Red Hat Enterprise Linux 9

Essentials

Red Hat Enterprise Linux 9 Essentials

ISBN-13: 978-1-951442-66-8

The content of this book is provided for informational purposes only. Neither the publisher nor the author offers any warranties or representation, express or implied, with regard to the accuracy of information contained in this book, nor do they accept any liability for any loss or damage arising from any errors or omissions.

This book contains trademarked terms that are used solely for editorial purposes and to the benefit of the respective trademark owner. The terms used within this book are not intended as infringement of any trademarks.

Rev: 1.0

Table of Contents

1. Introduction

Arguably one of the most highly regarded and widely used enterprise level operating systems available today is the Red Hat Enterprise Linux (typically shortened to RHEL and pronounced *rell*) distribution. Not only is it considered to be among the most stable and reliable operating systems, it is also backed by the considerable resources and technical skills of Red Hat, Inc. Red Hat Enterprise Linux 9 Essentials is designed to provide detailed information on the installation, use and administration of the Red Hat Enterprise Linux 9 distribution. For beginners, the book covers topics such as operating system installation, the basics of the GNOME desktop environment, configuring email and web servers and installing packages and system updates using App Streams. Additional installation topics such as dual booting with Microsoft Windows are also covered, together with all important security topics such as configuring a firewall and user and group administration.

For the experienced user, topics such as remote desktop access, the Cockpit web interface, logical volume management (LVM), disk partitioning, swap management, KVM virtualization, Secure Shell (SSH), Linux Containers and file sharing using both Samba and NFS are covered in detail to provide a thorough overview of this enterprise class operating system.

1.1 Superuser Conventions

RHEL 9, in common with Linux in general, has two types of user account, one being a standard user account with restricted access to many of the administrative files and features of the operating system, and the other a superuser (*root*) account with elevated privileges. Typically, a user can gain root access either by logging in as the root user, or using the *su* - command and entering the root password. In the following example, a user is gaining root access via the *su* - command:

```
[neil@demo-server ~]$ su -
Password:
[root@demo-server ~]#
```

Note that the command prompt for a regular user ends with a $ sign while the root user has a # character. When working with the command-line, this is a useful indication as to whether or not you are currently issuing commands as the root user.

If the *su* - command fails, the root account on the system has most likely been disabled for security reasons. In this case, the *sudo* command can be used instead as outlined below.

Using *sudo*, a single command requiring root privileges may be executed by a non-root user. Consider the following attempt to update the operating system with the latest patches and packages:

```
[neil@demo-server ~]$ dnf update
Not root, Subscription Management repositories not updated
```

```
Error: This command has to be run under the root user.
```

Optionally, user accounts may be configured so that they have access to root level privileges. Instead of using the *su* - command to first gain root access, user accounts with administration privileges are able to run otherwise restricted commands using *sudo*:

```
[neil@demo-server]$ sudo dnf update

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for neil:
Updating Subscription Management repositories.
.
.
```

To perform multiple commands without repeatedly using the sudo command, a command prompt with persistent super-user privileges may be accessed as follows:

```
[neil@demo-server]$ sudo su -
[neil@demo-server]#
```

The reason for raising this issue so early in the book is that many of the command-line examples outlined in this book will require root privileges. Rather than repetitively preface every command-line example with directions to run the command as root, the command prompt at the start of the line will be used to indicate whether or not the command needs to be performed as root. If the command can be run as a regular user, the command will be prefixed with a $ command prompt as follows:

```
$ date
```

If, on the other hand, the command requires root privileges, the command will be preceded by a # command prompt:

```
# dnf install openssh
```

1.2 Opening a Terminal Window

If you are using the GNOME desktop and need to access a command prompt you will need to open a Terminal window. To do this, either press the keyboard Windows key or click on the Activities button in the top left-hand corner of the screen, then select the Terminal from the dash as shown in Figure 1-1:

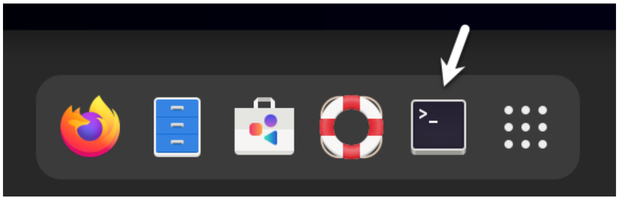


Figure 1-1

1.3 Editing Files

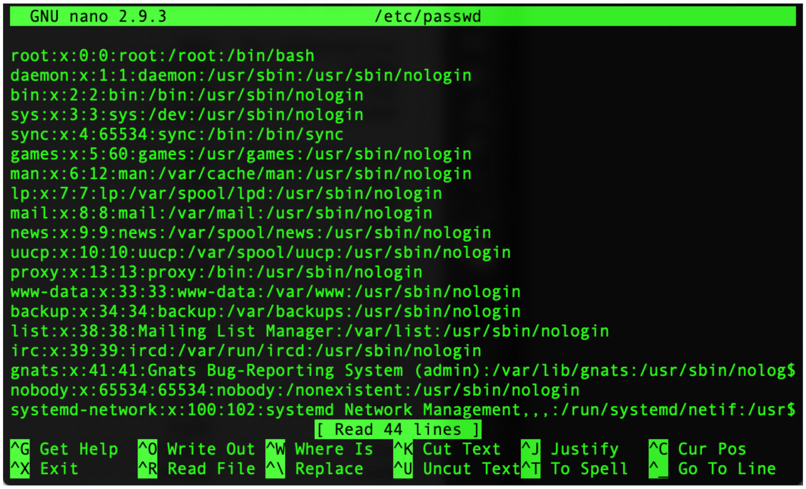
Configuring a Linux system typically involves editing files. For those new to Linux it can be unclear which editor to use. If you are running a terminal session and do not already have a preferred editor we recommend using the *nano* editor. To launch *nano* in a terminal window simply enter the following command:

```
# nano <file>
```

Where <file> is replaced by the path to the file you wish to edit. For example:

```
# nano /etc/passwd
```

Once loaded, *nano* will appear as illustrated in Figure 1-2:



```
  GNU nano 2.9.3                    /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nolog$
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr$
                              [ Read 44 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

Figure 1-2

To create a new file simply run *nano* as follows:

```
# nano
```

When you have finished editing the file, type Ctrl-S to save the file followed by Ctrl-X to exit. To open an existing file, use the Ctrl-R keyboard shortcut.

If you prefer to use a graphical editor within the GNOME desktop environment *gedit* is a useful starting point for basic editing tasks. To launch *gedit* from the desktop press Alt-F2 to display the Enter a Command window as shown in Figure 1-3:

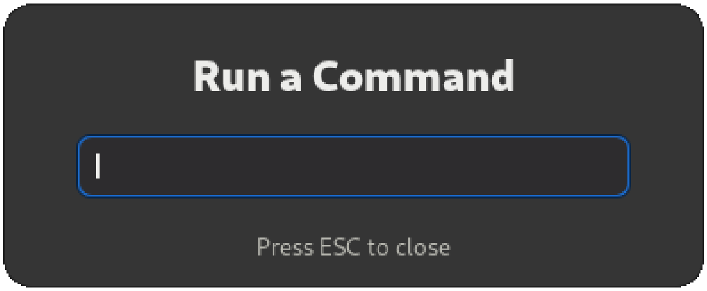
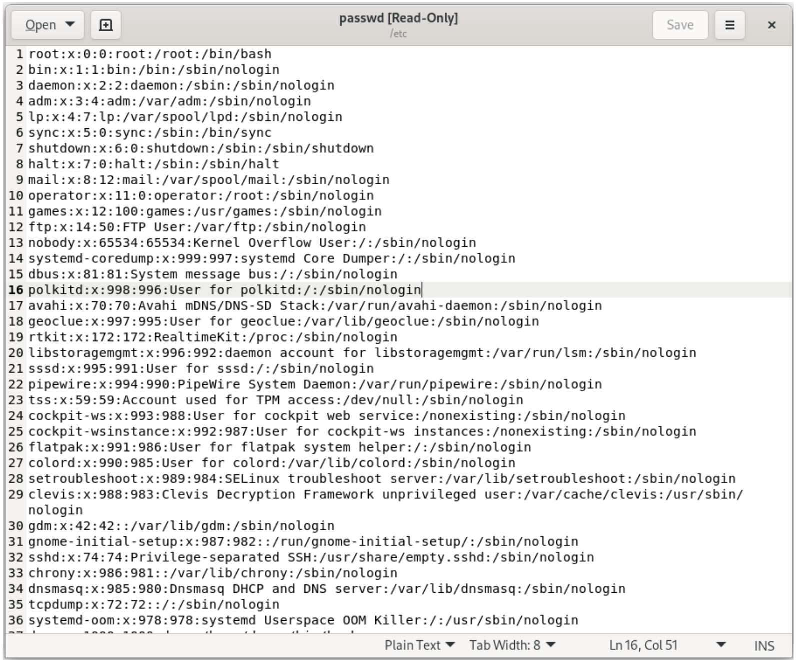


Figure 1-3

Enter *gedit* into the text field and press the Enter key. After a short delay, gedit will load ready to open, create and edit files:



```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
geoclue:x:997:995:User for geoclue:/var/lib/geoclue:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
libstoragemgmt:x:996:992:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
sssd:x:995:991:User for sssd:/:/sbin/nologin
pipewire:x:994:990:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
cockpit-ws:x:993:988:User for cockpit web service:/nonexisting:/sbin/nologin
cockpit-wsinstance:x:992:987:User for cockpit-ws instances:/nonexisting:/sbin/nologin
flatpak:x:991:986:User for flatpak system helper:/:/sbin/nologin
colord:x:990:985:User for colord:/var/lib/colord:/sbin/nologin
setroubleshoot:x:989:984:SELinux troubleshoot server:/var/lib/setroubleshoot:/sbin/nologin
clevis:x:988:983:Clevis Decryption Framework unprivileged user:/var/cache/clevis:/usr/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:987:982::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:986:981::/var/lib/chrony:/sbin/nologin
dnsmasq:x:985:980:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
systemd-oom:x:978:978:systemd Userspace OOM Killer:/:/usr/sbin/nologin
```

Figure 1-4

Alternatively, launch gedit from a terminal window either with or without the path to the file to open:

```
# gedit
```

```
# gedit /etc/passwd
```

1.4 Feedback

We want you to be satisfied with your purchase of this book. If you find any errors in the book, or have any comments, questions or concerns please contact us at *feedback@ebookfrenzy.com*.

1.5 Errata

While we make every effort to ensure the accuracy of the content of this book, it is inevitable that a book covering a subject area of this size and complexity may include some errors and oversights. Any known issues with the book will be outlined, together with solutions, at the following URL:

https://www.ebookfrenzy.com/errata/rhel9.html

In the event that you find an error not listed in the errata, please let us know by emailing our support team at *feedback@ebookfrenzy.com*.

Chapter 2

2. A Brief History of Red Hat Linux

Red Hat Enterprise Linux is one of a number of variants (also referred to as *distributions*) of the Linux operating system and is the product of a U.S. company named Red Hat, Inc., based in Raleigh, North Carolina. The company was founded in the mid-1990s through the merger of two companies owned at the time by Marc Ewing and Bob Young. The origins of Linux, however, go back even further. This chapter will outline the history of both the Linux operating system and Red Hat, Inc.

2.1 What exactly is Linux?

Linux is an operating system in much the same way that Windows is an operating system (and there any similarities between Linux and Windows end). The term operating system is used to describe the software that acts as a layer between the hardware in a computer and the applications that we all run on a daily basis. When programmers write applications, they interface with the operating system to perform such tasks as writing files to the hard disk drive and displaying information on the screen. Without an operating system, every programmer would have to write code to directly access the hardware of the system. In addition, the programmer would have to be able to support every single piece of hardware ever created to be sure the application would work on every possible hardware configuration. Because the operating system handles all of this hardware complexity, application development becomes a much easier task. Linux is just one of a number of different operating systems available today.

2.2 UNIX Origins

To understand the history of Linux, we first have to go back to AT&T Bell Laboratories in the late 1960s. During this time AT&T had discontinued involvement in developing a new operating system named Multics. However, two AT&T engineers, Ken Thompson and Dennis Ritchie, decided to take what they had learned from the Multics project and create a new operating system named UNIX which quickly gained popularity and wide adoption both with corporations and academic institutions.

A variety of proprietary UNIX implementations eventually came to market, including those created by IBM (AIX), Hewlett-Packard (HP-UX), and Sun Microsystems (SunOS and Solaris). In addition, a UNIX-like operating system named MINIX was created by Andrew S. Tanenbaum and designed for educational use with source code access provided to universities.

2.3 Who Created Linux?

The origins of Linux can be traced back to the work and philosophies of two people. At the heart of the Linux operating system is something called the *kernel*. This is the core set of features necessary for the operating system to function. The kernel manages the system's resources and handles communication between the hardware and the applications. The Linux kernel was developed by

Linus Torvalds, who, taking a dislike to MS-DOS and impatient for the availability of MINIX for the new Intel 80386 microprocessor, decided to write his own UNIX-like kernel. When he had finished the first version of the kernel, he released it under an open-source license that enabled anyone to download the source code and freely use and modify it without having to pay Linus any money.

Around the same time, Richard Stallman at the Free Software Foundation, a strong advocate of free and open-source software, was working on an open-source operating system of his own. Rather than focusing initially on the kernel, however, Stallman began by developing open-source versions of all the UNIX tools, utilities and compilers necessary to use and maintain an operating system. By the time he had finished developing this infrastructure, the obvious solution was to combine his work with the kernel Linus had written to create a complete operating system. This combination became known as GNU/Linux. Purists insist that Linux always be referred to as GNU/Linux (in fact, at one time, Richard Stallman refused to give press interviews to any publication which failed to refer to Linux as GNU/Linux). This is not unreasonable given that the GNU tools developed by the Free Software Foundation make up a significant and vital part of GNU/Linux. Unfortunately, most people and publications simply refer to Linux as Linux and this will probably always continue to be the case.

2.4 The Early Days of Red Hat

In 1993 Bob Young created a company named ACC Corporation which, according to Young, he ran from his "wife's sewing closet". The name ACC was intended to represent a catalog business but was also an abbreviation of a small business his wife ran called "Antiques and Collectibles of Connecticut". Among the items sold through the ACC catalog business were Linux CDs and related open-source software.

Around the same time, Marc Ewing had created his own Linux distribution company which he named Red Hat Linux (after his propensity to wear a red baseball cap while at Carnegie Mellon University).

In 1995, ACC acquired Red Hat, adopted the name Red Hat, Inc. and experienced rapid and significant growth. Bob Young stepped down as CEO shortly after the company went public in August of 1999 and has since pursued a number of business and philanthropic efforts including a print-on-demand book publishing company named Lulu and ownership of two Canadian professional sports teams. In 2018, IBM acquired Red Hat, Inc. in a deal valued at $34 billion.

2.5 Red Hat Support

Early releases of Red Hat Linux were shipped to customers on floppy disks and CDs (this, of course, predated the widespread availability of broadband internet connections). When users encountered problems with the software they were only able to contact Red Hat by email. In fact, Bob Young often jokes that this was effective in limiting support requests since, by the time a customer realized they needed help, their computer was usually inoperative and therefore unavailable to be used to send an email message seeking assistance from Red Hat's support team. In later years Red Hat provided better levels of support tied to paid subscriptions and now provides a variety of support levels ranging from "self help" (no support) up to premium support.

2.6 Open Source

Red Hat Enterprise Linux 9 is the current commercial offering from Red Hat and is primarily targeted at corporate, mission critical installations. It is also the cornerstone of an expanding ecosystem of products and services offered by Red Hat. RHEL is an open-source product in that you can download the source code free of charge and build the software yourself if you wish to do so (a task not to be undertaken lightly). If, however, you wish to download a pre-built, ready to install binary version of the software (either with or without support), you have to pay for it.

2.7 The Fedora Project

Red Hat also sponsors the Fedora Project, the goal of which is to provide access to a free Linux operating system (in both source and binary distributions) in the form of Fedora Linux. Fedora Linux also serves as a proving ground for many of the new features that are eventually adopted into the Red Hat Enterprise Linux operating system family and the CentOS derivative.

2.8 CentOS Stream - The Free Alternative

For users unable to afford a Red Hat Enterprise Linux subscription, another option is the CentOS Stream operating system. The CentOS project, initially a community-driven effort but now owned by Red Hat, takes the Red Hat Enterprise Linux source code, removes the Red Hat branding and subscription requirements, compiles it, and provides the distribution for download. Like Fedora, CentOS Stream field tests new operating system features before they are included in a future RHEL release. As such, it may lack stability but provides access to cutting-edge features.

2.9 Summary

The origins of the Linux operating system can be traced back to the work of Linus Torvalds and Richard Stallman in the form of the Linux kernel combined with the tools and compilers built by the GNU project.

Over the years, the open-source nature of Linux has resulted in the release of a wide range of different Linux distributions. One such distribution is Red Hat Enterprise Linux, created by Red Hat, Inc., a company founded by Bob Young and Mark Ewing. Red Hat specializes in providing enterprise level Linux software solutions combined with extensive technical support services.

Chapter 3

3. Installing RHEL 9 on a Clean Disk Drive

There are now two ways in which a RHEL 9 system can be deployed. One method is to either purchase new hardware or re-purpose an existing computer system on which to install and run the operating system. Another option is to create a cloud-based operating system instance using services such as Amazon AWS, Google Cloud, or Microsoft Azure (to name but a few). Since cloud-based instances are typically created by selecting a pre-configured, ready-to-run operating system image already optimized for the cloud platform and using that as the basis for the RHEL system, there is no need to perform a manual operating system installation in this situation.

If, on the other hand, you plan to install RHEL 9 on your own hardware, the first step on the path to learning about Red Hat Enterprise Linux 9 involves installing the operating system.

RHEL can be installed either in a clean disk environment (where an entire disk is cleared of any existing partitions and dedicated entirely to RHEL) or in a dual boot environment where RHEL co-exists with another operating system on the disk (typically a member of the Microsoft Windows family of operating systems).

This chapter will cover the clean disk approach to installation from local or remote installation media. Dual boot installation with a Windows 11 system will be covered in *"Dual Booting RHEL 9 with Windows"*.

3.1 Obtaining the RHEL 9 Installation Media

Although RHEL is an open-source operating system, and as such, the source code is freely accessible, the binary installation images are only available as part of a paid Red Hat Enterprise Linux subscription. In addition to access to the installation images for RHEL, subscription levels are available that provide technical support for the operating system. If you already have a paid subscription, log into your Red Hat account to download the operating system. If you want to try RHEL before purchasing, Red Hat provides a 30-day trial. To register for a trial, visit the Red Hat website at the following address:

https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/try-it

Red Hat Developer program members are also provided access to RHEL 9. Details on joining this program can be found on the Red Hat Developers website at *https://developers.redhat.com*.

RHEL 9 is available for Intel 64-bit (x86_64) and ARM64 (AArch64) CPU architecture families. The installation media can be downloaded as a DVD or boot ISO image for each architecture. The DVD ISO image is self-contained, including all of the packages necessary to install a RHEL system, and is named using the following convention:

```
rhel-baseos-<version>-<architecture>-dvd.iso
```

For example, the RHEL 9.1 DVD image for 64-bit Intel systems is named as follows:

```
rhel-baseos-9.1-x86_64-dvd.iso
```

On the other hand, the boot ISO image only contains the files necessary to begin the installation and requires access to the full installation media located on a remote server. The boot image can also perform rescue operations on an existing RHEL system. The boot ISO is named using the following convention:

```
rhel-baseos-<version>-<architecture>-boot.iso
```

The RHEL 9.1 boot image for AArch64, for example, is named as follows:

```
rhel-baseos-9.1-x86_64-boot.iso
```

Having downloaded an image, either burn it to disk, or use the steps in the next section to write the media to a USB drive, configure your virtualization environment to treat it as a DVD drive or use the steps outlined later in this chapter to access the installation image over a network connection.

3.2 Writing the ISO Installation Image to a USB Drive

These days it is more likely that an operating system installation will be performed from a USB drive than from a DVD. Having downloaded the ISO installation image for RHEL 9, the steps to write that image to a USB drive will differ depending on whether the drive is attached to a Linux, macOS, or Windows system. The steps outlined in the remainder of this section assume that the USB drive is new or has been reformatted to remove any existing data or partitions:

3.2.1 Linux

The first step in writing an ISO image to a USB drive on Linux is identifying the device name. Before inserting the USB drive, identify the storage devices already detected on the system by listing the devices in */dev* as follows:

```
# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2
```

Attach the USB drive to the Linux system and run the *dmesg* command to get a list of recent system messages, one of which will be a report that the USB drive was detected and will be similar to the following:

```
[  406.241717] sd 6:0:0:0: [sdb] Attached SCSI removable disk
```

This output tells us that we should expect the device name to include "sdb" which we can confirm by listing device names in */dev* again:

```
# ls /dev/sd*
/dev/sda   /dev/sda1   /dev/sda2   /dev/sdb
```

This output shows that the USB drive has been assigned to /dev/sdb. The next step before writing the ISO image to the device is to run the *findmnt* command to make sure it has not been auto-mounted:

```
# findmnt /dev/sdb
TARGET                                           SOURCE   FSTYPE OPTIONS
/run/media/neil/d6bf9574-7e31-4f54-88b1          /dev/sdb ext3   rw,nosuid,no
```

If the *findmnt* command indicates that the USB drive has been mounted, unmount it before continuing:

```
# umount /run/media/neil/d6bf9574-7e31-4f54-88b1
```

Once the filesystem has been unmounted, use the *dd* command as follows to write the ISO image to the drive:

```
# dd if=/path/to/iso/<image name>.iso of=/dev/sdb bs=512k
```

The writing process can take some time (as long as 10 - 15 minutes) depending on the image size and speed of the system on which it is running. Once the image has been written, output similar to the following will appear, and the USB drive will be ready to be used to install RHEL 9:

```
5956+0 records in
5956+0 records out
3122659328 bytes (3.1 GB, 2.9 GiB) copied, 426.234 s, 7.3 MB/s
```

3.2.2 macOS

The first step in writing an ISO image to a USB drive attached to a macOS system is to identify the device using the *diskutil* tool. Before attaching the USB device, open a Terminal window and run the following command:

```
$ diskutil list
/dev/disk0 (internal, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                        *1.0 TB     disk0
   1:                        EFI EFI                     209.7 MB   disk0s1
   2:                 Apple_APFS Container disk2         1000.0 GB  disk0s2

/dev/disk1 (internal):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      GUID_partition_scheme                         28.0 GB    disk1
   1:                        EFI EFI                     314.6 MB   disk1s1
   2:                 Apple_APFS Container disk2         27.7 GB    disk1s2

/dev/disk2 (synthesized):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:      APFS Container Scheme -                      +1.0 TB     disk2
                                 Physical Stores disk1s2, disk0s2
   1:                APFS Volume Macintosh HD            473.6 GB   disk2s1
   2:                APFS Volume Preboot                 42.1 MB    disk2s2
   3:                APFS Volume Recovery                517.0 MB   disk2s3
   4:                APFS Volume VM                      1.1 GB     disk2s4
```

Having established a baseline of detected devices, insert the USB drive into a port on the macOS system and rerun the command. The same results should appear with one additional entry for the USB drive resembling the following:

```
/dev/disk3 (external, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                                                   *16.0 GB    disk3
```

In the above example, the USB drive has been assigned to /dev/disk3. Before proceeding, unmount the disk as follows:

```
$ diskutil unmountDisk /dev/disk3
Unmount of all volumes on disk3 was successful
```

Finally, use the *dd* command to write the ISO image to the device, taking care to reference the raw disk device (/dev/**r**disk3) and entering your user password when prompted:

```
$ sudo dd if=/path/to/iso/image.iso of=/dev/rdisk3 bs=1m
```

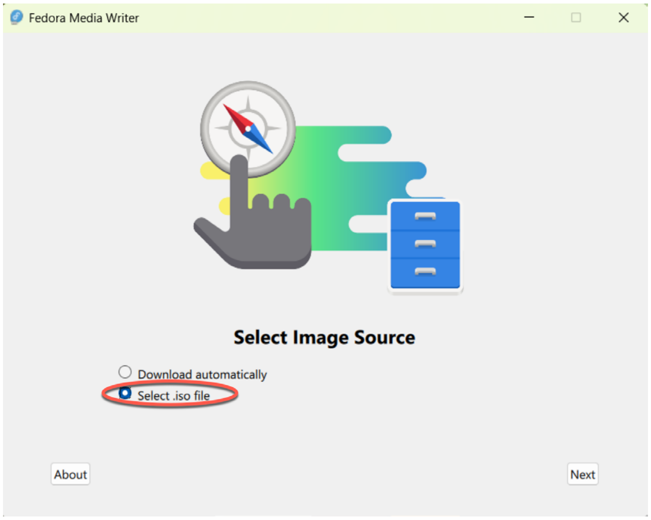
Once the image has been written, the USB drive is ready.

3.2.3 Windows/macOS

Several free tools are available for Windows and macOS that will write an ISO image to a USB drive, but one written specifically for writing Linux ISO images is the Fedora Media Writer tool which can be downloaded from the following URL:

https://getfedora.org/en/workstation/download/

Once installed, insert the destination USB drive, launch the writer tool, and choose the *Select .iso file* option as highlighted in Figure 3-1:

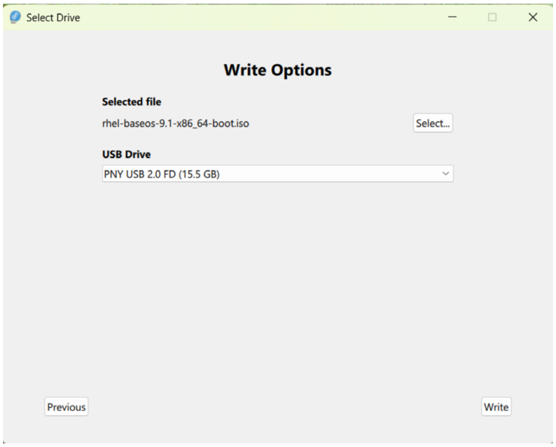


Select Image Source

○ Download automatically
● Select .iso file

Figure 3-1

Click Next to proceed to the Write Options screen and select the USB Drive before clicking on the *Select...* button:

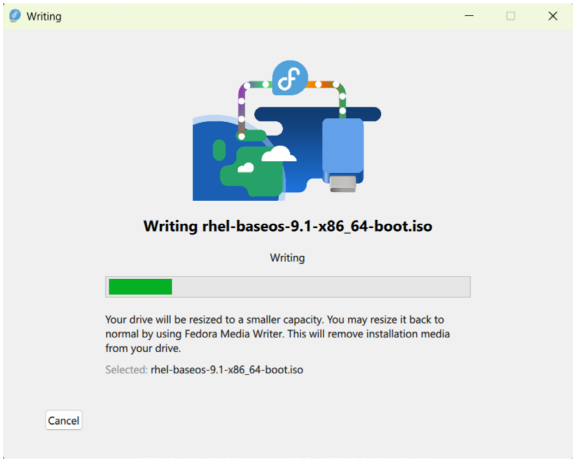


Selected file
rhel-baseos-9.1-x86_64-boot.iso

USB Drive
PNY USB 2.0 FD (15.5 GB)

Figure 3-2

In the resulting file selection dialog, navigate to and select the RHEL 9 installation ISO image and click the *Open* button. Finally, click the Write button to start writing the image to the USB drive:



Writing rhel-baseos-9.1-x86_64-boot.iso

Writing

Your drive will be resized to a smaller capacity. You may resize it back to normal by using Fedora Media Writer. This will remove installation media from your drive.

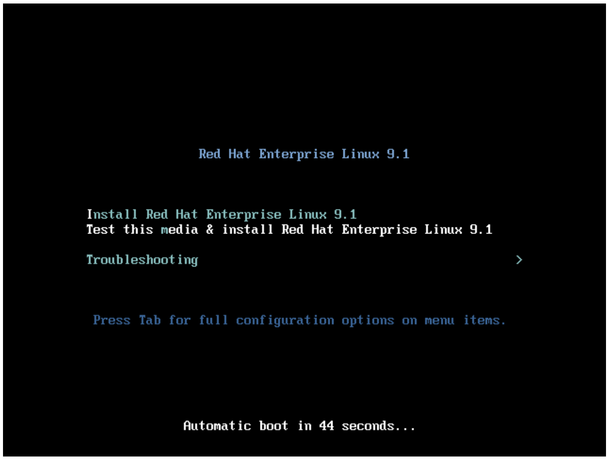
Selected: rhel-baseos-9.1-x86_64-boot.iso

Figure 3-3

Once the image has been written to the device, the device is ready to perform the installation.

3.3 Installing Red Hat Enterprise Linux 9

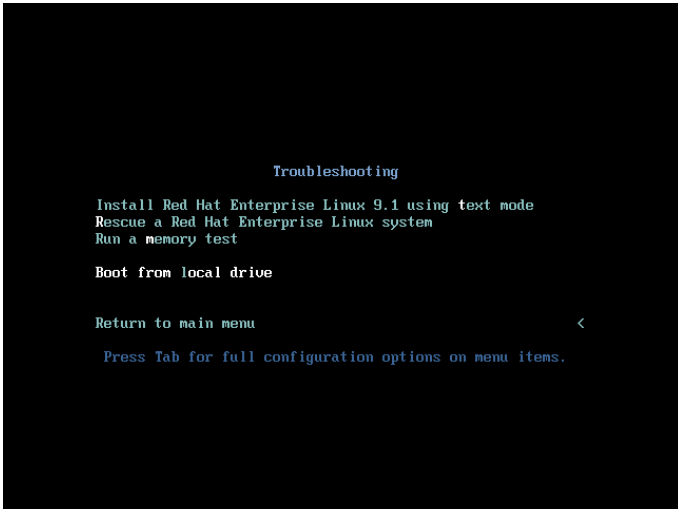
Insert the RHEL 9 installation media into the appropriate drive and power on the system. If the system tries to boot from the hard disk drive, you will need to enter the BIOS set up for your computer and change the boot order so that it boots from the installation media drive first. Once the system has booted, you will be presented with the following screen:



```
Red Hat Enterprise Linux 9.1

Install Red Hat Enterprise Linux 9.1
Test this media & install Red Hat Enterprise Linux 9.1

Troubleshooting                                          >

Press Tab for full configuration options on menu items.

