



Integrated Cloud Applications & Platform Services

Oracle Linux System Administration I

Student Guide - Volume II

D103151GC10

Edition 1.0 | February 2019 | D106142



Learn more from Oracle University at education.oracle.com

ORACLE®

Author

Eric Craig

**Technical Contributors
and Reviewers**

Michael O'Reilly

Antoinette O'Sullivan

Craig McBride

Michael O'Reilly

Dave Goff

Steve Miller

Rowan Puttergill

Nita Heieck

Craig Carl

Wim Coekaerts

Sergio Leunissen

Keshav Sharma

Hanlin Chien

Jim Williams

Ankur Kemkar

Jared Greenwald

Lawrence Gabriel

Tim Caster

Steven B. Nelson

Harish Niddagatta

Sebastien Colas

Matt Slingsby

Al Flournoy

Jamie Iles

Simon Coter

Avi Miller

John Haxby

Saar Maoz

Todd Vierling

Sreejith Mohan

Paul Kasewurm

Corey Leong

Graphic Designer

Yogita Chawdhary

Editors

Raj Kumar

Moushmi Mukherjee

Copyright © 2019, 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.

Publishers

Giri Venugopal

Sumesh Koshy

Raghunath M

Contents

1 Course Introduction

- Course Objectives 1-2
- Course Schedule 1-3
- Objectives 1-5
- Virtualization with Oracle VM Server for x86 1-6
- Oracle VM Server for x86 in the Classroom 1-7
- Working with Classroom Virtual Machines 1-8
- Summary 1-9
- Practice 1: Overview 1-10

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
- Oracle Linux Release/Update Targets Relative to Red Hat Enterprise Linux (RHEL) Releases/Updates 2-18
- The Unbreakable Enterprise Kernel (UEK) 2-19
- DTrace 2-20
- Btrfs File System 2-21
- Oracle Linux Release Notes 2-22
- Quiz 2-24
- Summary 2-26
- Practice 2: Overview 2-27

3 Oracle Cloud Computing

- Objectives 3-2
- What Is Infrastructure as a Service (IaaS)? 3-3
- What Is Oracle Private Cloud Appliance? 3-4

What Is Oracle OpenStack? 3-5
What Are Oracle Cloud Infrastructure Services? 3-6
Key Concepts and Terms 3-8
Oracle-Provided Images 3-12
Oracle-Provided Bare Metal Shapes 3-13
Oracle-Provided VM Shapes 3-14
Key Concepts and Terms 3-15
Task Flow to Launch an Oracle Cloud Infrastructure Instance 3-16
Setting Up a Virtual Cloud Network (VCN) 3-18
Setting Up VCN Subnets 3-19
Viewing Instance Details and Accessing Your Instance 3-20
OCI Utilities / oci-utils 3-22
Creating a Block Volume 3-23
Attaching a Block Volume to an Instance 3-24
Connecting a Block Volume to an Instance's Guest OS 3-25
Oracle Cloud Computing Resources 3-26
Quiz 3-27
Summary 3-29
Practice 3: Overview 3-30
4 Installing Oracle Linux 7
Objectives 4-2
Obtaining Oracle Linux 4-3
Oracle Software Delivery Cloud 4-4
Anaconda Installer 4-5
Oracle Linux Installation Menu 4-6
Boot Options 4-7
Welcome to Oracle Linux 4-8
Installation Summary 4-9
Date and Time Configuration 4-10
NTP Configuration 4-11
Keyboard Layout 4-12
Language Support 4-13
Software Installation Source 4-14
Selecting the Software to Install 4-15
Installation Destination 4-16
Automatic Partitioning 4-17
Manual Partitioning 4-18
Summary of Partitioning 4-19
Network and Host Name Configuration 4-20
Network Connection Settings 4-21

Security Policy 4-22
KDUMP 4-23
Completing the Installation 4-24
Setting the root Password 4-25
Creating an Initial User 4-26
Installation Complete 4-27
Initial Setup 4-28
GUI Login Window 4-29
Inplace Upgrade from Oracle Linux 6 4-30
Launching an Oracle Cloud Infrastructure Instance 4-31
Quiz 4-32
Summary 4-34
Practice 4: Overview 4-35

5 Oracle Linux 7 Boot Process

Objectives 5-2
The Oracle Linux 7 Boot Process: BIOS Mode 5-3
The Initial RAM File System 5-5
Master Boot Record (MBR) 5-8
GRUB 2 Bootloader 5-9
The /etc/default/grub File 5-10
Unified Extensible Firmware Interface (UEFI) Overview 5-12
Globally Unique Identifier (GUID) Partition Table: GPT 5-13
GPT Layout 5-14
Installation and Administration with UEFI Using a Local Disk 5-15
Installation with UEFI: EFI System Partition (ESP) 5-16
The efibootmgr Utility 5-17
The Oracle Linux 7 Boot Process: UEFI Mode Using Local Disk 5-18
Secure Boot with UEFI 5-19
Kernel Boot Parameters 5-20
GRUB 2 Configuration File 5-21
GRUB 2 Menu 5-22
Editing a GRUB 2 Menu Option 5-23
GRUB 2 Command Line 5-24
systemd Introduction 5-25
systemd Features 5-27
systemd Service Units 5-28
Displaying the Status of Services 5-29
Starting and Stopping Services 5-31
Enabling and Disabling Services 5-32
systemd Target Units 5-33

System-State and Equivalent Run Level Targets	5-34
Working with Target Units	5-35
Rescue Mode and Emergency Mode	5-37
Shutting Down, Suspending, or Rebooting Commands	5-38
Quiz	5-39
Summary	5-43
Practice 5: Overview	5-44

6 System Configuration

Objectives	6-2
Configuring System Date and Time During Installation	6-3
Configuring System Date and Time from the Command Line	6-4
Using the timedatectl Utility	6-6
Using Network Time Protocol	6-8
Configuring NTP by Using Chrony	6-10
The /etc/sysconfig Directory	6-12
The proc File System	6-13
Top-Level Files in /proc	6-15
Process Directories in /proc	6-17
Other Directories in /proc	6-18
The /proc/sys Directory	6-19
The sysctl Utility	6-20
The sysfs File System	6-22
The lshw Utility	6-24
The lshw-gui Utility: Graphical Hardware Configuration Interface	6-25
Quiz	6-26
Summary	6-30
Practice 6: Overview	6-31

7 Package Management

Objectives	7-2
Package Management: Introduction	7-3
The rpm Utility	7-4
Oracle Linux Yum Server	7-6
The yum Configuration	7-7
Modular yum Repository Configurations	7-11
Systems With Modular yum Repository Configurations	7-14
Modular yum Repository Configurations: Legacy Systems	7-15
The yum Utility	7-16
yum Groups	7-18
Unbreakable Linux Network (ULN)	7-19

ULN Channels	7-20
Switching from RHN to ULN	7-22
Registering Your System with ULN	7-24
Replacement of SSL Certificates for ULN	7-25
The Software Collection Library for Oracle Linux	7-28
Using Software Collections with Oracle Linux 7	7-29
Installing Software Collections and Running Commands	7-30
Creating a Shell Environment and Viewing Optional Packages	7-31
Quiz	7-32
Summary	7-35
Practice 7: Overview	7-36

8 Automating Tasks

Objectives	8-2
Automating System Tasks	8-3
Configuring cron Jobs	8-4
Other cron Directories and Files	8-6
The crontab Utility	8-8
Configuring anacron Jobs	8-9
Checking cron Job Status	8-11
at and batch	8-12
Quiz	8-14
Summary	8-15
Practice 8: Overview	8-16

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 Oracle Ksplice

Objectives	10-2
Oracle Ksplice: Introduction	10-3
Ksplice Addresses System Administrator Challenges	10-4
Ksplice: Features and Benefits	10-5
Ksplice Patching	10-6
Ksplice for Diagnosing Issues	10-7

Ksplice Track Record	10-8
Ksplice Case Study	10-9
How Ksplice Works	10-10
Online Ksplice Implementations	10-11
Offline Ksplice Implementations	10-12
Implementing the Ksplice Online Standard Client	10-13
Subscribing to the Ksplice Channel	10-14
Ksplice Packages on ULN	10-15
Using Ksplice Uptrack	10-16
Ksplice Uptrack Command Summary	10-17
Ksplice in Oracle Cloud Infrastructure	10-18
Installing uptrack for Ksplice Without ULN Registration	10-19
Ksplice Web Interface: System Status	10-22
Ksplice Web Interface: System Status Detail	10-23
The Ksplice Offline Standard Client	10-24
Modifying a Local Yum Server to Act as a Ksplice Mirror	10-25
Updating a Local Yum Server with Ksplice Channels	10-26
Configuring Ksplice Offline Standard Clients to Use the Local Ksplice Mirror	10-27
The Ksplice Enhanced Client	10-28
Implementing the Ksplice Online Enhanced Client	10-29
Managing the Ksplice Enhanced Client	10-30
Configuring Ksplice Offline Enhanced Clients to Use a Local Ksplice Mirror	10-31
Quiz	10-33
Summary	10-35
Practice 10: Overview	10-36

11 User and Group Administration

Objectives	11-2
Introduction to Users and Groups	11-3
User and Group Configuration Files	11-4
Adding a User Account	11-6
Modifying or Deleting User Accounts	11-9
Group Account Administration	11-10
User Private Groups	11-12
Password Configuration	11-14
The /etc/login.defs File	11-16
The User Manager Tool	11-17
Restricting Use of the su Command	11-18
Allowing Use of the sudo Command	11-19
User/Group Administration in Oracle Cloud Infrastructure	11-21
User/Group Administration in the Enterprise	11-22

Quiz 11-23

Summary 11-25

Practice 11: Overview 11-26

12 Partitions, File Systems, and Swap

Objectives 12-2

Disk Partitions 12-3

Partition Layout: Example 12-4

Partition Table Manipulation Utilities 12-5

The fdisk Utility 12-6

Using the fdisk Utility 12-8

The cfdisk Utility 12-10

The parted Utility 12-11

File System Types 12-13

Making File Systems 12-15

Mounting File Systems 12-17

Block Volumes in Oracle Cloud Infrastructure 12-20

The /etc/fstab File 12-21

Maintaining File Systems 12-22

Access Control Lists (ACLs) 12-24

getfacl and setfacl Utilities 12-25

Swap Space 12-27

Quiz 12-29

Summary 12-31

Practice 12: Overview 12-32

13 Network Configuration

Objectives 13-2

Network Interface File Names 13-3

Network Interface File Parameters 13-5

Additional Network Configuration Files 13-7

Starting the Network Service 13-9

The ethtool Utility 13-10

NetworkManager 13-11

Network Settings 13-12

View or Edit an Existing Network Connection 13-13

Network Connections Editor 13-14

The nmcli Utility 13-15

The nmcli general Object 13-16

The nmcli networking Object 13-18

The nmcli radio Object 13-20

The nmcli connection Object 13-21
The nmcli connection show Command 13-22
nmcli connection up|down Commands 13-23
The nmcli connection add Command 13-25
The nmcli connection edit Command 13-27
The nmcli connection modify Command 13-29
nmcli connection delete | reload | load Commands 13-30
The nmcli device Object 13-31
The nmtui Utility 13-33
The ip Utility 13-34
The ip addr Object 13-36
The ip link Object 13-38
Address Resolution Protocol (ARP) 13-40
The ip route Object 13-42
Networking in Oracle Cloud Infrastructure 13-44
Quiz 13-46
Summary 13-49
Practice 13: Overview 13-50

14 IPv6

Objectives 14-2
IPv6: Overview 14-3
IPv6: Features 14-4
IPv6 Addresses 14-5
IPv6 Address Representation 14-6
IPv6 Address Types 14-7
High-Order Bits Specified for Address Types 14-8
Prefix Notation 14-9
Unicast Addressing 14-10
Link-Local Unicast Address Layout 14-11
Global Unicast Address Layout 14-12
Multicast Addressing 14-13
Multicast Address Layout 14-14
Anycast Addressing 14-15
Neighbor Discovery Protocol (NDP) 14-16
Obtaining IPv6 Addresses 14-17
Stateless Address Autoconfiguration—SLAAC 14-18
The SLAAC Process 14-19
Interface ID—Modified EUI-64 Format Created from MAC Addresses 14-21
Interface Identifier Creation—Modified EUI-64 Format 14-22
Stateless DHCPv6 Autoconfiguration 14-23

Stateful DHCPv6 Autoconfiguration	14-24
Other Address Types	14-25
Using IPv6 with Oracle Linux	14-26
IPv6 Settings for Specific Interfaces	14-27
Some IPv6 Commands	14-28
Quiz	14-29
Summary	14-32
Practices for Lesson 14: Overview	14-33

15 OpenSSH

Objectives	15-2
OpenSSH: Introduction	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
Key Pairs for Oracle Cloud Infrastructure Instances	15-14
Using ssh-agent	15-15
Quiz	15-16
Summary	15-17
Practices for Lesson 15: Overview	15-18

16 Security Administration

Objectives	16-2
chroot Jail: Overview	16-3
chroot Utility	16-4
Implementing a chroot Jail	16-5
Running Services in a chroot Jail	16-7
Packet-Filtering Firewalls: Introduction	16-9
firewalld: Introduction	16-10
firewalld Zones	16-11
Predefined firewalld Zones	16-12
Setting the Default firewalld Zone	16-14
firewalld Services	16-15
Starting firewalld	16-17
firewalld Configuration Tool	16-18
firewall-cmd Utility	16-19
iptables: Introduction	16-21

iptables Terminology	16-22
Beginning iptables Maintenance	16-24
Adding a Rule by Using the iptables Utility	16-26
iptables Rule Specs	16-28
More iptables Options	16-29
NAT Table	16-30
TCP Wrappers	16-32
TCP Wrappers Configuration	16-33
TCP Wrapper Command Options	16-35
Security Control in Oracle Cloud Infrastructure	16-37
Quiz	16-38
Summary	16-40
Practices for Lesson 16: Overview	16-41

17 Oracle on Oracle

Objectives	17-2
Oracle Pre-Install RPM	17-3
Oracle Software User Accounts	17-5
Oracle Software Group Accounts	17-6
Oracle Automatic Storage Management (ASM) Groups	17-8
System Resource Tuning	17-9
Linux Shared Memory	17-10
Semaphores	17-11
Network Tuning	17-13
Setting the File Handles Parameter	17-14
Asynchronous I/O (AIO)	17-15
Oracle-Related Shell Limits	17-16
HugePages	17-18
Configuring HugePages	17-20
Oracle Database Smart Flash Cache (DBSFC)	17-22
Oracle ASM	17-23
ASM Library Driver (ASMLib)	17-26
Using ASMLib Commands	17-28
ASM Filter Driver (ASMF D)	17-29
Quiz	17-30
Summary	17-31
Practices for Lesson 17: Overview	17-32

18 System Monitoring and Management

Objectives	18-2
sosreport Utility	18-3

iostat Utility	18-5
mpstat Utility	18-7
vmstat Utility	18-9
sar Utility	18-11
top Utility	18-13
iotop Utility	18-15
strace Utility	18-16
netstat Utility	18-17
netstat Command Alternatives	18-19
tcpdump Utility	18-21
Wireshark	18-23
OSWatcher (OSWbb)	18-24
OSWbb Diagnostic Data Output	18-27
OSWatcher Analyzer (OSWbba)	18-31
Analyzing OSWbb Archive Files	18-34
Oracle Enterprise Manager Cloud Control	18-36
Oracle Enterprise Manager Cloud Control Components	18-37
Oracle Enterprise Manager Cloud Control GUI Sections	18-39
Oracle Enterprise Manager Cloud Control GUI—Enterprise Summary	18-40
Oracle Enterprise Manager Cloud Control GUI—Oracle Linux Home	18-41
Oracle Linux Home Page	18-42
Spacewalk: Overview	18-43
Spacewalk: Features and Functionality	18-44
Spacewalk Web Interface - Overview Page	18-46
Quiz	18-47
Summary	18-49
Practices for Lesson 18: Overview	18-50

19 System Logging

Objectives	19-2
System Logging: Introduction	19-3
rsyslog Configuration	19-4
rsyslog Filter Options	19-6
Facility/Priority-Based Filters	19-7
rsyslog Actions	19-9
rsyslog Templates	19-11
Configuring Log Rotation (logrotate)	19-13
The logwatch Utility	19-15
journald: Introduction	19-16
The journalctl Utility	19-17
journald Metadata	19-19

Process Accounting	19-20
Process Accounting Utilities	19-21
The accton Utility - Turn Process Accounting On/Off	19-22
Quiz	19-23
Summary	19-25
Practices for Lesson 19: Overview	19-26

20 Troubleshooting

Objectives	20-2
Two-Phased Approach to Troubleshooting	20-3
Gathering Information	20-4
Operating System Logs	20-5
dmesg Utility	20-6
Troubleshooting Resources	20-7
My Oracle Support	20-8
Causes of Common Problems	20-9
Troubleshooting Boot Problems	20-11
Typical Causes of NFS Problems	20-12
Quiz	20-13
Summary	20-14
Practices for Lesson 20: Overview	20-15

User and Group Administration

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain 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 administration in Oracle Cloud Infrastructure
- Describe user and group implementation in the enterprise



ORACLE

Copyright © 2019, 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



ORACLE

Copyright © 2019, 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 by 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 © 2019, 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:1000:1000:Oracle Student:/home/oracle:/bin/bash
```

The following describes this entry:

- **oracle**: Username
- **x**: Indicates that shadow passwords are used
- **1000**: UID, these begin with 1000 and increment by 1 for each newly added user. UIDs below 1000 are reserved for system use.
- **1000**: GID of the user's primary group. These begin with 1000 and increment by 1 for each new group. Users can belong to more than one group.
- **Oracle 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:

```
# grep oracle /etc/shadow
oracle:$6$zGK...:17429:0:99999:7:::
```

The following describes this entry:

- **oracle:** Username
- **\$6\$zGK...:** 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.
- **17429:** Number of days since the 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 the 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 was 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:

```
# grep oracle /etc/group
oracle:x:1000:oracle
```

The following describes this entry:

- **oracle:** Group name
- **x:** Indicates that shadow passwords are used
- **1000:** GID
- **oracle:** List of users that are members of the group

Each group can have multiple users. Users can also belong to more than one group. The GID stored in the user's entry in /etc/passwd is the user's primary group.

/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:

```
# grep oracle /etc/gshadow
oracle:!!:oracle
```

The following describes this entry:

- **oracle:** Group name
- **!!:** Hashed password. The !! indicates that the account is locked.
- **oracle:** List of users that are members of the group

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
```

ORACLE

Copyright © 2019, 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:1001:1001::/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 that 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 by 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 white space.
- `-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," 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 the purposes of running a service such as SMTP, FTP, or running a web server. A user without a login shell cannot log in to a system and, therefore, cannot run any commands interactively on the system. Processes, however, can run as that user.

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 `/etc/nologin.txt` file 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=1017):

```
# usermod -aG 1017 user_name
```

- Use the `userdel` command to delete a user:

```
# userdel [options] user_name
```

- Options to `userdel` include:

- `-f`: Force removal even if the user is logged in.
- `-r`: Remove the user's home directory.



Copyright© 2019, 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 1017 /etc/group
students:x:1017:
# usermod -aG 1017 mary
# grep 1017 /etc/group
students:x:1017: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 © 2019, 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 the `oracle` user belongs to two groups, `oracle` (primary group) and `students` (secondary group):

```
$ grep oracle /etc/passwd
oracle:x:1000:1000:Oracle Student:/home/oracle/bin/bash
$ grep oracle /etc/group
oracle:x:1000:
students:x:1056:student1,student2,oracle
```

The **groups** command (logged in 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=1000(oracle) gid=1000(oracle)
groups=1000(oracle),1066(students)...
```

Note that the `gid` equals `1000(oracle)`.

```
$ ps
 PID TTY      TIME CMD
20279 pts/0  00:00:00 bash
20411 pts/0  00:00:00 ps
$ newgrp students
$ id
uid=1000(oracle) gid=1066(students)
groups=1000(oracle),1066(students)...
```

Note that the `gid` now equals `1066(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 the 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.



ORACLE®

Copyright © 2019, 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 prevented 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 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, and set the setgid bit for the *project* directory, all *project* users are able to edit the *project* files and create new files in the *project* directory. Any files these users create retain their *project* group status. Other *project* users can always 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 © 2019, 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 user `frank`'s entry before making any changes:

```
# grep frank /etc/shadow
frank:$6$XB1Um6w...:15284:0:99999:7:::
```

If the minimum password age is set to 14, the user cannot change their password for 14 days. The maximum password set to 30 means the user has 16 days remaining before having to change his password. These changes would appear 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

Several options are available for the `chage` command.

To list aging information:

```
# chage -l frank
Last password change : Sep 24, 2014
Password expires      : Oct 24, 2014
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
```

The /etc/login.defs File

- The /etc/login.defs file provides default user account settings.
- Default values include:
 - The location of user mailboxes
 - Password aging controls
 - Values for automatic UID selection
 - Values for automatic GID selection
 - User home directory creation options
 - The umask value
 - The encryption method used to encrypt passwords



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

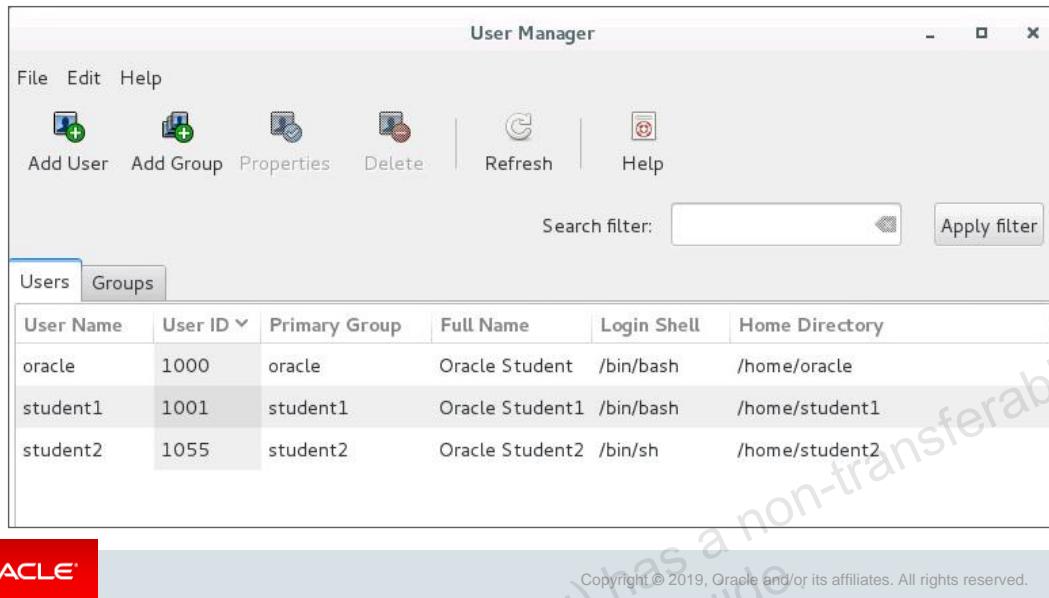
The /etc/login.defs file provides default configuration information for several user account parameters. The useradd, usermod, userdel, and groupadd commands, and other user and group utilities take default values from this file. Each line consists of a directive name and associated value. The following is a partial list of /etc/login.defs directives:

- Location of user mailboxes
- Password aging controls
- Minimum and maximum values for automatic UID selection (1000 to 60000)
- Minimum and maximum values for automatic GID selection (1000 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 sets 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.

The User Manager Tool

The `system-config-users` command starts User Manager.



This slide shows the graphical tool used to perform user and group account administration. 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. This package is not installed by default. Install it with yum as needed:

```
# yum install system-config-users
```

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 `su` command access 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
```



Copyright © 2019, 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 `su` command access to the `oracle` user, add the `oracle` user to the `wheel` group as follows:

```
# usermod -aG wheel oracle
```

Pluggable Authentication Modules (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 `root` access only 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 them 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.
- By default, commands specified with sudo are run as the root user.

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

You can also allow a regular user to run a command as root by preceding the 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 first "ALL" parameter gives the root user permission to run commands from any host. The second "ALL" parameter (in parentheses) permits the root user to execute commands as any user. The ending "ALL" parameter indicates that the root user can execute any command.

The following default entry in the /etc/sudoers file allows any user in the wheel group to run any command from any host as any user:

```
## Allows people in group wheel to run all commands  
%wheel  ALL=(ALL)          ALL
```

With a user added to the wheel group, this user can run any command, preceding them with sudo. The user does not need to be added to the /etc/sudoers file in this case.

When a user is added to the wheel group, you must log out and log back in to make new wheel membership known in the current terminal session.

If this %wheel entry in /etc/sudoers is commented out, a user must be added to /etc/sudoers in order to run commands using sudo, whether the user is a member of the wheel group or not.

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 from any host as any user (`root` is the default), create the following entry in the `/etc/sudoers` file:

```
# visudo  
oracle ALL=(ALL) ALL
```

For the `oracle` user to be authorized to run commands with `sudo`, an entry for the `oracle` user in the `/etc/sudoers` file is necessary if the `oracle` user is not a member of the `wheel` group. It is also necessary if the `oracle` user is a member of the `wheel` group and this `%wheel` entry is commented out in `/etc/sudoers`:

```
## Allows people in group wheel to run all commands  
# %wheel  ALL=(ALL)          ALL
```

The `oracle` user can then run administrative commands by preceding them with `sudo`.

You can also restrict use of `sudo` to specific commands. For example, to allow the `oracle` user only to add and delete users, create the following entry in the `/etc/sudoers` file:

```
oracle ALL=(ALL) /usr/sbin/useradd, /usr/sbin/userdel
```

This allows the user `oracle` to run the `/usr/sbin/useradd` and `/usr/sbin/userdel` commands from any host as any user. The absolute path to specific commands must be provided.

To allow the `oracle` user only to run the `systemctl` command from any host as any user, create the following entry in the `/etc/sudoers` file:

```
oracle ALL=(ALL) /usr/bin/systemctl
```

See the `sudoers` (5) man page for details.

User/Group Administration in Oracle Cloud Infrastructure

- `opc` is the default user for Oracle Linux image instances.
 - Created when launching an instance
- `opc` is added to the `wheel` group automatically.
- `root` privileges are obtained entirely via `sudo`.
- To obtain sustained `root` privileges:

```
[opc@instance1 ~]$ sudo bash  
[root@instance1 opc]#
```

- `root` privileges are available until `exit` is issued:

```
[root@instance1 opc]# exit  
exit  
[opc@instance1 ~]$
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

User/Group Administration in the Enterprise

- User and group account information is often centralized.
- Centralized information can be retrieved by 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 © 2019, 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.

Lightweight Directory Access Protocol (LDAP)

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



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a

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.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, d

Password aging information is stored in the /etc/shadow file

Summary

In this lesson, you should have learned how to:

- Explain 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 administration in Oracle Cloud Infrastructure
- Describe user and group implementation in the enterprise



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practice 11: Overview

This practice covers the following topics:

- Administering user accounts
- Administering group accounts
- Implementing user private groups
- Configuring password aging
- Using the User Manager GUI
- Restricting the use of the `su` command
- Allowing the use of the `sudo` command



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Partitions, File Systems, and Swap

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe disk partitioning
- Use disk partitioning utilities
- List supported file system types
- Explain file system creation, mounting, and maintenance
- Describe block volumes in Oracle Cloud Infrastructure
- Describe access control lists (ACLs)
- Describe and maintain swap space



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

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.



ORACLE®

Copyright © 2019, 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.

Partition Layout: Example

- Oracle Linux virtual machine is installed and running under Oracle VM.
- Three primary partitions are created during installation:
 - xvda1, mounted on /boot
 - xvda2, mounted on /
 - xvda3, mounted on /home
- A fourth partition, xvda4, is created as an extended partition.
 - Extended partitions can be subdivided into logical partitions.
- One logical partition, xvda5, is created on the extended partition.
 - This logical partition is designated as a swap partition.
- Partition devices are listed in the /proc/partitions file:

```
# cat /proc/partitions
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The slide shows an example of a partition layout for an Oracle Linux virtual machine installed and running under Oracle VM. The system was installed with three primary partitions: xvda1, xvda2, and xvda3. A 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.

Partition devices are listed in the /proc/partitions file:

```
# cat /proc/partitions
major minor #blocks name
 202      0   16777216 xvda
 202      1    1048576 xvda1
...
...
```

The columns are described as follows:

- **major:** The major number of the device. This corresponds with the block device in the /proc/devices file.
- **minor:** The minor number of the device. This relates to the number at the end of the partition name.
- **#blocks:** The number of physical disk blocks contained in the partition
- **name:** The name of the partition

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 © 2019, 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. Creating a new partition on a device that is in use is not recommended. 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.

The 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 sectors
 - **Blocks:** The number of blocks allocated to the partition
 - **Id and System:** The partition type
- Partition naming (example: `/dev/xvda1`)
 - `/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



Copyright © 2019, 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: 17.2 GB, 17179869184 bytes, 33554432 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000e9780
```

Device	Boot	Start	End	Blocks	Id	System
<code>/dev/xvda1</code>	*	2048	2099199	1048576	83	Linux
<code>/dev/xvda2</code>		2099200	23070719	10485760	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 17.2 GB virtual disk (`xvda`) with 33554432 sectors. With zone density recording (ZDR), a disk surface is divided into 16 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 sectors 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.

```
# fdisk <device_name>
Command (m for help):
```

- 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:

```
# partprobe <device_name>
```

ORACLE

Copyright © 2019, 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 several different partition table types. The supported partition types are displayed when viewing the fdisk menu.

To use the interactive interface, enter the fdisk command followed by the device name:

```
# fdisk /dev/xvdb
```

Informational messages are displayed:

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.

Be careful before using the write command.

The following message is displayed if the device does not contain a recognized partition table:

Device does not contain a recognized partition table

Building a new DOS disklabel with disk identifier ...

The fdisk command prompt then appears:

```
Command (m for help):
```

Enter m to display the fdisk commands.

fdisk Commands

```
Command (m for help): m
Command action
  a    toggle a bootable flag
  b    edit bsd disklabel
  c    toggle the dos compatibility flag
  d    delete a partition
  g    create a new empty GPT partition table
  G    create an IRIX (SGI) partition table
  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 create a new 5 GB primary partition:

```
Command (m for help): n
Partition type:
  p      primary partition (0 primary, 0 extended, 4 free)
  e      extended
Select (default p): ENTER
Using default response p
Partition number (1-4, default): ENTER
First sector (2048-10485759, default 2048): ENTER
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-10485759, default
10485759): +5GB
Partition 1 of type Linux and of size 4.7 GiB is set
```

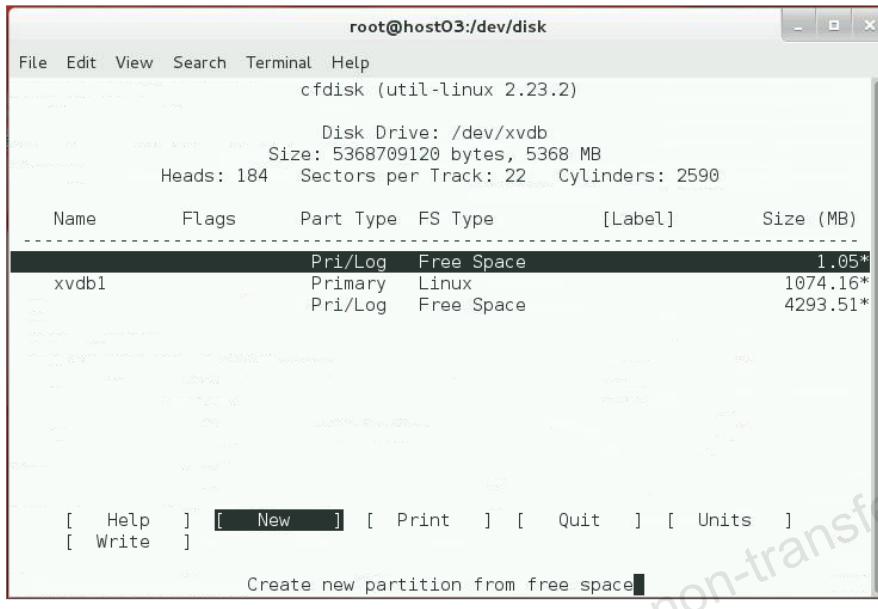
The fdisk command prompt is displayed again. Enter **w** to write the table to disk and exit.

The 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/xvdb
```

The cfdisk Utility



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The 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/xvdb
```

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.

