

Oracle Linux 9

Installing Oracle Linux



F51031-15
November 2024



Oracle Linux 9 Installing Oracle Linux,

F51031-15

Copyright © 2022, 2024, Oracle and/or its affiliates.

Contents

Preface

Documentation License	vi
Conventions	vi
Documentation Accessibility	vi
Access to Oracle Support for Accessibility	vi
Diversity and Inclusion	vi

1 About Deploying Oracle Linux

2 Preparing to Install

System Requirements	2-1
System Configuration	2-2
Obtaining Installation ISO Images	2-2
Verifying ISO Downloads	2-3
Preparing Installation Media	2-3
USB Flash Drive	2-3
DVD or CD	2-5
Network Drive	2-6
Driver Update Disk	2-6

3 Customizing Boot Loader Actions

Configuring the Boot Loader	3-1
Installation Boot Options	3-2
Installation Type	3-2
Installation Source	3-2
Network Configuration	3-3
Kickstart Installations	3-3
Remote Installations	3-4
Miscellaneous Boot Options	3-4

4 Installing Oracle Linux Manually

Installing in Graphics Mode	4-1
Welcome Screen	4-1
Installation Summary	4-2
Localization	4-3
Keyboard Layout	4-3
Language Support	4-3
Date and Time	4-4
Software	4-5
Installation Source	4-5
Software Selection	4-6
System	4-6
Installation Destination	4-7
KDUMP	4-8
Network and Host Name	4-9
Security Profile	4-10
User Settings	4-10
Root Password	4-11
User Creation	4-12
Completing the Installation	4-13
Installing in Text Mode	4-13
Postinstallation Configuration	4-14

5 Automating the Installation by Using Kickstart

Customizing the Kickstart Configuration File	5-1
Installation Options	5-2
Packages to Install	5-3
Preinstallation Options	5-4
Postinstallation Options	5-5
Starting the Kickstart Installation	5-6

6 Using a Driver Update Disk During Installation

7 Installing a System With a Btrfs root File System

Using the GUI Installer	7-1
Using the Text Installer	7-3
Verifying the Btrfs Installation	7-3

8 Troubleshooting Oracle Linux Installations

Troubleshooting Menu Options	8-1
Debug and Log Information	8-1

A Creating a Network Installation Setup

Setting Up the Network Server	A-1
How to Configure NFS	A-1
How to Configure dnsmasq	A-3
Configuring PXE Boot Loading	A-4
How to Configure for BIOS-based PXE Clients	A-5
How to Configure for UEFI-based PXE Clients	A-6
Supporting Different Kinds of Clients	A-8

Preface

[Oracle Linux 9: Installing Oracle Linux](#) provides information about how to install the Oracle Linux 9 release.

Documentation License

The content in this document is licensed under the [Creative Commons Attribution–Share Alike 4.0 \(CC-BY-SA\)](#) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and

the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Deploying Oracle Linux

Oracle Linux can be installed in various ways. This document focuses on installation of Oracle Linux by using an installation image that's available in ISO format. You can download the installation image and load it to bootable media to boot a bare metal system or virtual machine into the installer. The installer is used to install the operating system onto the hard disks or block devices that are available to the system.

You can also obtain media for other install environments, such as container images, cloud images in `qcow2` and `ova` formats, and a Raspberry Pi installation image. See <https://yum.oracle.com/oracle-linux-downloads.html> for more information about the available formats that you can download.

For developers who use the Raspberry Pi hardware platform, Oracle provides an unsupported developer release image, which includes the required firmware to boot this platform. For more information about using the Raspberry Pi hardware platform, see [Install Oracle Linux on a Raspberry Pi](#).

Many cloud providers include Oracle Linux as an option for cloud instance deployment. In these cases, the images are predefined and you can select the appropriate cloud image for the Oracle Linux release that you want to use and deploy that. Oracle Linux releases are provided as platform images on Oracle Cloud Infrastructure (OCI). You can use these images to deploy Oracle Linux to compute instances. For information about these images, see the release notes for the specific image that you're using on the OCI Documentation page at <https://docs.oracle.com/iaas/images/>. For information about using Oracle Linux instances on OCI, <https://docs.oracle.com/en-us/iaas/oracle-linux/home.htm>.

2

Preparing to Install

Whether you're installing on a single system or on multiple systems, plan ahead to ensure a successful installation.

- [System Requirements](#)
- [System Configuration](#)
- [Obtaining Installation ISO Images](#)
- [Preparing Installation Media](#)



Note:

To upgrade from the latest version of the previous major Oracle Linux release to the current major release version, use the Leapp utility. See [Oracle Linux 9: Upgrading Systems With Leapp](#).

System Requirements

Verify that the system fulfills the following minimum requirements. In general, having more resources in the system improves a system's performance.

- Minimum of 2 logical CPUs up to 2048 logical CPUs
- Minimum x86-64-v2 micro architecture level for CPUs when installing on x86_64 hardware



Tip:

If you're unsure whether the hardware can support x86-64-v2 or higher, consider installing Oracle Linux 8 or Oracle Linux 7 and run the following command:

```
sudo /lib64/ld-linux-x86-64.so.2 --help | grep x86-64-v
```

If the output includes **(supported, searched)**, next to the micro architecture level, then that level is supported on the CPU. For example, the following shows that micro architecture levels 2 and 3 are supported but not 4:

```
sudo /lib64/ld-linux-x86-64.so.2 --help | grep x86-64-v
x86-64-v4
x86-64-v3 (supported, searched)
x86-64-v2 (supported, searched)
```

- 1.5 GB of memory per logical CPU, up to a maximum of 64 TB
- At least 10 GB of disk space (20 GB is the recommended minimum)

- On UEFI systems, ensure that the target disk uses GPT (GUID Partition Table), as some UEFI firmwares don't support UEFI/MBR boot

See the following extra resources for information related to installation issues and system requirements:

- <https://linux.oracle.com/hardware-certifications>
- The Release Notes of the Oracle Linux version you're installing at [General Oracle Linux documentation](#)
- [Unbreakable Enterprise Kernel documentation](#)

System Configuration

The configuration of the system itself also has an impact on the installation. Consider the following:

Storage

- Storage device and partition on which the operating system is installed. The installation program displays a warning if disk space is insufficient.
- Storage space for each file system (`/`, `/boot`, `/home`, `/var/tmp`, and so on), the file system type, and whether to encrypt the block device underlying each file system.
- Layout and configuration of the storage devices such as use of logical volume management RAID configuration, encryption, and others.
- For iSCSI or FCoE connections, the WWID or the port, target, and LUN to be used.

Network

- Required network setup using DHCP or static addresses, FQDN or host name, and so on.
- Other specialized network interfaces to be configured during installation, such as VLANs and network bonding.

Software

- URLs of any other repositories and proxy settings to be used during installation.
- Software packages to be installed based on system's intended purpose, such as a web server.

Obtaining Installation ISO Images

To install Oracle Linux, download the installation images from one of the following locations:

- Oracle Linux yum server at <https://yum.oracle.com/oracle-linux-isos.html>.
- Oracle Software Delivery Cloud at <https://edelivery.oracle.com>.

The following installation images are available for both the x86_64 platform and the aarch64 platform, unless indicated otherwise:

- Full ISO of Oracle Linux for typical on-premises installations.
- Boot ISO of Oracle Linux for network installations.
- Boot ISO of the supported UEK release for installations in which you want to use the Btrfs file system or for installing on hardware that's supported only on UEK. For instructions to install Oracle Linux with the Btrfs file system, see [Installing a System With a Btrfs root File System](#).

- Source DVDs that contain the source code for the software packages in the release.

Verifying ISO Downloads

To verify an Oracle Linux ISO download, you need the following:

- The checksum file corresponding to the downloaded ISO.
- The public GPG key to verify the checksum file signature.

The following steps show you how you can verify an Oracle Linux ISO download:

1. Download and import the public GPG key.

For example, to download the GPG key for Oracle Linux 9, run the following command:

```
curl https://yum.oracle.com/RPM-GPG-KEY-oracle-ol9 | sudo gpg --import
```

2. Download the appropriate checksum file to the directory containing the downloaded ISO.

For example, to download the checksum for Oracle Linux 9 (x86_64 platform) you would run:

```
curl -O https://linux.oracle.com/security/gpg/checksum/OracleLinux-R9-U4-Server-x86_64.checksum
```

3. Verify the integrity of the checksum file you have downloaded:

```
sudo gpg --verify OracleLinux-R9-U4-Server-x86_64.checksum
```

4. Use the `sha256sum -c` command to verify the checksum for downloaded ISO:

```
sudo grep OracleLinux-R9-U4-x86_64-dvd.iso OracleLinux-R9-U4-Server-x86_64.checksum | sudo sha256sum -c
```

For more information, and download links for other Oracle Linux release keys and checksum files, see <https://linux.oracle.com/security/gpg/>

Preparing Installation Media

Before you can use an ISO image to install Oracle Linux, you must first store it on bootable installation media, such as the following:

- [USB Flash Drive](#)
- [DVD or CD](#)
- [Network Drive](#)
- [Driver Update Disk](#)

USB Flash Drive

You can install Oracle Linux by using a boot image on portable devices such as a USB flash drive or an SD card, if the system's firmware supports booting from those devices.

To create a bootable drive, use the `dd` or `xorriso-dd-target` command. Or, use a separate third-party utility to write the ISO image to a drive. See, for example, [Create USB Installation Media for Oracle Linux with Fedora Media Writer](#).

 Caution:

This procedure destroys any existing data on the drive. Ensure that you specify the correct device name for the USB drive on the system.

1. Insert a USB flash drive into an Oracle Linux system.
2. Use the `xorriso-dd-target` command to list available block devices and to identify likely candidate devices for use.

```
xorriso-dd-target -with_sudo -list_all
```

The command presents a password prompt as it uses `sudo` to access all devices on the system. Output similar to the following is displayed:

```
sda : YES : usb+ has_vfat+ : SanDisk Cruzer Switch
nvme0n1 : NO : not_usb- has_vfat+ has_xfs- has_crypto_LUKS- has_swap- :
PM9A1 NVMe Samsung 512GB
```

The command identifies pluggable block devices that aren't used as system disk or sincere data storage. In the example output, the command identified a USB device at `/dev/sda`, that could be used to write an ISO image.

3. Ensure that any file systems on the device are unmounted.

For example, to unmount the first partition on `/dev/sda`:

```
sudo umount /dev/sda1
```

4. Write the contents of the ISO image file to the USB device.

Do one of the following to write the ISO image file to the USB device:

- Use the `dd` command directly:

```
sudo dd if=./full_image.iso of=/dev/sda bs=512k
```

- Use the `xorriso-dd-target` command to guide you through this process:

```
xorriso-dd-target -with_sudo -plug_test -DO_WRITE -image_file ./
full_image.iso
```

The command guides you through testing for appropriate devices and finally prompts you to select and approve writing to the device. Example output follows:

```
sudo /bin/lsblk seems ok.
```

```
Caused by option -plug_test: Attempt to find the desired device
by watching it appear after being plugged in.
```

Step 1:
Please make sure that the desired target device is plugged *_out_* now.
If it is currently plugged in, make sure to unmount all its filesystems
and then unplug it.
Press the Enter key when ready.

Found and noted as *_not_* desired: nvme0n1

Step 2:
Please plug in the desired target device and then press the Enter key.

Waiting up to 10 seconds for a new device to be listed found: sda
Now waiting 5 seconds to let it settle
Found and noted as desired device: sda

sda : YES : usb+ has_vfat+ : SanDisk Cruzer Switch

Step 3:
Last chance to abort. Enter the word 'yes' to start REAL WRITING.
yes
Looking for mount points of sda:
Performing:
sudo /bin/dd if=/dev/zero of=/dev/'sda' bs=512 seek='30595071'
count=1 status=none
sudo /bin/dd if='OracleLinux.iso' of=/dev/'sda' bs=1M status=progress
oflag=dsync ; sync

The USB flash drive is now ready to be used to boot a system and start the installation.

