



Integrated Cloud Applications & Platform Services



# Oracle Linux System Administration II

Student Guide

D103155GC10 | D106226

Learn more from Oracle University at [education.oracle.com](https://education.oracle.com)



1003202019

**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.

## Contents

### 1 Course Introduction

- Course Goals 1-2
- Schedule 1-3
- Objectives 1-5
- Virtualization with Oracle VM Server for x86 1-6
- Classroom System Configuration 1-7
- Local Yum Repository 1-9
- Summary 1-10
- Practice 1: Overview 1-11

### 2 Network Addressing and Name Services

- Objectives 2-2
- Introduction to DHCP 2-3
- Configuring a DHCP Server 2-4
- Additional DHCP Server Declarations 2-6
- Starting and Stopping a DHCP Server 2-8
- Specifying Command-Line Arguments 2-9
- Configuring a DHCP Client 2-11
- Introduction to DNS 2-13
- Nameserver Types 2-14
- BIND 2-15
- Starting a DNS Cache-Only Nameserver 2-16
- Configuring an Authoritative Nameserver 2-17
- Zone Files 2-18
- The /etc/named.conf File 2-20
- The /etc/named.rfc1912.zones File 2-22
- Reverse Name Resolution 2-23
- rndc Utility 2-25
- host and dig Utilities 2-27
- Quiz 2-28
- Summary 2-29
- Practice 2: Overview 2-30

### **3 Authentication and Directory Services**

- Objectives 3-2
- Authentication Options 3-3
- Authentication Configuration GUI 3-4
- NIS Authentication 3-6
- Lightweight Directory Access Protocol (LDAP) 3-7
- OpenLDAP 3-9
- OpenLDAP Server Directories 3-10
- OpenLDAP Server Utilities 3-11
- OpenLDAP Client Utilities 3-12
- OpenLDAP Server Configuration 3-13
- The ldapmodify Utility 3-14
- The slappasswd Utility 3-15
- Loading the Standard Schemas 3-16
- Populating an OpenLDAP Directory 3-17
- Using the migrationtools Utilities 3-18
- Configuring LDAP Authentication 3-20
- Configuring User Authentication from an OpenLDAP Client 3-22
- Configuring Winbind Authentication 3-24
- Winbind Security Model Options 3-26
- Quiz 3-28
- Summary 3-29
- Practice 3: Overview 3-30

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

- Objectives 4-2
- Introduction to PAM 4-3
- PAM Configuration Files 4-4
- PAM Authentication Modules 4-5
- PAM Module Types 4-6
- PAM Control Flags 4-7
- PAM: Example #1 4-9
- PAM: Example #2 4-11
- Quiz 4-13
- Introduction to SELinux 4-14
- Summary 4-16
- Practice 4: Overview 4-17

### **5 Web and Email Services**

- Objectives 5-2
- Apache HTTP Server 5-3

Configuring Apache	5-4
Testing Apache	5-6
Apache Containers	5-7
Apache Virtual Hosts	5-9
Quiz	5-11
Email Program Classifications	5-12
Email Protocols	5-13
Postfix SMTP Server	5-15
Sendmail SMTP Server	5-16
Configuring Sendmail on a Client	5-18
Quiz	5-19
Summary	5-20
Practice 5: Overview	5-21

## **6 Installing Oracle Linux by Using PXE and Kickstart**

Objectives	6-2
Kickstart Installation Method	6-3
Kickstart File	6-4
Verifying the Kickstart File	6-5
Setting Up a Preboot Execution Environment (PXE) Server	6-6
What Is PXE?	6-7
PXE Execution Process	6-8
PXE Pre-Installation Steps	6-9
PXE Pre-Installation Steps (Cont.)	6-10
PXE Pre-Installation (Cont.)	6-12
Configuring DHCP and TFTP Services	6-13
Configuring TFTP Services	6-17
Beginning a Kickstart Installation	6-21
Quiz	6-22
Summary	6-23
Practice 6: Overview	6-24

## **7 Samba Services**

Objectives	7-2
Samba: Introduction	7-3
Samba Daemons and Services	7-4
Samba Server Configuration	7-5
Samba Server	7-7
Samba Server Types	7-8
Accessing Linux Shares from Windows	7-10
Accessing Windows Shares from Linux	7-12

Samba Utilities 7-13

Quiz 7-15

Summary 7-16

Practice 7: Overview 7-17

## **8 Advanced Software Package Management**

Objectives 8-2

Software Management with RPM and Yum 8-3

RPM Packages 8-5

The Binary RPM Build Process 8-6

BUILD Directory Structure 8-7

spec File to Build a Binary RPM Package 8-8

spec File: Example 8-10

Managing RPM-Based Software with Yum 8-11

Yum Cache 8-12

Yum History 8-14

Extending Yum Functionality with Plug-Ins 8-16

Popular Yum Plug-Ins 8-18

Using yum-config-manager 8-19

Managing a Yum Repo 8-20

Managing Errata 8-21

Important Resources for Errata Information 8-23

PackageKit Software Package Manager GUI 8-24

Using PackageKit Software Update 8-25

PackageKit Commands: Summary 8-26

Quiz 8-27

Summary 8-29

Practice 8: Overview 8-30

## **9 Advanced Networking**

Objectives 9-2

Network Bonding: Introduction 9-3

Network Bonding: Configuration 9-4

Using the NetworkManager TUI to Configure Network Bonding 9-6

Network Bonding Modes 9-7

Network Bonding Link Monitoring 9-9

Using the nmcli Utility to Configure Network Bonding 9-10

Using the nmcli Utility to Add the Slaves to the Bond 9-12

Activate the Bond 9-14

Viewing Network Bonding Information 9-15

Virtual Local Area Networks: Introduction 9-17

Using the nm-connection-editor GUI to Configure 802.1Q VLAN Tagging	9-18
Using the nmcli Utility to Configure VLAN Tagging	9-19
Viewing VLAN Information	9-21
Virtual Private Networks: Introduction	9-22
The libreswan RPM Package	9-23
Site-to-Site VPN	9-24
Site-to-Site VPN: Configuration	9-25
Example: “sitetosite” Connection	9-27
Multi-Factor Authentication	9-28
Two-Factor Authentication (2FA)	9-29
Pluggable Authentication Modules (PAM)	9-30
Stacking Multiple Modules	9-31
2FA PAM and Smart Cards	9-32
Internet Protocol Security (IPSec)	9-33
Port Address Translation (PAT)	9-34
IPSec with Port Address Translation	9-35
IPSec VPN Connection to On-Premises Network	9-36
802.3ad LAG (Link Aggregation Group)	9-37
Link Aggregation Control Protocol (LACP)	9-38
Virtual Port Channel (vPC)	9-39
Network Analysis	9-40
ip command	9-41
ss command	9-42
Tcpdump command	9-43
Wireshark	9-44
Capturing Packets with Wireshark	9-45
Analyzing Packets with Wireshark	9-46
tshark	9-47
Network Performance Tuning	9-48
top command	9-49
Virtual Memory Statistics (vmstat)	9-50
List Open Files (lsof)	9-51
Quiz	9-52
Summary	9-53
Practice 9: Overview	9-54

## **10 Implementing the XFS File System**

Objectives	10-2
XFS File System	10-3
Creating an XFS File System	10-4
xfs_growfs Utility	10-6

xfs_admin Utility	10-7
Enabling Disk Quotas on an XFS File System	10-8
xfs_quota Utility	10-10
Setting Project Quotas	10-12
Backing Up and Restoring XFS File Systems	10-13
XFS File System Maintenance	10-15
Quiz	10-16
Summary	10-19
Practice 10: Overview	10-20

## **11 Implementing the Btrfs File System**

Objectives	11-2
Btrfs: Introduction	11-3
Btrfs with Oracle Linux	11-5
Creating a Btrfs File System	11-6
The btrfs Utility	11-8
Btrfs Subvolumes	11-9
btrfs subvolume Utilities	11-11
Btrfs Snapshots	11-12
Taking a Snapshot of a File	11-13
Mounting a Subvolume or Snapshot	11-14
btrfs filesystem Utilities	11-16
The btrfs filesystem df Utility	11-17
btrfs filesystem show sync Utilities	11-19
The btrfs filesystem defragment Utility	11-20
The btrfs filesystem resize Utility	11-21
btrfs device Utilities	11-22
The btrfs device Utility: Examples	11-23
btrfs scrub Utilities	11-25
The btrfs scrub Utility: Examples	11-26
Converting Ext File Systems to Btrfs	11-28
Quiz	11-29
Summary	11-32
Practice 11: Overview	11-33

## **12 Storage Administration**

Objectives	12-2
Logical Volume Manager (LVM)	12-3
LVM Configuration: Example	12-4
Physical Volume Utilities	12-5
Volume Group Utilities	12-7

Logical Volume Utilities 12-9  
Making Logical Volumes Usable 12-11  
Backing Up and Restoring Volume Group Metadata 12-13  
LVM Thin Provisioning 12-14  
Snapper 12-16  
Redundant Array of Independent Disks (RAID) 12-19  
The mdadm Utility 12-21  
Making RAID Devices Usable 12-23  
Quiz 12-24  
Summary 12-25  
Practice 12: Overview 12-26

## **13 Advanced Storage Administration**

Objectives 13-2  
Disk Quotas 13-3  
Enabling Disk Quotas 13-4  
Summary of Quota Commands 13-6  
Encrypted Block Devices 13-9  
cryptsetup Utility 13-10  
Making an Encrypted Device Usable 13-12  
kpartx Utility 13-13  
Udev: Introduction 13-15  
Udev Rule Files and Directories 13-16  
Sample Udev Rules 13-17  
udevadm Utility 13-19  
Creating a Symbolic Link to a Device Node 13-21  
Quiz 13-22  
Summary 13-24  
Practice 13: Overview 13-25

## **14 File Sharing**

Objectives 14-2  
Introduction to NFS 14-3  
The NFS Server and RPC Processes 14-4  
NFS Server Configuration 14-6  
Starting the NFS Service 14-8  
The exportfs Utility 14-9  
NFS Client Configuration 14-10  
Automounting File Systems 14-12  
Direct Maps 14-13  
Indirect Maps 14-14

Host Maps 14-16  
Introduction to vsftpd 14-17  
vsftpd Configuration Options 14-18  
Quiz 14-20  
Summary 14-21  
Practice 14: Overview 14-22

## **15 Kerberos and IPA Services**

Objectives 15-2  
The Kerberos Protocol 15-3  
Configuring a Kerberos Server 15-4  
Configuring a Key Distribution Center 15-5  
Creating the Kerberos Database 15-6  
The kadmin.local command 15-7  
Starting and Enabling Kerberos Services 15-8  
Configuring Kerberos Authentication 15-9  
IPA Identity Management and Authentication Services 15-10  
Installing IPA Server with DNS 15-11  
Installing IPA Client 15-12  
IPA Client Discovery 15-13  
Configuring Advanced Options 15-14  
Configuring Password Options 15-16  
System Security Services Daemon 15-18  
Configuring SSSD Services 15-19  
Configuring SSSD Domains 15-21  
Quiz 15-23  
Summary 15-24  
Practice 15: Overview 15-25

# Course Introduction



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

# Course Goals

In this course, you learn how to:

- Configure network addressing and name services
- Configure authentication and directory services
- Configure Pluggable Authentication Modules (PAM)
- Configure web and email services
- Perform a PXE/Kickstart installation
- Configure Samba services
- Perform advanced software package management
- Perform advanced storage administration
- Perform advanced network configuration



ORACLE®

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

## Schedule

Session	Module
<b>Day 1</b>	Lesson 1: Course Introduction Lesson 2: Network Addressing and Name Services Lesson 3: Authentication and Directory Services
<b>Day 2</b>	Lesson 4: Pluggable Authentication Modules (PAM) Lesson 5: Web and Email Services Lesson 6: Installing Oracle Linux with PXE and Kickstart
<b>Day 3</b>	Lesson 7: Samba Services Lesson 8: Advanced Software Package Management Lesson 9: Advanced Networking



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

# Schedule

Session	Module
<b>Day 4</b>	Lesson 10: Implementing the XFS File System Lesson 11: Implementing the Btrfs File System Lesson 12: Storage Administration
<b>Day 5</b>	Lesson 13: Advanced Storage Administration Lesson 14: File Sharing Lesson 15: Kerberos and IPA Services



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

## Objectives

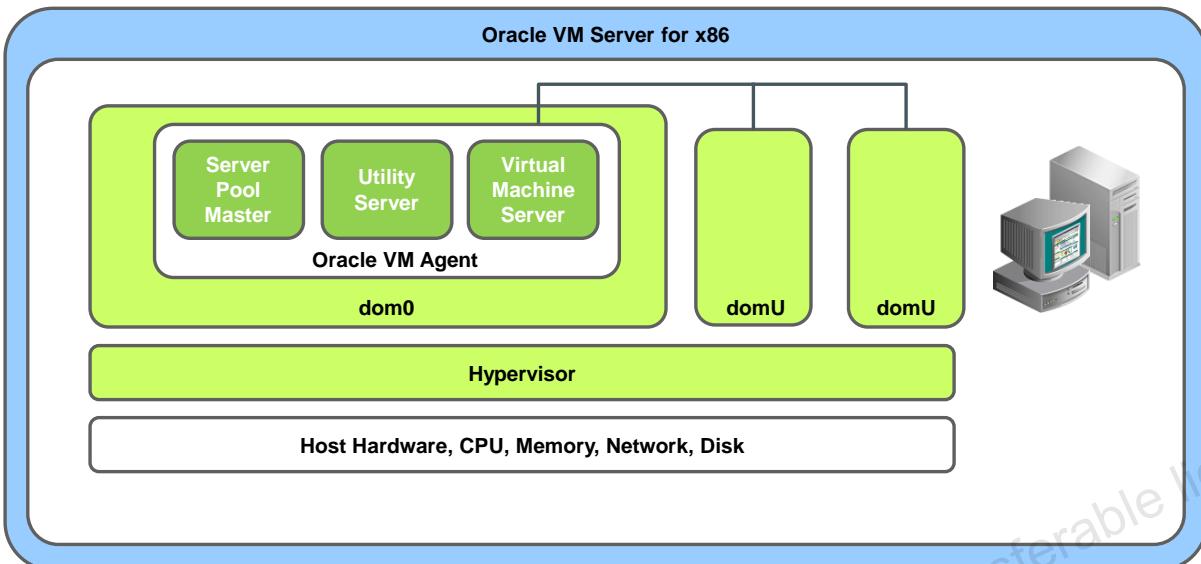
After completing this lesson, you should be able to:

- Describe the classroom environment used for the practice sessions
- Log in to a virtual machine on your student desktop



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

# Virtualization with Oracle VM Server for x86



ORACLE®

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

## Virtualization

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

## Hypervisor

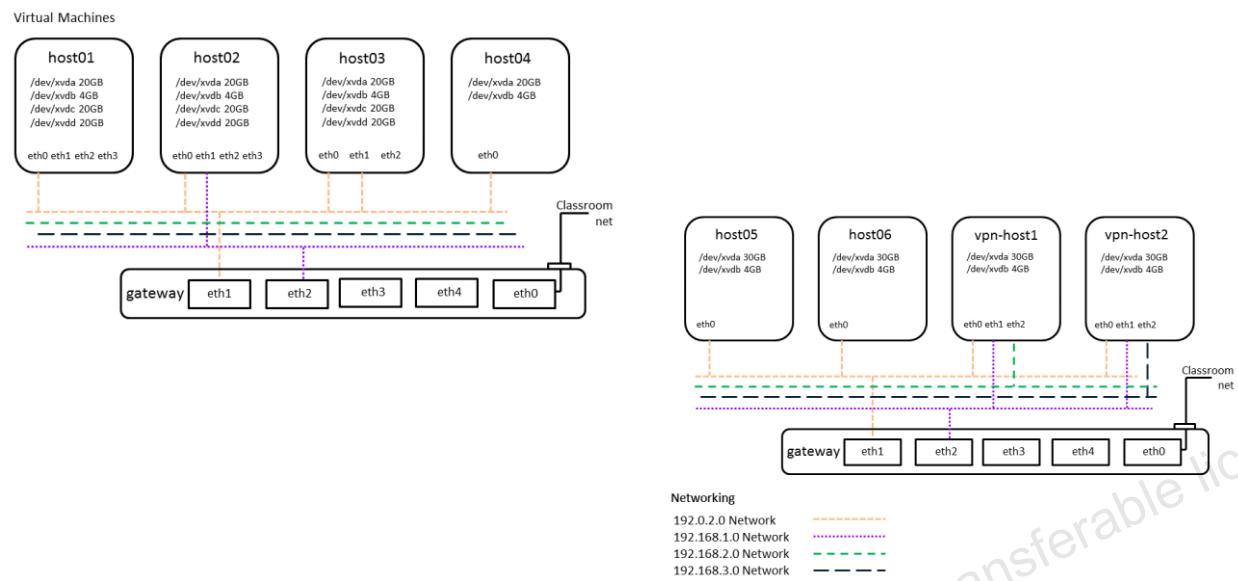
A hypervisor is a virtualization software, also known as a virtual machine monitor (VMM), that creates and runs the virtual machines. There are two different types of hypervisors:

- A type 2 hypervisor, such as VirtualBox, that runs on the host operating system and, in turn, runs the guest virtual machines. A type 2 hypervisor is a distinct software layer.
- A type 1 hypervisor, such as Oracle VM Server for x86 or VMware ESX, that provides a small footprint host operating system and exposes the server's resources to the guest virtual machines that run directly on top of the hypervisor. Because this type of hypervisor communicates directly with the hardware, it is known as a bare metal hypervisor.

## Oracle VM Server for x86 Domains

Oracle VM Server for x86 guests are referred to as *domains*. Dom0 is always present, providing management services for the other domains running on the same server.

# Classroom System Configuration



ORACLE®

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

The **host02** has two network interfaces:

- **eth0 is on the 192.0.2 subnet.**
- **eth1 is on the 192.168.1.0 subnet.**

The **host03** VM has two network interfaces:

- **eth0 and eth1 are on the 192.0.2 subnet.**

The **host04**, **host05**, and **host06** VMs have a single network interface, **eth0**, which is on the 192.0.2 subnet.

The **vpn-host1** has three network interfaces:

- **eth0 is on the 192.0.2 subnet.**
- **eth1 is on the 192.168.1.0 subnet.**
- **eth2 is on the 192.168.2.0 subnet.**

The **vpn-host2** has three network interfaces:

- **eth0 is on the 192.0.2 subnet.**
- **eth1 is on the 192.168.1.0 subnet.**
- **eth2 is on the 192.168.3.0 subnet.**

## Local Yum Repository

- The host $0n$  and vpn-host $n$  VMs are configured to access a local Yum Repository on gateway.
- The following `local.repo` file exists on each host $0n$  and vpn-host $n$  VM, which points to the Yum repository on gateway (192.0.2.1):

```
# cat /etc/yum.repos.d/local.repo
[local_repo]
Name="local_repo packages"
baseurl=http://192.0.2.1/local_repo/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```



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

A local Yum repository exists on the gateway VM. Students use the `yum` command to install and upgrade software packages on host $0n$  VMs from this local Yum repository.

The following `local.repo` file exists in the `/etc/yum.repos.d` directory on each host $0n$  VM, which points to the local Yum repository on gateway:

```
# cat /etc/yum.repos.d/local.repo
[local_repo]
name=local_repo packages
baseurl=http://192.0.2.1/local_repo
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

On gateway, the Yum repository is provided as an `http` service. The following directory on gateway contains the `repodata` and all the `rpm` files that students need to install in this course:

```
[root@gateway ~]# ls -l /var/www/html/local_repo/
total 20
drwxr-xr-x 2 root root 4096 ... repodata
drwxr-xr-x 2 root root 16384 ... rpms
```

## Summary

In this lesson, you should have learned how to:

- Describe the classroom environment used for the practice sessions
- Log in to a virtual machine on your student desktop



ORACLE®

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

## Practice 1: Overview

The practices for this lesson cover the following:

- Exploring the gateway environment
- Exploring the host01 VM
- Exploring the host02 VM
- Exploring the host03 VM
- Exploring the host04 VM
- Exploring the host05 VM
- Exploring the host06 VM
- Exploring the vpn-host1 VM
- Exploring the vpn-host2 VM
- Logging off from your system



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.

# Network Addressing and Name Services

ORACLE®

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

# Objectives

After completing this lesson, you should be able to:

- Describe DHCP
- Configure a DHCP server
- Start and stop a DHCP server
- Configure a DHCP client and request a lease
- Describe DNS
- Describe nameserver types
- Describe BIND
- Configure a cache-only nameserver
- Describe and configure zone files
- Describe and configure reverse name resolution
- Use the `rndc` utility
- Use the `host` and `dig` utilities



ORACLE®

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

## Introduction to DHCP

- Client machines automatically obtain network configuration information from a DHCP server.
- The client “leases” the network information.
  - The terms of the lease are configurable.
  - The lease is renewed automatically by the client while the network is in use.
- The DHCP server can provide static IP addresses.
- DHCP is broadcast based.
  - This requires the client and the server to be on the same subnet.



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

Dynamic Host Configuration Protocol (DHCP) allows client machines to automatically obtain network configuration information from a DHCP server each time they connect to the network. The DHCP server is configured with a range of IP addresses and other network configuration parameters.

When the client machine is configured to use DHCP, the client daemon, `dhclient`, contacts the server daemon, `dhcpd`, to obtain the networking parameters. Because DHCP is broadcast based, both the client and the server must be on the same subnet.

The server provides a lease on the IP address to the client. The client can request specific terms of the lease, such as its duration. The server can also be configured to limit the terms of the lease. While connected to the network, `dhclient` automatically renews the lease before it expires. You can configure the DHCP server to provide the same IP address each time to specific clients.

The advantages of using DHCP include ease of adding a new client machine to the network and centralized management of IP addresses. In addition, the number of total IP addresses needed is reduced because IP addresses can be reused. DHCP is also useful if you want to change the IP addresses of a large number of systems. Instead of reconfiguring each system individually, edit the DHCP configuration file on the server and enter the new set of IP addresses.

# Configuring a DHCP Server

- Install the `dhcp` package.

```
# yum install dhcp
```

- Specify network information for clients in the `/etc/dhcp/dhcpd.conf` configuration file. Example:

```
option subnet-mask          255.255.255.0;
option domain-name         "example.com";
option domain-name-servers 192.0.2.1;
option broadcast-address   192.168.1.255;
default-lease-time        21600;
max-lease-time            43200;
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.254;
}
```



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

To configure a system as a DHCP server, install the `dhcp` package:

```
# yum install dhcp
```

The main configuration file for DHCP is `/etc/dhcp/dhcpd.conf`. Use this file to store network information for the clients. A sample configuration file is also installed with the `dhcp` package:

`/usr/share/doc/dhcp-<version>/dhcpd.conf.sample`. You can copy this file to `/etc/dhcp/dhcpd.conf` and use it as a template.

## Options

Information in the `option` lines is sent to each client when it requests a lease. Each option declaration begins with the “`option`” keyword, followed by the option name and the option data. Each option declaration must be terminated with a semicolon (`;`). See the `dhcp-options (5)` man page for a description of available options.

## Lease Times

There are time-related configuration entries in the sample configuration file. These are described as follows. Each of these entries is also terminated with a semicolon.

- `default-lease-time`:** Specifies the number of seconds the IP lease remains valid if the client requesting the lease does not specify a duration
- `max-lease-time`:** Specifies the maximum number of seconds allowed for a lease

## Subnet Declaration

The `subnet` declaration defines a range of IP addresses that the DHCP server can assign to clients. The example in the slide includes a subnet declaration for the `192.168.1.0` subnet. The declaration defines a netmask value and a range of IP addresses between `192.168.1.200` and `192.168.1.254`. The range of addresses is surrounded by curly braces (`{ }` ).

You can declare multiple subnets and specify parameters inside or outside of the braces. Parameters specified within the braces apply to the clients on the subnet. In the following example, two subnet declarations are defined. The options apply only to the clients on the `192.168.1.0` subnet:

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    option domain-name          "example.com";  
    option domain-name-servers 192.168.1.1;  
    range 192.168.1.200 192.168.1.254;  
}  
  
subnet 192.168.2.0 netmask 255.255.255.0 {  
    range 192.168.2.200 192.168.2.254;  
}
```

Parameters configured outside of a subnet declaration are global and apply to all client systems.

## Additional DHCP Server Declarations

- Host declarations for static IP address assignment:
  - Assign a static IP address to a specific client system.
  - Include the MAC address and the static IP address within the host declaration.

```
host <name> { ... }
```

- Shared-network declarations for multiple subnets:
  - Group subnets that share the same physical network within the shared-network declaration

```
shared-network <name> { ... }
```

- Group declarations apply global parameters:
  - Use them to apply global parameters to a group of declarations.
  - Shared networks, subnets, and hosts can be grouped.

```
group { ... }
```



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

### Host Declaration

Use a `host` declaration to provide a static IP address to a specific client system. Include the MAC address of the client and the static IP address to be assigned to the client. Example:

```
host host01 {
    hardware ethernet          00:16:3E:00:01:01;
    fixed-address              192.168.1.101;
    max-lease-time             84600;
}
```

In this example, IP address of 192.168.1.101 is always assigned to the system with the MAC address of 00:16:3E:00:01:01. The `max-lease-time` included within the declaration is specific to this host and overrides the global parameter defined outside the curly brackets.

### Shared-Network Declaration

Declare all subnets that share the same physical network within a `shared-network` declaration. Parameters within the shared network, but outside the enclosed subnet declarations, are considered to be global parameters.

The following is an example of a shared-network declaration with two subnets. The `routers` parameter applies to both subnets:

```
shared-network name {  
    option routers 192.168.0.254;  
    subnet 192.168.1.0 netmask 255.255.252.0 {  
        range 192.168.1.200 192.168.1.254;  
    }  
    subnet 192.168.2.0 netmask 255.255.252.0 {  
        range 192.168.2.200 192.168.2.254;  
    }  
}
```

## Group Declaration

Use the `group` declaration to apply global parameters to a group of declarations. Shared networks, subnets, and hosts can be grouped. The following is an example of a `group` declaration with two host declarations:

```
group {  
    option routers 192.168.1.254;  
    host host01 {  
        hardware ethernet 00:16:3E:00:01:01;  
        fixed-address 192.168.1.101;  
    }  
    host host02 {  
        hardware ethernet 00:16:3E:00:01:02;  
        fixed-address 192.168.1.102;  
    }  
}
```

# Starting and Stopping a DHCP Server

- To enable the `dhcpd` service to start at boot time:

```
# systemctl enable dhcpd
```

- A symbolic link is created when you enable a service.

- To disable the `dhcpd` service from starting at boot time:

```
# systemctl disable dhcpd
```

- The symbolic link is removed when you disable the service.

- To start the `dhcpd` service:

```
# systemctl start dhcpd
```

- The service fails to start if the `/var/lib/dhcpd/dhcpd.leases` file does not exist.

- To stop the `dhcpd` service:

```
# systemctl stop dhcpd
```



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

Use the `systemctl` command to enable the `dhcpd` service to start at boot time:

```
# systemctl enable dhcpd
```

Created symlink from `/etc/systemd/system/multi-user.target.wants/dhcpd.service` to `/usr/lib/systemd/system/dhcpd.service`.

Notice that the command enables a service by creating a symbolic link for the lowest-level system-state target at which the service starts. In the example, the command creates the symbolic link `dhcpd.service` for the `multi-user` target.

Use the `systemctl` command to disable the `dhcpd` service from starting at boot time. The symbolic link is removed when the service is disabled:

```
# systemctl disable dhcpd
```

Removed symlink `/etc/systemd/system/multi-user.target.wants/dhcpd.service`.

Use the `systemctl` command to start the `dhcpd` service:

```
# systemctl start dhcpd
```

The `dhcpd` service fails to start if the `/var/lib/dhcpd/dhcpd.leases` file does not exist. You can use the `touch` command to create the file. The `dhcpd.leases` file stores the client lease information. Do not edit this file.

## Specifying Command-Line Arguments

- Copy the `/usr/lib/systemd/system/dhcpd.service` file to the `/etc/systemd/system/` directory:

```
# cp /usr/lib/systemd/system/dhcpd.service
/etc/systemd/system/
```

- Edit the “`ExecStart`” line in the `/etc/systemd/system/dhcpd.service` file:
  - This example adds `eth2` as a command-line argument.
  - This causes the DHCP server to listen only on `eth2`.

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf
-user dhcpcd -group dhcpcd --no-pid eth2
```

- Enabling the service creates a symbolic link to the `/etc/systemd/system/dhcpd.service` file.

```
# systemctl enable dhcpcd
```



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

To specify command-line arguments and options when the `dhcpcd` service is started, copy the `/usr/lib/systemd/system/dhcpd.service` file to the `/etc/systemd/system/` directory:

```
# cp /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/
```

You can then edit the `/etc/systemd/system/dhcpd.service` file and append command-line arguments and options to the `ExecStart` line.

For example, if your DHCP server has multiple network interfaces (`eth0`, `eth1`, `eth2`) but you want only the `dhcpcd` service to listen for DHCP requests on `eth2`, include `eth2` as a command-line argument:

```
# vi /etc/systemd/system/dhcpd.service
```

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpcd -group
dhcpcd --no-pid eth2
```

When you enable the service to start at boot time, a symbolic link is created to the `dhcpcd.service` file in the `/etc/systemd/system/` directory rather than the file in the `/usr/lib/systemd/system/` directory:

```
# systemctl enable dhcpcd
```

```
Created symlink from /etc/systemd/system/multi-
user.target.wants/dhcpcd.service to /etc/systemd/system/dhcpcd.service.
```

Refer to the `dhcpcd(8)` man page for additional command-line options and arguments. Some of the available options are described:

- **`-p <port>`:** Specifies the UDP port number on which `dhcpcd` listens. The default is port 67.
- **`-f`:** Runs the `dhcpcd` as a foreground process instead of a background daemon. This is helpful when debugging a problem.
- **`-d`:** Logs the DHCP server daemon to the standard error descriptor. This is helpful when debugging. If this is not specified, `dhcpcd` logs all output using `syslog`.
- **`-cf <filename>`:** Specifies the location of the configuration file. The default configuration file is `/etc/dhcp/dhcpcd.conf`.
- **`-lf <filename>`:** Specifies the location of the lease database file. The default lease file is `/var/lib/dhcpcd/dhcpcd.leases`.
- **`-q`:** Specifies to be quiet at startup. This suppresses printing of the entire copyright message when starting the daemon.
- **`--no-pid`:** Disables writing pid (Process ID) files. With this option, the service does not check for an existing server process.

# Configuring a DHCP Client

1. Install the `dhclient` package.
2. Set `BOOTPROTO=dhcp` in the `/etc/sysconfig/network-scripts/ifcfg-<device>` file.
3. Ensure that the network service is running on the client.
4. Optionally, enter any custom configuration information in the DHCP client configuration file, `/etc/dhcp/dhclient.conf`.
5. Run the `dhclient` command to request a lease from the server.

After being configured to use DHCP, the `dhclient` command runs at boot time.



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

To configure a system as a DHCP client, install the `dhclient` package:

```
# yum install dhclient
```

Change the `BOOTPROTO` directive in the `/etc/sysconfig/network-scripts/ifcfg-<interface>` file for the device to `dhcp`. For example, to use DHCP on `eth1`, perform the following:

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
BOOTPROTO=dhcp
```

- You also need to start the network service on the DHCP client.  
`# systemctl start network`
- The next time the client system connects to the network, `dhclient` requests a lease from the DHCP server and configures the client's network interface. You can also run `dhclient` from the command line to request a lease and make a connection:  
`$ dhclient`
- To request on a specific interface, include the interface as an argument. The following example only requests a lease for `eth1`:  
`$ dhclient eth1`

The DHCP client configuration file, /etc/dhcp/dhclient.conf, is required only for custom configurations. A sample file exists in /usr/share/doc/dhclient-<version>/dhclient.conf.example.

When the client has requested and established a lease, information about the lease is stored in /var/lib/dhclient/dhclient.leases. Example:

```
# cat /var/lib/dhclient/dhclient.leases
lease {
    interface "eth1";
    fixed-address 192.168.1.251;
    option subnet-mask 255.255.255.0;
    option dhcp-lease-time 21600;
    option dhcp-message-type 5;
    option domain-name-servers 192.0.2.1;
    option dhcp-server-identifier 192.168.1.103;
    option broadcast-address 192.168.1.255;
    option domain-name "example.com";
    renew 5 2015/04/17 20:18:58;
    rebind 6 2015/04/17 23:15:26;
    expire 6 2015/04/17 00:00:26;
}
```

## Introduction to DNS

- DNS is a network service that maps, or resolves, domain names to their respective IP addresses.
  - `wiki.us.oracle.com > 139.185.51.248`
- DNS performs the function of the `/etc/hosts` file, but on the Internet.
- The DNS database is hierarchical and distributed.
  - Each level of hierarchy is divided by a period (.)
- Resolution occurs from right to left:
  1. `.com` is resolved.
  2. `oracle.com` is resolved.
  3. `us.oracle.com` is resolved.
  4. The IP address of `wiki.us.oracle.com` is returned to the client.
- DNS servers are called nameservers.



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

Domain Name System (DNS) is a network service that maps, or resolves, domain names to their respective IP addresses. It reduces the need for users to remember IP addresses because they can refer to machines on the network by name. The mapping done by `/etc/hosts` on a small local area network (LAN) is handled by DNS on large networks, including the Internet.

The DNS database is hierarchical and distributed. Each level of the hierarchy is divided by a period (.). Consider the following example of a fully qualified domain name (FQDN):

`wiki.us.oracle.com`.

The root domain is represented by a period (.) and is frequently omitted except in zone files. The top-level domain in this example is `.com`, `oracle` is a subdomain of `.com`, `us` is a subdomain of `oracle`, and `wiki` is the host name. For administrative purposes, each of these domains is grouped into zones. A DNS server, called a nameserver, holds all the information to resolve all domains within a zone. The DNS server for a zone also holds pointers to DNS servers responsible for resolving a domain's subdomains.

When a client requests resolution of `wiki.us.oracle.com` from a nameserver, if the nameserver cannot resolve the FQDN, it queries a root nameserver, which returns the nameserver that can resolve `.com`. This nameserver is queried and returns the nameserver to resolve `oracle.com`. This nameserver is queried and returns the nameserver to resolve `us.oracle.com`, which is queried and returns the IP address for the FQDN to the client.

# Nameserver Types

- Authoritative nameservers:
  - Answer queries about names that are part of their zones only
  - Can be either primary (master) or secondary (slave)
- The primary nameserver holds the master copy of zone data.
- Secondary nameservers copy zone data from the master nameserver or another slave nameserver.
- Caching-only, or recursive, nameservers:
  - Offer resolution services, but are not authoritative
  - Cache the answers from previous queries
  - Respond to queries from the cache if possible
    - Otherwise, they forward the query to an authoritative server.



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

An authoritative nameserver responds to queries about names that are part of their zones only. Authoritative nameservers can be either primary (master) nameservers or secondary (slave) nameservers. Each zone has at least one authoritative DNS server. A DNS query returns information about a domain and specifies which DNS server is authoritative for that domain.

A primary nameserver, or master nameserver, is the authoritative server that holds the master copy of zone data. Secondary nameservers, or slave nameservers, are also authoritative but copy zone information from the master nameserver or from another slave nameserver. A nameserver can also serve as a primary or secondary server for multiple zones at the same time.

Caching-only nameservers, or recursive nameservers, offer resolution services, but they are not authoritative for any zone. These DNS cache nameservers store answers to previous queries in cache (memory) for a fixed period of time. When a caching-only nameserver receives a query, it answers from cache if it can. If it does not have the answer in cache, it forwards the query to an authoritative server.

Although it is not recommended for reasons of security, nameservers can also be configured to give authoritative answers to queries in some zones, while acting as a caching-only nameserver for all other zones.

# BIND

- The DNS server included in Oracle Linux is called BIND.
- The BIND server daemon is `named`.
- The remote administration utility is `rndc`.
- Configuration files and directories include:
  - `/etc/named.conf`: The main configuration file
  - `/var/named`: The default directory for storing zone files
  - `/etc/named.rfc1912.zones`: The base configuration file for implementing a caching-only nameserver
  - `/var/named/named.ca`: Contains a list of the 13 root authoritative DNS servers
- The default installation provides a caching-only nameserver.



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

The DNS server included in Oracle Linux is called Berkeley Internet Name Domain (BIND). BIND includes the DNS server daemon, `named`, tools for working with DNS such as the `rndc` administration utility, and several configuration files.

The primary BIND configuration files and directories are:

- `/etc/named.conf`: The main configuration file that lists the location and characteristics of all your domain's zone files. See the `named.conf` (5) man page for configuration options.
- `/var/named`: The default directory in which zone files are stored. This is specified by the "directory" option in the `/etc/named.conf` file.
- `/etc/named.rfc1912.zones`: The base configuration file for implementing a caching-only nameserver. This file contains five defined zones.
- `/var/named/named.ca`: A file that contains IP addresses of the 13 root authoritative DNS servers for the root domain.

The default installation of the `bind` package provides a caching-only nameserver. This nameserver is not authoritative for any domain. It stores only the results of queries in memory.

# Starting a DNS Cache-Only Nameserver

## 1. Install BIND:

```
# yum install bind
```

## 2. Add to the beginning of /etc/resolv.conf:

```
nameserver 127.0.0.1
```

## 3. If NetworkManager is running, add to

/etc/sysconfig/network-scripts/ifcfg-<interface>:

```
DNS1=127.0.0.1
```

## 4. Start the network service (if necessary):

```
# systemctl start network
```

## 5. Enable and start the named service:

```
# systemctl enable named  
# systemctl start named
```



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

To configure a system as a DNS cache-only nameserver, perform the following steps (as the `root` user):

- Install the `bind` package:  
`# yum install bind`
- Add the following line to the beginning of the `/etc/resolv.conf` file. This line indicates use of the local system as the primary nameserver:  
`nameserver 127.0.0.1`
- If NetworkManager is running, add the following line to the `/etc/sysconfig/network-scripts/ifcfg-<interface>` file:  
`DNS1=127.0.1.1`
- Ensure that the network service is running:  
`# systemctl start network`
- Enable and start the named service:  
`# systemctl enable named  
# systemctl start named`

# Configuring an Authoritative Nameserver

- Define the zones in /etc/named.conf. Example:

```
zone "example.com" {
    type master;
    file "data/master-example.com";
    allow-update { key "rndckey"; };
    notify yes;
};
```

- This example specifies the **type** as `master`, meaning the nameserver is authoritative for the `example.com` domain.



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

## Zone Files

- Zone files:
  - Store information about domains in the DNS database
  - Contain directives (optional) and resource records
- Resource records contain (not all fields are required):
  - Name: The domain name or IP address
  - TTL: Time to live
  - Class: Always IN for Internet
  - Type: Record type
  - Data: Varies with record type
- Common types of resource records include:
  - A, CNAME, MX, NS, PTR, SOA



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

Information about domains in the DNS database is stored in zone files. A zone file consists of directives and resource records. Directives tell the nameserver to perform tasks or apply special settings to the zone. Resource records define the parameters of the zone and store host information. Directives are optional, but resource records are required.

A resource record has the following fields (some fields are optional, depending on the **Type**):

- **Name:** The domain name or IP address
- **TTL:** Time to live, maximum time a record is cached before checking for a newer one
- **Class:** Always IN for Internet
- **Type:** Record type
- **Data:** Varies with record type

More than 30 types of resource records exist. The more common ones are:

- **A:** IPv4 address
- **CNAME:** Canonical name or alias
- **MX:** Mail exchange, specifies the destination for mail addressed to the domain
- **NS:** Nameserver, specifies the system that provides DNS records for the domain
- **PTR:** Maps an IP address to a domain name for reverse name resolution
- **SOA:** Start of authority, designates the start of a zone

The following is an example of a zone file:

```
$TTL 86400      ; 1 day
example.com IN SOA dns.example.com. root@example.com. (
                  57          ; serial
                  28800       ; refresh (8 hours)
                  7200        ; retry (2 hours)
                 2419200    ; expire (4 weeks)
                  86400       ; minimum (1 day)
)
IN NS dns.example.com.

dns      IN      A      192.0.2.1
example.com IN      A      192.0.2.1
host01   IN      A      192.0.2.101
host02   IN      A      192.0.2.102
host03   IN      A      192.0.2.103
```

The **\$TTL** entry is a directive that defines the default time to live for all resource records in the zone. Each resource record can have a **TTL** value, which overrides this global directive.

The next line in the example is the SOA record. All zone files must have one SOA record. The following information is included in the SOA record:

- **example.com**: The name of the domain
- **dns.example.com.**: The FQDN of the nameserver
- **root@example.com.**: The email address of the user who is responsible for the zone
- **serial**: A numerical value that is incremented each time the zone file is altered to indicate when it is time for the `named` service to reload the zone
- **refresh**: The elapsed time after which the primary nameserver notifies secondary nameservers to refresh their database
- **retry**: The time to wait after which a refresh fails before trying to refresh again
- **expire**: The time after which the zone is no longer authoritative and the root nameservers must be queried
- **minimum**: The amount of that time that other nameservers cache the zone's information

The **NS** (Nameserver) record announces authoritative nameservers for a particular zone by using the format:  
IN NS dns.example.com.

The **A** (Address) records specify the IP address to be assigned to a name by using the format:

```
hostname IN A IP-address
```

## The /etc/named.conf File

The /etc/named.conf file contains the following sections:

- options
  - Defines global server configuration options
- logging
  - Enables logging
  - /var/named/data/named.run
- zone
  - Specifies authoritative servers for the root domain
  - /var/named/named.ca
- include
  - Specifies files to include
  - /etc/named.rfc1912.zones



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

The default configuration of the /etc/named.conf file provides a caching-only nameserver. The file has four main sections described as follows.

### options

The options statement defines global server configuration options and sets defaults for other statements. The following options are defined in the default /etc/named.conf file:

- **listen-on**: Instructs named to listen on port 53 on the local system for both IPv4 and IPv6 queries
- **directory**: Specifies the default working directory for the named service
- **dump-file**: Specifies the location where BIND dumps the database (cache) in the event of a crash
- **statistics-file**: Specifies the location to which data is written when the command rndc stats is issued
- **memstatistics-file**: Specifies the location to which BIND memory usage statistics are written
- **allow-query**: Specifies which IP addresses (localhost by default) are allowed to query the server
- **recursion**: Instructs the nameserver to perform recursive queries. Recursive queries cause a nameserver to query another nameserver if necessary to respond with an answer.

The options defined in the default /etc/named.conf file continue:

- **dnssec-enable:** Specifies that a secure DNS service is being used
- **dnssec-validation:** Instructs the nameserver to validate replies from DNSSEC-enabled (signed) zones
- **dnssec-lookaside:** Enables DNSSEC Lookaside Validation (DLV) by using /etc/named.iscdlv.key

### **logging**

The logging statement turns on logging and causes messages to be written to the data/named.run file. The severity parameter controls the logging level. A severity value of dynamic means assume the global level defined by either the command-line parameter -d or by running the rndc trace command. The default logging statement follows:

```
logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
}
```

### **zone**

The default zone section specifies the initial set of root servers by using a hint zone, whose name is a period (.). This zone specifies that the nameserver must look in /var/named/named.ca for IP addresses of authoritative servers for the root domain when the nameserver starts or does not know which nameserver to query. The default zone section follows:

```
zone "." IN {
    type hint;
    file "named.ca";
};
```

Zone options include the following:

- **type:** Specifies the zone type, such as master, delegation-only, forward, hint, or slave. Type master designates the nameserver as authoritative for this zone. A zone is set as master if the zone file resides on this system.
- **file:** Specifies the name of the zone file, which is stored in the working directory defined by the directory option
- **allow-update:** Specifies which hosts are allowed to dynamically update information in their zone

### **include**

The include statement allows files to be included. This can be done for readability, ease of maintenance, or so that potentially sensitive data can be placed in a separate file with restricted permissions. This include statement includes the /etc/named.rfc1912.zones file as though it were present in this file.

## The /etc/named.rfc1912.zones File

This file specifies five predefined zones:

- **localhost.localdomain**
  - Specifies that `localhost.localdomain` points to `127.0.0.1`
- **localhost**
  - Sets up the normal server on the local system
- **1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa**
  - Sets up IPv6 reverse name resolution
- **1.0.0.127.in-addr.arpa**
  - Sets up IPv4 reverse name resolution
- **0.in-addr.arpa**
  - Specifies that IP addresses that start with 0 have their reverse lookup handled by the local server



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

The `/etc/named.rfc1912.zones` file is listed in the `include` section of the `/etc/named.conf` file. The `/etc/named.rfc1912.zones` file contains five zone sections.

Domains are grouped into zones, and zones are configured through the use of zone files. The `zone` statement defines the characteristics of a zone, the location of its zone file, and zone-specific options, which override the global options statements. The following zones are defined in the `/etc/named.rfc1912.zones` file:

- **localhost.localdomain:** Specifies that `localhost.localdomain` points to `127.0.0.1`, preventing the local server from looking upstream for this information
- **localhost:** Sets up the normal server on the local system
- **1.0.ip6.arpa:**
  - Sets up IPv6 reverse name resolution
- **1.0.0.127.in-addr.arpa:** Sets up IPv4 reverse name resolution
- **0.in-addr.arpa:** Specifies that IP addresses that start with 0 have their reverse lookup handled by the local server, preventing the local server from looking upstream

## Reverse Name Resolution

- Normal, or forward resolution returns an IP address when the domain name is known.
- Reverse name resolution returns the domain name when an IP address is known.
- DNS implements reverse name resolution by use of the following special domains:
  - **in-addr.arpa**: For IPv4
  - **ip6.arpa**: For IPv6
- Zone names reverse the network portion of the IP address and append the special domain name:
  - 2.0.192.in-addr.arpa
- Resource records use type PTR.



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

DNS also provides reverse name resolution, which returns a domain name for a given IP address. DNS implements reverse name resolution by use of the following special domains:

- **in-addr.arpa**: For IPv4
- **ip6.arpa**: For IPv6

The zone characteristics are defined in /etc/named.conf, for example:

```
zone "2.0.192.in-addr.arpa" IN {  
    type master;  
    file "data/reverse-192.0.2";  
    allow-update { key "rndckey"; };  
    notify yes;  
};
```

The zone name consists of **in-addr.arpa** preceded by the network portion of the IP address for the domain. In this example, the network is 192.0.2, which in reverse is 2.0.192.

Resource records in these domains have **Name** fields that contain IP addresses and **Type** fields of PTR.

The following is an example of the 2.0.192.in-addr.arpa zone file:

```
$TTL 86400      ; 1 day
2.0.192.in-addr.arpa IN SOA dns.example.com. root@example.com. (
    57          ; serial
    28800       ; refresh (8 hours)
    7200        ; retry (2 hours)
    2419200     ; expire (4 weeks)
    86400        ; minimum (1 day)
)
IN NS dns.example.com.

1      IN      PTR      dns
1      IN      PTR      example.com
101    IN      PTR      host01
102    IN      PTR      host02
103    IN      PTR      host03
```

## rndc Utility

- `rndc` is a command-line administration tool for `named`.
- Use the `rndc` key to prevent unauthorized access.
- The `rndc` key is generated by using the following command:

```
# rndc-confgen -a
```

- Configure `named` to use the key in `/etc/named.conf`.
- Type `rndc` to display usage of the utility and a list of available commands.



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

The `rndc` utility is a command-line tool to administer the `named` service, both locally and from a remote machine. To prevent unauthorized access to the service, `rndc` must be configured to listen on the selected port (port 953 by default), and an identical key must be used by both the service and the `rndc` utility. The `rndc` key is generated by using the following command:

```
# rndc-confgen -a
wrote key file "/etc/rndc.key"
```

This command creates the `/etc/rndc.key` file, which contains the key. To configure `named` to use the key, include the following entries in `/etc/named.conf`:

```
include "/etc/rndc.key";
controls {
    inet 127.0.0.1 allow { localhost; } keys { "rndckey"; }
};
```

The `include` statement allows files to be included so that potentially sensitive data can be placed in a separate file with restricted permissions. To ensure that only `root` can read the file, enter the following:

```
# chmod o-rwx /etc/rndc.key
```

The `controls` statement defines access information and the various security requirements necessary to use the `rndc` command.

- **inet:** The example allows you to control `rndc` from a console on the localhost (127.0.0.1).
- **keys:** Keys are used to authenticate various actions and are the primary access control method for remote administration. The example specifies using `rndckey`, which is defined in the `/etc/rndc.key` include file.

Type `rndc` to display usage of the utility and a list of available commands:

```
# rndc
Usage: rndc [-c config] [-s server] [-p port] [-key key-file] [-y key] [-V] command
Command is one of the following:
reload    Reload configuration file and zones
...
reconfig Reload configuration file and new zones only.
stats      Write server statistics to the statistics file.
querylog   Toggle query logging.
dumpdb     Dump cache(s) to the dump file (named_dump.db)
stop       Save pending updates to master files and stop the server.
halt       Stop the server without saving pending updates.
...
status     Display status of the server
...
```

The following is an example of some of the `rndc` commands:

Use the `rndc status` command to check the current status of the `named` service:

```
# rndc status
number of zones: 3
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/1000
tcp clients: 0/100
server is up and running
```

Use the `rndc reload` command to reload both the configuration file and zones:

```
# rndc reload
server reload successful
```

## host and dig Utilities

- host and dig are command-line tools to perform DNS lookups.
- The host command has more options.
- Examples of queries using host:

```
# host  
# host -a dns.example.com  
# host -a host01  
# host -a 192.0.2.101
```

- Examples of queries using dig:

```
# dig dns.example.com  
# dig -x 192.0.2.101  
# dig example.com NS  
# dig example.com A
```



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

The host and dig utilities are used for performing DNS lookups. The host utility returns the same information as dig and has more options. When no arguments are given, host displays a summary of its command-line arguments and options. Include the -a option to host for more verbose output.

To look up the IP address for host01:

```
# host host01  
# dig host01.example.com
```

To perform a reverse lookup, that is, to query DNS for the domain name that corresponds to an IP address:

```
# host 192.0.2.101  
# dig -x 192.0.2.101
```

To query DNS for the IP address that corresponds to a domain:

```
# host dns.example.com  
# dig dns.example.com
```



## Quiz

Which of the following is the main configuration file for BIND?

- a. /var/bind.conf
- b. /var/named.conf
- c. /etc/named.conf
- d. /etc/bind.conf



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe DHCP
- Configure a DHCP server
- Start and stop a DHCP server
- Configure a DHCP client and request a lease
- Describe DNS
- Describe nameserver types
- Describe BIND
- Configure a cache-only nameserver
- Describe and configure zone files
- Describe and configure reverse name resolution
- Use the `rndc` utility
- Use the `host` and `dig` utilities



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

ORACLE®

## Practice 2: Overview

The practices for this lesson cover the following:

- Configuring a DHCP server
- Configuring a DHCP client
- Viewing and Testing the DNS configuration
- Configuring a caching-only nameserver



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

# Authentication and Directory Services



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

# Objectives

After completing this lesson, you should be able to:

- Describe authentication options
- Describe the Authentication Configuration Tool
- Describe LDAP
- Describe OpenLDAP
- Describe OpenLDAP server and client utilities
- Configure LDAP authentication
- Configure Winbind authentication



ORACLE®

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

## Authentication Options

- Authentication is the verification of the identity of a user.
- Local account verification authenticates user information from local files:
  - /etc/passwd
  - /etc/shadow
- A local system can also access other directory services:
  - Network Information Service (NIS)
  - Lightweight Directory Access Protocol (LDAP)
  - Identity Policy Audit (IPA)
  - Winbind
- The Authentication Configuration GUI provides for the selection of user account databases and authentication configurations.



