ORACLE University

Integrated Cloud Applications & Platform Services

Oracle Linux System Administration I

Student Guide - Volume I

D103151GC10

Edition 1.0 | February 2019 | D106141

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

ORACLE

Author

Eric Craig

Technical Contributors and Reviewers

Michael Oreilly
Antoinette O'Sullivan
Craig McBride
Michael O'Reilly
Dave Goff
Steve Miller
Rowan Puttergill
Nita Heieck
Craig Carl
Wim Coekaerts
Sergio Leunissen
Keshav Sharma
Hanlin Chien
Jim Williams
Ankur Kemkar
Jared Greenwald
Lawrence Gabriel
Tim Caster
Steven B. Nelson
Harish Niddagatta
Sebastien Colas
Matt Slingsby
Al Flournoy
Jamie Iles
Simon Coter
Avi Miller
John Haxby
Saar Maoz
Todd Vierling
Sreejith Mohan
Paul Kasewurm
Corey Leong

Graphic Designer

Yogita Chawdhary

Editors

Raj Kumar
Moushmi Mukherjee

Publishers

Giri Venugopal
Sumesh Koshy
Raghunath M

Contents

1

Course Introduction

Course Objectives

After completing this course, you should be able to:

- Describe Oracle's contributions and commitment to Linux
- Describe Oracle Cloud Infrastructure features and functions
- Install and complete initial setup of Oracle Linux
- Describe and configure the Unbreakable Enterprise Kernel (UEK) for Oracle Linux
- Describe Oracle Ksplice for updates without rebooting
- Configure users, storage, and network interfaces on Oracle Linux
- Describe the preparation of the Oracle Linux server for installation of Oracle Database
- Monitor and troubleshoot Oracle Linux

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

You will learn about Oracle Cloud Infrastructure, including how to create an instance, using SSH key pairs for instance access, adding storage to an instance, and network setup with instances.

You will install Oracle Linux and perform system administration tasks that can be engaged in after an installation. You will be introduced to the Unbreakable Enterprise Kernel and explore kernel configuration. You learn about Oracle Ksplice for kernel and select user space updates without rebooting. You will configure users, storage, and network interfaces and learn how to monitor and troubleshoot your systems to ensure successful implementation.

.

Course Schedule

Session	Module
Day 1	Lesson 1: Course Introduction Lesson 2: Introduction to Oracle Linux Lesson 3: Oracle Cloud Computing Lesson 4: Installing Oracle Linux 7
Day 2	Lesson 5: Oracle Linux 7 Boot Process Lesson 6: System Configuration Lesson 7: Package Management Lesson 8: Automating Tasks

Course Schedule

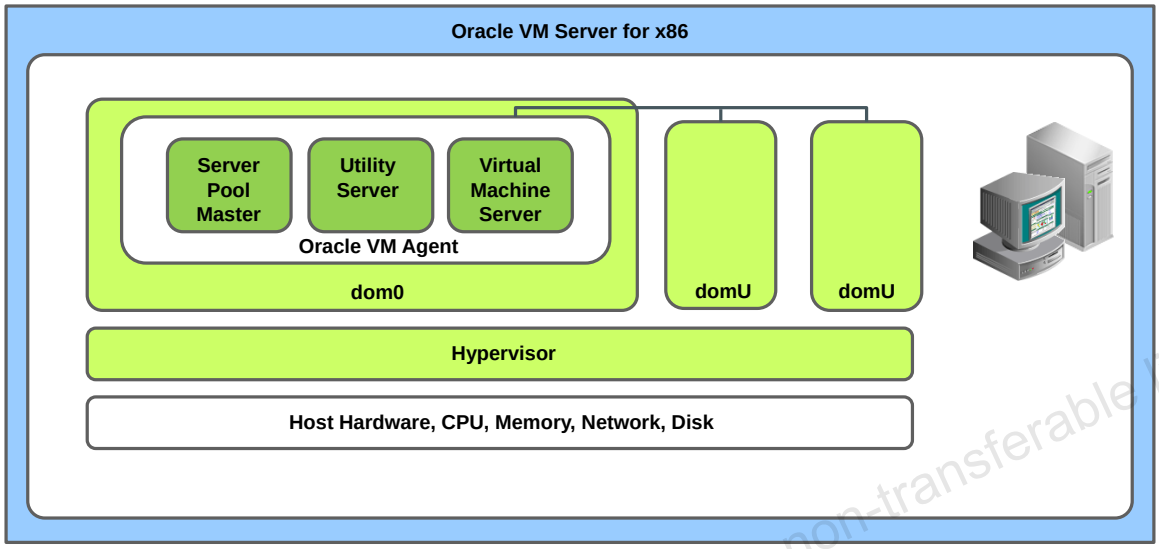
Session	Module
Day 3	Lesson 9: Kernel Module Configuration Lesson 10: Oracle Ksplice Lesson 11: User and Group Administration Lesson 12: Partitions, File Systems, and Swap
Day 4	Lesson 13: Network Configuration Lesson 14: IPv6 Lesson 15: OpenSSH Lesson 16: Security Administration
Day 5	Lesson 17: Oracle on Oracle Lesson 18: System Monitoring and Management Lesson 19: System Logging Lesson 20: Troubleshooting

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

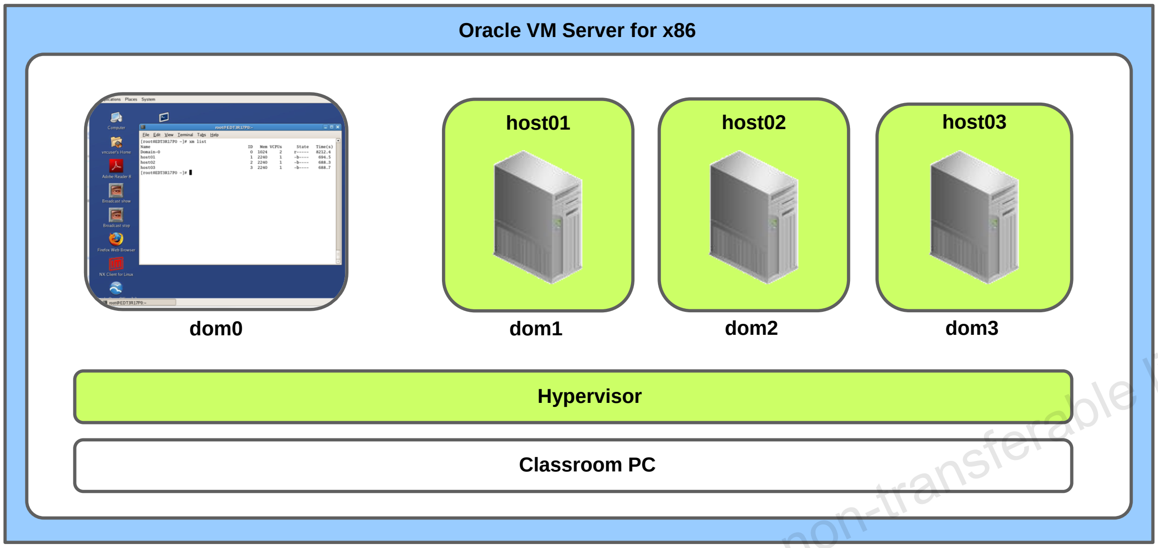
A hypervisor is a virtualization software, also known as a virtual machine monitor (VMM), that creates and runs the virtual machines. There are two different types of hypervisors:

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

Oracle VM Server for x86 Domains

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

Oracle VM Server for x86 in the Classroom



Self-Contained Multihost 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 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>`: Not graceful shut down
  - `xm reset <VM_name>`: Resets the specified VM
- The following commands connect to VM guests:
  - `ssh <VM_name>`
  - `ssh -X <VM_name>`
  - `xm  <VM_name>&`
- 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` 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 and Running GUI Utilities

When you need to run GUI utilities, the practice exercises direct you to use `xm vncviewer <VM name>&` to connect from dom0 to your guests. The "&" character causes the command to run in the background, giving the foreground prompt back. Example:

```
# xm vncviewer host01&
```

Alternatively, use secure shell to create a connection as the `root` user from dom0 to your guests. You can also include the `-X` option, which enables X11 forwarding. Examples:

```
# ssh root@host01
# ssh -X root@host01
```

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

Practice 1: Overview

This practice covers the following topics:

- Exploring the dom0 environment *
- Starting, stopping, and listing guests
- Exploring the host01 VM
- Exploring the host02 VM
- Logging off from your student PC

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

2

Introduction to Oracle Linux

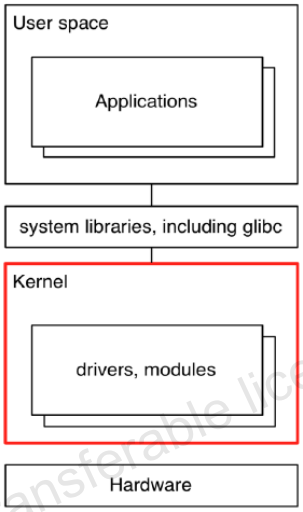
Objectives

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)
- The Unbreakable Enterprise Kernel
- Dtrace and Btrfs

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.

ORACLE

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.
- 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. The 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 two to 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 two to 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.

Several 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 ER (Enhancement Request) through My Oracle Support (MOS) at https://support.oracle.com. A Service Request (SR) must be created containing a detailed explanation of why the feature is being requested. Not all ERs get implemented. This might be because an ER is not in line with Oracle's future product direction or because the Enhancement is too specific to the given customer environment and would be better handled as a customization.

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 ERs 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/git/. All kernel changes are pushed to this Git repository. For example, Oracle's public GIT repository for kernel version 4.1 is available at https://oss.oracle.com/git/gitweb.cgi?p=linux-uek.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 7 is supported on x86-64 (64 bit) hardware architectures.

ORACLE

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 regarding 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 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 the x86 (32 bit) architecture.

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
- Examples:
  - 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. Examples of 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 the Oracle RPMs as well as errata released in between installation DVDs are also available via http://yum.oracle.com.

Some of the different groups of packages for Oracle Linux are:

- 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
- Oracle Ksplice: Apply kernel and select user space library updates on a running system

Oracle offers a comprehensive Linux solution including:

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

Oracle Ksplice allows you to apply kernel and select user space library updates without having to reboot the system. Updates come from either Oracle or the Linux community. These go into Oracle Linux releases, are tested by Oracle, and then are turned into Ksplice binary updates from Oracle Linux packages. You apply updates using Ksplice tools and the patches are 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 File System (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 manifested 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 12*c*, there is a new ASM filter driver (ASMFD) that prevents accidental corruption or deletion of the ASM devices. For more information, refer to: 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 File System (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. For more information, refer to: 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) benefits 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 nonfunctional 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 several 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.

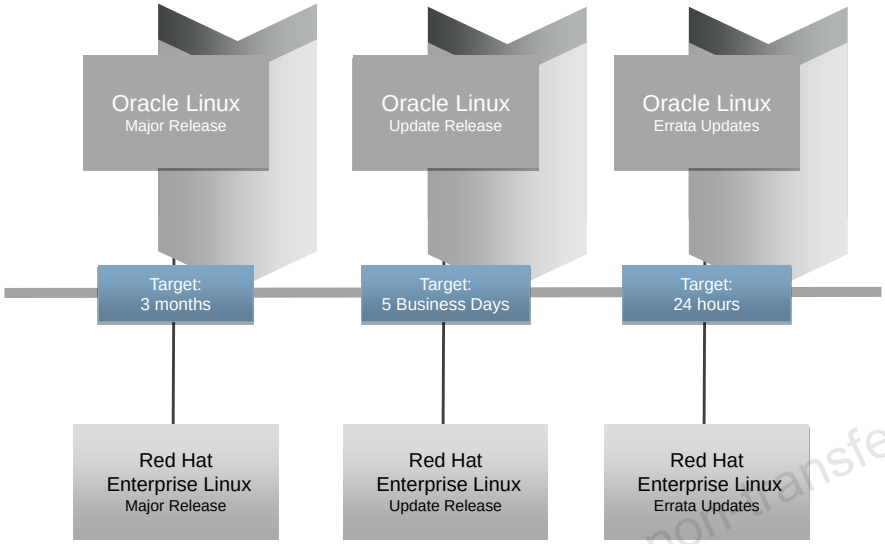
If a package has trademarks or Oracle Linux–specific changes, Oracle examines the source code and compares it against the bug fixes that have been already applied and released as part of Oracle Linux. If the Oracle patches are still relevant, they are re-applied, but if the problems have been fixed in 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://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.

Oracle Linux Release/Update Targets
Relative to Red Hat Enterprise Linux (RHEL) Releases/Updates



To ensure that the Oracle Linux distribution remains binary compatible with Red Hat, Oracle has developed business processes to ensure that as new releases/updates are made available by Red Hat, the binary compatible releases/updates for Oracle Linux will shortly follow. The slide shows the timelines we have for this.

The Unbreakable Enterprise Kernel (UEK)

- 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 Update 5.
- Since Oracle Linux 5 Update 5, you have a choice:
  - Red Hat Compatible Kernel
  - The 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 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 Update 5, 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 Update 6, the Unbreakable Enterprise Kernel became the default kernel. The Unbreakable Enterprise Kernel Release 4 (UEK R4) is a kernel release for Oracle Linux 6 Update 7 or later and Oracle Linux 7 Update 1 or later on the x86-64 architecture. It is based on the mainline Linux kernel version 4.1.12. The Unbreakable Enterprise Kernel Release 5 (UEK R5) is a kernel release for Oracle Linux 7 Update 5 and later on the x86-64 and 64-bit ARM (aarch64) architectures. It is based on the mainline Linux kernel version 4.14.35.

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.

The 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`.

Release notes for different kernel versions can be viewed here:
https://docs.oracle.com/cd/E93554_01/index.html

DTrace

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

ORACLE

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

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

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

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

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

Btrfs File System

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

Btrfs is a general-purpose file system that scales very well for large systems and large file systems.

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

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

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

A UEK Boot ISO image is available that boots up the Unbreakable Enterprise Kernel as the install kernel and uses Btrfs as the default file system for installation.

Oracle Linux Release Notes

- Oracle publishes release notes for each version.
- Release notes and all product documentation for Oracle Linux 7 are available at: http://docs.oracle.com/cd/E52668_01/.
- New features and changes
- 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

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” from the initial release of Oracle Linux 7:

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 recommended 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

Systemwide 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 nonpersistent 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.

The lesson titled “Installing Oracle Linux 7” describes requirements for upgrading from Oracle Linux 6.

Quiz

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

a. True
b. False

Answer: a

Quiz

Which of the following statements are true about the Unbreakable Enterprise Kernel (UEK)?

a. The UEK is the default kernel since Oracle Linux 5 Update 6.
b. The UEK is heavily tested with Oracle workloads.
c. Existing applications run unchanged when using the UEK.
d. The UEK is the only kernel available with Oracle Linux.

Answer: a, b, c

The Red Hat Compatible Kernel (RHCK) is also included with Oracle Linux.

Summary

In this lesson, you should have learned how 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)
- The Unbreakable Enterprise Kernel
- Dtrace and Btrfs

Practice 2: Overview

This practice covers the following topics:

- Quiz – Introduction to Oracle Linux
- Viewing kernel information

3

Oracle Cloud Computing

Objectives

After completing this lesson, you should be able to:

- Define Infrastructure as a Service (IaaS)
- Describe Oracle Private Cloud Appliance
- Describe Oracle OpenStack
- List some Oracle Cloud Infrastructure services
- Describe some key concepts and terms used in Oracle Cloud Infrastructure
- List some Oracle-provided images and available shapes in Oracle Cloud Infrastructure
- Describe the task flow to launch an Oracle Cloud Infrastructure instance
- Describe creation of an Oracle Cloud Infrastructure Virtual Cloud Network and subnet
- Describe oci-utils
- Describe attachment of an Oracle Cloud Infrastructure block volume to an instance

ORACLE

There are many Oracle cloud solutions and combinations to suit different customer needs:

- A Private Cloud solution (for example: Oracle Private Cloud Appliance) or building a solution with Oracle OpenStack
- An Oracle Cloud Infrastructure solution
- A hybrid solution

This lesson provides an introduction to each of these Oracle cloud solutions.

What Is Infrastructure as a Service (IaaS)?

- It is a form of cloud computing that provides infrastructure services over the internet.
- It is required by all applications, databases, and middleware deployments.
- Users can access:
  - Computer processors
  - Storage
  - Networks
  - Other infrastructure resources
- IaaS is one of the three main “service models” in cloud computing. The other two are:
  - SaaS: Software as a Service
  - PaaS: Platform as a Service

Infrastructure as a Service (IaaS) is a form of cloud computing that provides infrastructure services over the Internet. IaaS provides the following benefits:

- The user has access to the foundation level of the computing “task”—compute capacity, network bandwidth, and storage capacity.
- The IaaS provider is responsible for providing the hardware, maintenance, and cloud infrastructure up to the virtualization level.
- The user can create virtual machines as required and decide what operating system and/or apps to use.

The other two service models in cloud computing are:

- **Software as a Service (SaaS):** The user has access only to the applications provided by the SaaS provider. The provider has full control of the entire infrastructure, including applications.
  - Example: An email provider
- **Platform as a Service (PaaS):** The user is able to deploy applications using programs and tools provided by the provider. The provider has full control of the infrastructure up to the point where users are able to modify their applications and environment.
  - Example: A web hosting provider that offers tools and programs to deploy a website

One of the main features of these cloud computing service models is that users can access their applications and data from anywhere, as long as they can access the Internet.

What Is Oracle Private Cloud Appliance?

- It is preconfigured for stability, high availability, and automation.
- Orchestration software automatically handles new server hardware.
- It supports the provisioning of Infrastructure and Platform as a Service (IaaS and PaaS) on demand.
- Private Cloud Appliance Virtualization
  - Focus on services, not managing hardware
  - Created to handle demanding workloads
  - Unified support for the software solution stack
- Private Cloud Appliance simplifies application deployment and management.

ORACLE

Oracle Private Cloud Appliance, shown in the slide, is a converged infrastructure appliance. Because the servers, switches, and storage are preconfigured, Private Cloud Appliance is ready to run almost immediately after installation.

The hardware is preconfigured at the factory for high availability and automation. This minimizes the risk of misconfiguration later on. Insert additional servers and attach them to the preconfigured cables to connect them to the network. Orchestration software automatically detects and integrates new servers to increase processor and memory capacity for virtualization.

The Private Cloud Appliance environment supports the provisioning of Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) among other services. With templates and assemblies, you can rapidly meet demands for applications without having to consider additional network and storage capacity, or cabling and power, to host them.

Virtualization enables you to provide application platforms and infrastructure as services. You manage policies to provide services on demand, on a schedule, and for a particular length of time. You manage hardware to maintain the virtualization environment and not in response to application demands.

The Private Cloud Appliance virtualization environment has been created with significant capacity for demanding workloads. Private Cloud Appliance is built with Oracle hardware and software. Because Private Cloud Appliance is a unified solution, Oracle supports the entire system and its software. This approach separates the concerns of managing hardware in the data center from the demands of rapidly changing applications. The result is simplified application deployment and management. For more information about Private Cloud Appliance, go to http://www.oracle.com/servers/private-cloud-appliance or take the *Oracle Private Cloud Appliance Administration* course at http://education.oracle.com/engineeredsystems.

What Is Oracle OpenStack?

- Oracle OpenStack is a cloud management software for managing large pools of compute, storage, and networking resources.
- Based on an OpenStack community release, Oracle OpenStack is an enterprise-grade solution for managing an entire IT environment.
- It rapidly deploys Oracle and third-party applications across shared compute, network, and storage resources.
- It accelerates application deployment with self-service VM creation.

Oracle OpenStack offers the ability to deploy different deployment configurations depending on the demand. As an IaaS cloud management platform, it centralizes enterprises' cloud operations for OpenStack operators to manage and deploy resources within an IT network environment.

Oracle OpenStack is a software-defined network (SDN) solution for deploying VMs and virtual networks abstracted from the physical infrastructure of a data center. Optimized for Oracle Linux, Oracle OpenStack securely streamlines the deployment of compute resources while decreasing the time to deploy applications.

To learn more, go to http://www.oracle.com/openstack or take the *Oracle OpenStack for Oracle Linux Getting Started* course at https://education.oracle.com/s-oracle-openstack-for-oracle-linux-getting-started-ed-1/courP_7638.

What Are Oracle Cloud Infrastructure Services?

A set of complementary cloud services that enable you to build and run a wide range of applications and services in a highly-available hosted environment. Examples include:

- **Compute Service:** Provision and manage bare metal compute instances or virtual machine instances.
- **Networking Service:** Create and manage the network components for your cloud resources.
- **Identity and Access Management (IAM) Service:** Control access to cloud resources.
- **Load Balancing Service:** Create a public or private load balancer within your virtual cloud network (VCN).

Oracle Cloud Infrastructure is a set of complementary cloud services that enable you to build and run a wide range of applications and services in a highly-available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

The **Compute Service** lets you provision and manage compute hosts, known as instances. You can launch instances as needed to meet your compute and application requirements. After you launch an instance, you can access it securely from your computer, restart it, attach and detach volumes, and terminate it when you're done with it. Any changes made to the instance's local drives are lost when you terminate it. Any saved changes to volumes attached to the instance are retained.

Use the **Networking Service** to create and manage the network components for your cloud resources. You can configure your virtual cloud network (VCN) with access rules and gateways to support routing of public and private Internet traffic.

Use the **Identity and Access Management (IAM) Service** to control who has access to your cloud resources. You can control what type of access a group of users and/or instances have and to which specific resources.

The **Load Balancing Service** provides automated traffic distribution from one entry point to multiple servers reachable from your virtual cloud network (VCN). The service offers a load balancer with your choice of a public or private IP address, and provisioned bandwidth.

What Are Oracle Cloud Infrastructure Services?

- **Block Volume Service:** Dynamically provision and manage block storage volumes.
- **Object Storage Service:** Manage data as objects stored in logical containers called buckets. There are two storage class tiers:
  - Standard Object Storage (for data needing fast, immediate frequent access)
  - Archive Storage (for data that is seldom or rarely accessed)
- **File Storage Service:** Provide a durable, scalable, distributed, enterprise-grade network file system.
- **Database Service:** Provision a user-managed DB system or autonomous database.
- **Audit Service:** Track activity in your environment.

The **Block Volume Service** lets you dynamically provision and manage block storage volumes. You can create, attach, connect and move volumes as needed to meet your storage and application requirements. Once attached and connected to an instance, you can use a volume like a regular hard drive. Volumes can also be disconnected and attached to another instance without the loss of data.

The **Object Storage Service** offers two distinct storage class tiers to address the need for both performant, frequently accessed "hot" storage, and less frequently accessed "cold" storage. Storage tiers help you maximize performance where appropriate and minimize costs where possible. Use Standard Object Storage for data to which you need fast, immediate, and frequent access. Use Archive Storage for data to which you seldom or rarely access, but that must be retained and preserved for long periods of time.

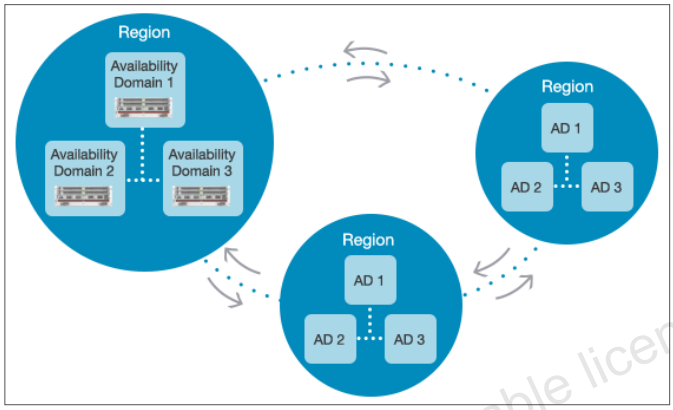
The **File Storage Service** provides a durable, scalable, distributed, enterprise-grade network file system. You can connect to a File Storage service file system from any bare metal, virtual machine, or container instance in your Virtual Cloud Network (VCN).

The **Database Service** offers autonomous and user-managed Oracle Database solutions. Autonomous databases are preconfigured, fully-managed environments that are suitable for either transaction processing or for data warehouse workloads. User-managed solutions are bare metal, virtual machine, and Exadata DB systems that you can customize with the resources and settings that meet your needs.

The **Audit Service** automatically records calls to all supported Oracle Cloud Infrastructure public application programming interface (API) endpoints as log events. Currently, all services support logging by Audit. Log events recorded by the Audit service include API calls made by the Oracle Cloud Infrastructure Console, Command Line Interface (CLI), Software Development Kits (SDK), your own custom clients, or other Oracle Cloud Infrastructure services.

Key Concepts and Terms

- Regions and Availability Domains
  - A region is a localized geographic area composed of several Availability Domains.
  - An Availability Domain is one or more data centers located within a region.
  - Availability Domains are isolated from each other, fault tolerant, and very unlikely to fail simultaneously.
  - A logical grouping
- Fault Domain
  - A logical grouping of hardware and infrastructure within an availability domain



The following section lists and describes key concepts and terms used in Oracle Cloud Infrastructure.

Regions and Availability Domains

Oracle Cloud Infrastructure is physically hosted in Regions and Availability Domains. A region is a localized geographic area, and an Availability Domain is one or more data centers located within a region. A region is composed of several Availability Domains. Oracle Cloud Infrastructure resources are either region-specific, such as a virtual cloud network, or availability domain-specific, such as a compute instance.

Availability domains are isolated from each other, fault tolerant, and very unlikely to fail simultaneously. When you configure your cloud services, use multiple availability domains to ensure high availability and to protect against resource failure. Be aware that some resources must be created within the same Availability Domain, such as an instance and the storage volume attached to it.

Fault Domain

A logical grouping of hardware and infrastructure within an availability domain to provide isolation of resources in case of hardware failure or unexpected software changes.

Key Concepts and Terms

- Tenancy
  - A tenancy is a secure and isolated partition within Oracle Cloud Infrastructure where you can create, organize, and administer your cloud resources.
  - Oracle creates a tenancy for your company when you sign up for Oracle Cloud Infrastructure.
  - Your tenancy is the root compartment that holds all your cloud resources.
- Compartments
  - Compartments allow you to organize and control access to your Oracle Cloud Infrastructure resources.
  - You can create additional compartments within the tenancy (root compartment).
  - When you create an Oracle Cloud Infrastructure resource, you must specify to which compartment you want the resource to belong.

Tenancy

A tenancy is a secure and isolated partition within Oracle Cloud Infrastructure where you can create, organize, and administer your cloud resources. When you sign up for Oracle Cloud Infrastructure, Oracle creates a tenancy for your company. A tenancy can be subscribed to multiple regions.

Compartments

Compartments allow you to organize and control access to your cloud resources. A compartment is a collection of related resources (such as instances, virtual cloud networks, block volumes) that can be accessed only by certain groups that have been given permission by an administrator. A compartment is to be thought of as a logical group, rather than a physical container. When you begin working with resources in the Console, the compartment acts as a filter for what you are viewing.

Your tenancy is the root compartment that holds all your cloud resources. You then create additional compartments within the tenancy (root compartment) and corresponding policies to control access to the resources in each compartment. When you create a cloud resource such as an instance, block volume, or cloud network, you must specify to which compartment you want the resource to belong.

Note: After you create a resource, you cannot move it to another compartment.

The correct use of compartments ensures that each user has access to only the resources they need.

Key Concepts and Terms

- Console
  - A web-based user interface
  - Provides access to manage Oracle Cloud Infrastructure
  - A navigation menu lets you access service pages to create, manage, and view cloud resources
  - A Help menu provides links to documentation and Oracle Support
- Tags
  - Defined keys and values that are associated with resources

Console

The console is a web-based user interface you can use to access and manage Oracle Cloud Infrastructure.

Tags

Tagging allows you to define keys and values and associate them with resources. You can then use the tags to help you organize and list resources based on your business needs. Defined tags are set up in your tenancy by an administrator. Only users granted permission to work with the defined tags can apply them to resources. Free-form tags can be applied by any user with permissions on the resource.

Key Concepts and Terms

- Instance
  - An instance is a compute host running in the cloud.
  - Bare metal compute instances run on bare metal servers without a hypervisor.
    - You maintain sole control of the physical CPU, memory, and Network Interface Card.
    - You do not share the physical machine with any other tenants.
  - Managed Virtual Machine (VM) instances are also available for workloads that don't require dedicated physical servers or the high performance of bare metal instances.
- Image
  - The image is a template of a virtual hard drive that defines the operating system and other software for an instance (for example, Oracle Linux).
  - When you launch an instance, you define its characteristics by choosing its image.
- Shape
  - The shape specifies the number of CPUs and the amount of memory allocated to an instance, and specifies if the instance is a bare metal instance or a VM instance.

Instance

An instance is a compute host running in the cloud. An Oracle Cloud Infrastructure compute instance allows you to use hosted physical hardware, as opposed to the traditional software-based virtual machines, ensuring a high level of security and performance.

Image

The image is a template of a virtual hard drive that defines the operating system and other software for an instance (for example, Oracle Linux). When you launch an instance, you can define its characteristics by choosing its image. Oracle provides a set of images you can use. You can also save an image from an instance that you have already configured to use as a template to launch more instances with the same software and customizations.

Shape

In the Compute Service, the shape specifies the number of CPUs and the amount of memory allocated to an instance. The shape also specifies if the instance is a bare metal instance or a virtual machine instance.

