

CentOS Stream 9 Essentials

CentOS Stream 9 Essentials

ISBN-13: 978-1-951442-73-6

© 2023 Neil Smyth / Payload Media, Inc. All Rights Reserved.

This book is provided for personal use only. Unauthorized use, reproduction and/or distribution strictly prohibited. All rights reserved.

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	1
1.1 Superuser Conventions.....	1
1.2 Opening a Terminal Window	2
1.3 Editing Files	3
1.4 Feedback.....	4
1.5 Errata.....	4
2. A Brief History of CentOS Stream	5
2.1 What exactly is Linux?.....	5
2.2 UNIX Origins	5
2.3 Who Created Linux?	5
2.4 The Early Days of Red Hat	6
2.5 Red Hat Support	6
2.6 Open Source.....	7
2.7 The Fedora Project	7
2.8 CentOS Stream - The Free Alternative	7
2.9 Summary	7
3. Installing CentOS Stream 9 on a Clean Disk Drive.....	9
3.1 Obtaining the CentOS Stream 9 Installation Media.....	9
3.2 Writing the ISO Installation Image to a USB Drive.....	10
3.2.1 Linux.....	11
3.2.2 macOS	11
3.2.3 Windows/macOS.....	12
3.3 Installing CentOS Stream 9.....	14
3.4 Partitioning a Disk for CentOS Stream 9	18
3.5 Disk Encryption	21
3.6 User Settings	22
3.7 The Physical Installation	22
3.8 Final Configuration Steps.....	23
3.9 Installing Updates.....	23
3.10 Displaying Boot Messages.....	24
3.11 Summary	25
4. Dual Booting CentOS Stream 9 with Windows.....	27
4.1 Partition Resizing	27
4.2 Changing the Default Boot Option.....	29
4.3 Accessing the Windows Partition from CentOS 9.....	30
4.4 Summary	31

Table of Contents

5. Allocating Windows Disk Partitions to CentOS 9	33
5.1 Unmounting the Windows Partition.....	33
5.2 Deleting the Windows Partitions from the Disk	33
5.3 Formatting the Unallocated Disk Partition	36
5.4 Mounting the New Partition	36
5.5 Summary	37
6. A Guided Tour of the GNOME 40 Desktop	39
6.1 Installing the GNOME Desktop.....	39
6.2 An Overview of the GNOME 40 Desktop	39
6.3 Activity Overview.....	41
6.4 Managing Windows	43
6.5 Using Workspaces	44
6.6 Calendar and Notifications	44
6.7 GNOME Desktop Settings	45
6.8 Beyond Basic Customization	46
6.9 Installing GNOME Desktop Apps	47
6.10 Summary	47
7. An Overview of the Cockpit Web Interface	49
7.1 An Overview of Cockpit.....	49
7.2 Installing and Enabling Cockpit.....	50
7.3 Accessing Cockpit	50
7.4 Overview	51
7.5 Logs	52
7.6 Storage.....	53
7.7 Networking.....	53
7.8 Accounts	54
7.9 Services	54
7.10 Applications	55
7.11 Virtual Machines.....	55
7.12 Software Updates.....	56
7.13 Terminal	56
7.14 Connecting to Multiple Servers	57
7.15 Enabling Stored Metrics	58
7.16 Summary	59
8. Using the Bash Shell on CentOS Stream 9	61
8.1 What is a Shell?.....	61
8.2 Gaining Access to the Shell.....	61
8.3 Entering Commands at the Prompt.....	62
8.4 Getting Information about a Command.....	62
8.5 Bash Command-line Editing	63

8.6 Working with the Shell History	63
8.7 Filename Shorthand	64
8.8 Filename and Path Completion	64
8.9 Input and Output Redirection	64
8.10 Working with Pipes in the Bash Shell	65
8.11 Configuring Aliases	65
8.12 Environment Variables	66
8.13 Writing Shell Scripts	67
8.14 Summary	68
9. Managing CentOS Stream 9 Users and Groups	69
9.1 User Management from the Command-line	69
9.2 User Management with Cockpit	71
9.3 User Management using the Settings App	73
9.4 Summary	75
10. CentOS Stream 9 Software Installation and AppStreams	77
10.1 Repositories	77
10.2 The BaseOS Repository	78
10.3 The AppStream Repository	79
10.4 Summary	82
11. Managing CentOS Stream 9 systemd Units	83
11.1 Understanding CentOS 9 systemd Targets	83
11.2 Understanding CentOS 9 systemd Services	83
11.3 CentOS 9 systemd Target Descriptions	83
11.4 Identifying and Configuring the Default Target	85
11.5 Understanding systemd Units and Unit Types	86
11.6 Dynamically Changing the Current Target	87
11.7 Enabling, Disabling, and Masking systemd Units	87
11.8 Working with systemd Units in Cockpit	89
11.9 Summary	90
12. CentOS Stream 9 Network Management	91
12.1 An Introduction to NetworkManager	91
12.2 Installing and Enabling NetworkManager	92
12.3 Basic nmcli Commands	92
12.4 Working with Connection Profiles	96
12.5 Interactive Editing	98
12.6 Configuring NetworkManager Permissions	100
12.7 Summary	100
13. CentOS Stream 9 Firewall Basics	101
13.1 Understanding Ports and Services	101

Table of Contents

13.2 Securing Ports and Services	101
13.3 CentOS Stream 9 Services and iptables Rules	102
13.4 Well-Known Ports and Services	103
13.5 Summary	106
14. CentOS Stream 9 Firewall Configuration with firewalld	107
14.1 An Introduction to firewalld.....	107
14.1.1 Zones	107
14.1.2 Interfaces.....	109
14.1.3 Services.....	109
14.1.4 Ports.....	109
14.2 Checking firewalld Status	109
14.3 Configuring Firewall Rules with firewall-cmd	110
14.3.1 Identifying and Changing the Default Zone	110
14.3.2 Displaying Zone Information	110
14.3.3 Adding and Removing Zone Services.....	111
14.3.4 Working with Port-based Rules.....	112
14.3.5 Creating a New Zone.....	112
14.3.6 Changing Zone/Interface Assignments	112
14.3.7 Masquerading	113
14.3.8 Adding ICMP Rules	113
14.3.9 Implementing Port Forwarding.....	113
14.4 Managing firewalld from the Cockpit Interface.....	114
14.5 Managing firewalld using firewall-config	115
14.6 Summary	116
15. Configuring SSH Key-based Authentication on CentOS Stream 9	117
15.1 An Overview of Secure Shell (SSH).....	117
15.2 SSH Key-based Authentication	117
15.3 Setting Up Key-based Authentication	118
15.4 Installing and Starting the SSH Service.....	118
15.5 SSH Key-based Authentication from Linux and macOS Clients.....	118
15.6 Managing Multiple Keys	120
15.7 SSH Key-based Authentication from Windows Clients	121
15.8 SSH Key-based Authentication using PuTTY	123
15.9 Generating a Private Key with PuTTYgen	124
15.10 Summary	125
16. CentOS Stream 9 Remote Desktop Access with VNC	127
16.1 Secure and Insecure Remote Desktop Access	127
16.2 Installing the GNOME Desktop Environment	127
16.3 Installing VNC on CentOS Stream 9.....	129
16.4 Configuring the VNC Server.....	129

16.5 Connecting to a VNC Server	131
16.6 Establishing a Secure Remote Desktop Session	131
16.7 Establishing a Secure Tunnel on Windows using PuTTY	133
16.8 Shutting Down a Desktop Session	134
16.9 Troubleshooting a VNC Connection	134
16.10 Summary	135
17. Displaying CentOS Stream 9 Applications Remotely (X11 Forwarding)	137
17.1 Requirements for Remotely Displaying CentOS Stream 9 Applications	137
17.2 Displaying a CentOS 9 Application Remotely	138
17.3 Trusted X11 Forwarding	138
17.4 Compressed X11 Forwarding	138
17.5 Displaying Remote CentOS 9 Apps on Windows.....	138
17.6 Summary	141
18. Using NFS on CentOS Stream 9 to Share Files with Remote Systems.....	143
18.1 Ensuring NFS Services are running on CentOS Stream 9	143
18.2 Configuring the CentOS 9 Firewall to Allow NFS Traffic	144
18.3 Specifying the Folders to be Shared	144
18.4 Accessing Shared Folders	145
18.5 Mounting an NFS Filesystem on System Startup	145
18.6 Unmounting an NFS Mount Point	145
18.7 Accessing NFS Filesystems in Cockpit.....	145
18.8 Summary	147
19. Sharing Files between CentOS Stream 9 and Windows with Samba	149
19.1 Accessing Windows Resources from the GNOME Desktop	149
19.2 Samba and Samba Client.....	150
19.3 Installing Samba on CentOS 9.....	150
19.4 Configuring the CentOS 9 Firewall to Enable Samba	150
19.5 Configuring the smb.conf File.....	151
19.5.1 Configuring the [global] Section	151
19.5.2 Configuring a Shared Resource	151
19.5.3 Removing Unnecessary Shares	152
19.6 Configuring SELinux for Samba	152
19.7 Creating a Samba User	154
19.8 Testing the smb.conf File.....	154
19.9 Starting the Samba and NetBIOS Name Services	155
19.10 Accessing Samba Shares.....	156
19.11 Accessing Windows Shares from CentOS Stream 9	158
19.12 Summary	160
20. An Overview of Virtualization Techniques	161
20.1 Guest Operating System Virtualization	161

Table of Contents

20.2 Hypervisor Virtualization	162
20.2.1 Paravirtualization	163
20.2.2 Full Virtualization.....	164
20.2.3 Hardware Virtualization	164
20.3 Virtual Machine Networking.....	165
20.4 Summary	165
21. Installing KVM Virtualization on CentOS Stream 9	167
21.1 An Overview of KVM	167
21.2 KVM Hardware Requirements.....	167
21.3 Preparing CentOS Stream 9 for KVM Virtualization	168
21.4 Verifying the KVM Installation.....	168
21.5 Summary	170
22. Creating KVM Virtual Machines on CentOS Stream 9 using Cockpit	171
22.1 Installing the Cockpit Virtual Machines Module	171
22.2 Creating a Virtual Machine in Cockpit	171
22.3 Starting the Installation	173
22.4 Working with Storage Volumes and Storage Pools.....	175
22.5 Summary	177
23. Creating KVM Virtual Machines on CentOS Stream 9 using virt-manager	179
23.1 Starting the Virtual Machine Manager.....	179
23.2 Configuring the KVM Virtual System	180
23.3 Starting the KVM Virtual Machine	184
23.4 Summary	185
24. Creating KVM Virtual Machines with virt-install and virsh	187
24.1 Running virt-install to build a KVM Guest System	187
24.2 An Example CentOS 9 virt-install Command	187
24.3 Starting and Stopping a Virtual Machine from the Command-Line	189
24.4 Creating a Virtual Machine from a Configuration File.....	189
24.5 Summary	189
25. Creating a CentOS Stream 9 KVM Networked Bridge Interface	191
25.1 Getting the Current Network Manager Settings.....	191
25.2 Creating a Network Manager Bridge from the Command-Line	193
25.3 Declaring the KVM Bridged Network	194
25.4 Using a Bridge Network in a Virtual Machine	195
25.5 Creating a Bridge Network using nm-connection-editor.....	197
25.6 Summary	200
26. Managing KVM using the virsh Command-Line Tool.....	201
26.1 The virsh Shell and Command-Line.....	201
26.2 Listing Guest System Status	202

26.3 Starting a Guest System	203
26.4 Shutting Down a Guest System	203
26.5 Suspending and Resuming a Guest System	203
26.6 Saving and Restoring Guest Systems	203
26.7 Rebooting a Guest System	204
26.8 Configuring the Memory Assigned to a Guest OS	204
26.9 Summary	204
27. An Introduction to Linux Containers.....	205
27.1 Linux Containers and Kernel Sharing.....	205
27.2 Container Uses and Advantages.....	206
27.3 CentOS 9 Container Tools	207
27.4 The CentOS Container Registry.....	207
27.5 Container Networking.....	208
27.6 Summary	208
28. Working with Containers on CentOS Stream 9.....	209
28.1 Installing the Container Tools	209
28.2 Pulling a Container Image	209
28.3 Running the Image in a Container	211
28.4 Managing a Container.....	212
28.5 Saving a Container to an Image	213
28.6 Removing an Image from Local Storage	213
28.7 Removing Containers	214
28.8 Building a Container with Buildah.....	214
28.9 Building a Container from Scratch.....	214
28.10 Container Bridge Networking.....	215
28.11 Managing Containers in Cockpit.....	218
28.12 Summary	221
29. Setting Up a CentOS Stream 9 Web Server.....	223
29.1 Requirements for Configuring a CentOS Stream 9 Web Server	223
29.2 Installing the Apache Web Server Packages	223
29.3 Configuring the Firewall	224
29.4 Port Forwarding	224
29.5 Starting the Apache Web Server	224
29.6 Testing the Web Server	224
29.7 Configuring the Apache Web Server for Your Domain	225
29.8 The Basics of a Secure Website	226
29.9 Configuring Apache for HTTPS	227
29.10 Obtaining an SSL Certificate	227
29.11 Summary	229
30. Configuring a CentOS Stream 9 Postfix Email Server	231