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

Authentication is the verification of the identity of a user. A user logs in by providing a username and a password and is authenticated by comparing this information to data stored on the system. If the login credentials match and the user account is active, the user is authenticated and can successfully access the system.

The information to verify the user can be located on the local system in the /etc/passwd and /etc/shadow files. A local system can also reference data stored on remote systems by using services such as Lightweight Directory Access Protocol (LDAP), Network Information Service (NIS), Identity Policy Audit (IPA), and Winbind. Additionally, LDAP, IPA, and NIS data files can use Kerberos authentication.

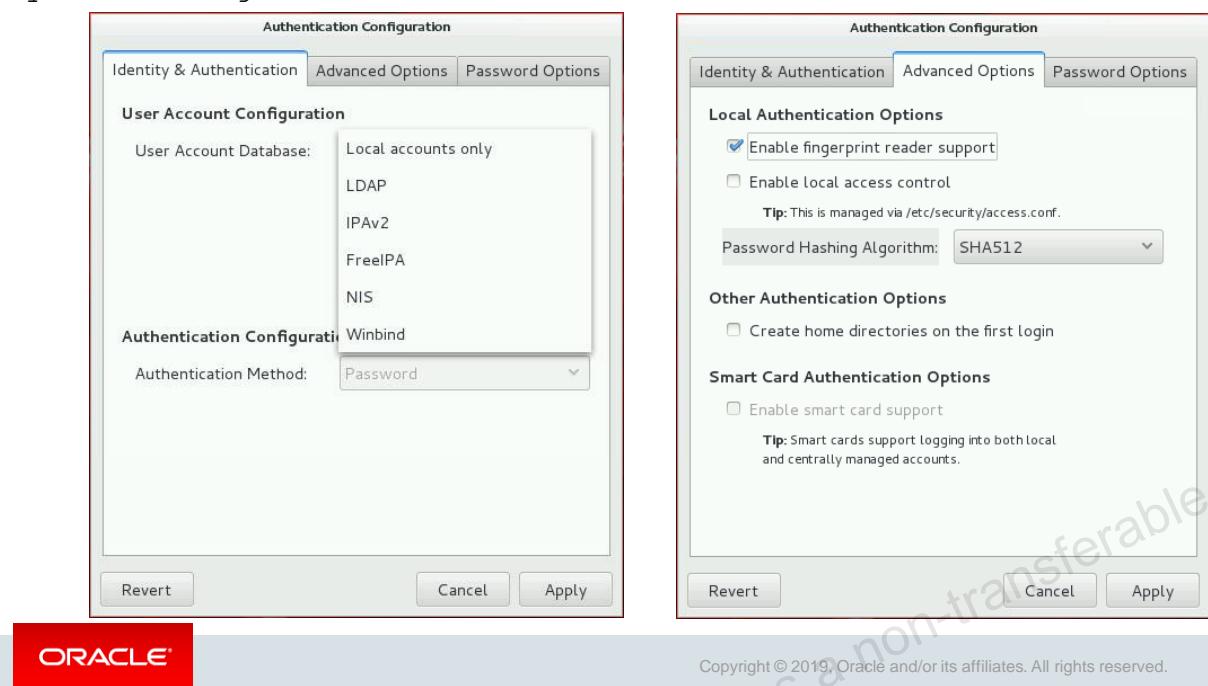
NIS simplifies the maintenance of common administration files such as /etc/passwd by keeping them in a central database and having clients retrieve information from this database server.

An LDAP directory can hold many types of information, including usernames, network services, and authentication data. Much like NIS, LDAP clients contact a centralized server to access this information.

Oracle Linux includes a GUI for selecting user databases and configuring associated authentication options—the Authentication Configuration GUI.

# Authentication Configuration GUI

system-config-authentication



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

ORACLE®

The software package `authconfig-gtk` provides the `system-config-authentication` utility.

To use the Authentication Configuration GUI, enter the command:

```
# system-config-authentication
```

There are three tabs on the Authentication Configuration GUI: Identify & Authentication, Advanced Options, and Password Options. The slide shows the first two tabs.

## Identity & Authentication

Click this tab to select how users should be authenticated. Under the User Account Configuration section of the page, select one of the six options for the User Account Database field:

- Local accounts only: Users and passwords are checked against local system accounts.
- LDAP
- IPA2
- FreeIPA
- NIS
- Winbind

Additional User Account Configuration fields appear, depending on which user account database is selected. The Authentication Configuration section of the page also changes, depending on which user account database is selected.

## **Advanced Options**

Click this tab to enable fingerprint reader support, enable local access control by using the `/etc/security/access.conf` file, configure smart card authentication options, change the password hashing algorithm, and configure other authentication options.

## **Password Options**

Click this tab to configure minimal password requirements, which include the length of the password and the number of character classes. You can also configure the required character classes and the maximal consecutive character repetition values.

# NIS Authentication

- NIS Domain:
  - A network of systems that share a common set of configuration files
- NIS Server:
  - A single system that stores the configuration files
- Authentication Method:
  - NIS password
  - Kerberos password



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

NIS was among the first directory services, but it has largely been replaced by other technologies, such as LDAP. NIS stores administrative information such as usernames, passwords, and host names on a centralized server. Client systems on the network can access this common data. This allows users the freedom to move from machine to machine without having to remember different passwords and copy data from one machine to another. Storing administrative information centrally, and providing a means of accessing it from networked systems, also ensures the consistency of that data.

An NIS network of systems is called an NIS domain. Each system within the domain has the same NIS domain name, which is different from a DNS domain name. The DNS domain is used throughout the Internet to refer to a group of systems. An NIS domain is used to identify systems that use files on an NIS server. An NIS domain must have exactly one master server but can have multiple slave servers. When using the Authentication Configuration Tool and selecting NIS as the user account database, you are prompted to enter the NIS Domain and the NIS Server.

For Authentication Method, NIS allows simple NIS password authentication or Kerberos authentication. Kerberos is an authentication protocol that allows nodes communicating over a nonsecure network to prove their identity to one another in a secure manner.

# Lightweight Directory Access Protocol (LDAP)

- LDAP is a protocol for accessing directory services.
- A directory is a hierarchical database.
- LDAP can also be used to authenticate users.
- An entry is the basic unit of information in a directory.
- Each entry has attributes.
- Required attributes are defined in a schema.
- Entries are uniquely identified and referenced by their distinguished name (DN).
- Example of a DN:
  - uid=oracle,ou=People,dc=example,dc=com
- LDIF is a plain-text representation of a DN.



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

Lightweight Directory Access Protocol (LDAP) is used to access centrally stored information over a network. LDAP servers store information in a database called a directory, which is optimized for searches. Directory entries are arranged in a hierarchical tree-like structure. This directory can store a variety of information such as names, addresses, phone numbers, network services, printers, and other types of data. LDAP can also be used to authenticate users, allowing users to access their accounts from any machine on the LDAP network.

An entry is the basic unit of information within an LDAP directory. Each entry has one or more attributes. Each attribute has a name, a type, or description and one or more values. An example of a type would be `cn` for a common name or `mail` for an email address. In addition, LDAP allows you to control which attributes are required and which are optional through the use of a special attribute called `objectClass`. The values of the `objectClass` attribute determine the schema rules that the entry must obey.

Each entry is uniquely identified and referenced by its distinguished name (DN). The DN is constructed by taking the name of the entry itself (called the “relative distinguished name” or RDN) and concatenating the names of its ancestor entries. For example, the DN for a user with an RDN of `uid=oracle` would be something like `uid=oracle,ou=People,dc=example,dc=com`. In this example, `ou` stands for “organizational unit” and `dc` stands for “domain component.”

The following is an example of the information needed for the `oracle` user:

```
dn: uid=oracle,ou=People,dc=example,dc=com
uid: oracle
cn: Oracle Student
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0...
shadowLastChange: 15880
shadowMin: 0
shadowMax: 9999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/oracle
gecos: Oracle Student
```

The following is an example group:

```
dn: cn=students,ou=Group,dc=example,dc=com
objectClass: posixGroup
objectClass: top
cn: students
userPassword:: e2NyeXB0...
gidNumber: 1008
memberUid: oracle
memberUid: student1
memberUid: student2
```

LDAP Data Interchange Format (LDIF) is a plain-text representation of an LDAP entry. It takes the following form:

```
[id] dn: distinguished_name
attribute_type: attribute_value...
attribute_type: attribute_value...
```

The optional `id` number is determined by the application that is used to edit the entry. Each entry can contain as many `attribute_type` and `attribute_value` pairs as needed, as long as they are all defined in a corresponding schema file.

# OpenLDAP

- OpenLDAP is an open-source implementation of LDAP.
- Packages include:
  - **openldap**: OpenLDAP libraries
  - **openldap-clients**: Client command-line utilities
  - **openldap-servers**: Server package, includes `slapd`
  - **nss-pam-ldapd**: Required for LDAP authentication
- OpenLDAP service is the stand-alone LDAP daemon, `slapd`.
- Use the `systemctl` utility to enable and start the service:

```
# systemctl enable slapd
# systemctl start slapd
```



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

Oracle Linux includes OpenLDAP, which is an open source implementation of the LDAP protocols. To begin configuring a system as an OpenLDAP server, install the following OpenLDAP packages:

- **openldap**: Contains the libraries necessary to run the OpenLDAP server and client applications
- **openldap-clients**: Contains the command-line utilities for viewing and modifying directories on an LDAP server
- **openldap-servers**: Contains the services and utilities to configure and run an LDAP server. This includes the stand-alone LDAP daemon, `slapd`.
- **nss-pam-ldapd**: Contains `nsLCD`, a local LDAP name service that allows a user to perform local LDAP queries. This package is only required to authenticate by using OpenLDAP.

Additional OpenLDAP packages, not required for a standard configuration, are:

- **compat-openldap**: Includes older versions of the OpenLDAP-shared libraries that might be required by some applications
- **bind-dyndb-ldap**: A new LDAP driver for BIND9. It allows you to read data and also write data back (DNS Updates) to an LDAP backend.

# OpenLDAP Server Directories

- Previous versions of OpenLDAP used a configuration file:
  - `/etc/openldap/slapd.conf`
- The current version of OpenLDAP uses a configuration database located in:
  - `/etc/openldap/slapd.d`
- The directory containing additional configuration files:
  - `/etc/openldap/slapd.d/cn=config`
- The directory containing the schema files:
  - `/etc/openldap/schema`
- The directory containing the database:
  - `/var/lib/ldap`



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

Previous versions of OpenLDAP used a configuration file:

`/etc/openldap/slapd.conf`

OpenLDAP now uses a configuration database located in the following directory:

`/etc/openldap/slapd.d`

The following list summarizes the OpenLDAP configuration that is stored in the `/etc/openldap` directory:

- `/etc/openldap/ldap.conf`: The configuration file for client applications
- `/etc/openldap/slapd.d`: The directory containing the `slapd` configuration
- `/etc/openldap/schema`: The directory containing the schema files

The schema used by OpenLDAP can be extended to support additional attribute types and object classes.

This is described at:

<http://www.openldap.org/doc/admin24/schema.html>

OpenLDAP uses one of two varieties of the Berkeley DB storage format:

- **bdb**: The standard Berkeley DB format
- **hdb**: A newer version for hierarchical databases like LDAP

The database is stored in the `/var/lib/ldap` directory.

## OpenLDAP Server Utilities

- **slapac1:** Checks the access to a list of attributes
- **slapadd:** Adds entries from an LDIF file
- **slapauth:** Checks permissions
- **slapcat:** Generates LDIF output from an LDAP directory
- **slapdn:** Checks a list of DNs based on schema syntax
- **slapindex:** Re-indexes the directory
- **slappasswd:** Is a password utility
- **slapschema:** Checks compliance of a directory
- **slaptest:** Checks the LDAP server configuration



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

The `openldap-servers` package also includes the following utilities:

- **slapac1:** Checks the access to a list of attributes
- **slapadd:** Adds entries from an LDIF file to an LDAP directory
- **slapauth:** Checks a list of IDs for authentication and authorization permissions
- **slapcat:** Generates LDIF output from an LDAP directory
- **slapdn:** Checks a list of distinguished names (DNs) based on schema syntax
- **slapindex:** Re-indexes the directory. Run `slapindex` whenever indexing options are changed in the configuration file
- **slappasswd:** Is a password utility for creating an encrypted user password
- **slapschema:** Checks compliance of a database with the corresponding schema
- **slaptest:** Checks the LDAP server configuration

## OpenLDAP Client Utilities

- **ldapadd:** Adds entries to an LDAP directory
- **ldapmodify:** Modifies entries in an LDAP directory
- **ldapcompare:** Compares a given attribute with an entry
- **ldapdelete:** Deletes entries from an LDAP directory
- **ldapexop:** Performs extended LDAP operations
- **ldapmodrdn:** Modifies the RDN value of an entry
- **ldappasswd:** Is a password utility for an LDAP user
- **ldapsearch:** Is an LDAP directory search tool
- **ldapurl:** Is an LDAP URL formatting tool
- **ldapwhoami:** Performs a `whoami` operation



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

The `openldap-clients` package installs the following utilities:

- **ldapadd:** Adds entries to an LDAP directory either from a file or from standard input. `ldapadd` is a symbolic link to `ldapmodify -a`.
- **ldapmodify:** Modifies entries in an LDAP directory
- **ldapcompare:** Compares a given attribute with an LDAP directory entry
- **ldapdelete:** Deletes entries from an LDAP directory
- **ldapexop:** Performs extended LDAP operations
- **ldapmodrdn:** Modifies the RDN value of an LDAP directory entry
- **ldappasswd:** Is a password utility for an LDAP user
- **ldapsearch:** Is an LDAP directory search tool
- **ldapurl:** Is an LDAP URL formatting tool
- **ldapwhoami:** Performs a `whoami` operation on an LDAP server

There are several LDAP client software applications that provide a graphical user interface (GUI) for maintaining LDAP directories, but none of them are included in Oracle Linux.

# OpenLDAP Server Configuration

- Install the packages:

```
# yum install openldap-servers openldap-clients migrationtools
```

- The `openldap-clients` package provides the `ldap` command-line utilities.
- The `migrationtools` package is optional but allows you to migrate information from existing name services.

- Enable and start the `slapd` service.
  - This creates the database in the `/var/lib/ldap` directory.
- Update the files in the configuration directory:
  - `/etc/openldap/slapd.d/cn=config`
- Use the `ldapmodify` and `ldapadd` commands to update these files.



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

To configure an OpenLDAP server, install the following packages:

```
# yum install openldap-servers openldap-clients migrationtools
```

The `openldap-clients` package provides the LDAP command-line utilities used to update the configuration database. The `migrationtools` package is optional but it provides a set of Perl scripts, which allows you to migrate users, groups, and other information from existing name services.

Use the `systemctl` command to enable and start the `slapd` service.

```
# systemctl enable slapd
# systemctl start slapd
```

Use `ldapmodify` and `ldapadd` commands to update the files in the configuration directory:

```
# ls /etc/openldap/slapd.d/cn=config
-rw-----. 1 ldap ldap ... olcDatabase={0}config.ldif
-rw-----. 1 ldap ldap ... olcDatabase={-1}frontend.ldif
-rw-----. 1 ldap ldap ... olcDatabase={1}monitor.ldif
-rw-----. 1 ldap ldap ... olcDatabase={2}hdb.ldif
```

## The ldapmodify Utility

- The following example uses the `ldapmodify` utility to set the `olcSuffix` directive in the `olcDatabase={2}hdb.ldif` file to “dc=example,dc=com”:

```
# ldapmodify -Q -Y EXTERNAL -H ldapi:/// <<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> replace: olcSuffix
> olcSuffix: dc=example,dc=com
>
> EOF
```

- After issuing the `ldapmodify` command, the prompt changes to `>`.
- The command terminates after entering the final “EOF”.



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

The example in the slide uses the `ldapmodify` command to update the `olcSuffix` parameter in the `olcDatabase={2}hdb.ldif` file. The example sets `olcSuffix` to “dc=example,dc=com”.

Use the `ldapmodify` utility to update the OpenLDAP domain component in the configuration database. The default setting after an initial installation is:

`dc=my-domain,dc=com`

You can use the `grep` command and search for occurrences of “my-domain” in the configuration database directory:

```
# grep my-domain /etc/openldap/slapd.d/cn=config/*
olcDatabase={1}monitor.ldif: ,cn=auth" read by dn.base="cn=Manager,dc=my-
domain,dc=com" read by * none
olcDatabase={2}hdb.ldif:olcSuffix: dc=my-domain,dc=com
olcDatabase={2}hdb.ldif:olcRootDN: cn=Manager,dc=my-domain,dc=com
```

In this example, “my-domain” needs to be changed to your company’s DN in two files:

- `olcDatabase={1}monitor.ldif`
- `olcDatabase={2}hdb.ldif`

## The slappasswd Utility

- Use the `slappasswd` command to create an encrypted user password. Example:

```
# slappasswd
New password: <password>
Re-enter new password: <password>
{SSHA}4SOiIaqwQYftwkdr1FbqVNEmI3Am0wJT
```

- Use the `ldapmodify` command to add the `olcRootPW` directive to the `olcDatabase={2}hdb.ldif` file.
  - Set the value of `olcRootPW` to the encrypted result of the `slappasswd` command.
- Subsequent modifications to the OpenLDAP directory require the password. Example:

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f base.ldif
Enter LDAP Password:
```



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

Use the `slappasswd` command to create an encrypted user password. The encrypted password shown is a sample only.

```
# slappasswd
New password: <password>
Re-enter new password: <password>
{SSHA}4SOiIaqwQYftwkdr1FbqVNEmI3Am0wJT
```

Use the `ldapmodify` command to add the `olcRootPW` directive to the `olcDatabase={2}hdb.ldif` file. Set the value of `olcRootPW` to the encrypted result of the `slappasswd` command:

```
# ldapmodify -Q -Y EXTERNAL -H ldap:// //<<EOF
> dn: olcDatabase={2}hdb,cn=config
> changetype: modify
> add: olcRootPW
> olcRootPW: {SSHA}4SOiIaqwQYftwkdr1FbqVNEmI3Am0wJT
>
> EOF
```

## Loading the Standard Schemas

- The attributes of each data object in an OpenLDAP directory are defined in a schema.
  - The schema must be loaded into the configuration database before the object they define can be used.
  - The standard schemas are provided as LDIF files in the /etc/openldap/schema directory.
- The following four schemas provide the objects and attributes of a typical organization:
  - core, cosine, inetorgperson, nis
- Use the `ldapadd` command to load the schema. Example:

```
# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/core.ldif
```



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

The attributes of each data object in an OpenLDAP directory are defined in a schema. The schema must be loaded into the configuration database before the object they define can be used. The standard schemas are provided as LDIF files in the /etc/openldap/schema directory.

```
# ls /etc/openldap/schema
collective.ldif    cosine.schema    java.ldif      openldap.schema
collective.schema  duacnf.ldif     java.schema    pmi.ldif
...
```

The following four schemas define the basic objects and attributes needed to describe a typical organization: core, cosine, inetorgperson, nis.

Use the `ldapadd` utility to load the schemas.

```
# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/core.ldif
adding new entry "cn=core,cn=schema,cn=config"
# ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
adding new entry "cn=cosine,cn=schema,cn=config"
```

## Populating an OpenLDAP Directory

- Create a base domain text file in the LDIF format.
- The example in the notes defines the base domain and the top-level DN for users and groups:
  - dn: dc=example,dc=com
  - dn: ou=People,dc=example,dc=com
  - dn: ou=Group,dc=example,dc=com
- Use the `ldapadd` command to import the base information to the LDAP directory.

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f base.ldif
Enter LDAP Password: <password>
adding new entry "dc=example,dc=com"
adding new entry "ou=People,dc=example,dc=com"
adding new entry "ou=Group,dc=example,dc=com"
```



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

You can create a text file containing base information for the LDAP directory and then use the `ldapadd` command to import the information into the directory. When executing the above command, the `-x` option uses simple authentication, the `-W` option prompts for simple authentication, and the `-D` option binds the distinguished name to the LDAP directory. The `-f` option reads the `base.ldif` file instead of standard input.

Create the base information in the LDIF format. The following example contains top-level information for users and groups:

```
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
dn: ou=People,dc=example,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit
dn: ou=Group,dc=example,dc=com
ou: Group
objectClass: top
objectClass: organizationalUnit
```

## Using the migrationtools Utilities

- The `migrationtools` Perl scripts allow you to migrate information from existing name services.
- The `migrationtools` files are installed in the `/usr/share/migrationtools` directory.
- Update the `migrate_common.ph` file for the correct domain.

```
# vi /usr/share/migrationtools/migrate_common.ph
$DEFAULT_MAIL_DOMAIN = "example.com";
$DEFAULT_BASE = "dc=example,dc=com";
```

- Extract existing information into text files.

```
# grep ":100[0-9]" /etc/passwd > passwd
# grep ":100[0-9]" /etc/group > group
```



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

The `migrationtools` Perl scripts allow you to migrate users, groups, and other information from existing name services. The `migrationtools` files are installed in the `/usr/share/migrationtools` directory.

You must first update the `migrate_common.ph` file for the correct domain. The following example sets two directives to the “example.com” domain:

```
# vi /usr/share/migrationtools/migrate_common.ph
$DEFAULT_MAIL_DOMAIN = "example.com";
$DEFAULT_BASE = "dc=example,dc=com";
```

You can use the `grep` command to extract existing users and groups into text files. The following example extracts users from the `/etc/passwd` file with UID in the 1000–1009 range and writes the output to the `passwd` file:

```
# grep ":100[0-9]" /etc/passwd > passwd
```

The following example extracts groups with GID in the 1000–1009 range and writes the output to the `group` file:

```
# grep ":100[0-9]" /etc/group > group
```

## Using the migrationtools Utilities

- Use the `migrate_passwd.pl` command to migrate user information into an LDAP format.

— The following example converts the `passwd` file to LDIF format:

```
# /usr/share/migrationtools/migrate_passwd.pl passwd > users.ldif
```

- Use the `migrate_group.pl` command to migrate group information into an LDAP format.

— The following example converts the `group` file to LDIF format:

```
# /usr/share/migrationtools/migrate_group.pl group > group.ldif
```

- Use the `ldapadd` command to import migrated information into the LDAP directory.



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

Run the `migrate_passwd.pl` command to migrate user information in the `passwd` file into an LDAP format. This example redirects the output to `users.ldif`:

```
# /usr/share/migrationtools/migrate_passwd.pl passwd > users.ldif
```

Run the `migrate_group.pl` command to migrate group information in the `group` file into an LDAP format. This example redirects the output to `group.ldif`:

```
# /usr/share/migrationtools/migrate_group.pl group > group.ldif
```

You can then use the `ldapadd` command to import the user and group information to the LDAP directory:

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f users.ldif
```

Enter LDAP Password: <*password*>

adding new entry "uid=oracle,ou=People,dc=example,dc=com"

...

```
# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f group.ldif
```

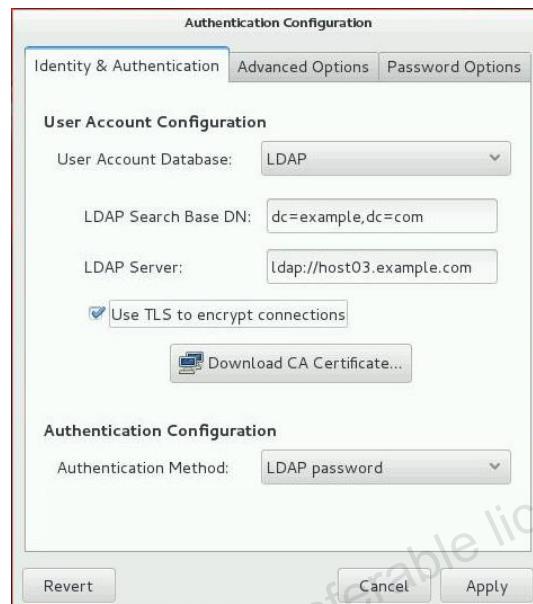
Enter LDAP Password: <*password*>

adding new entry "cn=oracle,ou=Group,dc=example,dc=com"

...

# Configuring LDAP Authentication

- LDAP Search Base DN
  - The distinguished name (DN), or root suffix, for the user directory
- LDAP Server
  - The URL of the LDAP server
- Authentication Method
  - LDAP password
  - Kerberos password



ORACLE®

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

To configure LDAP authentication from the Authentication Configuration Tool on the OpenLDAP server, select LDAP as the user account database. You are then prompted to enter:

- LDAP Search Base DN
- LDAP Server

For LDAP Search Base DN, enter the DN, or the root suffix, for the user directory. The LDAP directory is hierarchical, so all the user entries used for identity and authentication exist below this parent entry. An example of a base DN would be `dc=example,dc=com`.

For LDAP Server, enter the URL of the LDAP server and optionally include the port number. Examples of this would be:

- `ldap://host03.example.com:389`
- `ldaps://host03.example.com`

You also have the option to “Use TLS to encrypt connections.” TLS stands for Transport Layer Security. It provides a secure connection by encrypting the connections to the LDAP server. Selecting TLS enables the “Download CA Certificate” button. CA stands for certificate authority or certification authority and is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key. Clicking the “Download CA Certificate” button prompts you to enter the URL from which to download the CA certificate.

Do not select “Use TLS to encrypt connections” if the server URL uses a secure protocol (`ldaps`).

For Authentication Method, select either one of the following:

- LDAP password
- Kerberos password

The LDAP password option uses Pluggable Authentication Modules (PAM) applications for LDAP authentication. This option requires either a secure URL or the use of TLS to connect to the LDAP server.

You can also enable and configure LDAP from the command line by using the `authconfig` command. To use an LDAP identity data store, use the `--enableldap` flag. To use LDAP as the authentication source, use the `--enableldapauth` flag. Include the LDAP server name, the base DN for the user suffix, and TLS information if used. Use the full LDAP URL, including the protocol (`ldap` or `ldaps`) and the port number.

The following is an example:

```
# authconfig --enableldap --enableldapauth  
--ldapserver=ldap://host03.example.com:389  
--ldapbasedn="dc=example,dc=com" --enableldaptls  
--ldaploadcacert=https://ca.server.example.com/caCert.crt --update
```

Basic configuration of the OpenLDAP server is now complete.

OpenLDAP uses port 389 to communicate over the network. Ensure that the firewall is configured (or disabled) to allow access to this port. When using `firewalld`, add the `ldap` service. Example:

```
# firewall-cmd --permanent --zone=public --add-service=ldap
```

If you configure LDAP over Secure Sockets Layer (SSL), the port number is 636. When using `firewalld`, add the `ldaps` service.

```
# firewall-cmd --permanent --zone=public --add-service=ldaps
```

# Configuring User Authentication from an OpenLDAP Client

- Install the following packages on the client:  

```
# yum install openldap-clients nss-pam-ldapd
```
- Edit the /etc/openldap/ldap.conf file on the client to point to the LDAP server:
  - BASE dc=example,dc=com
  - URI ldap://192.0.2.103/
  - Also update /etc/nslcd.conf
- Set USELDAP=yes in /etc/sysconfig/authconfig.
- Enable the pam\_ldap module in the /etc/pam.d/system-auth file.
- Add ldap to the passwd, shadow, and group directives in the /etc/nsswitch.conf file.
- Start the nslcd service.



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

OpenLDAP clients can search a remote server for information. Users on an OpenLDAP client can also be authenticated over the network by the server. This allows all user accounts to exist only in the LDAP directory. Users can then log in from clients and be authenticated by the server. Install the following packages to configure an OpenLDAP client:

```
# yum install openldap-clients nss-pam-ldapd
```

Edit the /etc/openldap/ldap.conf file on the client to point to the LDAP server. Example:

```
BASE      dc=example,dc=com
URI       ldap://192.0.2.103/
```

You also need to configure the /etc/nslcd.conf file to point to the LDAP server. This is the configuration file for the Naming Services LDAP Client Daemon. Example (both directives are in lowercase in these files):

```
uri       ldap://192.0.2.103/
base     dc=example,dc=com
```

Edit the /etc/sysconfig/authconfig file and set USELDAP=yes:

```
USELDAP=yes
```

The `pam_ldap` module allows OpenLDAP clients to authenticate against LDAP directories, and to change their passwords in the directory. Pluggable Authentication Modules (PAM) is covered in a subsequent lesson in this course.

Changes are needed to the `/etc/pam.d/system-auth` file to enable this PAM module. Items in bold are added to this file.

```
# vi /etc/pam.d/system-auth
...
auth      requisite      pam_succeed_if.so uid >= 1000 quiet
auth      sufficient    pam_ldap.so use_first_pass
...
account   sufficient    pam_succeed_if.so uid < 1000 quiet
account  [default=bad success=ok user_unknown=ignore] pam_ldap.so
...
password  sufficient    pam_unix.so sha512 shadow nullok try_first_pass
use_authtok
password  sufficient    pam_ldap.so use_authtok
...
session   required      pam_unix.so
session   optional     pam_ldap.so
session   optional     pam_mkhomedir.so skel=/etc/skel umask=077
```

Add `ldap` to the `passwd`, `shadow`, and `group` directives in the `/etc/nsswitch.conf` file:

```
# vi /etc/nsswitch.conf
passwd:  files ldap
shadow:  files ldap
group:   files ldap
```

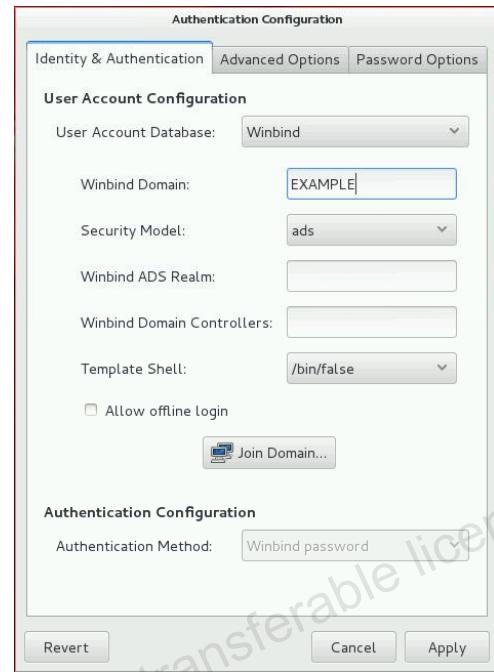
Use the `systemctl` command to start the `nslcd` service:

```
# systemctl start nslcd
```

You can now log in to the OpenLDAP client as a user, which has an account only on the LDAP server. The new “`session`” entry in the `/etc/pam.d/system-auth` file, which loads the `pam_mkhomedir` module, creates the user’s home directory on the client if needed.

# Configuring Winbind Authentication

- Winbind Domain
- Security Model
  - ads, domain, server, user
- Winbind ADS Realm
  - ads only
- Winbind Domain Controllers
  - ads, domain only
- Template Shell
  - ads, domain only



**ORACLE®**

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

To configure Winbind authentication, install the `samba-winbind` package:

```
# yum install samba-winbind
```

This package includes the `winbindd` daemon, controlled by the `winbind` service. `winbind` is a client-side service used to connect to Windows servers. It resolves user and group information on a Windows server, allowing Linux to understand Windows users and groups.

Select Winbind as the user account database in the Authentication Configuration Tool. You are then prompted for information required to connect to a Microsoft workgroup, Active Directory, or Windows NT domain controller.

Select the security model to use for Samba clients. The security model options are:

- **ads**: This configures Samba to act as a domain member in an Active Directory Server realm. To use ads, you need to install the `krb5-server` package and configure Kerberos.
- **domain**: Samba validates the user by authenticating through a Windows domain controller.
- **server**: A local Samba server validates the user by authenticating it through another server.
- **user**: This requires a client to log in with a valid username and password.

To complete Winbind authentication configuration, provide the following information:

- **Winbind ADS Realm:** The Active Directory realm that the Samba server joins. This is required only when using the `ads` security model.
- **Winbind Domain Controllers:** The domain controller to use
- **Template Shell:** The login shell to use for Windows NT user account settings
- **Allow offline login:** Allows user authentication while the system is offline. Authentication information is stored in a local cache provided by System Security Services Daemon (SSSD).

Winbind authentication can also be configured from the command line by using the `authconfig` command. For `user` and `server` security models, only the domain (or workgroup) name and the domain controller host names are required:

```
# authconfig --enablewinbind --enablewinbindauth --smbsecurity  
user|server --enablewinbindoffline --smbservers=ad.example.com  
--smbworkgroup=EXAMPLE --update
```

For `ads` and `domain` security models, specify using the `--smbsecurity` flag and append the template shell and realm (`ads` only) flags to the previous example:

```
--smbrealm EXAMPLE.COM --winbindtemplateshell=/bin/sh -update
```

# Winbind Security Model Options

- User Security Model
  - This model requires a client to log in with a valid username and password.
  - The client can mount multiple shares without specifying a separate username and password for each instance.
- Server Security Model
  - This model presents numerous security issues.
- Domain Security Model
  - The Samba server has a domain security trust account.
  - Samba authenticates username and password through a domain controller.
- Activity Directory Server (ADS) Security Model
  - Samba acts as a domain member in an ADS realm.



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

## User Security Model

User-level security is the default setting for Samba. It requires a client to log in with a valid username and password. This mode does support encrypted passwords. If the server accepts the client's username and password, the client can then mount multiple shares without specifying a password for each instance.

## Server Security Model

In this model, a local Samba server validates the username and password by authenticating it through another server, such as a Windows NT server. Do not use this model, because it presents numerous security issues. It was designed before Samba could act as a domain member server.

## Domain Security Model

In this model, the Samba server has a machine account (domain security trust account), and Samba validates the username and password by authenticating it through a domain controller.

## Activity Directory Server (ADS) Security Model

In this model, Samba is configured to act as a domain member in an ADS realm. Windows requires Kerberos tickets for Active Directory authentication. In addition, a machine account must be created on the Active Directory domain server.

Install the `krb5-server` package and configure Kerberos before attempting to join a domain.

```
# yum install krb5-server
```

Use the `kinit` command to create an administrative Kerberos ticket as follows:

```
# kinit administrator@EXAMPLE.COM
```

The `kinit` command references the Active Directory administrator account and Kerberos realm. **A Kerberos realm contains key distribution center information.** It then obtains and caches a Kerberos ticket, which is required for authentication.

Assuming that Kerberos has been initialized, click the Join Domain button to attempt to join the domain immediately. Alternatively, use the following command to join:

```
# net ads join -S <active_directory_server> -U administrator%password
```

This command creates the appropriate machine account on the Active Directory server and grants permissions to the Samba domain member server to join the domain.



## Quiz

Which of the following stores information in a structure, called a directory, that is optimized for searches?

- a. NIS
- b. OpenLDAP
- c. Winbind
- d. Kerberos
- e. SSSD



ORACLE®

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

## Summary

In this lesson, you should have learned how to:

- Describe authentication options
- Describe the Authentication Configuration Tool
- Describe LDAP
- Describe OpenLDAP
- Describe OpenLDAP server and client utilities
- Configure LDAP authentication
- Configure Winbind authentication



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

## Practice 3: Overview

The practices for this lesson cover the following topics:

- Configuring an OpenLDAP server
- Implementing OpenLDAP authentication
- Authenticating from an OpenLDAP client



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

# Pluggable Authentication Modules (PAM)

ORACLE®

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

## Objectives

After completing this lesson, you should be able to:

- Describe the purpose of PAM
- Describe PAM configuration files
- Describe PAM authentication modules
- Describe PAM module types
- Describe PAM control flags
- Walk through PAM authentication examples



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

## Introduction to PAM

- PAM allows you to configure how applications use authentication to verify the identity of a user.
- Configuration files are located in the `/etc/pam.d` directory.
- Each configuration file has the same, or a similar, name as the application it authenticates, for example:
  - `login`, `sudo`, `sshd`, `su`, `passwd`
- Each configuration file lists authentication modules that contain the authentication code.
- Authentication modules are shared libraries located in `/lib/security` (and `/lib64/security`).
- PAM documentation includes man pages for most modules and SAG in `/usr/share/doc/pam-<version>`.



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

Pluggable Authentication Modules (PAM) is an authentication mechanism that allows you to configure how applications use authentication to verify the identity of a user. Many system applications in Oracle Linux use PAM for authentication and authorization.

The PAM configuration files are located in the `/etc/pam.d` directory. Each of these files have names that are the same as, or similar to, the name of the application that they authenticate. For example, there is a configuration file for the `su` process named `su`.

Each PAM configuration file contains a group of directives that define the authentication module as well as any controls or arguments. The following example is from the `su` file:

```
account sufficient pam_succeed_if.so uid = 0 use_uid quiet
```

The four directives in this example are known by the following names:

- **module\_interface**: account
- **control\_flag**: sufficient
- **module\_name**: pam\_succeed\_if.so
- **module\_arguments**: uid = 0 use\_uid quiet

The PAM authentication module name in this example is `pam_succeed_if.so`. The authentication modules are shared libraries and are located in the `/lib/security` and `/lib64/security` directories.

## PAM Configuration Files

- Configuration files are located in the `/etc/pam.d` directory.
- Each PAM configuration file contains a list, or stack of calls to authentication modules.  
Example:

```
# cat /etc/pam.d/su
auth sufficient pam_rootok.so
auth required pam_wheel.so use_uid
auth include system-auth
account sufficient pam_succeed_if.so uid = 0 use_uid
account include system-auth
password include system-auth
session include system-auth
session optional pam_xauth.so
```

- The authentication modules have a `.so` extension.
- The `system-auth` is a configuration file that is “included.”



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

Each PAM configuration file contains a list, or stack, of calls to authentication modules. The example shows the contents of the `su` configuration file.

```
# cat /etc/pam.d/su
auth sufficient pam_rootok.so
auth required pam_wheel.so use_uid
auth include system-auth
account sufficient pam_succeed_if.so uid = 0 use_uid quiet
account include system-auth
password include system-auth
session include system-auth
session optional pam_xauth.so
```

This file references four authentication modules: `pam_rootok.so`, `pam_wheel.so`, `pam_succeed_if.so`, and `pam_xauth.so`. The `pam_wheel.so` and `pam_succeed_if.so` modules include module arguments.

The `system-auth` file is not an authentication module but is a common configuration file for PAM-ified services. Contents of the `system-auth` file are appended to the `su` configuration file and processed as if they were part of the file.

## PAM Authentication Modules

- PAM provides several authentication modules in shared libraries.
  - These are located in /lib/security (or /lib64/security for 64-bit Linux).
- Example:

```
# ls /lib64/security
pam_access.so      pam_limits.so      pam_chroot.so
pam_cap.so        pam_listfile.so    pam_sss.so
pam_chroot.so     pam_localuser.so   pam_stress.so
pam_console.so    pam_loginuid.so   pam_succeed_if.so
pam_cracklib.so   pam_mail.so       pam_systemd.so
pam_debug.so       pam_mkhomedir.so  pam_tally2.so
pam_deny.so        pam_motd.so      pam_time.so
...
...
```

- This pluggable, modular architecture provides flexibility in setting authentication policies.



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

PAM provides several authentication modules in shared libraries, which are located in /lib/security (or /lib64/security for 64-bit Linux).

This pluggable, modular architecture provides flexibility in setting authentication policies for a system. With the authentication code separate from the application code, you can configure the authentication mechanism for a given application without ever touching the application.

There are manual pages for most of the PAM modules. The following example lists a partial man page for the **pam\_access** module.

```
# man pam_access
...
NAME
  pam_access - PAM module for logdaemon style login access...
SYNOPSIS
  pam_access.so [debug] [nodefgroup] [noaudit] [accessfile...]
DESCRIPTION
  The pam_access PAM module is mainly for access manage...
...
```

## PAM Module Types

- The first column in the /etc/pam.d configuration file (`auth` in this example) is the module interface:

```
auth sufficient pam_rootok.so
```

- Module types represent a different aspect of the authorization process.

- Four types are available:

- `auth`: Proves the user is authorized to use the service
- `account`: Determines whether an already authenticated user is allowed to use the service
- `password`: Updates user authentication credentials
- `session`: Configures and manages user sessions



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

The following lists partial contents of the /etc/pam.d/su configuration file.

```
auth sufficient pam_rootok.so
auth required pam_wheel.so use_uid
session optional pam_xauth.so
```

The first column is the PAM module interface or module type indicator. Four module types are available. Only the `auth` and `account` types determine authorization to run a command.

- auth**: Proves that the user is authenticated or authorized to use the service. This can be done by requesting and verifying the validity of a password. Modules with this interface can also set credentials, such as group memberships or Kerberos tickets.
- account**: Verifies whether an already authenticated user is allowed access to the application. For example, it could check whether a user account has expired or whether a user is allowed to use this service at a particular time of day.
- password**: Is used when a user tries to update his or her authentication token
- session**: Configures and manages user sessions. Performs tasks that are needed to allow access, such as mounting a user's home directory and unmounting when the service is terminated.

## PAM Control Flags

- The second column in the `/etc/pam.d` configuration file (sufficient in this example) is the control flag:
  - `auth sufficient pam_rootok.so`
- Each PAM module generates a success or failure result.
- Control flags tell PAM what to do with the result:
  - `required`: The module must pass before access is granted. The user is not notified immediately if the module fails.
  - `requisite`: This is similar to `required` control flag except that the user is notified immediately if the module fails.
  - `sufficient`: Failure is not necessarily fatal, depending on other module test results.
  - `optional`: The module result is ignored unless this is the only module.
  - `include|substack`: This includes lines from another file.



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

The following lists partial contents of the `/etc/pam.d/su` configuration file.

```
auth      sufficient  pam_rootok.so
auth      required    pam_wheel.so use_uid
session   optional   pam_xauth.so
```

The second column is the PAM control flag. PAM reads the stack from top to bottom and calls the modules listed in the configuration file. Each PAM module generates a success or failure result when called. Control flags tell PAM what to do with the result.

Modules can be stacked in a particular order, and the control flags determine how important the success or failure of a particular module is to the overall goal of authenticating the user to the service. The available control flags are listed:

- **required**: All required modules are tried, and all must “pass” before access is granted by PAM. If the module fails at this point, the user is not notified until all modules in the stack have been executed.
- **requisite**: Similar to `required`, in which success of the module is required for authentication to continue. However, if the module fails, no further modules are executed. The user is notified immediately of the first failed `required` or `requisite` module test.

- **sufficient**: Success indicates that this module type has succeeded and no subsequent required modules of this type are executed. Failure is not fatal to the stack of this module type, however. PAM processes the remaining modules listed to decide whether access is allowed.
- **optional**: The module result is generally ignored. A module flagged as optional becomes necessary for successful authentication when it is the only module in the stack for a particular service.
- **include**: Unlike the other controls, this does not relate to how the module result is handled. This control flag pulls in all lines of the given type from the configuration file specified as an argument to this control.
- **substack**: This control is similar to include in that it includes all lines of the given type from the configuration file specified as an argument. The difference from include is that evaluation of the done and die actions in a substack does not cause skipping the rest of the complete module stack, but only of the substack.

PAM also includes some predefined actions in the control flag field, which are included within brackets as follows:

```
[value1=action1 value2=action2 ...]
```

The use of brackets in the control flag field gives you full control of PAM's actions. When the result returned by a function matches value, action is evaluated.

## PAM: Example #1

- The contents of the `/etc/pam.d/sshd` file are listed in the notes.
- The `pam_sepermit` PAM module allows or denies login depending on the SELinux enforcement state.
  - SELinux is covered in a subsequent lesson in this course.
- The `pam_nologin` module prevents users from logging into the system when either the `/var/run/nologin` file or the `/etc/nologin` file exists.
- The `pam_selinux` module sets up the default SELinux security context for the next executed process.
- The `pam_loginuid` module records the user's login UID.
- The `pam_keyinit` module is the kernel session keyring initializer.



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

To see PAM in action, the following example steps through the processing of the `sshd` application's authentication stack. The contents of the `sshd` configuration file are listed.

```
# cat /etc/pam.d/sshd
#%PAM-1.0
auth    required    pam_sepermit.so
auth    substack    password-auth
auth    include     postlogin
account required   pam_nologin.so
account include   password-auth
password include  password-auth
session required   pam_selinux.so close
session required   pam_loginuid.so
session required   pam_selinux.so open env_params
session optional   pam_keyinit.so force revoke
session include   password-auth
session include   postlogin
```

Lines that begin with # are comments and are not processed. There are three comment lines in this file. The remaining lines tell PAM to do something as part of the authentication process.

The first line is:

```
auth      required      pam_sepermit.so
```

This line uses the `pam_sepermit.so` module to attempt to authenticate the user. Because the control flag is required, this module is required to succeed if the authentication stack is to succeed.

There are manual pages for most of the PAM modules. The man page for `pam_serpermit` states that the PAM module allows or denies login depending on the SELinux enforcement state. SELinux is covered in a subsequent lesson in this course. The man page goes on to say that “when the user which is logging in matches an entry in the configuration file, `sepermit.conf`, the user is allowed access only when the SELinux is in enforcing mode.” Otherwise the user is denied access. The configuration file, `sepermit.conf`, also has a man page that describes the configuration options.

The next two lines include a couple of configuration files:

```
auth      substack      password-auth  
auth      include       postlogin
```

These files are located in the `/etc/pam.d` directory. These files also have man pages. The `password-auth` configuration file is for applications that handle authentication from different types of devices via simultaneously running individual conversations instead of one aggregate conversation. The `postlogin` configuration file is included from all individual service configuration files that provide login service with shell or file access.

The fourth line is:

```
account    required      pam_nologin.so
```

This line uses the `pam_nologin.so` module to verify whether an already authenticated user is allowed access to the application. The man page for `pam_nologin` states that the PAM module prevents users from logging into the system when `/var/run/nologin` or `/etc/nologin` exists. Because the control flag is required, this module is required to succeed if the authentication stack is to succeed.

The fifth and sixth lines are `account` and `password` type entries, both of which include the `password-auth` configuration file. The remaining lines are `session` type entries that use the following PAM modules:

```
session    required      pam_selinux.so close  
session    required      pam_loginuid.so  
session    required      pam_selinux.so open env_params  
session    optional     pam_keyinit.so force revoke
```

`pam_selinux` is a PAM module that sets up the default SELinux security context for the next executed process. These entries have module arguments, `close` and `open`. When a session is ended, the `close_session` part of the module restores old security contexts that were in effect before the change made by the `open_session` part of the module.

`pam_loginuid` is a PAM module that records the user's login UID, which is necessary for applications to be correctly audited.

`pam_keyinit` is the kernel session keyring initializer PAM module.

## PAM: Example #2

### Example # 2:

- Uses `value=action` pairs in the control flag field, allowing full control of PAM actions
  - [user\_unknown=ignore success=ok ignore=ignore default=bad]
- Uses authentication module arguments
  - pam\_unix.so nullok try\_first\_pass
  - pam\_succeed\_if.so uid >= 500 quiet
- Includes the contents of the common configuration file, system-auth
  - system-auth is included in nearly all individual service configuration files.
  - system-auth is auto-generated each time the authconfig command runs.



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

To demonstrate another example of PAM in action, the following is a partial listing of the `login` service's authentication stack:

```
# cat /etc/pam.d/login
#%PAM-1.0
auth [user_unknown=ignore success=ok ignore=ignore default=bad]
pam_securetty.so
auth           substack      system-auth
account required        pam_nologin.so
account include system-auth
password include system-auth
```

Processing of each line in the stack is described:

```
auth [user_unknown=ignore success=ok ignore=ignore default=bad
pam_securetty.so
```

The first noncommented line calls the `pam_securetty.so` module, which allows root logins only if logging in on a secure TTY, as defined in the `/etc/securetty` file. This module has no effect on nonroot users. This entry also contains specific actions for the control flag.

Several predefined control flag actions are available. This example uses the following actions and values:

- **`user_unknown=ignore`:**
  - `user_unknown`: The user is not known to the underlying authentication module.
  - `ignore`: The return status does not contribute to the return code.
- **`success=ok`:**
  - `success`: Successful function return
  - `ok`: If the module fails, the total stack state is fail. If the stack is already in fail status, the return code of this module does nothing.
- **`ignore=ignore`:**
  - `ignore`: Ignore underlying account modules regardless of whether the control flag is required, optional, or sufficient.
  - `ignore`: The return status does not contribute to the return code.
- **`default=bad`:**
  - `default`: All not explicitly mentioned values
  - `bad`: The return status is set to fail.

#### **`auth substack system-auth`**

The next line includes the contents of a common configuration file, `system-auth`, into the `/etc/pam.d/login` file. The `system-auth` configuration file is included in nearly all individual service configuration files. It checks that the user who is logging in is authorized to do so, including verification of the username and password. The `system-auth` configuration file is auto-generated each time the `authconfig` command is executed. Examples of `auth` entries in a `system-auth` file are:

```
auth    required    pam_env.so
auth    sufficient  pam_fprintd.so
auth    sufficient  pam_unix.so nullok try_first_pass
auth    requisite   pam_succeed_if.so uid >= 1000 quiet
auth    required   pam_deny.so
```

The `pam_env.so` module allows the setting and unsetting of environment variables. The `pam_fprintd.so` module is used for fingerprint authentication. The `pam_unix.so` module is the standard UNIX password authentication module. The `nullok` argument overrides the default action to not permit the user access to a service if the user's password is blank. The `try_first_pass` argument tries the previous stacked module's password before prompting the user for his or her password. The `pam_succeed_if.so` module tests account characteristics. In this case, it tests for `uid >= 1000`. The `quiet` argument means do not log success or failure to the system log file. The `pam_deny.so` module is the locking-out PAM module and can be used to deny access.

#### **`account required pam_nologin.so`**

The next line uses the `pam_nologin.so` module. This module prevents users from logging in to the system when the `/var/run/nologin` file or the `/etc/nologin.txt` file exists. This module has no effect on the `root` user.

Some of the remaining entries also include the contents of the `system-auth` file for module types of `account`, `password`, and `session`.



## Quiz

Which of the following are examples of PAM module types?

- a. requisite
- b. required
- c. auth
- d. account
- e. password
- f. sufficient
- g. session



ORACLE®

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

## Introduction to SELinux

- Standard Linux security is based on DAC.
- SELinux provides finer grained control.
- SELinux runs in three modes:
  - Enforcing
  - Permissive
  - Disabled
- Display the SELinux mode with `sestatus` or `getenforce` commands.
- SELinux also provides "Booleans."



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

In practice 4-1, you are presented with a new topic in Oracle Linux, SELinux. SELinux stands for “Security-Enhanced Linux” and is covered in another course. This brief introduction helps you understand the tasks you are directed to perform and why in practice 4-1 titled “Configuring PAM for a Single Login Session.”

Standard Linux security is based on Discretionary Access Control (DAC). With DAC, access to files and devices is based solely on user identity and ownership. Each file can have read, write, and execute permissions for the owner of the file, for the group, and for other users.

SELinux was created by the US National Security Agency to provide a finer-grained level of control over files, processes, users, and applications in the system. It is an enhancement to the Linux kernel, and it implements a different type of security called Mandatory Access Control (MAC). MAC policy is centrally managed rather than being managed by the user.

SELinux runs in one of three modes:

- **Enforcing:** Access is denied to users and programs unless permitted by SELinux security policy rules.
- **Permissive:** The security policy rules are not enforced, but SELinux sends denial messages to a log file.
- **Disabled:** SELinux does not enforce a security policy because no policy is loaded in the kernel. Only DAC rules are used for access control.

You can use the `sestatus` command to display the SELinux mode as well as some additional information about SELinux.