DVD or CD

Because of storage limits, optical media such as CDs or DVDs might not have capacity to accommodate most installation ISO images. However they can be used to store the boot ISO image.

1. Insert an empty recordable CD or DVD into the CD or DVD writer device.
2. Open a terminal and use `cdrecord` to write the ISO file to the device.

To write the downloaded ISO image file to a CD or DVD, use a command such as `cdrecord`, for example:

```
sudo cdrecord -v -eject speed=16 dev=/dev/sr0 file_name.iso
```

To display the device that corresponds to the CD or DVD writer, use the `cdrecord --devices` command.

The CD or DVD is now ready to be used to boot a system and start the installation.

Network Drive

You can copy installation media to a network drive to use as part of a network installation process. For the network drive, the image that you download can either be the full ISO image or the boot image, while the network server can be of any type, such as NFS or a web server.

To copy the ISO image to a network drive, follow these steps:

1. Mount the ISO image to a location on a system that has access to the network drive location.

For example you can loopback mount an ISO to the `/mnt` path as follows:

```
sudo mount -o loop full_image.iso /mnt
```

2. Copy the contents of the ISO image to the network drive location.

```
sudo cp -a -T path-to-mounted-ISO-image network_dir
```

For example, if you're using a web server to host the network drive, and have ISO image mounted at `/mnt`, you might run the following command:

```
sudo cp -a -T /mnt /var/www/html/OSimage/OL9
```

3. Unmount the ISO image after you have finished copying the contents.

For example, run:

```
sudo umount /mnt
```

Using a network drive is part of network installation, which requires you to build a network configuration that provides network installation functionality. For details, see [Creating a Network Installation Setup](#).

Driver Update Disk

A Driver Update Disk (DUD) provides a mechanism for delivering updated device drivers during system installation. On some systems, hardware might not be fully supported for an Oracle Linux release. In these cases, a DUD might be released later to help installation on newer hardware.

DUDs are released as modules and become available for previously unsupported hardware. The DUD is provided in the form of an ISO and is available in the Oracle Software Delivery Cloud or through MyOracle Support.

The DUD must be stored in an appropriate media or an alternative storage device before it can be used to for installation.

You can prepare a DUD in the same way that you prepare other installation media, by copying it to a USB, CD, or DVD device, or by storing it at an accessible network location. See [Preparing Installation Media](#) for more information.

3

Customizing Boot Loader Actions

Boot options control how the installation proceeds. In addition to default settings, you can add more options before you start the installation.

- [Configuring the Boot Loader](#)
- [Installation Boot Options](#)

Configuring the Boot Loader

You can configure the boot loader to use specific boot options that can control installation at the point of boot. For example, you can provide the network location of an install ISO, or the location of a Kickstart file used to automate installation.

Boot options are specified at the boot command line of the boot menu.

1. Boot the system.
2. Access the boot options in the boot menu.

When the boot menu is displayed, select any installation option, then press either **E** on UEFI-based systems or **Tab** on BIOS-based systems.

The boot options line is displayed with some default options already defined.

3. Add any required options at the end of the line.

Separate boot options with a space. Options that require parameters must be in the `option=parameter` format.

For a list of common boot options, see [Installation Boot Options](#).

4. Save the changes.

Save the changes by pressing either **Ctrl+X** on UEFI-based systems or **Return** on BIOS-based systems.

To discard changes and return to the boot menu, press **Esc**.

This example shows settings for the `inst.repo` and `inst.ks` options:

```
inst.repo=nfs:nfs.example.com:/ISOs/OL9/full_image.iso \  
inst.ks=nfs:nfs.example.com:/kickstart/OL9/server-ks.cfg ip=dhcp
```

With these directives, the installation program is instructed to do the following:

- Use the full installation image stored on an NFS share directory.
- Start the installation automatically by using a file also stored on an NFS share directory.

Installation Boot Options

This list contains some commonly used boot options that further control how the installation proceeds. You specify these options at the installation menu before you start the installation process.

The following resources provide more information about boot options:

- <https://anaconda-installer.readthedocs.io/en/latest/boot-options.html>
- The `dracut.cmdline(7)` manual page provides Dracut kernel command line options.

Installation Type

`inst.graphical`

Specifies a graphical-based installation.

`inst.gpt`

Applies only to BIOS-based systems with less than 2 TiB sized disk. Specifies to use the GUID partition table when boot loader is installed instead of the default MBR.

Installation Source

`inst.repo=cdrom[:device]`

Specifies a CD or DVD drive as the location that contains everything needed to install the software.

The installation program searches all the system's CD or DVD drives, unless a *device* is specified.

If access to a network is required and no network boot options are specified, the installation program enables DHCP on all available network devices.

`inst.repo=ftp://user : password@FTP_server/path`

Specifies an FTP server as the location that contains everything needed to install the software.

If access to a network is required and no network boot options are specified, the installation program enables DHCP on all available network devices.

`inst.repo=hd:device:path`

Specifies a local disk as the location that contains everything needed to install the software.

You can specify the *device* by its device name (`sdb2`), label (`LABEL=label`), or UUID (`UUID=uuid`).

`inst.repo=[http|https]://HTTP_server/path`

Specifies a web server as the location that contains everything needed to install the software.

If the system has access to the Internet, you can use the BaseOS repository on the Oracle Linux yum server for the Oracle Linux release that you're installing. For example, you could set this value to `https://yum.oracle.com/repo/OracleLinux/OL9/baseos/latest/x86_64`

`inst.repo=nfs:[options]:NFS_server:path`

Specifies an NFS share as the location that contains everything needed to install the software.

Use *options* to specify a comma-separated list of NFS mount options.

The NFS share can be the path to an ISO image or a directory.

inst.stage2=[*installation_source*]

Specifies the location to fetch the installer runtime image; packages are ignored. If this option is not specified, `inst.repo` is used instead. The path specified for *installation_source* can match any of the protocol and path options used for `inst.repo`.

The directory path specified for the *installation_source* should contain a valid `.treeinfo` file that specifies the location of the runtime image. If a `.treeinfo` file is not present at the source, the installer uses `LiveOS/squashfs.img` as the default location at the specified source.

! Important:

When specifying a runtime image in UEK network installations, use the UEK ISO for the installation repository as well. Do not combine using the standard Oracle Linux ISO image as the installation repository while specifying the Oracle Linux UEK Boot ISO for the runtime image.

Network Configuration

ip=[*interface*]:dhcp|dhcp6|auto6|ibft

Specifies a network automatic configuration method. If *interface* is not specified, all interfaces are configured. Use `ibft` to use the MAC address of the interface specified by the iSCSI Boot Firmware Table (iBFT) in the system BIOS or firmware.

ip=*ip::gateway:netmask:hostname:interface:none*

Specifies a static IP configuration for *interface*. Enclose IPv6 addresses in square brackets, for example `[2509:f0d0:1001::0004]`.

nameserver=*IP*

Specifies the IP address of a DNS server to use during installation. Multiple `nameserver` options can be used.

bootdev=*interface*

Specifies the primary network interface. Required if you use more than one `ip` option.

inst.dhcpclass

Specifies a vendor class identifier to DHCP.

Kickstart Installations

inst.ks=cdrom[:*device*]/*path*

Specifies a Kickstart file on a CD or DVD drive.

inst.ks=ftp://*user* : *password*@*FTP_server*/*path*

Specifies a Kickstart file on an FTP server.

inst.ks=hd:*device*:*path*

Specifies a Kickstart file a local disk.

You can specify the *device* by its device name (`sdb2`), label (`LABEL=label`) or UUID (`UUID=uuid`).

inst.ks=[http|https:]//*HTTP_server*/*path*

Specifies a Kickstart file on a web server.

inst.ks=nfs:[options:]NFS_server:path

Specifies a Kickstart file on an NFS share.

Use *options* to specify a comma-separated list of NFS mount options.

Remote Installations

inst.vnc

Enables remote graphical-based installation by starting a VNC server.

A VNC client can connect by using a command such as `vncviewer server:port`, where *server* is the IP address of the system being installed.

After installation, the system starts in text mode even if a graphical desktop environment is selected as the base environment.

inst.vncconnect=client[:port]

Specifies the VNC client and optional port that is listening for connections from a VNC server (`vncviewer -listen`). The default port is 5900.

inst.vncpassword=password

Specifies the password for client connections using VNC.

Miscellaneous Boot Options

inst.keymap=layout

Specifies the keyboard layout for installation.

inst.lang=language

Specifies the language for installation.

inst.loglevel=level

Specifies the logging level for installation: *critical*, *debug*, *error*, *info*, or *warning*. The default level is *info*.

4

Installing Oracle Linux Manually

Installation begins when you boot the system from the installation media. The installation menu displays 3 options:

- Install Oracle Linux
- Test the media and then install Oracle Linux (default)
- Troubleshooting

Two modes are available to manually install Oracle Linux:

- **Graphics mode** where a graphical user interface guides you through the installation process. Selecting either the first or the second option in the menu starts the installation in graphic mode by default.
- **Text mode** which has limited options for installing the operating system.

Installing in Graphics Mode

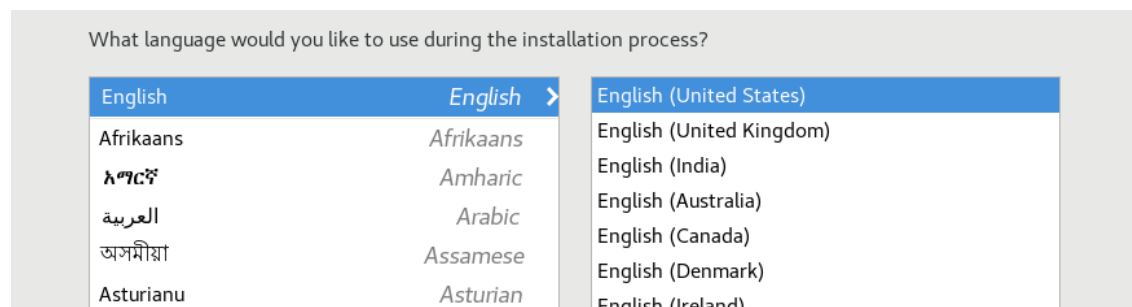
The graphics mode consists of installation directives in a series of graphical screens where you click representative icons and buttons to set the directives.

- **Welcome Screen**
- **Installation Summary**
- **Localization**
- **Software**
- **System**
- **User Settings**

Welcome Screen

The Welcome Screen is the first screen to appear after the system completes the boot process.