Table of Contents

30.1 The Structure of the Email System	231
30.1.1 Mail User Agent	231
30.1.2 Mail Transfer Agent	231
30.1.3 Mail Delivery Agent	231
30.1.4 SMTP	232
30.1.5 SMTP Relay	232
30.2 Configuring a CentOS 9 Email Server	232
30.3 Postfix Pre-Installation Steps	232
30.4 Firewall/Router Configuration	233
30.5 Installing Postfix on CentOS Stream 9	233
30.6 Configuring Postfix	233
30.7 Configuring DNS MX Records	235
30.8 Starting Postfix on a CentOS 9 System	235
30.9 Testing Postfix	235
30.10 Sending Mail via an SMTP Relay Server	236
30.11 Summary	237
31. Adding a New Disk Drive to a CentOS Stream 9 System	239
31.1 Mounted File Systems or Logical Volumes	239
31.2 Finding the New Hard Drive	239
31.3 Creating Linux Partitions	240
31.4 Creating a File System on a CentOS 9 Disk Partition	241
31.5 An Overview of Journaled File Systems	242
31.6 Mounting a File System	242
31.7 Configuring CentOS 9 to Mount a File System Automatically	243
31.8 Adding a Disk Using Cockpit	243
31.9 Summary	245
32. Adding a New Disk to a CentOS Stream 9 Volume Group and Logical Volume	247
32.1 An Overview of Logical Volume Management (LVM)	247
32.1.1 Volume Group (VG)	247
32.1.2 Physical Volume (PV)	248
32.1.3 Logical Volume (LV)	248
32.1.4 Physical Extent (PE)	248
32.1.5 Logical Extent (LE)	248
32.2 Getting Information about Logical Volumes	248
32.3 Adding Additional Space to a Volume Group from the Command-Line	251
32.4 Adding Additional Space to a Volume Group Using Cockpit	253
32.5 Summary	255
33. Adding and Managing CentOS Stream 9 Swap Space	257
33.1 What is Swap Space?	257
33.2 Recommended Swap Space for CentOS Stream 9	257

33.3 Identifying Current Swap Space Usage	257
33.4 Adding a Swap File to a CentOS Stream 9 System.....	258
33.5 Adding Swap as a Partition	259
33.6 Adding Space to a CentOS Stream 9 LVM Swap Volume.....	259
33.7 Adding Swap Space to the Volume Group	260
33.8 Summary	262
34. CentOS Stream 9 System and Process Monitoring.....	263
34.1 Managing Processes	263
34.2 Real-time System Monitoring with top.....	267
34.3 Command-Line Disk and Swap Space Monitoring.....	268
34.4 Summary	269
Index	271

1. Introduction

CentOS Stream 9 Essentials is intended to provide detailed information on the installation, use, and administration of the CentOS 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

CentOS Stream 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:

```
[demo@demo-server ~]$ su -  
Password:  
[demo@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 of whether 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 administrative privileges are able to run otherwise restricted commands using *sudo*:

Introduction

```
[demo@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 demo:
```

```
.  
.
```

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

```
[demo@demo-server]$ sudo su -  
[demo@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:

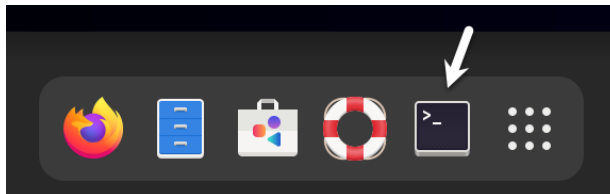


Figure 1-1

1.3 Editing Files

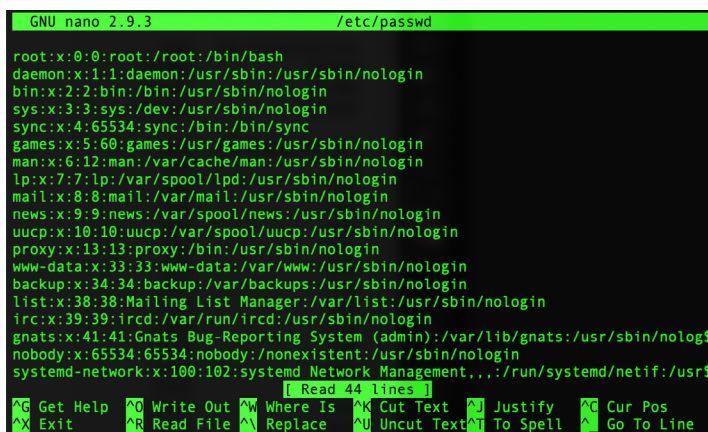
Configuring a Linux system typically involves editing files. For those new to Linux it can be unclear which editor to use. If you are running a terminal session and do not already have a preferred editor we recommend using the *nano* editor. To launch *nano* in a terminal window simply enter the following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

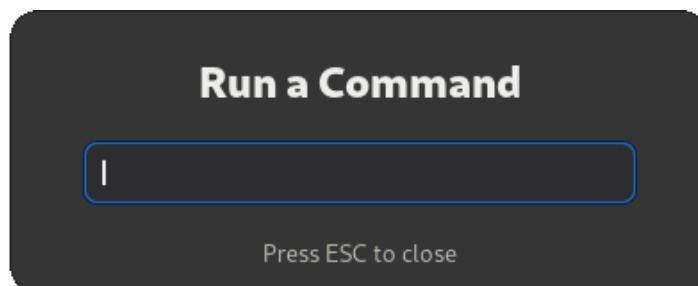


Figure 1-3

Introduction

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:

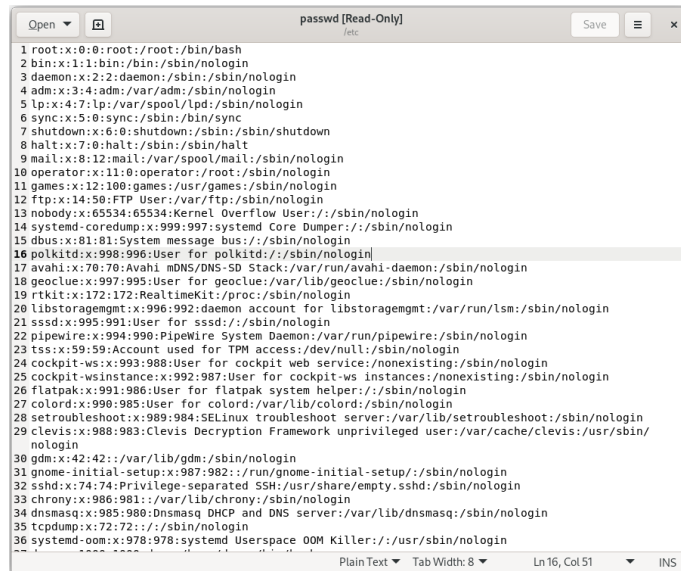


Figure 1-4

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/centos9.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.

2. A Brief History of CentOS Stream

CentOS Stream 9 is one of several variants (also referred to as *distributions*) of the Linux operating system. It is based on the source code of the Red Hat Enterprise Linux distribution (RHEL), developed by 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. before explaining how CentOS fits into this picture.

2.1 What exactly is Linux?

Linux is an operating system in much the same way that Windows is an operating system (and there are many similarities between Linux and Windows). 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

A Brief History of CentOS Stream

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 refer to Linux as Linux, which 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 could only 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 Red Hat offers. 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. CentOS Stream 9 is developed in close collaboration with Red Hat, Inc. and serves as a proving ground for the new features that will eventually be included in future Red Hat Enterprise Linux releases.

3. Installing CentOS Stream 9 on a Clean Disk Drive

There are now two ways in which a CentOS Stream 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 CentOS system, there is no need to perform a manual operating system installation in this situation.

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

CentOS can be installed either in a clean disk environment (where an entire disk is cleared of any existing partitions and dedicated entirely to CentOS) or in a dual boot environment where it 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 CentOS Stream 9 with Windows*”.

3.1 Obtaining the CentOS Stream 9 Installation Media

The CentOS distribution can be downloaded from the CentOS Project website using the following URL:

<https://www.centos.org/download/>

The installation media is provided as an ISO image and is available in the following forms:

- **CentOS Stream DVD ISO** - Contains a continuously evolving version of CentOS that will eventually become the next version of RHEL and CentOS.
- **CentOS Stream Container Images** - Pre-built images for use when building CentOS containers (a topic that will be covered in “*An Introduction to Linux Containers*”).
- **CentOS Cloud** - Images provided for cloud-based deployment of CentOS systems.
- **CentOS Stream RPM Packages** - The individual packages that comprise the CentOS 9 Stream operating system.

Installing CentOS Stream 9 on a Clean Disk Drive

For this book, we will download the DVD ISO image. The DVD image is self-contained, including all of the packages necessary to install a CentOS system, and is named using the following convention:

```
CentOS-Stream-9-<version>-<architecture>-dvd1.iso
```

For example, the most recent CentOS Stream 9 DVD image for 64-bit Intel systems is named as follows:

```
CentOS-Stream-9-latest-x86_64-dvd1.iso
```

To access the ISO image, open a web browser and navigate to <https://www.centos.org/download/>. Once the page has loaded, click on the CentOS Stream option (marked A in Figure 3-1), followed by version 9 (B):

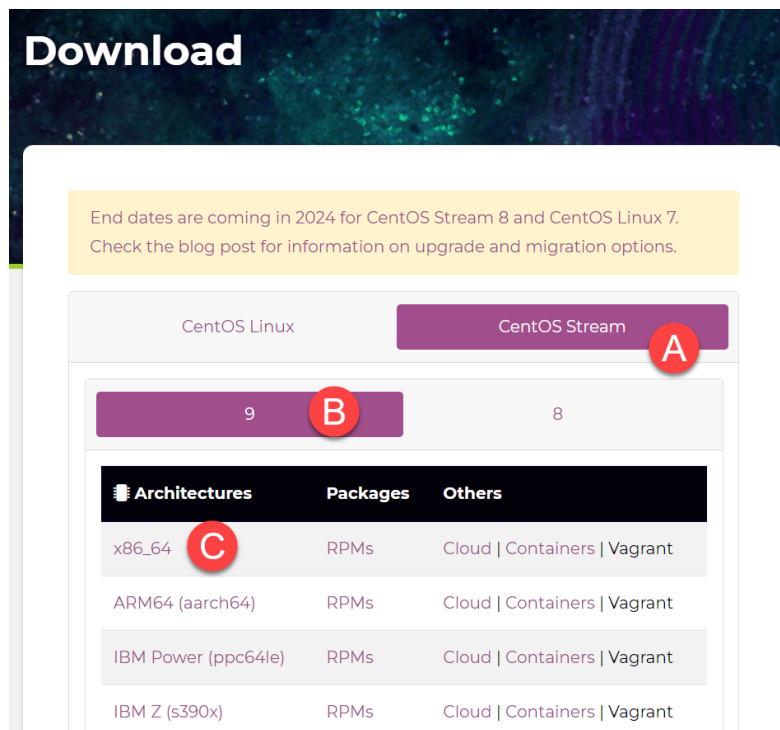


Figure 3-1

Finally, click on the architecture link (C) that matches your CPU to complete the ISO image download. Having downloaded the 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 CentOS 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 CentOS 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

Installing CentOS Stream 9 on a Clean Disk Drive

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/rdisk3`) 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 3-2:

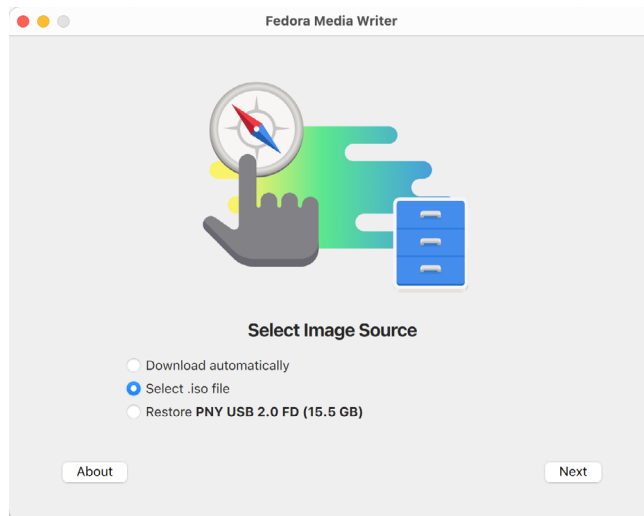


Figure 3-2

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

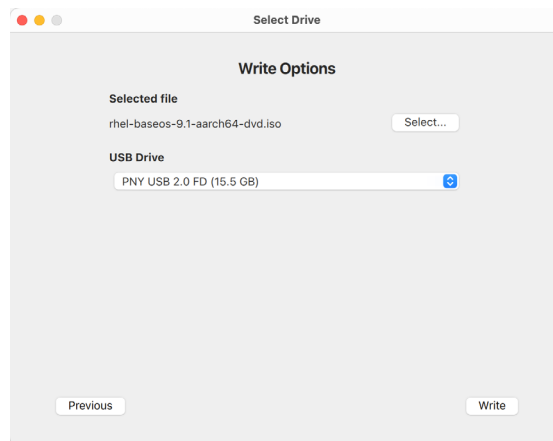


Figure 3-3

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

Installing CentOS Stream 9 on a Clean Disk Drive

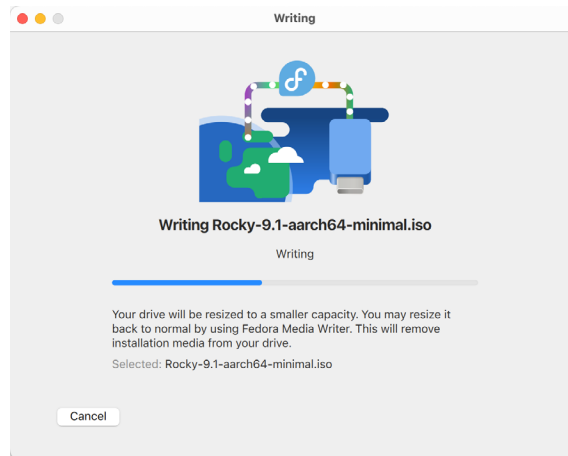


Figure 3-4

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

3.3 Installing CentOS Stream 9

Insert the CentOS 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:

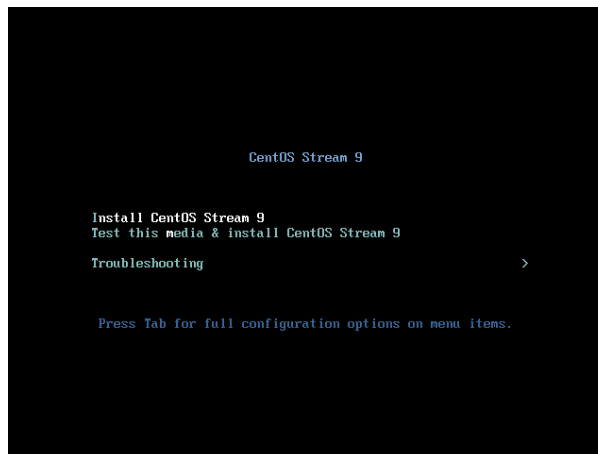


Figure 3-5

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 3-6 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 CentOS 9 system. An option is also available to perform the installation in basic graphics mode for systems without a graphical console:

Installing CentOS Stream 9 on a Clean Disk Drive

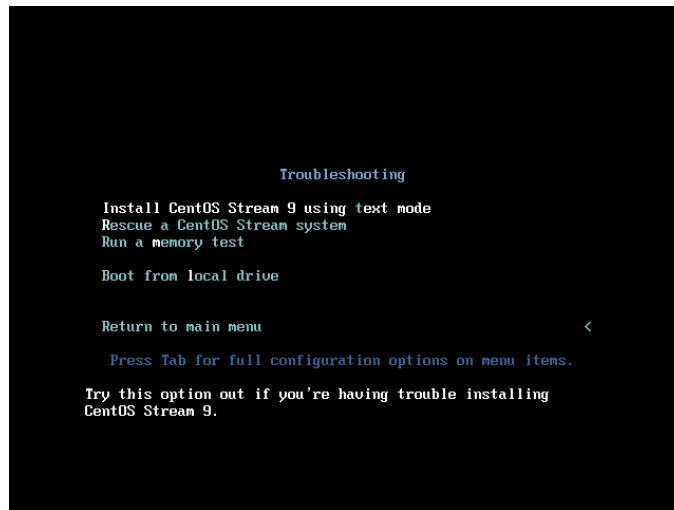


Figure 3-6

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

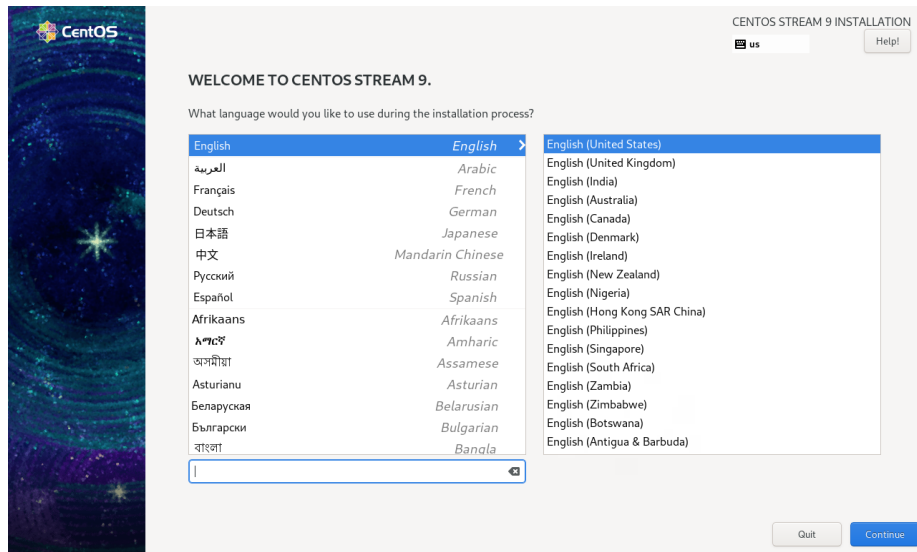


Figure 3-7

Select your preferred language on the first screen before clicking *Continue* to proceed to the main installation screen as shown in Figure 3-8:

Installing CentOS Stream 9 on a Clean Disk Drive

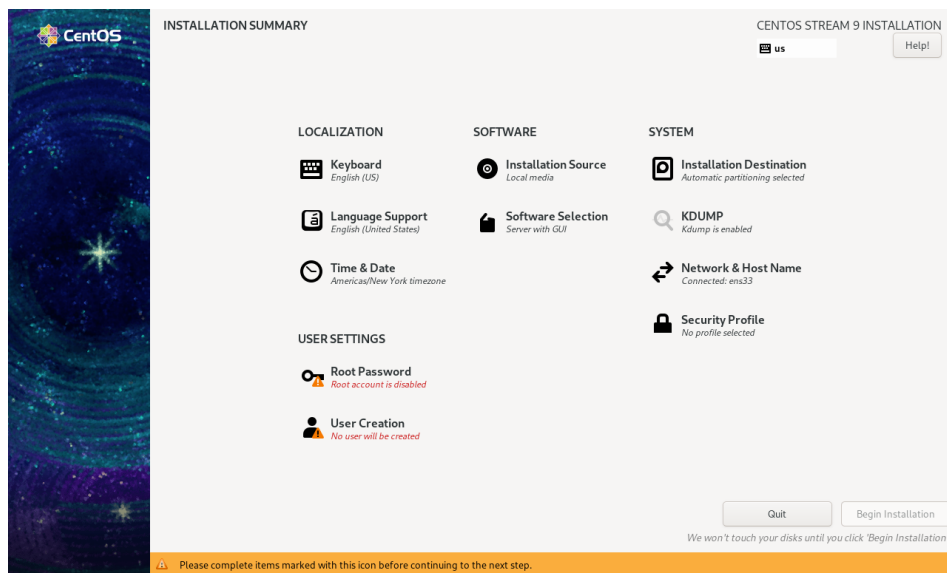


Figure 3-8

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

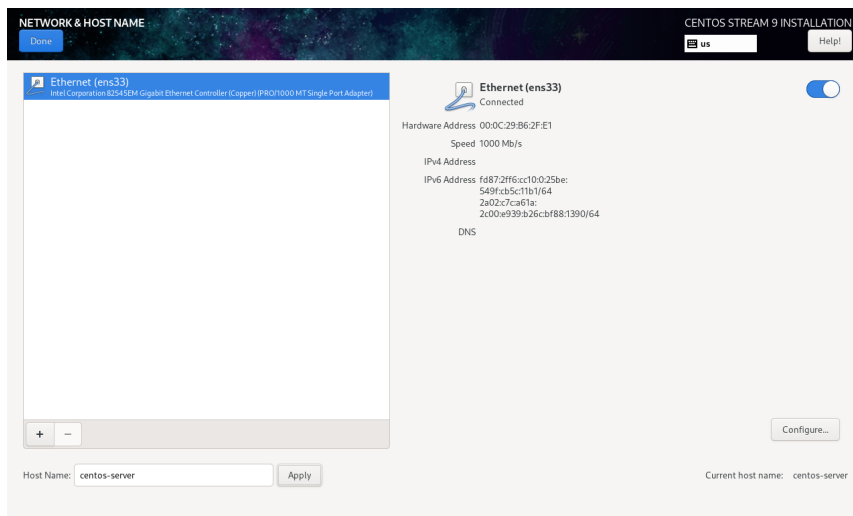


Figure 3-9

If your network connection requires additional settings, click on the *Configure...* button to access the advanced network settings screen illustrated in Figure 3-10:

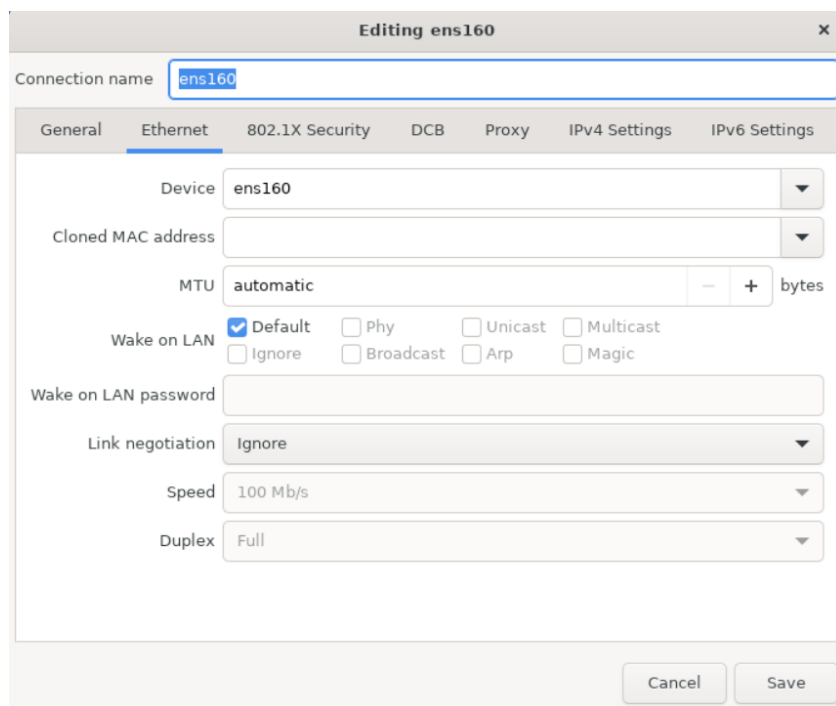


Figure 3-10

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 CentOS Stream 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 CentOS Stream 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:

Installing CentOS Stream 9 on a Clean Disk Drive

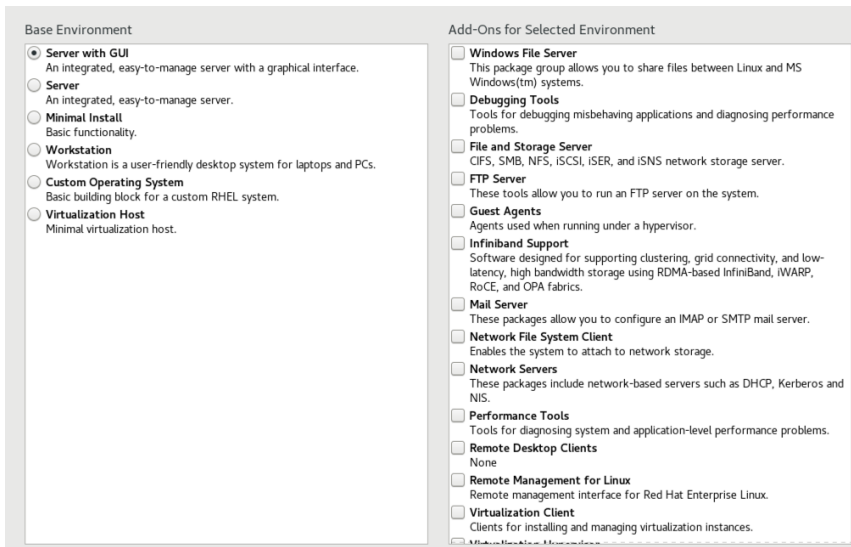


Figure 3-11

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.