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 example in the slide, one primary partition exists, `xvdb1`. The free space is selected in the upper portion of the window and the `[New]` menu option is selected at the bottom of the window. Press Enter to create a new partition from the free space.

The parted Utility

- The `parted` utility provides a command-line interface:

```
# parted [option] <device_name> [command [argument]]
```

- The `parted` utility also has an interactive mode:

```
# parted <device_name>
(parted)
```

- 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.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The GNU `parted` utility is also used to view the partition table, resize partitions, or create new partitions. 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 3.1
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
```

The following example creates a new partition table by using the `mklabel` command:

```
(parted) mklabel gpt
```

The disk label type must be one of the following: aix, amiga, bsd, dvh, gpt, mac, msdos, pc98, sun, or loop.

The following example uses the `mkpart` command to create a new partition:

```
(parted) mkpart
Partition name? []
File system type? [ext2]?
Start? 0
End? 20%
(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
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 5369MB
Sector size (logical/physical): 512/512
Partition Table: gpt
Number  Start   End     Size    File system  Name  Flags
 1      17.4kB  1074MB
```

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
- **XFS:** High-performance journaling file system
- **Btrfs:** Addresses scalability requirements of large storage systems



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

After partitioning the disk device, you need to make file systems on the partitions. Making a file system writes information to the device and creates an ordered structure from the empty partition 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 several 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 can 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.

XFS

The XFS file system is a high-performance journaling file system, supporting large file system sizes. You can create an XFS file system on a regular disk partition and on a logical volume. With Oracle Linux 7, the default file system type is XFS. XFS is discussed further in the lesson titled “Implementing the XFS File System.”

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. It provides extent-based file storage supporting large file system sizes. Btrfs offers the ability to create both readable and writable snapshots and the capability to roll back to a prior, known good state. It 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.

See the following Oracle guidelines for file and file system limits for ext4, XFS, and Btrfs:

https://docs.oracle.com/cd/E52668_01/E88149/html/section-shl-nvd-kn.html

See the following URL for a comparison between XFS and ext4 and Btrfs:

<https://lwn.net/Articles/476263/>.

The following URL discusses all the major file systems:

http://en.wikipedia.org/wiki/Comparison_of_file_systems.

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`
- Default 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 © 2019, 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/mkdirct      /sbin/mkfs          /sbin/mkfs.ext3      /sbin/mkfs.msdos
/sbin/mklost+found
/sbin/mkdosfs     /sbin/mkfs.btrfs    /sbin/mkfs.ext4      /sbin/mkfs.vfat
/sbin/mkswap
/sbin/mkdumprd   /sbin/mkfs.cramfs  /sbin/mkfs.fat       /sbin/mkfs.xfs
/sbin/mke2fs      /sbin/mkfs.ext2     /sbin/mkfs.minix
/sbin/mkhomedir_helper
```

Not all of these `mk*` files are used to create file systems. For example, the `mkhomedir_helper` command is a helper utility that creates home directories. The `mkdosfs`, `mkfs.msdos`, and `mkfs.vfat` files are symbolic links to `mkfs.fat`.

Using `mkfs`

The default file system type created when using the `mkfs` command is ext2. As previously mentioned, `mkfs` is 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

Several 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, enter:

```
# 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, assign a label name, and display information, enter:

```
# mkfs -t ext4 -L ProjectA /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: LABEL="ProjectA" UUID="f6f32f..." TYPE="ext4"
PARTUUID="..."
```

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 nosuper,ro /dev/xvdd1 /test
```

- Use the `umount` command to unmount file systems:

```
# umount /dev/xvdd1
```



Copyright © 2019, 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
```

To explicitly specify a file system type with the `mount` command, use the `-t` option:

```
# mount -t ext4 /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: LABEL="ProjectA" UUID="9d7ab..." TYPE="ext4"  
# mount LABEL="ProjectA" /test  
# mount UUID="9d7ab..." /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 with the `-a` option
- `loop`: Mounts a file and makes it accessible as a block device
- `noauto`: Does not allow the use of the `-a` option
- `noexec`: Does not allow the execution of binary files
- `nouser`: Does not allow a nonroot user to mount and unmount the file system
- `remount`: Remounts the mounted file system
- `ro`: Mounts the file system for read-only
- `rw`: Mounts the file system for both read/write
- `user`: Allows a nonroot user 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 exists), enter:

```
# mount -o ro,loop OracleLinux-R7-U5-Server-x86_64-dvd.iso
/media/cdrom
```

Journaling Mount Options

The `ext3` and `ext4` file systems have three journaling levels that can 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 can 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
```

Block Volumes in Oracle Cloud Infrastructure

- Dynamically provisioned
- iSCSI or paravirtualized attachment types
 - iSCSI - Bare metal or VM
 - Paravirtualized - VM
- Create, attach, connect, and move volumes
- Created with the Create Block Volume function within the Oracle Cloud Infrastructure console
- Attached from the instance details page
- If iSCSI attachment type used, volume is connected to an instance with `iscsiadm` commands
- Volume size affects block volume performance



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Oracle Cloud Infrastructure Block Volume Service lets you dynamically provision and manage block storage volumes. The attachment type can be paravirtualized or iSCSI. iSCSI attachments are required for bare metal instances.

You can create, attach, connect, and move volumes as needed to meet your storage and application requirements. When attached and connected to an instance, you can use a volume like a regular hard drive. Volumes can also be disconnected and attached to another instance without the loss of data.

The performance of iSCSI-attached Block Volume storage varies with volume size. Larger sizes generally reduce performance. With larger volumes, iSCSI attachments provide better performance than paravirtualized attachments due to the virtualization overhead associated with them.

When creating a Block Volume, you choose a compartment to hold your block volume resource and provide a user-friendly name or description. The Block Volume must be in the same Availability Domain as the instance. You choose the size of the block volume and assign any tags desired for administration.

You attach a Block Volume to an instance by navigating to the instance details in the Oracle Cloud Infrastructure console and selecting the Attached Block Volumes resource.

To set up an iSCSI connection, after the volume is attached, you log on to the instance and use `iscsiadm` tool commands to configure the iSCSI connection. These commands are provided for you from the block volume details page and include the iSCSI target IP address and port, and volume IQN.

Copy the provided commands and paste them into an SSH session with the instance to configure the iSCSI connection. After you configure the iSCSI connection, you can partition, format, mount and use the volume like a normal hard drive.

The /etc/fstab File

- The /etc/fstab file is the file system mount table. 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 © 2019, 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
UUID=...      /      ext4  defaults  1  1
UUID=...      /boot  ext4  defaults  1  2
UUID=...      /home  ext4  defaults  1  2
UUID=...      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.

To mount entries in fstab, use:

```
# mount -a
```

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.



ORACLE

Copyright © 2019, 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 the `fsck` Command

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 from util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
Test: clean, 11/65536 files, 8859/262139 blocks
```

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/btrfsck  /sbin/fsck          /sbin/fsck.ext2
/sbin/fsck.fat   /sbin/fsck.vfat
/sbin/dosfsck   /sbin/fsck.btrfs   /sbin/fsck.ext3
/sbin/fsck.minix /sbin/fsck.xfs
/sbin/e2fsck    /sbin/fsck.cramfs /sbin/fsck.ext4
/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 for `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 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 run a file system check on the ext2 file system before and after converting it to an ext3 file system. Use the following syntax to convert ext2 to ext3:

```
# tune2fs -j <ext2_file_system>
```

The `-j` option adds an ext3 journal to the file system.

The most commonly used options for `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
```

Access Control Lists (ACLs)

- An ACL provides a more fine-grained access control mechanism than traditional Linux access permissions.
- The file system containing the file or directory must be mounted with ACL support:

```
# mount -t ext3 -o acl /dev/xvdd1 /test
```

- Include the `acl` option in the `/etc/fstab` file:
 - `LABEL=/work /work ext3 acl 0 0`
- There are two types of ACL rules:
 - **access ACLs**: Specify access information for a single file or directory
 - **default ACLs**: Pertain to a directory only. The default access information for any file within the directory that does not have an access ACL.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Traditional Linux access permissions for files and directories consist of setting a combination of read, write, and execute permissions for the owner of the file or directory, a member of the group the file or directory is associated with, and everyone else (other). Access control lists (ACLs) provide a finer-grained access control mechanism than these traditional Linux access permissions.

Before using ACLs for a file or directory, install the `acl` package:

```
# yum install acl
```

The file system containing the file or directory must also be mounted with ACL support. The following is the syntax to mount a local ext3 file system with ACL support:

```
# mount -t ext3 -o acl <device-name> <mount-point>
```

If the partition is listed in the `/etc/fstab` file, include the `acl` option:

```
LABEL=/work /work ext3 acl 0 0
```

An ACL consists of a set of rules that specify how a user or group can access the file or directory the ACL is associated with. There are two types of ACL rules:

- **access ACLs**: Specify access information for a single file or directory
- **default ACLs**: Pertain to a directory only. It specifies default access information for any file within the directory that does not have an access ACL.

getfacl and setfacl Utilities

- Use the `getfacl` utility to display a file's ACL.
 - When a file does not have an ACL, `getfacl` displays the same information as `ls -l`, although in a different format.
- Use the `setfacl` utility to add or modify rules.
- The rules are in the following form:
 - `u:name:permissions`
 - `g:name:permissions`
 - `m:permissions`
 - `o:permissions`
- To add an ACL rule that gives the `oracle` user read and write permission to the `test` file:

```
# setfacl -m u:oracle:rwx test
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `getfacl` utility to display a file's ACL. When a file does not have an ACL, it displays the same information as `ls -l`, although in a different format. For example, the file `test` does not have an ACL:

```
# ls -l test
-rw-rw-r-- 1 oracle oracle 25 Mar 5 10:10 test
```

Sample `getfacl` output of the `test` file:

```
# getfacl test
# file: test
# owner: oracle
# group: oracle
user::rw-
group::rw-
other::r--
```

Use the `setfacl` utility to add or modify one or more rules in a file's ACL. The syntax is:

```
# setfacl -m <rules> <files>
```

The rules are in the following form:

- **u:name:permissions**: Sets the access ACL for a user (username or UID)
- **g:name:permissions**: Sets the access ACL for the group (group name or GID)
- **m:permissions**: Sets the effective rights mask. This is the union of all permissions of the owning group and all of the user and group entries.
- **o:permissions**: Sets the access ACL for everyone else (others)

The permissions are the traditional `r`, `w`, and `x` for read, write, and execute, respectively. The following example adds a rule to the ACL for the `test` file that gives the `oracle` user read and write permission to that file:

```
# setfacl -m u:oracle:rwx test
```

The output of `getfacl` includes the ACL rule:

```
# getfacl test
# file: test
# owner: oracle
# group: oracle
user::rw-
user:oracle:rwx
group::rw-
mask::rwx
other::r--
```

When a file has an ACL, `ls -l` displays a plus sign (+) following the permissions:

```
# ls -l test
-rw-rwrxr--+ 1 oracle oracle 25 Mar 5 10:10 test
```

Use the `-x` option without specifying any permissions to remove rules for a user or group. To remove the ACL itself, use the `-b` option:

```
# setfacl -x u:oracle test
# setfacl -b test
```

To set a default ACL, add `d:` before the rule and specify a directory instead of a file name:

```
# setfacl -m d:o:rx /share
```

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, respectively:

```
# swapon {device|file}
```

```
# swapoff {device|file}
```



Copyright © 2019, 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, 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/xvda5 17412096 25165823 3876864 82 Linux swap / ...
```

To view the swap partition (or file) in the /etc/fstab file, enter:

```
# grep swap /etc/fstab
UUID=... swap defaults 0 0
```

To display the contents of the /proc/swaps file, enter:

```
# cat /proc/swaps
Filename      Type      Size      Used   Priority
/dev/xvda5    partition 3876860  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, respectively. 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:

```
UUID=... swap defaults 0 0
/swapfile 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:        3876860 kB
SwapFree:         3876860 kB
```

To view swap usage by using the free command, enter:

```
# free |grep -i swap
Swap: 3876860 0 3876860
```

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



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz



Which of the following statements are 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.



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a,b,c,d

Summary

In this lesson, you should have learned how to:

- Describe disk partitioning
- Use disk partitioning utilities
- List supported file system types
- Explain file system creation, mounting, and maintenance
- Describe block volumes in Oracle Cloud Infrastructure
- Describe access control lists (ACLs)
- Describe and maintain swap space



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practice 12: Overview

This practice covers the following topics:

- Listing the current disk partitions
- Partitioning a storage device
- Creating ext3 and ext4 file systems
- Implementing access control lists
- Increasing swap space
- Removing partitions and additional swap space



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Network Configuration

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe network interface configuration files
- Describe additional network configuration files
- Start the network service
- Use the `ethtool` utility
- Describe NetworkManager and use the NetworkManager GUI
- Use the Network Connections editor
- Use the `nmcli` and `nmtui` utilities
- Describe ARP and the ARP cache
- Use the `ip` utility
- Describe networking in Oracle Cloud Infrastructure



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Network Interface File Names

- Each physical network device has an associated network interface configuration file.
- Network interface configuration files are located in the `/etc/sysconfig/network-scripts` directory.
- The naming scheme automatically assigns interface names that are predictable, for example:

```
# ls /etc/sysconfig/network-scripts/ifcfg*
ifcfg-enp134s1f0    ifcfg-enp134s1f1    ifcfg-lo
```

 - Names persist across system reboots, hardware reconfiguration, and kernel and device driver updates.
- The naming scheme can be changed by boot parameters:
 - `biosdevname=1`
 - `net.ifnames=0`



Copyright © 2019, 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.

In the example in the slide, there are two Ethernet interfaces, represented by `ifcfg-enp134s1f0` and `ifcfg-enp134s1f1`, and one loopback interface (`ifcfg-lo`). The system uses these files during the boot process to configure the network interfaces.

```
# ls /etc/sysconfig/network-scripts/ifcfg*
ifcfg-enp134s1f0    ifcfg-enp134s1f1    ifcfg-lo
```

In Oracle Linux 7, `systemd` and `udev` support different network interface name schemes. By default, the network interface name is derived from a device's firmware and location. In the example, the network interfaces are on a PCI card and are named “en” for Ethernet followed by `p<bus>s<slot>`, and `f<function_number>`. The following shows the names in `/sys`:

```
# ls -lR /sys | grep 134
lrwxrwxrwx... enp134s1f0 ->
/sys/devices/pci0000:80/0000:80:11.0/0000:86:01.0/net/enp134s1f0
lrwxrwxrwx... enp134s1f1 ->
/sys/devices/pci0000:80/0000:80:11.0/0000:86:01.1/net/enp134s1f1
```

This naming scheme automatically assigns interface names that are predictable and ensures that the names persist across system reboots, hardware reconfiguration, and updates to device drivers and the kernel.

By default, `systemd` assigns a two-character prefix based on the type of interface:

- **en**: Ethernet
- **wl**: Wireless LAN (WLAN)
- **ww**: Wireless wide area network (WWAN)

The prefix is followed by a suffix based on the hardware configuration, system bus configuration, or MAC address of the device, described as follows:

- **oN**: Onboard device with index number N . For example, `eno1`.
- **sS[fF] [dD]**: Hot-plug device with slot number S , optional function number F , and optional device ID D
- **xM**: Device with MAC address M
- **pBsS[fF] [dD]**: PCI device with bus number B , slot number S , optional function number F , and optional device ID D . Example:
 - `enp134s1f0`: Ethernet, bus number 134, slot number 1, function number 0
 - `enp134s1f1`: Ethernet, bus number 134, slot number 1, function number 1
- **pBsS[fF] [uP] [cC] [iI]**: USB device with bus number B , slot number S , optional function number F , optional port number P , optional configuration number C , and optional interface number I

The kernel assigns a legacy, unpredictable network interface name (`eth N` and `wlan N`) only if it cannot discover any information about the device that would allow it to differentiate the device from other such devices. You can use the `net.ifnames=0` boot parameter to reinstate the legacy naming scheme. The following gives the network interface names in `/sys` when using the `net.ifnames=0` boot parameter:

```
# ls -lR /sys | grep eth
lrwxrwxrwx... eth0 ->
/sys/devices/pci0000:80/0000:80:11.0/0000:86:01.0/net/eth0
lrwxrwxrwx... eth1 ->
/sys/devices/pci0000:80/0000:80:11.0/0000:86:01.1/net/eth1
```

The name of embedded network interfaces, PCI card network interfaces, and virtual function network interfaces can also be changed by the `biosdevname` udev helper utility. This feature requires that you install the `biosdevname` software package, and that you enable the `biosdevname` boot option as follows:

```
# yum install biosdevname
biosdevname=1
```

Note that using the `net.ifnames` or `biosdevname` boot parameters to change the naming scheme can render existing firewall rules invalid. This is discussed further in the lesson titled “Security Administration.” Changing the naming scheme can also affect other software that refers to legacy network interface names.

Network Interface File Parameters

Configuration parameters include:

- TYPE=Ethernet
- BOOTPROTO=none
- DEFROUTE=yes
- NAME=eth0
- ONBOOT=yes
- HWADDR=00:14:4F:8D:B0:BC
- IPADDR0=NN.NN.NN.NN
- PREFIX0=23
- GATEWAY0=NN.NN.NN.NN
- DNS1=NN.NN.NN.NN
- DNS2=NN.NN.NN.NN
- DOMAIN=example.com



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.



The system reads the network interface files during the boot process to determine which interfaces to bring up and how to configure them. The following is a sample:

```
# cat /etc/sysconfig/network-scripts/ifcfg-enp134s1f0
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=enp134s1f0
UUID=...
ONBOOT=yes
HWADDR=00:14:4F:8D:B0:BC
IPADDR0=NN.NN.NN.NN
PREFIX0=23
GATEWAY0=NN.NN.NN.NN
DNS1=NN.NN.NN.NN
```

```
DNS2=NN.NN.NN.NN  
DOMAIN=example.com  
IPV6_PEERDNS=yes  
IPV6_PEERROUTES=yes
```

A description of some of these configuration parameters is as follows:

TYPE=device_type: The type of network interface device

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).

DEFROUTE | IPV6_DEFROUTE=answer: Where *answer* is one of the following:

- **yes:** This interface is set as the default route for IPv4|IPv6 traffic.
- **no:** This interface is not set as the default route.

IPV6INIT=answer: Where *answer* is one of the following:

- **yes:** Enable IPv6 on this interface. If **IPV6INIT=yes**, the following parameters could also be set in this file:
 - **IPV6ADDR=**IPv6 address
 - **IPV6_DEFAULTGW=**The default route through the specified gateway
- **no:** Disable IPv6 on this interface.

IPV4_FAILURE_FATAL | IPV6_FAILURE_FATAL=answer: Where *answer* is one of the following:

- **yes:** This interface is disabled if IPv4 or IPv6 configuration fails.
- **no:** This interface is not disabled if configuration fails.

ONBOOT=answer: Where *answer* is one of the following:

- **yes:** This interface is activated at boot time.
- **no:** This interface is not activated at boot time.

HWADDR=MAC-address: The hardware address of the Ethernet device

IPADDRN=address: The IPv4 address assigned to the interface

PREFIXN=N: Length of the IPv4 netmask value

GATEWAYN=address: The IPv4 gateway address assigned to the interface. Because an interface can be associated with several combinations of IP address, network mask prefix length, and gateway address, these are numbered starting from 0.

DNSN=address: The address of the Domain Name Servers (DNS)

DOMAIN=DNS_search_domain: The DNS search domain

See the `nm-settings-ifcfg-rh(5)` man page (search on "ipv4 setting" and "ipv6 setting"), `/usr/share/doc/initscripts-*/sysconfig.txt` (search on "ifcfg" for parameters specific to interface configuration files), and

https://docs.oracle.com/cd/E52668_01/E54669/html/oI7-about-netconf.html for further parameter descriptions.

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 servers and search domain
- `/etc/sysconfig/network` specifies global information for all network interfaces.
- `/etc/nsswitch.conf` lists the order of host name searches.



ORACLE®

Copyright © 2019, 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 used. Larger networks would use Domain Name Service (DNS) 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 is shown as 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 example.com
nameserver NN.NN.NN.NN
nameserver NN.NN.NN.NN
nameserver NN.NN.NN.NN
```

/etc/sysconfig/network

This file specifies global network settings. For example, you can specify the default gateway in this file:

```
# cat /etc/sysconfig/network
GATEWAY=192.0.2.1
```

/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, and 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`, and then by querying the DNS server if the host name is not resolved:

```
# cat /etc/nsswitch.conf
...
hosts:      files dns
...
```

Starting the Network Service

- Use the `systemctl` command to start, stop, and restart the network service:

```
# systemctl restart network
```

- Interface control scripts in `/etc/sysconfig/network-scripts` can also be used.

- Use `ifup <interface_name>` to activate an interface:

```
# ifup enp134s1f0
```

- Use `ifdown <interface_name>` to deactivate an interface:

```
# ifdown enp134s1f0
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `systemd` service unit for networking is named `network.service`, shown as follows:

```
# systemctl list-units --type service | grep network
network.service loaded active exited LSB: Bring up/down networking
```

Use the `systemctl` command and specify `network` to start, stop, or view the status of network interfaces. The following example starts and stops all network interfaces:

```
# systemctl restart network
```

You can also use interface control scripts, located in the `/etc/sysconfig/network-scripts` directory, to start and stop network interfaces. The `ifup` and `ifdown` interface scripts are symbolic links to scripts in the `/usr/sbin` directory. When either of these scripts are called, they require the interface to be specified as an argument. For example, to bring the `enp134s1f0` interface down:

```
# ifdown enp134s1f0
```

To bring the `enp134s1f0` interface up:

```
# ifup enp134s1f0
```

There are additional interface control scripts in the `/etc/sysconfig/network-scripts` directory to start and stop different types of interfaces such as PPP, ISDN, PPP, and IPv6.

The ethtool Utility

- `ethtool` is used to query and set low-level network interface properties.
- Changes made by `ethtool` do not persist after a reboot.
- To show current low-level properties on an interface:

```
# ethtool enp134s1f0
...
```

- Use the `-s` option to set low-level properties on an interface. Example:

```
# ethtool -s enp134s1f0 speed 1000 autoneg on duplex full
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `ethtool` command 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
ethtool version 4.8
Usage:
    ethtool DEVNAME Display standard information about device
    ethtool -s|--change DEVNAME      Change generic options
        [ speed %d ]
        [ duplex half|full]
    ...

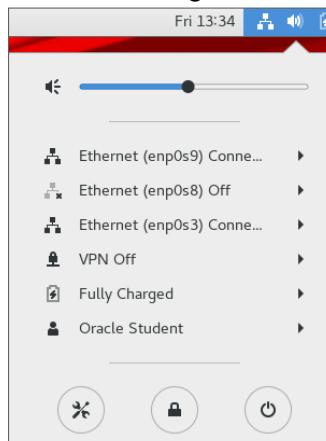
```

The following example configures the `enp134s1f0` interface to 1000 Mb/sec, full duplex, and enables autonegotiate:

```
# ethtool -s enp134s1f0 speed 1000 autoneg on duplex full
```

NetworkManager

- NetworkManager:
 - Dynamically detects and configures network connections
 - Includes a GNOME Notification Area network icon
- Click the icon to display status and to manage network connections.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

NetworkManager is the default networking service in Oracle Linux 7. It dynamically detects and configures network connections and also attempts to keep network interfaces up and active. To check whether NetworkManager is active and enabled:

```
# systemctl status NetworkManager
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service;
  enabled; vendor preset: enabled)
    Active: active (running) since ... MDT; ... ago
      ...

```

If needed, use the following commands to ensure that the package is installed and that the service is started:

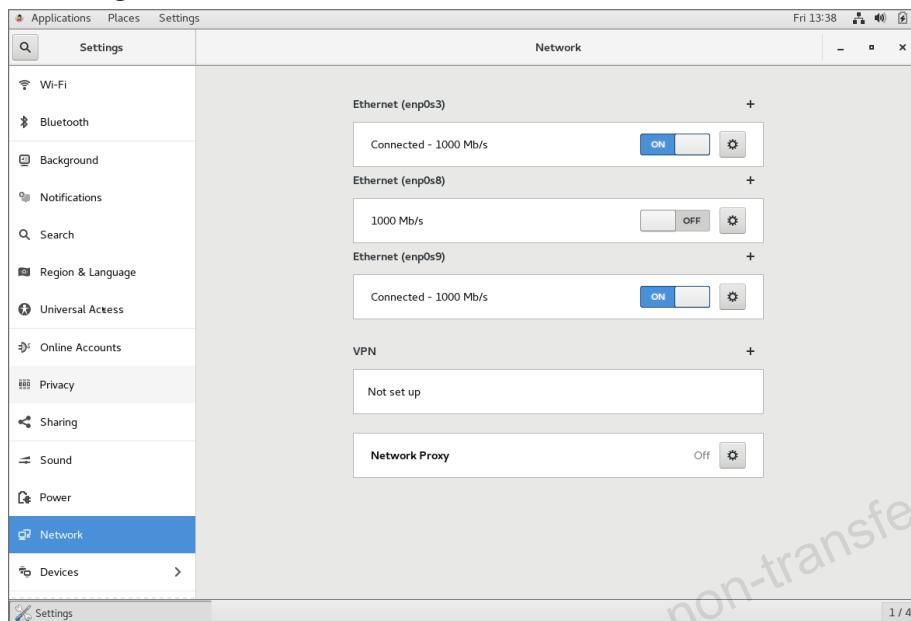
```
# yum install NetworkManager
# systemctl start NetworkManager
```

Run the following command to ensure that NetworkManager starts at boot time:

```
# systemctl enable NetworkManager
```

The GNOME Shell provides a network icon in the Notification Area, which is combined with other settings when clicked. Network connection states as reported by NetworkManager can be seen and managed from here. A sample status screen is shown in the slide. In this example, there are three networking interfaces, `enp0s9` (Connected), `enp0s8` (Off), and `enp0s3` (Connected). By clicking an interface, you can connect it and/or go to Network settings for further configuration options.

Network Settings



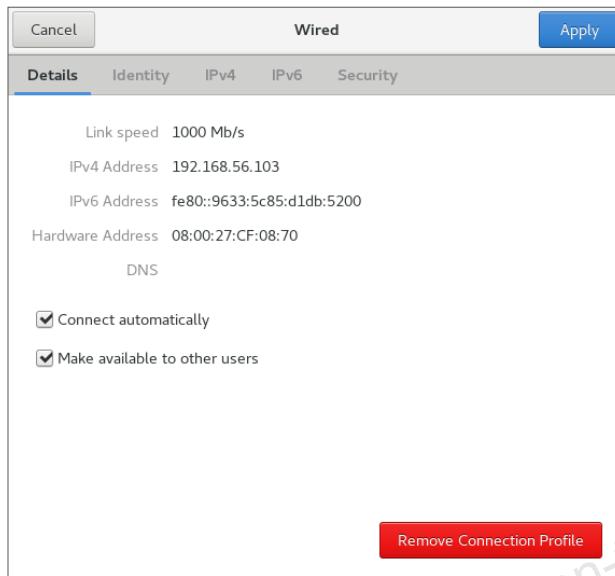
ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Network settings window is shown in the slide. It is a menu option in the GNOME Settings interface. The Network window shows existing network interfaces as well as options to configure a VPN and a network proxy. Details can be displayed for a given interface by clicking on the gear icon to the right of an entry. You can toggle the **ON/OFF** switch to enable or disable an entry.

Click the plus sign (+) to the right of an interface name to add a new connection profile. A profile is a group of connection settings for an interface that is given a name. You can define more than one profile for an interface and apply each profile as needed. When adding a profile, NetworkManager opens the same window used for editing an existing connection. When the process of creating a profile is complete, a new configuration file is created.

View or Edit an Existing Network Connection



ORACLE

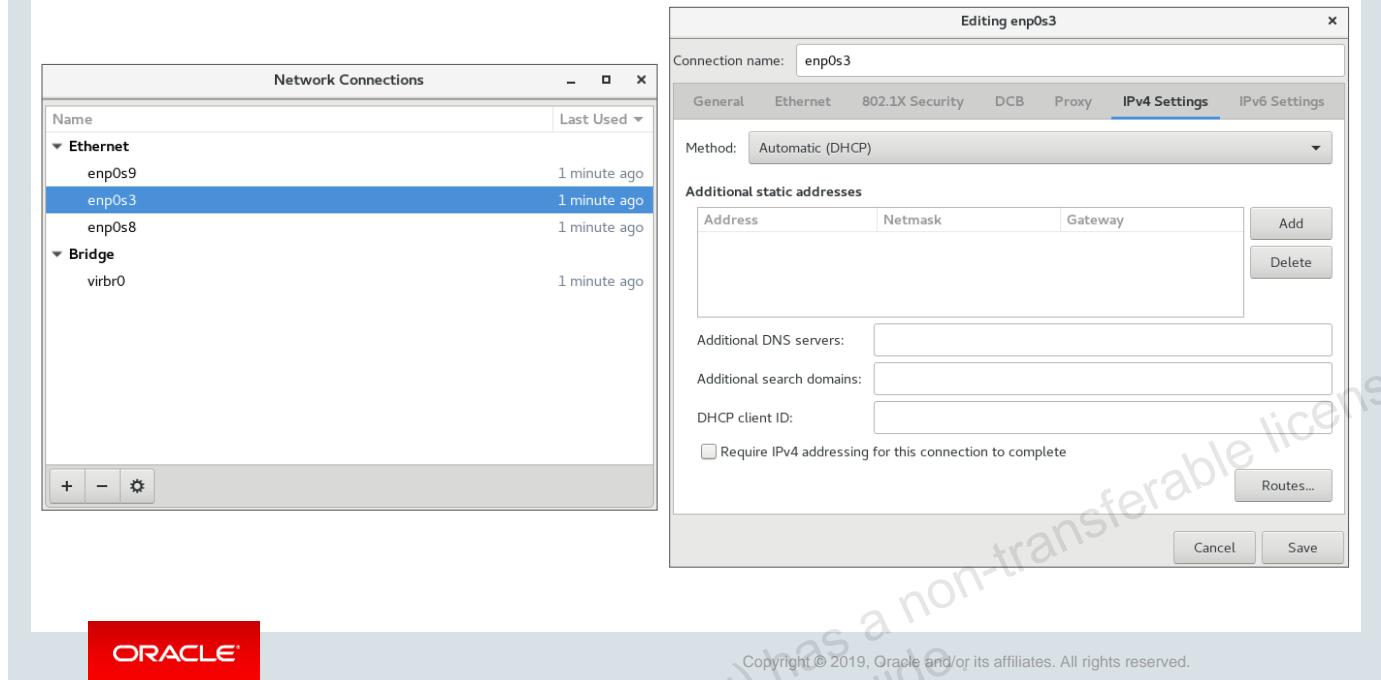
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

To view connection settings for an interface or to edit settings, click the gear icon to the right of the desired entry. A window like the one shown in the slide appears. You can then select from the following list of options:

- **Details:** Displays IPv4 and IPv6 addresses, the hardware address, default route, and any DNS servers. There are check boxes for connecting automatically and making the profile available to other users.
- **Identity:** Specify the device name, MAC address, cloned address, and MTU.
- **IPv4:** Specify a configuration method of DHCP, Manual, or Link-Local only, or disable the interface. Specify DNS servers and enable automatic DNS configuration. Specify routes, with address, netmask, gateway, and metric options and enable automatic route configuration.
- **IPv6:** Specify a configuration method of Automatic (Stateless Address Autoconfiguration), Automatic-DHCP only (stateful DHCPv6), Manual, or Link-Local only, or disable the interface. Specify DNS servers and enable automatic DNS configuration. Specify routes, with address, netmask, gateway, and metric options and enable automatic route configuration.
- **Security:** Enable or disable 802.1x security, choose an authentication method, and provide additional authentication information based on the chosen method. Available authentication methods are MD5 (message-digest algorithm), TLS (Transport Layer Security), PWD (Shared Password), FAST (Flexible Authentication via Secure Tunneling), Tunneled TLS, or Protected EAP (PEAP) (Protected Extensible Authentication Protocol).

Click **Remove Connection Profile** to delete the profile.

Network Connections Editor



You can also use the Network Connections editor window to add, delete, or edit network interface information. You can use the `rpm` command to check whether the `nm-connection-editor` package is installed:

```
# rpm -q nm-connection-editor
nm-connection-editor-...
```

If needed, you can use yum to install the required package:

```
# yum install nm-connection-editor
```

To display the Network Connections window:

```
# nm-connection-editor
```

The example in the slide shows the use of the Network Connections window to configure IPv4 settings for an Ethernet connection. The window on the left shows the existing Ethernet connections. Select an entry from the list and then click the gear icon to modify parameters. Categories in the Editing window include:

- **General:** Specify to automatically connect and other general parameters.
- **Ethernet:** Specify the interface name and/or MAC address, cloned MAC address if needed, MTU, Wake on LAN, and Link negotiation parameters.
- **802.1x Security:** Enable 802.1x security and specify authentication information.
- **DCB:** Enable Data Center Bridging (DCB) and specify parameters.
- **Proxy:** Specify the method and PAC, as appropriate.
- **IPv4 Settings:** Specify IPv4 settings as shown in the slide.
- **IPv6 Settings:** Specify IPv6 settings.