Figure 4-1 Welcome Screen

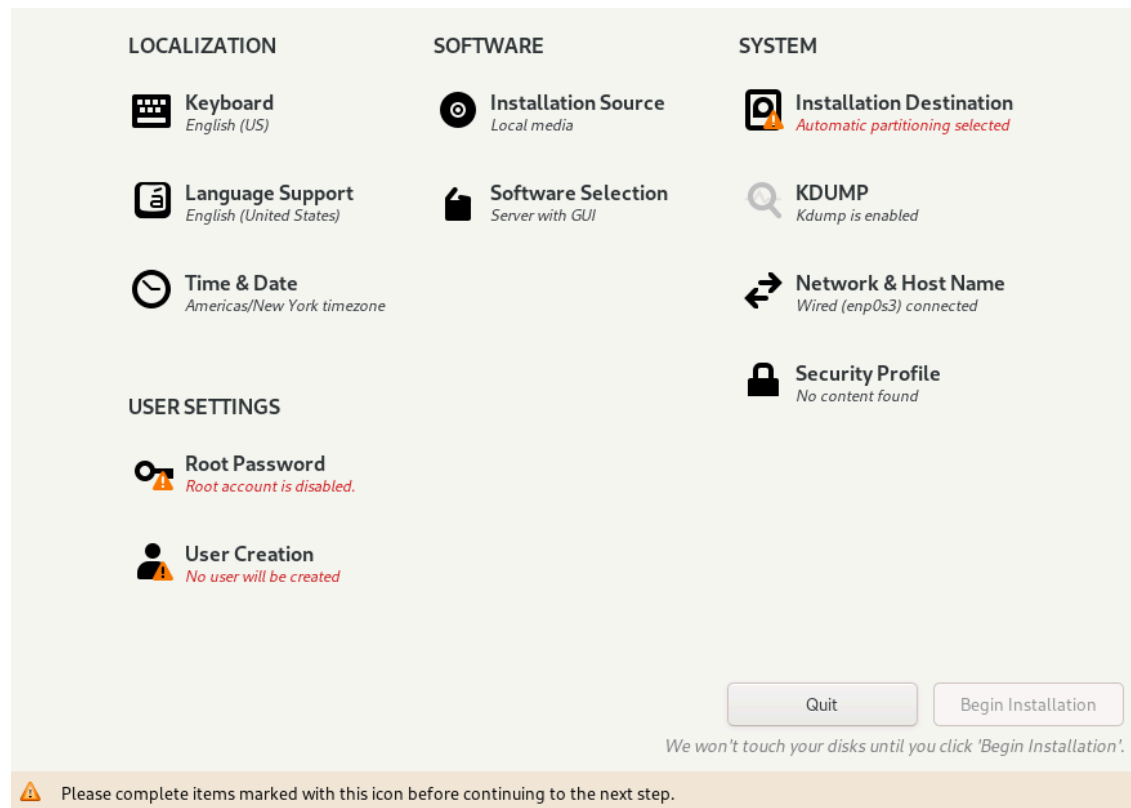


On this screen, select the preferred language to be used during the installation process. You can further select a specific locale, if any, for the selected language. Then click **Continue** to proceed.

Installation Summary

The Installation Summary screen appears after you have selected the installation language. It functions as the home or main screen.

Figure 4-2 Installation Summary



The screen provides four categories of options: LOCALIZATION, SOFTWARE, SYSTEM, and USER SETTINGS. Clicking an option under any of these categories opens other screens to configure the selected option.

The options under each category usually have default values. You can skip opening any of the options screens to use the default values. However, as a minimum, you **must** visit the options flagged with a warning icon.

After defining directives in a specific screen, click **Done** to return to the Installation Summary screen. Then you can configure other options. As you visit each configuration screen, pay attention to any warning messages that are displayed at the bottom of the screen.

You can continue to change installation configuration options until the installation begins. The installation doesn't begin until you click **Begin Installation** at the bottom of the screen. In turn, the **Begin Installation** button remains disabled until all the configuration warning flags have been cleared.

**Note:**

At the top right of the screen is the Keyboard switch. This switch appears in all the option screens to enable you to change to a different available keyboard layout at any time during configuration. See [Keyboard Layout](#) for more details about keyboard layouts.

Localization

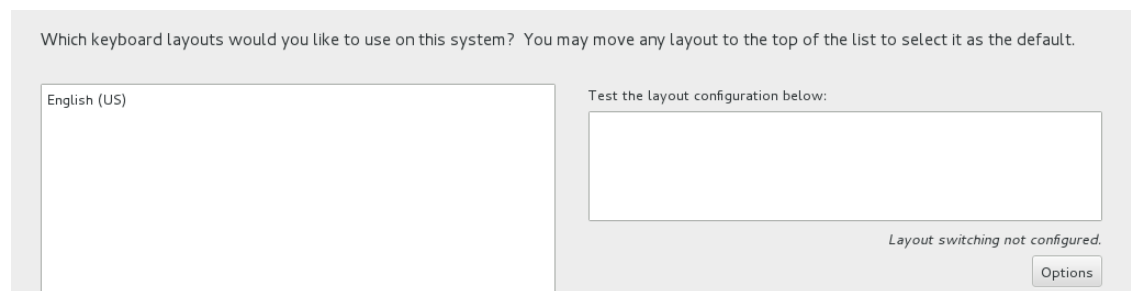
Under Localization, you configure the following options:

- [Keyboard Layout](#)
- [Language Support](#)
- [Date and Time](#)

After configuring any of these options, click **Done** to return to the Installation Summary screen.

Keyboard Layout

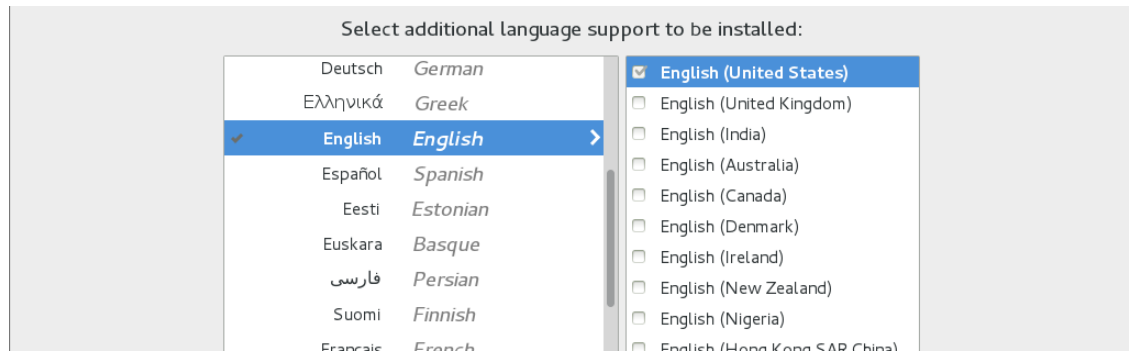
Figure 4-3 Keyboard Layout



On the left pane, you can add other keyboard layouts to the preselected default layout. You can also revise the order of the listed layouts. The layout at the top of the list becomes the default layout.

Language Support

Configuring language support consists of specifying other locales of the selected language that you want the system to support. This option is similar to the configuration of the keyboard layout at the beginning of the installation.

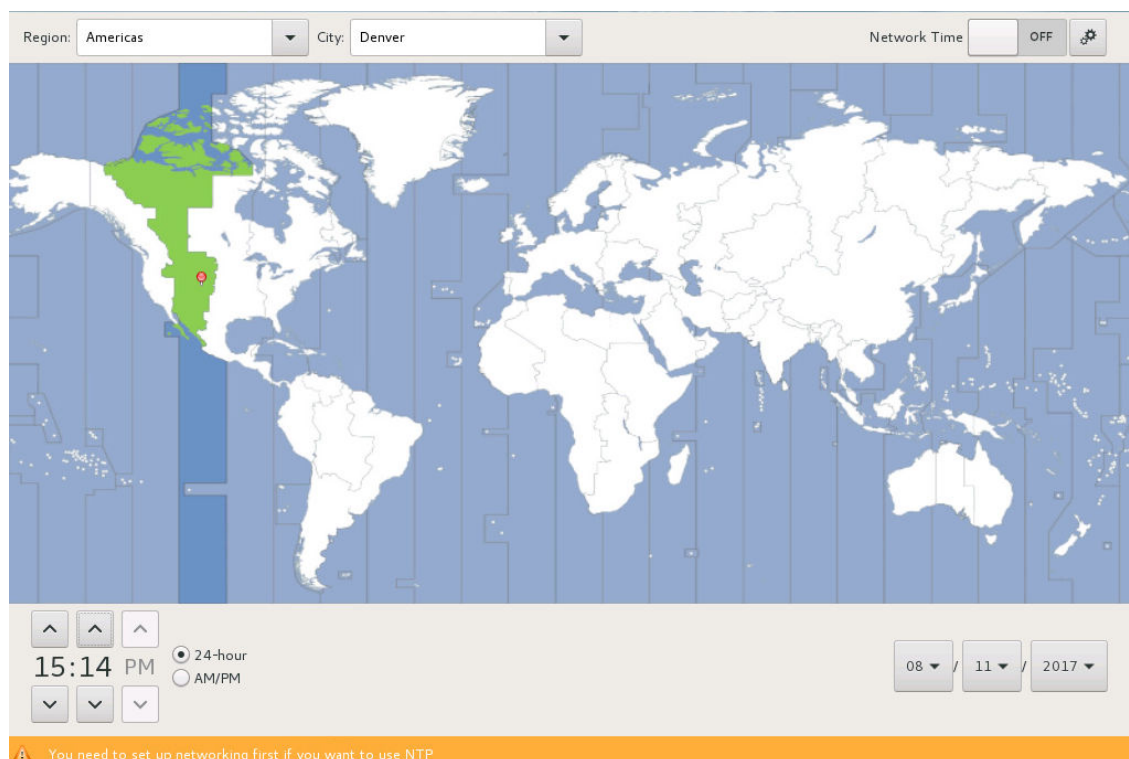
Figure 4-4 Language Support

From the list of languages on the left pane, select other languages for the system. Then, on the right pane, select from the available locales for that language.

Date and Time

The Date & Time screen enables you to set the following options:

- Time zone for the system
- Actual time and the format for displaying time
- Current date

Figure 4-5 Date & Time

To select the system's time zone, click an area on the map that approximates the site that you want the system to be located. Or, you can select from the Region and City drop down lists to specify that location.

To specify time, time format, and date, click the appropriate buttons at the bottom of the screen.

To enable NTP, switch the **Network Time** switch at the upper right corner of the screen to ON. Click **Settings** to display a dialog box, where you can configure the NTP servers that the system would use. You can skip configuring NTP until later by using the Chrony suite. See [Oracle Linux 9: Setting Up Networking](#). See also [Configure Chrony on Oracle Linux](#).

Software

Under Software, you configure the following options:

- [Installation Source](#)
- [Software Selection](#)

After configuring any of these options, click **Done** to return to the Installation Summary screen.

Installation Source

The Installation Source window identifies the source image that you use to install Oracle Linux.

Figure 4-6 Installation Source

The screenshot shows the 'Which installation source would you like to use?' window. The 'Auto-detected installation media' radio button is selected. Below it, the 'Device: sr0' and 'Label:' fields are visible, along with a 'Verify' button. The 'On the network' radio button is unselected. Below it, the 'Closest mirror' dropdown menu is open, showing 'repository URL' as the selected option. The 'URL type:' dropdown menu is also open, showing 'repository URL' as the selected option. The 'Additional repositories' section shows a table with columns 'Enabled' and 'Name'. The 'AppStream' repository is listed with the 'Enabled' checkbox checked. To the right of the table, the 'Name:' field is set to 'AppStream', the 'file://' dropdown menu is open, showing '/run/install/sources/mount-0000-cdrom/AppStream', the 'URL type:' dropdown menu is set to 'repository URL', and the 'Proxy URL:', 'User name:', and 'Password:' fields are empty.

If you use the *full ISO image* as the source, the install program detects that image. By default, the **Auto-detected installation media** button is selected and **Appstream** is listed as an additional installation repository. Because the image contains all the packages required for a system installation, you can use the default configuration to proceed with the installation.

If you use the *boot ISO image* as the source, the **On the network** button is selected as the installation mode and the **Closest mirror** option is selected as the repository source. Optionally, you can specify a local mirror as a repository source, in which case you would need to provide the mirror's path. If no path is specified, the Oracle Linux yum server is used by default. The installer automatically uses the required repositories from the network mirror to install the OS. However, the repositories aren't listed in the window. Because the installation image is configured to automatically use the Oracle Linux yum server if no mirror path is specified, you can use the default configuration to proceed with the installation.