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 CentOS Stream 9

When the *Installation Destination* option is selected, the installer will present a screen similar to the one illustrated in Figure 3-12 below:

Installing CentOS Stream 9 on a Clean Disk Drive

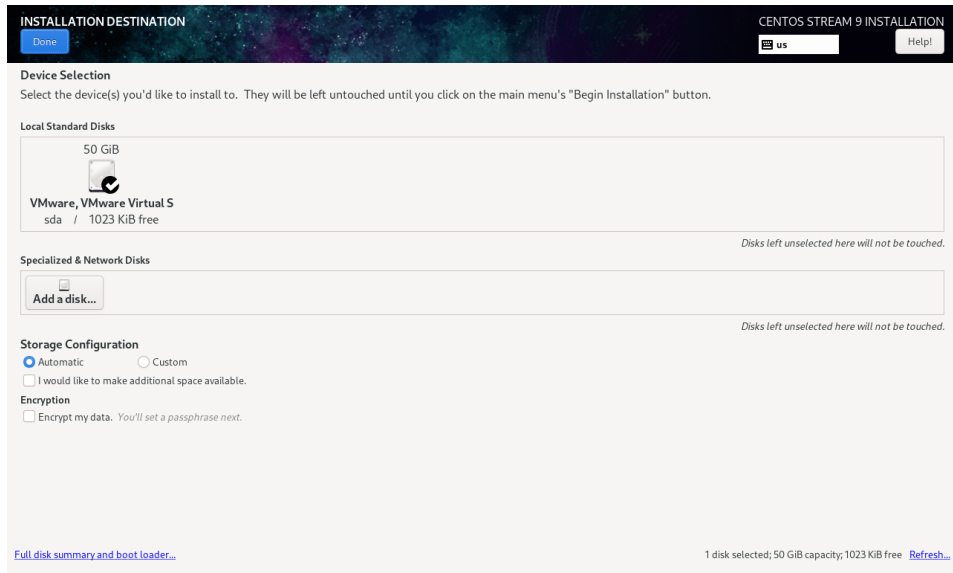


Figure 3-12

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 CentOS 9 has been installed. To remove any existing partitions so that they can be reclaimed and used by CentOS 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 3-13:

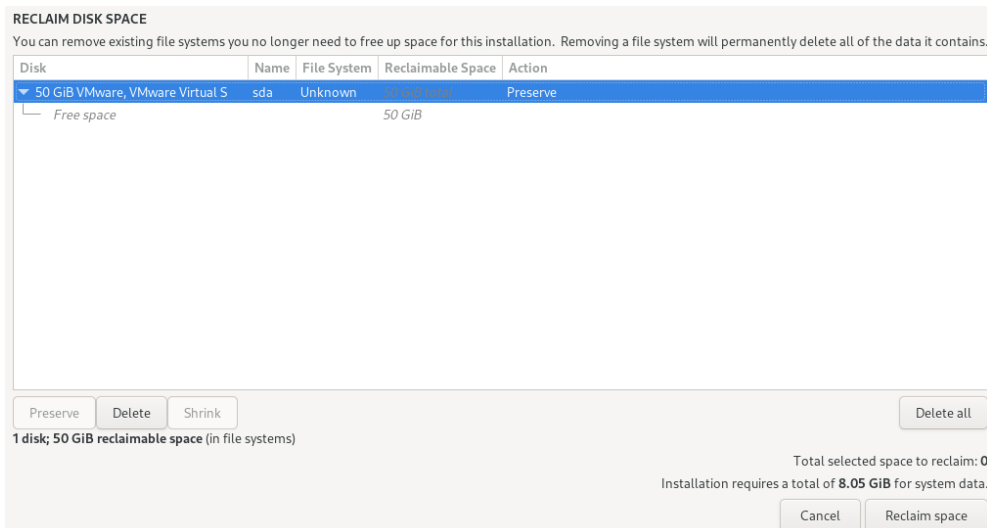


Figure 3-13

Installing CentOS Stream 9 on a Clean Disk Drive

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 CentOS 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 3-14):

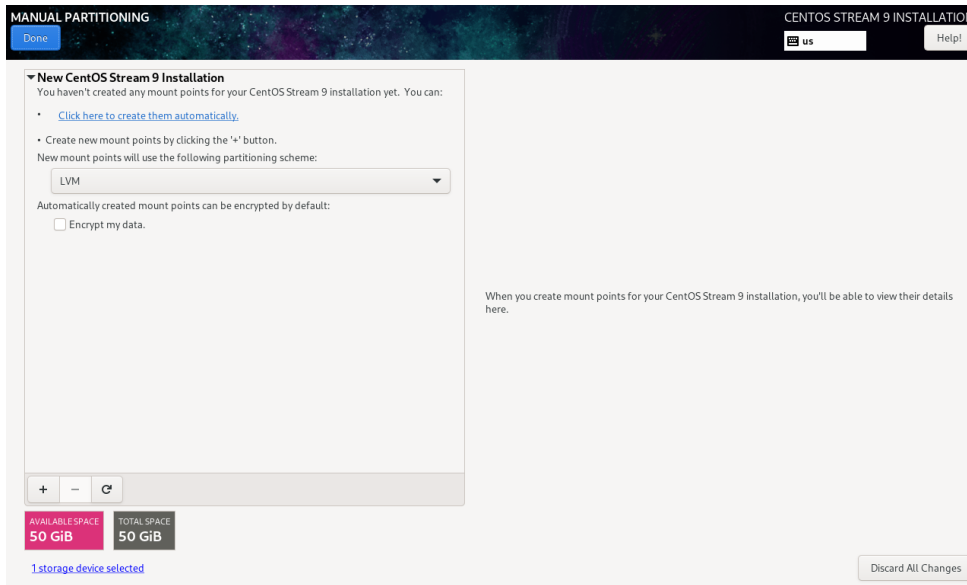


Figure 3-14

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 CentOS Stream 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 3-15, 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:

Installing CentOS Stream 9 on a Clean Disk Drive

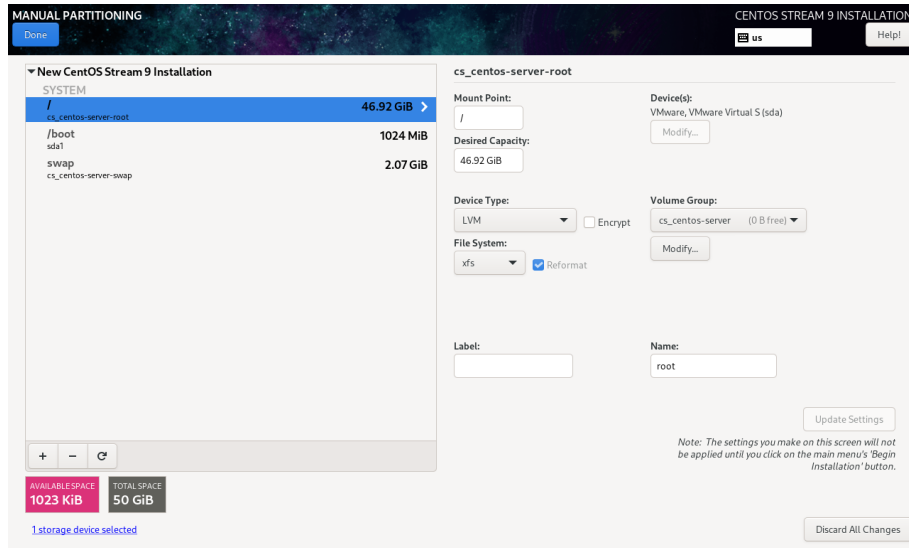


Figure 3-15

Once the disk has been configured, click on *Done*, review the summary of the changes to be made to the disk (Figure 3-16), followed by the *Accept Changes* button:

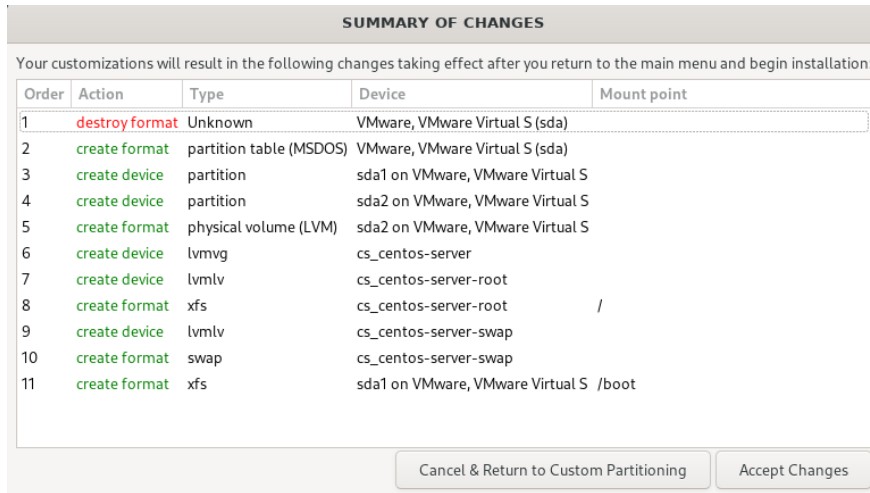


Figure 3-16

3.5 Disk Encryption

CentOS Stream 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 3-17:

Installing CentOS Stream 9 on a Clean Disk Drive

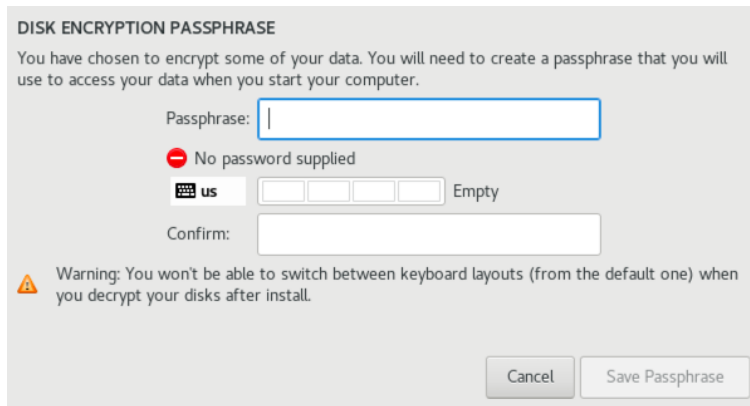


Figure 3-17

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 cover in the “*Configuring SSH Key-based Authentication on CentOS Stream 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 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 3-18 will appear:

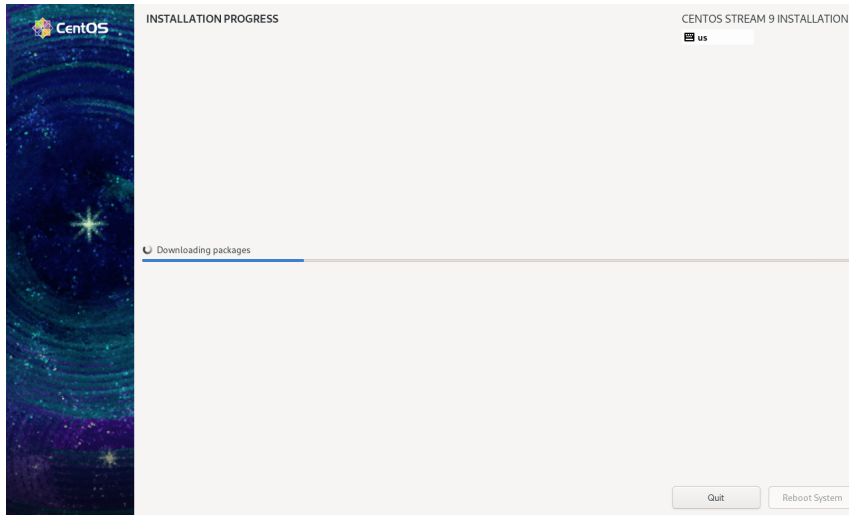


Figure 3-18

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

3.8 Final Configuration Steps

What appears on the console when the system starts will depend on which option was chosen from the Software Selection installation screen. If a GUI option 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 GUI installation, the GNOME initial setup tool will launch and offer a guided tour of the desktop environment.

3.9 Installing Updates

As with most operating systems today, each release of a CentOS 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 CentOS Stream 9 is to 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
```

Installing CentOS Stream 9 on a Clean Disk Drive

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 CentOS Stream 9 is ready for use.

3.10 Displaying Boot Messages

During the boot process, CentOS 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 Shells.
[ 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 dmw.
[ OK ] Finished Load Kernel Module fuse.
[ OK ] Mounted FUSE Control File System...
[ OK ] Started Journal Service.
[ OK ] Mounted FUSE Control File System.
[ OK ] Mounted UMass umblock Fuse Mount...
[ OK ] Finished Load Kernel Module configs.
[ OK ] Finished Load Kernel Modules.
[ OK ] Mounted Kernel Configuration File System...
[ OK ] Starting Apply Kernel Variables.
[ OK ] Mounted Kernel Configuration File System.
[ OK ] Finished Remount Root and Kernel File Systems.
[ OK ] Starting Flush Journal to Persistent Storage...
[ OK ] Starting Load/Save Random Seed...
[ OK ] Starting Create Static Device Nodes in /dev...
[ OK ] Finished Apply Kernel Variables.
[ OK ] Mounted UMass umblock Fuse Mount.
[ OK ] Finished Load/Save Random Seed.
[ OK ] Finished Flush Journal to Persistent Storage.
[ OK ] Finished Create Static Device Nodes in /dev.
[ OK ] Starting Rule-based Manager for Device Events and Files...
[ OK ] Finished Coldplug all udev Devices.
[ OK ] Starting Wait for udev To Complete Device Initialization...
[ OK ] Started Rule-based Manager for Device Events and Files.
[ OK ] Finished Monitoring of LVM mirrors, snapshots etc. using dmeventd or progress polling.
[ OK ] Starting Load Kernel Module configs...
[ OK ] Finished Load Kernel Module configs.
[ OK ] Starting Load Kernel Module fuse...
[ OK ] Finished Load Kernel Module fuse.
[ OK ] Started /usr/sbin/lsyncchange -say --autoactivation event rhel.
[ OK ] Finished Wait for udev To Complete Device Initialization.
[ OK ] Reached target Preparation for Local File Systems.
[ OK ] Mounted /boot...
[ OK ] Starting File System Check on /dev/disk/by-uuid/6CB7-6B7B...
[ OK ] Mounted /boot.
[ OK ] Finished File System Check on /dev/disk/by-uuid/6CB7-6B7B.
[ OK ] Mounted /bootefi...
[ OK ] Mounted Local File Systems.
[ OK ] Reached target Local File Systems.
[ OK ] Starting Tell Plymouth To Write Out Runtime Data...
[ OK ] Starting Create Volatile Files and Directories...
[ OK ] Finished Tell Plymouth To Write Out Runtime Data.
[ OK ] Finished Create Volatile Files and Directories.
[ OK ] 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
```

3.11 Summary

The first step in working with CentOS Stream 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.