Automatic boot in 44 seconds...
```

Figure 3-4

Use the arrow keys to navigate between the options and make a selection with the <Enter> key. For example, if the *Troubleshooting* option is selected, the screen shown in Figure 3-5 will appear, including options to boot from the current operating system on the local drive (if one is installed), test the system memory, or rescue an installed RHEL 9 system. An option is also available to perform the installation in basic graphics mode for systems without a graphical console:

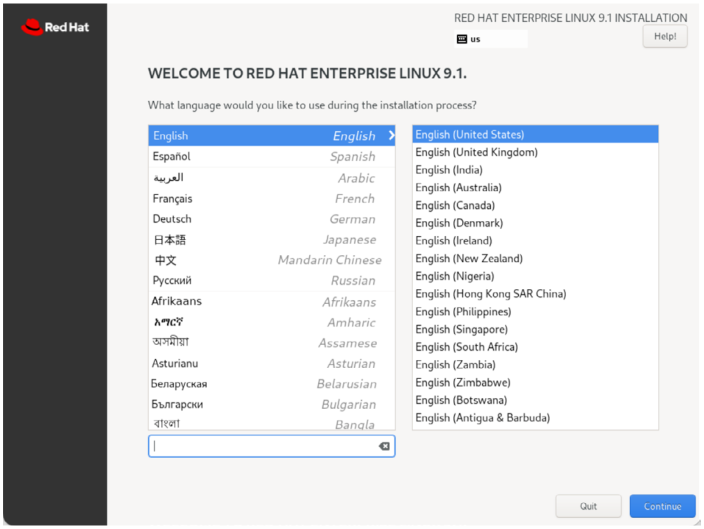


```
Troubleshooting

Install Red Hat Enterprise Linux 9.1 using text mode
Rescue a Red Hat Enterprise Linux system
Run a memory test

Boot from local drive

Return to main menu                                      <

Press Tab for full configuration options on menu items.
```

Figure 3-5

If you install the ARM64 image, the options to boot from the local drive and run a memory test may not be available on the troubleshooting screen.

Select the option on the main screen to install Red Hat Enterprise Linux and, after a short delay, the first screen of the graphical installer will appear:

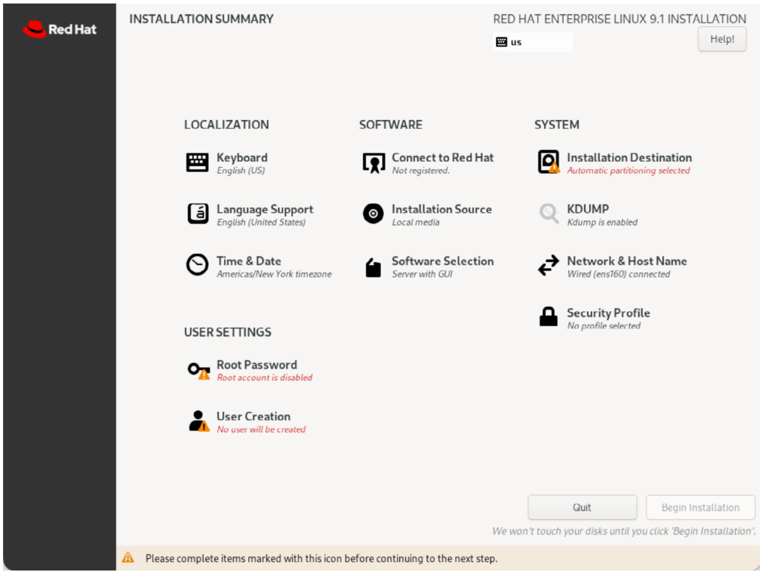


WELCOME TO RED HAT ENTERPRISE LINUX 9.1.

What language would you like to use during the installation process?

Figure 3-6

Select your preferred language on the first screen before clicking *Continue* to proceed to the main installation screen as shown in Figure 3-7:



INSTALLATION SUMMARY

LOCALIZATION

Keyboard
English (US)

Language Support
English (United States)

Time & Date
Americas/New York timezone

SOFTWARE

Connect to Red Hat
Not registered.

Installation Source
Local media

Software Selection
Server with GUI

SYSTEM

Installation Destination
Automatic partitioning selected

KDUMP
Kdump is enabled

Network & Host Name
Wired (ens160) connected

Security Profile
No profile selected

USER SETTINGS

Root Password
Root account is disabled

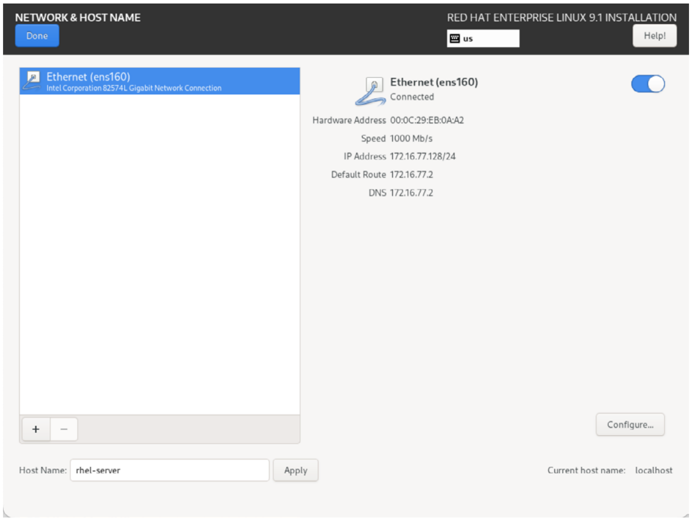
User Creation
No user will be created

We won't touch your disks until you click 'Begin Installation'.

⚠ Please complete items marked with this icon before continuing to the next step.

Figure 3-7

Begin by selecting the *Network & Host Name* option, enable a network device on your system, and enter a hostname before clicking the *Apply* button:



Ethernet (ens160)
Intel Corporation 82574L Gigabit Network Connection

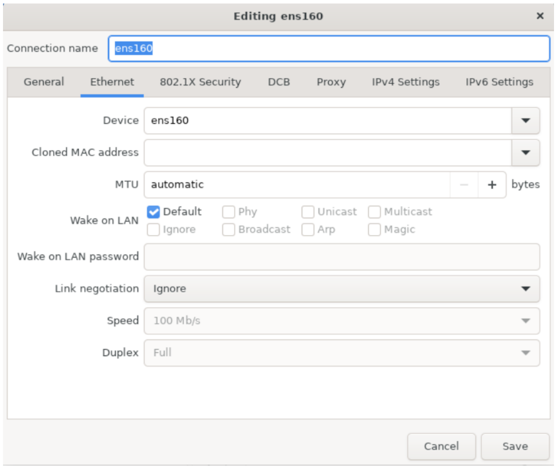
Ethernet (ens160)
Connected

Hardware Address 00:0C:29:EB:0A:A2
Speed 1000 Mb/s
IP Address 172.16.77.128/24
Default Route 172.16.77.2
DNS 172.16.77.2

Host Name: rhel-server
Current host name: localhost

Figure 3-8

If your network connection requires additional settings, click on the *Configure...* button to access the advanced network settings screen illustrated in Figure 3-9:



Connection name: ens160

Device: ens160
Cloned MAC address:
MTU: automatic − + bytes
Wake on LAN: ☑ Default ☐ Phy ☐ Unicast ☐ Multicast ☐ Ignore ☐ Broadcast ☐ Arp ☐ Magic
Wake on LAN password:
Link negotiation: Ignore
Speed: 100 Mb/s
Duplex: Full

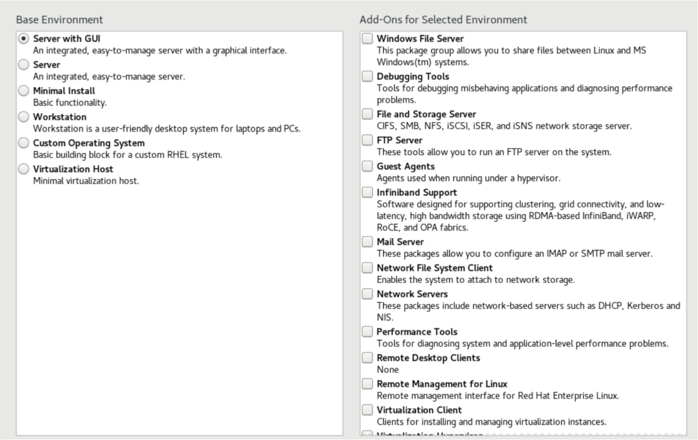
Figure 3-9

Once the hostname has been defined, and a network connection enabled, click *Done* to return to the main screen.

If you wish to change the keyboard, language, or time and date settings, select the corresponding option from the Localization column of the main screen. On the *Time & Date* screen, make a selection corresponding to your geographical location. The choice is also provided to use *Network Time* which automatically synchronizes the system with external Network Time Protocol servers.

Changes to the *Installation Source* settings should not be necessary since the installation is performed from local media. To complete the installation from media located on a remote server, select *Installation Source*, enable the *On the network* option, and then specify the location of the installation media and the type of URL being used. The installation media can, for example, be an ISO image installed either on a remote web server or on a system on the local network using NFS (the topic of NFS file sharing is covered in detail in the chapter entitled *"Using NFS on RHEL 9 to Share Files with Remote Systems"*), or the URL of a remote repository (repositories are essentially online collections of the software, packages, and applications that make up the operating system from which installations onto the local system can be performed).

By default, the installer will perform a Server with GUI installation of RHEL 9. This will consist of the minimum packages necessary to run the operating system and the graphical desktop environment. To select a different installation configuration, choose the *Software Selection* option to display the screen shown below:



Base Environment

● **Server with GUI**
An integrated, easy-to-manage server with a graphical interface.

○ **Server**
An integrated, easy-to-manage server.

○ **Minimal Install**
Basic functionality.

○ **Workstation**
Workstation is a user-friendly desktop system for laptops and PCs.

○ **Custom Operating System**
Basic building block for a custom RHEL system.

○ **Virtualization Host**
Minimal virtualization host.

Add-Ons for Selected Environment

□ **Windows File Server**
This package group allows you to share files between Linux and MS Windows(tm) systems.

□ **Debugging Tools**
Tools for debugging misbehaving applications and diagnosing performance problems.

□ **File and Storage Server**
CIFS, SMB, NFS, iSCSI, iSER, and iSNS network storage server.

□ **FTP Server**
These tools allow you to run an FTP server on the system.

□ **Guest Agents**
Agents used when running under a hypervisor.

□ **Infiniband Support**
Software designed for supporting clustering, grid connectivity, and low-latency, high bandwidth storage using RDMA-based InfiniBand, iWARP, RoCE, and OPA fabrics.

□ **Mail Server**
These packages allow you to configure an IMAP or SMTP mail server.

□ **Network File System Client**
Enables the system to attach to network storage.

□ **Network Servers**
These packages include network-based servers such as DHCP, Kerberos and NIS.

□ **Performance Tools**
Tools for diagnosing system and application-level performance problems.

□ **Remote Desktop Clients**
None

□ **Remote Management for Linux**
Remote management interface for Red Hat Enterprise Linux.

□ **Virtualization Client**
Clients for installing and managing virtualization instances.

Figure 3-10

Use the left-hand panel to select a basic configuration and the right-hand panel to add any additional packages you know you will need after the system starts up. If the desktop environment is not required and you are unsure which packages to install on the system, use the *Minimal install* option and install additional packages as needed once the system has booted. This avoids installing any packages that may never be required.

The *Security Policy* option allows additional security packages to be installed on the system that allow security restrictions that comply with the Security Content Automation Protocol (SCAP) standards to be imposed on the system. While selecting a profile from the list installs the necessary packages, the policy restrictions are not enforced until they are enabled on the running system. Unless you work for a government entity or company that mandates this level of security, selecting a policy on this screen is optional.

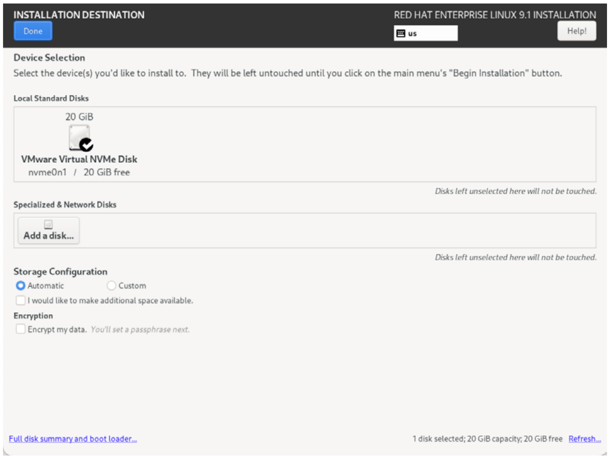
The *Kdump* feature, when enabled, writes out the state of the operating system kernel in the event of a system crash. This dump file can help identify the cause of a crash but takes up system memory when enabled. Unless memory is limited on your system, this can be left in the enabled state.

Installing RHEL 9 on a Clean Disk Drive

Having configured the basic settings, the next step is to decide how the hard disk drive will be partitioned to accommodate the operating system.

3.4 Partitioning a Disk for RHEL 9

When the *Installation Destination* option is selected, the installer will present a screen similar to the one illustrated in Figure 3-11 below:



INSTALLATION DESTINATION

RED HAT ENTERPRISE LINUX 9.1 INSTALLATION

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

20 GiB
VMware Virtual NVMe Disk
nvme0n1 / 20 GiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Storage Configuration

● Automatic ○ Custom

□ I would like to make additional space available.

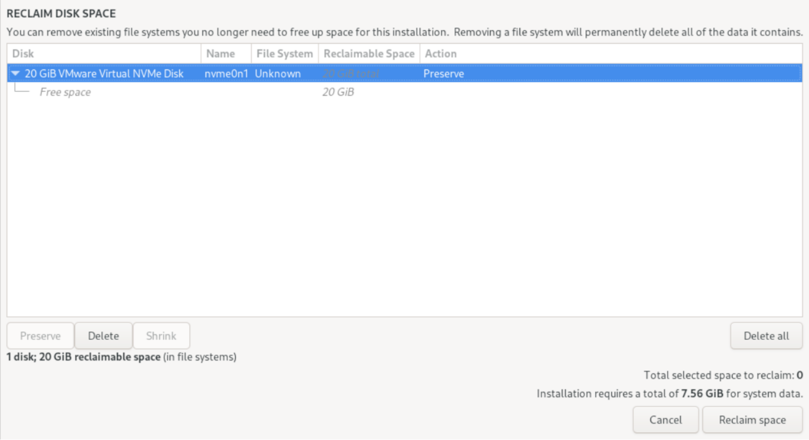
Encryption

□ Encrypt my data. *You'll set a passphrase next.*

1 disk selected; 20 GiB capacity; 20 GiB free

Figure 3-11

By default, the installer is configured to automatically install the operating system using the available space on the currently selected disk drive and to configure standard partition sizes. If the disk previously contained another operating system, these existing partitions will not be deleted, potentially leaving unused space on the drive after RHEL 9 has been installed. To remove any existing partitions so that they can be reclaimed and used by RHEL 9, enable the *I would like to make additional space available* option and click on the *Done* button to display the dialog shown in Figure 3-12:



RECLAIM DISK SPACE

You can remove existing file systems you no longer need to free up space for this installation. Removing a file system will permanently delete all of the data it contains.

Disk	Name	File System	Reclaimable Space	Action
▼ 20 GiB VMware Virtual NVMe Disk	nvme0n1	Unknown	20 GiB total	Preserve
Free space			*20 GiB*	

1 disk; 20 GiB reclaimable space (in file systems)

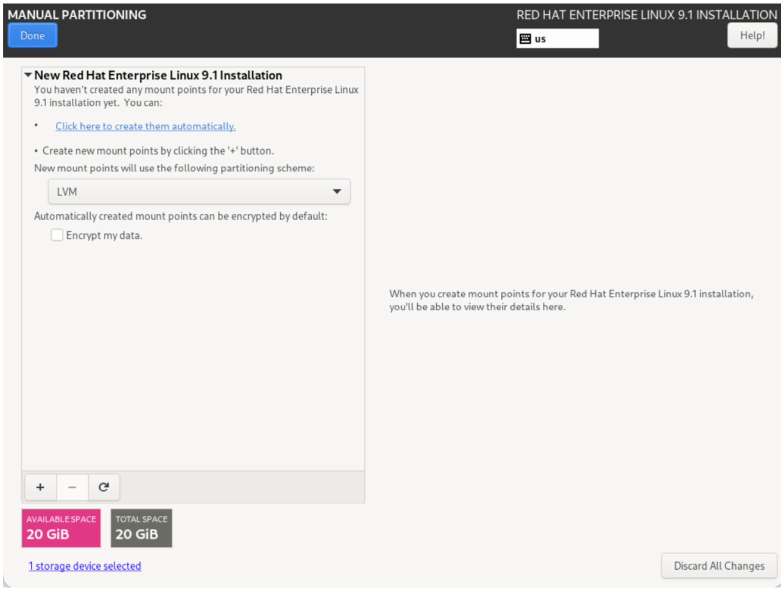
Total selected space to reclaim: **0**

Installation requires a total of **7.56 GiB** for system data.

Figure 3-12

To reclaim space, select a partition that is no longer needed and mark it for removal by clicking the *Delete* button. Once all the partitions to be removed have been selected, click on the *Reclaim Space* button to perform the deletion. The reclaimed space will now be used automatically by the RHEL 9 installer.

To manually configure the layout of the disk in terms of how the available space will be allocated, change the *Storage Allocation* setting from *Automatic* to *Custom* and click *Done* to display the Manual Partitioning screen (Figure 3-13):



New Red Hat Enterprise Linux 9.1 Installation

You haven't created any mount points for your Red Hat Enterprise Linux 9.1 installation yet. You can:

- Click here to create them automatically.
- Create new mount points by clicking the '+' button.

New mount points will use the following partitioning scheme:

LVM

Automatically created mount points can be encrypted by default:

□ Encrypt my data.

When you create mount points for your Red Hat Enterprise Linux 9.1 installation, you'll be able to view their details here.

AVAILABLE SPACE 20 GiB

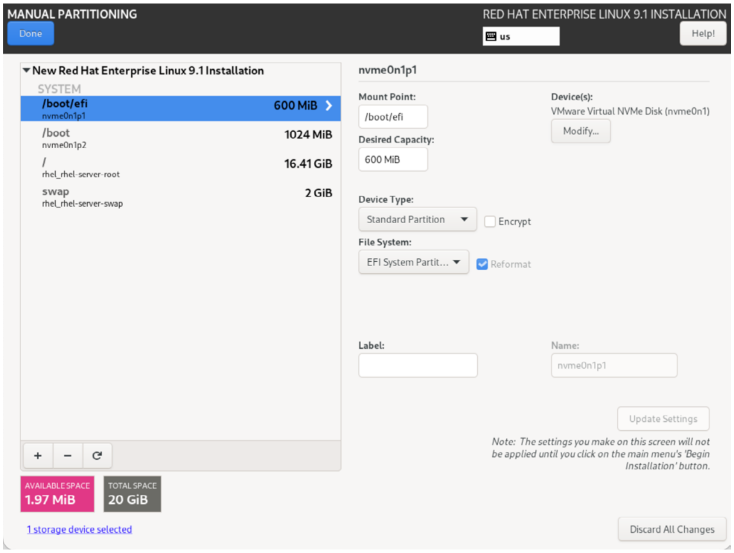
TOTAL SPACE 20 GiB

Figure 3-13

The manual partitioning screen allows configuring the disk using Logical Volume Management (LVM) or standard partitioning. LVM is the recommended option because it provides flexibility in terms of managing partition sizes once the system is up and running (LVM is covered in detail in the chapter entitled *"Adding a New Disk to a RHEL 9 Volume Group and Logical Volume").*

Once a partitioning scheme has been selected, the last step is to decide on the sizes of the partitions and the corresponding filesystem mount points. In general, the default configuration provided by the *Click here to create them automatically* option will meet most system needs. However, to manually create partitions and allocate mount points, click the + button to declare and assign each partition manually.

Another option is to select automatic partition creation and then use the resulting screen to change the partition configuration as needed manually. Figure 3-14, for example, shows the partition configuration defined by selecting the automatic creation option and allows the settings for each partition to be modified before any changes are made to the disk:



▾ New Red Hat Enterprise Linux 9.1 Installation

SYSTEM

/boot/efi nvme0n1p1	600 MiB ›
/boot nvme0n1p2	1024 MiB
/ rhel_rhel-server-root	16.41 GiB
swap rhel_rhel-server-swap	2 GiB

nvme0n1p1

Mount Point: /boot/efi

Device(s): VMware Virtual NVMe Disk (nvme0n1)

Modify...

Desired Capacity: 600 MiB

Device Type: Standard Partition ☐ Encrypt

File System: EFI System Partit... ☑ Reformat

Label:

Name: nvme0n1p1

Update Settings

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

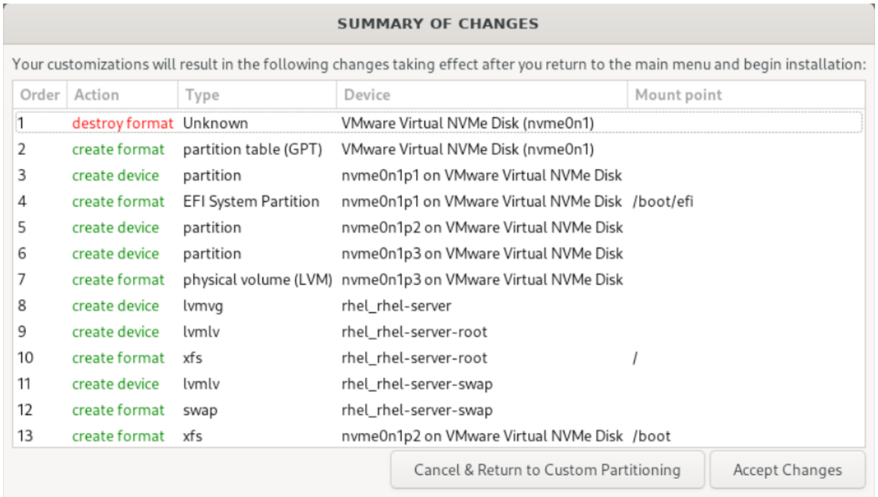
AVAILABLE SPACE 1.97 MiB

TOTAL SPACE 20 GiB

1 storage device selected

Figure 3-14

Once the disk has been configured, click on *Done*, review the summary of the changes to be made to the disk (Figure 3-15), followed by the *Accept Changes* button:


SUMMARY OF CHANGES

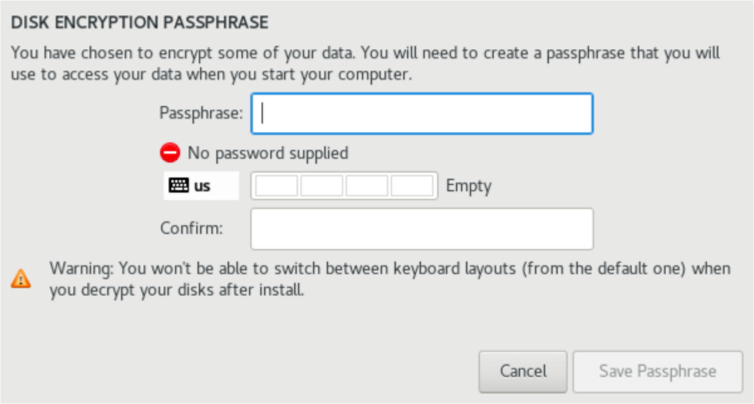
Your customizations will result in the following changes taking effect after you return to the main menu and begin installation:

Order	Action	Type	Device	Mount point
1	destroy format	Unknown	VMware Virtual NVMe Disk (nvme0n1)	
2	create format	partition table (GPT)	VMware Virtual NVMe Disk (nvme0n1)	
3	create device	partition	nvme0n1p1 on VMware Virtual NVMe Disk	
4	create format	EFI System Partition	nvme0n1p1 on VMware Virtual NVMe Disk	/boot/efi
5	create device	partition	nvme0n1p2 on VMware Virtual NVMe Disk	
6	create device	partition	nvme0n1p3 on VMware Virtual NVMe Disk	
7	create format	physical volume (LVM)	nvme0n1p3 on VMware Virtual NVMe Disk	
8	create device	lvmvg	rhel_rhel-server	
9	create device	lvmlv	rhel_rhel-server-root	
10	create format	xfs	rhel_rhel-server-root	/
11	create device	lvmlv	rhel_rhel-server-swap	
12	create format	swap	rhel_rhel-server-swap	
13	create format	xfs	nvme0n1p2 on VMware Virtual NVMe Disk	/boot

Figure 3-15

3.5 Disk Encryption

RHEL 9 also allows the disk's contents to be protected by encryption. This will require a passphrase to be entered each time the system boots. To enable encryption, select the *Encrypt my data* option on the main *Installation Destination* screen and, after clicking the *Done* button, enter the passphrase in the dialog shown in Figure 3-16:



DISK ENCRYPTION PASSPHRASE

You have chosen to encrypt some of your data. You will need to create a passphrase that you will use to access your data when you start your computer.

Warning: You won't be able to switch between keyboard layouts (from the default one) when you decrypt your disks after install.

Figure 3-16

3.6 User Settings

Before the installation can be completed, you must create a root password and an optional user account. The root, or super-user account, is a special user with administrative privileges on the system. While you will generally use your own account to log into the system, you will need root privileges to configure the system and perform other administrative tasks.

Select the Root Password option and enter a suitably secure password. Options are also available to lock the root account, so it cannot be accessed and to allow access to the root account over remote connections to the system via password authentication. In general, it is recommended to leave the root account unlocked but to disallow remote SSH access using password access. If remote root access is required, it should be implemented using SSH key-based authentication, a topic we will be covering in the *"Configuring SSH Key-based Authentication on RHEL 9"* chapter of this book.

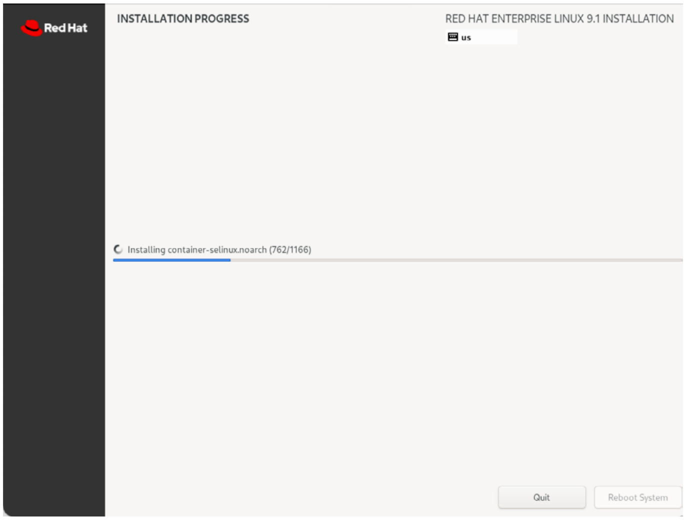
Next, select the User Creation option and enter a user name and password. If you would like this user to be able to perform administrative tasks using the *sudo* command, also enable the *Make this user administrator* checkbox. If this option is not enabled, the user must use the *su* command and enter the root password to gain root privileges (assuming that the option outlined above to lock the root account was not enabled). If the user you add will need to perform administrative tasks, sudo is the recommended option, and the checkbox should be enabled.

3.7 Registering the System

After the installation process has been completed and the system is running, the system must be registered with Red Hat before any updates, or additional packages can be installed. At the time of writing, installing the ARM64 version of RHEL 9 without first registering was possible, but the X86_64 version requires registration before installation. To register the system, select the Connect to Red Hat option and choose either the Account or Activation key option, depending on how you obtained your RHEL license. Once you have entered the information, click the Register button to complete the registration process.

3.8 The Physical Installation

Having made the appropriate package selections, clicking *Begin Installation* will start partitioning the disks and installing the packages that match the chosen installation settings. During this phase, the installation progress screen shown in Figure 3-17 will appear:



Installing container-selinux.noarch (762/1166)

Figure 3-17

Once all the system packages have been installed and configured, remove the installation media and click *Reboot System* to restart the system.

3.9 Final Configuration Steps

What appears on the console when the system has started will depend on whether the *Workstation* or *Server with GUI* option was chosen from the Software Selection installation screen. If either of these options was selected during installation, the GNOME Display Manager (GDM) login screen will appear. On the other hand, if the minimal or server configuration options were selected, the text-based login prompt will be displayed. Regardless of the configuration, log into the system as the user created during the installation process's final steps.

In the case of a *Workstation* or *Server with GUI* installation, the GNOME initial setup tool will launch and offer a guided tour of the desktop environment.

If you did not register your system before starting the installation, you can do so at the command prompt or a Terminal window in the case of the desktop environment. To launch a Terminal session on the desktop, click *Activities* in the menu bar followed by the *Terminal* icon in the panel at the bottom of the screen, as shown in Figure 3-18:

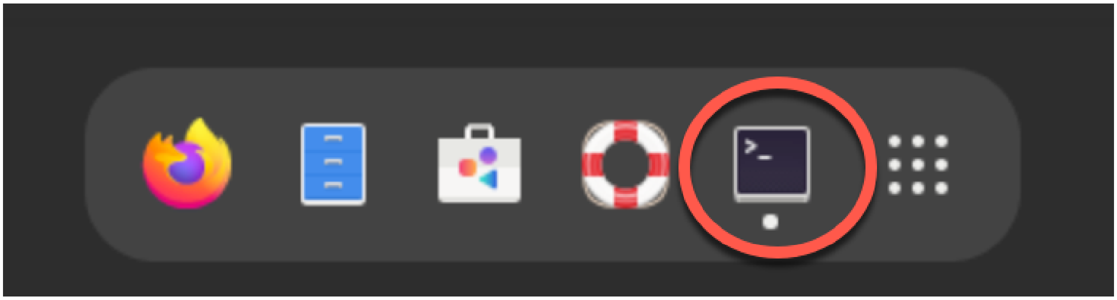


Figure 3-18

At the prompt, use the *sudo* command to gain root privileges and enter your user password:

```
$ sudo su -
```

Next, use the *subscription-manager* tool to register the system with Red Hat, entering your Red Hat Access username and password when prompted to do so:

```
# subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: myusername
Password:
The system has been registered with ID: 22f0ca3f-8d03-4933-be95-d6c4d8e3fe1a
The registered system name is: rhelserver1.
```

Finally, attach the system to a subscription service as follows:

```
# subscription-manager attach --auto
Installed Product Current Status:
Product Name: Red Hat Enterprise Linux for x86_64
Status:       Subscribed
```

3.10 Installing Updates

As with most operating systems today, each particular release of a RHEL distribution continues to evolve after it has been released to the public. This generally takes the form of bug fixes, security updates, and, occasionally, new features that may be downloaded over the internet and installed on your system.

Best practices dictate that the first step after installing RHEL is to make ensure any available updates are applied to the system. This can be achieved via the command-line prompt in a Terminal window using the *dnf* package manager tool. To check for the availability of updates, run the following command:

```
# dnf check-update
```

Any pending updates may be applied, once again, using the *dnf* tool:

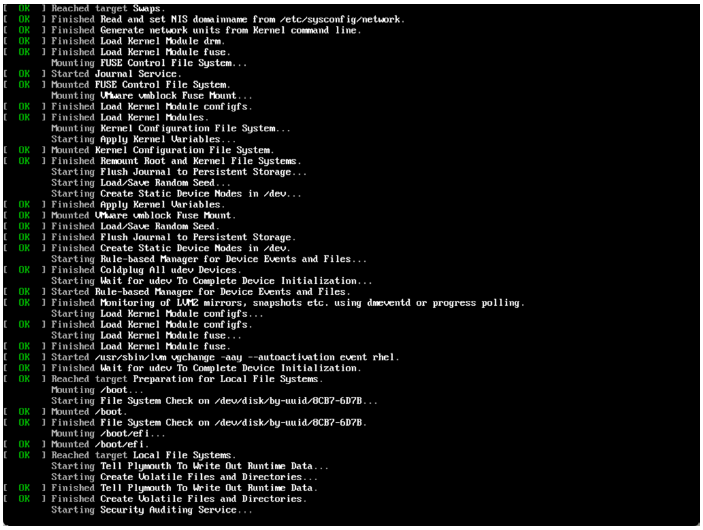
```
# dnf update
```

Upon execution, the *dnf* tool will provide a list of available packages for update and prompt for permission to perform the update.

Once the update is complete, the installation is essentially finished, and RHEL 9 is ready for use.

3.11 Displaying Boot Messages

During the boot process, RHEL 9 will display the Red Hat Graphical Boot (RHGB) screen, which hides from view all of the boot messages generated by the system as it loads. To make these messages visible during the boot process (as shown in Figure 3-19), press the keyboard Esc key while the system is starting:



```
[  OK  ] Reached target Swaps.
[  OK  ] Finished Read and set NIS domainname from /etc/sysconfig/network.
[  OK  ] Finished Generate network units from Kernel command line.
[  OK  ] Finished Load Kernel Module drm.
[  OK  ] Finished Load Kernel Module fuse.
         Mounting FUSE Control File System...
[  OK  ] Started Journal Service.
[  OK  ] Mounted FUSE Control File System.
         Mounting VMware vmblock Fuse Mount...
[  OK  ] Finished Load Kernel Module configfs.
[  OK  ] Finished Load Kernel Modules.
         Mounting Kernel Configuration File System...
         Starting Apply Kernel Variables...
[  OK  ] Mounted Kernel Configuration File System.
[  OK  ] Finished Remount Root and Kernel File Systems.
         Starting Flush Journal to Persistent Storage...
         Starting Load/Save Random Seed...
         Starting Create Static Device Nodes in /dev...
[  OK  ] Finished Apply Kernel Variables.
[  OK  ] Mounted VMware vmblock Fuse Mount.
[  OK  ] Finished Load/Save Random Seed.
[  OK  ] Finished Flush Journal to Persistent Storage.
[  OK  ] Finished Create Static Device Nodes in /dev.
         Starting Rule-based Manager for Device Events and Files...
[  OK  ] Finished Coldplug All udev Devices.
         Starting Wait for udev To Complete Device Initialization...
[  OK  ] Started Rule-based Manager for Device Events and Files.
[  OK  ] Finished Monitoring of LVM2 mirrors, snapshots etc. using dmeventd or progress polling.
         Starting Load Kernel Module configfs...
[  OK  ] Finished Load Kernel Module configfs.
         Starting Load Kernel Module fuse...
[  OK  ] Finished Load Kernel Module fuse.
[  OK  ] Started /usr/sbin/lvm vgchange -aay --autoactivation event rhel.
[  OK  ] Finished Wait for udev To Complete Device Initialization.
[  OK  ] Reached target Preparation for Local File Systems.
         Mounting /boot...
         Starting File System Check on /dev/disk/by-uuid/8CB7-6D7B...
[  OK  ] Mounted /boot.
[  OK  ] Finished File System Check on /dev/disk/by-uuid/8CB7-6D7B.
         Mounting /boot/efi...
[  OK  ] Mounted /boot/efi.
[  OK  ] Reached target Local File Systems.
         Starting Tell Plymouth To Write Out Runtime Data...
         Starting Create Volatile Files and Directories...
[  OK  ] Finished Tell Plymouth To Write Out Runtime Data.
[  OK  ] Finished Create Volatile Files and Directories.
         Starting Security Auditing Service...
```

Figure 3-19

The default behavior can be changed so that messages are always displayed by default by editing the */etc/default/grub* file and changing the GRUB_CMDLINE_LINUX setting, which, by default, will resemble the following:

```
GRUB_CMDLINE_LINUX="... rhgb quiet"
```

To remove the graphical boot screen so that messages are visible without pressing the Esc key, remove the "rhgb" option from the setting:

```
GRUB_CMDLINE_LINUX="... ~~rhgb~~ quiet"
```

This change will cause the system to display only a subset of the boot messages generated by the system. To show all messages generated by the system, also remove the "quiet" option:

```
GRUB_CMDLINE_LINUX="... ~~quiet~~"
```

Once the changes have been made, run the following command to generate a new boot configuration to take effect the next time the system starts:

```
# grub2-mkconfig --output=/boot/grub2/grub.cfg
```

3.12 Summary

The first step in working with RHEL 9 is to install the operating system. In the case of a cloud-based server, this task is typically performed automatically when an operating system image is selected for the system based on a range of options offered by the cloud service provider. Installation on your own hardware, however, involves downloading the installation media as an ISO image, writing that image to suitable storage such as a DVD or USB drive, and booting from it. Once running, the installation process allows a range of options to be configured, ranging from networking, whether the installation should proceed from the local media or via a remote server or repository, the packages to be installed, and the partitioning of the disk. Once installation is complete, it is important to install any operating system updates that may have been released since the original installation image was created.

Chapter 6

6. A Guided Tour of the GNOME 40 Desktop

Red Hat Enterprise Linux 9 includes the GNOME 40 desktop environment. Although lacking the complexity of Windows and macOS desktops, GNOME 40 provides an uncluttered and intuitive desktop environment that provides all of the essential features of a windowing environment with the added advantage that it can be learned quickly.

In this chapter, the main features of the GNOME desktop will be covered together with an outline of how basic tasks are performed.

6.1 Installing the GNOME Desktop

If the Workstation or Server with GUI software configuration was selected during the RHEL 9 installation process, the GNOME desktop will be installed and automatically launched each time the system starts.

If any other software configuration had been selected during the RHEL 9 installation process, the GNOME desktop would not have been included in the packages installed on the system. Installing a graphical desktop environment may seem redundant on server-based systems without a display attached. It is worth noting, however, that remote access to the GNOME desktop is also possible; even on headless servers (i.e., servers lacking a monitor, keyboard, and mouse), it may still be beneficial to install the GNOME desktop packages. The topic of establishing remote desktop access will be covered in detail in this book's *"RHEL 9 Remote Desktop Access with VNC"* chapter.

If the installation configuration did not include the GNOME desktop, it can be installed at any time using the following command:

```
# dnf groupinstall workstation
```

Once the installation is complete, the desktop environment may be launched from the command prompt on a monitor as follows:

```
$ startx
```

6.2 An Overview of the GNOME 40 Desktop

The screen shown in Figure 6-1 below shows the appearance of a typical, newly launched GNOME desktop session before any other programs have been launched or configuration changes made:

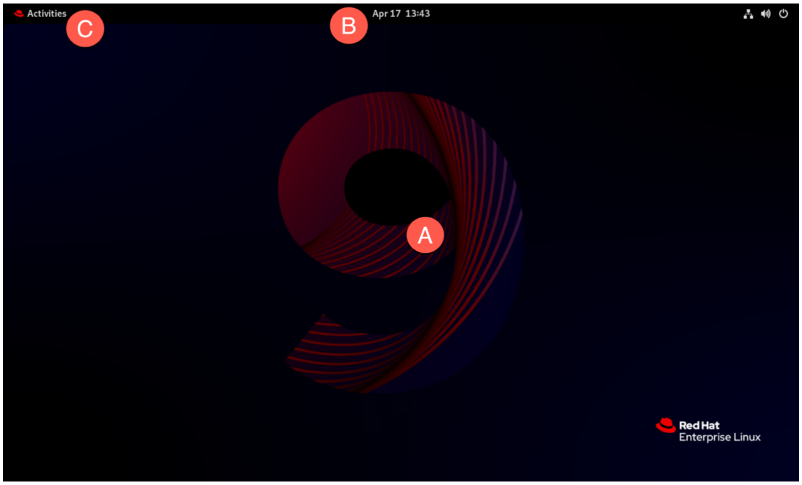


Figure 6-1

The main desktop area (marked A) is where windows will appear when applications and utilities are launched. Unlike other desktop environments, dragging and dropping files or applications onto the desktop is impossible, providing a clean and uncluttered workspace.

The bar at the top of the screen (B) is called the top bar and includes the Activities menu (C), the day and time, and a collection of buttons and icons, including network status, audio volume, battery power, and other status and account settings. In addition, the application menu for the currently active application running on the desktop will also appear in the top bar. Figure 6-2, for example, shows the application menu for the Terminal program:

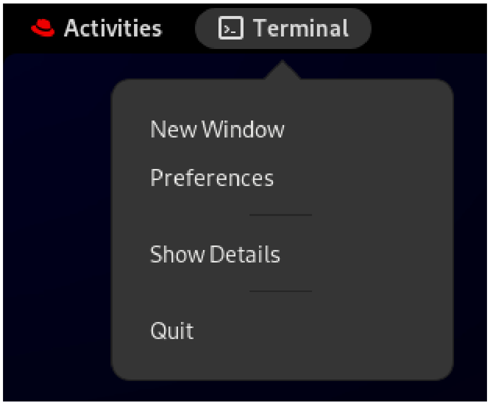


Figure 6-2

6.3 Activity Overview

Applications and utilities are launched using the Activities Overview, which may be displayed by clicking the Activities button in the top bar or pressing the special key on the keyboard. On Windows keyboards, this is the Windows key; on macOS, the Command key and Chromebooks, the key displaying a magnifying glass.

When displayed, the Activities Overview will resemble Figure 6-3 below:

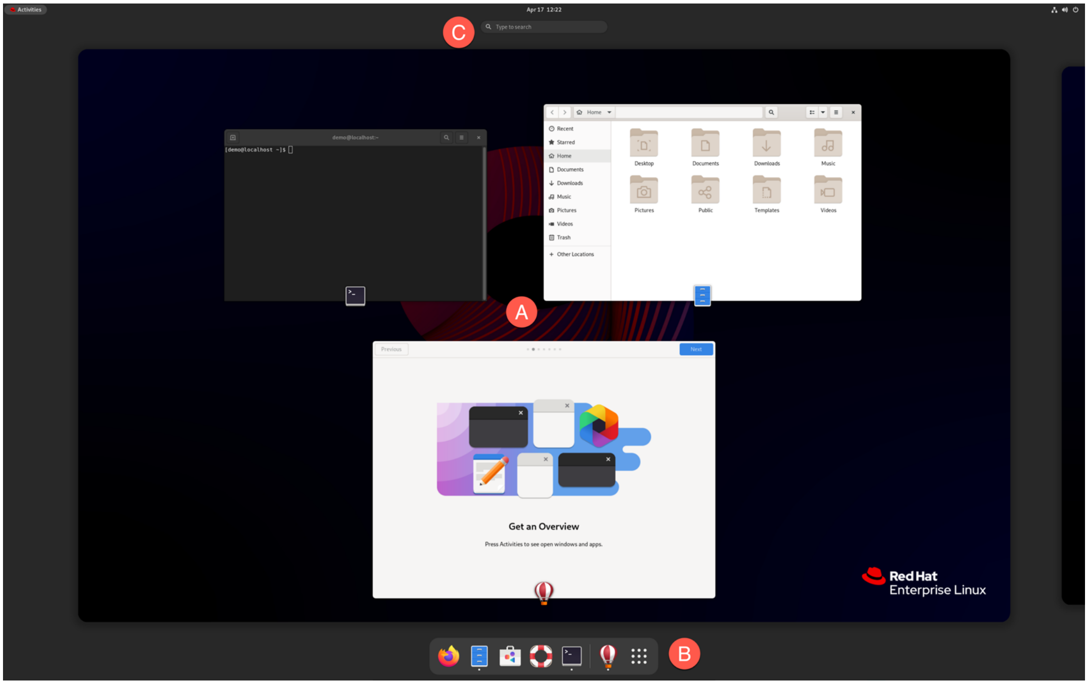


Figure 6-3

The central area of the overview (marked A in Figure 6-3 above) displays all of the currently open app windows. At the bottom of the screen (B) is the dashboard (also referred to as the *dash*) from which apps are launched. By default, the dash will display an icon for a predefined set of commonly used applications and an icon for any currently running applications. If the application is running, it will appear with a dot beneath the icon. To launch an application, simply click on the icon in the dash.

To find an application not included on the dash, one option is to select the right-most grid icon (the square comprising nine dots) to display a browsable list of applications as shown in Figure 6-4:

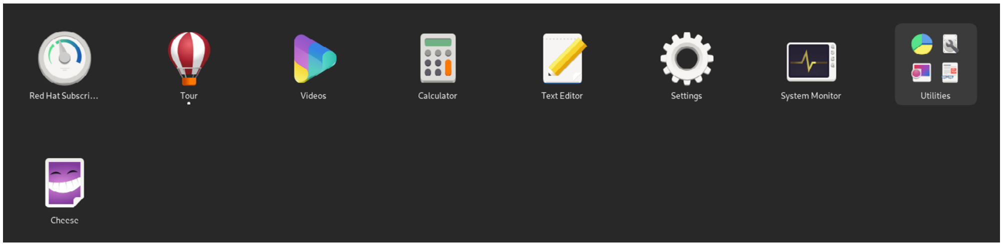
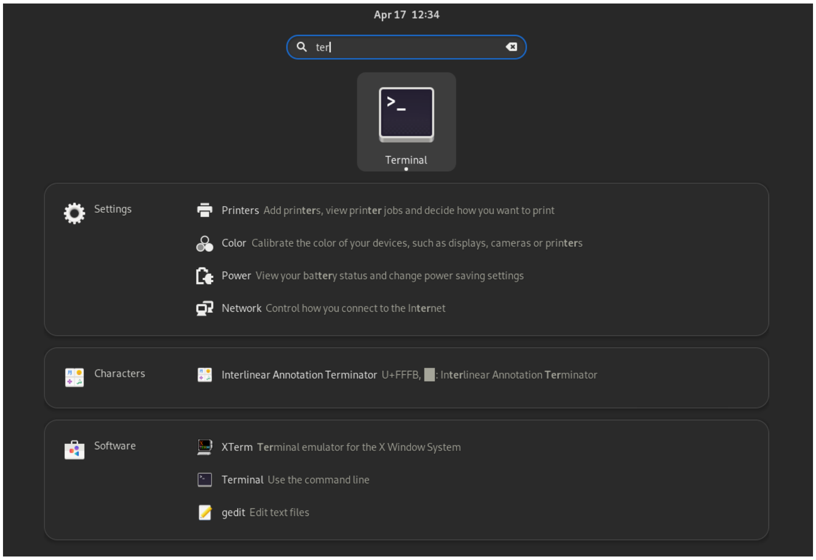


Figure 6-4

It is also important to be aware that some entries in the list are folders holding additional applications. Items in the grid may be repositioned by dragging and dropping the icons. To add an app to a folder, locate it in the grid and drag it into the folder. Moreover, dragging one icon and dropping it onto another will create a new folder containing both apps and to which other apps may be added.

An alternative to browsing the applications is to perform a search using the search bar (marked C in Figure 6-3 above) and shown in action in the figure below:



Apr 17 12:34

ter

Terminal

Settings

Printers Add printers, view printer jobs and decide how you want to print

Color Calibrate the color of your devices, such as displays, cameras or printers

Power View your battery status and change power saving settings

Network Control how you connect to the Internet

Characters

Interlinear Annotation Terminator U+FFFB, ￻: Interlinear Annotation Terminator

Software

XTerm Terminal emulator for the X Window System

Terminal Use the command line

gedit Edit text files

Figure 6-5

The list of possible matches will be refined as the text is typed into the search box.

To add an application to the dash for more convenient access, locate the icon for the application, right-click on it, and select the Add to Favorites menu option:

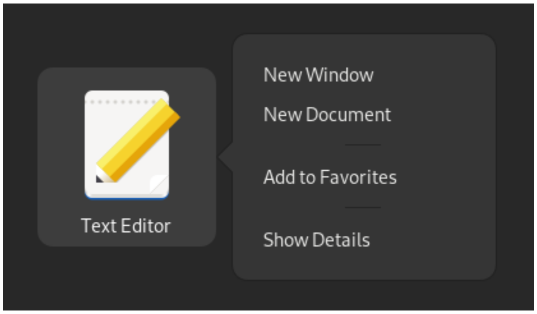


Figure 6-6

To remove an app from the dash, right-click on the icon in the dash and select Remove from Favorites. Alternatively, drag and drop icons to and from the dash to add and remove them.

6.4 Managing Windows

As with other desktop environments, applications run on GNOME in windows. When multiple application windows are open, the Super + Tab keyboard shortcut will display the switcher panel (Figure 6-7), allowing a different window to be chosen as the currently active window (the Super key is either the Windows key or, in the case of a Mac keyboard, the Cmd key):

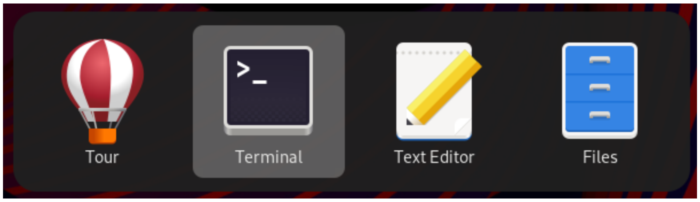


Figure 6-7

If a single application has more than one window open, the switcher will display those windows in a second panel so that a specific window can be selected:

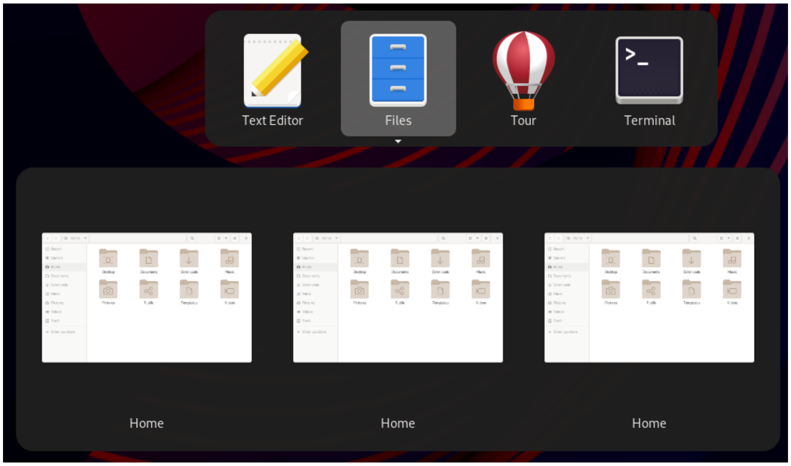


Figure 6-8

To cycle backward through the icons in the switcher, use the Shift+Tab keyboard shortcut.

To maximize a window so it fills the entire screen, click the title bar and drag the window to the top of the screen. To return the window to its original size, click on the title bar and drag it downwards. Alternatively, double-click on the title bar to toggle between window sizes. Similarly, dragging a window to the left or right side of the screen will cause the window to fill that half of the screen.

6.5 Using Workspaces

The area of the screen where the application windows appear is referred to as the workspace, and GNOME 40 allows multiple workspaces to be configured. The GNOME desktop will launch with two workspaces you can switch between using the Super+Alt+Left and Super+Alt+Right key combinations. When a window is added to a blank workspace, another blank workspace is added to the workspace panel, allowing additional workspaces to be created.

Thumbnails of the current workspaces appear beneath the search field in the Activities Overview as shown in Figure 6-9:

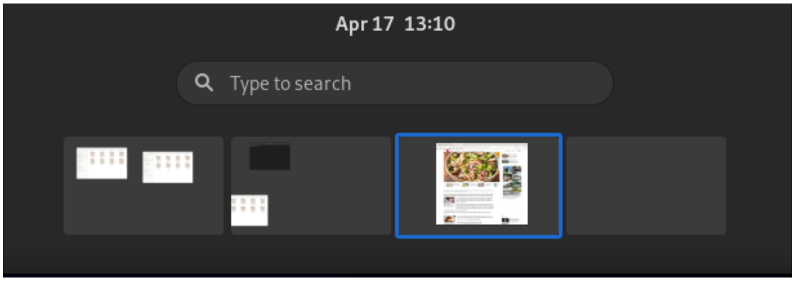
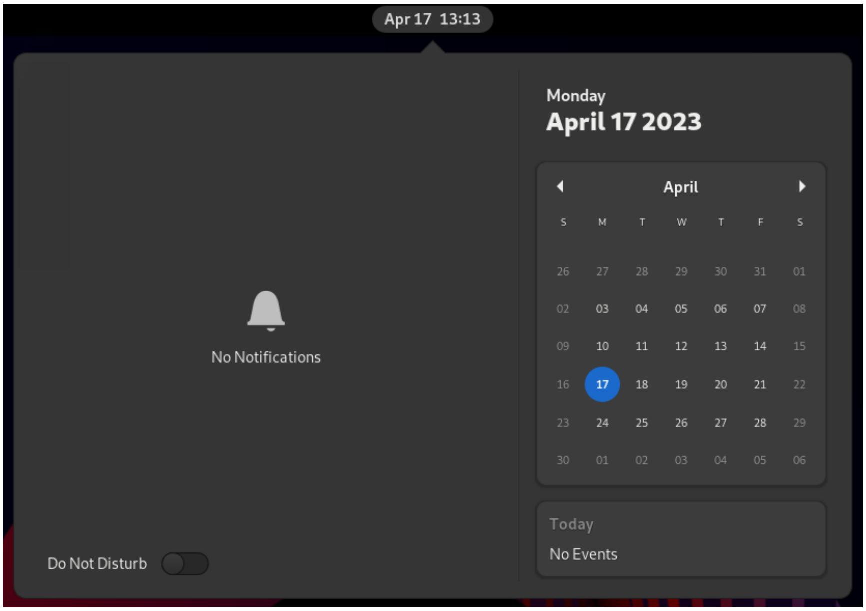


Figure 6-9

To switch to a different workspace, select the corresponding thumbnail. To move a window from one workspace to another, display the workspaces panel and drag and drop the application window (either the actual window from the current workspace or the thumbnail window in the workspaces panel) onto the destination workspace. To remove a workspace, close all its windows or move them to another workspace.

6.6 Calendar and Notifications

When the system needs to notify you of an event (such as the availability of system or application updates), a popup panel will appear at the top of the workspace. In addition, access to the calendar and any previous notifications is available by clicking on the day and time in the top bar, as shown in Figure 6-10:



Monday
April 17 2023

April

S	M	T	W	T	F	S
26	27	28	29	30	31	01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	01	02	03	04	05	06

Today
No Events

Figure 6-10

6.7 GNOME Desktop Settings

To access the Settings application, click on the row of icons on the right-hand side of the top bar and select Settings as shown in Figure 6-11:

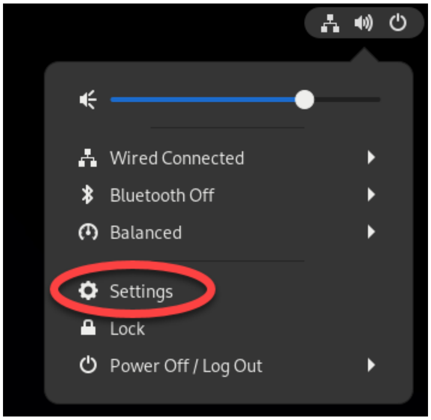
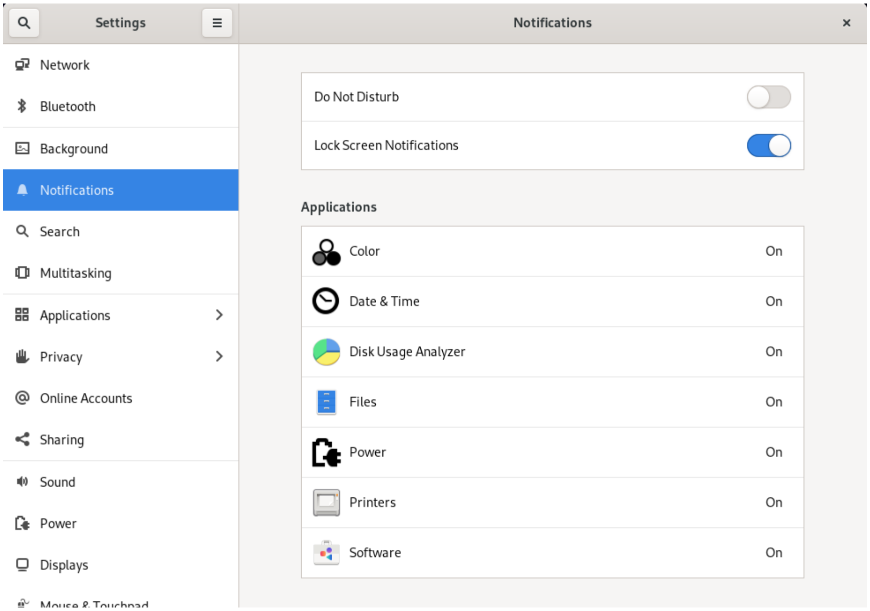


Figure 6-11

The Settings application provides many options, such as Ethernet and WiFi connections, screen background customization options, screen locking and power management controls, and language preferences. To explore the settings available in each category, select an option from the left-hand panel in the Settings window:



Notifications

Do Not Disturb ○
Lock Screen Notifications ●

Applications

Color	On
Date & Time	On
Disk Usage Analyzer	On
Files	On
Power	On
Printers	On
Software	On

Figure 6-12

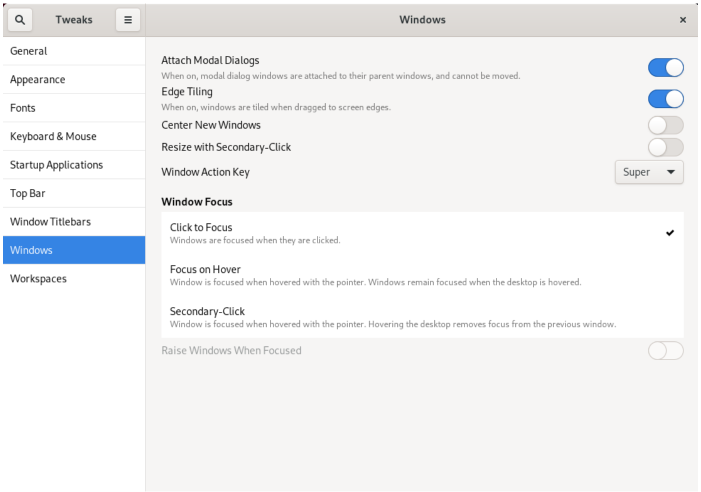
The menu shown in Figure 6-11 above also includes options to switch users, adjust audio volume, change to a different WiFi network, and log out, restart, or power off the system.

6.8 Beyond Basic Customization

The GNOME 40 desktop is, by design, a clean and uncluttered environment with minimal customization options. However, it is possible to make additional changes to the desktop. The GNOME Project has developed a tool called GNOME Tweaks for this purpose. Use the following commands to install and run this tool:

```
# dnf install gnome-tweaks
$ gnome-tweaks
```

Once GNOME Tweaks has loaded, the interface shown in Figure 6-13 will appear:



Attach Modal Dialogs
When on, modal dialog windows are attached to their parent windows, and cannot be moved.

Edge Tiling
When on, windows are tiled when dragged to screen edges.

Center New Windows

Resize with Secondary-Click

Window Action Key: Super

Window Focus

Click to Focus ✓
Windows are focused when they are clicked.

Focus on Hover
Window is focused when hovered with the pointer. Windows remain focused when the desktop is hovered.

Secondary-Click
Window is focused when hovered with the pointer. Hovering the desktop removes focus from the previous window.

Raise Windows When Focused

Figure 6-13

A wide range of options for customizing the desktop is now available. Too many to cover in this chapter, so experiment with these settings before proceeding to the next chapter.

6.9 Summary

RHEL 9 includes the GNOME 40 desktop environment, which may either be included during the initial installation or installed later using the *dnf group* package installation feature. Unlike most other desktop environments, GNOME 40 is intended to provide a clean and easy-to-use windowing user interface. Key areas of the GNOME 40 desktop include the top bar, Activities overview, and dash. In addition, GNOME 40 supports multiple workspaces keeping running applications organized and the screen uncluttered. Various configuration options are available within the Settings app, including desktop background settings, audio, network configuration, and WiFi network selection.

Chapter 10

10. Understanding RHEL 9 Software Installation and Management

It is doubtful that a newly installed RHEL 9 system will contain all of the software packages to perform the tasks for which it is intended. Even once all the required software has been installed, it is almost certain that newer versions of many of those packages will be released during the system's lifespan. In some cases, you will need to ensure that these latest package releases are installed on the system to ensure that any bugs are fixed. In other cases, however, an older version of a particular software package may need to be kept on the system for compatibility with other software.

This chapter introduces the basic concepts of software management on RHEL 9, explains how these issues are addressed, explains repositories, software packages, and the Red Hat Application Stream (AppStream), and explores how to list, install and remove the software packages that make up a functioning RHEL 9 system.

10.1 Repositories

Linux is essentially comprised of a set of base packages that provide the core functionality of the operating system together with a range of other packages and modules that add functionality and features on top of the base operating system.

Several packages will be installed when RHEL 9 is first installed, depending on the software options selected during the installation phase. Once the system is up and running, however, additional software can be installed as needed. Typically, all software that is part of RHEL 9 (in other words, software that a third-party vendor does not provide) is downloaded and installed on the system using the *dnf* command. As we have seen in earlier chapters, this typically consists of a command similar to the following being issued at the command prompt:

```
# dnf install httpd
```

When such a command is issued, the requested software is downloaded from a remote repository and installed on the local system. By default, RHEL 9 is configured to download software from repositories named BaseOS and AppStream. For example, running the following command will provide a list of the repositories the system is currently configured to use when downloading software:

```
# dnf repolist
Updating Subscription Management repositories.
repo id                              repo name
rhel-9-for-x86_64-appstream-rpms     Red Hat Enterprise Linux 9 for x86_64 - 
AppStream (RPMs)
rhel-9-for-x86_64-baseos-rpms        Red Hat Enterprise Linux 9 for x86_64 -
```

```
BaseOS (RPMs)
```

The above example shows that repositories for both AppStream and BaseOS are enabled on the system. Remember that repositories may be added for third-party software.

Additional repositories may be added to the system by placing entries in the */etc/dnf/dnf.conf* file, or by adding *.repo* files to the */etc/yum.repos.d/* directory. Alternatively, the repository may be added using the *dnf config-manager* tool, passing the URL of the *.repo* file for the repository as a command-line argument:

```
# dnf config-manager --add-repo https://url/of/repo/file
```

10.2 The BaseOS Repository

The BaseOS repository contains the packages that make up the core functionality of the operating system. These software elements are downloaded in the form of Red Hat Package Manager (RPM) package files and then installed on the system. A typical RHEL 9 system will have around 1500 RPM packages installed. To see a list of all the RPM packages currently installed on the system, run the *rpm* command as follows:

```
# rpm -qa | more
qemu-kvm-block-rbd-2.12.0-41.el8+2104+3e32e6f8.x86_64
kyotocabinet-libs-1.2.76-17.el8.x86_64
cyrus-sasl-scram-2.1.27-0.3rc7.el8.x86_64
curl-7.61.1-5.el8.x86_64
.
.
```

A list of packages available for installation from the BaseOS repository is available by running the *dnf* command as follows:

```
# dnf list
```

To obtain a list of packages that match a search string, use *dnf* as follows:

```
# dnf search "search string"
```

It is also possible to identify which package contains a specific file:

```
# dnf provides filename
```

For example:

```
Updating Subscription Management repositories.
httpd-2.4.51-7.el9_0.x86_64 : Apache HTTP Server
Repo        : rhel-9-for-x86_64-appstream-rpms
Matched from:
Filename    : /etc/httpd/conf/httpd.conf

httpd-core-2.4.53-7.el9.x86_64 : httpd minimal core
Repo        : rhel-9-for-x86_64-appstream-rpms
Matched from:
Filename    : /etc/httpd/conf/httpd.conf
```

```
httpd-core-2.4.53-7.el9_1.1.x86_64 : httpd minimal core
Repo        : rhel-9-for-x86_64-appstream-rpms
Matched from:
Filename    : /etc/httpd/conf/httpd.conf
```

To install a package, run the following command:

```
# dnf install packagename
```

Similarly, to delete a package:

```
# dnf delete packagename
```

When a newer version of a package is made available, it will be downloaded and installed when the system is next updated, typically via the *dnf* command:

```
# dnf update
```

Any updated packages will replace the older version currently installed on the system. While this is generally the ideal situation when working with base operating system packages, this is not necessarily the desired behavior when dealing with other packages, such as programming environments or development libraries, where upgrading to a new version may cause compatibility issues with other packages installed on the system. This issue is addressed by the AppStream repository.

10.3 The AppStream Repository

The AppStream repository manages software in terms of *packages*, *modules, streams,* and *profiles*. AppStream packages are, once again, RPM packages, as outlined in the previous section describing BaseOS. AppStream modules, on the other hand, are groups of packages that belong together or for which dependencies exist (for example, the group of packages that would need to be installed together when building a web server). Each module can have multiple streams, where each *module stream* represents a different version of the software module.

Consider, for example, a RHEL 9 system hosting a website that depends on version 7.2 of the PHP scripting language. The server still needs to receive any updates to PHP 7.2 to benefit from patches and bug fixes but is not compatible with the latest version of PHP (version 8.1). Before the introduction of AppStream, it would have been difficult to continue receiving version 7.2 updates when newer versions have been released.

To address this issue, the RHEL software management tools can use the AppStream repository to subscribe only to a specific stream for a specific module (in this case, the version 7.2 stream of the PHP module).

In addition to streams, modules may also be sub-classed by *module profile*. Module profiles provide different configurations of packages that make up a module to be installed, dependent on the requirements of the system. The nodejs JavaScript runtime environment module, for example, is available for installation using either the *development* or *minimal* profiles. On a system where development is taking place using nodejs, the development profile would most likely be used.

When the software developed using nodejs is deployed, the minimal system containing just the runtime might be installed instead.

To view the list of modules available for installation, use the *dnf* command as follows:

```
# dnf module list
Updating Subscription Management repositories.
Last metadata expiration check: 1:27:42 ago on Mon 27 Mar 2023 11:02:15 AM EDT.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
Name     Stream    Profiles                                 Summary
maven    3.8       common [d]                               Java project
management and project comprehension tool
nodejs   18        common [d], development, minimal, s2i    Javascript runtime
php      8.1       common [d], devel, minimal               PHP scripting language
.
.
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

The first column in the list is the name of the module, and the second is the stream name (which is typically the version of the module). The letter after the stream name ([d]) indicates whether the stream is the default (i.e., this is the stream that will be used for installation if no specific stream is referenced) or if it has been enabled for use when performing installations. The third column lists the profiles available for the corresponding package together with an indication of whether the profile is the default, has been installed, or is disabled.

The *dnf* command to list information about a specific module is structured as follows:

```
# dnf module list modulename
```

The following output, for example, lists information about the PHP modules available for installation:

```
# dnf module list php
.
.
Name       Stream      Profiles                        Summary
php        8.1 [d]     common [d], devel, minimal      PHP scripting language
```

Clearly, PHP version 8.1 will be installed on the system by default, and the module is available in common, development, and minimal profile configurations.