To install extra packages from other sources, use the **Additional repositories** box to add those sources, and then provide the necessary information about those sources in the text fields next to the box.

Software Selection

Software selection refers to the profile or base environment to be used during the installation.

Figure 4-7 Software Selection

The screenshot shows a software selection window with two main panels. The left panel, titled 'Base Environment', contains a list of radio buttons for selecting a base environment. The 'Minimal Install' option is selected and highlighted with a blue background. The right panel, titled 'Additional software for Selected Environment', contains a list of checkboxes for selecting additional software. The 'Guest Agents' checkbox is checked, while all other checkboxes are unchecked.

Base Environment	Additional software for Selected Environment
☐ Server with GUI An integrated, easy-to-manage server with a graphical interface.	☒ Guest Agents Agents used when running under a hypervisor.
☐ Server An integrated, easy-to-manage server.	☐ Standard The standard installation of Oracle Linux.
☒ Minimal Install Basic functionality.	☐ Legacy UNIX Compatibility Compatibility programs for migration from or working with legacy UNIX environments.
☐ Workstation Workstation is a user-friendly desktop system for laptops and PCs.	☐ Console Internet Tools Console internet access tools, often used by administrators.
☐ Custom Operating System Basic building block for a custom OL system.	☐ Container Management Tools for managing Linux containers
☐ Virtualization Host Minimal virtualization host.	☐ Development Tools A basic development environment.
	☐ .NET Development Tools to develop and/or run .NET applications
	☐ Graphical Administration Tools Graphical system administration tools for managing many aspects of a system.
	☐ Headless Management Tools for managing the system without an attached graphical console.

Each **Base Environment** represents a base set of functionalities and the required packages and software to fulfill those functionalities. By default, the **Server with GUI** environment is used. Choose the base environment that best fits the purpose of the system on which you're installing Oracle Linux. From the right pane, you can add other profiles to the base environment that you selected on the left pane. The other profiles group packages that you might need to perform different functions on the system.

System

Under System, you configure the following options:

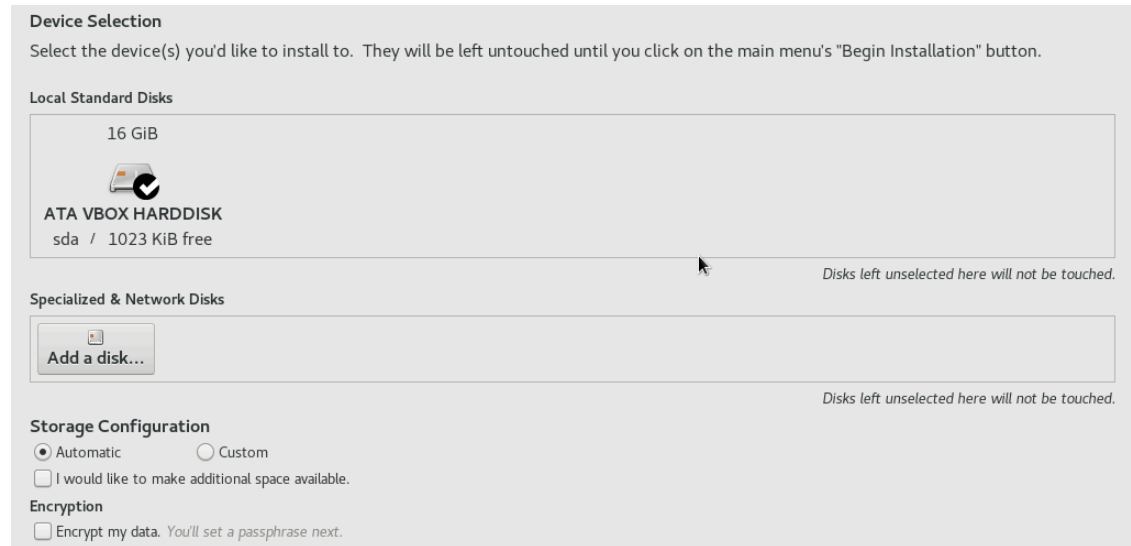
- [Installation Destination](#)
- [KDUMP](#)
- [Network and Host Name](#)
- [Security Profile](#)

After configuring any of these options, click **Done** to return to the Installation Summary screen.

Installation Destination

The installation destination is used to configure the disks where Oracle Linux is installed. Even if you accept the default settings, you must still open the Installation Destination screen to clear the option's warning icon.

Figure 4-8 Installation Destination



You can configure the installer to use local storage devices in the **Local Standard Disks** section of the window. A check mark icon on the disk device is displayed for any selected devices.

If you need to add network based or specialized storage such as iSCSI or NVMe devices, you can click the **Add a disk...** button in the **Specialized & Network Disks** section of the window.

You can configure other options for the target destination in the **Storage Configuration** section of the window:

- **Automatic:** radio button enables automatic partitioning.
 - **I would like to make additional space available:** set this checkbox to reclaim space from an existing partitioning layout.
 - **Encrypt my data:** set this checkbox to encrypt partitions using Linux Unified Key Setup (LUKS). You're prompted to set a LUKS encryption passphrase when you click the **Done** button.
- **Custom:** radio button enables custom partitioning. If you set this option, the **Manual Partitioning** configuration page is displayed after you click the **Done** button.

Manual Partitioning

Detected mount points are listed in the left pane in the window. Mount points are grouped by any detected operating system installations. Options for each mount point are displayed in the right pane when you click a mount point in the left pane.

If the system contains existing file systems, ensure that enough space is available for the installation. To remove any partitions, select them in the list and click the - button.

If no partitions exist on the disk and you want to create a set of partitions as a starting point, select a partitioning scheme from the left pane, and click **Click here to create them automatically**. The installer automatically creates standard partitions and mount points that you can customize and adjust.

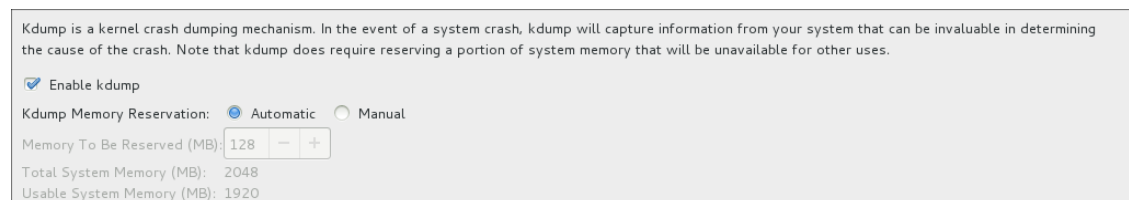
The following options are available when configuring a mount point:

- **Mount point:** If a file system is the root file system, enter /, enter /boot for the /boot file system, and so on. No mount point is required for a swap file system.
- **Desired Capacity:** Set the value to the size of the file system that you want to create. You can use common size units such as KiB or GiB. The default size unit is MiB.
- **Device Type:** Set the device type to one of `Standard Partition`, `LVM`, or `LVM Thin Provisioning`. You can configure RAID options if two or more disks are configured for manual partitioning and you select an LVM device type. You can also configure the LVM **Volume Group**, if an LVM device type is selected.
- **File System:** Select the file system type that you want to use for the partition. Note that Oracle Linux also includes support for the Btrfs file system type, but to configure a Btrfs file system you must use the *UEK boot image* to load the installer. See [Installing a System With a Btrfs root File System](#) for more information.
- **Reformat:** Set this checkbox format the partition. If the partition isn't a newly created one, you can unset the checkbox to retain existing data.
- **Label:** Optionally label the partition to easily recognize individual partitions in other tooling.
- **Name:** You can name a partition, but standard partitions are named automatically when they're created.

KDUMP

In the event of a system crash, Kdump captures information that helps in finding the cause.

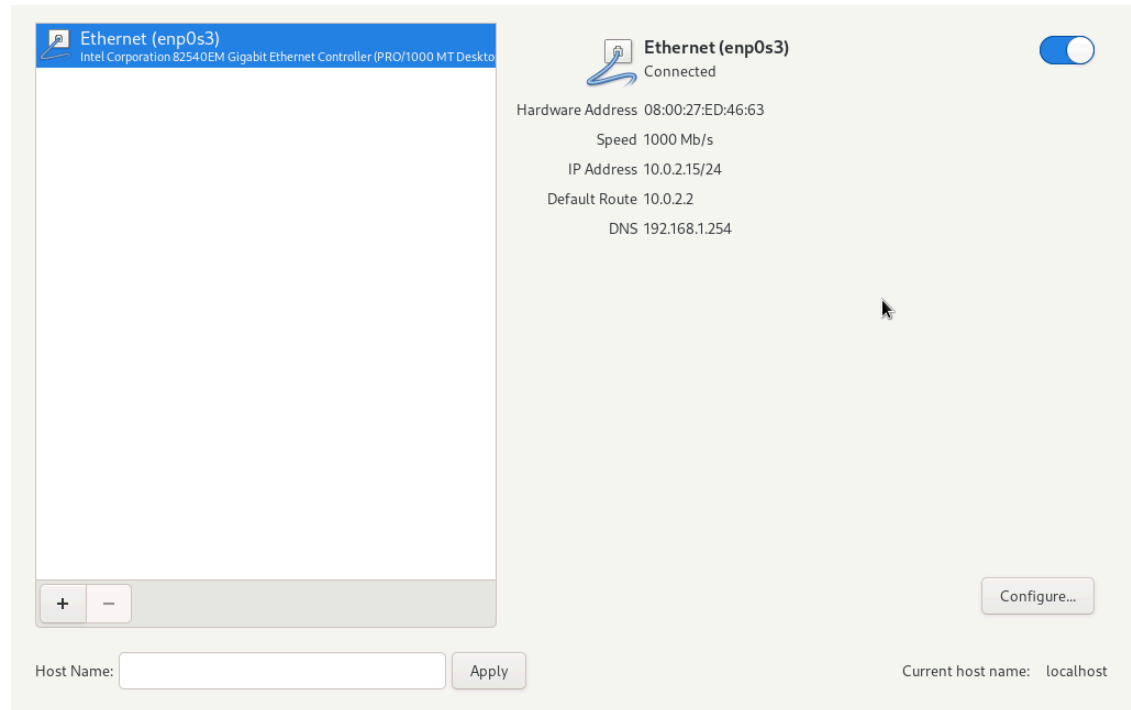
Figure 4-9 Kdump



By default, Kdump is enabled and the amount of memory reserved for Kdump is calculated automatically. Select the **Manual** option to set the amount of reserved memory yourself.

Network and Host Name

Figure 4-10 Network & Host Name



Beginning with Oracle Linux 9, network connections are automatically enabled. By default, network configuration uses DHCP for IPv4 addresses. IPv6 addresses are configured automatically. The default settings are sufficient for the system to provide basic network functionality. However, you can customize the network configuration, for example, by providing a custom host name, including a fully qualified domain name (FQDN). You can further opt to use static addresses instead of using DHCP, configure proxy settings, network bonds, and so on. To do these advanced types of configuration, click **Configure** and go through other configuration screens.

Security Profile

Because security policy isn't required on all systems, use the Security Policy screen only if you need to enforce a specific security policy as defined by your organization or by government regulations.

Figure 4-11 Security Policy

The screenshot shows a window titled "Security Policy". At the top, there is a "Change content" button and a toggle switch for "Apply security policy:" which is currently turned on. Below this, the text "Choose profile below:" is displayed. A large rectangular box contains the following text: "Standard System Security Profile for Oracle Linux 9", "This profile contains rules to ensure standard security baseline of Oracle Linux 9 system. Regardless of your system's workload all of these checks should pass." Below this box is a "Select profile" button. At the bottom of the window, there is a section titled "Changes that were done or need to be done:" followed by an empty rectangular box.