5. Allocating Windows Disk Partitions to CentOS 9

In the previous chapter, we looked at installing CentOS 9 on the same disk as Windows. This so-called “dual boot” configuration allows the user to have both operating systems installed on a single disk drive with the option to boot one or the other when the system is powered on.

This chapter is intended for users who have decided they like CentOS 9 enough to delete Windows entirely from the disk and use the resulting space for Linux. In the following sections, we will work through this process step by step.

5.1 Unmounting the Windows Partition

If the steps in the “*Dual Booting CentOS Stream 9 with Windows*” chapter were followed to mount the Windows partition from within CentOS 9, steps should be taken to unmount the partition before continuing with this chapter. Assuming that the Windows partition was mounted as `/mnt/windows`, it can be unmounted as follows:

```
# umount /mnt/windows
```

The `/etc/fstab` file should also be edited to remove the `/mnt/windows` auto-mount if it was previously added.

5.2 Deleting the Windows Partitions from the Disk

The first step in freeing up the Windows partition for use by CentOS is to delete that partition. Before doing so, however, any data you need to keep must be backed up from both the Windows and partitions. Having done that, it is safe to proceed with this chapter.

To remove the Windows partitions, we first need to identify the disk on which they reside using the `fdisk` tool:

```
# fdisk -l
Disk /dev/nvme0n1: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7A38CD86-091E-4781-BFB0-928FD383C935
```

Device	Start	End	Sectors	Size	Type
/dev/nvme0n1p1	2048	206847	204800	100M	EFI System
/dev/nvme0n1p2	206848	239615	32768	16M	Microsoft reserved

Allocating Windows Disk Partitions to CentOS 9

```
/dev/nvme0n1p3      239616   49362943  49123328  23.4G Microsoft basic data
/dev/nvme0n1p4 132933632 134213631  1280000   625M Windows recovery environment
/dev/nvme0n1p5   49362944   51460095  2097152    1G Linux filesystem
/dev/nvme0n1p6   51460096 132933631  81473536  38.8G Linux LVM
```

In the above example output, the system contains one physical disk drive referenced by device name `/dev/nvme0n1`. On that disk drive are six partitions accessed via the device names `/dev/nvme0n1p1` through `/dev/nvme0n1p6`, respectively. Based on the values in the Types column, three Windows-related partitions exist. The first is the Windows system partition, while the second, much larger, partition is the Windows boot partition containing the Windows operating system and user data, followed by the Windows recovery partition.

To remove the partitions, start the *fdisk* tool using the device name of the disk containing the partition (`/dev/nvme0n1` in this instance) and follow the instructions to display the partition and sector information once again:

```
# fdisk /dev/nvme0n1
```

```
Welcome to fdisk (util-linux 2.37.4).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Command (m for help): p
```

```
Disk /dev/nvme0n1: 64 GiB, 68719476736 bytes, 134217728 sectors
```

```
Disk model: VMware Virtual NVMe Disk
```

```
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: gpt
```

```
Disk identifier: 7A38CD86-091E-4781-BFB0-928FD383C935
```

Device	Start	End	Sectors	Size	Type
<code>/dev/nvme0n1p1</code>	2048	206847	204800	100M	EFI System
<code>/dev/nvme0n1p2</code>	206848	239615	32768	16M	Microsoft reserved
<code>/dev/nvme0n1p3</code>	239616	49362943	49123328	23.4G	Microsoft basic data
<code>/dev/nvme0n1p4</code>	132933632	134213631	1280000	625M	Windows recovery environment
<code>/dev/nvme0n1p5</code>	49362944	51460095	2097152	1G	Linux filesystem
<code>/dev/nvme0n1p6</code>	51460096	132933631	81473536	38.8G	Linux LVM

```
Partition table entries are not in disk order.
```

```
Command (m for help):
```

Before proceeding, note the start and end addresses of the partitions we will be deleting (in other words, the start of `/dev/nvme0n1p2` and the sector before the start of `/dev/nvme0n1p5`).

At the command prompt, delete the Windows partitions (these being partitions 2, 3, and 4 on our example system):

```
Command (m for help): d
Partition number (1-6, default 6): 2
```

Partition 2 has been deleted.

```
Command (m for help): d
Partition number (1,3-6, default 6): 3
```

Partition 3 has been deleted.

```
Command (m for help): d
Partition number (1,4-6, default 6): 4
```

Partition 4 has been deleted.

Now that we have deleted the Windows partitions, we need to create the new CentOS partition in the vacated disk space. The partition number must match the number of the first partition removed (in this case, 2). It will also be necessary to enter the Start and End sectors of the partition precisely as reported for the old partition (*fdisk* will typically offer the correct values by default, though it is wise to double-check). If you are prompted to remove the NTFS signature, enter Y:

```
Command (m for help): n
Partition number (2-4,7-128, default 2): 2
First sector (206848-134217694, default 206848):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (206848-49362943, default
49362943):
```

Created a new partition 2 of type 'Linux filesystem' and of size 23.4 GiB.

```
Command (m for help):
```

Having made these changes, the next step is to check that the settings are correct:

```
Command (m for help): p
Disk /dev/nvme0n1: 64 GiB, 68719476736 bytes, 134217728 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 7A38CD86-091E-4781-BFB0-928FD383C935
```

Device	Start	End	Sectors	Size	Type
/dev/nvme0n1p1	2048	206847	204800	100M	EFI System
/dev/nvme0n1p2	206848	49362943	49156096	23.4G	Linux filesystem
/dev/nvme0n1p5	49362944	51460095	2097152	1G	Linux filesystem

Allocating Windows Disk Partitions to CentOS 9

```
/dev/nvme0n1p6 51460096 132933631 81473536 38.8G Linux LVM
```

To commit the changes, we now need to write the new partition information to disk and quit from the *fdisk* tool:

```
Command (m for help): w
The partition table has been altered.
Syncing disks.
```

5.3 Formatting the Unallocated Disk Partition

To make the new partition suitable for use by CentOS 9, it needs to have a file system created on it. The recommended file system type for the current release of CentOS is XFS which will be covered in greater detail in the chapter entitled “*Adding a New Disk Drive to a CentOS Stream 9 System*”. Creation of the file system is performed using the *mkfs.xfs* command as follows:

```
# mkfs.xfs -f /dev/nvme0n1p2
meta-data=/dev/nvme0n1p2      isize=512    agcount=4, agsize=1536128 blks
=                               sectsz=512    attr=2, projid32bit=1
=                               crc=1        finobt=1, sparse=1, rmapbt=0
=                               reflink=1    bigtime=1 inobtcount=1
data      =                   bsize=4096    blocks=6144512, imaxpct=25
=                               sunit=0      swidth=0 blks
naming    =version 2          bsize=4096    ascii-ci=0, ftype=1
log       =internal log      bsize=4096    blocks=3000, version=2
=                               sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096    blocks=0, rtextents=0
```

5.4 Mounting the New Partition

Next, we need to mount the new partition. In this example, we will mount it in a directory named */data*. You are free, however, to mount the new partition using any valid mount point you desire or to use it as part of a logical volume (details of which are covered in the chapter entitled “*Adding a New Disk to a CentOS Stream 9 Volume Group and Logical Volume*”). First, we need to create the directory to act as the mount point:

```
# mkdir /data
```

Secondly, we need to edit the mount table in */etc/fstab* so that the partition is automatically mounted each time the system starts. At the bottom of the */etc/fstab* file, add the following line to mount the new partition (modifying the */dev/nvme0n1p2* device to match your environment):

```
/dev/nvme0n1p2    /data    xfs defaults 0 0
```

Finally, we can manually mount the new partition (note that this will not be necessary on subsequent reboots as the partition will automount due to the setting we added to the */etc/fstab* file above):

```
# mount /data
```

To check the partition, run the following command to display the available space:

```
# df -h /data
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/nvme0n1p2	24G	200M	24G	1%	/data

5.5 Summary

The Windows partitions in a dual boot configuration can be removed anytime to free up space for a CentOS system by identifying which partitions belong to Windows and then deleting them. Once deleted, the unallocated space can be used to create a new filesystem and mounted to make it available to the CentOS system.

9. Managing CentOS Stream 9 Users and Groups

During the installation of CentOS Stream 9, the installer created a root or superuser account and required that a password be configured. The installer also provided the opportunity to create a user account for the system. We should remember that CentOS 9 is an enterprise-class, multi-user, and multi-tasking operating system. To use the full power of CentOS 9, therefore, it is likely that more than one user will need to be given access to the system. Each user should have their own user account login, password, home directory, and privileges.

Users are further divided into groups for easier administration, and those groups can have different levels of privileges. For example, you may have a group of users who work in the Accounting department. In such an environment, you can create an accounts group and assign all the Accounting department users to that group.

This chapter will cover the steps to add, remove and manage users and groups on a CentOS 9 system. Users and groups may be managed on CentOS 9 using command-line tools, the Cockpit web interface, and the desktop Settings app. In this chapter, we will look at each of these approaches to user management.

9.1 User Management from the Command-line

New users may be added to a CentOS 9 system via the command line using the *adduser* utility. To create a new user account, enter a command similar to the following:

```
# useradd john
```

By default, this will create a home directory for the user in the */home* directory (in this case, */home/john*). To specify a different home directory, use the *-d* command-line option when creating the account:

```
# useradd -d /users/johnsmith john
```

Once the account has been created, a password needs to be assigned using the *passwd* tool before the user will be able to log into the system:

```
# passwd john
```

```
Changing password for user john.
```

```
New password:
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

An existing user may be deleted via the command line using the *userdel* utility. While this will delete the account, the user's files and data will remain intact on the system:

Managing CentOS Stream 9 Users and Groups

```
# userdel john
```

It is also possible to remove the user's home directory and mail spool as part of the deletion process:

```
# userdel --remove john
```

All users on a CentOS 9 system are members of one or more groups. By default, new users are added to a private group with the same name as the user (in the above example, the account created for user john was a member of a private group also named john). However, as an administrator, it makes sense to organize users into more logical groups. For example, all salespeople might belong to a sales group, and accounting staff might belong to the accounts group, and so on. New groups are added from the command line using the *groupadd* command-line tool, for example:

```
# groupadd accounts
```

Use the *usermod* tool to add an existing user to an existing group from the command line:

```
# usermod -G accounts john
```

To add an existing user to multiple existing groups, run the *usermod* command with the -G option:

```
# usermod -G accounts,sales,support john
```

Note that the above commands remove the user from supplementary groups not listed after the -G but to which the user is currently a member. To retain any current group memberships, use the -a flag to append the new group memberships:

```
# usermod -aG accounts,sales,support john
```

An existing group may be deleted from a system using the *groupdel* utility:

```
# groupdel accounts
```

Note that if the group to be deleted is the primary or initial group for any user, it cannot be deleted. The user must first be deleted or assigned a new primary group using the *usermod* command before the group can be removed. A user can be assigned to a new primary group using the *usermod* -g option:

```
# usermod -g sales john
```

```
# groupdel accounts
```

Run the *groups* command to find out the groups to which a user belongs. For example:

```
$ groups john
john : accounts support
```

By default, a user account cannot perform tasks requiring superuser (root) privileges unless they know the root password. It is, however, possible to configure a user account so that privileged tasks can be performed using the *sudo* command. This involves adding the user account as a member of the *wheel* group, for example:

```
# usermod -aG wheel john
```

Once added to the *wheel* group, the user will be able to perform otherwise restricted tasks using *sudo* as follows:

```
$ sudo dnf update
[sudo] password for demo:
Updating Subscription Management repositories.
.
.
```

The sudo capabilities of the wheel group may be modified by editing the `/etc/sudoers` file and locating the following section:

```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)        ALL

## Same thing without a password
# %wheel          ALL=(ALL)        NOPASSWD: ALL
```

To disable sudo for all wheel group members, comment out the second line as follows:

```
## Allows people in group wheel to run all commands
# %wheel  ALL=(ALL)        ALL
```

To allow wheel group members to use sudo without entering a password (for security reasons, this is not recommended), uncomment the corresponding line in the sudoers file:

```
## Same thing without a password
%wheel          ALL=(ALL)        NOPASSWD: ALL
```

Behind the scenes, all these commands are simply changing the `/etc/passwd`, `/etc/group`, and `/etc/shadow` files on the system.

