ORACLE
UNIVERSITY

Oracle Linux 7: System Administration

Student Guide | Volume 1

D88168GC10
Edition 1.0 | January 2015 | D89906

Learn more from Oracle University at **oracle.com/education/**

ORACLE

Author

Craig McBride

Technical Contributors and Reviewers

Yasar Akthar
Gavin Bowe
Avi Miller
Chris Potter
Tim Hill
Manish Kapur
Wim Coekaerts
Al Flournoy
Joel Goodman
Harald Van Breederode
Michele Dady
Steve Miller
Antoinette O'Sullivan

Editors

Malavika Jinka
Raj Kumar
Smita Kommini

Graphic Designer

Seema Bopaiah

Publishers

Joseph Fernandez
Giri Venugopal

Contents

12 XFS File System

13 Btrfs File System

1

Course Introduction

Course Objectives

After completing this course, you should be able to:

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

Course Schedule

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

Course Schedule

Session	Module
Day 3	Lesson 11: Partitions, File Systems, and Swap Lesson 12: Implementing the XFS File System Lesson 13: Implementing the Btrfs File System Lesson 14: Storage Administration
Day 4	Lesson 15: Network Configuration Lesson 16: File Sharing Lesson 17: OpenSSH Lesson 18: Security Administration

Course Schedule

Session	Module
Day 5	Lesson 19: Oracle on Oracle Lesson 20: System Monitoring Lesson 21: System Logging Lesson 22: Troubleshooting

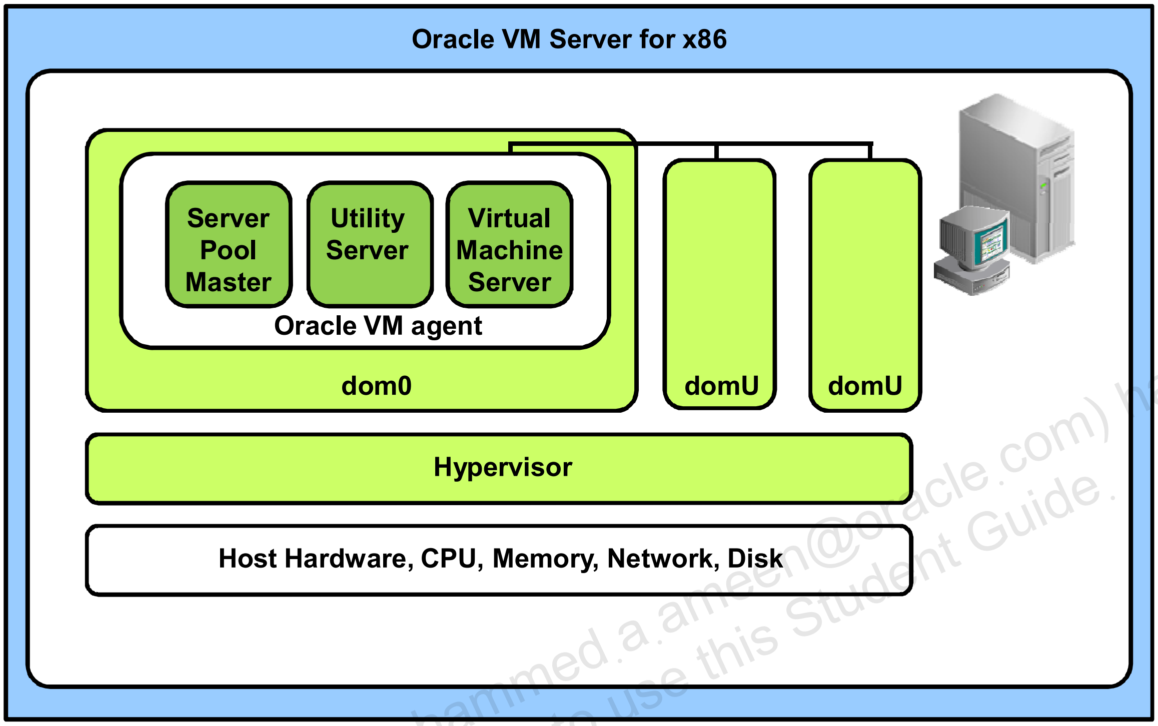
ORACLE

Lesson Objectives

After completing this lesson, you should be able to:

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

Virtualization with Oracle VM Server for x86



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

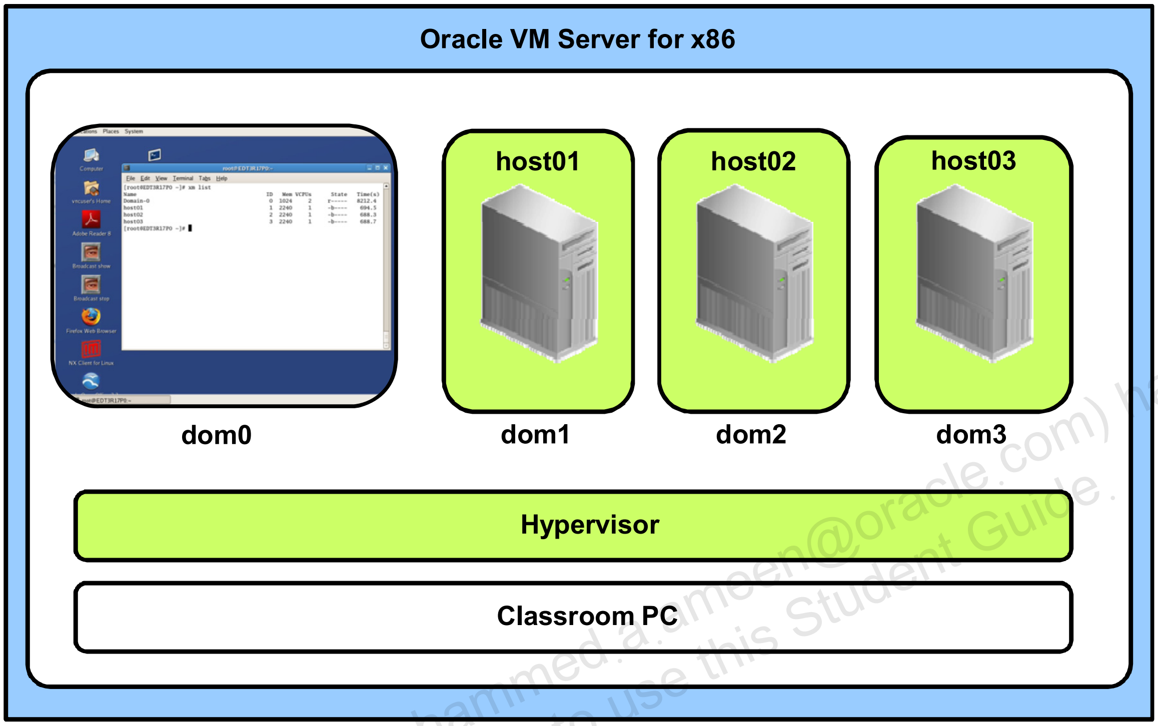
A hypervisor is virtualization software, also known as a virtual machine monitor (VMM), that creates and runs the virtual machines. There are two different types of hypervisors:

- 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.

Oracle VM Server for x86 in the Classroom



Self-Contained Multi-Host Environment

Your student PC is running Oracle VM Server for x86, where you can run up to three guests (as required) to work through the practice sessions. Guests running on your machine can see each other and can see outside the environment. Out of the box, Oracle VM Server for x86 does not offer a GUI front end; however, your dom0 has been modified to include the Gnome interface. When you log in to the machine, you are presented with a graphical interface that can also act as an X-server for your guests.

Logging In to Your Machine

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

Where to Find Your Guests

The guest VMs reside in their own directories under the `/OVS/running_pool` directory on dom0. For example, the files for the host01 VM reside in the `/OVS/running_pool/host01` directory and the files for the host02 VM reside in the `/OVS/running_pool/host02` directory.

Working with Classroom Virtual Machines

- Use the `xm` command-line tool to manually manage guests.
  - `xm list`: Lists the currently active guests
  - `xm create vm.cfg`: Starts a VM guest
  - `xm shutdown -w <VM_name>`: Gracefully shuts down the specified VM
  - `xm destroy <VM_name>`: Non-graceful shut down
  - `xm reset <VM_name>`: Resets the specified VM
- Use the following commands to connect to VM guests.
  - `ssh <VM_name>`
  - `vncviewer` (provide `localhost:<VNC_port_number>` when prompted)
- Each practice specifies whether to use `ssh` or `vncviewer`.

Starting, Stopping, and Listing Guests

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

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

Connecting to Guests

The practice exercises direct you to become the `root` user on dom0 and use secure shell (`ssh`) or `vncviewer` to connect from dom0 to your guests. For example, to use secure shell to create a connection from dom0 to the host01 guest:

```
# ssh host01
```

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

Using `vncviewer`

In some of the practices, you are instructed to use the `vncviewer` command, instead of using the `ssh` command, to connect to your guest VM. You can use `vncviewer` in all practices, but it is required only when specified in the applicable practices.

Before running the `vncviewer` command, you first must determine the port number by running the following command from dom0:

```
# xm list -l host03 | grep location
           (location 0.0.0.0:5904)
           (location 3)
```

In this example, the port number for host03 is `5904`. Each host has a different port number. Port numbers are assigned when the virtual machine is started. Therefore, run the above `xm list` command before running `vncviewer` because the port number can change.

Run the `vncviewer` command and you are prompted for **VNC Viewer: Connection Details**. Enter `localhost:<port_number>`, substituting the port number obtained from the previous `xm list` command. Example:

```
# vncviewer&
localhost:5904
```

Instead of entering `5904` as the port number, you can omit the "`590`" and enter the "`4`" as follows:

```
localhost:4
```

You are instructed to use `vncviewer` in the practices that require access to the GNOME desktop.

Summary

In this lesson, you should have learned how to:

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

Practices: Overview

This practice covers the following topics:

- Logging in to your classroom PC *
- Exploring the dom0 configuration and directory structure
- Starting, stopping, and listing VM guests
- Logging in to each VM guest
- Exploring the VM configurations
- Logging off from your classroom PC

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

ORACLE

2

Introduction to Oracle Linux

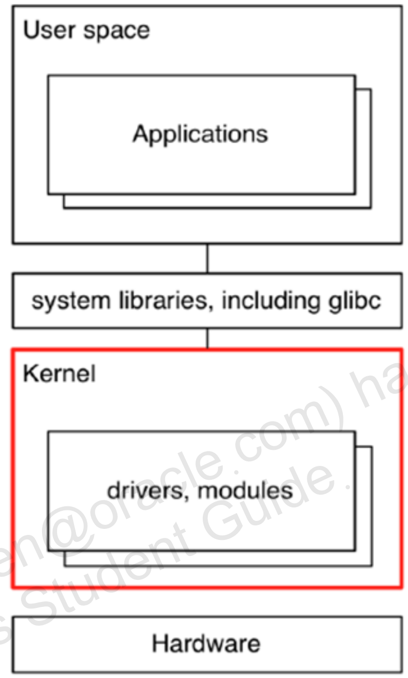
Objectives

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

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

Linux Kernel

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



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

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

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

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

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

"Hello everybody out there using minix –

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

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

Linus (torvalds@kruuna.helsinki.fi)"

The GNU Project

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

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

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

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

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

GNU General Public License (GPL)

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

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

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

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

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

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

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

Linux Kernel Development Model

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

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

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

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

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

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

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

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

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

Continuous Mainline Kernel Development

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

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

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

Oracle Linux 7 is supported on the x86-64 (64 bit) hardware architecture. Previous versions of Oracle Linux are also supported on x86 (32 bit) architecture. For a list of Oracle Linux supported releases by hardware platform, see https://linux.oracle.com/supported.html.

Linux Distributions

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

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

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

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

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

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

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

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

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

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

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

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

Oracle Linux

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

Oracle offers a comprehensive Linux solution including:

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

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

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

Oracle's Technical Contributions to Linux

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

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

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

- **ASMLib:** A support library for the Automatic Storage Management feature of Oracle Database that simplifies database administration. In Oracle Database 12c there is a new ASM filter driver (ASMFD) that prevents accidental corruption or deletion of the ASM devices. Refer to the following for more information: http://docs.oracle.com/database/121/LADBI/oraclerestart.htm#LADBI8076.
- **Asynchronous IO (AIO) Kernel Subsystem:** Used to make system calls asynchronous in a generic fashion to ensure that Oracle databases run properly on Linux
- **Btrfs file system:** Designed to address the expanding scalability requirements of large storage subsystems

- **Ext3 file system data-guarded mode:** Oracle's Linux Engineering Team has proposed this new ext3 mode. It maintains all the security protections of data-ordered mode, without requiring all the dirty data on the file system to be written during a single fsync.
- **Kernel I/O Subsystem Tuning:** Oracle Linux kernel engineers are working on creating novel approaches in the area of block I/O, to fully exploit the higher disk speeds of Solid State Disks (SSD).
- **Libstdc++:** Oracle is a major contributor and maintainer of this GNU standard C++ library.
- **NFS on IPv6:** Oracle Linux kernel engineers are working to enable the Linux Network File System (NFS) to run natively on IPv6 networks. The maintainer of NFS, Chuck Lever, is an Oracle employee.
- **Oracle Cluster Filesystem (OCFS) 2 v.1.4:** OCFS2 is an open source, general-purpose, extent-based clustered file system that Oracle developed and contributed to the Linux community. It was accepted into Linux kernel 2.6.16.
- **Oracle Linux Test (OLT) Kit:** Available as open source under the GPL and Artistic licenses, the Oracle Linux Test Kit, derived from the Oracle Validated Configurations program, is designed to verify Linux kernel functionality and stability essential for Oracle Database.
- **Oracle-Validated Configurations for Linux and Virtualization:** These are pre-tested, validated architectures with software, hardware, storage, and networking components with documented best practices for deployment included.
- **PHP:** The Oracle Linux engineering team devotes resources to the improvement and maintenance of PHP and its Oracle-specific extensions. Newer PHP packages in RPM format are available to Linux users for free download.
- **RDS:** Reliable Datagram Sockets (RDS) is an effort to provide a socket API that is uniquely suited to the way Oracle does network Interprocess Communication (IPC). The Oracle Linux kernel development team created an open source implementation of the API for the Linux kernel. The code is now integrated into the OpenFabrics Enterprise Distribution (OFED) stack. OFED aims to deliver a unified, cross-platform, transport-independent software stack for remote directory memory access (RDMA), including a range of standard protocols.
- **T10 Protection Information Model (also known as DIF):** Oracle, in collaboration with Emulex, is implementing a leading, first-of-its-kind initiative to bring enterprise-class data integrity to the Linux platform. An open source interface is being implemented by Oracle to expose the T10 Protection Information Model (also known as DIF—data integrity framework) standard to the Linux kernel and end-user applications. See the following for more information: http://oss.oracle.com/~mkp/.
- **Testing of Open Source Projects:** Testing the mainline kernel is essential so the Linux community can get a long-term regression picture of how the kernel performs and works. Mainline kernel testing and quality assurance (QA) benefit the entire community.
- **Yet Another Setup Tool (YaST):** YaST helps make system administration easier by providing a single utility for configuring and maintaining Linux systems. Available under GPL, this code can be freely accessed by anyone. The Oracle Linux Engineering team ported the YaST to OL from SUSE. Oracle Linux support customers have access to the YaST functionality integrated with the Oracle Management Pack for Linux.
- **Xen Hypervisor:** Consisting of Xen's open source server software and an integrated web browser–based management console, Oracle VM is free, scalable server virtualization software that supports Oracle and non-Oracle applications. Oracle's engineering team contributes heavily to feature the development of Xen mainline software.

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

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

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

Strict Binary Compatibility

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

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

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

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

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

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

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

Fully Compatible Updates and Errata

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

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

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

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

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

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

Unbreakable Enterprise Kernel

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

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

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

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

Using the Unbreakable Enterprise Kernel instead of the Red Hat compatible kernel changes only the kernel. Nothing changes in the user space. Existing applications run unchanged regardless of which kernel is used. Using a different kernel does not change system libraries such as `glibc`. The `glibc` version in Oracle Linux 7 is 2.17, regardless of the kernel version.

Unbreakable Enterprise Kernel Release 1

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

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

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

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

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

Latest OFED Stack (1.5.1)

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

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

Advanced Support for Large NUMA Systems

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

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

Receive Packet Steering (RPS)

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

To enable RPS, you have to place a CPU mask into:

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

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

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

Receive Flow Steering (RFS)

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

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

SSD Detection

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

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

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

Data Integrity Up to SAN

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

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

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

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

OCFS2 1.6

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

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

Unbreakable Enterprise Kernel Release 2

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

ORACLE

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

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

DTrace

DTrace allows static and dynamic tracing of your applications and your kernel. It does static tracing by allowing instrumentation compiled into the kernel and the applications. At specific points of execution in your code, you can activate the probes and designate actions, such as collecting and displaying information.

Btrfs File System

Btrfs is a general-purpose file system that scales very well for large systems and large file systems. Btrfs provides extent-based file storage with a maximum file size of 50 TB and a maximum file system size of 50 TB. All data and metadata is copy-on-write.

Transcendent Memory: Cleancache

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

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

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

Resource Isolation: Cgroups

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

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

OS Isolation: Linux Containers

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

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

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

Transparent Huge Pages

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

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

Transmit Packet Steering (XPS)

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

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

Networking: Built in Virtual Switch

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

Other Scalability Improvements

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

Unbreakable Enterprise Kernel Release 3

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

ORACLE

The Unbreakable Enterprise Kernel Release 3 (UEK R3) is Oracle's third major release of its heavily tested and optimized operating system kernel for Oracle Linux 6 on the x86_64 architecture. It is based on the mainline Linux version 3.8.13.

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

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

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

Technology Preview

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

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

Unbreakable Enterprise Kernel Release 3 can be installed on Oracle Linux 6 Update 4 or newer, running either the Red Hat–compatible kernel or a previous version of Unbreakable Enterprise Kernel. Unbreakable Enterprise Kernel Release 3 is supported on the x86_64 architecture but not on x86.

Release notes are available at http://docs.oracle.com/cd/E52668_01/index.html.

Oracle Linux Release Notes

- Oracle publishes release notes for each version.
- Release notes and all product documentation for Oracle Linux 7 are available at: http://docs.oracle.com/cd/E52668_01/.
- New Features and Changes
  - Notable changes from Oracle Linux 6
- Known Issues
- Installation and Availability
  - Upgrading from Oracle Linux 6
- Changes from the Upstream Release:
  - Removed, modified, new packages from upstream release
  - Packages added by Oracle

ORACLE

Refer to the release notes for information regarding each release of Oracle Linux. For example, the following is part of “Notable Changes from Oracle Linux 6.”

Exporting System Configuration Parameters

Parameters defined in `/etc/sysconfig` files are automatically exported in Oracle Linux 7. You no longer need to use the `export` command.

Host Name Configuration

The host name is now defined in `/etc/hostname` instead of in `/etc/sysconfig/network`.

Predictable Network Interface Naming

Network interface names are now based on information derived from the system BIOS or alternatively from a device's firmware, system path, or MAC address. This feature ensures that interface names persist across system reboots, hardware reconfiguration, and updates to device drivers and the kernel.

NFS v2

NFS v2 is not supported for use with Oracle Linux 7. You should use NFS v3 or NFS v4 instead.

`ifconfig` Output Has Changed.

It is recommendation to use the `ip` command instead.

Initial Setup Supersedes Firstboot

Although legacy support for Firstboot allows third-party modules to continue to function, it is recommended that you rewrite these modules to work with the installer and Initial Setup.

Layout of the root File System

As `initrd` is now able to mount the `/usr` file system at boot time, the files in `/bin`, `/lib`, `/lib64`, and `/sbin` have been moved to `/usr/bin`, `/usr/lib`, `/usr/lib64`, and `/usr/sbin`, respectively. Symbolic links in `/` provide backward compatibility for programs. For example:

```
# ls -ld /bin
lrwxrwxrwx ... /bin -> usr/bin
# ls -ld /lib
lrwxrwxrwx ... /lib -> usr/lib
# ls -ld /lib64
lrwxrwxrwx ... /lib64 -> usr/lib64
# ls -ld /sbin
lrwxrwxrwx ... /sbin -> usr/sbin
```

Localization Settings

System-wide default localization settings such as the default language, keyboard, and console font are now defined in `/etc/locale.conf` and `/etc/vconsole.conf` instead of in `/etc/sysconfig/i18n`.

System Logging

The new logging daemon, `journald`, records system messages in non-persistent journal files in memory and in `/run/log/journal`. The `journald` daemon forwards messages to `rsyslog`, which processes and archives only `syslog` messages by default. If required, you can configure `rsyslog` to archive any other messages that `journald` forwards, including kernel, boot, `initrd`, `stdout`, and `stderr` messages.

`systemd` Replaces Upstart and `init`

The `systemd` daemon replaces Upstart for managing system run levels and services. Replacing `init`, `systemd` is the first process that starts after the system boots, and is the final process that is running when the system shuts down. `systemd` controls the final stages of booting and prepares the system for use. `systemd` also speeds up booting by loading services concurrently.

Many of these notable changes are covered in detail in subsequent lessons in this course.

Summary

In this lesson, you should have learned:

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

Quiz

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

a. True
b. False

Quiz

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

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

Practice 2: Overview

The practices cover the following topics:

- Introduction to Oracle Linux quiz
- Viewing kernel information

3

Installing Oracle Linux 7

Objectives

After completing this lesson, you should be able to:

- Obtain Oracle Linux 7 operating system software
- Describe the Anaconda installer
- Install Oracle Linux 7
- Describe the Firstboot utility

Many of the steps required during installation, such as configuring the network, partitioning storage devices, creating file systems, configuring the GRand Unified Bootloader (GRUB), and installing software packages are covered in later lessons. The objective of this lesson is familiarize you with the Oracle Linux 7 installation process.

Obtaining Oracle Linux

- Obtain Oracle Linux from:
  - Oracle Software Delivery Cloud: https://edelivery.oracle.com/linux/
  - Oracle Public Yum Server: http://public-yum.oracle.com
  - Source and debug information available from: http://oss.oracle.com/
- Obtain errata for free from http://public-yum.oracle.com.
  - Subscribe to Oracle Linux errata mailing list from this site.
- Oracle Linux 7 is available for x86 64-bit systems.

ORACLE

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

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

A new set of ISO DVD images is made available for free download for every minor release (for example, Oracle Linux 6 Update 5) of Oracle Linux. You can obtain errata (bug fixes, security fixes, enhancements) updates for free and you can subscribe to the Oracle Linux errata mailing list from the public yum server.

Oracle Software Delivery Cloud

Oracle Linux 7 download page: Example

This screen lists the download options for Oracle Linux 7 from the Oracle Software Delivery Cloud: http://edelivery.oracle.com/linux. As shown, Oracle Linux 7 comes in two DVD ISO images for x86_64 bit architecture. One is the complete distribution (part # V46135-01) and the other is the boot ISO image. The full distribution ISO is what is typically needed for installations of Oracle Linux 7. The source code is also available for download from this page.

The boot ISO image is 343 MB and contains only the boot kernel and installer. It does not have all the software packages (RPMs). You can use this image file to produce minimal boot media such as bootable CDs, DVDs, or USB devices with which you can boot a system when you plan to complete the installation from an installation source available on a hard disk or over a network connection. You do not need to download the source DVDs to install Oracle Linux.

To install Oracle Linux 7 from DVD media, download the compressed binary DVD images. Verify the downloaded media file by comparing its `sha1sum` or `md5sum` with the published `sha1sum` or `md5sum`. Use DVD burning software to write the DVD image directly to DVD. Do not copy the files onto the DVD. You must use a DVD burner that can accept an ISO image as input, and that can create a bootable DVD from it. To test if you have burned the images correctly, insert a burned DVD and ensure that multiple files and directories are visible. Insert the DVD into your system, boot from DVD, and follow the on-screen instructions to deploy Oracle Linux.

Anaconda Installer

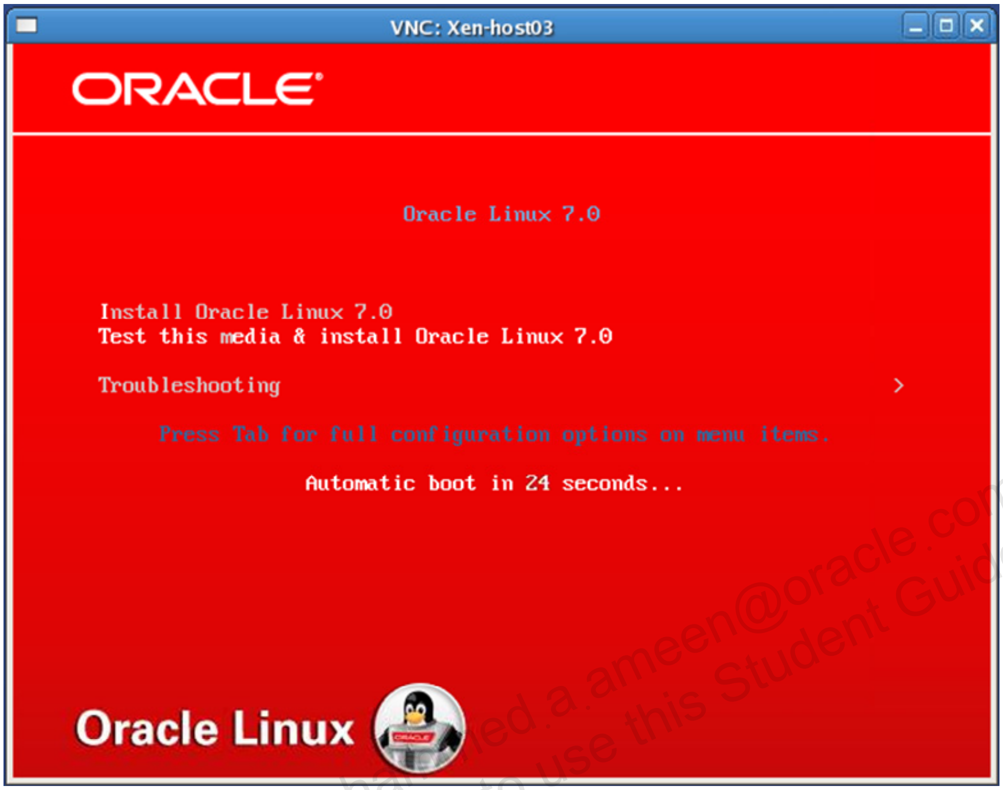
Anaconda:

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

ORACLE

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

Oracle Linux 7.0 Installation Menu



This slide shows the Oracle Linux 7.0 installation menu, which is the first window to appear during an installation. The default option of “Test this media & install Oracle Linux 7.0” is automatically selected within 60 seconds. Press Esc at the boot menu to get a boot prompt.

Install Oracle Linux 7.0

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

Test this media & install Oracle Linux 7.0

Use this default option to check the integrity of the installation media before starting the installation. Testing takes a few extra minutes but is worthwhile if you have concerns about corruption or errors with the media.

The **Troubleshooting** option displays another menu that contains the following options:

- **Install in basic graphics mode:** Select if the default option causes a distorted or blank screen, which is the result of the inability to load the correct driver for your video card.
- **Rescue installed system:** Select if you are unable to boot an installed system. Rescue allows you to repair partitions, edit configuration files, and fix a variety of boot problems.
- **Memory test:** Select to run a memory test utility to verify RAM on your system.
- **Boot from local drive:** Select to boot the system from an installed hard disk.
- **Return to main menu:** Select to return to the main installation menu.

Boot Options

- Press Esc at the boot menu to display the `boot:` prompt.
- Use the following syntax to provide boot options:

```
boot: linux option1 option2 option3
```

- Some examples of boot settings include:
  – Language
  – Display resolution
  – Interface type
  – Installation method
  – Network settings

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

```
boot: linux option1 option2 option3
```

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

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

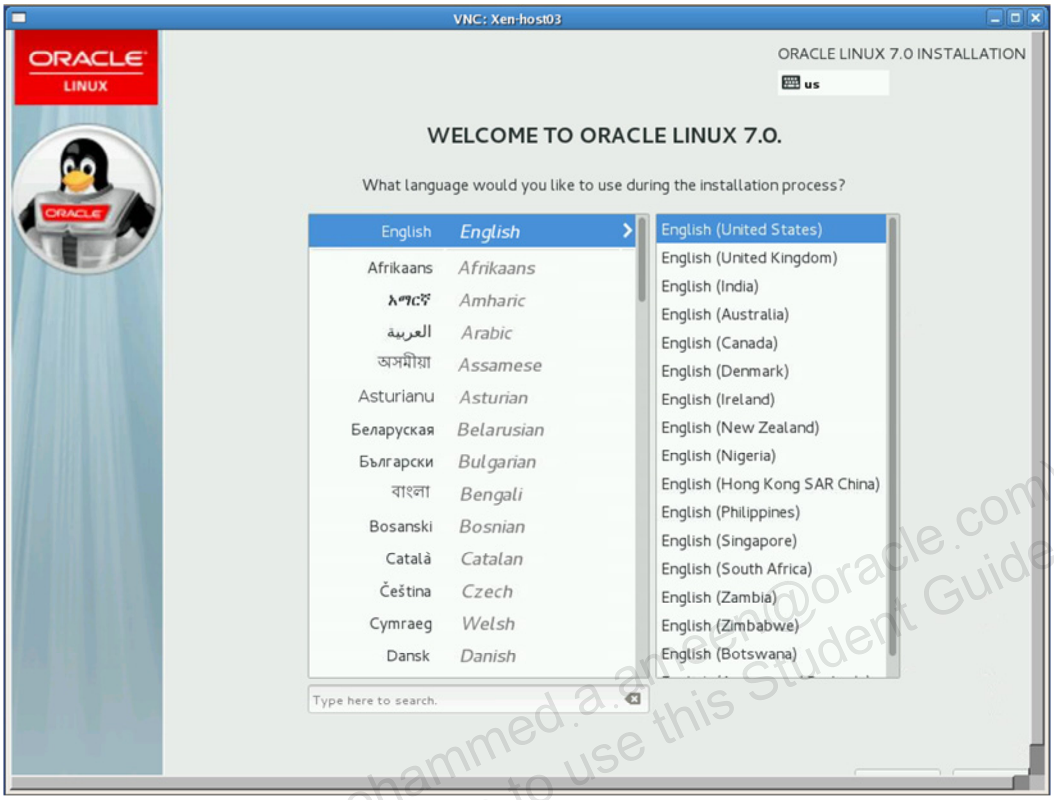
For example, to specify language and keyboard layout, enter:

```
boot: linux lang=value keymap=value
```

To run the installation in text mode, enter:

```
boot: linux text
```

Welcome to Oracle Linux 7

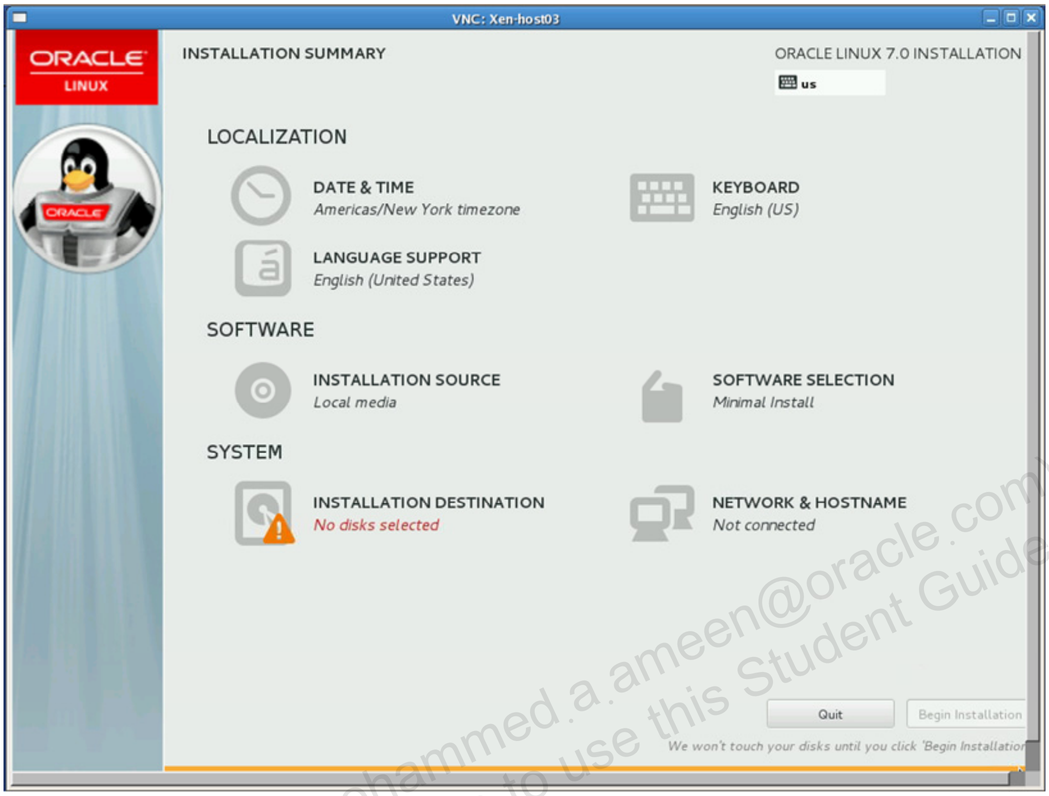


The Welcome to Oracle Linux 7 window appears. In the left-hand box, select the language to use for the installation. The language becomes the default language for the operating system. The language is also used to target the time zone configuration later during the installation process.

In the right-hand box, select the locale.

Scroll down if necessary and click Continue to display the INSTALLATION SUMMARY screen. You can also choose to Quit the installation.

Installation Summary



The Installation Summary window appears. The interface is a hub-and-spoke model rather than the linear model used in previous releases of Oracle Linux. You can select any option in any order.

Only those options with the warning icon must be configured. A message associated with the warning appears at the bottom of the screen. Only the SYSTEM option has a warning in this example, which indicates you must select the installation destination disks.

After all installation options are configured, click Begin Installation.

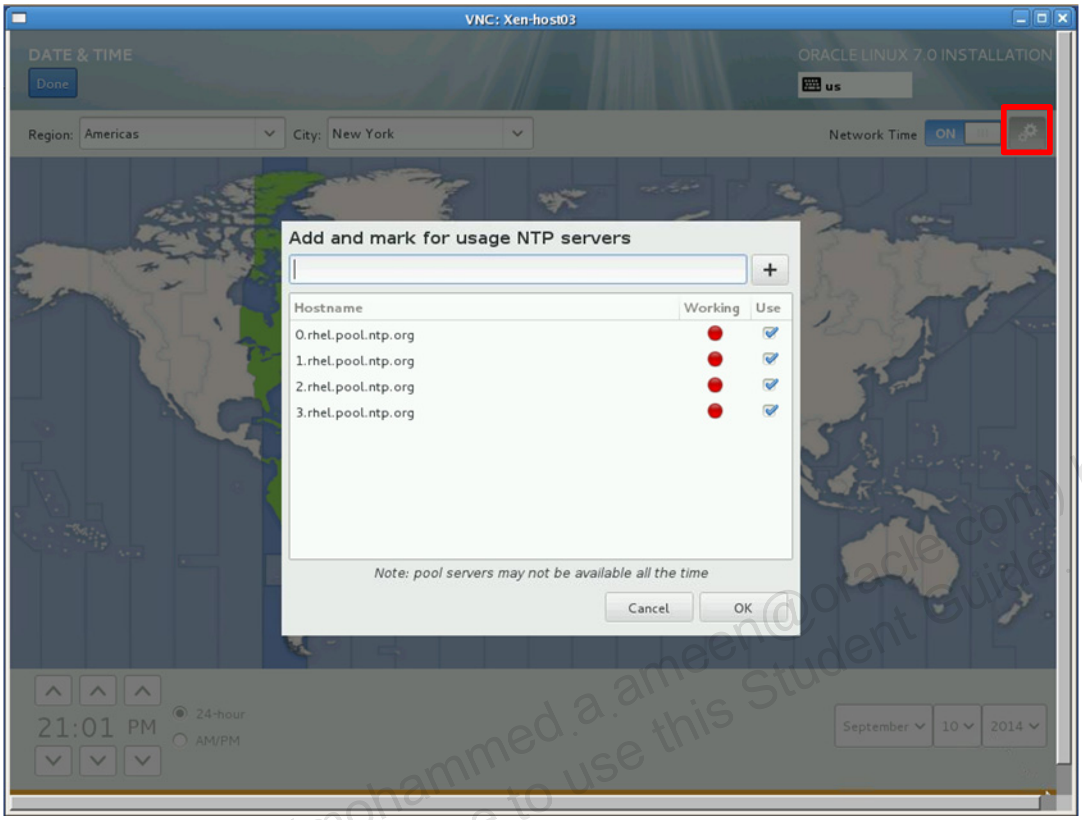
Date & Time Configuration

Selecting DATE & TIME from the INSTALLATION SUMMARY screen displays the screen shown in the slide. Select the appropriate time zone by selecting the country and city closest to the location of your system. To select the time zone, either click the map or select from the Region and City drop-down menus. You can also select Etc at the bottom of the Region menu and then select your time zone in the City menu adjusted to Greenwich Mean Time (GMT).