Available policies are listed in the profile window. These security policies follow the recommendations and guidelines that are defined by the Security Content Automation Protocol (SCAP) standard. Select the security profile you want to enforce and ensure that the Apply security policy is switched to On.

If you prefer not apply any security policy, toggle the switch to Off.

For more information on SCAP policies and profiles, see [Oracle Linux 9: Using OpenSCAP for Security Compliance](#).

User Settings

Under User Settings, you configure the following options:

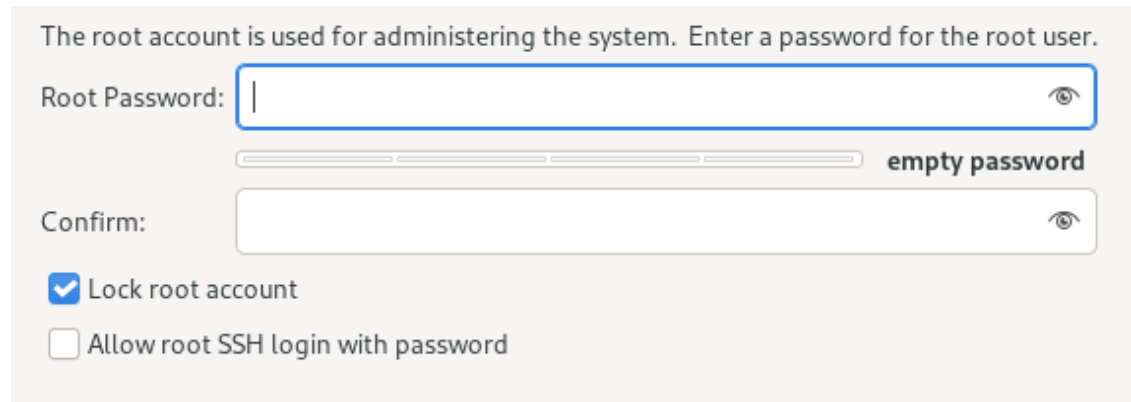
- [Root Password](#)
- [User Creation](#)

After completing root and user configuration, click **Done** to return to the Installation Summary screen.

Root Password

The Root Password window allows you to set the root account password or lock the root account.

Figure 4-12 Setting the root password



The root account is used for administering the system. Enter a password for the root user.

Root Password: 

empty password

Confirm: 

☒ Lock root account

☐ Allow root SSH login with password

By default, the root account is locked as a security feature implementation. If you unlock the root account, you must also set and confirm the root password.

Password strength is monitored and the following rules are applied:

- Password must be eight characters or longer
- Password contains numbers, letters (uppercase and lowercase) and symbols
- Password is case-sensitive

Weak passwords are permitted, but require that you press the **Done** button twice.

A checkbox option lets you configure whether to allow root account access over SSH. By default, this option is disabled.

User Creation

This option lets you configure a user's credentials to enable access to the system. You can optionally configure the user to have administrative privileges. If the root account is disabled, you must configure the user as an administrator.

Figure 4-13 Create User

The screenshot shows a 'Create User' form with the following elements:

- Full name:** A text input field with a blue border.
- User name:** A text input field.
- Make this user administrator:** An unchecked checkbox.
- Require a password to use this account:** A checked checkbox.
- Password:** A text input field with a password strength indicator below it showing 'empty password'.
- Confirm password:** A text input field.
- Advanced...:** A button at the bottom.

The following options are available:

- **Full name:** Enter the full name of the user account. This field is used to show the user account in the graphical login manager.
- **User name:** Enter the username that the user uses to login to the system on the command line or by using SSH.
- **Make this user administrator:** Set the checkbox if the user requires administrative privileges. The user is added to the `wheel` group, which is in the `sudoers` configuration by default.

An administrator user can use the `sudo` command to perform tasks that are only available to `root` by using the user password, instead of the `root` password.

- **Require a password to use this account:** Set this checkbox to require a password for login. Always set this checkbox, particularly if you configure the user account with administrative privileges.
- **Password:** Enter the user password into this field.
- **Confirm password:** Enter a matching user password into this field.

Password strength is monitored and the following rules are applied:

- Password must be eight characters or longer
- Password contains numbers, letters (uppercase and lowercase) and symbols
- Password is case-sensitive

Weak passwords are permitted, but require that you press the **Done** button twice.

After the options are configured, the **Advanced** button becomes available to let you configure other user account options, such as the home directory location for the user, an alternative user ID value, an alternative group ID value, and a list of groups that the account belongs to.

Completing the Installation

From the Installation Summary screen, click **Begin Installation**. This button becomes available only when warning flags on option icons have been cleared.

The installation takes a while. After it finishes, reboot the system as prompted. Then log in to the system by using the credentials you set in User Settings during the installation configuration.

For other configuration options you can set on the system, see [Postinstallation Configuration](#).

Installing in Text Mode

The Oracle Linux installation program can also run in text mode. Text mode is used automatically under certain circumstances, for example, if the system has insufficient memory or the video card isn't supported. You can manually switch to the text mode by specifying `inst.text` as a boot option on the boot command line. See [Configuring the Boot Loader](#) for more information.

! Important:

Text mode provides a limited subset of functionality that's available in the graphical installer. Most notably, it doesn't provide all the custom partitioning options for full control over disk layout. We recommend either using the graphical installer or using Kickstart to automate an installation.

Figure 4-14 Text-Based Installation Menu

```
=====
1) [x] Language settings                2) [x] Time settings
   (English (United Kingdom))          (Europe/London timezone)
3) [!] Installation source             4) [!] Software selection
   (Setting up installation            (Processing...)
   source...)
5) [!] Installation Destination        6) [x] Kdump
   (Processing...)                   (Kdump is enabled)
7) [x] Network configuration           8) [!] Root password
   (Connected: enp1s0)                (Root account is disabled)
9) [!] User creation
   (No user will be created)

Please make a selection from the above ['b' to begin installation, 'q' to quit,
'r' to refresh]:
```

The numbered options on the menu are the equivalent of the screen options in graphic based installations. Each option is preceded by a flag surrounded by brackets:

- [] - Option isn't configured.
- [x] - Option is configured with the default setting.
The setting is displayed between parentheses under the option.
- [!] - Option is configured but needs examination in case you want to specify a different setting.

To configure an option, type the option's number. The screen displays numbered values that are available for that option. Choose the value by typing the value's number. Then type `c` to continue. Continuing either returns you to the main menu screen, or displays other related options for you to configure. Type `c` also to skip screens.

After configuring all the necessary menu options, type `b` to begin installing. At the end of the installation, the system reboots.

Follow all the remaining prompts. At the end of the process, log in to the system and review the license agreement at `/usr/share/oraclelinux-release/EULA`.

For other configuration options you can set on the system, see [Postinstallation Configuration](#).

Postinstallation Configuration

Configuring Login

If you selected System with GUI as the OS profile to install and configured root passwords without creating users, then at the initial login as root, you're prompted to create an initial user. Setting up the first user can be completed in two ways:

- Create a regular user account, which is the default user setup screen. Enter the required user information as prompted, then click **Next** to complete the process.
- Create a domain account. For this option, click **Enterprise Login** on the default screen. A new window opens that prompts for the domain credentials. Use this option if the environment is configured with Active Directory or Identity Management domains for storing all user information. In this manner, the user can use domain credentials to log in to the system's GNOME desktop.

Registering the System

After you install Oracle Linux on a system, you have the option to register the system with the Unbreakable Linux Network (ULN), if you have an account. Registering enables the system to obtain extra packages, updates, and fixes. To register the system, choose one of the following methods:

- Visit <https://linux.oracle.com>. To obtain Oracle Linux updates from ULN, you must have an Oracle Linux support subscription.
- Use the `uln_register` shell command, which opens an interactive process.
- Use the Oracle Linux GNOME desktop menu. From the menu, select **Activities** and then search for `ULN Registration`. Click the **ULN Registration** shortcut icon to start the graphical registration wizard.

For more information about installing packages and managing software on the system, see [Oracle Linux: Managing Software on Oracle Linux](#).

If you have an Oracle Linux Premier Support account, you can opt to use Ksplice, to keep systems secure and highly available by automatically updating them with the latest kernel

security errata and other critical updates. For more information, see [Oracle Linux: Ksplice User's Guide](#).

Updating Software Packages

After installation is complete, we recommend updating all software packages to the latest available versions on the Oracle yum servers, or from ULN, because software packages are often updated for errata bug and security fixes, and package updates might already be available for packages that are provided on the installation media. To update all packages, run:

```
sudo dnf update -y
```

Next Steps

After a basic Oracle Linux installation, you might want to further configure the system for optimization and customization purposes, such as setting system date and time, scheduling tasks, obtaining updates, and so on. For reference, go to the Oracle Linux tutorial page at <https://docs.oracle.com/en/operating-systems/oracle-linux/tutorials.html> which lists tutorials for different administrative tasks. Tutorials are available for tasks that you run at the command line or through the Cockpit web console.

For more detailed information about different features of Oracle Linux, go to the Oracle Linux library at [General Oracle Linux documentation](#).

5

Automating the Installation by Using Kickstart

The Kickstart feature enables you to automate the OS installation. A Kickstart installation uses a configuration file that instructs the installer how it should implement a specific installation. The feature offers the following benefits:

- No user intervention is required during the installation process.
- Greater facility in installing on multiple systems.
- The configuration file is useful for troubleshooting a boot-time problem with an installed system.

You can use Kickstart to install Oracle Linux locally. However, the best use of this feature is in the installation of the OS on multiple systems over the network. In network installations, a Kickstart operation would include the following components:

- Kickstart configuration file
- Configured network that supports network installs such as providing connectivity so that different client systems can access the necessary installation and configuration files specific to those clients.

For an example of the configuration of a network installation, see [Creating a Network Installation Setup](#).

Customizing the Kickstart Configuration File

The Kickstart configuration file consists of installation instructions in the form of options and parameter definitions.

To create a Kickstart configuration file, choose one of the following methods:

- Use the `/root/anaconda-ks.cfg` file of an existing Oracle Linux system.

Every Oracle Linux installation creates a file called `/root/anaconda-ks.cfg`. The file contains configuration definitions based on the system on which Oracle Linux is installed and which can be read by Kickstart. Thus, the file is ready to be used either as is for other installations, or as a template for automating various types of installations for specific target systems.

As a template, the file's contents need to be revised to be applicable to other Oracle Linux installations, such as which OS versions to install on specific clients.

If the system contains the `pykickstart` package, the following useful tools can help you with revising the file's contents:
 - `ksverdiff` identifies removed or deprecated options.
 - `ksvalidator` checks that the syntax in the file is correct.
- Install an Oracle Linux release manually, which generates the `/root/anaconda-ks.cfg` file. Use the file to automatically install the same Oracle Linux release on other clients. You might need to adjust some configuration options in the file depending on the specifications you want to apply to those clients.

For more information, see <https://pykickstart.readthedocs.io/en/latest/>.

The configuration file is divided into parts. Each part contains a group of options as follows:

- [Installation Options](#)
- [Packages to Install](#)
- [Preinstallation Options](#)
- [Postinstallation Options](#)

Installation Options

Installation options define parameters for system storage, keyboard definitions, language setting, network information, and so on. In the following example, the option definition in bold enables PXE to bring up the network interface and is important in network installations.



Note:

To use the Btrfs file system that's supported in UEK, ensure that you're using the UEK installation media as described in [Btrfs installation chapter](#). Then you can use the `autopart` Kickstart option to automatically partition disks using Btrfs, as shown in the following example under `# Partition Information`.