Oracle-Provided Images

Image	Name	Description
Oracle Linux 7 Unbreakable Enterprise Kernel Release 4	Oracle-Linux-7.*x*-<*date*>-<*number*>	The Unbreakable Enterprise Kernel (UEK) is Oracle's optimized operating system kernel for demanding Oracle workloads. GPU shapes are supported with this image.
Oracle Linux 6 Unbreakable Enterprise Kernel Release 4	Oracle-Linux-6.*x*-<*date*>-<*number*>	The Unbreakable Enterprise Kernel (UEK) is Oracle's optimized operating system kernel for demanding Oracle workloads.
CentOS 7	CentOS-7-<*date*>-<*number*>	CentOS is a free, open-source Linux distribution suitable for use in enterprise cloud environments. For more information, see https://www.centos.org/.

This slide shows some of the available Oracle-provided images you can select when creating an Oracle Cloud Infrastructure instance.

For more images, see:

https://docs.cloud.oracle.com/iaas/Content/Compute/References/images.htm

Oracle-Provided Bare Metal Shapes

- Bare Metal Shapes

Shape	Instance Type	OCPU	Memory (GB)	Local Disk	Network Bandwidth[1]	Max VNICs Total: Linux	Max VNICS Total: Windows
BM.Standard2.52	X7-based standard compute capacity	52	768	Block storage only	2 x 25 Gbps	52 total (26 per physical NIC)	27 total (1 on the first physical NIC, 26 on the second)
BM.DenseO2.52	X7-based dense I/O compute capacity	52	768	51.2 TB NVMe SSD (8 drives)	2 x 25 Gbps	52 total (26 per physical NIC)	27 total (1 on the first physical NIC, 26 on the second)

This slide shows some of the available Oracle-provided bare metal shapes you can select when creating an Oracle Cloud Infrastructure instance.

For more shapes, see:

https://docs.cloud.oracle.com/iaas/Content/Compute/References/computeshapes.htm

Oracle-Provided VM Shapes

- VM Shapes

Shape	OCPU	Memory (GB)	Local Disk (TB)	Network Bandwidth[1]	Max VNICs Total: Linux	Max VNICs Total: Windows
VM.Standard2.1	1	15	Block Storage only	1 Gbps	2	2
VM.Standard2.2	2	30	Block Storage only	2 Gbps	2	2
VM.Standard2.4	4	60	Block Storage only	4.1 Gbps	4	4

This slide shows some of the available Oracle-provided VM shapes you can select when creating an Oracle Cloud Infrastructure instance.

For more shapes, see:

https://docs.cloud.oracle.com/iaas/Content/Compute/References/computeshapes.htm

Key Concepts and Terms

- Virtual Cloud Network (VCN):
  - Is a virtual version of a traditional network on which your instances run
  - Includes subnets, route tables, and gateways
  - Resides within a single region, but subnets can belong to multiple availability domains
  - Can have an optional internet gateway to handle public traffic
  - Can have an optional IPSec VPN connection to securely extend your on-premises network
- Subnet
  - Subdivision of a cloud network
  - Exists in a single availability domain
  - Consists of a contiguous range of IP addresses that do not overlap with other subnets in the cloud network

Virtual Cloud Network (VCN)

A Virtual Cloud Network is a virtual version of a traditional network—including subnets, route tables, and gateways—on which your instances run. A cloud network resides within a single region but can cross multiple availability domains. You can define subnets for a cloud network in different availability domains, but the subnet itself must belong to a single availability domain. You need to set up at least one cloud network before you can launch instances. You can configure the cloud network with an optional internet gateway to handle public traffic, and an optional IPSec VPN connection to securely extend your on-premises network.

Subnet

A subnet is a subdivision of a cloud network. Each instance resides in a subnet. Subnets contain virtual network interface cards (VNICs), which attach to instances. Each subnet in a VCN exists in a single availability domain and consists of a contiguous range of IP addresses that do not overlap with other subnets in the cloud network. Example: 172.16.1.0/24. The first two IP addresses and the last in the subnet's CIDR are reserved by the Networking service. You can't change the size of the subnet after creation, so it's important to think about the size of subnets you need before creating them. Also, the subnet acts as a unit of configuration: all instances in a given subnet use the same route table, security lists, and DHCP options.

Subnets can be either public or private. You choose this during subnet creation, and you can't change it later.

Task Flow to Launch an Oracle Cloud Infrastructure Instance

1. Create an SSH key pair.
2. Create or choose a compartment for your resources.
3. Create or choose a virtual cloud network (VCN).
4. Create or choose a subnet for the VCN.
5. Create an instance.
6. Connect to your instance.
7. Provision and manage block storage volumes (optional).
    a) Add a block storage volume.
    b) Attach the volume to an instance.
    c) Connect a volume to an instance's guest OS.

Instances use an SSH key pair instead of a password to authenticate a remote user. If you do not already have a key pair, your first task is to create one. You can use PuTTYgen on Windows or use the `ssh-keygen` utility on Oracle Linux to create an SSH key pair. You can have multiple key pairs to use for different instances, or you can use the same key pair for all instances.

Next, choose a compartment to hold your cloud resources. Your tenancy is the root compartment, but you can create additional compartments within the tenancy (root compartment) and corresponding policies to control access to the resources in each compartment. When you create a cloud resource such as an instance, block volume, or cloud network, you must specify to which compartment you want the resource to belong.

Next, create a VCN (virtual cloud network) with one or more subnets. You need to set up at least one cloud network before you can create instances.

Next, provide input and selections to create the instance. You can provide your own instance name instead of the name provided for you. An Oracle Cloud Identifier (OCID) uniquely identifies the instance. You specify an availability domain, select an image source and version, instance type (Virtual Machine or Bare Metal Machine) and shape. The image determines the operating system and other software for the instance. The shape determines the number of CPUs, amount of memory, and other resources allocated to a newly created instance.

You can change the boot volume size and encrypt boot volume data if desired.

You must specify the public SSH key(s) from your generated key pair(s). You can choose the SSH key file(s) from your computer, paste in the public key(s) from your key pair(s), or drag and drop key files.

You select a VCN compartment and VCN, as well as a subnet compartment and subnet to attach the instance to. The subnets are either public or private. Private means the instances in that subnet cannot have public IP addresses. When you create an instance, it is automatically attached to a Virtual Network Interface Card (VNIC) in the cloud network's subnet and given a private IP address from the subnet's CIDR.

You can optionally provide an available private IP address of your choice from the subnet's CIDR (otherwise the private IP address is automatically assigned). You can choose whether to assign a public IP address to the instance if the subnet is public. A public IP address is required to communicate with the instance over the internet, and to establish a Secure Shell (SSH) or Remote Desktop Protocol (RDP) connection to the instance from outside the cloud network.

You can optionally provide a host name to be used for DNS within the cloud network. This is available only if the VCN and subnet both have DNS labels.

You can specify a fault domain. If you do not specify the fault domain, the system selects one for you. Once the instance has been created, if you want to change the fault domain you need to terminate the instance and launch a new instance in the preferred fault domain. You can specify a startup script that will run when your instance boots up or restarts. Startup scripts can be used to install software and updates, and to ensure that services are running within the virtual machine. Tags can be specified to help you organize and track resources within your tenancy.

The instance is then launched. You must wait for a period of time while the instance is provisioned and transitions to a running state. You then need to wait another minute for the operating system to boot before you can SSH to the instance to connect to it.

Lastly, you can optionally provision block storage volumes for instances to meet your storage and application needs.

Setting Up a Virtual Cloud Network (VCN)

- Navigate to the interface for creating a VCN.
- Select an appropriate compartment.
- You can optionally provide a friendly name.
- A CIDR block range must be specified, which cannot later be changed.
- You can specify a DNS label for the VCN.
- If you create the VCN only, you must specify a CIDR block and subsequently create:
  - At least one subnet
  - An internet gateway
  - A route rule
- You can create the VCN with related resources, which automatically provides a CIDR block, subnets, a gateway, and a route rule for you.

ORACLE

When you work with Oracle Cloud Infrastructure, one of the first steps is to set up a Virtual Cloud Network (VCN) for your cloud resources. First, select an appropriate compartment. You can then begin VCN creation. You can optionally assign a friendly name to the cloud network. It does not have to be unique, and you can change it later (with the API, not the Console). Oracle automatically assigns the cloud network a unique identifier called an Oracle Cloud ID (OCID).

For the cloud network, you must specify a single, contiguous IPv4 Classless Inter-Domain Routing (CIDR) block. Oracle recommends the private IP address ranges specified in RFC 1918 (`10.0.0.0/8`, `172.16/12`, and `192.168/16`). Example: `10.0.0.0/16`. The cloud network can range from `/16` to `/30` and must not overlap with your on-premises network or another VCN you peer with (VCN peering = connecting multiple VCNs). The subnets in a given VCN must not overlap with each other. The Networking service reserves the first two IP addresses and the last one in each subnet's CIDR. It is possible to use a publicly routable range. You cannot change the size of the cloud network after creation.

If you want the instances in the VCN to have DNS hostnames (which can be used with the Internet and VCN Resolver, a built-in DNS capability in the VCN), specify that you will use DNS Hostnames in this VCN (this option is selected by default). Then you can specify a DNS label for the VCN, or the Console generates one for you. The dialog box automatically displays the corresponding DNS Domain Name for the VCN (<VCN_DNS_label>.oraclevcn.com).

If you only create the VCN at this point, you need to specify a CIDR block and subsequently set up at least one subnet, gateway, and route rule to have a working Virtual Cloud Network. You can select an option to create additional resources along with the VCN. This will automatically specify a CIDR block, provide an internet gateway, provide a route table with a default route rule, and create subnets in each availability domain.

Setting Up VCN Subnets

- Navigate to the interface for creating a VCN subnet.
- You can optionally provide a friendly name.
- An availability domain must be selected.
- A CIDR block range must be specified, which cannot later be changed.
- By default, subnets are considered public, which is required to access the internet.
  - You can change this to private
  - This choice cannot later be changed.
- You can optionally specify a route table, one or more security lists, and DHCP options.
  - These can be changed at any time.
- If compartment selection is enabled, you specify the compartment where you want the subnet to reside, as well as for the route table, security lists, and DHCP options.

ORACLE

When creating a subnet, you can optionally assign a friendly name to the subnet. Oracle automatically assigns the subnet a unique identifier called an Oracle Cloud ID (OCID). An availability domain is required and A CIDR Block must be specified. Example: `192.168.0.0/24`. You cannot change the CIDR block range of the subnet after creation.

You can optionally specify a route table for the subnet. If you do not, the cloud network's default route table is associated with the subnet. You can change which route table the subnet uses at any time. By default, subnets are considered public, which means instances in that subnet are allowed to have public IP addresses. You can change this to private and your choice for this setting cannot be changed later. You can add a DNS label for the subnet, which is required if you want the subnet's instances to use the Internet and VCN Resolver feature for DNS.

You can optionally associate a set of DHCP options for the subnet. All instances in the subnet receive the configuration specified in that set of DHCP options. If you do not specify a set, the cloud network's set of default DHCP options is associated with the subnet. You can change which set of DHCP options the subnet uses at any time. You can also optionally specify security lists for the subnet. If you do not specify any, the cloud network's default security list is associated with the subnet. You can change which security list the subnet uses at any time.

If you enable compartment selection, you specify the compartment where you want the subnet to reside, as well as for the Route Table, Security Lists and DHCP Options.

Viewing Instance Details and Accessing Your Instance

- Navigate to view instance details.
- You can view instance information:
  - Availability domain, fault domain, compartment
  - Shape, image, VCN
- VNIC information is provided:
  - IP addresses
  - FQDN
  - Subnet
- Any attached block volumes are shown.
- Log in to an instance using a public IP address and private key.

Instance details can be viewed from the console. This information includes the instance availability domain, fault domain and compartment. The shape, image, and VCN are displayed. VNIC information includes assigned IP addresses, the internal FQDN and subnet. Any attached block volumes are also displayed.

A public IP address is needed in order to access the instance over the internet. For example, if the assigned public IP address is `129.146.99.236`, you use this IP address to access your instance from a remote system. For instances created from Oracle Linux images, the default username is `opc`. To log on to an instance using `ssh`, specify the path and file name containing the private key associated with the instance, along with the `opc` username. For example, to log on with a private key contained in the file `id_rsa` in the `/root/.ssh` directory:

```
# ssh -i ~/.ssh/id_rsa opc@129.146.99.236
```

Some command output is shown after logging on to the Oracle Linux instance:

```
[opc@instance1 ~]$ hostname
instance1
[opc@instance1 ~]$ whoami
opc
[opc@instance1 ~]$ ls /home
opc
[opc@instance1 ~]$ uname -r
4.14.35-1818.3.3.el7uek.x86_64
```

Additional command output is shown on the next page.

```
[opc@instance1 ~]$ cat /etc/oracle-release
Oracle Linux Server release 7.5
[opc@instance1 ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq
state UP group default qlen 1000
    link/ether 02:00:17:00:b2:61 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 brd 192.168.1.255 scope global dynamic
ens3
       valid_lft 79247sec preferred_lft 79247sec
[opc@instance1 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs         14G     0   14G   0% /dev
tmpfs            14G     0   14G   0% /dev/shm
tmpfs            14G   25M   14G   1% /run
tmpfs            14G     0   14G   0% /sys/fs/cgroup
/dev/sda3        39G  1.7G   37G   5% /
/dev/sda1       200M  9.8M  191M   5% /boot/efi
tmpfs           2.8G     0  2.8G   0% /run/user/1000
[opc@instance1 ~]$ sudo fdisk -l
WARNING: fdisk GPT support is currently new, and therefore in an
experimental phase. Use at your own discretion.

Disk /dev/sda: 50.0 GB, 50010783744 bytes, 97677312 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disk label type: gpt
Disk identifier: BA353020-4A62-4D9F-BFCE-542E8BD9A16A

#         Start          End    Size  Type            Name
 1         2048       411647    200M  EFI System      EFI System
Partition
 2       411648     17188863      8G  Linux swap
 3     17188864     97675263   38.4G  Microsoft basic
```

OCI Utilities / `oci-utils`

- Preinstalled set of utilities to make it easier to work with Oracle Linux images
- Service component
  - `ocid`
  - Normally runs as a daemon started via `systemd`.
- Command line tools
  - `oci-growfs`: Expands the root file system of the instance to its configured size
  - `oci-iscsi-config`: Displays and configures iSCSI devices attached to a compute instance
  - `oci-metadata`: Displays metadata for the compute instance
  - `oci-network-config`: Lists or configures Virtual Network Interface Cards (VNICs) attached to the compute instance
  - `oci-network-inspector`: Displays a detailed report for a given compartment or network
  - `oci-public-ip`: Displays the public IP address of the current system in either human-readable or JSON format

Instances created using Oracle-provided images based on Oracle Linux include a preinstalled set of utilities that are designed to make it easier to work with Oracle Linux images. The package name is `oci-utils`. These utilities consist of a service component and related command-line tools that can help with managing block volumes (attach, remove, and automatic discovery), secondary VNIC configuration, discovering the public IP address of an instance, and retrieving instance metadata.

Creating a Block Volume

- Navigate to the interface for creating a block volume.
- Select a compartment.
- Provide a friendly name.
- An availability domain must be selected and be the same as for the instance.
- Select a size in 1 GiB increments.
- A backup policy can be selected.
- Tags can optionally be applied.
- You can specify that data be encrypted.

The Oracle Cloud Infrastructure Block Volume Service lets you dynamically provision and manage block storage volumes. You can create, attach, connect, and move volumes as needed to meet your storage and application requirements. When attached and connected to an instance, you can use a volume like a regular hard drive. Volumes can also be disconnected and attached to another instance without the loss of data.

To create a Block Storage Volume, choose a compartment to hold your block volume resource. Provide a user-friendly name or description. An availability domain must be selected and must be the same availability domain as the instance. Choose the size of the block volume. A backup policy can be selected and tags can be applied. You can optionally choose that the data on the volume be encrypted.

After the volume is created and available, it is ready to attach to an instance.

Attaching a Block Volume to an Instance

- Navigate to the interface for attaching a block volume to an instance.
- Select the iSCSI (the default) or Paravirtualized attachment type.
- Select a compartment.
- Select the block volume desired.
- CHAP credentials can optionally be enabled.
- Select READ/WRITE or READ-ONLY for the access type.

ORACLE

You can attach a volume to an instance to expand the available storage on the instance. When you attach a block volume to an instance, you select either iSCSI or Paravirtualized as the volume attachment type. iSCSI attachments are the only option when connecting block volumes to bare metal instances, VM instances based on Windows images published before February 2018, or VM instances based on Linux images published before December 2017. Once the volume is attached, you need to log in to the instance and use the `iscsiadm` command-line tool or use `oci-utils` to configure the iSCSI connection.

Paravirtualized attachments are an option when attaching volumes to VM instances. For VM instances launched from Oracle-provided images, you can select this option for Linux-based images published December 2017 or later, and Windows images published February 2018 or later. For VM instances launched from custom images, the volume attachment type is based on the volume attachment type from the VM the custom image was created from. Once you attach a volume using the paravirtualized attachment type, it is ready to use - you do not need to run any additional commands. However, due to the overhead of virtualization, this reduces the maximum IOPS performance for larger block volumes.

Select the volume compartment and volume you want to attach.

For iSCSI attachments, you can optionally enable CHAP (Challenge-Handshake Authentication Protocol). This is a security protocol used by iSCSI for authentication between a volume and an instance.

Select READ/WRITE or READ-ONLY for the access type.

When attached, you must still connect and mount the volume from the instance for the volume to be usable.

Connecting a Block Volume to an Instance's Guest OS

- For iSCSI attachments, iSCSI commands must be issued to connect the volume to the OS.
- The Oracle Cloud Infrastructure console provides the necessary iSCSI commands.
- Issue the given commands on your instance.
- Alternatively, run the `oci-iscsi-config` utility from the OCI utilities.
- If the `ocid` daemon is started, devices are connected automatically.
- You can then partition the new block device, create a file system on the device, create a mount point, and mount the new file system.
- Include the `_netdev` and `nofail` options on every nonroot block volume in the `/etc/fstab` file.

You use the iSCSI protocol to connect a block volume to an instance when the iSCSI attachment type is selected. After the volume is attached, you log on to the instance and use the `iscsiadm` command-line tool to configure the iSCSI connection. After you configure the volume, you can mount it and use it like a normal hard drive. The Oracle Cloud Infrastructure console provides the necessary iSCSI commands to make the connection. Log on to your instance and issue the `iscsiadm` commands provided.

Alternatively, you can run the `oci-iscsi-config` utility from the `oci-utils` package. This will automatically make the proper iSCSI connection to your instance. See the `oci-iscsi-config(1)` man page for details. If you start the `ocid` daemon, it will automatically connect devices without running the `oci-iscsi-config` command, if they do not require manual interaction.

After the iSCSI connection has been established, you can partition the block device, create a file system and mount the file system.

If you use `/etc/fstab` to automatically mount volumes upon reboot, be sure to include the `_netdev` and `nofail` options on every nonroot block volume in the `/etc/fstab` file.

By default, the `/etc/fstab` file is processed before the initiator starts. The `_netdev` option causes the mount process to initiate before the volumes are mounted.

When you create a custom image of an instance where the volumes, excluding the root volume, are listed in the `/etc/fstab` file, instances will fail to launch from the custom image. Specifying the `nofail` option in the `/etc/fstab` file prevents this issue.

Oracle Cloud Computing Resources

Refer to the following resources for further information about Oracle Cloud Computing:

- https://www.oracle.com/cloud
- https://cloud.oracle.com/home
- https://cloud.oracle.com/cloud-infrastructure
- https://docs.cloud.oracle.com
- https://education.oracle.com/oracle-cloud-learning-subscriptions

Quiz

Which of the following statements are true? (Select all that apply.)

a. Oracle Cloud Infrastructure is physically hosted in regions and Availability Domains.
b. Oracle Cloud Infrastructure is physically hosted in tenancies and compartments.
c. When you sign up for Oracle Cloud Infrastructure, Oracle creates a tenancy for your company.
d. Compartments allow you to organize and control access to your cloud resources.

Answer: a, c, d

Quiz

Which of the following statements about Oracle Cloud Infrastructure instances are true? (Select all that apply.)

a. You need to set up at least one Virtual Cloud Network (VCN) before you can launch instances.

b. Bare metal compute instances run on bare metal servers without a hypervisor.

c. Managed Virtual Machine (VM) instances are also available for workloads that don't require dedicated physical servers or the high performance of bare metal instances.

d. When launching an instance, you also need to select an image, a shape, a private IP address, and a public IP address.

Answer: a, b, c

Answer d is false. You can optionally enter an available private IP address of your choice from the subnet's CIDR; otherwise, the private IP address is automatically assigned. You can optionally assign a public IP address to an instance. This option is available only if the subnet you selected for the instance is public.

Summary

In this lesson, you should have learned how to:

- Define Infrastructure as a Service (IaaS)
- Describe Oracle Private Cloud Appliance
- Describe Oracle OpenStack
- List some Oracle Cloud Infrastructure services
- Describe some key concepts and terms used in Oracle Cloud Infrastructure
- List some Oracle-provided images and available shapes in Oracle Cloud Infrastructure
- Describe the task flow to launch an Oracle Cloud Infrastructure instance
- Describe creation of an Oracle Cloud Infrastructure Virtual Cloud Network and subnet
- Describe oci-utils
- Describe attachment of an Oracle Cloud Infrastructure block volume to an instance

Practice 3: Overview

This practice covers Quiz – Oracle Cloud Computing.

4

Installing Oracle Linux 7

Objectives

After completing this lesson, you should be able to:

- Obtain the Oracle Linux operating system software
- Describe the Anaconda installer
- Install Oracle Linux
- Complete initial setup
- Upgrade from Oracle Linux 6
- Describe steps to launch an Oracle Cloud Infrastructure instance

ORACLE

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 to familiarize you with the Oracle Linux installation process.

Obtaining Oracle Linux

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

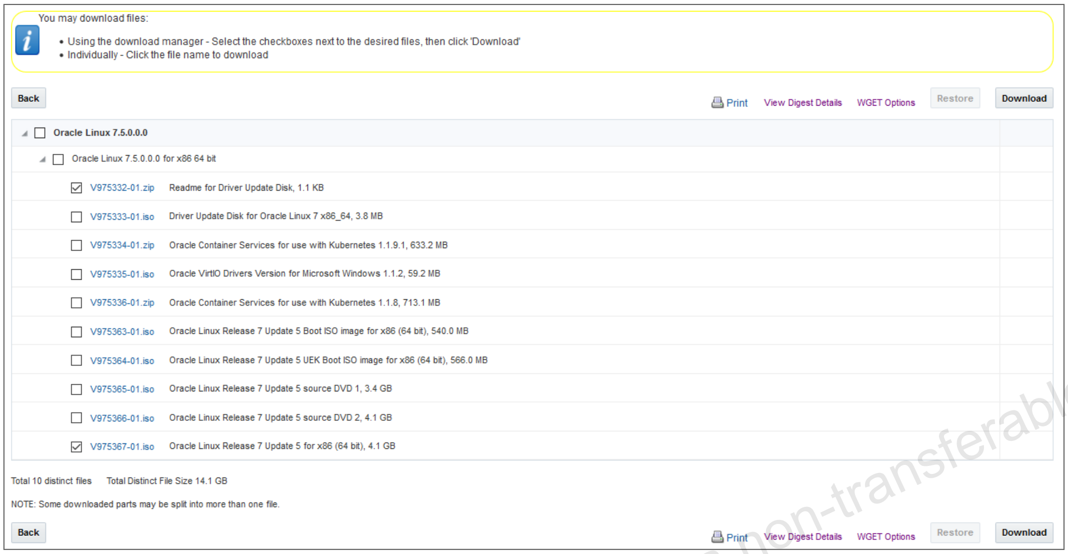
True to the open source philosophy, Oracle has provided Oracle Linux software as a free download from day 1. Anyone can download the binaries, installation media, and the source code for free without a support subscription.

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://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 7 Update 1) 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 Oracle Linux yum server.

Oracle Software Delivery Cloud

Oracle Linux download page: Example



ORACLE

This screen lists the download options for Oracle Linux 7 Update 5 from the Oracle Software Delivery Cloud: http://edelivery.oracle.com/linux. As shown, Oracle Linux 7 Update 5 comes in different ISO images for the x86_64 bit architecture. One is the complete distribution and others include boot ISO images (one is the Unbreakable Enterprise Kernel (UEK) and the other is the Red Hat Compatible Kernel [RHCK]). The full distribution ISO is what is typically needed for installations of Oracle Linux. The source code is also available for download from this page, listed as DVD 1 and DVD 2.

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

To install Oracle Linux 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.

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. The installation can be automated with Kickstart for unattended installation.

Oracle Linux Installation Menu

This slide shows the Oracle Linux 7.5 installation menu, which is the first window to appear during an installation. The default option of "Test this media & install Oracle Linux 7.5" is automatically selected within 60 seconds. Press Esc at the boot menu to get a boot prompt.

Install Oracle Linux 7.5

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

Test this media and install Oracle Linux 7.5

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 a 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 installation 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

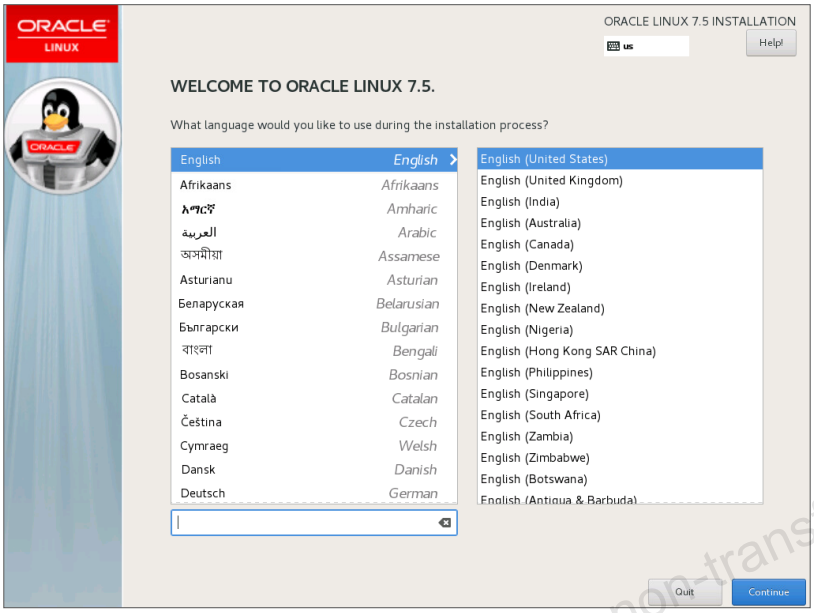
For example, to specify language and keyboard layout, enter:

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

To run the installation in text mode, enter:

```
boot: linux text
```

Welcome to Oracle Linux

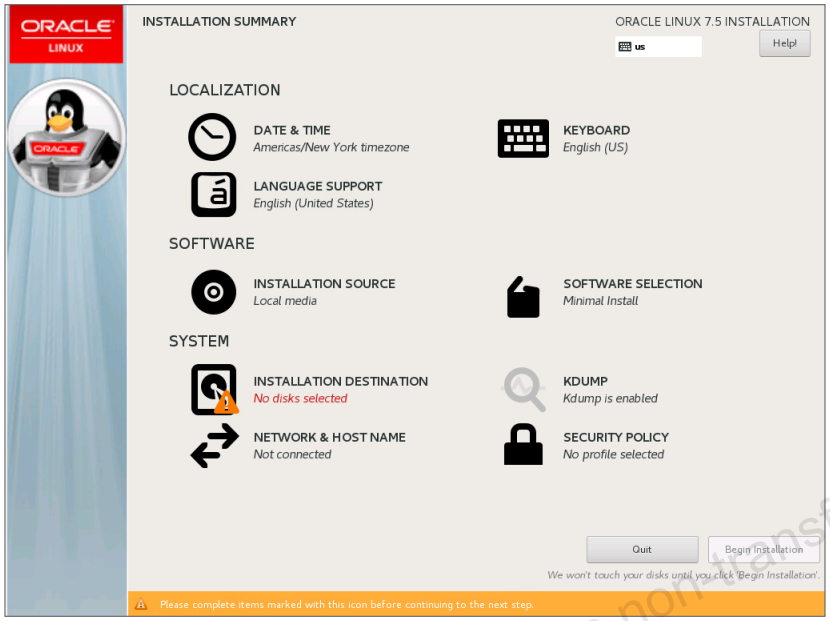


The Welcome to Oracle Linux window appears. In the box on the left, select the language to use for the installation. The language becomes the default language for the operating system. The language is also used to target the time zone configuration later during the installation process.

In the box on the right, 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

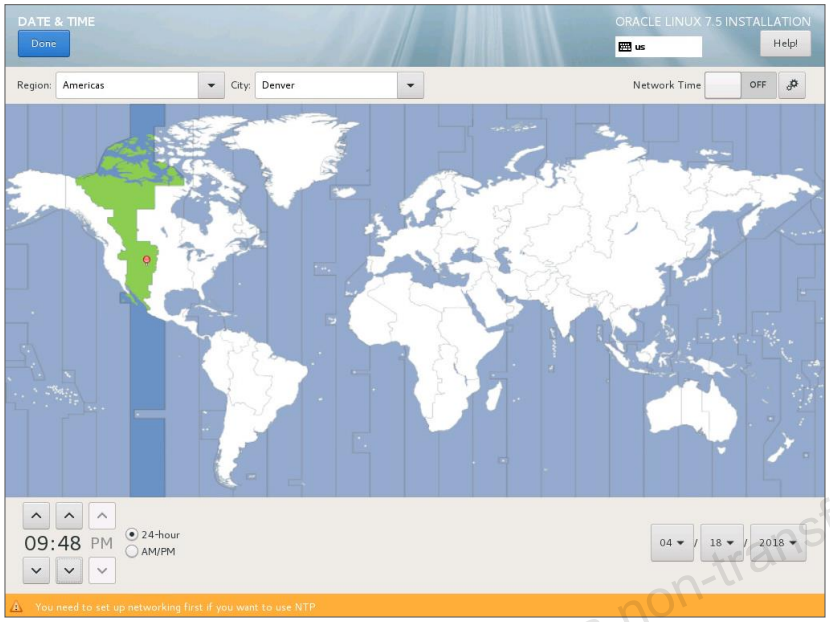


The Installation Summary window appears. The interface is a hub-and-spoke model rather than the linear model used in previous releases of Oracle Linux. You can select any option in any order.