9.2 User Management with Cockpit

If the Cockpit web interface is installed and enabled on the system (a topic covered in the chapter entitled “*An Overview of the Cockpit Web Interface*”), several user management tasks can be performed within the Accounts screen shown in Figure 9-1 below:

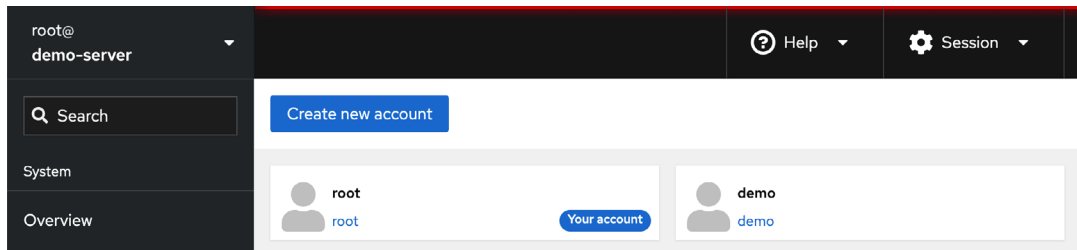


Figure 9-1

The screen will display any existing user accounts on the system and provides a button to add additional accounts. To create a new account, click the Create New Account button and enter the requested information in the resulting dialog (Figure 9-2). Note that the option is also available to create the account but to lock it until later:

Managing CentOS Stream 9 Users and Groups

Create new account

Full name

User name

Password

Confirm

Authentication

☒ Use password

☐ Require password change on first login

☐ Disallow password authentication ?

Create

Cancel

Figure 9-2

To modify a user account, select it from the main screen and make any modifications to the account details:

Accounts > demo

demo

Terminate session

Delete

Full name

demo

User name

demo

Roles

☒ Server administrator ?

Last login

Mar 28, 2023, 3:59 PM

Options

☐ Disallow interactive password ?

Never expire account

edit

Password

Set password

Force change

Never expire password

edit

Authorized public SSH keys

Add key

There are no authorized public keys for this account.

Figure 9-3

This screen allows various tasks, including locking or unlocking the account, changing the password, or forcing the user to configure a new password to be performed. In addition, if the Server Administrator option is selected, the user will be added to the wheel group and permitted to use sudo to perform administrative tasks. A button is also provided to delete the user from the system.

If the user is accessing the system remotely using an SSH connection with key encryption, the

user's public key may be added within this screen. SSH access and authentication will be covered later in “*Configuring SSH Key-based Authentication on CentOS Stream 9*”.

9.3 User Management using the Settings App

A third user account management option is available via the GNOME desktop settings app. This app is accessed by clicking on icons in the top right-hand corner of the GNOME desktop and selecting the Settings option, as shown in Figure 9-4:

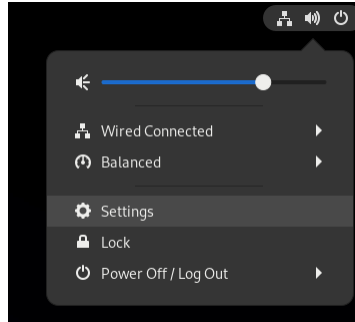


Figure 9-4

When the main settings screen appears, click the Users option in the left-hand navigation panel. By default, the settings will be locked, and making any changes to the user accounts on the system will not be possible. To unlock the Settings app, click the Unlock button in Figure 9-5 below and enter your password. Note that it will only be possible to unlock the settings if you are logged in as a user with sudo privileges:

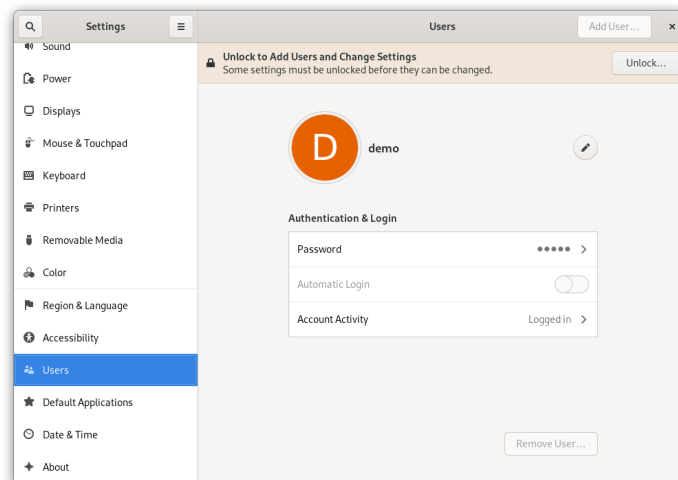


Figure 9-5

Once the app has been unlocked, a button labeled *Add User...* will appear in the title bar. Click this button to display the dialog shown in Figure 9-6 below:

Managing CentOS Stream 9 Users and Groups

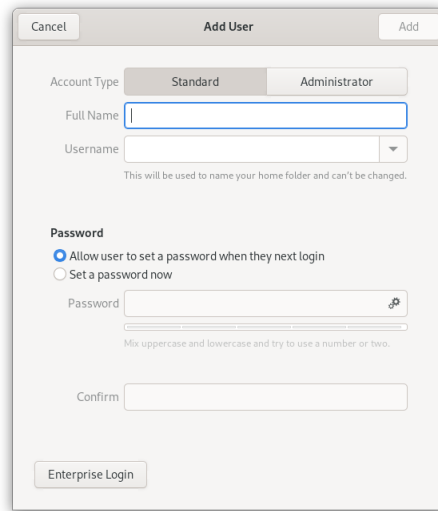


Figure 9-6

Select the Administrator account to assign sudo access to the new user; otherwise, leave Standard selected. Next, enter the user's full name and username and assign a password now, or allow the user to set up the password when they first log into their account. Once the information has been entered, click the Add button to create the account.

The settings for an existing user can be viewed, modified, or the account deleted at any time by selecting the corresponding icon within the Users screen, as shown in Figure 9-7. The option is also available to view the user's login activity. Note that it will be necessary to unlock the Settings app again before any changes can be made to an account:

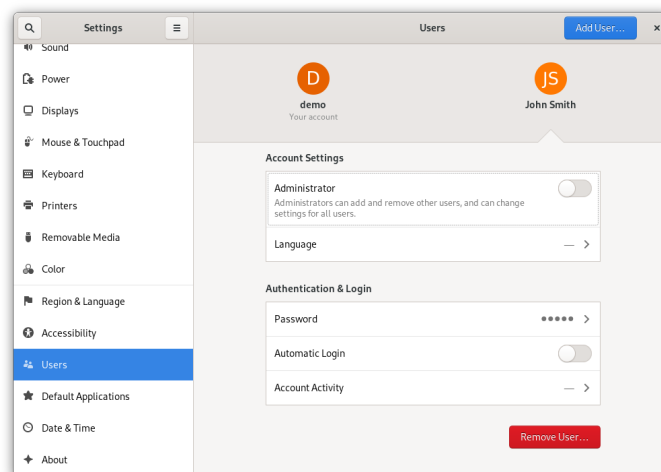


Figure 9-7

9.4 Summary

As a multi-user operating system, CentOS Stream 9 has been designed to support controlled access for multiple users. During installation, the root user account was created and assigned a password, and the option to create a user account was also provided. Additional user accounts may be added to the system using a set of command-line tools via the Cockpit web interface or the GNOME Settings app. In addition to user accounts, Linux also implements the concept of groups. New groups can be added, and users assigned to those groups using command-line tools, and each user must belong to at least one group. By default, a standard, non-root user does not have permission to perform privileged tasks. Users that are members of the special *wheel* group, however, may perform privileged tasks by making use of the `sudo` command.

13. CentOS Stream 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.

CentOS Stream 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 CentOS 9 will be covered in the chapter entitled “*CentOS Stream 9 Firewall Configuration with firewallld*”.

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 CentOS Stream 9 Services and iptables Rules

By default, a newly installed 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 CentOS 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 CentOS 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 “CentOS Stream 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 a CentOS 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 CentOS Stream 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 by implementing 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.

21. Installing KVM Virtualization on CentOS Stream 9

Earlier versions of CentOS provided two virtualization platforms: Kernel-based Virtual Machine (KVM) and Xen. In recent releases, support for Xen has been removed, leaving KVM as the only bundled virtualization option supplied with CentOS Stream 9. In addition to KVM, third-party solutions are available in products such as VMware and Oracle VirtualBox. Since KVM is supplied with CentOS 9, however, this virtualization solution will be covered in this and subsequent chapters.

Before plunging into installing and running KVM, it is worth discussing how it fits into the various types of virtualization outlined in the previous chapter.

21.1 An Overview of KVM

KVM is categorized as a Type-1 hypervisor virtualization solution that implements full virtualization with support for unmodified guest operating systems using Intel VT and AMD-V hardware virtualization support.

KVM differs from many other Type-1 solutions in that it turns the host Linux operating system into the hypervisor, allowing bare metal virtualization to be implemented while running a complete, enterprise-level host operating system.

21.2 KVM Hardware Requirements

Before proceeding with this chapter, we must discuss the hardware requirements for running virtual machines within a KVM environment. First and foremost, KVM virtualization is only available on certain processor types. As previously discussed, these processors must include either Intel VT or AMD-V technology.

To check for virtualization support, run the *lscpu* command in a terminal window:

```
# lscpu | grep Virtualization:
```

If the system contains a CPU with Intel VT support, the above command will provide the following output:

```
Virtualization:          VT-x
```

Alternatively, the following output will be displayed when a CPU with AMD-V support is detected:

```
Virtualization:          AMD-V
```

If the CPU does not support virtualization, no output will be displayed by the above *lscpu* command.

Installing KVM Virtualization on CentOS Stream 9

Note that while the above commands only report whether the processor supports the respective feature, it does not indicate whether it is currently enabled in the BIOS. This is because, in practice, virtualization support is typically disabled by default in the BIOS of most systems. Therefore, you should check your BIOS settings to ensure the appropriate virtualization technology is enabled before proceeding with this tutorial.

Unlike a dual-booting environment, a virtualized environment involves running two or more complete operating systems concurrently on a single computer system. This means the system must have enough physical memory, disk space, and CPU processing power to comfortably accommodate all these systems in parallel. Therefore, before beginning the configuration and installation process, check on the minimum system requirements for both CentOS 9 and your chosen guest operating systems and verify that your host system has sufficient resources to handle the requirements of both systems.

21.3 Preparing CentOS Stream 9 for KVM Virtualization

Unlike Xen, it is not necessary to run a special version of the kernel to support KVM. As a result, KVM support is already available for use with the standard kernel via installing a KVM kernel module, thereby negating the need to install and boot from a special kernel.

To avoid conflicts, however, if a Xen-enabled kernel is currently running on the system, reboot the system and select a non-Xen kernel from the boot menu before proceeding with the remainder of this chapter.

The tools required to set up and maintain a KVM-based virtualized system are only installed by default if selected explicitly during the CentOS 9 operating system installation process. To install the KVM tools from the command prompt, execute the following command in a terminal window:

```
# dnf install qemu-kvm qemu-img libvirt virt-install libvirt-client
```

If you have access to a graphical desktop environment, the *virt-manager* package is also recommended:

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

21.4 Verifying the KVM Installation

It is worthwhile checking that the KVM installation worked correctly before moving forward. When KVM is installed and running, two modules will have been loaded into the kernel. The presence or otherwise of these modules can be verified in a terminal window by running the *lsmod* command:

```
# lsmod | grep kvm
```

Assuming that the installation was successful, the above command should generate output similar to the following:

```
kvm_intel          237568    0
kvm                737280    1 kvm_intel
irqbypass         16384    1 kvm
```

Note that if the system contains an AMD processor, the `kvm` module will likely read *kvm_amd* rather than *kvm_intel*.