The nmcli Utility

- This is a command-line tool used to control NetworkManager.
 - Useful for scripting and for controlling NetworkManager without a GUI
- The command provides seven categories, or objects:
 - general: NetworkManager's general status and operations
 - networking: Overall networking control
 - radio: NetworkManager radio switches
 - connection: NetworkManager's connections
 - device: Devices managed by NetworkManager
 - agent: NetworkManager secret and/or polkit agent
 - monitor: Monitor NetworkManager changes
- See the `nmcli-examples(5)` manual page for examples.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

NetworkManager includes a command-line tool, `nmcli`, which is used to control NetworkManager. You can use `nmcli` to create, display, edit, delete, activate, and deactivate network connections, as well as control and display network device status. The syntax is:

```
nmcli [OPTIONS] OBJECT { COMMAND | help }
```

There are seven different objects: general, networking, radio, connection, device, agent, and monitor. Use the `help` argument to display the options and information about the seven different objects:

```
# nmcli help

...
OPTIONS
...
OBJECT
  g[eneral]      NetworkManager's general status and operations
  n[etworking]   overall networking control
  r[adio]        NetworkManager radio switches
  c[onnection]   NetworkManager's connections
  d[evice]       devices managed by NetworkManager
  a[gent]         NetworkManager secret agent or polkit agent
  m[onitor]     monitor NetworkManager changes
```

The nmcli general Object

The `nmcli general` object provides the following commands:

- `status`: Show the overall status of NetworkManager.
- `hostname`: Get or change the system host name. The system host name is stored in `/etc/hostname`.
- `permissions`: Show permissions for the various authenticated operations that NetworkManager provides.
- `logging`: Get or change NetworkManager logging level for domains. See the `NetworkManager.conf(5)` man page for descriptions of log levels and domains.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `nmcli general` object to show NetworkManager status and permissions. This command also allows you to view and change the system host name and the NetworkManager logging level. The following command provides help on the `nmcli general` object:

```
# nmcli general help
Usage: nmcli general { COMMAND | help }
COMMAND := { status | hostname | permissions | logging }
...
```

Some examples of using this command follow. Use the following command to display the overall status of NetworkManager. The `status` argument is the default and can be omitted.

```
# nmcli general status
STATE      CONNECTIVITY   WIFI-HW   WIFI      WWAN-HW   WWAN
connected   full          enabled    enabled    enabled    enabled
```

The `hostname` argument is used to display or change the system host name. The host name is stored in the `/etc/hostname` file. The following example changes the host name to `my_host.example.com` and updates the `/etc/hostname` file:

```
# nmcli general hostname my_host.example.com
```

The permissions argument shows the permissions a caller has for the various authenticated operations that NetworkManager provides. The following example shows some of the permissions for enabling and disabling networking, changing Wi-Fi and WWAN state, modifying connections, and other operations:

```
# nmcli general permissions
PERMISSION                                     VALUE
org.freedesktop.NetworkManager.enable-disable-network  yes
org.freedesktop.NetworkManager.enable-disable-wifi    yes
org.freedesktop.NetworkManager.enable-disable-wwan   yes
org.freedesktop.NetworkManager.enable-disable-wimax  yes
org.freedesktop.NetworkManager.sleep-wake           yes
org.freedesktop.NetworkManager.network-control     yes
org.freedesktop.NetworkManager.wifi.share.protected yes
org.freedesktop.NetworkManager.wifi.share.open      yes
org.freedesktop.NetworkManager.settings.modify.system yes
org.freedesktop.NetworkManager.settings.modify.own   yes
org.freedesktop.NetworkManager.settings.modify.hostname yes
...
...
```

The logging argument is used to get and change NetworkManager logging level and domains. Without any argument, the current logging level and domains are shown as follows:

```
# nmcli general logging
LEVEL DOMAINS
INFO  PLATFORM,RFKILL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,IP4,IP6,
      AUTOIP4,DNS,VPN,SHARING,SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,
      DEVICE,OLPC,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,TEAM,
      CONCHECK,DCB,DISPATCH,AUDIT,SYSTEMD,PROXY
```

To change logging state, provide the level and/or domain parameters using the following syntax:

```
nmcli general logging [level <log level>] [domains <log domains>]
```

The logging level can be one of the following (listed in order of verbosity):

- **ERR**: Logs only critical errors
- **WARN**: Logs warnings that might reflect operation
- **INFO**: Logs various informational messages that are useful for tracking state and operations
- **DEBUG**: Enables verbose logging for debugging purposes
- **TRACE**: Enables more verbose logging than **DEBUG**

The following example sets the logging level to **DEBUG** for the **IPV4** domain:

```
# nmcli general logging level DEBUG domains IPV4
```

The following example sets the logging level to **INFO** for all domains:

```
# nmcli general logging level INFO domains ALL
```

For information about configuring NetworkManager logging and for domain descriptions, see the **NetworkManager.conf(5)** man page.

The nmcli networking Object

- The nmcli networking object provides the following commands:
 - on: Enable networking by NetworkManager.
 - off: Disable networking by NetworkManager.
 - connectivity [check]: Get network connectivity state.
- To display networking status:

```
# nmcli networking  
enabled
```

- To get network connectivity state:

```
# nmcli networking connectivity check  
full
```

- Possible states: none, portal, limited, full, unknown



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the nmcli networking object to show NetworkManager networking status, or to enable and disable networking. Disabling networking removes the configuration from all devices and changes them to the “unmanaged” state. The following command provides help on the nmcli networking object:

```
# nmcli networking help  
Usage: nmcli networking { COMMAND | help }  
COMMAND := { [ on | off | connectivity ] }  
...
```

Some examples of using this command are shown as follows. The following sequence of commands displays the networking status and then disables and enables networking:

```
# nmcli networking  
enabled  
# nmcli networking off  
# nmcli networking  
disabled  
# nmcli networking on
```

The connectivity argument shows the network connectivity state. An optional check argument tells NetworkManager to recheck the connectivity. Without the check argument, the command displays the most recent known connectivity state without rechecking. The following example includes the check argument:

```
# nmcli networking connectivity check  
full
```

Possible states are:

- **none**: The host is not connected to any network.
- **portal**: The host is behind a captive portal and cannot reach the full Internet.
- **limited**: The host is connected to a network, but it has no access to the Internet.
- **full**: The host is connected to a network and has full access to the Internet.
- **unknown**: The connectivity status cannot be determined.

The nmcli radio Object

- The nmcli radio object provides the following commands:
 - wifi [on | off]: Get or set status of Wi-Fi in NetworkManager
 - wwan [on | off]: Get or set status of WWAN (mobile broadband)
 - all [on | off]: Get or set status of all the above
- To display status of all the radio switches:

```
# nmcli radio
WIFI-HW  WIFI      WWAN-HW  WWAN
enabled   enabled   enabled   enabled
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the nmcli radio object to show radio switch status, or to enable and disable the switches. The following command provides help on the nmcli radio object:

```
# nmcli radio help
Usage: nmcli radio { COMMAND | help }
COMMAND := { all | wifi | wwan }
...
```

Some examples of using this command are given. The following sequence of commands displays the radio switch status and then disables Wi-Fi in NetworkManager:

```
# nmcli radio
WIFI-HW  WIFI      WWAN-HW  WWAN
enabled   enabled   enabled   enabled
# nmcli radio wifi off
# nmcli radio
WIFI-HW  WIFI      WWAN-HW  WWAN
enabled  disabled   enabled   enabled
```

The nmcli connection Object

- NetworkManager stores all network configuration information as connections.
- Connections contain all the information required to create or connect to a network.
- There can be multiple connections for a given device.
- Only one connection can be active on a device at a time.
- The `nmcli connection` object provides the following commands, as displayed, by using the `help` argument:

```
# nmcli connection help
Usage: nmcli connection { COMMAND | help }
COMMAND := {show | up | down | add | modify | clone | edit | delete |
           monitor | reload | load | import | export }
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `nmcli connection` object to start, stop, and manage network connections.

NetworkManager stores all network configuration information as connections. Connections contain all the information, such as MAC address and IP address, required to create or connect to a network. A connection is active when a device uses that connection's configuration to create or connect to a network.

There can be multiple connections for a given device but only one of them can be active on that device at any given time. The additional connections can be used to allow quick switching between different networks and configurations. For example, you can have a connection defined for a network interface that uses static IP addressing. You could have a second connection defined for the same network interface that uses DHCP.

The following command provides help on the `nmcli connection` object:

```
# nmcli connection help
Usage: nmcli connection { COMMAND | help }
COMMAND := { show | up | down | add | modify | clone | edit | delete |
            monitor | reload | load | import | export }
...
```

The various commands for the `nmcli connection` object are described in the following slides.

The nmcli connection show Command

- Use the `show` argument to list connection profiles.
- Include the `--active` option to list only the active profiles.

```
# nmcli connection show --active
NAME      UUID      TYPE      DEVICE
eth0      ...      ethernet   eth0
virbr0    ...      bridge     virbr0
```

- View detailed information by specifying an `<ID>` keyword and associated value:

```
# nmcli connection show id eth0
connection.id:           eth0
...
connection.type:          802-3-ethernet
connection.autoconnect:   yes
...
```

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `show` argument to list connection profiles. Include the `--active` option to list only the active profiles. Example:

```
# nmcli connection show --active
NAME      UUID      TYPE      DEVICE
eth0      ...      ethernet   eth0
virbr0    ...      bridge     virbr0
```

You can also view detailed information for a specific connection by specifying an optional `<ID>` keyword followed by an associated value. The `<ID>` can be `id`, `uuid`, `path`, or `apath`. The following example uses the `id` keyword to show detailed information for the `eth0` connection. Only partial output is shown:

```
# nmcli connection show id eth0
...
ipv4.addresses:           192.0.2.103/24
ipv4.gateway:             192.0.2.1
...
GENERAL.DBUS-PATH: /org/freedesktop/NetworkManager/ActiveConnection/49
...
```

nmcli connection up|down Commands

- Use the `up` argument to activate a connection.
- The connection is specified by its ID, UUID, or D-Bus path.
- The following example uses the connection ID:

```
# nmcli connection up id eth1
Connection successfully activated (D-Bus active path:
  /org/freedesktop/NetworkManager/ActiveConnection/13)
```

- Use the `down` argument to deactivate a connection.

```
# nmcli connection down id eth1
```

- If the connection is set to autoconnect, the connection starts automatically on the disconnected device again.
- In this case, use the `nmcli device disconnect` command instead of `nmcli connection down`.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `up` argument to activate a connection. The connection is specified by its name, UUID, or D-Bus path. When requiring a particular device on which to activate the connection, use the `ifname` option with the interface name.

The following example activates the `eth1` connection. The `show` argument is issued before and after to illustrate the result of the `up` argument:

```
# nmcli connection show
NAME      UUID            TYPE      DEVICE
eth0      ...             ethernet   eth0
virbr0    ...             bridge    virbr0
eth1      ...             ethernet   --
# nmcli connection up id eth1
Connection successfully activated (D-Bus active path:
  /org/freedesktop/NetworkManager/ActiveConnection/13)
# nmcli connection show
NAME      UUID            TYPE      DEVICE
eth0      ...             ethernet   eth0
eth1      ...             ethernet   eth1
virbr0    ...             bridge    virbr0
```

Use the `down` argument to deactivate a specific active connection. The following example deactivates the `eth1` connection. The `show` argument is issued before and after to illustrate the result of the `down` argument:

```
# nmcli connection show
NAME      UUID            TYPE      DEVICE
eth0      ...             ethernet  eth0
eth1      ...             ethernet  eth1
virbr0    ...             bridge    virbr0
# nmcli connection down id eth1
# nmcli connection show
NAME      UUID            TYPE      DEVICE
eth0      ...             ethernet  eth0
virbr0    ...             bridge    virbr0
eth1      ...             ethernet  --
```

If the connection has the “`connection.autoconnect`” flag set to “yes,” the connection automatically starts on the disconnected device again. In this case, use the `nmcli device disconnect` command instead of the `nmcli connection down` command.

The nmcli connection add Command

- Use the `add` argument to add a connection for NetworkManager.
- The command accepts the following options:
 - Common options: `type`, `ifname`, `con-name`, and so on
 - Type-specific options: `mac`, `ssid`, `mtu`, and so on
 - IP options: `ip4`, `ip6`, `gw4`, `gw6`
- Example:

```
# nmcli connection add con-name new-eth0 ifname eth0 type ethernet ip4
  192.168.2.100/24 gw4 192.168.2.1
Connection 'new-eth0' (<UUID>) successfully added.
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `add` argument to add a connection for NetworkManager. The syntax is as follows:

```
nmcli connection add COMMON_OPTIONS TYPE_SPECIFIC_OPTIONS
SLAVE_OPTIONS IP_OPTIONS
```

The `COMMON_OPTIONS` for the `add` argument are described as follows:

- **`type <type>`:** Connection type. Valid types of connections are `ethernet`, `wifi`, `wimax`, `pppoe`, `gsm`, `cdma`, `infiniband`, `bluetooth`, `vlan`, `bond`, `bond-slave`, `team`, `team-slave`, `bridge`, `bridge-slave`, `vpn`, `olpc-mesh`, `adsl`, `tun`, `ip-tunnel`, `macvlan`, `xvlan`, and `dummy`.
- **`ifname <ifname>`:** Interface to bind the connection to. A special value of "*" can be used for interface-independent connections.
- **`con-name <connection_name>`:** Connection name. This is optional. When not provided, a default name is generated, `<type>[-<ifname>] [-<num>]`.
- **`autoconnect yes|no`:** Whether the connection profile can be automatically activated. This is optional. The default is `yes`.
- **`save yes|no`:** Whether the connection is persistent. This is optional. The default is `yes`.

Some of the `TYPE_SPECIFIC_OPTIONS` for the `add` argument are shown on the next page. Refer to the `nmcli(1)` man page for the complete list of options.

The following lists the **TYPE_SPECIFIC_OPTIONS** for Ethernet and Wi-Fi connections:

- **ethernet TYPE_SPECIFIC_OPTIONS:**
 - **mac <MAC_address>**: MAC address of the device to which this connection is locked
 - **cloned-mac <cloned_MAC_address>**: Clone MAC address
 - **mtu <MTU>**: MTU
- **wifi TYPE_SPECIFIC OPTIONS:**
 - **ssid <SSID>**: SSID
 - **mac <MAC_address>**: MAC address of the device to which this connection is locked
 - **cloned-mac <cloned_MAC_address>**: Clone MAC address
 - **mtu <MTU>**: MTU
 - **mode infrastructure|ap|adhoc**

The **IP_OPTIONS** for the `add` argument are described as follows:

- **ip4 <IPv4_address> gw4 <IPv4_address>**: IPv4 addresses
- **ip6 <IPv6_address> gw6 <IPv6_address>**: IPv6 addresses

The following example adds an Ethernet connection. The `nmcli connection show` command is issued afterwards to view the results. Only partial output is shown.

```
# nmcli connection add con-name new-eth0 ifname eth0 type ethernet
ip4 192.168.2.100/24 gw4 192.168.2.1

Connection 'new-eth0' (<UUID>) successfully added.

# nmcli connection show

NAME      UUID           TYPE      DEVICE
eth0      ...            ethernet  eth0
virbr0    ...            bridge    virbr0
eth1      ...            ethernet  --
new-eth0  ...            ethernet  --
```

Each new connection creates an associated network interface configuration file in the `/etc/sysconfig/network-scripts` directory. Example:

```
# cd /etc/sysconfig/network-scripts
# ls ifcfg*
ifcfg-eth0  ifcfg-eth1  ifcfg-lo  ifcfg-new-eth0
```

The nmcli connection edit Command

- Use the `edit` argument to edit a connection by using an interactive editor. Example:

```
# nmcli connection edit new-eth0
...
You may edit the following settings: connection, 802-3-ethernet (ethernet),
802-1x, dcb, ipv4, ipv6, proxy
nmcli>
```

- Use the ‘?’ key or type ‘help’ to view commands.

- Use the `edit` argument without specifying a connection to add a new connection. Example:

```
# nmcli connection edit
...
Enter connection type:
...
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `edit` argument to edit an existing connection, identified by the connection ID, UUID, or D-Bus path. The following example specifies editing of the `new-eth0` connection:

```
# nmcli connection edit new-eth0
====| nmcli interactive connection editor |===
Editing existing '802-3-Ethernet' connection: 'new-eth0'
Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.
You may edit the following settings: connection, 802-3-ethernet
(ethernet), 802-1x, dcb, ipv4, ipv6, proxy
nmcli>
```

Use the ‘?’ key or type ‘help’ to display the available commands. Only partial output is shown as follows:

```
nmcli> ?
...
set    [<setting>.<prop> <value>]  :: set property value
...
quit                         :: exit nmcli
```

Use the `edit` argument without specifying a connection identifier to add a new connection. The interactive editor guides you through the connection editing. The following example adds a new Ethernet connection:

```
# nmcli connection edit  
Valid connection types: adsl, bluetooth, ..., wlan, vpn, wimax, 802-  
3-ethernet (ethernet), 802-11-wireless (wifi), ...  
Enter connection type: 802-3-ethernet  
==| nmcli interactive connection editor |===  
Adding a new '802-3-ethernet' connection  
Type 'help' or '?' for available commands.  
Type 'describe [<setting>.<prop>]' for detailed property  
description.  
You may edit the following settings: connection, 802-3-ethernet  
(ethernet), 802-1x, dcb, ipv4, ipv6, proxy  
nmcli> set connection.id new-eth1  
nmcli> set connection.interface-name eth1  
nmcli> set connection.autoconnect yes  
nmcli> set 802-3-ethernet.mtu auto  
nmcli> set ipv4.method manual  
nmcli> set ipv4.addresses 192.168.2.101/24  
nmcli> set ipv6.method auto  
nmcli> save  
Saving the connection with 'autoconnect=yes'. That might result in  
an immediate activation of the connection.  
Do you still want to save? (yes/no) [yes] ENTER  
Connection 'new-eth1' (<UUID>) successfully saved.  
nmcli> quit
```

The following `connection show` command lists the new 'new-eth1' connection:

```
# nmcli connection show  
NAME      UUID            TYPE      DEVICE  
...  
new-eth1  ...          ethernet    eth1  
...
```

A new network interface configuration file is created in the `/etc/sysconfig/network-scripts` directory:

```
# cd /etc/sysconfig/network-scripts; ls ifcfg*  
ifcfg-eth0  ifcfg-eth1  ifcfg-lo  ifcfg-new-eth0  
ifcfg-new-eth1
```

The nmcli connection modify Command

- Use the `modify` argument to modify properties in the connection profile.
 - The following example modifies the IPv4 DNS server address property (`ipv4.dns`) for the `new-eth1` connection:

```
# nmcli connection modify new-eth1 ipv4.dns NN.NN.NN.NN
```

- Use the '+' prefix to append a value:

```
# nmcli connection modify new-eth1 +ipv4.dns NN.NN.NN.NN
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `modify` argument to modify one or more properties in the connection profile. Identify the connection to modify by its ID, UUID, or D-Bus path. The provided value overwrites the existing property value.

Use an empty value ("") to set the property value to the default. You can use the + prefix for the property name to append an item to the existing value, or use the - prefix to remove a specified value. The following example modifies the IPv4 DNS server address. The `show` argument displays the values before and after the modification:

```
# nmcli connection show new-eth1
...
ipv4.dns:          --
...
# nmcli connection modify new-eth1 ipv4.dns NN.NN.NN.NN
# nmcli connection show new-eth1
...
ipv4.dns:          NN.NN.NN.NN
...
```

nmcli connection delete | reload | load Commands

- Use the `delete` argument to delete a configured connection. Example:

```
# nmcli connection delete new-eth1
```

- Use the `reload` argument to reload all connection files from disk. Example:

```
# nmcli connection reload
```

- Use the `load` argument to load or reload one or more configuration files from disk. Example:

```
# nmcli connection load /etc/sysconfig/network-scripts/ifcfg-new-eth0
```

- Set `monitor-connection-files` to `true` in a NetworkManager configuration file to cause NetworkManager automatically to reload configuration files when changed.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The remaining three arguments to the `nmcli connection` command are given in the slide.

Use the `delete` argument to delete a configured connection profile. Identify the connection to delete by its ID, UUID, or D-Bus path.

Use the `reload` argument to reload all configuration files from disk. Use this command to tell NetworkManager to reread the connection profiles from disk whenever a change was made to them. Set `monitor-connection-files` to `true` in a NetworkManager configuration file so that NetworkManager will automatically reload configuration files when they change. See the `NetworkManager.conf(5)` man page for configuration file details and issues if this is set to `true`.

Use the `load` argument to load or reload one or more specific configuration files from disk. This is not needed if the auto-loading feature is enabled for the connection.

The nmcli device Object

- The nmcli device object provides the following commands, among others:
 - status: Display the status of all devices.
 - show [<filename>]: Show detailed information about devices.
 - connect <filename>: Connect the device.
 - disconnect <filename>: Disconnect the device.
 - wifi list | connect | hotspot | rescan: List Wi-Fi access points, connect to a Wi-Fi network, create a Wi-Fi hotspot, or rescan for available access points.
- To display status of all the devices:

```
# nmcli device
DEVICE      TYPE      STATE           CONNECTION
...
eth0        ethernet  connected       eth0
...
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the nmcli device object to show and manage network interfaces. The following command provides help on the nmcli device object:

```
# nmcli device help
Usage: nmcli device { COMMAND | help }
COMMAND := { status | show | connect | reapply | modify | disconnect |
delete | monitor | wifi | llidp }
...
```

Some examples of using this command are shown as follows. The following sequence of commands displays the status of all devices. The status argument is the default.

```
# nmcli device
DEVICE      TYPE      STATE           CONNECTION
virbr0      bridge    connected       virbr0
eth0        ethernet  connected       eth0
eth1        ethernet  disconnected   --
lo          loopback  unmanaged      --
virbr0-nic tun       unmanaged      --
```

The following example displays detailed information about a device:

```
# nmcli device show
GENERAL.DEVICE:                         eth0
GENERAL.TYPE:                            ethernet
GENERAL.HWADDR:                          00:16:3E:00:01:03
GENERAL.MTU:                             1500
GENERAL.STATE:                           100 (connected)
GENERAL.CONNECTION:                      eth0
...
IP4.ADDRESS[1]:                          192.0.2.103/24
IP4.GATEWAY:                            192.0.2.1
...
IP6.ADDRESS[1]:                          fe80::b28b:49e3:68d1:c916/64
IP6.GATEWAY:                            --
...
GENERAL.DEVICE:                          eth1
GENERAL.TYPE:                            ethernet
GENERAL.HWADDR:                          00:16:3E:00:02:03
GENERAL.MTU:                             1500
GENERAL.STATE:                           100 (connected)
GENERAL.CONNECTION:                      eth1
...
```

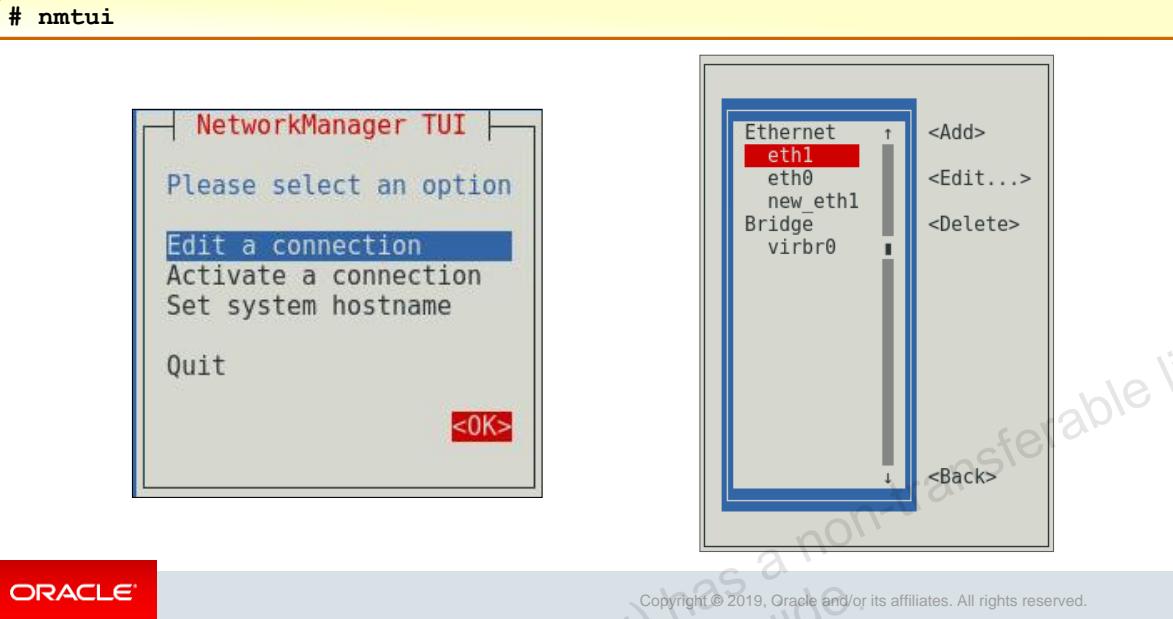
The following example shows the effect of using the disconnect and connect arguments:

```
# nmcli device disconnect eth1
# nmcli device
DEVICE      TYPE      STATE           CONNECTION
...
eth1        ethernet  disconnected   --
...
# nmcli device connect eth1
Device 'eth1' successfully activated with <UUID>.
# nmcli device
DEVICE      TYPE      STATE           CONNECTION
...
eth1        ethernet  connected      eth1
...
```

The "nmcli device wifi" command provides the following arguments:

- **list**: List available Wi-Fi access points.
- **connect <(B) SSID>**: Connect to a Wi-Fi network specified by Service Set Identifier (SSID) or Basic Service Set Identifier (BSSID).
- **hotspot**: Set up a Wi-Fi hotspot.
- **rescan**: Request that NetworkManager rescan for available Wi-Fi access points.

The nmtui Utility



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

A text-based user interface (TUI) for NetworkManager exists to add or edit a network connection, to activate a connection, and to set the system host name. Enter the `nmtui` 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 “Edit a connection” and press Enter to display the second screen. From this screen, you can edit and delete an existing connection, or add a new connection.

You can use the `rpm` command to check whether the `nmtui` utility is installed:

```
# rpm -q NetworkManager-tui
```

```
NetworkManager-tui-...
```

If needed, use the following command to install the package that provides the `nmtui` utility:

```
# yum install NetworkManager-tui
```

The ip Utility

- The `ip` utility:
 - Is used to display and manipulate devices, routing, policy routing, and tunnels
 - Replaces the `ifconfig` command
 - Provides several OBJECT arguments, such as:
 - `link`: Network device
 - `address` (or `addr`): IPv4 or IPv6 address on a device
 - `route`: Routing table entry
 - Provides a number of COMMANDS for each OBJECT, such as:
 - `add`, `change`, `del`, `show`, and so on
- Use `help` to show COMMANDS available for an OBJECT.

```
# ip addr help
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

You can use the `ip` command to display the status of an interface, configure network properties, or for debugging or tuning the network. The `ip` command replaces the `ifconfig` command, which is deprecated. The syntax of the `ip` utility is as follows:

```
ip [ OPTIONS ] OBJECT { COMMAND | help }
```

OBJECT

The following describes the OBJECT field:

- `link`: Network interface device
- `address` (or `addr`): Protocol (IPv4 or IPv6) address on a device
- `addrlabel`: Label configuration for protocol address selection
- `route`: Routing table entry
- `rule`: Rule in routing policy database
- `neighbour` (or `neigh`): Manage ARP or NDISC cache entries.
- `ntable`: Manage the neighbor cache's operation.
- `tunnel`: Tunnel over IP
- `tuntap`: Manage TUN/TAP devices.
- `maddress` (or `maddr`): Multicast address

The OBJECT field description continues:

- **mroute**: Multicast routing cache entry
- **mrule**: Rule in multicast routing policy database
- **monitor**: Watch for netlink messages.
- **xfrm**: Manage IPSec policies.
- **netns**: Manage network namespaces.
- **l2tp**: Tunnel ethernet over IP (L2TPv3)
- **tcp_metrics** (or **tcpmetrics**): Manage TCP metrics.
- **token**: Manage interface identifiers that involve tokens.

COMMAND

The COMMAND specifies the action to perform on the object. The set of possible actions depends on the object type. In general, you can add, delete, and show objects. Some objects do not allow all of these operations; some objects allow more operations. The `help` command displays a list of available commands and syntax for a specified object. The following example gives the commands available for the `addr` object. Only partial output is shown.

```
# ip addr help

Usage: ip address {add|change|replace} IFADDR dev IFNAME [ LIFE...
       ip address del IFADDR dev IFNAME [mngtmpaddr]
       ip address {save|flush} [ dev IFNAME ] [ scope ... ]
       ip address [ show [ dev IFNAME ] [ scope SCOPE-ID ] ...
       ip address {showdump|restore}

IFADDR := PREFIX | ADDR peer PREFIX
...
```

As shown in this example, the commands for the `addr` object are `add`, `change`, `replace`, `del` (or `delete`), `show`, `save`, `flush`, `showdump`, and `restore`. The `addr` object is discussed further in the next slide.

OPTIONS

The following describes some of the available OPTIONS for the `ip` utility:

- **-v, --Version**: Display the version of the `ip` utility.
- **-b, --batch <FILENAME>**: Read commands from `<FILENAME>` or from standard input and execute them. A failure in a command in batch mode terminates `ip`.
- **-force**: Do not terminate `ip` on errors in batch mode.
- **-s, --stats, --statistics**: Display more information. If the option appears twice or more, the amount of information increases.
- **-l, --loops <COUNT>**: Specify the maximum number of loops the ‘`ip addr flush`’ logic attempts. The default is 10. Zero (0) means loop until all addresses are removed.
- **-f, --family <FAMILY>**: Specify the protocol family: `inet`, `inet6`, `bridge`, `ipx`, `dnet`, or `link`. The `link` family is a special identifier meaning that no networking protocol is involved. The respective shortcuts are `-4`, `-6`, `-B`, `-I`, `-D`, and `-0`.

The ip addr Object

- Use the `ip addr` object to show and manage IPv4 or IPv6 addresses on a device.
 - Changes made with `ip` are not persistent.
- To show the status of all active devices:

```
# ip addr
```

- To add an IPv4 address to a network interface device:

```
# ip addr add 192.168.2.101/24 dev eth1
```

- To remove an IPv4 address from a device:

```
# ip addr del 192.168.2.101/24 dev eth1
```

- To remove all IP addresses from a device:

```
# ip addr flush dev eth1
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `ip addr` object to show and manage IPv4 or IPv6 addresses on a device. The following example shows IP status for some active devices. The `show` command is the default.

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    state UP qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
            valid_lft forever preferred_lft forever
        inet6 fe80::fb2:7716:9092:8be1/64 scope link
            valid_lft forever preferred_lft forever
...
```

The following example uses the `add` argument to add the IPv4 address `192.168.2.101/24` to the `eth1` interface. The `show` argument is given afterwards to display the result. This example assumes the interface already has `192.168.1.103/24` assigned to it.