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 section 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 and Time Configuration

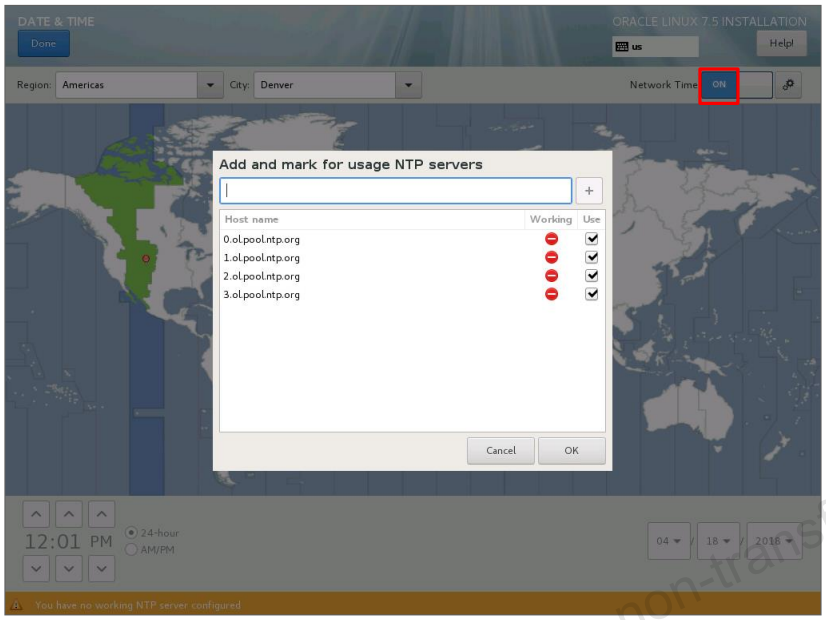


Selecting DATE & TIME from the INSTALLATION SUMMARY screen displays the screen shown in the slide. Select the appropriate time zone by selecting the country and city closest to the location of your system. To select the time zone, either click the map or select from the Region and City drop-down menus.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

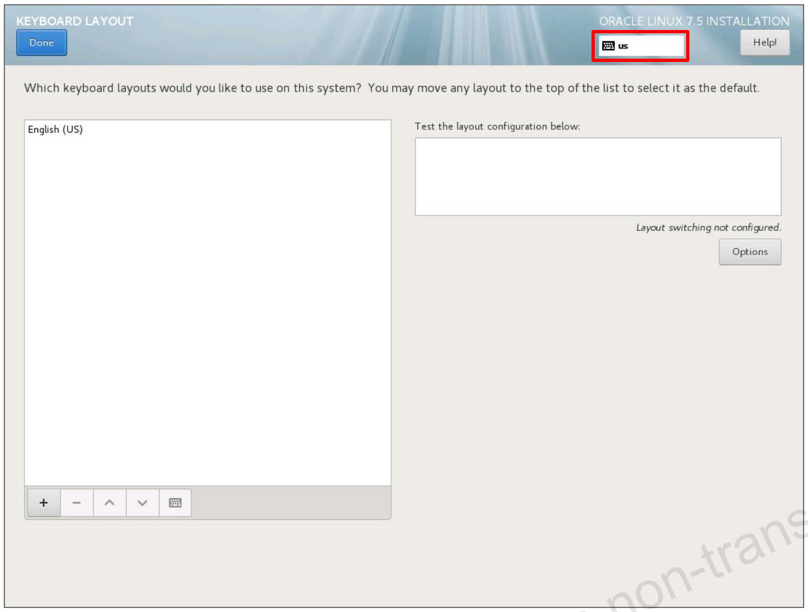


This slide shows the screen to configure Network Time Protocol (NTP). You must be connected to a network before you can enable NTP.

Move the Network Time switch to the ON position (outlined in red on the slide) and click the configuration icon 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

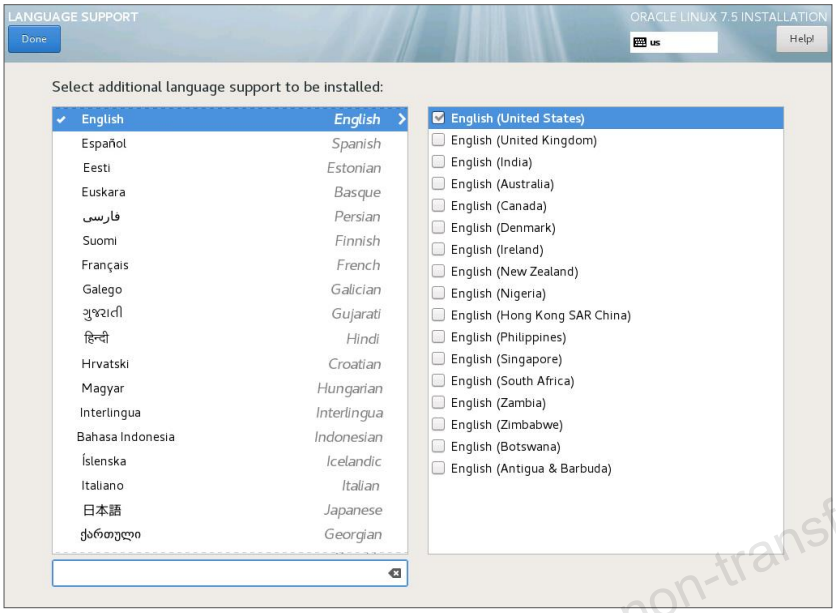


Selecting KEYBOARD from the INSTALLATION SUMMARY screen displays the screen shown in the slide. The language you selected in the welcome screen is listed as the keyboard layout in the left pane. To add support for any additional keyboards for your system, click the + button at the bottom of the screen.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

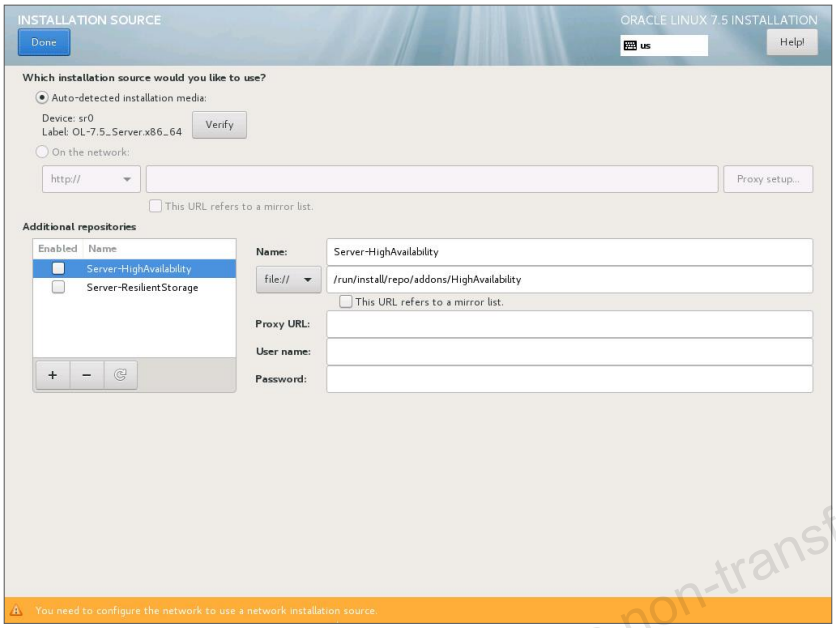
Language Support



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 box on the left, or type in the search box, and then select one or more locales in the box on the right.

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.

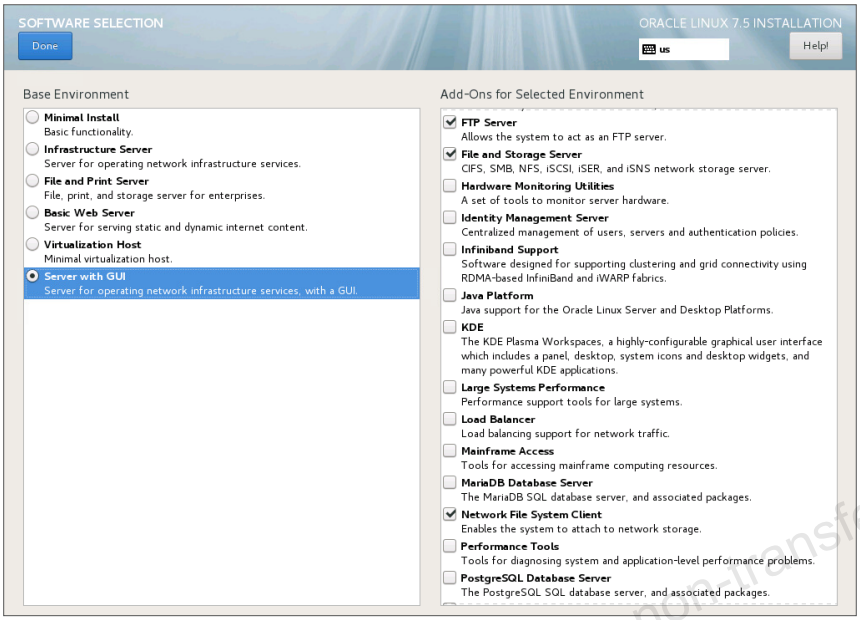
On the Network

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. Short descriptions are provided in the SOFTWARE SELECTION panes, giving input into which selections are appropriate for various environments.

Base Environment

Select the Base Environment from the left pane. The Base Environments are predefined 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 desktop.

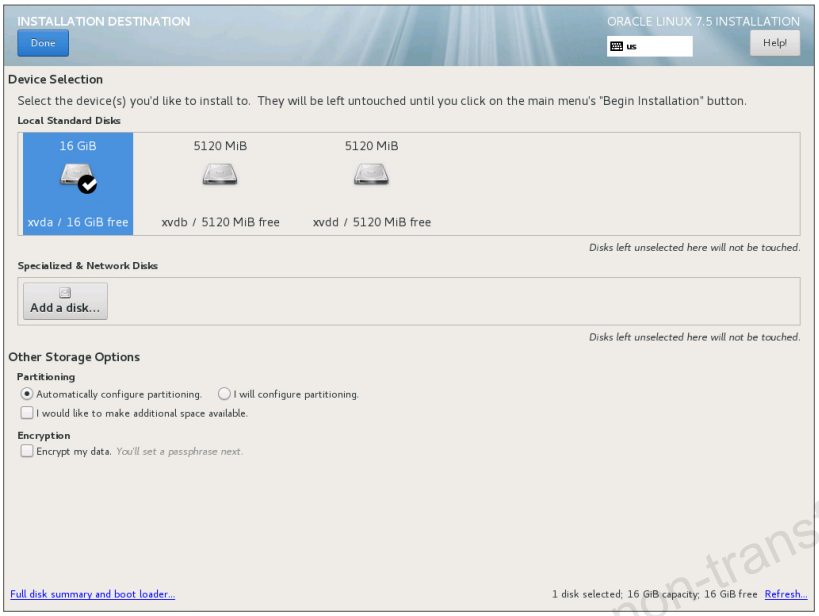
The Minimal Install base environment is the default environment. It contains only the minimum set of packages required to run Oracle Linux.

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 configure the storage.

Local Standard Disks

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 check mark.

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.

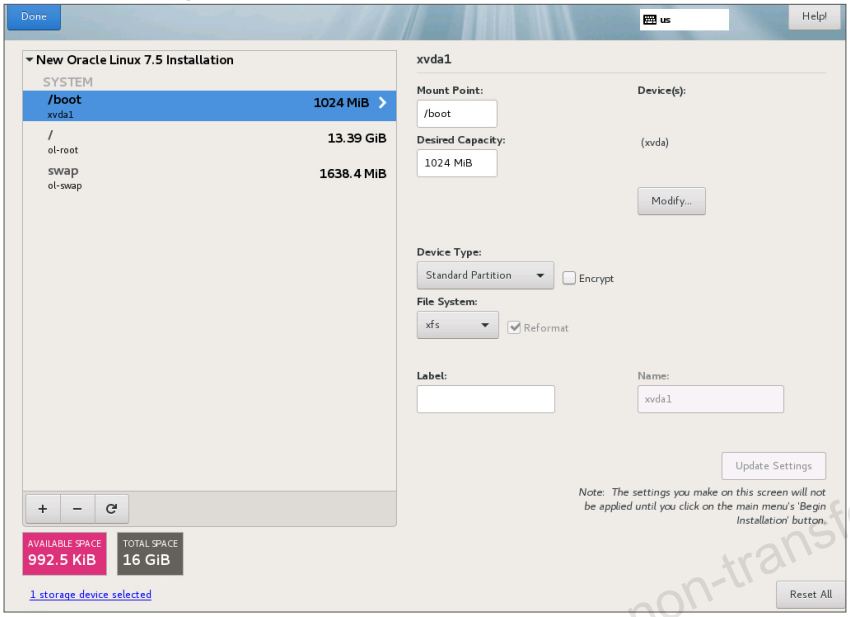
Other Storage Options

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.

The installer shows device sizes in binary units (MiB and GiB) rather than decimal units (MB and GB). It is preferable that partitions be aligned on MiB boundaries. Using binary units also can allow partitions to be aligned with physical block sizes, improving transfer speeds.

Automatic Partitioning



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

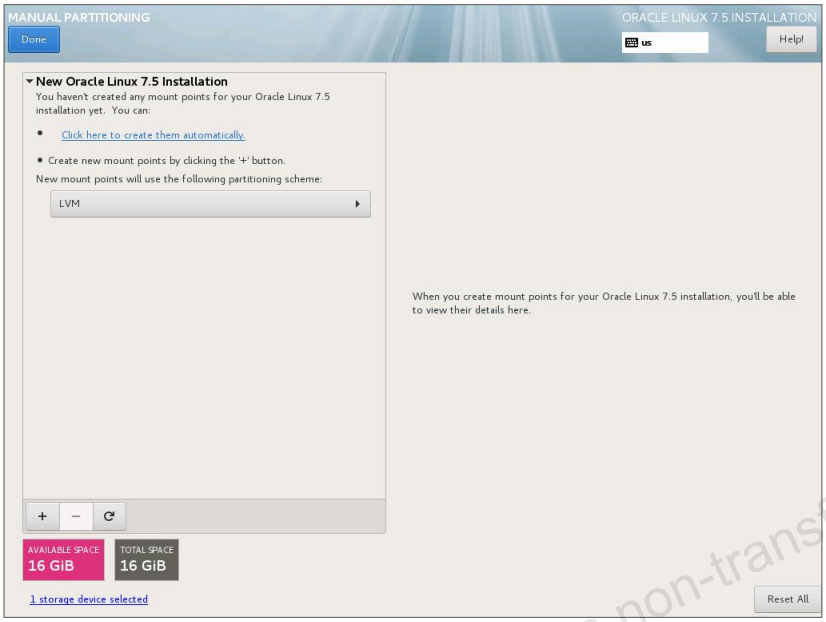
- A partition for an XFS file system that contains `/boot`
- A partition in the remaining disk space configured as a Logical Volume Manager (LVM) physical volume for an LVM volume group that contains:
  - A logical volume for an XFS file system that contains the root file system (`/`)
  - A logical volume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

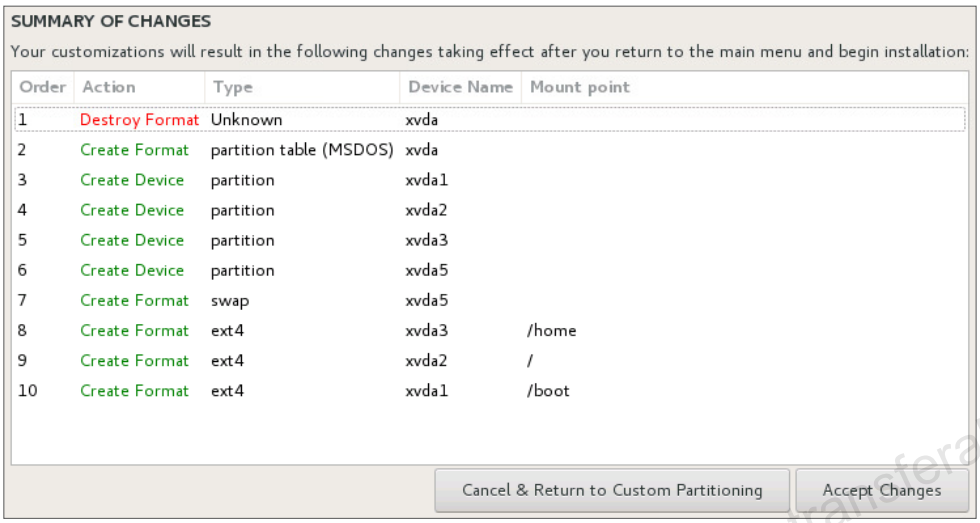
- Standard Partition
- BTRFS
- LVM
- LVM Thin Provisioning

Menu options are available to perform the following actions:

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

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

Summary of Partitioning

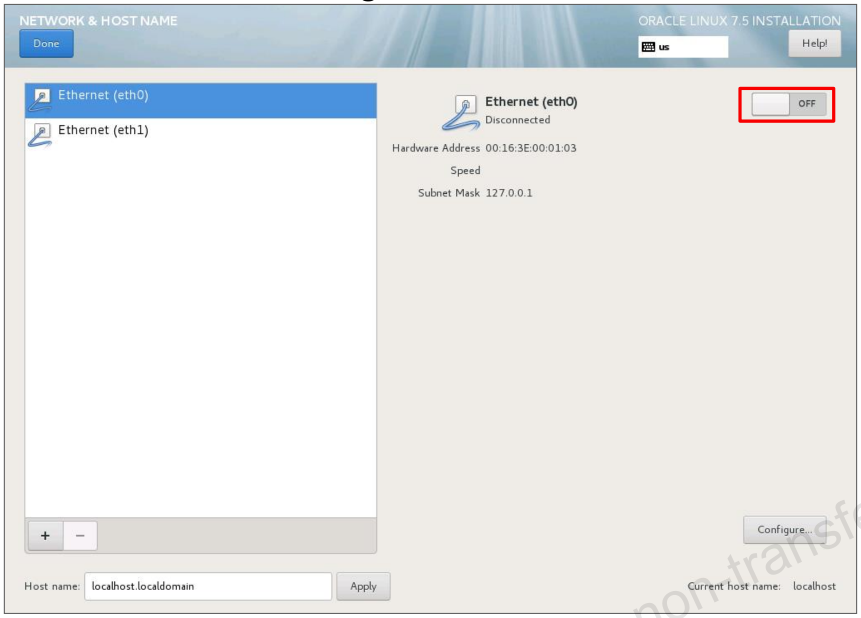


Click Done after partitioning your storage devices. The window in the slide appears summarizing your partitioning customizations. Click Accept Changes to continue or click Cancel & Return to Custom Partitioning to make further changes.

Partitioning and file system creation are discussed in the lesson entitled “Partitions, File Systems, and Swap.” The XFS file system is discussed in the lesson titled “Implementing the XFS File System” in the Oracle Linux System Administration II course. The Btrfs file system is discussed in the lesson titled “Implementing the Btrfs File System” in the Oracle Linux System Administration II course. Logical Volume Management (LVM) and RAID are discussed in the lesson titled “Storage Administration” in the Oracle Linux System Administration II course.

Click Accept Changes to return to the INSTALLATION SUMMARY screen.

Network and Host Name 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 host name for your system.

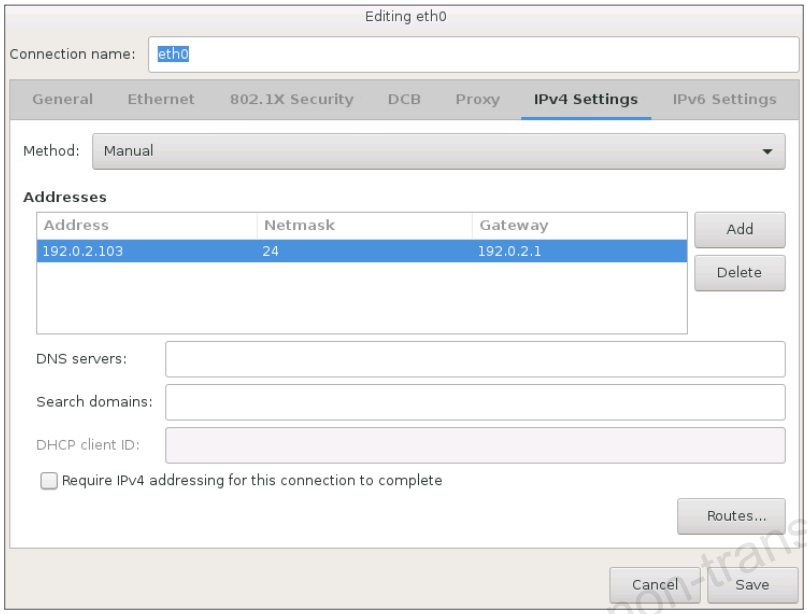
You need to configure a network if network access is required when your system boots for the first time and if you desire to configure NTP at installation time. Otherwise, you can configure the network after the installation is complete.

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

Network Connection Settings

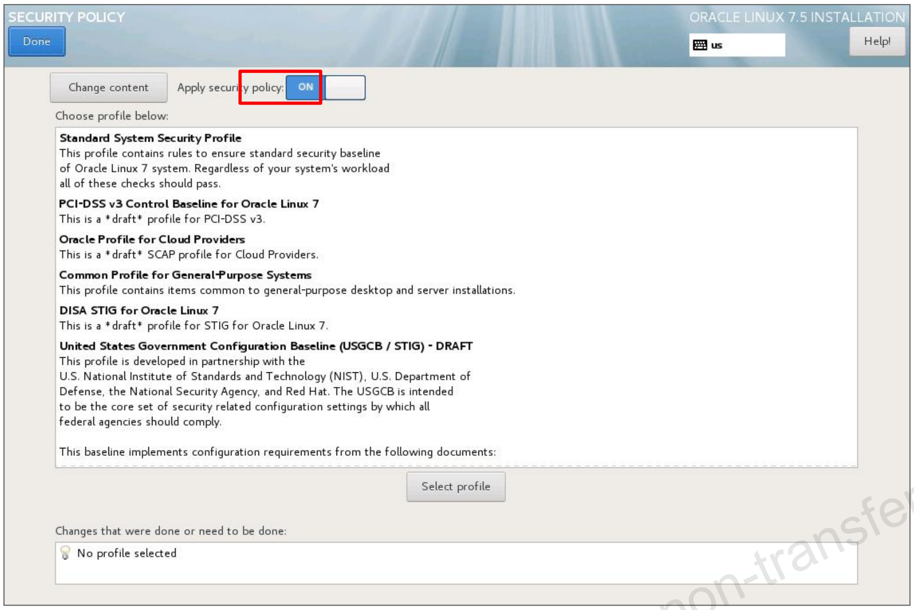


Selecting Configure from the NETWORK & HOST NAME screen and the IPv4 Settings tab displays the screen shown in the slide. The configuration options presented depend on whether the connection is wired, wireless, mobile broadband, VPN, or DSL. The lesson titled “Network Configuration” provides details of the configuration options.

This screen shows manual IPv4 configuration for a wired interface. The IP address, netmask, and gateway are provided. Domain Name Service (DNS) servers and search domains can be configured on this screen.

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

Security Policy



ORACLE

Selecting SECURITY POLICY from the INSTALLATION SUMMARY screen displays the screen shown in the slide. Select this screen to apply a security policy to your system. Not all systems require a security policy. Apply a security policy if it is required by your organization or by government regulations.

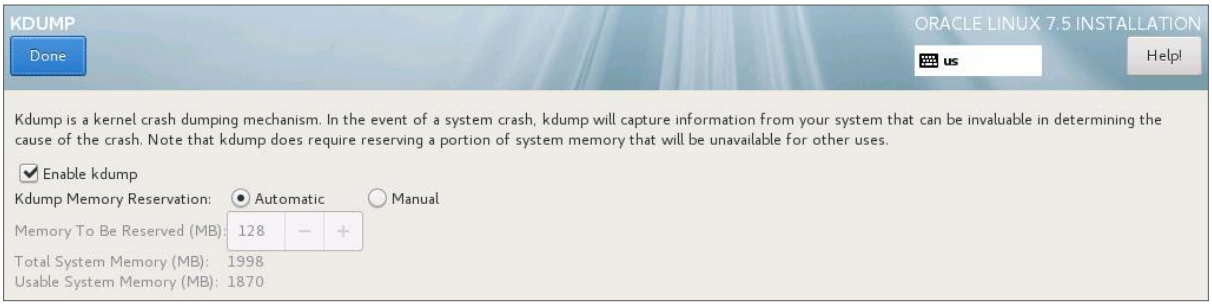
To enable a security policy, ensure that the “Apply security policy” switch is in the ON position (outlined in red) and select one of the security policies. These predefined policies are provided by the Security Content Automation Protocol (SCAP) Security Guide. For more information, refer to: http://www.open-scap.org/security-policies/choosing-policy/.

If you click Change content, you are prompted to either “Use SCAP Security Guide” or enter the data stream content or archive URL. You can then click Fetch if you have entered an archive URL from which to fetch content.

If you apply a security policy, your system is installed by using the restrictions and recommendations defined by the selected policy. OpenSCAP is installed and your system is automatically scanned to verify compliance. See the Oracle Linux 7 Security Guide at the following location for more information about using OpenSCAP to scan for vulnerabilities: http://docs.oracle.com/cd/E52668_01/E54670/E54670.pdf.

Click Done to return to the INSTALLATION SUMMARY screen.

KDUMP



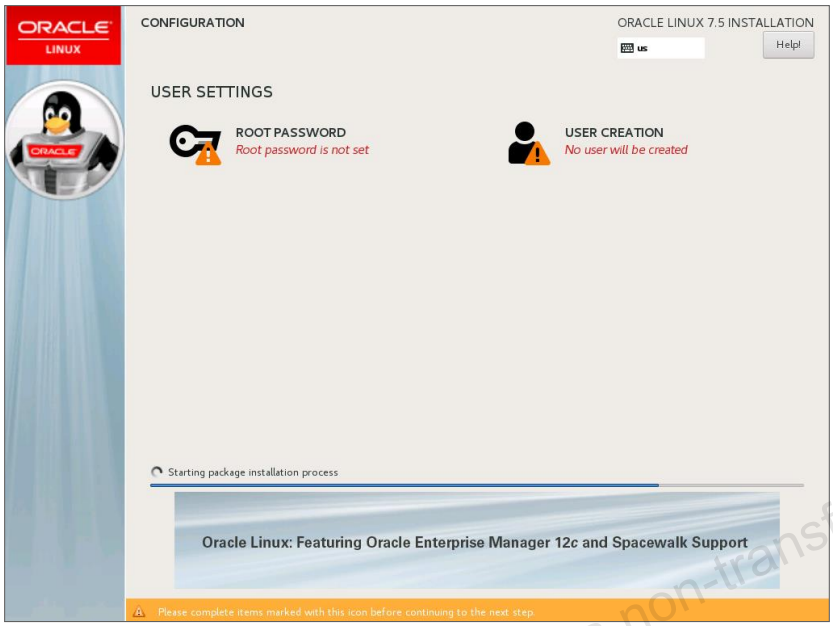
- Crash dumping mechanism for the Linux kernel
- Enabled by default

Selecting KDUMP from the INSTALLATION SUMMARY screen displays the screen shown in the slide. Kdump is enabled by default during the installation process. Select this screen to disable it.

Kdump is the Linux kernel crash dumping mechanism. In the event of a system crash, Kdump provides a memory dump (vmcore) image. This image can assist in determining the cause of the crash. It is highly recommended that you keep the Kdump feature enabled.

With Kdump enabled, a portion of the system memory is reserved for the capture kernel. This portion of memory is unavailable for other uses. By default, the Kdump Memory Reservation is automatically selected for you. You can manually change the memory reservation if desired.

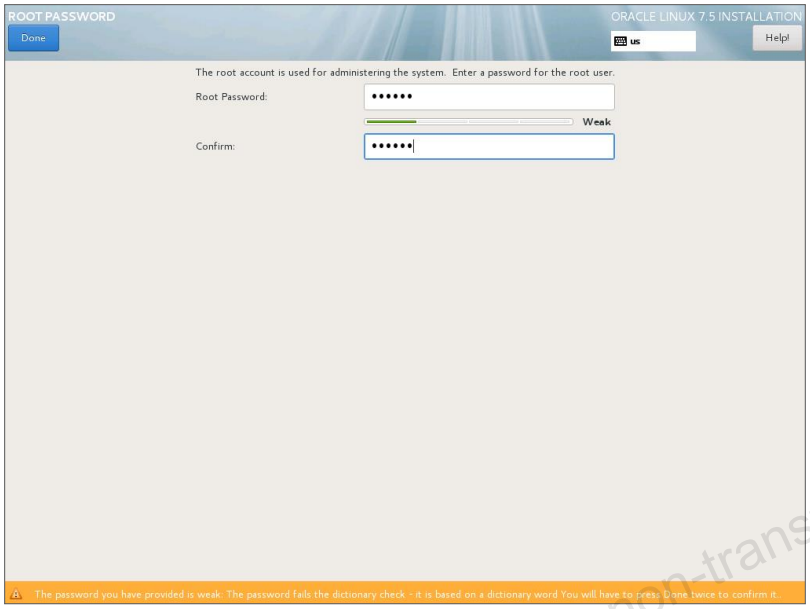
Completing the Installation



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 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.