At the bottom of the screen you can manually adjust the date and time as needed.

After you have made your selection, click Done to return to the INSTALLATION SUMMARY screen.

NTP Configuration

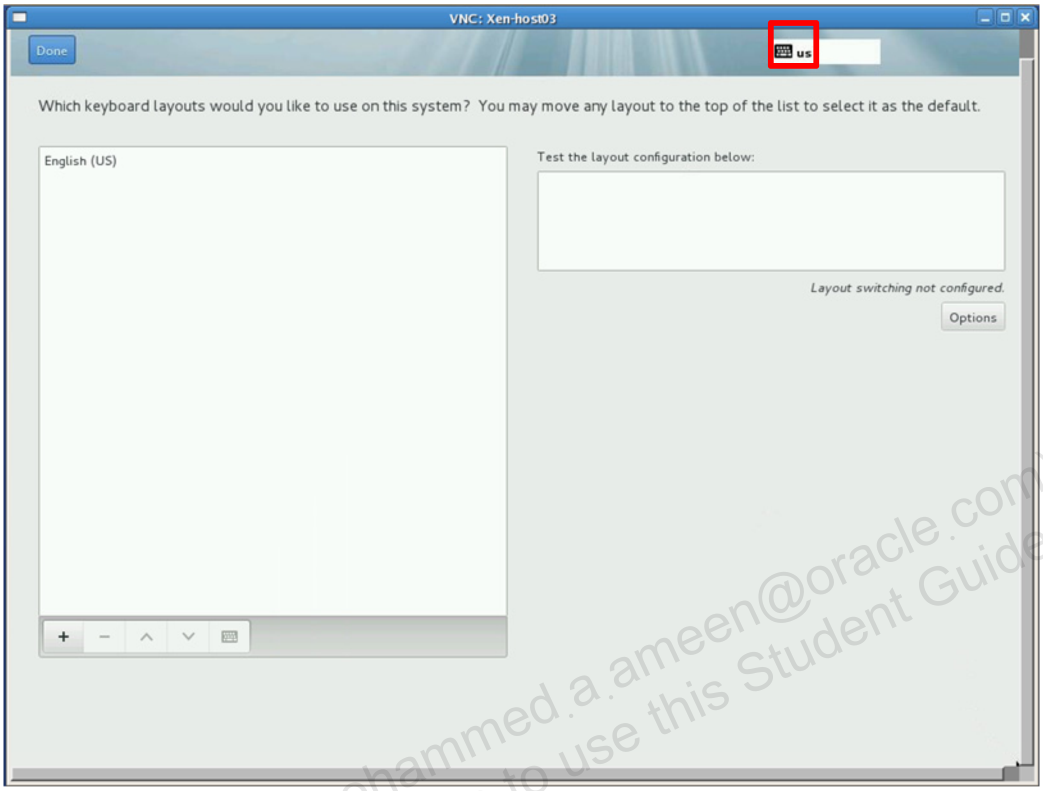


This slide shows the screen to configure Network Time Protocol (NTP). You must be connected to a network before you can enable NTP.

Move the Network Time switch to the ON position and click the configuration icon (outlined in red in the slide) to set the date and time by using NTP. You can then select which NTP servers to use.

After you have made your selection, click Done to return to the INSTALLATION SUMMARY screen.

Keyboard Layout

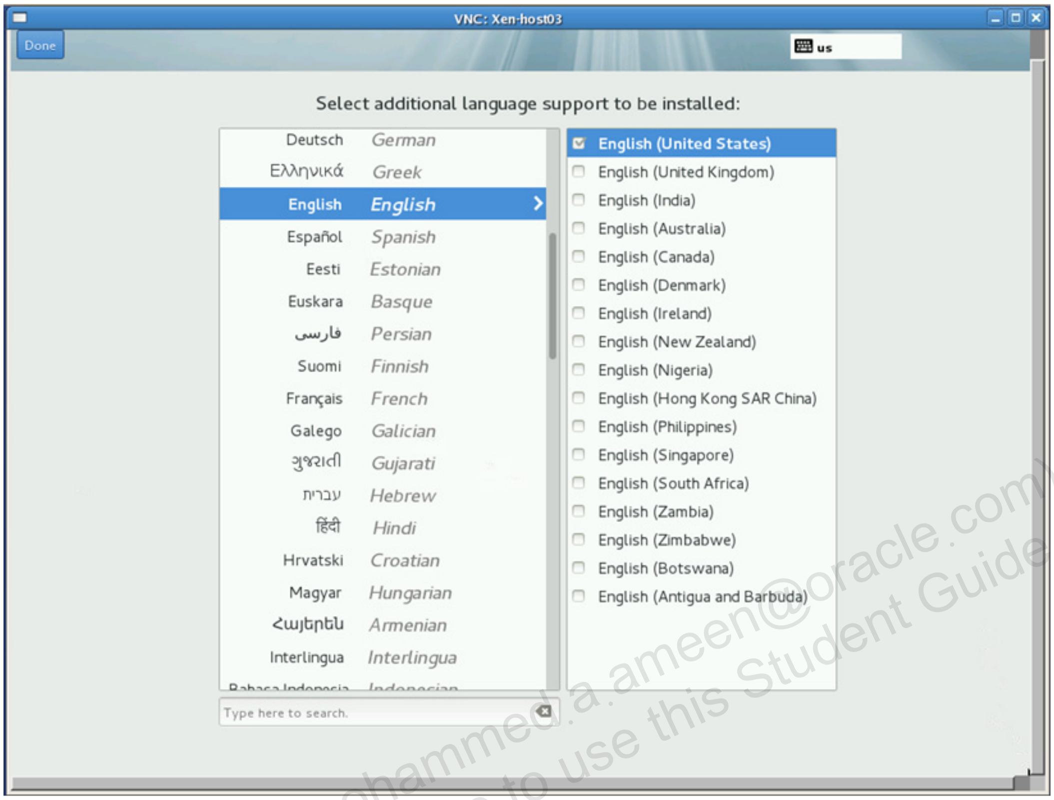


Selecting KEYBOARD from the INSTALLATION SUMMARY screen displays the screen shown in the slide. The language you selected in the welcome screen is listed as the keyboard layout in the left pane. To add support for any additional keyboards for your system, click the + button at the bottom of the screen. To delete a keyboard layout, select the layout in the left pane and click the - button at the bottom of the screen.

Click the keyboard icon at the top of the screen (outlined in red in the slide) to change the current keyboard. Type some text in the text box on the right to test a layout and to confirm that your selection functions correctly.

Click the Options button to configure layout switching options. Click Done to return to the INSTALLATION SUMMARY screen.

Language Support

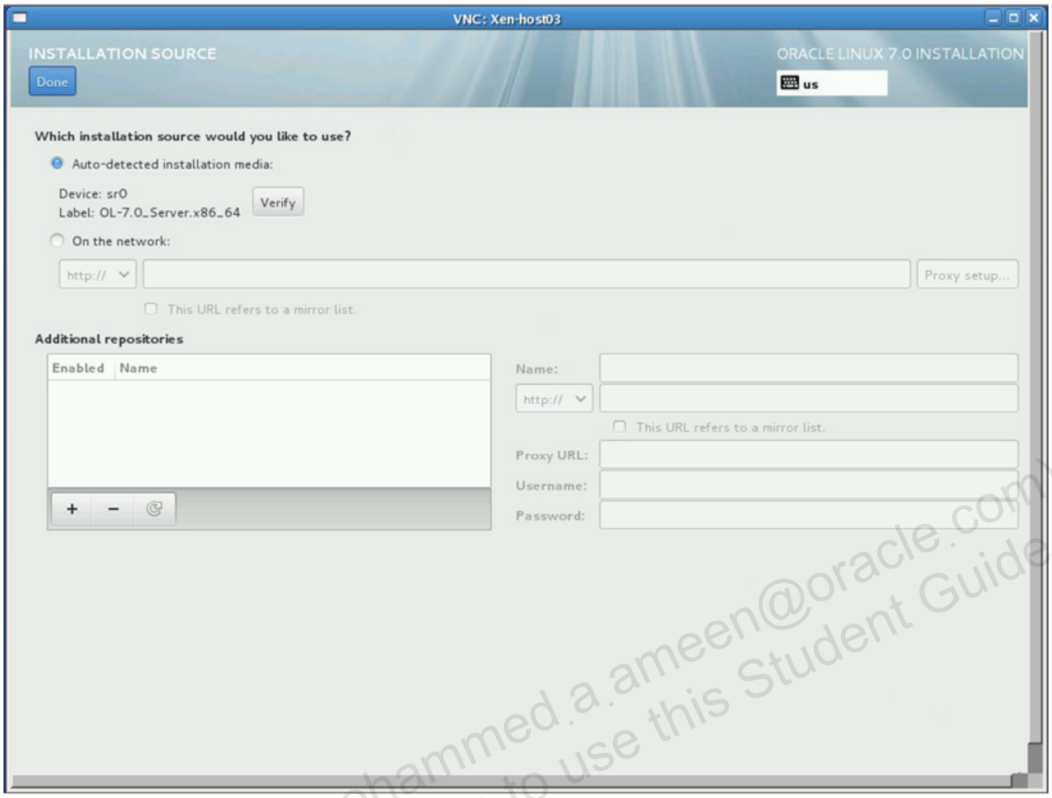


ORACLE

Selecting LANGUAGE SUPPORT from the INSTALLATION SUMMARY screen displays the screen shown in the slide. Select this screen to install support for additional locales and language dialects. Select a language in the left-hand box, or type in the search box, and then select one or more locales in the right-hand box.

Click Done to return to the INSTALLATION SUMMARY screen.

Software Installation Source



This slide displays the INSTALLATION SOURCE screen. On this screen, you can specify locally available installation media or a network location.

Auto-detected installation media

This option is available if the installation program detected a local device, such as a DVD or USB drive, that contains the full installation image. Click Verify to check the media.

ISO file

This option (not shown on the slide) is available if the installation program detected a local hard drive with a mountable file system. Click the Choose an ISO button and then browse to the installation ISO file's location to select an ISO image. Click Verify to check the media.

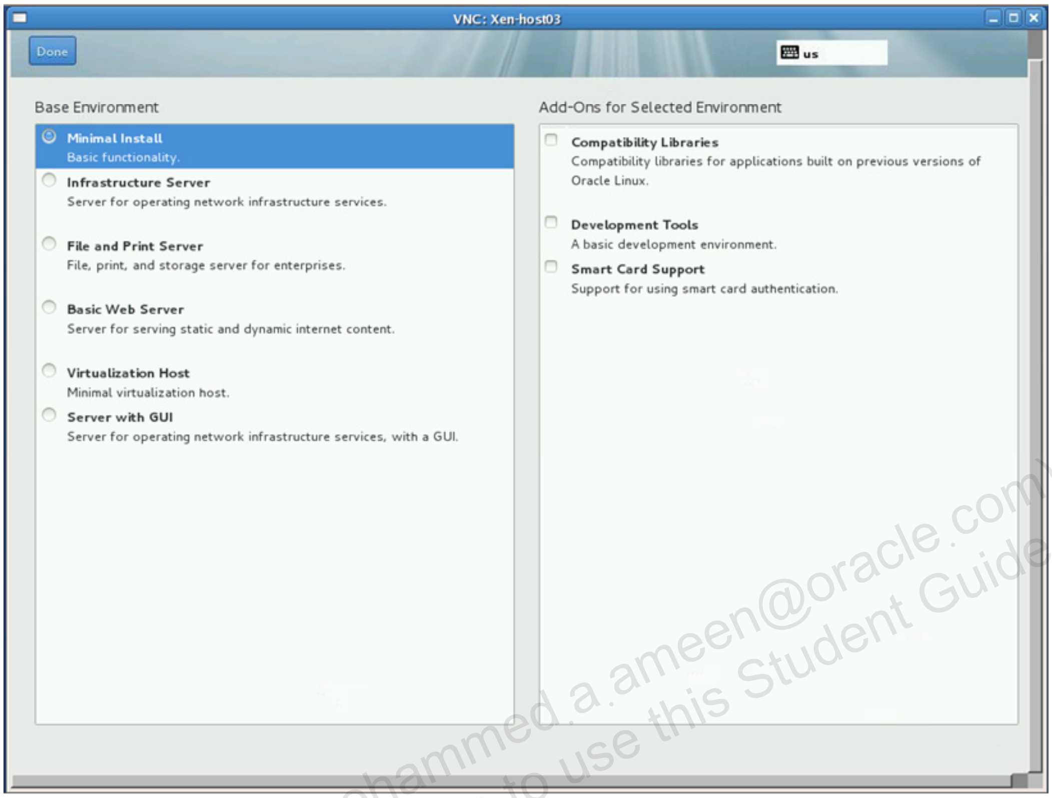
On the network

Select this option to specify a network location as the installation source. Select the network protocol (`http`, `https`, `ftp`, or `nfs`) required to access a network installation server from the drop-down list. For `http` or `https`, enter the URL of the installation image and optionally click Proxy setup to configure a proxy server. For `nfs` and `ftp`, enter the domain name or IP address of the server and the path to the directory that contains the installation image.

Additional repositories

Click the + button to specify additional repositories containing more installation environments and software add-ons. Click Done to return to the INSTALLATION SUMMARY screen.

Selecting the Software to Install



Selecting SOFTWARE SELECTION from the INSTALLATION SUMMARY screen displays the screen shown in the slide. From this screen you can specify which software packages you want to install.

Base Environment

Select the Base Environment from the left pane. The Base Environments are pre-defined sets of packages with a specific purpose (for example, Server with GUI contains the GNOME desktop software packages). The Server with GUI base environment is the only environment that displays a graphical desktop when the system boots. All other base environments boot into a command-line environment. By default, the Server with GUI base environment installs the GNOME 3 desktop.

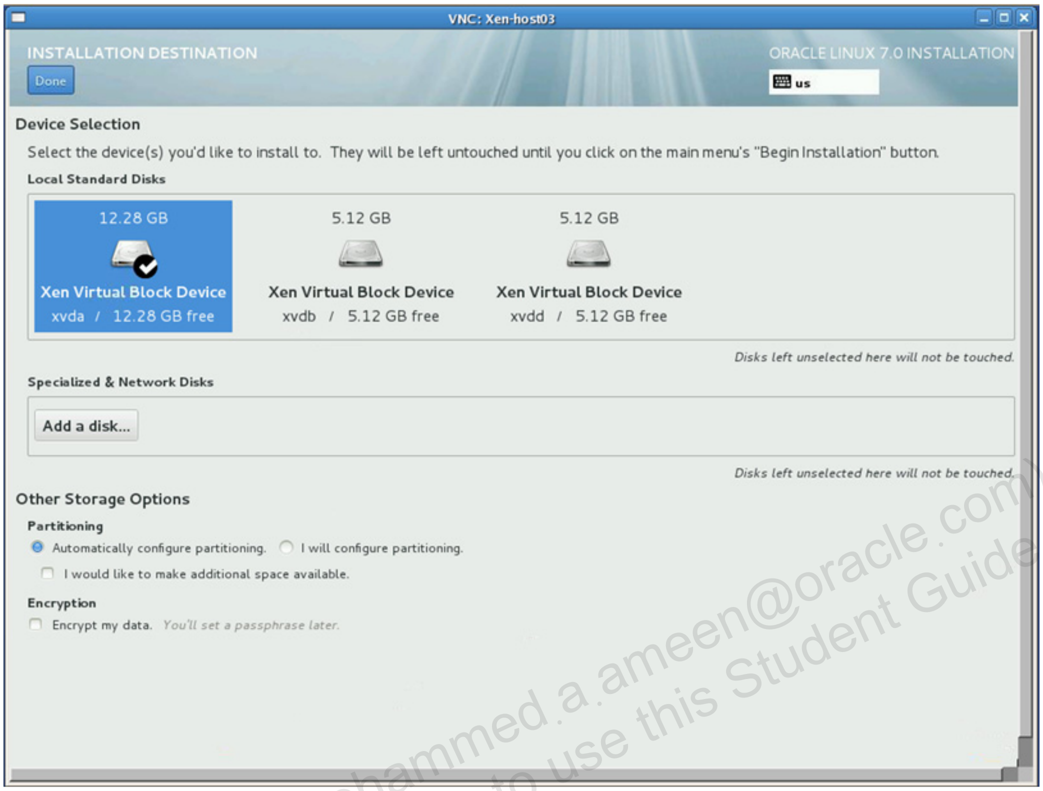
The Minimal Install base environment is the default environment. It contains only the minimum set of packages required to run Oracle Linux.

Add-Ons for Selected Environment

For each Base Environment, you can select from a list of specific Add-Ons in the right pane. For some environments, the add-on pane contains a horizontal line, which separates the list of add-ons. Add-ons listed above the line are specific to the Base Environment you selected. Add-ons below the line are available for all environments.

Click Done to return to the INSTALLATION SUMMARY screen.

Installation Destination



Selecting INSTALLATION DESTINATION from the INSTALLATION SUMMARY screen displays the screen shown in the slide. From this screen you can specify where to install the software and also configure the storage.

Local Standard Disk

This section displays the storage devices directly connected to your computer. Each disk is marked with its device name, size, and available space. Click the disk(s) on which you want to install Oracle Linux. Selected disks are marked with a tick icon as shown in the slide.

Specialized & Network Disks

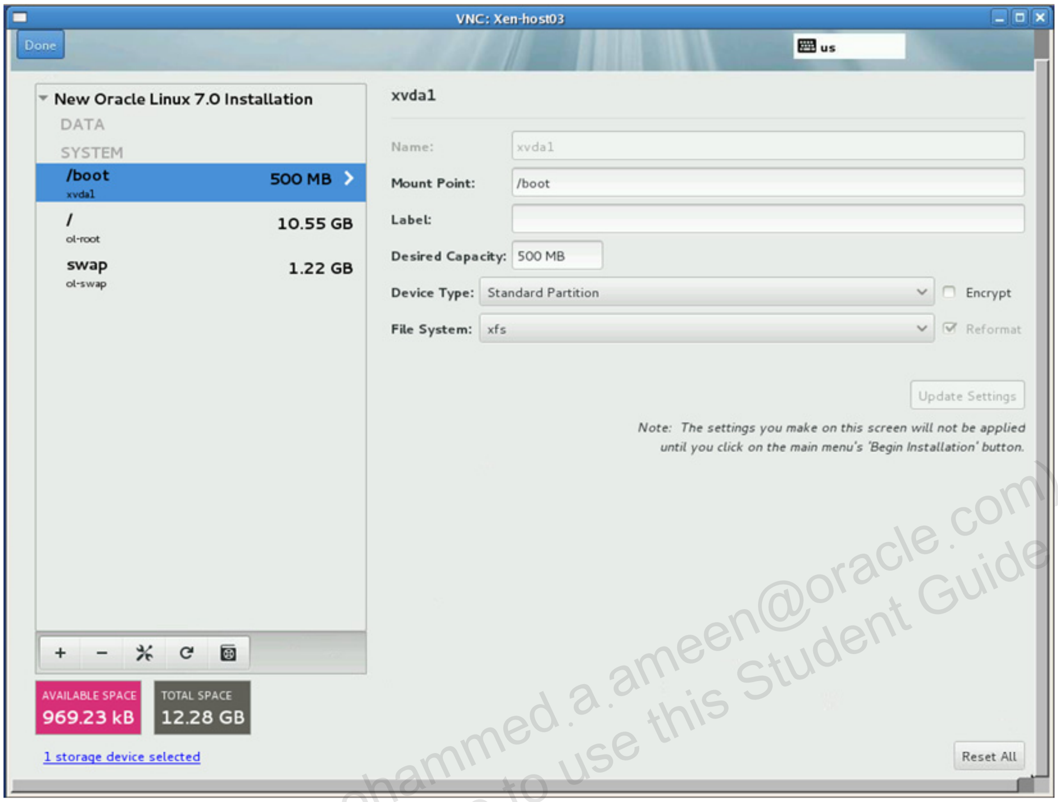
In this section, you can select hardware RAID, iSCSI, or Fibre Channel over Ethernet (FCoE) storage devices. Click Add Disk to display the options for adding and configuring these device types.

Other Storage Options

In this section, select how you want to partition the selected disks, either automatically or manually. By default, the Automatically configure partitioning option is selected. In this section, you can also choose to encrypt your data.

At the bottom of this screen is an option to view Full disk summary and bootloader information.

Automatic Partitioning



This slide displays the automatic partition table for a new Oracle Linux 7 installation. Information for the `/boot` partition is displayed. Automatic disk partitioning creates the following layout on the selected disks:

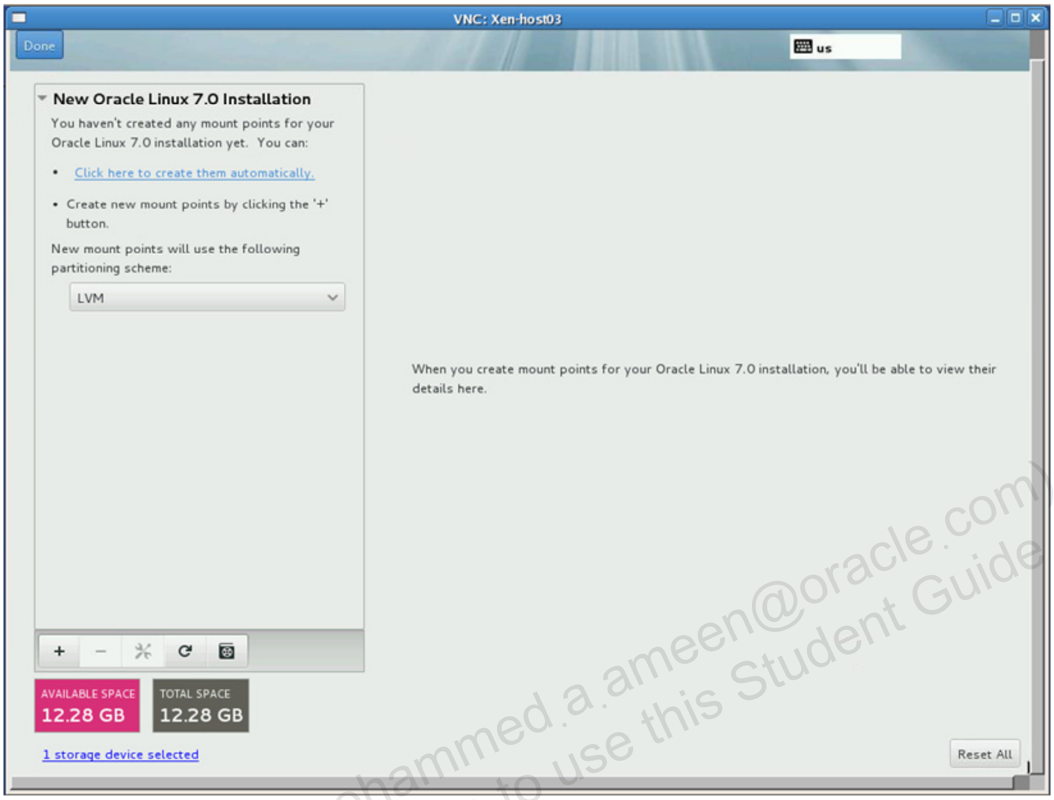
- A 500 MB partition for an XFS file system that contains `/boot`
- A partition in the remaining disk space configured as a Logical Volume Manager (LVM) physical volume for an LVM volume group that contains:
  - A logical volume for an XFS file system that contains the root file system (`/`)
  - A logical volume for an XFS file system that contains the `/home` file system (if the selected disks are larger than 50 GB in total)
  - A logical volume for a swap partition

For the logical volumes, the default volume group name is `ol`. The logical volume names are:

- `ol-home`: For the `/home` file system
- `ol-root`: For the `/` file system
- `ol-swap`: For the swap partition

Menu options exist at the bottom of this screen to customize this default partition table. You can add or remove a mount point, or customize an existing mount point.

Manual Partitioning



Clicking "I will configure partitioning" from the INSTALLATION DESTINATION window displays the screen displayed in this slide. You can choose "Click here to create them automatically." Automatic partitioning is discussed in the previous slide.

For manual partitioning, select the partitioning scheme from the drop-down menu. Options include:

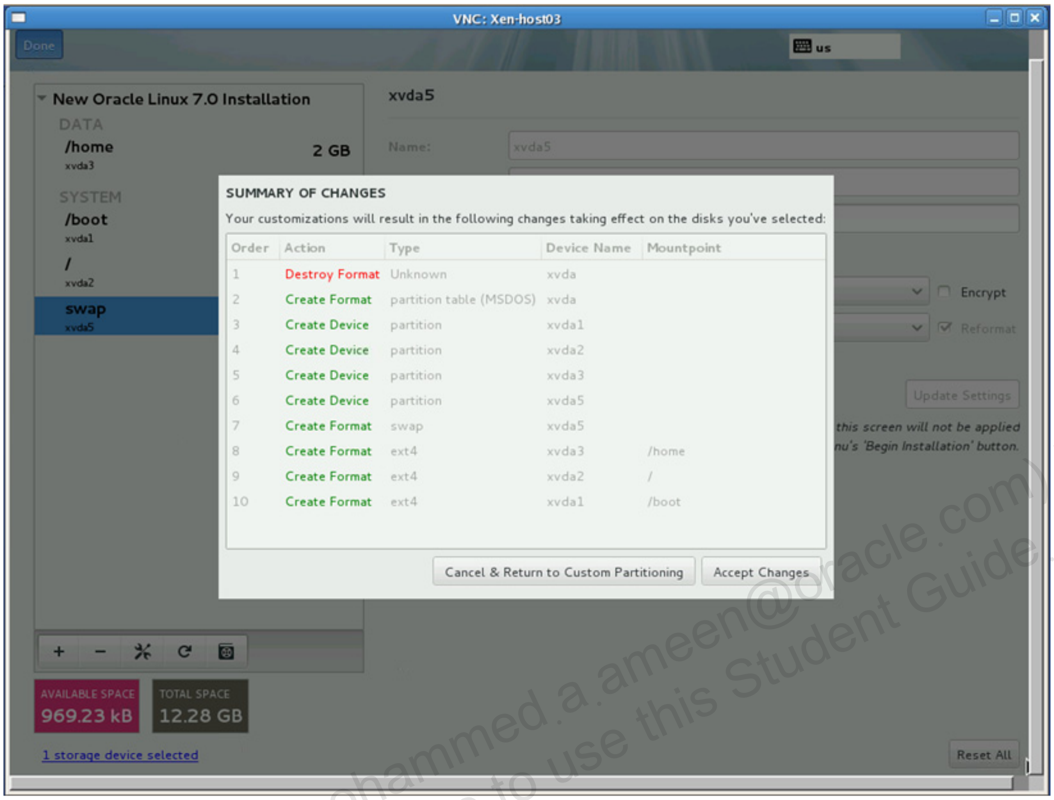
- Standard Partition
- BTRFS
- LVM
- LVM Thin Provisioning

Menu options are available at the bottom of this screen to perform the following actions:

- Add a new mountpoint
- Remove the selected mountpoint(s)
- Configure the selected mountpoint
- Reload storage configuration from disk
- Click for help

When adding a new mountpoint, you are prompted for the Mount Point and the Desired Capacity. You can then label and encrypt the partition and also change the file system type.

Summary of Partitioning

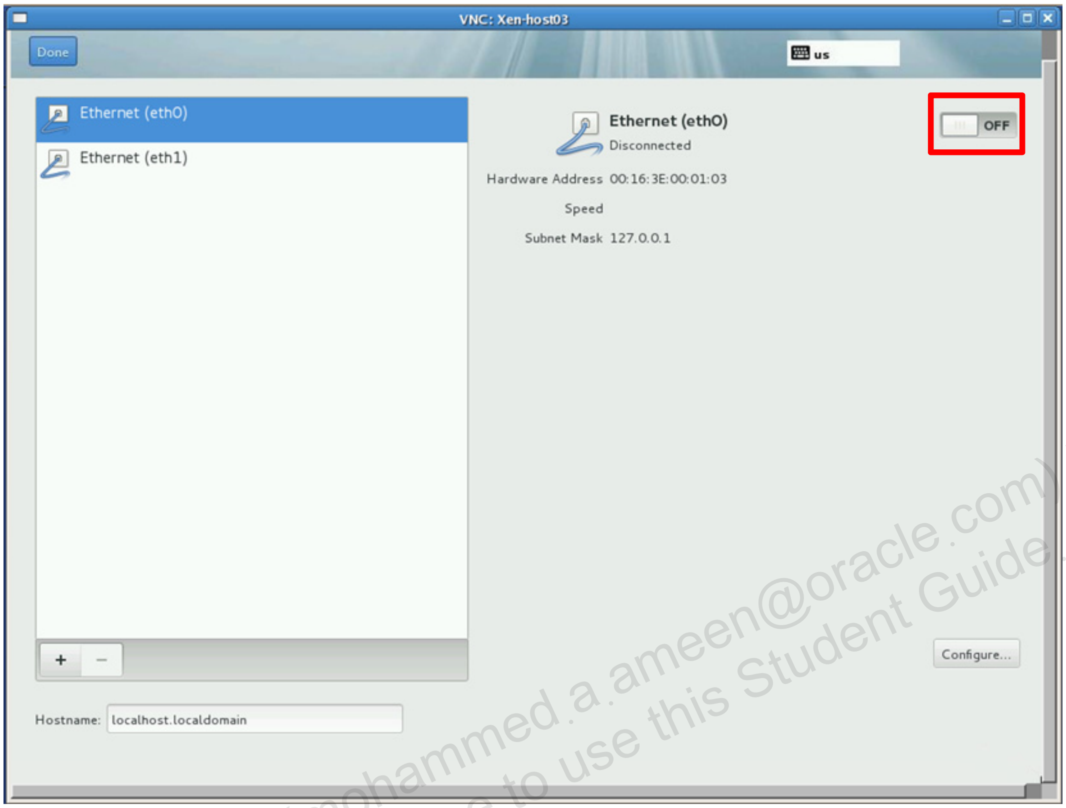


Click Done after partitioning your storage devices. The window in the slide appears summarizing your partitioning customizations. Click Accept Changes to continue or click Cancel & Return to Custom Partitioning to make further changes.

Partitioning and file system creation is discussed in the lesson titled "File Systems." The XFS file system is discussed in the lesson titled "Implementing the XFS File System." The Btrfs file system is discussed in the lesson titled "Implementing the Btrfs File System." Logical Volume Management (LVM) and RAID is discussed in the lesson titled "Storage Administration."

Click Done to return to the INSTALLATION SUMMARY screen.

Network and Hostname Configuration



Selecting NETWORK & HOSTNAME from the INSTALLATION SUMMARY screen displays the screen shown in the slide. From this screen you can configure networking features for your system. At the bottom of this window, you can also set the hostname for your system.

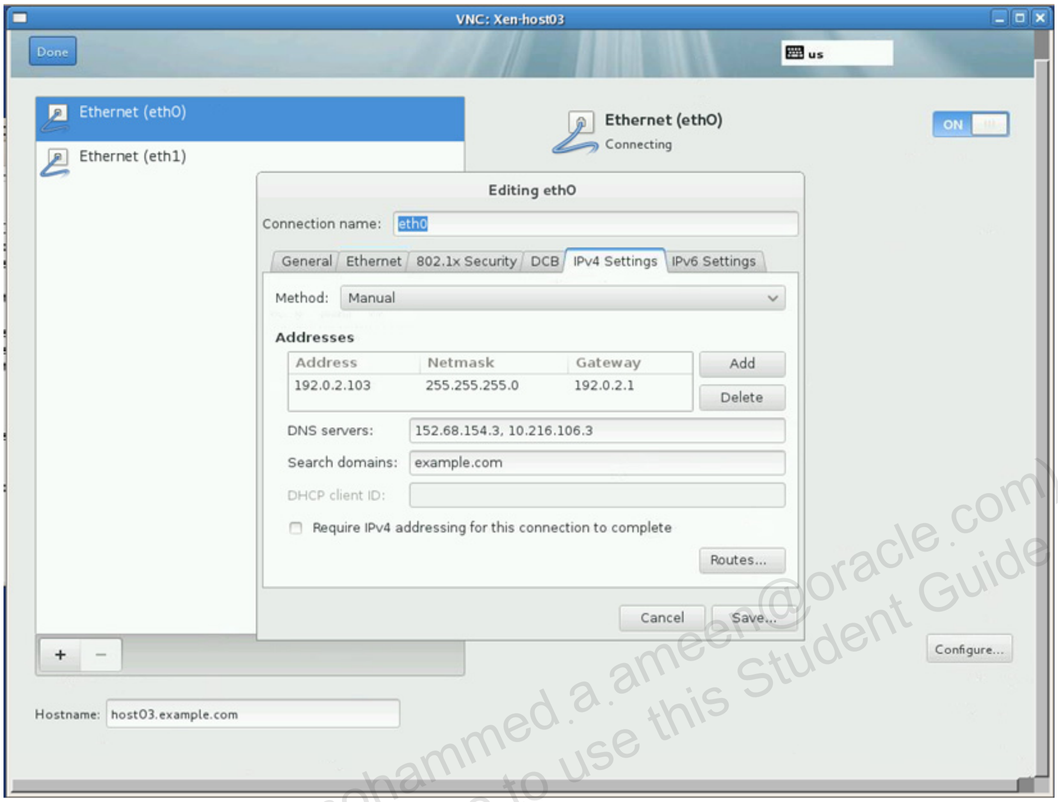
You need to configure a network only if network access is required when your system boots for the first time. Otherwise, you can configure the network after the installation is complete.

Locally accessible network interfaces are detected automatically and are listed in the left-hand pane of the window. These network interfaces cannot be manually deleted. Details about the selected interface are displayed on the right-hand pane.

This example displays a system with two wired Ethernet connections, `eth0` and `eth1`. You can also add special network devices such as an interface bond or team, or a VLAN (virtual local area network) device. Click the + button to access the settings for configuring these devices.

To enable a selected network interface, move the switch screen (outlined in red in the slide) to the ON position. By default, the IP settings for a network interface are configured automatically by using DHCP for IPv4, and the Automatic method for IPv6. To edit the connection, click the Configure button to display the connection settings window as shown in the next slide.

Network Connection Settings

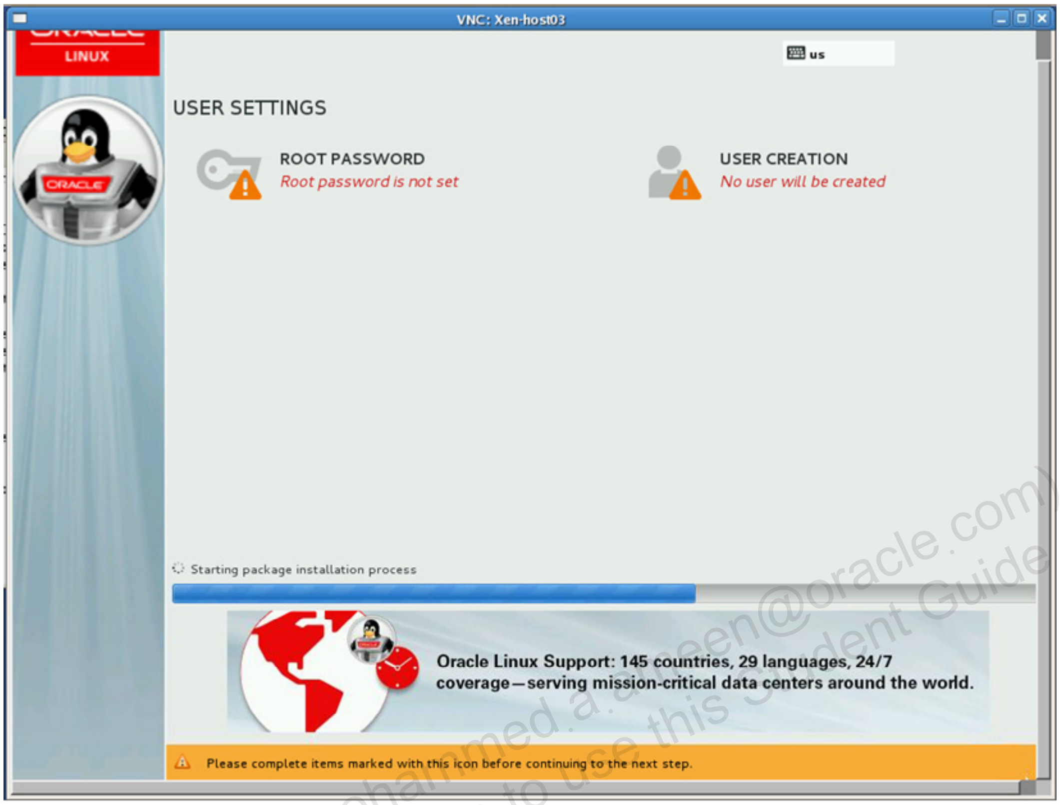


Selecting Configure from the NETWORK & HOSTNAME screen displays the screen shown in the slide. The configuration options presented depends on whether the connection is wired, wireless, mobile broadband, VPN, or DSL. The lesson titled “Network Configuration” provides details of the configuration options.