To install a module using the default stream and profile, the *dnf* command can be used with the following syntax:

```
# dnf install @modulename
```

For example:

```
# dnf install @php
```

Alternatively, a specific stream may be specified from which to perform the installation:

```
# dnf install @modulename:stream
```

For example:

```
# dnf install @php:8.1
```

Finally, a profile may also be declared as follows:

```
# dnf install @modulename:stream/profile
```

For example, to install the minimal set of packages for PHP 8.1:

```
# dnf install @php:8.1/minimal
```

After installing the module using the above command, the PHP modules will now be listed as follows:

```
php        8.1 [e]      common [d], devel, minimal [i]   PHP scripting language
```

The "[e]" indicator in the stream column tells us that the 8.1 stream has been enabled, while the "[i]" in the profile column shows that the module has been installed using the minimal profile.

To enable a stream without installing a module, use *dnf* as follows:

```
# dnf module enable modulename
```

Similarly, a stream may be disabled as follows:

```
# dnf module disable modulename
```

To uninstall a module, use the following syntax:

```
# dnf module remove modulename
```

Additional information about a module may be identified using the following *dnf* command:

```
# dnf module info modulename
```

To find out which RPM packages make up the different profiles of a specific module and stream combination, use *dnf* as follows:

```
# dnf module info --profile modulename:stream
```

For example:

```
# dnf module info --profile php:8.1
Updating Subscription Management repositories.
Last metadata expiration check: 1:44:32 ago on Mon 27 Mar 2023 11:02:15 AM EDT.
Name    : php:8.1:9010020220706080036:9:x86_64
common  : php-cli
        : php-common
        : php-fpm
        : php-mbstring
        : php-xml
devel   : php-cli
        : php-common
        : php-devel
        : php-fpm
        : php-mbstring
        : php-pecl-zip
        : php-process
        : php-xml
minimal : php-cli
        : php-common
```

Finally, to switch from one module stream to another, run the installation command referencing the new stream as follows:

```
# dnf install @modulename:otherstream
```

This command will download the packages for the new stream and either upgrade or downgrade the existing packages to the specified version. Once this process is complete, resynchronize module packages for the new stream:

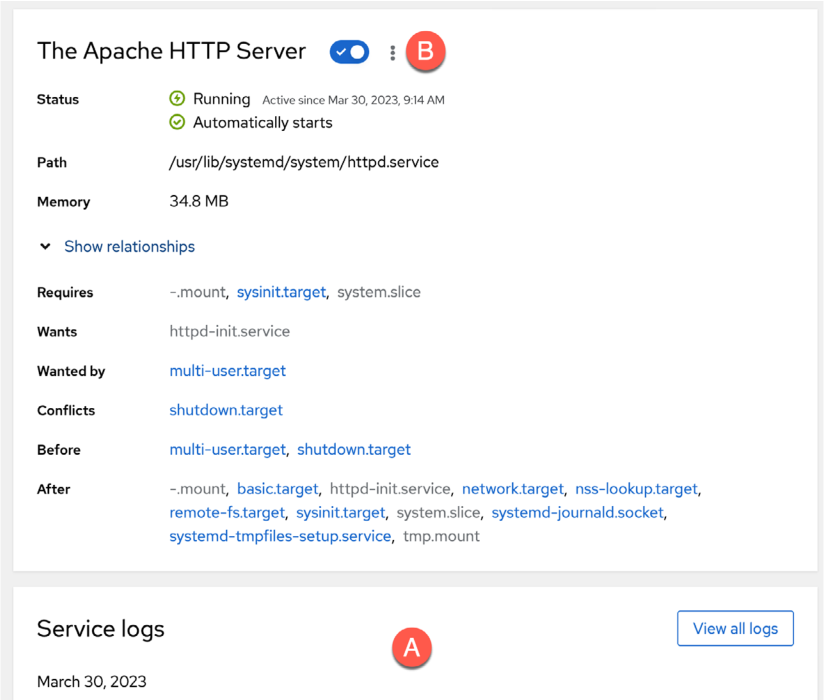
```
# dnf distro-sync
```

10.4 Summary

The RHEL 9 system comprises RPM format software packages that are downloaded and installed from the Red Hat BaseOS and AppStream repositories. Additional repositories can be added to the system for installation packages as needed.

The BaseOS repository contains the packages that implement the base core functionality of the operating system. The AppStream packages, on the other hand, provide additional features and functionality that will be installed selectively depending on the purpose for which the system is being configured. In a complex system of this nature, there can be a significant amount of package interdependency where part of the system may require a specific version of another software package to function correctly. AppStreams allow modules and profiles to be created that contain all of the dependent packages necessary to install a particular feature together at the correct version level. AppStreams also allow installed packages to receive updates to the current version without downloading the next major version, thereby avoiding disrupting the dependencies of other packages.

Selecting a unit from the list will display detailed information. Figure 11-4, for example, shows the detail screen for an httpd instance, including service logs (A) and a switch and menu (B) for performing tasks such as starting, stopping, enabling/disabling, and masking/unmasking the unit:



The Apache HTTP Server

Status: Running Active since Mar 30, 2023, 9:14 AM
Automatically starts

Path: /usr/lib/systemd/system/httpd.service

Memory: 34.8 MB

Show relationships

Requires: -.mount, sysinit.target, system.slice

Wants: httpd-init.service

Wanted by: multi-user.target

Conflicts: shutdown.target

Before: multi-user.target, shutdown.target

After: -.mount, basic.target, httpd-init.service, network.target, nss-lookup.target, remote-fs.target, sysinit.target, system.slice, systemd-journald.socket, systemd-tmpfiles-setup.service, tmp.mount

Service logs

March 30, 2023

Figure 11-4

11.9 Summary

A newly installed RHEL 9 system includes a base set of systemd units, many of which run in the background to provide much of the system's functionality. These units are categorized by type, the most common being targets and services. A target unit is a group of other units to be started collectively. The system has a default target unit that defines the other units to be started up each time the system boots. The most common targets are those which boot the system to either multi-user or graphical mode. In addition, the *systemctl* command-line tool provides a range of options for performing systemd unit configuration tasks, many of which are also available through the Cockpit web-based interface.

```
# nmcli con mod static_ip connection.permissions user:root,john,caitlyn
```

Once NetworkManager has reloaded the connection profiles, the *static_ip* connection will only be active and accessible to other users when at least one designated user is logged in to an active session on the system. As soon as the last of these users logs out, the connection will go down and remain inactive until one of the users signs back in.

In addition, only users with permission can change the connection status or configuration.

12.7 Summary

The NetworkManager service handles network management on RHEL 9. NetworkManager views a network as consisting of network interface devices and connections. A network device can be a physical Ethernet or WiFi device or a virtual device used by a virtual machine guest. Connections represent the network to which the devices connect and are configured by connection profiles. A configuration profile will, among other settings, define whether the connection has a static or dynamic IP address, the IP address of any gateway used by the network, and whether or not the connection should be established automatically each time the system starts up.

NetworkManager can be administered using several tools, including the *nmcli* and *nmtui* command-line tools, the *nm-connection-editor* graphical tool, and the network settings section of the Cockpit web interface. In general, the nmcli command-line tool provides the most features and flexibility.

Chapter 13

13. RHEL 9 Firewall Basics

A firewall is a vital component in protecting an individual computer system or network of computers from external attacks (typically from an internet connection). Any computer connected directly to an internet connection should ideally run a firewall to protect against malicious activity. Similarly, any internal network must have some form of firewall between it and an external internet connection.

RHEL 9 is supplied with powerful firewall technology known as *iptables* built-in. Entire books can, and indeed have, been written about configuring iptables. If you would like to learn about iptables, we recommend the following:

https://www.linuxtopia.org/Linux_Firewall_iptables/index.html

This chapter will cover some basic concepts of firewalls, TCP/IP ports, and services. Firewall configuration on RHEL 9 will be covered in the chapter entitled *"RHEL 9 Firewall Configuration with firewalld"*.

13.1 Understanding Ports and Services

The predominant network communications protocol in use these days is TCP/IP. It is the protocol used by the internet and, as such, has swept away most of the formerly popular protocols used for local area networks (LANs).

TCP/IP defines a total of 65,535 ports, of which 1023 are considered *well-known ports*. It is essential to understand that these are not physical ports into which network cables are connected but rather virtual ports on each network connection which can be used by applications and services to communicate over a TCP/IP network connection. In reality, the number of ports used by popular network clients and services comprises an even smaller subset of the well-known group of ports.

An operating system can provide several different TCP/IP services. A comprehensive list of such services is provided in the table at the end of this chapter. Still, such services include HTTPS for running a secure web server, FTP for allowing file transfers, SSH for providing secure remote login access and file transfer, and SMTP for transporting email messages. Each service is, in turn, assigned to a standard TCP/IP port. For example, HTTPS is assigned to port 443, while SSH communication occurs on port 22.

13.2 Securing Ports and Services

A large part of securing servers involves defining roles and, based on the roles, defining which services and ports should be enabled. For example, a server that acts solely as a web server should only run the HTTPS service (in addition to perhaps SSH for remote administration access). All other services should be disabled and, ideally, removed entirely from the operating system (thereby making it harder for an intruder to re-enable the service).

Securing a system involves removing any unnecessary services from the operating system and ensuring that the ports associated with the non-essential services are blocked using a firewall. The rules that define which ports are accessible and under what circumstances are determined using iptables.

Many operating systems are installed with several services installed and activated by default. Before installing a new operating system, the installation must be carefully planned. This planning involves deciding which services are not required and identifying which services have been installed and enabled by default. Deployment of new operating system installations should never be rushed. The fewer services and open ports available on a system, the smaller the surface area and opportunities for attackers.

13.3 RHEL 9 Services and iptables Rules

By default, a newly installed RHEL 9 system has no iptables rules defined to restrict access to ports. The following command may be executed in a terminal window to view the current iptables settings:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

As illustrated in the above output, no rules are currently defined. While this may appear to be an unsafe configuration, it is essential to remember that a newly installed RHEL 9 system also has few services running by default, making the ports useless to a potential attacker. For example, accessing a web server on a newly installed RHEL 9 system is impossible because no web server services are installed or running by default. Once services begin to be activated on the system, it will be important to establish a firewall strategy by defining iptables rules.

Several methods are available for defining iptables rules, including using command line tools and configuration files. For example, to block access to port 25 (used by the SMTP mail transfer protocol) from IP address 192.168.2.76, the following command could be issued in a terminal window:

```
# iptables -A INPUT -s 192.168.2.76 -p tcp --destination-port 25 -j DROP
```

If we now check the current rules, we will see that this one is currently listed:

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
```

```
DROP       tcp  --  192.168.2.76         anywhere             tcp dpt:smtp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

The rule may subsequently be removed as follows:

```
# iptables -D INPUT -s 192.168.2.76 -p tcp --destination-port 25 -j DROP
```

Given the complexity of iptables it is unsurprising that several user-friendly configuration tools have been created to ease the rule creation process. One such tool is the *firewall-cmd* command-line tool which will be covered in the chapter *"RHEL 9 Firewall Configuration with firewalld"*.

13.4 Well-Known Ports and Services

Before moving on to cover more complex firewall rules, it is first worth taking time to outline some of the key services that can be provided by an RHEL 9 system, together with the corresponding port numbers:

Port	Assignment	Description
20	FTP	**File Transfer Protocol (Data)** - The File Transfer protocol provides a mechanism for transferring specific files between network-connected computer systems. The transfer is typically performed using the ftp client. Most modern web browsers can also browse and download files on a remote FTP server. FTP uses TCP (rather than UDP) to transfer files, which is considered a highly reliable transport mechanism. FTP does not encrypt data and is not considered a secure file transfer protocol. Secure Copy Protocol (SCP) and Secure File Transfer Protocol (SFTP) are strongly recommended in place of FTP.
21	FTP	**File Transfer (Control)** - Traditionally, FTP has two ports assigned (port 20 and port 21). Port 20 was initially considered the data transfer port, while port 21 was assigned to communicate control information. However, in modern implementations, port 20 is rarely used, with all communication taking place on port 21.

22	SSH	**Secure Shell** - The Secure Shell provides a safe, encrypted, remote login session to a host over a TCP/IP network. The original mechanism for remote access was the Telnet protocol. However, because Telnet transmits data in plain text, its use is strongly discouraged in favor of the secure shell, which encrypts all communications, including login and password credentials. SSH also provides the mechanism by which files can be securely transferred using the Secure Copy Protocol (SCP) and is also the basis for the Secure File Transfer Protocol (SFTP). SSH also replaces both the rsh and rlogin clients.
23	Telnet	**Telnet** - Telnet is a terminal emulation protocol that can log into a remote system over a TCP/IP connection. The access is text-based, allowing the user to type into a command prompt on the remote host, and text displayed by the remote host is displayed on the local Telnet client. Telnet encrypts neither the password nor the text communicated between the client and server. As such, the use of telnet is strongly discouraged. Most modern systems will have port 23 closed and the telnet service disabled to prevent its use. SSH should be used in place of Telnet.
25	SMTP	**Simple Mail Transfer Protocol** - SMTP defines the mechanism by which email messages are sent from one network host to another. SMTP is a straightforward protocol requiring the mail service to always be available at the receiving host. Typically the receiving host will store incoming messages in a spool for subsequent access by the recipient using the POP3 or IMAP protocols. In addition, SMTP uses the TCP transport protocol to ensure error-free message delivery.
53	DNS	**Domain Name Server** - The service used by TCP/IP networks to translate host names and Fully Qualified Domain Names (FQDN) to IP addresses.
69	TFTP	**Trivial File Transfer Protocol** - TFTP is a stripped-down version of the File Transfer Protocol (FTP). It has a reduced command set and lacks authentication. The most significant feature of TFTP is that it uses UDP to transfer data. This results in high-speed transfer speeds but, consequently, lacks data reliability. TFTP is typically used in network-based booting for diskless workstations.

80	HTTP	**Hypertext Text Transfer Protocol** - HTTP is used to download text, graphics, and multimedia from a web server to a web browser. It defines the command and control mechanism between the browser and server, defining client requests and server responses. HTTP is based on the TCP transport protocol and, as such, is a connection-oriented protocol.
110	POP3	**Post Office Protocol** - The POP3 protocol is a mechanism for storing and retrieving incoming email messages from a server. In most corporate environments, incoming email is stored on an email server and then downloaded to an email client running on the user's desktop or laptop when the user checks email. However, POP3 downloads all new messages to the client and does not allow the user to choose which messages to download, view headers, or download only parts of messages. For this reason, the IMAP protocol is increasingly being used in place of POP3.
119	NNTP	**Network News Transfer Protocol** - The protocol responsible for posting and retrieving messages to and from Usenet News Servers (i.e., newsgroups and discussion forums hosted on remote servers). NNTP operates at the Application layer of the OSI stack and uses TCP to ensure error-free message retrieval and transmission.
123	NTP	**Network Time Protocol** - A protocol designed to synchronize computer clocks with an external time source. Using this protocol, an operating system or application can request the current time from a remote NTP server. The remote NTP server is usually based on the time provided by a nuclear clock. NTP is useful for ensuring that all systems in a network are set to the same, accurate time of day. This is of particular importance in security situations when, for example, the time a file was accessed or modified on a client or server is in question.
143	IMAP4	**Internet Message Access Protocol, Version 4** - IMAP4 is an advanced and secure email retrieval protocol. IMAP is similar to POP3, allowing users to access email messages stored on an email server. However, IMAP includes many additional features, such as the ability to selectively download messages, view message headers, search messages, and download part of a message. In addition, IMAP4 uses authentication and fully supports Kerberos authentication.

161	SNMP	**Simple Network Management Protocol** - Provides a mechanism whereby network administrators can collect information about network devices (such as hubs, bridges, routers, and switches). The SNMP protocol enables agents running on network devices to communicate their status to a central manager and, in turn, allows the manager to send new configuration parameters to the device agent. The agents can further be configured to notify the manager when certain events, known as traps, occur. SNMP uses UDP to send and receive data.
443	HTTPS	**Hypertext Transfer Protocol Secure** - The standard HTTP (non-secure) protocol transfers data in clear text (i.e., with no encryption and visible to anyone who might intercept the traffic). While this is acceptable for most web browsing purposes, it poses a severe security risk when confidential information such as credit card details needs to be transmitted from the browser to the web server. HTTPS addresses this using the Secure Sockets Layer (SSL) to send encrypted data between the client and server.
2049	NFS	**Network File System** - Originally developed by Sun Microsystems and subsequently widely adopted throughout the industry, NFS allows a file system on a remote system to be accessed over the network by another system as if the file system were on a local disk drive. NFS is widely used on UNIX and LINUX-based systems. Later versions of Microsoft Windows can also access NFS-shared file systems on UNIX and LINUX-based systems.