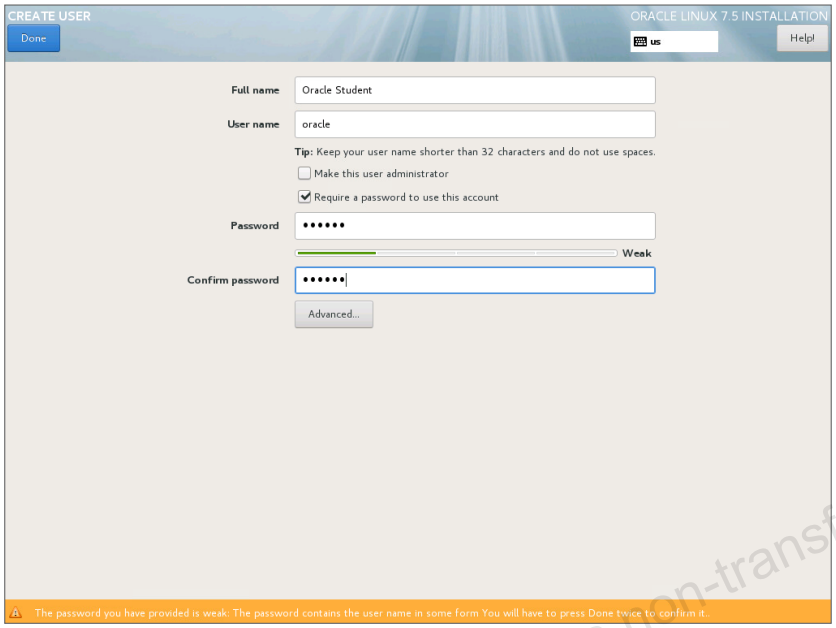
Setting the `root` Password



Clicking ROOT PASSWORD from the CONFIGURATION screen displays the screen shown in the slide. On this screen, enter your new password in the Root Password field. Your entry does not display the characters for security. There is an 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 screen 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.

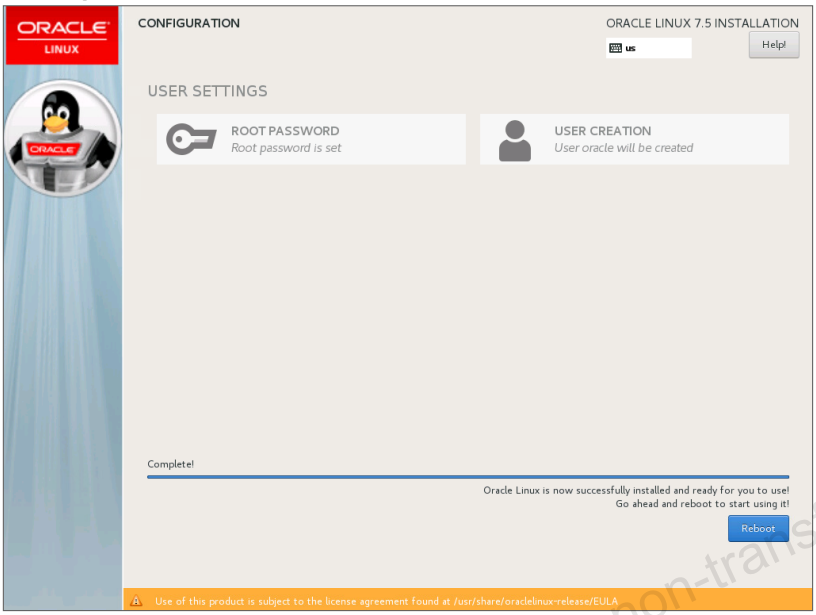
Enter the Password twice. There is an indication about the strength of the password.

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 the user to run administrative commands 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



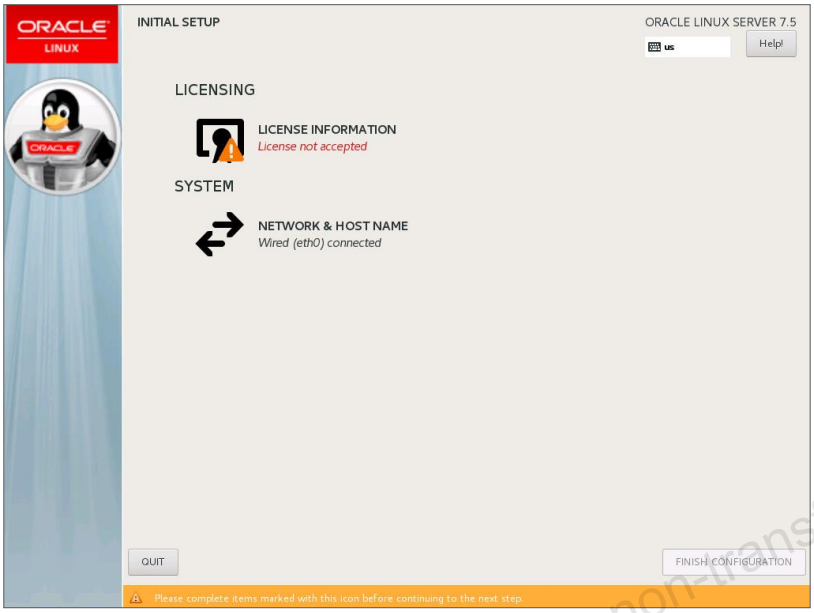
After the installation and configuration process is complete, a "Complete!" message appears, and the CONFIGURATION screen prompts you to Reboot. Remove any installation media if it is not ejected automatically upon reboot.

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 the Server with GUI base environment, the system reboots as a nongraphical system and displays a `login:` prompt.

If you installed the Server with GUI environment, the system reboots and displays the Initial Setup screen.

Initial Setup

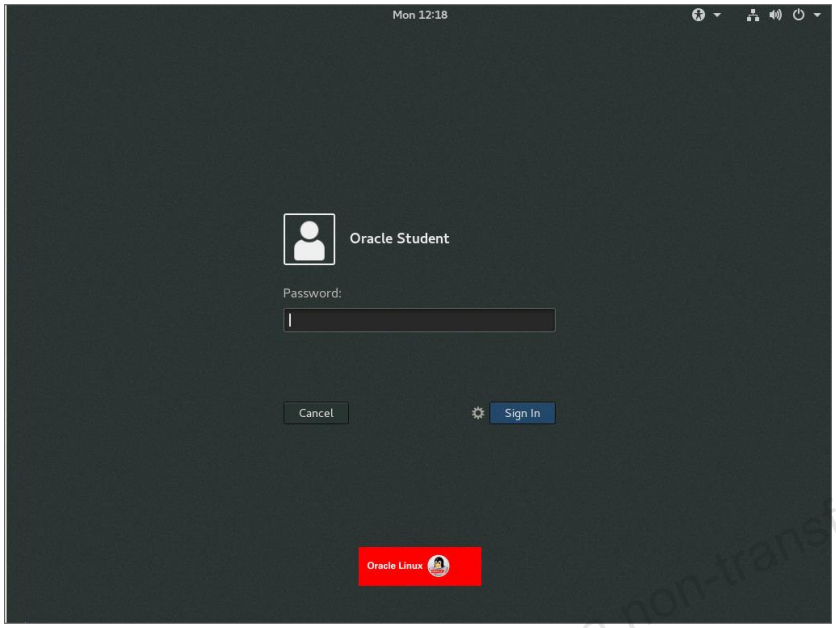


If you install the Server with GUI environment, the system reboots and the Initial Setup screen is displayed as shown in the slide. A USER CREATION option appears on this window if you did not already create an initial nonroot user.

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

Click FINISH CONFIGURATION in the INITIAL SETUP window.

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
- **Privacy:** Allow or disallow geographical location detection. Defaults to "ON".
- **Online accounts:** Connect to your existing data in the cloud

When these selections are complete, click Start using Oracle Linux Server.

The GNOME help window opens after you click Start using Oracle Linux Server. Close the help window to display the GNOME desktop. The Gnome-initial-setup and help windows open only the first time you log in.

Inplace Upgrade from Oracle Linux 6

An inplace upgrade to Oracle Linux 7 from Oracle Linux 6 Update 5 or later is possible under the following conditions:

- The system meets the minimum installation requirements as described in the Oracle Linux 7 Release Notes.
- The Oracle Linux 6 system has been completely updated from the ol6_x86_64_latest channel or ol6_latest repository.
- UEK R3 or UEK R4 is installed on the system to be upgraded and is the default boot kernel.
  - Upgrading from UEK R2 is not supported.
- No Oracle product stack is present on the system.
- The system to be upgraded has the Minimal Install base environment.
- See “Upgrading an Oracle Linux System” in the *Oracle Linux 7 Installation Guide* for complete instructions.

It is possible to upgrade to Oracle Linux 7 from Oracle Linux 6 Update 5 or later under the following conditions:

- The system meets the minimum installation requirements for Oracle Linux 7. See the Section titled “System Requirements” in the Oracle Linux 7 Installation Guide for Release 7 at https://docs.oracle.com/cd/E52668_01/E54695/html/ol7-install-requirements.html.
- The Oracle Linux 6 system has been completely updated from the ol6_x86_64_latest channel or ol6_latest repository.
- UEK R3 or UEK R4 is installed on the system to be upgraded and is the default boot kernel. Upgrading from UEK R2 is not supported.
- The system to be upgraded has the Minimal Install base environment.
- No Oracle product stack is present on the system.

If the Red Hat Compatible Kernel (RHCK) is installed, it is upgraded as part of the process, but it must not be set as the default boot kernel.

To perform an inplace upgrade, you must install some additional packages and their dependencies. The system should have access to the Unbreakable Linux Network (ULN), Oracle yum, or a local yum server.

For instructions about how to perform an upgrade, see “Upgrading an Oracle Linux System” in the *Oracle Linux 7 Installation Guide* at http://docs.oracle.com/cd/E52668_01/E54695/html/ol7-upgrade.html.

Launching an Oracle Cloud Infrastructure Instance

- Launching an instance does not require system installation, as described in this lesson.
- An image and shape are selected, providing an operating system template as well as CPU and memory resources.
- A Virtual Cloud Network and subnet are selected.
- One or more SSH public keys are provided.
- The operating system with accompanying resources is deployed for you.
- After the operating system boots, you can log in to the instance.

Quiz

To enable NTP during installation, you must be connected to a network.

a. True
b. False

Answer: a

Quiz

The default File System type when partitioning disks is:

a. ext4
b. ext3
c. xfs
d. vfat

ORACLE

Answer: c

Summary

In this lesson, you should have learned how to:

- Obtain the Oracle Linux operating system software
- Describe the Anaconda installer
- Install Oracle Linux
- Complete initial setup
- Upgrade from Oracle Linux 6
- Describe steps to launch an Oracle Cloud Infrastructure instance

Practice 4: Overview

This practice covers the following topics:

- Installing Oracle Linux
- Completing initial setup
- Booting up the host03 VM guest

5

Oracle Linux 7 Boot Process

Objectives

After completing this lesson, you should be able to:

- Describe the Oracle Linux boot process - BIOS and UEFI
- Explain and configure the GRUB 2 bootloader
- Describe features of UEFI, the GUID Partition Table (GPT), and Secure Boot
- List and configure kernel boot parameters
- Describe the `systemd` system and service manager
- Explain `systemd` service units
- Configure services
- Describe `systemd` target units
- Configure Rescue Mode and Emergency Mode
- Perform shutdown, suspend, and reboot operations

The Oracle Linux 7 Boot Process: BIOS Mode

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 files linked by `/etc/systemd/system/default.target`
   C. Brings the system to the state defined by the system target
   D. Executes `/etc/rc.local` (under certain conditions)

It is important to understand the Linux boot process when troubleshooting boot problems. The slide shows the high-level steps in the boot process. You need to be aware of the 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 (points to `/etc/rc.d/rc.local`) 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/*4.1.12-112.16.4*
...
/boot/initramfs-4.1.12-112.16.4.el7uek.x86_64.img
...
/boot/vmlinuz-4.1.12-112.16.4.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
```

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 `4.1.12-112.16.4`, there is a `vmlinuz` file and an `initramfs` file containing the same version number:

```
# ls /boot/*4.1.12-112.16.4*
...
/boot/initramfs-4.1.12-112.16.4.el7uek.x86_64.img
...
/boot/vmlinuz-4.1.12-112.16.4.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 `initramfs` 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
```

The simplest form of the `dracut` utility takes no parameters:

```
# dracut
```

This creates an `initramfs` image using the currently running kernel version, creating an image file with the name `initramfs-<kernel version>.img` in the `/boot` directory, using the currently running kernel version in the image file name.

For example, assume the currently running kernel version is determined as follows:

```
# uname -r
3.10.0-514.el7.x86_64
```

Running the `dracut` utility with no additional parameters will create the following image, using the Red Hat Compatible Kernel (RHCK) version `3.10.0-514.el7.x86_64`:

```
/boot/initramfs-3.10.0-514.el7.x86_64.img
```

The following form of `dracut` provides the image file name as a single parameter, specifying the `/boot` directory location for the file:

```
# dracut /boot/initramfs-3.10.0-514.el7.x86_64.img
```

This will create an `initramfs` image with file name `initramfs-3.10.0-514.el7.x86_64.img` using the currently booted kernel version. With this form, if a directory or path is not specified for the image file location, it will be created in the current working directory. You normally want these files in the `/boot` directory and may want to navigate to `/boot` before running `dracut` commands, to avoid having to add `/boot` to the parameter specifications.

The form allows you to create `initramfs` images of any file name, but it is important to specify file names with versions for `initramfs` images that match kernel image versions you intend to boot from, to avoid boot issues. For example, assume the Unbreakable Enterprise Kernel (UEK) version `4.1.12-61.1.18.el7uek.x86_64` is running and the following `dracut` command is run:

```
# dracut /boot/initramfs-3.10.0-514.el7.x86_64.img
```

This will create an `initramfs` image file name of `initramfs-3.10.0-514.el7.x86_64.img` using the currently booted UEK version of `4.1.12-61.1.18.el7uek.x86_64`, instead of the desired RHCK version of `3.10.0-514.el7.x86_64`. Because this `initramfs-3.10.0-514.el7.x86_64.img` does not correspond to the RHCK kernel executable `vmlinuz-3.10.0-514.el7.x86_64`, attempting to boot into the RHCK will fail.

To correct this situation, you would need to boot into UEK or the rescue kernel, and create a correct `initramfs` image for the appropriate kernel version. If you boot into the rescue kernel, for example, assume you are running the RHCK version `3.10.0-514.el7.x86_64`. You can then run `dracut` with the `--force` option to overwrite the previous, incorrectly created `initramfs` image as follows:

```
# dracut --force /boot/initramfs-3.10.0-514.el7.x86_64.img
```

As mentioned previously, this form of the `dracut` command uses the currently running kernel version to create the `initramfs` image. Because RHCK version `3.10.0-514.el7.x86_64` is running, this will create the proper `initramfs` image, with the prior, incorrect `initramfs` image overwritten with the `--force` option.

Alternatively, assume you boot into UEK version `4.1.12-61.1.18.el7uek.x86_64` to correct the issue. You then must specify a second parameter to indicate which kernel version the `initramfs` image is being created for, because the RHCK is not running. The following example shows the `initramfs` image name as the first parameter (`initramfs-3.10.0-514.el7.x86_64.img`), and the RHCK version as the second parameter (`3.10.0-514.el7.x86_64`). The `--force` option again is necessary to overwrite the previous, incorrect `initramfs` image:

```
# dracut --force /boot/initramfs-3.10.0-514.el7.x86_64.img 3.10.0-514.el7.x86_64
```

This form is useful for creating various kernel version images, whether running or not, and also makes it very explicit about exactly which kernel version image is being created.

The following is a simpler way to create an `initramfs` image of the form `initramfs-<kernel version>.img` in the default `/boot` location by only specifying the kernel version along with the `--kver` option:

```
# dracut --kver 3.10.0-514.el7.x86_64
```

This will create `initramfs-3.10.0-514.el7.x86_64.img` in `/boot` for RHCK version `3.10.0-514.el7.x86_64`. If `initramfs-3.10.0-514.el7.x86_64.img` already exists (as in the troubleshooting example), add the `--force` option:

```
# dracut --force --kver 3.10.0-514.el7.x86_64
```

There are many command line options available for creating `initramfs` images. See the `dracut.cmdline(7)` man page for details.

`dracut` configuration files allow the specification of parameters to `dracut` other than real-time on the command line. Relevant files are:

```
/etc/dracut.conf
/etc/dracut.conf.d/*.conf
/usr/lib/dracut/dracut.conf.d/*.conf
```

Files in `/etc/dracut.conf.d/*.conf` take precedence over those in `/usr/lib/dracut/dracut.conf.d/*.conf` if they have the same name. Parameters specified in these files take precedence over any specified in `/etc/dracut.conf`, which is an older configuration file. It is recommended to use `/etc/dracut.conf.d/*.conf` for your own configuration settings.

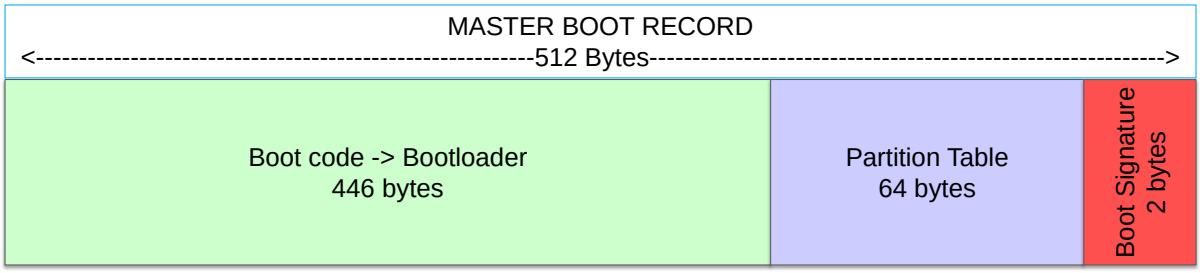
The following is a sample list of files in `/usr/lib/dracut/dracut.conf.d`:

```
# ls -l
...
-rw-r--r--. 1 root 524 Nov  4  2016 01-dist.conf
-rw-r--r--. 1 root  22 Sep  2  2016 01-microcode.conf
-rw-r--r--. 1 root  26 Nov  4  2016 02-rescue.conf
-rw-r--r--. 1 root  65 Sep  6  2016 50-nss-softokn.conf
```

Files are processed in alphanumeric order. Command line parameters take precedence over *.conf file values specified.

See the `dracut.conf(5)` man page for details.

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 further bootloader code somewhere else on the disk or on another disk. The next 64 bytes contain the partition table for the disk. The last two bytes are the boot signature, 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  01_users  20_linux_xen     30_os-prober  41_custom
00_tuned   10_linux  20_ppc_terminfo  40_custom     README
```

The scripts in this directory are read in alphanumeric 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: The 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="rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

If changes are made to any of these parameters, you need to run grub2-mkconfig to regenerate 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 setting shown on the slide 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`. The default value is `saved`. When set to `saved`, the value associated with the `saved_entry` directive in the GRUB 2 environment file, `/boot/grub2/grubenv`, determines the kernel that will be loaded. To view this setting, run the following:

```
# grub2-editenv list
saved_entry=Oracle Linux Server (4.1.12-112.16.4.el7uek.x86_64
with Unbreakable Enterprise Kernel) 7.5
```

- The default value for `saved_entry` is determined by the values of the `UPDATEDEFAULT` and `DEFAULTKERNEL` directives in `/etc/sysconfig/kernel`. The following example specifies the most recently installed UEK as the default:

```
# cat /etc/sysconfig/kernel
# UPDATEDEFAULT specifies if new-kernel-pkg should make
# new kernels the default
UPDATEDEFAULT=yes

# DEFAULTKERNEL specifies the default kernel package type
DEFAULTKERNEL=kernel-uek
```

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 the 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 this lesson.
- `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.

Unified Extensible Firmware Interface (UEFI) Overview

- UEFI is a specification implemented in firmware, replacing Basic Input/Output System (BIOS) firmware.
  - A newer method for booting systems
  - Interface between system hardware and the Operating System
- Some BIOS constraints
  - 16-bit processing
  - 2 TiB drive size limit
- UEFI improvements over BIOS
  - Can boot from drives over 2 TiB
  - More memory along with 32-bit and 64-bit processing
  - Greater security—Secure Boot available
- Many UEFI systems can boot in legacy BIOS mode.
- It requires GUID Partition Table (GPT) vs. MBR partition table used with BIOS.

Intel originally developed the Extensible Firmware Interface (EFI) specification and later joined other companies in the development of the Unified Extensible Firmware Interface (UEFI) specification. UEFI is overseen by the Unified Extensible Firmware Interface Forum (uefi.org).

The desire was to overcome BIOS constraints and UEFI is seen as a replacement for BIOS. Along with other improved features, UEFI provides greater security with the optional/additional feature of Secure Boot.

UEFI is implemented in firmware in accordance with the UEFI Specification (see http://uefi.org/specifications). Many UEFI systems can boot in legacy BIOS mode for backward compatibility via a Compatibility Support Module (CSM). Intel plans to end support for legacy BIOS mode booting by 2020.

Globally Unique Identifier (GUID) Partition Table: GPT

- Type of partition table required by UEFI
- Developed by Intel and became part of the UEFI specification
- Globally Unique Identifier (GUID) - 128 bit number
  - Identifies partition types and provides unique IDs for each partition
  - GPT header also includes a disk GUID
- Some MBR constraints
  - Four primary partitions or three primary and one extended partition
  - No data checking or redundancy
  - 2 TiB drive size maximum
- GPT enhancements over MBR
  - Larger number of partitions (128 maximum)
  - Cyclic Redundancy Checks (CRCs) used for data integrity checking
  - Partition table duplicated for backup
  - > 2 TiB drive sizes supported

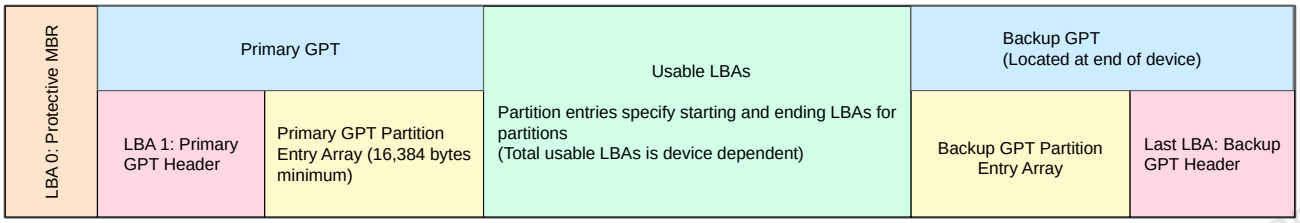
The Globally Unique Identifier (GUID) Partition Table (abbreviated "GPT") layout was initiated by Intel and is now part of the UEFI specification.

In GPT, GUIDs are 128 bit numbers that identify partition types, provide unique IDs for every partition, and provide disk identifiers. An EFI System partition has a GUID partition type of C12A7328-F81F-11D2-BA4B-00A0C93EC93B.

GPT provides enhancements over MBR, such as larger drive sizes, more partitions, and data protection.

GPT Layout

- 64-bit Logical Block Addresses (LBAs) used



The GPT layout uses 64 bit Logical Block Addresses.

LBA 0 has a "protective MBR" when using the GPT layout. This protective MBR identifies a single partition encompassing all disk space (starting after the protective MBR - up to the limit of space that MBR can manage), with a partition type of GPT. This is intended to prevent operating system tools/utilities/commands that do not work with GPT from modifying the disk and damaging data.

LBA 1: Start of the primary GPT. The primary GPT Header is located at LBA 1 followed by the primary GPT Partition Entry Array. In addition to other information, the GPT Header specifies the first and last usable LBAs on the device and has checksums for data integrity. The Partition Entry Array contains a set of GPT Partition Entries. Among other things, each entry contains a GUID defining the partition type, a unique GUID for the partition itself (every partition has a unique identifier), and the starting and ending LBAs of the partition.

The primary GPT is followed by the usable LBAs for partition data. The number of usable LBAs will vary, depending on the size of the disk.

A backup GPT follows the last usable LBA. The backup GPT Partition Entry Array is first, followed by the backup GPT Header, which is located at the last LBA of the device. The backup GPT can be used to restore the primary GPT if it becomes corrupted, and vice versa. CRCs and other information are checked to determine the validity of the primary and backup GPTs. The primary and backup GPTs are required to be kept in sync.

Installation and Administration with UEFI Using a Local Disk

- UEFI booting requires system installation with UEFI firmware present. UEFI mode is detected at installation time.
- A GPT is automatically set up at install time when in UEFI mode.
- Install Oracle Linux with a /boot/efi partition. Automatic partitioning gives 200 MiB size
  - The file system type is "EFI System Partition" (ESP).
  - The files required for UEFI support are installed in the /boot/efi path.
- The `/boot/efi/EFI/redhat` directory includes:
  - A first-stage bootloader (shim)
  - GRUB 2 bootloader (`grubx64.efi`)
  - GRUB 2 configuration file (`grub.cfg`)
- The grub file (user settings for `grub.cfg`) remains in `/etc/default` (same as in BIOS mode)
- To update `grub.cfg` with UEFI:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

ORACLE

To boot in UEFI mode, Oracle Linux is installed on a system with UEFI firmware implemented. The system detects that it is in UEFI mode and uses the GUID Partition Table (GPT) for the system drive. GPT is required by UEFI. With automatic partitioning, the installer will set up an EFI System Partition (ESP) of 200 MiB on the `/boot/efi` mount point, which will contain the files needed for UEFI booting. With manual partitioning, the ESP must be explicitly specified.

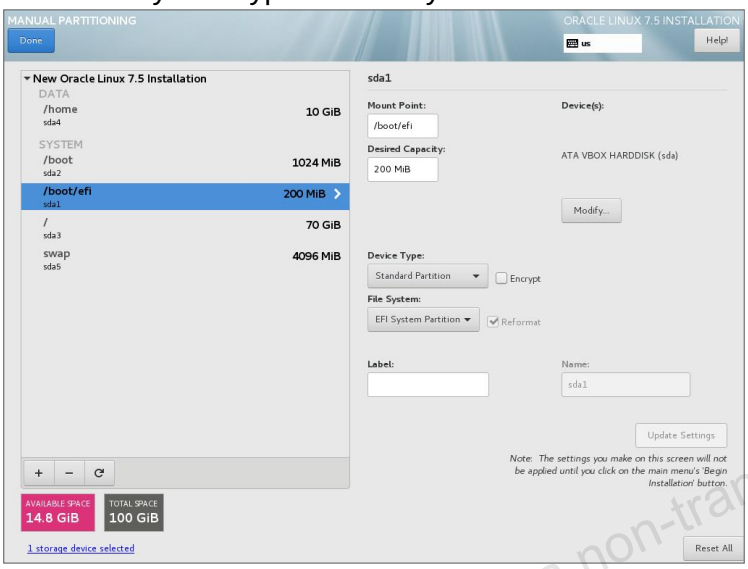
The `/boot/efi/EFI/redhat` directory contains a first-stage bootloader (shim), the GRUB 2 bootloader (`grubx64.efi`), and the GRUB 2 configuration file (`grub.cfg`). This location for `grub.cfg` differs from its location in `/boot/grub2` when in BIOS mode. The grub file, which contains user settings for `grub.cfg`, remains in the `/etc/default` directory - the same location as it is in BIOS mode - if `grub.cfg` needs to be updated.

To rebuild `grub.cfg` (after modifying `/etc/default/grub`), use the `grub2-mkconfig` command, specifying the output file as `/boot/efi/EFI/redhat/grub.cfg`, instead of `/boot/grub2/grub.cfg`:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

Installation with UEFI: EFI System Partition (ESP)

- ESP is shown with a file system type of "EFI System Partition":



The slide shows a partition layout in the Anaconda installer with the required EFI System Partition highlighted. The mount point is /boot/efi and the file system type is "EFI System Partition" (ESP).

The ESP in this example has 200 MiB allocated to it.

The `efibootmgr` Utility

- `efibootmgr` is a utility that can help manage the boot process.
  - Provides a boot menu showing boot entries
- `efibootmgr` allows manipulation of boot entries. For example, you can:
  - Alter the boot order
  - Create boot entries
  - Remove boot entries
  - Specify the boot entry for the next boot
- To view a summary of boot entries, run `efibootmgr` with no options:

```
# efibootmgr
```

- To view boot entries with more detail, add the `-v` option:

```
# efibootmgr -v
```

- See the `efibootmgr` man page for a description of options to manage boot entries.

The `efibootmgr` utility allows administrators to view a boot menu and manipulate boot entries. For example, the boot order can be changed, boot entries can be created and removed, and the boot entry for the next boot can be specified.

To view a summary of boot entries, run the `efibootmgr` with no options:

```
# efibootmgr
```

To view boot entries with more detail, add the `-v` option:

```
# efibootmgr -v
```

See the `efibootmgr` man page for a description of options to manage boot entries.