```
# ip addr add 192.168.2.101/24 dev eth1
# ip addr show eth1
...
link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global ...
        valid_lft forever preferred_lft forever
    inet 192.168.2.101/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::fb2:7716:9092:8be1/64 scope link ...
...
Use the del argument to delete the IPv4 address. Example:
```

```
# ip addr del 192.168.2.101/24 dev eth1
# ip addr show eth1
...
link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.103/24 brd 192.168.1.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::fb2:7716:9092:8be1/64 scope link
...
Use the flush argument to remove all the IPv4 addresses assigned to an interface. Example:
```

```
# ip addr flush dev eth1
# ip addr show eth1
...
link/ether 00:16:3e:00:02:03 brd: ff:ff:ff:ff:ff:ff
```

Any settings that you configure for network interfaces using `ip` do not persist across system reboots. To make the changes permanent, set the properties in the `/etc/sysconfig/network-scripts/ifcfg-<interface>` file.

Refer to the `ip-address (8)` man page for more information about using commands available for the `ip addr` object.

The ip link Object

- Use the `ip link` object to show and manage the attributes of network interfaces on the system.
 - Changes made with `ip` are not persistent.
- To show the status of a specific device:


```
# ip link show eth1
```
- To bring a specific network interface down:


```
# ip link set eth1 down
```
- To bring a specific network interface up:


```
# ip link set eth1 up
```
- To change a device attribute, for example, MTU:


```
# ip link set eth1 mtu 1000
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `ip link` object to show and manage the state of network interface devices on the system. The following example gives the commands available for the `link` object. Only partial output is shown.

```
# ip link help
Usage: ip link add [link DEV] [ name ] NAME
...
ip link delete { DEVICE | dev DEVICE ... type TYPE [ARGS]
...
ip link set { DEVICE | dev DEVICE | group DEVGROUP} [{up |
down}]
...
ip link show [ DEVICE | group GROUP] [up] ...
...
```

As this example shows, the commands for the `link` object are `add`, `delete`, `set`, and `show`. The `TYPE` argument can be any of the following: `vlan`, `veth`, `vcan`, `dummy`, `ifb`, `macvlan`, `macvtap`, `bridge`, `bond`, `team`, `ipoib`, `ip6tnl`, `ipip`, `sit`, `vxlan`, `gre`, `gretap`, `ip6gre`, `ip6gretap`, `vti`, `nlmon`, `team_slave`, `bond_slave`, `ipvlan`, `geneve`, `bridge_slave`, `vrf` and `macsec`.

Refer to the `ip-link(8)` man page for more information.

The following example shows the status of some active devices. The `show` argument is the default. Notice that the output is similar to that of the `ip addr` command, but without the IP address information.

```
# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT ...
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
...
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
...
```

Use the `set` argument to change device attributes. The `up` and `down` arguments change the state of the device. The following example brings the `eth1` interface down and then back up. The `show` argument displays the results of the `set` argument.

```
# ip link set eth1 down
# ip link show eth1
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
# ip link set eth1 up
# ip link show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

The following example uses the `set` argument to change the MTU attribute to 1000:

```
# ip link set eth1 mtu 1000
# ip link show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1000 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff
```

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 become stale after 60 seconds.
- Use the `ip neigh` object to display, add, or delete entries in the ARP table.
 - The `arp` command is deprecated.
- To display all entries:

```
# ip neigh  
...
```

- To delete all entries:

```
# ip neigh flush all
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

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. The following example uses the `ip addr show` command to display the MAC address, `00:16:3e:00:02:03`, and the IP address, `192.168.1.103`, for the `eth1` network interface:

```
# ip addr show eth1  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast  
state UP group default qlen 1000  
    link/ether 00:16:3e:00:02:03 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.103/24 brd 192.168.1.255 scope global ...  
        valid_lft forever preferred_lft forever
```

For performance reasons, ARP caches resolve IP addresses and associate MAC addresses in an ARP table (or cache). By default, entries are considered stale after 60 seconds. Stale entries are updated before being used again. This value can be modified on a per-network interface basis. For example, the following file stores the timeout value for the `eth1` interface:

```
# cat /proc/sys/net/ipv4/neigh/eth1/gc_stale_time
```

60

Use the `ip neigh` object to display the ARP table, to delete an ARP entry, or to add an entry to the table. The `ip neigh` object replaces the `arp` command, which is deprecated. The ARP table is also known by another name, the IP neighbor table.

Use the following command to display the commands available for the `ip neigh` object. Only partial output is displayed.

```
# ip neigh help
Usage: ip neigh { add | del | change | replace } { ADDR ...
...
        ip neigh { show | flush } [ proxy ] [ to PREFIX ] [ dev
DEV ] [ nud STATE ]
```

The `ip neigh` object commands are summarized as follows:

- `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.

The following example displays the ARP table. The `show` command is the default.

```
# ip neigh
192.0.2.102 dev eth0 lladdr 00:16:3e:00:01:02 REACHABLE
192.168.1.101 dev eth1 lladdr 00:16:3e:00:02:01 STALE
192.0.2.101 dev eth0 lladdr 00:16:3e:00:01:01 REACHABLE
192.168.1.102 dev eth1 FAILED
192.0.2.1 dev eth0 lladdr fe:ff:ff:ff:ff:ff REACHABLE
```

The following example clears all entries in the ARP table with verbosity:

```
# ip -s -s neigh flush all
192.0.2.102 dev eth0 lladdr 00:16:3e:00:01:02 used 75/73/43
probes 4 STALE
192.168.1.101 dev eth1 lladdr 00:16:3e:00:02:01 used 116/116/98
probes 4 STALE
192.0.2.101 dev eth0 lladdr 00:16:3e:00:01:01 used 70/68/38
probes 4 STALE
192.0.2.1 dev eth0 lladdr fe:ff:ff:ff:ff:ff ref 1 used 0/0/3
probes 0 DELAY

*** Round 1, deleting 4 entries ***
*** Flush is complete after 1 round ***
```

The following example removes entries in the ARP table on device `eth1`:

```
# ip neigh flush dev eth1
```

Refer to the `ip-neighbour (8)` man page for more information.

The ip route Object

- Use the `ip route` object 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:

```
# ip route
```

- To add an entry to the routing table:

```
# ip route add default via NN.NN.NN.2 dev enp134s1f0 proto static
# ip route add 192.0.2.1 via NN.NN.NN.2 dev enp134s1f0
```

- Configure permanent static routes in the `/etc/sysconfig/network-scripts/route-interface` file.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `ip 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.

To create a default route, include a `GATEWAY` entry in the `/etc/sysconfig/network` file. Use the `GATEWAYDEV` parameter to designate a specific interface. Example:

```
# cat /etc/sysconfig/network
GATEWAY=NN.NN.NN.1
GATEWAYDEV=enp134s1f0
```

Network traffic destined for hosts on another network would be handled by the `NN.NN.NN.NN` gateway on the local area network.

Displaying the Routing Table

Use the `ip route` command to display the routing table. Example:

```
# ip route
default via NN.NN.NN.1 dev enp134s1f0 proto static metric 1024
NN.NN.NN.0/23 dev enp134s1f0 proto kernel scope link src ...
192.168.122.0/24 dev virbr0 proto kernel scope link src ...
```

In this example, the gateway IP address of NN.NN.NN.1 was obtained from the entry in the /etc/sysconfig/network file. Refer to the ip-route (8) man pages for more information.

You can also use the netstat -r command to display the route table:

```
# netstat -r
```

Adding a Route

Use the ip 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 directed to the NN.NN.NN.2 IP address:

```
# ip route add default via NN.NN.NN.2 dev enp134s1f0 proto static
```

The following example adds a static route to a host address via a specific network interface.

```
# ip route add 192.0.2.1 via NN.NN.NN.2 dev enp134s1f0
```

Deleting a Route

Use the ip route delete command to delete an entry from the routing table, for example:

```
# ip route delete default via NN.NN.NN.2
```

```
# ip route delete 192.0.2.1
```

Configuring Permanent Static Routes

Any changes that you make to the routing table by using ip route do not persist across system reboots. 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 enp134s1f0 interface would be stored in the /etc/sysconfig/network-scripts/route-enp134s1f0 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 enp134s1f1 interface (NN.NN.10.1):

```
# cat /etc/sysconfig/network-scripts/route-enp134s1f1
```

```
198.168.2.0/24 via NN.NN.10.1 dev enp134s1f1
```

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 use of the IP command arguments to define the same entry:

```
ADDRESS0=198.168.2.0
```

```
NETMASK0=255.255.255.0
```

```
GATEWAY0=NN.NN.10.1
```

Start at 0 (as shown) and increment by one for each additional static route.

Networking in Oracle Cloud Infrastructure

- An instance must run within a subnet within a Virtual Cloud Network (VCN).
- VCN contains addresses in the IPv4 CIDR block.
 - Oracle recommends private ranges: 10.0.0.0/8, 172.16/12, and 192.168/16
 - Example CIDR block: 192.168.0.0/16
 - Publicly routable ranges are possible, but all addresses are termed "private."
 - Comes with a default route table
- Subnet is a subset of VCN, specified as either public or private.
 - CIDR is block specified, for example, 192.168.1.0/24
 - Private subnet has no public IP addresses.
 - Has a primary virtual network interface card (VNIC), and can have more
 - VNIC allows connection to the VCN.
- Instance is attached to a VNIC within a specified subnet.
 - Public subnet allows SSH connection over the Internet.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

A Virtual Cloud Network is a virtual version of a traditional network—including subnets, route tables, and gateways—on which your instances run. A cloud network resides within a single region but can cross multiple Availability Domains. You must have at least one VCN before you can launch instances.

When you create a VCN, you specify a single, contiguous IPv4 CIDR block. An example of a CIDR block is 192.168.0.0/16. The range of IP addresses in this example is 192.168.0.0 through 192.168.255.255. A portion of this IP address range is used for each Subnet created for the VCN.

A subnet is a subdivision of a cloud network and each instance resides in a subnet. You must have at least one subnet before you can launch instances. Subnets contain virtual network interface cards (VNICS), which attach to instances. Each subnet exists in a single Availability Domain (which must be specified) and consists of a contiguous range of IP addresses that do not overlap with other subnets in the cloud network (a CIDR Block must be specified).

An example of a CIDR block is 192.168.1.0/24. This example specifies 256 IP addresses. The range of IP addresses in this example is 192.168.1.0 through 192.168.1.255.

The first two IP addresses and the last IP address are reserved.

0 – network

1 - default route

255 – broadcast

Networking in Oracle Cloud Infrastructure

- Virtual route tables allow sending traffic outside a VCN, such as to the Internet or an on-premises network.
 - Default route table has no rules, which can be added later.
 - Route table is only used if destination IP address is outside the VCNs CIDR block.
- Gateways are virtual routers that provide paths for network traffic.
 - For example, an Internet gateway provides access to the Internet
- Secondary VNICs can be added after an instance is launched.
 - Can connect to multiple subnets in a VCN or to multiple VCNs
- Secondary private IP addresses can be added after an instance is launched.
 - Can be used on a standby instance if an instance is not running correctly
- VCN Peering allows direct, private connections between VCNs.
 - Can be local (within a region) or remote (between regions)



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Route rules specify a destination CIDR block and target for any traffic that matches the specified CIDR block. The target can be “Internet gateway” for example, to provide access to the Internet.

Some examples of gateways include: Internet gateway (access to the Internet), dynamic routing gateway (allows private traffic between your VCN and on-premises network), and service gateway (allows private traffic between your VCN and another Oracle Cloud Infrastructure service).

Secondary VNICs can be useful for communication between different subnets or VCNs. A secondary VNIC can be in a subnet in the same VCN as the primary VNIC or a different VCN. All the VNICs must be in subnets in the same availability domain as the instance.

Secondary IP addresses can be useful in failover situations as well as when it is desired to run a number of services on one instance, for example.

VCN peering network traffic is direct and private between VCNs. It does not traverse an on-premises network. Cross-region VCN peering allows backups in different regions, for example.

Quiz



Which of the following statements is true?

- a. Network interface configuration files are located in the `/etc/sysconfig/network` directory.
- b. The NetworkManager GUI 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 © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: c

Network interface configuration files are located in the `/etc/sysconfig/network-scripts/` directory. The `ifup eth0` command activates the `eth0` interface.

Quiz



Which of the following tasks can you perform by using the `nmcli` utility?

- a. Disable and enable networking operations.
- b. Manage network connection profiles.
- c. Change the system host name.
- d. Manage routing.
- e. Manage the ARP cache.



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz



Which of the following tasks can you perform by using the `ip` utility?

- a. Manage network devices.
- b. Manage network addressing.
- c. Change the system host name.
- d. Manage routing.
- e. Manage the ARP cache.



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, d, e

Summary

In this lesson, you should have learned how to:

- Describe network interface configuration files
- Describe additional network configuration files
- Start the network service
- Use the `ethtool` utility
- Describe NetworkManager and use the NetworkManager GUI
- Use the Network Connections editor
- Use the `nmcli` and `nmtui` utilities
- Describe ARP and the ARP cache
- Use the `ip` utility
- Describe networking in Oracle Cloud Infrastructure



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practice 13: Overview

This practice covers the following topics:

- Configuring the `eth1` network interface
- Using NetworkManager with the GNOME GUI
- Using the Network Connection Editor
- Using the `nmcli` utility
- Using the `nmtui` utility
- Using the `ip` utility



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Explain the background for IPv6 development
- Describe some features of IPv6
- Explain the IPv6 address format
- List and describe IPv6 address types
- Describe aspects of IPv6 link-local unicast, global unicast, and multicast address layouts
- List the three types of autoconfiguration for obtaining IPv6 addresses
- Explain Stateless Address Autoconfiguration (SLAAC)
- Explain two IPv6 settings for specific interfaces
- Describe some IPv6 commands



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6: Overview

- This is the most recent version of the IP specification.
- In the 1990s, the Internet Engineering Task Force (IETF) spearheaded efforts to address a pending IP address shortage.
- Classful addressing resulted in inefficient address allocation.
- Efforts were made to mitigate the IP address issue:
 - Classless Inter-Domain Routing (CIDR) with Variable-Length Subnet Masking (VLSM)
 - Network Address Translation (NAT)
- RFC 8200 is the current IPv6 specification—July 2017:
 - Succeeds IPv4, though both coexist currently
 - Addresses the IPv4 address shortage
 - Provides new functionality



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6 implements a 128-bit address scheme that supports 3.4×10^{38} nodes. IPv4, with a 32-bit address scheme, provides for more than four billion addresses. However, many of these addresses were not usable because classful addressing techniques wasted large numbers of possible IPv4 addresses. With classful addressing, network portions of IP addresses were specified in multiples of 8 bits (octets). For example, the network portion of a Class A address was 8 bits, of a Class B address was 16 bits, and of a Class C address, 24 bits. This led to address allocations that could be too small or too large.

Classless Inter-Domain Routing (CIDR) provided a way to utilize IPv4 addresses more efficiently. An IP address is given a suffix indicating the number of bits of the network portion (prefix) of the IP address. The number of bits specifying the network prefix is not restricted to octet multiples, but can vary as desired. This is referred to as variable-length subnet masking (VLSM), and it provides greater flexibility in specifying the size of a network.

Another technique that has helped to alleviate the IPv4 address shortage is commonly referred to as Network Address Translation or NAT. NAT uses IP addresses on private networks without exposing them to the Internet, and there are different implementations of NAT. In the NAT implementation that helps alleviate the address shortage, multiple private IPv4 addresses are translated into a single global IPv4 address before packets are placed on the public network. This reduces the number of global IPv4 addresses needed.

RFC 8200 ("Internet Protocol, Version 6 (IPv6) Specification") is the current IPv6 specification document.

IPv6: Features

- Increased address space
- Autoconfiguration
- Header is fixed length with fewer fields—less processing overhead
- Extension headers can be specified to support various options
- Flow label
- Scope

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The IPv6 features are as follows:

- Increased address space: The address size is increased from 32-bit addresses to 128-bit addresses.
- Autoconfiguration: IPv6 systems can configure their IPv6 addresses automatically. Administrators, however, still have to administer the name-to-IPv6 address mapping.
- Fixed length header with fewer fields: The number of header fields in an IPv6 datagram is reduced from 14 fields to 8.
- Extension headers: Extension headers can optionally be used in addition to the primary header. They are located between the required IPv6 packet header and the payload.
- Flow label: A flow label in the header provides for flows. Flows identify a sequence of datagrams from the same source to the same destination when the source requests special handling of the specified datagram sequence by the intervening routers.
- Scope: Part of the address indicating the network range in which the address can be properly used, for example, local (the local link) or global (the Internet)

IPv6 Addresses

- IPv6 Addresses use 128 bits.
- The first part of the address is a prefix:
 - Analogous to the network portion of an IPv4 address
 - Specified with the form: prefix/prefix-length-in-bits
 - Contents of the prefix depend on the address type
- The second part of the address is the interface identifier (ID), which can be set in various ways:
 - Derived from the system's MAC address—modified EUI-64 format
 - Randomly generated to protect privacy
 - Manually
- Autoconfiguration simplifies address assignment.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6 addressing uses 128 bits. The first part of the address is a prefix, which is like the network portion of an IPv4 address. The second part of the address is the interface identifier, which uniquely identifies a node. It is analogous to the IPv4 host portion and is derived from the system's MAC address or in other ways. For example, for privacy considerations, an interface identifier can be randomly generated, to avoid an attacker using an unchanging identifier to identify patterns when gathering information from network communications. See RFC 4941 ("Privacy Extensions for Stateless Address Autoconfiguration in IPv6") and RFC 7217 ("A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)") for details. RFC 4941 describes random interface IDs that are temporary, while RFC 7217 describes interface IDs that are randomly generated, but remain the same while a node is within the current subnet. RFC 4291 ("IPv6 Addressing Architecture") describes identifiers besides those that use 48-bit MAC addresses as well.

When the autoconfiguration capability of IPv6 is used, address configuration is very simple.

IPv6 Address Representation

- 128-bit IPv6 addresses contain eight sets of 4 hexadecimal numbers delimited by colons (each number is 4 bits), for example:

2001:0db8:0000:0000:0a00:20ff:feb5:4137

- Guideline is to use the shortest representation possible:

- Leading 0s (zeros) are removed, giving:

2001:db8:0:0:a00:20ff:feb5:4137

- Consecutive 16-bit sets of zeros are represented with a double colon (::). You can do this only once in any address:

2001:db8::a00:20ff:feb5:4137

- The double colon must not be used in place of only one 16-bit set of zeros.
 - See RFC 5952 ("IPv6 Text Representation") for more details on address representations.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

In an IPv6 address, a set of 4 hexadecimal numbers is 16 bits or a hexet. Each individual hexadecimal number within a hexet comprises 4 bits (a "nibble").

Using a double colon in place of consecutive sets of zeros results in an address that is compressed.

RFC 5952 ("IPv6 Text Representation") provides details on address representations.

RFC 5952 is in accord with RFC 4291 ("IPv6 Addressing Architecture") in which a more abbreviated guideline is provided.

IPv6 Address Types

- IPv6 supports the following address types:
 - Unicast addresses
 - Multicast addresses
 - Anycast addresses
- Specific examples of unicast addresses are:
 - Global unicast
 - Link-local unicast
- IPv6 differs from IPv4 in that IPv6 does not provide broadcast addresses as a mechanism for communicating with other hosts on a subnet.
- The high-order bits of the prefix determine the address type.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6 has three general types of addresses that you can use to communicate across a network. For sending messages, IPv6 supports:

- Unicast addresses
- Multicast addresses
- Anycast addresses

Examples of specific types of unicast addresses are global unicast and link-local unicast.

IPv6 differs from IPv4 in that IPv6 does not provide broadcast addresses as a mechanism for communicating with other hosts on a subnet.

In IPv6, it is normal for more than one IPv6 addresses to be assigned to the same physical interface. There must be one link-local unicast address, and there may be one or more global unicast addresses.

High-Order Bits Specified for Address Types

- The high-order bits in an IPv6 address determine the address type.
- Some common address types:

Address Type	High-Order Bits	Prefix
Link-local unicast addresses	1111 1110 10	fe80::/10
Global unicast addresses	001	2000::/3
Multicast addresses	1111 1111	ff00::/8



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The table in the slide shows some common types of IPv6 addresses with their high-order bits and prefixes.

Refer to RFC 4291 ("IPv6 Addressing Architecture") for further information. The three high-order bits (001) of the global unicast address type is equivalent to 0x2 as in the table, not 0x1, since it is 3 of the 4 bits in the high-order nibble.

A special prefix has been defined in RFC 3849 ("IPv6 Address Prefix Reserved for Documentation"), to be used for "documentation" - 2001:db8::/32. The intent is to provide a prefix for use in examples, to avoid conflicts with existing, actual addresses. It is part of the global unicast address space, but should never be routed. This documentation prefix is used in some examples in this lesson.

Prefix Notation

- CIDR notation specifies the prefix length of an IPv6 address.
- An example of an address with the prefix specified is:

2001:db8::a00:20ff:feb5:4137/64

- /64 indicates that the prefix is 64 bits in length.
- 2001:db8:: is the 64-bit prefix—32-bit global unicast prefix followed by 32 bits of zeros.
- a00:20ff:feb5:4137 is the 64-bit interface identifier.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

RFC 4291 describes how IPv6 addresses use the prefix notation. Specifying addresses in this way is like CIDR notation in IPv4.

An example of a prefix address is:

2001:db8::a00:20ff:feb5:4137/64

/64 indicates that the prefix is 64 bits in length. The address is divided into a prefix and an interface identifier.

- 2001:db8:: is the 64-bit prefix (this is the documentation prefix).
- a00:20ff:feb5:4137 is the 64-bit interface identifier.

Unicast Addressing

- A unicast packet is sent to a single interface with the matching destination IPv6 address.
- Two types of unicast addresses are:
 - Link-local unicast addresses are used on a local network link.
 - Global unicast addresses (also called aggregatable global unicast addresses) are routable across the Internet.



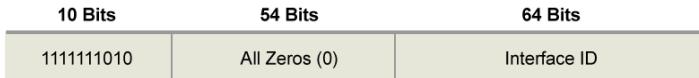
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

With the unicast address type, a unique address is assigned to an interface. A unicast packet is sent to a single node with the matching destination IPv6 address.

Two kinds of unicast addresses are global unicast (global scope) and link-local unicast (local scope).

Link-Local Unicast Address Layout

- The following represents the link-local unicast address layout:



- The first 10 bits of the address prefix identify an address as a link-local address:
 - 1111 1110 10 in binary
 - fe8 in hexadecimal
- The remainder of the prefix is padded with 54 bits of zeros and is followed by the 64-bit Interface ID.
- Example (compressed):
`fe80::a00:20ff:feb5:4137`
- The prefix is fe80:: and the Interface ID is a00:20ff:feb5:4137.
- Link-local addresses are not forwarded by routers.

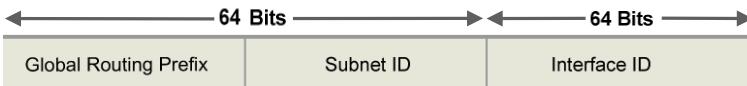
ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

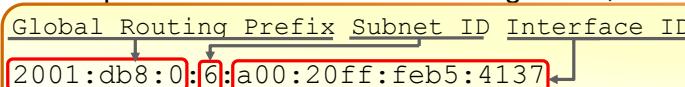
The first 10 bits of a link-local address prefix identify an address as a link-local address. The 10 high-order bits are 1111 1110 10 in binary or fe8 in hexadecimal, as shown in the slide.

Global Unicast Address Layout

- The following represents the global unicast address layout:



- The first three bits are always set to 001 (binary), 0x2 (hexadecimal)
- Global routing prefix—size is variable.
 - Value is assigned by an address registry or ISP to identify a site.
- Subnet ID—size is dependent on the size of the Global Routing Prefix.
 - Identifier of a subnet or link within a site, assigned by a system administrator
- Global routing prefix size + subnet ID size = 64 bits
- Interface ID—uniquely identifies an interface and is 64 bits in size
- Example with a 48-bit Global Routing Prefix, 16-bit Subnet ID, and 64-bit Interface



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The format of a global unicast address includes the following:

- Global routing prefix: A value assigned by an address registry or ISP to identify a site. Its size is variable.
- Subnet ID: An identifier of a subnet or link within a site. The number of bits in this field depends on the size of the global routing prefix.
- Interface ID: The 64-bit portion of the IP address that identifies the interface.
- The number of bits within the Global Routing Prefix and Subnet ID parts of the address can vary, but they should add up to 64 bits total—the total size of the prefix (Global Routing Prefix + Subnet ID).

The example shows a 48-bit Global Routing Prefix of 2001:db8:0. Note that this is the documentation prefix and is used for purposes of the example. A routable prefix would be different. The Subnet ID is the 16-bit hexadecimal number 0x6 (shortened from 0006), giving an overall prefix length of 64 bits. The Interface ID is the 64-bit hexadecimal number a00:20ff:feb5:4137. Note that the leading zero of the first hexet in the interface ID is omitted, so 0a00 is presented as a00.

Multicast Addressing

- A packet that is addressed to a multicast address is delivered to all the nodes that belong to the multicast group.
- It allows many systems to be members of a multicast group, which is represented by a single multicast address.
- The multicast address is a destination address and has a scope:
 - Source address is unicast.
 - Scope can range from local to global.
- Multicast addressing is also part of IPv4.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The multicast address type allows many systems to be members of a multicast group, which is represented by a single multicast address. An IPv6 multicast address can be thought of as a single identifier for a group of IPv6 systems that belong to the multicast group.

The scope setting allows further restriction of how many interfaces will receive a multicast packet. Multicast addressing is also part of IPv4.

Multicast Address Layout

- The following represents the multicast address layout:

Flags 8 Bits	Scope 4 Bits	Multicast Group ID 112 Bits
11111111	ORPT	XXXX

- The 8 high-order bits of a multicast address are:
 - 11111111 in binary
 - ff in hexadecimal
- A fourth flag bit ("T") of zero means the address is permanent or well known—assigned by the Internet Assigned Numbers Authority (IANA).
- Scope bits define the reach of the address, for example, `0x2` means link-local scope.
- Example of an all-routers multicast address with link-local scope:

`ff02::2`

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The low-order 112 bits in an IPv6 address identify the multicast group to which the packet belongs. A single interface can have multiple IPv6 addresses assigned to it, and it may also join multiple IPv6 multicast groups.

The 8 high-order bits of `11111111` or `ff` in hexadecimal format in an address identify the address as being a multicast address.

Multicast addresses include 4 1-bit flags after the initial `ff` in the prefix. The first flag is reserved and set to zero. The "R" flag is described in RFC 3956, the "P" flag is described in RFC 3306, and the "T" flag is described in RFC 4291. The "T" flag bit is set to 0 if a well-known IANA-assigned multicast address is used; it is set to 1 if a multicast address that is not permanent is used. See RFC 2375 ("IPv6 Multicast Address Assignments") for lists of well-known addresses. See RFC 4291 for greater details about multicast addresses, including a list of scope values/definitions.

Anycast Addressing

- An address is assigned to a group of interfaces.
 - These interfaces share the same anycast address.
- Packets are delivered to the nearest interface member, instead of being delivered to all members of a group.
 - Routers determine the "nearest" interface.
- Intent is to provide desired access to a service hosted on multiple nodes.
- Some potential benefits:
 - Help balance load
 - Provide quicker response time



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

With the anycast address type, an address is assigned to a group of systems. Anycast addresses identify the nearest member of a group of systems that provide a particular type of service. Packets are delivered to the nearest interface member, as identified by the routing protocol, instead of being delivered to all members of a group.

Anycast addresses use the global unicast address space. The address type portion of the prefix does not identify these addresses separately from unicast addresses.

Neighbor Discovery Protocol (NDP)

- Includes five ICMPv6 message types:
 - Router Solicitation
 - Router Advertisement
 - Neighbor Solicitation
 - Neighbor Advertisement
 - Redirect
- "Solicitation"—request for information
- "Advertisement"—response or sent independently of a solicitation
- Some Neighbor Solicitation/Advertisement purposes:
 - Discover a link-layer address, determine whether a neighbor is reachable, determine whether an address is a duplicate (Duplicate Address Detection—DAD)
- Some Router Solicitation/Advertisement information provided:
 - Routable prefixes for address autoconfiguration, default gateway, hop limit
- Redirect—routers let nodes know of a better first hop to a destination

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Neighbor Discovery Protocol (NDP) for IPv6 includes a set of ICMPv6 messages that serve a variety of functions and provide information for nodes. The NDP communication between nodes (any device using IPv6) is all on the same local link. Selecting one comparison, determining a link-layer (MAC) address using Neighbor Solicitation/Neighbor Advertisements in IPv6 corresponds to Address Resolution Protocol (ARP) in IPv4.

RFC 4861 describes NDP.

Obtaining IPv6 Addresses

- Two general methods:
 - Manually—configure them yourself
 - Automatically with autoconfiguration
- Autoconfiguration means automatically creating or obtaining routable IPv6 addresses for network interfaces.
- Types of autoconfiguration:
 - Stateless Address Autoconfiguration (SLAAC)
 - Stateless DHCPv6—uses SLAAC for IP address and DHCPv6 for other information
 - Stateful DHCPv6—like IPv4 DHCP



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6 addresses can be obtained by either creating them manually or via autoconfiguration.
IPv6 provides three methods of autoconfiguration as seen on the slide.

Stateless Address Autoconfiguration—SLAAC

- New in IPv6
- SLAAC process begins when an interface is enabled.
- Nodes automatically generate IPv6 addresses:
 - No manual node configuration required
 - No DHCP servers necessary
- Routers do not use this method.
- No state information about addresses is maintained.
- Facilitates current and future device addressing:
 - Consider the volume of devices in the Internet of Things (IoT).



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Stateless Address Autoconfiguration (SLAAC) is a new feature of IPv6. Address configuration is accomplished between the node needing a routable IP address and other nodes and routers on the same local link. DHCPv6 is not involved with this method.

An interface may become enabled in various ways, for example, being brought up on a network for the first time or upon system reboot. The SLAAC process is carried out at the level of specific interfaces that need IPv6 addresses.

SLAAC is very simple and can be used for current addressing needs, as well as for the ongoing proliferation of networked devices, such as with IoT.

RFC 4862 ("IPv6 Stateless Address Autoconfiguration") describes SLAAC.

The SLAAC Process

- An interface creates a unique 64-bit interface ID in modified EUI-64 format or randomly.
- A 128-bit link-local address is generated by combining the interface ID and the 64-bit link-local prefix (`fe80::/10`).
 - At this point, this IPv6 address is considered *tentative*.
- This newly formed address is checked for duplicates on the local link—Duplicate Address Detection (DAD).
 - Neighbor Solicitation ICMPv6 message is sent to Solicited Node multicast address.
 - The system receives a Neighbor Advertisement from any device that is currently using the address.
 - If the address is already in use, autoconfiguration stops and an address must be configured manually.
 - If no reply is received, the link-local address is presumed to be unique on the local link and is assigned to the interface—the address is then *preferred*.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

With SLAAC, globally routable addresses are created from a unique interface ID combined with a routable prefix provided by a router. In order to get to that point, an interface ID must be generated and a link-local address created. The link-local address is *tentative* at this stage.

Duplicate address detection (DAD) is conducted with the newly formed link-local address before it is assigned to an interface as described on the slide. If a response is received, the address is in use and cannot be used. In this case, autoconfiguration cannot continue and a unique address must be configured manually. If no response is received, the node assumes that the address is available for use and assigns the address to the interface. The address is then considered *preferred*.

See RFC 4862 for other address states.

The SLAAC Process

- A Router Solicitation ICMPv6 message is sent to the all-routers multicast group to obtain a routable prefix.
- Routers respond with ICMPv6 Router Advertisements that advertise routable prefixes:
 - Router Advertisement has "Autonomous address-configuration" flag (A flag) set to 1—tells the node to use SLAAC.
- The node forms a globally routable address by combining an advertised prefix with the unique interface identifier.

ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Once the initial link-local address is verified as unique via DAD and assigned to the interface, routers on the local link are contacted in order to obtain routable prefixes. Globally routable addresses are created by combining routable prefixes with the interface ID.

Communication is conducted via ICMPv6 messages in order to accomplish the steps outlined in the SLAAC process.

If there are no routers, nodes can use link-local addresses to communicate on the local link only.

See RFC 4862 for more details on SLAAC.

Interface ID—Modified EUI-64 Format Created from MAC Addresses

- Example of a 64-bit modified EUI-64 interface ID created from a 48-bit MAC address
- Example 48-bit MAC address in hexadecimal format:

08:00:20:b5:41:37

- MAC address in binary format:

0	8	0	0	2	0	b	5	4	1	3	7
0000	1000	0000	0000	0010	0000	1011	0101	0100	0001	0011	0111

- Two octets, 0xff and 0xfe, are inserted between the third and fourth octets:

0	8	0	0	2	0	f	f	f	e	b	5	4	1	3	7
0000	1000	0000	0000	0010	0000	1111	1111	1111	1110	1011	0101	0100	0001	0011	0111

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Modified 64-bit Extended Unique Identifier (modified EUI-64) formed from a 48-bit MAC address is one way SLAAC obtains an interface ID. Another method is by the interface ID being randomly generated. This example shows how a modified EUI-64 interface ID is created from a MAC address.

Appendix A of RFC 4291 describes the process of creating IPv6 modified EUI-64 interface identifiers.

The slide shows the first part of creating a modified EUI-64 interface ID using a MAC address.

The initial MAC address is 08:00:20:b5:41:37.

First, the octets 0xff and 0xfe are inserted between the first three and last three octets of the MAC address.