Table 13-1

13.5 Summary

A newly installed RHEL 9 system is generally considered secure due to the absence of services running on the system ports. Once the system begins to be configured for use, however, it is important to ensure that it is protected from attack through the implementation of a firewall. When configuring firewalls, it is important to understand the various ports and the corresponding services.

Several firewall options are available, the most basic being the command-line configuration of the iptables firewall interface. More intuitive and advanced options are available via firewalld, which will be covered in the next chapter.

Chapter 23

23. Creating KVM Virtual Machines on RHEL 9 using virt-manager

The previous chapter explored how to create KVM virtual machines on RHEL 9 using the Cockpit web tool. With the caveat that *virt-manager* may one day be discontinued once the Virtual Machines Cockpit extension is fully implemented, this chapter will cover using this tool to create new virtual machines.

23.1 Starting the Virtual Machine Manager

If you have not already done so, install the *virt-manager* package as follows:

```
# dnf install virt-manager
```

Next, launch Virtual Machine Manager from the command line in a terminal window by running *virt-manager*. Once loaded, the virtual machine manager will prompt for the password of the currently active user before displaying the following screen:

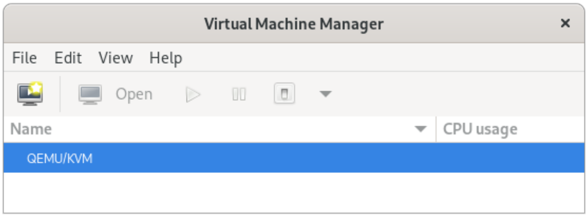
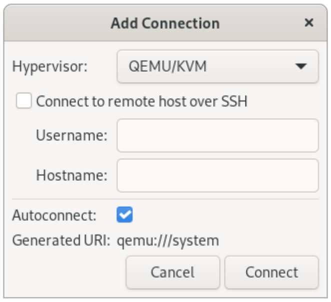


Figure 23-1

The main screen lists the current virtual machines running on the system. By default, the manager should be connected to the system libvirtd instance. If it is not, connect to the host system by right-clicking on the entry in the list and selecting *Connect* from the popup menu. To manage session-based virtual machines, select the *File -> Add Connection...* menu option to display the dialog shown in Figure 23-2:



Hypervisor: QEMU/KVM

□ Connect to remote host over SSH

Username:

Hostname:

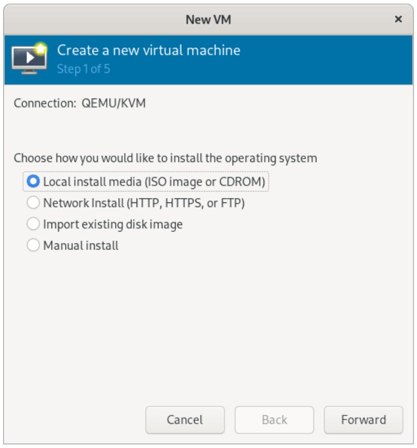
Autoconnect: ☑

Generated URI: qemu:///system

Figure 23-2

Select QEMU/KVM user session from the Hypervisor menu and click the Connect button within this dialog. On returning to the main *virt-manager* screen, the QEMU/KVM session should now be listed as shown in Figure 23-1 above.

To create a virtual system, click on the new virtual machine button (the far left button in the toolbar) to display the first screen of the New VM wizard. In the Name field, enter a suitably descriptive name for the virtual system. On this screen, also select the location of the media from which the guest operating system will be installed. This can either be a CD or DVD drive, an ISO image file accessible to the local host, a network install using HTTP, FTP, NFS, or PXE, or the disk image from an existing virtual machine:


New VM

Create a new virtual machine

Step 1 of 5

Connection: QEMU/KVM

Choose how you would like to install the operating system

● Local install media (ISO image or CDROM)

○ Network Install (HTTP, HTTPS, or FTP)

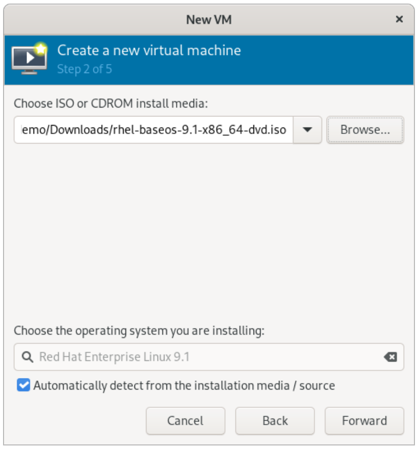
○ Import existing disk image

○ Manual install

Figure 23-3

23.2 Configuring the KVM Virtual System

Clicking Forward will display a screen seeking additional information about the installation process. The displayed screen and information required will depend on selections made on the initial screen. For example, if a CD, DVD, or ISO is selected, this screen will ask for the specific location of the ISO file or physical media device. This screen also attempts to identify the type and version of the guest operating system (for example, the Windows version or Linux distribution) based on the specified installation media. If it is unable to do so, uncheck the *Automatically detect from installation media/source* option, type in the first few characters of the operating system name, and select an option from the list of possible matches:



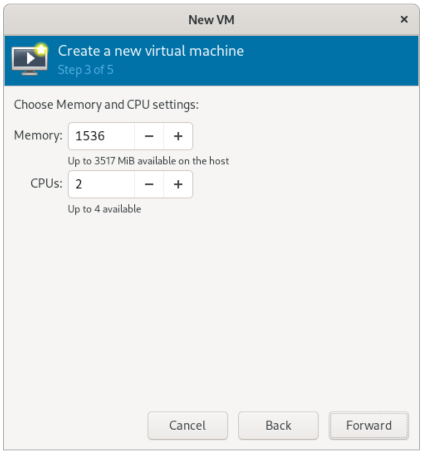
Choose ISO or CDROM install media: emo/Downloads/rhel-baseos-9.1-x86_64-dvd.iso

Choose the operating system you are installing: Red Hat Enterprise Linux 9.1

☑ Automatically detect from the installation media / source

Figure 23-4

Once these settings are complete, click the Forward button to configure CPU and memory settings. The optimal settings will depend on the number of CPUs and amount of physical memory present in the host, together with the requirements of other applications and virtual machines that will run in parallel with the new virtual machine:



Choose Memory and CPU settings:

Memory: 1536 − +

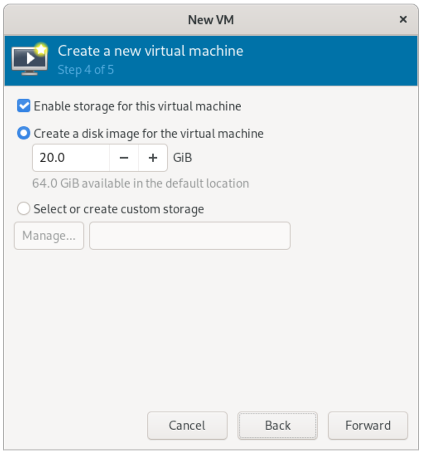
Up to 3517 MiB available on the host

CPUs: 2 − +

Up to 4 available

Figure 23-5

On the next screen, options are available to create an image disk of a specified size, select a pre-existing volume, or create a storage volume of a specified format (raw, vmdk, ISO, etc.). Unless you have a specific need to use a particular format (for example, you might need to use vmdk to migrate to a VMware-based virtualization environment at a later date) or need to use a dedicated disk or partition, it is generally adequate to specify a size on this screen:



New VM

Create a new virtual machine
Step 4 of 5

☑ Enable storage for this virtual machine

● Create a disk image for the virtual machine

20.0 GiB

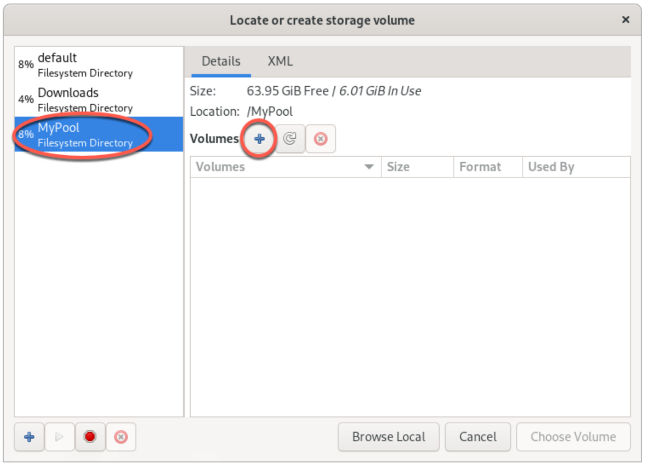
64.0 GiB available in the default location

○ Select or create custom storage

Figure 23-6

If the default settings are used here, the virtual machine will use a storage volume within the default storage pool for the virtual disk drive. To use the custom "MyPool" storage pool created earlier in the chapter, enable the *Select or create custom storage* option before clicking the *Manage...* button.

In the storage volume dialog, select the MyPool entry in the left-hand panel, followed by the + button in the main panel to create a new storage volume:



Locate or create storage volume

8% default
Filesystem Directory

4% Downloads
Filesystem Directory

8% MyPool
Filesystem Directory

Size: 63.95 GiB Free / *6.01 GiB In Use*

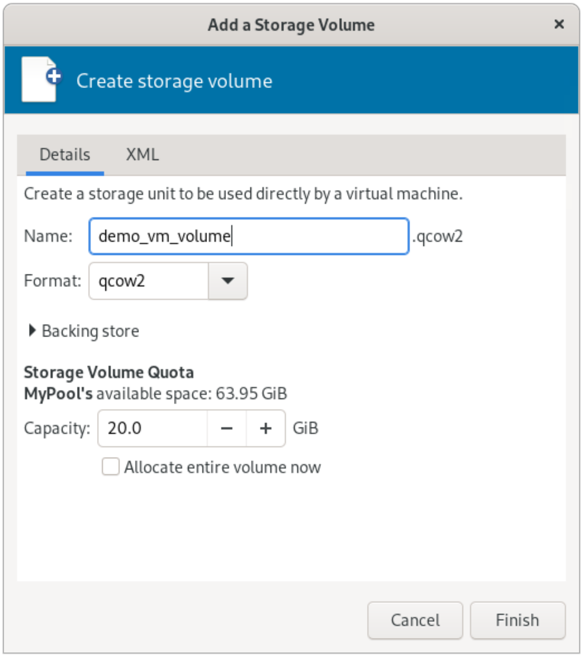
Location: /MyPool

Volumes	Size	Format	Used By

Figure 23-7

Note that the + button in the bottom left-hand corner of the dialog may also be used to create new storage pools as an alternative to using the Cockpit interface.

In the configuration screen (Figure 23-8), name the storage volume, select the volume size, and click on the *Finish* button to create the volume and assign it to the virtual machine:



Add a Storage Volume

Create storage volume

Details XML

Create a storage unit to be used directly by a virtual machine.

Name: demo_vm_volume .qcow2

Format: qcow2

Backing store

Storage Volume Quota

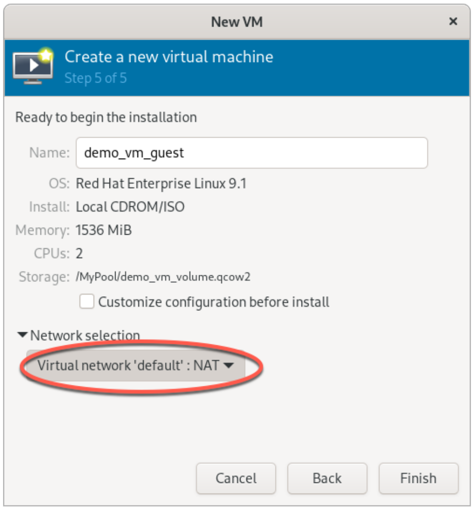
MyPool's available space: 63.95 GiB

Capacity: 20.0 GiB

□ Allocate entire volume now

Figure 23-8

Once these settings are configured, select the new volume and click the *Choose Volume* button. Then, click the Forward button once more. The final screen displays a summary of the configuration. Review the information displayed. Advanced options are also available to change the virtual network configuration for the guest, as shown in Figure 23-9:



New VM

Create a new virtual machine

Step 5 of 5

Ready to begin the installation

Name: demo_vm_guest

OS: Red Hat Enterprise Linux 9.1

Install: Local CDROM/ISO

Memory: 1536 MiB

CPUs: 2

Storage: /MyPool/demo_vm_volume.qcow2

□ Customize configuration before install

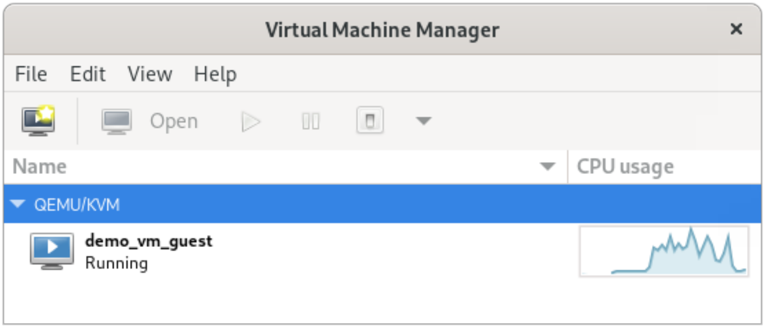
Network selection

Virtual network 'default' : NAT

Figure 23-9

23.3 Starting the KVM Virtual Machine

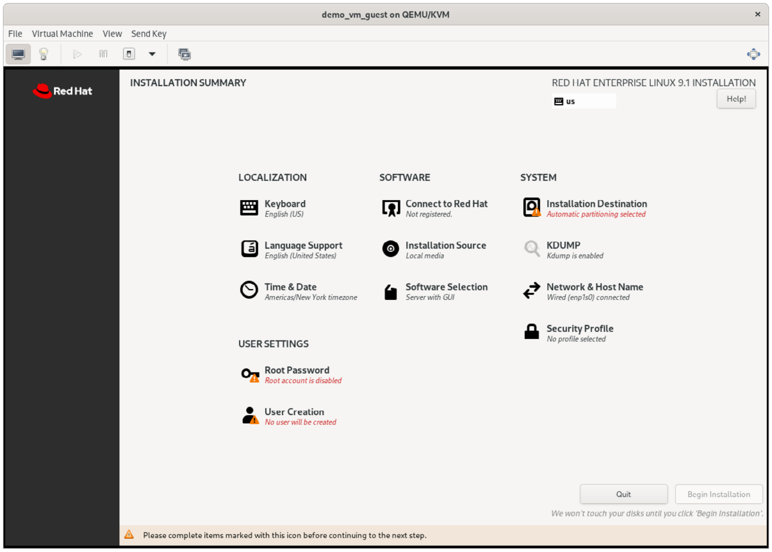
Click on the Finish button to begin the creation process. The virtualization manager will create the disk and configure the virtual machine before starting the guest system. Finally, the new virtual machine will appear in the main *virt-manager* window with the status set to *Running* as illustrated in Figure 23-10:



demo_vm_guest
Running

Figure 23-10

By default, the console for the virtual machine should appear in the virtual machine viewer window. To view the console of the running machine at any future time, ensure that it is selected in the virtual machine list and select the Open button from the toolbar. The virtual machine viewer should be ready for the installation process to begin:



INSTALLATION SUMMARY

RED HAT ENTERPRISE LINUX 9.1 INSTALLATION

us

LOCALIZATION

Keyboard
English (US)

Language Support
English (United States)

Time & Date
Americas/New York timezone

SOFTWARE

Connect to Red Hat
Not registered.

Installation Source
Local media

Software Selection
Server with GUI

SYSTEM

Installation Destination
Automatic partitioning selected

KDUMP
Kdump is enabled

Network & Host Name
Wired (enp1s0) connected

Security Profile
No profile selected

USER SETTINGS

Root Password
Root account is disabled

User Creation
No user will be created

We won't touch your disks until you click 'Begin Installation'.

⚠ Please complete items marked with this icon before continuing to the next step.

Figure 23-11

From now on, follow the installation instructions to install the guest OS in the KVM virtual machine.

23.4 Summary

There are several ways to create new KVM-based virtual machines on a RHEL 9 host system. This chapter uses the *virt-manager* graphical tool to create, configure, and run a guest operating system, including creating a new storage volume. With these basics covered, the next chapter will cover the creation of virtual machines from the command line.

Index