The Oracle Linux 7 Boot Process: UEFI Mode Using Local Disk

1. UEFI firmware performs POST.
2. A first-stage bootloader is loaded from the EFI System Partition (ESP).
3. The first-stage bootloader loads the GRUB 2 bootloader, found in the ESP.
4. The boot process continues with the same steps as for BIOS mode booting:
    a. GRUB 2 bootloader loads the `vmlinuz` kernel image.
    b. GRUB 2 extracts the contents of the `initramfs` image.
    c. The kernel loads driver modules from `initramfs`.
    d. The kernel starts the system's first process, `systemd` .
    e. The `systemd` process takes over.

The slide describes the UEFI boot process when using a local disk. UEFI mode booting differs from BIOS mode booting in its initial steps. UEFI does not use an MBR like BIOS, but looks for and accesses an EFI System Partition (ESP) for files required for booting. The ESP resides on a device which uses a GUID Partition Table (GPT), which is required by UEFI.

UEFI performs POST and then loads a first-stage bootloader from the ESP. The first-stage bootloader then loads the GRUB 2 bootloader, located on the ESP in the `/boot/efi/EFI/redhat` directory.

After GRUB 2 is loaded, the boot process follows the same steps as for BIOS mode booting.

Secure Boot with UEFI

- Secure Boot is an optional feature with UEFI.
  - Enabled in UEFI firmware interface for a given platform
- It provides greater security than was available with BIOS (avoids malware attacks).
- The boot process is the same as UEFI booting without Secure Boot, except:
  - Components must be signed and authenticated in order to be loaded/executed
- A “chain of trust" is established.
  - First-stage bootloader (shim) - signed/authenticated - loads GRUB 2 bootloader
  - GRUB 2 bootloader - signed/authenticated - loads the kernel
  - kernel - signed/authenticated and executed - loads signed/authenticated kernel modules only

Secure Boot is an additional/optional feature associated with UEFI, intended to help avoid execution of malware during the boot process.

Secure Boot steps are identical to UEFI booting without Secure Boot, with the important exception that it requires components to be signed and authenticated in order to be loaded/executed (private/public key pairs are used for authentication).

A "chain of trust" is established, beginning with a first-stage bootloader, shim (signed by Microsoft and Oracle). When shim is authenticated, the GRUB 2 bootloader (signed by Oracle), is loaded after being authenticated by shim. GRUB 2 then validates the kernel signature (signed by Oracle), allowing the kernel to be executed. All kernel modules (included with the kernel RPM and those used with Oracle Ksplice) have Oracle signatures and the (signed/authenticated) running kernel must validate them or they will not be loaded.

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`. Example:

```
# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-4.1.12-112.16.4.el7uek.x86_64 root=UUID=... ro 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. However, these settings are not persistent 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="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 are viewable after boot.

GRUB 2 Configuration File

```
### BEGIN /etc/grub.d/10_linux ###
menuentry 'Oracle Linux Server (4.1.12-112.16.4.el7uek.x86_64 with Unbreakable Enterprise
Kernel) 7.5' --class oracle --class gnu-linux --class gnu --class os --unrestricted
$menuentry_id_option 'gnulinux-4.1.12-112.16.4.el7uek.x86_64-advanced-274e601a-635a-4253-
839c-2747d2be1b4b' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1'  9b0dc085-a263-41f0-
98fb-f368987f4feb
        else
          search --no-floppy --fs-uuid --set=root 9b0dc085-a263-41f0-98fb-f368987f4feb
        fi
        linux16 /vmlinuz-4.1.12-112.16.4.el7uek.x86_64 root=UUID=274e601a-635a-4253-839c-
2747d2be1b4b ro rhgb quiet
        initrd16 /initramfs-4.1.12-112.16.4.el7uek.x86_64.img
}
```

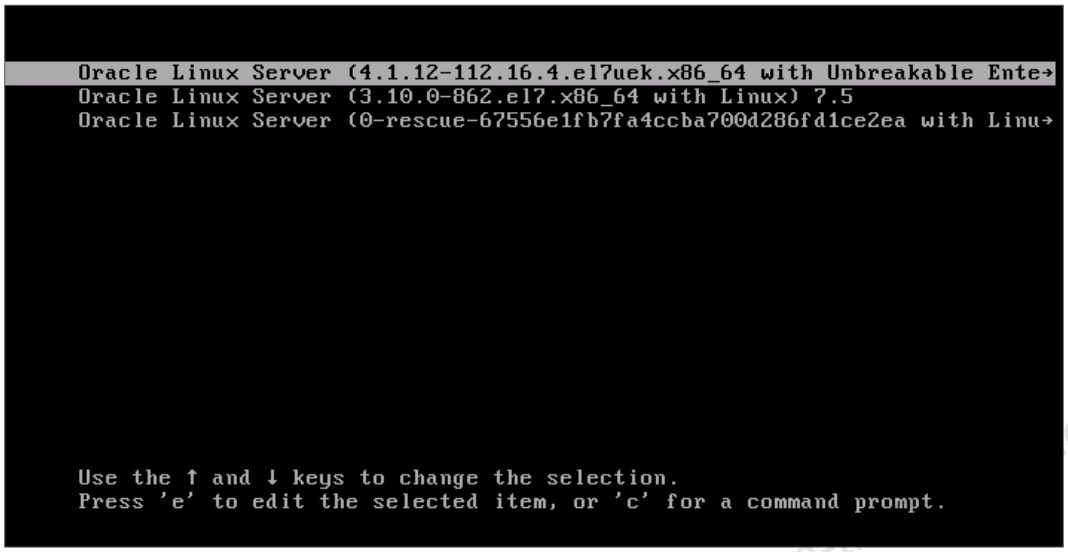

The GRUB 2 configuration file, `/boot/grub2/grub.cfg`, contains `menuentry` stanzas, which represent an installed Linux kernel. Each stanza begins with the `menuentry` keyword with options. Each `menuentry` is also a single boot menu entry in the GRUB 2 menu. The associated block of code is enclosed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

The screen shows a sample `menuentry` stanza for the Unbreakable Enterprise Kernel (UEK), version `4.1.12-112.16.4.el7uek.x86_64`. Indent everything within the braces 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. That is, 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:

- The Unbreakable Enterprise Kernel (UEK) (`4.1.12-112.16.4.el7uek.x86_64`)
- The Red Hat Compatible Kernel (RHCK) (`3.10.0-862.el7.x86_64`)
- The rescue kernel

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

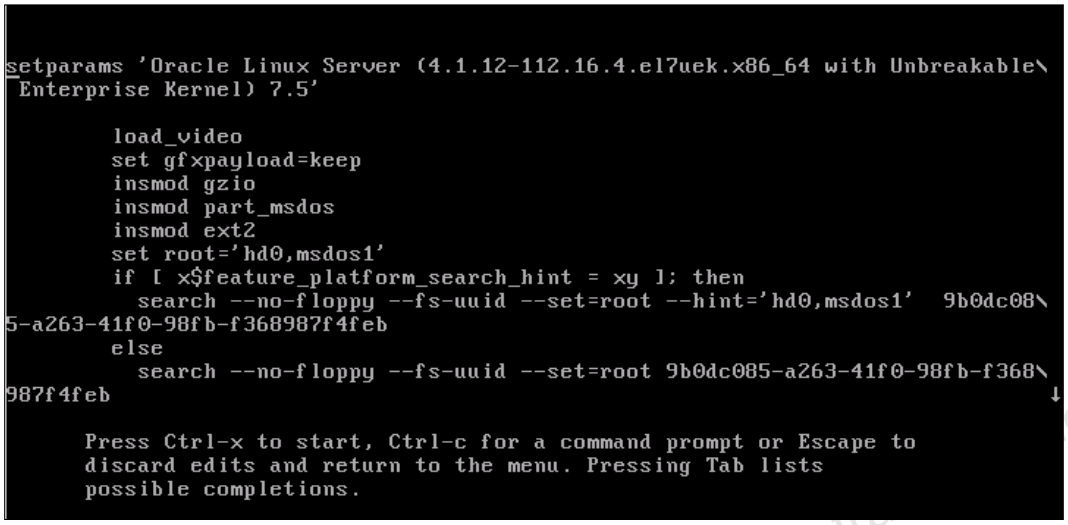
```
# grep '^menuentry' /boot/grub2/grub.cfg
menuentry 'Oracle Linux Server (4.1.12-112.16.4.el7uek.x86_64 with
Unbreakable Enterprise Kernel) 7.5'... {
menuentry 'Oracle Linux Server (3.10.0-862.el7.x86_64 with Linux)
7.5'... {
menuentry 'Oracle Linux Server (0-rescue-...with Linux) 7.5'... {
```

In 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 when 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 regenerate `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.
```

Press Esc to exit the GRUB 2 command-line interface and return to the previous screen.

systemd Introduction

- 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. For example:
  - services (*name*.service)
  - targets (*name*.target)
  - devices (*name*.device)
  - file system mount points (*name*.mount)
  - sockets (*name*.socket)

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 included with installed RPM packages.
- /run/systemd/system: systemd units created by a running program. These take precedence over units in the /usr/lib/systemd/system directory.
- /etc/systemd/system: systemd units created and managed by the system administrator. These take precedence over all other units.

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

- **Service units:** 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 units:** 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 units:** 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 units:** Expose kernel devices in `system`. They 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 units:** 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 units:** 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 units:** 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 units:** 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 units:** 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 units:** 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 units:** 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 units:** 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
pev.mqueue.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.

The systemd system and service manager provides the following features:

- Listening sockets are created at boot time for certain system services. These services can start in parallel and can also be restarted without losing any message content.
- System services can be started on-demand when a client attempts to communicate, when a piece of hardware becomes available, and when a file or directory state changes.
- 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. 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, the unit load state, the high-level (ACTIVE) and low-level (SUB) unit activation state, and a description of the service unit are displayed. 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 `iscsid` service:

```
# systemctl status iscsid
● iscsid.service - Open-iSCSI
...
```

- 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 `iscsid.service` service unit. You can omit the `.service` extension:

```
# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; disabled;
vendor preset: disabled)
   Active: inactive (dead)
     Docs: man:iscsid(8)
           man:iscsiadm(8)
```

The following information is among that available for a 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 time stamp
- `Main PID:` The Process ID of the corresponding system service and the service name
- `Status:` Status information about the system service
- `Process:` Process information including related processes
- `CGroup:` Control Groups information

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
 Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled;
vendor preset: enabled)
   Active: active (running) since <date_time; time> ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 906 ExecStart=/usr/sbin/sshd $OPTIONS (code=exited,
status=0/SUCCESS)
 Main PID: 929 (sshd)
   CGroup: /system.slice/sshd.service
           └─929 /usr/sbin/sshd
<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 iscsid
disabled
```

Starting and Stopping Services

service Utility	systemctl Utility	Description
service *name* start	systemctl start *name*	Start a service
service *name* stop	systemctl stop *name*	Stop a service
service *name* restart	systemctl restart *name*	Restart a service
service *name* condrestart	systemctl try-restart *name*	Restart a service if it is running
service *name* reload	systemctl reload *name*	Reload a configuration
service *name* status	systemctl status *name*	Check whether a service is running
service --status-all	systemctl list-units --type service --all	Display the status of all services

In previous versions of Oracle Linux, the `service` utility is used to stop and start services. This only affects services during run time, not at system boot. 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 `iscsid` service, starts the service, and checks the status a second time:

```
# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; disabled;
vendor preset: disabled)
   Active: inactive (dead)
...
# systemctl start iscsid
# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; disabled;
vendor preset: disabled)
   Active: active (running) since <date_time; time> ago
...
```

Enabling and Disabling Services

chkconfig Utility	systemctl Utility	Description
`chkconfig *name* on`	`systemctl enable *name*`	Enable a service
`chkconfig *name* off`	`systemctl disable *name*`	Disable a service
`chkconfig --list *name*`	`systemctl status *name*` `systemctl is-enabled *name*`	Check whether a service is enabled
`chkconfig --list`	`systemctl list-unit-files --type service`	List all services and check whether they are enabled

In previous versions of Oracle Linux, the `chkconfig` utility is used to enable and disable services at boot time, as well as to list their status. 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 `iscsid` service:

```
# systemctl enable iscsid
Created symlink from /etc/systemd/system/multi-
user.target.wants/iscsid.service to
/usr/lib/systemd/system/iscsid.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 `iscsid.service` for the `multi-user` target.

To disable the `iscsid` service:

```
# systemctl disable iscsid
Removed symlink /etc/systemd/system/multi-
user.target.wants/iscsid.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/runlevel5.target
/usr/lib/systemd/system/runlevel6.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 path name of these predefined systemd run level target units:

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

System-State and Equivalent Run Level Targets

System-State Targets	Equivalent Run Level Targets	Description
poweroff.target	runlevel0.target	Shut down and power off
rescue.target	runlevel1.target	Set up a rescue shell
multi-user.target	runlevel[234].target	Set up a nongraphical multi-user shell
graphical.target	runlevel5.target	Set up a graphical multi-user shell
reboot.target	runlevel6.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. The system-state target links are shown here:

```
# cd /usr/lib/systemd/system
# ls -l runlevel*.target
lrwxrwxrwx ... runlevel0.target -> poweroff.target
lrwxrwxrwx ... runlevel1.target -> rescue.target
lrwxrwxrwx ... runlevel2.target -> multi-user.target
lrwxrwxrwx ... runlevel3.target -> multi-user.target
lrwxrwxrwx ... runlevel4.target -> multi-user.target
lrwxrwxrwx ... runlevel5.target -> graphical.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 Basic System
cryptsetup.target      loaded 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 ->
/usr/lib/systemd/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
Removed symlink /etc/systemd/system/default.target.
Created symlink from /etc/systemd/system/default.target to
/usr/lib/systemd/system/multi-user.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 ->
/usr/lib/systemd/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. This 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 to 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`	Halt the system
`poweroff`	`systemctl poweroff`	Halt and power off the system
`reboot`	`systemctl reboot`	Reboot the system
`pm-suspend`	`systemctl suspend`	Suspend the system
`pm-hibernate`	`systemctl hibernate`	Put the system into hibernation
`pm-suspend-hybrid`	`systemctl hybrid-sleep`	Put the system into hibernation and suspend 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.

Running the following command saves the system state to memory:

```
# systemctl suspend
```

Most of the devices are powered off. When you bring the system back up, the system state is restored from memory and doesn't need to go through the entire boot process. Restoring system state from memory is faster than restoring from a hard disk drive; however, the system state is lost if a power outage occurs.

Run the following command to save the system state to the hard disk drive:

```
# systemctl hibernate
```

This command also powers off your machine. When you bring the system back up, the system state is restored from the hard disk drive without going through the entire boot process. This is slower than restoring from memory, but is safer because data is not lost during a power outage.

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`

Answer: c

You edit `/etc/default/grub` per the following:

The GRUB 2 configuration file is `/boot/grub2/grub.cfg`. You 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`.

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

Answer: e

Quiz

Which of the following statements are true about 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.

Answer: b, d

Only the `enable` option has an effect at boot time. The others have effects during run time.

Quiz

A GUID Partition Table (GPT) is required by UEFI.

a. True
b. False

Answer: a

Summary

In this lesson, you should have learned how to:

- Describe the Oracle Linux boot process - BIOS and UEFI
- Explain and configure the GRUB 2 bootloader
- Describe features of UEFI, the GUID Partition Table (GPT), and Secure Boot
- List and configure kernel boot parameters
- Describe the `systemd` system and service manager
- Explain `systemd` service units
- Configure services
- Describe `systemd` target units
- Configure Rescue Mode and Emergency Mode
- Perform shutdown, suspend, and reboot operations

Practice 5: Overview

This practice covers 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

6

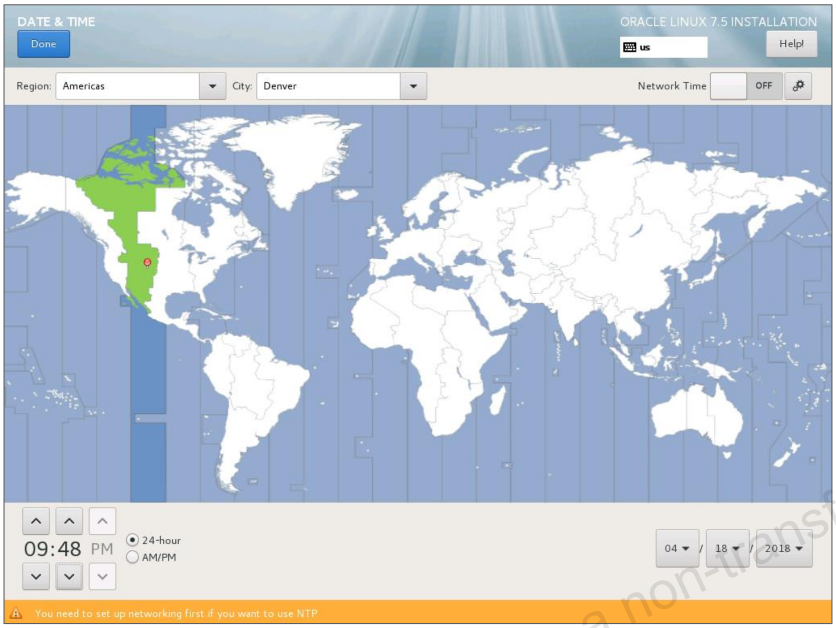
System Configuration

Objectives

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

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

Configuring System Date and Time During Installation



ORACLE

The slide shows the DATE & TIME selection window, which is accessible during installation of Oracle Linux. From this screen, you can select the time zone and 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 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 has three command-line utilities to configure the system date and time: the `date` command, the `hwclock` command, and the `timedatectl` command, which comes with `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
Tue Sep 25 09:18:51 MDT 2018
```

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

```
# date +%A
Tuesday
```

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"
Wed Sep 25 16:45:29 MDT 2019
```

To display the date one month in the past:

```
# date -d "1 month ago"
Sat Aug 25 16:46:02 MDT 2018
```

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) 11 July, 2018:

```
# date +%D -s 2018-07-11
07/11/18
```

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
Fri Aug 24 14:15:17 MDT 2018
# hwclock -s
# hwclock
Tue 25 Sep 2018 05:01:57 PM MDT  -0.639239 seconds
# date
Tue Sep 25 17:02:14 MDT 2018
```

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, RTC time, 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 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 Saving 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: Tue 2018-09-25 17:03:11 MDT
  Universal time: Tue 2018-09-25 23:03:11 UTC
        RTC time: Tue 2018-09-25 23:03:11
       Time zone: 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 2018-03-11 01:59:59 MST
                  Sun 2018-03-11 03:00:00 MDT
 Next DST change: DST ends (the clock jumps one hour backwards) at
                  Sun 2018-11-04 01:59:59 MDT
                  Sun 2018-11-04 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 the latest estimate of system clock frequency. This file is updated as necessary by the NTP daemon. 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 to display a list of peers known to the server as well as a summary of their state. The -n option causes numeric host addresses to be displayed. For example:

```
# ntpq -np
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*192.0.2.1       LOCAL(0)        11 u  130  256  377    0.109   -0.023   0.012
```

The * indicates your system is synchronized with the server with IP address 192.0.2.1. 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
synchronised to NTP server (192.0.2.1) at stratum 12
   time correct to within 27 ms
   polling server every 256 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 several 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.

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 can be started at boot time, runs in the background, and makes necessary adjustments to the system clock. Chrony also includes `chronyc`, a command-line utility used to monitor performance and make changes to operating parameters during run time.

The `chronyc` utility provides a command prompt and several commands. Use the `help` command to display the commands. For example:

```
# chronyc
...
chronyc> help
Commands:
...
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 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 communicating with the Internet the majority of time, or on private networks that have NTP servers
- Systems that seldom connect to true NTP servers
- Networks with no connectivity to systems having access to reference clocks
- Systems with dial-up connections to the Internet that are 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 a notification that the link to the Internet is present.
- **`keyfile`**: File containing the administrator password. The 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 about these files, see /usr/share/doc/initscripts*/sysconfig.txt.
- The contents of the directory vary depending on the programs installed.
- Files in the directory describe:
  - Configuration parameters
  - Arguments for respective daemons
  - Custom options for commands
  - System defaults
  - Kernel information
  - Much more

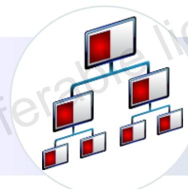
The /etc/sysconfig directory contains files that control the system configuration. See /usr/share/doc/initscripts*/sysconfig.txt for information about these files. The files found in the /etc/sysconfig directory vary, depending on what is installed on a given system. Some of the files found in the /etc/sysconfig directory are described as follows:

- **atd:** This file is used to specify additional command-line arguments for the atd daemon.
- **authconfig:** This file provides authorization directives for a host. For example, USEMKHOMEDIR=no disables creating a home directory for a user on the first login.
- **autofs:** This file defines custom options for automatically mounting devices and controlling the automounter operation.
- **crond:** This file is used to pass arguments to the crond daemon at boot time.
- **firewalld:** This file passes arguments to the firewalld daemon when a system boots.
- **ntpd:** This file passes arguments to the ntpd daemon when a system boots.

Many of the files and directories in the /etc/sysconfig directory are described further in other 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, and less to view most files.
- Utilities exist to view other files. Example:
  – lspci, free, top
- Some files can be modified to adjust kernel settings:
  – echo value > /proc/<file or path-to-file>



ORACLE

The proc directory hierarchy consists of files that provide information about system hardware and processes. These files provide insight into kernel operations and those with write permission can be modified to change the kernel configuration. 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 became quite messy over the years so Linux created the sysfs file system to clean it up.

Virtual File System

proc can be called a virtual file system because the files within it are virtual files. There is a great deal of information available from these files, but most show zero size when listed. In addition, most of the files are updated frequently.

Virtual files such as /proc/interrupts, /proc/meminfo, /proc/cpuinfo, and /proc/devices provide a view of the system's hardware. Others, such as /proc/filesystems and the files under /proc/sys, provide information about the system's configuration and allow this configuration to be modified.

Files that contain information about related topics are grouped into virtual directories. For example, a separate directory exists in `/proc` for each process that is currently running 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, `/proc/cpuinfo` contains information about the system's CPUs:

```
# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 58
model name      : Intel(R) Core(TM) i5-3550 CPU @ 3.30GHz
stepping        : 9
...
```

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 the contents of 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

The majority of files within the `/proc` directory cannot be modified. However, some are writable and can be used to adjust settings in the kernel, particularly for files in the `/proc/sys` directory. To change the value of a virtual file, use the following syntax:

```
echo value > /proc/sys/<sub-directory>/file
```

For example, to change the host name, enter:

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

Others take binary or Boolean values, with `0` meaning "off" or "false" and `1` meaning "on" or "true." In this example, network packets are not being forwarded, as shown by the `0`:

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

Change the value to `1` to allow packet forwarding:

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

Top-Level Files in `/proc`

```
1       1244    1410    243     3331    484     545     911             dma             loadavg         sys
10      1285    1415    245     3336    485     547     916             driver          locks           sysrq-trigger
103     1293    1420    246     3600    49      578     923             execdomains     mdstat          sysvipc
1049    1294    1426    25      364     5       579     924             fb              meminfo         thread-self
1051    13      1435    250     3713    50      582     927             filesystems     misc            timer_list
11      1303    1456    2597    3771    502     588     928             fs              modules         timer_stats
1152    1316    15      26      3796    522     590     947             interrupts      mounts          tty
1153    1323    16      2602    3797    523     591     acpi            iomem           mtrr            uptime
1186    1328    17      27      392     525     6       buddyinfo       ioports         net             version
12      1330    18      270     398     526     600     bus             irq             pagetypeinfo    vmallocinfo
1202    1337    2       28      402     527     601     cgroups         kallmodsyms     partitions      vmstat
1209    1338    20      280     44      528     7       cmdline         kallsyms        sched_debug     xen
1212    1345    21      281     45      529     70      consoles        kcore           scsi            zoneinfo
1215    1350    22      29      46      530     71      cpuinfo         keys            self
1219    1360    23      3       47      532     8       crypto          key-users       slabinfo
1224    1395    239     31      48      533     9       cyclicinfo      kmsg            softirqs
1227    14      24      32      481     542     907     devices         kpagecount      stat
1236    1400    241     3311    482     544     910     diskstats       kpageflags      swaps
```


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`:** Provides information for diagnosing memory fragmentation
- **`/proc/cmdline`:** Lists parameters passed to the kernel at boot time
- **`/proc/cpuinfo`:** Provides CPU information
- **`/proc/crypto`:** Provides information about installed cryptographic cyphers
- **`/proc/devices`:** Lists the names and major device numbers for all currently configured character and block devices
- **`/proc/dma`:** Lists the Direct Memory Access (DMA) channels that are currently in use
- **`/proc/execdomains`:** Lists the execution domains for binaries that the Oracle Linux kernel supports

- **/proc/filesystems:** Shows file system types that the kernel supports. Entries beginning with `nodev` are not currently being used.
- **/proc/interrupts:** Records the number of interrupts per interrupt request queue (IRQ) for each CPU since system startup
- **/proc/iomem:** Lists the system memory map for all physical devices
- **/proc/ioports:** Lists the range of I/O port addresses that the kernel uses with devices
- **/proc/kcore:** Presents a core file format of physical memory and is intended to be analyzed with debugging tools such as gdb or crash. It is not human-readable.
- **/proc/kmsg:** Records kernel-generated messages, which are in turn used by programs such as `dmesg`
- **/proc/loadavg:** Displays the system load averages (number of queued processes) for the past 1, 5, and 15 minutes, the number of running processes, the total number of processes, and the PID used for the most recent process
- **/proc/locks:** Displays information about the file locks that the kernel is currently holding on behalf of processes
- **/proc/mdstat:** Lists information about multiple-disk RAID devices
- **/proc/meminfo:** Reports the system's usage of memory in more detail than is available using the `free` or `top` commands
- **/proc/modules:** Displays information about the modules that are currently loaded into the kernel. The `lsmod` command formats and displays much of the same information.

Process Directories in `/proc`

- Process directories have names using process ID (PID) numbers.
- They provide information about processes:
  - `cmdline`: The command issued
  - `cwd`: Symbolic link to the current working directory
  - `environ`: Environment variables
  - `fd`: File descriptors
  - `maps`: Memory maps to executables and library files
  - `root`: Symbolic link to the root directory of the process
  - `status`: Run state and memory usage

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 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

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

- **`/proc/bus`:** Contains information about buses which are present. The presence or absence of actual devices affects what information is available with the `/proc/bus` hierarchy.
- **`/proc/bus/pci`**, **`/proc/bus/usb`:** Contains information about PCI or USB devices. You can view these devices by running the `cat` command on the `devices` file within these directories. Because the content of these `devices` files is not formatted for readability, the `lspci` and `lsusb` commands can be run to make it more understandable.
- **`/proc/driver`:** Contains information about drivers
- **`/proc/fs`:** Contains information about file systems
- **`/proc/irq`:** Contains information about IRQs. The affinity between IRQs and CPUs can be configured.
- **`/proc/self/net`:** Contains a series of files and directories providing detailed network configuration information. The `/proc/net` file is a symbolic link to this directory.
- **`/proc/scsi`:** Provides information about SCSI devices. `/proc/scsi/scsi` contains details about configured SCSI devices.
- **`/proc/sysvipc`:** Provides information about the usage of System V Interprocess Communication (IPC) resources for messages (msg), semaphores (sem), and shared memory (shm).
- **`/proc/tty`:** Contains information about terminal devices. The `drivers` file is a list of the current `tty` devices in use.

The /proc/sys Directory

- /proc/sys has a dual role.
- Provides information about the system
- Contains files used to enable, disable, or modify 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.
  - Some files contain Boolean values to turn features on or off.
- Subdirectories in /proc/sys include dev, fs, kernel, and net.

/proc/sys provides system information and allows you to enable, disable, or modify kernel features in real time. Kernel settings can be changed by modifying files that have write permission. The echo command is used to change /proc/sys file values. For example, to change the host name to www.example.com:

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

Other files contain binary or Boolean values. 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 are lost upon system reboot. To make configuration changes persistent, add them to files created in the /etc/sysctl.d directory.

- **/proc/sys/dev:** Contains device parameters
- **/proc/sys/fs:** Contains file system parameters
- **/proc/sys/kernel:** Contains kernel configuration parameters
- **/proc/sys/net:** Contains networking parameters

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 files created in `/etc/sysctl.d`
  - Use `sysctl -p <filename>` to make changes immediate.

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 files created in `/etc/sysctl.d`. Values added to these files take effect each time the system boots.