```
# sestatus
SELinux status:      enabled
...
Current mode:        enforcing
...
```

You can use the `getenforce` command to display the SELinux mode. This command displays the current mode: “Enforcing,” “Permissive,” or “Disabled.” Example:

```
# getenforce
Enforcing
```

SELinux running in “Enforcing” mode is often the cause of a problem in Linux. In some of the practices in this course, you are directed to change the SELinux mode to “Permissive” to get a function of Linux to work properly. You can use the `setenforce` command to change the mode to either “Enforcing” (1) or “Permissive” (0). Example:

```
# setenforce 0
# getenforce
Permissive
```

SELinux also provides “Booleans,” which allow parts of a SELinux policy to be changed at run time, without reloading or recompiling a SELinux policy. You can display a list of Booleans, state information, and a description of the Boolean by running the following command:

```
# semanage boolean -l
SELINUX boolean   State  Default Description
ftp_home_dir      (off, off)    Allow ftp to read and write ...
...
...
```

You can change the state of a specific Boolean to either `on` or `off` by using the `setsebool` command. For example, to turn the `ftp_home_dir` Boolean to `on`:

```
# setsebool ftp_home_dir on
```

Use the `getsebool` command to display the state of a specific Boolean. Example:

```
# getsebool ftp_home_dir
ftp_home_dir --> on
```

SELinux provides many other features, functions, and configuration options. This brief introduction helps you understand the tasks you are directed to perform and why in the following practice:

- Practice 4-1: Configuring PAM for a Single Login Session

## Summary

In this lesson, you should have learned how to:

- Describe the purpose of PAM
- Describe PAM configuration files
- Describe PAM authentication modules
- Describe PAM module types
- Describe PAM control flags
- Walk through PAM authentication examples



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

## Practice 4: Overview

The practices for this lesson cover the following topics:

- Configuring PAM for a single login session
- Configuring PAM to prevent non-root login

SELinux is referenced in the following practice:

- Configuring PAM for a single login session



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.

# Web and Email Services



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

# Objectives

After completing this lesson, you should be able to:

- Describe the Apache HTTP Web Server
- Configure Apache directives
- Configure Apache containers
- Configure Apache virtual hosts
- Describe email program classifications: MUA, MTA, MDA
- Describe email protocols: SMTP, POP, IMAP
- Describe the Postfix SMTP server
- Describe the Sendmail SMTP server
- Configure Sendmail on a client system



ORACLE®

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

# Apache HTTP Server

- Apache HTTP Web Server is included with Oracle Linux.
- Install the package:  

```
# yum install httpd
```
- Start the HTTP daemon:  

```
# systemctl start httpd
```
- The main configuration file:
  - /etc/httpd/conf/httpd.conf
- The auxiliary configuration directory:
  - /etc/httpd/conf.d
- The modules configuration directory:
  - /etc/httpd/conf.modules.d
- Use the apachectl command to check for configuration errors and to reload the configuration.

ORACLE

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

The Apache HTTP Server, an open-source web server developed by the Apache Software Foundation, is included with Oracle Linux. The Apache server is used to host web content. It responds to requests for content from web browsers, such as Internet Explorer and Firefox. To configure your system as a web server, begin by installing the `httpd` software package.

```
# yum install httpd
```

Use the `systemctl` utility to enable the HTTP daemon to start at boot time and also to start the daemon immediately.

```
# systemctl enable|start httpd
```

The main configuration file for Apache is `/etc/httpd/conf/httpd.conf`. An auxiliary directory, `/etc/httpd/conf.d`, also exists to store configuration files that are included in the main configuration file. Configuration files that load modules are in the `/etc/httpd/conf.modules.d` directory.

A new `apachectl` command is available in Oracle Linux 7. The following example uses the `configtest` subcommand to check the configuration for possible errors.

```
# apachectl configtest
```

Use the `graceful` subcommand to reload the configuration without affecting active requests.

```
# apachectl graceful
```

# Configuring Apache

Examples of configuration directives in the configuration file:

- Listen 192.168.2.1:8080
- ServerName www.example.com:80
- ServerRoot /etc/httpd
- DocumentRoot /var/www/html
- UserDir enabled oracle
- ErrorLog logs/error\_log
- LoadModule auth\_basic\_module modules/mod\_auth\_basic.so
- Order deny,allow
- Deny from all
- Allow from .example.com
- Timeout 60



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

Apache runs as installed, but you can modify configuration directives in this file to customize Apache for your environment. Some of these directives are described here. An index of all the directives is available at <http://httpd.apache.org/docs/2.4/mod/directives.html>.

## **Listen [IP address:]port**

Tells the server to accept incoming requests on the specified port or IP address and port combination. By default, the server responds to requests on all IP interfaces on port 80. If you specify a port number other than 80, a request to the server must include the port number (as in [www.example.com:8080](http://www.example.com:8080)). This is a required directive. Examples are as follows:

```
Listen 80  
Listen 192.168.2.1:8080
```

## **ServerName FQDN[:port]**

Specifies the fully qualified domain name or IP address of the server and an optional port that Apache listens on. The FQDN must be able to be resolved by DNS. If no FQDN is specified, Apache performs a DNS reverse name lookup on the IP address. If no port is specified, the server uses the port from the incoming request, as shown in the following example:

```
ServerName www.example.com:80
```

**ServerRoot *directory-path***

The top of the directory hierarchy under which the Apache server's configuration, error, and log files are kept. The default is /etc/httpd. Do not add a slash at the end of *directory-path*:

```
ServerRoot /etc/httpd
```

**DocumentRoot *directory-path***

The top of the directory hierarchy that holds the Apache server content. Do not end the path name with a slash. The Apache user needs read access to any files and execute access to the directory and any subdirectories in the hierarchy. The following is the default:

```
DocumentRoot /var/www/html
```

**UserDir *directory-path* | *disabled* | *enabled user-list***

Allows users identified by the *user-list* argument to publish content from their home directories. The *directory-path* is the name of a directory in a user's home directory from which Apache publishes content. If *directory-path* is not defined, the default is ~/public\_html. The following example enables this feature for user oracle. Assuming that the ServerName is [www.example.com](http://www.example.com), browsing to <http://www.example.com/~oracle> displays the oracle user's webpage.

```
UserDir enabled oracle
```

**ErrorLog *filename* | *syslog[:facility]***

Specifies the name of the file, relative to ServerRoot, that Apache sends error messages to. Alternatively, *syslog* specifies that Apache must send errors to rsyslogd. The optional *facility* argument specifies which rsyslogd facility to use. The default facility is local7.

```
ErrorLog logs/error_log
```

**LoadModule *module* *filename***

Apache, like the Linux kernel, uses external modules to extend functionality. These modules are called dynamic shared objects (DSOs). The *module* argument is the name of the DSO, and *filename* is the path name of the module, relative to ServerRoot. More than 60 modules are included with Apache, and more than 50 of these are loaded by default. An index of all the modules is available at <http://httpd.apache.org/docs/2.4/mod/>.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

**Allow from All | host [host ...]**

Specifies which clients can access content. All serves content to any client. Alternatively, you can list the specific hosts that are allowed access to content.

**Deny from All | host [host ...]**

Specifies which clients are not allowed access to content.

**Order deny,allow | allow,deny**

Specifies the order in which Allow and Deny directives are evaluated. deny,allow evaluates deny directives first and then allow directives. The following example grants access to clients from the example.com domain only, by first denying access to all and then allowing it from .example.com:

```
Order deny,allow
Deny from all
Allow from .example.com
```

**Timeout *num***

Specifies the number of seconds Apache waits for network operations to finish. The default is 60.

## Testing Apache

The screenshot shows a Mozilla Firefox window with the title bar "Apache HTTP Server Test Page powered by Linux - Mozilla Firefox". The address bar shows "localhost". The main content area displays the "Apache 2 Test Page" with the subtext "powered by the **Apache httpd server**". Below this, a message states: "This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that the Apache HTTP server installed at this site is working properly." There are two sections of text: "If you are a member of the general public:" and "If you are the website administrator:". The "general public" section says: "The fact that you are seeing this page indicates that the website you just visited is either experiencing problems or is undergoing routine maintenance. If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general mail sent to the name". The "website administrator" section says: "You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf. You are free to use the images below on Apache Linux powered HTTP servers."

ORACLE®

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

You can confirm that Apache is working by pointing a browser on the local system to <http://localhost> as shown. From a remote system, point a browser to `http://` followed by the `ServerName` directive that you specified in the configuration file. The test page, shown in the slide, confirms that Apache is working correctly.

To test the display of actual content, create an HTML file named `index.html` in the directory specified by the `DocumentRoot` directive (the default directory is `/var/www/html`). Apache automatically displays the `index.html` file in this directory, if it exists.

# Apache Containers

- Containers are special directives that group other directives.
  - `<Directory directory-path>` applies directives to directories within `directory-path`.
  - `<IfModule module-name>` applies directives if `module-name` is loaded.
  - `<IfModule !module-name>` applies directives if `module-name` is not loaded.
  - `<Limit method [method] ...>` limits access control directives to specified methods.
- Containers can be nested.



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

Apache containers are special configuration directives that group other directives. The containers use XML-style tags, meaning that the beginning of a container is `<name>` and the end is `</name>`. An index of all the container directives is available at <http://httpd.apache.org/docs/current/sections.html>. The following are examples of containers:

**`<Directory directory-path>`**

This container applies directives to directories within `directory-path`. The example applies Deny, Allow, and AllowOverride directives to all files and directories within the `/var/www/html/test` directory hierarchy. Indenting is for readability only.

```
<Directory /var/www/html/test>
    Deny from all
    Allow from 192.168.2.
    AllowOverride All
</Directory>
```

The `AllowOverride` directive in this container specifies classes of directives that are allowed in `.htaccess` files. The `.htaccess` files are other configuration files that typically contain user authentication directives. The `ALL` argument to `AllowOverride` means that all classes of directives are allowed in the `.htaccess` files. There are classes of directives that control authorization, control client access, control directory indexing, and others.

**<IfModule [!]module-name>**

This container applies directives if *module-name* is loaded. With the optional exclamation point, Apache does the inverse; that is, it sets the directives in the container if the *module-name* is not loaded. An example is as follows:

```
<IfModule mod_userdir.c>
    UserDir disabled
</IfModule>
```

**<Limit method [method] ...>**

This container limits access control directives to specified methods. An HTTP method specifies actions to perform on a Uniform Resource Identifier (URI). Examples of methods are GET (the default), PUT, POST, and OPTIONS. The following example disables HTTP uploads (PUT) from systems that are not in the example.com domain:

```
<Limit PUT>
    Order deny,allow
    Deny from all
    Allow from .example.com
</Limit>
```

**<LimitExcept method [method] ...>**

This container is the opposite of the Limit container in that it limits access control directives to all except specified methods.

The following example uses the LimitExcept container but also illustrates that containers can be nested. This example controls access to UserDir directories by restricting these directories to be read-only:

```
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch \
        IncludesNoExec
    <Limit GET POST OPTIONS>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

The Options directive controls server features by directory. Some of these are described:

- **Multiviews:** Allows a page to be displayed in different languages, for example
- **Indexes:** Generates a directory listing if the DirectoryIndex directive is not set
- **SymLinksIfOwnerMatch:** Follows symbolic links if the file or directory being pointed to has the same owner as the link

## Apache Virtual Hosts

- A single Apache server can respond to requests directed to multiple IP addresses or host names.
- Each virtual host can provide different content and can be configured differently.
- Use the `<VirtualHost host-name>` container:

```
<VirtualHost www.example1.com>
    ServerName www.example1.com
    DocumentRoot /var/www/example1
    ErrorLog example1.error_log
</VirtualHost>
<VirtualHost www.example2.com>
    ...
</VirtualHost>
```



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

Apache supports virtual hosts, meaning that a single Apache server can respond to requests directed to multiple IP addresses or host names. Each virtual host can provide different content and can be configured differently.

You can configure virtual hosts in two ways:

- IP-based Virtual Hosts (host-by-IP)
- Name-based Virtual Hosts (host-by-name)

With host-by-IP, each virtual host has its own IP address and port combination. The Apache web server responds to the IP address that the host resolves as. Host-by-IP is required for serving HTTPS requests due to restrictions in the Secure Sockets Layer (SSL) protocol.

With host-by-name, all virtual hosts share the common IP address. Apache responds to the request by mapping the host name in the request to `ServerName` and `ServerAlias` directives in the particular virtual host's configuration file.

Use the `<VirtualHost host-name>` container to implement virtual hosts. After the first `VirtualHost` is defined, all of the content served by Apache must also be moved into virtual hosts.

The following example is a simple name-based virtual hosts configuration:

```
<VirtualHost *:80>
    ServerName example1.com
    ServerAlias www.example1.com
    DocumentRoot /var/www/example1
    ErrorLog example1.error_log
</VirtualHost>
<VirtualHost *:80>
    ServerName example2.com
    ServerAlias www.example2.com
    DocumentRoot /var/www/example2
    ErrorLog example2.error_log
</VirtualHost>
```



## Quiz

On which port does Apache listen for client requests by default?

- a. 443
- b. 8080
- c. 80
- d. 280



**ORACLE®**

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

# Email Program Classifications

- Mail User Agent (MUA):
  - An email client application to create and read email messages
  - Some MUAs are capable of sending outbound messages to MTA.
  - Some MUAs are capable of retrieving messages from remote servers by using POP or IMAP.
- Mail Transfer Agent (MTA):
  - An email server that transports email messages by using SMTP
  - Examples: Sendmail, Postfix, Fetchmail
- Mail Delivery Agent (MDA):
  - Invoked by MTA
  - Puts incoming email in the recipient's mailbox file
  - Examples: Procmail or mail



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

An email message is created by a mail client program called a Mail User Agent (MUA) and delivered to the recipient's email server by Mail Transfer Agents (MTAs). From here, a Mail Delivery Agent (MDA) puts the message in the recipient's mailbox file.

## **Mail User Agent (MUA)**

An MUA is an email client application that allows you to create and read email messages, set up mailboxes to store and organize messages, and send outbound messages to an MTA. Many MUAs can also retrieve email messages from remote servers by using Post Office Protocol (POP) or Internet Message Access Protocol (IMAP).

## **Mail Transfer Agent (MTA)**

An MTA transports email messages between systems by using Simple Mail Transfer Protocol (SMTP). It provides mail delivery services from a client program to a destination server, possibly traversing several MTAs along the way. Oracle Linux offers two MTAs, Postfix and Sendmail, and also includes a special purpose MTA called Fetchmail.

## **Mail Delivery Agent (MDA)**

An MDA, such as Procmail, is invoked by the MTA to put incoming email in the recipient's mailbox file. MDAs perform the actual delivery. They distribute and sort messages on the local system for an email client application to access.

## Email Protocols

- Simple Mail Transfer Protocol (SMTP):
  - This is a transport protocol (an MTA).
  - Specify the SMTP server when configuring the client program.
  - Configure relay restrictions to limit junk email.
- Post Office Protocol (POP):
  - An email access protocol
  - Used by client programs to retrieve email messages
- Internet Message Access Protocol (IMAP):
  - This is similar to POP.
  - Email is kept on the server when using IMAP.
- POP and IMAP services are provided by the `dovecot` package.



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

Several different network protocols are required to deliver email messages. These protocols work together to allow different systems, often running different operating systems and different email programs, to send and receive email. The most commonly used protocols for transferring email are described here:

### **Simple Mail Transfer Protocol (SMTP)**

SMTP is considered to be a transport protocol (an MTA), as opposed to an email access protocol. It provides mail delivery services from a client program to a server and from an originating server to the destination server. You must specify the SMTP server when configuring the client program. You can also specify a remote SMTP server for outgoing email.

SMTP does not require authentication. Anyone can send anyone email, including junk email, also known as spam or unsolicited bulk email. You can configure relay restrictions that limit users from sending email through your SMTP server. Servers without any restrictions are called open relay servers.

The SMTP programs provided by Oracle Linux are Postfix and Sendmail. Because these programs use SMTP, they are often referred to as SMTP server programs.

## Post Office Protocol (POP)

POP is an email access protocol used by client programs to retrieve email messages from remote servers. Users on client systems usually have an email account on a server run by their employer or an Internet Service Provider (ISP). On Linux systems, the MUA on the receiving system either reads the mailbox file or retrieves the email from a remote SMTP server, using POP or IMAP. Unless you own a domain at which you want to receive email, you do not need to set up Sendmail as an incoming SMTP server.

## Internet Message Access Protocol (IMAP)

IMAP is similar to POP in that it is an email access protocol used to retrieve email remotely. The IMAP server is provided by the same `dovecot` package that provides the POP server. There are no new software packages to install.

While POP email clients typically delete the message on the server after it has been successfully retrieved, email is kept on the server when using IMAP. The entire message is downloaded only when it is opened. Messages can be read or deleted while still on the server. Both POP and IMAP allow you to manage mail folders and create multiple mail directories to organize and store email.

Oracle Linux includes the `dovecot` package to implement both the POP and IMAP protocols. To install the package:

```
# yum install dovecot
```

Start the daemon by entering the following command:

```
# systemctl start dovecot
```

To ensure that the service starts at boot time:

```
# systemctl enable dovecot
```

By default, `dovecot` runs IMAP and POP together with their secure versions using Secure Sockets Layer (SSL) encryption for client authentication and data transfer sessions. When starting `dovecot`, it reports that it started the IMAP server, but it also starts the POP server. The servers provided by `dovecot` are configured to work as installed. You typically do not need to modify the configuration file, `/etc/dovecot/dovecot.conf`. Refer to `/usr/share/doc/dovecot*` for more information.

## Postfix SMTP Server

- This is the default MTA with Oracle Linux.
- The main configuration files are in the `/etc/postfix` directory:
  - `access`: Specifies which hosts can connect to Postfix
  - `main.cf`: The global Postfix configuration file
  - `master.cf`: Specifies how Postfix processes interact
  - `transport`: Maps email addresses to relay hosts
- Restart the service after making any configuration changes:

```
# systemctl restart postfix
```

- Refer to [www.postfix.org](http://www.postfix.org) for complete documentation.



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

Postfix and Sendmail are two MTAs (SMTP servers) included with Oracle Linux. Postfix is configured as the default MTA. It is easier to administer than Sendmail, but does not include as many features. Postfix has a modular design that consists of a master daemon and several smaller processes. It stores its configuration files in the `/etc/postfix` directory. Some of the configuration files are described. Refer to [www.postfix.org](http://www.postfix.org) for complete documentation.

- **access**: This file is used for access control and specifies which hosts are allowed to connect to Postfix.
- **main.cf**: This is the global Postfix configuration file in which most of the configuration options are specified.
- **master.cf**: This file specifies how the Postfix master daemon interacts with the smaller processes to deliver email.
- **transport**: This file maps email addresses to relay hosts.

By default, Postfix does not accept network connections from any system other than the local host. To enable mail delivery for other hosts, edit the `/etc/postfix/main.cf` file and configure the domain, host name, and network information for the other hosts. Restart the service after making any configuration changes:

```
# systemctl restart postfix
```

## Sendmail SMTP Server

- This is an MTA included with Oracle Linux.
- One of the oldest and most common MTAs on the Internet
- Install two packages:
- Configuration files are located in /etc/mail:
  - sendmail.mc: Is the main configuration file
  - access: Specifies a relay host
  - virtusertable: Serves email to multiple domains
  - mailertable: Forwards email from one domain to another
- You must regenerate the configuration files after editing:

```
# yum install sendmail sendmail-cf  
# systemctl restart sendmail  
# make all -C /etc/mail
```



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

Sendmail is also included with Oracle Linux. It is one of the oldest and most commonly used MTAs on the Internet. The main purpose of Sendmail is to transfer email between systems, but it is highly configurable and capable of controlling almost every aspect of how email is handled.

To use Sendmail, install the following packages. The `sendmail-cf` package is required to configure Sendmail. The `procmail` package is installed as a dependency. In the default setup, the Sendmail MTA uses Procmail as the local MDA. The Procmail application writes email to the recipient's mailbox file.

```
# yum install sendmail sendmail-cf
```

The Sendmail configuration files are located in `/etc/mail`. The main configuration file is `sendmail.cf`, but it is not intended to be edited by using a text editor. Make any configuration changes in the `sendmail.mc` file and then generate a new `sendmail.cf` file by restarting the `sendmail` service:

```
# systemctl restart sendmail
```

You can also run the following `make` command, which calls the `Makefile` file in the `/etc/mail` directory and regenerates mail configuration files that have been modified.

```
# make all -C /etc/mail
```

Some of the other configuration files in the `/etc/mail` directory are described here:

- **access:** This file sets up a relay host. A relay host processes outbound mail for other systems. The default configuration is to relay mail only from the local host:

Connect: localhost.localdomain	RELAY
Connect: localhost	RELAY
Connect: 127.0.0.1	RELAY

To configure your system to relay mail from other systems (for example, the 192.168 subnet), add the following entry:

Connect: 192.168	RELAY
------------------	-------

- **virtusertable:** This file serves email to multiple domains. Each line starts with the address that the email was sent to, followed by the address Sendmail forwards the email to. For example, the following entry forwards email addressed to any user at `foo.org` to the same username at `example.com`:
- **mailertable:** This file forwards email from one domain to another. The following example forwards email sent to the `foo.org` domain to the SMTP server for the `example.com` domain:  
`foo.org smtp: [example.com]`

The configuration files in the `/etc/mail` directory have corresponding `.db` files. Example:

```
# ls /etc/mail/*table*
domaintable    mailertable    virtusertable
domaintable.db mailertable.db virtusertable.db
```

Make any configuration changes to the files without extensions. Sendmail uses the `.db` files, however. To update or regenerate the `.db` files for Sendmail, either restart the `sendmail` service or run the `make` command after making any configuration changes.

#### **/etc/aliases**

The `/etc/aliases` file can also be used to forward incoming email messages. Use the file to map inbound addresses to local users, files, commands, and remote addresses. The following example forwards mail sent to `admin` on the local system to several users, including `user4`, who is on a different system:

```
admin: user1, user2, user3, user4@different.com
```

To direct email to a file, specify the absolute path name of the file in place of the destination address. The `/etc/aliases` file is writable only by the `root` user.

#### **~/.forward**

Individual users can forward incoming email messages by creating a `.forward` file in their home directory. Simply specify a different email address in the `~/.forward` file. Example:

```
user1@host02.example.com
```

You can also specify another user, or a file, or a command to pipe the email to. Separate multiple entries with a comma or a newline.

## Configuring Sendmail on a Client

- Sendmail on a client system simply relays outbound mail to an SMTP server.
- A remote SMTP server, typically an ISP, relays email to its destination.
- Edit the following line in /etc/mail/sendmail.mc:
  - `dnl define(`SMART_HOST', `smtp.your.provider')dnl`
- Remove the `dnl` at the beginning of the line.
- Include the ISP's SMTP server name:
  - `define(`SMART_HOST', `smtp.isp.com')dnl`
- Restart the sendmail service:

```
# systemctl restart sendmail
```



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

Sendmail on a client system simply relays outbound mail to an SMTP server. A remote SMTP server, typically an ISP, relays outbound email to its destination. The following example configuration sends email only to the SMTP server that originates on the local system. It does not forward email originating from other systems. This configuration does not handle inbound email either. Client systems normally use POP or IMAP to receive email.

To configure the system as described, locate the following entry in the main email configuration file, /etc/mail/sendmail.mc:

```
dnl define(`SMART_HOST', `smtp.your.provider')dnl
```

The `dnl` at the beginning of the line is a comment. Delete these characters. Replace “`smtp.your.provider`” with the FQDN of your ISP's SMTP server. You can choose to delete the `dnl` characters at the end of the line or not delete them. Assuming that your ISP's SMTP server is `smtp.isp.com`, change the line to appear as follows:

```
define(`SMART_HOST', `smtp.isp.com')dnl
```

Restart the sendmail service, which regenerates the `sendmail.cf` file.

```
# systemctl restart sendmail
```

Send an email message to an account that you have on a remote server to ensure that `sendmail` is relaying your email.



## Quiz

Sendmail is an example of what type of email program classification?

- a. MUA
- b. MTA
- c. MDA
- d. MMA



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

## Summary

In this lesson, you should have learned how to:

- Describe the Apache HTTP Web Server
- Configure Apache directives
- Configure Apache containers
- Configure Apache virtual hosts
- Describe email program classifications: MUA, MTA, MDA
- Describe email protocols: SMTP, POP, IMAP
- Describe the Postfix SMTP server
- Describe the Sendmail SMTP server
- Configure Sendmail on a client system



ORACLE®

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

## Practice 5: Overview

The practices for this lesson cover the following topics:

- Configuring the Apache Web Server



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.

# Installing Oracle Linux by Using PXE and Kickstart



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

## Objectives

After completing this lesson, you should be able to:

- Describe the Kickstart file
- Verify the Kickstart file
- Describe the role of PXE installations
- Describe the PXE process
- Explore setting up a Preboot Execution Environment (PXE)
- Start a PXE/Kickstart installation



ORACLE®

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

## Kickstart Installation Method

To automate the installation of Oracle Linux:

- Create a Kickstart file that contains installation parameters.
- Make the Kickstart file available on a boot disk, on a boot CD, or on the network.
- Make the Oracle Linux installation tree available from the installation CD, from the ISO image stored on a hard drive, or over the network.
- Use NFS, FTP, or HTTP to provide access to the installation tree over the network.
- Initiate the PXE/Kickstart installation from the boot prompt.



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

The Kickstart installation method allows you to perform an unattended installation of Oracle Linux. Requirements to implement Kickstart include the creation of a Kickstart file, which contains the answers to all the questions you are asked during a normal installation. You then must make the Kickstart file available on a boot disk, on a CD, or on the network. Kickstart also needs access to the Oracle Linux installation tree from the installation CD, from the ISO image stored on a hard drive, or over the network. You can use NFS, FTP, or HTTP to provide access to the installation tree over the network.

## Kickstart File

- The Kickstart file contains answers to installation questions.
  - The Command section defines installation options and associated values.
  - The %packages section contains the names of package groups and individual package names to be installed.
  - The optional %pre section contains commands to run before the installation begins.
  - The optional %post section contains commands to run after the installation is completed.
- Every installation creates a Kickstart file, /root/anaconda-ks.cfg.
  - This file can be used as a template for future installations.



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

To perform a Kickstart or PXE installation, one of the key requirements is the existence of a Kickstart file.

A Kickstart file contains the answers to all the questions posed during an installation. The Kickstart file contains the following sections:

- **Command section:** Defines the installation options and associated values
- **%packages section:** Defines the packages to install
- **%pre and %post sections:** Defines preinstallation and postinstallation commands

Every installation creates a Kickstart file, /root/anaconda-ks.cfg. This file can be used “as is” to repeat the installation, or it can be modified to specify different settings.

The following example demonstrates the syntax for providing system information in the Command section. The example defines values for the language, keyboard type, root password, time zone, and network. In this example, the root password is encrypted:

- lang en\_US.UTF-8
- keyboard --vckeymap=us --xlayouts='us'
- rootpw --iscrypted \$6\$...
- timezone America/Denver --isUtc --nontp
- network --bootproto=static --device=eth0 --gateway=192.0.2.1 --ip=192.0.2.107 --netmask=255.255.255.0 --ipv6=auto --activate

## Verifying the Kickstart File

- Use the `ksvalidator` utility to check for errors in the Kickstart file.
- It does not validate the syntax of `%pre` and `%post` scripts.
- Install the `pykickstart` package:

```
# yum install pykickstart
```

- Provide path to the Kickstart file as argument:

```
# ksvalidator /root/ks.cfg
The following problem occurred on line 15 of the kickstart file:
no such option: -b
```



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

You can use the `ksvalidator` utility to verify the syntax of a Kickstart file. The utility checks for proper quoting, valid options, and ensures that any required options have values.

The `ksvalidator` utility does not validate the syntax of the `%pre` and `%post` scripts. It also does not guarantee that a Kickstart file installs properly because `ksvalidator` cannot deal with the complexities of partitioning and what potentially already exists on the disk.

Install the `pykickstart` package:

```
# yum install pykickstart
```

Provide the Kickstart file name as an argument. For example:

```
# ksvalidator /root/ks.cfg
```

Output only occurs if syntax errors are found.

# Setting Up a Preboot Execution Environment (PXE) Server

This section explores the following PXE topics:

- What is PXE
- How PXE works
- Pre-installation configuration
- The PXE server configuration steps



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

This section begins with a discussion on what the preboot execution environment is, why you use it, and how it works. The section begins with the pre-installation tasks and ends with a walkthrough of the configuration steps.

## What Is PXE?

The Preboot Execution Environment (PXE):

- Enables remote installation of multiple servers with the same basic configuration
- Is an open industry standard supported by a number of hardware and software vendors
- Utilizes a combination of protocols:
  - DHCP
  - TFTP
  - HTTP/NFS
- Enables a server's Network Interface Card (NIC) to function like a boot device



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

### What Is PXE?

Pre-boot Execution Environment (PXE) is one method of provisioning a server to boot over the network instead of using a CD-ROM. PXE is an open industry standard supported by a number of hardware and software vendors. Using this automated approach is useful when you are deploying tens or even hundred of servers.

The PXE environment utilizes a combination of protocols, which enables the booting and provisioning of servers over the network.

PXE works with the Network Interface Card (NIC) of the system by making it function like a boot device.

# PXE Execution Process

The pre-boot execution process works as follows:

1. You boot the client (target) machine.
2. Client machine's NIC card initiates a DHCP request.
3. DHCP server intercepts the request and responds with IP networking information, the location of the TFTP server, and the boot image file.
4. Client contacts the TFTP server for the boot image file.
5. TFTP server sends the boot image file to the client, and the client executes the file. The boot image searches the TFTP server for the boot configuration files (kernel and root file system).
6. Client downloads and loads the files needed.
7. Client reboots and uses Kickstart method to automate the installation.



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

## How PXE Works

The PXE execution process is as follows:

1. Client machine boots.
2. The Client's Network Interface Card (NIC) triggers a DHCP request.
3. The DHCP server intercepts the request and responds with standard network information (IP, subnet mask, gateway, DNS, etc.). In addition, it provides information about the location of a TFTP server and boot image (e.g., pxelinux.0).
4. When the client receives this information, it contacts the TFTP server in order to obtain the boot image.
5. The TFTP server sends the boot image, and the client executes it. By default, the boot image searches the `pxelinux.cfg` directory on the TFTP server for the boot configuration files on the TFTP server.
6. The client downloads all the files it needs (kernel and root file system) and then loads them.
7. The target machine reboots.

The Provisioning application uses the Kickstart method to automate the installation on target machines. The host specific boot configuration file contains the location of the kickstart file.

# PXE Pre-Installation Steps

To prepare for a PXE-based installation, you need to:

1. Install and set up an HTTP or an NFS server and share the Oracle Linux ISO installation repository on to the network.

```
# yum install httpd
# rpmquery httpd
httpd-2.4.6-31.0.1.el7.x86_64
```

- a. Mount the ISO image and copy the entire installation tree to either a web server or an NFS server.

```
# mkdir /mnt
# mount -t iso9660 -o loop ol7u5-server-x86-64-dvd.iso /mnt
# mkdir /var/www/html/OL75
# cp -a -T /mnt/* /var/www/html/OL75/
# umount /mnt
```



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

## Complete the Pre-Installation Steps

To prepare for a PXE-based network installation, you need to stage the complete Oracle Linux ISO image and make it available on the network. In addition, you need to set up the following services and configure them to support PXE clients:

- Either an HTTP or an NFS server. These servers stage the Oracle Linux distribution and the Kickstart file for the remote installation.
- The example on this slide shows the commands to install the HTTP service and copy the entire ISO installation tree to a designated directory on the HTTP server.
- The ISO image contains everything you need to boot and install Oracle Linux over the network.

Instructions for configuring NFS shares on Oracle Linux systems can be found in the Oracle Linux Administrator's Guide.

## PXE Pre-Installation Steps (Cont.)

- b. Edit the `/etc/httpd/conf/httpd.conf` file and enclose the following directives within a new `<Directory>` section to the bottom of the file.

```
<Directory "/var/www/html/OSimage/OL75">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
    Require all granted
</Directory>
```



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

### Configure the Apache Server (`httpd.conf`) File

Configuring the `httpd.conf` file sets up the HTTP directory of the network server that hosts the installation RPM packages.

Add a `<Directory>` section that points to the location of the ISO installation packages. These options and directives allow the following:

- `<Directory "/var/www/html/OL75">` - This is the full path to the directory of the ISO installation files.
- `Options Indexes FollowSymLinks` - The `Indexes` option returns a formatted listing of the directory when there is no `index.html` file in that directory. The `FollowSymLinks` option allows the server to follow symbolic links in the directory.
- `AllowOverride None` directive disables the Apache server from running `.htaccess` files, which slows down the server.
- `Order allow,deny` directive turns off authentication and allows access from all hosts in the domain.

**Note:** These directives are for configuring the HTTP directories for Oracle Linux 7 systems. For other versions of Oracle Linux, consult the appropriate Oracle Linux documentation.

## PXE Pre-Installation Steps (Cont.)

### 2. Install network services to support PXE:

#### a. A DHCP server

- Install the `dhcp` package.

```
# yum install dhcp
# rpmquery dhcp
dhcp-4.2.5-36.0.1.el7.x86_64
```

#### b. A TFTP server

- Install the `tftp-server` package. This also installs the `xinetd` package.

```
# yum install tftp-server
# rpmquery tftp-server xinetd
tftp-server-5.2-11.el7.x86_64
xinetd-2.3.15-12.el7.x86_64
```



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

### Installing network services (DHCP and TFTP)

In addition to setting up the ability to share the Oracle Linux ISO on the network, you need to install the packages for the DHCP and TFTP servers. To install Oracle Linux on PXE clients, the DHCP and TFTP do not have to run on the same host. If you configure separate DHCP and TFTP servers, the DHCP server must define the TFTP server from which a client can download the boot loader, installation kernel, and initial ram-disk files.

The examples on the slide show executing the `yum install` command to install the `dhcp` and `tftp-server` packages. The `rpmquery` command confirms that the packages did install successfully and the version of each package.

The `tftp-server` package also contains the `xinetd` package. The `xinetd` daemon process starts programs that provide Internet services, in the same manner as the `inetd` program. The `xinetd` process remains dormant until a connection request arrives. `xinetd` listens on all service ports for the services listed in its configuration file. When a request comes in, `xinetd` starts the appropriate server.

## PXE Pre-Installation (Cont.)

### 3. Obtain the boot loader files:

- Install the **syslinux** package for BIOS-based PXE clients.
  - You use the `pxelinux.0` file.

```
# yum install syslinux
# rpmquery syslinux
syslinux-4.05-12.el7.x86_64
```

- Install the **grub2-efi** package for UEFI-based PXE clients.
  - You use the `grubx64.efi` file.

```
# yum install grub2-efi
# rpmquery grub2-efi
grub2-efi-2.02-0.16.0.3.el7.x86_64
```



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

### Obtaining the Boot Loader Files

The last package you need to install is the package(s) for either BIOS-based or UEFI-based PXE clients. **syslinux** is the bootloader for Linux BIOS-base systems.

The GRand Unified Bootloader (GRUB) contains a customizable bootloader for EFI systems.

# Configuring DHCP and TFTP Services

After installing the required packages, you need to configure the DHCP and TFTP services to support PXE clients.

## 1. Configure the DHCP server:

- a. Edit the `/etc/dhcp/dhcpd.conf` file. Add entries for PXE clients.
- b. Define a PXE type:

```
set vendorclass = option vendor-class identifier;
option pxe-system-type code 93 = unsigned integer 16;
set pxetype = option pxe-system-type;
```



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

## Configuring DHCP Server

After package installation, the configuration file is empty of any options or directives. You need to configure this file to meet your networking design, plus add entries for the PXE clients.

Edit the `/etc/dhcp/dhcpd.conf` file and configure entries for the PXE clients. The first entries to add defines the type of architecture (BIOS or UEFI) and the PXE type. Add these lines to the top of the file.

## Configuring DHCP and TFTP Services

- b. Associate the boot loader that the `filename` parameter specifies to a PXE type.

```
if substring(vendorclass, 0, 9)="PXEClient" {  
    if pxetype=00:06 or pxetype=00:07 {  
        filename "efi/grubx64.efi";  
    } else {  
        filename "pxelinux/pxelinux.0";  
    }  
}
```

- c. Configures a pool of generally available IP addresses within a specified range. Specify the PXE boot server within that range.

```
pool {  
    range 192.0.2.100 192.0.2.200;  
}  
    next-server 192.0.2.103;  
}
```



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

### Configuring DHCP Service

The following page shows an example of a `dhcpd.conf` file configured to support PXE clients.

```
# cat /etc/dhcp/dhcpd.conf
set vendorclass = option vendor-class-identifier;
option pxe-system-type code 93 = unsigned integer 16;
set pxetype = option pxe-system-type;

subnet 192.0.2.0 netmask 255.255.255.0 {
    interface eth0;
        option routers                  192.0.2.1;
        option subnet-mask              255.255.255.0;
        option domain-name              "example.com";
        option broadcast-address       192.0.2.255;
        option time-offset              -25200; # Mountain Standard Time
        default-lease-time            21600;
        max-lease-time                43200;

    if substring(vendorclass, 0, 9)=="PXEClient" {
        if pxetype=00:06 or pxetype=00:07 {
            filename "efi/grubx64.efi";
        } else {
            filename "pxelinux/pxelinux.0";
        }
    }
    pool {
        range 192.0.2.200 192.0.2.254;
    }
        next-server 192.0.2.103;
}
```

After configuring the file, you need to start the DHCP service, and configure it to start after a reboot.

```
# systemctl start dhcpd
# systemctl enable dhcpd
```

# Configuring DHCP and TFTP Services

- Configure the TFTP service by editing the `/etc/xinetd.d/tftp` and modify the `disable` and `server_args` attributes:

```
service tftp
{
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /var/lib/tftpboot
    disable = no
    per_source = 11
    cps = 100 2
    flags = IPv4
}
```



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

## Configuring TFTP Service

In the `/etc/xinetd.d/tftp` file, change the `disabled` parameter from `yes` to `no` to enable `xinetd` to start the TFTP service (`in.tftpd`). Also, define the TFTP server directory to be `/var/lib/tftpboot`, which is the default. When `xinetd` receives a TFTP request, it starts `in.tftpd` and directs the request to it.

For more information about the configuration attributes, see the `xinetd.conf(5)` manual page.

# Configuring TFTP Services

- a. Create the `/var/lib/tftpboot` subdirectories:

- I. For BIOS-based clients, create the `pxelinux/pxelinux.cfg` directories:

```
# mkdir -p /var/lib/tftpboot/pxelinux/pxelinux.cfg
```

- II. For UEFI-based clients, create the `efi` directory:

```
# mkdir -p /var/lib/tftpboot/efi
```

- b. Copy the boot loader files, the installation kernel (`vmlinuz`), and the ram-disk image file (`initrd.img`) to the TFTP server subdirectories.

- I. For BIOS-based clients:

```
# cp /usr/share/syslinux/pxelinux.0  
    /var/lib/tftpboot/pxelinux  
# wget http://192.0.2.1/OL75/isolinux/vmlinuz  
    -O /var/lib/tftpboot/pxelinux/vmlinuz
```



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

## Configuring TFTP Service

Create the `/var/lib/tftpboot` subdirectories and copy the boot loader and ram-disk file to the `/var/lib/tftpboot/pxelinux` directory.

The `/tftpboot/pxelinux.cfg` directory contains the PXE configuration file(s) for the PXE clients to PXE/kickstart.

# Configuring TFTP Services

## II. For UEFI-based clients:

```
# cp /tempdir/boot/efi/EFI/redhat/grubx64.efi /var/lib/tftpboot/efi  
# cp /tempdir/boot/efi/EFI/redhat/shim.efi  
    /var/lib/tftpboot/efi  
# cp /tempdir/boot/efi/EFI/redhat/MokManager.efi  
    /var/lib/tftpboot/efi  
# wget http://10.0.0.11/OSimage/OL7/isolinux/vmlinuz -O  
    /var/lib/tftpboot/efi/vmlinuz  
# wget http://10.0.0.11/OSimage/OL7/isolinux/initrd.img  
    /var/lib/tftpboot/efi/initrd.img
```



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

## Configuring TFTP Service

For UEFI-based clients, copy the BIOS boot loader file, the installation kernel, and the ram-disk image file to the `efi` directory. You only need to copy the `shim.efi` and `MokManager.efi` files if you need to support Secure Boot on clients. The `MokManager.efi` provides utilities for managing the keys used to sign EFI binaries.

# Configuring TFTP Services

## c. Create the boot loader configuration file(s).

- The `default` file is the default boot loader configuration file for BIOS-based PXE clients.

```
# cat default
prompt 0
default o17
timeout 0
label o17
kernel vmlinuz
append initrd=initrd.img inst.ks=http://192.0.2.1/ks.cfg
```



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

## Configuring TFTP Service

The `default` file is the default boot loader configuration file for BIOS-based PXE clients and uses pxelinux configuration settings. An example of the file is on the slide.

With the `prompt` directive, you can toggle displaying a boot prompt by changing the value between 0 and 1, with 0 (zero) not displaying the prompt.

The `default` directive identifies the default boot entry by its label value, `o17`.

Pxelinux boots the client using the default boot entry after `timeout/10` seconds.

The `kernel` directive defines the name of the kernel executable and the `append` directive defines any parameters that should be appended when loading the kernel, such as the name of the ram-disk image and the location of a kickstart file.

# Configuring TFTP Services

- The `grub.cfg` file is the default boot loader configuration file for UEFI-based PXE clients.

```
# cat grub.cfg
set default 0
set timeout=10
Boot Loader Configuration for UEFI-Based PXE Clients
16
menuentry 'ol7' {
    echo "Loading efi/vmlinuz"
    linuxefi efi/vmlinuz inst.repo=http://192.0.2.1/OSimage/OL7 inst.ks.sendmac \
    inst.ks=http://192.0.2.1/ksfiles/ol7_cfg.ks
    echo "Loading efi/initrd.img"
    initrdefi efi/initrd.img
    echo "Booting installation kernel"
}
```



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

## Configuring TFTP Service

The `grub.cfg` file is the default boot loader configuration file for UEFI-based PXE clients and uses GRUB 2 configuration settings. An example of the file is on the slide.

The `linuxefi` directive defines the name of the kernel executable and defines any parameters that should be appended when loading the kernel, such as the location of the installation packages and how to access these packages. This example uses HTTP to install the packages from the specified URL. The `initrdefi` directive defines the name of the ram-disk image.

The kernel and ram-disk image file paths are assumed to be relative to the subdirectory that contains the boot loader, for example `efi`. If you place the `vmlinuz` and `initrd.img` files in a subdirectory such as `efi/OL7`, ensure you have the correct relative paths.

By default, GRUB 2 does not provide any indication that it is transferring the kernel and ram-disk images files. The `echo` statements in the example above provide a simple indication of progress.

## Beginning a Kickstart Installation

- Booting the PXE client initiates the kickstart installation.

```
Xen-host07 - TigerVNC
gPXE 1.0.0 -- Open Source Boot Firmware -- http://etherboot.org
Features: AoE HTTP iSCSI DNS TFTP bzImage COMBOOT ELF Multiboot PXE PXEXT

net0: 00:16:3e:00:01:07 on PCI00:04.0 (open)
  [Link:up, TX:0 RX:0 RXE:0]
DHCP (net0 00:16:3e:00:01:07).... ok
net0: 192.0.2.200 255.255.255.0 gw 192.0.2.1
Booting from filename "pxelinux/pxelinux.0"
tftp://192.0.2.103/pxelinux/pxelinux.0. ok

PXELINUX 4.05 0x548cd619 Copyright (C) 1994-2011 H. Peter Anvin et al
!PXE entry point found (we hope) at 9AB5:0395 via plan A
UNDI code segment at 9AB5 len 0794
UNDI data segment at 9B2F len 2D10
Getting cached packet 01 02 03
My IP address seems to be C00002C8 192.0.2.200
Ip=192.0.2.200:192.0.2.103:192.0.2.1:255.255.255.0
BOOTIF=01-00-16-3e-00-01-07
SYSUUID=f5858bcc-142f-34d2-9513-f4c18da88083
TFTP prefix: pxe/linu/
Trying to load: pxe/linu/.cfg/default
Loading vmlinuz..... ok
Loading initrd.img..... ready.
Probing EDD (edd=off to disable)... ok
```

ORACLE®

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

To begin a PXE/Kickstart installation, boot the client system. By using the protocols, network services, and directives in the various configuration files modified, booting the client initiates the PXE/Kickstart installation. You can see where the client's NIC triggers a DHCP request and the point where the DHCP server responds with standard network information (IP, subnet mask, gateway, DNS, etc.). The TFTP server locates the files required to boot and initiate the client kickstart installation. A short time duration exists from initial boot to the start of the Oracle Linux installation.



## Quiz

The %packages section in the Kickstart file can contain package group names as well as individual package names.

- a. True
- b. False



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe the Kickstart file
- Verify the Kickstart file
- Describe the role of PXE installations
- Describe the PXE process
- Explore setting up a Preboot Execution Environment (PXE)
- Start a PXE/Kickstart installation



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

## Practice 6: Overview

The practices cover the following topics:

- Creating a Kickstart File
- Configure gateway as an HTTP Server
- Configuring PXE Services
- Video of Performing a PXE/Kickstart Installation



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

# Samba Services



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

## Objectives

After completing this lesson, you should be able to:

- Describe the purpose of Samba
- Describe Samba services and daemons
- Configure a Samba server
- Describe Samba server types
- Access Samba shares from a client



ORACLE®

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

## Samba: Introduction

- Samba:
  - Is an open source implementation of the Server Message Block (SMB) protocol
  - Enables Linux and Windows systems to share files and printers
- Samba packages included with Oracle Linux:
  - **samba**: SMB/CIFS server package
  - **samba-client**: Allows clients to access SMB/CIFS shares and printers
  - **samba-common**: Provides files necessary for both the server and client Samba packages
  - **samba-winbind**: Provides the `winbind` daemon and client tools
  - **samba-winbind-clients**: Provides the NSS library and PAM modules needed to communicate with `winbind`



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

Samba is an open-source implementation of the Server Message Block (SMB) protocol. It allows Linux to work with the Windows operating system, as both a server and a client. Samba shares Linux files and printers with Windows systems, and also gives Linux users access to files on Windows systems. Samba uses NetBIOS over TCP/IP (NetBT) protocols and does not need the NetBEUI (Microsoft Raw NetBIOS frame) protocol.

Several Samba packages are included with the Oracle Linux distribution:

- **samba**: Provides an SMB/Common Internet File System (CIFS) server that can be used to provide network services to SMB/CIFS clients
- **samba-client**: Provides some SMB/CIFS clients to complement the built-in SMB/CIFS file system in Linux. These clients allow access to SMB/CIFS shares and printing to SMB/CIFS printers
- **samba-common**: Provides files necessary for both the server and client Samba packages
- **samba-winbind**: Provides the `winbind` daemon and client tools. `winbind` enables Linux membership in Windows domains and the use of Windows user and group accounts
- **samba-winbind-clients**: Provides the Network Security Services (NSS) library and Pluggable Authentication Modules (PAM) needed to communicate with `winbind`

## Samba Daemons and Services

- The `samba` server package includes two daemons:
  - `smbd`: Provides file and print services for Samba clients
  - `nmbd`: NetBIOS nameserver
- The `samba-winbind` package includes one daemon:
  - `winbindd`: Resolves user/group information on Windows
- Each daemon has an associated service:
  - `smb`
  - `nmb`
  - `winbind`



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

Use `yum install <package_name>` to install the packages. The `samba` server package includes the following daemons and associated services:

- **`smbd`:** This server daemon provides file-sharing and printing services to Windows clients. It is also responsible for user authentication, resource locking, and data sharing through the SMB protocol.
- **`nmbd`:** The NetBIOS nameserver daemon replies to name-service requests produced by SMB/CIFS in Windows-based systems. It also provides browsing support in the Windows Network Neighborhood view.

These daemons are controlled by their associated services, `smb` and `nmb`, for example:

```
# systemctl start smb  
# systemctl start nmb
```

The `samba-winbind` package includes the `winbindd` daemon and associated service:

- **`winbindd`:** Resolves user and group information on a server running Windows and makes this information understandable by Linux

This daemon is controlled by the `winbind` service:

```
# systemctl start winbind
```

# Samba Server Configuration

- The main configuration file for Samba is: /etc/samba/smb.conf
- The configuration file contains the following sections:
  - [global]: Defines global parameters
  - [homes]: Defines shares in the homes directory
  - [printers]: Defines printers
  - [share name]: Defines a share
- Example of share definition:

```
[tmp]
path = /tmp
writable = yes
guest ok = yes
```



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

The main configuration file for Samba is /etc/samba/smb.conf. This file is divided into sections, each beginning with text surrounded by square brackets. With the exception of the [global] section, each section describes a shared resource, known as a “share.” Typical sections are:

- [global]: Defines global parameters
- [homes]: Defines shares in the home directory
- [printers]: Defines printers
- [share name]: Defines a share

Parameters within the section define the share attributes. Assuming that the global parameters are configured properly, the following example defines a share that gives any Windows user read/write permissions to the local /tmp directory:

```
[tmp]
comment = Insert a comment here
path = /tmp
writable = yes
guest ok = yes
```

Refer to the `smb.conf(5)` man page for a description of all the parameters that you can set in the configuration file. There are global parameters, security parameters, logging parameters, browser parameters, communication parameters, and share parameters. There are also several graphical user interfaces to configure and manage Samba. A list of these can be found at <http://www.samba.org/samba/GUI/>.

### [homes] Share

Samba provides this share to make it easy for users to share their Linux home directories with a Windows system. The following is an example:

```
[homes]
comment = Insert a comment here
browsable = no
writable = yes
```

These settings prevent users other than the owners from browsing home directories, while allowing logged-in owners full access.

### Starting a Samba Server

To start a Samba server:

```
# systemctl start smb
```

When making configuration changes to the `/etc/samba/smb.conf` file, issue a restart or reload:

```
# systemctl restart smb
# systemctl reload smb
```

The `reload` argument does not stop and start the `smb` service; it only reloads the configuration file.

Use the `systemctl` command to configure the service to start at boot time. Example:

```
# systemctl enable smb
```

# Samba Server

- Three server types to configure a Samba server:
  - Stand-alone server
  - Domain member server
  - Domain controller
- Difference between a workgroup and a domain:
  - A Windows workgroup is a smaller, peer-to-peer network with no centralized management.
  - A Windows domain is a larger network of computers that share security and access control. Centralized management is provided by a domain controller.



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

There are three server types to configure a Samba server:

- Stand-alone server
- Domain member server
- Domain controller

To understand the differences between the different server types, a brief introduction to Windows environments is necessary.

## Windows Workgroups and Domains

Computers running Windows on a network belong to a workgroup or to a domain. Workgroup networks consist of a small number of computers when compared to a domain. Domains are best suited for corporate networks with many systems networked together.

A workgroup environment is a peer-to-peer network. Computers do not rely on each other for services. There is no centralized management. Each computer is sustainable on its own. Each computer has its own set of user accounts, its own access control, and its own resources. The systems can share resources, however, if configured to do so.

A domain is a trusted group of computers that share security and access control. Domains provide centralized management and security from a separate computer called a domain controller. Most modern Windows domains use Active Directory.

# Samba Server Types

- Server type is configured in the [global] section of the /etc/samba/smb.conf file.
- A stand-alone server can be a workgroup server or a member of a workgroup.
- A domain member server logs in to a domain controller and is subject to the domain's security rules.
- A Samba server can be a domain controller in a Windows NT domain but not in an Active Directory domain.