Interface Identifier Creation—Modified EUI-64 Format

- Bit 7 (from the left), the universal/local bit, is toggled. This completes the process:

0	a	0	0	2	0	f	f	f	e	b	5	4	1	3	7
0000	1010	0000	0000	0010	0000	1111	1111	1111	1110	1011	0101	0100	0001	0011	0111

- The modified EUI-64 interface ID in IPv6 hexadecimal format:

0a00:20ff:feb5:4137

- This unique interface identifier is 64 bits of the 128-bit IPv6 address. It is combined with a prefix to form a complete IPv6 address.
- Example 128-bit IPv6 address formed from a global unicast (documentation) prefix and the modified EUI-64 interface ID:

2001:0db8:0000:0000:0a00:20ff:feb5:4137

- The same address compressed:

2001:db8::a00:20ff:feb5:4137



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Next, bit 7 (from the left), the universal/local (U/L) bit, is toggled.

The result is a 64-bit interface ID that is then combined with a prefix to form a 128-bit IPv6 address. The example shows an IPv6 global unicast (documentation) address formed from the documentation prefix (2001:db8:/32) and the interface ID.

Stateless DHCPv6 Autoconfiguration

- IPv6 address is obtained via SLAAC.
- Other information is provided by a DHCPv6 server.
 - for example, DNS server address(es)
- Router Advertisement has "Other Configuration" (O flag) set to 1.
 - This tells a node to autoconfigure its IPv6 address
 - Then to request DHCPv6 information that is needed
- No state information about addresses is maintained.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

With stateless DHCPv6, nodes receive IP addresses via SLAAC and other needed information from a DHCPv6 server.

Stateful DHCPv6 Autoconfiguration

- Provides greater control over addressing than SLAAC
 - Might be desired in some enterprise environments
- Requires setup of a DHCPv6 server
- Router advertisement has "Managed address configuration" (M flag) set to 1
 - Tells the node to request address and other information from a DHCPv6 server
- DHCPv6 server supplies address information like DHCP in IPv4
- DHCP Unique Identifier (DUID) used for client and server identification rather than MAC addresses as in DHCP for IPv4
- Can coexist and operate together with stateless autoconfiguration as defined in IPv6



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Stateful autoconfiguration requires the setup of a DHCPv6 server and may be desired in some environments. Stateful autoconfiguration and stateless autoconfiguration, as defined in IPv6, can coexist and operate together. Stateful autoconfiguration supplies address and other information similar to the way that DHCP provides information in IPv4.

A DHCPv6 server maintains state information about addresses in this case.

Other Address Types

- IPv4-mapped IPv6 addresses:
 - Carrying an IPv4 address in an IPv6 packet is called an IPv4-mapped IPv6 address.
 - This format is used for nodes that only support IPv4.
- Unspecified address:
 - The source address of a system that has not had an address assigned will be all zeros - :: in compressed format.
 - It is analogous to the IPv4 address 0.0.0.0.
- Loopback address:
 - ::1 in compressed format
 - This address is analogous to the 127.0.0.1 local address used by IPv4 systems.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv4-mapped IPv6 address

The IPv4-mapped IPv6 address can be used on nodes that only support IPv4.

An example of an IPv4-mapped IPv6 address is:

0000:0000:0000:0000:0000:ffff:yyy.yyy.yyy.yyy

The first 80 bits are all zeros. The next 16 bits are ffff, indicating that an embedded IPv4 address is present, and the final 32 bits are yyy.yyy.yyy.yyy, representing the IPv4 address in dot-decimal format. An example compressed address is ::ffff:10.2.4.16.

Unspecified Address

For example: 0000:0000:0000:0000:0000:0000:0000:0000, 0:0:0:0:0:0:0:0, or :: compressed.

Loopback Address

IPv6 systems use the loopback address of 0000:0000:0000:0000:0000:0000:0000:0001, 0:0:0:0:0:0:1, or ::1 to send packets to themselves. This is ::1/128 in prefix notation.

Using IPv6 with Oracle Linux

- IPv6 is enabled in Oracle Linux 7 by default.
- Check this by looking for `inet6` addresses as follows:

```
# ip addr|grep inet6
inet6 ::1/128 scope host
inet6 fe80::2f49:669e:6cb1:2a1/64 scope link ...
inet6 fe80::d86b:7b28:2646:effb/64 scope link ...
```

- The example shows the IPv6 loopback address of `::1/128`; the scope is `host` - the node itself only.
- It also shows two link-local addresses for two different interfaces:
 - `fe80::` is the link-local prefix; the scope is `link` - the local link only.
 - Prefix length is 64 bits.
 - The interface IDs are `2f49:669e:6cb1:2a1` and `d86b:7b28:2646:effb`.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

IPv6 is enabled by default in Oracle Linux 7. The example shows an IPv6 loopback address and two link-local addresses. Note that the interface IDs of the link-local addresses do not have the modified EUI-64 format. An indication of this is the absence of the hexadecimal characters `ff` and `fe` in the middle of the interface IDs. These characters would be present if a modified EUI-64 address was created using the interface's MAC address. This means these interface IDs were generated from another method.

IPv6 Settings for Specific Interfaces

- IPv6 parameters can be set in ifcfg-<interface> files in /etc/sysconfig/network-scripts/ for specific interfaces, for example:
 - IPV6ADDR=<IPv6 address>[/prefix length]

 - Allows a primary static IPv6 address to be provided
 - 64 is the default prefix length if not specified
 - IPV6_ADDR_GEN_MODE=stable-privacy|eui64
 - Specifies the types of interface IDs used for creating addresses
 - "stable-privacy" enables random creation of interface IDs that do not change within the current subnet (see RFC 7217).
 - "eui64" specifies that interface IDs are configured in modified EUI-64 format from the MAC address
 - Default is eui64



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Shown on the slide are two of the settings that can be used in the ifcfg-<interface> files, found in /etc/sysconfig/network-scripts/.

If needed, you can manually specify static IPv6 addresses.

IPV6_ADDR_GEN_MODE specifies the types of interface IDs used when generating addresses.

The nm-settings-ifcfg-rh(5) man page provides further information on IPv6 settings. Search on "ipv6 setting" to find the table with settings specific to IPv6. You can also find information in /usr/share/doc/initscripts-*/*sysconfig.txt. Search on "IPV6" for details on IPv6 settings in this file. Interface configuration file setting information can also be found here:

https://docs.oracle.com/cd/E52668_01/E54669/html/oI7-about-netconf.html.

Some IPv6 Commands

- Linux commands specific to IPv6 can be specified using the "-6" option. For example:

- Show IPv6 addresses:

```
# ip -6 addr
```

- Ping an IPv6 address:

```
# ping -6 2001:db8::a00:20ff:feb5:4137
```

- To ping an IPv6 link-local address, the interface name must be appended to the IPv6 address with the "%" character:

```
# ping -6 fe80::e87d:a31:aecf:a5a2%eth0
```

- Or the interface name must be specified with the "-I" option:

```
# ping -6I eth0 fe80::e87d:a31:aecf:a5a2
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The "-6" option can be used to specify IPv6 when running Linux commands.

To show only IPv6 address information:

```
# ip -6 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::ba72:373a:52de:9b90/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
...
...
```

To ping a link-local IPv6 address, append the "%" character to the interface name or use the "-I" option. The following shows use of the "%" character, appending the interface name to the IPv6 address:

```
# ping -6 fe80::e87d:a31:aecf:a5a2%eth0
PING fe80::e87d:a31:aecf:a5a2%eth0(fe80::e87d:a31:aecf:a5a2%eth0) 56
data bytes
64 bytes from fe80::e87d:a31:aecf:a5a2%eth0: icmp_seq=1 ttl=64
time=0.178 ms
...
...
```

Quiz



IPv6 supports which of the following address types? (Select all that apply)

- a. Unicast
- b. Multicast
- c. Anycast
- d. Broadcast



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a, b, c

Quiz



A unicast packet is sent to a single interface.

- a. True
- b. False



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz



Which of the following are the three high order bits of a Global Unicast Address?

- a. 100
- b. 010
- c. 001
- d. 000



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: c

Summary

In this lesson, you should have learned how to:

- Explain the background for IPv6 development
- Describe some features of IPv6
- Explain the IPv6 address format
- List and describe IPv6 address types
- Describe aspects of IPv6 link-local unicast, global unicast, and multicast address layouts
- List the three types of autoconfiguration for obtaining IPv6 addresses
- Explain Stateless Address Autoconfiguration (SLAAC)
- Explain two IPv6 settings for specific interfaces
- Describe some IPv6 commands



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practices for Lesson 14: Overview

This practice covers using IPv6.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

OpenSSH

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

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
- Describe the use of key pairs for Oracle Cloud Infrastructure instances
- Use `ssh-agent` and `ssh-add`



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

OpenSSH: Introduction

OpenSSH:

- Is a suite of secure network connectivity tools:
 - **ssh**: Secure shell command
 - **scp**: Secure copy command
 - **sftp**: Secure file transfer protocol (FTP) command
 - **sshd**: OpenSSH daemon
 - **ssh-keygen**: Creates ECDSA or RSA authentication keys
- Is a secure alternative to **telnet**, **rcp**, **rsh**, **rlogin**, and **ftp**
- Encrypts all communication between the client and server
- Supports the SSH2 protocol on client and server
 - SSH1 removed from server side as of Oracle Linux 7 Update 4
- Provides X11 forwarding and port forwarding



ORACLE

Copyright © 2019, 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**: OpenSSH daemon
- **ssh-keygen**: Creates ECDSA or RSA host/user authentication keys:
 - ECDSA (Elliptic Curve 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 by using a key known only by the local and remote systems.

OpenSSH supports SSH protocol version 2 (SSH2) on the client and server sides. As of Oracle Linux 7 Update 4, support for the SSH1 protocol is removed from the OpenSSH server side. This is because SSH1 is considered vulnerable and insecure. SSH clients are provided with Oracle Linux 7 Update 4 and later can still connect to (legacy) SSH1 servers. SSH1 on the server side is disabled by default in Oracle Linux 5 and subsequent releases.

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 ECDSA and RSA public and private key files
 - PAM configuration file: `/etc/pam.d/sshd`
 - Configuration file for `sshd`: `/etc/sysconfig/sshd`
- User files include:
 - `config`: Overrides global `ssh_config` file
 - `known_hosts`: Contains host keys of SSH servers accessed by the user
 - Various user ECDSA and RSA public and private key files



Copyright © 2019, 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`: Default OpenSSH client configuration file. Entries are overridden by a user's `~/.ssh/config` file.
- `sshd_config`: Configuration file for the `sshd` daemon
- `ssh_host_ecdsa_key`: ECDSA private key used by the `sshd` daemon
- `ssh_host_ecdsa_key.pub`: ECDSA public key used by the `sshd` daemon
- `ssh_host_key`: RSA private key for version SSH1
- `ssh_host_key.pub`: RSA public key for version SSH1
- `ssh_host_rsa_key`: RSA private key for version SSH2
- `ssh_host_rsa_key.pub`: RSA public key for version SSH2

There is also a PAM configuration file for the `sshd` daemon, `/etc/pam.d/sshd`, and a configuration file for the `sshd` service, `/etc/sysconfig/sshd`.

To connect to a legacy SSH1 server from a current SSH client, the client's SSH configuration file, `/etc/ssh/ssh_config`, must specify SSH version 1 in addition to version 2. Example:

```
Host ssh1_server.example.com
  Protocol 2,1
```

~/.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_ecdsa**: ECDSA private key of the user
- **id_ecdsa.pub**: ECDSA public key of the user
- **id_rsa**: RSA private key for version SSH2
- **id_rsa.pub**: RSA public key for version SSH2
- **identity**: RSA private key for version SSH1
- **identity.pub**: 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:

```
# systemctl start sshd
```

 - Configure the service to start at boot time:

```
# systemctl enable sshd
```
- To configure an OpenSSH client:
 - The following packages are installed by default:
 - openssh
 - openssh-clients
 - There are no services to start on the client.



Copyright © 2019, 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:

```
# systemctl start sshd
```

Use the systemctl command to automatically start the sshd service at boot time:

```
# systemctl enable sshd
```

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-clients
```

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 © 2019, 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, do not try to connect to that machine until you have resolved the situation.

Using the ssh Command

- The `ssh` command allows you to:
 - Connect to a remote system
 - 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 © 2019, 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`, enter only the following:

```
$ ssh host03
```

This command attempts to connect to the remote host with the same username that you are logged in 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 © 2019, 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
```

ORACLE

Copyright © 2019, 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 in 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.
- Use the `-t` option to specify the key type. Example:

```
$ ssh-keygen -t rsa
```

- `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:
 - Copy the public key to `~/.ssh` on the remote system.
 - Do one of the following:
 - Rename the public key file `authorized_keys`.
 - Append the public key to the `authorized_keys` file on the remote system to allow connection from multiple clients.



Copyright © 2019, 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:

```
$ 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 `-t` option to specify the type of key to create. Possible values are “`rsa1`” for protocol version 1 and “`dsa`”, “`ecdsa`”, or “`rsa`” for protocol version 2. If the `-t` option is not specified, the key type defaults to `rsa` for protocol 2.

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-keygen` command in the example generated two keys in the `~/.ssh` directory:

```
$ ls ~/.ssh  
id_rsa  
id_rsa.pub
```

To log in 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.

You can use the `ssh-copy-id` script to copy the public key to a remote system, using this form:

```
$ ssh-copy-id user@hostname
```

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
```

`ssh-copy-id` will append keys to the end of the `authorized_keys` file, if it is already present on the system.

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.

Key Pairs for Oracle Cloud Infrastructure Instances

- Generated key pairs are required for Oracle Linux instances.
 - Authenticates remote users instead of using passwords.
- Public keys are provided when launching instances.
- Private keys are specified when connecting to instances with SSH.
- For example, to log in with a private key contained in the `id_rsa` file in the `/root/.ssh` directory:

```
# ssh -i ~/.ssh/id_rsa opc@<public-ip-address>
```

- Oracle Linux images use the default user name `opc`.
- You supply the public IP address of your instance.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Using ssh-agent

- ssh-agent is an authentication agent that handles passwords for SSH private keys.
 - Use ssh-keygen to generate authentication key pairs.
 - Provide a passphrase, for example “password”, when creating the key pairs.
 - Copy the public key to `~/.ssh/authorized_keys` on the remote system.
- 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 © 2019, 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 is forgotten when you log out.

Quiz



OpenSSH connectivity tools encrypt all network traffic, including passwords.

- a. True
- b. False



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a

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
- Describe the use of key pairs for Oracle Cloud Infrastructure instances
- Use `ssh-agent` and `ssh-add`



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practices for Lesson 15: Overview

This practice covers the following topics:

- Connecting to a remote server by using ssh
- Configuring OpenSSH to connect without a password



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Security Administration

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the `chroot` jail
- Use the `chroot` utility
- Describe packet-filtering firewalls
- Describe `firewalld`
- Configure `firewalld` packet filters
- Describe `iptables`
- Configure `iptables` packet filters
- Describe TCP wrappers
- Configure TCP wrappers
- Describe security control in Oracle Cloud Infrastructure



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Several methods of securing your computer system are covered in this lesson.

chroot Jail: Overview

- 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 it is running in 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
 - 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 © 2019, 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 be 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 it is running in 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 [OPTION] NEWROOT [COMMAND [ARG] . . .]`
- The `NEWROOT` directory becomes the artificial root.
- `chroot` changes to `NEWROOT` and runs the optional command.
 - Alternatively, it runs the `SHELL` variable if the command is omitted.
- Assuming the `NEWROOT` directory exists, the following command fails unless the necessary files are copied into the `NEWROOT` directory before running `chroot`:

```
# chroot /home/oracle/jail  
chroot: failed to run command '/bin/bash': No such file or directory
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

To use a `chroot` jail, use the following command (`NEWROOT` must be an existing directory):

```
# chroot [OPTION] NEWROOT [COMMAND [ARG] . . . ]
```

The `NEWROOT` directory becomes the artificial root directory. `chroot` changes to `NEWROOT` and runs the optional command. Without specifying a command as an argument, `chroot` changes to `NEWROOT` 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
```



Copyright © 2019, 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 => (0x0000...)  
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x0000...)  
libdl.so.2 => /lib64/libdl.so.2 (0x0000...)  
libc.so.6 => /lib64/libc.so.6 (0x0000...)  
/lib64/ld-linux-x86-64.so.2 (0x0000...)
```

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-4.2#
```

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-4.2# 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-4.2# ls
bash: ls: command not found
```

Use the `exit` command to exit the `chroot` jail.

```
bash-4.2# exit
exit
#
```

Running Services in a chroot Jail

- DNS and FTP include 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 © 2019, 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 or her home directory.

Packet-Filtering Firewalls: Introduction

- Packet filtering firewalls accept or deny network packets.
- The Linux kernel has built-in packet filtering functionality.
 - Netfilter is the kernel component that stores filtering rules.
- Two services are available in Oracle Linux 7 to create, maintain, and display the rules stored by Netfilter:
 - `firewalld`
 - `iptables`
- The default firewall service in Oracle Linux 7 is `firewalld`.
- `firewalld` offers several advantages over `iptables`:
 - Changes do not require restart of `firewalld` service.
 - Networks can be separated into different zones based on the level of trust.



ORACLE

Copyright © 2019, 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 called Netfilter. Netfilter consists of a set of tables that store rules that the kernel uses to control network packet filtering. Oracle Linux provides the `firewalld` service and the `iptables` services to manage the rules stored by Netfilter.

In Oracle Linux 7, the default firewall service is `firewalld`. You can configure `firewalld` by using the `firewall-cmd` command-line interface. You can also use the `firewall-config` GUI to configure `firewalld`.

The `firewalld`-based firewall has the following advantages over `iptables`:

- Unlike the `iptables` command, the `firewall-cmd` command does not restart the firewall and disrupt established TCP connections.
- `firewalld` supports dynamic zones. Zones are discussed in subsequent slides.
- `firewalld` supports D-Bus for better integration with services that depend on firewall configuration.

firewalld: Introduction

- Firewalld:
 - Is a dynamic firewall manager for Oracle Linux 7
 - Supports firewall (network) zones
 - Supports IPv4, IPv6, and Ethernet bridges
 - Provides a D-BUS interface
 - Provides two configuration modes:
 - **Runtime:** Configuration changes are immediate.
 - **Permanent:** Changes are written to configuration files and are applied when the firewalld service restarts.
- Configuration files exist in two directories:
 - /usr/lib/firewalld: Contains default configuration files.
Do not make changes to these files.
 - /etc/firewalld: Configuration changes are written to files in this directory.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The firewalld service is a dynamic firewall manager for Oracle Linux 7. It provides support for network zones that allow you to define trusted services for specific network connections. Trusted services are network services, such as ssh and dhcp, which are accessible from other systems. The firewalld service has support for IPv4, IPv6, and Ethernet bridges.

The firewalld service also provides a D-BUS interface. Services or applications already using D-BUS can add or request changes to firewall rules directly through the D-BUS interface. Refer to the firewalld.dbus(5) man page for more information.

The firewalld service has two types of configuration options:

- **Runtime:** Changes to firewall settings take effect immediately but are not permanent. Changes made in runtime configuration mode are lost when the firewalld service is restarted.
- **Permanent:** Changes to firewall settings are written to configuration files. These changes are applied when the firewalld service restarts.

Configuration files for firewalld exist in two directories:

- /usr/lib/firewalld: Contains default configuration files. Do not make changes to these files. An upgrade of the firewalld package overwrites this directory.
- /etc/firewalld: Changes to the default configuration files are stored in this directory. Files in this directory overload the default configuration files.

firewalld Zones

A `firewalld` zone defines the following firewall features:

- **Services:** Predefined or custom services that are trusted
- **Ports:** Additional ports and associated protocols to trust
- **Protocols:** Allow access for specified protocols
- **Source Ports:** Allow access for specific source ports/port ranges by protocol
- **Masquerading:** Translates IPv4 addresses to a single external address
- **Port Forwarding:** Forward inbound network traffic to an alternative port or IPv4 address
- **ICMP Filter:** Blocks selected ICMP messages
- **Rich Rules:** Extend existing `firewalld` rules to include additional source and destination addresses and logging and auditing actions.
- **Interfaces:** Network interfaces bound to a zone. The zone for an interface is specified with `ZONE=<zone>` in the `/etc/sysconfig/network-scripts/ifcfg` file. If the option is missing, the interface is bound to the default zone.
- **Sources:** Provide source addresses or areas bound to a zone



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `firewalld` service allows you to separate networks into different zones based on the level of trust you want to place on the devices and traffic within a specific network. For each zone, you can define the following features:

- **Services:** Predefined or custom services to trust. Trusted services are a combination of ports and protocols that are accessible from other systems and networks
- **Ports:** Additional ports or port ranges and associated protocols that are accessible from other systems and networks
- **Protocols:** Allow access for specified protocols
- **Source Ports:** Allow access for specific ports/port ranges by protocol
- **Masquerading:** Translates IPv4 addresses to a single external address. With masquerading enabled, addresses of a private network are mapped to and hidden behind a public address.
- **Port Forwarding:** Forward inbound network traffic from a specific port or port range to an alternative port on the local system or to a port on another IPv4 address
- **ICMP Filter:** Blocks selected Internet Control Message Protocol messages
- **Rich Rules:** Extend existing `firewalld` rules to include additional source and destination addresses and logging and auditing actions
- **Interfaces:** Network interfaces bound to a zone. The zone for an interface is specified with the `ZONE=option` in the `/etc/sysconfig/network-scripts/ifcfg` file. If the option is missing, the interface is bound to the default zone.
- **Sources:** Provide source addresses or areas bound to a zone

Predefined firewalld Zones

- The firewalld software package includes a set of predefined network zones in the following directory:

```
# ls /usr/lib/firewalld/zones/
block.xml  drop.xml   home.xml  public.xml  work.xml
dmz.xml    external.xml internal.xml trusted.xml
```

- The zone files contain preset settings, which can be applied to a network interface. For example:

```
# grep -i service /usr/lib/firewalld/zones/public.xml
<service name="ssh"/>
<service name="dhcpcv6-client"/>
```

- In this example, network interfaces bound to the public zone trust only two services, ssh and dhcpcv6-client.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The firewalld package includes a set of predefined network zones. Zone settings for each zone are stored in the following directory:

```
# ls /usr/lib/firewalld/zones/
block.xml  drop.xml   home.xml  public.xml  work.xml
dmz.xml    external.xml internal.xml trusted.xml
```

The zone files contain a description and preset settings, which can be applied to a network interface. Example:

```
# cat /usr/lib/firewalld/zones/public.xml
...
<description>For use in public areas. You do not trust the other
computers on networks to not harm your computer. Only selected
incoming connections are accepted.</description>
<service name="ssh"/>
<service name="dhcpcv6-client"/>
```

In this example, all network interfaces bound to the public zone trust only two services, ssh and dhcpcv6-client.

A brief explanation of each zone is as follows:

- **drop**: Any incoming network packets are dropped and there is no reply. Only outgoing network connections are possible.
- **block**: Any incoming network connections are rejected with an `icmp-host-prohibited` message for IPv4 and `icmp6-adm-prohibited` message for IPv6. Only network connections initiated from within the system are possible.
- **home**: For use in home areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
- **public**: For use in public areas. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
- **work**: For use in work areas. You mostly trust the other computers on networks to not harm your computer. Only selected incoming connections are accepted.
- **dmz**: For computers in your demilitarized zone that are publicly accessible with limited access to your internal network. Only selected incoming connections are accepted.
- **external**: For use on external networks with masquerading enabled especially for routers. You do not trust the other computers on the network to not harm your computer. Only selected incoming connections are accepted.
- **internal**: For use on internal networks. You mostly trust the other computers on the networks to not harm your computer. Only selected incoming connections are accepted.
- **trusted**: All network connections are accepted.

Setting the Default firewalld Zone

- After an initial installation, the `public` zone is the default zone as specified in the configuration file, `/etc/firewalld/firewalld.conf`.

```
# grep -i defaultzone /etc/firewalld/firewalld.conf
DefaultZone=public
```

- Network interfaces are bound to the default zone unless specified with `ZONE=<zone>` in the `ifcfg` file.
- You can use the `firewall-cmd` command to change the default zone:

```
# firewall-cmd --set-default-zone=work
success
```

- You can also use the `firewall-config` GUI to change the default zone.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `public` zone is initially defined as the default zone, but this can be changed by editing the `/etc/firewalld/firewalld.conf` file. Network connections are bound to the default zone unless the zone is specified in the `ifcfg` file with the `ZONE=<zone>` option.

The following command shows the interfaces that are bound to the `public` zone:

```
# firewall-cmd --get-active-zone
public
    interfaces: eth0 eth1
```

You can use the `firewall-cmd` command to change the default zone. The following sequence of commands displays the default zone, then changes the default zone to the `work` zone.

```
# firewall-cmd --get-default-zone
public
# firewall-cmd --set-default-zone=work
success
```

You can also use the `firewall-config` GUI to change the default zone. From the menu bar, select **Options > Change Default Zone** and then select a zone from a pop-up list.

firewalld Services

- A `firewalld` service is a combination of local ports, protocols, modules, and destination addresses.
- A `firewalld` service can also include Netfilter kernel modules that are automatically loaded when a service is enabled.
- The `firewalld` software package includes a set of predefined services in the following directory:

```
# ls /usr/lib/firewalld/services/
amanda-client.xml ipp-client.xml mysql.xml rpc-bind.xml
bacula-client.xml ipp.xml      nfs.xml    samba-client.xml
...
```

- Services can be enabled for a zone in runtime mode.
- Service definitions can be edited only in permanent mode.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

A `firewalld` service is a combination of local ports, protocols, modules, and destination addresses. For each service, you can limit network traffic to a particular destination address and Internet Protocol (IPv4 or IPv6). A service can also include Netfilter kernel modules that are automatically loaded when a service is enabled.

Trusted services are accessible from all hosts and networks. You can choose to trust a service or choose not to trust a service for a selected zone at any time. In **runtime** configuration mode, changes are implemented immediately without the need to restart the `firewalld` service or to disrupt existing network connections and services.

The `firewalld` package includes a set of predefined services. Configuration files for these services are stored in the following directory:

```
# ls /usr/lib/firewalld/services/
amanda-client.xml ipp-client.xml mysql.xml rpc-bind.xml
bacula-client.xml ipp.xml      nfs.xml    samba-client.xml
bacula.xml        ipsec.xml     ntp.xml    samba.xml
...
...
```

Service definition settings can be changed only in the **permanent** configuration mode.

The service files contain a description and preset settings. Example:

```
# cat /usr/lib/firewalld/services/samba.xml
...
<description>This option allows you to access and participate in
Windows file and printer sharing networks. You need the samba
package installed for this option to be useful.</description>
<port protocol="udp" port="137"/>
<port protocol="udp" port="138"/>
<port protocol="tcp" port="139"/>
<port protocol="tcp" port="445"/>
<module name="nf_conntrack_netbios_ns"/>
...
```

Starting firewalld

- To start firewalld:

```
# systemctl start firewalld
```
- To ensure firewalld starts at boot time:

```
# systemctl enable firewalld
```
- To check if firewalld is running:

```
# systemctl status firewalld
# firewall-cmd --state
```
- Three methods to configure the firewalld service:
 - firewall-cmd: Command-line interface
 - firewall-config: Graphical user interface
 - Edit various XML configuration files.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the following command to install the firewalld package and the GUI tool.

```
# yum install firewalld firewall-config
```

To start firewalld, run the following commands as root:

```
# systemctl start firewalld
# systemctl enable firewalld
```

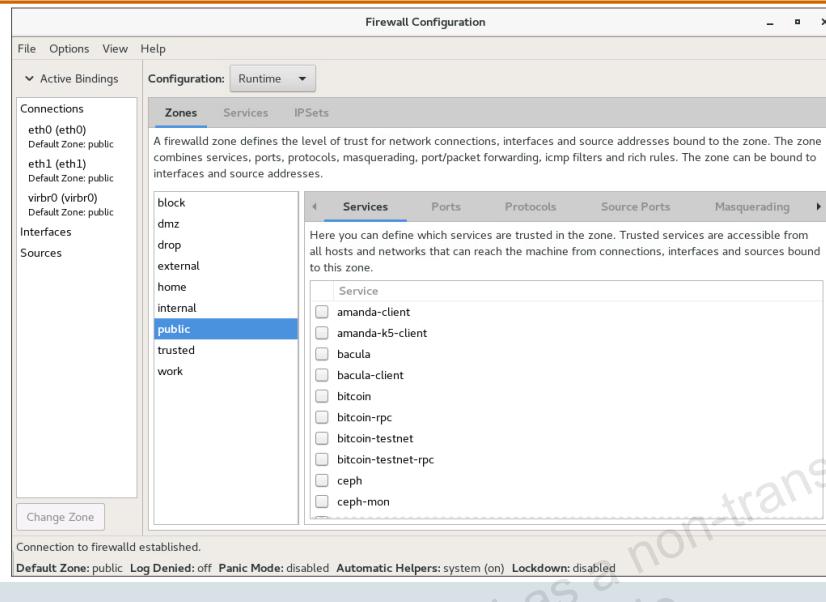
Use either of the following commands to check if firewalld is running.

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service...)
   Active: active (running) since ...
             ...
# firewall-cmd --state
running
```

The firewalld service can be configured by using the firewall-config GUI, by using the firewall-cmd command-line interface, and by editing the various XML configuration files.

firewalld Configuration Tool

```
# firewall-config
```



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The slide shows the Firewall Configuration GUI, which can be used to configure `firewalld`. Enter the following command to start the GUI:

```
# firewall-config
```

The message “Connection to firewalld established” in the lower left corner indicates that the `firewalld` service is running. Start `firewalld` if “Trying to connect to firewalld, waiting...” appears in the lower left corner and in a pop-up message.

The Configuration drop-down menu offers two options:

- **Runtime:** Changes to current firewall settings take effect immediately. Changes are not permanent and are lost when the `firewalld` service restarts.
- **Permanent:** Changes are written to configuration files and are applied when the `firewalld` service restarts. You can restart the `firewalld` from the GUI by selecting **Options > Reload Firewalld** from the menu bar. If you select **Permanent**, a row of icons and/or buttons appears below, allowing you to add, edit, remove, or load defaults relative to the configuration item selected above.

The GUI has three main tabs visible by default—Zones, Services, and IPSets—which allow you to configure different firewall characteristics. More tabs can be made visible by selecting desired check boxes from the drop-down list of the View menu item. Selecting each tab displays a short description along with a set of tabs below it relating to the selected main tabs above. The slide shows the Zones tab selected with its related categories (Services, Ports, Protocols, Source Ports...), along with a short description of a zone. Active Bindings are shown on the left. The bottom left shows the **Default Zone**, which is `public`.

firewall-cmd Utility

- The command-line interface to configure the `firewalld` service
- To get help on the `firewall-cmd` command:

```
# firewall-cmd --help
```

- To list information for all zones:

```
# firewall-cmd --list-all-zones
```

- To permit access by HTTP clients for the `public` zone:

```
# firewall-cmd --zone=public --add-service=http
```

- To permanently permit access by HTTP clients for the `public` zone:

```
# firewall-cmd --permanent --zone=public --add-service=http
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The command-line tool `firewall-cmd` is part of the `firewalld` application, which is installed by default. It can be used to make permanent and nonpermanent runtime changes.

Enter the following command to view the help output.

```
# firewall-cmd --help
```

```
...
```

The `firewall-cmd` command offers categories of options such as General, Status, Permanent, Zone, IcmpType, Service, Adapt and Query Zones, Direct, Lockdown, Lockdown Whitelist, and Panic. Refer to the `firewall-cmd(1)` man page for more information.

Some examples are given. Use the following command to list information for all zones. Only partial output is displayed.

```
# firewall-cmd --list-all-zones
...
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: eth0 eth1
  sources:
  services: ssh dhcpcv6-client
  ports:
  ...
...
```

To permit access by HTTP clients for the `public` zone:

```
# firewall-cmd --zone=public --add-service=http  
success
```

To list services that are allowed for the public zone:

```
# firewall-cmd --zone=work --list-services  
dhcpv6-client http ssh
```

Using this command changes only the `runtime` configuration and does not update the configuration files.

The following sequence of commands shows that configuration changes made in `runtime` configuration mode are lost when the `firewalld` service is restarted:

```
# systemctl restart firewalld  
# firewall-cmd --zone=work --list-services  
dhcpv6-client ssh
```

To make changes permanent, use the `--permanent` option. Example:

```
# firewall-cmd --permanent --zone=work --add-service=http  
success
```

Changes made in `permanent` configuration mode are not implemented immediately. Example:

```
# firewall-cmd --zone=work --list-services  
dhcpv6-client ssh
```

However, changes made in `permanent` configuration are written to configuration files.