/usr/lib/sysctl.d contains files with kernel configuration settings placed there by application or rpm installation. This allows settings to be specified in designated files without having multiple sources modifying the now deprecated /etc/sysctl.conf file. Here are some files in /usr/lib/sysctl.d for example:

```
# ls -l
total 12
-rw-r--r--. 1 root  293 Nov  4  2016 00-system.conf
-rw-r--r--. 1 root 1205 Nov  3  2016 50-default.conf
-rw-r--r--. 1 root  499 Nov  4  2016 60-libvirtd.conf
```

Here is some content in one of these files:

```
# cat 00-system.conf
# Kernel sysctl configuration file
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Disable netfilter on bridges.
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

Files in /etc/sysctl.d are created by administrators to specify custom kernel values. Settings found in this directory take precedence over those in /usr/lib/sysctl.d. Files are processed in alphanumeric order, so it is recommended to name these files with a two-digit number followed by a dash and the name of the function being configured. This allows for proper ordering as well as keeping configuration parameters for similar functions together clearly. Files should end with ".conf".

For example, to allow IPv4 packet forwarding persistently, the file 25-ipv4_settings.conf could be created in /etc/sysctl.d with the following setting:

```
net.ipv4.ip_forward = 1
```

Though the file /etc/sysctl.conf is deprecated, the following shows a link to /etc/sysctl.conf for backward compatibility within /etc/sysctl.d:

```
# ls -l
...
lrwxrwxrwx. 1 root ... 99-sysctl.conf -> ../sysctl.conf
```

Run sysctl -p /etc/sysctl.d/<filename> to enable specified settings immediately. sysctl --system enables settings in all files in a variety of paths where these files are found. See the sysctl(8) man page for a list of more configuration file locations. Also see the sysctl.d(5) man page.

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 ~]# ls -l /sys
total 0
drwxr-xr-x.   2 root root 0 Sep 26 08:17 block
drwxr-xr-x.  31 root root 0 Sep 26 08:17 bus
drwxr-xr-x.  57 root root 0 Sep 26 08:17 class
drwxr-xr-x.   4 root root 0 Sep 26 08:17 dev
drwxr-xr-x.  19 root root 0 Sep 26 08:17 devices
drwxr-xr-x.   5 root root 0 Sep 26 08:17 firmware
drwxr-xr-x.   8 root root 0 Sep 26 08:17 fs
drwxr-xr-x.   5 root root 0 Sep 26 08:17 hypervisor
drwxr-xr-x.  10 root root 0 Sep 26 08:17 kernel
drwxr-xr-x. 157 root root 0 Sep 26 08:44 module
drwxr-xr-x.   2 root root 0 Sep 26 08:17 power
```

ORACLE

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. The 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 --file-type /sys/block/xvda

alignment_offset  device@            mq/        ro          trace/   xvda4/
badblocks         discard_alignment  power/     size        uevent   xvda5/
bdi@              ext_range          queue/     slaves/     xvda1/
capability        holders/           range      stat        xvda2/
dev               inflight           removable  subsystem@  xvda3/
```

- **/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 nonperipheral 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  parameters  sections    taint
drivers   initsize  notes      refcnt      srcversion  uevent
...
```

- **/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 `lshw` Utility

- `lshw` is a command-line tool that provides hardware configuration information.
- Example output for a virtual machine:

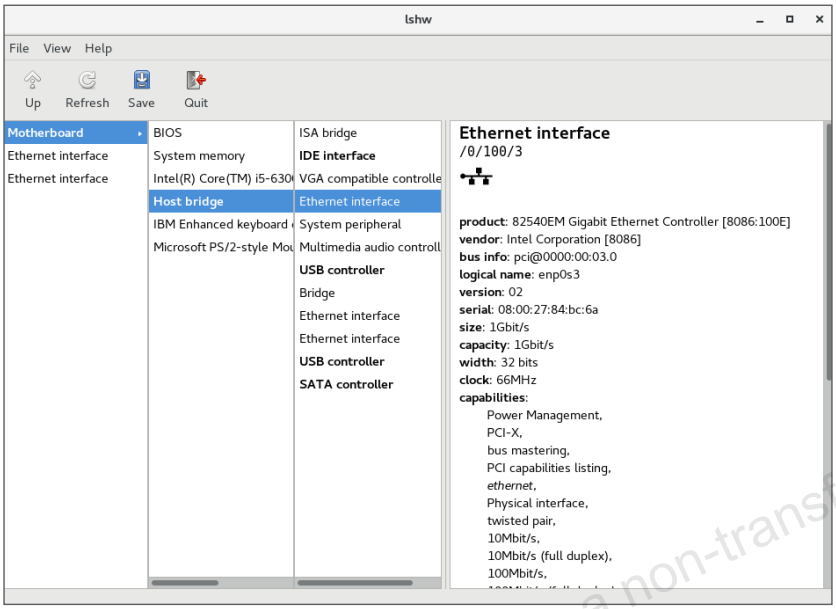
```
# lshw
host03.example.com
    description: Computer
    product: HVM domU
    vendor: Xen
    version: 4.1.3OVM
    serial: ...
    width: 64 bits
    capabilities: smbios-2.4 dmi-2.4 vsyscall32
    configuration: boot=normal uuid=...
...
```


`lshw` provides hardware information about a system. This includes information about memory, firmware, CPU, network interfaces and more. The output is displayed in a tree format.

The example in the slide shows the initial output when running `lshw` with no options.

This utility is part of the `ol7_latest` Oracle Linux yum server repository or the `ol7_x86_64_latest` Unbreakable Linux Network channel.

The `lshw-gui` Utility: Graphical Hardware Configuration Interface



The slide shows sample output from running the `lshw-gui` utility:

```
# lshw-gui
```

This provides a graphical interface for viewing hardware information. You can look through the hardware configuration by clicking items successively through the tree. You can double-click items in bold to drill down to see further items available to be viewed.

`lshw-gui` is available in the `ol7_optional_latest` Oracle Linux yum server repository or the `ol7_x86_64_optional_latest` channel on the Unbreakable Linux Network.

Quiz

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

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

Answer: a, b, c

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

Answer: a

Quiz

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

a. /etc/sysconfig

b. /proc/sys

c. /var/spool

d. /var/log

Answer: b

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

Answer: d

Summary

In this lesson, you should have learned how to describe:

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

ORACLE

Practice 6: Overview

This practice covers the following topics:

- 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

7

Package Management

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Linux package management
- Use the `rpm` utility
- Describe the Oracle Linux yum server
- Explain and configure yum repositories
- Use the `yum` utility
- Describe the Unbreakable Linux Network (ULN)
- Describe how to register a system with ULN
- List the steps to switch from Red Hat Network (RHN) to ULN
- Explain replacement of SSL certificates for ULN
- Describe Software Collections and how to use them

Package Management: Introduction

- 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
- The Oracle Linux yum server:
  - Offers a free way to install packages
  - Has free errata
- Unbreakable Linux Network (ULN):
  - Is a comprehensive resource for support subscribers
  - Offers access to software patches, updates, and fixes

The software on Linux systems comprises 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 Linux Yum Server

The Oracle Linux 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 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.

The 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
...
strace-4.12-6.el7.x86_64
basesystem-10.0-7.0.1.el7.noarch
libpeas-loader-python-1.20.0-1.el7.x86_64
perl-File-Path-2.09-2.el7.noarch
gnome-vfs2-2.24.4-14.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     : 25.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-2.8.71-9.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-5.0.2-28.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-4.1.12-112.16.4.el7uek.x86_64.rpm
```

Removing Packages

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

```
# rpm -e zsh
```

Oracle Linux Yum Server

- Oracle offers free packages from Oracle Linux installation media.
- Errata are also available from the Oracle Linux yum server.
  - Subscribe to the Oracle errata mailing list from this site.
- Packages can be accessed at:
  - https://yum.oracle.com/
- Repository files are used to provide access to yum repositories needed for your system.
  - Previously, the main repository file was `public-yum-ol7.repo`.
- A modular `yum` repository configuration provides multiple, more targeted repository files.
  - These repository files replace `public-yum-ol7.repo`.
  - Existing `public-yum-ol7.repo` files can still be used, but will no longer be updated.

The Oracle Linux 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 yum server. You can also subscribe to the Oracle Linux errata mailing list from this site. The yum server is offered without support of any kind and can be accessed at https://yum.oracle.com/.

Note that the `public-yum-ol7.repo` file will no longer be updated with current repository lists. It will still provide access to `yum` repositories that are defined in the file. See the slides referring to Modular `yum` Repository Configurations in this lesson for the modular approach to managing `yum` repositories.

The yum Configuration

- /etc/yum.conf:
  - Is the primary configuration file
  - Holds global settings
- /etc/yum.repos.d:
  - Is the directory that 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 is:

```
# 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 the authenticity of the packages. The `gpgkey` directive specifies the location of the GPG key.
- **`payload_gpgcheck`:** When set to `1`, `yum` checks the GPG signature on payloads of packages as opposed to just package headers, which **`gpgcheck`** does. This can be used in conjunction with **`gpgcheck`**.
- **`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 repositories in files in the `/etc/yum.repos.d` directory. Examples of files in this directory from the modular `yum` configuration include `oracle-linux-ol7.repo`, `uek-ol7.repo`, `and virt-ol7.repo`. In a non-modular `yum` configuration, the `public-yum-ol7.repo` file would normally be present.

The `repo` files define which repositories to use. Each `repo` file includes specifications (or definitions) for related repositories. Examples of specifications in the `oracle-linux-ol7.repo` file include `[ol7_latest]`, `[ol7_optional_latest]`, and `[ol7_addons]`. This is the `[ol7_latest]` repository definition in this file:

```
# cat oracle-linux-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum$ociregion.oracle.com/repo/OracleLinux/OL7/latest
/$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.

Example contents of a `public-yum-ol7.repo` file:

```
# cat public-yum-ol7.repo
[ol7_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=http://yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
...
[ol7_UEKR4]
...
[ol7_optional_latest]
...
[ol7_software_collections]
...
```

The `public-yum-ol7.repo` file contains many repository definitions for different groups of RPMs. This file is active when the modular `yum` repository configuration is not in effect.

Modular `yum` repository files contain fewer repository definitions, more specific to groups of RPMs. For example, the modular `yum` repository file, `oracle-linux-ol7.repo`, does not contain UEK repository definitions or a Software Collections definition as you see in the `public-yum-ol7.repo` file. The modular `yum` repository file, `uek-ol7.repo`, only contains definitions specific to UEK, for example:

```
# cat uek-ol7.repo
[ol7_UEKR5]
name=Latest Unbreakable Enterprise Kernel Release 5 for Oracle
Linux $releasever ($basearch)
baseurl=https://yum$ociregion.oracle.com/repo/OracleLinux/OL7/UEKR5
/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
...
[ol7_UEKR4]
...
```

The `baseurl` for modular `yum` repository files includes a variable for systems running in Oracle Cloud Infrastructure, `$ociregion`. Each region in Oracle Cloud Infrastructure has its own Oracle Linux yum server mirror to enable high speed access without incurring extra network charges. When launched, Oracle Linux images have a `yum` variable set in /etc/yum/vars/ociregion to ensure they connect to a mirror in the appropriate region. If a system is not running in Oracle Cloud Infrastructure, this variable will be empty.

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
Loaded plugins: ...
Installed Packages
...
NetworkManager.x86_64          1:1.10.2-13.el7    @anaconda/7.5
...
NetworkManager-libnm.x86_64    1:1.10.2-13.el7    @anaconda/7.5
...
authconfig.x86_64              6.2.8-30.el7       @anaconda/7.5
...
```

Modular `yum` Repository Configurations

- In the past, the `public-yum-ol7.repo` file contained a large number of repository definitions for many RPMs.
  - As new packages were added, more repository definitions were added to this single file.
- A modular `yum` repository definition approach is now implemented.
  - Smaller, more targeted repository files are provided.
  - This makes it easier to enable specific repositories and keep `yum` repository definitions current.
- New release RPMs are available in the `ol7_latest` repository.
  - These release RPMs let you configure and update repositories for specific RPMs you are interested in.
- The `public-yum-ol7.repo` file will no longer be updated with new repository definitions.

As new packages were added to the Oracle Linux yum server over time, more repository definitions were added to the single, monolithic `public-yum-ol7.repo` file, which contained definitions for many RPMs. By providing smaller, more targeted repository files, it is easier to enable specific repositories for the Oracle Linux yum server and to keep your `yum` repository definitions up to date.

A set of release RPMs are published on the Oracle Linux yum server in the `ol7_latest` repository. When installed, these RPMs provide repository files for more specific sets of RPMs.

The following are release RPMs provided on the Oracle Linux yum server as part of the modular `yum` repository configuration:

Software	Release RPM
Oracle Linux, UEK & Virtualization tools	`oraclelinux-release-el7, oraclelinux-release-el6`
Software Collection Library for Oracle Linux	`oracle-softwarecollection-release-el7, oracle-softwarecollection-release-el6`
Oracle OpenStack for Oracle Linux	`oracle-openstack-release-el7, oracle-openstack-release-el6`
Spacewalk Server	`oracle-spacewalk-server-release-el7, oracle-spacewalk-server-release-el6`
Spacewalk Client	`oracle-spacewalk-client-release-el7, oracle-spacewalk-client-release-el6`
Gluster Storage	`oracle-gluster-release-el7, oracle-gluster-release-el6`
Ceph Storage	`oracle-ceph-release-el7`
Oracle Instant Client	`oracle-release-el7, oracle-release-el6`
EPEL for Oracle Linux	`oracle-epel-release-el7`
Packages for Developers and Oracle Cloud Infrastructure	`oraclelinux-developer-release-el7, oraclelinux-developer-release-el6`
MySQL Community releases	`mysql-release-el7, mysql-release-el6`
Stable releases of the Go programming language	`oracle-golang-release-el7`
Stable PHP releases	`oracle-php-release-el7, oracle-php-release-el6`
Stable Node.js releases	`oracle-nodejs-release-el7, oracle-nodejs-release-el6`

The following release RPMs are for Oracle Cloud Infrastructure only:

Software	Release RPM
Oracle Linux patch repositories	`oraclelinux-patchonly-release-el7, oraclelinux-patchonly-release-el6`
Ksplice Utilities	`oracle-ksplice-release-el7, oracle-ksplice-release-el6`

Systems With Modular `yum` Repository Configurations

- You can verify whether your system is configured with the modular `yum` approach by checking for installation of the `oraclelinux-release-el7` RPM:

```
# rpm -q oraclelinux-release-el7
# oraclelinux-release-el7 ...
```

- You also will no longer have an active `public-yum-ol7.repo` file.
- Install other release RPMs you desire and `yum` repository files will be configured accordingly.
- To see a current list of installed and available release RPMs:

```
# yum list *release-el7
```

Systems using the modular `yum` repository configuration have the `oraclelinux-release-el7` RPM installed and no longer use the `public-yum-ol7.repo` file.

You can install other release RPMs associated with the packages you need and the `yum` repo files will be configured properly.

You can check for available release RPMs using the `yum list` command as shown on the slide.

Modular `yum` Repository Configurations: Legacy Systems

- For legacy Oracle Linux installations with a monolithic `public-yum-ol7.repo` file, initial new repository files are obtained by:
  - Running `yum update`, installing the `oraclelinux-release-el7` RPM, or installing one of the other release RPMs
  - These repository files have a `.disabled` extension initially.
- The script `/usr/bin/ol_yum_configure.sh` is installed and must be run as `root`, to complete the transition.
  - Currently enabled repositories are checked.
  - Needed release RPMs are installed, providing further required repositories.
  - New repository files will then have `.repo` extensions and be active.
  - The `public-yum-ol7.repo` file will no longer be used.
- If no action is taken, the existing `public-yum-ol7.repo` file will work, but will not be updated.

For systems with an existing monolithic `public-yum-ol7.repo` file, you can obtain initial new repository files by running `yum update`, installing the `oraclelinux-release-el7` RPM, or installing one of the other release RPMs. You will then have a new set of repository files with `.disabled` extensions.

You then must run the `/usr/bin/ol_yum_configure.sh` script to complete the transition to the new `.repo` files. This will check for currently enabled repositories, installing release RPMs to provide necessary repository files. The new repository files will then have `.repo` extensions and will be active. The existing `public-yum-ol7.repo` file will no longer be used.

If no action is taken to configure legacy systems for the modular `yum` configuration, the `public-yum-ol7.repo` file will still work, as the content of the repositories and the URLs where they are located will not change. Updated versions of this file with updated repository lists simply will not be available.

The 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, and remove packages:

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

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 or installed on your system. 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
Loaded plugins: ...
at-3.1.13-23.el7.x86_64 : Job spooling tools
...
```

Checking for Updates

Use the following command to check for available updates to installed packages:

```
# 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 updates` returns the same output.

Updating Packages

You can update individual packages, multiple packages, or all packages. Dependent packages are also updated.

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

Use the following command to update all packages and their dependencies:

```
# yum update
```

Installing Packages

Use the `yum install` command to install a new package. Dependent packages are also installed. 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

Use the `yum remove` command to remove a package. 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 group list
# yum group info groupname
# yum group install groupname
# yum group update groupname
# yum group remove 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 group list
```

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

```
# yum group info <groupname>
```

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

```
# yum group install <groupname>
```

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

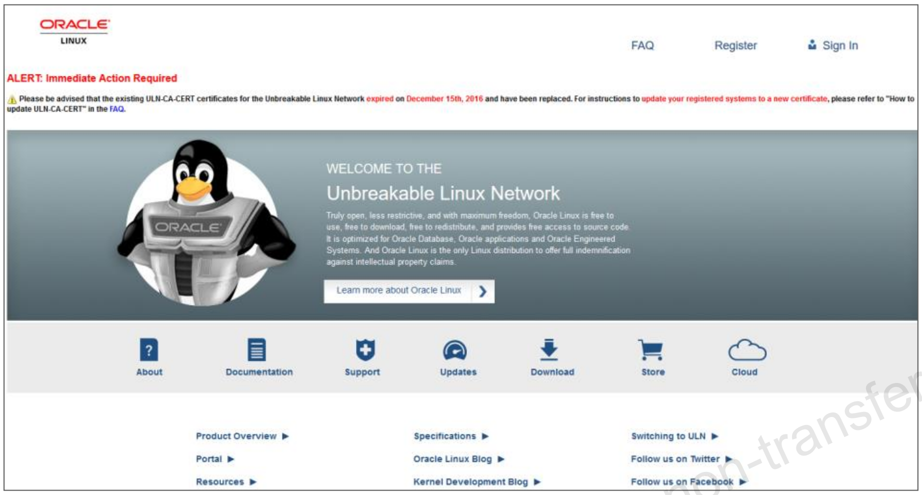
```
# yum group update <groupname>
```

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

```
# yum group remove <groupname>
```

Unbreakable Linux Network (ULN)

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



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, Ksplice, 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/.

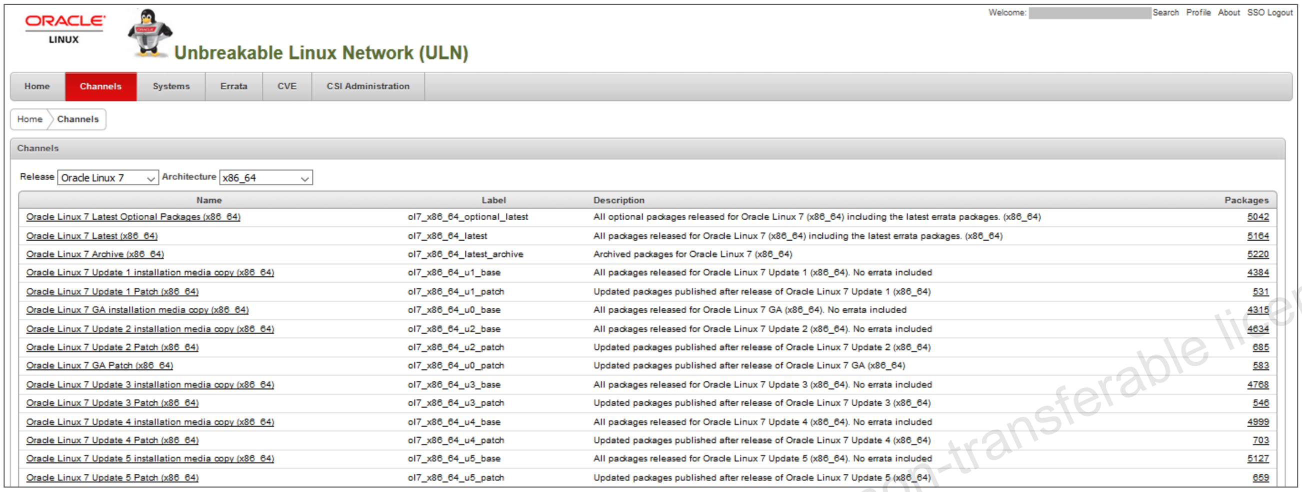
The Update Agent (`yum`)

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 one or more ULN channels for each system. The ULN channel 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 packages 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 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)	5042
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)	5164
Oracle Linux 7 Archive (x86_64)	ol7_x86_64_latest_archive	Archived packages for Oracle Linux 7 (x86_64)	5220
Oracle Linux 7 Update 1 installation media copy (x86_64)	ol7_x86_64_u1_base	All packages released for Oracle Linux 7 Update 1 (x86_64). No errata included	4384
Oracle Linux 7 Update 1 Patch (x86_64)	ol7_x86_64_u1_patch	Updated packages published after release of Oracle Linux 7 Update 1 (x86_64)	531
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 Update 2 installation media copy (x86_64)	ol7_x86_64_u2_base	All packages released for Oracle Linux 7 Update 2 (x86_64). No errata included	4634
Oracle Linux 7 Update 2 Patch (x86_64)	ol7_x86_64_u2_patch	Updated packages published after release of Oracle Linux 7 Update 2 (x86_64)	685
Oracle Linux 7 GA Patch (x86_64)	ol7_x86_64_u0_patch	Updated packages published after release of Oracle Linux 7 GA (x86_64)	583
Oracle Linux 7 Update 3 installation media copy (x86_64)	ol7_x86_64_u3_base	All packages released for Oracle Linux 7 Update 3 (x86_64). No errata included	4768
Oracle Linux 7 Update 3 Patch (x86_64)	ol7_x86_64_u3_patch	Updated packages published after release of Oracle Linux 7 Update 3 (x86_64)	546
Oracle Linux 7 Update 4 installation media copy (x86_64)	ol7_x86_64_u4_base	All packages released for Oracle Linux 7 Update 4 (x86_64). No errata included	4999
Oracle Linux 7 Update 4 Patch (x86_64)	ol7_x86_64_u4_patch	Updated packages published after release of Oracle Linux 7 Update 4 (x86_64)	703
Oracle Linux 7 Update 5 installation media copy (x86_64)	ol7_x86_64_u5_base	All packages released for Oracle Linux 7 Update 5 (x86_64). No errata included	5127
Oracle Linux 7 Update 5 Patch (x86_64)	ol7_x86_64_u5_patch	Updated packages published after release of Oracle Linux 7 Update 5 (x86_64)	659

ORACLE

There are more than 400 unique channels supported by ULN. The slide shows some of the channels available for Oracle Linux 7 x86_64 architecture on ULN. The architectures currently supported are i386, x86_64, aarch64 (starting with UEK R5), and ia64 (starting with Oracle Linux 4 update 6 and Oracle Linux 5 update 4).

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. Descriptions of some channels are provided here:

`_latest` Channels

The `_latest` channels provide RPMs released in the `_base` and `_patch` channels for the latest update (that is, the version of any RPM downloadable on the `_latest` channels is always the most recently 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 `ol7_<arch>_latest` channel for Oracle Linux 7 contains the combination of the latest `ol7_<arch>_u<number>_base` and `ol7_<arch>_u<number>_patch` channels.

`_latest_archive` Channels

The `_latest_archive` channels provide older versions of RPMs that have been updated in the `_latest` channels. Packages are moved to a `_latest_archive` channel when newer versions of the same packages are added to the `_latest` channel.

`_base` Channels

The `_base` channels (also known as installation media channels) provide RPMs for each major version and minor update of Oracle Linux as released on their respective installation media (DVD or ISO). There is a `_base` channel for each update level of an Oracle Linux release. 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 fixes provided, these channels always provide the most recent version of such a package.

`_addons` Channels

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

`_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. The current `ol7_<arch>_u<number>_base` and `ol7_<arch>_u<number>_patch` channels remain available and do not include the new updates, making it 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 `ol7_<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 to the `ol7_<arch>_latest` channel, for example, you must subscribe to the `ol7_<arch>_u<number>_patch` and `ol7_<arch>_u<number>_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 recommends that you keep your system subscribed to the `_latest` channel. If you unsubscribe from the `_latest` channel, your system will become vulnerable to security-related issues when a new update is released.

Some of the other channels for Oracle Linux 7 are:

- **`ol7_x86_64_optional_latest`**: Optional packages including the latest errata
- **`ol7_x86_64_MySQL<nn>_community`**: Latest MySQL n.n packages
- **`ol7_x86_64_Dtrace_userspace`**: Latest Dtrace user space tools
- **`ol7_x86_64_ksplice`**: Oracle Ksplice clients, updates, and dependencies
- **`ol7_x86_64_UEKR4_OFED`**: Latest OFED tools for UEK R4

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_ol7.tgz`
     - `uln_register-gnome_ol7.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).

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. For Oracle Linux 7, the file names to download are:

- `uln_register_ol7.tgz`
- `uln_register-gnome_ol7.tgz`

2. Extract the Registration Packages

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

```
# tar -xzf uln_register_ol7.tgz
# tar -xzf uln_register-gnome_ol7.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

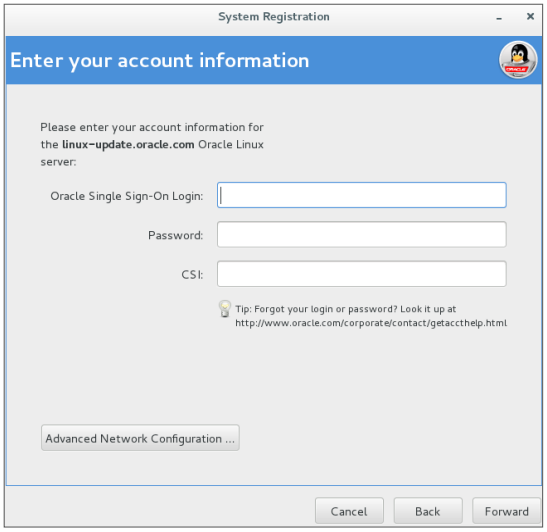
To register your system, run the following command as the `root` user in a terminal window or on the command line:

```
# 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.

Registering Your System with ULN

These are the initial screens when you run the `uln_register` command from a terminal window in an X Window system.



You run `uln_register` to register a system with ULN. Running this command from an X Window system brings up the GUI interface.

After an initial window describing the registration process, click Forward to get to the window on the left of the slide. This window prompts for login information, which requires an Oracle support subscription. CSI stands for Customer Support Identifier.

After you have entered your accounting information for ULN, you are prompted to create a profile. Create a system name that is used by ULN to identify this system. This system name does not need to be the same as the host name. Select the check boxes, as shown, to provide information about the system hardware and the packages installed on the system.

Your system information is then sent to ULN, which takes a few minutes. Newly registered systems are automatically subscribed to two ULN channels:

- The Oracle Linux latest channel for your version of Oracle Linux and the system architecture
- The latest UEK channel that is supported by your version of Oracle Linux

After your system is registered with ULN, you can use the ULN web interface to subscribe to additional channels and download errata and other software packages.

See the Unbreakable Linux Network User's Guide for more information on using ULN:
https://docs.oracle.com/cd/E52668_01/E39381/html/index.html

Replacement of SSL Certificates for ULN

- Oracle has replaced Symantec-branded certificates with Digicert-branded certificates across all of its infrastructure.
  - Change in certificates occurred on October 9, 2018.
- It requires that client certificates on all Oracle Linux systems directly subscribed to and receiving updates from ULN be updated.
- Connection to ULN will not be possible after October 9, 2018 until certificates are updated.
- For Oracle Linux 7, the following packages must be updated to these versions or later:
  - `rhn-client-tools-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-setup-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-check-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-setup-gnome-2.0.2-21.0.9.el7.noarch.rpm` (only required if a previous version is already installed)
  - If these packages were installed before October 9, 2018, no further action is required.

Oracle replaced Symantec-branded certificates with Digicert-branded certificates across all of its infrastructure to prevent trust warnings once the Symantec root certificate authority is removed from several web browsers, including Firefox and Chrome. This change in certificates occurred on October 9, 2018.

Because of the nature of how Oracle Linux systems connect to Unbreakable Linux Network (ULN), this change requires that client certificates on all Oracle Linux systems directly subscribed to and receiving updates from ULN be updated.

The four packages shown on the slide must be updated to the indicated versions or later. If this was done before October 9, 2018, no further action is required.

Replacement of SSL Certificates for ULN

- If the four packages were not updated as indicated before October 9, 2018, connection to ULN will not be possible until certificates are manually updated.
- In this case, the SSL certificate on Oracle Linux systems must first be replaced by following these steps:

```
# cp /usr/share/rhn/ULN-CA-CERT /usr/share/rhn/ULN-CA-CERT.old
# wget https://linux-update.oracle.com/rpms/ULN-CA-CERT.sha2
# cp ULN-CA-CERT.sha2 /usr/share/rhn/ULN-CA-CERT
```

If the four packages already shown are not updated before October 9, 2018, the SSL certificate on Oracle Linux systems must be manually replaced in order to connect to ULN.

In this case, before updating the SSL certificate, you will receive one of the following errors:

```
The certificate /usr/share/rhn/ULN-CA-CERT is expired. Please ensure you
have the correct certificate and your system time is correct.
```

or

```
There was an SSL error: [('SSL routines', 'SSL3_GET_SERVER_CERTIFICATE',
'certificate verify failed')]