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

## Stand-Alone Server

A stand-alone Samba server can be a workgroup server or a member of a workgroup environment and does not participate in a Windows domain in any way. The following is an example of configuring the [global] directives in /etc/samba/smb.conf for a stand-alone server:

```
[global]
workgroup = workgroup_name
netbios name = netbios_name
security = share
```

The security parameter set to share indicates share-level security as opposed to user-level security. With share-level security, the server accepts only a password without an explicit username from the client. The server expects a password for each share, independent of the username. The use of share-level security is discouraged in favor of user-level security. There are four different ways to implement user-level security—user, server, domain, and ads—each of which is discussed in the “Authentication and Directory Services” lesson.

## Domain Member Server

A domain member server is similar to a stand-alone server, but the server is logged in to a domain controller (either Windows or Samba) and is subject to the domain's security rules. An example of a domain member server would be a departmental server running Samba that has a machine account on the Primary Domain Controller (PDC). All of the department's clients still authenticate with the PDC, but the departmental server controls printer and network shares. To set up a domain member server, you must first join the domain or Active Directory by using the `net join` command before starting the `smb` service.

The following is an example of configuring `/etc/samba/smb.conf` to implement an Active Directory domain member server. Samba authenticates users for services being run locally, but is also a client of the Active Directory.

```
[global]
realm = EXAMPLE.COM
security = ADS
password server = kerberos.example.com
```

The `realm` directive identifies the Kerberos realm and must be capitalized. Kerberos is an authentication protocol that allows nodes communicating over a nonsecure network to prove their identity to one another. Windows requires Kerberos for Active Directory authentication. The `password server` directive is required only if Active Directory and Kerberos are running on different servers.

The following is an example of configuring `/etc/samba/smb.conf` to implement a Windows NT4-based domain member server. NT4-based domains do not use Kerberos in their authentication method.

```
[global]
workgroup = workgroup_name
netbios name = netbios_name
security = domain
```

## Domain Controller

A Samba server cannot be configured as an Active Directory Primary Domain Controller (PDC) but it can be configured to appear as a Windows NT4-style domain controller. For Windows NT, a domain controller is similar to a Network Information Service (NIS) server in a Linux environment. They both host user and group information databases and other services. Domain controllers are mainly used for security, including the authentication of users accessing domain resources. Authentication services are discussed in the lesson titled "Authentication and Directory Services."

## Accessing Linux Shares from Windows

- Browse using the `\servername\sharename` syntax.
  - `servername` is the Linux Samba server.
- Provide a Windows username and a Samba password.
- The Windows username must map to the Linux username.
  - `/etc/samba/smbuser` maps Linux > Windows usernames.
  - `oracle = wuser`
- Use the `smbpasswd` command to add a Samba password.

```
# smbpasswd -a oracle
```



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

To access a share on a Linux Samba server from Windows, open My Computer or Explorer and enter the host name of the Samba server and the share name in the following format:

`\servername\sharename`

If you enter `\servername`, Windows displays the directories that the Linux system is sharing. You can also map a network drive to a share name by using the same syntax.

### **smbusers File**

For a Windows user to access a Samba share on a Linux system, the user must provide a Windows username and a Samba password. The Windows username must be the same as the Linux username or must map to a Linux username. Samba stores these username maps in the `/etc/samba/smbusers` file. Users with the same username on Linux and Windows do not need an entry in this file, but they still need a Samba password.

The `/etc/samba/smbusers` file has two default entries:

```
root = administrator admin
nobody = guest pcguest smbguest
```

The first entry maps the Linux `root` user to the `administrator` and `admin` users in Windows. The second entry maps the Linux user `nobody` to three Windows usernames.

To map the Windows username of `wuser` to the Linux username of `oracle`, add the following entry to `/etc/samba/smbusers`:

```
oracle = wuser
```

Samba uses Samba passwords, not Linux passwords, to authenticate users. Add a password for the `oracle` user with the following command:

```
# smbpasswd -a oracle
```

New SMB password:

Retype new SMB password:

Added user oracle.

# Accessing Windows Shares from Linux

- Utilities to query Samba servers:
  - findsmb
  - smbtree
- From GNOME and KDE desktops file managers, enter `smb:` in the location bar.
- Use the `smbclient` utility to connect to a Windows share from the command line:
  - `smbclient //<servername>/<sharename> [-U <username>]`
- Use the `mount.cifs` command to mount a Samba share.
- The `mount` command requires the `cifs-utils` package.



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

Use the `findsmb` command to query a subnet for Samba servers. The command displays the IP address, NetBIOS name, workgroup, operating system, and version for each server found.

You can also use the `smbtree` command, which is a text-based SMB network browser. It displays a hierarchy diagram with all the known domains, the servers in those domains, and the shares on the servers.

The GNOME and KDE desktops provide browser-based file managers to view Windows shares on the network. Enter `smb:` in the location bar of the file managers to browse shares.

Use the `smbclient` utility to connect to a Windows share from the command line. The format is as follows:

```
smbclient //<servername>/<sharename> [-U <username>]
```

The `smb:>` prompt is displayed after successfully logging in. Type `help` to display a list of commands. Type `exit` to exit `smbclient`.

To mount Samba shares, install the `cifs-utils` package:

```
# yum install cifs-utils
```

Use the `mount.cifs` command with the following format to mount Samba shares:

```
mount.cifs //<servername>/<sharename> /mount-point -o  
username=<username>,password=<password>
```

## Samba Utilities

Samba packages include several command-line utilities:

- **net**: Works like the `net` utility for Windows and MS-DOS
- **nmblookup**: Resolves NetBIOS names to IP addresses
- **smbstatus**: Displays the status of Samba server connections
- **smbtar**: Backs up and restores Windows-based share files and directories to a local Linux tape archive
- **testparm**: Checks the syntax of the `/etc/samba/smb.conf` file
- **wbinfo**: Displays information from the `winbindd` daemon
- **smbget**: A `wget`-like utility for downloading files over SMB



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

The following list summarizes the command-line utilities included with the Samba packages. Use the `which` utility command to display the absolute path name of the command. Include the output as an argument to the `rpm -qf` command to display which Samba package provides the command.

Example:

```
# which smbtree  
/bin/smbtree  
  
# rpm -qf /bin/smbtree  
samba-client-<version>
```

The Samba command-line utilities include the following:

- **smbtree**: Is a text-based SMB network browser
- **smbclient**: Is an FTP-like client to access SMB/CIFS resources on servers
- **smbpasswd**: Is used to add or modify a user's SMB password
- **smbcacls**: Modifies Windows ACLs on files and directories shared by a Samba server or a Windows server
- **nmblookup**: Is used to query NetBIOS names and map them to IP addresses

- **net:** Is a tool for the administration of Samba and remote CIFS servers. It is designed to work like the `net` utility used for Windows and MS-DOS. The syntax is:

```
net <protocol> [options]
```

The `<protocol>` argument specifies the protocol to use when executing a command. Specify the type of server connection by using `ads` (Active Directory), `rap` (Win9x/NT3), or `rpc` (Windows NT4/2000/2003/2008). If the protocol argument is not specified, `net` automatically tries to identify it. Use `net -h` for online help and usage examples.

- **rpcclient:** Is a tool for executing client-side Microsoft RPCs functions
- **smbcontrol:** Sends control messages to the running `smbd`, `nmbd`, or `winbinddd` daemons
- **smbspool:** Sends a print file to an SMB printer
- **smbstatus:** Displays the status of current connections to a Samba server
- **smbtar:** Backs up and restores Windows-based share files and directories to a local Linux tape archive
- **testparm:** Checks the syntax of the `/etc/samba/smb.conf` file
- **wbinfo:** Displays information from the `winbinddd` daemon (The `winbinddd` daemon must be running.)
- **smbcquotas:** Manipulates quotas on NT file system (NTFS) SMB file shares
- **smbget:** Is a `wget`-like utility for downloading files over SMB



## Quiz

Which of the following statements are true?

- a. Samba allows Linux clients to mount exported file systems on remote Windows systems.
- b. Samba allows Windows clients to mount exported file systems on remote Linux systems.
- c. Samba allows Linux clients to mount exported file systems on remote Linux systems.



ORACLE®

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

## Summary

In this lesson, you should have learned how to:

- Describe the purpose of Samba
- Describe Samba services and daemons
- Configure a Samba server
- Describe Samba server types
- Access Samba shares from a client



ORACLE®

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

## Practice 7: Overview

The practices for this lesson cover the following:

- Configuring a Samba server
- Accessing Samba shares from a Samba client host
- Accessing a Linux Samba share from a Windows system



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.

# Advanced Software Package Management



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

## Objectives

After completing this lesson, you should be able to:

- Describe the contents of an RPM package
- Perform a binary RPM build
- Use the tools to perform package maintenance by using Yum
- Manage Yum cache and Yum history
- Install and use Yum plug-ins
- Describe and use the programs offered by PackageKit



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

# Software Management with RPM and Yum

- Red Hat Package Manager (RPM) is a software package management system.
- The RPM toolset includes:
  - The `rpm` command, which is also called the RPM Package Manager
  - Additional utilities such as `rpmquery`, `rpminfo`, and `rpmbuild`
  - The RPM database
  - The `.rpm` package format
- Use the `yum` utility for RPM-based package maintenance:
  - To resolve dependencies during installation, upgrade, and removal
  - To access and query local or remote RPM-based repositories



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

## RPM Package Management

You can use the RPM package management system for packaging software in the Linux environment and manage these packages. The `rpm` command is the most widely used command for RPM package management.

Using the `rpm` command, you can:

- Install, upgrade, and remove packages
- List and query installed packages. For example, use the `rpm -ql <package name>` command to list the files that make up a package
- Manage the RPM database

In addition to the `rpm` command, you can utilize additional commands to manage your RPM-based software. Three of these commands are listed with an example for each command:

- The `rpminfo` command provides information about installed packages. The following example lists executable (-e) files included in the `bash` package:

```
# rpminfo -e bash
bash-4.2.46-12.el7.x86_64
/usr/bin/bash          PIC
```

- The `rpmbuild` command builds source or binary RPM packages. The following example builds a binary package (`-bb`) in verbose (`-v`) mode:

```
# rpmbuild -bb -v myspecfile.spec
```

- The `rpmquery` command displays information about packages, the RPM database or other package-related items. The following example displays files in a package:

```
$ rpmquery -l xorg-x11-server-Xorg
/etc/X11/xorg.conf.d
/etc/pam.d/xserver
/etc/security/console.apps/xserver
/usr/bin/X
...
/usr/share/man/man5/xorg.conf.5.gz
/usr/share/man/man5/xorg.conf.d.5.gz
```

Note that the following `rpm` command produces the same output as the preceding one:

```
$ rpm -ql xorg-x11-server-Xorg
/etc/X11/xorg.conf.d
/etc/pam.d/xserver
/etc/security/console.apps/xserver
/usr/bin/X
...
/usr/share/man/man5/xorg.conf.5.gz
/usr/share/man/man5/xorg.conf.d.5.gz
```

## **Yum Package Management**

Yum is also a package management system. It offers more functionality than is available with RPM-based tools such as the `rpm` command. Whether you use the `rpm` command or the `yum` command, the packages are the same. They are built by using the RPM format and are distributed in files with the `.rpm` identifier. Yum functionality is discussed later in this lesson.

## RPM Packages

- An RPM binary package file (for example, `bash-4.1.2-15.el6_4.x86_64.rpm`) contains programs, configuration files, documentation.

```
# yum install bash
Loaded plugins: aliases, langpacks
Setting up Install Process
Package bash-4.2.46-12.el7.x86_64 already installed ...
Nothing to do
```

- An RPM source package file (for example, `bash-4.2.46-12.el7.src.rpm`) contains the source code and all necessary files to re-create the binary package file.

```
# yumdownloader --source bash
Loaded plugins: aliases, langpacks
bash-4.2.46-12.el7.src.rpm
...
```



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

# The Binary RPM Build Process

1. As the `root` user, install the RPM tools package called `rpmdevtools`.
2. As a user other than `root`:
  1. Create a directory structure for the build.
  2. Add the source files to a compressed (`.tar.gz`) file and store them in the `SOURCES` directory.
  3. Create the `spec` file.
  4. Build the binary package by using the `rpmbuild -bb` command.
3. As the `root` user, install the package and verify the installation.
4. Optional: Upload the package to a repository.

ORACLE®

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

The steps to build a binary RPM package are shown in the slide. Install the `rpmdevtools` package, which installs the `rpm-build` package as a dependency. These two packages provide the basic tools necessary to build binary and source RPM packages. These tools include the following utilities:

- `rpmdev-setuptree`: Creates the directory structure for the package build
- `rpmdev-newspec`: Creates a skeleton `spec` file
- `rpmbuild`: Builds the binary RPM (also used to create source RPM packages)

Other tools are installed as well. Use the following commands to list all the executables that are installed as part of the `rpmdevtools` package:

```
# rpm -q1 rpmdevtools | grep bin
/usr/bin/annotate-output
...
/usr/bin/rpmdev-newspec
...
/usr/bin/rpmdev-setuptree
...
```

## BUILD Directory Structure

- Use the `rpmdev-setuptree` command to build the directory structure for the RPM build process.
- This command creates the following directories:
  - BUILD
  - RPMS
  - SOURCES
  - SPECS
  - SRPMS



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

The directories listed in the slide are shown in alphabetical order.

In the following notes, the directories are listed as they are used during the binary (or source) RPM build process:

### SPECS

This is the directory where you store the `spec` file for building a source or binary package.

### SOURCES

This is the location where you store the files to build your package. The files can be source code or binary files.

### BUILD

The files in the SOURCES directory are copied or extracted to this directory before building the software. This directory is used as temporary space to compile the software.

### RPMS

The output of the RPM build process is an RPM package. If the build is a binary build, this package is stored in the RPMS directory.

### SRPMS

This is the same as the RPMS directory except that it is used to store source RPMs.

## spec File to Build a Binary RPM Package

- The `spec` file describes the package and lists the steps to build the software in the package.
- It contains the following sections:
  - Header: Describes the package with a collection of tags
  - `%prep`: Prepares files for the build
  - `%build`: Builds the software
  - `%install`: Copies files to their installation location
  - `%clean`: Cleans the build directory tree
  - `%files`: Identifies the files to be packaged



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

### Header

This section describes the package by using tags and directives.

For example, the following tags describe a package: Name, Arch, Version, Release, URL, and License. The `%description` directive describes the usage for the package. After the package is built, the tags and directive information become part of the package.

You can display the tags and other information in an existing package by using the `yum info <package name>` command:

```
# yum info bash
Loaded plugins: aliases, langpacks
Installed Packages
Name        : bash
Arch       : x86_64
Version    : 4.2.46
Release   : 12.el7
...
Description : The GNU Bourne Again shell (Bash) is a shell ...
...
```

Each section following the header information is a step in the build process. Any command, script, macro or directive in a section is executed like a script when that step is executed.

Before each step is executed, several environment variables are set. For example, the value for the `RPM_BUILD_DIR` variable is set.

#### **%prep**

During the execution of this step, the source files are unpacked into the build directory. If present, patches are applied. The `%setup` macro is responsible for unpacking the source files.

#### **%build**

There are no special macros for this section. Generally, you specify one or more `make` commands in this section to build the software.

#### **%install**

The role of this section is to install the newly built software. This means that the files that make up the package along with their directory location are copied into a directory structure. After the files are copied, the build reads the list of files in the `%files` section and creates the binary RPM package.

#### **%clean**

At this step, the packaging is already done. This macro cleans the directory tree where the software is installed (default is variable `RPM_BUILD_ROOT`) or any directory specified in this section.

#### **%files**

This section lists the files that are to be part of the final RPM. In this section, you can use macros to set file and directory permissions.

As stated in the description of the sections in the `spec` file, you can use macros and directives to perform specific steps in each section.

Example:

- `%setup` macro in the `%prep` section unpacks the source files.
- `%config` directive in the `%files` section labels files as configuration files.

You can find more information about the tags, macros, and directives in the RPM-based `spec` file at this location: <http://ftp.rpm.org/max-rpm/>.

## spec File: Example

```
Name:          hello
Version:       1.0
Release:       1%{?dist}
Summary:       hello program

License:       GPL
#URL:
Source0:       hello-1.0.tar.gz

#BuildRequires:
#Requires:

%description
A program to display Hello World

%prep
%setup -q
```

```
%build

%install
rm -rf $RPM_BUILD_ROOT
#%make_install
install -d      $RPM_BUILD_ROOT/usr/local/bin
install hello
               $RPM_BUILD_ROOT/usr/local/bin/hello

%files
/usr/local/bin/hello

%changelog
```



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

Use the `rpmdev-newspec` command to create a skeleton `spec` file. The following example creates the `hello.spec` file in the `SPECS` directory:

```
# rpmdev-newspec SPECS/hello.spec
```

From the contents of the `hello.spec` file shown in the slide, you can gather the following information:

- Both the package and the program in the package are called `hello`.
- This is version 1.0 of the software being packaged (`Version: 1.0`).
- This is version 1 of the package itself with the distribution appended (`Release: 1.el7.x86_64`).
- There is no step for the `%build` section. The build process goes from the `%prep` section to the `%install` section without any build command or script. Generally, this section contains build instructions.
- There is only one file in the final package, the `hello` program, which is installed in the `/usr/local/bin` directory when the package is installed.

# Managing RPM-Based Software with Yum

- Using Yum greatly simplifies package maintenance in your Linux environment.
- Yum includes:
  - The `yum` command
  - Several utilities such as `yum-config-manager`, `repoquery`, and `yumdownloader`
  - The ability to create, query, and control access to repositories
  - Plug-ins that extend Yum's functionality
  - Caching to increase performance for Yum operations



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

Yum tools, including the `yum` command, provide more services and functionality than is available with the `rpm` command and other RPM-based tools.

With Yum tools and plug-ins, you can:

- List software packages, both installed and available, in local or remote repositories
- Check for package dependencies (packages required to install a package)
- Check for dependent software (packages that depend on another package)
- Create new repositories and enable or disable access to existing repositories
- Speed up package installation by using cached information (Yum cache)
- Extend Yum's functionality with plug-ins such as the `aliases` plug-in (to create and view aliases for Yum commands)
- Use package management GUI tools such as PackageKit. PackageKit uses Yum tools.

PackageKit is discussed later in this lesson.

Note that when creating packages, either binary or source RPMs, you use RPM-type tools such as `rpmbuild` and `rpmdev-setuptree`. These commands were discussed in the RPM Packages topic earlier in this lesson.

## Yum Cache

- yum stores temporary files in the /var/cache/yum directory.
- Temporary package files are deleted after a yum operation completes successfully.
- You can enable caching to retain package files in cache directories:
  - These packages can be reused when there is no network connection to repositories.
- Clean information in the cache with:

```
# yum clean metadata  
# yum clean headers  
# yum clean packages  
# yum clean all
```



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

For some operations (for example, a yum install operation), Yum downloads the packages to install into the Yum cache. The cached packages are located in a subdirectory structure from /var/cache/yum that reflects the architecture, the distribution release, and the repository from where the packages were downloaded.

After successful installation, the packages are deleted from the cache. To retain the cached packages, change the keepcache setting to 1 in the /etc/yum.conf file as follows:

```
keepcache = 1
```

You can also change the location for the cache by modifying the cachedir parameter, which by default is set to:

```
cachedir=/var/cache/yum/$basearch/$releasever
```

### Cleaning the Yum Cache

Clean the Yum cache to reclaim disk space or to clear errors due to corrupted metadata files.

To remove cached packages only, use:

```
# yum clean packages
```

To delete metadata for each enabled repository, use the following command:

```
# yum clean metadata
```

To delete package headers, use the following command:

```
# yum clean headers
```

To clean all cached information, use the following command:

```
# yum clean all
```

If you get the message “Metadata file does not match checksum” during a Yum operation, clearing the metadata from the cache might not help. In this case, adding the following line to /etc/yum.conf resolves the problem:

```
http_caching=none
```

## Yum History

- Yum keeps detailed information about transactions in Yum history.
  - Each transaction is assigned an ID.
- Yum history is stored in /var/lib/yum/history/.
- To display the transactions:

```
# yum history list
# yum history info <transaction ID>
# yum history package-list <package name>
```

- To undo a transaction:

```
# yum history undo <transaction ID|last>
```

- To start a new history db:

```
# yum history new
```



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

The following command displays the last 20 transactions in Yum history:

```
# yum history list
Loaded plugins: aliases, langpacks
ID      | Login user          | Date and time       | Action(s)      | Altered
-----
 7 | root <root>        | 2014-01-28 03:13 | Erase          | 2
 6 | root <root>        | 2014-01-28 03:03 | Install         | 2
 5 | System <unset>     | 2014-01-27 09:31 | Update          | 1 <
 4 | root <root>        | 2014-01-27 08:10 | Install         | 4 >
 3 | root <root>        | 2014-01-27 07:46 | Install         | 1
 2 | root <root>        | 2014-01-27 07:45 | Update          | 2
 1 | System <unset>     | 2014-01-27 06:57 | Install         | 1131
history list
```

To obtain detailed information for transaction ID 6, which installed the `rpmdevtools` package:

```
# yum history info 6
Loaded plugins: aliases, langpacks
Transaction ID : 6
Begin time      : ...
...
Return-Code     : Success
Command Line    : install rpmdevtools
Transaction performed with:
...
```

In this example, transaction ID 6 installed the `rpmdevtools` package. Transaction ID 7 (not shown) uninstalled the `rpmdevtools` package. The following command performs transaction ID 6 again, which installs `rpmdevtools`:

```
# yum history redo 6
Loaded plugins: aliases, langpacks
...
Repeating transaction 6...
...
Installing:
  rpmdevtools ...
...
```

## Extending Yum Functionality with Plug-Ins

- Yum uses plug-ins to extend its functionality.
- Yum plug-ins are installed as packages.
- Installed Yum plug-ins reside in /usr/lib/yum-plugins.
- Each plug-in has an associated configuration file in /etc/yum/pluginconf.d.
  - To enable a plug-in, edit the configuration file and set enable=1.
  - To disable a plug-in, edit the configuration file and set enable=0.
- Use the --disableplugin=<plugin\_name> option to the yum command to disable a plug-in for a single command.
  - The following example disables the aliases plug-in when running the yum update command:

```
# yum update --disableplugin=aliases
```



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

Yum uses plug-ins to extend its functionality.

Each installed plug-in has a configuration file located in the /etc/yum/pluginconf.d directory. For example, the configuration file for the langpacks plug-in contains:

```
# cat /etc/yum/pluginconf.d/langpacks.conf
[main]
enabled=1
langpack_locales = en_US.UTF-8
```

By default, all plug-ins are enabled in /etc/yum.conf by using the following statement:

```
plugins=1
```

Do not disable plug-ins globally from /etc/yum.conf by changing plugins=1 to 0. This action can cause problems with some Yum services. If you want to disable a plug-in without uninstalling it, you can:

- Disable the plug-in from its configuration file in /etc/yum/pluginconf.d
- Disable the plug-in for a single operation by appending --disableplugin=<plug-in name> to the yum command

Yum plug-ins are Python scripts or programs that are stored in `/usr/lib/yum-plugins` when they are installed:

```
# ls -l /usr/lib/yum-plugins
-rw-r--r-- ... langpacks.py
-rw-r--r-- ... langpacks.pyc
-rw-r--r-- ... langpacks.pyo
```

Each plug-in in this example has three files associated with it: a `.py`, `.pyc`, and `.pyo` file. These files are part of the Python application. For more information about these Python file extensions, see <http://docs.python.org/release/1.5.1p1/tut/node43.html>.

## Popular Yum Plug-Ins

- **kabi:** Checks if newly installed kernel module packages conform with the kernel Application Binary Interface (kABI)
- **aliases:** Allows you to create and view aliases for Yum commands. It includes the `/etc/yum/aliases.conf` file, which contains many predefined Yum command aliases.
- **changelog:** Allows you to view package change logs before and after updating
- **tmprepo:** Allows you to use a temporary Yum repository. It ensures that these repositories are safe and does not allow GPG checking to be disabled.



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

### **kabi**

The package name is `kabi-yum-plugin`. This plug-in checks the newly installed kernel module packages to ensure that they conform with kABI.

### **aliases**

The package name is `yum-plugin-aliases`. This plug-in allows you to create and view aliases for Yum commands. This plug-in includes the `/etc/yum/aliases.conf` file, which contains many predefined Yum command aliases.

### **changelog**

The package name is `yum-plugin-changelog`. This plug-in allows you to view package change logs before and after updating. Yum invokes the plug-in if either the `--changelog` option or the `changelog` command is used with the `yum` command.

### **tmprepo**

The package name is `yum-plugin-tmprepo`. This plug-in provides the `--tmprepo <URL:repo_file>` option to the `yum` command. The repository file is downloaded from the URL and enabled for a single Yum transaction. This plug-in attempts to ensure that temporary repositories are safe to use. The plug-in does not allow GNU Privacy Guard (GPG) checking to be disabled by default.

## Using yum-config-manager

Yum repositories and configurations are managed by either manual editing or using the `yum-config-manager` program.

- Install the `yum-utils` package to use `yum-config-manager`:

```
# yum install yum-utils
```

- After installing, run the `man` command for verifying installation of `yum-config-manager` and for reviewing command options:

```
# man yum-config-manager
```

- For adding a remote repository, use the command form:

```
# yum-config-manager --add-repo repository_url
```

- For example, to add the repo `ol7_software_collections.repo`, run the command:

```
# yum-config-manager --add-repo ol7_software_collections
```



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

After adding the `ol7_software_collections` repo, a new file, `ol7_software_collections.repo`, is added to `/etc/yum.repos.d/`. The repo configurations for `ol7_software_collections.repo` are as follows:

```
[ol7_software_collections]
Name=added from: ol7_software_collections
baseurl=ol7_software_collections
enabled=1
```

After this repo is added and enabled, you can install software collections such as PHP, Python, Java, and Ruby.

## Managing a Yum Repo

You can enable and disable yum repositories with `yum-config-manager` for managing access of a repo.

- Use the following command template to enable a yum repo:

```
# yum-config-manager --enable repo_name
```

- To enable the `ol7_latest` repo, run the following command:

```
# yum-config-manager --enable ol7_latest
```

- Use the following command template to disable a yum repo:

```
# yum-config-manager --disable repo_name
```

- To disable the `ol7_latest` repo, run the following command:

```
# yum-config-manager --disable ol7_latest
```



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

After enabling a repo, the repo's enabled value equals 1.

```
[ol7_latest]
...
enabled=1
```

When a repo is disabled, the repo's enabled value equals 0.

```
[ol7_latest]
...
enabled=0
```

# Managing Errata

- To list all of the available errata:

```
# yum updateinfo list
```

- To filter errata information:
  - By security priority (critical, important, moderate)

```
# yum updateinfo list --sec-severity=Important
```

- By erratum

```
# yum updateinfo --advisory ELSA-2014-0097
```

- By CVE

```
# yum updateinfo info --cve CVE-2013-5896
```



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

Use the `yum updateinfo` command to view information about errata (security, bug fixes, and enhancements), but also to update your Linux system by using options that act like filters to select only certain updates from the errata. In previous versions of Oracle Linux, this capability was provided by the `yum-security` plug-in. Oracle Linux 7 includes an updated version of Yum and the `yum-security` plug-in has been integrated into `yum`.

## Obtaining Errata Information

To list all of the available errata:

```
# yum updateinfo list
```

...

ELSA-2015-0672 Moderate/Sec. bind-libs-32:9.9.4-18.el7\_1.1...

...

ELBA-2015-0741 bugfix binutils-2.23.52.0.1-30.el7\_1.1...

...

ELEA-2015-0969 enhancement crash-7.0.9-5.el7\_1.x86\_64

...

ELSA-2015-0265 Critical/Sec. firefox-31.5.0-2.0.1.el7\_0...

...

This list contains all of the errata by errata ID. Errata include security patches, bug fixes, and feature enhancements. Security fixes are listed by priority: Critical, important, or moderate.

To obtain only the security errata with priority set to Important:

```
# yum updateinfo list --sec-severity=Important
ELSA-2014-0097 Important/Sec. java-1.6.0-openjdk-1:1.6.0.0-3...
ELSA-2014-0097 Important/Sec. java-1.6.0-openjdk-devel-1:1.6...
ELSA-2013-1801 Important/Sec. kernel-2.6.32-431.1.2.el6.x86_64
ELSA-2013-1801 Important/Sec. kernel-2.6.32-431.5.1.el6.x86_64
...
...
```

To list detailed information for a particular erratum from the previous list:

```
# yum updateinfo info --advisory ELSA-2014-0097
=====
java-1.6.0-openjdk security update
=====
Update ID : ELSA-2014-0097
Release : Oracle Linux 6
Type : security
Status : final
Issued : 2014-01-27
CVEs : CVE-2013-5878
      : CVE-2013-5884
      ...
      : CVE-2014-0423
      : CVE-2014-0428
Description : [1:1.6.0.1-3.1.13.0]
      : - updated to icedtea 1.13.1
      : -
      : http://blog.fuseyism.com/index.php/2014/01/...
      ...
Severity : Important
updateinfo info done
```

To obtain information for a particular Common Vulnerabilities and Exposures (CVE):

```
# yum updateinfo info --cve CVE-2013-5896
```

To update all packages for which security errata exist to the latest version of the packages:

```
# yum --security update
```

To update all packages for which security errata exist to the version that contains the security fix, ignoring newer releases of the packages:

```
# yum --security update-minimal
```

## Important Resources for Errata Information

- Refer to the following for errata listings:
  - <https://linux.oracle.com/errata>
- Refer to the following for CVE listings:
  - <https://linux.oracle.com/cve>
- Finding important errata and CVE information on ULN:
  - <https://blogs.oracle.com/linux/finding-important-errata-and-cve-information-on-uln>
- Updates to errata on ULN and yum.oracle.com:
  - [https://blogs.oracle.com/linux/entry/updates\\_to\\_errata\\_on\\_uln](https://blogs.oracle.com/linux/entry/updates_to_errata_on_uln)



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

Refer to the following for errata listings: <https://linux.oracle.com/errata>.

Using this link, you can view all errata releases available, listed by type, severity, advisory, summary, and release date. In addition, you are also able to filter this list by release and/or type (Bug, Security, or Enhancement).

Refer to the following for CVE listings: <https://linux.oracle.com/cve>

Using this link, you can find information about security errata by CVE identifier.

### Read These Blogs

The following blog discusses finding important errata and CVE information on ULN:

<https://blogs.oracle.com/linux/finding-important-errata-and-cve-information-on-uln>

The following blog discusses updates to errata on ULN and yum.oracle.com:

[https://blogs.oracle.com/linux/entry/updates\\_to\\_errata\\_on\\_uln](https://blogs.oracle.com/linux/entry/updates_to_errata_on_uln)

## PackageKit Software Package Manager GUI

- PackageKit is designed to facilitate package management.
- It interfaces with a GUI front end. Examples:
  - GNOME (gnome-packagekit)
  - KDE (KPackageKit)
- The package operations are carried out by a back-end package management:
  - Yum (for Oracle Linux and Red Hat)
  - APT for Debian systems



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

The PackageKit program provides a graphical interface for package management systems for different Linux distributions. PackageKit for Oracle Linux and Red Hat distributions uses a Yum back end to perform operations on packages, such as installing, updating, and removing packages. The front-end graphical interface is GNOME, and this support is provided by the `gnome-packagekit` package and its dependencies.

# Using PackageKit Software Update

Launch Software Update through one of the following methods:

- Select Software Update under Applications > System Tools.
- Run `gpk-update-viewer` from the command line.



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

PackageKit provides a graphical tool to add or remove software and another tool to update your system.

For Oracle Linux users, installing updates with the PackageKit program is equivalent to installing fixes for bugs and/or security issues or enhancements announced in the errata and made available in the \_latest repository for your particular Oracle Linux release. Only the updates for the currently installed software packages are shown.

You can use the Software Update and the Software programs as a nonprivileged user. When requesting to add/update/remove packages, you are prompted for the `root` password.

## PackageKit Commands: Summary

- Commands to launch PackageKit graphical programs:
  - `gpk-application` → Add/Remove Software
  - `gpk-log` → Software Log Viewer
  - `gpk-prefs` → Software Update Preferences
  - `gpk-update-viewer` → Software Update Viewer
- Commands that interface with PackageKit:
  - `pk12util` → Utility to import and export keys and certificates
  - `pkaction` → Command to receive details of a registered action
  - `pkcheck` → Command to check authorization of a process
  - `pkcon` → CLI client for PackageKit
  - `pkcs1-conv` → Command for private and public RSA keys format conversion



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

The `gpk-update-viewer` launches the Software Update graphical program as discussed in the previous slide. Similarly, the `gpk-application` program launches the Software graphical program to install and remove packages. The `gpk-log` command allows you to view the software log. The `gpk-prefs` command allows you to change preferences for keeping your system up to date.

Following is a list of all the available “`gpk`” commands:

```
# ls /bin/gpk*
/bin/gpk-application  /bin/gpk-log    /bin/gpk-prefs  /bin/gpk-update-viewer
```

There are man pages for each of the “`gpk`” commands.

Similarly, you can view the available “`pk`” commands.

```
# ls /bin/pk*
/bin/pk12util      /bin/pkaction     /bin/pkcheck     /bin/pkcon      /bin/pkcs1-
conv
...
...
```



## Quiz

Which statements are true about Yum plug-ins?

- a. Yum plug-ins extend the functionality of Yum.
- b. New plug-ins are automatically included in the latest release of the Yum package.
- c. Each plug-in has its own configuration file.
- d. By default, all Yum plug-ins are enabled in /etc/yum.conf.



ORACLE®

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



## Quiz

Which of the following commands can be used to clean information in the Yum cache?

- a. yum clean metadata
- b. yum clean cache
- c. yum clean packages
- d. yum delete cache



ORACLE®

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

## Summary

In this lesson, you should have learned how to:

- Describe the contents of an RPM package
- Perform a binary RPM build
- Use the tools to perform package maintenance by using Yum
- Manage Yum cache and Yum history
- Install and use Yum plug-ins
- Describe and use the programs offered by PackageKit



ORACLE®

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

## Practice 8: Overview

The practices for this lesson cover the following:

- Exploring the host04 VM
- Managing Yum plug-ins
- Using Yum utilities to manage errata and to download software
- Creating a binary RPM package
- Managing software updates with PackageKit
- Working with Yum history and Yum cache



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

# Advanced Networking



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

## Objectives

After completing this lesson, you should be able to:

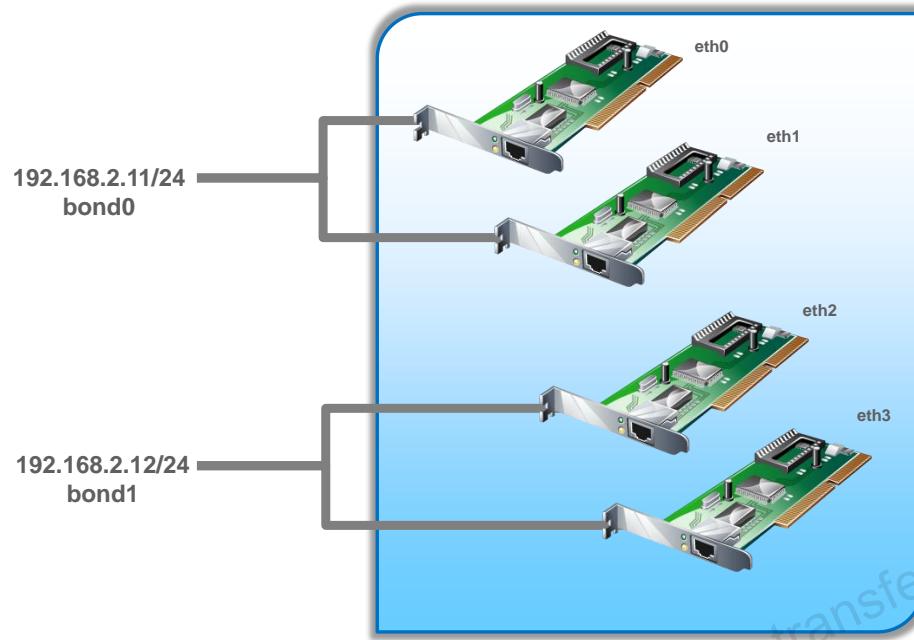
- Describe Network Bonding
- Configure Network Bonding
- Describe Virtual Local Area Networks (VLANs)
- Configure a VLAN
- Describe Virtual Private Networks (VPNs)
- Configure a Site-to-Site VPN
- Describe Multi-Factor Authentication
- Describe Internet Protocol Security (IPSec)
- Describe Network Analysis
- Describe Network Performance Tuning



ORACLE®

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

## Network Bonding: Introduction



ORACLE®

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

Network interface bonding is called by many names: Port Trunking, Channel Bonding, Link Aggregation, NIC teaming, and others. It combines or aggregates multiple network connections into a single channel bonding interface. This allows two or more network interfaces to act as one, to increase throughput and to provide redundancy or failover.

The Linux kernel comes with the bonding driver for aggregating multiple physical network interfaces into a single logical interface (for example, aggregating `eth0` and `eth1` into `bond0`). For each bonded interface, you can define the mode and the link monitoring options. There are seven different mode options, each providing specific load balancing and fault tolerance characteristics.

The slide shows four network interface cards (NICs): `eth0`, `eth1`, `eth2`, and `eth3`. The `eth0` and `eth1` NICs are combined into a single `bond0` interface with an IP address of `192.168.2.11/24`. The `eth2` and `eth3` NICs are combined into a single `bond1` interface with an IP address of `192.168.2.12/24`.

# Network Bonding: Configuration

- To configure network bonding:
  - Manually create a bonding interface file in the `/etc/sysconfig/network-scripts/` directory
  - Or use the `nm-connection-editor` GUI
  - Or use the `nmtui` utility
  - Or use the `nmcli` utility
- The physical network interfaces included in the bonding interface are called “slaves.”
  - Each “slave” also has an interface file in the `/etc/sysconfig/network-scripts/` directory.
  - The configuration file includes `MASTER` and `SLAVE` directives.



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

## Creating a Bonding Interface File

You can manually create a bonding interface file in the `/etc/sysconfig/network-scripts` directory. You can also use the `nm-connection-editor` GUI, the `nmtui` text user interface, or use the `nmcli` command-line interface. You first create the bonding interface, and then you add the physical network interfaces to the bond. These physical network interfaces are called “slaves.”

For example, in the previous slide, the slaves for the `bond0` interface are `eth0` and `eth1`. The slaves for `bond1` are `eth2` and `eth3`.

## Creating Slave Interface Files

The “slaves” also have an associated file in the `/etc/sysconfig/network-scripts` directory. These files identify the bond by using the `MASTER` directive, and the slaves by using the `SLAVE` directive.

Examples of configuration files in the `/etc/sysconfig/network-scripts` directory are shown on the following page.

The following is an example of a bonding interface file:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
BONDING_OPTS="miimon=1 updelay=0 downdelay=0 mode=active-backup"
TYPE=Bond
BONDING_MASTER=yes
BOOTPROTO=none
IPADDR=192.168.2.12
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
...
NAME=bond0
UUID=...
ONBOOT=yes
```

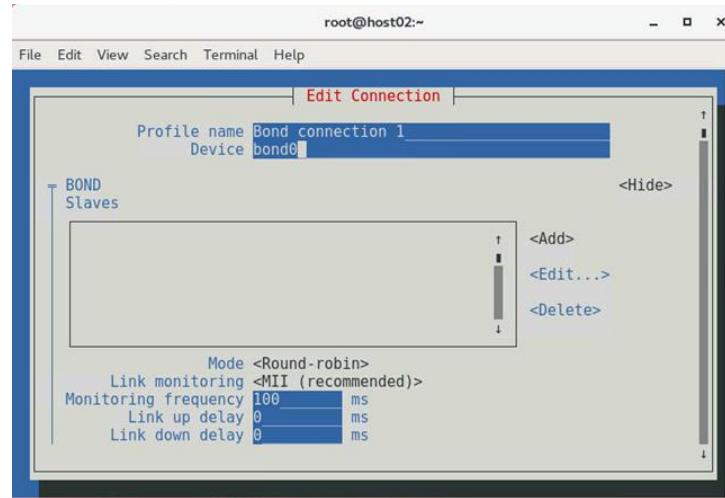
The following example defines the eth0 physical network interface as a slave for bond0:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth0
TYPE=Ethernet
NAME=bond-slave-eth0
UUID=...
DEVICE=eth0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

The following example defines the eth1 physical network interface as a slave for bond0:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth1
TYPE=Ethernet
NAME=bond-slave-eth1
UUID=...
DEVICE=eth1
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

# Using the NetworkManager TUI to Configure Network Bonding



ORACLE®

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

The slide shows the NetworkManager TUI used to configure a network bonding connection. You can access this text user interface by using the following command:

```
# nmcli connection edit
```

Select "Edit a connection" and click "OK". Use the right arrow to highlight "Add" and hit return. You are prompted to choose a connection type. Select "Bond" as the connection type to display the GUI shown in the slide.

From the "Bond" tab as shown in the slide, you can do the following:

- Provide a name for the bonded interface, which defaults to `bond0`
- Click "Add" to add the physical network interfaces, which are known as "slaves"
- Configure the Mode, which defaults to "Round-robin"
- Configure Link Monitoring, which defaults to MII (Media Independent Interface)
- Configure the Monitoring frequency, Link up delay, and Link down delay

Click "Save" and the interface files are created in `/etc/sysconfig/network-scripts/`.

## Network Bonding Modes

The following bonding policy modes are available:

- **Round-robin:** This is the default mode. Network traffic occurs on each bonded slave interface in sequential order.
- **Active backup:** Only one slave in the bond is active at a time.
- **XOR:** This mode works best for traffic on the same link.
- **Broadcast:** All network transmissions are sent on all slaves.
- **802.3ad:** This uses a dynamic link aggregation policy.
- **Adaptive transmit load balancing:** Outgoing network traffic is distributed according to the current load on each slave.
- **Adaptive load balancing:** It provides both transmit load balancing and receive load balancing for IPv4 traffic.

See `/usr/share/doc/iputils-*/README.bonding`.



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

The following bonding policy modes are available:

- **Round-robin:** This is the default mode. Network transmissions are in sequential order beginning with the first available slave. This mode provides load balancing and fault tolerance.
- **Active backup:** Only one slave in the bond is active. Another slave interface becomes active if the active slave interface fails. The bond's MAC address is externally visible on only one network adapter to avoid confusing a network switch. This mode provides fault tolerance.
- **XOR (exclusive-or):** Network transmissions are based on a transmit hash policy. The default policy derives the hash by using MAC addresses. In this mode, network transmission destined for specific peers are always sent over the same slave interface. This mode works best for traffic to peers on the same link or local network. This mode provides load balancing and fault tolerance.
- **Broadcast:** All network transmissions are sent on all slave interfaces. This mode provides fault tolerance.
- **802.3ad:** This uses an IEEE 802.3ad dynamic link aggregation policy. Aggregation groups share the same speed and duplex settings. This mode transmits and receives network traffic on all slaves in the active aggregator. This mode requires an 802.3ad-compliant network switch.

- **Adaptive transmit load balancing (TLB):** Outgoing network traffic is distributed according to the current load on each slave interface. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed slave. This mode does not require any special switch support.
- **Adaptive load balancing (ALB):** This mode includes transmit load balancing (tlb) and receive load balancing (rlb) for IPv4 traffic and does not require any special switch support. Receive load balancing is achieved by ARP negotiation.

See the `/usr/share/doc/iputils-*/README.bonding` file for complete descriptions of the available bonding policy modes. This information is also available at <http://www.kernel.org/doc/Documentation/networking/bonding.txt>.

# Network Bonding Link Monitoring

The bonding driver supports two methods to monitor a slave's link state:

- MII (Media Independent Interface) monitor
  - This is the default, and recommended, link monitoring option.
  - It monitors the carrier state of the local network interface.
  - You can specify the monitoring frequency and the delay.
  - Delay times allow you to account for switch initialization.
- ARP monitor
  - This sends ARP queries to peer systems on the network and uses the response as an indication that the link is up.
  - You can specify the monitoring frequency and target addresses.



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

## MII (Media Independent Interface) Monitor

This is the default link monitoring option. This method monitors only the carrier state of the local network interface. It relies on the device driver for carrier state information, or queries the MII registers directly, or uses `ethtool` to attempt to obtain carrier state. You can specify the following information for MII monitoring:

- **Monitoring frequency:** The time in milliseconds between querying carrier state
- **Link up delay:** The time in milliseconds to wait before using a link that is up
- **Link down delay:** The time in milliseconds to wait before switching to another link when the active link is reported as down

## ARP Monitor

This method of link monitoring sends APR queries to peer systems on the network and uses the response as an indication that the link is up. The ARP monitor relies on the device driver to keep the last receive time, and the transmit start time, updated. If the device driver is not updating these times, the ARP monitor fails any slaves that use that device driver. You can specify the following information for APR monitoring:

- **Monitoring frequency:** The time in milliseconds that ARP queries are sent
- **ARP targets:** A comma-separated list of IP addresses that ARP queries are sent to

## Using the `nmcli` Utility to Configure Network Bonding

- The following command creates a bonded interface named `bond0`, defines the interface as `bond0`, sets the mode to “active-backup”, and assigns an IP address to the bond:

```
# nmcli con add type bond con-name bond0 ifname bond0 mode active-backup  
ip4 192.168.2.12/24
```

- The `nmcli con` command shows the new bond connection:

```
# nmcli con  
NAME      UUID           TYPE      DEVICE  
eth0      ...            ethernet  eth0  
eth1      ...            ethernet  eth1  
bond0     ...            bond      bond0
```

- An `ifcfg-bond0` file is created in the `/etc/sysconfig/network-scripts` directory.



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

Use the `nmcli connection` command without any arguments to view the existing network connections. You can also shorten the “connection” argument to “con” or a shorter abbreviation. Example:

```
# nmcli con  
NAME      UUID           TYPE      DEVICE  
eth0      ...            ethernet  eth0  
eth1      ...            ethernet  eth1
```

Include the “add type bond” arguments, and any additional information to create a network bond connection. The following example creates a bonded interface named `bond0`, defines the interface as `bond0`, sets the mode to “active-backup”, and assigns an IP address to the bonded interface.

```
# nmcli con add type bond con-name bond0 ifname bond0 mode active-backup  
ip4 192.168.2.12/24
```

The `nmcli con` command shows the new bond connection.

```
# nmcli con  
NAME      UUID           TYPE      DEVICE  
bond0     ...            bond      bond0
```

The nmcli con add type bond command creates an interface configuration file in the /etc/sysconfig/network-scripts directory. For example:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
BONDING_OPTS=mode=active-backup
TYPE=Bond
BONDING_MASTER=yes
BOOTPROTO=none
IPADDR=192.168.2.12
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=bond0
UUID=...
ONBOOT=yes
```

The ip addr command shows the new bond0 interface:

```
# ip addr
...
6: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.2.12/24 brd 192.168.2.255 scope global noprefixroute
            bond0
...
...
```

## Using the nmcli Utility to Add the Slaves to the Bond

- The following commands add the eth2 interface and the eth3 interface as a "bond-slave" type connection for bond0:

```
# nmcli con add type bond-slave ifname eth2 master bond0
# nmcli con add type bond-slave ifname eth3 master bond0
```

- The nmcli con command shows the new connections:

NAME	UUID	TYPE	DEVICE
bond-slave-eth3	...	ethernet	eth3
bond-slave-eth2	...	ethernet	eth2
bond0	...	bond	bond0

- Interface configuration files are created for the slaves.

```
# ls /etc/sysconfig/network-scripts/*slave*
ifcfg-bond-slave-eth2
ifcfg-bond-slave-eth3
```

ORACLE®

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

For each interface that you want to bond, use the nmcli con add type bond-slave command. The following example adds the eth2 interface as a bond slave. The command does not include the con-name argument so a name is automatically generated.

```
# nmcli con add type bond-slave ifname eth2 master bond0
Connection 'bond-slave-eth2' (<UUID>) successfully added.
```

The following example adds the eth3 interface as a "bond-slave."

```
# nmcli con add type bond-slave ifname eth3 master bond0
Connection 'bond-slave-eth3' (<UUID>) successfully added.
```

The nmcli con command shows the new connections.

NAME	UUID	TYPE	DEVICE
bond-slave-eth3	...	ethernet	eth3
bond-slave-eth2	...	ethernet	eth2
bond0	...	bond	bond0
eth0	...	ethernet	eth0
eth1	...	ethernet	eth1
eth2	...	ethernet	--
eth3	...	ethernet	--

The nmcli con add type bond-slave commands create interface configuration files in the /etc/sysconfig/network-scripts directory. For example:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth2
TYPE=Ethernet
NAME=bond-slave-eth2
UUID=...
DEVICE=eth2
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

The “bond-slave” configuration file for the eth3 interface is shown:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond-slave-eth3
TYPE=Ethernet
NAME=bond-slave-eth3
UUID=...
DEVICE=eth3
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

The ip addr command includes “SLAVE” for the eth2 and eth3 interfaces and also includes “master bond0”.

```
# ip addr
...
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ... master
bond0 state UP ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 ... master
bond0 state UP ...
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default
    link/ether 00:16:3e:00:03:02 brd ff:ff:ff:ff:ff:ff
        inet 192.168.2.12/24 brd 192.168.2.255 scope global noprefixroute
bond0
...
```

Note that the bond and the two slave entries have the same MAC address.

## Activate the Bond

- You can use the `nmcli` command to bring up the connections.
- Bring up the slaves first, and then bring up the bond interface.
- The following commands bring up the slaves:

```
# nmcli con up bond-slave-eth2  
# nmcli con up bond-slave-eth3
```

- The following command brings up the `bond0` interface:

```
# nmcli con up bond0
```



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

The bond connection and the slave connections are up by default after configuring these connections. However, you can use the following commands as needed.

You can use the `nmcli` command to bring up the connections. Bring up the slaves first, and then bring up the bond interface. The following commands bring up the slaves:

```
# nmcli con up bond-slave-eth2  
# nmcli con up bond-slave-eth3
```

The following command brings up the `bond0` interface:

```
# nmcli con up bond0
```

The `ip addr` command, or the `ip link` command, now shows the slave and the bond interfaces that are up.

```
# ip link  
...  
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> ... state UP  
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> ... state UP  
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> ... state UP  
...
```

Stopping the master bond interface also stops the slave interfaces.

## Viewing Network Bonding Information

- Each network interface contains a directory in the `/sys/class/net` directory. For example:

```
# ls /sys/class/net  
bond0 bonding_masters eth0 eth1 eth2 eth3 lo
```

- The `bond0` directory contains a `bonding` directory as well as a directory for each of the slaves. For example:

```
# ls /sys/class/net/bond0  
bonding lower_eth2 lower_eth3 ...
```

- The `/proc/net/bonding` directory also contains a file with the same name as the bond that provides configuration information. For example:

```
# ls /proc/net/bonding/  
bond0
```



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

Each network interface contains a directory in the `/sys/class/net` directory. For example:

```
# ls /sys/class/net  
bond0 bonding_masters eth0 eth1 eth2 eth3 lo
```

In this example, a network bond named `bond0` exists. A directory of the same name exists that contains configuration information for that bond. For example:

```
# ls /sys/class/net/bond0  
addr_assign_type carrier ifalias netdev_group lower_eth3 ...  
...
```

Within this directory is a `bonding` directory that contains information for the `bond0` interface. For example:

```
# ls /sys/class/net/bond0/bonding  
active_slave all_slaves_active mimon primary_reselect ...  
...
```

There are also directories that contain information for each of the slaves. For example:

```
# ls /sys/class/net/bond0/lower_eth2  
addr_assign_type device ifalias mtu rxbuf_cur ...
```

Following are some examples of viewing files in the /sys/class/net directory.

```
# cat /sys/class/net/bonding_masters
bond0
# cat /sys/class/net/bond0/operstate
up
# cat /sys/class/net/bond0/address
00:16:3e:00:03:01
# cat /sys/class/net/bond0/bonding/active_slave
eth2
# cat /sys/class/net/bond0/bonding	mode
active-backup 1
# cat /sys/class/net/bond0/bonding/slaves
eth2 eth3
```

Following is an example of viewing the /proc/net/bonding/bond0 file.