This screen shows manual IPv4 configuration for a wired interface. The IP address, netmask, and gateway are provided. In addition, Domain Name Service (DNS) parameters are configured in this screen.

Click Save and then click Done to return to the INSTALLATION SUMMARY screen.

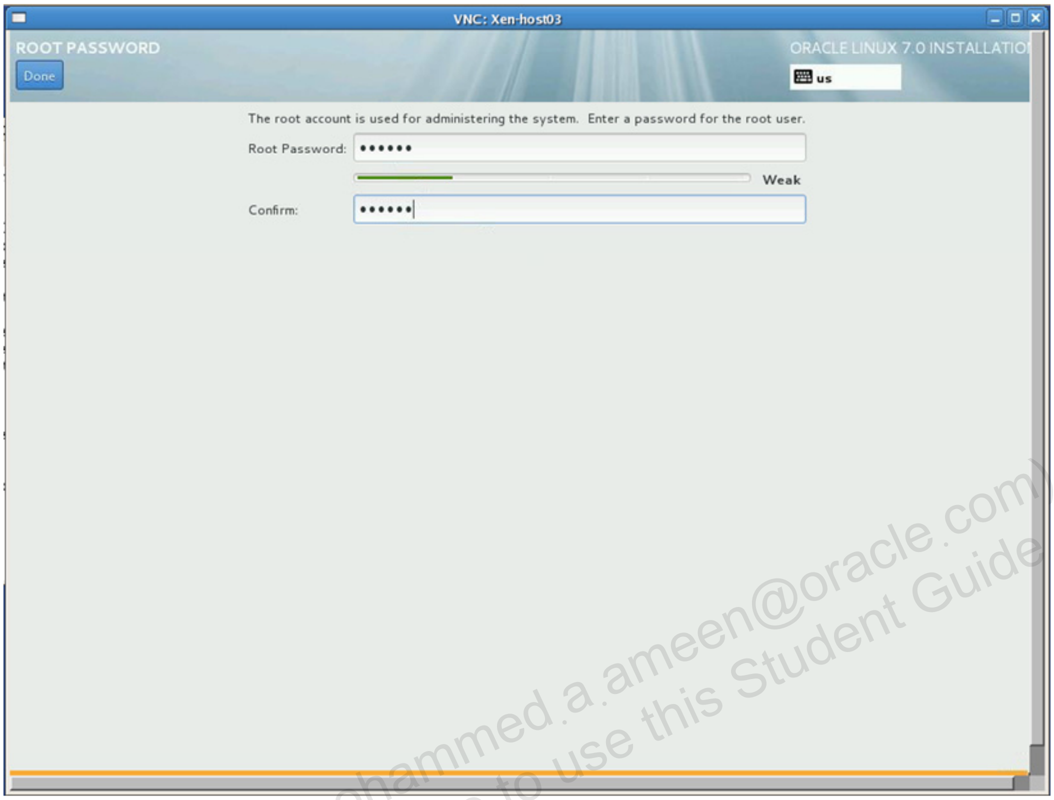
Completing the Installation



Click Begin Installation from the INSTALLATION SUMMARY to display the screen shown in the slide. From this screen you can configure the `root` password and also add an initial non-root user. A status bar also shows the progress of the installation.

You can use the menu options while the installation is in progress. You must set the `root` password to complete the installation. Creating a user is optional.

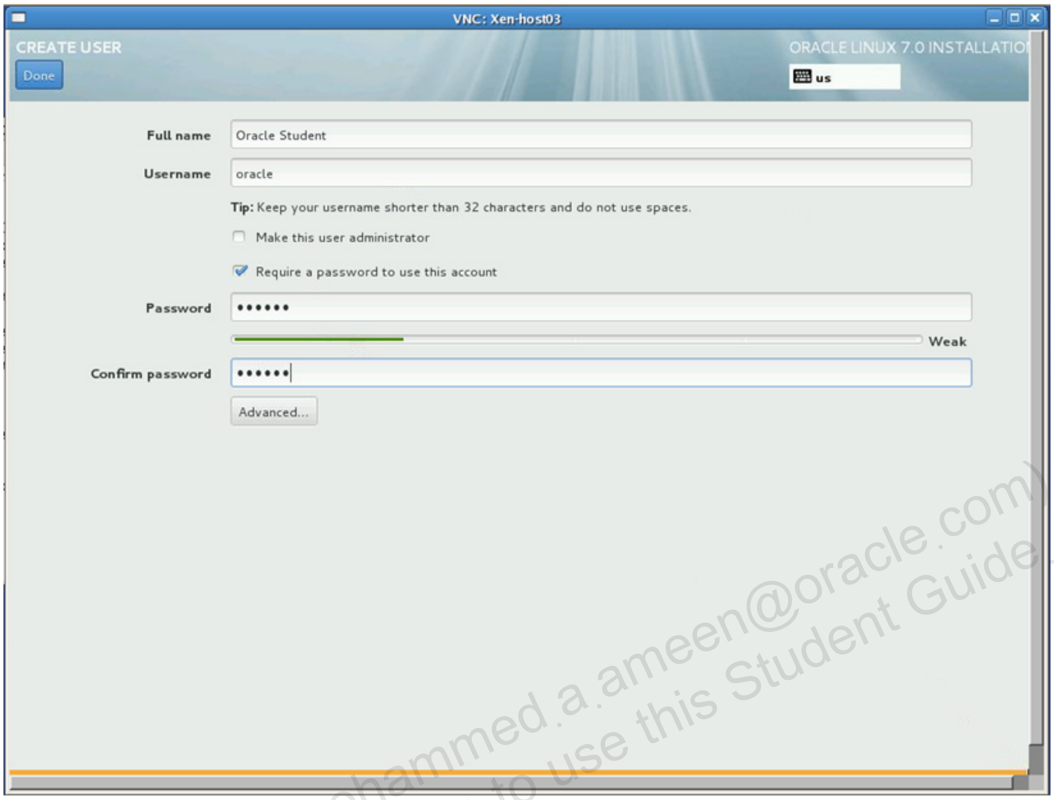
Setting the `root` Password



Clicking ROOT PASSWORD from the CONFIGURATION displays the screen shown in the slide. On this screen, enter your new password into the Root Password field. Your entry displays the characters as asterisks for security. There is a visual indication about the strength of the password. Enter the same password into the Confirm field to ensure it is set correctly.

After you set the `root` password, click Done. If you set a weak password, you have to click Done twice.

Creating an Initial User



Clicking USER CREATION from the CONFIGURATION displays the screen shown in the slide. Enter the Full name of the local user account you want to create. A Username is derived automatically from the Full name but you can change this.

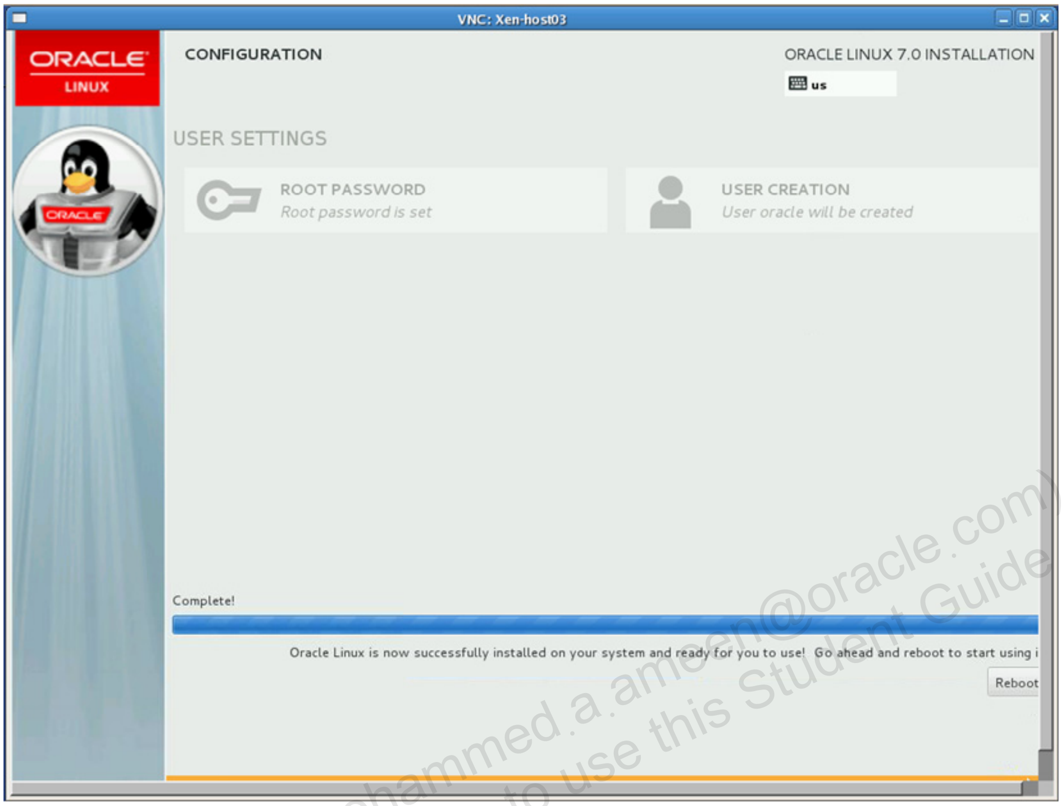
Enter the Password twice, again there is a visual indication about the strength of the password.

Select Make this user administrator if you want the user to be able to administer the system. Selecting this option adds the user to the `wheel` user group, which enables them to run administrative tools by using the `sudo` command.

Click Advanced to specify the user's home directory, user ID, or group ID. Creating users is discussed in the lesson titled “User and Group Administration.”

Click Done to return to the CONFIGURATION screen. Click Done again if you have set a weak password.

Installation Complete



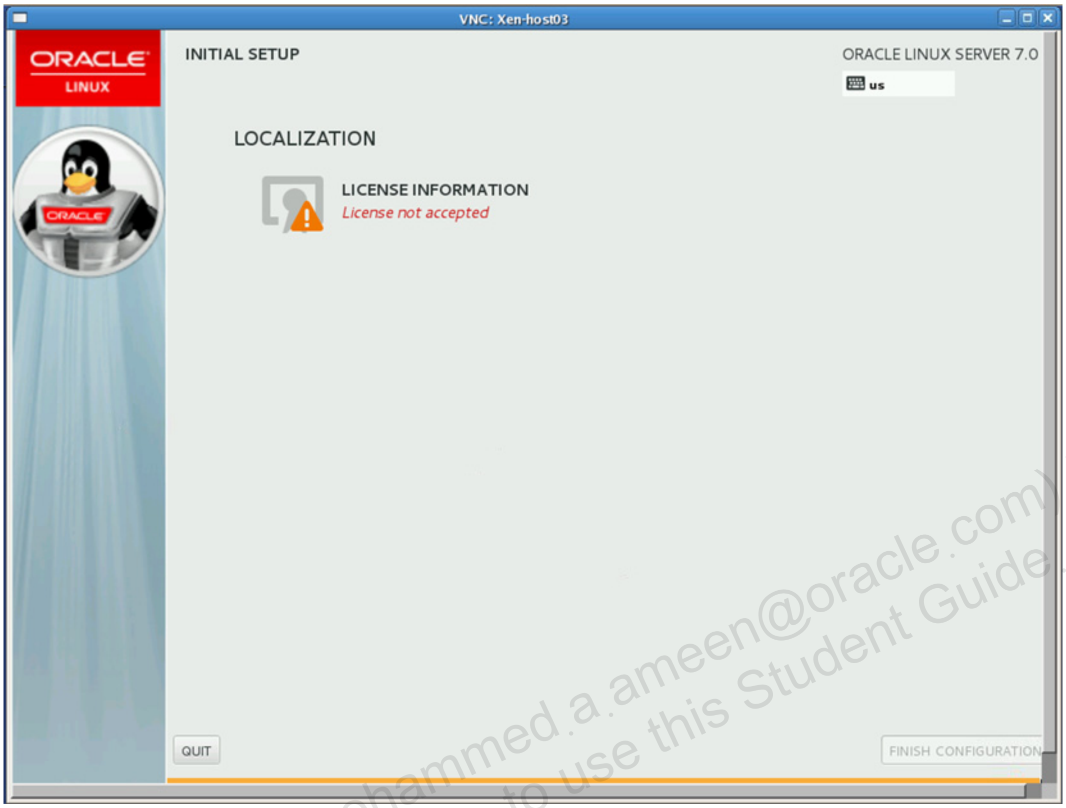
After you have completed the user configuration, click Finish Configuration. The installation completes and you are prompted to Reboot. Remove any installation media if it is not ejected automatically upon reboot.

Log files created during your installation can be found in the `/var/log/anaconda/` directory after you reboot your system.

If you installed a Base Environment other than Server with GUI base environment, the system reboots as a non-graphical system and displays a `login:` prompt.

If you installed the Server with GUI environment, the system reboots and displays the Initial Setup screen as shown in the next slide.

Initial Setup and Firstboot



If you install the Server with GUI environment, the system reboots and the Initial Setup screen is displayed as shown in the slide. A USER CREATION option appears on this window if you did not already create an initial non-root user.

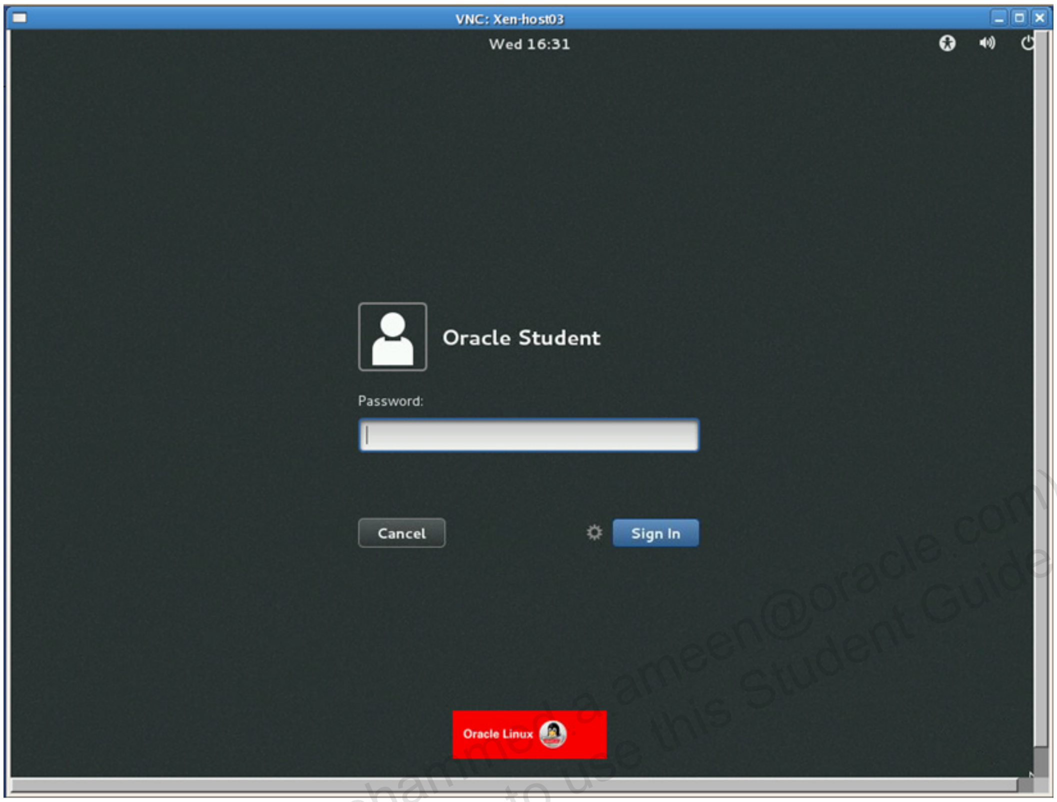
Select the License Information option and accept the license agreement. Click Done after reading and accepting the license agreement.

Click FINISH CONFIGURATION on this INITIAL SETUP window. Firstboot launches next, prompting you for the final configuration items:

- **Kdump:** Kernel crash dumping mechanism
- **Set Up Software Updates:** Connect to Unbreakable Linux Network (ULN) for software updates.

If you connect to ULN, you are prompted for ULN login and also to create a ULN profile. Connecting to ULN is discussed in the lesson titled “Package Management.” Kdump is covered in the advanced administration course.

GUI Login Window



You are presented with a GUI login window if you have installed the Server with GUI base environment. Click the user you created during the installation and you are prompted for the user password as shown in the slide. Enter the password and click Sign In.

The first time you log in, you are taken through the Gnome-initial-setup. The following screens appear:

- **Language selection:** Defaults to language selected during the installation
- **Input sources:** Defaults to the keyboard type selected during the installation
- **Online accounts:** Connect to your existing data in the cloud

The GNOME help window also opens the first time you log in. Close the help window to display the GNOME desktop. The Gnome-initial-setup and help windows open only the first time you log in.

Quiz

After installing the Minimal Install base environment, the Firstboot tool runs to perform initial configuration tasks.

a. True

b. False

Summary

In this lesson, you should have learned how to:

- Obtain Oracle Linux 7 operating system software
- Describe the Anaconda installer
- Install Oracle Linux 7
- Describe the Firstboot utility

Practice 3: Overview

The practices cover the following topics:
- Installing Oracle Linux 7
- Using Firstboot
- Logging in to Oracle Linux 7 and shutting down
- Re-creating the host03 VM guest

4

Oracle Linux 7 Boot Process

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe the Oracle Linux 7 boot process
- Describe and configure the GRUB 2 bootloader
- Describe and configure kernel boot parameters
- Describe the `systemd` system and service manager
- Describe `systemd` service units
- Configure services
- Describe `systemd` target units
- Configure Rescue Mode and Emergency Mode
- Perform shutdown, suspend, and reboot operations

Oracle Linux 7 Boot Process

1. The computer's BIOS performs POST.
2. BIOS reads the MBR for the bootloader.
3. GRUB 2 bootloader loads the `vmlinuz` kernel image.
4. GRUB 2 extracts the contents of the `initramfs` image.
5. The kernel loads driver modules from `initramfs`.
6. Kernel starts the system's first process, `systemd`.
7. The `systemd` process takes over. It:
   A. Reads configuration files from the `/etc/systemd` directory
   B. Reads file linked by `/etc/systemd/system/default.target`
   C. Brings the system to the state defined by the system target
   D. Executes `/etc/rc.local`

ORACLE

It is important to understanding the Linux boot process to troubleshoot boot problems. These are the high-level steps in the boot process. You need to be aware of files involved in the boot process because errors in these files can cause boot problems.

The default bootloader program used on Oracle Linux 7 is GRUB 2. GRUB stands for GRand Unified Bootloader. GRUB 2 replaces the older GRUB bootloader, which was used by previous versions of Oracle Linux. The older GRUB bootloader is now referred to as "GRUB Legacy."

The bootloader loads the `vmlinuz` kernel image file into memory and extracts the contents of the `initramfs` image file into a temporary, memory-based file system (`tmpfs`). The initial RAM disk (`initramfs`) is an initial root file system that is mounted before the real root file system.

After the newly loaded kernel gets far enough in its initialization sequence (disks probed, memory mapped, and so on), it then switches over to using the real root file system as specified by the `root` directive in the GRUB 2 configuration. This contains, among other things, the `/etc/fstab` file identifying the rest of the file systems to be mounted.

The kernel starts the `systemd` process with a process ID of 1 (PID 1). `systemd` is the ancestor of all processes on a system. `systemd` reads its configuration from files in the `/etc/systemd` directory. The `/etc/systemd/system.conf` file controls how `systemd` handles system initialization.

`systemd` reads the file linked by `/etc/systemd/system/default.target` (for example, `/usr/lib/systemd/system/multi-user.target`) to determine the default system target. The system target file defines the services that `systemd` starts.

`systemd` brings the system to the state defined by the system target, performing system initialization tasks such as:

- Setting the host name
- Initializing the network
- Initializing SELinux based on its configuration
- Printing a welcome banner
- Initializing the system hardware based on kernel boot arguments
- Mounting the file systems, including virtual file systems such as the `/proc` file system
- Cleaning up directories in `/var`
- Starting swapping

If you have made `/etc/rc.local` executable and have copied `/usr/lib/systemd/system/rc-local.service` to `/etc/systemd/system`, `systemd` runs any actions that you have defined in `/etc/rc.local`. However, the preferred way of running such local actions is to define your own `systemd` unit.

For information on `systemd` and on how to write `systemd` units, see the `systemd(1)`, `systemd-system.conf(5)`, and `systemd.unit(5)` manual pages.

The Initial RAM File System

- The kernel mounts `initramfs` as part of a two-stage boot process.
  – The initial RAM disk image preloads the block device modules so the root file system can be mounted.
- The `initramfs` is bound to the Linux kernel executable:

```
# ls /boot/*3.8.13-35.3.1*
initramfs-3.8.13-35.3.1.el7uek.x86_64.img
vmlinuz-3.8.13-35.3.1.el7uek.x86_64
```

- The `dracut` utility creates `initramfs` whenever a new kernel is installed.
- Use the `lsinitrd` command to view the contents of the image created by `dracut`:

```
# lsinitrd | less
```

ORACLE

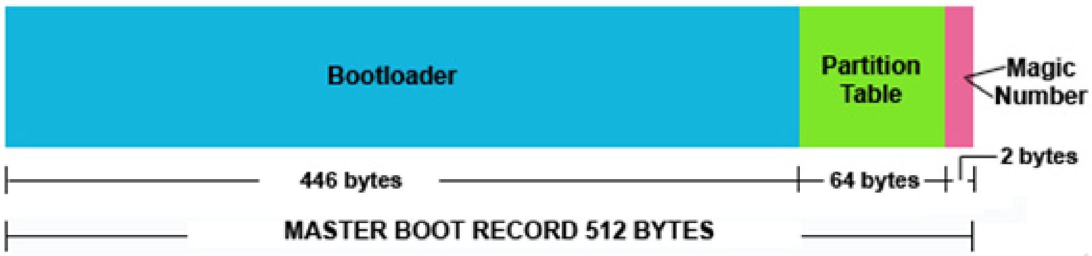
The job of the initial RAM file system is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted. The `initramfs` is bound to the kernel and the kernel mounts this `initramfs` as part of a two-stage boot process. For example, if the kernel version is `3.8.13-35.3.1`, there is a `vmlinuz` file and an `initramfs` file containing the same version number:

```
# ls /boot/*3.8.13-35.3.1*
initramfs-3.8.13-35.3.1.el7uek.x86_64.img
vmlinuz-3.8.13-35.3.1.el7uek.x86_64
```

Whenever a new kernel is installed, the `dracut` utility is always called by the installation scripts to create an `initramfs`. Refer to the `dracut(8)` man page for information on using `dracut` to generate an `initramfs` image, inspecting the content of the image created by `dracut`, adding kernel modules and boot parameters, specifying the root device and keyboard settings, blacklisting kernel modules, speeding up the boot process, injecting custom files to the `initramfs` image, creating a network-aware `initfamfs` image (if your root partition is on a network drive), reducing the image size, and troubleshooting the boot process. You can use the `lsinitrd` command to view the contents of the image created by `dracut`:

```
# lsinitrd | less
```

Master Boot Record (MBR)



On a computer with x86 architecture, the Master Boot Record (MBR) is the first 512 bytes of the boot drive that is read into memory by the BIOS. The first 446 bytes of that 512 bytes contain low-level boot code. For some bootloaders, the code in the MBR points to a bootloader somewhere else on the disk or on another disk. With GRUB 2 however, the code all fits in the MBR. The next 64 bytes contain the partition table for the disk. The last two bytes are the "Magic Number," which is used for error detection.

Bootloader

The bootloader software runs when a computer starts. It is responsible for loading and transferring control to the kernel. The kernel then initializes the rest of the operating system. Many bootloaders are available. The most common bootloaders for Linux are LILO (LInux LOader) and GRUB (GRand Unified Bootloader). The default bootloader used on Oracle Linux 7 is GRUB 2.

The GRUB 2 manual is available at http://www.gnu.org/software/grub/manual/grub.html.

GRUB 2 Bootloader

- Oracle Linux 7 uses the GRUB 2 bootloader.
- The GRUB 2 configuration file is `/boot/grub2/grub.cfg`.
  - Do not edit this file directly.
- Use the `grub2-mkconfig` command to generate `grub.cfg`.

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- The `grub2-mkconfig` command uses:
  - Template scripts in the `/etc/grub.d/` directory
  - Boot menu-configuration settings in `/etc/default/grub`

GRUB 2 is the bootloader used by Oracle Linux 7. GRUB is a product of the GNU project. GRUB conforms to the multiboot specification, which allows it to load many free operating systems directly as well as to chain-load proprietary operating systems. GRUB understands file systems and kernel executable formats, allowing it to load an arbitrary operating system without recording the physical position of the kernel on the disk. The kernel can load just by specifying its file name and the drive and partition where the kernel resides.

The GRUB 2 configuration file is `/boot/grub2/grub.cfg`. Do not edit this file directly. Use the `grub2-mkconfig` command to generate `grub.cfg`. This command uses the template scripts in `/etc/grub.d` and menu-configuration settings taken from `/etc/default/grub` when generating `grub.cfg`. The `/etc/grub2.cfg` file is a symbolic link to `/boot/grub2/grub.cfg`.

The following lists the contents of the `/etc/grub.d` directory:

```
# ls /etc/grub.d
00_header   20_linux_xen      30_os-prober   41_custome
10_linux    20_ppc_terminfo   40_custom      README
```

The scripts in this directory are read in alphabetical order. Therefore, you can rename the scripts to change the boot order of specific menu entries.

The /etc/default/grub File

- Provides GRUB 2 menu-configuration settings in `/boot/grub2/grub.cfg`
- Examples include:
  – `GRUB_TIMEOUT`: Number of seconds to display the boot menu before the boot process continues
  – `GRUB_DEFAULT`: The default menu entry to boot, starting with zero (`0`). The value of `saved` allows the use of the following commands:

```
# grub2-set-default <menuentry>
# grub2-reboot <menuentry>
```

  – `GRUB_CMDLINE_LINUX`: Kernel boot parameters
- Run `grub2-mkconfig` after making any changes:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```


GRUB 2 menu-configuration settings are taken from `/etc/default/grub` when generating `grub.cfg`. The following lists the contents of the `/etc/default/grub` file:

```
# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto vconsole.font=latarcyrheb-sun16 rhgb quiet
GRUB_DISABLE_RECOVERY="true"
```

If changes are made to any of these parameters, you need to run `grub2-mkconfig` to re-generate the `/boot/grub2/grub.cfg` file. For example:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Values are described as follows:

- GRUB_TIMEOUT: The time in seconds after the menu is displayed to boot the default entry, unless a key is pressed. The default is 5. Set to 0 to boot immediately without displaying the menu, or to -1 to wait indefinitely.
- GRUB_DISTRIBUTOR: Set by distributors of GRUB and is used to generate more informative menu entry titles. The example evaluates to Oracle Linux Server.
- GRUB_DEFAULT: The default menu entry to boot. A value of 0 boots the first menuentry. A value of 1 boots the second menuentry. A value of saved instructs GRUB 2 to load the last successfully loaded operating system.

A GRUB_DEFAULT value of saved also allows you to use the grub2-set-default and grub2-reboot commands to specify the default entry. These two commands are described as follows:

- grub2-set-default: Sets the default entry for all subsequent reboots
- grub2-reboot: Sets the default entry for the next reboot only

For example, with GRUB_DEFAULT=saved, the following command sets the default entry for all subsequent reboots to the second menuentry:

```
# grub2-set-default 1
```

- GRUB_DISABLE_SUBMENU: By default, the grub2-mkconfig command generates a top level menu entry for the kernel with highest version number and puts all other found kernels or alternative menu entries for recovery mode in a submenu. Setting GRUB_DISABLE_SUBMENU=true disables this.
- GRUB_TERMINAL_OUTPUT: The terminal output device. When specifying multiple devices, separate the valid terminal output names with spaces.
- GRUB_CMDLINE_LINUX: Kernel boot parameters. More information on kernel boot parameters is provided in the next slide.
- GRUB_DISABLE_RECOVERY: By default, two menu entries are generated for each Linux kernel: one default entry and one entry for recovery mode. Setting GRUB_DISABLE_RECOVERY="true" disables this.

Kernel Boot Parameters

Kernel boot parameters:

- Modify the behavior of the kernel
- Can be included in the GRUB 2 configuration file (persistent)
- Can be issued from the GRUB 2 command-line interface
- Are exported to `/proc/cmdline`, for example:

```
# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.8.13-35.2.1.el7uek.x86_64
   root=UUID=... ro vconsole.keymap=us crashkernel=auto
   vconsole.font=latarcyrheb-sun16 rhgb quiet
```


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

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

Parameters can be passed to the kernel from the GRUB 2 command-line interface. These settings are not persistent however and apply only for a single boot. To make the settings persistent, define the parameters as values for the `GRUB_CMDLINE_LINUX` parameter in the `/etc/default/grub` file. For example:

```
GRUB_CMDLINE_LINUX="vconsole.keymap=us crashkernel=auto
vconsole.font=latarcyrheb-sun16 rhgb quiet
```

Refer to the Oracle Linux 7 Administration Guide for a description of kernel boot parameters.

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

GRUB 2 Configuration File

```
root@host03:~
File Edit View Search Terminal Help
menuentry 'Oracle Linux Server, with Unbreakable Enterprise Kernel 3.8.13-35.3.1
.el7uek.x86_64' --class oracle --class gnu-linux --class gnu --class os --unrest
ricted $menuentry_id_option 'gnulinux-3.8.13-35.3.1.el7uek.x86_64-advanced-161be
b9e-4942-432f-83ef-23855602e694' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1'  1e0af6c4-
bf07-4f50-bfde-ac2fc3a21807
        else
          search --no-floppy --fs-uuid --set=root 1e0af6c4-bf07-4f50-bfde-ac2fc3
a21807
        fi
        linux16 /vmlinuz-3.8.13-35.3.1.el7uek.x86_64 root=UUID=161beb9e-4942-432
f-83ef-23855602e694 ro vconsole.keymap=us crashkernel=auto  vconsole.font=latarc
yrheb-sun16 rhgb quiet
        initrd16 /initramfs-3.8.13-35.3.1.el7uek.x86_64.img
}
menuentry 'Oracle Linux Server, with Linux 0-rescue-92d15ea5dd7f4c58981df85c7c86
:
```

ORACLE

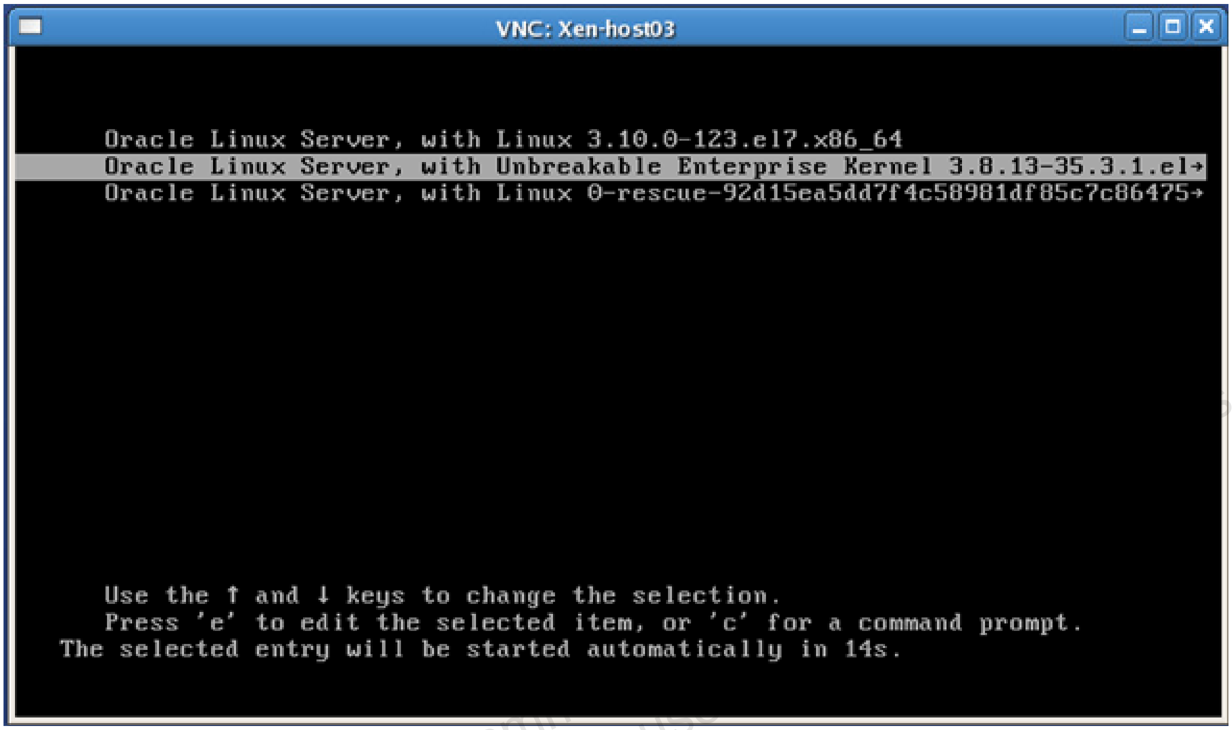
The GRUB 2 configuration file, `/boot/grub2/grub.cfg`, contains **`menuentry`** stanzas, which represent an installed Linux kernel. Each stanza begins with the **`menuentry`** keyword with options. Each **`menuentry`** is also a single boot menu entry in the GRUB 2 menu. The associated block of code is enclosed in curly brackets, `{ }`.

The screen shows a sample `menuentry` stanza for the Unbreakable Enterprise Kernel (UEK), version `3.8.13-35.3.1.el7uek.x86_64`. Indent everything within the curly brackets for readability. The stanza includes a `linux16` directive followed by the path to the kernel and an `initrd16` directive followed by the path to the `initramfs` image. The `linux16` directive specifies the kernel version number to be booted as well as kernel boot parameters. A separate `/boot` partition was created; therefore, the path to the kernel (as well as to the `initramfs` image) are relative to `/boot`.

The `initrd16` directive must point to the location of the `initramfs` file corresponding to the same kernel version. In other words, the kernel as given on the `linux16 /vmlinuz-<kernel_version>` line must match the version number of the `initramfs` image given on the `initrd16 /initramfs-<kernel_version>.img` line of each stanza.

The real root file system is specified by the `root` directive in the `linux16` line. The root file system contains, among other things, the `/etc/fstab` file identifying the rest of the file systems to be mounted.