A common cause of this error is the system time being incorrect. Verify
that the time on this system is correct.
```

Follow the three steps on the slide to replace the SSL certificate.

Replacement of SSL Certificates for ULN

- After manually replacing the SSL certificate on your Oracle Linux systems, install the four packages as part of your patching cycle:
  - `rhn-client-tools-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-setup-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-check-2.0.2-21.0.9.el7.noarch.rpm`
  - `rhn-setup-gnome-2.0.2-21.0.9.el7.noarch.rpm` (only required if a previous version is already installed)
  - This presumes these four packages were not installed before October 9, 2018.
- Systems managed by Oracle Enterprise Manager or that are subscribed to a local Spacewalk instance are unaffected by this SSL certificate issue.

After the SSL certificate is manually replaced on your Oracle Linux systems, the four packages shown on the slide must then be updated to the versions shown or later.

See this blog posting for lists of needed packages for other versions of Oracle Linux: https://blogs.oracle.com/linux/action-required:-replacement-of-ssl-certificates-for-the-unbreakable-linux-network

The Software Collection Library for Oracle Linux

- Set of software providing support mainly for developers
- Provides later versions of software than are available from a standard Oracle Linux installation (for example: GCC, Python, Perl, and `strace`)
  - Developers may need to work with more current features that are available.
- Software from a collection is installed and executed separately from versions installed with Oracle Linux.
- Collections are available for Oracle Linux 6 and Oracle Linux 7. Check for differences as necessary.
- Packages are available on the ULN and Oracle Linux yum server.
- Example collection: `devtoolset-7`
  - Set of tools for developers

The Software Collection Library for Oracle Linux provides a set of software to assist developers by making later versions of software available than are provided by a standard Oracle Linux installation. This makes more current features available that developers may need or desire to work with.

Packages can be obtained from either the ULN or the Oracle Linux yum server. Software from a collection is installed in a separate path and executed independently from software versions installed with Oracle Linux.

As an example, the `devtoolset-7` collection is a developer toolset available within the Software Collection Library for Oracle Linux.

Using Software Collections with Oracle Linux 7

- Ensure you are subscribed to the following ULN channels:
  - `ol7_x86_64_latest`, `ol7_x86_64_optional_latest` and `ol7_x86_64_SoftwareCollections`
- Or ensure the following Oracle Linux yum server repositories are enabled:
  - `ol7_latest`, `ol7_optional_latest` and `ol7_software_collections`
- To view available packages within the ULN software collections channel, run:

```
# yum --disablerepo="*" --enablerepo="*SoftwareCollections" list available
```

- To view available packages within the yum software collections repository, run:

```
# yum --disablerepo="*" --enablerepo="*software_collections" list available
```

- To see software collections packages that are already installed from the yum software collections repository, run:

```
# yum list installed|grep software_collections
```

ORACLE

To use collections and packages from the Software Collection Library with Oracle Linux 7, three ULN channels or Oracle Linux yum server repositories are needed, as shown in the slide.

Packages available (not yet installed) within the ULN `ol7_x86_64_SoftwareCollections` channel or yum `ol7_software_collections` repository can be viewed using the `yum` commands shown in the slide. In these cases, all repos are disabled for the duration of the `yum` command, and only the software collections channel or repository is enabled.

Installed software collections packages can be seen using the `yum` command combined with `grep`. The example in the slide will show packages installed from the yum `ol7_software_collections` repository, if any. To list packages installed from the ULN repository, `ol7_x86_64_SoftwareCollections`, use this form of the command:

```
# yum list installed|grep SoftwareCollections
```

A complete list of collections can be found in the Release Notes for the Software Collection Library for Oracle Linux, available here: https://docs.oracle.com/cd/E52668_01/index.html.

Installing Software Collections and Running Commands

- The `scl-utils` package must be installed in order to run software from a collection, if not installed already:

```
# yum install scl-utils
```

  - The `scl` utility from the `scl-utils` package is used to execute commands or utilities from a software collection with the `enable` action.
- Use `yum` to install a collection. For example, to install Developer Toolset version 7:

```
# yum install devtoolset-7
```

- Software collection packages are installed in the `/opt` path.
- `strace` is one of the utilities in the `devtoolset-7` collection. Use the `scl` utility to run the `devtoolset-7` software collection version of `strace`. Example:

```
# scl enable devtoolset-7 -- strace -e trace=/stat$ ls /boot/vmlinuz*
```

To view man pages for software collections, use the `scl` utility as in these examples:

```
# scl enable devtoolset-7 'man devtoolset-7'
# scl enable devtoolset-7 'man strace'
```

ORACLE

The `scl-utils` package must be installed in order to run software from a collection. This package provides the `scl` utility, which enables execution of a command from a software collection rather than using the version of a command available from the installation of Oracle Linux. The `enable` action enables use of one or more software collections, followed by a command or utility to execute. If the command or utility has one or more arguments, it must be preceded by the "`--`" command separator or enclosed in quotation marks. See the `scl(1)` man page for more information about this utility.

To use commands within a software collection, install the software collection first. For example, to install the `devtoolset-7` collection, run:

```
# yum install devtoolset-7
```

Software collection packages are installed in the `/opt` path. Then, if it is desired to utilize `strace` from the `devtoolset-7` collection, the following is an example:

```
# scl enable devtoolset-7 -- strace -e trace=/stat$ ls /boot/vmlinuz*
```

`strace` provides information about system calls as well as other information, and can be useful with the debugging process. In the example above, `strace` from the `devtoolset-7` collection is called to trace the execution of the `ls` command, which will list the `vmlinuz*` files in the `/boot` directory. A regular expression (`stat$`) is used to select system calls ending with the string "stat". This is a capability of the `strace` version included with the `devtoolset-7` collection that is unavailable in the `strace` version included with the installation of Oracle Linux 7 Update 5.

`man` pages specific to software collections can be viewed using the `scl` utility as shown in the slide. In these examples, the `devtoolset-7` collection has a `man` page that can be viewed, in addition to the specific `strace` utility contained within that collection. `man` pages are included with the various software collections, so a given collection must be installed in order to view its related `man` pages.

Creating a Shell Environment and Viewing Optional Packages

- Commands or utilities associated with one or more collections can be made available in a shell environment without running the `scl` utility within the shell. Example:

```
# scl enable devtoolset-7 bash
```

  – The path to the software collection version of `strace` is known within the shell:

```
# which strace
/opt/rh/devtoolset-7/root/usr/bin/strace
```

- To enable two or more software collections within the same shell environment, list them on the command line. Example:

```
# scl enable devtoolset-7 python33 bash
```

- Collections can have optional packages that are not installed with the main package and its dependencies.
  – Use the `yum` command to see available optional packages. Example:

```
# yum list available devtoolset-7-\*
```

ORACLE

A shell environment can be generated in which commands or utilities from one or more collections can be run without using the `scl` utility. The commands or utilities from software collections enabled within such a shell environment are executed instead of those installed with Oracle Linux. A shell environment is created by running the `scl` utility specifying the collections to be enabled within the shell, followed by "bash." For example, the commands and utilities within the `devtoolset-7` collection can be run within their own shell as follows:

```
# scl enable devtoolset-7 bash
```

The version of `strace` that will be run is found in the `/opt` path, rather than in its installed Oracle Linux directories, `/bin` or `/usr/bin`. `strace` can then be run within the shell environment using a regular expression for system calls without using `scl`:

```
# strace -e trace=/stat$ ls /boot/vmlinuz*
```

To enable multiple software collections within a shell environment, enter the collection names on the command line delimited by spaces. For example:

```
# scl enable devtoolset-7 python33 bash
```

This will allow commands and utilities from the `devtoolset-7` and `python33` collections to be run within the same shell environment.

When installing a software collection, the main package and dependencies are installed. There can be additional optional packages not installed, that may be desired. These can be viewed for any given software collection as shown in the slide.

Quiz

Yum repositories can be both local and remote.

a. True
b. False

Answer: a

Quiz

Errata are available only by subscribing to ULN.

a. True
b. False

Answer: b

Quiz

Which of the following statements about the Software Collection Library for Oracle Linux are true? (Select all that apply.)

a. The collections are primarily intended for developers.

b. The collections provide later versions of software than are available from an Oracle Linux installation.

c. Packages from the collections are installed in the same directory path as packages from an Oracle Linux installation.

d. Packages are available from the Unbreakable Linux Network (ULN) and the Oracle Linux yum server.

Answer: a, b, d

Answer c is false. Packages are installed in the `/opt` path.

Summary

In this lesson, you should have learned how to:

- Describe Oracle Linux package management
- Use the `rpm` utility
- Describe the Oracle Linux yum server
- Explain and configure yum repositories
- Use the `yum` utility
- Describe the Unbreakable Linux Network (ULN)
- Describe how to register a system with ULN
- List the steps to switch from Red Hat Network (RHN) to ULN
- Explain replacement of SSL certificates for ULN
- Describe Software Collections and how to use them

Practice 7: Overview

This practice covers the following topics:

- Using the `rpm` utility
- Accessing the Oracle Linux yum server
- Creating a local `yum` repository
- Using the `yum` utility
- Using Oracle Linux Software Collections
- Using the Unbreakable Linux Network (ULN)

8

Automating Tasks

Objectives

After completing this lesson, you should be able to:

- Describe the available automated task 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 task utilities such as `cron`, `anacron`, `at`, and `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 the system is up.
- `crond` searches multiple files and directories for scheduled jobs:
  - `/var/spool/cron/`
  - `/etc/anacrontab`
  - `/etc/cron.d`

Oracle Linux can run programs automatically. Programs are also referred to as tasks or jobs. Utilities to automate these tasks or jobs include `cron`, `anacron`, `at`, and `batch`. This lesson describes each of these utilities.

`cron` and `anacron`

Use the `cron` and `anacron` utilities to schedule jobs to run on certain days or times. You can use `cron` to run jobs as often as every minute. A `cron` job will not run if the system is down during the time that the job was scheduled to run.

Use the `anacron` utility to run a job once a day. If the system is down when the `anacron` job is scheduled to run, the `anacron` job will run when the system comes back up. Therefore, you can use `anacron` to run `cron` jobs that failed to run because the system was down.

The `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 the name of the month
  - `day-of-week`: From 0 to 7, or the abbreviated name of day
  - `command`: Program 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 system-wide `crontab`. Users can also have `cron` jobs. `cron` jobs are specified in the following format:

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

- **`minute`:** From 0 to 59
- **`hour`:** From 0 to 23
- **`day`:** From 1 to 31 (must be a valid day for the specified month)
- **`month`:** From 1 to 12 (you can also use Jan, Feb, Mar, and so on)
- **`day-of-week`:** Either 0 or Sun, 1 or Mon, 2 or Tue, 3 or Wed, 4 or Thu, 5 or Fri, 6 or Sat, 7 or Sun
- **`command`:** The program or script to execute

Other special characters can be used:

- An asterisk (`*`) represents all valid values.
- A hyphen (`-`) between integers specifies a range of integers.
- A comma (`,`) between a list of values.
- A forward slash (`/`) specifies step values. Examples are given on the next page.

`cron` Examples

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

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

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

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

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

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

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

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

Other `cron` Directories and Files

- `/etc/cron.d`
  - Contains files with the same syntax as `/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`.

These files are checked each time a user tries to add or delete a `cron` job.

Specify users, each user on a separate line, that can use `cron` in the `cron.allow` file.

Specify users, each user on a separate line, that cannot use `cron` in the `cron.deny` file.

The `cron.deny` file is only checked if the `cron.allow` file does not exist.

Only the `root` user can use `cron` if neither file exists.

The `crontab` Utility

- The `crontab` utility allows users other than `root` to configure `cron` tasks.
- User-defined `crontabs` are stored in:
  - `/var/spool/cron/<user>`
- To create or edit a `crontab`:
  - Enter the following command:

```
$ crontab -e
```

  - Use the same format as `/etc/crontab` without specifying a user.
- To list the contents of a user-defined `crontab`, enter:

```
$ crontab -l
```

- The `root` user can use the `-u <username>` option to create, edit, remove, or list the `crontab` for a specific user. For example:

```
# crontab -u oracle -l
```

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
```

The `root` user can use the `-u <username>` option to create, edit, remove, or list the `crontab` for a specific user. Be aware that the `-r` option removes a `crontab` with no prompting. For example:

```
# crontab -u oracle -e
# crontab -u oracle -l
# crontab -u oracle -r
```

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

`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=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     dailyjob          nice run-parts /etc/cron.daily
7        0     weeklyjob         nice run-parts /etc/cron.daily
@monthly 0     monthlyjob        nice run-parts /etc/cron.daily
```

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 hour range that scheduled jobs can start

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 to 45 minutes and can be executed between 03:00 and 22:00.

The first job runs daily and executes all programs in the `/etc/cron.daily` directory. Delay is 5 minutes + RANDOM_DELAY for `cron.daily`.

The second job runs weekly and executes all programs in the `/etc/cron.weekly` directory.

The third job runs monthly and executes all programs in the `/etc/cron.monthly` directory.

Checking `cron` Job Status

- `cron` messages logs are specified in `/etc/rsyslog.conf`.
- Sample `/etc/rsyslog.conf` entry that logs all `cron` messages to `/var/log/cron`:

```
# Log cron stuff
cron.*                  /var/log/cron
```

- Information about `cron` and `anacron` jobs are stored here.
- Sample log entries:

```
Aug  7 09:01:01 host03 CROND[5607]: (root) CMD (run-parts
/etc/cron.hourly)
Aug  7 09:01:01 host03 run-parts(/etc/cron.hourly)[5607]:
starting 0anacron
Aug  7 09:01:01 host03 run-parts(/etc/cron.hourly)[5618]:
finished 0anacron
```


`/etc/rsyslog.conf` is described in detail in the lesson titled “System Logging.”

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 the 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)

Use the at command to execute a one-time task at a specific time. Use the batch command to execute a one-time task when system load levels permit, that is, when the 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> <Ctrl+d>
Job 1 at <date_time>
# atq
1           <date_time> 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`.

Usage of these files is similar to the usage of the `cron.allow` and `cron.deny` files.

These files are checked each time a user tries to use either the `at` or `batch` command.

Specify users, each user on a separate line, that can use the commands in the `at.allow` file.

Specify users, each user on a separate line, that cannot use the commands in the `at.deny` file.

The `at.deny` file is only checked if the `at.allow` file does not exist.

The `root` user can always execute `at` and `batch` commands.

Quiz

Both `at` and `batch` are used to schedule the execution of recurring tasks.

a. True
b. False

Answer: b

– These commands are for scheduling a one-time task.

Summary

In this lesson, you should have learned how to:

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

Practice 8: Overview

This practice covers 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>
```

- In UEK R4, kernel module file names have a "`.ko`" extension, for "kernel object".
  – For example, the `ext4` kernel module name in UEK R4 is `ext4.ko`.
- In UEK R5, kernel modules are compressed, and have "`.xz`" added to "`.ko`".
  – The `ext4` kernel module name in UEK R5 is `ext4.ko.xz`.
- Kernel modules often have dependencies.

ORACLE

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
...
bridge                    114688   1 ebtable_broute
stp                        16384   1 bridge
llc                        16384   2 stp,bridge
...
```

The output lists the name of the kernel module and the amount of memory the module uses. The "Used by" column gives the total of the number of processes that are using the module in addition to any modules that depend on the module, with the names of modules that depend on the given module shown last. For example, the `stp` and `bridge` modules depend on the `llc` module. 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-20-21.0.1.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 `ext4` kernel module, enter:

```
# modinfo ext4
filename:       /lib/modules/4.1.12-
112.16.4.el7uek.x86_64/kernel/fs/ext4/ext4.ko
license:        GPL
description:    Fourth Extended Filesystem
author:         Remy Card, Stephen Tweedie, Andrew Morton, ...
alias:          fs-ext4
srcversion:     6AD825CC26E69409C80CA01
depends:        jbd2,mbcache2
...
parm:           num_prealloc_crypto_pages:Number of crypto pages
to preallocate (uint)
parm:           num_prealloc_crypto_ctxs:Number of crypto contexts
to preallocate (uint)
```

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/4.1.112.16.4.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                   286720  0
lockd                  94208  1 nfs
fscache                65536  1 nfs
sunrpc                344064  3 nfs,lockd
```

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/4.1.12-
112.16.4.el7uek.x86_64/kernel/fs/fscache/fscache.ko
insmod /lib/modules/4.1.12-
112.16.4.el7uek.x86_64/kernel/fs/nfs_common/grace.ko
insmod /lib/modules/4.1.12-
112.16.4.el7uek.x86_64/kernel/fs/lockd/lockd.ko
insmod /lib/modules/4.1.12-
112.16.4.el7uek.x86_64/kernel/fs/nfs/nfs.ko
```

Note that `modprobe` uses the `insmod` command to load the modules into the kernel. However, do not use `insmod`, 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
rmmod fscache
rmmod lockd
rmmod grace
```

Modules are unloaded in the reverse order, with the `nfs` 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 ASM Cluster File System (ACFS) and ASM Dynamic Volume Manager (ADVM) drivers and utilities. Automatic Storage Management (ASM) is a feature within 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 Oracle High Availability Service Daemon (OHASD) 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:
  - `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:

```
alias, options, install, remove, blacklist
```

Alias

Use the `alias alias_name module_name` syntax to create alternative names for kernel modules. You can also use shell wildcards in alias names. Example:

```
alias usbdevfs usbcore
```

Options

Use the `options module_name option(s)` syntax to add options to `module_name`. Example:

```
options b43 nohwcrypt=1 qos=0
```

Install

Use the `install module_name command(s)` syntax 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 `remove module_name command(s)` syntax to tell `modprobe -r` to run shell commands rather than unload the module from the kernel.

Blacklist

Use the `blacklist module_name` syntax 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, 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>`

Answer: b

The `modinfo` command displays detailed information about a kernel module. The `lsmod` command shows what kernel modules are currently loaded. The `modprobe` command adds and removes modules from the Linux kernel. The `depmod` command generates the `modules.dep` file.

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

This practice covers using loadable kernel modules.

10

Oracle Ksplice

Objectives

After completing this lesson, you should be able to:

- Describe the purpose of Oracle Ksplice
- Describe the features, benefits, and operation of Oracle Ksplice
- Implement the Ksplice Online Standard Client
- View Ksplice packages on ULN
- Describe Ksplice implementation in Oracle Cloud Infrastructure
- Use Ksplice commands
- Use the Ksplice web interface
- Use the Ksplice Offline Standard Client
- Implement the Ksplice Online Enhanced Client
- Configure Ksplice Offline Clients to use a local Ksplice mirror

Oracle Ksplice: Introduction

Ksplice:

- Updates the kernel and select user space libraries on a running system
- Applies the latest security errata (CVEs) and critical bug fixes
  - Patches are effective immediately without rebooting
- Does not halt the system
- Does not restart applications
- Applies updates in the background
- Requires an Oracle Premier support subscription
- Works with both the Unbreakable Enterprise Kernel and Red Hat Compatible Kernel
- Has an easy-to-use website

Customers running Oracle Linux 6 or later with a Premier support subscription have access to Oracle Ksplice technology. Oracle Linux 5 requires Extended Support for Ksplice access.

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 and select user space library patches that can be applied on a running system. Ksplice patches are applied to the running Linux kernel and/or running user space processes, and are effective immediately. Oracle Ksplice kernel patches are only applied to the running kernel and on subsequent reboot, kernel patches are applied at boot time. In addition to updating running user space processes, user space libraries are updated on disk, which makes them available for subsequent processes as needed. User space updates are available for Oracle Linux 6 or later.

You can apply the latest security errata (CVEs: Common Vulnerabilities and Exposures) and crucial bug fixes without rebooting. No halt to the system or restart of applications is required. Updates are applied in the background with a negligible impact.

The Ksplice Uptrack website has an easy-to-use interface that lets you view the status of registered systems for kernel updates. For each system, the website shows available kernel updates that have not yet been applied as well as installed updates for your running kernel. Each update has a one-line description. You can also set policies to allow or deny kernel updates. User space updates are not shown on this website.

Ksplice Addresses System Administrator Challenges

- Security updates and crucial bug fixes need to be applied.
  - Results in down time
  - Maintenance windows must be coordinated and scheduled.
  - Time and disruption is costly.
- Administrators must balance keeping systems up-to-date and secure versus down time.
- Ksplice mitigates these challenges.
  - Updates are applied with no down time.
  - Updates are fast.
  - Systems are secured and bug fixes are applied quickly.
  - System availability is improved.
  - Costs are lowered.

System administrators must balance the need for secure and up-to-date systems with the down time that can be required to apply updates.

With Ksplice, updates are applied quickly with no down time. This makes systems more available and lowers the costs involved in managing down time with system restarts.

Ksplice: Features and Benefits

Features	Benefits
• Update/patch Oracle Linux systems with no down time	• Systems remain available and are secured in real time • Reduces cost/disruption
• Roll back updates	• Updates can be removed without reboot • Can integrate with change control
• Web and API interfaces	• Centralized management
• No agents or daemons	• No impact on performance
• Oracle Linux on bare metal, Oracle VM, or Oracle Cloud supported	• Multiple platforms
• Patching supported with applications, middleware, and database installed	• No impact on running Oracle software

Ksplice Patching

- Oracle Linux
  - The Unbreakable Enterprise Kernel (UEK)
  - Red Hat Compatible Kernel (RHCK)
- Oracle VM Server (dom0)
  - VMs continue running
  - No migrations needed
- User space RPMs
  - glibc
  - OpenSSL
  - Must install Ksplice-aware versions from ULN with one-time reboot
- Scalability
  - Up to 5,000 kernel versions at a time, for widespread issues
  - Complex patches are handled

For vulnerabilities or bugs that affect many different kernel versions, ksplice patches can be created for up to 5,000 different kernels.

Ksplice for Diagnosing Issues

- Work with Oracle support.
  - Create a Service Request.
- If the issue has no known resolution, support can work with development for a diagnostic Ksplice patch.
  - Collect further data/information for diagnosis.
- After the resolution is identified, the Ksplice diagnostic patch is removed.
- A final Ksplice patch is developed for issue resolution.
  - Errata is provided to Unbreakable Linux Network (ULN).
  - The final Ksplice patch can be applied using uptrack.
  - It is available to Oracle Linux Premier Support customers.
- Everything is done with no rebooting and no down time.

Ksplice Track Record

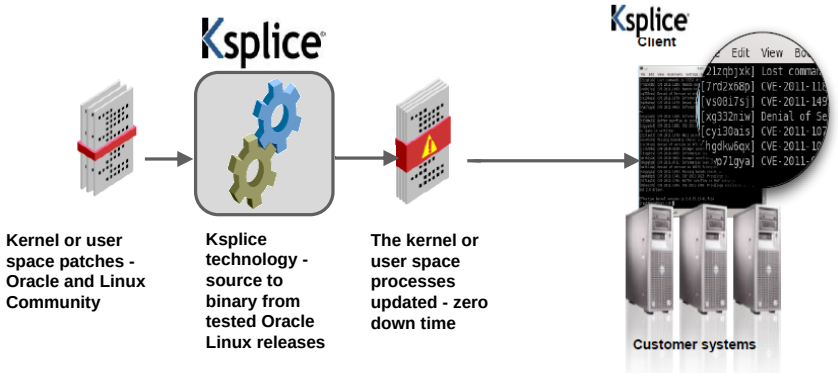
- Over 1.5 million updates shipped
- Example of a kernel vulnerability addressed:
  - CVE-2016-5696 Challenge-ACK TCP hijacking
    - Ksplice patch available five days before releases from other vendors.
- Examples of user space vulnerabilities remedied:
  - CVE-2015-7547 Glibc vulnerabilities
    - Ksplice was the only option for patching without a restart.
  - CVE-2016-0800 (DROWN) OpenSSL vulnerability
    - Ksplice again was the only solution supporting patching without reboot.

Ksplice Case Study

- Oracle Compute
  - 1,000 Oracle Linux systems running EDA simulations
  - Rolling upgrade 10% at a time
  - Each upgrade took ~54 hours of sysadmin time, 7 hours of user time lost
  - 28 days for full upgrade
- With Ksplice, time reduced to ~1 hour

EDA = Electronic Design Automation

How Ksplice Works



- Patches go into new kernel or user space releases, are tested by Oracle.
  - They are then turned into Ksplice updates from Oracle Linux packages.
- `glibc` and `openssl` user space libraries are updated on disk as well.
  - Updates are available as shared object files for applications to load as needed.

Kernel or user space updates come from Oracle or the Linux community. These go into Oracle Linux releases, are tested by Oracle, and then are turned into Ksplice binary updates from Oracle Linux packages.

Kernel updates are inserted into a running kernel. User space updates are developed to patch running processes for glibc and openssl. glibc and openssl libraries are also updated on disk to be available permanently. You apply patches using Ksplice tools, and the patches are 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.

Online Ksplice Implementations

- For client systems with internet access
  - Installation packages are obtained from ULN.
  - Updates are obtained from the Oracle Uptrack server.
- Standard Client: Provides kernel updates
  - uptrack package installed
  - Uses Ksplice Uptrack commands and updates the kernel
- Enhanced Client: Provides kernel and user space updates
  - ksplice package installed, with dependencies (includes uptrack)
  - Ksplice-aware user space libraries, glibc and openssl, installed with one-time reboot
  - Uses Ksplice commands
  - Updates the kernel along with glibc and openssl shared libraries

Online implementations access ULN for installation packages and the Oracle Uptrack server for kernel and user space updates.

These cron jobs are installed when the online clients are installed:

- `/etc/cron.d/uptrack`

  Checks for available kernel updates
- `/etc/cron.d/ksplice`

  Checks for available user space updates

If `autoinstall = yes` in `/etc/uptrack/uptrack.conf`, available updates are applied automatically when these cron jobs run. If `autoinstall = no` (the default), available updates are not applied when these cron jobs run, and ksplice commands must be executed separately to apply updates.

Offline Ksplice Implementations

- For client systems with no internet access
  - For example: Security purposes
  - Updates are packaged in RPMs.
  - Use a local Ksplice mirror or Oracle Cloud Infrastructure mirror for offline clients to obtain installation packages and updates.
- Standard Client: Provides kernel updates
  - uptrack-offline package installed
  - Uses Ksplice Uptrack commands and updates the kernel
- Enhanced Client: Provides kernel and user space updates
  - ksplice-offline package installed, with dependencies (includes uptrack-offline)
  - Ksplice-aware user space libraries, glibc and openssl, installed with one-time reboot
  - Uses Ksplice commands
  - Updates the kernel along with glibc and openssl shared libraries

The offline Ksplice implementations are intended for client systems without open access to the Internet. One example would be for security reasons. The offline versions of the Ksplice packages are installed, which are: `ksplice-offline-<version>` and `uptrack-offline-<version>`. Updates are packaged in RPMs for patching the kernel and `glibc/openssl` user space libraries.

For the Standard Client, kernel patch RPMs use the naming convention: `uptrack-updates-<kernel version>` and are found in the channel: "`Ksplice for Oracle Linux (x86_64)`". For the Enhanced Client, user space patch RPMs use the naming convention: `ksplice-updates-glibc-<version>` and `ksplice-updates-openssl-<version>`, and are found in the channel: "`Ksplice aware userspace packages for Oracle Linux 7 (x86_64)`".

These cron jobs are installed when the offline clients are installed:

- `/etc/cron.d/uptrack`

  Checks for available kernel updates. Additional kernel updates might be applied if the `uptrack-updates-<kernel version>` RPM installation was unable to apply all available updates at the time, possibly due to a busy system.

- `/etc/cron.d/ksplice`

  Checks for available user space updates. Installing the user space packages `ksplice-updates-glibc-<version>` and `ksplice-updates-openssl-<version>` does not apply updates. This cron job can apply available updates to running processes.

If `autoinstall = yes` in `/etc/uptrack/uptrack.conf`, available updates are applied automatically when these cron jobs run. If `autoinstall = no` (the default), available updates are not applied when these cron jobs run, and ksplice commands must be executed separately to apply updates.

Implementing the Ksplice Online Standard Client

A summary of actions to get started with Ksplice:

1. Register your systems 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 the 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:

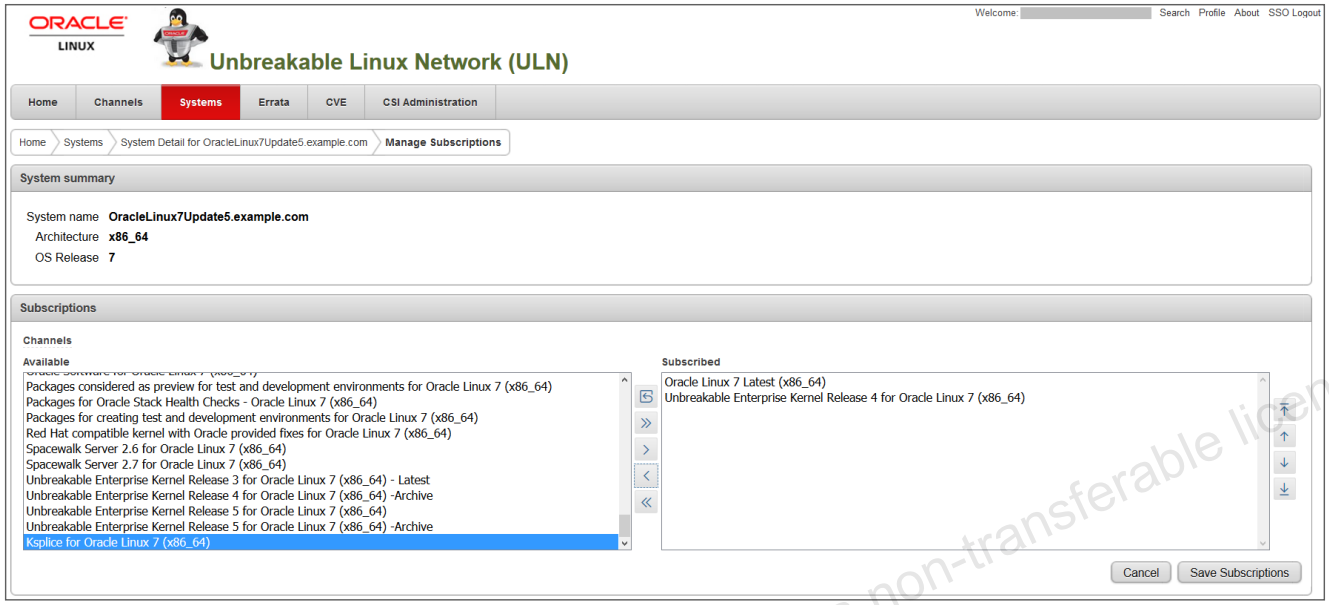
1. Register your system(s) with ULN.
2. Subscribe to the appropriate Ksplice channel.