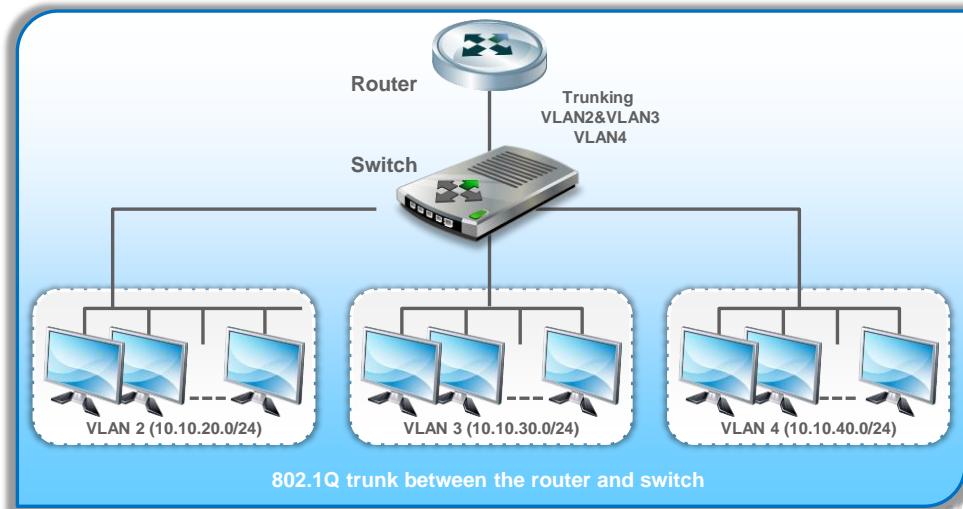
```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth2
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth2
MII Status: up
Speed: ...
Duplex: ...
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0

Slave Interface: eth3
MII Status: up
Speed: ...
Duplex: ...
Link Failure Count: 0
Permanent HW addr: 00:16:3e:00:03:01
Slave queue ID: 0
```

## Virtual Local Area Networks: Introduction



**ORACLE®**

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

A VLAN is a type of local area network that does not have its own dedicated physical infrastructure, but instead uses another local area network to carry its traffic. The traffic is encapsulated so that a number of logically separated VLANs can be carried by the same physical LAN. With VLANs, you can create multiple distinct broadcast domains that are mutually isolated. With VLANs, network switches (not routers) create the broadcast domain.

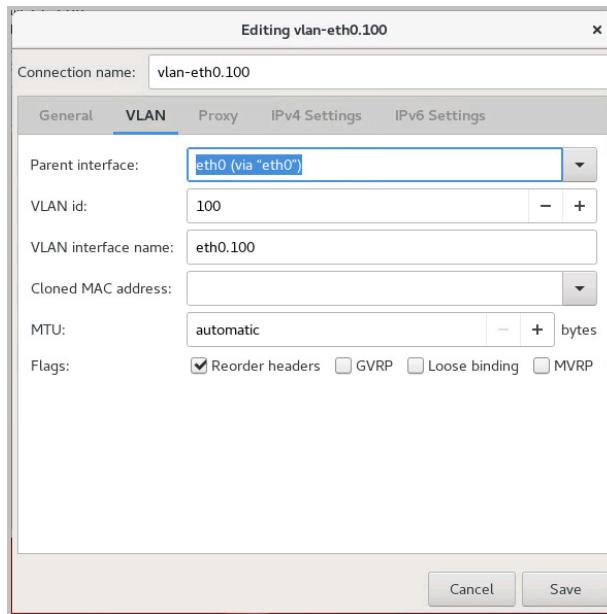
Each VLAN is identified by a VID (VLAN Identifier) in the range 1 to 4094 inclusive. Switch ports are assigned to a VLAN ID, and all ports assigned to a single VLAN are in a single broadcast domain. The VID is stored in an extra 4-byte header that is added to the packet called the Tag. Adding a Tag to a packet is called tagging.

IEEE 802.1Q is a protocol for carrying VLAN traffic on Ethernet. There are two ways in which a machine can be connected to a switch carrying 802.1Q VLAN traffic:

- Via an *access* (or *untagged*) port, where VLAN support is handled by the switch (so the machine sees ordinary, *untagged* Ethernet frames)
- Via a *trunk* (or *tagged*) port, where VLAN support is handled by the attached machine (which sees 802.1Q-tagged Ethernet frames)

Linux has the ability to use an Ethernet interface as an 802.1Q trunk port, allowing it to concurrently send and receive traffic on multiple VLANs. This is provided by the 8021q kernel module.

# Using the nm-connection-editor GUI to Configure 802.1Q VLAN Tagging



ORACLE®

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

The slide shows the nm-connection-editor GUI used to configure an 802.1Q VLAN connection. You can access this GUI by using the following command:

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

Select “VLAN” as the connection type to display the GUI shown in the slide.

From the “VLAN” tab as shown in the slide, you can do the following:

- Provide a name for the VLAN connection, which defaults to “VLAN connection 1”
- Select the “Parent interface” from a drop-down list of available physical interfaces
- Provide the “VLAN id,” which is set to 100 in this example
- Provide the VLAN interface name,” which is set to eth0.100 in this example
- Optionally provide “Cloned MAC address” and “MTU” settings

Click “Save” and the interface files are created in /etc/sysconfig/network-scripts/.

## Using the `nmcli` Utility to Configure VLAN Tagging

- The following command creates a VLAN device named `eth0.100` and a VLAN connection named `vlan-eth0.100`:

```
# nmcli con add type vlan con-name vlan-eth0.100 ifname eth0.100 dev eth0
    id 100 ip4 192.168.100.1/24
```

- The example also sets the following parameters:
  - VLAN ID: 100
  - IPv4 address: 192.168.100.1/24
  - Physical (parent) device: `eth0`
- Both `eth0` and `eth0.100` have the same MAC address.
- The `nmcli` command automatically creates the VLAN network interface configuration file:
  - `/etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100`



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

You can also use the `nmcli connection` command to create a VLAN connection. Include the “add type vlan” arguments and any additional information to create a VLAN connection. For example:

```
# nmcli con add type vlan con-name vlan-eth0.100 ifname eth0.100 dev eth0
    id 100 ip4 192.168.100.1/24
```

The example defines the following attributes of the VLAN connection:

- `con-name vlan-eth0.100`: Specifies the name of the new VLAN connection
- `ifname eth0.100`: Specifies the interface to bind the connection to
- `dev eth0`: Specifies the physical (parent) device this VLAN is on
- `id 100`: Specifies the VLAN ID
- `ip4 192.168.100.1/24`: Specifies IPv4 address to assign to the interface

The `nmcli con` command shows the new VLAN connection.

```
# nmcli con
      NAME           UUID           TYPE           DEVICE
      vlan-eth0.100  ...            wlan          eth0.100
```

This command creates the `ifcfg-vlan-eth0.100` file. Following is the contents of this file:

```
# cat /etc/sysconfig/network-scripts/ifcfg-vlan-eth0.100
VLAN=yes
TYPE=Vlan
DEVICE=eth0.100
PHYSDEV=eth0
VLAN_ID=100
REORDER_HDR=0
BOOTPROTO=none
IPADDR=192.168.100.1
PREFIX=24
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=vlan-eth0.100
UUID=...
ONBOOT=yes
```

You can use the `ip addr` command to view the protocol address information for the network devices. The following shows the VLAN interface, `eth0.100`:

```
# ip addr
6: eth0.100@eth0: ...
    link/ether 00:16:3e:00:01:01 brd ff:ff:ff:ff:ff:ff
        inet 192.168.100.1/24 brd 192.168.100.255 scope ... eth0.100
...
...
```

The `nmcli dev` command shows the `eth0.100` device.

```
# nmcli dev
      NAME           TYPE      STATE      CONNECTION
      eth0.100       vlan     connected   wlan-eth0.100
```

The `nmcli con` command shows the `wlan-eth0.100` connection.

```
# nmcli con
      NAME           UUID           TYPE      DEVICE
      wlan-eth0.100  ...           wlan     eth0.100
```

## Viewing VLAN Information

- Each VLAN interface contains a directory in the `/sys/class/net` directory. For example:

```
# ls /sys/class/net
eth0.100 ...
```

- The `eth0.100` directory contains configuration information for the VLAN interface.

- The `/proc/net/vlan` directory also contains information about the VLAN. For example:

```
# ls /proc/net/vlan/
config eth0.100
```

- You can use the `tcpdump` command to view tagged and untagged packets on the wire. For example:

```
# tcpdump -e -i eth0
```



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

Each network interface contains a directory in the `/sys/class/net` directory. For example:

```
# ls /sys/class/net
eth0.100 eth0 eth1 eth2 eth3 lo
```

In this example, a VLAN interface exists named `eth0.100` and a directory of the same name exists that contains configuration information for that interface. For example:

```
# ls /sys/class/net/eth0.100
addr_assign_type carrier flags link_mode power ...
...
```

There are also files in the `/proc/net/vlan` directory that describe the VLAN interface. For example:

```
# ls /proc/net/vlan
config eth0.100
```

You can use the `tcpdump` utility to see tagged and untagged packets to ensure traffic is showing up on the expected interfaces. The `-e` option specifies the Ethernet header that includes 802.1Q tags. Use the `-i` option to specify the interface. For example:

```
# tcpdump -e -i eth0
```

# Virtual Private Networks: Introduction

- A virtual private network (VPN) is a secure connection between two or more endpoints over an untrusted network.
  - VPNs create a secure tunnel that typically uses authentication and encryption between the two endpoints.
- There are two main types of VPNs:
  - Site-to-Site: Each site has a VPN gateway that provides a secure link between client systems at each site.
  - Remote Access: Also referred to as mobile VPN or Road Warrior. Each client has VPN software and connects to a remote VPN gateway.
- IPSec (IP Security) is the preferred method for creating a VPN.
- IPSec includes various protocols including AH, ESP, and IKE.



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

A virtual private network (VPN) enables secure communication between two or more private networks over an untrusted network such as the Internet. VPNs create a secure tunnel that typically uses authentication and encryption between the two networks.

There are different types of VPNs:

- **Site-to-Site VPN:** Each site has a VPN server that encrypts data and provides a secure link between the sites. Client systems at each site do not have VPN client software, they send and receive normal TCP/IP traffic through the VPN gateway server.
- **Remote Access VPN:** A secure connection is made from an individual computer to a VPN gateway. Every host must have VPN client software.

IPSec (IP Security) is the preferred method for creating a VPN. IPSec is a layer 3 protocol and can protect any protocol that runs on top of IP. IPSec consists of various protocols and algorithms including the following:

- **Authentication Header (AH):** Protects the integrity of the entire packet including header information such as the source and destination IP addresses
- **Encapsulating Security Payload (ESP):** Protects the application-level data and provides both integrity checking and encryption
- **Internet Key Exchange (IKE):** Protocol that uses a system of automatic keying to derive a key for IPSec communications

## The libreswan RPM Package

- libreswan is an open source IPSec implementation included with Oracle Linux 7.
  - Uses the Network Security Services (NSS) cryptographic library that is required for FIPS security compliance
  - Provides the ipsec command-line utility and a number of utilities in the /usr/libexec/ipsec directory
  - Provides the IPSec configuration directory, /etc/ipsec.d
  - Provides the main configuration file for IPSec, /etc/ipsec.conf
- Use the ipsec command to generate security keys, view security keys, and view connection status. Example:

```
# ipsec newhostkey --configdir /etc/ipsec.d --output
    /etc/ipsec.d/www.example.com.secrets
# ipsec showhostkey --left
# ipsec auto --status
```



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

In Oracle Linux 7, IPSec is provided by the libreswan RPM package. libreswan is an open source IPSec implementation. It uses the Network Security Services (NSS) cryptographic library that is required for Federal Information Processing Standard (FIPS) security compliance. Use the yum command to install this package and all required dependencies.

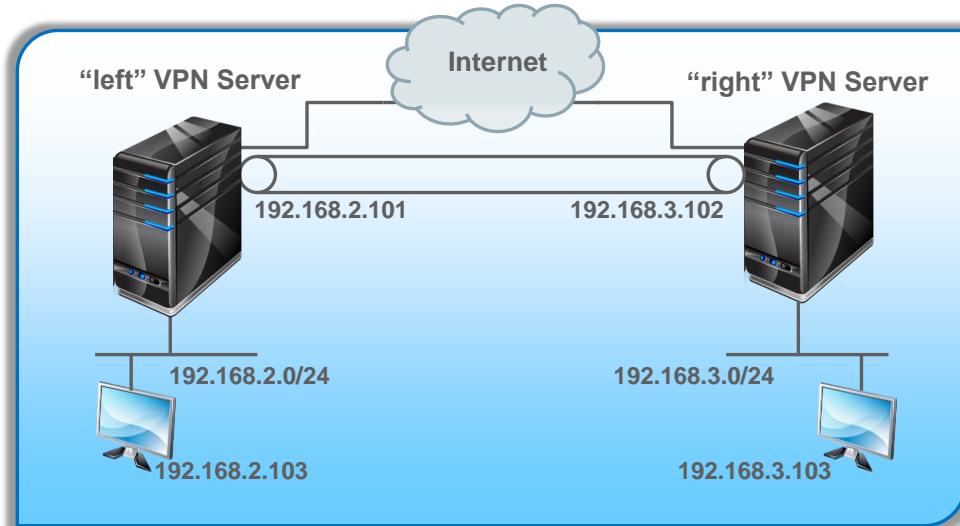
```
# yum install libreswan
```

The libreswan package provides the ipsec command-line utility, a number of utilities in the /usr/libexec/ipsec directory, and several other files including man pages and documentation. The IPSec configuration directory is /etc/ipsec.d and the main configuration file for IPSec is /etc/ipsec.conf.

The ipsec utility provides a number of commands. Use the ipsec --help command to view a list of the commands. The following describes a few of the ipsec commands:

- ipsec newhostkey: Generate a new RSA authentication key for a host. Include the --output <filename>.secrets option to store the RSA authentication key.
- ipsec showhostkey: Display a host's authentication key. The output is suitable for copying and pasting the key in to the /etc/ipsec.conf configuration file.
- ipsec auto --status: View VPN connection status and supported ESP and IKE algorithms.

## Site-to-Site VPN



ORACLE®

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

The slides show a VPN configuration that is referred to as a site-to-site, or gateway-to-gateway VPN. In this configuration, there are two VPN servers that each have a public Internet IP address. The VPN servers also have connections to two separate private networks, 192.168.2.0/24 and 192.168.3.0/24.

Each VPN server is running IPSec. IPSec is configured to provide a secure VPN tunnel connection between the two private networks.

The main configuration file for IPSec is `/etc/ipsec.conf`. You need to configure a "sitetosite" connection in this file and provide configuration information for each VPN server.

libreswan uses the terms "left" and "right" to refer to the VPN servers at each site. In this example, the server with the IP address of 192.168.2.101 is the "left" side of the connection and the server with the IP address of 192.168.3.102 is the "right" side of the connection. Parameters for the "left" and "right" VPN servers are defined for the "sitetosite" connection in the `/etc/ipsec.conf` file.

## Site-to-Site VPN: Configuration

On each VPN server:

- Use the `sysctl` command to enable IP forwarding
- Use the `ipsec newhostkey` command to generate an RSA authentication key
- Use the `ipsec showhostkey --left` command to view the key on the left host
- Use the `ipsec showhostkey --right` command to view the key on the right host
- Copy and paste the output of each `ipsec showhostkey` command into the `/etc/ipsec.conf` file
- Complete the “`sitetosite`” connection configuration in the `/etc/ipsec.conf` file (details in the next slide)
- Add `firewalld` rules to trust the `ipsec` protocols
- Use the `systemctl` command to start the `ipsec` service



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

For the site-to-site VPN configuration in the previous slide, each of the VPN servers acts as a router and forward packets destined for other hosts. Therefore, IP forwarding needs to be enabled on each VPN server. You can use the `sysctl` command to enable IPv4 forwarding on each VPN server:

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

On each VPN server, use the `ipsec newhostkey` command to generate an RSA authentication key. Example:

```
# ipsec newhostkey --configdir /etc/ipsec.d --output
/etc/ipsec.d/www.example.com.secrets
```

In this example, an RSA key pair is generated by using the Network Security Services (NSS) database and is written to the `/etc/ipsec.d/www.example.com.secrets` file.

In the example configuration in the slide titled “Site-to-Site VPN,” the server with the IP address of 192.168.2.101 is the “left” side of the connection. From this server, use the `ipsec showhostkey --left` command to view the key on the left host. The `--left` option causes the output to be in the `ipsec.conf` format as a “`leftrsasigkey`” parameter.

```
# ipsec showhostkey --left
...
leftrsasigkey=...
```

From the “right” VPN server, use the `ipsec showhostkey --right` command to view the key on the right host. The `--right` option causes the output to be in the `ipsec.conf` format as a “`rightrsasigkey`” parameter.

```
# ipsec showhostkey --right  
...  
rightrsasigkey=...
```

Complete the “sitetosite” connection configuration in the `/etc/ipsec.conf` file with the “`leftrsasigkey`” parameter, the “`rightrsasigkey`” parameter, and the following information. This assumes the configuration represents the example in the “Site-to-Site VPN” slide.

```
# vi /etc/ipsec.conf  
conn sitetosite  
    leftid=192.168.1.101  
    left=192.168.1.101  
    leftsourceip=192.168.2.101  
    leftsubnet=192.168.2.0/24  
    lefrsasigkey=...  
  
    rightid=192.168.1.102  
    right=192.168.1.102  
    rightsourceip=192.168.3.102  
    rightsubnet=192.168.3.0/24  
    rightrsasigkey=...  
  
    authby=rsasig  
    auto=start
```

Use the following command to check the syntax of the `/etc/ipsec.conf` file. If any errors are returned, the line number is included. If no errors are returned, the syntax is correct:

```
# /usr/libexec/ipsec/addconn --config /etc/ipsec.conf --checkconfig
```

Copy the completed `/etc/ipsec.conf` file to the other VPN server so that both the “left” and the “right” VPN servers have the same configuration file.

Either add `firewalld` rules to trust the `ipsec` protocols, or stop the `firewalld` service on both the “left” and the “right” VPN servers. The following example uses the `systemctl` command to stop the `firewalld` service.

```
# systemctl stop firewalld
```

Use the `systemctl` command to start the `ipsec` service on both the “left” and the “right” VPN servers.

```
# systemctl start ipsec
```

## Example: “sitetosite” Connection

```
conn sitetosite
    leftid=192.168.1.101
    left=192.168.1.101
    leftsourceip=192.168.2.101
    leftsubnet=192.168.2.0/24
    leftrsasigkey=...

    rightid=192.168.1.102
    right=192.168.1.102
    rightsourceip=192.168.3.102
    rightsubnet=192.168.3.0/24
    rightrsasigkey=...

    authby=rsasig
    auto=start
```



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

The slide shows the “sitetosite” connection configuration in the /etc/ipsec.conf file. This assumes that the configuration represents the example in the “Site-to-Site VPN” slide. The parameters are described as follows. Refer to the ipsec.conf(5) man page for more information.

- **conn:** A connection specification defining a network connection to be made by using IPSec. In this example, the connection name is “sitetosite” but could be named whatever you want.
- **leftid|rightid:** Identifies the “left|right” VPN servers for authentication purposes. This could be an IP address or a fully qualified domain name.
- **left|right:** The IP address of the “left|right” VPN servers.
- **leftsourceip|rightsourseip:** The IP address of the “left|right” VPN server to use when transmitting a packet to the other side of the link.
- **leftsubnet|rightsubnet:** The private subnet behind the “left|right” participant.
- **leftrsasigkey|rightrsasigkey:** The output of the ipsec showhostkey command for the “left|right” participant. This key is generated by using the ipsec newhostkey command.
- **authby:** How the two security gateways authenticate each other.
- **auto:** What operation to perform at IPSec startup.

## Multi-Factor Authentication

Is a method to establish a higher level of confidence between a user and a system

- Many systems today still require only two credentials:
  - Username
  - Password
- However, these days more credential factors are necessary to enable greater security between users and systems.
- Three types of credential factors are:
  - Something you **know**
  - Something you **have**
  - Something you **are**



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

An example of a something you know is a username or password. A token is the something you have, whereas the something you are could be a form of biometric authentication such as a fingerprint or eye scan.

## Two-Factor Authentication (2FA)

Is a subset of multi-factor authorization by requiring a user to provide a token to the system for authenticating

- A token can be issued by software application or hardware device
  - Examples of tokens are:
    - Receiving a number code via SMS messaging
    - Randomly generated code from One Time Password (OTP) Token Device
    - USB Key
    - Smart Card
  - Numbered and generated tokens may be time sensitive to use.
  - Users must still guard against hackers by not losing token devices, keys, mobile phones or cards.



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

Two-Factor Authentication is not a silver bullet for protecting against cyber hackers. Users must protect their token devices and online accounts from potential hackers. However, 2FA does provide an extra layer of security for preventing potential hackers from gaining access to online accounts or even bank accounts. Some services allow multiple tokens for gaining into one's account by requiring a username, password along with an SMS Code or OTP tokens. If either tokens do not work, the user may use a USB key or smart card to still gain access to one's account.

## Pluggable Authentication Modules (PAM)

Is an authentication framework abstracted from software applications and an operating system

- PAM provides a pluggable modules model for enabling multi-factor authentication.
- PAM provides stacking multiple modules for system administrators to easily add new authentication mechanisms to a system.
- PAM comprises three components:
  - Authentication library API
  - Authentication mechanism-specific modules
  - Service Provider Interface (SPI)



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

PAM's pluggable model provides an easy method for adding or removing a module from an Oracle Linux operating system. Stacking modules is an approach for system administrators to order the listed modules in a top-down fashion and require users to authenticate according to specified conditions. The authentication library API comprises the core component applications, such as ftp, ssh, and login. The authentication mechanism-specific modules include the added modules for user authentication such as Kerberos, fingerprinting, and smart cards. The service provider interface (SPI) connects the front-end authentication library API to the back-end authentication mechanism-specific modules.

## Stacking Multiple Modules

service	module_type	control_flag	module_path	options
login	auth	required	pam_unix_auth.so	nowarn
login	session	required	pam_unix_session.so	
login	account	required	pam_unix_account.so	
ftp	auth	required	pam_skey_auth.so	
ftp	session	required	pam_unix_session.so	
sshd	session	required	pam_unix_session.so	
login	password	required	pam_unix_passwd.so	
passwd	password	required	pam_unix_passwd.so	



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

## 2FA PAM and Smart Cards

PAM offers two-factor authentication by integrating a smart card authentication module.

- For example, smartcard-auth tests for a smart card in line 2:

```
auth    required    pam_env.so
auth    [success=done ignore=ignore default=die] pam_pkcs11.so  nodebug
wait_for_card
auth    required    pam_deny.so
...
```

- The `pam_pkcs11.so` module checks for the insertion of a smart card by waiting due to the `wait_for_card` option.
- If a valid smart card is not used, access is not authorized as specified by `pam_deny.so`.



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

# Internet Protocol Security (IPSec)

Is designed to provide security for the IP layer, IPv4 and IPv6, using cryptography

- Essentially, IPSec protects the following types of network paths:
  - Between two hosts
  - Between two security gateways
  - Between a host and security gateways
- IPSec implements two primary protocols for providing security:
  - Authentication Header (AH)
    - Provides connectionless integrity, data origin authentication, and protection from replaying data sequences
  - Encapsulating Security Payload (ESP)
    - Provides confidentiality to encapsulated data



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

To be IPSec compliant, a host implementation must support secure connectivity between two hosts and a host and security gateway. A security gateway must support secure connectivity between two hosts, between two security gateways, and between a host and security gateways. The authentication header is an IP header that offers messaging integrity while protecting as much of the IP datagram as it can when in transit. AH exists between the transport header and IP header. The transport header may be TCP, UDP, ICMP, or an alternative IP header when tunneling is implemented. The encapsulating security payload provides confidentiality to AH as well as what data is encapsulated. ESP is an optional authentication service. ESP offers mixed services such as implementing without other security support, with AH, or even nested within itself. ESP exists between the IP header and the next layer protocol header or before an encapsulated IP header.

For more information about IPSec, read and review RFC 4301 at <https://tools.ietf.org/html/rfc4301>. To learn more about authentication header, see RFC 4302 at <https://tools.ietf.org/html/rfc4302>. More information for encapsulating security payload is described by RFC 4303 at <https://tools.ietf.org/html/rfc4303>.

## Port Address Translation (PAT)

Is a network mapping scheme for looking up and forwarding incoming connections to alternate Internet ports

- PAT uses a routing table for listing source ports to destination ports.
- An example of a PAT routing table is as follows:

Source Port	Destination
8080	10.0.0.2, 80
22	10.0.0.3, 10022
...	...

- All incoming connections requesting port 8080 are forwarded to the machine at 10.0.0.2 port 80.
- All incoming ssh requests are forwarded to the machine at 10.0.0.3 port 10022.



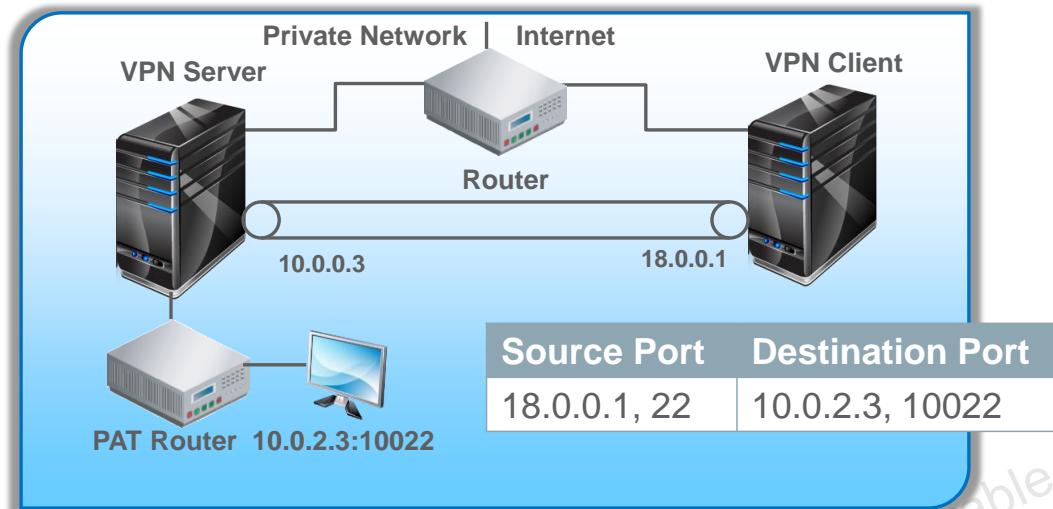
Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

PAT is an ideal networking mapping scheme for working with private networks implementing network address translation (NAT). NAT maps a single private IP address to a NAT device. Three common private IP ranges are:

- Class A: 10.0.0.0 - 10.255.255.255/8
- Class B: 172.16.0.0 - 172.31.255.255/12
- Class C: 192.168.0.0 - 192.168.255.255/16

In the slide, an example of a PAT routing table is defined by a system administrator. The left column of the table cites the source port number while the right column references the destination machine and specific port. Incoming traffic is routed according to the PAT routing table. A system administrator may enter additional source and destination ports as needed. In addition, a system administrator is able to update or remove source to destination port configuration.

## IPSec with Port Address Translation



ORACLE®

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

## IPSec VPN Connection to On-Premises Network

You can connect your on-premises network to a virtual cloud network (VCN) by using IPSec VPN.

- Before connecting to a VCN, the following are required:
  - Configuration for asymmetric routing
  - Knowledge and creation of cloud network resources
  - Knowledge of your on-premises router
- After connecting to a VCN, you can test the IPSec VPN connection by launching a cloud resource instance followed by pinging the new instance from your on-premises network.



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

The three requirements mentioned on this slide are important for successful IPSec connection to a VCN. For the first requirement, asymmetric routing is a method for routing network traffic from a source to a destination using a particular path and then routing the returning network traffic from destination to source using an alternate path. The second requirement involves the existing knowledge, experience, and creation of VCN resources. You must also know the following:

- VCN ID
- VCN Classless Inter-Domain Routing (CIDR)
- VCN CIDR subnet
- VPN headend (Oracle's IPSEC tunnel endpoint IP address) for each IPSec Tunnel
- Pre-shared key (PSK) for each IPSec Tunnel

The third requirement covers the knowledge of your existing on-premises router. You must possess specific information about the inside and outside interfaces for your on-premises router.

## 802.3ad LAG (Link Aggregation Group)

Is an IEEE specification for aggregating Ethernet interfaces for higher bandwidth throughput from a single interface

- 802.3ad LAG bundles multiple physical interfaces to form one interface increasing port density while lowering cost.
- Links are full duplex, point-to-point, or same data rate.
- Additional features of 802.3ad LAG include:
  - Graceful recoveries when link failures occur
  - Traffic throughput distributed in packets
  - Same messages sent over same links for eliminating the order of messages
  - Realizes high utilization when simultaneous message conversations occur



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

## Link Aggregation Control Protocol (LACP)

Is a protocol for aggregating Ethernet interfaces

- LACP is an IEEE effort that supports 802.3ad Link Aggregation.
- LACP helps in the maintenance and control of ports with the exchange of system information for LAG bundling.
- LACP monitors and detects newly configured Ethernet interfaces.
- Additional features of LACP are:
  - Assigns highest port priorities to LAG bundles
  - Utilizes Protocol Data Units (PDUs) between links and bundles
  - Dynamically configure LAG bundles with IPv6 addresses
  - Route IP, VLAN, PPPoE, or MPLS traffic over the links



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

A version of the IEEE specification for 802.3ad is located at [http://www.ieee802.org/3/ad/public/mar99/seaman\\_1\\_0399.pdf](http://www.ieee802.org/3/ad/public/mar99/seaman_1_0399.pdf). PPPoE stands for Point-to-Point Protocol over Ethernet. MPLS is an acronym for Multiprotocol Label Switching, which is a method of data transfer protocol used by telecommunication organizations.

## Virtual Port Channel (vPC)

Is a method for connecting two physical switches together to act and appear as a single device to another device

- vPC creates a single data link across two separate switches.
- Each vPC enabled switch remains physically and logically separate from one another.
- One vPC switch is designated as a control plane and the other vPC switch is the data plane.
- Even though the switches are connected, they are not clustered.
  - Important benefits of vPC:
    - Provides high availability
    - Provides link-level resiliency
    - Provides all available uplink bandwidth



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

# Network Analysis

Why should you analyze a network?

- To conduct daily system administration tasks and operations
- To resolve current network issues and potential problems
- To defend against cyber attacks, intruders, and malicious software
- To plan for scaling current network and for future network expansion
  - Methods for Analyzing Networks
    - Using command line utilities
    - Installing and configuring third-party software
    - Receiving feedback and comments from internal and external users



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

As networks become more and more complex, the need to analyze network traffic and data packets is an important task for system administrators. Moreover, cybersecurity continues to be a growing concern for enterprise organizations due to the proliferation of malicious software and network attacks by bad actors. Enterprise networks must also scale to meet new demands by its employees or additional network services required. System administrators may leverage network analysis tools to analyze traffic, eliminate malicious software, and for scaling out a network.

## ip command

Is a command-line utility for showing and managing network devices and routing information

- The `ip` command replaces the deprecated `ifconfig` command.
- To show an enumerated list of network devices, run the following:

```
# ip addr show
```

- To display information for a specific network device, run the following:

```
# ip addr show dev eno1
```

- To stop network traffic on a network device, run the following:

```
# ip link set eno1 down
```

- To return network traffic on a network device, run the following:

```
# ip link set eno1 up
```



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

## ss command

Is a command-line utility for investigating network sockets

- The `ss` command replaces the deprecated `netstat` command.
- To show all listening and non-listening network sockets, use the command:

```
# ss -a
```

- For viewing only the listening sockets on a system, run the command:

```
# ss -l
```

- To view all tcp-related sockets, use the following command:

```
# ss -t -a
```

- For viewing all udp-only sockets, run the command:

```
# ss -u -a
```



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

## Tcpdump command

Is a command line program that captures packets in a network

- Listens to network interfaces for monitoring network packets
- Displays capture packets in real time or enables you to save to a file for reviewing at a later time
- Lists data link types for an attached network interface

```
# tcpdump -i eno1 port 80
```

- To display packets for a network device on port 80, run the following:

```
# tcpdump -i eno1 -nn port 80
```

- To display packets without resolving host names, run the following:

```
# tcpdump -i eno1 -nn port 80 -w capturefile.pcap
```

- For saving captured packets, add the -w option and a filename:

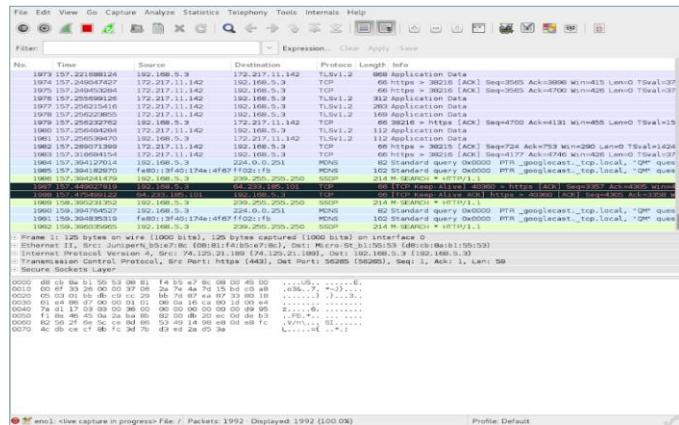


Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

# Wireshark

Is a GUI program for capturing and analyzing network packets

- Analyzes packets in real-time and offline
- Inspects different types of network protocols
- Decrypts packets such as IPsec, Kerberos, and SSL/TLS
- Color-codes captured packets for grouping by a particular protocol



**ORACLE®**

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

Official documentation for Wireshark can be found at [https://www.wireshark.org/docs/wsug\\_html/](https://www.wireshark.org/docs/wsug_html/).

## Capturing Packets with Wireshark

To capture packets from your local host, install Wireshark:

```
# yum install wireshark
```

- Start Wireshark as root:

```
# wireshark
```

- The startup screen for the Wireshark GUI will open.
- Under Start, select the desired network interface or interfaces for capturing packets.
- Next, click the green shark fin icon on the Wireshark toolbar to begin capturing local packets.
- Network packets are captured in real time and color-coded for easier viewing.
- To stop capturing packets, click the red square on the toolbar.



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

You can save captured packets to a pcap file or other file types for viewing and analyzing in offline mode. Under File, select Save As... and choose the file type Wireshark/tcpdump/... -pcap. Name the file and click Save.

## Analyzing Packets with Wireshark

You can inspect packets after capturing by opening a pcap file.

- To inspect a capture file, select File on the toolbar.
- Locate a file, for example a pcap file, and click the Open button.
- You will then see the captured packets in the Wireshark main window.
- You can scroll down for viewing all of the captured packets which are automatically numbered as they were captured.
- Packets are sorted by time captured, but you can reorder packets by source, destination, protocol, length, or info.
- By clicking a packet, you can inspect a packet further by its details and bytes.



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

## tshark

Is the command line version of Wireshark supporting the same options

- You must first install Wireshark before running tshark.
- tshark is helpful when a GUI is not available.
- It uses the same capture file format pcap as tcpdump and Wireshark.
- To view a list of available network interfaces, run the command:

```
# tshark -D
```

- To set a network interface for live capturing, run the following:

```
# tshark -i en01
```

- To save captured packets to a file, run the commands:

```
# touch tsharkpackets.pcap
```

```
# tshark -i en01 -w tsharkpackets.pcap
```



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

# Network Performance Tuning

Performance tuning should be implemented for improving network operations.

- Three types of system elements that require maintaining for enhancing network performance are:
  - Processes
  - Memory
  - Files
- To manage these system elements, three related system tools are available to install and administer:
  - top
  - vmstat
  - lsof



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

## top command

Is a command-line utility for displaying and monitoring Linux processes in real time

- Top displays information about tasks, CPU usage, main memory and swap memory.
- Top displays a table with process IDs, usernames, and so on.
- To run `top`, run the following command:

```
# top
```

— To quit `top`, press the “q” key.

- To show all of the options for `top`, run this command:

```
# top -h
```

- To show all of the processes and threads owned by an owner, run the following command with the `oracle` user as the option:

```
# top -u oracle
```



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

## Virtual Memory Statistics (`vmstat`)

Is a command-line utility that reports statistics on virtual memory

- `vmstat` displays processes, memory, paging, block IO, and so on.
- Information about the last reboot averages is first displayed.
- Reports on process and memory are displayed in real time.
- To run `vmstat`, type the following command:

```
# vmstat
```

- To display active and inactive memory, type the following:

```
# vmstat -a
```

- To display statistics about event counters and memory, run the following command:

```
# vmstat -s
```



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

## List Open Files (`lsof`)

Is a command-line utility that lists open files on a system

- `lsof` provides a column-based table with the name of the process, owner of the file, size of file, and so on.
- An open file may be a regular file, directory, a library, and so on.
- To display all open files that belong to active processes, run the `lsof` command as follows:

```
# lsof
```

- To view all of the opened files owned by the `oracle` user, run the command:

```
# lsof -u oracle
```

- To return all open files found on the `/dev/sda2` device, run the following command:

```
# lsof /dev/sda2
```



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



## Quiz

Which of the following statements are true?

- a. Network bonding allows you to combine multiple physical network interfaces together into a single interface.
- b. In network bonding, the physical interfaces are called “slaves” and the logical bonded interface is called “master.”
- c. You need to load the 8021q kernel module to implement network bonding.
- d. A VLAN does not have its own dedicated physical infrastructure and instead uses another LAN to carry its traffic.
- e. ESP is the preferred method for creating a VPN in Oracle Linux 7.



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe Network Bonding
- Configure Network Bonding
- Describe Virtual Local Area Networks (VLANs)
- Configure a VLAN
- Describe Virtual Private Networks (VPNs)
- Configure a Site-to-Site VPN
- Describe Multi-Factor Authentication
- Describe Internet Protocol Security (IPSec)
- Describe Network Analysis
- Describe Network Performance Tuning



ORACLE®

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

## Practice 9: Overview

This practice covers the following topics:

- Configuring Network Bonding by Using the TUI
- Configuring Network Bonding from the Command Line
- Working with Bonded Interfaces
- Configuring 802.1Q VLAN Tagging by Using the GUI
- Configuring 802.1Q VLAN Tagging from the Command Line
- Working with VLAN Interfaces
- Configuring a Site-to-Site VPN



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

# Implementing the XFS File System

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 XFS for Oracle Linux
- Create an XFS file system
- Use the `xfs_growfs` utility
- Use the `xfs_admin` utility
- Enable disk quotas on an XFS file system
- Use the `xfs_quota` utility
- Set project quotas
- Use the `xfsdump` and `xfsrestore` utilities
- Use XFS file system maintenance utilities



ORACLE®

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

# XFS File System

XFS:

- Is a high-performance journaling file system
- Is available for Oracle Linux x86\_64 architecture
- Is the default file system for Oracle Linux 7
- Can be created on a regular disk partition and on a logical volume
- Supports extended attributes
  - Used by Access Control Lists (ACL) and SELinux
- Supports user, group, and project disk quotas



ORACLE®

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

The XFS file system is a high-performance journaling file system. XFS in Oracle Linux is available for the x86\_64 architecture and has been supported in the Unbreakable Enterprise Kernel since Release 2 (2.6.39). XFS is the default file system for Oracle Linux 7 and supports large file system sizes. You can create an XFS file system on a regular disk partition and on a logical volume.

The data section of an XFS file system contains the file system metadata (inodes, directories, and indirect blocks) and the user file data. The data section is partitioned into allocation groups, which are virtual storage regions of fixed size. Any files and directories that you create can span multiple allocation groups. Each allocation group manages its own set of inodes and free space independently of other allocation groups to provide both scalability and parallelism of I/O operations.

The XFS journal (or log) can be located internally in the data section of the file system, or externally on a separate device to reduce the number of disk seeks. The journal stores changes to the running file system metadata until those changes are written to the data section. XFS journaling guarantees consistency of the file system following loss of power or a system crash. When mounting a file system after a crash, the journal is read to complete operations that were in progress at the time of the crash.

See the following Oracle guidelines for file and file system limits:

[https://docs.oracle.com/cd/E52668\\_01/E93593/html/ol7-system-requirements-limits.html](https://docs.oracle.com/cd/E52668_01/E93593/html/ol7-system-requirements-limits.html).

## Creating an XFS File System

- Use the `mkfs.xfs` command to create an XFS file system:

```
# mkfs.xfs /dev/xvdd1
meta-data=/dev/xvdd1 isize=256 agcount=4, agsize=...
          = sectsz=512 attr=2, projid32bit=1
          = crc=0
data      = bsize=4096 blocks=1310719, ...
          = sunit=0 swidth=0 blks
naming    =version 2 bsize=4096 ascii-ci=0 ftype=0
log       =internal log bsize=4096 blocks=2560, ...
          = sectsz=512 sunit=0 blks, ...
realtime  =none extsz=4096 blocks=0, ...
```

- See the `mkfs.xfs(8)` man page for more information.



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

Use the `mkfs.xfs` or `mkfs -t xfs` command to create an XFS file system. The following example creates an XFS file system with an internal log on the `/dev/xvdd1` partition. As shown in the slide, parameters for the file system are displayed as output.

```
# mkfs.xfs /dev/xvdd1
meta-data=/dev/xvdd1 isize=256 agcount=4, agsize=327680 blks
          = sectsz=512 attr=2, projid32bit=1
          ...
          
```

The following example creates an XFS file system on `/dev/xvdb1` but places the journal on another device, `/dev/xvdd1`. The `size` option specifies a 10000 block journal:

```
# mkfs.xfs -l logdev=/dev/xvdd1,size=10000b /dev/xvdb1
```

The following example creates an XFS file system with a stripe-unit size of 32 KB and 6 units per stripe on a logical volume:

```
# mkfs.xfs -d su=32k,sw=6 /dev/myvolg/myvol
```

XFS uses the stripe-unit size and the number of units per stripe information to align data, inodes, and the journal appropriately for the storage. On LVM and Multiple Devices (MD) volumes and some hardware RAID configurations, XFS can automatically select the optimal stripe parameters.

The following example includes the output of the `mkfs.xfs` command. The `-f` option forces the overwrite of an existing file system type. The `-L` option sets the file system label to “XFS”. The `-b size=512` option sets the logical block size to 512 bytes.

```
# mkfs.xfs -f -L XFS -b size=512 /dev/xvdb1
meta-data=/dev/xvdb1      isize=256    agcount=4, agsize=530129 blks
                          =           sectsz=512  attr=2, projid32bit=1
                          =           crc=0
data          =           bsize=512   blocks=2120516, imaxpct=25
                =           sunit=0    swidth=0 blks
naming        =version 2   bsize=4096  ascii-ci=0
log           =internal log bsize=512   blocks=20480, version=2
                =           sectsz=512  sunit=0 blks, lazy-count=1
realtime      =none       extsz=4096  blocks=0, rtextents=0
```

The output shows that an XFS file system has up to three parts: a `data` section, a `log` section (journal), and a `realtime` section. When using the default `mkfs.xfs` options, the `realtime` section is absent, and the `log` area is contained within the `data` section. The `naming` area specifies the settings for the file system directory.

The following are some additional options for the `mkfs.xfs` command:

- **`-b <block_size>`:** Each section of the file system is divided into a certain number of blocks. XFS allows you to choose the logical block size for each section of the file system. The physical disk blocks are always 512 bytes. The default value of the logical block size is 4 KB. This is the recommended block size for file systems larger than 100 MB. The minimum logical block is 512 bytes and is recommended for file systems smaller than 100 MB and for file systems with many small files. The maximum block size is the page size of the kernel.
- **`-d <data_section_options>`:** These options specify the location, size, and other parameters of the data section of the file system. The `data` section of the file system is divided into allocation groups to improve the performance of XFS. More allocation groups imply that you can achieve more parallelism when allocating blocks and inodes. Use the `-d agcount=<value>` option to select the number of allocation groups. The default number of allocation groups is 8 when the file system size is between 128 MB and 8 GB. Alternatively, you can use the `-d agsize=<value>` option to select the size of allocation groups. The `agcount` and `agsize` parameters are mutually exclusive. The minimum allocation group size is 16 MB; the maximum size is just under 1 TB. Increase the number of allocation groups from the default if there is sufficient memory and a lot of allocation activity. Do not set the number of allocation groups too high, because this can cause the file system to use large amounts of CPU time, especially when the file system is nearly full.
- **`-n <naming_options>`:** These options specify the version and size parameters for the file system directory (or naming area). This allows you to choose a logical block size for the file system directory that is greater than the logical block size of the file system. For example, in a file system with many small files, the file system logical block size could be small (512 bytes) and the logical block size for the file system directory could be large (4 KB). This can improve the performance of directory lookups, because the tree storing the index information has larger blocks.

Refer to the man page for `mkfs.xfs` (8) to view a description of all available options.

## xfs\_growfs Utility

- Use this to increase the size of a mounted XFS file system.
- There must be space available on the underlying device.
  - For example, XFS file system on a logical volume
- The syntax of the `xfs_growfs` command is as follows:

```
# xfs_growfs [options] mount-point
```

- Options include:
  - `-d`: Expand the data section to use all available space.
  - `-D <size>`: Specify the size, in number of blocks, to expand the data section of the file system.
  - `-L <size>`: Specify the new size of the log area.
- See the `xfs_growfs (8)` man page for more information.



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

Use the `xfs_growfs` command to increase the size of an XFS file system. The XFS file system must be mounted and there must be space available on the underlying device. The `xfs_growfs` utility is most often used with logical volumes. The syntax of the `xfs_growfs` command is as follows:

```
xfs_growfs [options] mount-point
```

The following options are available for the `xfs_growfs` command:

- `-d`: Expand the data section of the file system to the maximum size of the underlying device.
- `-D <size>`: Specify the size to expand the data section of the file system. The `<size>` argument is expressed in the number of file system blocks.
- `-L <size>`: Specify the new size of the log area. This does not expand the size, but specifies the new size of the log area. Therefore, this option can be used to shrink the size of the log area. You cannot shrink the size of the data section of the file system.
- `-m <maxpct>`: Specify the new value for the maximum percentage of space in the file system that can be allocated as inodes. With the `mkfs.xfs` command, this option is specified with the `-i maxpct=<value>` option.

For more information, see the `xfs_growfs (8)` manual page.

## xfs\_admin Utility

- Use this to change and view the parameters of an XFS file system.
  - Unmount the XFS file system before changing parameters with `xfs_admin`.
- You can change the file system label and the UUID:
  - File system label (`-L <new_label>`)
  - File system UUID (`-U <new_UUID>`)
- You can enable or disable XFS lazy counters.
  - To enable lazy counters (`-c 1`)
  - To disable lazy counters (`-c 0`)
- XFS enables lazy counters by default.
- Lazy counters improve performance by not requiring superblock updates when other counters are changed.



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

Use the `xfs_admin` command to change the parameters of an XFS file system. You can also use the `xfs_admin` command to view the file system label and UUID:

```
# xfs_admin -lu /dev/xvdb1
label = "XFS"
UUID = ...
```

You must unmount the file system before changing parameters with the `xfs_admin` command. With the file system unmounted, you can change the following parameters:

- `-L <label>`: Use this option to change the file system label.
- `-U <UUID>`: Use this option to change the file system UUID.

You can also use the `xfs_admin` command to enable or disable lazy counters. With lazy counters enabled, the superblock is not modified or logged when changes are made to the free-space and inode counters. Information is stored in other parts of the file system to maintain the counter values. This provides significant performance improvements in some configurations. Enabling and disabling lazy counters is time-consuming on large file systems because the entire file system must be scanned. To enable and disable lazy counters:

- `-c 1`: Enables lazy counters
- `-c 0`: Disables lazy counters

For more information, see the `xfs_admin(8)` manual page.

# Enabling Disk Quotas on an XFS File System

- XFS supports quotas by user, by group, and by project.
  - Project quotas set limits on directory hierarchies.
- Limit disk space (blocks) and/or the number of files (inodes).
  - Hard limits and soft limits on blocks and inodes
- Enable quotas by using XFS file system mount options:
  - `quota|uquota|usrquota`: Enable user quotas and enforce usage limits.
  - `gquota|grpquota`: Enable group quotas and enforce usage limits.
  - `pquota|prjquota`: Enable project quotas and enforce usage limits.
  - `uqnoenforce|gqnoenforce|pqnoenforce`: Enable quotas but do not enforce usage limits.



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

XFS supports disk quotas by user, by group, and by project. Project disk quotas allow you to limit the amount of disk space on individual directory hierarchies. You can configure both hard and soft limits on the number of disk blocks (or disk space), and the number of inodes, which limit the number of files a user can create. Quotas do not apply to the `root` user.

You must first enable quotas for users, groups, and/or projects by using a mount option when mounting for the XFS file system. After enabling quotas, use the `xfs_quota` command to set limits to view quota information.

## Enabling Quotas

To enable quotas for users on an XFS file system, include the `quota` option in the `/etc/fstab` entry for the file system, or mount the file system with the `quota` option:

```
# mount -o quota /dev/xvdb1 /xfs
```

To enable quotas for groups, include the `gquota` option in the `/etc/fstab` entry for the file system, or mount the file system with the `gquota` option:

```
# mount -o gquota /dev/xvdb1 /xfs
```

To enable quotas for projects, include the `prjquota` option in the `/etc/fstab` entry for the file system, or mount the file system with the `prjquota` option:

```
# mount -o prjquota /dev/xvdb1 /xfs
```

Alternatively, you can include the quota mount options in the `/etc/fstab` file. The following example shows entries in the `/etc/fstab` file to enable quotas for users, groups, and projects, respectively, on an XFS file system. These examples also mount the file system with read/write permissions:

```
/dev/xvdb1    /xfs    xfs    rw,quota    0  0
/dev/xvdb1    /xfs    xfs    rw,gquota   0  0
/dev/xvdb1    /xfs    xfs    rw,prjquota  0  0
```

## XFS Quota Mount Options

Other “quota” mount options for XFS file systems are available. The following is a complete list of mount options to enable user quotas on XFS file systems:

- `quota|uquota|usrquota`: Enable user quotas and enforce usage limits.
- `uqnoenforce`: Enable user quotas. Report usage but do not enforce usage limits.

Group quota mount options include:

- `gquota|grpquota`: Enable group quotas and enforce usage limits.
- `gqnoenforce`: Enable group quotas. Report usage but do not enforce usage limits.

Project quota mount options include:

- `pquota|prjquota`: Enable project quotas and enforce usage limits.
- `pqnoenforce`: Enable project quotas. Report usage but do not enforce usage limits.