GRUB 2 Menu



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

- Red Hat Compatible Kernel (RHCK) (`3.10.0-123.el7.x86_64`)
- Oracle Linux Unbreakable Enterprise Kernel (UEK) (`3.8.13-35.3.1.el7uek.x86_64`)
- Rescue kernel

These entries correspond to the `menuentry` line in the `/boot/grub2/grub.cfg` file:

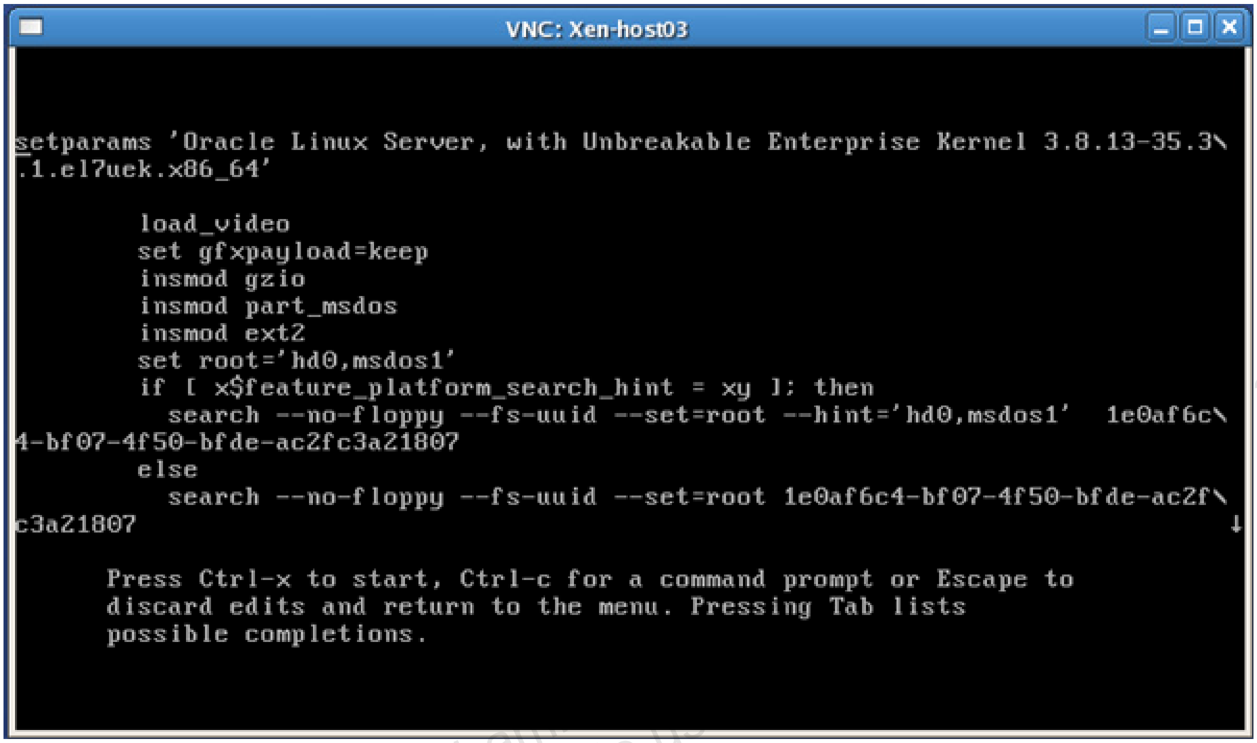
```
# grep '^menuentry' /boot/grub2/grub.cfg
menuentry 'Oracle Linux Server, with Linux 3.10... {
menuentry 'Oracle Linux Server, with Unbreakable Linux ... {
menuentry 'Oracle Linux Server, with Linux 0-rescue-... {
```

At the GRUB 2 menu, a different boot menu item can be selected by using the up arrow and down arrow keys. Kernel parameters can be changed or additional parameters can be passed to the kernel by using the `e` command. Highlight the kernel you want to modify, and enter `e` to edit directives within the `menuentry` stanza.

Press `c` to access the GRUB 2 command-line interface. Press ENTER to boot the highlighted kernel.

The rescue kernel boots a Red Hat Compatible Kernel (RHCK) with a known good `initramfs`. If your system does not boot with the desired kernel, you can boot the rescue kernel and correct the problem. Another use case for the rescue kernel is booting the system with different hardware, such as the disk is being accessed remotely or has been physically moved between systems.

Editing a GRUB 2 Menu Option



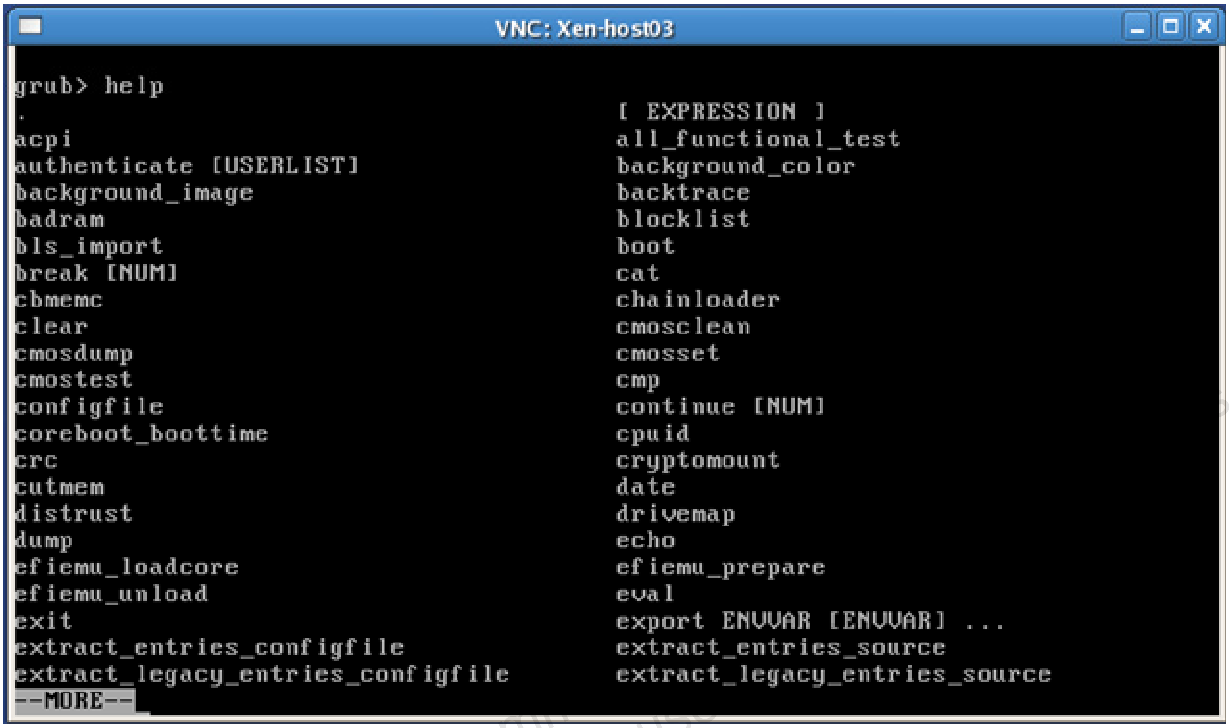
The slide shows the `menuentry` stanza for the selected kernel entry. You can edit anything within the `{ }` stanza for the selected entry. The entire content of the stanza is not shown on the screen. Use the down arrow key to display the lines at the end of the stanza.

Use the arrow keys to position the cursor at the point that you want to modify. The Backspace and Delete keys can also be used. Press Enter to save changes, or press Esc to discard changes.

All editing changes at boot time are temporary. GRUB 2 does not update the configuration file. For changes to be permanent, edit the template scripts in `/etc/grub.d` and/or the menu-configuration settings in `/etc/default/grub` and use the `grub2-mkconfig` command to re-generate `grub.cfg` as described previously.

Press Ctrl-x to start the boot sequence using the newly made changes. Press Ctrl-c to display the `grub>` command-line entry prompt.

GRUB 2 Command Line



The example in the slide shows the output of entering `help` at the `grub>` prompt to display available GRUB 2 commands. Note that not all of the commands are displayed on the screen. Type `help` to display a list of the most useful GRUB 2 commands. Type `help` followed by a command to display a brief synopsis and description of the command specified. Example:

```
grub> help background_image
Usage: background_image [-m (stretch|normal)] FILE
Load background image for active terminal.

-m, --mode=stretch|normal  Background image mode.
-h, --help                 Display this help and exit.
-u, --usage                Display the usage of this command and exit.
```

Introduction to `systemd`

- New system and service manager in Oracle Linux 7
- Backward compatible with SysV init scripts
- Replaces Upstart as the default initialization system
- The first process that starts after the system boots
- Speeds up booting by loading services concurrently
- Allows you to manage various types of units on a system, example:
  - services (*name*.service)
  - targets (*name*.target)
  - devices (*name*.device)
  - file system mount points (*name*.mount)
  - sockets (*name*.socket)

ORACLE

`systemd` is the new system and service manager in Oracle Linux 7. It is backward compatible with SysV `init` scripts used by previous versions of Oracle Linux including Oracle Linux 6. It replaces Upstart as the default initialization system.

`systemd` is the first process that starts after the system boots, and is the final process that is running when the system shuts down. It controls the final stages of booting and prepares the system for use. It also speeds up booting by loading services concurrently.

`systemd` allows you to manage various types of units on a system, including services (*name*.service) and targets (*name*.target), devices (*name*.device), file system mount points (*name*.mount), and sockets (*name*.socket).

`systemd` units are defined by unit configuration files located in the following directories:

- `/usr/lib/systemd/system`: `systemd` units distributed with installed RPM packages.
- `/run/systemd/system`: `systemd` units created at runtime. This directory takes precedence over the directory with installed service units.
- `/etc/systemd/system`: `systemd` units created and managed by the system administrator. This directory takes precedence over the directory with runtime units.

A complete list of available `systemd` unit types is as follows:

- **Service unit:** Start and control daemons and the processes they consist of. Service unit file names have a `.service` extension. See the `systemd.service(5)` man page for details.
- **Target unit:** Group units or provide well-known synchronization points during boot-up. Target unit file names have a `.target` extension. See the `systemd.target(5)` man page for details.
- **Automount unit:** Provide automount capabilities for on-demand mounting of file systems as well as parallelized boot-up. Automount unit file names have a `.automount` extension. See the `systemd.automount(5)` man page for details.
- **Device unit:** Expose kernel devices in `systemd` and can also be used to implement device-based activation. Device unit file names have a `.device` extension. See the `systemd.device(5)` man page for details.
- **Mount unit:** Control mount points in the file system. Mount unit file names have a `.mount` extension. See the `systemd.mount(5)` man page for more details.
- **Path unit:** Activate other services when file system objects change. Path unit file names have a `.path` extension. See the `systemd.path(5)` man page for details.
- **Scope unit:** Similar to service units but manage foreign processes instead of starting them as well. Scope unit file names have a `.scope` extension. See the `systemd.scope(5)` man page for details.
- **Slice unit:** Group units that manage system processes, such as service units and scope units, in a hierarchical tree for resource management purposes. Slice unit file names have a `.slice` extension. See the `systemd.slice(5)` man page for details.
- **Snapshot unit:** Can be used to temporarily save the state of the set of `systemd` units, which can later be restored by activating the saved snapshot unit. Snapshot unit file names have a `.snapshot` extension. See the `systemd.snapshot(5)` man page for details.
- **Socket unit:** Encapsulate local interprocess communication (IPC) or network sockets in the system, which are useful for socket-based activation. Socket unit file names have a `.socket` extension. See the `systemd.socket(5)` man page for details. For details on socket-based activation and other forms of activation, see the `daemon(7)` man page.
- **Swap unit:** Encapsulate memory swap partitions or swap files. Swap unit file names have a `.swap` extension. See the `systemd.swap(5)` man page for details.
- **Timer unit:** Use to trigger activation of other units based on timers. Timer units have a `.timer` extension. See the `systemd.timer(5)` man page for details.

Use the `systemctl` command as follows to list all installed unit files:

```
# systemctl list-unit-files
UNIT FILE                               STATE
proc-sys-fs-binfmt_misc.automount       static
dev-hugepages.mount                     static
proc-fs-nfsd.mount                      static
...
```

See the `systemctl(1)` man page for a description of the STATE column.

systemd Features

- Services are started in parallel wherever possible by using socket-based activation and D-Bus.
- Daemons can be started on demand.
- Processes are tracked by using Control Groups (cgroups).
- Snapshotting of the system state and restoration of the system state from a snapshot is supported.
- Mount points can be configured as `systemd` targets.

ORACLE

The `systemd` system and service manager provides the following features:

- Listening sockets are created at boot time for all system services that support this type of activation. This allows services to start in parallel and also allows restarting of services without loss of any messages. The corresponding socket remains accessible and all messages are queued.
- System services that use D-Bus for interprocess communication can be started on-demand when a client application attempts to communicate.
- System services that support device-based activation can be started on-demand when a particular type of hardware becomes available.
- System services that support path-based activation can be started on-demand when a particular file or directory changes its state.
- Processes are tracked by using Control Groups (cgroups). A cgroup is a collection of processes that are bound together so that you can control their access to system resources.
- Snapshotting of the system state and restoration of the system state from a snapshot is supported.
- Mount and automount points can be monitored and managed by `systemd`.

`systemd` Service Units

- Previous versions of Oracle Linux use scripts in the `/etc/rc.d/init.d` directory to control services.
- In Oracle Linux 7, these scripts have been replaced by `systemd` service units.
- Use the `systemctl` command to list information about service units.
- To list all loaded service units:

```
# systemctl list-units --type service --all
```

- To see which service units are enabled:

```
# systemctl list-unit-files --type service
```


Previous versions of Oracle Linux use init scripts located in the `/etc/rc.d/init` directory to start and stop services. In Oracle Linux 7, these init scripts have been replaced with `systemd` service units. Service units have a `.service` extension. Use the `systemctl` command as follows to list all loaded service units:

```
# systemctl list-units --type service --all
UNIT                LOAD   ACTIVE  SUB     DESCRIPTION
abrt-ccpp.service   loaded active  exited  Install ABRT core...
abrt-oops.service   loaded active  running ABRT kernel log ...
...
```

The name of the service unit is displayed, the unit load state, the high-level (ACTIVE) and low-level (SUB) unit activation state, and a description of the service unit. Omit the `--all` option to list only the active service units. Use the `list-unit-files` option to see which service units are enabled:

```
# systemctl list-unit-files --type service
UNIT FILE                          STATE
abrt-ccpp.service                  enabled
...
```

Displaying the Status of Services

- `systemd` service units correspond to system services.
- To display detailed information about the `httpd` service:

```
# systemctl status httpd
httpd.service - The Apache HTTP Server
...
```

- To check whether a service is running (active) or not running (inactive):

```
# systemctl is-active sshd
active
```

- To check whether a service is enabled:

```
# systemctl is-enabled sshd
enabled
```


`systemd` service units correspond to system services. Use the following command to display detailed information about a service unit. This example displays information about the `httpd.service` service unit. You can omit the `.service` extension:

```
# systemctl status httpd.service
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
Active: inactive (dead)
```

The following information is available for the specified service unit:

- `Loaded`: If the service is loaded, the absolute path to the service unit file, and if the service unit is enabled
- `Active`: If the service unit is running and a timestamp
- `Main PID`: The Process ID of the corresponding system service and the service name
- `Status`: Additional information about the corresponding system service
- `Process`: Additional information about related processes
- `CGroup`: Additional information about related Control Groups

The most recent log entries are displayed if the command is executed by the `root` user.

The following example displays detailed information about an active service, `sshd`:

```
# systemctl status sshd
sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
Active: active (running) since <date_time; hours> ago
Process: 32626 ExecStartPre=/usr/sbin/sshd-keygen (code=exited,
status=0/SUCCESS)
Main PID: 32640 (sshd)
CGroup: /system.slice/sshd.service
        |-32640 /usr/sbin/sshd -D
<Log file entries>
```

Use the following command to check whether a service is running (active) or not running (inactive):

```
# systemctl is-active sshd
active
```

Use the following command to check whether a service is enabled or disabled:

```
# systemctl is-enabled sshd
enabled
# systemctl is-enabled httpd
disabled
```

Starting and Stopping Services

`service` Utility	`systemctl` Utility	Description
`service` *`name`* `start`	`systemctl start` *`name`*	Starts a service
`service` *`name`* `stop`	`systemctl stop` *`name`*	Stops a service
`service` *`name`* `restart`	`systemctl restart` *`name`*	Restarts a service
`service` *`name`* `condrestart`	`systemctl try-restart` *`name`*	Restarts a service only if it is running
`service` *`name`* `reload`	`systemctl reload` *`name`*	Reloads a configuration
`service` *`name`* `status`	`systemctl status` *`name`*	Checks whether a service is running
`service --status-all`	`systemctl list-units --type service --all`	Displays the status of all services

In previous versions of Oracle Linux, the `service` utility is used to stop and start services. In Oracle Linux 7, the `systemctl` utility provides an equivalent set of subcommands. The table in the slide shows a comparison of the `service` utility with `systemctl`.

The following set of commands checks the status of the `httpd` service, starts the service, and checks the status a second time:

```
# systemctl status httpd
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
Active: inactive (dead)
# systemctl start httpd
# systemctl status httpd
httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
Active: active (running) since <date_time>; 3s ago
Main PID: 13775 (httpd)
...
```

Enabling and Disabling Services

`chkconfig` Utility	`systemctl` Utility	Description
`chkconfig` *`name`* `on`	`systemctl enable` *`name`*	Enables a service
`chkconfig` *`name`* `off`	`systemctl disable` *`name`*	Disables a service
`chkconfig --list` *`name`*	`systemctl status` *`name`* `systemctl is-enabled` *`name`*	Checks whether a service is enabled
`chkconfig --list`	`systemctl list-unit-files --type service`	Lists all services and checks whether they are enabled

In previous versions of Oracle Linux, the `chkconfig` utility is used to enable and disable services. In Oracle Linux 7, the `systemctl` utility provides an equivalent set of subcommands. The table in the slide shows a comparison of the `chkconfig` utility with `systemctl`.

The following example enables the `httpd` service:

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

To disable the `httpd` service:

```
# systemctl disable httpd
rm '/etc/systemd/system/multi-user.target.wants/httpd.service'
```

Notice that the symbolic link is removed when the service is disabled.

`systemd` Target Units

- Previous versions of Oracle Linux use SysV init run levels to allow a system to be used for a specific purpose.
  – Specific services are started at a specific run level.
- In Oracle Linux 7, run levels have been replaced with `systemd` target units.
- Target units have a `.target` extension.
- Target units allow you to start a system with only the services that are required for a specific purpose.
- To list the predefined `systemd` run level target units:

```
# find / -name "runlevel*.target"
/usr/lib/systemd/system/runlevel4.target
/usr/lib/systemd/system/runlevel5.target
...
```


Previous versions of Oracle Linux use SysV `init` run levels. These run levels provided the ability to use systems for different purposes and only start the services needed for a specific purpose, at a specific run level. In Oracle Linux 7, run levels have been replaced with `systemd` target units. Target units have a `.target` extension and similar to run levels, target units allow you to start a system with only the services that are required for a specific purpose.

Oracle Linux 7 is distributed with a set of predefined targets that are similar to run levels in previous versions of Oracle Linux. The following command returns the absolute pathname of these predefined `systemd` run level target units:

```
# find / -name "runlevel*.target"
/usr/lib/systemd/system/runlevel4.target
/usr/lib/systemd/system/runlevel5.target
/usr/lib/systemd/system/runlevel6.target
/usr/lib/systemd/system/runlevel1.target
/usr/lib/systemd/system/runlevel2.target
/usr/lib/systemd/system/runlevel3.target
/usr/lib/systemd/system/runlevel0.target
```

Comparision of SysV Run Levels and Target Units

Run Level	Target Units	Description
0	runlevel0.target, poweroff.target	Shut down and power off
1	runlevel1.target, rescue.target	Set up a rescue shell
2,3,4	runlevel[234].target, multi-user.target	Set up a nongraphical multi-user shell
5	runlevel5.target, graphical.target	Set up a graphical multi-user shell
6	runlevel6.target, reboot.target	Shut down and reboot the system

The table in the slide shows that there are system-state targets that are aliases for the equivalent run level targets. Each `runlevel[0123456].target` file is a symbolic link to the system-start target equivalents. For example:

```
# cd /usr/lib/systemd/system
# ls -l runlevel0.target
lrwxrwxrwx ... runlevel0.target -> poweroff.target
# ls -l runlevel1.target
lrwxrwxrwx ... runlevel1.target -> rescue.target
# ls -l runlevel2.target
lrwxrwxrwx ... runlevel2.target -> multi-user.target
# ls -l runlevel5.target
lrwxrwxrwx ... runlevel5.target -> graphical.target
# ls -l runlevel6.target
lrwxrwxrwx ... runlevel6.target -> reboot.target
```

Working with Target Units

- To view which target unit is used by default:

```
# systemctl get-default
graphical.target
```

- To list the currently active targets on a system:

```
# systemctl list-units --type target
...
```

  – Specify the `--all` option to view all targets on the system.

- To change the default target to `multi-user.target`:

```
# systemctl set-default multi-user.target
...
```

- To change the active target to `multi-user.target`:

```
# systemctl isolate multi-user.target
```


Use the following command to view which target unit is used by default:

```
# systemctl get-default
graphical.target
```

The `graphical.target` target unit indicates that the system is running in a graphical, multi-user state. This is similar to run level 5 in a SysV init system. This is confirmed by the following command:

```
# runlevel
N 5
```

Notice that the `runlevel` command still exists but is only included for compatibility reasons.

Use the following command to list the currently active targets on a system:

```
# systemctl list-units --type target
UNIT             LOAD   ACTIVE SUB    DESCRIPTION
basic.target     loaded active active Basic System
cryptsetup.target loaded active active Encrypted Volumes
...
```

The default target unit is represented by the `/etc/systemd/system/default.target` file. This file is a symbolic link to the current default target unit. In the following example, the `graphical.target` unit is the default:

```
# systemctl get-default
graphical.target
```

Notice the `default.target` symbolic file points to this file:

```
# ls -l /etc/systemd/system/default.target
lrwxrwxrwx ... /etc/systemd/system/default.target ->
/lib/system/system/graphical.target
```

Use the following command to change the default target unit (for example, to change the default to the `multi-user.target` unit):

```
# systemctl set-default multi-user.target
rm '/etc/systemd/system/default.target'
ln -s '/usr/lib/systemd/system/multi-user.target'
'/etc/systemd/system/default.target'
```

Notice that the `default.target` symbolic link has changed, and is now pointing to the `multi-user.target` unit:

```
# ls -l /etc/systemd/system/default.target
lrwxrwxrwx ... /etc/systemd/system/default.target ->
/lib/system/system/multi-user.target
```

The "`set-default`" command does not change the state of the system. Use the following command to change the currently active system target (for example, to change the currently active system target to `multi-user.target`):

```
# systemctl isolate multi-user.target
```

This command is similar to using `telinit <runlevel>` to change the current run level. This `telinit` command still exists but is only included for compatibility reasons.

You can also use the following command to enter the default target unit:

```
# systemctl default
```

This is equivalent to the following command:

```
# systemctl isolate default.target
```

Rescue Mode and Emergency Mode

- Rescue mode is the same as single-user mode.
  - Attempts to mount local file systems and start some system services.
  - Does not start the network service and does not allow other users to log on to the system.
  - To change to rescue mode:

```
# systemctl rescue
```

- Emergency mode allows you attempt repairs even when your system is unable to enter rescue mode.
  - To change to emergency mode:

```
# systemctl emergency
```

- Changing to rescue mode and emergency mode prompts for the `root` password.

Rescue mode is the same as single-user mode. This mode can be used when a condition exists that prevents your system from completing the regular boot process. The system attempts to mount local file systems and start some system services. But rescue mode does not start the network service and does not allow other users to log on to the system. Changing to rescue mode prompts for the `root` password.

Use the following command to enter rescue mode:

```
# systemctl rescue
```

This command sends a message to all users currently logged on that the system is going down. Alternatively, use the following command. The following command does not send a message to the users.

```
# systemctl isolate rescue.target
```

Emergency mode mounts the root system read-only and does not attempt to mount any other local file systems. Emergency mode allows you attempt repairs even when your system is unable to enter rescue mode. Use the following command to enter emergency mode. This mode requires the `root` password and also sends a message to all logged in users.

```
# systemctl emergency
```

Shutting Down, Suspending, or Rebooting Commands

Older Command	`systemctl` equivalent	Description
`halt`	`systemctl halt`	Halts the system
`poweroff`	`systemctl poweroff`	Powers off the system
`reboot`	`systemctl reboot`	Restarts the system
`pm-suspend`	`systemctl suspend`	Suspends the system
`pm-hibernate`	`systemctl hibernate`	Hibernates the system
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Hibernates and suspends the system

In Oracle Linux 7, the `systemctl` utility replaces some older power management commands used in previous versions of Oracle Linux. The table in the slide compares the older commands with the equivalent `systemctl` command. The old commands listed in the slide still exist for compatibility reasons.

Suspending the system saves the system state in RAM and with the exception of the RAM module, powers off most of the devices in the machine. When you turn the machine back on, the system then restores its state from RAM without having to boot again. Because the system state is saved in RAM and not on the hard disk, restoring the system from suspend mode is significantly faster than restoring it from hibernation, but as a consequence, a suspended system state is also vulnerable to power outages.

Hibernating the system saves the system state on the hard disk drive and powers off the machine. When you turn the machine back on, the system then restores its state from the saved data without having to boot again. Because the system state is saved on the hard disk and not in RAM, the machine does not have to maintain electrical power to the RAM module, but as a consequence, restoring the system from hibernation is significantly slower than restoring it from suspend mode.

Summary

In this lesson, you should have learned to:

- Describe the Oracle Linux 7 boot process
- Describe and configure the GRUB 2 bootloader
- Describe and configure kernel boot parameters
- Describe the `systemd` system and service manager
- Describe `systemd` service units
- Configure services
- Describe `systemd` target units
- Configure Rescue Mode and Emergency Mode
- Perform shut down, suspend, and reboot operations

Quiz

Which of the following files must you edit to define persistent kernel boot parameters?

a. `/etc/grub.d/00_header`

b. `/boot/grub2/grub.cfg`

c. `/etc/default/grub`

d. `/proc/cmdline`

Quiz

Which of the following statements are true about `systemd`?

a. It is a system and service manager in Oracle Linux 7.
b. It is backward compatible with SysV init scripts.
c. It provides management of services, devices, file system mount points, and sockets.
d. It runs as the first process on boot.
e. All of the above

Quiz

Which of the following statements are true of the `systemctl` utility?

a. The `systemctl start <name>` command causes the `<name>` service to start at boot time.

b. The `systemctl enable <name>` command causes the `<name>` service to start at boot time.

c. The `systemctl is-active <name>` command checks whether the `<name>` service is configured to start at boot time.

d. The `systemctl status <name>` command checks whether the `<name>` service is running and whether the `<name>` service is enabled.

Practice 4: Overview

The practices for this lesson cover the following topics:

- Exploring the GRUB 2 bootloader
- Booting different kernels
- Using the GRUB 2 menu
- Exploring `systemd` units
- Working with `systemd` target and service units

ORACLE

5

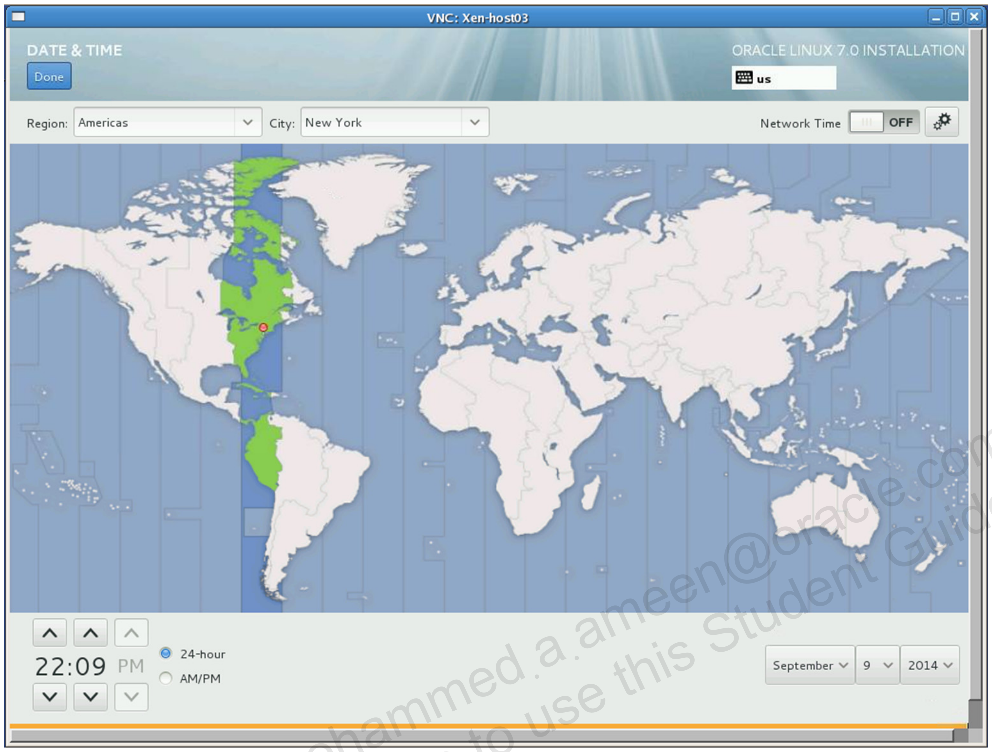
System Configuration

Objectives

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

- System date and time configuration
- Configuring NTP and chrony
- The `/etc/sysconfig` directory
- The `proc` file system
- The `sysfs` file system
- The `sysctl` utility

Configuring System Date and Time During Installation



The slide shows the DATE & TIME selection window, which is accessible during installation of Oracle Linux 7. From this screen you can select the time zone and can manually adjust the date and time.

You can also enable and configure Network Time Protocol (NTP) during installation. However, you need to set up networking first if you want to use NTP. NTP provides a method of verifying and correcting your computer's time by synchronizing it with another system. Assuming you are connected to a network, you can enable NTP by moving the Network Time switch to the ON position. You can then click the configuration icon, located next to the Network Time switch, to set the date and time by using NTP. Configuring NTP is described in more detail later in this lesson.

Configuring System Date and Time from the Command Line

- Oracle Linux 7 has three command-line utilities to configure the system date and time:
  - `date`
  - `hwclock`
  - `timedatectl`
- Run the `date` command to display or set the system date and time.
  - Can also be used to display dates in the past or future
- Run the `hwclock` command to display or set the hardware clock date and time.
  - Can sync hardware clock with system time
  - Can sync system time with hardware clock

Oracle Linux 7 has three command-line utilities to configure the system date and time, the traditional `date` command, the `hwclock` command, and a new command, `timedatectl`, which is part of `systemd`.

Use the `date` command to display or set the system date and time. Run the `date` command with no arguments to display the current date and time:

```
# date
Fri Aug 22 12:02:03 MDT 2014
```

The `date` command provides a variety of output formatting options. For example, "`date +%A`" only displays the day of the week.

```
# date +%A
Friday
```

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

```
# date -d "1 year"
Sat Aug 22 12:06:12 MDT 2015
```

To display the date one month in the past:

```
# date –d "1 month ago"
Tue Jul 22 12:06:22 MDT 2014
```

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

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

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

```
# date +%D -s 2013-07-11
07/11/13
```

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

```
date +%T%p -s <HH:MM:SS>AM|PM –u
```

Run the `date --help` command to view additional formatting options and usage.

The `hwclock` Command

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

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

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

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

```
# date
Thu Jul 11 11:41:38 MDT 2013
# hwclock -s
# hwclock
Fri Aug 22 2014 10:38:31 AM MDT  -0.688168 seconds
# date
Fri Aug 22 10:38:32 MDT 2014
```

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

Using the `timedatectl` Utility

- The `timedatectl` utility is part of the `systemd` system and service manager.
- To display local, universal, and RTC time and time zone, NTP configuration, and DST information:

```
# timedatectl
```

- Use the following syntax to change the date and time:

```
# timedatectl set-time <YYYY-MM-DD>
# timedatectl set-time <HH:MM:SS>
```

- Use the following syntax to change the time zone:

```
# timedatectl set-timezone <time_zone>
```

- To enable clock synchronization over NTP:

```
# timedatectl set-ntp yes
```


Oracle Linux 7 offers another utility to configure and display date and time information, `timedatectl`. This utility is part of the `systemd` system and service manager. Running `timedatectl` with no arguments displays:

- Local time
- Universal time
- RTC time
- Time zone
- Network Time Protocol (NTP) configuration
- Daylight Savings Time (DST) information

The utility allows you to:

- Change the current date and time
- Set the time zone
- Enable automatic synchronization of the system clock with a remote server

Examples of using the command are given in the slide and on the following page.

To display date and time information:

```
# timedatectl
      Local time: Wed 2014-08-13 14:49:29 MDT
  Universal time: Wed 2014-08-13 20:49:29 UTC
        RTC time: Wed 2014-08-13 20:49:52
        Timezone: America/Denver (MDT, -0600)
     NTP enabled: no
NTP synchronized: no
 RTC in local TZ: no
      DST active: yes
 Last DST change: DST began at
                  Sun 2014-03-09 01:59:59 MST
                  Sun 2014-03-09 03:00:00 MDT
 Next DST change: DST ends (the clock jumps one hour backwards
...
                  Sun 2014-11-02 01:59:59 MDT
                  Sun 2014-11-02 01:00:00 MST
```

Change Current Date and Time

Use the `set-time` argument to change the current date and the current time. To change the date, use the following syntax:

```
# timedatectl set-time <YYYY-MM-DD>
```

To change the current time, use the following syntax. Enter the hour by using a 24-hour clock.

```
# timedatectl set-time <HH:MM:SS>
```

To configure your system to maintain the clock in the local time, use the following command:

```
# timedatectl set-local-rtc yes
```

To configure your system to use UTC, use the following command:

```
# timedatectl set-local-rtc no
```

Change Time Zone

Use the following command to list all available time zones:

```
# timedatectl list-timezones
```

Use the `set-timezone` argument to change the time zone. The following example changes the time zone to `America/New_York`:

```
# timedatectl set-timezone America/New_York
```

Enable or Disable Synchronizing the System Clock with a Remote Server over NTP

Use the `set-ntp` argument to enable or disable automatic synchronization of your system clock with a remote server over the Network Time Protocol (NTP). Use the following command to enable this feature:

```
# timedatectl set-ntp yes
```

Use the following command to disable this automatic synchronization over NTP:

```
# timedatectl set-ntp no
```

Using Network Time Protocol

- Synchronize your computer's time with another system.
  - NTP servers are arranged in layers, or strata, from 1 to 15.
  - Specify NTP servers in `/etc/ntp.conf`.
- Start the `ntpd` service as follows:

```
# systemctl start ntpd
```

- Use the `ntpq` command to query NTP operations and performance.
  - To view which NTP server your system is synchronized with:

```
# ntpq -p
```

- Use the `ntpstat` command to view network time synchronization status:

```
# ntpstat
synchronized to NTP server (10.150.36.208) at stratum 5
...
```


NTP provides a method of verifying and correcting your computer's time by synchronizing it with another system. By default, there are four public server entries in the NTP configuration file, `/etc/ntp.conf`, which are specified by the `server` directive.

```
# grep server /etc/ntp.conf
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
server 3.rhel.pool.ntp.org
```

NTP Strata

Notice that each public server listed above has a number, or stratum, associated with it. NTP servers are arranged in layers, or strata, from number 1 through 15. The source of time signals for these NTP servers are atomic clocks, which are provided by many standards bodies around the world. Atomic clocks are referred to as stratum 0 because they are the source of the time signals. All stratum 0 atomic clocks are attached to a server, which is referred to as stratum 1. The NTP servers are then layered and numbered accordingly. A stratum 2 server reads from stratum 1 and serves to lower strata. A stratum 3 server reads from stratum 2 and serves to lower strata.

This process continues down to stratum 15, which is the lowest valid stratum. Stratum 16 indicates an unsynchronized state. More information about these time servers can found at www.pool.ntp.org.

Instead of using a predefined public server, you can specify a local reference server in the `/etc/ntpd.conf` file. For example:

```
# vi /etc/ntpd.conf
server 192.0.2.1
```

Another directive in the configuration file is `driftfile`. The default setting is as follows:

```
driftfile /var/lib/ntp/drift
```

This drift file contains one value used to adjust the system clock frequency after every system or service start. Refer to the `ntp.conf(5)` man page for a description of the configuration parameters.

The NTP daemon

The `ntpd` program is the user space daemon that synchronizes the system clock with remote NTP time servers or local reference clocks. The daemon reads the configuration file at system start or when the service is restarted. After editing the `/etc/ntp.conf` file, use the `systemctl` command to start the NTP daemon:

```
# systemctl start ntpd
```

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

```
# systemctl enable ntpd
```

You also need to open UDP port 123 in the firewall for NTP packets. Refer to the `ntpd(8)` man page for more information on the NTP daemon.