If you're using local mirror repositories, ensure that these are current and that the Kickstart configuration includes the UEK repository required for Btrfs to function. Either use the Oracle Linux yum server to mirror the required repositories; or if you're unable to access the Oracle Linux yum server, you can mirror the repositories included on the full installation ISO, but you must use the UEK installation ISO to boot the installer. In the following example, the system is assumed to have access to the Internet and the publicly available Oracle Linux yum server repositories are used directly.

Alternatively, you also manually partition the disks to use Btrfs as described in the upstream documentation.

```
#platform=x86, AMD64, or Intel EM64T
#version=OL9
# Firewall configuration
firewall --enabled --service=ssh

# Use Oracle Linux yum server repositories as installation source
repo --name="ol9_AppStream" --baseurl="https://yum.oracle.com/repo/
OracleLinux/OL9/appstream/x86_64/"
repo --name="ol9_UEKR7" --baseurl="https://yum.oracle.com/repo/
OracleLinux/OL9/UEKR7/x86_64/"
url --url="https://yum.oracle.com/repo/OracleLinux/OL9/baseos/latest/x86_64"

# Root password
rootpw --iscrypted SHA512_password_hash

# Use graphical install
graphical
firstboot --disable

# Keyboard layouts
```

```
keyboard --vckeymap=us --xlayouts='us'

# System language
lang en_US.UTF-8

# SELinux configuration
selinux --enforcing

# Installation logging level
logging --level=info

# System timezone
timezone America/Los_Angeles

# Network information
network --bootproto=dhcp --device=em1 --onboot=yes
--hostname=hostname

# System bootloader configuration
bootloader --location=mbr --boot-drive=sda

# Non-administrative user
user --name=user --homedir=/home/user --password=SHA512_password_hash --iscrypted

# Partition information

clearpart --all --initlabel --drives=sda
autopart --type=btrfs
```

Packages to Install

Packages to be installed are listed under the group heading `%packages`. The list stops at the `%end line`.

```
%packages
@base
@core
@desktop-debugging
@dial-up
@fonts
@gnome-desktop
@guest-agents
@guest-desktop-agents
@input-methods
@internet-browser
@multimedia
@print-client
@print-server
@x11
mttools
pax
python-dmidecode
oddjob
wodim
```

```
sgpio
genisoimage
device-mapper-persistent-data
abrt-gui
samba-winbind
certmonger
openldap-clients
pam_krb5
krb5-workstation
ldapjdk
slapi-nis
libXmu
perl-DBD-SQLite
perl-Mozilla-LDAP
%end
```

The list includes package groups and individual packages. Names of package groups use the @ prefix, such as @base, to distinguish them from individual packages.

To help decide which packages to include in the file, use the `dnf group list` command on an existing Oracle Linux server. The command displays both the installed package groups and the package groups that are available to install.

To specify more than one package for a match, you can use the wildcard character (*). To exclude a package from the installation, insert the - character as a prefix to the package name.

The `%packages` keyword can take options, such as the following useful ones.

--ignoremissing

Installs the available packages without prompting about missing packages. Without this option, Kickstart would interrupt the installation and prompt you to continue or cancel the installation.

--multilib

Sets the `multilib` policy in `dnf` configuration to `all` so that 32-bit packages can be installed on the system.

Instead of listing packages directly into the configuration file, you can compile these names into a file and store it in an accessible location, such as locally in a Kickstart's ramdisk file system, or on an HTTP server or an NFS share. Then on the configuration file, specify the full path to the list in an `%include` statement, for example:

```
%packages --ignoremissing
%include /tmp/package-list
%end
```

Preinstallation Options

Preinstallation options define the actions that the installer must perform before beginning the installation process. Configuring this group of options is optional. The options are under the `%pre` heading and terminated by the `%end` line.

In the following example, the installer is instructed to run the script `config-partitions` that is stored on an HTTP server. Further, the installer should also download a list of packages from

the web server for use with a `%include /tmp/package-list` statement in the `%packages` section.

```
%pre
%include http://192.168.1.100/scripts/config-partitions
wget -q -O- http://192.168.1.100/scripts/package-list > /tmp/package-list
%end
```

In this example, the `wget` command saves the package list in Kickstart's file system, which exists as a ramdisk in memory.

An included script or file must be accessible at the specified path or URL. If no name service is available to identify hosts, then use IP addresses.

Postinstallation Options

Postinstallation options define any actions to be completed by the installer at the end of the installation. Configuring this group of options is optional.

The options are under the `%post` heading and terminated by the `%end` line.

Beginning with Oracle Linux 9, the `root` account is locked by default. If you want to enable SSH root logins to the system, add the following line to the Kickstart file.

```
%post
echo "PermitRootLogin yes" > /etc/ssh/sshd_config.d/01-permitrootlogin.conf
%end
```

By default, Kickstart runs postinstallation tasks in a `chroot` environment that's based on the root file system of the newly installed system. If you need to access any files that are outside the `chroot` environment, specify the `--nochroot` option to `%post` line. You can then access files in the Kickstart file system with the newly installed system's root file system being mounted at `/mnt/sysimage`.

In the following example, the script `/tmp/post-config` is run at the end of the installation.

```
%post --nochroot
%include /tmp/post-config
%end
```

If you configure the installed system's network interface to obtain its settings using DHCP, you must either use IP addresses instead of domain names or set up a temporary `resolv.conf` file, for example:

```
%post
wget -q -O- http://192.168.1.100/scripts/resolv.conf > /etc/resolv.conf
%include http://instsvr.example.com/scripts/post-config
.
.
.
%end
```

Starting the Kickstart Installation

To start the Kickstart installation, ensure that you have prepared the following:

- Access to the full installation media, which can be on a local CD-ROM drive or USB drive. See the [instructions on media preparation](#).
- Access to the Kickstart configuration file.

Installing Over the Network

1. Boot the system.
2. During the boot sequence, access the system's BIOS.
3. On the BIOS screen, change the system settings to boot from the network.
4. Save the changes and continue with the boot process.
5. When the boot menu appears, select an installation option then press either **E** (UEFI-based systems) or **Tab** (BIOS-based systems) to access the boot prompt.
6. Add the the location of the Kickstart file in the network. The location must specify the network server's protocol and FQDN or IP address. For example, if you're using an NFS server, then you might enter the following line:

```
nfs:options:fdqn/path-to-file
```

Note:

Another way of installing from the network is by using a boot server. In this scenario, the installation automatically begins when you turn on the system. See [Creating a Network Installation Setup](#).

7. Save the boot configuration revisions and continue the boot process.
At the end of the boot process, the installation proceeds immediately.

Installing Locally

1. Boot the system from the local boot media.
2. When the boot menu appears, select an installation option then press either **E** (UEFI-based systems) or **Tab** (BIOS-based systems) to access the boot prompt.
3. Add the appropriate option that identifies the location of the Kickstart file, for example:

```
inst.ks=cdrom:/dev/sbd1/tmp/ks.cfg
```

For more information about boot options, see [Customizing Boot Loader Actions](#).

4. Save the boot configuration revisions and continue the boot process.
At the end of the boot process, the installation proceeds immediately.

6

Using a Driver Update Disk During Installation

To use the DUD during an installation, choose from one of the following methods depending on where the DUD image is stored.

- Use a DUD image in an attachable media (USB)
 1. Boot the system from the standard Oracle Linux installation media.
 2. While the system is booting, but before the installer starts, attach the media that contains the DUD image.

As an example, attach the media when the system displays the GRUB boot prompt.

The installer recognizes the DUD and uses it. The installer might also prompt you to select the drivers to install.

Note:

The kernel used by the installer might not include support for USB 3.0. When using the DUD on USB media, ensure that you use a USB 2.0 compatible port when connecting the USB media to the system.

- Use a DUD image on the network
 1. While the system is booting, press the key that enables you to edit the boot options.
On BIOS-based systems, press Tab. On UEFI-based systems, press E.
 2. Add the following line to the boot options:

```
inst.dd=network-location
```

The network location can be a URL, such as `http://www.example.com/dd.iso`, or the full path of the NFS share directory.

If you're using DUD in a Kickstart installation, indicate the DUD location in the Kickstart file. The specific entry to add depends on the location of the DUD image.

- DUD image is in an attached block device:

```
driverdisk /dev/sdb1
```

- DUD image is on the network location:

```
driverdisk --source=network-location
```

The network location can be a URL, such as `http://www.example.com/dd.iso`, or the full path of the NFS share directory.

7

Installing a System With a Btrfs root File System

Oracle Linux uses XFS as the default file system, but it also supports other file systems such as ext2, ext4, and so on.

The Btrfs file system is supported in Oracle Linux, but only on UEK. Thus, to install Oracle Linux with Btrfs, you must use the UEK ISO image.

For more information about file systems in Oracle Linux, see [Oracle Linux 9: Managing Local File Systems](#).

Like any installation of a standard Oracle Linux release, installing the OS with a Btrfs file system can also be completed in two modes:

- [Using the GUI Installer](#)
- [Using the Text Installer](#)

You can also install a system with a Btrfs root file system using Kickstart automation as described in [Automating the Installation by Using Kickstart](#). If you choose to perform an automated installation and you create your own network installation setup, as described in [Creating a Network Installation Setup](#), ensure that you configure the network boot to use the installation kernel and ram-disk image provided on the UEK ISO image and that you have mirrored all required repositories, or you might get an error stating that Btrfs isn't supported for the partitioning type.

Using the GUI Installer

The following steps are limited to what's relevant to a Btrfs installation. For the complete procedure, see [Installing in Graphics Mode](#).

1. Boot the system from the installation media using the Unbreakable Enterprise Kernel boot ISO file.
2. On the Installation Summary screen, select **Installation Destination**, then select the local disks and add any **Specialized & Network** disks that you want to use for the installation.
3. Under Storage Configuration, select **Custom** and then click **Done**.

The Manual Partitioning screen is displayed. Custom partitioning can be either automatic, where mount points are automatically created, or manual, where you individually create and configure mount points.

Automatic Partitioning

- a. From the list of partitioning schemes, select **Btrfs**.
- b. Click **Click here to create them automatically**.

The following mount points are created:

- /home
- /boot

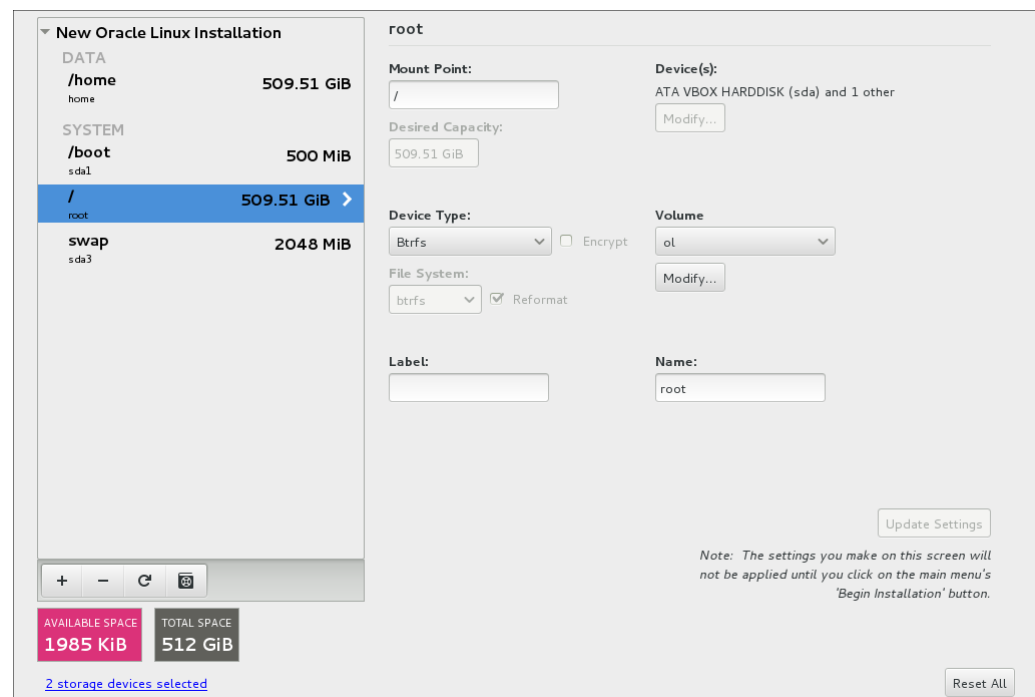
- / (root)
 - swap
- c. Optionally, select each mount point and verify in the corresponding information that the mount point's device type is **Btrfs**.