## Reporting Quota State Information

You can use the following `xfs_quota` command to report the overall quota state information:

```
# xfs_quota -x -c state
User quota state on /xfs (/dev/xvdb1)
  Accounting: ON
  Enforcement: ON
  Inode: #37 (1 blocks, 1 extents)
Group quota state on /xfs (/dev/xvdb1)
  Accounting: OFF
  Enforcement: OFF
  Inode: N/A
Project quota state on /xfs (/dev/xvdb1)
  Accounting: OFF
  Enforcement: OFF
  Inode: N/A
  Blocks grace time: [7 days 00:00:30]
  Inodes grace time: [7 days 00:00:30]
  Realtime Blocks grace time: [7 days 00:00:30]
```

This command reports whether user, group, and project disk quota accounting is enabled and whether limits are being enforced. The grace period for blocks and inodes is also reported. The timer for the grace period is enabled whenever the soft limit is exceeded. If soft limits continue to be exceeded after the grace period expires, no more disk space or inodes are allocated.

## xfs\_quota Utility

- Use to report file system quota information and perform quota management operations on XFS file systems.
  - Set block and inode limits.
  - Enable or disable quota enforcement.
  - Modify the quota enforcement timeout information.
- It includes an interactive interface:

```
# xfs_quota  
xfs_quota>
```

- Include **-x** and **-c** options to modify and report quota information from the command line:

```
# xfs_quota -x -c 'limit -u bsoft=5m bhard=6m john' /xfs  
# xfs_quota -x -c state
```



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

After enabling quotas by using XFS file system mount options, use the `xfs_quota` command to report file system quota information, set block and inode limits, enable or disable quota enforcement, modify the quota enforcement timeout information (grace period), and perform other quota management operations on XFS file systems.

The `xfs_quota` utility provides a number of user and administrator subcommands. These subcommands can be issued in interactive mode or included as arguments to the `xfs_quota` command. Enter `xfs_quota` without any options or arguments to enter interactive mode. An `xfs_quota>` prompt appears in interactive mode. Enter `help` or `?` to view the available subcommands. You can also enter `help commandname` to display additional information on a specific subcommand.

```
# xfs_quota  
xfs_quota>
```

When including `xfs_quota` subcommands from the command line, use the `-c <command>` option. Any modifications to the quota system from the command line also require the `-x` option (enable expert mode). The following example uses the `limit` subcommand to set a soft limit of 5 MB, and a hard limit of 6 MB on the XFS file system for user `john`:

```
# xfs_quota -x -c 'limit -u bsoft=5m bhard=6m john' /xfs
```

The following example sets a soft limit of 100 inodes and a hard limit of 150 inodes for the `students` group:

```
# xfs_quota -x -c 'limit -g isoft=100 ihard=150 students' /xfs
```

## Displaying Quota Information

Use the `xfs_quota` command to display information about disk quotas. To list all paths with devices and identifiers:

```
# xfs_quota -x -c print
Filesystem      Pathname
/xfs           /dev/xvdb1 (uquota)
```

To report file system usage for blocks (-b) and inodes (-i):

```
# xfs_quota -x -c 'free -hb'
Filesystem  Size   Used   Avail  Use%  Pathname
/dev/xvdb1  1.0G  12.4M  1013.1M  1%   /xfs
# xfs_quota -x -c 'free -hi'
Filesystem  Inodes  Used   Free  Use%  Pathname
/dev/xvdb1    1.1m     7   1.1m   0%   /xfs
```

To report file system quota information:

```
# xfs_quota -x -c report
User quota on /xfs (/dev/xvdb1)
          Blocks
User ID    Used   Soft   Hard   Warn/Grace
-----
root      8513     0     0     00 [-----]
oracle      0   4096   5120     00 [-----]
```

To report quota information in human-readable form on /xfs:

```
# xfs_quota -x -c 'report -h' /xfs
```

For more information, see the `xfs_quota(8)` manual page.

## Setting Project Quotas

- XFS allows you to set quotas on directory hierarchies.
- Project quotas are initially enabled by using a mount option:
  - pquota or prjquota or pqnoenforce
- Associate a unique project ID with an XFS directory hierarchy in the /etc/projects file. Example:
  - 50:/xfs
- Associate a project name to the project ID in the /etc/projid file. Example:
  - test:50
- Use the project name when setting limits. Example:

```
# xfs_quota -x -c 'limit -p bsoft=5m bhard=6m test' /xfs
```



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

XFS allows you to set quotas on individual directory hierarchies. You can create an entry in the /etc/projects file that associates the XFS file system directory hierarchy with a unique project ID. For example, the following entry in /etc/projects associates a project ID of 50 with the /xfs directory:

50:/xfs

You can optionally use the /etc/projid file to associate a project name to a project ID. For example, the following entry in the /etc/projid file associates project name test with project ID 50:

test:50

After defining a project in /etc/projects, use the xfs\_quota command to initialize its project directory:

```
# xfs_quota -x -c 'project -s test' /xfs
```

Use the xfs\_quota command to set limits for projects with initialized directories. The following example sets a soft limit of 5 MB and a hard limit of 6 MB for the test project:

```
# xfs_quota -x -c 'limit -p bsoft=5m bhard=6m test' /xfs
```

For more information, see the `projects(5)`, `projid(5)`, and `xfs_quota(8)` manual pages.

# Backing Up and Restoring XFS File Systems

- Use `xfsdump` to back up an XFS file system.
  - You can back up entire XFS file systems or selected files and directories from an XFS file system.
- Use `xfsrestore` to restore files to an XFS file system.
  - You can restore entire XFS file systems or selected files and directories to an XFS file system.
- To perform a full (level 0) backup of the XFS file system mounted on `/xfs` to a local SCSI tape device, `/dev/st0`:

```
# xfsdump -l 0 -f /dev/st0 /xfs
```
- To restore from a backup that was written to the `/usr/tmp/backup` file to the `/xfs` directory:

```
# xfsrestore -f /usr/tmp/backup /xfs
```



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

Use the `xfsdump` and `xfsrestore` utilities to back up and restore files in an XFS file system. You can back up files to directly attached tape drives or hard drives, or to remote drives that are accessible over the network. You can back up an entire XFS file system, only the files that have changed since a previous backup, or selected directories or files.

You can restore all files from a full or incremental backup, or selected files and directories. You can restore data to its original location or to another location within an XFS file system. The `xfsrestore` utility can also be run interactively, allowing you to select files that you want to restore.

## Using `xfsdump`

Use the `-l <level>` option to specify a full or incremental backup. Level 0 is a full backup of an entire XFS file system. Levels 1-9 are incremental backups that back up all files that have changed since a backup with a lower level number. The following example performs a level 0 backup of the XFS file system mounted on `/xfs` to a local SCSI tape device, `/dev/st0`. The `-L <session_label>` option allows you to assign a label to the backup.

```
# xfsdump -l 0 -L "Level 0 backup of /xfs" -f /dev/st0 /xfs
```

Backups can span multiple tape media if necessary. If the end of the tape media is reached before the backup is complete, `xfsdump` prompts you to insert additional media. Multiple backups can also be stored on the same media. The tape is automatically advanced to the end of any existing backups before beginning a new backup.

The following example performs a level 1 backup to a tape device attached to a remote system (host01). Use a colon between the remote host name (or IP address) and the tape device.

```
# xfsdump -l 1 -f host01:/dev/st0 /xfs
```

You can also use `xfsdump` to back up data to a file instead of a tape device. The following example performs a full backup (level 0) of the XFS file system mounted on `/xfs` to a local file, `/usr/tmp/full_Monday`. Note that if a level is not specified, a full backup is performed.

```
# xfsdump -f /usr/tmp/full_Monday /xfs
```

Use the `-s` option to back up specific files or directories in an XFS file system. The following example backs up `file` and `directory` to a file on a remote host, `host01:/usr/tmp/back`. Both `file` and `directory` are located in the XFS file system mounted on `/xfs`.

```
# xfsdump -f host01:/usr/tmp/back -s file -s directory /xfs
```

### Examining `xfsdump` Inventory

The `xfsdump` utility keeps an inventory in the `/var/lib/xfsdump` directory of all backups. You can examine the inventory contents by using the `-I` option.

```
# xfsdump -I
```

The inventory records are in sequential order and indented for readability and to emphasize the hierarchical nature of the `xfsdump` information.

### Using `xfsrestore`

The following example restores an `xfsdump` from a SCSI tape device to an XFS file system mounted on `/xfs`.

```
# xfsrestore -f /dev/st0 /xfs
```

The following example restores the contents of an `xfsdump` that was written to the `/usr/tmp/backup` file to the `/xfs` directory.

```
# xfsrestore -f /usr/tmp/backup /xfs
```

You can perform cumulative restores from tape media that contains full (level 0) and incremental backups. Contents of the level 0 `xfsdump` are restored first, then contents are restored from the next higher level, and so forth until all incremental backups are restored. Use the `-r` option to perform a cumulative restore.

The following example performs a cumulative restore from `xfsdump` backups on a SCSI tape device to an XFS file system mounted on `/xfs_restore`.

```
# xfsrestore -f /dev/st0 -r /xfs_restore
```

A cumulative restore creates an `xfsrestorehousekeepingdir` directory in the directory that is restored. Files in this directory pass information from one execution of `xfsrestore` to the next. This directory can be removed after the cumulative restore is complete.

For more information, see the `xfsdump` (8) and `xfsrestore` (8) manual pages.

## XFS File System Maintenance

- `xfs_fsr`: Improve performance of an XFS file system by reorganizing and improving the layout of the file extents.
- `xfs_repair`: Repair a corrupted or damaged XFS file system. Unmount the file system before running this command.
- `xfs_db`: Debug an XFS file system. This utility provides a command set that allows you to perform scans on the file system, and to navigate and display its data structures.



ORACLE®

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

Additional XFS utilities are available to perform file system maintenance. These utilities include the following:

- `xfs_fsr`: XFS is an extent-based file system. The `xfs_fsr` utility reorganizes and improves the layout of the file extents, which improves overall performance. Run this command on a mounted XFS file system or on individual files in the file system.
- `xfs_repair`: Repair a corrupted or damaged XFS file system. Unmount the file system before running this command. If the file system cannot be repaired, restore files from a backup with `xfsrestore`.
- `xfs_db`: Debug an XFS file system. This utility provides a command set that allows you to perform scans on the file system, and to navigate and display its data structures.

For more information, see the `xfs_fsr(8)`, `xfs_repair(8)`, and `xfs_db(8)` manual pages.



## Quiz

Which of the following statements are true?

- a. XFS in Oracle Linux is available for the x86\_64 architecture.
- b. XFS in Oracle Linux requires the Unbreakable Enterprise Kernel Release 3 (3.8.13).
- c. XFS is supported for use with the root (/) and /boot file systems.
- d. All of the above



**ORACLE®**

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



## Quiz

Which of the following statements are true?

- a. XFS supports quotas by user, by group, and by project.
- b. Disk quotas are enabled by using XFS file system mount options.
- c. Use the `xfs_admin` command to set limits on disk space and the number of files.
- d. Use the `xfs_quota` command to report file system quota information.
- e. All of the above



**ORACLE®**

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



## Quiz

Which of the following statements are true?

- a. The `xfsdump` and `xfsrestore` utilities can be used with remote storage devices.
- b. The `xfsdump` and `xfsrestore` utilities support full and incremental backups and restores.
- c. The `xfsdump` and `xfsrestore` utilities allow backups and restores of individual files on an XFS file system.
- d. All of the above



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe XFS for Oracle Linux
- Create an XFS file system
- Use the `xfs_growfs` utility
- Use the `xfs_admin` utility
- Enable disk quotas on an XFS file system
- Use the `xfs_quota` utility
- Set project quotas
- Use the `xfsdump` and `xfsrestore` utilities
- Use XFS file system maintenance utilities



ORACLE®

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

## Practice 10: Overview

This practice covers the following topics:

- Creating an XFS file system
- Setting disk quotas on an XFS file system
- Backing up and restoring XFS file systems



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

# Implementing the Btrfs File System

The Oracle logo, consisting of the word "ORACLE" in white capital letters inside a red rectangular box.

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

## Objectives

After completing this lesson, you should be able to:

- Describe the features of the Btrfs file system
- Create a Btrfs file system
- Create Btrfs subvolumes and snapshots
- Take a snapshot of a file in a Btrfs subvolume
- Mount Btrfs subvolumes and snapshots
- Defragment and resize a Btrfs file system
- Add and remove devices in a Btrfs file system
- Check and repair the integrity of a Btrfs file system
- Convert ext file systems to Btrfs



ORACLE®

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

## Btrfs: Introduction

- Development began at Oracle
  - Now jointly developed by several companies
- Extent-based file storage
- All data and metadata written via copy-on-write
- Readable and writable snapshots
- Integrated volume management and RAID capabilities
- CRCs for all metadata and data
- Online resizing and defragmentation
- Transparent compression
- Efficient storage for small files
- SSD optimizations and TRIM support



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



Btrfs is an open-source, general-purpose file system for Linux. The name is derived from the use of B-trees to store internal file system structures. Different names are used for the file system, including “Butter F S” and “B-tree F S.” Development of Btrfs began at Oracle in 2007, and now several companies (including Red Hat, Fujitsu, Intel, SUSE, and many others) are contributing to the development effort. Btrfs is included in the mainline Linux kernel.

Btrfs provides extent-based file storage supporting large file and file system sizes. All data and metadata is copy-on-write, meaning blocks of data are not changed on disk. Btrfs just copies the blocks and then writes out the copies to a different location. Not updating the original location eliminates the risk of a partial update or data corruption during a power failure. The copy-on-write nature of Btrfs also facilitates file system features such as replication, migration, backup, and restoration of data.

Btrfs allows you to create both readable and writable snapshots. A snapshot is a copy of an entire Btrfs subvolume taken at a given point in time. The snapshots appear as normal directories, and you can access the snapshot as you would any other directory. Writable snapshots allow you to roll back a file system to a previous state. You can take a snapshot, perform a system upgrade, and reboot into the snapshot if the upgrade causes problems. All snapshots are writable by default, but you also have the option to create read-only snapshots.

See the following Oracle guidelines for file and file system limits:

[https://docs.oracle.com/cd/E52668\\_01/E93593/html/oI7-system-requirements-limits.html](https://docs.oracle.com/cd/E52668_01/E93593/html/oI7-system-requirements-limits.html)

Btrfs allows a file system to span multiple devices. This is different from the logical volume management (LVM) style of volume management. Btrfs does not create block devices; it just creates subvolumes in the file system that can then be mounted like a regular file system.

Btrfs also has built-in RAID support for RAID-0, RAID-1, and RAID-10 levels. RAID-5 and RAID-6 features are implemented, but these RAID levels are not recommended for production use. Btrfs's RAID is not a multi-disk RAID like the software RAID devices created by using the `mdadm` command. It is not block RAID either because it does not mirror block devices. Btrfs's RAID just ensures that for every block, there are "x" number of copies. For RAID-1, for example, Btrfs just stores two copies of everything on two different devices.

Btrfs maintains CRCs for all metadata and data so everything is checksummed to preserve the integrity of data against corruption. With a RAID-1 or RAID-10 configuration, if checksum fails on the first read, data is taken from another copy.

Btrfs has online resizing and defragmentation. You can add or remove devices while the file systems remain online. When a device is removed, the extents stored on it are redistributed to the other devices in the file system. You can also replace devices while Btrfs is online. Btrfs rebalances the extents across the new disk, and then you can drop the old disk from a Btrfs array.

Btrfs has transparent compression and supports two compression methods: zlib and LZO (the default). LZO offers a better compression ratio, whereas zlib offers faster compression. Btrfs only compresses blocks when it determines it is possible. You enable compression and specify the compression method by using a `mount` option. For example, to enable LZO or zlib compression:

```
# mount -o compress=lzo|zlib <device> <mount_point>
```

You can also force Btrfs to always compress data:

```
# mount -o compress-force <device> <mount_point>
```

Btrfs provides efficient storage for small files. All Linux file systems address storage in block sizes (for example, 4 KB). With other file systems, a file that is smaller than 4 KB wastes the leftover space. Btrfs stores these smaller files directly into the metadata, thereby providing a significant performance advantage over other file systems when creating and reading small files.

Btrfs automatically detects solid state drives (SSD) and turns off all optimizations for rotational media. For example, on spinning disks, it is important to store related data close together to reduce seeking. This requires CPU cycles to get good data locality on spinning disks, which is not as important with SSD. TRIM support is also an optimization for SSD. It tells the SSD which blocks are no longer needed and are available to be written over.

## Btrfs with Oracle Linux

- Btrfs has been production-ready for Oracle Linux since UEK R2.
- Btrfs has been a technology preview with the RHCK.
  - Deprecated in the RHCK as of Oracle Linux 7 Update 4
  - Continues to be fully supported by Oracle in UEK R4 and UEK R5
- Refer to the Oracle Linux 7 and UEK release notes at [http://docs.oracle.com/cd/E52668\\_01/index.html](http://docs.oracle.com/cd/E52668_01/index.html).



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

## Creating a Btrfs File System

- Btrfs utilities are provided by the `btrfs-progs` software package:

```
# rpm -qf btrfs-progs
```

- Use the `mkfs.btrfs` command to create a file system:

```
mkfs.btrfs [options] block_device [block_device ...]
```

- To create a Btrfs file system across two devices:

```
# mkfs.btrfs /dev/xvdb /dev/xvdd
```

- Mount the Btrfs file system by using the `mount` command, referencing either device:

```
# mount /dev/xvdb /btrfs
```



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

The Btrfs utilities are provided by the `btrfs-progs` software package. Use the following command to list the files provided by the package:

```
# rpm -qf btrfs-progs
```

Use the `mkfs.btrfs` command to create a Btrfs file system. The syntax is:

```
mkfs.btrfs [options] block_device [block_device ...]
```

You can create a Btrfs file system on a single device or on multiple devices. Devices can be disk partitions, loopback devices (disk images in memory), multipath devices, or LUNs that implement RAID in hardware.

Some of the available options for the `mkfs.btrfs` command are:

- A offset**: Specify the offset from the start of the device for the file system. The default is 0, which is the start of the device.
- b size**: Specify the size of the file system. The default is all the available storage.
- d type**: Specify how the file system data is spanned across the devices. The `type` argument must be `raid0`, `raid1`, `raid10`, or `single`.
- l size**: Specify the leaf size, the least data item in which Btrfs stores data. The default is the page size. This is the same as `-n|--nodesize` and has been deprecated.
- L name**: Specify a label name for the file system.

- **-m profile:** Specify how the file system metadata is spanned across the devices. The *profile* argument must be `raid0`, `raid1`, `raid10`, `single`, or `dup`.
- **-M:** Mix data and metadata chunks together for more efficient space utilization. This option affects performance for larger file systems and is recommended only for file systems that are 1 GB or smaller.
- **-n size:** Specify the node size. The default is the page size.
- **-s size:** Specify the sector size, which is the minimum block allocation.
- **-v:** Print the `mkfs.btrfs` version and exit.

### **`mkfs.btrfs`: Examples**

To create a Btrfs file system on a single block device (for example, `/dev/xvdb`):

```
# mkfs.btrfs /dev/xvdb
```

To create a Btrfs file system on two block devices (for example, `/dev/xvdb` and `/dev/xvdd`):

```
# mkfs.btrfs /dev/xvdb /dev/xvdd
```

The default configuration for a file system with multiple devices is:

- **-d raid0:** Stripe the file system data across all devices.
- **-m raid1:** Mirror the file system metadata across all devices.

To create a Btrfs file system with multiple devices (`/dev/xvdb` and `/dev/xvdd`) and stripe both the data and the metadata:

```
# mkfs.btrfs -m raid0 /dev/xvdb /dev/xvdd
```

To create a Btrfs file system with multiple devices (`/dev/xvdb` and `/dev/xvdd`) and mirror both the data and the metadata:

```
# mkfs.btrfs -d raid1 /dev/xvdb /dev/xvdd
```

When you specify a single device, metadata is duplicated on that device unless you specify only a single copy. To create a Btrfs file system on a single block device (for example, `/dev/xvdb`) and to specify not to duplicate the metadata:

```
# mkfs.btrfs -m single /dev/xvdb
```

For RAID-10 data or metadata, you must specify an even number of at least four devices. To create a Btrfs file system and stripe the data and metadata across mirrored devices (RAID-10):

```
# mkfs.btrfs -d raid10 -m raid10 /dev/xvd[bcde]
```

### **Mounting the File System**

Use the `mount` command or make an entry in `/etc/fstab` as you would when mounting any other type of Linux file system. You can reference either device when your file system contains multiple devices. You can also reference the file system label or the UUID.

Example:

```
# mount /dev/xvdb /btrfs
```

## The btrfs Utility

- The btrfs utility requires a subcommand:

```
# btrfs
usage: btrfs [--help] [--version] <group> [<group>...] <command> [<args>]
...
```

- Some available subcommands include:

- subvolume
- filesystem
- device | replace
- scrub
- check | rescue | restore
- inspect-internal
- send | receive
- quota | qgroup

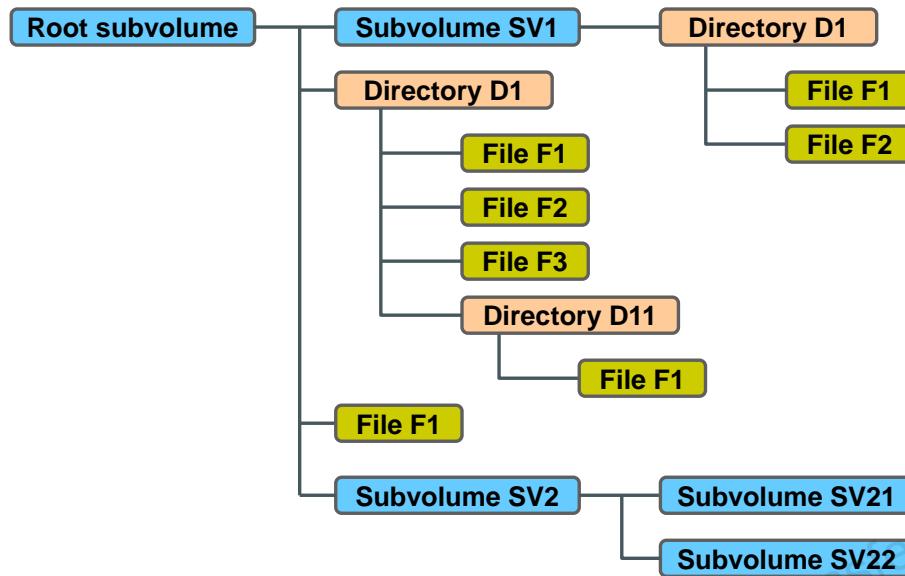


Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

Use the btrfs command to manage and display information about a Btrfs file system. The command requires a subcommand. Enter btrfs without any arguments to list the subcommands:

```
# btrfs
Usage: btrfs [--help] [--version] <group> [<group>...] <command> [<args>]
btrfs subvolume create [-i <qgroupid>] [<dest>/]<name>
    Create a subvolume
btrfs subvolume delete <subvolume> [<subvolume>...]
    Delete subvolume(s)
...
btrfs filesystem df <path>
    Show space usage information for a mount point
btrfs filesystem show [--all-devices] [<uuid>|<label>]
    Show the structure of a filesystem
...
```

## Btrfs Subvolumes



ORACLE®

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

This slide illustrates a Btrfs file system hierarchy that consists of subvolumes, directories, and files. Btrfs subvolumes are named B-trees that hold files and directories. Subvolumes can also contain subvolumes, which are themselves named B-trees that can also hold files and directories. The top level of a Btrfs file system is also a subvolume and is known as the *root subvolume*.

The root subvolume is mounted by default, and Btrfs subvolumes appear as regular directories within the file system. However, a subvolume can be mounted, and only files and directories in the subvolume are accessible. The following example lists the hierarchy displayed in the slide, with the default root subvolume mounted on /btrfs:

```
# ls -l /btrfs
drwxr-xr-x ... SV1
drwxr-xr-x ... D1
-rw-r--r-- ... F1
drwxr-xr-x ... SV2
```

Mounting the SV1 subvolume or the SV2 subvolume on /btrfs allows access only to the files and directories within the respective subvolumes. Remount the root subvolume to gain access to the entire hierarchy.

Use the **btrfs subvolume** command to manage and report on Btrfs subvolumes. A list of the available subvolume commands is as follows:

```
# btrfs subvolume
usage: btrfs subvolume <command> <args>
btrfs subvolume create [-i <qgroupid>] [<dest>/]<name>
        Create a subvolume
btrfs subvolume delete <subvolume> [<subvolume>...]
        Delete a subvolume(s)
btrfs subvolume list [options] [-G [+|-]value] [-C [+|-]value] [--sort=gen,ogen,rootid,path] <path>
        List subvolumes (and snapshots)
btrfs subvolume snapshot [-r] <source> <dest>| [<dest>/]<name>
btrfs subvolume snapshot [-r] [-i <qgroupid>] <source>
<dest>| [<dest>/]<name>
btrfs subvolume get-default <path>
        Get the default subvolume of a filesystem
btrfs subvolume set-default <subvolid> <path>
        Set the default subvolume of a filesystem
btrfs subvolume find-new <path> <lastgen>
        List the recently modified files in a filesystem
btrfs subvolume show <subvol-path>
        Show more information of the subvolume
btrfs subvolume sync <path> [<subvol-id>...]
        Wait until given subvolume(s) are completely removed from the
filesystem.

        manage subvolumes: create, delete, list, etc
```

The word “subvolume” in the **btrfs** command can be abbreviated to “sub”. For example, both the following commands are valid:

```
# btrfs subvolume create /btrfs/SV1
# btrfs sub create /btrfs/SV1
```

The abbreviation applies to other **btrfs** subcommands as well. For example, both the following subcommands are valid:

```
# btrfs filesystem df /btrfs
# btrfs file df /btrfs
```

## btrfs subvolume Utilities

- Use the `btrfs subvolume create` command to create a subvolume on a mounted Btrfs file system, such as:

```
# btrfs subvolume create /btrfs/SV1
```

- The subvolume appears as a normal directory when the `ls` command is used (only a partial output is shown):

```
# ls -l /btrfs  
drwxr-xr-x ... SV1
```

- Use the `btrfs subvolume list` command to view the subvolumes in a Btrfs file system, as in this example:

```
# btrfs subvolume list /btrfs  
ID 258 gen 10 top level 5 path SV1
```



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

Use the `btrfs subvolume create` command to create a subvolume. The following example creates a subvolume named `SV1` on a Btrfs file system mounted on `/btrfs`:

```
# btrfs subvolume create /btrfs/SV1  
Create subvolume '/btrfs/SV1'
```

The subvolume appears as a regular directory. The following example creates a regular directory in `/btrfs` and then displays the content:

```
# mkdir /btrfs/D1  
# ls -l /btrfs  
drwxr-xr-x ... D1  
drwxr-xr-x ... SV1
```

Use the `btrfs subvolume list` command to view only the subvolumes in a Btrfs file system, as in this example:

```
# btrfs subvolume list /btrfs  
ID 258 gen 10 top level 5 path SV1
```

This command also displays the subvolume ID (258), root ID generation of the B-tree (10), and the top-level ID (5). These fields are described later in this lesson.

## Btrfs Snapshots

- A snapshot is a point-in-time copy of a subvolume.
- Snapshots are created quickly and initially consume very little disk space.
- Use the `btrfs subvolume snapshot` command to create a snapshot of a subvolume.
- The following example creates a writable/readable snapshot named `SV1-snap` of the `SV1` subvolume:

```
# btrfs subvolume snapshot /btrfs/SV1 /btrfs/SV1-snap
```

- Use the `-r` option to create a read-only snapshot:

```
# btrfs subvolume snapshot -r /btrfs/SV1 /btrfs/SV1-rosnap
```



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

Btrfs subvolumes can be snapshotted and cloned, which creates additional B-trees. A snapshot starts as a copy of a subvolume taken at a point in time. You can make a snapshot writable and use it as an evolving clone of the original subvolume. Or you can use the snapshot as a stable image of a subvolume for backup purposes or for migration to other systems. Snapshots can be created quickly and initially consume very little disk space.

Use the `btrfs subvolume snapshot` command to create a writable/readable snapshot of a subvolume. The following example creates a snapshot of the `SV1` subvolume:

```
# btrfs subvolume snapshot /btrfs/SV1 /btrfs/SV1-snap
```

Create a snapshot of '/btrfs/SV1' in '/btrfs/SV1-snap'

Use the `btrfs subvolume snapshot -r` option to create a read-only snapshot:

```
# btrfs subvolume snapshot -r /btrfs/SV1 /btrfs/SV1-rosnap
```

Create a readonly snapshot of '/btrfs/SV1' in '/btrfs/SV1-rosnap'

The snapshots appear as a regular directory when the `ls` command is used. Snapshots also appear in the output of the `btrfs subvolume list` command.

## Taking a Snapshot of a File

- Use the `cp --reflink` command to take a snapshot of a file.
- A new file shares the same disk blocks as the original file.
- The copy operation is almost instantaneous and also saves disk space.
- This operation works only within the boundaries of the same Btrfs file system and within the same subvolume.
- Example:

```
# cp --reflink /btrfs/SV1/file /btrfs/SV1/copy_of_file
```



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

You can use the `cp --reflink` command to take a snapshot of a file. With this option, the file system does not create a new link pointing to an existing inode, but instead creates a new inode that shares the same disk blocks as the original copy. The new file appears to be a copy of the original file, but the data blocks are not duplicated. This allows the copy to be almost instantaneous and also saves disk space. As the file's content diverges over time, its amount of required storage grows. One restriction is that this operation can work only within the boundaries of the same file system and within the same subvolume.

The following example copies a file by using the `cp --reflink` command. The space used is given both before and after the copy operation after a sync is run. Note that the space used does not increase.

```
# sync
# df -h /btrfs
Filesystem      Size   Used   Avail   Use%   Mounted on
/dev/xvdb        5.0G   22M    4.8G    1%    /btrfs
# cp --reflink /btrfs/SV1/vmlinuz* /btrfs/SV1/copy_of_vmlinuz
# sync
# df -h /btrfs
Filesystem      Size   Used   Avail   Use%   Mounted on
/dev/xvdb        5.0G   22M    4.8G    1%    /btrfs
```

## Mounting a Subvolume or Snapshot

- To mount a subvolume or snapshot, you must first determine the ID number.
- Use the `btrfs subvolume list` command to display the ID numbers, as in this example:

```
# btrfs subvolume list /btrfs
ID 259 gen 26 top level 5 path SV1
ID 263 gen 23 top level 5 path SV1-snap
```

- Use the `btrfs subvolume set-default` command to change the ID number to the entity to be mounted:

```
# btrfs subvolume set-default 259 /btrfs
```

- Unmount and remount the file system.
- Alternatively, use `-o subvolid=#` when mounting the file system, but this does not change the default subvolume ID.



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

By default, Linux mounts the parent Btrfs volume, which has an ID of 0. In this example, the following `mount` command was issued before creating any subvolumes and snapshots:

```
# mount /dev/xvdb /btrfs
```

The subvolume `SV1` was created in `/btrfs`. The `ls` command shows the subvolume:

```
# ls -l /btrfs
drwxr-xr-x ... SV1
```

The following example copies files into `SV1`, creates a snapshot of `SV1`, and verifies that both the subvolume and the snapshot contain the same files:

```
# cp /boot/vmlinuz-3.10.0-693* /btrfs/SV1
# btrfs sub snapshot /btrfs/SV1 /btrfs/SV1-snap
# ls /btrfs/SV1*
/btrfs/SV1:
vmlinuz-3.10.0-693.el7.x86_64
/btrfs/SV1-snap:
vmlinuz-3.10.0-693.el7uek.x86_64
```

If you unmount /btrfs and remount it, the parent Btrfs volume is mounted by default:

```
# ls /btrfs
SV1  SV1-snap
# umount /btrfs
# mount /dev/xvdb /btrfs
# ls /btrfs
SV1  SV1-snap
```

You can, however, mount a btrfs subvolume or snapshot as though it were a disk device. If you mount a snapshot instead of its parent subvolume, you effectively roll back the state of the file system to the time that the snapshot was taken.

The following example copies a file to SV1 so that the content is different from the SV1-snap:

```
# cp ~/test-file /btrfs/SV1
# ls /btrfs/SV1*
/btrfs/SV1:
test-file      vmlinuz-3.10.0-693.el7uek.x86_64
/btrfs/SV1-snap:
vmlinuz-3.10.0-693.el7uek.x86_64
```

To mount a subvolume or snapshot, you must first determine the ID number of the subvolume that you want to mount. Use the btrfs subvolume list command to display the ID numbers. In the following example, the ID of the root subvolume is 5:

```
# btrfs subvolume list /btrfs
ID 259 gen 26 top level 5 path SV1
ID 263 gen 23 top level 5 path SV1-snap
```

Use the btrfs subvolume set-default command to set the default subvolume of a file system. For example, to mount the SV1 Btrfs subvolume, which has an ID of 259:

```
# btrfs subvolume set-default 259 /btrfs
```

You then need to unmount and remount the Btrfs file system. The root level then contains the contents of the SV1 subvolume, and the root subvolume is no longer visible:

```
# umount /btrfs
# mount /dev/xvdb /btrfs
# ls /btrfs
test-file      vmlinuz-3.10.0-693.el7uek.x86_64
```

You can also use the -o subvolid option to the mount command to mount the root subvolume or a subvolume or snapshot. For example, to mount the root subvolume:

```
# umount /btrfs
# mount -o subvolid=5 /dev/xvdb /btrfs
# ls /btrfs
SV1  SV1-snap
```

## btrfs filesystem Utilities

- Use the `btrfs filesystem` command to manage and report on Btrfs file systems.
- Available commands include:
  - `btrfs filesystem df`
  - `btrfs filesystem show`
  - `btrfs filesystem sync`
  - `btrfs filesystem defragment`
  - `btrfs filesystem resize`
  - `btrfs filesystem balance`
  - `btrfs filesystem label`
- For example, to display the file system label:

```
# btrfs filesystem label /btrfs
Btrfs
```



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

Use the `btrfs filesystem` command to manage and report on Btrfs file systems. A partial list of the available commands is as follows:

```
# btrfs filesystem
usage: btrfs filesystem [<group>] <command> [<args>]

        btrfs filesystem df [options] <path>
            Show space usage information for a mount point
        btrfs filesystem du [options] <path> [<path>..]
            Summarize disk usage of each file.
        btrfs filesystem show [options] [<path>|<uuid>|<device>|label]
            Show the structure of a filesystem
        btrfs filesystem sync <path>
            Force a sync on a filesystem
        ...
```

## The btrfs filesystem df Utility

- Use the `btrfs filesystem df` command to show accurate space usage information for a mount point.
- The following example is for a RAID-1 Btrfs file system created with two 5 GB disks:

```
# btrfs filesystem df /btrfs
Data, RAID1: total=1.00GiB, used=6.11MiB
System, RAID1: total=8.00MiB, used=16.00KiB
Metadata, RAID1: total=1.00GiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

- Btrfs allocates space on disks in chunks.
  - A chunk is 1 GB for data and 256 MB for metadata.
  - A chunk also has a specific RAID profile associated with it.



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

Some information is presented when you create a Btrfs file system. The following example creates a Btrfs file system with two 5 GB devices (`/dev/xvdb` and `/dev/xvdd`) and mirrors both the data and the metadata (metadata is mirrored by default):

```
# mkfs.btrfs -L Btrfs -d raid1 /dev/xvdb /dev/xvdd
btrfs-progs v4.9.1
...
Node size:          16384
Sector size:        4096
Filesystem size:    10.00GiB
...
```

The preceding output shows that the block size is 4 KB with a total of 10 GiB of space. But because the array is RAID-1, you can fit only 5 GB of data on this file system. You actually have less than 5 GB because space is needed for the metadata as well. The example continues with creating a mount point and mounting the file system:

```
# mkdir /btrfs
# mount /dev/xvdb /btrfs
```

As previously discussed, you can mount by referencing either device in the array, the LABEL, or the UUID.

Even the `/proc/mounts` file does not show the second device for the Btrfs file system:

```
# grep btrfs /proc/mounts  
/dev/xvdb /btrfs btrfs rw,seclabel,relatime,ssd,space_cache 0 0
```

For example, the following command copies a file to the Btrfs file system:

```
# cd /btrfs  
# cp /boot/vmlinuz-3.10* .  
# ls -l  
-rwxr-xr-x ... vmlinuz-3.10...
```

When the file system is mounted and has a file copied to it, the output of the `df` command produces inaccurate information for the Btrfs file system:

```
# sync  
# df -h .  
Filesystem Size Used Avail Use% Mounted on  
/dev/xvdb 5.0G 23M 4.0G 1% /btrfs
```

This output shows that the file system has a size of 5.0 G, which is accurate because this is a RAID-1 array (one disk mirrored to the other). To get further space information for a Btrfs file system, use the `btrfs filesystem df` command:

```
# btrfs filesystem df /btrfs  
Data, RAID1: total=1.00GiB, used=6.11MiB  
System, RAID1: total=8.00MiB, used=16.00KiB  
Metadata, RAID1: total=1.00GiB, used=112.00KiB  
GlobalReserve, single: total=16.00MiB, used=0.00B
```

Btrfs allocates space on disks in chunks. A chunk is 1 GB for data and 256 MB for metadata. A chunk also has a specific RAID profile associated with it, which allows Btrfs to have different allocation profiles for data and for metadata. The output of the `btrfs filesystem df` command shows that it has allocated only a 1 GB chunk of RAID-1 at this time.

Btrfs is not yet actually “RAIDing” the entire device. For example, if you specify RAID-1 for metadata and RAID-0 for data, metadata writes are mirrored across all the disks and data writes are striped across the disks.

The output of the `btrfs filesystem df` command shows that you are currently using 6.11 MB. The disk (system RAID1) has a total allocated space of 8 MB and has used 16 KB. Metadata is allocated 1 GB of space as well; it has used 112 KB of it.

## btrfs filesystem show| sync Utilities

- Use the `btrfs filesystem show` command to display the structure of a file system, as in this example:

```
# btrfs filesystem show
Label: 'Btrfs'  uuid: ...
      Total devices 2 FS bytes used 6.23MiB
      devid 1 size 5.00GiB used 2.01GiB path /dev/xvdb
      devid 2 size 5.00GiB used 2.01GiB path /dev/xvdd
```

- Use the `btrfs filesystem sync` command to force a sync for the file system:

```
# btrfs filesystem sync /btrfs
```



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

Use the `btrfs filesystem show` command to display the structure of a file system. The syntax is as follows:

```
btrfs filesystem show [options]<path>|<uuid>
```

If you omit the optional path and uuid, the command shows information about all the Btrfs file systems.

The following example displays the structure of a Btrfs file system:

```
# btrfs filesystem show
Label: 'Btrfs'  uuid: ...
      Total devices 2 FS bytes used 6.23MiB
      devid    1 size 5.00GiB used 2.01GiB path /dev/xvdb
      devid    2 size 5.00GiB used 2.01GiB path /dev/xvdd
```

Use the `btrfs filesystem sync` command to force a sync for the file system. The file system must be mounted. To force a sync of the file system mounted on /btrfs:

```
# btrfs filesystem sync /btrfs
```

## The btrfs filesystem defragment Utility

- Use the `btrfs filesystem defragment` command to defragment a file system, file, or directory.

- To defragment a file system:

```
# btrfs filesystem defragment /btrfs
```

- To defragment and compress a file system:

```
# btrfs filesystem defragment -c /btrfs
```

- Set up automatic defragmentation by specifying the `autodefrag` option with the `mount` command:

```
# mount -o autodefrag /dev/sdb /btrfs
```



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

Btrfs provides online defragmentation of a file system, file, or directory. The online defragmentation facility reorganizes data into contiguous chunks wherever possible to create larger sections of available disk space and to improve read and write performance. Use the `btrfs filesystem defragment` command to defragment a file or a directory.

```
btrfs filesystem defragment [options] <file>|<dir> [...]
```

The available options include the following:

- `-v`: Verbose
- `-c`: Compress file contents while defragmenting
- `-r`: Defragment files recursively
- `-f`: Flush file system after defragmenting
- `-s start`: Defragment only from byte `start` onward
- `-l len`: Defragment only up to `len` bytes
- `-t size`: Defragment files less than or equal to `size` bytes

You can set up automatic defragmentation by specifying the `-o autodefrag` option when you mount the file system. Do not defragment with kernels up to version 2.6.37 if you have created snapshots or made snapshots of files by using the `cp --reflink` option. Btrfs in these earlier kernels unlinks the copy-on-write copies of data.

## The btrfs filesystem resize Utility

- Use the `btrfs filesystem resize` command to resize a file system.
- To accommodate the resizing, you must have space available on the underlying devices.
- To reduce the file system by 2 GB:

```
# btrfs filesystem resize -2G /btrfs
```

- To increase the file system by 2 MB:

```
# btrfs filesystem resize +2M /btrfs
```

- To have the file system occupy all available space:

```
# btrfs filesystem resize max /btrfs
```



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

Btrfs provides online resizing of a file system. Use the `btrfs filesystem resize` command to resize a file system. You must have space available to accommodate the resizing because the command has no effect on the underlying devices. The syntax is as follows:

```
btrfs filesystem resize [devid:] [+/-] <newsize>[kKmMgGtTpPeE] | [devid:]max<path>
```

Descriptions of the parameters:

- `+ newsize`: Increases the file system size by the `newsize` amount
- `- newsize`: Decreases the file system size by the `newsize` amount
- `Newsize`: Specifies the `newsize` amount
- `kKmMgGtTpPeE`: Specifies the unit of `newsize`. If no units are specified, the parameter defaults to bytes.
- `Max`: Specifies that the file system occupies all available space

For example, to reduce the size of the file system by 2 GB:

```
# btrfs filesystem resize -2G /btrfs
Resize '/btrfs/' of '-2G'
```

## btrfs device Utilities

- Use the `btrfs device` command to manage devices on Btrfs file systems.
- The available commands include:
  - `btrfs device add|delete|scan|ready|stats`
- The `btrfs device scan` command scans physical devices looking for members of a Btrfs volume.
  - This allows a multiple-disk Btrfs file system to be mounted without specifying all the disks on the `mount` command.
- Udev automatically runs `btrfs device scan` on boot.
- The `btrfs device ready` command checks whether all devices are in cache for mounting.
- The `btrfs device stats` command shows IO stats.



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

Use the `btrfs device` command to manage devices on Btrfs file systems. A list of the available commands is as follows:

```
# btrfs device
Usage: btrfs device <command> [<args>]
      btrfs device add [options] <device> [<device>...] <path>
          Add a device to a filesystem
      btrfs device delete <device>|<devid> [<device>|<devid>...] <path>
      btrfs device remove <device>|<devid> [<device>|<devid>...] <path>
          Remove a device from a filesystem
      btrfs device scan [(-d|--all-devices) |<device> [<device>...]]
          Scan devices for a btrfs filesystem
      ...
```

The `btrfs device scan` command scans physical devices looking for members of a Btrfs volume. This command allows a multiple-disk Btrfs file system to be mounted without specifying all the disks on the `mount` command.

You do not need to run `btrfs device scan` from the command line, because udev automatically runs `btrfs device scan` on boot.

## The btrfs device Utility: Examples

- Use the `btrfs device add` command to add a device to a mounted file system, as in this example:

```
# btrfs device add /dev/xvdd /btrfs
```

- Use the `btrfs filesystem balance` command after adding a device:

```
# btrfs filesystem balance /btrfs
```

- Use the `btrfs device delete` command to remove a device from a file system:

```
# btrfs device delete /dev/xvdd /btrfs
```



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

Use the `btrfs device add` command to add a device to a file system. In this example, the current file system structure is as follows:

```
# btrfs filesystem show
Label: 'Btrfs'  uuid: ...
          Total devices 1  FS bytes used 5.79MiB
          devid      1  size 5.00GiB used 276.00MiB path /dev/xvdb
```

The `btrfs filesystem df` command shows:

```
# btrfs filesystem df /btrfs
Data, single: total=8.00MiB, used=5.67MiB
System, single: total=4.00MiB, used=16.00KiB
Metadata, single: total=264.00MiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

The output of the `df` command shows:

```
# df -h /btrfs
Filesystem  Size  Used  Avail  Use%  Mounted on
/dev/sdb     5.0G  22M   4.8G    1%   /btrfs
```

Add a 5 GB disk, /dev/xvdd, to the file system mounted on /btrfs by using the btrfs device add command:

```
# btrfs device add /dev/xvdd /btrfs
```

The output of the btrfs filesystem show command shows the newly added device:

```
# btrfs file show
```

```
Label: 'Btrfs'  uuid: ...
        Total devices 2  FS bytes used 5.79MiB
        devid      1 size 5.00GiB used 276.00MiB path /dev/xvdb
        devid      2 size 5.00GiB used 0.00  path /dev/xvdd
```

The output of the btrfs filesystem df command shows no difference after adding the new device:

```
# btrfs filesystem df /btrfs
```

```
Data, single: total=8.00MiB, used=5.67MiB
System, single: total=4.00MiB, used=16.00KiB
Metadata, single: total=264.00MiB, used=112.00KiB
GlobalReserve, single: total=16.00MiB, used=0.00B
```

There is no difference in the output because the newly added device has not yet been allocated for either data or metadata.

The additional size is reflected in the output of df:

```
# df -h /btrfs
Filesystem  Size  Used   Avail  Use%  Mounted on
/dev/sdb     10g   22M   9.8G   1%   /btrfs
```

After adding a device, it is recommended that you run the following balance command on the file system:

```
# btrfs filesystem balance /btrfs
```

Running this command redistributes space by balancing the chunks of the file system across all the devices. This command also reclaims any wasted space.

Use the btrfs device delete command to remove a device from a file system.

Example:

```
# btrfs device delete /dev/xvdd /btrfs
```

## btrfs scrub Utilities

- Use the `btrfs scrub` command to manage scrubbing on Btrfs file systems.
- Scrubbing is performed in the background by default. It attempts to report and repair bad blocks on the file system.
- Available commands include:
  - `btrfs scrub start`
  - `btrfs scrub cancel`
  - `btrfs scrub resume`
  - `btrfs scrub status`



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

You can initiate a check of the entire file system by triggering a file system scrub job. The scrub job runs in the background by default and scans the entire file system for integrity. It automatically attempts to report and repair any bad blocks that it finds along the way. Instead of going through the entire disk drive, the scrub job deals only with data that is actually allocated. Depending on the allocated disk space, this is much faster than performing an entire surface scan of the disk.

Scrubbing involves reading all the data from all the disks and verifying checksums. If any values are not correct, the data can be corrected by reading a good copy of the block from another drive. The scrubbing code also scans on read automatically. It is recommended that you scrub high-usage file systems once a week and all other file systems once a month.

The following is a partial list of the available `btrfs scrub` commands:

```
# btrfs scrub
Usage: btrfs scrub <command> [options] <path>|<device>
      btrfs scrub start [-BdqrR] [-c ioprio_class ...]
                        Start a new scrub
      ...
      btrfs scrub status [-dR] <path>|<device>
                        Show status of running or finished scrub
```

## The btrfs scrub Utility: Examples

- Use the `btrfs scrub start` command to start a scrub on all the devices of a file system or on a single device:

```
# btrfs scrub start /btrfs
```

- Use the `btrfs scrub status` command to get the status of a scrub job. The following example includes detailed scrub information about each device in the file system:

```
# btrfs scrub status -dR /btrfs
```

- Use the `btrfs scrub cancel` command to cancel a running scrub job:

```
# btrfs scrub cancel /btrfs
```

- Use the `btrfs scrub resume` command to resume a previously canceled or interrupted scrub:

```
# btrfs scrub resume /btrfs
```



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

Use the `btrfs scrub start` command to start a scrub on all the devices of a file system or on a single device. The syntax is as follows:

```
btrfs scrub start [-BdqrR] [-c ioprio_class ...]
```

Description of options:

- `-B`: Do not run in the background and print statistics when finished.
- `-d`: Print separate statistics for each device of the file system. This option is used in conjunction with the `-B` option.
- `-q`: Run in quiet mode, omitting error messages and statistics.
- `-r`: Run in read-only mode, not correcting any errors.
- `-R`: Raw print mode. Print full data instead of the summary.
- `-c ioprio_class`: Set IO priority class (see `ionice(1)` man page).
- `-n ioprio_classdata`: Set IO priority classdata (see `ionice(1)` man page).
- `-f`: Force a scrub to start. This may be needed if a scrub is currently running and if there is a damaged scrub stats record file.

The following example starts a scrub on the Btrfs file system that is mounted on `/btrfs`:

```
# btrfs scrub start /btrfs
scrub started on /btrfs, fsid ... (pid=...)
```

Use the `btrfs scrub status` command to get the status of a scrub job. Two options are available:

- `-d`: Print separate statistics for each device of the file system.
- `-R`: Print detailed scrub statistics.

The following shows partial output from the `btrfs scrub status` command with the `-dR` options:

```
# btrfs scrub status -dR /btrfs
scrub status for ...
scrub device /dev/xvdb (id 1) history
    scrub started at ... and finished after 00:00:00
    data_extents_scrubbed: 97
    tree_extents_scrubbed: 8
    data_bytes_scrubbed: 6205440
    tree_bytes_scrubbed: 131072
    read_errors: 0
    csum_errors: 0
    verify_errors: 0
    no_csum: 80
    csum_discards: 0
    super_errors: 0
    malloc_errors: 0
    uncorrectable_errors: 0
    unverified_errors: 0
    corrected_errors: 0
    last_physical: 1396703232
```

You can also cancel a running scrub job. Progress is saved in the scrub progress file, and you can resume scrubbing later.

To cancel a scrub:

```
# btrfs scrub cancel /btrfs
```

To resume a canceled or interrupted scrub job:

```
# btrfs scrub resume /btrfs
```

The `scrub resume` command has the same options as the `scrub start` command.

Btrfs stores the last two minutes of root ID generations at 30-second intervals. Btrfs continues to keep rolling these generations, even if there are no changes in the file system.

If a scrub does not correct errors, you can use the following mount option to roll back to a known good B-tree, given that the rest of the tree is available because of copy-on-write:

```
# mount -o recovery /dev/xvdb /btrfs
```

## Converting Ext File Systems to Btrfs

- Use the `btrfs-convert` utility to convert an ext2, ext3, or ext4 file system to a Btrfs file system.
- To convert a nonroot ext file system:
  1. Unmount the ext file system.
  2. Use `fsck` to check the integrity of the ext file system.
  3. Use the `btrfs-convert` utility to convert the file system.
  4. Edit `/etc/fstab` and change the file system type to `btrfs`.
  5. Mount the converted file system on the original mount point.
- The syntax of the `btrfs-convert` utility is as follows:

```
btrfs-convert [options] device
```
- You cannot convert the `root` file system or a bootable partition, such as `/boot`, to Btrfs.

ORACLE®

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

Btrfs supports the conversion of ext2, ext3, and ext4 file systems to Btrfs file systems. The original ext file system metadata is stored in a snapshot named `ext#_saved` so that the conversion can be reversed if necessary.

Use the `btrfs-convert` utility to convert an ext file system. Always make a backup copy before converting a file system. To convert a nonroot ext file system, perform the steps listed in the slide.

You cannot convert the `root` file system or a bootable partition, such as `/boot`, to Btrfs.

To install a system with a Btrfs root file system, refer to the following:

[http://docs.oracle.com/cd/E52668\\_01/E54695/html/ol7-install-btrfs-filesystem.html](http://docs.oracle.com/cd/E52668_01/E54695/html/ol7-install-btrfs-filesystem.html)

## Quiz

Which of the following statements are true?

- a. Btrfs is a general-purpose file system.
- b. All Btrfs data and metadata is written via copy-on-write.
- c. Btrfs supports only read-only snapshots.
- d. Btrfs supports online resizing and defragmentation.