Other NTP Utilities

Use the `ntpq` command to query the NTP daemon operations and to determine performance. Run `ntpq` with no arguments to initiate an interactive session. Use the `help` command to list available `ntpq` commands and to request help on an individual `ntpq` command. For example:

```
# ntpq
ntpq> help
ntpq> help peers
```

Use the `-p` option (or `peers` command) to display a list of peers known to the server as well as a summary of their state. For example:

```
# ntpq -p
*ovmdev1.us.orac LOCAL(0) ...
```

The `*` indicates your system is synchronized with the `ovmdev1` server. Refer to the `ntpq(8)` man page for additional information and usage of this command.

Use the `ntpstat` command to show network time synchronization status. If your local system is synchronized to a reference time source, `ntpstat` reports the approximate time accuracy. For example:

```
# ntpstat
synchronized to NTP server (10.150.36.208) at stratum 5
    time correct to within 17 ms
    polling server every 128 s
```

Configuring NTP by Using Chrony

- Chrony provides another implementation of NTP.
- Chrony is designed for systems that are often powered down or disconnected from the network.
- The main configuration file is `/etc/chrony.conf`.
  - Parameters are similar to those in the `/etc/ntp.conf` file.
- `chronyd` is the daemon that runs in user space.
- `chronyc` is a command-line program that provides a command prompt and a number of commands. Examples:
  - `tracking`: Displays system time information
  - `sources`: Displays information about current sources
- The user guide is accessible at http://chrony.tuxfamily.org/manual.html.

ORACLE

Chrony is a suite of utilities that provides another implementation of NTP. Chrony is designed for mobile systems and virtual machines that are often powered down or disconnected from the network. Systems that are not permanently connected to a network take a relatively long time to adjust their system clocks with the NTP daemon, `ntpd`.

Chrony consists of `chronyd`, a daemon that runs in user space, and `chronyc`, a command-line program for making adjustments to `chronyd`. The `chronyd` daemon makes adjustments to the system clock that is running in the kernel. It uses NTP to synchronize with another system when network access is available. When network access is not available, `chronyd` uses the last calculated drift stored in the drift file to synchronize the system time.

You can also make corrections manually with the `chronyc` utility. This utility provides a command prompt and a number of commands to monitor `chronyd` performance and make changes to `chronyd`. Use the `help` command to display the commands. For example:

```
# chronyc
...
chronyc> help
tracking: Display system time information
sources: Display information about current sources
```

Install the chrony package by using the following command:

```
# yum install chrony
```

Use the following commands to start chronyd and to ensure chronyd starts at boot time:

```
# systemctl start chronyd
# systemctl enable chronyd
```

The user guide is accessible at http://chrony.tuxfamily.org/manual.html. The user guide provides typical operating scenarios for using chrony such as:

- Computers connected to the Internet most of the time, or on a private network with NTP servers
- Infrequent connection to true NTP servers
- Isolated networks with no reference clocks
- A home computer with a dial-up connection to the Internet and is turned off when not in use

The main configuration file is /etc/chrony.conf. Configuration file parameters are described at http://chrony.tuxfamily.org/manual.html#Configuration-file. The following configuration example is for systems that have infrequent connection to true NTP servers:

```
# cat /etc/chrony.conf
server a.b.c offline
server d.e.f offline
server g.h.i offline
keyfile /etc/chrony.keys
generatecommandkey
driftfile /var/lib/chrony/drift
makestep 10 3
```

The parameters are described as follows:

- **server**: Identifies the NTP servers you want to use. The offline keyword indicates that the servers are not contacted until chronyd receives notification that the link to the Internet is present.
- **keyfile**: File containing administrator password. Password allows chronyc to log in to chronyd and notify chronyd of the presence of the link to the Internet.
- **generatecommandkey**: Generates a random password automatically on the first chronyd start.
- **driftfile**: Location and name of file containing drift data.
- **makestep**: Step (start anew) system clock if a large correction is needed. The parameters 10 and 3 would step the system clock if the adjustment is larger than 10 seconds, but only in the first three clock updates.

Refer to http://www.ntp.org/ntpfaq/NTP-s-algo.htm for an explanation of clock step. This URL also provides more information on how NTP works.

The `/etc/sysconfig` Directory

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

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

- **`atd`:** This file is used to specify additional command-line arguments for `atd` daemon.
- **`authconfig`:** This file sets the authorization to be used on the host. For example, `USEMKHOMEDIR=no` disables creating a home directory for a user on the first login.
- **`autofs`:** This file defines custom options for the automatic mounting of devices. It controls the operation of the `automount` daemons.
- **`crond`:** This file is used to pass arguments to the `crond` daemon at boot time.
- **`iptables-config`:** This file stores information used by the kernel to set up packet filtering services at boot time or when the service is started.
- **`iptables`:** This file stores the actual firewall configuration rules.
- **`ntpd`:** This file is used to pass arguments to the `ntpd` daemon at boot time.

Many of the files and directories in the `/etc/sysconfig` directory are described further in subsequent lessons in this course.

The `proc` File System

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

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

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

Virtual File System

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

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

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

Viewing Virtual Files

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

```
# cat /proc/cpuinfo
processor  : 0
vendor_id  : GenuineIntel
cpu family : 6
model      : 6
model name : Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz
...
```

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

```
# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC ...
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA ...
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE ...
...
```

Changing Virtual Files

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

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

For example, to change the host name, enter:

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

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

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

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

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

Top-Level Files Within `/proc`

```
root@host03:~
File  Edit  View  Search  Terminal  Help
[root@host03 ~]# ls /proc
1     22     29680  3      464   580        driver         mtd
10    220    29684  30     466   6          execdomains    mtrr
11    221    29689  30069  467   647        fb             net
1127  23     29692  30072  482   65         filesystems    pagetypeinfo
1134  24     29698  30073  490   7          fs             partitions
1146  26     29702  30106  491   7334       interrupts     sched_debug
1167  27     29706  30115  492   7417       iomem          schedstat
12    28     29716  315    494   7427       ioports        scsi
1259  29     29719  333    495   7493       irq            self
1261  29421  29731  336    497   7505       kallmodsyms    slabinfo
13    29432  29737  339    5     7515       kallsyms       softirqs
1305  29443  29741  341    507   8          kcore          stat
1389  29451  29747  382    508   9          keys           swaps
14    29452  29749  39     5092  acpi       key-users      sys
15    29517  29763  40     5123  buddyinfo  kmsg           sysrq-trigger
1525  29527  29770  41     513   bus        kpagecount     sysvipc
16    29594  29788  419    5130  cgroups    kpageflags     timer_list
17    29615  29819  420    514   cmdline    latency_stats  timer_stats
18    29619  29825  422    523   consoles   loadavg        tty
19    29623  29836  423    529   cpuinfo    locks          uptime
192   29641  29844  428    532   crypto     mdstat         version
193   29642  29846  43     548   cyclicinfo meminfo        vmallocinfo
2     29649  29853  442    559   devices    misc           vmstat
20    29670  29883  452    566   diskstats  modules        xen
21    29672  29893  454    573   dma        mounts         zoneinfo
[root@host03 ~]#
```


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

- `/proc/buddyinfo`: This file is used primarily for diagnosing memory fragmentation issues.
- `/proc/cmdline`: This file shows the parameters passed to the kernel at the time it is started.
- `/proc/cpuinfo`: This virtual file identifies the type of processor used by your system.
- `/proc/crypto`: This file lists all installed cryptographic ciphers used by the Linux kernel, including additional details for each.
- `/proc/devices`: This file displays the various character and block devices currently configured (not including devices whose modules are not loaded).
- `/proc/dma`: This file contains a list of the registered ISA DMA channels in use.
- `/proc/execdomains`: This file lists the execution domains currently supported by the Linux kernel, along with the range of personalities they support.

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

Process Directories in `/proc`

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

ORACLE

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

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

Other Directories in `/proc`

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

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

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

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

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

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

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

  Changes made by using the `echo` command are not persistent and disappear when the system is restarted. To make configuration changes take effect after the system is rebooted, add them to the `/etc/sysctl.conf` file.
- **`/proc/sys/dev`**: This directory provides parameters for particular devices on the system.
- **`/proc/sys/fs`**: This directory contains options and information concerning various aspects of the file system, including quota, file handle, and inode information.
- **`/proc/sys/kernel`**: This directory contains a variety of different configuration files that directly affect the operation of the kernel.
- **`/proc/sys/net`**: This directory contains subdirectories concerning various networking topics. You can alter the files within these directories to adjust the network configuration on a running system.
- **`/proc/sysvipc`**: This directory contains information about System V Interprocess Communication (IPC) resources. The files in this directory relate to System V IPC calls for messages (`msg`), semaphores (`sem`), and shared memory (`shm`).
- **`/proc/tty`**: This directory contains information about the available and currently used `tty` devices on the system. The `drivers` file is a list of the current `tty` devices in use.

The `sysfs` File System

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

```
root@host03:~
File  Edit  View  Search  Terminal  Help
[root@host03 ~]# ls -l /sys
total 0
drwxr-xr-x.  2 root root 0 Sep 19 04:55 block
drwxr-xr-x. 24 root root 0 Sep 19 04:55 bus
drwxr-xr-x. 42 root root 0 Sep 19 04:55 class
drwxr-xr-x.  4 root root 0 Sep 19 04:55 dev
drwxr-xr-x. 19 root root 0 Sep 19 04:55 devices
drwxr-xr-x.  4 root root 0 Sep 19 04:55 firmware
drwxr-xr-x.  6 root root 0 Sep 19 04:55 fs
drwxr-xr-x.  5 root root 0 Sep 19 04:55 hypervisor
drwxr-xr-x.  6 root root 0 Sep 19 04:55 kernel
drwxr-xr-x. 99 root root 0 Sep 19 04:55 module
drwxr-xr-x.  2 root root 0 Sep 19 04:56 power
[root@host03 ~]#
```


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

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

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

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

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

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

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

```
# ls /sys/module/xen*
...
/sys/module/xen_blkfront:
coresize  holders   initstate  parameters  sections    taint
drivers   initsize  notes      refcnt      srcversion  uevent

/sys/module/xen_netfront:
coresize  holders   initstate  refcnt    srcversion  uevent
drivers   initsize  notes      sections  taint
...
```

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

The `sysctl` Utility

- The `sysctl` utility is used to assign values to writable files in `/proc/sys`.
- To view kernel settings, enter:

```
# sysctl -a
```

- To assign values, the syntax is:

```
# sysctl -w <kernel parameter>=<value>
```

- Example:

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

- Changes are lost when the system is rebooted.
- To preserve settings, add them to the config file: `/etc/sysctl.conf`

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

```
# sysctl -a
abi.vsyscall32 = 1
...
```

This is the same information seen if each of the files were viewed individually, for example:

```
# cat /proc/sys/abi/vsyscall32
1
```

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

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

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

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

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

Quiz

Which of the following are Oracle Linux 7 command-line utilities to configure the system date and time?

a. `date`
b. `hwclock`
c. `timedatectl`
d. `datetimectl`
e. `chrony`

Quiz

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

a. `/etc/sysconfig`

b. `/proc/sys`

c. `/var/spool`

d. `/var/log`

Quiz

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

a. `/etc/sysconfig`

b. `/proc/sys`

c. `/var/spool`

d. `/var/log`

Quiz

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

a. `/etc`

b. `/proc/sys`

c. `/proc/pid`

d. `/proc`

Summary

In this lesson, you should have learned about:

- System date and time configuration
- Configuring NTP and chrony
- The `/etc/sysconfig` directory
- The `proc` file system
- The `sysfs` file system
- The `sysctl` utility

Practice 5: Overview

The practices for this lesson cover the following tasks:

- Configuring date and time
- Configuring NTP and chrony
- Exploring the `/etc/sysconfig` directory
- Exploring the `proc` file system
- Exploring the `sysfs` file system
- Using the `sysctl` utility

6

Package Management

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Linux package management
- Use the `rpm` utility
- Describe the Oracle Public Yum Server
- Describe and configure Yum repositories
- Use the `yum` utility
- Describe the Unbreakable Linux Network (ULN)
- Describe the steps to switch from Red Hat Network (RHN) to ULN

Introduction to Package Management

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

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

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

Oracle Public Yum Server

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

Unbreakable Linux Network (ULN)

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

rpm Utility

- Query options:

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

- Installing and updating packages:

```
# rpm -Uvh package_name
```

- Installing a new kernel:

```
# rpm -ivh kernel_package_name
```

- Removing packages:

```
# rpm -e package_name
```


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

Query Packages

To list all installed packages, use the following command:

```
# rpm -qa
spice-glib-0.20-8.el7.x86_64
opus-1.0.2-6.el7.x86_64
vinagre-3.8.2-4.el7.x86_64
libreport-python-2.1.11-10.0.2.el7.x86_64
...
```

The format of rpm package names is name-version-release.architecture. The example shows packages for version 7 of Oracle Linux (el7) with architectures of either:

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

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

```
# rpm –qi filesystem
Name        : filesystem
Version     : 3.2
Release     : 18.el7
Architecture: x86_64
...
```

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

```
# rpm –ql bash
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
/usr/sbin/alias
...
```

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

```
# rpm –qf /etc/hosts
setup-2.8.71-4.el7.noarch
```

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

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

Installing and Updating Packages

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

```
# rpm –Uvh zsh-5.0.2-7.el7.x86_64.rpm
```

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

Installing a New Kernel

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

```
# rpm –ivh kernel-uek-3.8.13-44.1.1.el7uek.x86_64.rpm
```

Removing Packages

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

```
# rpm –e zsh
```

Oracle Public Yum Server

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

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

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

For Oracle Linux 7, download the `yum repo` configuration file by running the following commands as `root`:

```
# cd /etc/yum.repos.d
# wget http://public-yum.oracle.com/public-yum-ol7.repo
```

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

```
# ls /etc/yum.repos.d
public-yum-ol7.repo
```

Contents of `public-yum-ol7.repo`:

```
# cat public-yum-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://public-
yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol7_u0_base]
name=Oracle Linux $releasever GA installation media copy
($basearch)
baseurl=http://public-
yum.oracle.com/repo/OracleLinux/OL7/0/base/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=0

[ol7_UEKR3]
...
[ol7_optional_latest]
...
[ol7_addons]
...
[ol7_MySQL56]
...
[ol7_MySQL55]
...
```

Enable the appropriate repository by editing the `yum repo` configuration file. Locate the section in the file for the repository that you plan to update from—for example, `[ol7_latest]`. Set `enabled` to `1` (enable) or `0` (disable). You can enable more than one repository.

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

```
# yum list
Installed Packages
GConf2.x86_64               3.2.6-8.el7       @anaconda/7.0
ModemManager.x86_64         1.1.0-6.git...el7 @anaconda/7.0
ModemManager-glib.x86_64    1.1.0-6.git...el7 @anaconda/7.0
...
```

Oracle Public Yum uses Akamai so when accessing Oracle Public Yum on an Oracle Network (in an office or while on VPN), you need to do the following (modify these settings for proper proxy URLs in geographies other than the US):

For `wget`, set the "`http_proxy`" environment variable:

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

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

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

You could use `ftp://gdsuln/YUM_local/OracleLinux` or `ftp://obiftp/YUM_local/OracleLinux`. Oracle hosts a copy of `ol6_latest`, `ol6_latest`, and `ol5_latest` on the utility servers worldwide. Sample configuration:

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

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

`yum` Configuration

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

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

```
# cat /etc/yum.conf
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
```

Global configurations are defined in the `[main]` section:

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

yum Repositories

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

```
# ls /etc/yum.repos.d
Ksplice-uptrack.repo    public-yum-ol7.repo
```

The `repo` files define which repositories to use. Each `repo` file includes specifications for several related repositories. For example, the `public-yum-ol7.repo` file holds `[ol7_latest]`, `[ol7_u0_base]`, `[ol7_UEKR3]`, and `[ol7_optional_latest]`:

```
# cat public-yum-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://public-
yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
...
```

The directives in the `repo` files include:

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

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

`yum` Utility

- List all packages:

```
# yum list
```

- List all installed packages:

```
# yum list installed
```

- List packages available to be installed:

```
# yum list available
```

- Check for updates to installed packages:

```
# yum check-update
```

- Update, install, remove packages:

```
# yum update package_name
# yum install package_name
# yum remove package_name
```

ORACLE

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

Listing Packages

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

To list all packages in all the repositories and all the packages installed on your system, use the following command:

```
# yum list
```

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

```
# yum list installed
```

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

```
# yum list available
```

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

```
# yum provides /etc/sysconfig/atd
at-3.1.13-17.el7.x86_64 : Job spooling tools
...
```

Checking for Updates

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

```
# yum check-update
```

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

Updating Packages

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

Updating a Single Package

To update a single package, use the following command syntax:

```
yum update package_name
```

For example, to update the `bind-libs` package, enter:

```
# yum update bind-libs
```

`yum` checks dependencies, displays dependencies resolved and a transaction summary, prompts "`Is this ok [y/N]`", waits for your response, and then downloads and installs the package and any dependent packages needed. Use `yum -y` to bypass the prompt.

Updating All Packages

To update all packages and their dependencies, enter `yum update` (without any arguments):

```
# yum update
```

Installing Packages

To install a new package together with any package dependencies, use the following syntax:

```
yum install package_name
```

For example, to install the `zsh` package, enter:

```
# yum install zsh
```

Updating and Installing Kernels

You do not need to worry about the distinction between installing and upgrading a kernel package when you use `yum`. `yum` always installs a new kernel regardless of whether you are using `yum update` or `yum install`.

Removing Packages

To remove a package, use the following syntax:

```
yum remove package_name
```

For example, to remove the `zsh` package, enter:

```
# yum remove zsh
```

yum Groups

- yum groups are collections of software packages referred to by a single "group" name.
- yum supports the following group commands:

```
# yum grouplist
# yum groupinfo groupname
# yum groupinstall groupname
# yum groupupdate groupname
# yum groupremove groupname
```


Software packages that interoperate with others can be packaged together as a group. The yum command includes several subcommands for managing these groups.

To list installed and available groups in two separate lists, use the following command:

```
# yum grouplist
```

To get detailed information about a specific group, use the following command:

```
# yum groupinfo groupname
```

To install or update packages in a group, use the following command:

```
# yum groupinstall groupname
# yum groupupdate groupname
```

To remove all packages in a group, use the following command:

```
# yum groupremove groupname
```

Unbreakable Linux Network (ULN)

ULN is accessed at https://linux.oracle.com/.

ORACLE

The Unbreakable Linux Network (ULN) is a comprehensive resource for Oracle Unbreakable Linux support subscribers. ULN offers Linux software patches, updates, and fixes, along with information on the `yum` program and support policies. Support subscribers can also download useful extra packages not included in the original distribution. ULN can be accessed at https://linux.oracle.com/. Also included on the ULN site are instructions to register with ULN, to create local `yum` repositories, and to switch from Red Hat Network (RHN) to ULN.

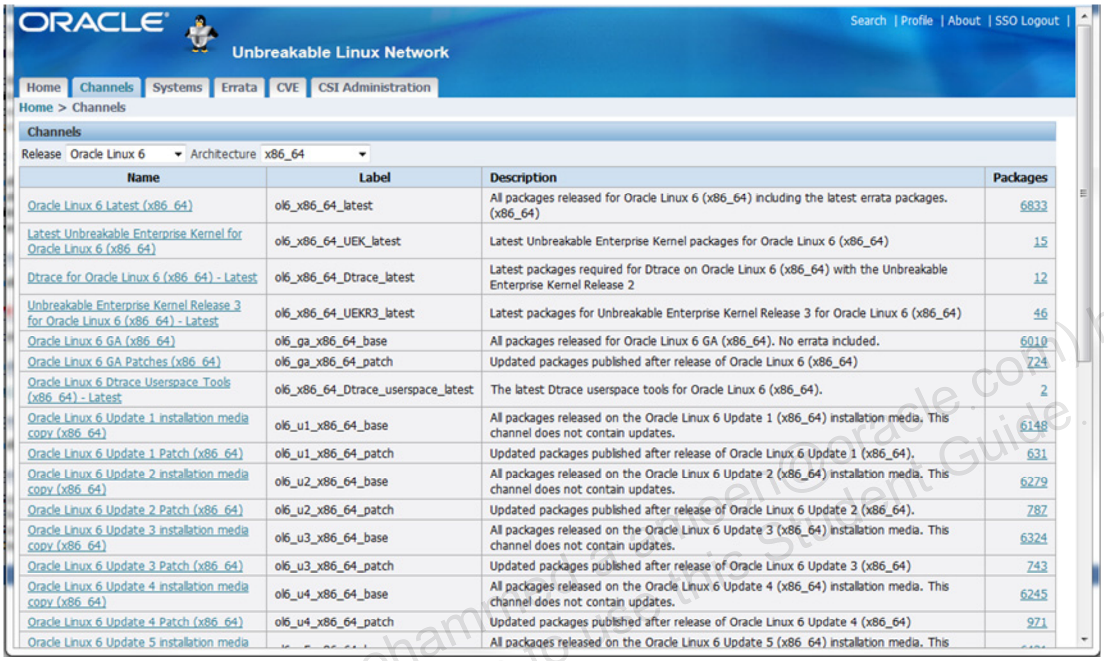
The Update Agent (`yum`)

ULN subscribers have the option of using `yum` to manage their systems. To use ULN and `yum`, users must register their systems with ULN and subscribe to a ULN channel. It is also possible to subscribe to multiple channels at once. There are several ULN channels available, and one containing the latest version is automatically chosen after registration, depending on the architecture and OS revision of the system to be managed.

After it is started, `yum` connects to the central ULN server repository and downloads the latest software packaged in RPM format. It then installs RPMs on the registered machine, maintaining a log. `yum` lets you choose which packages to update, because it is not necessary to install all the newly available packages. You can get a list of all the available packages and then choose which ones to download.

ULN Channels

Register your systems and subscribe to a ULN channel.



Name	Label	Description	Packages
Oracle Linux 6 Latest (x86_64)	ol6_x86_64_latest	All packages released for Oracle Linux 6 (x86_64) including the latest errata packages. (x86_64)	6833
Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)	ol6_x86_64_UEK_latest	Latest Unbreakable Enterprise Kernel packages for Oracle Linux 6 (x86_64)	15
Dtrace for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_Dtrace_latest	Latest packages required for Dtrace on Oracle Linux 6 (x86_64) with the Unbreakable Enterprise Kernel Release 2	12
Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64) - Latest	ol6_x86_64_UEKR3_latest	Latest packages for Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64)	46
Oracle Linux 6 GA (x86_64)	ol6_ga_x86_64_base	All packages released for Oracle Linux 6 GA (x86_64). No errata included.	6010
Oracle Linux 6 GA Patches (x86_64)	ol6_ga_x86_64_patch	Updated packages published after release of Oracle Linux 6 (x86_64)	724
Oracle Linux 6 Dtrace Userspace Tools (x86_64) - Latest	ol6_x86_64_Dtrace_userspace_latest	The latest Dtrace userspace tools for Oracle Linux 6 (x86_64).	2
Oracle Linux 6 Update 1 installation media copy (x86_64)	ol6_u1_x86_64_base	All packages released on the Oracle Linux 6 Update 1 (x86_64) installation media. This channel does not contain updates.	6148
Oracle Linux 6 Update 1 Patch (x86_64)	ol6_u1_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 1 (x86_64).	631
Oracle Linux 6 Update 2 installation media copy (x86_64)	ol6_u2_x86_64_base	All packages released on the Oracle Linux 6 Update 2 (x86_64) installation media. This channel does not contain updates.	6279
Oracle Linux 6 Update 2 Patch (x86_64)	ol6_u2_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 2 (x86_64).	787
Oracle Linux 6 Update 3 installation media copy (x86_64)	ol6_u3_x86_64_base	All packages released on the Oracle Linux 6 Update 3 (x86_64) installation media. This channel does not contain updates.	6324
Oracle Linux 6 Update 3 Patch (x86_64)	ol6_u3_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 3 (x86_64)	743
Oracle Linux 6 Update 4 installation media copy (x86_64)	ol6_u4_x86_64_base	All packages released on the Oracle Linux 6 Update 4 (x86_64) installation media. This channel does not contain updates.	6245
Oracle Linux 6 Update 4 Patch (x86_64)	ol6_u4_x86_64_patch	Updated packages published after release of Oracle Linux 6 Update 4 (x86_64)	971
Oracle Linux 6 Update 5 installation media	[illegible]	All packages released on the Oracle Linux 6 Update 5 (x86_64) installation media. This	[illegible]

There are more than 200 unique channels supported by ULN. The architectures currently supported are i386, x86_64, and ia64 (starting with Oracle Linux 4 update 6 and Oracle Linux 5 update 4). When you register a system, you are automatically subscribed to the channel containing the latest software for the architecture and OS revision of your system.

You can also choose a specific OS revision that you would like your system to remain at. Subscribe to the channel corresponding to the architecture of your system and the update level desired. Specific revisions of Oracle Linux have patches and errata issued, but you are not forced to upgrade from a given revision level to the next to get these fixes. Channels also exist for Oracle VM, OCFS2, RDS, and productivity applications. The installer determines which architecture to run.

`_base` Channels

The `_base` channels (also known as installation media channels) provide RPMs for each major version and subsequent minor updates of Oracle Linux as released on their respective installation media (DVD or ISO). For example, there is a `_base` channel for Oracle Linux 6 Update 3 as well as Oracle Linux 6. Security errata and bug fixes are not published to these channels.

_patch Channels

The _patch channels provide only the packages that have changed since the initial release of a particular version (whether a minor update or a major version). If multiple releases are created for the same package, due to multiple vulnerabilities found at different times, these channels always provide the most recent version of such a package.

_latest Channels

The _latest channels provide RPMs for all the packages in the distribution, including those errata also provided in the _patch channels (that is, the version of any RPM downloadable on the _latest channels is always the most recent available). For some RPMs, this corresponds to the same version distributed initially with the original distribution (if no vulnerabilities have been found to date). For others, the version is the same as was provided in the _patch channel for the highest update level. For example, the `ol6_<arch>_latest` channel for Oracle Linux 6 contains the combination of the `ol6_u3_<arch>_base` and `ol6_u3_<arch>_patch` channels.

_addons Channels

The _addons channels provide RPMs not included in the base distribution, such as RPMs to be used in creating a `yum` repository for Oracle Linux 6.

_oracle Channels

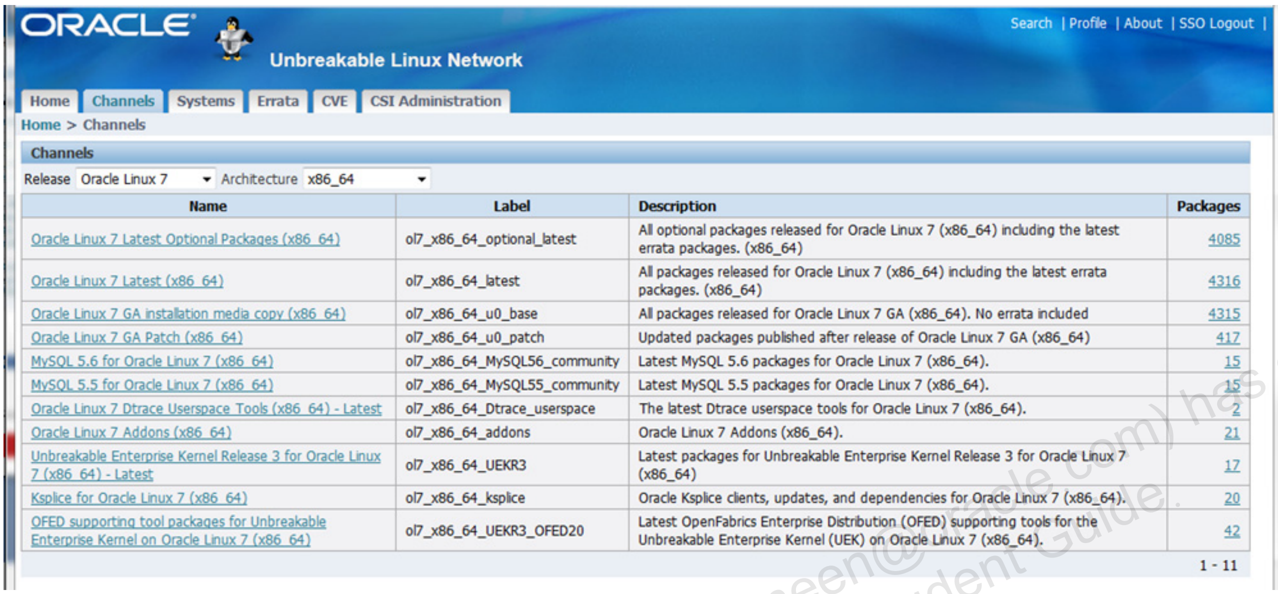
The _oracle channels provide distribution for Oracle freely downloadable software (in RPM format) that runs on Linux (for instance, Oracle Instant Client and `asmlib`).

New Channels

As new major releases and new minor updates of Oracle Linux become available, new channels are created by Oracle, to distribute the new RPMs. That is, the current `ol6_u1_<arch>_base` and `ol6_u1_<arch>_patch` channels remain available and do not include the new updates, making it therefore possible for ULN subscribers to remain on a specific release level of Oracle Linux and selectively apply errata on top of that. Every time a new minor update is released, two new channels (_base and _patch) are created for each architecture. The `ol6_<arch>_latest` channels continue to distribute the highest possible version of any package, and therefore follow the “head” of the development tree, independent of the update level. A similar philosophy is followed with the channels for major versions of Oracle Linux.

If you prefer to remain at a certain update level, but are currently subscribed, for example, to the `ol6_<arch>_latest` channel, you must subscribe to the `ol6_u<number>_<arch>_patch` and `ol6_u<number>_<arch>_base` channels for the desired update level and architecture and then unsubscribe from the _latest channel. This can be done through the web interface.

Oracle Linux 7 x86_64 Channels on ULN



Name	Label	Description	Packages
Oracle Linux 7 Latest Optional Packages (x86_64)	ol7_x86_64_optional_latest	All optional packages released for Oracle Linux 7 (x86_64) including the latest errata packages. (x86_64)	4085
Oracle Linux 7 Latest (x86_64)	ol7_x86_64_latest	All packages released for Oracle Linux 7 (x86_64) including the latest errata packages. (x86_64)	4316
Oracle Linux 7 GA installation media copy (x86_64)	ol7_x86_64_u0_base	All packages released for Oracle Linux 7 GA (x86_64). No errata included	4315
Oracle Linux 7 GA Patch (x86_64)	ol7_x86_64_u0_patch	Updated packages published after release of Oracle Linux 7 GA (x86_64)	417
MySQL 5.6 for Oracle Linux 7 (x86_64)	ol7_x86_64_MySQL56_community	Latest MySQL 5.6 packages for Oracle Linux 7 (x86_64).	15
MySQL 5.5 for Oracle Linux 7 (x86_64)	ol7_x86_64_MySQL55_community	Latest MySQL 5.5 packages for Oracle Linux 7 (x86_64).	15
Oracle Linux 7 Dtrace Userspace Tools (x86_64) - Latest	ol7_x86_64_Dtrace_userspace	The latest Dtrace userspace tools for Oracle Linux 7 (x86_64).	2
Oracle Linux 7 Addons (x86_64)	ol7_x86_64_addons	Oracle Linux 7 Addons (x86_64).	21
Unbreakable Enterprise Kernel Release 3 for Oracle Linux 7 (x86_64) - Latest	ol7_x86_64_UEKR3	Latest packages for Unbreakable Enterprise Kernel Release 3 for Oracle Linux 7 (x86_64)	17
Ksplice for Oracle Linux 7 (x86_64)	ol7_x86_64_ksplice	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 7 (x86_64).	20
OFED supporting tool packages for Unbreakable Enterprise Kernel on Oracle Linux 7 (x86_64)	ol7_x86_64_UEKR3_OFED20	Latest OpenFabrics Enterprise Distribution (OFED) supporting tools for the Unbreakable Enterprise Kernel (UEK) on Oracle Linux 7 (x86_64).	42

1 - 11

The slide shows the channels available for Oracle Linux 7 x86_64 architecture on ULN. In addition to the `_base`, `_patch`, `_latest`, and `_addons` channels, the following channels are available:

- **`ol7_x86_64_optional_latest`**: Optional packages include the latest errata.
- **`ol7_x86_64_MySQL56_community`**: Latest MySQL 5.6 packages for OL7
- **`ol7_x86_64_MySQL55_community`**: Latest MySQL 5.5 packages for OL7
- **`ol7_x86_64_Dtrace_userspace`**: Latest Dtrace userspace tools for OL7
- **`ol7_x86_64_UEKR3`**: Latest packages for UEK R3 for OL7
- **`ol7_x86_64_ksplice`**: Oracle Ksplice clients, updates, and dependencies for OL7
- **`ol7_x86_64_UEKR3_OFED20`**: Latest OFED tools for the UEK on OL7

DTrace stands for Dynamic Tracing. OFED stands for OpenFabrics Enterprise Distribution.

Switching from RHN to ULN

- RHEL uses the `rhn_register` program to register a system with Red Hat Network (RHN).
- Oracle has a `uln_register` program to switch to ULN if you are running RHEL.
- Do the following:
  1. Download the following files from https://linux-update.oracle.com/rpms:
     - `uln_register.tgz`
     - `uln_register-gnome.tgz`
  2. Extract the packages.
  3. Install the packages.
  4. Create a ULN account at https://linux.oracle.com/register.
  5. Register your system with ULN by running `uln_register`.

Red Hat Enterprise Linux has an application called `rhn_register` to register a system with Red Hat Network (RHN). This application normally runs as part of the FirstBoot program on a new installation of RHEL. A system can be registered with RHN at a later time by running `rhn_register`.

Oracle makes it easy to switch from RHN to Unbreakable Linux Network (ULN). Oracle's `uln_register` program allows you to switch to ULN if you are running Red Hat Enterprise Linux. Details are available at https://linux.oracle.com/switch.html.

1. Download the Registration Packages

The packages that are required to register your system to ULN for i386 and x86_64 architectures are available from http://linux-update.oracle.com/rpms. The file names to download are:

- `uln_register.tgz`
- `uln_register-gnome.tgz`

2. Extract the Registration Packages

Extract the `.tgz` files on RHEL by using the following commands:

```
# tar -xzf uln_register.tgz
# tar -xzf uln_register-gnome.tgz (only if rhn-setup-gnome is
already installed)
```

3. Install the Registration Packages

After extracting the files, change to the `uln_migrate` directory and install the registration packages as follows:

```
# cd uln_migrate
# rpm -Uvh *.rpm
```

4. Create a ULN Account

Before you can register a server, you must first create a ULN account. You can create a ULN account at http://linux.oracle.com/register.

5. Register Your System with ULN

To register your system, run the following command as the `root` user in a terminal window or on the command line:

```
# uln_register
```

Follow the instructions on the screen and provide the requested information. `uln_register` also collects machine information and uploads it to Oracle's server.