ULN channels that are available for Ksplice kernel updates on Oracle Linux:

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

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

Subscribing to the Ksplice Channel



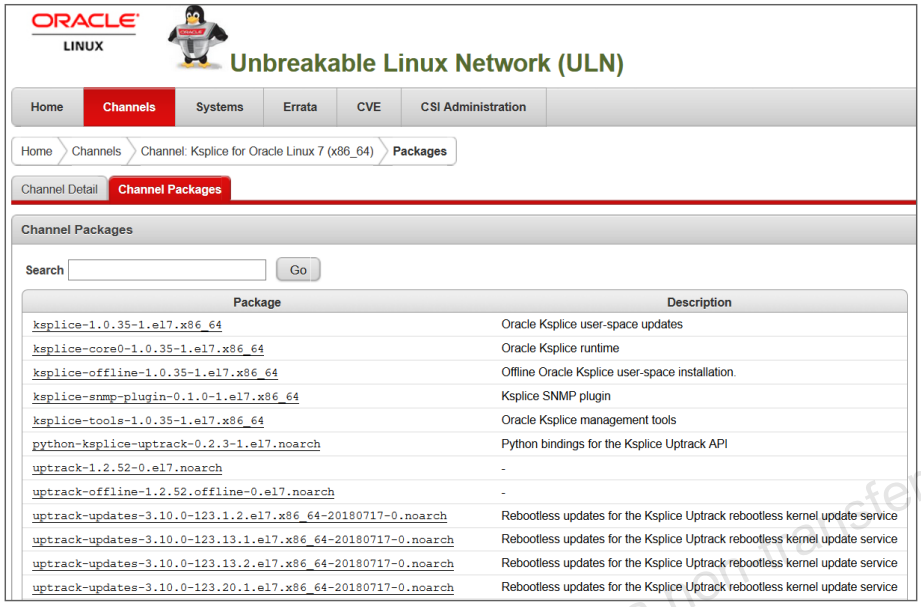
You have the option of subscribing to the Ksplice channel when you register your system with ULN. Otherwise, you can log in to the ULN web interface and subscribe to the Ksplice channel for your system's version of Oracle Linux.

Select the system name and click Manage Subscriptions. Scroll down the list of Available channels and select the appropriate Ksplice channel for your system.

The slide shows that the "Ksplice for Oracle Linux 7 (x86_64)" ULN channel is selected for the system named `OracleLinux7Update5.example.com`. This corresponds to `ol7_x86_64_ksplice`.

With the Ksplice channel selected, click the > arrow to move the selected entry to the Subscribed Channels pane. Click the Save Subscriptions button to continue.

Ksplice Packages on ULN



This slide shows a sample of the 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.

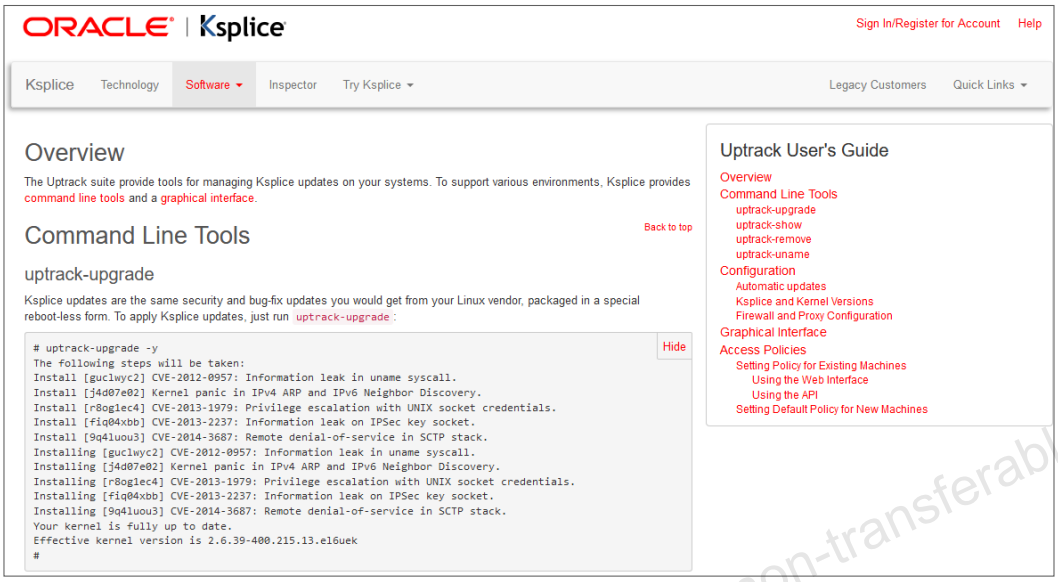
Your server must have access to the Internet. If a proxy is used, set the proxy in your shell before running the Uptrack commands. The commands to set the proxy are:

```
# 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 Command Line Tools page at https://ksplice.oracle.com/uptrack/guide. It lists Ksplice Uptrack commands and provides examples and sample output. It describes how to configure your systems for automatic updates. It also describes access policies and provides other information about Ksplice.

There is also a Ksplice user's guide available at https://docs.oracle.com/cd/E37670_01/E39380/html/index.html.

Ksplice Uptrack Command Summary

- `uptrack-upgrade`: Apply available updates.
- `uptrack-install <Ksplice_ID>`: Install a specific update by specifying the Ksplice ID.
- `uptrack-show`: List the active updates in your running kernel.
- `uptrack-show --available`: List the available updates to be installed.
- `uptrack-remove`: Remove applied updates from the running system and return to the original kernel version.
- `uptrack-remove <Ksplice_ID>`: Remove a specific update by specifying the Ksplice ID.
- `uptrack-uname`: Display the effective kernel version based on active Oracle Ksplice updates.

The following provides a summary of the Ksplice Uptrack Commands:

- **`uptrack-upgrade`**: Connect to the Uptrack update server and apply available updates.
- **`uptrack-install <Ksplice_ID>`**: Apply a specific update by specifying the Ksplice ID. This will first install any dependencies, then the specified update ID.

It is recommended to apply all updates regularly, rather than specifying a specific Ksplice ID with the **`uptrack-install`** command.

- **`uptrack-show`**: List the active Oracle Ksplice updates in your running kernel.
- **`uptrack-show --available`**: List the available updates to be installed.
- **`uptrack-remove`**: Remove applied updates from the running system and return to the original kernel version and state.
- **`uptrack-remove <Ksplice_ID>`**: Remove a specific update by specifying the Ksplice ID. This will remove any later updates, then the update ID specified.
- **`uptrack-uname`**: This is the modified version of `uname` that knows how to read the effective kernel version based on active Oracle Ksplice updates.
- To remove the `uptrack` package, enter:

```
# yum remove uptrack
```

You can also find the uninstall instructions on the Uptrack website.

Ksplice in Oracle Cloud Infrastructure

- Ksplice is available with Oracle Cloud Infrastructure instances.
- Ksplice for kernel patches (`uptrack`) is automatically installed on instances launched from Oracle Linux images on or after August 25, 2017.
  - You can run `uptrack-upgrade` immediately to apply available updates, with no further configuration.
  - Set `autoinstall = yes` in `/etc/uptrack/uptrack.conf` to apply updates automatically.
- For instances launched from custom images or for instances launched before August 25, 2017, the `uptrack` package must be installed.
  - If you register with ULN, the `uptrack` package can be installed from ULN with yum using the `ol7_x86_64_ksplice` channel.
- Ksplice for online user space library updates requires registration with ULN or use of a ULN mirror, with installation/setup necessary.
  - Ksplice for offline user space updates uses an Oracle Cloud Infrastructure mirror, with no ULN registration or access.

Users who run Oracle Linux in Oracle Cloud Infrastructure have full access to Ksplice as part of being Oracle Cloud Infrastructure customers, with no extra charge. Ksplice is used directly within instances.

Ksplice for kernel patches (`uptrack`) is automatically installed on instances launched from Oracle Linux images in Oracle Cloud Infrastructure on or after August 25, 2017. The kernel can be updated using the `uptrack-upgrade` command immediately. Instances launched from custom images or before this date require installation of the `uptrack` package. If installation of `uptrack` for kernel updates is needed, registration with ULN allows installation from ULN with yum.

Ksplice for online user space library updates requires ULN registration or use of a ULN mirror, installation of the appropriate Ksplice package, and steps to be followed in the slide titled "Implementing the Ksplice Online Enhanced Client."

Ksplice for offline user space updates does not access ULN, but uses an Oracle Cloud Infrastructure mirror.

Installing `uptrack` for Ksplice Without ULN Registration

- If uptrack must be installed in Oracle Cloud Infrastructure and you do not register with ULN, follow these steps:
  - Set up a proxy if needed.
  - Get the `install-uptrack-oc` script:

```
# wget -N https://www.ksplice.com/uptrack/install-uptrack-oc
```

  - Run the `install-uptrack-oc` script. The `--autoinstall` option causes updates to be applied automatically:

```
# sh install-uptrack-oc --autoinstall
```

  - Or, the following requires manual updates:

```
# sh install-uptrack-oc
```


If you do not register with ULN and uptrack must be installed, the `install-uptrack-oc` script must be downloaded and run. To do this, run the following command as `root` on the instance:

```
# wget -N https://www.ksplice.com/uptrack/install-uptrack-oc
...
Saving to: 'install-uptrack-oc'

100%[======================================>] 10,154      --.-K/s   in
0.05s

<date_time> (218 KB/s) - 'install-uptrack-oc' saved [10154/10154]
```

After downloading the `install-uptrack-oc` script for Oracle Cloud Infrastructure, the following installs `uptrack`, enabling automatic kernel updates with the `--autoinstall` option:

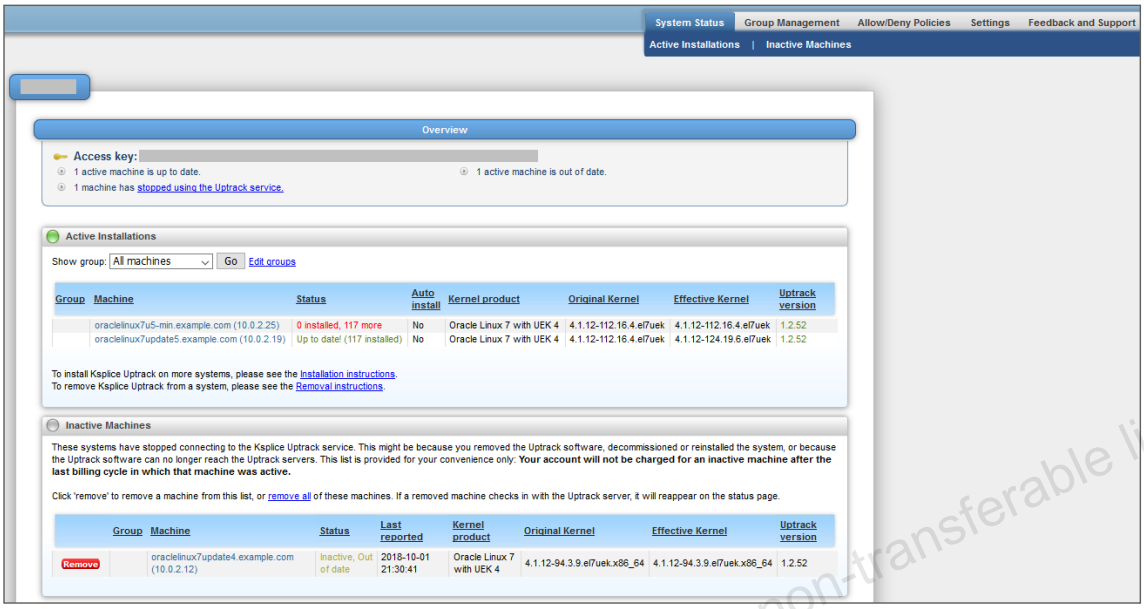
```
# sh install-uptrack-oc --autoinstall
[ Release detected: ol ]
...
Saving to: ‘ksplice-uptrack-release.noarch.rpm’
...
<date_time> (82.6 MB/s) - ‘ksplice-uptrack-release.noarch.rpm’ saved
[6876/6876]
[ Installing Uptrack ]
...
Preparing packages...
ksplice-uptrack-release-1-3.noarch
Loaded plugins: langpacks, ulninfo
...
Resolving Dependencies
--> Running transaction check
...
--> Finished Dependency Resolution
Dependencies Resolved
================================================================================
 Package       Arch      Version        Repository             Size
================================================================================
Installing:
 uptrack      noarch     1.2.41-0.el7   ksplice-uptrack       298 k
Installing for dependencies:
 perl-autodie noarch     2.16-2.el7     ol7_latest             77 k

Transaction Summary
================================================================================
...
 Installed:
  uptrack.noarch 0:1.2.41-0.el7
 Dependency Installed:
  perl-autodie.noarch 0:2.16-2.el7
 Complete!
```

After `uptrack` is installed, available kernel updates are listed:

```
Effective kernel version is 4.1.12-94.2.1.el7uek
The following steps will be taken:
Install [czul7xgo] Improve the interface to freeze tasks.
Install [aqmwi3fw] CVE-2017-7895: Remote information leak in
kernel NFS server.
Install [5ttn4813] NULL pointer dereference in iSCSI target
communication.
Install [ghun2xxv] CVE-2016-10229: Remote code execution when
receiving UDP packet with short buffers.
Install [tazqd9ku] Denial-of-service in BTRFS reflinked file
removal.
Install [f6fr74n6] NULL pointer dereference in Hyper-V key/value
store.
Install [7p6hizfc] NULL pointer dereference in IPv6 DTRACE probe.
Install [t6v4yb39] CVE-2017-8890: Denial-of-service in TCP and
DCCP socket manipulation.
Install [nqey27lx] CVE-2017-7308: Memory corruption in AF_PACKET
socket options.
Install [8cpcuyra] CVE-2017-1000364: Increase stack guard size to
1 MiB.
Install [dnixen2b] CVE-2017-7645: Remote denial-of-service via
overly sized NFS2/3 RPC call.
Install [ta5frh0y] CVE-2017-7477: Remote Denial-of-service in
802.1AE implementation.
Install [sfmj44ki] Denial-of-service when bonding multiple IPOIB
devices.
[ Installation Complete! ]
[ Please run '/usr/sbin/uptrack-upgrade -y' to bring your system
up to date ]
```

As indicated, run `uptrack-upgrade -y` to install the updates without confirmation.

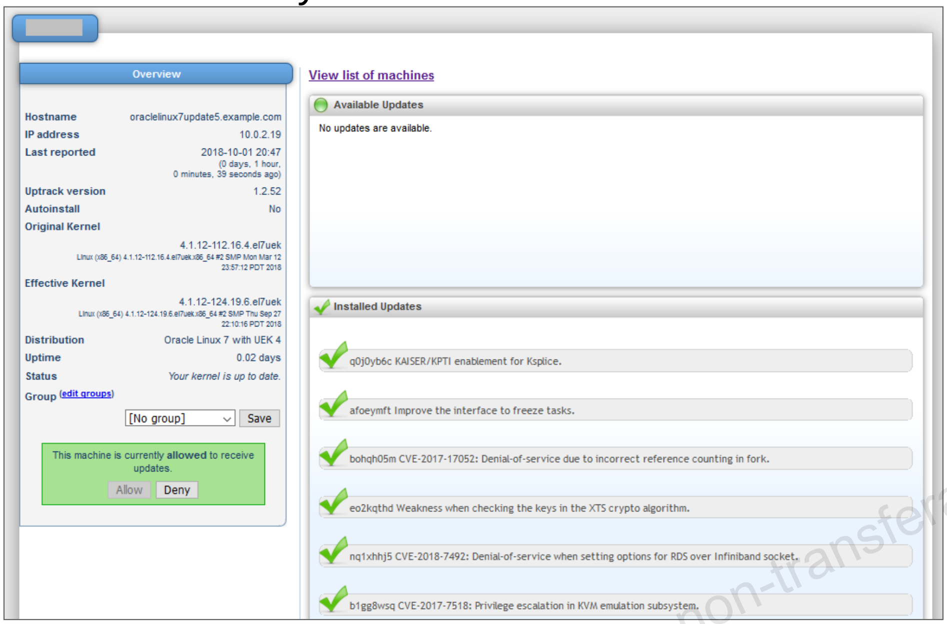
Ksplice Web Interface: System Status



This screen shows the System Status page of the Ksplice web interface at https://status-ksplice.oracle.com. Online Ksplice implementations for kernel updates are listed on this page. Offline Ksplice implementations and user space update status are not reflected here. Systems listed under Active Installations are connected to the Ksplice servers. Inactive Machines are listed at the bottom of the page and are those that have stopped using Ksplice Uptrack services.

- The **Machine** column lists the individual systems. Click a machine name to display detailed information about available updates, installed updates, and additional information.
- The **Status** in the example in the slide shows that there are updates available for one of the active systems. 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 shown. The Original and Effective Kernel become the same if you install the new kernel on disk using `yum` and reboot. If you reboot without updating with `yum`, you boot into the Original Kernel, and Ksplice applies updates automatically at boot time. This returns the system to the Original and Effective Kernel versions in effect before rebooting.
- The **Uptrack version** column shows the Ksplice Uptrack version.

Ksplice Web Interface: System Status Detail



This screen shows the System Status details page of the Ksplice web interface for one system. Access this screen by clicking the Machine link on the main System Status page.

This screen shows system information in the Overview section. Available and Installed Updates are shown in separate sections.

There are no available updates for the selected system. Because updates have been previously applied, the Effective Kernel is 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 Ksplice Offline Standard Client

- Systems running the Ksplice Offline Standard 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.
  - A Local Yum Server requires a network connection to ULN.
  - Ksplice Offline Standard Clients require access only to the Local Yum Server.
- In Oracle Cloud Infrastructure, an Oracle Cloud Infrastructure mirror is used for offline updates.

The Ksplice Offline Standard Client removes the requirement for a server to have a 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. For example, you can download the RPM from ULN to a memory stick, 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 ULN. After installing the Ksplice Offline Standard Client on your other systems, these systems need only to be able to connect to the Local Yum Server.

In Oracle Cloud Infrastructure, an Oracle Cloud Infrastructure mirror is used for offline updates instead of creating a Local Yum Server that acts as a Ksplice mirror. The Oracle Cloud Infrastructure mirror is already set up for you.

Systems that are running the Ksplice Offline Standard 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

- A 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 the needed channels.
  7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.

ORACLE

If you have a system that is registered with ULN and 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. Add Ksplice channels you want to make available to local Ksplice offline clients.
7. When you have finished selecting channels, click Save Subscriptions and log out of ULN.

For details about creating a local ULN mirror, see https://docs.oracle.com/cd/E37670_01/E39381/html/ol_createlocal_repo.html and https://www.oracle.com/technetwork/articles/servers-storage-admin/yum-repo-setup-1659167.html.

Updating a Local Yum Server with Ksplice Channels

- After Ksplice channels have been added to the local yum server, repositories are updated daily by the `/usr/sbin/uln-yum-mirror` script.
- The `uln-yum-mirror` package is installed when creating a local ULN mirror and is a prerequisite to mirroring Ksplice channels.
- This package installation adds an `anacron` job that runs once per day:
  - `/etc/cron.daily/uln-yum-mirror`
- To synchronize the local Ksplice channels without waiting for the `anacron` job to run, or if the `anacron` job has been disabled, run the `uln-yum-mirror` script manually:

```
# /usr/bin/uln-yum-mirror
```


Configuring a local Ksplice mirror presumes a local ULN mirror has been previously set up, as noted before, and described here:

https://docs.oracle.com/cd/E37670_01/E39381/html/ol_createlocal_repo.html and https://www.oracle.com/technetwork/articles/servers-storage-admin/yum-repo-setup-1659167.html.

Ksplice channels are merely added to any existing channels on the pre-existing local ULN mirror.

As part of setting up a local ULN mirror, the `uln-yum-mirror` package is installed. This creates the `/etc/cron.daily/uln-yum-mirror` `anacron` job, which runs once per day.

The behavior of the `uln-yum-mirror` script can be configured with the /`etc/sysconfig/uln-yum-mirror` file. For example, to disable the daily run of the `anacron` job, edit this file to set the following value:

```
CRON_ENABLED=0
```

If there is a need to synchronize the Ksplice repositories without waiting for the `anacron` job to run, or if the `anacron` job has been disabled, the /`usr/sbin/uln-yum-mirror` script can be run manually:

```
# /usr/sbin/uln-yum-mirror
```

`cron` and `anacron` are discussed in the lesson titled “Automating Tasks.”

Configuring Ksplice Offline Standard Clients to Use the Local Ksplice Mirror

To configure a system as a Ksplice Offline Standard Client, perform the following steps:

1. In the `/etc/yum.repos.d` directory, edit any existing repository files 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 Standard 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 above 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 Standard Client:

1. In the `/etc/yum.repos.d` directory, edit any existing repository files 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:

```
[local_ol7_x86_64_ksplice]
name=Ksplice for $releasever - $basearch
baseurl=http://<IP_address_of_local_yum_server>/yum/OracleLinux/OL
    7/ksplice/$basearch/
...
enabled=1
```

3. Install the Ksplice Offline Standard Client:

```
# yum install uptrack-offline
```

4. Install the Ksplice updates that are available for the kernel:

```
# yum install uptrack-updates-`uname -r`
```

See https://docs.oracle.com/cd/E37670_01/E39380/html/ol_cfgolc_ksplice.html for further details.

In Oracle Cloud Infrastructure, an Oracle Cloud Infrastructure mirror is used for offline updates, rather than a separately configured Ksplice mirror. Since this mirror is configured for you, in this case, steps 1 and 2 are replaced with ensuring the proper repository is enabled and then continuing with the remaining steps on the slide.

The Ksplice Enhanced Client

- Is available for Oracle Linux 6 and Oracle Linux 7
- Allows updates to Ksplice-aware `glibc` and `openssl` shared libraries, in addition to kernel updates
- Online and offline configurations can be set up.
- Managed with the `ksplice` command instead of `uptrack` commands

Implementing the Ksplice Online Enhanced Client

1. Register your systems with the Unbreakable Linux Network (ULN) or have access to ULN channels on a mirror.
2. Subscribe to the appropriate Ksplice channels.
3. Use the `yum` command to install the `ksplice` package.
4. Install the Ksplice-aware versions of the `glibc` and `openssl` user space libraries.
5. Perform any required configuration.
6. Reboot your system once to use the new user space libraries.
7. Install available user space and kernel updates.
8. View the status from the System Status page of the Ksplice web interface at https://status-ksplice.oracle.com.

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

The following channels are available for the Ksplice Enhanced Client on Oracle Linux 7:

- **`ol7_x86_64_ksplice`**: Ksplice clients, updates, and dependencies for Oracle Linux 7 (x86_64)
- **`ol7_x86_64_userspace_ksplice`**: Latest Ksplice-aware user space packages for Oracle Linux 7 (x86_64)

Subscribe to both the ksplice and userspace_ksplice channels.

The Ksplice package is installed with:

```
# yum install -y ksplice
```

The Ksplice-aware versions of `glibc` and `openssl` are then installed with other updates:

```
# yum update
```

Or by only selecting the Ksplice-aware versions of `glibc` and `openssl`:

```
# yum update glibc* openssl*
```

Reboot your system once to use the new Ksplice-aware libraries.

Install available user space and kernel updates:

```
# ksplice -y user upgrade
# ksplice -y kernel upgrade
```

To enable automatic update installation, set `autoinstall = yes` in the `uptrack` configuration file `/etc/uptrack/uptrack.conf`.

Managing the Ksplice Enhanced Client

- Use the `ksplice` command instead of `uptrack` commands. The following are examples:
  - Display user space processes the client can patch:
    ```
    # ksplice all list-targets
    ```
  - Display updates that have been applied:
    ```
    # ksplice all show
    ```
  - List available kernel updates:
    ```
    # ksplice -n kernel upgrade
    ```
  - Install available user space updates:
    ```
    # ksplice -y user upgrade
    ```
  - Show the effective kernel:
    ```
    # ksplice kernel uname -r
    ```
- See the `ksplice(8)` man page for further details.

ORACLE

Quoted from the `ksplice` man page:

"The ksplice tool manages Ksplice updates for multiple subsystems. For each command the SUBSYSTEM parameter may be specified as `kernel`, `user` or `all` to apply the commands to the kernel, user-space, and both respectively.

There are a number of commands for managing Oracle Ksplice updates. The supported commands are:

ksplice [OPTIONS] SUBSYSTEM list-targets [COMAND_OPTIONS...]

ksplice [OPTIONS] SUBSYSTEM show [COMAND_OPTIONS...]

ksplice [OPTIONS] SUBSYSTEM apply [COMAND_OPTIONS...] UPDATE

ksplice [OPTIONS] SUBSYSTEM undo [COMAND_OPTIONS...] KID

ksplice [OPTIONS] SUBSYSTEM upgrade [COMAND_OPTIONS...]

ksplice [OPTIONS] SUBSYSTEM remove [COMAND_OPTIONS...] [KIDs]

Additionally, the `uname` command is available for the kernel subsystem:

ksplice [OPTIONS] kernel uname [COMAND_OPTIONS...]"

Configuring Ksplice Offline Enhanced Clients to Use a Local Ksplice Mirror

After configuring a local Ksplice mirror:

1. In the `/etc/yum.repos.d` directory, edit any existing repository files and disable all entries by setting `enabled=0`, and then create the `local-yum.repo` file.
2. Install the Ksplice Offline Enhanced Client:
   ```
   # yum install ksplice-offline
   ```
3. Install the following user space packages:
   ```
   # yum install ksplice-updates-glibc ksplice-updates-openssl
   ```
4. Install the kernel package and apply updates to your kernel version:
   ```
   # yum install uptrack-updates-`uname -r`
   ```
5. Install updates to Ksplice-aware user space libraries:
   ```
   # yum update glibc* openssl*
   ```
6. Reboot once to make the Ksplice-aware user space libraries available.

When setting up a local yum server that can act as a Ksplice mirror, ensure that proper channels are selected. For example: `ol7_x86_64_ksplice, ol7_x86_64_userspace_ksplice, ol7_latest, ol7_UEK4_latest, ol7_addons`

Import the GPG key if not already done:

```
# rpm --import /usr/share/rhn/RPM-GPG-KEY
```

The `local-yum.repo` file contains entries such as the following for each needed repo, per the above, prefaced with "`local`":

```
[local_ol7_x86_64_ksplice]
name=Ksplice for $releasever - $basearch
baseurl=http://<IP_address_of_local_yum_server>/yum/OracleLinux/OL
    7/ksplice/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY
gpgcheck=1
enabled=1
```

When using the "`local`" prefix for the user space channel `ol7_x86_64_userspace_ksplice` in the `local-yum.repo` file, the following directive must be placed in the `/etc/uptrack/uptrack.conf` file:

```
[User]
yum_userspace_ksplice_repo_name = local_ol7_x86_64_userspace_ksplice
```

If the "`local`" prefix is not used and the channel label exactly matches what is on ULN, the previous `uptrack.conf` entry is not needed.

In Oracle Cloud Infrastructure, an Oracle Cloud Infrastructure mirror is used for offline updates, rather than a separately configured Ksplice mirror. Since this mirror is configured for you, in this case, step 1 on the previous slide is replaced with ensuring the proper repositories are enabled and then continuing with the remaining steps on the slide.

The system must be rebooted once after running the `yum update` command, described previously, to use the new user space libraries. After this, reboots are not required when applying Ksplice updates to these libraries.

The following installs the relevant uptrack package and applies any updates that are available for the kernel:

```
# yum install uptrack-updates-`uname -r`
```

Change the autoinstall setting to "yes" in `/etc/uptrack/uptrack.conf` to enable automatic installation of updates:

```
autoinstall = yes
```

When `ksplice-updates-glibc` and `ksplice-updates-openssl` packages are installed with `autoinstall = yes`, the ksplice cron job will check for and apply any available updates to the user space running processes. Otherwise, run `ksplice -y user upgrade` to apply available updates to the user space processes manually.

You can set up a cron job to install the `uptrack-updates-`uname -r``, `ksplice-updates-glibc`, and `ksplice-updates-openssl` packages on a regular basis.

Alternatively, once the `ksplice-updates-glibc` and `ksplice-updates-openssl` packages are installed initially, `yum update` will pick up any new in-memory updates for the `glibc` and `openssl` user space running processes, by updating these packages. Then, with `autoinstall = yes`, the ksplice cron job will check for and apply any available updates to the user space running processes. Otherwise, `ksplice -y user upgrade` must be run to apply available updates to the user space processes manually.

In addition to patching existing user space processes in memory, `ksplice user upgrade` (via `cron` or manually) updates shared objects (.so) on disk so that new processes load patched versions of `glibc` and `openssl` libraries.

See https://docs.oracle.com/cd/E37670_01/E39380/html/ksplice-enhanced-offline.html

for further details.

Quiz

Ksplice kernel patches are effective immediately without requiring a reboot.

a. True
b. False

Answer: a

Quiz

Which of the following commands is primarily used to manage the Ksplice Enhanced Client?

a. uptrack

b. ksplice

Answer: b

Summary

In this lesson, you should have learned how to:

- Describe the operation and features of Oracle Ksplice
- Implement the Ksplice Online Standard Client
- View Ksplice packages on ULN
- Describe Ksplice implementation in Oracle Cloud Infrastructure
- Use Ksplice commands
- Use the Ksplice web interface
- Use the Ksplice Offline Standard Client
- Implement the Ksplice Online Enhanced Client
- Configure Ksplice Offline Clients to use a local Ksplice mirror

Practice 10: Overview

This practice covers the following topics:

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