The installation process should also have configured the `libvirtd` daemon to run in the background. Once again, using a terminal window, run the following command to ensure `libvirtd` is running:

```
# systemctl status libvirtd
libvirtd.service - Virtualization daemon
   Loaded: loaded (/usr/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-03-06 14:41:22 EST; 3min 54s ago
```

If the process is not running, start it as follows:

```
# systemctl enable --now libvirtd
# systemctl start libvirtd
```

If the desktop environment is available, run the *virt-manager* tool by selecting Activities and entering “virt” into the search box. When the Virtual Machine Manager icon appears, click it to launch it. When loaded, the manager should appear as illustrated in the following figure:

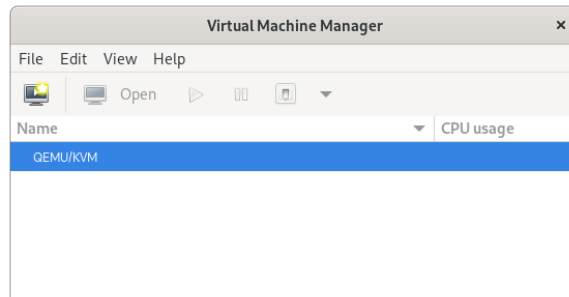


Figure 21-1

If the QEMU/KVM entry is not listed, select the *File -> Add Connection* menu option and, in the resulting dialog, select the QEMU/KVM Hypervisor before clicking on the Connect button:

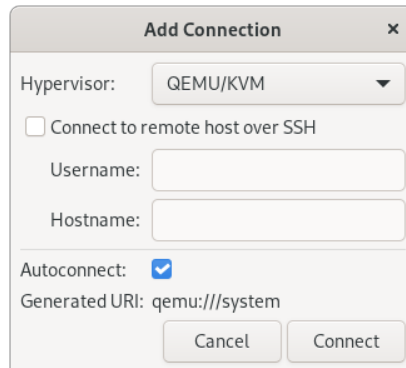


Figure 21-2

Installing KVM Virtualization on CentOS Stream 9

If the manager is not currently connected to the virtualization processes, right-click on the entry listed and select Connect from the popup menu.

21.5 Summary

KVM is a Type-1 hypervisor virtualization solution that implements full virtualization with support for unmodified guest operating systems using Intel VT and AMD-V hardware virtualization support. It is the default virtualization solution bundled with CentOS 9 and can be installed quickly and easily on any CentOS 9 system with appropriate processor support. With KVM support installed and enabled, the following chapters will outline some options for installing and managing virtual machines on a CentOS 9 host.

27. An Introduction to Linux Containers

The preceding chapters covered the concept of virtualization, emphasizing creating and managing virtual machines using KVM. This chapter will introduce a related technology in the form of Linux Containers. While there are some similarities between virtual machines and containers, key differences will be outlined in this chapter, along with an introduction to the concepts and advantages of Linux Containers. The chapter will also introduce some CentOS Stream container management tools. Once the basics of containers have been covered in this chapter, the next chapter will work through some practical examples of creating and running containers on CentOS 9.

27.1 Linux Containers and Kernel Sharing

In simple terms, Linux containers are a lightweight alternative to virtualization. A virtual machine contains and runs the entire guest operating system in a virtualized environment. The virtual machine, in turn, runs on top of an environment such as a hypervisor that manages access to the physical resources of the host system.

Containers work by using a concept referred to as kernel sharing, which takes advantage of the architectural design of Linux and UNIX-based operating systems.

To understand how kernel sharing and containers work, it helps first to understand the two main components of Linux or UNIX operating systems. At the core of the operating system is the kernel. In simple terms, the kernel handles all the interactions between the operating system and the physical hardware. The second key component is the root file system which contains all the libraries, files, and utilities necessary for the operating system to function. Taking advantage of this structure, containers each have their own root file system but share the host operating system's kernel. This structure is illustrated in the architectural diagram in Figure 27-1 below.

This type of resource sharing is made possible by the ability of the kernel to dynamically change the current root file system (a concept known as change root or chroot) to a different root file system without having to reboot the entire system. Linux containers are essentially an extension of this capability combined with a container runtime, the responsibility of which is to provide an interface for executing and managing the containers on the host system. Several container runtimes are available, including Docker, lxd, containerd, and CRI-O. Earlier versions of CentOS used Docker by default, but Podman has supplanted this as the default in CentOS 9.

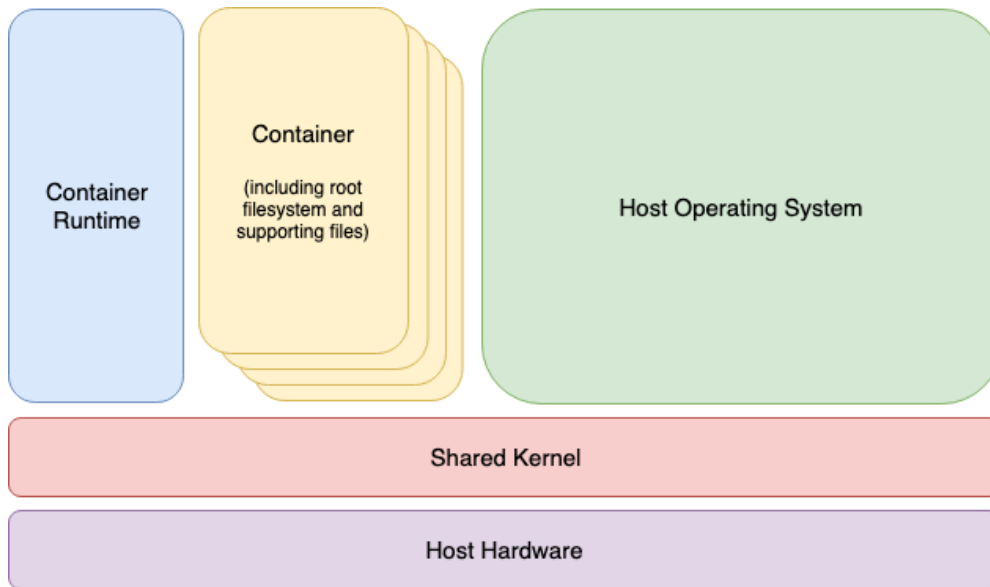


Figure 27-1

27.2 Container Uses and Advantages

The main advantage of containers is that they require considerably less resource overhead than virtualization allowing many container instances to be run simultaneously on a single server. They can be started and stopped rapidly and efficiently in response to demand levels. In addition, containers run natively on the host system providing a level of performance that a virtual machine cannot match.

Containers are also highly portable and can be easily migrated between systems. Combined with a container management system such as Docker, OpenShift, and Kubernetes, it is possible to deploy and manage containers on a vast scale spanning multiple servers and cloud platforms, potentially running thousands of containers.

Containers are frequently used to create lightweight execution environments for applications. In this scenario, each container provides an isolated environment containing the application together with all of the runtime and supporting files required by that application to run. The container can then be deployed to any other compatible host system that supports container execution and runs without any concerns that the target system may not have the necessary runtime configuration for the application - all of the application's dependencies are already in the container.

Containers are also helpful when bridging the gap between development and production environments. By performing development and QA work in containers, they can be passed to production and launched safely because the applications run in the same container environments in which they were developed and tested.

Containers also promote a modular approach to deploying large and complex solutions. Instead of developing applications as single monolithic entities, containers can be used to design applications

as groups of interacting modules, each running in a separate container.

One possible drawback of containers is that the guest operating systems must be compatible with the shared kernel version. It is not, for example, possible to run Microsoft Windows in a container on a Linux system. Nor is it possible for a Linux guest system designed for the 2.6 version of the kernel to share a 2.4 version kernel. These requirements are not, however, what containers were designed for. Rather than being seen as limitations, these restrictions should be considered some of the key advantages of containers in providing a simple, scalable, and reliable deployment platform.

27.3 CentOS 9 Container Tools

CentOS Stream 9 provides several tools for creating, inspecting, and managing containers. The main tools are as follows:

- **buildah** – A command-line tool for building container images.
- **podman** – A command-line based container runtime and management tool. Performs tasks such as downloading container images from remote registries and inspecting, starting, and stopping images.
- **skopeo** – A command-line utility used to convert container images, copy images between registries and inspect images stored in registries without downloading them.
- **runc** – A lightweight container runtime for launching and running containers from the command line.
- **OpenShift** – An enterprise-level container application management platform consisting of command-line and web-based tools.

All of the above tools comply with the Open Container Initiative (OCI), a set of specifications designed to ensure that containers conform to the same standards between competing tools and platforms.

27.4 The CentOS Container Registry

Although CentOS Stream 9 is provided with a set of tools designed to be used in place of those provided by Docker, those tools still need access to CentOS images for use when building containers. For this purpose, a set of CentOS Stream container images is hosted on a repository located on Red Hat's Quay.io web site at the following URL:

<https://quay.io/repository/centos/centos?tab=tags>

In addition to downloading (referred to as “pulling” in container terminology) container images from Quay.io and other third-party host registries, you can also use registries to store your own images. This can be achieved by hosting your own registry or using existing services such as Quay.io, Docker, Amazon AWS, Google Cloud, Microsoft Azure, and IBM Cloud, to name a few options.

27.5 Container Networking

By default, containers are connected to a network using a Container Networking Interface (CNI) bridged network stack. In the bridged configuration, all the containers running on a server belong to the same subnet and, as such, can communicate with each other. The containers are also connected to the external network by bridging the host system's network connection. Similarly, the host can access the containers via a virtual network interface (usually named `podman0`) which will have been created as part of the container tool installation.

27.6 Summary

Linux Containers offer a lightweight alternative to virtualization and take advantage of the structure of the Linux and Unix operating systems. Linux Containers share the host operating system's kernel, with each container having its own root file system containing the files, libraries, and applications. As a result, containers are highly efficient and scalable and provide an ideal platform for building and deploying modular enterprise-level solutions. In addition, several tools and platforms are available for building, deploying, and managing containers, including third-party solutions and those provided by Red Hat.

Index

Symbols

! 64
 #! 67
 >> 65
 | 65
 \$DISPLAY variable 138
 50-redhat.conf 137
 .bashrc 67
 /etc/containers/networks 218
 /etc/default/grub 29
 /etc/exports 144
 /etc/fstab 31, 33, 145, 147, 243
 /etc/gdm/custom.conf 137
 /etc/group 71
 /etc/httpd 225
 /etc/passwd 71
 /etc/samba/smb.conf 150
 /etc/shadow 71
 /etc/sshd_config 137
 /etc/ssh/sshd_config.d 137
 /etc/sudoers 71
 /etc/systemd/system 88, 130
 /etc/yum.repos.d/ directory 78
 /home 69
 /proc/swaps 257
 .requires 89
 .ssh 118, 120
 /usr/lib/systemd/system 88
 /var/log/maillog 236
 .wants 88

A

ACC Corporation 6
 access control list 174
 ACL 174

Activity Overview 41
 adduser 69
 AIX 5
 alias 65
 Aliases 65
 AMD-V 164
 Andrew S. Tanenbaum 5
 Apache
 mod_ssl 227
 Apache web server 223
 Application Stream 77
 AppStream 77, 79
 modules 79
 packages 79
 profiles 79
 AppStream repository 77
 authorized_keys file 122

B

BaseOS repository 77, 78
 Bash
 scripts 67
 Bash shell 61
 aliases 65
 .bashrc 67
 chmod 68
 command-line editing 63
 do loops 67
 echo 66
 environment variables 66
 filename completion 64
 Filename shorthand 64
 for loop 67
 history 63
 HOME 66
 input and output redirection 64
 PATH 66
 path completion 64
 pipes 65

Index

- sh 68
- stderr 65
- stdin 64
- stdout 64
- basic.target 84
- Bell Labs 61
- Bourne Again SHell 61
- Bourne shell 61
- Brian Fox 61
- buildah 207
 - containers 214
 - from registry 214
 - from scratch 214
 - installroot 215
 - run 215
 - umount 215
- Buildah 209

C

- CA 226
- cat 64
- CentOS
 - history of 5
- certbot 227
- Certificate Authority 226
- change root 205
- chmod 68
- chroot 205
- cifs filesystem 159
- CNI 208, 215
- Cockpit 89
 - accessing 50
 - account management 54
 - applications 55
 - cockpit-machines 171
 - cockpit-storaged 146
 - create VM 171
 - Drives 244
 - enabling 50
 - extensions 49, 55
 - firewall management 114

- installing 50
- logs 52
- Multiple Servers 57
- networking 53
- NFS 145
- overview 49
- persistent metrics 58
- Podman Containers 218
- port 50
- select bridge 195
- services 54, 89
- storage 53, 243
- system 51
- systemd 89
- terminal access 56
- user management 71
- virtual machines 55
- volume groups 253
- cockpit-machines 171
- cockpit.socket 50
- cockpit-storaged 146, 253
- Compressed X11 Forwarding 138
- Connection Profiles 96
- containerd 205
- Container Networking Interface 208, 215
- Containers
 - attaching to an image 212
 - bridge networking 215
 - buildah 214
 - networking 215
 - Networking Interface 215
 - overview 205
 - pulling an image 209
 - removing an image 213
 - running an image 211
 - saving to an image 213
 - stopping 212
- context labels 153
- CoreUtils 215
- CRI-O 205
- C shell 61

D

- daemon 83
- dash 41
- David Korn 61
- dd 11
- DDNS 223
- Default Boot Option 29
- default.target 86
- df 252, 268
- discretionary access control 152
- disk drive
 - detecting 239
- disk I/O 269
- Disk partition
 - formatting 36
- disk usage 268
- diskutil 11
- DISPLAY variable 138
- dmesg 11
- dmz 108
- dnf 23
 - groupinfo 128
 - groupinstall 128
 - grouplist 128
- dnf.conf file 78
- DNS 104
- DNS MX Records 235
- Docker 205, 206
- do loops 67
- Domain Name Server 104
- DoS 113
- dual boot 33
- Dual Booting 27
- Dynamic DNS 223
- DynDNS 223