Restarting the `firewalld` service reads the configuration files and implements the changes.

Example:

```
# systemctl restart firewalld  
# firewall-cmd --zone=work --list-services  
dhcpv6-client http ssh
```

iptables: Introduction

- Another firewall mechanism available in Oracle Linux 7
- Firewall configuration information is stored in the `/etc/sysconfig/` directory:
 - `iptables` file is used for IPv4 configuration.
 - `ip6tables` file is used for IPv6 configuration.
- Every configuration change flushes old firewall rules and reloads new firewall rules.
- Stop the `firewalld` service before using `iptables`.
- Install the `iptables-services` package before attempting to start `iptables` services.
 - `iptables` and `ip6tables` are found here.
- Use the `iptables` command-line utility to create firewall configuration rules.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `iptables` firewall mechanism is also available in Oracle Linux 7. Firewall configuration information is stored in the `/etc/sysconfig/` directory.

- IPv4 information is stored in the `/etc/sysconfig/iptables` file.
- IPv6 information is stored in the `/etc/sysconfig/ip6tables` file.

With the `iptables` service, every configuration change flushes all the old firewall rules and reads all the new rules from the configuration file.

In Oracle Linux 7, the `firewalld` service is enabled by default and the `iptables` and `ip6tables` services are disabled. To use the `iptables` service, you must first stop and disable the `firewalld` service. Use the following command to stop the `firewalld` service:

```
# systemctl stop firewalld
```

Use the following command to disable the `firewalld` service so `firewalld` does not start when your system boots:

```
# systemctl disable firewalld
```

Install the `iptables-services` package, in which `iptables` and `ip6tables` are found:

```
# yum install iptables-services
```

This is necessary before either of these services can be started.

Use the `iptables` command-line interface to configure the `iptables` service. Earlier versions of Oracle Linux included a GUI for configuring `iptables`, but this tool no longer exists in Oracle Linux 7.

iptables Terminology

- The Netfilter component is a set of tables:
 - Filter: Default table
 - NAT: Network Address Translation table
 - Mangle: 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 © 2019, 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:** Default table. This table is used primarily to `DROP` or `ACCEPT` packets based on their content.
- **NAT:** 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:** Continues processing the packet
- **DROP:** Ends 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.

Beginning iptables Maintenance

- To start the service:

```
# systemctl start iptables
```

- To configure iptables to start at boot time:

```
# systemctl enable iptables
```

- 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:

```
# service iptables save
```

ORACLE

Copyright © 2019, 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:

```
# systemctl start iptables
```

To ensure that `iptables` starts at boot time, enter the following command:

```
# systemctl enable iptables
```

Run the same commands for `ip6tables` if you are 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 to the /etc/sysconfig/iptables file so that they are loaded when the iptables service starts, 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 © 2019, 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. rule_specs consist of `-p tcp -m tcp --dport 80`. If information in the packet header matches the rule, the action taken is `ACCEPT`.

`rule_specs` in this example are defined as follows:

- **-p tcp:** Matches if the packet uses the TCP protocol. The 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:** 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 © 2019, 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 :** 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*:** Specifies what to do if the packet matches the rule specification
- **-g, --goto *chain*:** Specifies that the processing continues in a user-specified chain
- **-i, --in-interface *name*:** Matches if the packet came from the *name* interface
- **-o, --out-interface *name*:** Matches if the packet is to be sent to the *name* interface

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 © 2019, 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 *rule_spec* or *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 *rule_spec* or *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 © 2019, Oracle and/or its affiliates. All rights reserved.

Network Address Translation (NAT) is a 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 must 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 the `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.



Copyright © 2019, 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 © 2019, 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 a 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
```



Copyright © 2019, 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.

Security Control in Oracle Cloud Infrastructure

- Private subnets – Disallow public IP addresses for instances in the subnet
- Security lists – Virtual firewalls that control packet level traffic for instances
 - Configured for subnets - all instances in the subnet have the same rules
 - Specify ingress and egress rules for traffic types
- Firewall rules – Control packet level traffic for instances
 - Configured on instances directly
 - Use `firewalld` in Oracle Linux 7
- Gateways and route tables – Control traffic between a Virtual Cloud Network (VCN) and destinations such as the Internet, on-premises network, Oracle Cloud Infrastructure services, and other VCNs
- IAM policies – Control access to the Oracle Cloud Infrastructure API and Console



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

There are multiple ways and levels to secure networks in Oracle Cloud Infrastructure.

Subnets are configured as either public or private. Public subnets allow IP addresses that can access the Internet, and this is the default. Private subnets cannot have public IP addresses and therefore cannot access the Internet.

Both security lists and firewall rules control packet level network traffic for instances, but they are set at different levels. Security lists control traffic for all instances in a given subnet, while firewall rules are set for individual instances. In Oracle Linux 7, `firewalld` is used to set firewall rules.

Gateways and route tables control network traffic at the cloud network level (VCN) to various destinations as seen on the slide.

IAM policies affect access to compartments, which are collections of related resources that require the proper permissions to access. For example, VCNs and Internet Gateways are created within specific compartments. Compartments can be specified for other resources as well.

Quiz



Which of the following has a specific purpose of allowing or denying access to network services?

- a. chroot jail
- b. firewalld
- c. iptables
- d. TCP wrappers



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz



Which of the following statements are true?

- a. The default firewall service in Oracle Linux 7 is firewalld.
- b. When using firewalld, configuration changes do not require restart of firewalld service.
- c. With iptables, networks can be separated into different zones based on the level of trust.
- d. Both the firewalld and the iptables services include a GUI to make configuration changes.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: a, b

Summary

In this lesson, you should have learned how to:

- Describe a `chroot` jail
- Use the `chroot` utility
- Describe packet-filtering firewalls
- Describe `firewalld`
- Configure `firewalld` packet filters
- Describe `iptables`
- Configure `iptables` packet filters
- Describe TCP wrappers
- Configure TCP wrappers
- Describe security control in Oracle Cloud Infrastructure



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practices for Lesson 16: Overview

This practice covers the following topics:

- Configuring a chroot jail
- Configuring a chroot jail for ftp users
- Exploring firewalld
- Configuring firewalld
- Configuring iptables
- Configuring a TCP wrapper
- Restoring VM configurations



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license
to use this Student Guide.

Oracle on Oracle

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Prepare your Oracle Linux server for Oracle Database installation
- Use Oracle Pre-Install RPM
- 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)
- Describe Oracle ASM, ASM Library Driver (ASMLib), and ASM Filter Driver (ASMFD)



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Oracle Pre-Install RPM

Oracle Database Pre-Install RPM for Oracle Linux:

- Completes most pre-installation configuration tasks
- Downloads and installs various software packages and specific versions needed for database installation
- Creates the user `oracle` and the groups `oinstall` and `dba`
- Updates kernel parameters
 - These reside in `/etc/sysctl.d/99-oracle-database-server-18c-preinstall-sysctl.conf`
- Sets hard and soft shell resource limits in the `/etc/security/limits.d` directory
- Sets `numa=off` in the kernel boot parameters for `x86_64` machines
- Sets `transparent_hugepage=never` in the kernel boot parameters



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Oracle Database Pre-install RPM package is designed specifically for Oracle Linux to aid in the installation of the Oracle Database. You can complete most pre-installation configuration tasks by using this package, which is available from the Unbreakable Linux Network or from the Oracle Linux yum repository.

This package was formerly known as `oracle-validated`. For Oracle Linux 6 or newer, the names of packages for different Oracle Database versions are:

- `oracle-rdbms-server-11gR2-preinstall`
- `oracle-rdbms-server-12cR1-preinstall`
- `oracle-database-server-12cR2-preinstall`
- `oracle-database-preinstall-18c`

The pre-install RPM configures an Oracle Linux 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`
- Creates the user `oracle` and the groups `oinstall` and `dba`, which are the defaults used during database installation

The pre-install package also performs the following tasks:

- Updates kernel parameters. These reside in `/etc/sysctl.d/99-oracle-database-server-18c-preinstall-sysctl.conf`.
- Changes 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 or later 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.
- It sets `transparent_hugepage=never` in the kernel boot parameters.

More details on setting kernel parameters for the 18c version can be found at

<https://docs.oracle.com/en/database/oracle/oracle-database/18/ladbi/changing-kernel-parameter-values.html#GUID-FB0CC366-61C9-4AA2-9BE7-233EB6810A31>.

Oracle Software User Accounts

- The 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 a database, and create other `OSDBA` members.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Oracle software installation requires a Linux user to be designated as the Oracle software owner. The Oracle software owner runs the Oracle Universal Installer (OUI) 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 and drop a database and create other DBA privileged users, in addition to the privileges of `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 might 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 Oracle Grid Infrastructure & ASM, there is a user called `grid` and three groups: `asmadmin`, `asmdba`, and `asmoper`. The owner of the Grid Infrastructure is commonly the “grid” user.

Oracle Software Group Accounts

- OSDBA:
 - This is commonly named dba.
 - Members of the OSDBA group have database administration privileges (SYSDBA).
- OSOPER:
 - This is an optional group.
 - 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.



ORACLE

Copyright © 2019, 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.

The following are additional administrative user accounts:

- **SYSBACKUP**: Facilitates Oracle Recovery Manager (RMAN) backup and recovery operations from either RMAN or SQL*Plus
- **SYSDG**: Facilitates Data Guard operations. The user can perform operations either with Data Guard Broker or with the DGMGRL command-line interface.
- **SYSKM**: Facilitates Transparent Data Encryption keystore operations
- **SYSRAC**: Facilitates Oracle Real Application Clusters (Oracle RAC) operations by connecting to the database by the Clusterware agent on behalf of Oracle RAC utilities such as SRVCTL.

Each of these accounts provides a designated user for the new administrative privilege with the same name. See the following for a list of these accounts with links to documents for functions they relate to: <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/tcpsg/securing-the-database-installation-and-configuration.html#GUID-9107985A-AA8A-4A81-A644-863884083AB7>.

Oracle Automatic Storage Management (ASM) Groups

- The OSDBA group for Oracle ASM:
 - This is commonly named asmdba.
 - The Oracle Grid Infrastructure software owner (typically, grid) must be a member.
 - Membership in this group enables access to the files managed by Oracle ASM.
- The OSASM group for Oracle ASM Administration:
 - This is commonly named asmadmin.
 - Create this group as a separate group if you want to have separate administration privileges groups for Oracle ASM and Oracle Database administrators.
- The OSOPER group for Oracle ASM:
 - This is an optional group.
 - This is commonly named asmoper.
 - Create this group if you want a separate group of operating system users to have a limited set of Oracle instance administrative privileges.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Create the following operating system groups if you are installing Oracle Grid Infrastructure:

- The OSDBA group for Oracle ASM (typically, asmdba)
- The OSASM group for Oracle ASM Administration (typically, asmadmin)
- The optional OSOPER group for Oracle ASM (typically, asmoper)

The OSDBA group (typically, asmdba) for Oracle ASM can be the same group used as the OSDBA group for the database, or you can create a separate OSDBA group for Oracle ASM to provide administrative access to Oracle ASM instances. The Oracle Grid Infrastructure software owner (typically, grid) must be a member of this group. Membership in this group enables access to the files managed by Oracle ASM.

Create the OSASM (typically, asmadmin) group as a separate group if you want to have separate administration privileges groups for Oracle ASM and Oracle Database administrators. Members of this group are granted the SYSASM system privileges to administer Oracle ASM. In Oracle documentation, the operating system group whose members are granted SYSASM privileges is called the OSASM group and, in command lines, is referred to as asmadmin.

The OSOPER group (typically, asmoper) is an optional group. Create this group if you want a separate group of operating system users to have a limited set of Oracle instance administrative privileges (the SYSOPER for ASM privilege), including starting up and stopping the Oracle ASM instance. By default, members of this group also have all privileges granted by the SYSOPER for ASM privilege.

Add the oracle user to the asmadmin and asmdba groups:

```
# usermod -aG asmadmin oracle
# usermod -aG asmdba oracle
```

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 © 2019, 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 can have hundreds of processes executing and thousands of files open.

Linux Shared Memory

- Three shared memory-related kernel parameters:
 - SHMMNI: The maximum number of system-wide 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 >= the largest SGA.
- Set shared memory kernel parameters in: /etc/sysctl.d/97-oracle-database-sysctl.conf
- Parameters are viewable in:

```
# ls /proc/sys/kernel/sh*
shmmax     shmall     shmmni
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The following are the memory-related kernel parameters:

- **SHMMNI**: The maximum number of system-wide 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 system global area (SGA).

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.

Set shared memory kernel parameters in /etc/sysctl.d/97-oracle-database-sysctl.conf:

- kernel.shmmni = 4096
- kernel.shmmax = 4398046511104
- kernel.shmall = 1073741824

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.d/97-oracle-database-sysctl.conf`:
 - `semms1`: Maximum number of semaphores per set
 - `semnms`: Total number of semaphores in the system
 - `semopm`: Maximum number of operations per `semop` call
 - `semnni`: Maximum number of semaphore sets



Copyright © 2019, 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.d/97-oracle-database-sysctl.conf` and viewable in `/proc/sys/kernel/sem`. The four parameters are:

- `semms1`: Maximum number of semaphores per set
- `semnms`: 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 `semmni`: 128

Because these parameters are positional, the following illustrates setting the parameters as indicated in `/etc/sysctl.d/97-oracle-database-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
```

ORACLE

Copyright © 2019, 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.

```
# cd /proc/sys/net/core  
# ls *[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.d/97-oracle-database-sysctl.conf`:
`fs.file-max = 6815744`
- View the setting in:

```
# cat /proc/sys/fs/file-max  
6815744
```



Copyright © 2019, 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.d/97-oracle-database-sysctl.conf` and is viewable in `/proc/sys/fs/file-max`:

```
# cat /proc/sys/fs/file-max  
6815744
```

Asynchronous I/O (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.d/97-oracle-database-sysctl.conf:
`fs.aio-max-nr = 1048576`
- View the setting in:

```
# cat /proc/sys/fs/aio-max-nr  
1048576
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Asynchronous I/O (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 © 2019, Oracle and/or its affiliates. All rights reserved.

You must set three limits 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 must 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 one time. 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 © 2019, 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 KB 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 and 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 Automatic Memory Management (AMM) and HugePages are not compatible. You must disable AMM to be able to use HugePages.

Size of a HugePage

HugePages can be used with 32-bit and 64-bit architectures. The kernel version and hardware architecture affect HugePage sizes, which range from 2 MB to 256 MB.

Configuring HugePages

Configuring your Linux OS for HugePages is a delicate process. If you do not configure properly, the system can 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 the appropriate file in `/etc/security/limits.d` 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.d/97-oracle-database-sysctl.conf` and set the `vm.nr_hugepages` parameter.
 - Reboot your system.
- To check: `grep HugePages /proc/meminfo`



Copyright © 2019, 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 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 (11g R2) or the appropriate file in `/etc/security/limits.d` (12c R1 or later). 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 Database pre-install package installed
- Oracle Exadata DB compute nodes

Step 2: Log in 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.d/97-oracle-database-sysctl.conf` file and set the `vm.nr_hugepages` parameter so that it is set properly with each reboot:

```
vm.nr_hugepages = 1496
```

Step 6: Stop all the database instances and reboot the server.

The performed configuration is based on the RAM installed and combined size of SGA of database instances that you are running. If any of the following changes occur, 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) has 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 vary. For a valid configuration, ensure that the `HugePages_Free` value is smaller than `HugePages_Total` and that there are some `HugePages_Rsvd`. The sum of `HugePages_Free` and `HugePages_Rsvd` can be smaller than your total combined SGA as instances allocate pages dynamically and proactively as needed.

Additionally, Oracle recommends disabling Transparent HugePages before starting installation, because they may cause delays in accessing memory that can result in node restarts in Oracle RAC environments or performance issues or delays for Oracle Database single instances. `transparent_hugepage=never` is set in the kernel boot parameters by the Oracle Database pre-install RPM automatically, ensuring that Transparent HugePages is disabled.

Oracle Database Smart Flash Cache (DBSFC)

- DBSFC:
 - Is available for both Oracle Solaris and Oracle Linux customers with the 11g R2, 12c, and 18c databases
 - 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 <https://www.oracle.com/technetwork/articles/servers-storage-admin/smart-flash-cache-oracle-perf-361527.html>.



Copyright © 2019, 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, 12c, 18c databases. 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 <https://www.oracle.com/technetwork/articles/servers-storage-admin/smart-flash-cache-oracle-perf-361527.html> for more information. Note that multiple devices can be used in 12c and 18c without requiring a volume manager when specifying flash cache file names.

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 relevant initialization parameters are `DB_FLASH_CACHE_FILE` and `DB_FLASH_CACHE_SIZE`. These initialization parameters are described here, for Oracle Database 18c:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/admin/managing-memory.html#GUID-AA1EDDB2-203B-43DA-84D5-73FD7DF3ECE2>.

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 © 2019, 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.

Multiple ASM disks should not be configured by using different partitions on the same device, as this causes performance issues.

Grid Installation Owner and ASMOPER

During installation, in the Privileged Operating System Groups window, it is 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 is not 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 by 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 by 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 that use the same storage from accidentally overwriting each other's files.

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 reestablish 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.

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 © 2019, Oracle and/or its affiliates. All rights reserved.

If you intend to use ASM for database storage for Linux, you can use ASMLib 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:

<https://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.

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. The `oracleasmlib` package is available from ULN or the Oracle Technology Network, but not from the Oracle Linux yum server. The `oracleasm` kernel driver is included in UEK. You do not need to install any driver package when using this kernel. The `oracleasm` kernel driver for the 64-bit RHCK (`kmod-oracleasm`) is not built in and must be installed manually. It is available from ULN or the Oracle Linux yum server.

The following Oracle Technology Network webpage describes getting ASMLib from ULN:
<https://www.oracle.com/technetwork/server-storage/linux/uln-095759.html>.

If you do not have access to ULN, you can obtain ASMLib from this Oracle Technology Network webpage—Oracle ASMLib Downloads for Oracle Linux 7:
<https://www.oracle.com/technetwork/server-storage/linux/asmlib/ol7-2352094.html>.

Oracle ASMLib Release Notes for Oracle Linux 7 are available from this Oracle Technology Network webpage: <https://www.oracle.com/technetwork/server-storage/linux/release-notes-092521.html>.

Installation and configuration details are part of the Oracle Database Documentation. For Oracle Database 18c, see: <https://docs.oracle.com/en/database/oracle/oracle-database/18/cwlin/configuring-storage-device-path-persistence-using-oracle-asmlib.html#GUID-6B1DA5DB-2E93-4616-B517-18ABDEE72AE4>.

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 scan for Oracle ASM disks on boot

The user to own the driver interface is the same user that owns the software installation, typically `oracle`. The group to own the driver interface is the group used for DBAs, typically `dba`. You want to scan for Oracle ASM disks on boot.

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
```

Using ASMLib Commands

Some options for the `oracleasm` program:

- **configure:** Configure the ASMLib driver. Options:
 - `-i` - interactive mode; `-e` - enable ASMLib startup on boot; `-d` - disable ASMLib startup on boot.
- **init:** Load and initialize the ASMLib driver.
- **exit:** Stop the ASMLib driver.
- **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:** Check block devices for ASM disks.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

To administer the Automatic Storage Management library driver and disks, use the `oracleasm` program with different options. The following describes some options for the `oracleasm` program:

- **configure:** Use this to configure the Automatic Storage Management library driver. Some options for the `configure` option are: `-i` - interactive mode; `-e` - enable ASMLib startup on boot; `-d` - disable ASMLib startup on boot
- **init:** Load and initialize the ASMLib driver.
- **exit:** Shut down and unload the ASMLib driver.
- **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.

See the `oracleasm(8)` man page and the man pages for specific options listed within the `oracleasm(8)` man page.

ASM Filter Driver (ASMFD)

- Available on Linux systems starting with Oracle Database 12c Release 1
- Is an alternative to ASMLib
- Prevents accidental corruption or deletion of ASM devices
- Provides persistent paths and permissions for storage devices used with ASM
 - No need to update `udev` files with storage device paths and permissions.
- Cannot be run concurrently with ASMLib
 - If ASMLib is installed, it must be deinstalled before running ASMFD.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

For Linux systems, starting with Oracle Database 12c Release 1, there is an ASM filter driver (ASMFD) that prevents accidental corruption or deletion of the ASM devices. Like with ASMLib, ASMFD provides persistent paths and permissions for storage devices used with ASM, eliminating the need for updating `udev` files with storage device paths and permissions.

ASMF D is an alternative to the use of ASMLib. Either ASMF D or ASMLib must be chosen to run with Oracle ASM—they cannot be run concurrently. Refer to the following for more information:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/ladbi/about-oracle-asm-with-oracle-asm-filter-driver-asmd.html#GUID-02BAA12B-51A3-4E05-B1C7-76DD05A94F51>.

Steps to configure Oracle ASMF D are provided at:

<https://docs.oracle.com/en/database/oracle/oracle-database/18/ostmg/administer-filter-driver.html#GUID-BB2B3A64-4B83-4A6D-816C-6472FAF9B27A>

Quiz



Which of the following statements is true regarding ASM?

- a. ASMLib is required to use ASM.
- b. The `oracleasm` kernel driver is included in the Red Hat Compatible Kernel (RHCK).
- c. When using ASMLib, 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.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Answer: c

The `oracleasm` kernel driver is not built into the RHCK as it is with UEK. It must be installed manually and is available on the ULN or Oracle Linux yum server. Example:

```
# yum install kmod-oracleasm
```

For further information, see

Summary

In this lesson, you should have learned how to:

- Prepare your Oracle Linux server for Oracle Database installation
- Use Oracle pre-install RPM
- 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)
- Describe Oracle ASM, ASM Library Driver (ASMLib), and ASM Filter Driver (ASMFD)



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Practices for Lesson 17: Overview

This practice covers the following topics:

- Using `scp` to upload `oracle` packages
- Installing and running Oracle Database Pre-Install
- Preparing disks for ASM use
- Installing and configuring ASMLib
- Reverting changes made to host03



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

System Monitoring and Management

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

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 (OSWbb) tool
- Use OSWatcher Analyzer (OSWbba)
- Describe Oracle Enterprise Manager Cloud Control
- Describe Spacewalk



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

sosreport Utility

- The `sosreport` utility:
 - Collects debugging information about a system
 - Stores the information in a compressed file in `/var/tmp`
- Run the tool as follows:

```
# sosreport
...
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 specified plug-in(s).



Copyright © 2019, 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. This information is stored in a single compressed file in the `/var/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 become part of the file name created by the tool. After the tool completes, you can uncompress the file and view the contents, by running the following commands:

```
# cd /var/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:

```
# ls /var/tmp/sosreport*
boot/      hostname      lsof      root/      uname
chkconfig  installed-rpms lspci     route     uptime
date       ip_addr       mount    sos_commands/  usr/
...
...
```

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:

abrt	Automatic Bug Reporting Tool
acpid	ACPI daemon information
...	

The following plugins are currently disabled:

activemq	inactive	ActiveMQ message broker
apache	inactive	Apache http daemon
...		

The following plugin options are available:

abrt.detailed	off	collect detailed info for every report
boot.all-images	off	collect lsinitrd for all images
...		

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.
- **-a:** Enable all (Boolean) options for all loaded plug-ins.
- **--tmp-dir *DIRECTORY*:** Specify an alternative temporary directory.

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
- 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 © 2019, 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 4.1.12-94.3.9.el7uek.x86_64 (host03.example.com) 10/25/2017
_x86_64_           (1 CPU)

avg-cpu: %user   %nice %system %iowait  %steal   %idle
          0.08    0.00    0.04    0.56    0.61   98.70

Device:    tps   kB_read/s   kB_wrtn/s   kB_read   kB_wrtn
xvda       0.96      6.88      4.91    615926    439527
xvdb       0.01      0.18      0.02     15912     2052
xvdd       0.01      0.18      0.02     15928     2052
scd0       0.01      0.06      0.00      5080      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 CPUs 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
- **kB_read/s**: Amount of data read from the device expressed in number of kilobytes per second
- **kB_wrtn/s**: Amount of data written to the device expressed in number of kilobytes per second
- **kB_read**: Total number of kilobytes read
- **kB_wrtn**: Total number of kilobytes written

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.
- **-m**: Display statistics in megabytes 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.
- Execute `mpstat` continuously at a specific *interval*, up to *count* times.

```
# mpstat -P ALL
```

- For example, to run `mpstat` every 2 seconds for 5 times:

```
# mpstat interval count
```

```
# mpstat 2 5
```



Copyright © 2019, 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 4.1.12-94.3.9.el7uek.x86_64 (host03.example.com) 10/26/2017
_x86_64_          (1 CPU)

07:25:58 AM   CPU    %usr    %nice    %sys %iowait    %irq    %soft
%steal    %guest    %gnice    %idle
07:25:58 AM   all    0.06    0.00    0.03    0.52    0.00    0.00
0.60    0.00    0.00   98.79
```

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 CPUs 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(s) while the hypervisor was servicing another virtual processor
- **%guest**: Percentage of time spent by the CPU(s) to run a virtual processor
- **%qnice**: Percentage of time spent by the CPU(s) to run a niced guest
- **%idle**: Percentage of time that the CPU was (or the CPUs 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`: Number of processes in wait or sleep states
 - `Memory`: Amount of memory free and amount 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
```

ORACLE

Copyright © 2019, 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
```

		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 about 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 about 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 about 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 about 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 I/O
- **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 as follows:

- **-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: Is the sar back-end tool that does the data collection
 - sa1: Is 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 © 2019, 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/sysconfig/sysstat                            /usr/bin/sar
/etc/sysconfig/sysstat.ioconf                     /usr/lib64/sa
/usr/bin/cifsiostat                            /usr/lib64/sa/sa1
/usr/bin/iostat                                /usr/lib64/sa/sa2
/usr/bin/mpstat                                /usr/lib64/sa/sadc
/usr/bin/nfsiostat                            /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 1 1
# 0 * * * * root /usr/lib64/sa/sa1 600 6 &
# Generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib64/sa/sa2 -A
```

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 4.1.12-94.3.9.el7uek.x86_64 (host03.example.com) ... (1 CPU)

12:00:01 AM   CPU   %user   %nice   %system   %iowait   %steal   %idle
12:10:01 AM   all    0.03    0.00    0.03     0.48      0.60    98.87
12:20:01 AM   all    0.02    0.00    0.02     0.47      0.59    98.90
12:30:01 AM   all    0.02    0.00    0.02     0.48      0.60    98.89
...
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 space usage statistics.
- **-w:** Report swapping 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` or `f`: Display the field management screen.
 - `i`: Toggle the display of all tasks or just active tasks.
 - `q`: Quit.



Copyright © 2019, 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 - 08:30:45 up 1 day, 23:12, 2 users, load average: 0.00, 0.01,
0.05

Tasks: 127 total, 2 running, 125 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0/0.3 0[

KiB Mem : 2045360 total, 871460 free, 247752 used, 926148 buff/cache
KiB Swap: 4281340 total, 4281340 free, 0 used. 1729276 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR   S %CPU %MEM     TIME+ COMMAND
        754 root      20   0      0      0      0   S  0.3  0.0   0:00.58
[kworker/0:2]
          1 root      20   0 128172  8288  5452   S  0.0  0.4   0:10.10
/usr/lib/systemd/systemd --swit+
          2 root      20   0      0      0      0   S  0.0  0.0   0:00.01
[kthreadd]
```

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 process ID numbers (PIDs), the user who owns the process, running time, and CPU and memory that the processes use. 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 or f**: Field Management
 - Allows you to select columns to display
 - Allows you to rearrange order of columns
 - Allows you to sort by a specific column
- **O or o**: Allows you to set a filter
- **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

```
Total DISK READ : 0.00 B/s | Total DISK WRITE : 0.00 B/s
Actual DISK READ: 0.00 B/s | Actual DISK WRITE: 0.00 B/s
  TID PRIO USER      DISK READ  DISK WRITE  SWAPIN   IO>    COMMAND
    1 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  systemd --
switched--~tem --deserialize 21
    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]
    6 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [kworker/u30:0]
    7 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [rcu_sched]
    8 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [rcu_bh]
    9 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [rcuos/0]
   10 be/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [rcuob/0]
   11 rt/4 root      0.00 B/s   0.00 B/s  0.00 %  0.00 %  [migration/0]
```



Copyright © 2019, 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.
- `strace <command>` traces the given command.
- `strace -p <PID>` traces the active process given by the process ID (PID).
 - Terminate tracing with CTRL + C.
- `strace -p <PID> -f` traces child processes created by the process being traced with PID.



Copyright © 2019, 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("/usr/bin/ls", ["ls"], /* 26 vars */) = 0
brk(NULL)                               = 0x220c000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7fce5da34000
access("/etc/ld.so.preload", R_OK)        = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=73186, ...}) = 0
mmap(NULL, 73186, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fce5da22000
...
...
```

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 extended information.



Copyright © 2019, 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.

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. Example:

```
# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address      Foreign Address          State
tcp        0      0 host03...        example.com:...      ESTABLISHED
...
...
```

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, inet6, unix, ipx, ax25, netrom, ddp, and econet. Example:

```
# netstat -A unix
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags  Type      State       I-Node Path
unix     5      [ ]    DGRAM          8453    /run/systemd/journal/socket
...
...
```

Some of the other options for netstat are listed:

- **-r or --route:** Display the kernel routing table:

```
# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask        Flags MSS Window irtt Iface
default         example.com   0.0.0.0       UG      0 0          0 eth0
192.0.2.0      0.0.0.0       255.255.255.0 U        0 0          0 eth0
...
```

- **-i or -I=iface:** Display a table of all network interfaces or the specified *iface*:

```
# netstat -I=eth0
Iface      MTU     RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR ...
eth0      1500    14703     0      0 0      7993     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 0.0.0.0:sunrpc  0.0.0.0:*          LISTEN
tcp        0      0 host03.example...  0.0.0.0:*          LISTEN
...
```

- **-g or --groups:** Display multicast group membership information for IPv4 and IPv6:
- **-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.

netstat Command Alternatives

- The `netstat` command is deprecated.
- Some current alternatives for the `netstat` command and options are shown here:

Netstat command	Alternative Command
<code>netstat</code>	<code>ss</code>
<code>netstat -A FAMILY</code>	<code>ss -f FAMILY</code>
<code>netstat -n</code>	<code>ss -n</code>
<code>netstat -r</code>	<code>ip route</code>
<code>netstat -i</code>	<code>ip -s link</code>
<code>netstat -g</code>	<code>ip maddr</code>



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `netstat` command still functions, but it is obsolete.

The slide shows alternatives for some `netstat` commands. The following are examples of running alternative commands:

- Show socket statistics:

```
# ss
Netid  State  Recv-Q  Send-Q Local Address:Port          Peer Address:Port
u_str  ESTAB  0        0      @/tmp/dbus-...           * 244607
u_str  ESTAB  0        0      * 244113                * 244114
u_str  ESTAB  0        0      /run/systemd/...         * 14794
...

```

- Show sockets for the given FAMILY:

```
# ss -f unix (Note that # ss -x is equivalent to # ss -f unix)
Netid  State  Recv-Q  Send-Q Local Address:Port          Peer Address:Port
u_str  ESTAB  0        0      @/tmp/dbus-...           * 244607
u_str  ESTAB  0        0      * 244113                * 244114
u_str  ESTAB  0        0      /run/systemd/...         * 14794
...