 Note:

The swap partition can't be configured as a Btrfs file system.

The mount point information might resemble the following example:

Figure 7-1 Mount Points



- d. Optionally for each mount point, configure the respective volumes.
- e. Click **Done**, then on the partition summary that's displayed, click **Accept Changes**.

Manual Partitioning

- a. From the list of partitioning schemes, select **Btrfs**.
- b. Click + to create the mount point.

The **Add A New Mount Point** window opens.

- c. Select the mount point and enter its size as prompted, then click **Add mount point**.

As a minimum, create mount points for /, /boot, /home, and swap.

- d. Select each mount point you created and verify that their corresponding information is correct.

**Note:**

The `swap` partition can't be configured as a Btrfs file system.

- e. Optionally, change the volume of each mount point, as needed.
- f. Click **Done** and on the partition summary that's displayed, click **Accept Changes**.
4. Complete the configuration of any other sections in the Installation Summary screen.
5. Click **Begin Installation**.

Using the Text Installer

The following steps are limited to what's relevant to a Btrfs installation. For the complete procedure, see [Installing in Text Mode](#).

1. From the text based installation menu, type `5` to configure Installation Destination.
 - a. **Installation Destination:** If the preferred disk is already selected, type `c` to continue. Otherwise, type the number corresponding to the disk you want to use, press Enter, then type `c` to continue.
 - b. **Partitioning Options:** If the preferred option is already selected, type `c` to continue. Otherwise, type the number corresponding to the preferred option, press Enter, then type `c` to continue.
 - c. **Partition Scheme Options:** Type `2` to use Btrfs, press Enter, then type `c` to return to the main menu.
2. Configure all the remaining settings that are flagged with the warning symbol (!).
3. After you have completed configuring all the installation options, type `b` to begin the installation.

Verifying the Btrfs Installation

When the entire installation process has completed and you can log in to the system, you can use different tools to verify the file system configuration. The following example shows the output of the `sudo df -Thk` command:

```
sudo df -Thk
```

```
Filesystem Type 1K-blocks    Used Available Use% Mounted on
...
/dev/sda3 btrfs    209606660 67708388 141898272  33% /
/dev/sda2 btrfs    4128768   756460  3372308   19% /boot
/dev/sda1 vfat     1033224   12556   1020668    2% /boot/efi
/dev/sda4 btrfs    267254276 95711592 171542684  36% /home
...
```

Optionally, you can complete other tasks as described in [Postinstallation Configuration](#).

8

Troubleshooting Oracle Linux Installations

This chapter describes some options for troubleshooting Oracle Linux installations. Also check the Oracle Linux release notes of the specific Oracle Linux release you are installing in [Oracle Linux](#) for any known issues.

For known hardware issues, see the release notes for your UEK release in the [Unbreakable Enterprise Kernel documentation](#).

Troubleshooting Menu Options

The boot menu on the Oracle Linux installation media contains a Troubleshooting option with the following options:

Install the Oracle Linux release in basic graphics mode

Use this option if the screen goes blank or appears distorted when you try to install Oracle Linux in graphics mode.

Rescue a Oracle Linux system

Use this option to boot an installed system in a mode that enables you to edit partitions or configuration files to fix various boot problems.

Run a memory test

Use this option to run the `memtest86` utility, which verifies the integrity of the system's RAM.

Boot from local drive

Use this option to boot an installed system from the hard disk.

Debug and Log Information

During an installation, you can press `Ctrl+Alt+F1` to display the virtual console for the installation program. This console contains messages and debugging information output for the installation program. Additional virtual consoles are available to display log information from different sources, as described in the following section. Press `Ctrl+b`, and then press either `n` (for next) or `b` (for back) to switch between the virtual consoles. To return to the graphical installation program, press `Ctrl+Alt+F6`.

During an installation, several log files are generated, which capture messages from the following sources:

Anaconda program

The `/tmp/anaconda.log` file contains Anaconda logs relating to the installation.

During the installation you can access the messages stored in this log by pressing `Ctrl+Alt+F1` to display the virtual console, then press `Ctrl+b`, and then press `3`.

If the installation succeeds, the log file is copied to `/var/log/anaconda/anaconda.log`.

Hardware detection and configuration

The `/tmp/syslog` file contains messages relating to the system hardware.

If the installation succeeds, the log file is copied to `/var/log/anaconda/syslog`.

Kickstart

The `/tmp/ks-name.log` file contains logs from Kickstart installations.

If the installation succeeds, the log file is copied to `/var/log/anaconda/ks-script-name.log`.

Network detection and configuration

The `/tmp/ifcfg.log` file contains logs relating to network configuration.

If the installation succeeds, the log file is copied to `/var/log/anaconda/ifcfg.log`.

Other programs

The `/tmp/program.log` file contains logs from all other programs used during the installation.

During the installation you can view the messages stored in this log by pressing Ctrl+Alt+F1 to display the virtual console, then press Ctrl+b, and then press 5.

If the installation succeeds, the log file is copied to `/var/log/anaconda/program.log`.

Package installation

The `/tmp/package.log` file contains package installation messages output by the `dnf` and `rpm` commands.

If the installation succeeds, the log file is copied to `/var/log/anaconda/package.log`.

Storage detection and configuration

The `/tmp/storage.log` file contains logs from the storage modules.

During the installation you can view the messages stored in this log by pressing Ctrl+Alt+F1 to display the virtual console, then press Ctrl+b, and then press 4.

If the installation succeeds, the log file is copied to `/var/log/anaconda/storage.log`.

If the installation fails, the messages from these log files are combined into a single log file at `/tmp/anaconda-tb-name`.

To access a shell prompt as the `root` user during the installation, press Ctrl+Alt+F1 to display the virtual console. Then, press Ctrl+b, followed by pressing the number 2. You can use the shell prompt to access the log files and to copy them to a local storage device such as a USB device. Or, you can copy the log files to a network location by using the `scp` command.

A

Creating a Network Installation Setup

Network installations are useful and advantageous especially in scenarios where you have to install the OS on multiple systems.

If you use a boot ISO or the Preboot eXecution Environment (PXE) to install the OS on systems, you can set up a network installation configuration that consists of the following components:

- A network installation server that provides the IP and network configuration, the PXE configuration files, kernel and boot images, and kernel boot directives. In this example, `dnsmasq` is used to provide these services.
- A network accessible file system server over a protocol such as NFS or HTTP, where packages, the Kickstart file, and other required configuration files might be stored. In this document, NFS is used as the file system. Note that if your systems have direct access to the Internet, you can configure your Kickstart installation to use the BaseOS and AppStream repositories directly available on the Oracle Linux yum server for the package installation. You can also create a yum mirror of these repositories as described in [Oracle Linux: Managing Software on Oracle Linux](#) so that systems that do not have direct access to the Internet have access to a complete set of packages. However, you would still need to provide a mechanism to serve the Kickstart file up to the system when it boots.

The two components can be on separate systems. Also, they are not required to run the latest Oracle Linux version. A previous release would suffice. For convenience, the scenario in this example assumes that the two components are hosted in one system. It also assumes that Kickstart installation is used.

Setting Up the Network Server

Preparing a server for a network installation consists of the following general tasks:

- [How to Configure NFS](#)
- [How to Configure dnsmasq](#)

How to Configure NFS

If you have an existing NFS server, you can use this service to host the extracted contents of an ISO and any Kickstart configuration files. Ensure that the exported share directories are accessible to the IP ranges specified for the PXE boot hosts.

If you don't have an available NFS server, use the following procedure to install and configure the service to enable network installation.



Note:

These NFS configuration steps are limited only to what's relevant to a network installation.

1. Install the `nfs-utils` package:

```
sudo dnf install nfs-utils -y
```

2. If you're running a firewall service, add the `nfs` service to the firewall rules.

```
sudo firewall-cmd --add-service nfs --permanent
```

If you're using an alternative firewall service or you have an external firewall device, ensure that you configure rules that grant PXE boot hosts access to the NFS service on this system.

3. Create a directory to store the ISO image, for example:

```
sudo mkdir /var/nfs-exports/ISOs
```

4. Export the NFS share directory.

```
sudo exportfs -i -o ro [subnet]:/var/nfs-exports/ISOs
```

This syntax grants world access to the NFS share with read-only permissions. Adding *subnet*, for example `192.0.2.0/24:/var/nfs-exports/ISOs`, limits access only to the subnet's clients.

Alternatively, you can add an entry for exporting the share directory in the `/etc/exports` file, for example:

```
/var/nfs-exports/ISOs 192.0.2.0/24(ro)
```

Then, reload the `/etc/exports` to implement the entry:

```
sudo exportfs -ra
```

For more information, see the `exportfs(8)`, `exports(5)`, and `showmount(8)` manual pages.

5. Enable and start the `nfs-server` service:

```
sudo systemctl enable --now nfs-server
```

6. Extract the downloaded ISO image to a subdirectory of the NFS share directory:

```
sudo cp -a T path-to-download-image /var/nfs-exports/ISOs/ol9
```

7. If using Kickstart, put the Kickstart files in a subdirectory of the NFS share directory also, such as `/var/nfs-exports/ISOs/ksfiles`.

8. (Optional) From a different system, verify that the NFS share directory is accessible, for example:

```
sudo mount -t nfs NFS-server-ip:/var/nfs-exports/ISOs /mnt
```

How to Configure dnsmasq

The dnsmasq router advertisement server is designed to act as a DNS forwarder, DHCP server, and TFTP server. Dnsmasq is applicable in most network installation scenarios and is therefore a convenient alternative to configuring separate DHCP and TFTP services.

For more information about dnsmasq, see the `dnsmasq(8)` manual page, the `/usr/share/doc/dnsmasq-version` file, and <https://thekelleys.org.uk/dnsmasq/doc.html>.

1. Install the `dnsmasq` package.

```
sudo dnf install dnsmasq -y
```

2. Configure parameters in the `/etc/dnsmasq.conf` file.

- At a minimum, you must have the **`enable-tftp`** entry and a defined TFTP server directory for **`tftp-root`**. See the entries in bold in the following example:

```
interface=em1
dhcp-range=10.0.0.101,10.0.0.200,6h
dhcp-host=80:00:27:c6:a1:16,10.0.0.253,svr1,infinite
dhcp-boot=pxelinux/pxelinux.0
dhcp-match=set:efi-x86_64,option:client-arch,8
dhcp-boot=tag:efi-x86_64,shim.efi
enable-tftp
tftp-root=/var/lib/tftpboot
```

 Note:

If SELinux is enabled in enforcing mode on the system and you configured a TFTP server directory other than `/var/lib/tftpboot`, install the `policycoreutils-python` and `policycoreutils` packages to enable you to run the following commands:

```
sudo /usr/sbin/semanage fcontext -a -t tftpd_dir_t "/var/
tftpboot(/.*)?"
sudo /sbin/restorecon -R -v /var/tftpboot
```

These commands define the default file type of the TFTP server directory hierarchy as `tftpd_dir_t` and apply the file type to the entire directory hierarchy.