Quiz

Yum repositories can be both local and remote.

a. True

b. False

Quiz

Errata are available only by subscribing to ULN.

a. True
b. False

Summary

In this lesson, you should have learned how to:

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

Practice 6: Overview

The practices for this lesson cover the following topics:

- Using the `rpm` utility
- Accessing the Oracle Public Yum Server
- Creating a local Yum repository
- Using the `yum` utility
- Using the Unbreakable Linux Network

7

Ksplice

Objectives

After completing this lesson, you should be able to:

- Describe the purpose of Ksplice
- Describe how Ksplice works
- Describe Ksplice implementation steps
- View Ksplice packages on ULN
- Use Ksplice Uptrack commands
- Use the Ksplice web interface
- Use Ksplice Offline Client

Introduction to Ksplice

Ksplice:

- Updates the kernel on a running system
- Applies the latest kernel security errata (CVEs)
  – Patches are effective immediately without rebooting.
- Does not halt the system
- Does not restart applications
- Applies updates in the background
- Requires Oracle Premier support subscription
- Works with both Unbreakable Enterprise Kernel and Red Hat Compatible Kernel
- Easy-to-use website

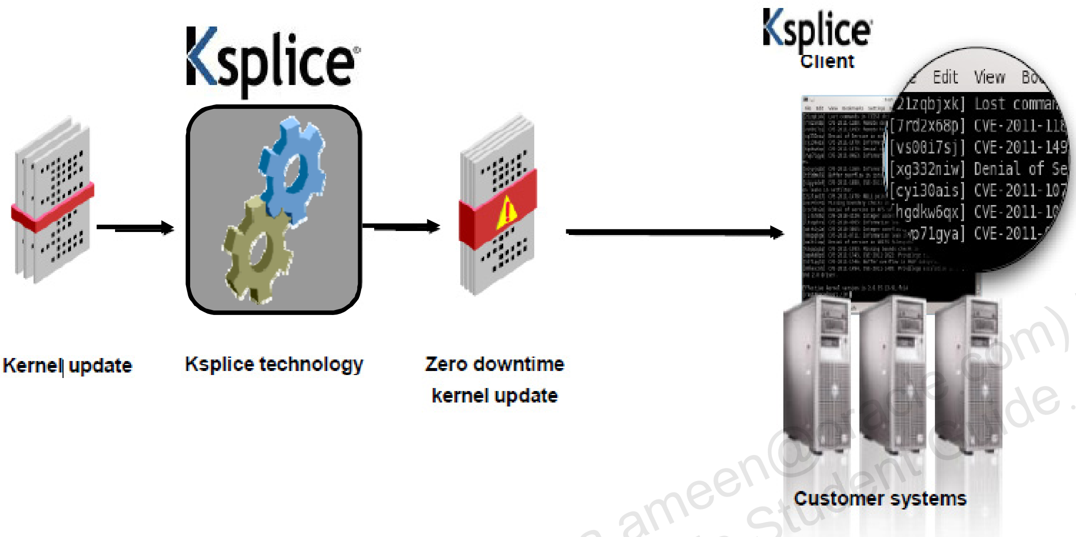
Oracle Linux customers with a Premier support subscription have access to Oracle Ksplice technology. Ksplice updates are available for systems running Oracle Linux 5 or later with either the Unbreakable Enterprise Kernel or the Red Hat Compatible Kernel.

Oracle Ksplice updates are kernel updates that can be applied on a running system. The Ksplice patches are applied to the running Linux kernel and are effective immediately. Oracle Ksplice patches only the running kernel. On subsequent reboot, these patches are applied at boot time.

This technology allows you to apply the latest kernel security errata (CVEs: Common Vulnerabilities and Exposures) without rebooting. It does not halt the system or restart applications; the updates are just applied in the background with a negligible impact, and usually only a millisecond pause.

The Ksplice Uptrack website has an easy-to-use interface that lets you view registered systems, patches installed, available patches, and statuses of systems. It also lets you create access control groups. For each server, the website shows available updates that have not yet been applied. Each update lists a one-line description of the update. Further down the website is a list of the installed updates on your running kernel.

How Ksplice Works



A kernel update comes from either Oracle or from the kernel community. The Ksplice team takes the update and works it into a binary patch that is inserted into a running kernel. You apply the patch by using the Ksplice tools, and the patch is up and running immediately.

Because you do not need to reboot or bring your system down, you can apply security updates as they become available without having to wait for your users to tell you that it is okay to take down the system. With Ksplice, you can keep your systems secure without jeopardizing high availability.

Ksplice Implementation

A summary of actions to get started with Ksplice:

1. Register your system(s) with Unbreakable Linux Network (ULN).
2. Subscribe to the appropriate Ksplice channel.
3. Use the `yum` command to install the `uptrack` package.
4. Perform any required configuration.
5. View status from the System Status page of the Ksplice web interface at https://status-ksplice.oracle.com.

After you register your system with ULN and install the `uptrack` package, you receive an email containing instructions for logging on to the Ksplice web interface.

The Ksplice web interface can also be accessed from: https://uptrack.ksplice.com.

The following is a summary of actions to get started with Ksplice:

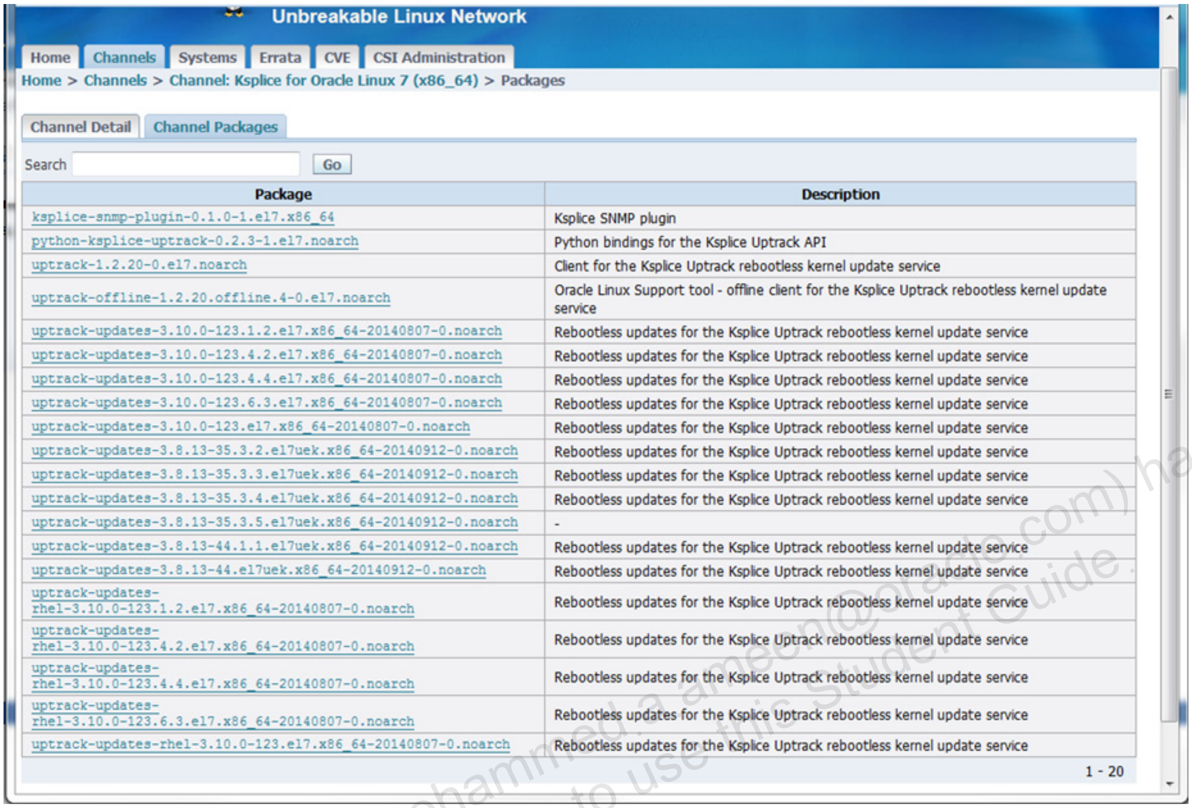
1. Register your system(s) with ULN.
2. Subscribe to the appropriate Ksplice channel.

ULN channels that are available for Ksplice on Oracle Linux:

- **`ol5_i386_ksplice`**: Ksplice for Oracle Linux 5 (i386)
- **`ol5_x86_64_ksplice`**: Ksplice for Oracle Linux 5 (x86_64)
- **`ol6_i386_ksplice`**: Ksplice for Oracle Linux 6 (i386)
- **`ol6_x86_64_ksplice`**: Ksplice for Oracle Linux 6 (x86_64)
- **`ol7_x86_64_ksplice`**: Ksplice for Oracle Linux 7 (x86_64)

3. Use the `yum` command to install the `uptrack` package.
4. Perform any required configuration.
5. View status from the System Status page of the Ksplice web interface at https://status-ksplice.oracle.com.

Ksplice Packages on ULN


Unbreakable Linux Network
Home | Channels | Systems | Errata | CVE | CSI Administration
Home > Channels > Channel: Ksplice for Oracle Linux 7 (x86_64) > Packages
Channel Detail | Channel Packages
Search Go

Package	Description
ksplice-snmp-plugin-0.1.0-1.el7.x86_64	Ksplice SNMP plugin
python-ksplice-uptrack-0.2.3-1.el7.noarch	Python bindings for the Ksplice Uptrack API
uptrack-1.2.20-0.el7.noarch	Client for the Ksplice Uptrack rebootless kernel update service
uptrack-offline-1.2.20.offline.4-0.el7.noarch	Oracle Linux Support tool - offline client for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.10.0-123.1.2.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.10.0-123.4.2.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.10.0-123.4.4.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.10.0-123.6.3.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.10.0-123.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.8.13-35.3.2.el7uek.x86_64-20140912-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.8.13-35.3.3.el7uek.x86_64-20140912-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.8.13-35.3.4.el7uek.x86_64-20140912-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.8.13-35.3.5.el7uek.x86_64-20140912-0.noarch	-
uptrack-updates-3.8.13-44.1.1.el7uek.x86_64-20140912-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-3.8.13-44.el7uek.x86_64-20140912-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-rhel-3.10.0-123.1.2.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-rhel-3.10.0-123.4.2.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-rhel-3.10.0-123.4.4.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-rhel-3.10.0-123.6.3.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service
uptrack-updates-rhel-3.10.0-123.el7.x86_64-20140807-0.noarch	Rebootless updates for the Ksplice Uptrack rebootless kernel update service

1 - 20

This slide shows packages available from the "Ksplice for Oracle Linux 7 (x86_64)" channel on ULN. Use the `yum` command to install the `uptrack` package:

```
# yum install uptrack
```

After the installation is complete, the tool automatically registers your system with the Uptrack service and checks for any available Oracle Ksplice updates for your running kernel. If new versions are available, the Uptrack tools provide you with the list of updates.

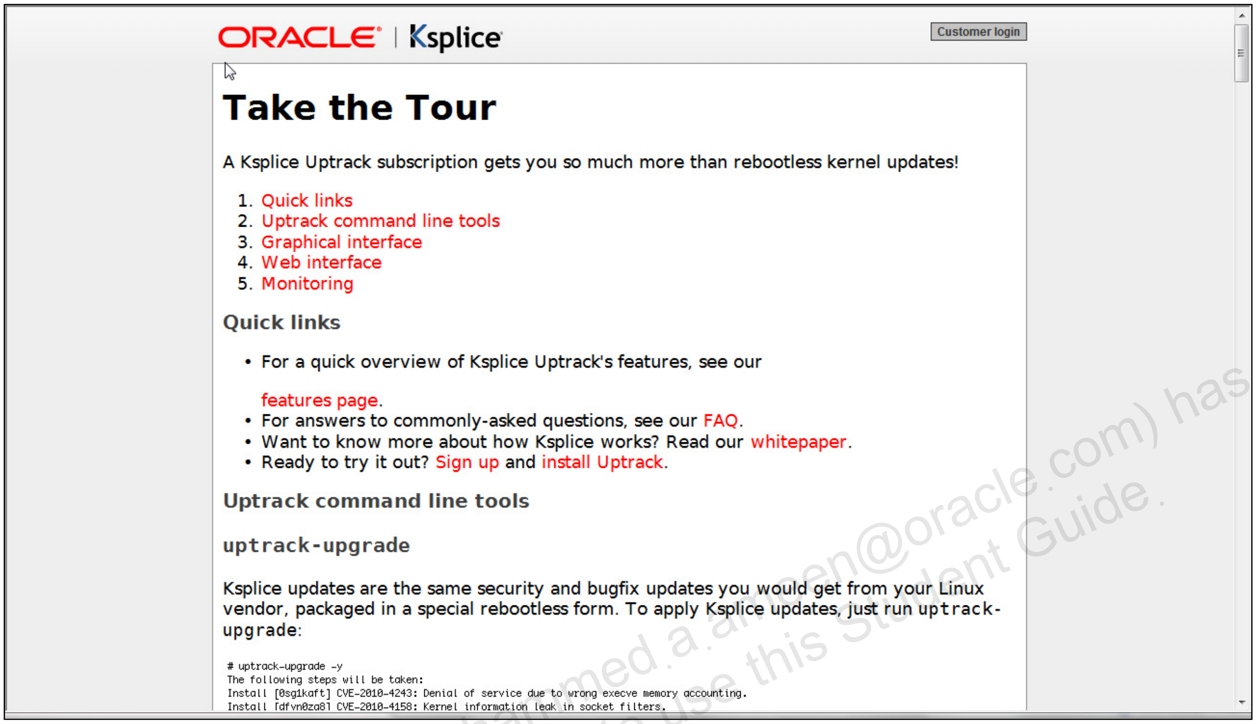
Your server must have access to the Internet. If a proxy is used, set the proxy in your shell before running the Uptrack commands. The commands to set the proxy are:

```
# export http_proxy=http://proxy.company.com port
# export https_proxy=http://proxy.company.com:port
```

The Uptrack configuration file location is `/etc/uptrack/uptrack.conf`. Modify this file to configure a proxy server, automatically install updates at boot time, or automatically check for new updates and apply them at the same time.

Oracle Ksplice patches are stored locally in `/var/cache/uptrack` and, by default, are automatically re-applied after a reboot (very early in the boot process). It is recommended that you also install the regular kernel RPM packages for released errata. This enables you to boot into a newer kernel version when you have a restart of the operating system. At that point, the Oracle Ksplice patches are applied starting from this new kernel as a baseline.

Using Ksplice Uptrack



This screen shows the "Take the Tour" page at https://ksplice.oracle.com/uptrack/using. It lists all the Ksplice Uptrack commands and provides examples and sample output. It describes how to configure your systems for automatic updates. It also describes access policies, how to configure email notifications, how to use the Uptrack API to monitor and control your Ksplice Uptrack account, and other features of Ksplice.

There is also a Ksplice User's Guide available at http://docs.oracle.com/cd/E37670_01/E39380/html/.

Ksplice Uptrack Command Summary

- `uptrack-upgrade`: Download and apply new updates.
- `uptrack-show`: List the active updates in your running kernel.
- `uptrack-remove`: Remove applied updates from the running system and return to the original kernel version.
- `uptrack-uname`: Display the effective kernel version based on active Oracle Ksplice updates.
- `uptrack-install`: This lets you install a specific update.

ORACLE

The following provides a summary of the Ksplice Uptrack Commands:

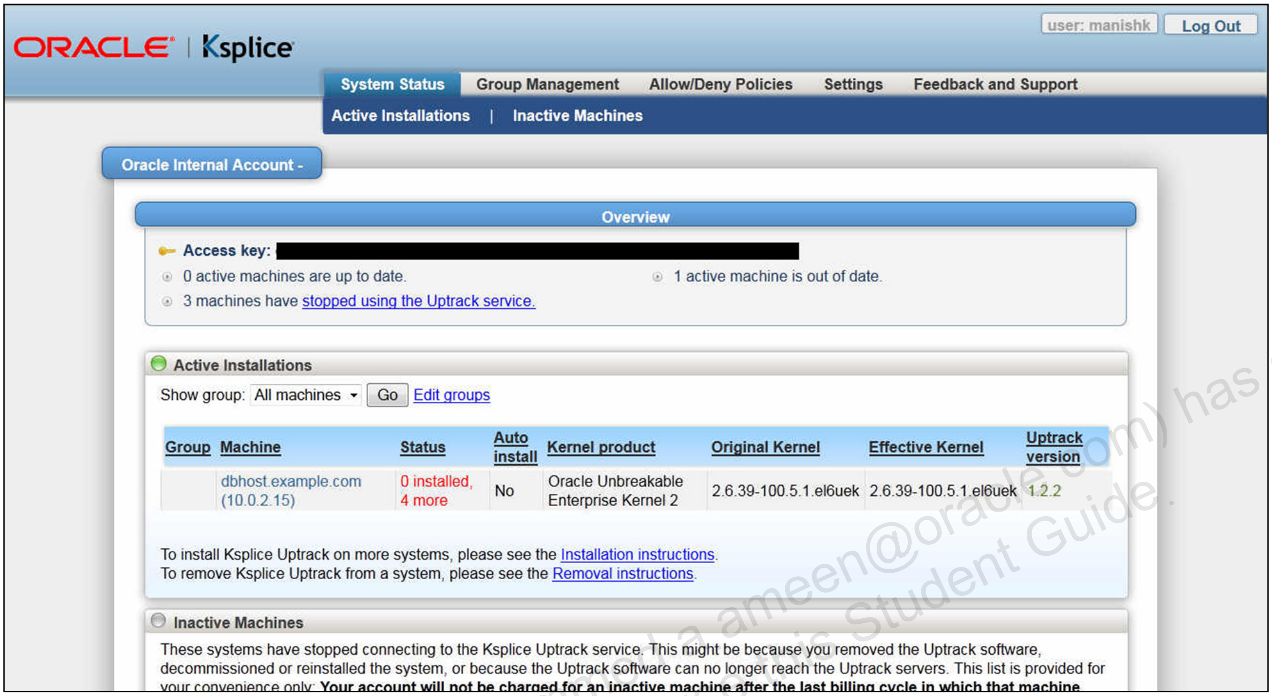
- **`uptrack-upgrade`:** Connect to the Uptrack update server and download new updates when available. These updates can be applied immediately as well.
- **`uptrack-show`:** List the active Oracle Ksplice updates in your running kernel.
- **`uptrack-remove`:** Remove applied updates from the running system and return to the original kernel version and state.
- **`uptrack-uname`:** The modified version of `uname` that knows how to read the effective kernel version based on active Oracle Ksplice updates
- **`uptrack-install`:** Install a specific update (if you do not want them all). This command is the opposite of `uptrack-remove`. The `uptrack-remove [id]` command takes an ID of the update to remove, and removes it. The `uptrack-install [id]` command takes the ID of an update to install, and installs it.

To remove the `uptrack` package, enter:

```
# yum remove uptrack
```

You can find the uninstall instructions on the Uptrack website as well.

System Status



This screen shows the System Status page of the Ksplice web interface at https://status-ksplice.oracle.com. All of your Ksplice-enabled systems are listed on this page. Systems listed under Active Installations are those systems that are connected to the Ksplice servers. Inactive Machines listed at the bottom of the page are those that have stopped using Ksplice Uptrack services.

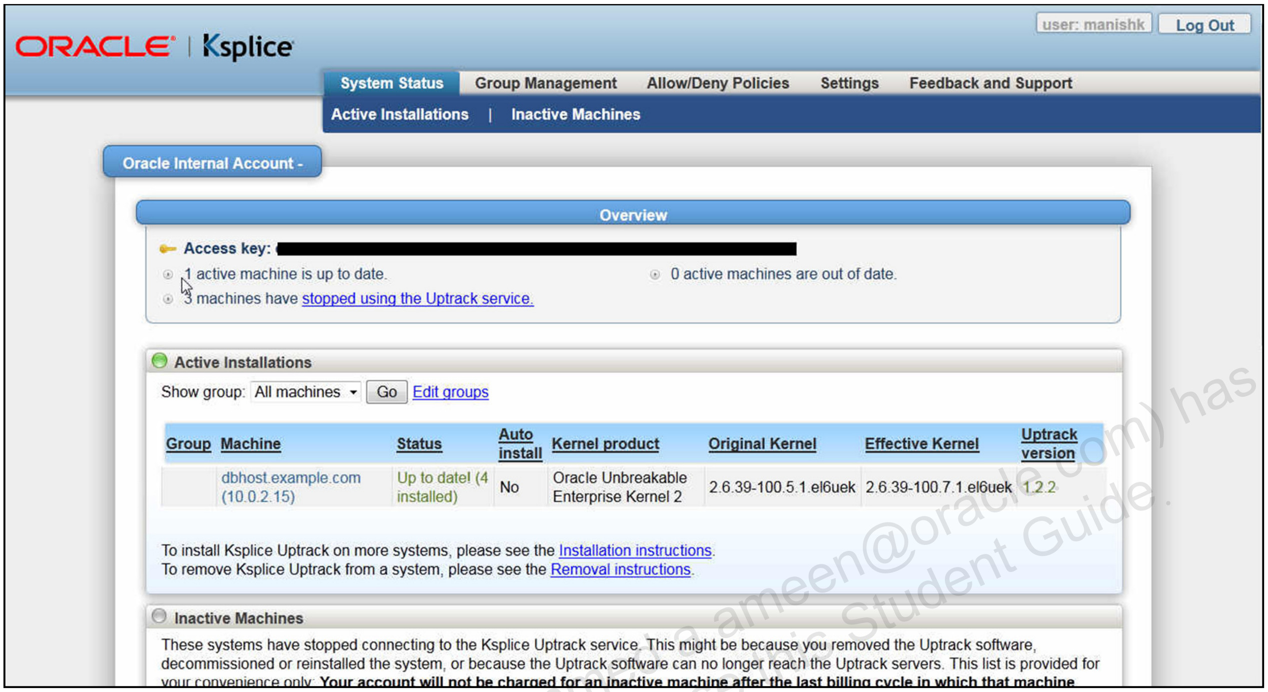
The Machine column lists the individual systems. Click the machine name to display detailed information on available updates, installed updates, and additional information.

The Status in the example in the slide shows there are four updates available. Run the `uptrack-upgrade` command to apply the updates.

Auto Install is set to No. This is a configuration parameter, `autoinstall`, in the `/etc/uptrack/uptrack.conf` file. Setting this directive to `yes` automatically downloads and installs Ksplice updates by using `cron`.

The Original Kernel and Effective Kernel are the same because no updates have been installed. The last column gives the Ksplice Uptrack version.

System Updated



This screen shows the System Status page of the Ksplice web interface after running the `uptrack-upgrade` command. The Effective Kernel is now at a different level than the Original Kernel. The Effective Kernel is shown using the `uptrack-uname` command. The original kernel version is shown using the `uname` command.

If the `install_on_reboot` parameter in the `/etc/uptrack/uptrack.conf` file is set to `yes`, the updates that were installed by using Ksplice are re-installed after reboot by the `uptrack` service.

The Original Kernel and Effective Kernel only become the same if you installed the new kernel on disk by using `yum` and rebooted. If you reboot without updating the kernel on disk by using `yum`, you boot back into the old kernel and Ksplice applies its updates automatically at boot-time, bringing you back to the same Original and Effective Kernel versions you had before you rebooted (in this example, `2.6.39-100.5.1.el6uek` and `2.6.39-100.7.1.el6uek`).

Ksplice Offline Client

- Systems running the Ksplice Offline Client do not need a network connection to the Oracle Uptrack server.
- Ksplice updates for each supported kernel version are bundled into an RPM and made available from ULN.
  - Ksplice updates for Oracle Linux 7 x86_64 systems are available on the `ol7_x86_64_ksplice` ULN channel.
- Download the RPM from ULN to a memory stick and then use the `rpm` command to install the update package.
- Alternatively, create a Local Yum Server that acts as a Ksplice mirror.
  - Local Yum Server requires network connection to ULN.
  - Ksplice Offline Clients require access only to the Local Yum Server.

Ksplice Offline Client removes the requirement for a server to have a direct network connection to the Oracle Uptrack server. All available Ksplice updates for each supported kernel version are bundled into an RPM and made available from ULN. For example, the Ksplice updates for Oracle Linux 7 x86_64 systems are available on the `ol7_x86_64_ksplice` ULN channel. The RPM for each kernel version at this channel is updated when a new Ksplice patch becomes available. You can download the RPM from ULN to a memory stick, for example, and then use the `rpm` command to install the update package.

You can also create a Local Yum Server that acts as a mirror of Ksplice for Oracle Linux channels on ULN. At regular intervals, download the latest Ksplice update packages from ULN to this Local Yum Server. Only the Local Yum Server requires access to the Oracle Uptrack server. After installing Ksplice Offline Client on your other systems, these systems need only to be able to connect to the Local Yum Server.

Systems that are running Ksplice Offline Client are not registered with https://status-ksplice.oracle.com; therefore, you cannot use the Ksplice web interface or the Ksplice Uptrack API on these unregistered systems.

Modifying a Local Yum Server to Act as a Ksplice Mirror

- Local Yum Server must be registered with ULN.
- Perform the configuration steps as follows:
  1. Log in to ULN at http://linux.oracle.com.
  2. From the ULN Systems tab, click the link for your system in the list of registered machines.
  3. On the System Details page, click Edit.
  4. On the Edit System Properties page, verify that the Yum Server check box is selected and click Apply Changes.
  5. On the System Details page, click Manage Subscriptions.
  6. On the System Summary page, select needed channels.
  7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.

If you have a system that is registered with ULN that is also configured as a Local Yum Server, you can modify the system to act as a Ksplice mirror. Perform the following steps from the Local Yum Server:

1. Log in to ULN at http://linux.oracle.com with the ULN username and password that you used to register the system.
2. From the ULN Systems tab, click the link for your system in the list of registered machines.
3. On the System Details page, click Edit.
4. On the Edit System Properties page, verify that the Yum Server check box is selected and click Apply Changes.
5. On the System Details page, click Manage Subscriptions.
6. On the System Summary page, select channels from the list of available or subscribed channels and click the arrows to move the channels between the lists.
7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.

For more information on available release channels, see http://www.oracle.com/technetwork/articles/servers-storage-admin/yum-repo-setup-1659167.html.

Updating a Local Yum Server with Ksplice Channels

- To update the Yum repositories for the registered channels, perform the following steps:
  1. Download the `167283.sh` script from http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/167283.sh.
  2. Edit the `167283.sh` script and set the value of the `REP_BASE` variable to the base directory for the repository.
  3. Run the `167283.sh` script to create the Yum repositories for the registered channels.
- It is recommended that you set up a `cron` job to run the `167283.sh` script on a regular basis.
- The `cron` facility is discussed in the lesson titled "Automating Tasks."

After subscribing to the Ksplice channel(s), perform the following steps to update the Yum repositories for the registered channels:

1. Download the `167283.sh` script from http://www.oracle.com/ocom/groups/public/@otn/documents/webcontent/167283.sh.
2. Edit the `167283.sh` script and set the value of the `REP_BASE` variable to the base directory for the repository, for example:

```
REP_BASE=/var/yum
```

3. Run the `167283.sh` script to create the Yum repositories for the registered channels.
   a. To download only binary RPMs, enter:

   ```
   sh 167283.sh
   ```

   b. To download both binary and source RPMs, enter:

   ```
   sh 167283.sh src
   ```

It is recommended that you set up a `cron` job to perform this task. For example, the following `crontab` entry for `root` runs the script twice per day at 6:00 AM and 6:00 PM:

```
0 6,18 * * * sh /var/downloads/yum/167283.sh
```

This example assumes that the `167283.sh` script is located in the `/var/downloads` directory. The `cron` facility is discussed in the lesson titled "Automating Tasks."

Configuring Ksplice Offline Clients to Use the Local Ksplice Mirror

- Perform the following steps to configure a system as a Ksplice Offline Client:
  1. In the /etc/yum.repos.d directory, edit the existing repository file and disable all entries by setting enabled=0.
  2. In the /etc/yum.repos.d directory, create the local-yum.repo file.
  3. Install the Ksplice Offline Client with the following command:
     ```
     # yum install uptrack-offline
     ```
  4. Install the Ksplice updates that are available for the kernel:
     ```
     # yum install uptrack-updates-`uname -r`
     ```
- The command given in step 4 installs the Ksplice updates and applies them. It is recommended that you set up a daily cron job to perform this task.

When you have set up a Local Yum Server that can act as a Ksplice mirror, you can configure your other systems to receive Ksplice updates. Perform the following steps to configure a system as a Ksplice Offline Client:

1. In the /etc/yum.repos.d directory, edit the existing repository file such as public-yumol6.repo and disable all entries by setting enabled=0.
2. In the /etc/yum.repos.d directory, create the local-yum.repo file, which contains entries such as the following for an Oracle Linux 7 client:
   ```
   [ol7_x86_64_ksplice]
   name=Ksplice for $releasever - $basearch
   baseurl=http://<IP_address_of_local_yum_server>/yum/OracleLinux
     /OL6/ksplice/$basearch/
   enabled=1
   ```
3. Install the Ksplice Offline Client:
   ```
   # yum install uptrack-offline
   ```
4. Install the Ksplice updates that are available for the kernel:
   ```
   # yum install uptrack-updates-`uname -r`
   ```

Quiz

Ksplice kernel patches are effective immediately without requiring a reboot.

a. True

b. False

Summary

In this lesson, you should have learned how to:

- Describe the purpose of Ksplice
- Describe how Ksplice works
- Describe Ksplice implementation steps
- View Ksplice packages on ULN
- Use Ksplice Uptrack commands
- Use the Ksplice web interface
- Use Ksplice Offline Client

Practice 7: Overview

The practices for this lesson cover the following:

- Using Ksplice Uptrack
- Installing the Ksplice Offline Client and kernel updates

8

Automating Tasks

Objectives

After completing this lesson, you should be able to:

- Describe available automated tasks utilities
- Configure `cron` jobs
- Describe `cron` directories and files
- Use the `crontab` utility
- Configure `anacron` jobs
- Use the `at` and `batch` utilities

Automating System Tasks

- Oracle Linux can run tasks automatically, and comes with automated tasks utilities: `cron`, `anacron`, `at`, `batch`.
- `cron` jobs can run as often as every minute.
  - A scheduled `cron` job is skipped if the system is down.
- `anacron` can run a job only once a day.
  - Scheduled jobs are remembered and run the next time that the system is up.
- `crond` searches multiple files and directories for scheduled jobs:
  - `/var/spool/cron/`
  - `/etc/anacrontab`
  - `/etc/cron.d`

Oracle Linux can be configured to automatically run tasks (or jobs) within a specified period of time, on a specified date, or when the system load average is below a specified number. You can use automated tasks to perform periodic backups, monitor the system, run custom scripts, and more. Oracle Linux comes with several automated tasks utilities.

`cron` and `anacron`

Both `cron` and `anacron` are used to schedule the execution of recurring tasks according to a combination of the time, day of the month, month, day of the week, and week. `cron` allows jobs to be run as often as every minute. However, if the system is down when a `cron` job is scheduled, the job is not executed. `anacron` can run a job only once a day, but `anacron` remembers scheduled jobs and runs them the next time the system is up. The main purpose of `anacron` is to run `cron` jobs that did not run because the system was down. It is most useful on laptop computers to run the daily or weekly `cron` jobs after the system is booted with a particular delay to avoid overloading the system.

`crond` Daemon

The `crond` daemon executes scheduled tasks. It searches `/var/spool/cron` for `crontab` files for individual users, `/etc/anacrontab`, and the files in the `/etc/cron.d` directory. It checks each command to see whether it should be run in the current minute. When a task is scheduled for execution, `crond` executes it as the user who owns the file describing the task.

Configuring `cron` Jobs

- `cron` jobs are defined in `/etc/crontab`.
- Jobs are defined by:
  - `minute`: From 0 to 59
  - `hour`: From 0 to 23
  - `day`: From 1 to 31
  - `month`: From 1 to 12, or short name of month
  - `day-of-week`: From 0 to 7, or short name of day
  - `user`: User under which the jobs are run
  - `command`: Shell command or script to execute
- Special characters can be used:
  - Asterisk, hyphen, comma, and forward slash

`cron` jobs are defined in the `/etc/crontab` configuration file. This is the systemwide `crontab`. Users can also have `cron` jobs. `cron` jobs are specified in the following format:

```
minute   hour   day   month   day-of-week   user   command
```

- **`minute`:** From 0 to 59
- **`hour`:** From 0 to 23
- **`day`:** From 1 to 31 (must be a valid day if a month is specified)
- **`month`:** From 1 to 12 (or the short name of the month such as Jan or Feb)
- **`day-of-week`:** From 0 to 7, where 0 or 7 represents Sunday (or the short name of the week such as Sun or Mon)
- **`user`:** The user under which the jobs are run
- **`command`:** The command to execute; can be a shell command or script

Other special characters can be used:

- An asterisk (`*`) can be used to specify all valid values.
- A hyphen (`-`) between integers specifies a range of integers.
- A list of values separated by commas (`,`) specifies a list.
- A forward slash (`/`) can be used to specify step values.

`cron` Examples

Examples for specifying a `cron` job to run by minutes:

- `*/5 * * * * command`: Run command every five minutes
- `*/10 * * * * command`: Run command every 10 minutes
- `*/15 * * * * command`: Run command every 15 minutes
- `0-59/2 * * * * command`: Run command every other minute

Examples for specifying a `cron` job to run by hours:

- `0 */2  * * * command`: Run command every two hours
- `0 */3  * * * command`: Run command every three hours
- `0 */4  * * * command`: Run command every four hours
- `0 */5  * * * command`: Run command every five hours

Examples for specifying a `cron` job to run by days:

- `0 0 * * 5 command`: Run command every Friday at midnight
- `0 0 * * 6 command`: Run command every Saturday at midnight

Examples for specifying a `cron` job to run by months:

- `0 0 1 5,10 * command`: Run command first of May at midnight and first of October at midnight
- `0 0 1 */3 * command`: Run command every third month at midnight

Other `cron` Directories and Files

- `/etc/cron.d`
  - Contains files with same syntax as the `/etc/crontab`
  - `root` privileges only
- Other `cron` directories in `/etc`:
  - `cron.hourly`
  - `cron.daily`
  - `cron.weekly`
  - `cron.monthly`
- Scripts in these directories run hourly, daily, weekly, or monthly, depending on the name of the directory.
- The `/etc/cron.allow` and `/etc/cron.deny` files restrict user access to `cron`.

The `/etc/cron.d` directory contains files that have the same syntax as the `/etc/crontab` file. Only `root` has permission to create and modify files in this directory.

```
# ls /etc/cron.d
0hourly  raid-check  sysstat  unbound-anchor  uptrack
# cat /etc/cron.d/0hourly
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
01 * * * * root run-parts /etc/cron.hourly
```

Also in `/etc` are `cron` directories named:

- `cron.hourly`
- `cron.daily`
- `cron.weekly`
- `cron.monthly`

Scripts in these directories run hourly, daily, weekly, or monthly, depending on the name of the directory. Create entries in the `/etc/anacrontab` file to schedule the execution of these scripts. Example:

```
# cat /etc/anacrontab
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs are started during the following hours only
START_HOURS_RANGE=3-22
#period in days  delay in minutes  job-identifier   command
1        5      cron.daily        nice run-parts /etc/cron.daily
7        25     cron.weekly       nice run-parts /etc/cron.weekly
@monthly 45     cron.monthly      nice run-parts /etc/cron.monthly
```

Controlling Access to `cron`

The `/etc/cron.allow` and `/etc/cron.deny` files are used to restrict access to `cron`. The access control files are checked each time a user tries to add or delete a `cron` job. If the `cron.allow` file exists, only users listed in it are allowed to use `cron`, and the `cron.deny` file is ignored. If the `cron.allow` file does not exist, users listed in `cron.deny` are not allowed to use `cron`. If neither file exists, only `root` can use `cron`. The format of both access control files is one username on each line.

`crontab` Utility

- The `crontab` utility allows users other than `root` to configure `cron` tasks.
- User-defined `crontabs` are stored in:
  – `/var/spool/cron/`*`<user>`*
- To create or edit a `crontab`:

```
# crontab -e
```

  – Use the same format as `/etc/crontab` without specifying a user.
- To list the contents of a user-defined `crontab`:

```
# crontab -l
```

ORACLE

Users other than `root` configure `cron` tasks by using the `crontab` utility. All user-defined `crontabs` are stored in the `/var/spool/cron/` directory and are executed by using the usernames of the users that created them.

To create or edit a `crontab` as a user, log in as that user and enter the `crontab -e` command. The file uses the same format as `/etc/crontab` with one exception, do not specify a user. When the changes to the `crontab` are saved, the `crontab` is stored according to username and written to the `/var/spool/cron/`*`<username>`* file. To list the contents of your own personal `crontab` file, use the `crontab -l` command.

The following set of commands illustrates the usage of the `crontab` utility (as `root`):

```
# crontab -e
```

Create an entry to execute a shell script named “full-backup” on June 10 at 8:30 AM.

```
30 08 10 06 * /full-backup
# ls /var/spool/cron
root
# crontab -l
30 08 10 06 * /full-backup
```

Configuring `anacron` Jobs

- `anacron` jobs are defined in `/etc/anacrontab`.
- Jobs are defined by:
  - Period in days: The frequency of execution in days
  - Delay in minutes: The minutes to wait before executing the job
  - Job-identifier: A unique name used in log files
  - Command: A shell command or script to execute

ORACLE

`anacron` jobs are defined in the `/etc/anacrontab` configuration file. The following is an example of the `/etc/anacrontab` file:

```
# cat /etc/anacrontab
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=30
# the jobs are started during the following hours only
START_HOURS_RANGE=16-20
#period in days  delay in minutes  job-identifier   command
1         20     dailyjob          nice run-parts /etc/cron.daily
7         25     weeklyjob         /etc/weeklyjob.bash
@monthly 45      monthlyjob        ls /proc >> /tmp/proc
```

The first five lines are variables used to configure the environment in which the `anacron` tasks are run:

- **`SHELL`:** The shell environment to use
- **`PATH`:** The path used to execute commands
- **`MAILTO`:** The username to email output of the `anacron` jobs to
- **`RANDOM_DELAY`:** The maximum number of minutes to be added to the delay in the minutes variable specified for each job (the minimum delay defaults to 6 minutes)
- **`START_HOURS_RANGE`**: An interval when scheduled jobs can be run

The remaining lines in the `/etc/anacrontab` file represent scheduled jobs:

- **`period in days`:** The frequency of execution of a job in days. This can be a macro. (`@daily` = 1, `@weekly` = 7, `@monthly` = once a month)
- **`delay in minutes`:** The number of minutes `anacron` waits, if necessary, before executing a job (`0` = no delay)
- **`job-identifier`:** A unique name of a job used in the log files
- **`command`:** A command to execute (can be a shell command or a script)

Jobs defined in this `anacrontab` file are randomly delayed by 6–30 minutes and can be executed between 16:00 and 20:00.