```

Some of the other alternatives for `netstat` are listed as follows:

- Display the routing table:

```
# ip route:  
default via 192.0.2.1 dev eth0 proto static metric 100  
192.0.2.0/24 dev eth0 proto kernel scope link src 192.0.2.103  
192.0.2.0/24 dev eth0 proto kernel scope link src 192.0.2.103  
metric 100  
192.168.1.0/24 dev eth1 proto kernel scope link src 192.168.1.103  
...
```

- Display network interface information:

```
# ip -s link  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    RX: bytes packets errors dropped overrun mcast  
        334828     3612      0      0      0      0  
    TX: bytes packets errors dropped carrier collsns  
        334828     3612      0      0      0      0  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000  
    link/ether 00:16:3e:00:01:03 brd ff:ff:ff:ff:ff:ff  
    RX: bytes packets errors dropped overrun mcast  
        2272714    36553     0      0      0      0  
    TX: bytes packets errors dropped carrier collsns  
        22168052   56954     0      0      0      0  
...
```

- Show multicast information:

```
# ip maddr  
1:      lo  
        inet  224.0.0.1  
        inet6 ff02::1  
        inet6 ff01::1  
2:      eth0  
        link  01:00:5e:00:00:01  
        link  33:33:00:00:00:01  
        link  01:00:5e:00:00:fb  
        link  33:33:ff:d1:c9:16  
        link  33:33:00:00:02:02  
        inet  224.0.0.251  
        inet  224.0.0.1  
...
```

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 © 2019, 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.virbr0
3.nflog (Linux netfilter log (NFLOG) interface)
4.nfqueue (Linux netfilter queue (NFQUEUE) interface)
5.eth1
6.usbmon1 (USB bus number 1)
7.any (Pseudo-device that captures on all interfaces)
8.lo [Loopback]
```

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 262144...
03:25:32.461734 IP host03.example.com.ssh > example.com.56106: Flags
[P.], seq 2445174393:2445174589, ack 2855305377, win 271, options
[nop,nop,TS val 237757443 ecr 1969597489], length 196
```

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. Examples:

```
# tcpdump -i 1 -v
# tcpdump -i 1 -vv
```

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`:

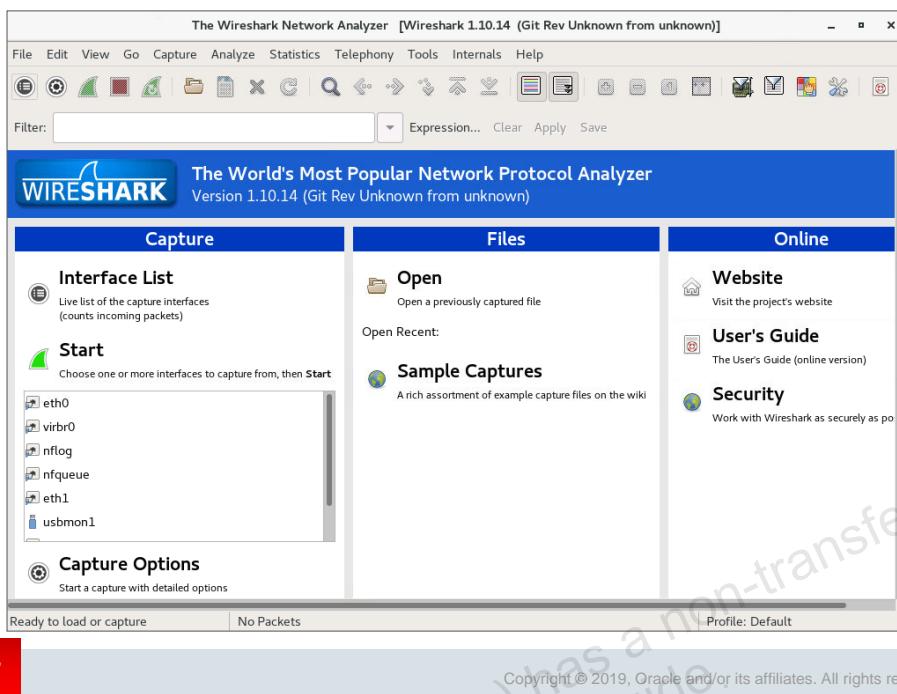
```
# tcpdump protocol
```

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 a time period.
- Enable MAC name resolution.
- Enable network name resolution.

You can also filter a capture based on MAC address, IP address and 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 (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 oswbb812.tar
```

- To start OSWbb, use the following command:

```
# ./startOSWbb.sh
```

- To stop OSWbb, use the following command:

```
# ./stopOSWbb.sh
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Oracle OSWatcher (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 UNIX utilities such 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 Diagnostic Data Tool (RAC-DDT) script file, but is not installed by RAC-DDT.

Installing and running OSWbb as part of performance diagnostic data collection can facilitate issue resolution with Oracle support.

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 Doc ID 301137.1 and can be downloaded as a TAR file named `oswbb<version>.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 oswbb<version>.tar
```

Extracting the TAR file creates a directory named `oswbb`, which contains all the files associated with OSWbb.

```
# ls oswbb
analysis           iosub.sh        oswsub.sh
archive            locks          psmemsub.sh
call_du.sh         ltop.sh        sarsub.sh
call_sar.sh        mpsub.sh      src
call_uptime.sh    nfssub.sh     startOSWbb.sh
data               OSWatcherFM.sh stopOSWbb.sh
docs               OSWatcher.sh   tar_up_full_archive.sh
Example_extras.txt OSWatcher.sh~  tar_up_partial_archive.sh
Exampleprivate.net oswbba.jar   tmp
genprvnet.sh       oswib.sh      topaix.sh
gif                oswnet.sh    vmsub.sh
ifconfigsub.sh    oswrds.sh    xtop.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
# ... Setting the archive log directory to /root/oswbb/archive
Testing for discovery of OS Utilities...
VMSTAT found on your system.
IOSTAT found on your system.
MPSTAT found on your system.
IFCONFIG found on your system.
NETSTAT found on your system.
TOP found on your system.
TRACEROUTE found on your system.
...
Discovery of CPU CORE 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
- Data files are located in subdirectories 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
- oswprvnet: Contains the status of RAC private networks
 - Requires you to manually create an executable file named private.net, which runs traceroute commands



Copyright © 2019, 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 ifconfig, top, vmstat, iostat, mpstat, netstat, and ps utilities and for /proc/meminfo, /proc/slabinfo and /proc/cpuinfo. There is also an optional collector for tracing private networks. The collected data files are stored in subdirectories in the archive subdirectory, which is created when OSWbb is started for the first time. The archive directory contains 11 subdirectories, one for each data collector.

```
# ls archive
oswcpuinfo    oswmeminfo   oswprvnet    oswtop
oswifconfig   oswmpstat    oswps        oswvmstat
oswiostat     oswnetstat   oswslabinfo
```

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

With the exception of oswcpuinfo, each entry in the file contains a time stamp prefixed by *** characters. The contents of each of the 11 archive directories are described as follows:

oswiostat

OSWbb runs the `iostat` utility at the specified interval and stores the data in this directory. 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 in the lower section of the report are not divided up on a per-interface basis, so you need to compare these 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 looks 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 into 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
- **oswtop:** Contains the output from the `top` utility
- **oswvmstat:** Contains the output from the `vmstat` utility
- **oswifconfig:** Contains the output from the `ifconfig` utility
- **oswcpuinfo:** Contents of the `/proc/cpuinfo` file



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The remaining data collection directories are described as follows.

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 in supporting information for RAC diagnostics. For example, the status of a process before a system crash might 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 might 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 over 200 pages per second, then there is a memory shortage. Disk problems might exist if the number of processes blocked exceeds the number of processes on the run queue.

oswifconfig

OSWbb runs the `ifconfig -a` utility at the specified interval and stores the data in this directory. The `ifconfig` command displays the current status of network interfaces. The `ifconfig -a` command utility is most commonly used to troubleshoot RAC network interface issues. The output of this command is used with the output of `netstat` and `private.net` to diagnose any network interface issues on your server.

oswcpuinfo

OSWbb reads the `/proc/cpuinfo` file at the specified interval and stores the data in this directory. Information about the processor(s), including cpu family, model, stepping, clock speed, number of cores, and other details, is included in this file.

OSWatcher Analyzer (OSWbba)

- OSWbba:
 - Is a graphing and analysis utility that is included with OSWbb v4.0.0 and higher
 - Graphically displays data collected and generates reports
 - 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 © 2019, Oracle and/or its affiliates. All rights reserved.

OSWatcher 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 utility log files contained in the `oswbb/archive` directory. However, only the `vmstat`, `iostat`, `top`, `ps` and `netstat` files are analyzed. 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 1.8.0_131:

```
# java -version  
openjdk version "1.8.0_131"  
...
```

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 created by OSWatcher previously. Use the `-i <archive_directory>` option to specify the input directory:

```
# java -jar oswbba.jar -i ~/oswbb/archive  
Starting OSW Analyzer V8.1.1  
...  
Parsing Data. Please Wait...  
...  
Parsing Completed.
```

After the parsing completes, the following menu is displayed, providing options to create graphs 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 GC to Generate All CPU Gif Files  
Enter GM to Generate All Memory Gif Files  
Enter GD to Generate All Disk Gif Files  
Enter GN to Generate All Network Gif Files  
Enter L to Specify Alternate Location of Gif Directory  
Enter Z to Zoom Graph Time Scale (Does not change analysis...)  
Enter B to Returns to Baseline Graph Time Scale (Does not ...)  
Enter R to Remove Currently Displayed Graphs  
Enter X to Export Parsed Data to Flat File  
Enter S to Analyze Subset of Data (Changes analysis dataset ...)  
Enter A to Analyze Data  
Enter D to Generate Dashboard  
Enter Q to Quit Program  
Please Select an Option:
```

The first three options display graphs of specific CPU components of `vmstat`. 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, average wait time, percent busy, throughput per second, and percent busy versus throughput/second. 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 GC, GM, GD – Generate image files of the graphs for these categories: GC - CPU options 1-3; GM - memory option 4; GD - disk option 5, respectively. Files are written to the `oswbb/gif/<month code...>` directory by default
- Option GN – Generates image files for network data
- Option L – Allows you to specify an alternative location for the image files that you created
- Option Z – 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 B – 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 X – Exports parsed data to flat files for spreadsheet analysis. Files are written to the `oswbb/data/<month code...>` directory
- Option S – Analyzes a subset of data
- Option A – Analyzes the files in the archive and produces a report
- Option D – Generates dashboard files for HTML page viewing
- Option Q – Exits the program

Analyzing OSWbb Archive Files

- Start the analyzer from the OSWatcher installation directory (oswbb).
 - 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

ORACLE

Copyright © 2019, 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.../analysis.txt` has been created.

Analyses are placed in the `analysis.txt` file under a directory named after the specific host. For example:

```
ls ~/oswbb/analysis/host03.example.com_Oct30103528_1509527093
analysis.txt
```

The analyzer output is divided into sections for easy readability.

### Section 1: System Status

Provides a brief 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: System Slowdown**

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: System General Findings**

Provides system general findings such as the following:

- CPU run queue observed very high spikes.
- Severe memory swapping was observed.

## **Section 4: CPU Detailed Findings**

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
- Other processes with high CPU consumption

## **Section 5: Memory Detailed Findings**

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: Disk Detailed Findings**

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 Network Detailed Findings**

Provides detailed network findings including data link findings, IP findings, UDP findings, and TCP findings

## **Section 8 Process Detailed Findings**

Provides detailed process findings ordered by time as well as top processes increasing memory

## Oracle Enterprise Manager Cloud Control

- System management software that provides centralized monitoring, administration, and life cycle management for IT infrastructures
- Allows you to set up, manage, and support enterprise clouds and traditional Oracle IT environments
- Provides functionality for Oracle Linux server life cycle management
  - Oracle Linux 7.x patching is supported from Oracle Enterprise Manager Cloud Control 13c version 13.2.
- Centralized web-based user interface for all management functions
- Select product features:
  - Asset discovery
  - Asset monitoring
  - Oracle Linux provisioning
  - Managing Oracle Linux patching and compliance
  - Running procedures to automate tasks



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Oracle Enterprise Manager Cloud Control provides a comprehensive management solution for Oracle environments. Key product capabilities include complete cloud life cycle management, integrated cloud stack management, and business-driven application management. Oracle Linux systems can be managed as part of Oracle Enterprise Manager Cloud Control. Host patching for Oracle Linux 7.x systems is supported from Oracle Enterprise Manager Cloud Control 13c version 13.2 and above. Oracle Linux Support customers can use Oracle Enterprise Manager Cloud Control to manage all their Oracle Linux installed servers.

Some features provided by this solution include the following:

- Asset discovery
- Asset monitoring, including incidents and problems
- Oracle Linux provisioning
- Managing Oracle Linux patching and compliance against RPM repositories
- Running procedures to automate tasks

Patching can be done for single hosts or groups of hosts (called a "Linux Patching Group").

Procedures (called "Deployment Procedures") for automating management tasks can be run against single hosts or groups of hosts as well.

Yum patching and PXE boot provisioning are utilized, following open Linux standards.

Oracle Ksplice is enabled for patching from Oracle Enterprise Manager Cloud Control 13c version 13.3.

See the white paper entitled "Oracle Linux Management with Oracle Enterprise Manager 13c" for more information, found here:

<http://www.oracle.com/us/technologies/linux/linux-with-enterprise-manager-1959006.pdf>.

## Oracle Enterprise Manager Cloud Control Components

The following components are part of Oracle Enterprise Manager Cloud Control:

- Oracle Management Agent
- Oracle Management Service (OMS)
- Oracle Management Repository
- Plug-ins
- Enterprise Manager Cloud Control Console
- Oracle JVMD Engine
- Oracle BI Publisher
- EMCTL
- EM CLI



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The following is a description of product components:

- **Oracle Management Agent:** A software component that enables you to convert an unmanaged host to a managed host in the Oracle Enterprise Manager system. The Management Agent works in conjunction with the plug-ins to monitor the targets running on that managed host.
- **Oracle Management Service (OMS):** A web-based application that orchestrates with the Management Agents and the plug-ins to discover targets, monitor and manage them, and store the collected information in a repository for future reference and analysis.
- **Oracle Management Repository:** A storage location where all the information collected by the Management Agent is stored. It consists of objects such as database jobs, packages, procedures, views, and tablespaces.
- **Plug-ins:** Offer special management capabilities customized for specific target types. Plug-ins work in conjunction with the OMS and the Management Agent to monitor every target in an environment. Therefore, they are deployed to the OMS as well as the Management Agent. These are default plug-ins included with Oracle Enterprise Manager Cloud Control:
  - Oracle Database - to discover, monitor, and manage Oracle Database and related targets
  - Oracle Fusion Middleware - to discover, monitor, and manage Oracle Fusion Middleware products such as Oracle WebLogic Domain and Oracle WebLogic Server
  - Oracle Exadata - to discover, monitor, and manage Oracle Exadata targets

- Oracle Cloud Framework - to access basic features that are common across cloud services such as Infrastructure as a Service (IaaS) and Database as a Service (DBaaS)
- Oracle System Infrastructure - to discover, monitor, and manage Oracle hardware systems and Super Cluster engineered systems
- **Oracle JVMD Engine:** Enables you to diagnose performance problems in Java applications in a production environment. Starting with Oracle Enterprise Manager 13c, as part of the Oracle Fusion Middleware Plug-in deployment, one JVMD Engine is installed and configured by default on the OMS. For every additional OMS deployed, you receive one JVMD Engine by default with that OMS. JVMD Agents must be manually deployed on the targeted JVMs.
- **Oracle BI Publisher:** Oracle's primary reporting tool for authoring, managing, and delivering highly formatted documents. Starting with Oracle Enterprise Manager 13c, Oracle BI Publisher is installed and configured by default on the OMS. For every additional OMS you deploy, you receive one Oracle BI Publisher by default with that OMS.
- **Enterprise Manager Cloud Console:** The user interface allowing you to centrally monitor and administer an entire computing environment
- **EMCTL:** A command-line tool that enables you to execute certain tasks on the OMS and Management Agents. You can use it for tasks such as starting or stopping OMS instances, setting properties on OMS instances, or getting a list of targets being monitored by a specific Management Agent. EMCTL commands are executed on a specific OMS or Management Agent.
- **EM CLI:** A command-line tool that is accessible through programming language constructs, enabling tasks to be created and run from either the command-line or programmatically. EM CLI enables you to access Enterprise Manager Cloud Control functionality from text-based consoles (shells and command-line windows) for various operating systems.

See the Oracle Enterprise Manager Cloud Control Online Documentation Library Release 13.3 for further information, found here: <https://docs.oracle.com/cd/cloud-control-13.3/index.htm>.

## Oracle Enterprise Manager Cloud Control GUI Sections

The screenshot displays the Oracle Enterprise Manager Cloud Control interface with several annotated sections:

- Global Menu (2)**: Offers access to key functionality. It includes links for Summary, Monitoring, Job Reports, Configuration, Compliance, Provisioning and Quality Management, My Oracle Support, Cloud Chargeback, and Consolidations.
- Global Search (4)**: Enables you to search based on various key search filters. It shows a search bar and a dropdown menu for "All Targets" with options like Target Name, Target Type, Target Status, Last Refreshed, Lifecycle Status, List of Resources, Components, Locations, Catalog, and Operating Systems.
- Global Menu (3)**: Offers access to setup and personalization options. It includes links for Help, Enterprise Manager Help, Documentation Library, All Videos, User Forum, Oracle Technology Network, and Sitemap.
- Personalized Home Page (1)**: Enables you to set a particular page as your home page. It shows a dropdown menu with options like Initial Setup Console, Add Target, Environment, Privacy Settings, Welcome Page, Notification Center, Environment Summary, and Environment Page and Logins.
- Enterprise Summary (5)**: Provides an overview of the targets' health. It shows a summary of targets with status 20 (100%) and a detailed breakdown of incidents, alerts, and problems.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This slide shows some different sections of the Oracle Enterprise Manager Cloud Control GUI interface.

The Home page is the first page presented when you log in to the Enterprise Manager Cloud Control Console. When you log in for the first time after installing the product, by default, the Select Enterprise Manager Home Page appears. You can select another page and set that as your Home page based on your job profile or role. This allows you to display a page with information of your choice and interest immediately after you log in.

Across the top of the GUI are aspects of a Global Menu, providing access to key functionality, setup, and personalization options. Global Search allows you to search using filters you select.

An Enterprise Summary page is displayed, showing an overview of targets' health.

Targets are entities such as host machines, databases, Fusion Middleware components, and server targets (hardware), that can be managed and monitored in Cloud Control.

## Oracle Enterprise Manager Cloud Control GUI—Enterprise Summary

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The top navigation bar includes links for Enterprise, Targets, Favorites, History, Setup, and a user account (RAKRNJA). A status bar at the bottom right indicates the page was refreshed on March 27, 2018, at 4:05:21 AM PDT.

**Enterprise Summary**

**Overview Status**

Targets with Status: 686

- Up (387) 56%
- Down (118) 2%
- Unknown (169) 25%
- Under Maintenance (12) 17%

**Incidents**

Updated in the last 24 hours: 113  
Updated in last 7 days: 219

**Breakdown of incidents updated in the last 7 days**

| Category     | Count |
|--------------|-------|
| Availability | 80    |
| Performance  | 76    |
| Security     | 52    |
| Others       | -     |

**Problem:**

Total Open: 43  
Updated in the last 24 hours: 10  
Without Service Request: 43

**Jobs**

Suspended Executions: 0 ✓  
Problem Executions: 0 ✓  
Action Required Executions: 0 ✓

**Inventory and Usage**

Show: Hosts

Details

Hosts distribution by operating system:

- Oracle Linux Server release 6.6: 25%
- Oracle Linux Server release 6.5: 25%
- Oracle Linux Server release 6.4 (Santiago): 25%
- Oracle Linux Server release 6.3: 15%
- Others: 5%

**Compliance Summary**

Standards: Targets

Use: Compliance Score (radio button selected) Violations

**Name**

| Name                                                 | Average Compliance Score |
|------------------------------------------------------|--------------------------|
| Oracle VM Manager secure configuration compliance    | 100                      |
| Oracle VM Manager supported configuration compliance | 100                      |
| etc14-compare - Drift                                | 100                      |

**Patch Recommendations**

View by: Classification (radio button selected) Target Type

Patch recommendations are not available.

⚠ My Oracle Support refresh job has not run successfully in 72 hours. Patch Recommendations information may be stale or unavailable. Either set the preferred My Oracle Support credentials in online mode or manually upload the metadata required in offline mode to submit a 'Refresh From My Oracle Support' job.

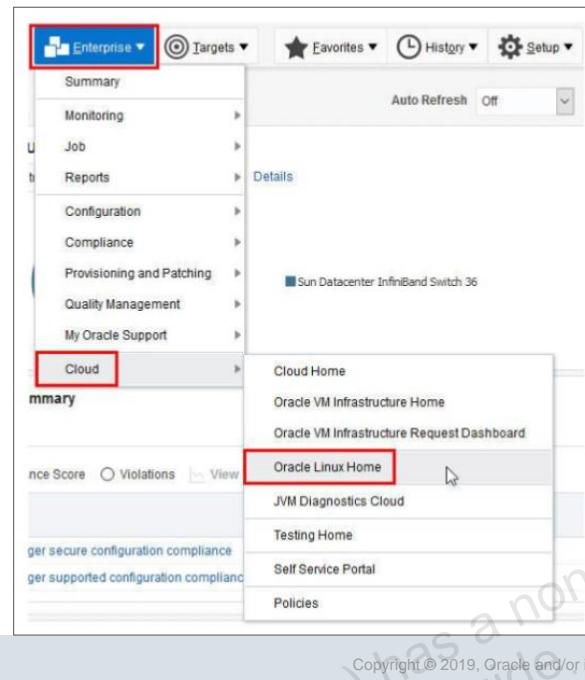
No recommendations to report. Learn More

**ORACLE**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This slide displays a view of the Enterprise Summary page of the Oracle Enterprise Manager Cloud Control GUI interface. This shows an overview of target health.

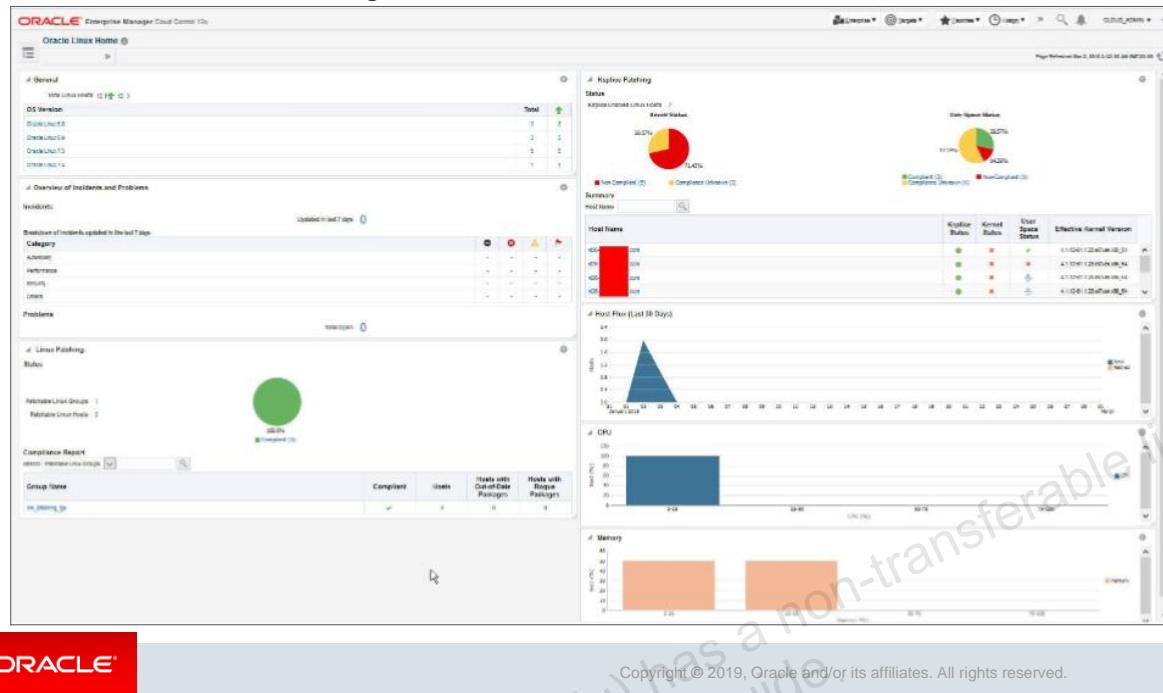
## Oracle Enterprise Manager Cloud Control GUI—Oracle Linux Home



This slide shows the Oracle Linux Home section of the Oracle Enterprise Manager Cloud Control GUI interface.

This is a new home page for Oracle Linux in Oracle Enterprise Manager Cloud Control 13c version 13.3. This interface allows management and monitoring of Oracle Linux hosts.

## Oracle Linux Home Page



This slide shows the Oracle Linux Home page, with these sections:

- **General:** Summary of the Oracle Linux hosts, showing total numbers of each Oracle Linux version, as well as their status
- **Overview of Incidents and Problems:** Incidents or problems relating to availability, performance, and security can be viewed.
- **Linux Patching:** Shows the status of Oracle Linux patching and compliance
- **Ksplice Patching:** The Status region shows the number of Ksplice enabled hosts with their compliance status. The Summary region shows whether Ksplice enabled hosts are online or offline, what the compliance status is for kernel and user space updates, and what the effective kernel version is.
- **Host flux:** Shows new or retired Oracle Linux hosts over a 30-day period
- **CPU:** CPU utilization is shown for a range of Oracle Linux hosts.
- **Memory:** Shows memory utilization for a range of Oracle Linux hosts

## Spacewalk: Overview

- Spacewalk is a full lifecycle management tool for RPM-based Linux distributions.
  - The community project can be found at <https://spacewalkproject.github.io>.
  - Documentation is available at [https://docs.oracle.com/cd/E92593\\_01/index.html](https://docs.oracle.com/cd/E92593_01/index.html).
- Open Source
- Basis for Red Hat Satellite Server 5.x and SUSE Manager
- The RPMs for Oracle Linux are available from the Oracle Linux yum server at [yum.oracle.com](http://yum.oracle.com).
  - Spacewalk Server for Oracle Linux 6 and 7 x86\_64
  - Spacewalk Client for Oracle Linux 7 x86\_64
  - Spacewalk Client for Oracle Linux 6 i386 and x86\_64 architectures
  - Spacewalk Client for Oracle Linux 5 i386 and x86\_64 architectures



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Spacewalk is the open source upstream project of Red Hat Satellite Server 5.x and is the basis for SUSE Manager. It is a full life cycle management tool for RPM-based Linux distributions. Spacewalk is similar to Oracle Enterprise Manager Cloud Control in that it allows you to install and update software on your systems, provision systems, and manage your Oracle Linux systems. While Oracle Enterprise Manager Cloud Control provides monitoring capabilities, Spacewalk does not.

Documentation is available at [https://docs.oracle.com/cd/E92593\\_01/index.html](https://docs.oracle.com/cd/E92593_01/index.html). Release notes, installation requirements (including database requirements), and other information can be found here, for different versions of Spacewalk.

Oracle has made a few changes to Spacewalk to ensure easy and complete support for Oracle Linux. Spacewalk Server and Client software is available from the Oracle Linux yum server at [yum.oracle.com](http://yum.oracle.com). Here you can select the desired version of Oracle Linux, as well as the desired version and architecture of Spacewalk.

Spacewalk Server is available for Oracle Linux 6 and 7 x86\_64 architecture.

Spacewalk Client is available for Oracle Linux 7 x86\_64, Oracle Linux 6 i386 and x86\_64, and Oracle Linux 5 i386 and x86\_64.

## Spacewalk: Features and Functionality

- Manages software updates
- Allows update staging through multiple environments
- Provides a central web-based administration interface
- Enables scheduling of mass updates to thousands of servers
- Allows delivery of software updates targeting specific errata
- Mirrors content from ULN and Oracle Linux yum locally
- Can manage internal and third-party Yum repositories
- Provides geographic distribution by using Spacewalk Proxy and Inter-Server Synchronization (ISS)



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Spacewalk provides software content management. It allows update staging through multiple environments, such as development environments, test environments, quality assurance (QA), near-production, and production environments. You can promote packages through your environments. And you can promote servers through your environments.

Spacewalk has a web-based administration interface. It also includes comprehensive application program interface (API) and command-line tools. Spacewalk enables mass updates of thousands of servers at one time. You can group servers by using System Groups and assign servers to one or more system groups. Spacewalk allows delivery of software updates from specific channels and also by errata or common vulnerabilities and exposures (CVEs). You can have Spacewalk send the patches that resolve a CVE to the affected servers.

Spacewalk can manage internal and third-party Yum repositories. The repositories do not need to come from Oracle or from Red Hat or from an official repository source. You can set up internal repositories within Spacewalk and manually push packages into them. You can set up repositories to sync packages from an upstream source.

Spacewalk supports geographic distribution. Instead of having multiple clients all connecting to the same server, you can set up multiple complete Spacewalk instances and use Inter-Server Synchronization (ISS) to link them together. The other option is to use Spacewalk Proxy. This is a proxy that sits between servers and the main Spacewalk instance. Spacewalk Proxy can provide packages and first-level processing for downstream clients.

## Spacewalk: Features and Functionality

- Provisioning of new physical and virtual servers
  - Supports PXE-based deployments by using kickstart
  - Automatically registers new servers
  - Can redeploy existing servers by using PXE
  - Can create new virtual instances
- Remote Management
  - Spacewalk clients connect with the Spacewalk server every 4 hours checking for any actions to run.
  - OSAD provides near-real time triggering of actions on clients.
  - rhnconfig provides local configuration file management and remote actions.
- Auditing
  - Spacewalk can trigger OpenSCAP-based XCCDF testing and provide results.



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Spacewalk supports provisioning of both physical and virtual servers. Physical server provisioning is based on Preboot Execution Environment (PXE) or booting over the network. Spacewalk supports multiple kickstart configurations. You can have a kickstart configuration for different versions of Oracle Linux and a kernel-based virtual machine (KVM) configuration for virtual instances. If you enable the Spacewalk client channel, it will automatically register your server with Spacewalk as part of the kickstart process.

Spacewalk can create Xen and KVM virtual instances. Note that it is not supported on Oracle VM. You cannot run the Spacewalk client that creates virtual images on Oracle VM. It also cannot replace Oracle VM Manager, but it can create KVM instances on Oracle Linux. It uses Spacewalk Proxy, so you do not need to kickstart from the central location. Spacewalk configures its proxies to do local kickstarting of the geographical locations.

By default, the `rhnscd` daemon on Spacewalk client systems connects with the Spacewalk server every 4 hours and performs any updates or actions that are scheduled. There is an additional client tool called Open Source Architecture Daemon (OSAD) that provides triggering of actions on clients. There is also a local configuration file manager called `rhnconfig` that allows you to send files and directories from your Spacewalk server down to your Spacewalk client. It also supports remote actions so you can send commands down to the client.

From an auditing perspective, Spacewalk can trigger OpenSCAP-based XCCDF testing on a daily or weekly basis.

## Spacewalk Web Interface - Overview Page

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The Spacewalk web interface Overview page is shown in the slide. The menu header provides the following administrative areas that you can select:

- **Overview:** The default Overview page presents a dashboard view of the state of the Spacewalk server. The page displays important information about systems, recently scheduled actions, relevant security errata, and links to administrative tasks.
- **Systems:** The System Overview page displays a summary of the numbers of available updates, errata, packages, configuration files, and crashes; the name of the base channel; and the entitlements for each managed client system.
- **Errata:** The Errata Relevant to Your Systems page displays information about the errata that are available for your registered systems.
- **Channels:** The Full Software Channel List page displays the channels to which you can subscribe your registered systems.
- **Audit:** The OpenSCAP Scans page displays a summary of any scans that you have performed on your systems.
- **Configuration:** The Configuration Overview page displays a summary of the configuration files known to Spacewalk, links to actions you can perform with configuration files, and scheduled deployments of configuration files.
- **Schedule:** The Pending Actions page displays a list of actions that are scheduled to be performed.
- **Users:** The Active Users page displays a list of administrators or other users and their allocated roles.
- **Admin:** The Organizations page displays the organizations configured to be administered, as well as the configured number of systems, administrators, and trusts.

## Quiz



Which of the following utilities allows you to collect system information for sending to Oracle support? (Select all that apply.)

- a. sosreport
- b. sar
- c. OSWbb
- d. strace



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

**Answer:** a, c

## Quiz



Which of the following utilities monitors system processes in real time?

- a. mpstat
- b. iostat
- c. vmstat
- d. top



**ORACLE®**

Copyright© 2019, Oracle and/or its affiliates. All rights reserved.

**Answer: d**

## 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 (OSWbb) tool
- Use OSWatcher Analyzer (OSWbba)
- Describe Oracle Enterprise Manager Cloud Control
- Describe Spacewalk



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## Practices for Lesson 18: 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 Analyzer



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

# System Logging

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## Objectives

After completing this lesson, you should be able to:

- Describe Oracle Linux 7 system logging options
- Describe the contents of the `rsyslog` configuration file
- Describe `rsyslog` filter options
- Describe facility/priority-based filters
- Describe `rsyslog` actions
- Describe `rsyslog` templates
- Configure log rotation
- Describe `logwatch`
- Describe `journald`
- Use the `journalctl` utility
- Explain process accounting



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## System Logging: Introduction

- Log files store system, kernel, service, and application messages.
  - Most log files are located in the `/var/log/` directory.
- Some log files are controlled by the `rsyslogd` daemon.
  - `/etc/rsyslog.conf` is the main configuration file.
  - It contains global directives, modules, and rules.
- With Oracle Linux 7, log files can also be managed by the `journald` daemon.
  - `journald` is a component of `systemd`.
  - `journald` captures various system messages, indexes them, and stores them in `/run/log/journal/`.
  - Use the `journalctl` utility to view the journal logs.



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Log files contain messages about the system, the kernel, services, and applications. Most of these log files are located in the `/var/log/` directory. Some log files are controlled by the `rsyslogd` daemon. The main configuration file for `rsyslogd` is `/etc/rsyslog.conf`, which contains global directives, modules, and rules.

- **Global Directives:** Configuration options that apply to the `rsyslogd` daemon
- **Modules:** Dynamically loaded modules that provide additional functionality and associated configuration directives
- **Rules:** Define a filter, which is a subset of `rsyslog` messages, and an action, which specifies what to do with the messages

With Oracle Linux 7, log files can also be managed by the `journald` daemon. The `journald` daemon is a component of `systemd` that captures various system messages, indexes them, and stores them in the `/run/log/journal/` directory. You can use the `journalctl` utility to view the journal logs.

This lesson begins with a discussion of `rsyslogd` and ends with a discussion of `journald`.

## rsyslog Configuration

- Global directives:
  - They specify configuration options that apply to the `rsyslogd` daemon.
  - All configuration directives must begin with a dollar sign (\$).
- Modules:
  - Are dynamically loaded by using the `$ModLoad` global directive
  - Provide additional functionality and configuration directives
  - Categories of modules include *Input*, *Output*, *Parser*, *Message modification*, *String generator*, and *Library*.
- Rules:
  - Specify a *filter* (`cron.*`) and *action* (log all `cron` messages to `/var/log/cron`):

```
$IncludeConfig /etc/rsyslog.d/*.conf
```

```
cron.* /var/log/cron
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `/etc/rsyslog.conf` configuration file contains global directives, modules, and rules.