The following list describes the other parameters in the `/etc/dnsmasq.conf` file:

interface

Specifies the interface to be monitored for incoming client requests.

dhcp-range

Identifies a range of available IP addresses. The `6h` setting in the example above specifies a six-hour lease of the addresses.

To configure static addresses with infinite leases, instead of a pool, specify a static network address and use the `static` and `infinite` keywords, for example:

```
dhcp-range=10.0.0.253,static,infinite
```

dhcp-host

Specifies a reserved IP address for a client system. The system is identified by its name and MAC address.

dhcp-boot

Specifies the location of the boot loader file for clients, such as `pxelinux/pxelinux.0` for BIOS-based clients. For UEFI-based clients, include the `tag:efi-x86_64` keyword in the setting before specifying the boot loader, for example:

```
dhcp-boot=tag:efi-x86_64,shim.efi
```

You must create separate entries for BIOS-based and UEFI-based clients.

- Uncomment the **tftp-no-blocksize** line in the file as shown:

```
# This option stops dnsmasq from negotiating a larger blocksize for TFTP
# transfers. It will slow things down, but may rescue some broken TFTP
# clients.
tftp-no-blocksize
```

3. (Optional) To use dnsmasq as a caching-only name server, do the following:
 - a. In the `/etc/resolv.conf` file, configure a name server entry for 127.0.0.1 that precedes other name server entries, for example:

```
nameserver 127.0.0.1
nameserver 10.0.0.8
nameserver 10.0.0.4
```

The dnsmasq server ignores the 127.0.0.1 entry and forwards DNS queries to the other listed name servers.

- b. Configure the firewall to accept DNS requests:

```
sudo firewall-cmd --add-service=dns --permanent
```

4. Enable and start the dnsmasq service.

```
sudo systemctl enable --now dnsmasq
```

Configuring PXE Boot Loading

The steps to configure a PXE boot loader depends on the type of system that is used for the boot server.

- [How to Configure for BIOS-based PXE Clients](#)
- [How to Configure for UEFI-based PXE Clients](#)

! Important:

You must use the correct kernel and ram-disk image for the installation that you intend to support. For example, if you intend to install a system with a Btrfs root partition, you must use the UEK boot ISO to obtain the correct kernel and image for the system to be able to install with Btrfs support.

How to Configure for BIOS-based PXE Clients

1. Install the `syslinux` package that contains the `pxelinux.0` boot loader.

```
sudo dnf install syslinux -y
```

2. Create the `pxelinux/pxelinux.cfg` directory under the TFTP server directory.

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

3. Copy the boot loader file to the `pxelinux` subdirectory.

```
sudo cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/pxelinux
```

4. From the NFS share directory, copy the installation kernel (`vmlinuz`), and the ram-disk image file (`initrd.img`) to the `pxelinux` subdirectory.

```
sudo cp /var/nfs-exports/ISOs/ol9/vmlinuz /var/lib/tftpboot/pxelinux/
vmlinuz
sudo cp /var/nfs-exports/ISOs/ol9/initrd.img /var/lib/tftpboot/pxelinux/
initrd.img
```

5. Copy the modules for displaying the boot loader as a menu to the boot loader directory.

```
sudo cp /usr/share/syslinux/*.c32 /var/lib/tftpboot/pxelinux
```

6. In the `pxelinux.cfg` subdirectory, create and add entries to a boot configuration file.

You can assign any name to the file, such as `pxe.conf`. The following example shows typical entries in the file:

```
DEFAULT menu.c32
TIMEOUT 400
ONTIMEOUT BootLocal
PROMPT 0
MENU INCLUDE pxelinux.cfg/pxe.conf
NOESCAPE 1
LABEL BootLocal
    localboot 0
    TEXT HELP
    Boot to local hard disk
    ENDTEXT

LABEL OL9
    MENU LABEL OL9
    kernel vmlinuz
```

```
append initrd=initrd.img inst.repo=/var/nfs-exports/ISOs/ol9/ \
inst.ks.sendmac inst.ks=/var/nfs-exports/ISOs/ksfiles/kstart-file
TEXT HELP
Install Oracle Linux 9
ENDTEXT
```

Based on these entries, the boot loader would automatically attempt to boot from the local drive if no user intervention occurs during the `TIMEOUT` period. If no OS is installed, then the boot loader would boot from the network and initiate the installation process.

This list explains some directives used in the configuration file:

- `DEFAULT` identifies the module you want to use for displaying the boot loader menu.
For a basic text display, the `menu.c32` module suffices. However, if you add directives for a graphical display, such as images and colors, then specify the `vesamenu.c32` module instead.
- `TIMEOUT` specifies the period in `timeout/10` seconds before the boot loader boots the client according to the subsequent directives. The next directive (`ontimeout`) specifies the action when the wait period expires.
- `PROMPT` determines whether the `boot: prompt` is displayed by default. If `PROMPT` is set to 1, the `boot: prompt` is displayed. If `PROMPT` is set to 0, the `boot: prompt` isn't displayed unless the user presses the `Shift` or `Alt` key at the console.
- `MENU INCLUDE` identifies the boot configuration file you created.
- `kernel` defines the name of the kernel executable.
- `append` defines any parameters to append when loading the kernel, such as the name of the ram-disk image and the location of a file. Note that the `inst.repo` variable can be set to point to the BaseOS repository on the Oracle Linux yum server if the system has access to the Internet. For example, `inst.repo` can be set to `https://yum.oracle.com/repo/OracleLinux/OL9/baseos/latest/x86_64` for an `x86_64` platform system.

! Important:

The kernel and ram-disk image file paths are assumed to be relative to the subdirectory that contains the boot loader, such as `pxelinux`. If you place the `vmlinuz` and `initrd.img` files in a subdirectory such as `pxelinux/OL9`, ensure you have the correct relative paths.

How to Configure for UEFI-based PXE Clients

1. Download the `grub2-efi` and `shim` packages:

```
mkdir /tmp/boot_rpms
sudo dnf download shim grub2-efi --downloadaddir=/tmp/boot_rpms
```

2. Go to the package location and extract their contents:

```
cd /tmp/boot_rpms
sudo rpm2cpio grub2-efi-version.rpm | cpio -idmv
sudo rpm2cpio shim-version.rpm | cpio -idmv
```

3. Create a subdirectory, for example, `efi`, under the TFTP server directory:

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

4. Copy the boot loader and other related files to the `efi` subdirectory:

```
sudo cp /tmp/boot_rpms/boot/efi/EFI/redhat/grubx64.efi /var/lib/
tftpboot/efi
sudo cp /tmp/boot_rpms/boot/efi/EFI/redhat/shim*.efi /var/lib/tftpboot/efi
sudo cp /tmp/boot_rpms/boot/efi/EFI/redhat/MokManager.efi /var/lib/
tftpboot/efi
```

Note:

The `shim.efi` and `MokManager.efi` files are needed to support Secure Boot on clients. The `MokManager.efi` provides utilities for managing the keys used to sign EFI binaries. As a passthrough boot loader, you would then specify `shim.efi` when setting `dhcp-boot` for UEFI-based clients in the `/etc/dnsmasq.conf` file. See [How to Configure dnsmasq](#).

5. From the NFS share directory, copy the installation kernel (`vmlinuz`) and the ram-disk image file (`initrd.img`) to the `efi` subdirectory.

```
sudo cp /var/nfs-exports/ISOs/ol9/vmlinuz -O /var/lib/tftpboot/efi/vmlinuz
sudo cp /var/nfs-exports/ISOs/ol9/initrd.img /var/lib/tftpboot/efi/
initrd.img
```

6. In the `efi` subdirectory, create and then add entries to the `grub.cfg` boot loader configuration file.

The configuration file includes the options for booting from the network and for booting locally. The client boots from the network to begin the installation process. Then after the client reboots at the end of the installation, the system boots from the local drive.

The following example shows typical entries in the file:

```
set default 0
set timeout=10

menuentry 'ol9 localboot' {
  echo "Booting from local disk"
  set root=(hd0,gpt1)
  chainloader efi/shim.efi
  boot
}

menuentry 'ol9' {
```

```
echo "Loading vmlinuz"
linuxefi vmlinuz inst.repo=/var/nfs-exports/ISOs/ol9/ inst.ks.sendmac \
inst.ks=/var/nfs-exports/ISOs/ksfiles/kstart-file
echo "Loading /initrd.img"
initrdefi initrd.img
echo "Booting installation kernel"
}
```

⚠ Caution:

Boot loader configuration isn't uniform across UEFI-based systems. Because of variables such as differing disk layout, a specific boot loader setup doesn't universally apply to all systems. The previous example illustrates only one way of configuring the boot loader. You must create boot loader configurations appropriate to the systems that you're setting up.

- `linuxefi` defines the name of the kernel executable and any parameters to append when loading the kernel, such as the location of the installation packages and the location of a file.
- `initrdefi` defines the name of the ram-disk image.

! Important:

The kernel and ram-disk image file paths are assumed to be relative to the subdirectory that contains the boot loader, such as `efi`. If you place the `vmlinuz` and `initrd.img` files in a subdirectory, such as `efi/OL9`, ensure you have the correct relative paths.

Supporting Different Kinds of Clients

To support different types of clients, you can create a configuration file with the name based on the following:

- Client's UUID, for example, `a8943708-c6f6-51b9-611e-74e6ac80b93d`

✎ Note:

A UUID-based file name is valid only for BIOS-based clients. Don't apply it for UEFI-based clients.

- Client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and using dashes to separate each byte value instead of colons, for example, `01-80-00-27-c6-a1-16`.

Use lowercase characters when typing the MAC address.

- Client's IP address expressed in hexadecimal without any leading `0x`, for example, `0A000FD` represents the IP address `10.0.0.253`.

To reduce the number of configuration files, you can group clients by IP address range, for example, `0A0000E` represents the IP address range 10.0.0.224 through 10.0.0.239.

If you're serving both types of clients, ensure that the file names are distinct from each other. Where the configuration file for BIOS-based clients is `A000FC`, for example, the equivalent file for UEFI-based clients can be `grub.cfg-A000FC`, and so on.

Place the configuration files in their respective boot loader subdirectories such as `pxelinux/pxelinux.cfg` for BIOS-based clients or `efi` subdirectory for UEFI-based clients.

The boot loader searches for a configuration file in the following order, until a matching file name is found:

- *UUID* (for example, `a8943708-c6f6-51b9-611e-74e6ac80b93d`)
- *01-MAC_address* (for example, `[grub2-cfg-]01-80-00-27-c6-a1-16`)
- Full 32 bits of the IP address (for example, `[grub.cfg-]0A0000FD`)
- Most significant 28 bits of the IP address (for example, `[grub.cfg-]0A0000F`)
- Most significant 24 bits of the IP address (for example, `[grub.cfg-]0A0000`)
- Most significant 20 bits of the IP address (for example, `[grub.cfg-]0A000`)
- Most significant 16 bits of the IP address (for example, `[grub.cfg-]0A00`)
- Most significant 12 bits of the IP address (for example, `[grub.cfg-]0A0`)
- Most significant 8 bits of the IP address (for example, `[grub.cfg-]0A`)
- Most significant 4 bits of the IP address (for example, `[grub.cfg-]0`)
- Default configuration file (either `default` for BIOS-based clients or `grub.cfg` for UEFI-based clients).

If several configuration files for a client type have identical content, you can use the `ln` command to link the files to a primary copy, for example:

```
sudo ln primary-ol-verson [grub.cfg-]0A0000FC
```

```
sudo ln primary-ol-verson [grub.cfg-]0A0000FD
```

```
sudo ln primary-ol-verson [grub.cfg-]0A0000FE
```

For more information about `pxelinux`, see <https://wiki.syslinux.org/wiki/index.php?title=PXELINUX>.

For more information about GRUB 2, run the `info grub` command to access the GRUB 2 manual.