The first job runs anywhere between 16:26 and 16:50 every day. The command executes all programs in the `/etc/cron.daily` directory (using the `run-parts` script, which takes a directory as a command-line argument and sequentially executes every program within that directory).

The second job runs once a week and executes the `/etc/weeklyjob.bash` script.

The third job is executed once a month and runs the `ls /proc >> /tmp/proc` command.

`at` and `batch`

- `at` and `batch` are utilities for scheduling one-time tasks.
- The `at` command executes a task at a specific time.
- The `batch` command executes a task when system load average drops below 0.8.
- The `atd` service must be running to use `at` or `batch`.
- `at` command syntax:
  - `at` *`time`*
  - The *`time`* argument is the time to execute the command.
  - The *`time`* argument accepts multiple formats.
- `batch` command syntax:
  - `batch` (the `at>` prompt is displayed)

The `at` command is used to schedule a one-time task at a specific time. The `batch` command is used to schedule a one-time task to be executed when the system load average drops below 0.8. The `atd` service must be running to use `at` or `batch`.

Using `at`

To schedule a one-time job at a specific time, enter the command `at time`, where `time` is the time to execute the command. Use one of the following `time` arguments:

- `HH:MM`
- `midnight`: At 12:00 AM
- `noon`: At 12:00 PM
- `teatime`: At 4:00 PM
- `month-name day year`
- `MMDDYY`, `MM/DD/YY`, or `MM.DD.YY`
- `now + time`: `time` is in minutes, hours, days, or weeks (for example, now + 5 days).

After entering the `at` command with the time argument, the `at>` prompt is displayed. Type the command to execute, either a shell command or script, and press Enter. Multiple commands can be specified by typing each command followed by pressing Enter.

After typing all the commands, press Enter to go to a blank line and press Ctrl + D. You are emailed standard output and standard error from commands. Use the `atq` command to view pending jobs. Example:

```
# at now + 2 hours
at> /full-backup
at> <EOT>
Job 1 at 2011-10-12 16:52
# atq
1           2011-10-12 16:52 a root
```

Using `batch`

`batch` is similar to `at` except commands or scripts are not executed until the load average is below 0.8. Type `batch` and the `at>` prompt is displayed. Enter multiple commands or scripts, pressing Enter after each entry. Press Ctrl + D on a blank line to end.

Controlling Access to `at` and `batch`

The `/etc/at.allow` and `/etc/at.deny` files are used to restrict access to `at` and `batch`. The access control files are checked each time a user tries to use either command. Usage of these files is similar to the usage of the `cron.allow` and `cron.deny` files. The `root` user can always execute `at` and `batch` commands, regardless of the access control files. If the `at.allow` file exists, only users listed in it are allowed to use the commands and the `at.deny` file is ignored. If `at.allow` does not exist, users listed in `at.deny` cannot use the commands.

Quiz

Both `at` and `batch` are used to schedule the execution of recurring tasks.

a. True

b. False

Summary

In this lesson, you should have learned how to:

- Describe available automated tasks utilities
- Configure `cron` jobs
- Describe `cron` directories and files
- Use the `crontab` utility
- Configure `anacron` jobs
- Use `at` and `batch` utilities

Practice 8: Overview

The practices for this lesson cover automating tasks by using the `crontab` and `at` commands.

9

Kernel Module Configuration

Objectives

After completing this lesson, you should be able to:

- Describe loadable kernel modules
- Dynamically load and unload kernel modules
- Configure kernel module parameters

Loadable Kernel Modules (LKM)

- LKM extend the functionality of a running kernel.
- Kernel modules are dynamically loaded and unloaded.
- To list currently loaded kernel modules:

```
# lsmod
```

- To view details about a specific kernel module:

```
# modinfo <module_name>
```

- Kernel modules often have dependencies.

The Linux kernel is loaded into memory by the boot loader. New code can be added to the kernel by including the source files in the kernel source tree and recompiling the kernel. But Linux also supports LKM that allow you to add code to a running kernel.

Kernel modules are dynamically loaded and unloaded on demand. They provide device drivers to allow the kernel to access new hardware, support for different file system types, and generally extend the functionality of the kernel.

Listing the Loaded Kernel Modules

To list which kernel modules are currently loaded into the kernel, use the `lsmod` command. This command produces output by reading the `/proc/modules` file. Example:

```
# lsmod
Module                  Size    Used by
ksplice_cwy849xp      141175    1
rfkill                 18918    3 bluetooth
parport                38056    2 ppdev,parport_pc
...
```

The output lists the name of the kernel module and the amount of memory the module uses. The "Used by" column gives the total number of processes that are using the module and the other modules that it depends on, followed by a list of those dependent modules. For example, the `parport` module depends on the `ppdev` and `parport_pc` modules, which would be loaded before loading `parport`. These three modules are currently used by two processes.

`kmod` Package

The `lsmod` command and other kernel module files and utilities such as `modinfo`, `modprobe`, `depmod`, `insmod`, and `rmmod` are provided by the `kmod` package:

```
# rpm -qf /sbin/lsmod
kmod-14-9.el7.x86_64
```

To list all files provided by the `kmod` package, enter:

```
# rpm -ql kmod
/etc/depmod.d
/etc/depmod.d/dist.conf
/etc/modprobe.d
...
```

Listing Module Details

The `modinfo` command displays detailed information about a specific kernel module. For example, to display information about the `ipv6` kernel module, enter:

```
# modinfo ipv6
filename:       /lib/modules/3.8.13-
44.1.1.el7uek.x86_64/kernel/net/ipv6/ipv6.ko
alias:          net-pf-10
license:        GPL
Description:    IPv6 protocol stack for Linux
...
parm:      disable: Disable IPv6 module such that it is non...
parm:      disable_ipv6: Disable IPv6 on all interfaces (int)
parm:      autoconf: Enable IPv6 address autoconfiguration ...
...
```

Description of the output includes:

- **`filename`:** The absolute path of the kernel object file
- **`description`:** The short description of the module
- **`alias`:** The internal alias names for the module, if any
- **`depends`:** A comma-separated list of modules that this module depends on, if any
- **`parm`:** The parameter name and a short description

Modules are loaded from the `/lib/modules/<kernel_version>/kernel` directory. For example, to display the absolute path of the `nfs` kernel object file, enter:

```
# modinfo -n nfs
/lib/modules/3.8.13-44.1.1.el7uek.x86_64/kernel/fs/nfs/nfs.ko
```

Loading and Unloading Kernel Modules

- To load kernel modules:

```
# modprobe <module_name>
```

- To unload kernel modules:

```
# modprobe -r <module_name>
```

- Kernel module dependencies are listed in `/lib/modules/<kernel_version>/modules.dep`.
  - The file is created by `depmod` when kernel modules are installed.
- The Oracle-supplied kernel modules used in Grid Infrastructure are:
  - `oracleacfs, oracleadvm, oracleoks`
- The directory to specify modules to load at boot time is:
  - `/etc/sysconfig/modules`

Loading Kernel Modules

Kernel modules are loaded by using the `modprobe` command. The device manager for the Linux kernel, `udev`, uses `modprobe` to load drivers for automatically detected hardware.

For example, to load the `nfs` kernel module, enter:

```
# modprobe nfs
```

To list the newly loaded module, enter:

```
# lsmod | grep nfs
nfs                178150     0
fscache            51498      1      nfs
...
sunrpc             262929      6    nfs,...
```

The `nfs` kernel module and many other kernel modules have dependencies. Dependent modules are loaded first. Some of these modules may, in turn, be dependent on other modules.

You can use `modprobe -v` (verbose) to view the dependency resolution when loading a kernel module.

The following example includes the `-v` (verbose) option:

```
# modprobe -v nfs
insmod /lib/modules/3.8.13-
44.1.1.el7uek.x86_64/kernel/fs/fscache/fscache.ko
insmod /lib/modules/3.8.13-
44.1.1.el7uek.x86_64/kernel/fs/nfs/nfs.ko
```

Note that `modprobe` uses the `insmod` command to load the modules into the kernel. Do not use `insmod`, however, because this command does not resolve dependencies.

Kernel module dependencies are listed in `/lib/modules/<kernel_version>/modules.dep`. The `modprobe` command queries this file to determine dependencies. The `modules.dep` file is created by `depmod` when kernel modules are installed on your system.

Unloading Kernel Modules

Unload kernel modules by using the `modprobe -r` command. You can also use the verbose option. For example, to unload the nfs kernel module, enter:

```
# modprobe -rv nfs
rmmod nfs.ko
rmmod fscache.ko
```

Modules are unloaded in the reverse order, with the `nfs.ko` kernel module being unloaded first followed by the modules it was dependent on. Modules being used by a process or modules needed by other loaded modules are not unloaded.

The `modprobe -r` command uses `rmmod` to unload the modules. But similar to `insmod`, it is not recommended to use `rmmod` directly to unload kernel modules.

ACFS and ADVM Drivers

The installation of the Oracle Grid Infrastructure (GI) stack also installs ACFS (ASM Cluster File System), and ADVM (ASM Dynamic Volume Manager) drivers and utilities. ASM (Automatic Storage Management) is a feature within the Oracle Database to simplify the management of database files. There are three drivers to support ACFS and ADVM. These drivers are dynamically loaded (in top-down order) by the OHASD (Oracle High Availability Service Daemon) process during Oracle Clusterware startup.

- **`oracleoks.ko`**: This is the kernel services driver, providing memory management, lock, and cluster synchronization primitives.
- **`oracleadvm.ko`**: The ADVM driver maps I/O requests against an ADVM Volume Device to blocks in a corresponding on-disk ASM file location. This ADVM driver provides volume management driver capabilities that directly interface with the file system.
- **`oracleacfs.ko`**: This is the ACFS driver that supports all ACFS file system file operations.

Use the `lsmod` command to list these kernel modules. Example:

```
# lsmod | grep oracle
Module              Size      Used by
oracleacfs        781476      5
oracleadvm        212736      9
oracleoks         224864      2 oracleacfs,oracleadvm
```

Using the `/etc/sysconfig/modules` Directory

You can specify modules to be loaded at boot time by creating a file in the `/etc/sysconfig/modules` directory. Files in this directory must be executable shell scripts and the file name must end with `.modules`.

The `/etc/sysconfig/modules/<file_name>.modules` could be a simple call to `modprobe`. Example:

```
#!/bin/sh
modprobe abc
```

Or the file could be more elaborate, as in the following example:

```
#!/bin/sh
if [ ! -c /dev/input/uinput ] ; then
      exec /sbin/modprobe uinput >/dev/null 2>&1
fi
```

Kernel Module Parameters

- Pass parameters to a kernel module:

```
modprobe <module_name> [parameter=value]
```

- Configuration directory for `modprobe`:
  - `/etc/modprobe.d`
- Create `*.conf` files in `/etc/modprobe.d` to:
  - Specify options
  - Create aliases
  - Override normal `modprobe` behavior
  - Blacklist kernel modules
- Valid commands to use in these files include the following:
  - `alias`, `options`, `install`, `remove`, `blacklist`

Just as the kernel can accept boot time parameters to modify the behavior of the kernel, kernel modules can also accept parameters to modify their behavior. The syntax for passing parameters to a kernel module with `modprobe` is:

```
modprobe <module_name> [parameter=value]
```

Multiple *parameter=value* pairs can be passed by separating the pairs with spaces. Ensure that the module is not previously loaded, because `modprobe` does not reload the module.

Configuration Directory for `modprobe`

The configuration directory for `modprobe` is `/etc/modprobe.d`. Files in this directory end with `.conf` and are used for the following purposes:

- To specify options used with kernel modules
- To create aliases or alternative names for a module
- To override the normal `modprobe` behavior for modules with special requirements

The format of these `.conf` files is one command per line. Valid commands to use in these files include the following:

```
alias, options, install, remove, blacklist
```

Alias

Use the syntax `alias alias_name module_name` to create alternative names for kernel modules. You can also use shell wildcards in alias names. Example:

```
alias usbdevfs usbcore
```

Options

Use the syntax `options module_name option(s)` to add options to `module_name`. Example:

```
options b43 nohwcrypt=1 qos=0
```

Install

Use the syntax `install module_name command(s)` to tell `modprobe` to run shell commands rather than inserting the module in the kernel. Example:

```
install net-pf-6 /bin/true
```

Remove

This is similar to the `install` command, except it is invoked when `modprobe -r` is run. Use the syntax `remove module_name command(s)` to tell `modprobe -r` to run shell commands rather than unloading the module from the kernel.

Blacklist

Use the syntax `blacklist module_name` to tell `modprobe` to ignore a module's internal aliases. Internal aliases are those seen when using the `modinfo` *<module_name>* command. The blacklist keyword is typically used when the associated hardware is not needed, or when two or more modules support the same devices, or a module invalidly claims to support a device.

Refer to the `modprobe.d(5)` man page for more information.

Quiz

Which of the following commands displays details about a kernel module?

a. `lsmod` *<module_name>*

b. `modinfo` *<module_name>*

c. `modprobe` *<module_name>*

d. `depmod` *<module_name>*

Summary

In this lesson, you should have learned how to:

- Describe loadable kernel modules
- Dynamically load and unload kernel modules
- Configure kernel module parameters

Practice 9: Overview

The practices for this lesson cover using loadable kernel modules.

10

User and Group Administration

Objectives

After completing this lesson, you should be able to:

- Describe user and group implementation
- Describe user and group configuration files
- Configure users and groups by using command-line utilities
- Implement user private groups (UPG)
- Configure password aging and the hashing algorithm
- Use the User Manager GUI tool
- Manage the use of `su` and `sudo` commands
- Describe user and group implementation in the enterprise

Introduction to Users and Groups

- User account information is stored in `/etc/passwd`.
- Group information:
  - Group information is stored in `/etc/group`.
  - Each user has a private group (UPG).
  - Users can belong to more than one group.
- Oracle Linux uses shadow passwords.
  - `/etc/shadow`: Hashed user passwords
  - `/etc/gshadow`: Hashed group passwords
  - `/etc/login.defs`: Security policies

Each user in Linux has a unique user ID (UID), which is an ordinary integer number, and an associated username. Users log in by using their usernames, but the system uses the associated UIDs. Each user account also has a home directory and a login shell. When users log in, they are placed in their home directory and their login shell executes. All of this user account information is stored in the `/etc/passwd` file.

Each user also belongs to one or more groups. Different users can be assigned to the same group. Access can be given to a group and all members of the group are granted the same access privileges. Each group account in Linux has a unique group ID (GID) and an associated group name. Group information is stored in the `/etc/group` file.

Oracle Linux uses a user private group (UPG) scheme. When a new user account is added, a new user private group is also created. The user private group has the same name as the user, and the new user is the only member of this group.

Both users and groups use shadow passwords. Passwords are hashed and stored in different files, `/etc/shadow` for users and `/etc/gshadow` for groups. Security improves by storing hashed passwords in “shadow” files, because these files are readable only by the `root` user. The use of shadow passwords also provides password aging parameters and allows security policies to be enforced, using the `/etc/login.defs` file.

Only the `root` user can add, modify, or delete user and group accounts.

User and Group Configuration Files

- Contents of `/etc/passwd`:
  - username: placeholder: UID: GID: GECOS: home dir: shell
- Contents of `/etc/shadow`:
  - username: hashed password: password aging information
- Contents of `/etc/group`:
  - groupname: placeholder: GID: comma-separated members
- Contents of `/etc/gshadow`:
  - groupname: hashed password: GID: comma-separated administrators: comma-separated members
  - Group passwords are rarely used.

/etc/passwd

When a new user is added, the information is stored as a single, colon-separated line in `/etc/passwd`. Here is an example of an entry in this file:

```
# tail -1 /etc/passwd
oracle:x:1000:1000:Oracle Student:/home/oracle:/bin/bash
```

The following describes this entry:

- **`oracle`**: Username
- **`x`**: Indicates that shadow passwords are used
- **`1000`**: UID, these begin with 1000 and increment by 1 for each newly added user. UIDs below 1000 are reserved for system use.
- **`1000`**: GID of the user's primary group. These begin with 1000 and increment by 1 for each new group. Users can belong to more than one group.
- **`Oracle Student`**: GECOS (General Electric Comprehensive Operating System) information, used only for informational purposes such as full name
- **`/home/oracle`**: Home directory for this user
- **`/bin/bash`**: Default shell for this user

/etc/shadow

With shadow passwords, a new entry is automatically added to `/etc/shadow` when a new user is created. This file can be viewed only by `root`. Here is an example of an entry in this file:

```
# tail -1 /etc/shadow
oracle:$6$Q4pxg...:16331:0:99999:7:::
```

The following describes this entry:

- **oracle:** Username
- **6Q4pxg...:** Hashed password value (partial value shown). The plain text password itself is not stored on the disk. An algorithm creates a unique string from a password.
- **16331:** Number of days since password has changed (counted in days since Jan 1, 1970).
- **0:** Number of days that need to pass before the password must be changed by the user.
- **99999:** Maximum number of days since the password changed that the password can be used. After this amount of days, the password must be changed by the user.
- **7:** Number of days before expire date that the user is warned about the pending password change policy. If the password is not changed after this number of days, the user account is locked.

The next field is empty but is used to store the last date when the account is locked (counted in days since Jan 1, 1970). The last field is also empty but is not used.

/etc/group

Because Oracle Linux uses a UPG scheme, a new entry is automatically created in `/etc/group` when a new user is added. The group name is the same as the username. Here is an example of an entry in this file:

```
# tail -1 /etc/group
oracle:x:1000:oracle
```

The following describes this entry:

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

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

/etc/gshadow

Hashed group passwords are stored in this file. However, group passwords are rarely used. Here is an example of an entry in this file:

```
# tail -1 /etc/gshadow
oracle:!!::oracle
```

The following describes this entry:

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

The last two fields are used to designate administrators and members.

Adding a User Account

- Use the `useradd` command to add a user:

```
# useradd [options] user_name
```

- Use the `passwd` command to create a password:

```
# passwd [options] user_name
```

- User default settings are stored in:
  – `/etc/default/useradd`
- Use the `-D` option to display or modify defaults:

```
# useradd -D [options]
```

- A new user's home directory is populated with files from:
  – `/etc/skel` directory
- To create a `nologin` user:

```
# useradd -s /sbin/nologin user_name
```


useradd

Use the `useradd` command to add a user account. The syntax is:

```
useradd [options] user_name
```

When creating a new user without any options, the default settings are applied. Example:

```
# useradd jim
# tail -1 /etc/passwd
jim:x:1001:1001::/home/jim:/bin/bash
```

Also by default, `useradd` creates a locked user account. To unlock the account and assign a password, run the `passwd` *`user_name`* command as root. Example:

```
# passwd jim
```

The `passwd` *`user_name`* command prompts you for a new password. Depending on the complexity of the password, you may be notified the password is bad (too short or too simple). Re-enter the same password to continue and unlock the user account.

The same `passwd` command is used to change a password. The root user can always change a user's password. Users are prompted to enter the current password first.

Default Settings

The default settings for a new user can be viewed and modified by using the `-D` option. Example:

```
# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

The `INACTIVE` directive sets the number of days after a password expires until the account is locked. A value of `0` locks the account as soon as the password expires. A value of `-1` disables the feature. Contents of the `SKEL` (`/etc/skel` by default) are copied to a new user's home directory when the user account is created. Default settings are stored in `/etc/default/useradd`. The following options, used with `-D`, change the `useradd` command defaults:

- **`-b default_home`**: The initial path prefix for a new user's home directory
- **`-e default_expire_date`**: The date on which the user account is disabled
- **`-f default_inactive`**: The number of days after a password has expired before the account is locked
- **`-g default_group`**: The group name or ID for a new user's initial group
- **`-s default_shell`**: The new user's login shell

For example, to change a new user's login shell to the Bourne shell, enter the following:

```
# useradd -D -s /bin/sh
```

`useradd` Options

Several options are available to the `useradd` command to override default settings. The following are some of the more commonly used options:

- **`-c`** ***`comment`***: The new user's GECOS information, such as full name
- **`-d`** ***`home_dir`***: The initial path prefix for a new user's home directory
- **`-e`** ***`expire_date`***: The date (format YYYY-MM-DD) when the user account is disabled
- **`-g`** ***`initial_group`***: The group name or number of the user's initial login group. The group name must exist. A group number must refer to an already existing group.
- **`-G`** ***`group`***: A list of secondary groups that the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace.
- **`-p`** ***`passwd`***: Set the new user's password.
- **`-s`** ***`shell`***: The name of the user's login shell

For example, to create a new username of "mary," and include the user's name, and change the login shell to the C shell, enter the following:

```
# useradd -c "Mary Smith" -s /bin/csh mary
```

`nologin` Shell

When you add a new user account, the user is granted shell access by default. You can create a user account with `nologin` shell for purposes of running a service such as SMTP, FTP, or running a web server, for example. A user without a login shell cannot log in to a system and, therefore, cannot run any commands interactively on the system. Processes can run as that user, however.

Logging in as a user with a `nologin` shell is politely refused and a message is displayed that the account is not available. If the file `/etc/nologin.txt` exists, `nologin` displays the file's contents rather than the default message.

To create a `nologin` user, first ensure that `nologin` exists in the `/etc/shells` file:

```
# cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/tcsh
/bin/csh
```

To add a new user called `test` with no shell access:

```
# useradd -s /sbin/nologin test
```

Attempting to log in as user `test` displays:

```
# su – test
This account is currently not available.
```

Modifying or Deleting User Accounts

- Use the `usermod` command to modify a user:

```
# usermod [options] user_name
```

- Example: To add a user to a secondary group (GID=1017):

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

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

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

- Options to `userdel` include:
  - `-f`: Force removal even if user is logged in
  - `-r`: Remove the user's home directory

usermod

Use the `usermod` command to modify an existing user account. The syntax is:

```
usermod [options] user_name
```

One of the most common uses of the `usermod` command is to add a user to another (secondary) group. Use the `-a` and `-G` options followed by a comma-separated list of the secondary groups to add the user to. The following example lists the contents of `/etc/group` before and after modifying a user and adding them to a secondary group:

```
# grep 1017 /etc/group
students:x:1017:
# usermod -aG 1017 mary
# grep 1017 /etc/group
students:x:1017:mary
```

userdel

Use the `userdel` command to delete a user account. Example:

```
# userdel mary
```

Group Account Administration

- Use the `groupadd` command to add a group account:

```
# groupadd [options] group_name
```

- Use the `groupmod` command to modify a group account:

```
# groupmod [options] group_name
```

- Use the `groupdel` command to delete a group account:

```
# groupdel group_name
```

- Use the `gpasswd` command to administer group accounts:

```
# gpasswd [options] group_name
```

- Example: To add a user (`jim`) to a group (`students`):

```
# gpasswd -a jim students
```

- The `groups` command prints the groups to which a user belongs.
- The `newgrp` command changes the real group identification.

ORACLE

groupadd

Use the `groupadd` command to add a group account. The syntax is:

```
groupadd [options] group_name
```

groupmod

Use the `groupmod` command to modify a group account. The syntax is:

```
groupmod [options] group_name
```

groupdel

Use the `groupdel` command to delete a group account. The syntax is:

```
groupdel group_name
```

You can remove groups even if there are members in the group. You cannot remove the primary group of any existing user. You must remove the user before removing the group.

gpasswd

Use the `gpasswd` command to administer `/etc/group` and `/etc/gshadow`. Every group can have administrators, members, and a password. The syntax is:

```
gpasswd [options] group_name
```

groups

The `groups` command displays the groups that a user belongs to. The following example illustrates that user `oracle` belongs to two groups, `oracle` (primary group) and `students` (secondary group):

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

The `groups` command (logged on as `oracle`) verifies these group memberships.

```
$ whoami
oracle
$ groups
oracle students
```

newgrp

The `newgrp` command executes a new shell and changes a user's real group identification. The following example illustrates the group ID before and after running the command. It also illustrates that a new shell is executed.

```
$ id
uid=1000(oracle) gid=1000(oracle)
groups=1000(oracle),1066(students)...
```

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

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

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

Also note that a new shell was executed:

```
$ ps
  PID TTY       TIME CMD
20279 pts/0 00:00:00 bash
20464 pts/0 00:00:00 bash
20486 pts/0 00:00:00 ps
```

The `newgrp` command does not recognize group ID numbers and you can only change your real group name to a group that you are a member of. Running the command without an argument sets the real group identification to the user's primary group.

User Private Groups

- Each user belongs to a unique group.
  – Eliminates the need for `umask=0022`, which makes groups read-only
  – Allows `umask=0002`, which gives write permission to group
- Additional steps to implement:
  1. Create a directory to share.
  2. Create a new group.
  3. Add users to this new group.
  4. Change the group ownership for the directory.
  5. Set the `setgid` bit on the directory.

As previously mentioned, Oracle Linux uses a user private group (UPG) scheme. Whenever new user accounts are added, they have a unique group. The purpose of this UPG scheme is to make Linux groups easier to use.

Traditionally on Linux systems, the `umask` was `0022`, which prevents other users and other members of a user's primary group from modifying a file. The following example illustrates the permissions on newly created files and directories when `umask` is set to `0022`:

```
$ umask
0022
$ mkdir project
$ ls -ld project
drwxr-xr-x.  project
$ touch project/testfile
$ ls -l project/testfile
-rw-r--r--.  project/testfile
```

As you can see, the permissions for the group are read-only when `umask` is set to `0022`.

With UPG, all users have their own private group, so the group protection provided by setting `umask` to `0022` is not needed. With UPG, the `umask` is set to `0002` in `/etc/profile` and `/etc/bashrc`. Permissions on newly created files and directories are:

```
$ umask
0002
$ mkdir project
$ ls –ld project
drwxrwxr-x.  project
$ touch project/testfile
$ ls –l project/testfile
-rw-rw-r--.  Project/testfile
```

Now, the group does have write permission on the newly created file and directory.

To allow multiple users write access to files within the same directory, create a new group, add the users to this new group, change the group ownership on this directory to the new group, and set the `setgid` bit on the directory. Files created in this directory have the group permission set to the directory's group, rather than the primary group ID of the user who created the file.

For example, if you created a *project* group, added users to this *project* group, created a *project* directory, changed the group ownership for this *project* directory to the *project* group, set the `setgid` bit for the *project* directory, all *project* users are able to edit the *project* files and create new files in the *project* directory. Any files these users create retain their *project* group status. Other *project* users can always edit these files.

Password Configuration

- Password aging requires users to change their password.
- Use the `chage` command to configure password aging:

```
# chage [options] user_name
```

- Current values are displayed and changed interactively:

```
Minimum Password Age [0]:
Maximum Password Age [99999]:
Last Password Change [2011-11-06]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date [1969-12-31]:
```

- Use the `authconfig` command to configure the password hashing algorithm:

```
# authconfig --passalgo=<algorithm> --update
```

ORACLE

Password aging requires users to change their password periodically. Use the `chage` command to configure password expiration. The syntax is:

```
chage [options] user_name
```

Enter the `chage` command, followed by a username, to display existing password aging values and make modifications. For example, to display and change values for user `frank`, type (as user `root`):

```
# chage frank
Changing the ageing information for frank
Enter the new value, or press ENTER for the default
        Minimum Password Age [0]:
        Maximum Password Age [99999]:
        Last Password Change (YYYY-MM-DD) [2011-11-06]:
        Password Expiration Warning [7]:
        Password Inactive [-1]:
        Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
```

Password aging information is stored in the `/etc/shadow` file. To view the user `frank`'s entry before making any changes:

```
# grep frank /etc/shadow
frank:$6$XBlUm6w...:15284:0:99999:7:::
```

Changing the minimum password age value to `14` and maximum password age value to `30` means that in `14` days the user has `30` days to change their password. The new entry appears as:

```
# grep frank /etc/shadow
frank:$6$XBlUm6w...:15284:14:30:7:::
```

Based on this information, the user is warned to change his password seven days before the date the password expires.

The `INACTIVE` directive is used to set the number of days of inactivity after a password has expired before the user account is locked. Setting `INACTIVE` to `-1` disables this feature.

`chage` Options

A number of options are available for the `chage` command.

To list aging information:

```
# chage -l frank
Last password change                                : Sep 24, 2014
Password expires                                    : Oct 24, 2014
Password inactive                                   : never
Account expires                                     : never
Minimum number of days between password change      : 14
Maximum number of days between password change      : 30
Number of days or warning before password expires: 7
```

To force a user to set a new password immediately (force immediate expiration), set the last password change value to `0`. Example:

```
# chage -d 0 frank
```

After login, the user is prompted to change his password.

`authconfig`

The Linux user password hashing algorithm is also configurable. Use the `authconfig` command to determine the current algorithm being used, or to set it to something different. To determine the current algorithm:

```
# authconfig --test | grep hashing
password hashing algorithm is sha512
```

To change the algorithm, use the `--passalgo` option with one of the following as a parameter: `descrypt`, `bigcrypt`, `md5`, `sha256`, or `sha512`, followed by the `--update` option. For example, to change the algorithm to MD5:

```
# authconfig --passalgo=md5 --update
```

/etc/login.defs File

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

ORACLE

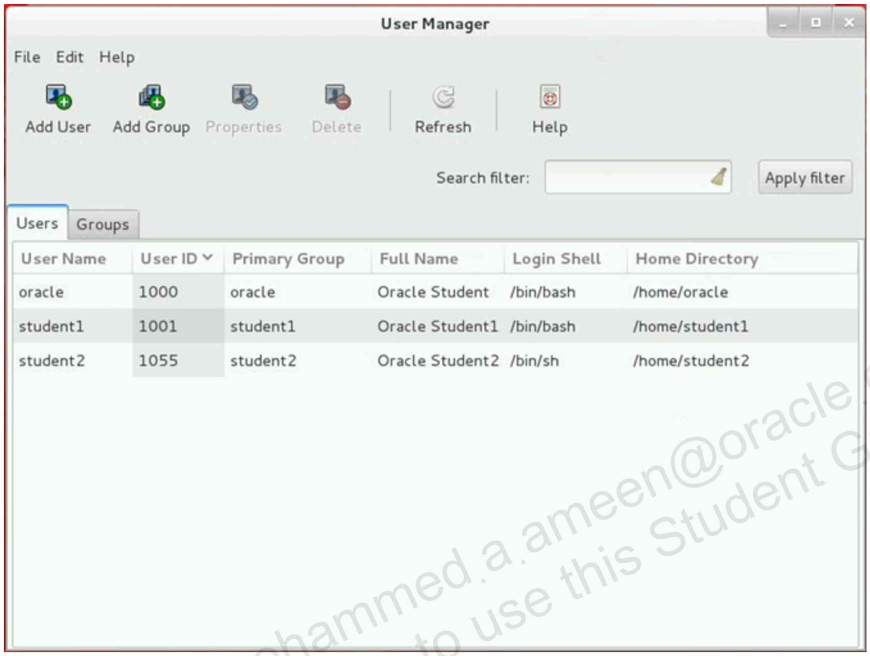
The /etc/login.defs file provides default configuration information for several user account parameters. The useradd, usermod, userdel, and groupadd commands, and other user and group utilities take default values from this file. Each line consists of a directive name and associated value. The following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

- Location of user mailboxes
- Password aging controls
- Minimum and maximum values for automatic UID selection (1000 to 60000)
- Minimum and maximum values for automatic GID selection (1000 to 60000)
- Whether home directories should be created when adding a new user
- Default umask
- Encryption method used to encrypt passwords

If the USERGROUPS_ENAB directive in /etc/login.defs is set to YES, a group is created for the user with the same name as the username. If the directive is set to NO, the useradd command sets the primary group of the new user to the value specified by the GROUP directive in the /etc/default/useradd file, or 100 by default.

User Manager Tool

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



The slide shows the graphical tool used to perform user and group account administration. The User Manager application provides a GUI to view, modify, add, and delete local users and groups. Use the `system-config-users` command to start the tool.

Two tabs are available, a Users tab for user administration, and a Groups tab for group administration. To add a user or group, click the Add User or Add Group button. To modify an existing user or group, select the entry from the list and click the Properties button. Select an entry from the list and click the Delete button to delete a user or group account.

A Search filter is available to find a specific user or group. Enter the first few letters of the name in the “Search filter” field and click the “Apply filter” button. You can also sort on any column by clicking the column header.

Restricting Use of the `su` Command

- You can limit access to the `su` command to only those users who are members of the `wheel` group.
- To limit access to the `su` command to the `oracle` user, add the `oracle` user to the `wheel` group as follows:

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

- Add the following line to the `/etc/pam.d/su` file to only permit `root` access to members of the `wheel` group:

```
auth   required   pam_wheel.so   use_uid
```


You can limit access to the `su` command to only those users who are members of the `wheel` group. Two steps are needed to limit user access to the `su` command:

1. Add user(s) to the `wheel` group.
2. Add an entry to the `/etc/pam.d/su` file.

For example, to limit access to the `su` command to the `oracle` user, add the `oracle` user to the `wheel` group as follows:

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

Pluggable Authentication Modules (PAM) is an authentication mechanism that allows you to configure how applications use authentication to verify the identity of a user. Add the following line to the `/etc/pam.d/su` file to permit only `root` access to members of the `wheel` group:

```
auth   required   pam_wheel.so   use_uid
```

The preceding entry is in the `/etc/pam.d/su` file by default, but is commented out. Users using the `su` command still need to know the `root` password. The following line allows members of the `wheel` group to `su` to `root` without knowing the password:

```
auth   sufficient   pam_wheel.so   trust   use_uid
```

Allowing Use of the `sudo` Command

- `sudo` privileges are configured in the `/etc/sudoers` file.
- The following entry is present in the `/etc/sudoers` file by default:

```
root ALL=(ALL)   ALL
```

- The following entry in `/etc/sudoers` allows the `oracle` user to use `sudo` to run administrative commands:

```
oracle ALL=(ALL)   ALL
```

- The `oracle` user can now run administrative commands by preceding the command with `sudo`, for example:

```
$ sudo useradd new_user
[sudo] password for oracle:
```

- You are prompted for the `oracle` user password, not the `root` user password.

ORACLE

You can also allow a regular user to run a command as `root` by preceding the administration command with `sudo`. The following example uses `sudo` to allow a regular user to add a new user account. You are prompted for your regular user password, not the `root` password.