E

- echo 66
- Email Server 231
- encryption
 - disk 21

- env 66
- Environment Variables 66
- EPEL 30
- Errata 4
- Exim 231
- export 66
- exportfs 144
- ext2 253
- ext3 253
- ext4 253
- Extra Packages for Enterprise Linux 30

F

- FAT16 27
- FAT32 27
- fdisk 31, 33, 240, 251
 - create partition 240
 - list partitions 240
- Fedora Linux 7
- Fedora Media Writer 12
- Fedora Project 7
- Fetchmail 231
- Filename Shorthand 64
- File System
 - creating 241
 - mounting 242
- File Transfer (Control) 103
- File Transfer Protocol (Data) 103
- findmnt 11
- Firewall
 - Interfaces 109
 - overview 101, 107
 - Port Forwarding 113
 - Ports 109
 - Services 109
 - web server settings 224
 - Zones 107
- firewall-cmd 103, 110, 130
 - enable NFS 144
 - mail settings 233
 - web server settings 224

Index

firewall-config 115
firewalld
 default zone 110
 display zone information 110
 firewall-cmd 110
 firewall-config 115
 ICMP rules 113
 interfaces 107, 109
 list services 111
 overview 107
 permanent settings 110
 port forwarding 113
 port rules 111, 112
 ports 107, 109
 reload 110
 runtime settings 110
 services 107, 110
 status 109
 zone creation 112
 zone/interface assignments 112
 zones 107
 zone services 111
for 67
ForwardX11Trusted 138
FQDN 104
free 258
 -s flag 268
Free Software Foundation 6
fsck 242
fstab 145, 243
FTP 101, 103
Full Virtualization 164
Fuse NTFS driver 30

G

GDM 23
gedit 138
getfacl 174
GNOME
 apps 47
 Software app 47

GNOME Desktop 39
 Activity Overview 41
 dash 41
 gnome-tweaks 46
 settings 45
 starting 39
 switcher 44
 tweaks 46
 windows 43
 Workspaces 44
GNOME Desktop Environment 127
GNOME Display Manager 23
gnome-system-monitor 265
gnome-tweaks 46
GNU/Linux 6
GNU project 7
graphical.target 84
groupadd 70
groupdel 70
groups 70
grub2-set-default 30
GRUB_SAVEDefault 29
Guest OS virtualization 161

H

Hardware Virtualization 164
Hewlett-Packard 5
history 62, 63
HOME 66
HP-UX 5
HTTP 105, 226
httpd 223, 227
httpd.conf 225, 227
httpd-le-ssl.conf 228
HTTPS 101, 106, 109, 226
hypercalls 163
Hypertext Text Transfer Protocol 105
Hypertext Transfer Protocol Secure 106
Hypervisor 162
 hypercalls 163
 type-1 162

type-2 162
Hypervisor Virtualization 162

I

IBM 5
ICMP 113
id_rsa file 118, 120, 121
id_rsa.pub file 118, 122
if statements 67
IMAP 104
IMAP4 105
Input and Output Redirection 64
installation
 disk partitioning 18
Installation
 clean disk 9
install packages
 listing 78
Intel VT 164
Intel x86
 Rings 163
Internet Control Message Protocol 113
Internet Message Access Protocol, Version 4 105
internet service provider 223
I/O redirection 65
iotop 269
 installing 269
IPSets 116
iptables 101, 103, 107
 rules 102
 tool 102
ip tool 94
ISO image
 write to USB drive 10
ISP 223

J

journalctl 130, 135
Journaled File Systems 242

K

Kdump 18
Kerberos 105
kernel 5
kill 264
 -9 flag 264
KMail 231
Korn shell 61
Kubernetes 206
KVM
 hardware requirements 167
 installation 168
 overview 167
 virt-manager 168
kvm_amd 169
kvm_intel 169
KVM virtualization 165

L

LE 248
Let's Encrypt 227
libvirt 108
libvirtd 179
libvirtd daemon 169
Linus Torvalds 6
Linux Containers. *See* Containers
Logical Extent 248
Logical Volume 248
Logical Volume Management 247
loopback interface 93
lost+found 242
ls 62, 65
lscpu 167
lsmod 168
LV 248
lvdisplay 249, 252, 259
lvextend 252
LVM 247
lxd 205

M

macOS

Index

- writing ISO to USB drive 11
- MacVTap 165
- Mail Delivery Agent 231
- Mail Exchanger 235
- Mail Transfer Agent 231
- Mail User Agent 231
- main.cf 233
- man 62
- mandatory access control 152
- Marc Ewing 6
- Martin Hellman 117
- MDA 231
- MINIX 5
- mkfs.xfs 36, 241
- mkswap 258, 259
- mod_ssl 227
- mount 31, 145, 158, 242, 248
- MTA 231
- MUA 231
- multi-user.target 84
- MX 235

N

- NAT 165, 173
- NetBIOS 155
- NetBIOS nameservice 155
- Network Address Translation 165
- Networked Bridge Interface 191
- Network File System 106
- NetworkManager
 - Connection Profiles 96
 - enabling 92
 - installing 92
 - permissions 100
- Network News Transfer Protocol 105
- Network Time Protocol 105
- NFS 106
 - Cockpit 145
 - firewall settings 144
- nfs-client.target 84
- NFS service 143

- nfs-utils 145
- NMB 155
- nmcli 91, 193
 - activate connection 95
 - add bridge 193
 - add connections 97
 - command line options 92
 - deactivate connection 95
 - delete connection 97
 - device status 93
 - general status 92
 - interactive 98
 - modify connection 96
 - permissions 100
 - reload 96
 - show connections 94
 - switch connection 94
 - wifi scan 95
- nm-connection-editor 91
 - create bridge 197
- nmtui 91
- NNTP 105
- NTFS 27
- NTP 105
- nvme 239

O

- OCI 207
- Open Container Initiative 207
- OpenShift 206, 207
- OSI stack 105

P

- Paravirtualization 163, 164
- Partition
 - mounting 36
- passwd 69
- PATH 66
- Path Completion 64
- PE 248
- Physical Extent 248

Physical Volume 248

Pipes 65

podman 207

attach 212

commit 213

exec 212, 213

images 211, 213

inspect 211

network commands 217

network connect 217

network create 217

network disconnect 217

network inspect 217

network ls 216

network rm 218

pause 213

ps -a 212

pull 210

rm 214

run 211

stop 212

unpause 213

Podman 205, 209

POP3 105

Port Forwarding 113, 224

Ports

securing 101

Postfix 231, 232

configuring 233

installing 233

main.cf 233

postmap 236

sasl_passwd 236

starting 235

testing 235

postmap 236

Post Office Protocol 105

poweroff.target 83

PowerShell 121

private key 117

ps 65, 153, 263

-a flag 263

-aux flags 264

-H flag 265

TERM signal 264

-u flag 263

public key 117

public key encryption 117

PuTTY 123

secure tunnel 133

X11 Forwarding 140

PuTTYgen 123

PuTTY Key Generator 123

PV 248

pvccreate 252, 261

pvdisplay 250

pwd 62

PXE 180

Q

QEMU 180

QEMU/KVM Hypervisor 169

Qmail 232

Quay.io 207

R

RealVNC 129

reboot.target 84

Red Hat, Inc. 5

Red Hat Package Manager 78

Red Hat Support 6

Remote Desktop Access

insecure 127

secure 127

remote-fs.target 84

remote installation 17

Remote Procedure Calls 143

Repositories 77

rescue.target 83

resize2fs 253

restorecon 154

RHGB 29

Index

Richard Stallman 6
rlogin 104
root password
 specifying during installation 22
root_t 154
root user 1
RPC 143
rpcbind 143
rpm 78
RPM 78
rsh 104
runc 207

S

Samba 143, 150
 add user 154
 firewall settings 150
 installing 150
 NetBIOS 155
 samba_share_t 153
 SELinux 152
 smbclient 155, 158
 smb.conf 150
 smbpasswd 154
 smb_t 153
 testparm 155
Samba Client 150
samba_share_t 153, 154
sasl_passwd 236
SATA 239
Secure File Transfer Protocol 103
Secure Shell 104, 117
Secure Socket Layer 226
Secure Sockets Layer 106
Secure Tunnel 133
SELinux
 context labels 153
 restorecon 154
 Samba 152
 sestatus 152
 type enforcement 153

SELinux
 enforcing mode 153
 permissive mode 153
Sendmail 231, 232
Server Message Block 149
Server with GUI 127
Services
 securing 101
sestatus 152
setfacl 174
Settings App 73
 users 73
SFTP 103
sh 68
Shell Scripts 67
Simple Mail Transfer Protocol 104
Simple Mail Transport Protocol 232
Simple Network Management Protocol 106
skopeo 207, 209
Skopeo 209
SMB 149
smbclient 155, 158
smb.conf 150
 testing 154
 testparm 154
smbpasswd 154
smb_t 153
SMTP 101, 104, 109, 232
SMTP Relay 232, 236
SNMP 106
sockets.target 84
Solaris 5
spawning 265
ssh
 -C flag 138
 X11 Forwarding 138
 -X flag 138
SSH 101, 104, 117, 132
 Microsoft Windows 121
 Multiple Keys 120
 VNC 132

- ssh client 119
- ssh-copy-id 119, 122
- sshd_config.d 137
- sshd_config.d directory 120
- sshd_config file 120
- sshd service 120
- ssh-keygen 118
- SSH Service
 - installing 118
 - starting 118
- SSH tunnel 132
- SSL 106, 226
- SSL certificate 226, 227
- SSL Labs 229
- startx 39, 129
- stderr 65
- stdin 64
- stdout 64
- storage devices
 - identify 11
- Storage Pools 175
- Storage Volumes 175
- su - command 1
- sudo 1
 - wheel group 70
- SunOS 5
- Superuser 1
- Swap 257
 - current size 257
 - free 258
 - logical volume 259
 - lvdisplay 259
 - mkswap 258, 259
 - /proc/swaps 257
 - pvccreate 261
 - recommendations 257
 - swapoff 258, 261
 - swapon 258
- swapoff 258, 261
- swapon 258
- system

- units 86
- unit types 86
- systemctl 85
 - daemon-reload 130
- systemd 83
 - services 83
 - targets 83
- System Monitor 265
- system processes 263

T

- TCP/IP 101, 109
 - Well-Known Ports 103
- Telnet 104
- Terminal window 2
- TERM signal 264
- testparm
 - smb.conf 154
- TFTP 104
- TigerVNC 129, 131
 - installation 129
 - viewer 129
- TightVNC 129
- TLS 226
- top 267
 - u flag 268
- Transport Layer Security 226
- Trivial File Transfer Protocol 104
- Trusted X11 Forwarding 138
- Type-1 hypervisors 163
- Type-2 hypervisors 163
- type enforcement 153

U

- UDP 103
- umount 11, 145
- UNIX 5, 61
 - origins of 5
- update packages 79
- USB drive
 - device name 11

Index

userdel 69
user_home_t 153
usermod 70
user processes 263
Users and Groups 69

V

VcXsrv 139
VG 247
vgdisplay 248
vgextend 252
vgs 249, 261
virbr0 191, 192
virsh 187, 189, 194, 201
 destroy 189, 203
 dumpxml 189
 edit 196
 help 202
 list 202
 reboot 204
 restore 203
 resume 203
 save 203
 setmem 204
 setmemmax 204
 shell 201
 shutdown 189, 203
 start 189, 203
 suspend 203
virt-install 171, 187, 195
virt-manager 168, 179, 196
 installation 168
 New VM wizard 180
 storage pools 182
VirtualBox 167
Virtualization 161
 AMD-V 164
 full 164
 guest 161
 hardware 164
 hypercalls 163

hypervisor 162
Intel VT 164
KVM virtualization 165
MacVTap 165
Type-1 162
Type-2 162
virt-manager 168
Virtual Machine Networking 165
Virtual Network Computing 127
virt-viewer 174, 188
vmdk 181
VMware 167
VNC 127
 firewall settings 130
 installation 129
 PuTTY 133
 secure 132
 server shutdown 134
 start server 130
 troubleshooting 134
 vncpasswd 129
 vncserver 131
 vncserver@.service 130
vncpasswd 129
vncserver 131
VNC Server
 configuring 129
vncserver@.service 130
vncviewer 132
Volume Group 247

W

Wayland 137
WaylandEnable 137
wc 65
Web Server 223
 testing 224
Well-Known Ports 103
wheel group 70
which 62
Whitfield Diffie 117

- wildcard character 64
- wildcards 64
- Windows
 - accessing partition from Linux 30
 - Disk Management 27
 - Dual Booting 27
 - shrink volume 28
 - writing ISO to USB drive 12
- Windows partition
 - reclaiming 33
 - unmounting 33
- Windows PowerShell 121
- wipefs 252
- Workspaces 44
- Workstation 127

X

- X11 Forwarding 137
 - compressed 138
- X11Forwarding 137
- x86 family 163
- Xen 168
- XFS file system 241
- XFS filesystem 252
- xfs_growfs 252
- XLaunch 139
- X.org 137
- X Window System 137

Y

- yum.repos.d directory 78