ORACLE®

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

## Quiz

Which of the following statements are true?

- a. Btrfs has built-in RAID support for RAID-0, RAID-1, RAID-5, RAID-6, and RAID-10 (though RAID-5 and RAID-6 are not recommended for production implementation).
- b. Btrfs supports transparent compression.
- c. Btrfs automatically detects and optimizes solid state drives.
- d. Oracle Linux with UEK R2 is the first release to officially support Btrfs.



ORACLE®

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

## Quiz

Which of the following are valid `btrfs` commands?

- a. `btrfs subvolume create`
- b. `btrfs snapshot create`
- c. `btrfs filesystem show`
- d. `btrfs device create`



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe the features of the Btrfs file system
- Create a Btrfs file system
- Create Btrfs subvolumes and snapshots
- Take a snapshot of a file in a Btrfs subvolume
- Mount Btrfs subvolumes and snapshots
- Defragment and resize a Btrfs file system
- Add and remove devices in a Btrfs file system
- Check and repair the integrity of a Btrfs file system
- Convert ext file systems to Btrfs



ORACLE®

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

## Practice 11: Overview

This practice covers the following topics:

- Creating a Btrfs file system
- Working with subvolumes and snapshots
- Recovering from data corruption



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.

# Storage Administration

The ORACLE logo, featuring the word "ORACLE" in white capital letters inside a red rectangular box.

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

## Objectives

After completing this lesson, you should be able to:

- Describe the Linux device mapper
- Explain Logical Volume Manager (LVM)
- Configure LVM components
- Back up and restore volume group metadata
- Describe LVM thin provisioning
- Describe snapper
- Explain the Linux kernel multi-disk (MD) driver
- Discuss RAID and configure RAID devices



ORACLE®

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

# Logical Volume Manager (LVM)

- LVM is a tool to facilitate the management of physical volumes, volume groups, and logical volumes.
  - **Physical volume (PV):** It is a physical storage device.
  - **Volume group (VG):** Physical volumes are grouped together into storage pools called volume groups.
  - **Logical volume (LV):** Each volume group is divided into LVs.
- File systems are created on LVs.
- Use LVM to increase the size of VGs and LVs without interrupting normal operations.



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

The Linux device mapper (DM) supplies an abstraction layer on top of the actual storage block devices and provides the foundation for Logical Volume Manager (LVM2), RAID, encryption, and other storage features. LVM2 manages multiple physical volumes and also supports mirroring and striping of logical volumes to provide redundancy and increase performance. To assist in understanding LVM, the following terms are defined:

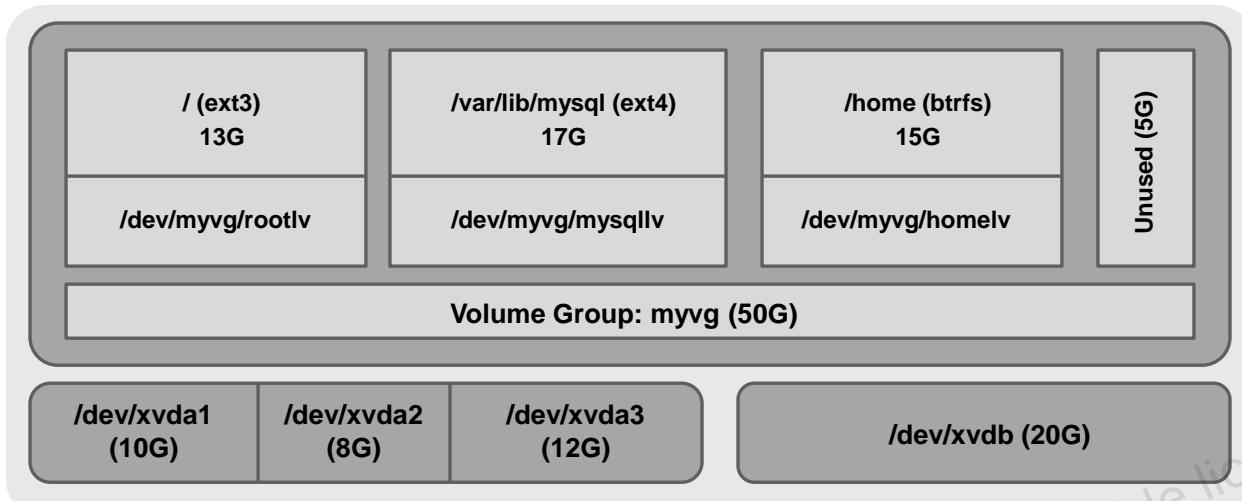
- **Physical volumes:** These are physical storage devices (hard drives, partitions, arrays).
- **Volume groups:** Physical volumes are grouped together into volume groups.
- **Logical volumes:** Each volume group is divided into logical volumes.

Each logical volume is analogous to a standard disk partition. Logical volumes, therefore, function as partitions that can span multiple physical disks.

File systems can be created on logical volumes and connected to the directory hierarchy through mount points. As these “partitions” become filled with data, use LVM to increase their capacity from free space in the volume group. New physical storage devices are added to volume groups to increase the capacity of these groups.

With LVM, the capacity is expanded in logical volumes “on the fly” (dynamically) without the need to back up the data on standard partitions, modify the partition table, and restore the data. Logical volume management does not interrupt usage and is transparent to users.

## LVM Configuration: Example



ORACLE®

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

This slide illustrates a possible LVM configuration. There are four physical volumes (PV), with three of these being partitions on one drive and the fourth being an entire hard drive:xvda1: 10 GB

- xvda1: 10 GB
- xvda2: 8 GB
- xvda3: 12 GB
- xvdb: 20 GB

All of the physical volumes are grouped into a single volume group (VG) named `myvg`. The storage capacity of this group is 50 GB, which is the total space of the four physical volumes.

The volume group is divided into three logical volumes (LV). The following lists the LV name, the size, the mount point, and the file system type of each logical volume:

- `rootlv`, 13 GB, / (root), ext3
- `mysql1lv`, 17 GB, /var/lib/mysql, ext4
- `homelv`, 15 GB, /home, btrfs

Finally, the illustration shows that there is 5 GB of unused space in the VG. This is available to be allocated to any of the existing logical volumes or to a new logical volume.

# Physical Volume Utilities

- Use the `pvcreate` command to create physical volumes:

```
# pvcreate -v /dev/xvdd1 /dev/xvdd2
```

- The following commands display physical volumes:
  - `pvdisplay`
  - `pvs`
  - `pvscan`

- Use the `pvremove` command to remove physical volumes:

```
# pvremove /dev/xvdd1
```

- Additional PV commands are available:
  - `pvchange`: Change the attributes of physical volumes.
  - `pvresize`: Resize physical volumes.
  - `pvck`: Check the consistency of physical volumes.
  - `pvmove`: Move extents from one physical volume to another.



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

The first step in implementing LVM is to create physical volumes. In addition to creating physical volumes, commands exist to display the attributes of physical volumes, remove physical volumes, and perform other functions on physical volumes.

## Creating Physical Volumes

Use the `pvcreate` command to create physical volumes. The syntax is:

```
pvcreate [options] device
```

You can initialize multiple disks or partitions for use by LVM in the same command. For example, the following command initializes two partitions. The `-v` option makes the output more verbose:

```
# pvcreate -v /dev/xvdd1 /dev/xvdd2
      Set up physical volume for "/dev/xvdd1" with ...
      Zeroing start of device /dev/xvdd1
      Writing physical volume data to disk "/dev/xvdd1"
Physical volume "/dev/xvdd1" successfully created
      Set up physical volume for "/dev/xvdd2" with ...
      Zeroing start of device /dev/xvdd2
...
```

## Displaying Physical Volumes

Use the `pvdisplay` command to display attributes of physical volumes.

```
# pvdisplay
"/dev/xvdd1" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name          /dev/xvdd1
...
```

In addition to `pvdisplay`, two other commands list information about physical volumes. The `pvs` command reports information about physical volumes in a more condensed form. The `pvscan` command scans all disks for physical volumes. Example:

```
# pvs
  PV        VG   Fmt  Attr  Psize  PFree
  /dev/xvdd1    lvm2  a--  1.00g  1.00g
  /dev/xvdd2    lvm2  a--  1.00g  1.00g
# pvscan
  PV /dev/xvdd1      lvm2  [1.00GiB]
  PV /dev/xvdd2      lvm2  [1.00GiB]
  Total: 2 [2.00GiB] / in use: 0 [0] / in no VG: 2 [2.00 GiB]
```

## Removing Physical Volumes

Use the `pvremove` command to remove a physical volume. Example:

```
# pvremove /dev/xvdd1
  Labels on physical volume "/dev/xvdd1" successfully wiped
# pvdisplay /dev/xvdd1
  No physical volume label read from /dev/xvdd1
  Failed to read physical volume "/dev/xvdd1"
```

## Additional PV Commands

The following are other commands that are associated with the manipulation of physical volumes:

- **`pvchange`**: Change the attributes of physical volumes.
- **`pvresize`**: Resize physical volumes.
- **`pvck`**: Check the consistency of physical volumes.
- **`pvmove`**: Move extents from one physical volume to another.

# Volume Group Utilities

- Use the `vgcreate` command to create volume groups:

```
# vgcreate -v myvolg /dev/xvdd1 /dev/xvdd2
```

- The following commands display volume groups:

- `vgdisplay`
- `vgs`
- `vgscan`

- Use the `vgremove` command to remove volume groups:

```
# vgremove myvolg
```

- Additional VG commands are available. For example:

- `vgchange`: Change volume group attributes.
- `vgck`: Check the consistency of volume groups.
- `vgextend`: Add physical volumes to a volume group.
- `vgreduce`: Remove physical volumes from a volume group.



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

The next step in implementing LVM is to assign the physical volumes to an existing or new volume group.

## Creating a Volume Group

Use the `vgcreate` command to create a new volume group. The space in a volume group is divided into "extents." The default physical extent size is 4 MB. The syntax is:

```
vgcreate [options] volume_group_name physical_volume
```

For example, to create a volume group named `myvolg` by using the `/dev/xvdd1` and `/dev/xvdd2` physical volumes with a default physical extent size of 4 MB, enter:

```
# vgcreate -v myvolg /dev/xvdd1 /dev/xvdd2
      Adding physical volume '/dev/xvdd1' to volume group 'myvolg'
      Adding physical volume '/dev/xvdd2' to volume group 'myvolg'
      Archiving volume group "myvolg" metadata (seqno 0).
      Creating volume group backup "/etc/lvm/backup/myvolg" ...
Volume group "myvolg" successfully created
```

## Displaying Volume Groups

Use the `vgdisplay` command to display attributes of volume groups:

```
# vgdisplay
--- Volume group ---
VG Name      myvolg
System ID
Format       lvm2
...
```

In addition to `vgdisplay`, two other commands list information about volume groups. The `vgs` command reports information about volume groups in a more condensed form. The `vgscan` command scans all disks for volume groups and rebuilds caches. Example:

```
# vgs
VG      #PV  #LV  #SN  Attr     Vsize   VFree
myvolg    2    0    0  wz--n-  5.01g   5.01g
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "myvolg" using metadata type lvm2
```

## Removing Volume Groups

Use the `vgremove` command to remove a volume group. Example:

```
# vgremove myvolg
Volume group "myvolg" successfully removed
# vgdisplay
No volume groups found
```

## Additional VG Commands

The following commands are used to manipulate volume groups:

- **`vgcfgbackup`**: Back up volume group configurations.
- **`vgcfgrestore`**: Restore volume group configurations.
- **`vgchange`**: Change volume group attributes.
- **`vgck`**: Check the consistency of volume groups.
- **`vgconvert`**: Change the volume group metadata format.
- **`vgexport`**: Unregister volume groups from the system.
- **`vgextend`**: Add physical volumes to a volume group.
- **`vgimport`**: Register an exported volume group with the system.
- **`vgmerge`**: Merge volume groups.
- **`vgmknodes`**: Create special files for volume group devices in `/dev`.
- **`vgreduce`**: Remove physical volumes from a volume group.
- **`vgrename`**: Rename a volume group.
- **`vgsplit`**: Move physical volumes into a new or existing volume group.

The use of the `vgcfgbackup` and `vgcfgrestore` commands is discussed in a later slide.

# Logical Volume Utilities

- Use the `lvcreate` command to create logical volumes:

```
# lvcreate -v --size 2g --name myvol myvolg
```

- The following commands display logical volumes:

- `lvdisplay`
- `lvs`
- `lvscan`

- Use the `lvremove` command to remove logical volumes:

```
# lvremove myvolg/myvol
```

- Additional LV commands are available. For example:

- `lvchange`: Change the attributes of logical volumes.
- `lvextend`: Add space to a logical volume.
- `lvreduce`: Reduce the size of a logical volume.



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

The next step in implementing LVM is to create logical volumes from the space allocated to volume groups.

## Creating Logical Volumes

Use the `lvcreate` command to create a new logical volume. This command automatically creates the block device nodes in the `/dev` directory. The syntax is:

```
lvcreate [options] --size <size> --name LV_name VG_name
```

The `--size` option defines the size of the logical volume by allocating logical extents from the free physical extent pool of the volume group. For example, to create a logical volume named `myvol` from the volume group named `myvolg` with a size of 2 GB, enter:

```
# lvcreate -v --size 2g --name myvol myvolg
      Setting logging type to disk
      Finding volume group "myvolg"
      Archiving volume group "myvolg" metadata (seqno 1).
      Creating logical volume myvol
      Create volume group backup "/etc/lvm/backup/myvolg" ...
...
...
```

## Displaying Logical Volumes

Use the `lvdisplay` command to display the attributes of logical volumes.

```
# lvdisplay
--- Logical volume ---
LV Path          /dev/myvolg/myvol
LV Name          myvol
VG Name          myvolg
LV UUID...
```

...

In addition to `lvdisplay`, two other commands list information about logical volumes. The `lvs` command reports information about logical volumes in a more condensed form. The `lvscan` command scans all disks for logical volumes. Example:

```
# lvs
  LV   VG     Attr      LSize  Pool Origin Data% Move Log Cpy...
  myvol myvolg -wi-a---- 2.00g
# lvscan
  ACTIVE    '/dev/myvolg/myvol' [2.00 GiB] inherit
```

## Removing Logical Volumes

Use the `lvremove` command to remove a logical volume. You must include the volume group name as well as the logical volume name. You are prompted to confirm your request. Example:

```
# lvremove myvol
  Volume group "myvol" not found
  Skipping volume group myvol
# lvremove myvolg/myvol
  Do you really want to remove active logical volume myvol? ...
  Logical volume "myvol" successfully removed
```

## Additional LV Commands

The following commands are used to manipulate logical volumes:

- **`lvchange`**: Change the attributes of logical volumes.
- **`lvconvert`**: Change logical volume layout.
- **`lvextend`**: Add space to a logical volume.
- **`lvmdiskscan`**: List devices that may be used as physical volumes.
- **`lvmsadc`**: Collect activity data.
- **`lvmsar`**: Create activity report.
- **`lvreduce`**: Reduce the size of a logical volume.
- **`lvrename`**: Rename a logical volume.
- **`lvresize`**: Resize a logical volume.

## Making Logical Volumes Usable

- Final steps:
  - Create a file system on the logical volume.
  - Create a mount point.
  - Attach the logical volume to the directory hierarchy.
- The `lvcreate` command creates two entries in the `/dev` directory for each logical volume. Example:
  - `/dev/mapper/myvolg-myvol`
  - `/dev/myvolg/myvol`
- Either of these block device names are usable as arguments to the `mkfs` command:

```
# mkfs -t ext4 /dev/mapper/myvolg-myvol
# mkfs -t ext4 /dev/myvolg/myvol
```



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

The last step in implementing LVM is to create a file system on the logical volume, create a mount point, and attach the logical volume to the directory hierarchy. There is nothing new here; these steps were discussed in the lesson titled “Partitions, File Systems, and Swap.” Logical volumes do not require a file system to be usable. For example, they can be used as Automatic Storage Management (ASM) disks or as a raw device.

The only thing different from creating a file system on a disk partition and creating a file system on a logical volume is the name of the block device in the `/dev` directory. The `lvcreate` command creates two entries in the `/dev` directory for each logical volume. For example, when creating the logical volume named `myvol` from the volume group named `myvolg`, the following two block device names in the `/dev` directory were automatically created:

```
/dev/mapper/myvolg-myvol
/dev/myvolg/myvol
```

Use either of these device names as arguments to the `mkfs` command when making a file system. For example, to make an ext4 file system on the `myvol` logical volume, enter either of the following commands:

```
# mkfs -t ext4 /dev/mapper/myvolg-myvol
# mkfs -t ext4 /dev/myvolg/myvol
```

The `blkid` command displays the same output (and the same UUIDs) when querying either of the logical volume device names:

```
# blkid /dev/mapper/myvolg-myvol
/dev/mapper/myvolg-myvol: UUID="9fa64e..." TYPE="ext4"
# blkid /dev/myvolg/myvol
/dev/myvolg/myvol: UUID="9fa64e..." TYPE="ext4"
```

Create a mount point and mount the new logical volume file system. Example:

```
# mkdir /test
# mount /dev/mapper/myvolg-myvol /test
```

Create an entry in `/etc/fstab` to mount the file system at boot time.

# Backing Up and Restoring Volume Group Metadata

- LVM metadata contains configuration details of volume groups.
- Metadata backups and archives are automatically created on every volume group and logical volume configuration change.
  - Backups are stored in /etc/lvm/backup.
  - Archives are stored in /etc/lvm/archive.
- Configuration settings are stored in /etc/lvm/lvm.conf.
- Use the vgcfgbackup command to manually back up LVM metadata.
- Use the vgcfgrestore command to restore from a backup to recover from corrupted or missing metadata.



ORACLE

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

LVM metadata contains configuration details of LVM volume groups. By default, metadata backups and archives are automatically created on every volume group and logical volume configuration change. Settings can be changed in the LVM configuration file, /etc/lvm/lvm.conf. The lvm dumpconfig command displays configuration settings.

Metadata backups are stored in the /etc/lvm/backup directory. Metadata archives are stored in the /etc/lvm/archive directory. You can manually back up the metadata by using the vgcfgbackup command. For example, the following command backs up the metadata of the myvolg volume group to the /etc/lvm/backup/myvolg file:

```
# vgcfgback myvolg
```

Omit the name of the volume group to back up metadata for all volume groups. Use the -f <filename> option to give the backup file a specific file name.

The following are examples of error messages you might get if the metadata area is corrupted or incorrect:

Couldn't find device with uuid '...`.

Couldn't find all physical volumes for volume group myvolg.

You can use the vgcfgrestore command to restore volume group metadata from a backup. Provide the name of the volume group as an argument to the vgcfgrestore command.

## LVM Thin Provisioning

- LVM thin provisioning allows you to over-commit the physical storage.
- You can create file systems, which are larger than the available physical storage.
- Use the `lvcreate` command to create a thin pool:

```
# lvcreate -L 100m -T myvolg/mythinpool
```

- Use the `lvcreate` command to create a thin volume.
  - A thin volume is a virtual disk inside a thin pool.
  - The size of the virtual disk can be greater than the size of the thin pool.

```
# lvcreate -V 1g -T myvolg/mythinpool -n mythinvol
```

- Use the `lvs` command to monitor the allocated pool data and add more capacity when it starts to become full.



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

LVM thin provisioning allows you to create virtual disks inside a thin pool. The size of the virtual disk can be greater than the available space in the thin pool. This allows you to over-commit the physical storage and create file systems that are larger than the actual available physical storage. It is important that you monitor the thin pool and add more capacity when it starts to become full.

Thin pools are created using the `lvcreate` command and, as such, are essentially logical volumes. Use the `-T` option, the `--thin` option, or the `--thinpool` option when creating a thin pool. The following example creates a thin pool named `mythinpool` from the `myvolg` volume group that is 100m in size:

```
# lvcreate -L 100m -T myvolg/mythinpool
```

This command creates a logical volume as shown by the following command:

```
# lvs
  LV        VG        Attr       LSize   Pool       Origin Data%...
  mythinpool  myvolg  twi-a-tz--  100.00m                  0.00
```

The “Data%” column shows the allocated pool data. The example shows 0.00% because virtual thin volumes have not yet been created in this thin pool. You can also use the `lvdiskusage` command to show the “Allocated pool data” percentage.

The thin pool logical volume is not mountable. That is, there is no entry in the /dev directory:

```
# ls /dev/myvolg*
ls: cannot access /dev/myvolg*: No such file or directory
```

Use the `lvcreate` command with the `-V` option to create a thin volume (a virtual disk) from a thin pool. The following example creates a 1 GB thin volume named `mythinvol` in the `myvolg/mythinpool` thin pool. Note that the size of the thin volume is larger than the size of the thin pool that contains it.

```
# lvcreate -V 1g -T myvolg/mythinpool -n mythinvol
```

This command creates a thin volume as shown by the following command:

```
# lvs
  LV      VG      Attr       LSize   Pool      Origin Data%...
  mythinpool myvolg twi-a-tz-- 100.00m
  mythinvol  myvolg Vwi-a-tz--  1.00g mythinpool      0.00
  mythinvol  myvolg Vwi-a-tz--  1.00g mythinpool      0.00
```

Note the difference in attributes. The thin volume has a "V" attribute for virtual disk. The Data% column shows 0.00 until you create a file system on the thin volume.

The virtual disk thin volume has a /dev entry as shown as follows. In this example, the entry is a symbolic link to the `dm-4` block device.

```
# ls -l /dev/myvolg*
lrwxrwxrwx ... mythinvol -> ../dm-4
```

You can create a file system on this thin volume and mount it. For example:

```
# mkfs.ext4 /dev/myvolg/mythinvol
# mkdir /myvol
# mount /dev/myvolg/mythinvol /myvol
```

Output of the `df` command shows the size of the file system is 976M, which is an over-allocation of the available storage in the thin pool.

```
# df -h
Filesystem      Size  Used  Avail  Use%  Mounted on
...
/dev/mapper/myvolg-mythinvol
      976M  2.6M  907M    1%  /myvol
```

Copy some data to `/myvol`, then run the `lvs` command to show the allocated pool data.

```
# cp /boot/vmlinuz* /myvol
# lvs
  LV      VG      Attr       LSize   Pool      Origin Data%...
  mythinpool myvolg twi-a-tz-- 100.00m      49.00
  mythinvol  myvolg Vwi-a-tz--  1.00g mythinpool      4.79
```

This shows you have used 49% of the allocated pool data. This also shows that the thin volume has used 4.79% of 1 GB. You can use the `lvextend` command to add space to a thin pool logical volume.

## Snapper

- Command-line utility in Oracle Linux 7 to create and manage snapshots
- Supports Btrfs and LVM thin volumes
- Requires a configuration file for each Btrfs and LVM volume
  - To create the `myvol1_snap` configuration file for ext4 file system on LVM thin volume mounted on `/myvol1`:

```
# snapper -c myvol1_snap create-config -f "lvm(ext4)" /myvol1
```

- Entry is added to `/etc/sysconfig/snapper`.
- The `.snapshots` directory is created in the `/myvol1` directory.
- The configuration file, `myvol1_snap`, is created in the `/etc/snapper/configs` directory.



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

Snapper is a command-line utility in Oracle Linux 7 used to create and manage snapshots of LVM thin volumes. It can create, delete, and compare snapshots and revert changes done between them. Snapper also allows for the easy creation and management of snapshots for Btrfs. Use the following command to install the snapper software package:

```
# yum install snapper
```

The snapper software package includes a `cron.hourly` file to create snapshots and a `cron.daily` file to clean up old snapshots.

```
# ls -l /etc/cron*/snapper
-rwxr-xr-x ... /etc/cron.daily/snapper
-rwxr-xr-x ... /etc/cron.hourly/snapper
```

To create a snapshot using snapper, a configuration file is required for the LVM thin volume or Btrfs subvolume. The LVM and Btrfs volumes must also have a mounted file system. Use the `create-config` command to create the configuration file. The following example creates a configuration file named `myvol1_snap` for an LVM ext4 file system mounted on `/myvol1`:

```
# snapper -c myvol1_snap create-config -f "lvm(ext4)" /myvol1
```

This command adds an entry to `/etc/sysconfig/snapper`, creates a `.snapshots` directory in the `/myvol1` directory, and creates the configuration file, `myvol1_snap`, in the `/etc/snapper/configs` directory.

# Snapper

- You can create three types of snapshots:
  - pre, post, and single
- Always associate a pre snapshot with a post snapshot.
- All snapshots have an associated number.
- To create a pre snapshot:

```
# snapper -c myvol1_snap create -t pre -p
```

- To create a post snapshot:

```
# snapper -c myvol1_snap create -t post -pre-num 4 -p
```

- To list the differences between a pre snapshot (#4) and a post snapshot (#5):

```
# snapper -c myvol1_snap diff 4..5
```



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

Each /etc/snapper/configs/\* file describes a snapper configuration. See the `snapper-configs(5)` man page for a description of the parameters in the snapper configuration file.

The hourly cron job performs automatic snapshot creation, but you can also create snapshots manually.

There are three types of snapshots that you can create by using snapper:

- **pre**: Use to record the state of a volume before a modification. Pre snapshots should always have a corresponding post snapshot.
- **post**: Use to record the state of a volume after a modification.
- **single**: These snapshots have no special relationship to other snapshots.

Use the `create -t` command to specify the type of snapshot to create. Possible values are `single`, `pre`, and `post`.

The following example creates a pre snapshot of the `/myvol1` volume. The `-p` option causes snapper to display the number of the snapshot. In this example, the number is 4.

```
# snapper -c myvol1_snap create -t pre -p
4
```

The snapshots are stored by snapshot number in the `.snapshots` subdirectory of the volume.

The following example creates a post snapshot of the /myvol1 volume. The --pre-num 4 option references the associated pre snapshot number. The -p option causes snapper to display the number of the snapshot. In this example, the number is 5.

```
# snapper -c myvol1_snap create -t post --pre-num 4 -p  
5
```

Use the snapper status command to display the files and directories that have been added, removed, or modified between a pre snapshot and a post snapshot. Example:

```
# snapper -c myvol1_snap status 4..5
```

Use the snapper diff command to display the differences between the contents of the files in a pre snapshot and a post snapshot. Example:

```
# snapper -c myvol1_snap diff 4..5
```

Use the snapper list command to list the snapshots that exist for a volume defined by the snapper configuration file. Example:

```
# snapper -c myvol1_snap list
```

Use the snapper delete command to delete a snapshot number. Example:

```
# snapper -c myvol1_snap delete 1
```

Use the snapper undochange command to revert the contents of a volume defined by a configuration file to the pre snapshot contents. Example:

```
# snapper -c myvol1_snap undochange 4..5
```

# Redundant Array of Independent Disks (RAID)

- The multi-disk (MD) driver supports software RAID.
  - MD organizes disk drives into RAID devices (arrays).
- Common RAID levels are supported:
  - Linear RAID: Concatenated drives
  - RAID-0: Striping
  - RAID-1: Mirroring
  - RAID-5: Distributed parity
  - RAID-6: Dual-distributed parity
  - Nested RAID levels:
    - RAID 0+1: Mirrored striping
    - RAID 1+0 (or RAID 10): Striped mirror



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

In addition to logical volume management with LVM2, the Linux kernel supports “software RAID” with the multi-disk (MD) driver. MD organizes disk drives into RAID devices, or arrays, and provides different RAID levels. RAID devices are virtual devices created from two or more real block devices.

RAID combines multiple disk drives into an array and allows data to be spread across the drives to increase capacity, achieve redundancy, and increase performance.

## Supported RAID Levels

The following RAID levels are the most commonly used levels supported by Oracle Linux:

- **Linear RAID:** Linear RAID simply groups drives together to create a larger virtual drive. Data is written to the first drive until it is full, and then it is written to the next drive. There is no redundancy or performance benefit. Reliability is actually decreased, because the entire array cannot be used if any one drive fails.
- **RAID-0:** RAID-0 is called striping and provides an increase in performance but offers no redundancy. Data is broken down into stripes and written across all the drives, rather than filling up the first drive before moving on to the next as is the case with Linear RAID. In addition, as with Linear RAID, the entire array cannot be used if any one drive fails.

- **RAID-1:** RAID-1 is called mirroring and provides redundancy by writing identical data to each drive in the array. If one drive fails, the mirror drive satisfies I/O requests. RAID-1 is expensive because the same information is written to all the disks in the array.
- **RAID-5:** RAID-5 is the most common type of RAID and uses striping with distributed parity. RAID-5 is able to recover from the loss of one drive in the array. Parity information is calculated based on the contents of the rest of the drives in the array. This information is used to reconstruct data when one drive in the array fails. The reconstructed data also satisfies I/O requests to the failed drive before it is replaced and repopulates the new disk after the failed one has been replaced. With RAID-5, the parity is distributed across all drives in the array.
- **RAID-6:** RAID-6 uses striping with double distributed parity. RAID-6 is able to recover from the loss of two drives in the array. RAID-6 is commonly used when it is more important to have data redundancy and preservation instead of performance.

### Nested RAID Levels

Nested RAID levels, also known as hybrid RAID, combine standard RAID levels for additional performance and/or redundancy. One example is **RAID 0+1**, which is a mirror (RAID-1) of striped (RAID-0) disks. Another example is **RAID 1+0**, sometimes called **RAID 10**, which is a stripe of mirrors.

Many Oracle Database customers use one of these nested RAID levels, but have the RAID implemented in the storage area network (SAN) arrays. RAID 5 and RAID 6 provide the redundancy needed, but add overhead to calculate parity. This can impact the performance of write-intensive databases.

## The mdadm Utility

- Use the `mdadm` command to build, manage, and monitor Linux MD devices (software RAID devices).
- To create a device:

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/xvdb2  
/dev/xvdd2 --spare-devices=1 /dev/xvdd3
```

- To get help on the `mdadm --create` command:

```
# mdadm --create --help
```

- To query a device:

```
# mdadm --query /dev/md0  
# mdadm --detail /dev/md0  
# cat /proc/mdstat
```



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

### Creating a RAID Device

The `mdadm` command is used to build, manage, and monitor Linux MD devices (software RAID devices). The basic syntax to create a new RAID array is:

```
mdadm --create <md_device> --level=<RAID_level> --raid-devices=<#> devices  
--spare-devices=<#> devices
```

For example, to create a RAID-1 device (`/dev/md0`) consisting of two block devices (`/dev/xvdd1` and `/dev/xvdd2`) and one spare device (`/dev/xvdd3`), enter:

```
# mdadm --create /dev/md0 --level=1 -raid-devices=2 /dev/xvdd1 /dev/xvdd2 -  
-spare-devices=1 /dev/xvdd3
```

Information about the RAID device to be created is displayed along with the following prompt:

Continue creating array?

Respond with “y” to create the array. View the `/proc/mdstat` file to check the status of your MD RAID devices:

```
# cat /proc/mdstat  
  
Personalities : [raid1]  
  
md0 : active raid1 xvdd3[2] (S) xvdd2[1] xvdb2[0]  
      1047552 blocks super 1.2 [2/2] [UU]
```

## Querying a RAID Device

You can also use the `mdadm` command to view information about the RAID device:

```
# mdadm --query /dev/md0
/dev/md0: 1023.00MiB raid1 2 devices, 1 spare. Use mdadm --detail for more
detail.

To see further details, enter:
# mdadm --detail /dev/md0
/dev/md0:
          Version : 1.2
          Creation Time : ...
          Raid Level : raid1
          Array Size : 1047552 (1023.00 MiB 1072.69 MB)
          Used Dev Size : 1047552 (1023.00 MiB 1072.69 MB)
          Raid Devices : 2
          Total Devices : 3
          Persistence : Superblock is persistent

          Update Time : ...
          State : clean
          Active Devices : 2
          Working Devices : 3
          Failed Devices : 0
          Spare Devices : 1

          Consistency Policy : unknown

          Name : host03.example.com:0 (local to host
host03.example.com)
          UUID : ...
          Events : 17

          Number  Major  Minor  RaidDevice State
              0      202      18        0  active sync   /dev/xvdb2
              1      202      50        1  active sync   /dev/xvdd2

              2      202      51        -  spare     /dev/xvdd3
```

## Making RAID Devices Usable

1. Create a file system on the RAID device.
2. Create a mount point.
3. Attach the RAID device to the directory hierarchy.
4. Update the `mdadm` configuration file:
  - `/etc/mdadm.conf`
  - `ARRAY /dev/md0 devices=/dev/xvdd1,/dev/xvdd2`



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

The last step in implementing a RAID device is to create a file system on the device, create a mount point, and attach the device to the directory hierarchy. These are the same steps required for standard partitions, logical volumes, or RAID devices. You can create logical volumes on top of RAID block devices.

Assuming that the RAID device name is `/dev/md0`, to create an ext4 file system on the device and mount it to `/raid`, enter:

```
# mkfs -t ext4 /dev/md0
# mkdir /raid
# mount /dev/md0 /raid
```

The last step is to update the `mdadm` configuration file, `/etc/mdadm.conf`, with RAID configuration information. This helps `mdadm` assemble existing arrays at system boot. You can either copy and adapt the sample configuration file from `/usr/share/doc/mdadm-4.0/mdadm.conf-example` or create the file from the beginning. This is an example entry for a RAID device:

```
ARRAY /dev/md0 devices=/dev/xvdd1,/dev/xvdd2
```



## Quiz

Which of the following commands is used to build, manage, and monitor software RAID devices?

- a. raidadm
- b. lvadm
- c. mdadm
- d. dmsetup



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe the Linux device mapper
- Explain Logical Volume Manager (LVM)
- Configure LVM components
- Back up and restore volume group metadata
- Describe LVM thin provisioning
- Describe snapper
- Explain the Linux kernel multi-disk (MD) driver
- Discuss RAID and configure RAID devices



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

## Practice 12: Overview

This practice covers the following topics:

- Creating Linux LVM partitions
- Creating a logical volume
- Creating a file system and mounting a logical volume
- Backing up volume group metadata
- Creating a logical volume snapshot
- Increasing the capacity of a logical volume
- Restoring volume group metadata
- Creating a thinly provisioned logical volume
- Using snapper with LVM thin provisioned logical volumes
- Creating a RAID device
- Removing partitions



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

# Advanced Storage Administration

ORACLE®

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

## Objectives

After completing this lesson, you should be able to:

- Describe and configure disk quotas
- Describe the Linux `dm-crypt` device driver
- Describe and configure encrypted block devices
- Describe the `kpartx` utility
- Describe Udev
- Create Udev rules
- Use the `udevadm` utility



ORACLE®

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

## Disk Quotas

- Disk quotas limit file system disk usage:
  - For users and groups
  - To a number of blocks (disk space)
  - To a number of inodes (files)
- Set hard limits and soft limits on blocks and inodes.
  - Hard limits are maximums.
  - Soft limits have a grace period.
- Disk quota tools are used for:
  - Configuration
  - Reporting
  - Enabling/disabling
  - Maintaining accuracy



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

Set disk quotas for your users to restrict disk space and to notify you when usage is reaching a specified limit. Configure disk quotas for individual users as well as user groups. Quotas are set to limit the number of disk blocks (or disk space), as well as the number of inodes, which limit the number of files a user can create.

Hard limits define the maximum number of blocks (disk space) or inodes (files) for the user or group on the file system. Users can exceed soft limits for a certain period of time, called a “grace period.” The grace period is configurable in time periods of days, hours, minutes, or seconds.

Various disk quota tools are summarized:

- `quotacheck`: This command creates disk usage tables, `aquota.user` and `aquota.group`.
- `edquota` / `setquota`: These commands configure the quotas for users and groups. `edquota` is interactive and is also used to configure the grace period for soft limits.
- `quota`: Use this command to verify that quotas are set for users and groups.
- `quotaon` / `quotaoff`: Use these commands to enable and disable quotas.
- `repquota`: This command reports disk usage.
- `quotacheck`: Use this command to ensure the accuracy of quota reporting.

## Enabling Disk Quotas

- Enable disk quotas with these file system mount options:
  - Use `usrquota` to enable for individual users
  - Use `grpquota` to enable for groups
- Use the `quotacheck` command to create a disk usage table for quota-enabled file systems:

```
# quotacheck -cug /home
```
- The above command creates the following files in the `/home` directory:
  - `aquota.user`
  - `aquota.group`

ORACLE®

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

### Enabling Quotas

Disk quotas are enabled by including mount options to entries in the `/etc/fstab` file. Include the `usrquota` mount option to configure disk quotas for individual users. Include the `grpquota` mount option to configure disk quotas for user groups. The following example enables both user and group quotas on the `/home` file system:

```
/dev/xvdb1 /home ext4 usrquota,grpquota 0 0
```

After making any changes to entries in `/etc/fstab`, the file system must be unmounted and remounted for the change to take effect. Either run the `umount` command followed by the `mount` command to remount the file system or use the `-o remount` option as follows:

```
# mount -o remount /home
```

### Creating the Quota Database Files

After the `/etc/fstab` file is modified and the file system is remounted, run the `quotacheck` command. This command creates disk usage tables for the quota-enabled file system. Two files are created, `aquota.user` and `aquota.group`. Use the `-cug` options to the `quotacheck` command and include the quota-enabled file system mount point as an argument. Example:

```
# quotacheck -cug /home
```

To generate the table of file system disk usage, run the following command:

```
# quotacheck -avug
```

The options are described as follows:

- a**: Check all quota-enabled, locally mounted file systems.
- v**: Display verbose status information as the quota check proceeds.
- u**: Check user disk quota information.
- g**: Check group disk quota information.

## Summary of Quota Commands

- **edquota:** Interactive utility to assign quotas and configure grace periods
- **setquota:** Command-line utility to assign quotas
- **quota:** Verify that quotas are set
- **quotaoff:** Disable quotas without modifying the limits
- **quotaon:** Enable quotas
- **repquota:** Report on disk usage
- **quotacheck:** Ensure the accuracy of quota reporting



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

### Assigning Quotas per User

The **edquota** command is an interactive command to configure the quotas for a user. For example, to configure the quota for user **john**, enter the following command:

```
# edquota john
```

A text file opens in the default editor, defined by the **EDITOR** variable, and allows you to specify the limits for the user. The following is an example of the file:

```
Disk quotas for user john (uid 500)
Filesystem    blocks   soft   hard   inodes   soft   hard
/dev/xvdb1     400428     0     0    30412     0     0
```

The first column is the file system that has quota enabled. The second column is the number of blocks that the user is currently using. The next two columns are used to set soft and hard block limits for the user on the file system. The inodes column shows how many inodes the user is currently using. The last two columns are used to set the soft and hard inode limits for the user on the file system.

Make any changes to soft and hard limits in the text file and save and close the file. Repeat the **edquota** command for each user to whom you want to assign hard and soft limits on blocks and inodes.

## Assigning Quotas per Group

The edquota command is also used to configure quotas for a group. Include the `-g group` option with the command. For example, to configure the quota for group teamA, enter the following command:

```
# edquota -g teamA
```

A text file opens, allowing you to set limits for the group. Set the soft and hard block limits and the soft and hard inode limits for the group on the file system and save and close the file.

## Setting Quotas from the Command Line

Use the setquota command to configure quotas from the command line. The syntax is:

```
setquota username block_soft_limit block_hard_limit inode_soft_limit
inode_hard_limit file_system
```

This command is also used to set quotas for groups. The `-p` option (prototype) allows quota settings from one user or group to be applied to another user or group.

## Verifying Quotas

Use the quota command to verify that quotas are set. Enter the following to verify quotas for user john:

```
# quota john
```

Enter the following command to verify that quotas are set for group teamA:

```
# quota -g teamA
```

## Setting the Grace Period

To configure the grace period for soft limits, use the edquota -t command. A text file opens, allowing you to set both the block and inode grace periods as follows:

```
# edquota -t
```

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem	Block grace period	Inode grace period
/dev/xvdb1	7days	7days

Make any changes and save and close the file.

## Enabling and Disabling Quotas

One way to disable quotas is to set the limits to 0. If any of the values are set to 0, that limit is not set. Disable quotas without modifying the limits by using the quotaoff command. For example, to turn all user and group quotas off, enter:

```
# quotaoff -vaug
```

To enable quotas, use the quotaon command:

```
# quotaon -vaug
```

Alternatively, to enable quotas on a specific file system (for example /home), enter:

```
# quotaon -vug /home
```

## Quota Reporting

Use the `repquota` command to report on disk usage. To view the report for all quota-enabled file systems, use the `-a` option:

```
# repquota -a
```

Include the file system as an argument to view the report for a specific quota-enabled file system. For example, to view the report for the `/home` file system, enter:

```
# repquota /home
```

## Quota Accuracy

Run the `quotacheck` command to ensure the accuracy of quota reporting. Quota inaccuracies are caused by unclean system shutdowns. Unmount the file system before running this command. Disable quotas before running the `quotacheck` command and enable quotas afterward. For example, to ensure the accuracy of quota reporting on the `/home` file system:

```
# quotaoff -vaug /home
# quotacheck -vaug /home
# quotaon -vaug /home
```

## Encrypted Block Devices

- The `dm-crypt` device driver is used to encrypt block devices.
- Encrypted volumes can be stored on:
  - Disk partitions
  - Logical volumes
  - Disk images
- `dm-crypt` can encrypt:
  - All file systems supported by Linux
  - Swap space
  - RAID volumes
  - LVM physical volumes



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

The Linux device mapper also supports the creation of encrypted block devices using the `dm-crypt` device driver. Data on these encrypted devices can only be accessed by providing the correct password at boot time. Because the encryption takes place on the underlying block device, `dm-crypt` can be used for encrypting all file systems supported by Linux, as well as swap space. Encrypted volumes can be stored on disk partitions, logical volumes, and disk images. `dm-crypt` can also be configured to encrypt RAID volumes and LVM physical volumes.

## cryptsetup Utility

- Use the `cryptsetup` command to create and activate encrypted volumes and to manage authentication.
- The `cryptsetup` command includes the Linux Unified Key Setup (LUKS) extension, a disk encryption standard.
- The basic syntax of the `cryptsetup` command is:
  - `cryptsetup [options] [action] [action args]`
- To initialize a volume and set an initial key:  
`# cryptsetup luksFormat /dev/xvdd1`
- To open the partition and create the device mapping:  
`# cryptsetup luksOpen /dev/xvdd1 cryptfs`
- The device mapping file is:
  - `/dev/mapper/cryptfs`



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

Because the `dm-crypt` device mapper is concerned only with encryption of the block device, it relies on user-space tools, such as `cryptsetup`, to set up `dm-crypt` managed device-mapper mappings. Use `cryptsetup` to create and activate encrypted volumes and to manage authentication.

The `cryptsetup` command also provides commands to deal with the LUKS on-disk format. LUKS is a standard for hard disk encryption. It standardizes a partition header, as well as the format of the bulk data.

### LUKS Actions

The following is a partial listing of valid LUKS actions:

- **`luksFormat`:** Initializes a LUKS partition and sets the initial key
- **`luksOpen`:** Opens the partition and creates the device mapping
- **`luksSuspend`:** Suspends the active device and wipes the encryption key from the kernel
- **`luksResume`:** Resumes the suspended device and reinstates the encryption key
- **`luksAddKey`:** Adds a new key passphrase
- **`luksRemoveKey`:** Removes the key from the LUKS device
- **`luksUUID`:** Prints the UUID if the device has a LUKS header
- **`luksClose`:** Closes the partition and removes the device mapping

## Using the cryptsetup Command

The basic syntax of the `cryptsetup` command is:

```
# cryptsetup [options] [action] [action args]
```

For example, to initialize a volume and set an initial key, enter the following command. A warning message appears, asking for confirmation to continue. You are prompted for the initial key (passphrase) twice to ensure that your password is typed correctly.

```
# cryptsetup luksFormat /dev/xvdd1
WARNING!
=====
This will overwrite data on /dev/xvdd1 irreversibly.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: <passphrase>
Verify passphrase: <passphrase>
```

To open the partition and create the device mapping, enter:

```
# cryptsetup luksOpen /dev/xvdd1 cryptfs
Enter passphrase for /dev/xvdd1: <passphrase>
```

The device mapping created in this example is `/dev/mapper/cryptfs`. Create the file system on this device mapping file, not the physical device (`/dev/xvdd1`). The device mapping file is actually a symbolic link to `/dev/dm-0`:

```
# ls -l /dev/mapper/cryptfs
lrwxrwxrwx, ... /dev/mapper/cryptfs -> ../dm-0
```

To check the status of the encrypted volume, enter:

```
# cryptsetup status cryptfs
/dev/mapper/cryptfs is active.
  type:  LUKS1
  cipher: aes-xts-plain64
  keysize: 256 bits
  device:  /dev/xvdd1
  offset:  4096 sectors
  size:    2093056 sectors
  mode:    read/write
```

To close the partition and remove the device mapping, enter:

```
# umount /cryptfs (if the file system is mounted)
# cryptsetup luksClose /dev/mapper/cryptfs
```

## Making an Encrypted Device Usable

1. Create a file system on the encrypted device.
2. Create a mount point.
3. Attach the encrypted device to the directory hierarchy.
4. Update the `/etc/crypttab` configuration file:  
— `cryptfs /dev/xvdd1 none luks`
5. Add the file system to `/etc/fstab`.
6. Enter the passphrase to mount the encrypted file system during boot.

ORACLE®

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

As is the case with any block device, to make the encrypted device usable, you must create a file system on the device, create a mount point, and attach the device to the directory hierarchy.

Assuming that the encrypted device name is `/dev/mapper/cryptfs` and that you want to create an ext4 file system on the device and mount it to `/crypt`, enter:

```
# mkfs -t ext4 /dev/mapper/cryptfs  
# mkdir /crypt  
# mount /dev/mapper/cryptfs /crypt
```

Then update the configuration file, `/etc/crypttab`. This ensures that the encrypted file system is properly set up and mounted at boot time. The following entry is appropriate for your example:

```
# cat /etc/crypttab  
# <target name> <source device> <key file> <options>  
cryptfs /dev/xvdd1 none luks
```

Finally, add the file system to `/etc/fstab` for the actual mounting to take place. You are prompted to enter your passphrase for the encrypted file system during the boot process.

## kpartx Utility

- The `kpartx` utility is used to create device maps from partition tables.
- It reads block devices and creates device mappings of partitions in `/dev/mapper`.
- Using the `system.img` drive image as an example, to list partitions found on the drive image:

```
# kpartx -l system.img
```

- To add device mappings for the detected partitions:

```
# kpartx -a system.img
```

- You can now mount the partitions in `/dev/mapper` and view the files that they contain.
- To disconnect the device:

```
# kpartx -d system.img
```



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

The `kpartx` utility can be used to set up device mappings for the partitions of any partitioned block device. It reads partition tables on the specified device and creates device maps for the detected partitions. After running `kpartx` on a partitioned block device, device files are created in `/dev/mapper`. These files represent a disk partition or a disk volume.

Using the `system.img` file that you installed Oracle Linux on as an example, the following set of commands illustrates the usage of `kpartx`. Use the `-l` option to list any partitions that are found on the drive image.

```
# kpartx -l system.img
loop0p1 : 0 204800 /dev/loop0 2048
loop0p2 : 0 12288000 /dev/loop0 206848
loop0p3 : 0 4096000 /dev/loop0 212494848
loop0p4 : 0 2 /dev/loop0 16590848
```

The output shows that the drive image contains four partitions. The first column gives the names of the device files that are created. Before adding the device files, a listing of `/dev/mapper` shows no devices.

```
# ls /dev/mapper
control
```

To add the device mappings for the detected partitions, use the `-a` option.

```
# kpartx -a system.img
```

To view the new device mappings:

```
# ls /dev/mapper
control  loop0p1  loop0p2  loop0p3  loop0p4
```

You can now mount the partitions and view the files that they contain. For example, create a mount point and mount the first partition:

```
# mkdir /mnt/sysimage
# mount /dev/mapper/loop0p1 /mnt/sysimage
```

View the files on this first partition:

```
# ls /mnt/sysimage
config-3.10.0-229.el7.x86_64
config-3.8.13-55.1.6.el7uek.x86_64
...
...
```

The `/boot` file system is mounted on the first partition. As expected, the files on `/boot` (`/dev/xvda1`) are the same:

```
# df -kh | grep xvda1
/dev/xvda1    97M   46M   46M  50% /boot
# ls /boot
config-3.10.0-229.el7.x86_64
config-3.8.13-55.1.6.el7uek.x86_64
...
...
```

To unmount the partition and disconnect the device, enter:

```
# umount /mnt/sysimage
# kpartx -d system.img
```

Notice that the mapping is gone in `/dev/mapper`:

```
# ls /dev/mapper
control
```

## Udev: Introduction

- Udev dynamically creates device file names at boot time.
- Udev is now part of `systemd`.
  - The Udev daemon, `systemd-udevd`, receives device uevents directly from the kernel whenever a device is added or removed.
  - For every event, `systemd-udevd` executes matching instructions specified in Udev rules.
- With Udev, device file names can change after reboot.
  - `/dev/sdc` could be named `/dev/sdb` after reboot.
- You can configure Udev to create persistent device names.
- You can use these names in the file system mount table, `/etc/fstab`, or as an argument to the `mount` command.



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

Udev is the device manager for the Linux kernel. Udev dynamically creates or removes device node files at boot time in the `/dev` directory for all types of devices. Udev is now part of `systemd` as you can see by viewing the “udev” file names included with the `systemd` RPM package.