### Global Directives

Global directives specify configuration options that apply to the `rsyslogd` daemon. All configuration directives are specified on a single line and must begin with a dollar sign (\$). 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 configuration directives and their descriptions can be found at [http://www.rsyslog.com/doc/rsyslog\\_conf\\_global.html](http://www.rsyslog.com/doc/rsyslog_conf_global.html).

### Modules

`rsyslog` has a modular design, which allows functionality to be loaded from modules dynamically. Each module provides configuration directives. Modules must be loaded for their configuration directives and functionality to be available. The following example uses the `$ModLoad` global directive to load the `imjournal` module:

```
$ModLoad imjournal
```

The `imjournal` module transfers data acquired by `journald` to `rsyslogd`. The `omjournal` module is available to transfer data from `rsyslogd` to `journald`.

The following describes the main categories of `rsyslogd` modules:

- **Input modules:** Gather messages from various sources. Input module names always start with the `im` prefix (examples: `imfile`, `imjournal`).
- **Output modules:** Output messages to various targets such as across a network, storing them in a database, or encrypting them. Output module names always start with the `om` prefix (examples: `omsnmp`, `omjournal`).
- **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 (examples: `pmciscoios`, `pmlastmsg`).
- **Message modification modules:** Change the content of an `rsyslog` message. Names of these modules always start with the `mm` prefix (examples: `mmcount`, `mmfields`).
- **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 (examples: `smfile`, `smfwd`).

Any output that is generated by `rsyslog` can be modified and formatted by using templates.

- **Library modules:** Library modules provide functionality for other loadable modules. These modules cannot be configured and are loaded automatically by `rsyslog` when needed.

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 might also modify the message itself. The internal representation is passed to output modules, which might output a message and 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

Every rule consists of two fields, a *filter* part and an *action* part. A *filter* specifies a subset of `rsyslog` messages to select. An *action* 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.

Following are examples of rules defined in the `/etc/rsyslog.conf` file. Lines beginning with the `#` sign are comments.

```
Log all kernel messages to the console.
kern.* /dev/console

Log all the mail messages in one place.
mail.* /var/log/maillog

Log cron stuff
cron.* /var/log/cron
```

## rsyslog Filter Options

There are three ways to filter rsyslog messages:

- Facility/priority-based filters:
  - Filters are based on two conditions: facility and priority.
  - Facility specifies the subsystem that produces the message.
  - Priority represents the priority of the message.
- Property-based filters:
  - Filter by comparing a property of the message to a value
- Expression-based filters:
  - Filter according to arithmetic, Boolean, or string operations
  - Use rsyslog scripting language. Syntax:

```
:msg, contains, "error"
```

```
if EXPRESSION then ACTION else ACTION
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The rsyslogd daemon offers three different ways to filter rsyslog messages:

### Facility/Priority-Based Filters

Facility/priority-based filters filter rsyslog messages based on two conditions: *facility* and *priority*. Facility specifies the subsystem that produces the message. Examples of facilities include `mail`, `kernel`, and `cron`. Priority represents the priority of the message. Examples of priorities include `debug` (7), `warning` (4), and `alert` (1).

### Property-Based Filters

Filter rsyslog messages by any property, such as `timegenerated` or `msg`. You can compare a property to a value by using one of several property-based compare operations. Compare operations include `contains`, `isequal`, and `startswith`. The following example filters for messages that contain the string "error" in the message text (`msg`):

```
:msg, contains, "error"
```

### Expression-Based Filters

Select rsyslog messages according to arithmetic, Boolean, or string operations by using an rsyslog scripting language. The following shows the basic syntax of expression-based filters:

```
if EXPRESSION then ACTION else ACTION
```

## 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 auth rsyslog messages with any priority:

***auth.\****

- Select all mail rsyslog messages with priority err and higher:

***mail.err***

- Select all user rsyslog messages except those with info or debug priority:

***user.!info,!debug***

**ORACLE**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Facility/priority-based filters select rsyslog messages based on two conditions: *facility* and *priority*. A facility-priority pair is called a selector. 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). All messages of the specified priority and higher are logged according to the given action.

- **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.

The following are examples of facility/priority-based selectors. To select all `mail` messages with priority `err` and higher:

```
mail.err
```

Special characters can be used. Use an asterisk (\*) to specify all facilities or priorities. For example, to select all `auth` messages with any priority:

```
auth.*
```

Use a comma (,) to specify multiple facilities and priorities. For example, to select both the `uucp` and `news` facilities with priority of `warning` or higher:

```
uucp,news.warning
```

Use a semicolon (;) to define multiple selectors on one line. Example:

```
*.info;mail.none;auth.none;cron.none
```

Use an equal sign (=) to specify a single priority. All other priorities are ignored. For example, to select `cron` messages of only `emerg` priority:

```
cron.=emerg
```

Precede a priority with an exclamation mark (!) to select all `rsyslog` messages except those with the defined priority. The following example selects all `user` messages, except those with the `info` or `debug` priority:

```
user.!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.
  - Write rsyslog messages into 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 © 2019, Oracle and/or its affiliates. All rights reserved.

Actions specify what to do with the messages filtered out by a selector. The following are some of the available actions.

### Saving rsyslog Messages to Log Files

To save an rsyslog message to a log file, specify the absolute path to the log file after the selector. The following example selects all cron messages, and the action saves them to the /var/log/cron.log log file:

```
cron.* /var/log/cron.log
```

You can specify an existing tty or /dev/console device to send rsyslog messages to standard output.

### Sending rsyslog Messages over the Network

Use the following syntax to forward rsyslog messages to a remote machine:

```
@ [zNUMBER] HOST: [PORT]
```

Use a single at sign (@) to specify UDP as the transport protocol. Use a double at sign (@@) to specify TCP. The optional zNUMBER field enables a level of zlib compression from 1 to 9. The HOST field specifies the receiving host. The optional PORT field specifies the port number on the receiving host.

For example, to forward messages to 192.0.2.101 using the UDP protocol:

```
. @192.0.2.101
```

To forward messages to port 18 on “host02.example.com” using the TCP protocol:

```
. @@host02example.com:18
```

### Sending rsyslog Messages to Specific Users

Specify the username to send rsyslog messages to. Separate usernames with a comma (,) to specify more than one user. Use an asterisk (\*) to send messages to every user that is currently logged on. The following example sends all kernel messages to user joe:

```
kern.* joe
```

### 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. The following example processes all kernel messages by the template knl and passes them on to the knl-prog program. Templates are discussed in the next slide.

```
kern.* ^knl-prog;knl
```

NOTE: Undesired commands can be injected and executed in the specified program when accepting messages from any host. Exercise caution when using this program execution method.

### Write rsyslog Messages into a Database

You can use the database writer action to write selected rsyslog messages directly into a database table. The database writer uses the following syntax:

```
:PLUGIN:DB_HOST, DB_NAME, DB_USER, DB_PASSWORD; [TEMPLATE]
```

The *PLUGIN* field specifies the plug-in that performs the database writing. rsyslog provides support for MySQL and PostgreSQL databases. MySQL integration requires the rsyslog-mysql software package. PostgreSQL requires the rsyslog-pgsql package. You also need to load the *ommysql* module for MySQL and the *ompgsql* module for PostgreSQL.

### Discarding rsyslog Messages

Use the tilde character (~) to discard selected messages. The following rule discards any news messages:

```
news.* ~
```

You can specify multiple actions for a selector by specifying subsequent actions on a new line and preceding the actions with an ampersand character (&). Specify the selector on the first action line. The following is an example of a rule with multiple actions:

```
kern.* joe
& ^knl-prog;knl
& @192.0.2.101
```

In the preceding example, all kernel messages are:

- Sent to user joe
- Processed by the template knl and passed on to the knl-prog executable
- Forwarded to 192.0.2.101 by using the UDP protocol

## rsyslog Templates

Templates modify and format output generated by rsyslog.

- Syntax to create a template:

```
$template TEMPLATE_NAME, "text %PROPERTY% text", [OPTION]
```

- Templates can be used to generate dynamic file names:

```
$template DynamicFile,
"/var/log/%timegenerated%-test.log"
```

- Example of a template definition:

```
$template class, "Time: %timestamp%, Facility: %syslogfacility-text%,
Priority: %syslogpriority-text%, Hostname: %hostname%, Message: %msg%\n"
```

- Example of using a template in a rule:

```
. /var/log/logfile;class
```

**ORACLE**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use templates to modify and format rsyslog output. The following is the syntax to create a template:

```
$template TEMPLATE_NAME, "text %PROPERTY% text", [OPTION]
```

The fields are described as follows:

- \$template**: Directive that defines a template
- TEMPLATE\_NAME**: Name of the template
- "text"**: Actual template text surrounded by quotation marks
- %PROPERTY%**: Specific message content surrounded by 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/%timegenerated%-test.log"
```

Specify the template name in a rule to modify rsyslog output. Dynamic files are represented by a template and a question mark (?) prefix. Example:

```
. ?DynamicFile
```

## Properties

You can use properties inside a template to reference specific contents of an `rsyslog` message. Use the following syntax to define a property inside a template:

```
%PROPERTY_NAME[:FROM_CHAR:TO_CHAR:OPTION]%
```

The fields are described as follows:

- **PROPERTY\_NAME**: Name of a property
- **FROM\_CHAR** and **TO\_CHAR**: Range of characters the specified property acts upon
- **OPTION**: Property options

A list of available properties and descriptions can be found at  
[http://www.rsyslog.com/doc/property\\_replacer.html](http://www.rsyslog.com/doc/property_replacer.html).

The following property represents the entire message text of an `rsyslog` message:

```
%msg%
```

The following example represents the first two characters of the message text:

```
%msg:1:2%
```

The following property represents the host name in an `rsyslog` message:

```
%hostname%
```

The following property represents the facility from the message in text form:

```
%syslogfacility-text%
```

### Template: Example

The following example defines a template named `class` that formats an `rsyslog` message to output the message's time stamp, facility in text form, priority in text form, host name, and message text and ends with a new line:

```
$template class, "Time: %timestamp%, Facility: %syslogfacility-text%,
Priority: %syslogpriority-text%, Hostname: %hostname%, Message:
%msg%\n"
```

To use the template for `/var/log/logfile` messages, include the template name as follows:

```
. /var/log/logfile;class
```

## Configuring Log Rotation (logrotate)

- `logrotate` is a utility to automatically manage log files.
  - It runs as a daily cron job using the `/etc/cron.daily/logrotate` file.
- The `/etc/logrotate.conf` file 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 © 2019, 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 using the `/etc/cron.daily/logrotate` file. The main configuration file for `logrotate` is `/etc/logrotate.conf`. There are also 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 File

The following is a sample /etc/logrotate.conf configuration file:

```
cat /etc/logrotate.conf
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/ Directory

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
bootlog glusterfs libvirtd.qemu psacct up2date yum
chrony iscsiuilog numad samba vsftpd
...
...
```

The following is an example of the /etc/logrotate.d/chrony configuration file:

```
cat /etc/logrotate.d/chrony
/var/log/chrony/* log {
 missingok
 nocreate
 sharedscripts
 postrotate
 /usr/libexec/chrony-helper command cyclelogs > /dev/nul...
 endscript
}
```

The options in the /etc/logrotate.d/chrony configuration file are described as follows:

- **missingok:** If the log file is missing, do not issue an error message.
- **nocreate:** New log files are not created.
- **postrotate/endscript:** The lines between these directives are executed after the log file is rotated.
- **sharedscripts:** The postrotate script runs only once, not once for each log that is rotated.

For a full list of directives and configuration options, refer to the `logrotate(8)` man page.

## The logwatch Utility

- **logwatch** is a utility to perform basic log file monitoring and analysis.
  - It runs as a daily cron job by using the `/etc/cron.daily/0logwatch` file.
- The local configuration file is:
  - `/etc/logwatch/conf/logwatch.conf`
- The main configuration file is:
  - `/usr/share/logwatch/default.conf/logwatch.conf`
- A HOWTO-Customize-Logwatch file exists in the following directory:
  - `/usr/share/doc/logwatch-<version>/`
- **logwatch** can also be run from command line. Example:

```
logwatch --help
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

**logwatch** is a customizable log monitoring system. It goes through system logs for a given time period and reports on specific areas of interest.

It might be necessary to install the **logwatch** package. 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.

The main configuration file is `/usr/share/logwatch/default.conf/logwatch.conf`, while local configuration options can be set in `/etc/logwatch/conf/logwatch.conf`. A HOWTO-Customize-Logwatch file exists in the `/usr/share/doc/logwatch-<version>/` directory. Following are some of the options you can configure:

- Level of detail
- Log files to report on
- Names of services 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 about using **logwatch**:

```
logwatch --help
```

## journald: Introduction

- Journald:
  - Is a logging service included with `systemd`
  - Collects and stores logging data in structured, indexed journals.
    - Supports advanced query options and faster search times
  - Adds structured metadata to the messages that assist in troubleshooting
  - Can be used together with, or in place of, `rsyslogd`
- Journals are nonpersistent by default.
  - Are stored in `/run/log/journal/`
- To configure persistent journal data storage:

```
mkdir -p /var/log/journal
systemctl restart systemd-journald
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Log files can also be managed by the `journald` daemon, which is part of `systemd`. The full service name is `systemd-journald.service`, and the full daemon name is `systemd-journald`. These can be seen referred to by the shortened name "journald" in documentation. In the example on the slide, the name `systemd-journald` is necessary to restart the service. `journald` collects and stores logging information from sources such as the kernel, system services, and user processes. `journald` stores data in structured, indexed journals that support advanced query options and faster search times than traditional log files. `journald` adds structured metadata to the messages that assist in troubleshooting. `journald` can be used together with, or in place of, `rsyslogd`.

By default, the journal stores log data in `/run/log/journal`. The `/run` mount point is a `tmpfs` file system that is mounted at boot time.

```
mount | grep run
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
...
```

A `tmpfs` file system stores its files in virtual memory. It is a temporary file system in the sense that data is lost at reboot or if the file system is unmounted. In addition, the amount of logged data depends on free memory. When you run out of free memory, the oldest entries in the journal are deleted.

You can make journal data persistent by creating the `/var/log/journal/` directory and then restarting the `systemd-journald` service.

```
mkdir -p /var/log/journal
systemctl restart systemd-journald
```

## The journalctl Utility

- Use the `journalctl` command to view the journal logs.

```
journalctl
```

- Output is displayed one page at a time.
- Time stamps are converted to your local time zone.
- Priority of entries is visibly marked.
  - Entries with error priority and higher are red.
  - Entries with notice and warning priority are in bold font.
- The beginning of the boot process has a special entry.
- With no options, all log data is displayed.
- By default, oldest entries are listed first.
- A number of query options are available. See:

```
journalctl -h
```

ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `journalctl` command to view the journal logs. By default, the listed entries include a time stamp, the host name, the application involved, and the given message.

```
journalctl
-- Logs begin at ..., end at ...
<date_time> <host name> systemd-journal[65]: ...
...
```

The output of the command is formatted as follows:

- Entries are displayed one page at a time.
- Time stamps are converted to your local time zone.
- Priority of entries is visibly marked. Entries with error priority and higher are red. Entries with notice and warning priority are in bold font.
- The beginning of the boot process is indicated with a special entry.

When running the `journalctl` command without any options or arguments, all log data is displayed, including rotated logs. Oldest entries are listed first. A number of options are available for the `journalctl` command. Examples of some of the options are given on the next page.

Use the `-r` option to display the newest log entries first.

```
journalctl -r
-- Logs begin at ..., end at ...
<date_time> <host name> CROND[27325]: (root) CMD (/usr...
<date_time> <host name> systemd[1]: Started session ...
...
```

Use the `-n <number>` option to display a specific number of the most recent log entries. The following example displays the three most recent log entries.

```
journalctl -n 3
-- Logs begin at ..., end at ...
<date_time> <host name> systemd[1]: Started session ...
<date_time> <host name> systemd[1]: Started session ...
<date_time> <host name> CROND[27452]: (root) CMD (/usr...
```

Use the `-p <priority>` option to display only log entries of a specific `<priority>`. Valid priorities are debug, info, notice, warning, err, crit, alert, and emerg. The following example displays only crit log entries. Entries with `err` priority and higher are in red.

```
journalctl -p crit
-- Logs begin at ..., end at ...
<date_time> host03.example.com smartd[512]: Problem creating ...
<date_time> host03.example.com smartd[512]: In the system's ...
<date_time> host03.example.com firewalld[506]: 2014-11-10 ...
...
```

Use the `-u <systemd_unit>` option to display only log entries for the specified systemd unit. The following example displays only log entries associated with the `crond` unit.

```
journalctl -u crond
-- Logs begin at ..., end at ...
<date_time> <host name> systemd[1]: Starting Command ...
<date_time> <host name> systemd[1]: Started Command ...
<date_time> <host name> crond[578]: (CRON) INFO (RAN...
...
```

Use the `-o <output_form>` option to format the output. Valid output formats are short, short-iso, short-precise, short-monotonic, verbose, export, json, json-pretty, json-see, and cat. Refer to the `journalctl(1)` man page for a description of the output formats. The following example displays log entries using the `verbose` format.

```
journalctl -o verbose
-- Logs begin at ..., end at ...
<date_time>
 PRIORITY=6
 _TRANSPORT=driver
 MESSAGE=Runtime journal is using 8.0M (max allowed 99.8M, trying ...
...
```

## journald Metadata

- journald adds structured metadata to the messages that assist in troubleshooting.
- To view all metadata for all journal entries:

```
journalctl -o verbose
```

- To view a list of the metadata fields:

```
journalctl <TAB> <TAB>
```

- To view a list of unique values that occur in a specific metadata field, use the following syntax:

```
journalctl -F <fieldname>
```

- To filter log entries for a specific metadata value, use the following syntax:

```
journalctl <fieldname>=<value>
```



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

journald adds structured metadata to the messages that assist in troubleshooting. The following command (also shown on the previous page) displays all metadata for all journal entries:

```
journalctl -o verbose
```

You can view a list of the metadata fields by pressing the <TAB> key twice after the journalctl command:

```
journalctl <TAB> <TAB>
```

For a description of the metadata fields, see the `systemd.journal-fields(7)` man page. Metadata values are usually text-based but can include binary data. Metadata fields can also have multiple values, but this is usually not the case. Metadata can be used for message filtering to assist in troubleshooting.

To view a list of unique values that occur in a specific metadata field, use the following syntax:

```
journalctl -F <fieldname>
```

You can specify a <value> for a <fieldname> to show only log entries that match the condition. Use the following syntax:

```
journalctl <fieldname>=<value>
```

You can specify multiple values for one field, and you can specify multiple field-value pairs.

## Process Accounting

- Process accounting provides a means to monitor activities on systems.
- It can be used for:
  - Billing based on system resource usage
  - System tuning based on usage patterns
  - Enhancing security—can discover attacker command usage
- The `psacct` package provides the process accounting service (`psacct`) and a set of utilities.
- Utilities provide information about connection times and commands executed.
- The default process accounting file is `/var/account/pacct` (for executed command information).
- Login/logout record maintained in `/var/log/wtmp` (for connection times)
- Monitor space usage of the process accounting file (`/var/account/pacct` by default) and the file recording logins and logouts (`/var/log/wtmp`).



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Process accounting provides information about connect times (for logins) and command execution via a set of utilities that report aspects of collected data. It can be used for various purposes, including security.

By default, process accounting data is stored in `/var/account/pacct`. When each process terminates, a record with information about the process is written to the accounting file. This file exists after system installation (empty) and is available for use when process accounting is enabled.

The `/var/account/pacct` (or alternative if specified) and `/var/log/wtmp` files should be monitored for space usage, as they can get very large. The `pacct` file is rotated once per day 31 times before being deleted. Rotated files are compressed (see `/etc/logrotate.d/psacct` and the `logrotate(8)` man page for details).

## Process Accounting Utilities

- `accton` - Turns process accounting on/off
  - Alternatively, the `psacct` service can be started, to turn on process accounting
- `ac` - provides connection time statistics
- `lastcomm` - Provides information about previously executed commands
- `sa` - Summarizes information about previously executed commands
- `dump-acct` - Provides human-readable output of select process accounting file data
- `dump-utmp` - Provides human-readable output of the `utmp` or `wtmp` file



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The `accton` utility turns process accounting on or off. An alternative to the default process accounting file can be specified as an argument to the `accton` utility. The `psacct` service can also be started as an alternative to using `accton` directly.

`ac` with no options provides total connect time for all users in hours. By default, the `ac` utility uses the `/var/log/wtmp` file to report login connection times. `wtmp` keeps records of *all* logins and logouts, as opposed to `/var/run/utmp`, which records information about *current* users. These files exist after system installation. An alternate file can be specified with `ac`.

`lastcomm` with no options provides information from the default process accounting file about all commands executed previously. An alternate file can be specified.

`sa` with no options provides summary information from the default process accounting file about all commands executed previously. An alternate file can be specified.

The `dump-acct` utility provides human-readable output from a process accounting file. A filename must be specified. For example, to view output from the default process accounting file:

```
dump-acct /var/account/pacct
```

`dump-utmp` provides human-readable output of the `/var/log/wtmp` file as well as of the `/var/run/utmp` file. A filename must be specified. For example, to view output from the `wtmp` file:

```
dump-utmp /var/log/wtmp
```

See the man pages for specific utilities for more information.

## The accton Utility - Turn Process Accounting On/Off

- `accton on` turns on process accounting using the default accounting file, `/var/account/pacct`:

```
accton on
Turning on process accounting, file set to the default
'/var/account/pacct'.
```

- `accton <filename>` specifies an alternate accounting filename:

```
accton /var/account/my_acct
Turning on process accounting, file set to '/var/account/my_acct'.
```

- `accton off` turns process accounting off:

```
accton off
Turning off process accounting.
```

- The `psacct` service can be started as an alternative—this calls the `accton` utility.



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

The slide shows forms of the `accton` utility, which turns process accounting on or off. If an alternate filename is specified rather than the default accounting file, `/var/account/pacct`, the file must first be created and the permissions changed to read/write by root. For example:

```
touch /var/account/my_acct
chmod 600 /var/account/my_acct
```

Alternatively, the `psacct` service can be started. It is a wrapper that calls the `accton` utility and can be enabled to start process accounting at boot time. In this case, the default accounting file, `/var/account/pacct`, is used and cannot be changed. Initially, the `psacct` service is inactive. To start the `psacct` service and check its status:

```
systemctl start psacct
systemctl status psacct
```

- `psacct.service` - Kernel process accounting

```
Loaded: loaded (/usr/lib/systemd/system/psacct.service; disabled;
vendor preset: disabled)
```

```
Active: active (exited) since <date/time>; 2s ago
Process: 10748 ExecStart=/usr/sbin/accton /var/account/pacct
(code=exited, status=0/SUCCESS)
Process: 10746 ExecStartPre=/usr/libexec/psacct/accton-create
(code=exited, status=0/SUCCESS)
Main PID: 10748 (code=exited, status=0/SUCCESS)
```

See the `accton(9)` man page for details.

## 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



ORACLE

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

### Answer: a, b

Answer b is correct because kern.warning logs kernel messages with priority of warning and higher. Priorities higher than warning are err, crit, alert, and emerg (see the section "Facility/Priority-Based Filters" above, for priority order).

## Quiz



Which of the following statements are true?

- a. To use the `journald` service, you must stop `rsyslogd`.
- b. Journals are nonpersistent by default.
- c. The `journald` service adds structured metadata to the messages that assist in troubleshooting.
- d. Use the `journald` command to view the journal log files.



**ORACLE®**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

**Answer: b, c**

## Summary

In this lesson, you should have learned how to:

- Describe Oracle Linux 7 system logging options
- Describe the contents of the `rsyslog` configuration file
- Describe `rsyslog` filter options
- Describe facility/priority-based filters
- Describe `rsyslog` actions
- Describe `rsyslog` templates
- Configure log rotation
- Describe `logwatch`
- Describe `journald`
- Use the `journalctl` utility
- Explain process accounting



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## Practices for Lesson 19: Overview

The practices for this lesson cover the following topics:

- Configuring system logging
- Using `rsyslog` templates
- Using `logwatch`
- Using `journald`
- Using process accounting



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

# Troubleshooting

The ORACLE logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## 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



ORACLE®

Copyright © 2019, 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 fault diagnosis phases.



Copyright © 2019, 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?



ORACLE®

Copyright © 2019, 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 might 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.

- You can monitor a log file in real time by using the following command:

```
tail -f <logfile>
```

- To view the journal in live view:

```
journalctl -f
```



ORACLE

Copyright © 2019, 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 `/var/log/messages` file 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. Use the `journalctl -f` command to monitor the journal in live view. These commands keep 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 can display human readable time stamps with the “-T” option:
  - dmesg -T
- The buffer can truncate when it is full.
  - /var/log/boot\*
  - /var/log/dmesg\*



Copyright © 2019, 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 the `dmesg(1)` man page 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 7 administration guides:
  - [http://docs.oracle.com/cd/E52668\\_01/](http://docs.oracle.com/cd/E52668_01/)
- The My Oracle Support website contains knowledge articles and other helpful information.
  - <https://support.oracle.com/>



ORACLE

Copyright © 2019, 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 `-d` or `-D` options are normally turned off or not specified by default because they display more information than you need when everything is running smoothly. However, when a problem occurs, the detailed information can be useful in troubleshooting where the problem might 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 contains valuable information to assist in troubleshooting system problems.

Internet searches can 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 the Oracle My Support interface. At the top, there's a navigation bar with links like Dashboard, Knowledge, Service Requests, Patches & Updates, Community, Certifications, Managed Cloud, CRM On Demand, and More... A status indicator shows '(Available)' with a green light. Below the navigation is a search bar with the query 'linux boot'. The search results page is titled 'KM Search Results' and shows a list of articles under 'Knowledge Base Search Results'. Each article entry includes a small icon, the date it was created, the title, and a link to view details. The results are paginated with 'Page 1 of 1' at the bottom.

| Date         | Title                                                                         | Link               |
|--------------|-------------------------------------------------------------------------------|--------------------|
| Sep 11, 2017 | How to Boot Linux OS to Rescue Mode with Multipathing                         | (Doc ID 2302834.1) |
| Jul 23, 2012 | Exalogic Compute Node Stuck At GRUB Linux Boot Sequence                       | (Doc ID 1370639.1) |
| Sep 24, 2017 | How To Boot Linux Paravirtualized VM Into Rescue Mode and in Single User Mode | (DocID 1602157.1)  |
| Nov 19, 2012 | How to boot Oracle Linux in Interactive Mode                                  | (Doc ID 1500952.1) |
| Nov 24, 2016 | ILOM Java console appeared as Linux System in Hung during boot                | (Doc ID 2190274.1) |
| Jan 29, 2016 | How to Boot Oracle Linux 5.x into Rescue Mode                                 | (Doc ID 1516777.1) |
| Sep 14, 2017 | Quickest and easiest way to boot a PVM Linux guest into Rescue Mode           | (Doc ID 2187003.1) |
| Aug 25, 2013 | How to Configure an Oracle Linux PXE Boot Server                              | (Doc ID 578582.1)  |

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 `systemctl` command to start a service or check the status of a service.
  - Use the `systemctl enable` command to start a service at boot time.
- Configuration errors:
- Firewall (`firewalld` and `iptables`) is prohibiting a connection.
  - Stop the service 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 © 2019, Oracle and/or its affiliates. All rights reserved.

Use the `systemctl` command to ensure a particular daemon (or service) is running. For example, to obtain the status of the `sshd` daemon:

```
systemctl status sshd
```

Always start (or restart) a service whenever making a change to an associated configuration file. For example, after changing parameters in a network interface file in the `/etc/sysconfig/network-scripts` directory, restart the network service:

```
systemctl restart network
```

Use the `systemctl enable` command to configure a service to start at boot time. For example, to configure the `vsftpd` daemon to start at boot time:

```
systemctl enable vsftpd
```

This command does not actually start a service. You need to run the following command to start the service:

```
systemctl start vsftpd
```

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-<version>/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 an appropriately named file in the `/etc/sysctl.d` directory. Changes made to specific files in the `/etc/sysctl.d` directory take effect immediately when issuing the following command:

```
sysctl -p /etc/sysctl.d/<filename>
```

A series of files are involved with kernel settings. See the `sysctl` and `sysctl.d` man pages for details. To cause settings in all files to take effect immediately, issue the following command:

```
sysctl --system
```

The `firewalld` and `iptables` services (firewall) are often the cause of a problem with a client-server process. Use the `systemctl stop firewalld|iptables` command to temporarily stop `firewalld` and `iptables`, respectively, 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 service.

PAM modules might be causing authentication errors. Entries are usually written to the `/var/log/secure` log file when PAM is denying access. PAM is covered in another course.

SELinux stands for “Security-Enhanced Linux” and is covered in another course in the Oracle Linux curriculum map. SELinux is often the cause of a problem. 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.

```
getenforce
Enforcing
setenforce 0
getenforce
Permissive
```

Notice that “Current mode” is now set to “Permissive.”

The `sestatus` command also reports the status of SELinux:

```
sestatus
SELinux status: enabled
...
Current mode: 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/grub2/grub.cfg
  - /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 `grub2-install` to re-install the boot loader.



ORACLE

Copyright © 2019, 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/grub2/grub.cfg` 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`. `/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, and use `rpm` to install or upgrade software packages and other commands to rescue your environment.

You can reinstall the GRUB boot loader if it has been corrupted or overwritten by another operating system. Use the `grub2-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 `firewalld` and `iptables` rules or stop the service.
- DNS host name resolution:
  - Ensure `/etc/resolv.conf` contains correct entries.



Copyright © 2019, 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 are 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 parenthesis 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 `firewalld` and `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 command is useful in determining if your system has hardware-related errors?

- a. service
- b. ps
- c. lsmod
- d. dmesg



**ORACLE®**

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

**Answer: d**

## Summary

In this lesson, you should have learned about:

- Two-phased approach to troubleshooting
- Type of information needed to troubleshoot a problem
- Available operating system logs to assist in troubleshooting
- Use of the `dmesg` utility
- Available troubleshooting resources
- Causes of common problems
- Troubleshooting boot problems
- Typical causes of NFS problems



ORACLE®

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

## Practices for Lesson 20: Overview

The practices for this lesson involve troubleshooting the following common problems:

- System boots into single-user mode
- Status commands fail
- A cron job fails to run
- A user cannot log in
- File system does not mount
- Network connectivity problem
- You cannot log in to remote hosts using ssh
- Log file is not getting updated



Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

In these practices, 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 preceding lessons when necessary and attempt to diagnose and fix the problem.

Unauthorized reproduction or distribution prohibited. Copyright© 2019, Oracle and/or its affiliates.

GANG LIU (gangl@baylorhealth.edu) has a non-transferable license  
to use this Student Guide.