```
$ sudo useradd new_user
```

The `sudo` privileges are configured in the `/etc/sudoers` file. The following entry is present in the `/etc/sudoers` file by default:

```
root ALL=(ALL)   ALL
```

The "`ALL=(ALL)`" argument gives the `root` user permission to run from any terminal, acting as any user. The ending "`ALL`" argument indicates that the `root` user can execute any command.

Use the `visudo` command to edit the `/etc/sudoers` file. This command locks the `/etc/sudoers` file against simultaneous edits. To authorize the `oracle` user to run any command with `root` privileges, create the following entry in the `/etc/sudoers` file:

```
# visudo
oracle ALL=ALL
```

The `oracle` user can now run administration commands by preceding them with `sudo`.

User/Group Administration in the Enterprise

- User and group account information is often centralized.
- Centralized information can be retrieved by using:
  – Lightweight Directory Access Protocol (LDAP)
  – Network Information Service (NIS)
- User home directories can also be centralized and accessed remotely.
  – Remote file systems can be auto-mounted.

In enterprise environments with possibly hundreds of servers and thousands of users, user and group account information can be stored in a central repository rather than in files on several servers. You can configure user and group information on a central server and retrieve this information by using services such as Lightweight Directory Access Protocol (LDAP) and Network Information Service (NIS). You can also create users' home directories on a central server and automatically mount, or access, these remote file systems.

LDAP (Lightweight Directory Access Protocol)

LDAP is a set of open protocols used to access information stored remotely over a network. LDAP is commonly used for centrally managed users and groups, user authentication, or system configuration.

Network Information Service (NIS)

NIS is a directory service that provides a centralized location for usernames, group names, and host names. NIS simplifies the maintenance of these common administrative files by keeping them in a central database. As with LDAP, other systems on the network contact the central server to retrieve information.

Quiz

Oracle Linux implements shadow passwords and user private groups.

a. True

b. False

Quiz

Which of the following statements are true?

a. User account information is stored in `/etc/passwd`.

b. Group account information is stored in `/etc/group`.

c. Password aging information is stored in `/etc/passwd`.

d. Default settings for a new user are stored in `/etc/default/useradd` and `/etc/login.defs`.

Summary

In this lesson, you should have learned how to:

- Describe user and group implementation
- Describe user and group configuration files
- Configure users and groups by using command-line utilities
- Implement user private groups (UPG)
- Configure password aging and the hashing algorithm
- Use the User Manager GUI tool
- Manage which users can use the `su` and `sudo` commands
- Describe user and group implementation in the enterprise

Practice 10: Overview

The practices for this lesson cover the following:

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

11

Partitions, File Systems, and Swap

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe disk partitioning
- Use disk partitioning utilities
- Describe supported file system types
- Describe file system creation, mounting, and maintenance
- Describe and maintain swap space

Disk Partitions

- Partitioning divides a disk drive into logical disks.
  - Each partition is treated as a separate disk.
  - A partition table defines the partitions.
- Minimum recommended partitions (file systems):
  - / (root)
  - `/boot`
  - `swap`
- Create additional partitions to simplify administration.
- Extended partitions allow the creation of more than four primary partitions.

Partitioning divides a disk drive into one or more logical disks. Each partition is treated as a separate disk with its own file system. Partition information is stored in a partition table.

Oracle Linux needs at least one partition for its root file system. It is also recommended to create a second partition dedicated as a swap partition. On Intel-compatible hardware, the BIOS that boots the system can often only access the first 1024 cylinders of the disk. For this reason, a third partition is often created as a boot partition to store the kernel image and a few other files needed at boot time.

Additional partitions can be created to simplify administration and backups, to increase system security, and to accommodate other needs, such as testing. For example, data that frequently changes, such as user home directories, databases, and log file directories, is typically created on separate partitions to facilitate backups.

Primary and Extended Partitions

The original partitioning scheme for PC hard disks allowed only four partitions, called primary partitions. To create more than four partitions, one of these four partitions can be divided into many smaller partitions, called logical partitions. When a primary partition is subdivided in this way, it is known as an extended partition. The partitioning tools presented in this lesson allow you to create primary or extended partitions.

Partitions Created During Installation

- During the installation of Oracle Linux, you created three primary partitions:
  - `xvda1`, mounted on `/boot`
  - `xvda2`, mounted on `/`
  - `xvda3`, mounted on `/home`
- A fourth partition, `xvda4`, was created as an extended partition.
  - Extended partitions can be subdivided into logical partitions.
- One logical partition, `xvda5`, was created on the extended partition.
  - This logical partition was designated as a `swap` partition.
- Partition devices are listed in the `/proc/partitions` file:

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


When you installed Oracle Linux in the lesson titled "Installing Oracle Linux 7," you created three primary partitions: `xvda1`, `xvda2`, and `xvda2`. A fourth partition, `xvda4`, was created as an extended partition, allowing it to be subdivided into logical partitions. One logical partition, `xvda5`, was created on the extended partition that was designated as a `swap` partition.

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

```
# cat /proc/partitions
major minor  #blocks name
 202     0  12582912 xvda
 202     1    512000 xvda1
...
```

The columns are described as follows:

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

Partition Table Manipulation Utilities

- Three partition utilities are presented in this lesson:
  - `fdisk`
  - `cfdisk`
  - `parted`
- Do not partition a device while it is in use.
- Ensure that file systems are unmounted:
  - Use the `umount` command.
- Ensure that swap space is disabled:
  - Use the `swapoff` command.

Various utilities are available to display and manipulate the partition table. Three of these utilities are presented in this lesson:

- `fdisk`
- `cfdisk`
- `parted`

If the file system is greater than 2 TB, you cannot use `fdisk` but you must use `parted`. When removing or resizing a partition by using any partition table manipulator command, the device on which that partition resides must not be in use. Also *not* recommended is creating a new partition on a device that is in use. Unmount the file system partitions and disable the swap partition before making any changes to the partition table.

Use the `umount` command to unmount file systems, and use the `swapoff` command to disable the swap space. These commands are discussed in more detail later in this lesson.

`fdisk` Utility

- The `fdisk` utility is a partition table manipulator for Linux.
- Use the `fdisk -l` option to list the partition table.
  - Device: Lists the partitions
  - Boot: `*` indicates that the partition contains boot files
  - Start and End: The starting and ending sectors
  - Blocks: The number of blocks allocated to the partition
  - Id and System: The partition type
- Partition naming (example: `/dev/xvda1`)
  - `/dev`: The directory containing device files
  - `sd`: SCSI disk; `hd`: IDE disk; `xvd`: Virtual disk
  - `a`: First disk; `b`: Second disk; `c`: Third disk
  - `1`: First partition; `2`: Second partition; `3`: Third partition

ORACLE

The `fdisk` utility is a common partition table manipulator for Linux. Use `fdisk -l` to list the partition table. Output varies depending on the number of attached disks and partitions. To display the partition for a specific device, include the device name as an argument. For example:

```
# fdisk -l /dev/xvda
Disk /dev/xvda: 12.9 GB, 12884901888 bytes, 25165824 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000d8871
    Device  Boot  Start       End      Blocks    Id    System
/dev/xvda1    *    2048   1026047      512000    83    Linux
/dev/xvda2       1026048  13314047    6144000    83    Linux
...
```

Without specifying a device as an argument, partitions in `/proc/partitions` are listed.

The first five lines of output from the `fdisk -l /dev/xvda` command are summary information about the device itself, `/dev/xvda`. The example output shows a 12.9 GB virtual disk (`xvda`) with 25165824 sectors. With zone density recording (ZDR), a disk surface is divided into 16 circumferential zones. The number of sectors for each track is different in each zone, with the outermost zone containing the most sectors and the innermost zone containing the smallest number of sectors. Therefore, if a partition spans zones, the number of sectors per track would not be the same.

The partition table is displayed after the summary information. Seven columns of information are listed in the partition table. The Device column shows five partitions: `/dev/xvda1`, `/dev/xvda2`, `/dev/xvda3`, `/dev/xvda4`, and `/dev/xvda5`. The Boot column shows that the first partition, `/dev/xvda1`, has an asterisk (`*`) indicating that this partition contains the files required by the boot loader to boot the system. The Start and End columns list the starting and ending sectors of each partition. The Blocks column lists the number of blocks allocated to the partition. The Id and System columns identify the partition type.

Partition Types

The partition types can be displayed and changed by using the `fdisk` utility. A partial list of partition types are:

- **`83:`** Linux
- **`82:`** Linux swap
- **`5:`** Extended
- **`8e:`** Linux LVM

Partition Naming

The Linux partition naming scheme is in the `/dev/`*`xxyN`* form. Elements of this naming scheme are described as follows:

- `/dev/`
  This is the directory in which all device files reside.
- `xx` (or `xxx`)
  The first two of three letters indicate the type of device on which the partition resides. These letters are usually `hd` (for IDE disks), `sd` (for SCSI disks), or `xvd` (for virtual disks).
- `y`
  This letter indicates which device the partition is on—for example, `/dev/sda` (the first SCSI hard disk) or `/dev/xvdb` (the second virtual disk).
- `N`
  This number indicates the partition. For example, `/dev/sdb1` is the first partition on the second SCSI device and `/dev/xvda3` is the third partition on the first virtual disk.

Using the `fdisk` Utility

- The `fdisk` utility provides an interactive interface.

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

- Basic `fdisk` commands include:
  - `d`: Delete a partition.
  - `l`: List the known partition types.
  - `m`: Print the available commands.
  - `n`: Add a new partition.
  - `p`: Print the partition table.
  - `w`: Write the table to disk and exit `fdisk`.
- To have the kernel re-read the partition table:

```
# partprobe <device_name>
```


The `fdisk` utility also provides an interactive interface for manipulating the partition table of a disk device. `fdisk` understands a number of different partition table types. The supported partition types are displayed when viewing the `fdisk` menu.

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

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

Informational messages are displayed:

```
Welcome to fdisk (util-linux 2.23.2).
Changes will remain in memory only, until you decide to write
the.
Be careful before using the write command.
```

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

```
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier ...
```

The `fdisk` command prompt then appears:

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

Enter `m` to display the `fdisk` commands.

`fdisk` Commands

```
Command (m for help): m
Command action
   a    toggle a bootable flag
   b    edit bsd disklabel
   c    toggle the dos compatibility flag
   d    delete a partition
   g    create a new empty GPT partition table
   G    create an IRIX (SGI) partition table
   l    list known partition types
   m    print this menu
   n    add a new partition
   o    create a new empty DOS partition table
   p    print the partition table
   q    quit without saving changes
   s    create a new empty sun disklabel
   t    change a partition's system id
   u    change display/entry units
   v    verify the partition table
   w    write table to disk and exit
   x    extra functionality (experts only)
```

To create a new 5 GB primary partition:

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

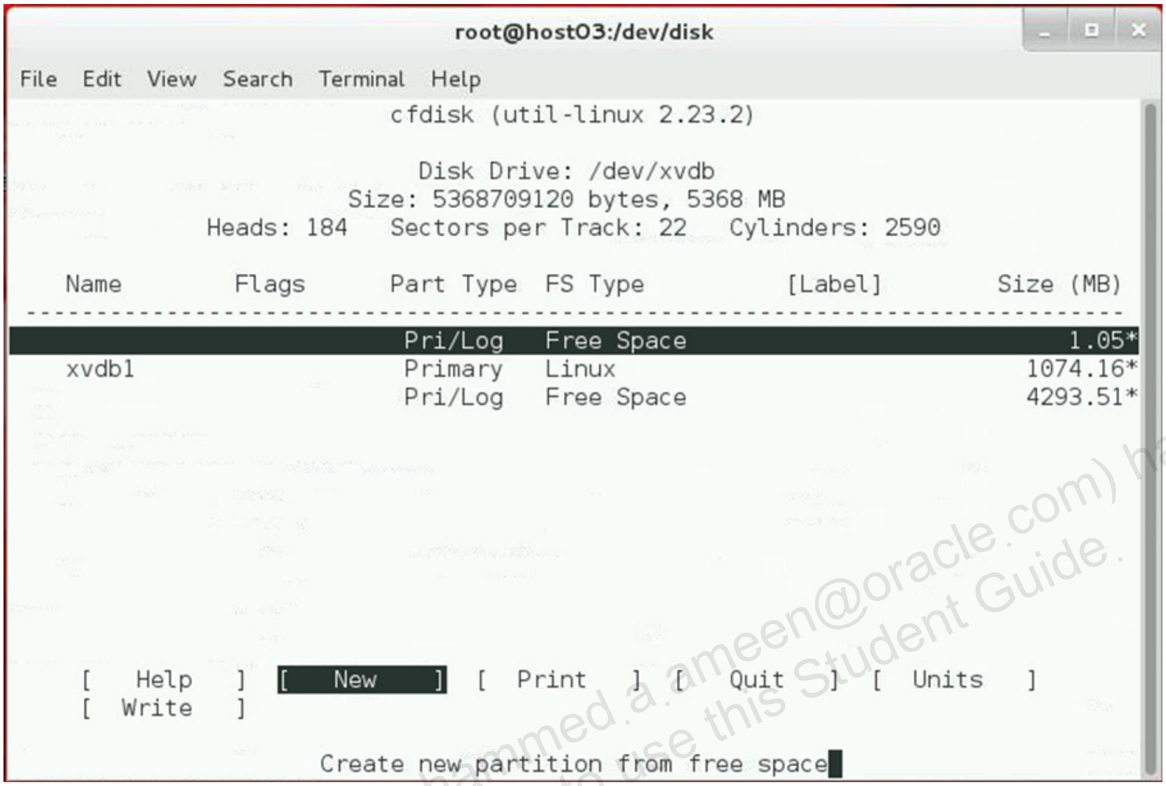
The `fdisk` command prompt displays again. Enter `w` to write the table to disk and exit.

`partprobe` Command

This command informs the kernel of partition table changes. Run this command with the device name as an argument to require the operating system to re-read the partition table:

```
# partprobe /dev/xvdb
```

cfdisk Utility



The screenshot in the slide shows the user interface of the `cfdisk` utility, which is used to create, delete, and modify partitions on a disk device. Enter the `cfdisk` command and include the device that you want to partition as an argument. Example:

```
# cfdisk /dev/xvdb
```

Summary information for the disk device is displayed at the top of the window. The partition table is displayed in the middle of the window. Selectable commands are displayed in brackets at the bottom of the window.

Use the up and down arrow keys to select a partition from the list. Use the right and left arrows to select a command. All partition-specific commands apply to the current partition.

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

parted Utility

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

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

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

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

- Interactive mode displays a `(parted)` prompt.
  - Enter `help` to view a list of available commands.
  - Enter `help command` to view detailed help on a specific command.

The GNU `parted` utility is also used to view the existing partition table, change the size of existing partitions, or add partitions from free space or additional hard drives. This utility is more advanced than the `fdisk` utility. It supports more disk label types and offers additional commands. `parted` syntax is:

```
parted [option] device [command [argument]]
```

Use `parted` interactively to enter commands one at a time. Include only the device as an argument to invoke interactive mode. Example:

```
# parted /dev/xvdd
GNU Parted 3.1
Using /dev/xvdd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

From the `(parted)` prompt, enter a command or type `help` to view the list of available commands. Get additional help on a specific command by typing `help` plus the command. Example:

```
(parted) help mkpart
```

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

```
(parted) mklabel gpt
```

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

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

```
(parted) mkpart
Partition name? []?
File system type? [ext2]?
Start? 0
End? 20%
(parted)
```

With a GPT disk label, you are first prompted to optionally give the partition a name. You are not required to name your partitions. Next, you are prompted for a file system type. A large number of file system types are supported. You are then prompted for the Start and End parameters for the partition.

To display the new partition, enter the `print` command. Example:

```
(parted) print
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdd: 5369MB
Sector size (logical/physical): 512/512
Partition Table: gpt
Number  Start   End     Size    File system  Name  Flags
1       17.4kB  1074MB  1074MB
```

The columns of output differ depending on the type of partition table. To exit the `parted` utility, enter `quit`:

```
(parted) quit
```

File System Types

- ext2
  - High performance for fixed disk and removable media
- ext3
  - Journaling version of ext2
- ext4
  - Supports larger files and file system sizes
- vfat
  - MS-DOS file system useful when sharing files between Windows and Linux
- XFS
  - High-performance journaling file system
- Btrfs
  - Addresses scalability requirements of large storage systems

After partitioning the disk device, you need to make a file system on the partitions. Making a file system writes information to the device and creates order of the empty space. This file system–related data consumes a small percentage of the space. The remaining space on the disk drive is split into small, consistently sized segments called blocks.

Oracle Linux supports a number of file system types, some of which are described as follows.

ext2

The second extended file system was introduced in Linux in January 1993. ext2 supports a maximum file system size of 8 TB and a maximum file size of 2 TB. ext2 file systems are susceptible to corruption during a power failure or any unclean system shutdown. In this case, mounted ext2 file systems must be checked for consistency. This consistency check causes delays in boot time, especially when file systems contain a large number of files.

ext3

The ext3 file system is an improvement on the ext2 file system and includes journaling capabilities. The journaling provided by the ext3 file system improves reliability and availability. The consistency checks required by ext2 file systems after an unclean shutdown are not necessary with ext3. ext3 supports a maximum file system size of 16 TB and a maximum file size of 2 TB. ext2 file systems are upgradeable to ext3 without reformatting.

A file system journal first performs a write operation in a journal. Then it performs the write on the file system itself and removes the entry from the journal. Journaling ensures that the file system is able to recover from power failures or system crashes by recovering from the journal and removing any incomplete entries.

ext4

The ext4 file system is a scalable successor to ext3. ext4 supports very large file systems and files, extents (contiguous physical blocks), pre-allocation, delayed allocation, faster file system checking, more robust journaling, and other enhancements. ext4 supports a maximum file system size of 1 EB and a maximum file size of 16 TB.

Journal options for ext3 and ext4 file systems can be specified on the command line by using `-J <journal-options>`. Journal options are comma-separated and can take an argument using the equals (=) sign. The following journal options are supported:

- **`size=journal-size`:** This creates an internal journal (stored inside the file system) with the size specified in megabytes.
- **`device=external-journal`:** This attaches the file system to the journal block device located on `external-journal`. The external journal must already have been created using the `mke2fs -O journal_dev external-journal` command.

vfat

The vfat file system (also known as FAT32) is an MS-DOS file system. It is supported by Linux but is not journaled and lacks many of the features available with the ext file system types. Because vfat file systems are readable by both Windows and Linux, they are useful for exchanging data between these operating systems.

XFS

The XFS file system is a high-performance journaling file system. XFS supports a maximum file system size of 100 TB and a maximum file size of 16 TB. You can create an XFS file system on a regular disk partition and on a logical volume. With Oracle Linux 7, the default file system type is XFS. XFS is discussed further in the lesson titled “Implementing the XFS File System.”

Btrfs

The Btrfs (B-Tree file system) is a copy-on-write file system for Linux designed to address the expanding scalability requirements of large storage subsystems. Btrfs provides extent-based file storage with a maximum file size of 16 EB. Btrfs offers the ability to create both readable and writable snapshots and the capability to roll back to a prior, known-good state. Btrfs includes checksum functionality to ensure data integrity, as well as transparent compression to save space. Btrfs includes integrated logical volume management operations making it easy to add and remove capacity and to use different RAID levels. Btrfs is discussed further in the lesson titled “Implementing the Btrfs File System.”

See the following URL for a comparison between XFS and ext4 and Btrfs: https://lwn.net/Articles/476263/.

The following URL discusses all of the major file systems: http://en.wikipedia.org/wiki/Comparison_of_file_systems.

Making ext File Systems

- Use the `mkfs` command to build a Linux file system:

```
# mkfs [options] <device>
```

- The `mkfs` command is a wrapper for other utilities:
  - `mkfs.ext2`
  - `mkfs.ext3`
  - `mkfs.ext4`
- Defaults parameters are specified in `/etc/mke2fs.conf`.
- Use the `blkid` command to display block device attributes.
- Use the `e2label` command to display and modify the file system label.

The command to build a Linux file system on a device, or hard disk partition, is `mkfs`. The syntax for the command is:

```
mkfs [options] device
```

The `mkfs` command is actually a front end for the different file system builder utilities such as `mkfs.ext2` and `mkfs.ext4`. These utilities are executable directly from the command line. When using the `mkfs` wrapper, include the `-t fstype` option to specify the type of file system to be built. If not specified, the default file system type, ext2, is created.

To see which supported file system types are installed, use the `ls /sbin/mk*` command:

```
# ls /sbin/mk*
mkdict     mkfs.btrfs   mkfs.fat    mkhomedir_helper
mkdosfs    mkfs.cramfs  mkfs.minux  mklost+found
mkdumprd   mkfs.ext2    mkfs.msdos  mkswap
mke2fs     mkfs.ext3    mkfs.vfat
mkfs       mkfs.ext4    mkfs.xfs
```

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

Using `mkfs`

The default file system type created when using the `mkfs` command is ext2. As previously mentioned, `mkfs` is a wrapper that calls other file system build utilities. Therefore, any of the following commands create an ext2 file system on the specified device:

```
# mkfs /dev/xvdd1
# mke2fs /dev/xvdd1
# mkfs.ext2 /dev/xvdd1
```

To create an ext3 file system, use any of the following commands:

```
# mkfs –t ext3 /dev/xvdd1
# mke2fs –t ext3 /dev/xvdd1
# mkfs.ext3 /dev/xvdd1
```

To create an ext4 file system, use any of the following commands:

```
# mkfs –t ext4 /dev/xvdd1
# mke2fs –t ext4 /dev/xvdd1
# mkfs.ext4 /dev/xvdd1
```

Configuration File

A number of options are available to customize block size, fragment size, blocks per group, journal options, number of inodes, and other parameters. Without including any options, the defaults that are specified in the `/etc/mke2fs.conf` configuration file are used.

File System Labels

A useful option for the file system build utilities is the `-L name` option. This assigns a label to the partition; this label can be used instead of the device name when mounting the file system. Labels are limited to a maximum size of 16 characters. For existing file systems, the `e2label` command is used to display or set a label.

File systems are automatically assigned a universally unique identifier (UUID). UUIDs can be used when mounting the file system. To display the UUID, the label, and the file system type, use the `blkid` command. The following examples illustrate creating different file systems, with and without a label, and displaying the information with the `blkid` command (only a partial UUID is displayed). To create an `ext2` file system and display information, enter:

```
# mkfs /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: UUID=:f6f32f...” TYPE=“ext2”
```

To create an ext3 file system and display information, enter:

```
# mkfs –t ext3 /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: UUID=“f6f32f...” SEC_TYPE=“ext2” TYPE=“ext3”
```

To create an ext4 file system, assign a label name, and display information, enter:

```
# mkfs –t ext4 –L ProjectA /dev/xvdd1
# blkid /dev/xvdd1
/dev/xvdd1: LABEL=“ProjectA” UUID=“f6f32f...” TYPE=“ext4”
PARTUUID=”...”
```

Mounting File Systems

- Use the `mount` command to attach a device to the directory hierarchy:

```
# mount [option] <device> <mount_point>
```

- Use the device name, the UUID, or the label:

```
# mount /dev/xvdd1 /test
# mount UUID="uuid_number" /test
# mount LABEL="label_name" /test
```

- Use the `-o` flag to specify mount options:

```
# mount -o nouser,ro /dev/xvdd1 /test
```

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

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


File systems on different partitions and removable devices, such as CDs, DVDs, or USB flash drives, must be attached to the directory hierarchy to be accessed. To attach a partition or device, a mount point must be created. A mount point is simply a directory created with the `mkdir` command. After a directory, or mount point, is created, attach the partition by using the `mount` command. Syntax for the mount command is:

```
mount [options] device_file mount_point
```

The following example creates a mount point (`/test`) and attaches the partition:

```
# mkdir /test
# mount /dev/xvdd1 /test
```

Alternatively, mount the partition or device by referencing the UUID or label. The following example displays the UUID and label, using the `blkid` command, and mounts the partition by referencing each (partial UUID is used):

```
# blkid /dev/xvdd1
/dev/xvdd1: LABEL="ProjectA" UID="9d7ab..." TYPE="ext4"
# mount LABEL="ProjectA" /test
# mount UUID="9d7ab..." /test
```

The `mount` command without any options displays all currently attached file systems:

```
# mount
/dev/xvdd1 on /test type ext4 (rw)
...
```

In this example, the `/dev/xvdd1` partition is mounted on `/test`. The file system type is ext4 and is mounted for both reading and writing.

The `df` command also displays mounted file systems. Example:

```
# df -h
Filesystem Size  Used  Avail  Use%  Mounted on
/dev/xvdd1 5.0G  139M   4.6G    3%  /test
...
```

The information in the `proc` file system displays mounted file systems. Example:

```
# cat /proc/mounts
...
```

Mount Options

To specify mount options, use the `-o` flag followed by a comma-separated string of options. The following are some of the available options for the `mount` command:

- **`auto`**: Allows the file system to be mounted automatically by using the `mount -a` command
- **`loop`**: Mounts the image as a loop device
- **`noauto`**: Disallows the automatic mount of the file system by using the `mount -a` command
- **`noexec`**: Disallows the execution of binary files on the file system
- **`nouser`**: Disallows an ordinary user (other than `root`) to mount and unmount the file system
- **`remount`**: Remounts the file system in case it is already mounted
- **`ro`**: Mounts the file system for reading only
- **`rw`**: Mounts the file system for both reading and writing
- **`user`**: Allows an ordinary user (other than `root`) to mount and unmount the file system

For example, to mount the `/dev/xvdd1` partition on the `/test` mount point as read-only with only the `root` user able to mount and unmount the file system, enter:

```
# mount -o nouser,ro /dev/xvdd1 /test
```

To mount an ISO image by using the `loop` device (assuming that the ISO image is present in the current directory and the mount point exist), enter:

```
# mount -o ro,loop OracleLinux-R6-U1-Server-x86_64-dvd.iso
/media/cdrom
```

Journaling Mount Options

The ext3 and ext4 file systems have three journaling levels that can be set with the `-o` option in the `mount` command or in the options section of `/etc/fstab`:

- **`data=journal`**: The highest level. The one that does the most journaling. This writes the journal entries for all the data and metadata changes. All data is committed into the journal before being written into the main file system.

- **`data=ordered`:** The default mode. All data is forced directly out to the main file system before its metadata is committed to the journal.
- **`data=writeback`:** The lowest level. Data ordering is not preserved. Data can be written into the main file system after its metadata has been committed to the journal.

Unmounting File Systems

To unmount a file system, use the `umount` command. The partition name, the device name, or the mount point is used as an argument. Example:

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

`/etc/fstab` File

- The `/etc/fstab` file is the file system mount table, and it:
  - Contains all the information needed by the `mount` command
  - Is read at boot time
- When creating a new file system, add a new entry to the file system mount table in the following format:

```
/dev/xvda3   /boot   ext4   defaults   0   0
```

  - The first column is the device to mount.
  - The second column is the mount point.
  - The third column is the file system type.
  - The fourth column specifies mount options.
  - The fifth column is used by the `dump` command.
  - The last column is used by the `fsck` command.

ORACLE

The `/etc/fstab` file is called the file system mount table and contains all the information that the `mount` command needs to mount devices. When adding a new file system, create the appropriate entry in `/etc/fstab` to ensure that the file system is mounted at boot time. The following is an example of entries in the `/etc/fstab` file:

```
# cat /etc/fstab
UUID=...     /       ext4   defaults   1   1
UUID=...     /boot   ext4   defaults   1   2
UUID=...     /home   ext4   defaults   1   2
UUID=...     swap    swap   defaults   0   0
```

The first column is the device to mount. The UUID or the label name should be used in place of the device name, because device names could change. The second column is the mount point, except the `swap` partition entry. Swap is discussed later. The third column is the file system type. The fourth column specifies mount options. The fifth column is used by the `dump` command. The number 1 means to dump the file system and 0 means the file system does not need to be dumped. The last column is used by the `fsck` program to determine the order in which file system checks are done at reboot time. The root file system should be specified with a value of 1 and the other file systems should have a value of 2. A value of 0 does not check the file system.

Maintaining File Systems

- The `fsck` command checks and repairs Linux file systems.
  - `fsck` runs at boot time based on configurable parameters.
  - Do not run `fsck` on mounted file systems.
- Use the `tune2fs` command to:
  - Configure the frequency of file system checks
  - Convert ext2 file systems to ext3
  - Adjust file system parameters on ext2, ext3, and ext4 file systems
  - Display current file system parameter values
- Use the `dumpe2fs` utility to print file system information.
- The `debugfs` utility is an interactive file system debugger.

The best tool for maintaining file systems is `fsck`, which checks and repairs Linux file systems. By default, `fsck` runs after 20 system reboots but should be run manually if your system runs for weeks or months with rebooting. The frequency of file system checks is changed by using the `tune2fs` command. Other utilities for performing file system maintenance include `dumpe2fs` and `debugfs`. The `dumpe2fs` utility prints the super block and blocks group information for the file system on the specified device. The `debugfs` utility is an interactive file system debugger.

Using the `fsck` Command

The `fsck` command accepts a device name, a mount point, a UUID, or a file system label as an argument. If no argument is given, `fsck` checks all file systems listed in `/etc/fstab`. Do not run `fsck` on mounted file systems, because it causes severe file system damage. To unmount the file system and run the `fsck` utility on `/dev/xvdd1`:

```
# umount /dev/xvdd1
# fsck /dev/xvdd1
fsck from util-linux 2.23.2
E2fsck 1.42.9 )28-Dec-2013)
Test: clean, 11/65536 files, 8859/262139 blocks
```

Notice that the `fsck` utility calls the `e2fsck` utility to check the file system. File system–specific commands are located in `/sbin`:

```
# ls /sbin/*fsck*
btrfsck  fsck.cramfs  fsck.fat   fsck.xfs
dosfsck  fsck.ext2    fsck.minix
e2fsck   fsck.ext3    fsck.msdos
fsck     fsck.ext4    fsck.vfat
```

If the file system is corrupted, you are prompted to respond to a series of questions during repair attempts. You can include the `-y` option to use "yes" as an answer to all questions. Additional options to `fsck` are given:

- **`-s`:** Serialize `fsck` operations. This is a good idea if you are checking multiple file systems and the checkers are in an interactive mode.
- **`-A`:** Walk through the `/etc/fstab` file and try to check all file systems in one run. This option is typically used from the `/etc/rc` system initialization file. The `root` file system is checked first. After that, file systems are checked in the order specified by the sixth field in the `/etc/fstab` file. File systems with a value of `0` in this field are skipped and are not checked.
- **`-R`:** When checking all file systems with the `-A` flag, skip the `root` file system (in case it is already mounted read-write).

Using `tune2fs`

The `tune2fs` utility is mainly used to set file system check options, and to convert an ext2 file system to ext3. You should always use the `e2fsck` utility before and after using `tune2fs`. To convert an ext2 file system to ext3, enter:

```
# tune2fs -j block_device
```

The *`block_device`* argument contains the ext2 file system that you want to convert. The `-j` option adds an ext3 journal to the file system.

The most commonly used options to `tune2fs` are:

- **`-c max-mount-counts`:** Adjust the maximum mount count between two file system checks.
- **`-C mount-count`:** Set the number of times the file system has been mounted.
- **`-i interval-between-checks[d|m|w]`:** Adjust the maximum time between two file system checks.
- **`-m reserved-blocks-percentage`:** Set the percentage of reserved file system blocks.
- **`-r reserved-blocks-count`:** Set the number of reserved file system blocks.

Use the `tune2fs` command to adjust various tunable file system parameters on ext2, ext3, and ext4 file systems. Current values are displayed by using the `-l` option. Example:

```
# tune2fs –l /dev/xvda1
```

Alternatively, use the `dumpe2fs` command to display file system parameters:

```
# dumpe2fs /dev/xvda1
```

Swap Space

- Swap space is used when there is insufficient RAM.
- Swap space is a partition, a file, or both.
- Use `fdisk`, `cfdisk`, or `parted` to create a swap partition.
- Use `dd` to create a swap file:

```
# dd if=/dev/zero of=/swapfile bs=1024 count=1000000
```

- Use `mkswap` to initialize a swap partition or file:

```
# mkswap {device|file}
```

- Use `swapon` and `swapoff` to enable and disable devices and files for swapping, respectively:

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

ORACLE

Swap space is used in Linux when there is insufficient physical memory (RAM) on your system to store the data currently being processed. When your system needs more memory, inactive memory pages are written to the disk, freeing up physical memory. Increasing swap space should not be considered as a solution to memory shortages. Swap space is located on disk drives, which have slower access times than physical memory. If your system is swapping often, you should add more physical memory, not more swap space.

Swap space in Linux is either a normal file in the file system, called a swap file, or a separate partition, or a combination of swap partitions and swap files. A dedicated swap partition is much faster, but it is easier to change the size of a swap file. If you know how much swap space you need, use a swap partition. If you are unsure, experiment with a swap file first, then make a swap partition when you know your requirements.

The swap partition is listed in the partition table, referenced in `/etc/fstab`, and viewable in the `/proc/swaps` file. There are also command-line utilities to display information about your swap space. To view the swap partition in the partition table, enter:

```
# fdisk -l | grep swap
/dev/xvda5   17412096   25165823   3876864   82   Linux swap / ...
```

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

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

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

```
# cat /proc/swaps
Filename     Type        Size      Used  Priority
/dev/xvda5  partition  3876860  0     -1
```

Swap Utilities

The `mkswap` command is used to initialize a swap partition or a swap file. The syntax is:

```
mkswap {device|file}
```

The `swapon` and `swapoff` utilities enable and disable devices and files for swapping, respectively. To display current swap information, use the `swapon -s` command. Output is identical to viewing the contents of `/proc/swaps`.

Adding Swap Space

The swap partition or swap file must exist before it is initialized. Use `fdisk` or `parted` to create a swap partition. A swap file is created by using the `dd` command. Example:

```
# dd if=/dev/zero of=/swapfile bs=1024 count=1000000
```

To initialize a swap partition, type:

```
# mkswap /dev/xvdd1
```

To initialize a swap file, type:

```
# mkswap /swapfile
```

Initialized swap space is enabled by using the `swapon` command. To enable swapping on a swap file, enter:

```
# swapon /swapfile
```

To enable swapping on a swap partition, enter:

```
# swapon /dev/xvda3
```

Update the `/etc/fstab` file to enable the swap partition or swap file at boot:

```
UUID=...     swap   swap   defaults   0   0
/swapfile    swap   swap   defaults   0   0
```

Viewing Swap Usage

View the `/proc/meminfo` file, or use other utilities such as `free`, `top`, and `vmstat` to view memory and swap space usage. Example:

```
# grep -i swap /proc/meminfo
SwapCached:         0 kB
SwapTotal:    3876860 kB
SwapFree:     3876860 kB
```

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

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

Quiz

Which of the following is the file system mount table for Oracle Linux?

a. `/etc/vfstab`

b. `/etc/filesystem`

c. `/etc/fstab`

d. `/boot/filesystem`

Quiz

Which of the following statements is true?

a. Use the `umount` command to unmount file systems.
b. Use the `swapoff` command to disable swap space.
c. File systems must be unmounted and swap partitions must be disabled before repartitioning a disk drive.
d. Do not run `fsck` on mounted file systems.
e. All of the above

Summary

In this lesson, you should have learned how to:

- Describe disk partitioning
- Use disk partitioning utilities
- Describe supported file system types
- Describe file system creation, mounting, and maintenance
- Describe and maintain swap space

Practice 11: Overview

The practices for this lesson cover the following:

- Listing current disk partitions
- Partitioning a storage device
- Creating `ext3` and `ext4` file systems
- Increasing swap space

12

XFS File System

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

XFS File System

- XFS is a high-performance journaling file system.
- XFS is available for Oracle Linux x86_64 architecture.
  – Requires UEK R2 or later UEK
- XFS is the default file system for Oracle Linux 7.
- It supports a maximum file size of 16 TB.
- It supports a maximum file system size of 500 TB.
- XFS can be created on a regular disk partition and on a logical volume.
- XFS supports extended attributes.
  – Used by Access Control Lists (ACL) and SELinux
- XFS supports user, group, and project disk quotas.

The XFS file system is a high-performance journaling file system. XFS in Oracle Linux is available for the x86_64 architecture and requires the Unbreakable Enterprise Kernel Release 2 (`2.6.39`) or higher. XFS is the default file system for Oracle Linux 7. XFS supports a maximum file system size of 500 TB and a maximum file size of 16 TB. 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 file system metadata while the file system is running until those changes are written to the data section. XFS journaling guarantees the 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.

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, agzise=...
         =             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.

ORACLE

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/xvdb1` partition. As shown in the slide, parameters for the file system are displayed as output.

```
# mkfs.xfs /dev/xvdb1
meta-data=/dev/xvdd1   isize=256  agcount=4, agzise=327680 blks
         =             sectsz=512 attr=2, projid32bit=1
...
```

The next 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 next 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 next 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, agzise=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 to 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.

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 update when other counters are changed.

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 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.

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 the following:

- **`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 the following:

- **`pquota|prjquota`**: Enable project quotas and enforce usage limits.
- **`pqnoenforce`**: Enable project quotas. Report usage but do not enforce usage limits.

Report 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
```


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
```


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 the project name `test` with the 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
```


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 are 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 re-organizing 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.

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 in Oracle Linux supports a maximum file system size of 500 TB and a maximum file size of 16 TB.

d. XFS is supported for use with the root (`/`) and `/boot` file systems.

e. All of the above

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 number of files.
d. Use the `xfs_quota` command to report file system quota information.
e. All of the above

Quiz

Which of the following statements are true?

a. The `xfsdump` and `xfsrestore` utilities can be used to access 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

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

Practice 12: Overview

The practices for this lesson cover the following:

- Creating an XFS file system
- Setting disk quotas on an XFS file system
- Backing up and restoring an XFS file system