```
# rpm -ql systemd | grep udev
/etc/udev
...
```

The Udev daemon, `systemd-udevd`, receives device uevents directly from the kernel whenever a device is added or removed from the system. For every event, `systemd-udevd` executes matching instructions specified in Udev rules.

Device file names can change when disks are removed from the system due to failure. For example, devices are named `/dev/sda`, `/dev/sdb`, and `/dev/sdc` at boot time. But on the next reboot, `/dev/sdb` fails and what was previously `/dev/sdc` is named `/dev/sdb`. Any configuration references to `/dev/sdb` now contain content originally referenced by `/dev/sdc`.

The solution to avoid this type of situation is to guarantee consistent names for devices through reboots. You can configure Udev to create persistent names and use these names in the file system mount table, `/etc/fstab`, or as an argument to the `mount` command.

## Udev Rule Files and Directories

- Udev rules determine how to identify devices and how to assign a persistent name.
- Udev rules files are located in the following directories:
  - /lib/udev/rules.d/ – The default rules directory
  - /etc/udev/rules.d/ – The custom rules directory
- Custom rules files override default rules files of the same name.
- Rules files are sorted and processed in lexical order.
- Sample rules file names:
  - 10-dm.rules
  - 50-udev-default.rules



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

Udev rules determine how to identify devices and how to assign a name that is persistent through reboots or disk changes. When Udev receives a device event, it matches the configured rules against the device attributes in sysfs to identify the device. Rules can also specify additional programs to run as part of device event handling.

Udev rules files are located in the following directories:

- /lib/udev/rules.d/ – The default rules directory
- /etc/udev/rules.d/ – The custom rules directory. These rules take precedence.

Rules files need to have unique names. Files in the custom rules directory override files of the same name in the default rules directory. Rules files are sorted and processed in lexical order. The following is a partial listing of rules files from the default and custom rules directories:

```
# ls /lib/udev/rules.d
100-balloon.rules  10-dm.rules  11-dm-lvm.rules
...
# ls /etc/udev/rules.d
70-persistent-ipoib.rules
```

## Sample Udev Rules

```
SUBSYSTEM=="virtio-ports", KERNEL=="vport*", ATTR{name}=="?*",  
SYMLINK+="virtio-ports/$attr{name}"  
  
SUBSYSTEM=="tty", KERNEL=="tty[0-9]*", GROUP="tty", MODE="0620"  
  
SUBSYSTEM=="mem", KERNEL=="mem|kmem|port", GROUP="kmem", MODE="0640"  
  
SUBSYSTEM=="usb", EBV{DEVTYPE}=="usb_device",  
ENV{ID_USB_INTERFACES}=="*:0701???:*", GROUP="lp"  
  
SUBSYSTEM=="block", GROUP="disk"  
  
SUBSYSTEM=="scsi_generic|scsi_tape", SUBSYSTEMS=="scsi",  
ATTRS{type}=="1|8", GROUP="tape"  
  
KERNEL=="tun", MODE="0666", OPTIONS+="static_node-net/tun"
```



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

This slide contains selected entries from the /lib/udev/rules.d/50-udev-default.rules file. This rules file contains over 60 entries. The selected entries assist in describing the syntax of the rules files.

Comments begin with a # sign. Each noncommented line in a rules file consists of a list of one or more key-value pairs separated by a comma. There are two types of keys:

- Match keys
- Assignment keys

If all match keys match their respective value, the rule gets applied and the assignment keys are assigned the specified value. Each key has a distinct operation, depending on the operator. Valid operators are:

- ==: Compare for equality
- !=: Compare for inequality
- =: Assign a value to a key
- +=: Add the value to the current values for the key
- :=: Assign the final value to the key. Disallow any later changes by any later rules.

Shell-style pattern matching (\*, ?, []) is also supported in Udev rules.

## Match Keys

The following key names are used to match against device properties. Some of the keys also match against properties of the parent devices in `sysfs`, and not just the device that has generated the event. If multiple keys are specified in a single rule, all these keys must match.

- **ACTION:** Match the name of the event action.
- **DEVPATH:** Match the devpath of the event device.
- **KERNEL:** Match the name of the event device.
- **NAME:** Match the name of a network interface. It can be used if the `NAME` key was set in one of the preceding rules.
- **SYMLINK:** Match the name of the symlink targeting the node. It can be used if a `SYMLINK` key was set in one of the preceding rules. There can be multiple symlinks, but only one needs to match.
- **SUBSYSTEM:** Match the subsystem of the event device.
- **TEST**{*octal mode mask*}:**Test** the existence of a file. You can specify *octal mode mask*.

Other match keys include `DRIVER`, `ATTR`{*filename*}, `KERNELS`, `SUBSYSTEMS`, `DRIVERS`, `ATTRS`{*filename*}, `TAGS`, `ENV`{*key*}, `TAG`, `PROGRAM`, and `RESULT`.

## Assignment Keys

The following keys can have values assigned to them:

- `NAME` – The name to use for a network interface. The name of a device node cannot be changed by Udev; only additional symlinks can be created.
- `SYMLINK` – The name of the symlink targeting the node
- `OWNER`, `GROUP`, `MODE` – The permissions for the device node
- `OPTIONS` – Rule and device options. The `ignore_remove` option used in the example means “Do not remove the device node when the device goes away.”

Other assignment keys include `ATTR`{*key*}, `ENV`{*key*}, `TAG`, `RUN`{*type*}, `LABEL`, `GOTO`, `IMPORT`{*type*}, `WAIT_FOR`, and `OPTIONS`.

## String Substitutions

The `NAME`, `SYMLINK`, `PROGRAM`, `OWNER`, `GROUP`, `MODE`, and `RUN` keys support many `printf`-like string substitutions. The substitutions used in the example are:

- `%M` – The kernel major number for the device
- `%m` – The kernel minor number for the device

Additional string substitutions are supported. Refer to the `udev`(7) man page for all supported substitutions and details on additional match keys, additional assignment keys, and additional rule and device options.

## udevadm Utility

- The udevadm utility is a management tool for Udev.
- To query the Udev database for all device information for /dev/xvdd:  

```
# udevadm info --query=all --name=/dev/xvdd
```
- To print all sysfs properties of /dev/xvdd:  

```
# udevadm info --attribute-walk --name=/dev/xvdd
```
- These properties can be used in Udev rules to match the device.
- It prints all devices along the chain, up to the root of sysfs.



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

The udevadm utility is a userspace management tool for Udev. Among other functions, you can use udevadm to query sysfs and obtain device attributes to help in creating Udev rules that match a device. To display udevadm usage:

```
# udevadm --help
Usage: udevadm [--help] [--version] [--debug] COMMAND [OPTIONS]
      info          query sysfs or the udev database
      trigger       requests events from the kernel
      settle        wait for the event queue to finish
      control       control the udev daemon
      monitor      listen to kernel and udev events
      hwdb         maintain the hardware database index
      test          test an event run
      test-builtin test a built-in command
```

You can also obtain usage for each of the udevadm commands. For example, to get help on using the info command:

```
# udevadm info --help
```

Some examples follow. To query the Udev database for the device path of /dev/xvdd:

```
# udevadm info --query=path --name=/dev/xvdd  
/devices/vbd-5696/block/xvdd
```

To query the Udev database for all device information for /dev/xvdd:

```
# udevadm info --query=all --name=/dev/xvdd  
P: /devices/vbd-5696/block/xvdd  
N: xvdd  
E: DEVNAME=/dev/xvdd  
E: DEVPATH=/devices/vbd-5696/block/xvdd  
E: DEVTYPE=disk  
E: MAJOR=202  
E: MINOR=48  
E: MPATH_SBIN_PATH=/sbin  
E: SUBSYSTEM=block  
E: TAGS=:systemd:  
E: USEC_INITIALIZED=12403
```

Enter the following to print all sysfs properties of /dev/xvdd. These properties can be used in Udev rules to match the device. It prints all devices along the chain, up to the root of sysfs.

```
# udevadm info --attribute-walk --name=/dev/xvdd  
...  
looking at device '/devices/vbd-5696/block/xvdd':  
KERNEL=="xvdd"  
SUBSYSTEM=="block"  
DRIVER==""  
ATTR{ro}=="0"  
ATTR{size}=="20971520"  
...  
looking at parent device '/devices/vbd-5696':  
KERNELS=="vbd-5696"  
SUBSYSTEMS=="xen"  
DRIVERS=="vbd"  
ATTR{devtype}=="vbd"  
ATTR{nodename}=="device/vbd/5696"
```

## Creating a Symbolic Link to a Device Node

- Create a rules file:

```
# vi /etc/udev/rules.d/10-local.rules  
KERNEL=="xvdd", SUBSYSTEM=="block", SYMLINK="my_disk"
```

- Run udevadm trigger to process the rules files:

```
# udevadm trigger
```

- View the symlink:

```
# ls -l /dev/my*  
lrwxrwxrwx. ... /dev/my_disk -> xvdd
```



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

The order in which rules are evaluated is important. When creating your own rules, you want these evaluated before the defaults. Because rules are processed in lexical order, create a rules file with a file name such as /etc/udev/rules.d/10-local.rules for it to be processed first.

The following rule creates the /dev/my\_disk symbolic link to the /dev/xvdd device node. You can create a Udev rule to change the name of a network interface, but the name of a device node cannot be changed by Udev. Only additional symlinks can be created for device nodes.

```
KERNEL=="xvdd", SUBSYSTEM=="block", SYMLINK="my_disk"
```

Run udevadm trigger to process the rules files:

```
# udevadm trigger
```

The symlink now exists.

```
# ls -l /dev/my*  
lrwxrwxrwx. ... /dev/my_disk -> xvdd
```

Remove the 10-local.rules file and run udevadm trigger to remove the symlink.

## Quiz



Which of the following can be encrypted?

- a. All file systems supported by Linux
- b. Swap space
- c. RAID volumes
- d. LVM physical volumes
- e. All of the above



**ORACLE®**

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

**Answer: e**

## Quiz



Which of the following statements are true?

- a. Udev is part of systemd.
- b. You can define Udev rules to create persistent symbolic links to device nodes.
- c. Run the `udevadm trigger` command to process rules files.
- d. The `start_udev` utility is a user-space management tool for Udev.



ORACLE®

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

**Answer: a, b, c**

## Summary

In this lesson, you should have learned how to:

- Describe and configure disk quotas
- Describe the Linux `dm-crypt` device driver
- Describe and configure encrypted block devices
- Describe the `kpartx` utility
- Describe Udev
- Create Udev rules
- Use the `udevadm` utility



ORACLE®

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

## Practice 13: Overview

The practices for this lesson cover the following topics:

- Creating and mounting a file system
- Setting disk quotas
- Exploring and configuring Udev rules
- Using kpartx



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.

# File Sharing

ORACLE®

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

## Objectives

After completing this lesson, you should be able to:

- Describe NFS
- Configure the NFS server and client
- Describe the `exportfs` utility
- Explain and configure automounter
- Discuss and configure `vsftpd`



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

## Introduction to NFS

- NFS allows a Linux server to share directory hierarchies with Linux clients over a network.
- NFS servers export the directory, and NFS clients mount the exported directory.
- Oracle Linux 7 supports two versions:
  - NFSv3
  - NFSv4
- NFS relies on Remote Procedure Calls (RPC) between clients and servers.
- Several `nfs` and `rpc` services work together, depending on which version of NFS is implemented.



ORACLE®

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

A Network File System (NFS) allows a server to share directory hierarchies (file systems) with remote systems over a network. NFS servers *export* the directory, and NFS clients *mount* the exported directory. The server directory then appears to the client systems as if they were local directories. NFS reduces storage needs and improves data consistency and reliability, because users are accessing files that are stored on a centralized server.

Oracle Linux 7 does not support NFS version 2 (NFSv2). The following two versions are supported:

- NFS version 3 (NFSv3), specification is RFC 1813
- NFS version 4 (NFSv4), specification is RFC 3530

NFS relies on Remote Procedure Calls (RPC) between clients and servers. RPC services are controlled by the `rpcbind` service. The `rpcbind` service replaces `portmap`, which was used in previous versions of Linux to map RPC program numbers to IP address port number combinations. `rpcbind` responds to requests for RPC services and sets up connections to the requested RPC service.

`rpcbind` is not used with NFSv4, because the server listens on well-known TCP port 2049. The mounting and locking protocols have also been incorporated into the NFSv4 protocol, so NFSv4 does not interact with the `lockd` and `rpc.statd` daemons either.

## The NFS Server and RPC Processes

- Starting the `nfs-server` service starts the NFS server and other RPC processes.  
`# systemctl start nfs`
- Several `nfsd` kernel threads are started. The number of threads are defined in `/proc/fs/nfsd/threads`.
- The RPC process includes:
  - `rpc.statd`: Implements monitoring protocol (NSM) between NFS client and NFS server
  - `rpc.mountd`: NFS mount daemon that implements the server side of the mount requests from NFSv3 clients
  - `rpc.idmapd`: Maps NFSv4 names and local UIDs and GIDs
  - `rpc.rquotad`: Provides user quota information for remote users



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

Starting the `nfs-server` service starts the NFS server and other RPC processes needed to service requests for shared NFS file systems. You can use the short name “`nfs`” rather than “`nfs-server`” when starting the service. Example:

```
# systemctl start nfs
```

This is the NFS server process that implements the user level part of the NFS service. The main functionality is handled by the `nfsd` kernel module. The user space program merely specifies what sort of sockets the kernel server listens on, which NFS versions it supports, and how many `nfsd` kernel threads it uses. Use the `ps -e` command to show the number of running threads. Only the partial output is shown:

```
# ps -e |grep nfs
...
... nfsd
... nfsd
...
...
```

The number of `nfsd` threads to run is defined in the `/proc/fs/nfsd/threads` file. In this example, 8 `nfsd` threads are specified:

```
# cat /proc/fs/nfsd/threads
8
```

Starting the `nfs-server` service also starts the RPC processes. You can use the `ps -e` command to display the names of the RPC processes. Only the partial output is shown:

```
# ps -e |grep rpc
...
... rpciod
... rpcbind
... rpc.statd
... rpc.mountd
... rpc.idmapd
... rpc.rquotad
```

#### **rpc.statd**

This process implements the Network Status Monitor (NSM) RPC protocol, which notifies NFS clients when an NFS server is restarted without being gracefully brought down. This is not used with NFSv4.

#### **rpc.mountd**

This is the NFS mount daemon that implements the server side of the mount requests from NFSv3 clients. It checks that the requested NFS share is currently exported by the NFS server and that the client is allowed to access it. For NFSv4, the `rpc.mountd` daemon is required only on the NFS server to set up the exports.

#### **rpc.idmapd**

This provides NFSv4 client and server upcalls, which map between on-the-wire NFSv4 names (which are strings in the form of `user@domain`) and local UIDs and GIDs. For `idmapd` to function with NFSv4, `/etc/idmapd.conf` must be configured. This service is required for use with NFSv4, although not when all hosts share the same DNS domain name.

#### **rpc.rquotad**

This process provides user quota information for remote users. It is started automatically by the `nfs` service and does not require user configuration. The results are used by the `quota` command to display user quotas for remote file systems and by the `edquota` command to set quotas on remote file systems.

#### **lockd**

This is a kernel thread that runs on both clients and servers. It implements the Network Lock Manager (NLM) protocol, which allows NFSv3 clients to lock files on the server. It is started automatically whenever the NFS server is run and whenever an NFS file system is mounted.

#### **nfslock**

Starting this service starts the RPC processes that allow NFS clients to lock files on the server.

# NFS Server Configuration

- Install the `nfs-utils` package.
- The configuration file for the NFS server is `/etc/exports`.
  - It contains a list of exported directory hierarchies that remote systems can mount.
- The format of `/etc/exports` entries is:

```
dir client1(options) [client2(options) ...]
```
- Example:

```
/export/directory 192.0.2.102(rw,async)
```
- Client options include (defaults are listed first):
  - `ro` / `rw`
  - `sync` / `async`
  - `wdelay` / `no_wdelay`
  - `no_all_squash` / `all_squash`



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

To begin configuring a system as an NFS server, install the `nfs-utils` package:

```
# yum install nfs-utils
```

The main configuration file for the NFS server is `/etc/exports`. This file stores a list of exported directory hierarchies that remote systems can mount. The format for entries is:

```
export-point client1(options) [client2(options) ... ]
```

The `export-point` is the absolute path name of the directory hierarchy to be exported. One or more client systems, each with specific options, can mount `export-point`. There are no spaces between the client attribute and the open bracket. When no client options are specified, the following default settings apply:

- **ro**: Read-only. Client hosts cannot change the data shared on the file system. To allow client hosts to make changes to the file system, specify the `rw` (read/write) option.
- **sync**: The NFS server replies to requests only after changes made by previous requests are written to disk. `async` specifies that the server does not have to wait.
- **wdelay**: The NFS server delays committing write requests when it suspects another write request is imminent. To disable the delay, use the `no_wdelay` option. `no_wdelay` is available only if the default `sync` option is also specified.

- **root\_squash**: Prevents root users connected remotely from having root privileges, effectively “squashing” the power of the remote root user. Requests appear to come from the nfsnobody user, an unprivileged user on the local system, or as specified by anonuid. To disable root squashing, specify the no\_root\_squash option.
- **no\_all\_squash**: Does not change the mapping of remote users. To squash every remote user (including root), use the all\_squash option.

To specify the user ID (UID) and group ID (GID) that the NFS server assigns to remote users, use the anonuid and anongid options as follows:

```
export-point client(anonuid=uid,anongid=gid)
```

The anonuid and anongid options allow you to create a special user and group account for remote NFS users to share. By default, access control lists (ACLs) are supported by NFS. To disable this feature, specify the no\_acl option when exporting the file system.

You can use wildcard characters, such as (\*) and (?) in client names. You can also export directories to all hosts on an IP network. To do this, specify an IP address and netmask pair as address/netmask. Either of the following forms is valid:

- 192.168.1.0/24
- 192.168.1.0/255.255.255.0

Other client options exist. Refer to `man exports` for descriptions of all options.

### /etc(exports Examples

In the following example, a client system with the IP address of 192.0.2.102 can mount the /export/directory with read/write permissions. All writes to the disk are asynchronous:

```
/export/directory 192.0.2.102(rw,async)
```

The following example exports the /exports/apps directory to all clients, converts all connecting users to the local anonymous nfsnobody user, and makes the directory read-only:

```
/exports/apps *(all_squash,ro)
```

The following example exports the /spreadsheets/proj1 directory with read-only permissions to all clients on the 192.168.1.0 subnet and read/write permissions to the client system named mgmtpc:

```
/spreadsheets/proj1 192.168.1.0/24(ro) mgmtpc(rw)
```

## Starting the NFS Service

- Start `rpcbind` before starting the `nfs` services:

```
# systemctl start rpcbind  
# systemctl start nfs  
# systemctl start nfslock
```

- Use the `systemctl enable` command to automatically start the services at boot time:

```
# systemctl enable nfs-server
```

- Specify configuration options and arguments in `/etc/sysconfig/nfs`.

- To display exported file systems:

```
# showmount -e
```



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

Note: NFSv3 requires the `rpcbind` service whereas NFSv4 does not.

The `rpcbind` service must be started before starting `nfs`. This command checks if the `rpcbind` service is enabled and running.

```
# systemctl status rpcbind
```

If the `rpcbind` service is running, the `nfs` service can be started. Restart `nfs` after making any configuration changes in `/etc/exports` or run the `exportfs -a` command.

```
# systemctl start nfs
```

Check if the `nfslock` service is enabled and running. Starting this service starts the RPC processes that allow NFS clients to lock files on the server.

```
# systemctl status nfslock
```

Use the `systemctl enable` command to automatically start the services at boot time.

```
# systemctl enable nfs
```

As an option, use the full name, `nfs-server`, when enabling the NFS:

```
# systemctl enable nfs-server
```

Specify configuration options and arguments by placing them in `/etc/nfs.conf`. The earlier configuration file, `/etc/sysconfig/nfs`, is still available for compatibility purposes.

Use the `showmount -e` command to display exported file systems.

```
# showmount -e
```

## The `exportfs` Utility

- `exportfs` exports or unexports directories and is run from the command line.
  - No need to change `/etc/exports`
  - No need to restart NFS service
- Syntax for the command:  
`exportfs [options] [client:dir ...]`
- Example:  
`# exportfs -i -o rw *:/Dev`
- This example does the following:
  - Exports `/Dev` to all clients systems (\*)
  - Allows read/write permission (`-o rw`)
  - Ignores `/etc/exports` entries (`-i`)



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

You can also configure an NFS server from the command line by using `exportfs`. This command allows the `root` user to selectively export or unexport directories without changing `/etc/exports` and without restarting the NFS service. The syntax for the command is:

`exportfs [options] [client:dir ...]`

The `client` argument is the name of the client system that `dir` is exported to. The `dir` argument is the absolute path name of the directory being exported. The following is a list of some of the options:

- **-r:** Re-export the entries in `/etc/exports` and synchronize `/var/lib/nfs/etab` with `/etc/exports`. The `/var/lib/nfs/etab` file is the master export table. `rpc.mountd` reads this file when a client sends an NFS `mount` command.
- **-a:** Export the entries in `/etc/exports` but do not synchronize `/var/lib/nfs/etab`. Run `exportfs -a` after making any configuration changes.
- **-i:** Ignore the entries in `/etc/exports` and use only command-line arguments.
- **-u:** Unexport one or more directories.
- **-o:** Specify client options as specified in `/etc/exports`.

## NFS Client Configuration

- Install the `nfs-utils` package.
- Use the `mount` command to mount exported file systems.
- Syntax for the command:

```
mount -t nfs -o options host:/remote/export /local/directory
```

- Example:  
`# mount -t nfs -o ro,nosuid abc:/home /abc_home`
- This example does the following:
  - It mounts `/home` from remote host `abc` on local mount point `/abc_home`.
  - The file system is mounted read-only, and users are prevented from running a setuid program (`-o ro,nosuid` options).
- Update `/etc/fstab` to mount NFS shares at boot time.



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

To begin configuring a system as an NFS client, install the `nfs-utils` package:

```
# yum install nfs-utils
```

Use the `mount` command to mount exported file systems (NFS shares) on the client side. The syntax for the command is:

```
mount -t nfs -o options host:/remote/export /local/directory
```

The following are descriptions of the arguments:

- `-t nfs`: Indicates that the file system type is `nfs`. With this option, `mount` uses NFSv4 if the server supports it; otherwise, it uses NFSv3.
- `-o options`: A comma-delimited list of mount options
- `host:/remote/export`: The host name exporting the file system, followed by a colon and by the absolute path name of the NFS share
- `/local/directory`: The mount point on the client system

For example, to mount the `/home` directory exported from host `abc` with read-only permissions (`ro` option) on local mount point `/abc_home` and prevent remote users from gaining higher privileges by running a setuid program (`nosuid` option):

```
# mount -t nfs -o ro,nosuid abc:/home /abc_home
```

For a list of options used for NFS mounts, see the MOUNT OPTIONS section of the `nfs(5)` man pages. For a list of client mount options, see the FILESYSTEM INDEPENDENT MOUNT OPTIONS section of the `mount(8)` man pages.

To mount NFS shares at boot time, add entries to the file system mount table, `/etc/fstab`. Entries are in the following format:

```
server:/exported-filesystem local_mount_point nfs options 0 0
```

For example, the `/etc/fstab` entry that replicates the `mount` command on the previous page is:

```
abc:/home /abc_home nfs ro,nosuid 0 0
```

The `df` command displays mounted file systems, including NFS-mounted file systems. For NFS mounts, the “File system” column displays the `server:/exported-filesystem` information. Use the `-T` option to include a “Type” column:

```
# df -hT  
Filesystem Type Size Used Avail Use% Mounted on  
...  
host03:/Dev nfs4 976M 2.5M 907M 1% /remote_dev
```

# Automounting File Systems

- Remote file systems are mounted only when accessed.
- Install the `autofs` package.
  - `autofs`: Kernel module
  - `automount`: User space daemon
- The main configuration file is `/etc/auto.master`.
- The format of `/etc/auto.master` entries is:

<code>/key</code>	<code>map-file</code>	<code>[options]</code>
-------------------	-----------------------	------------------------

- Example:

```
# cat /etc/auto.master
/- auto.direct
/misc  /etc/auto.misc
/net   -hosts
+auto.master
```

ORACLE

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

Automounting is an alternative to creating NFS mount entries in `/etc/fstab` or using the `mount` command from the command line to mount NFS shares. Automounting mounts remote file systems when they are accessed, rather than maintaining these remote mounts at all times. When the remote file systems are inactive, they are unmounted. This frees up system resources and improves overall system performance.

To implement automounting, first install the `autofs` package:

```
# yum install autofs
```

To start the `autofs` service:

```
# systemctl start autofs
```

The main configuration file, known as the master map file, is `/etc/auto.master`. This file lists mount points, known as keys, and corresponding map files that indicate which remote file systems can be mounted on the key. The format for entries in `/etc/auto.master` is:

<code>/key</code>	<code>map-file</code>	<code>[options]</code>
-------------------	-----------------------	------------------------

Automounting supports direct maps, indirect maps, and host maps. Direct maps use a special key, `/-`, in `/etc/auto.master`. Indirect maps specify a relative path name in their map files. Host maps use a special map, `-hosts`, in the `/etc/auto.master` file. Entries preceded with a plus sign (+) include a map from its source as if it were present in the master map.

## Direct Maps

- Direct maps always have a key of `/-` in `/etc/auto.master`. Example:

```
/-      auto.direct
```

- The sample entry in `auto.direct` map file:

```
/usr/man    -ro,soft      host01:/usr/man
```

- The format of direct and indirect map files:

key	[options]	location
-----	-----------	----------

- The “key” is the absolute path name of the mount point for direct mounts.
- The “location” is an exported NFS file system or a local file system of any supported file system type.
- Mount options included in map files override options specified in the master map file, `/etc/auto.master`.



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

The following entry in the `/etc/auto.master` file is an example of a direct map:

```
/-      auto.direct
```

Direct maps always have a key of `/-`. The map file in this example is `auto.direct`. With direct maps, the map file contains the absolute path name of the directory to be mounted. The following is an example of the contents of the `auto.direct` file:

```
/usr/man -ro,soft      host01:/usr/man
```

This entry mounts the file system `/usr/man` from the server `host01` on the local `/usr/man` mount point. `automount` creates the `/usr/man` directory if it does not already exist. If `/usr/man` does exist and is not empty, the mounted file system hides the local existing file system.

The `soft` option will allow an NFS request to fail after a specified number of retransmissions, after an NFS request times out. The number of retransmission is determined by the `retrans` option. If `retrans` is not specified, requests are tried three times.

The `hard` option (which is the default if neither `soft` nor `hard` is specified) allows unlimited NFS retransmission requests after an NFS request times out.

Direct map files and indirect map files have the following format:

key	[options]	location
-----	-----------	----------

The `key` can be a single directory name for an indirect map or the absolute path name of the mount point for direct mounts. Mount options can be included in map files. Any options specified in map files override options specified in the master map file. The location is the exported NFS file system, a local file system, or any other supported file system type.

## Indirect Maps

- Example of an indirect map entry in /etc/auto.master:

```
/misc      /etc/auto.misc
```

- Sample entries in the auto.misc map file:

xyz	-fstype=nfs	host01:/xyz
cd	-fstype=iso9660,ro,nosuid,nodev	:/dev/cdrom
abc	-fstype=ext3	:/dev/hda1

- The “key” in the indirect map file is relative to the autofs mount point, /misc, defined in /etc/auto.master.
- For example:
  - cd /misc/xyz mounts the /xyz directory from machine host01 locally on /misc/xyz.
  - cd /misc/abc mounts the ext3 file system on local device /dev/hda1 on /misc/abc.



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

The following entry in the /etc/auto.master file is an example of an indirect map:

```
/misc      /etc/auto.misc
```

Indirect maps are more common than direct maps. The following is an example of an indirect map file named /etc/auto.misc:

```
# cat /etc/auto.misc
xyz      -fstype=nfs                      host01:/xyz
cd       -fstype=iso9660,ro,nosuid,nodev    :/dev/cdrom
abc      -fstype=ext3                      :/dev/hda1
kernel   -ro,soft,intr                     ftp.kernel.org:/pub/linux
windoz   -fstype=smbfs                     ://windoz/c
```

The key field is relative to the actual location of the autofs mount point, /misc, from the master map file, /etc/auto.master.

For example, entering the cd /misc/xyz command mounts the /xyz directory from machine host01 locally on /misc/xyz. Only the /misc mount point needs to exist on the local machine. For indirect maps, the key is created when the file system is accessed and then removed when the file system is unmounted.

The second and third entries are examples of automounting local file systems:

```
cd      -fstype=iso9660,ro,nosuid,nodev          :/dev/cdrom  
abc     -fstype=ext3                           :/dev/hda1
```

The location field is the local file system path preceded with a colon (:). Entering the `ls /misc/cd` command would display the contents of the `iso` file on the `cdrom`. Entering the `ls /misc/abc` command would display the contents of the `ext3` file system on the `hda1` device.

The fourth line is an NFS mount (excluding the `-fstype` option defaults to NFS), which mounts the `/pub/kernel` directory from `ftp.kernel.org` on local mount point `/misc/kernel`:

```
kernel -ro,soft,intr           ftp.kernel.org:/pub/linux
```

The last line mounts a share exported from a Windows machine on `/misc/windoz`:

```
windoz -fstype=smbfs          :://windoz/c
```

## Host Maps

- Example of a host map entry in /etc/auto.master:

```
/net    -hosts
```

- The automount daemon creates a subdirectory under the “key” directory for every server listed in /etc/hosts.
- For example, entering the following command mounts all exports from host03 over the /net/host03 directory:

```
# cd /net/host03
```

- Entering the ls command lists exported file systems from host03.



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

The following entry in the /etc/auto.master file is an example of a host map:

```
/net          -hosts
```

When -hosts is given as the map, the automount daemon creates a subdirectory under the “key” directory, /net, for every server listed in the /etc/hosts file.

For example, entering the following command mounts all exports from host03 over the /net/host03 directory:

```
# cd /net/host03
```

All exports are mounted with the “no-suid, nodev, intr” options by default.

## Introduction to vsftpd

- vsftpd allows a system to function as an FTP server.
- vsftpd includes the following configuration files and directories:
  - /etc/vsftpd/vsftpd.conf
  - /etc/vsftpd/ftpusers
  - /etc/vsftpd/user\_list
  - /var/ftp
- To start the service:

```
# systemctl start vsftpd
```

- To start automatically at boot time:

```
# systemctl enable vsftpd
```



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

File Transfer Protocol (FTP) is a commonly used method of downloading and uploading files between systems on a network. FTP sites are typically public sites that allow anonymous users to log in and download software and documentation without needing a user account on the remote system.

The FTP server daemon included with Oracle Linux is called “very secure FTP” or vsftpd. To install the vsftpd package:

```
# yum install vsftpd
```

The following configuration files are installed with the package:

- **/etc/vsftpd/vsftpd.conf**: This is the main configuration file for vsftpd.
- **/etc/vsftpd/ftpusers**: This is a list of users not allowed to log in to vsftpd.
- **/etc/vsftpd/user\_list**: This file contains users who are denied access when the userlist\_deny directive is set to YES (default) in /etc/vsftpd/vsftpd.conf or users who are allowed access when userlist\_deny is set to NO.
- **/var/ftp**: The directory containing files served by vsftpd. It also contains the /var/ftp/pub directory for anonymous users.

To start the vsftpd service:

```
# systemctl start vsftpd
```

## vsftpd Configuration Options

- Local and anonymous users can download files by default.
  - local\_enable=YES
  - anonymous\_enable=YES
- Users can upload files by default, too.
  - write\_enable=YES
- Additional configuration parameters in /etc/vsftpd/vsftpd.conf include:
  - userlist\_enable
  - userlist\_deny
  - no\_anon\_password
  - xferlog\_enable
  - xferlog\_file



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

- **anon\_other\_write\_enable**: When set to `yes`, this setting allows anonymous users to make other changes to the file system, such as deleting, renaming, and modifying existing files.
- **anon\_upload\_enable**: This setting allows anonymous users to upload files to the server.
- **ascii\_download\_enable**: This setting allows conversion of text files transferred from the server to other operating systems. This can be a good idea if you are transferring text files from UNIX systems to Mac OS or Windows.
- **ascii\_upload\_enable**: This setting allows conversion of text files uploaded to the server.
- **xferlog\_enable**: This setting activates logging of uploads and downloads.
- **xferlog\_file**: This setting names the upload/download log file. The default is `/var/log/vsftpd.log`.



## Quiz

Which of the following statements are true?

- a. NFS allows Linux clients to mount exported file systems from remote Linux systems.
- b. Automounter allows NFS shares to be automatically mounted.
- c. The `vsftpd` daemon enables a system to be configured as an FTP server.
- d. All of the above



**ORACLE®**

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

## Summary

In this lesson, you should have learned how to:

- Describe NFS
- Configure the NFS server and client
- Describe the `exportfs` utility
- Explain and configure automounter
- Discuss and configure `vsftpd`



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

## Practice 14: Overview

This practice covers the following topics:

- Configuring an NFS server and an NFS client
- Removing the NFS Configuration
- Configuring an FTP server
- Downloading a file from an FTP server
- Restoring VMs to Original Configuration



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

# Kerberos and IPA Services



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

## Objectives

After completing this lesson, you should be able to:

- Configure Kerberos authentication
- Describe IPA Identity Management and Authentication Services
- Describe SSSD services and domains



ORACLE®

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

## The Kerberos Protocol

Is a network protocol created by Massachusetts Institute of Technology (MIT)

- Provides secure network connections between clients and servers using cryptography
- Clients authenticate their identities using a ticketing system.
- After authentication, messages between clients and servers are encrypted, averting network sniffing tools used by hackers.
- The three main configuration files are:
  - /etc/krb5.conf
  - /var/kerberos/krb5kdc/kdc.conf
  - /var/kerberos/krb5kdc/kadm5.acl
- For more information about the Kerberos protocol, please visit <https://web.mit.edu/kerberos>



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

Students learn how to configure Kerberos from the command line in the next set of slides.

# Configuring a Kerberos Server

- After installing the Kerberos server, edit the `/etc/krb5.conf` file to define the Key Distribution Center (KDC) server, `admin_server`, and Kerberos realm:

```
...
[realms]
EXAMPLE.COM = {
    kdc = krbsvr.example.com
    admin_server = krbsvr.example.com

[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
...
```



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

## Configuring a Key Distribution Center

- Port 88 is the default port for the Kerberos protocol.
- In /var/kerberos/krb5kdc/kdc.conf, assign the ports and define the access control list file in /var/kerberos/krb5kdc/kadm5.acl

```
[kdcdefaults]
  kdc_ports = 88
  kdc_tcp_ports = 88

[realms]
...
acl_file = /var/kerberos/krb5kdc/kadm5.acl
...
```



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

## Creating the Kerberos Database

The Kerberos database holds the following information:

- Principals
- Passwords
- Policies
- The `kdb5_util` command creates, deletes, loads, and dumps a database.
- To create a Kerberos database and store the database password in a stash file, run the following command:

```
# /usr/sbin/kdb5_util create -s
```

- Add an initial principal with administrative access in the `/var/kerberos/krb5kdc/kadm5.acl` file:

```
*/admin@EXAMPLE.com *
```



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

A stash file is an encrypted copy of the master key residing locally. The purpose of the stash key is to authenticate the KDC with itself. For security reasons, the stash file should be owned and readable by the root user and located only on the local disk of the KDC.

## The kadmin.local command

- After defining a principal user in the `/var/kerberos/krb5kdc/kadm5.acl` file, run the `kadmin.local` command to create a principal for each user:

```
# kadmin.local -q "addprinc root/admin"
```

- A keytab file provides a location to copy Kerberos server's key.
- You must create entries for the `kadmin/admin` and `kadmin/changepw` principals to the keytab file for authenticating with a KDC.

```
# kadmin.local -q "ktadd -k /etc/kadm5.keytab kadmin/root"
```

```
# kadmin.local -q "ktadd -k /etc/kadm5.keytab kadmin/changepw"
```



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

The `kadmin.local` and `kadmin` are commands for communicating with the Kerberos administration system. The main difference between these two commands is that `kadmin.local` directly accesses the KDC database, whereas `kadmin` communicates using `kadmind`, the daemon that starts the Kerberos administration server.

## Starting and Enabling Kerberos Services

- To start the KDC and administration services, run the following commands:

```
# systemctl start krb5kdc  
# systemctl start kadmin
```

- To enable both the KDC and administration services for starting at boot up, run the following:

```
# systemctl enable krb5kdc  
# systemctl enable kadmin
```



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

# Configuring Kerberos Authentication

- Realm
  - The realm for the Kerberos server
- Key Distribution Center (KDC)
  - A server that issues Kerberos tickets
- Admin Servers
  - The servers running the `kadmind` process in the realm



ORACLE®

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

LDAP, NIS, and IPA authentication support Kerberos authentication. Kerberos provides a secure connection over standard ports. It also uses credentials caching with the System Security Services Daemon (SSSD), which allows offline logins.

Install the following packages required for Kerberos authentication:

```
# yum install krb5-libs krb5-workstation
```

Select Kerberos password as the Authentication Method. You are then prompted for information required to connect to the Kerberos realm:

- **Realm:** The realm for the Kerberos server is the network that uses Kerberos. It is composed of Key Distribution Centers (KDCs) and client systems.
- **Key Distribution Center (KDC):** A comma-separated list of KDC servers that issue Kerberos tickets
- **Admin Servers:** A comma-separated list of administration servers that run the `kadmind` process in the realm

You can also select the check boxes to use DNS to resolve server host names and to find additional KDCs within the realm.

You can also use the `authconfig` command to configure Kerberos authentication.

# IPA Identity Management and Authentication Services

- There are two versions of Identity Policy Audit (IPA) that you can choose from the Authentication Configuration GUI:
  - FreeIPA (effectively IPAv1) and IPAv2
- These are open-source projects that combine the capabilities of Linux, LDAP (389 Directory Server), Kerberos, NTP, DNS, SSSD, and certificate authority functionality.
- You can use them to:
  - Enable single sign-on authentication for all your systems, services, and applications
  - Define Kerberos authentication and authorization policies for your identities
  - Control services like DNS, sudo, SELinux, and autofs
  - Integrate with Microsoft Active Directory
- See <https://www.freeipa.org> for more information.



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

There are two versions of IPA that you can choose from the Authentication Configuration GUI: FreeIPA (effectively IPAv1) and IPAv2. These are open-source projects that provide centralized identity management and authentication services. They combine the capabilities of Linux, LDAP (389 Directory Server), Kerberos, NTP, DNS, SSSD, and certificate authority functionality.

FreeIPA and IPAv2 allow you to enable single sign-on authentication for all your systems, services, and applications. They allow you to define Kerberos authentication and authorization policies for your identities. You can control services like DNS, sudo, SELinux, and autofs. You can also integrate FreeIPA and IPAv2 with other Identity Management systems like Microsoft Active Directory.

FreeIPA supports identity management and authentication of users and groups and does not require you to join your system to an IPA realm. When you select FreeIPA as the user account database, you are prompted to enter information about LDAP and Kerberos configurations.

IPAv2 supports identity management and authentication of machines and requires you to join your system to an IPA realm. You are prompted to enter information about the IPA domain configuration, optionally choose to configure NTP, and click Join Domain to create a machine account on the IPA server. After your system has obtained permission to join the IPA realm, you can select and configure the authentication method.

## Installing IPA Server with DNS

Besides managing users' identities, an IPA server can also manage a DNS server and DNS records.

- To install the IPA server with DNS, run the following command:

```
# yum install ipa-server bind bind-dyndb-ldap  
      ipa-server-dns
```

- After installing the required IPA packages, run the install script with the option for setting up DNS:

```
# ipa-server-install --setup-dns
```

- During the running of the install script, answer the following questions as shown:

```
Do you want to configure DNS forwarders? [yes] no
```

```
Do you want to configure the reverse zone? [yes] yes
```



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

For authentication purpose, you must run the following command to receive a ticket:

```
# kinit admin
```

Next, start your preferred browser and visit 127.0.0.1 for opening the web-based user interface. To log in, use the IPA admin account and password.

## Installing IPA Client

IPA clients must authenticate with an IPA server before joining a realm.

- You must first install the IPA client using the following command:

```
# yum ipa-client
```

- After installing the IPA client, run the install script:

```
# ipa-client-install
```

- During the installation, accept the default values by answering the following question as follows:

```
Continue to configure the system with these values? [no]yes
```

- Before authenticating with the IPA server, enter the password for the admin user for the IPA server as follows:

```
Password for admin@EXAMPLE.COM: *****
```



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

## IPA Client Discovery

IPA clients must first discover IPA servers in order to authenticate for joining a new realm.

- To discover an IPA server, run the following script command:

```
# ipa-client-install
```

- Once an IPA server is discovered, you are presented with configuration settings for accepting as shown below:

```
Discovery was successful!
```

```
Client hostname: host02.example.com
```

```
Realm: EXAMPLE.com
```

```
DNS Domain: example.com
```

```
IPA Server: host01.example.com
```

```
BaseDN: dc=example,dc=com
```

```
Continue to configure the system with these values? [no] yes
```



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

# Configuring Advanced Options

**ORACLE®**

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

The slide shows the Advanced Options tab from the Authentication Configuration GUI. On this tab, you can configure local authentication options that define authentication behavior on the local system. You can also configure your system to automatically create home directories the first time that a user logs in, and you can enable smart card authentication. Each of these configuration options is discussed.

## Enable Fingerprint Reader Support

Assuming that the appropriate hardware is in place, this allows fingerprint scans to be used to authenticate local users rather than using other credentials. Use the following command to enable fingerprint reader support, from the command line:

```
# authconfig --enablefingerprint --update
```

## Enable Local Access Control

This checks the `/etc/security/access.conf` file for local user authorization rules. This file specifies combinations for logins that are accepted or refused. The syntax of the entries is:

```
permission : users : origin
```

A plus sign (+) in the `permission` field means that the login is granted. Login is denied if the field contains a minus sign (-). The `users` field can be a username, group, or the `ALL` keyword. The `origin` field is a host name, network, TTY (terminal), or the `ALL` or `NONE` keywords.

## Password Hashing Algorithm

This sets the hashing algorithm to use to encrypt locally stored passwords. The options are:

- DESCRIPT
- BIGCRYPT
- MD5
- SHA256
- SHA512

You can also look at the password field in `/etc/shadow` to determine the algorithm. The field starts with a specific set of characters, depending on the hashing algorithm used, for example:

- MD5 starts with `$1$`
- SHA-256 starts with `$5$`
- SHA-512 starts with `$6$`

To determine the current algorithm from the command line:

```
# authconfig --test |grep hashing
```

You can also change the hashing algorithm from the command line. The following example changes it to SHA512:

```
# authconfig --passalgo=sha512 --update
```

## Other Authentication Options

To enable the creation of user home directories at the first login, from the command line:

```
# authconfig --enablemkhomedir --update
```

## Smart Card Authentication Options

A system can accept smart cards (or tokens) to authenticate users. The appropriate hardware must be available, and the following package must be installed:

```
# yum install pam_pkcs11
```

Enabling smart card support prompts for additional configuration information:

- **Require smart card for login:** This disables Kerberos password authentication and all other methods of authentication for logging in to the system.
- **Card removal action:** Sets the system's response to a smart card being removed during an active session. Options are `Ignore`, meaning that the system continues functioning, and `Lock`, which immediately locks the screen.

To enable smart card use, from the command line:

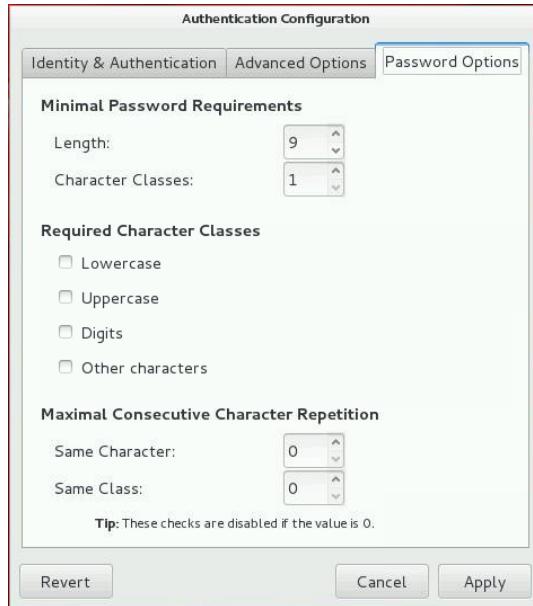
```
# authconfig --enablesmartcard --update
```

To enable smart cards and lock the system when the smart card is removed:

```
# authconfig --enablesmartcard --smartcardaction=0 --update
```

Setting `--smartcardaction=1` does not lock the system when the smart card is removed.

# Configuring Password Options

**ORACLE®**

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

The slide shows the Password Options tab from the Authentication Configuration GUI. From this tab, you can configure password complexity, which is a combination of length and variation of character classes. You can specify what types of characters can be used in a password and how those characters can be used within the password.

## Minimal Password Requirements

Use the “Length” field to specify the minimum length of the password. Use the “Character Classes” field to specify the minimum number of character classes, which must be used in the password.

## Required Character Classes

In this section, you can enable character classes, which must be used for passwords. For example, if the “Digits” check box is selected, a digit must be used in every password. All character types are allowed, but the selection of a character class means the character class is required.

## Maximal Consecutive Character Repetition

Set the number of times that a character or character class can be repeated consecutively. A zero setting means there is no repeat limit. The “Same Character” field sets how often a single character can be repeated. The “Same Class” field sets how many times any character from a character class can be repeated.

Password complexity can also be configured from the command line by using the `authconfig` command with the following options:

- `--passminlen`: The minimum length of a password
- `--passminclass`: The minimum number of different types of characters that must be used in a password
- `--passmaxrepeat`: The number of times a character can be repeated consecutively
- `--passmaxclassrepeat`: The number of times the same character can be used
- `--enablerequpper`: The password requires uppercase letters
- `--enablereqlower`: The password requires lowercase letters
- `--enablereqdigit`: The password requires numbers
- `--enablereqother`: The password requires special characters

The following example sets the minimum length of the password to eight characters, requires three different types of character classes, does not allow characters of classes to be repeated more than two times, and requires both numbers and special characters.

```
# authconfig --passminlen=8 --passminclass=3 --passmaxrepeat=2  
--passmaxclassrepeat=2 --enablereqdigit --enablereqother  
--update
```

# System Security Services Daemon

System Security Services Daemon (SSSD) provides access to remote identity and authentication providers.

- SSSD acts as an intermediary between local clients and any back-end providers.
  - Reduces the load on back-end providers
  - Allows offline authentication
  - Allows for single-user accounts
- Install the packages:

```
# yum install sssd sssd-client
```

- Start the service:

```
# authconfig --enablerssd --update
```



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

The System Security Services Daemon (SSSD) provides access to remote identity and authentication providers. Providers are configured as back ends with SSSD acting as an intermediary between local clients and any configured back-end provider. The local clients connect to SSSD, and then SSSD contacts the providers. Benefits of SSSD include:

- **Reduced load:** Clients do not have to contact the identification/authentication servers directly; they need to contact only SSSD.
- **Offline authentication:** SSSD can, optionally, keep a cache of user identities and credentials, allowing users to authenticate offline.
- **Single-user accounts:** SSSD maintains network credentials, allowing users to connect to network resources by authenticating with their local username on their local machine.

Install the following SSSD packages:

```
# yum install sssd sssd-client
```

To cause SSSD to start when the system boots, enter either of the following:

```
# systemctl enable sssd
# authconfig --enablerssd --update
```

# Configuring SSSD Services

The main configuration file is `/etc/sssd/sssd.conf`.

- SSSD services are configured in separate sections of this file.
- `[sssd]` section specifies:
  - Specialized services that run together with SSSD
  - Identity domains
- `[nss]` section:
  - Configuration parameters for NSS service
- `[pam]` section:
  - Configuration parameters for PAM service



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

The main configuration file for SSSD is `/etc/sssd/sssd.conf`. SSSD services and domains are configured in separate sections of this file, each beginning with a name of the section in square brackets. The following are examples:

```
[sssd]
```

```
[nss]
```

```
[pam]
```

## **[sssd] Section**

SSSD functionality is provided by specialized services that run together with SSSD. These specialized services are started and restarted by a special service called “monitor.” Monitor options and identity domains are configured in the `[sssd]` section of `/etc/sssd/sssd.conf`. The following is an example:

```
[sssd]
domains = LDAP
services = nss, pam
```

The `domains` directive can define multiple domains. Enter them in the order in which you want them to be queried. The `services` directive lists the services that are started when `sssd` itself starts.

## Services Sections

Each of the specialized services that run together with SSSD is configured in separate sections in `/etc/sssd/sssd.conf`. For example, the `[nss]` section is used to configure the Name Service Switch (NSS) service. The `[pam]` section is used to configure the PAM service.

### Configuring the NSS Service

Included in the `sssd` package is an NSS module, `sssd_nss`, which instructs the system to use SSSD to retrieve user information. This is configured in the `[nss]` section of `/etc/sssd/sssd.conf`. The following is an example that includes only a partial list of configurable directives:

```
[nss]
filter_groups = root
filter_users = root
reconnection_retries = 3
entry_cache_timeout = 300
```

The `filter_users` and `filter_groups` directives tell SSSD to exclude certain users and groups from being fetched from the NSS database. The `reconnection_retries` directive specifies the number of times to attempt to reconnect in the event of a data provider crash. The `enum_cache_timeout` directive specifies, in seconds, how long `sssd_nss` caches request information about all users.

### Configuring the PAM Service

The `sssd` package also provides a PAM module, `sssd_pam`, which is configured in the `[pam]` section of `/etc/sssd/sssd.conf`. The following is an example that includes only a partial list of configurable directives:

```
[pam]
reconnection_retries = 3
offline_credentials_expiration = 2
offline_failed_login_attempts = 3
offline_failed_login_delay = 5
```

The `offline_credentials_expiration` directive specifies, in days, how long to allow cached logins if the authentication provider is offline.

The `offline_failed_login_attempts` directive specifies how many failed login attempts are allowed if the authentication provider is offline.

To update the PAM configuration to reference all of the SSSD modules, use the `authconfig` command as follows to enable SSSD for system authentication:

```
# authconfig --update --enablenesssd --enablesssdauth
```

This command auto-generates the PAM configuration file to include the necessary `pam_sss.so` entries.

# Configuring SSSD Domains

SSSD domains are also configured in separate sections of `/etc/sssd/sssd.conf`.

- The syntax is:

```
[domain/Name]
  id_provider = type
  auth_provider = type
  provider_specific = value
  global = value
```

- Supported identity providers include `ldap`, `local`, or `proxy`.
- Supported authentication providers include `ldap`, `krb5`, `proxy`, or `none`.
- Provider-specific directives apply to the specified provider.
- Global directives apply to all domains.



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

SSSD domains are a combination of an identity provider and an authentication method. SSSD works with LDAP identity providers (including OpenLDAP, Red Hat Directory Server, and Microsoft Active Directory) and can use native LDAP authentication or Kerberos authentication. When configuring a domain, you define both where the user information is stored and how those users are allowed to authenticate to the system.

Similar to SSSD services, SSSD domains are also configured in separate sections of the `/etc/sssd/sssd.conf` file. The services and the domains are identified in the `[sssd]` section. Example:

```
[sssd]
domains = LDAP
services = nss, pam
```

This example specifies an LDAP domain. The domain section of the configuration would begin with the following header:

```
[domain/LDAP]
```

The domain configuration section would then specify the identity provider, the authentication provider, and any specific configuration to access the information in those providers.

The following is an example of a domain section:

```
[domain/LDAP]
id_provider = ldap
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
auth_provider = krb5
krb5_server = kerberos.example.com
krb5_realm = EXAMPLE.COM
min_id = 10000
max_id = 20000
```

### Identity Provider

The `id_provider` specifies the data provider identity back end to use for this domain. Supported back ends are:

- `proxy`: Supports a legacy NSS provider
- `local`: SSSD internal local provider
- `ldap`: LDAP provider

The `ldap_uri` directive gives a comma-separated list of the URIs (Universal Resource Identifiers) of the LDAP servers, in order of preference, to which SSSD connects.

The `ldap_search_base` directive gives the base DN to use for performing LDAP user operations.

### Authentication Provider

The `auth_provider` directive specifies the authentication provider used for the domain. If un-specified, the `id_provider` is used. Supported authentication providers are:

- `ldap`: Native LDAP authentication
- `krb5`: Kerberos authentication
- `proxy`: Relays authentication to some other PAM target
- `none`: Disables authentication explicitly

The `krb5_server` directive gives a comma-separated list of Kerberos servers, in order of preference, to which SSSD connects.

The `krb5_realm` directive gives the Kerberos realm to use for Simple Authentication and Security Layer (SASL)/Generic Security Services API (GSS-API) authentication. Configuration of SASL connections by using GSS-API is required before SSSD can use Kerberos to connect to the LDAP server.

The last two directives, `min_id` and `max_id`, are examples of global attributes that are available to any type of domain. Other attributes include cache and timeout settings. These two directives specify UID and GID limits for the domain. If a domain contains an entry that is outside these limits, it is ignored.

Start or restart the `sssd` service after making any configuration changes to domains or services:

```
# systemctl start sssd
```

## Quiz



Which two versions of Identity Policy Audit (IPA) are available from the Authentication Configuration GUI?

- a. FreeIPA
- b. IPA v2
- c. SSSD
- d. DNS

ORACLE®

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

## Summary

In this lesson, you should have learned how to:

- Configure Kerberos authentication
- Describe IPA Identity Management and Authentication Services
- Describe SSSD services and domains



ORACLE®

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

## Practice 15: Overview

- Kerberos Authentication
- Installing Identity Management
- Configuring SSSD Services



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.