

Red Hat Enterprise Linux 8

Essentials

Red Hat Enterprise Linux 8 Essentials

ISBN-13: 978-1-951442-04-0

© 2020 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: 2.0

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..... | 5 |
| 1.5 Errata..... | 5 |
| 2. A Brief History of Red Hat Linux | 7 |
| 2.1 What exactly is Linux?..... | 7 |
| 2.2 UNIX Origins | 7 |
| 2.3 Who Created Linux? | 7 |
| 2.4 The Early Days of Red Hat | 8 |
| 2.5 Red Hat Support | 8 |
| 2.6 Open Source..... | 9 |
| 2.7 The Fedora Project | 9 |
| 2.8 CentOS - The Free Alternative | 9 |
| 2.9 Summary | 9 |
| 3. Installing RHEL 8 on a Clean Disk Drive | 11 |
| 3.1 Obtaining the Red Hat Enterprise Linux Installation Media | 11 |
| 3.2 Writing the ISO Installation Image to a USB Drive..... | 12 |
| 3.2.1 Linux..... | 12 |
| 3.2.2 macOS | 13 |
| 3.2.3 Windows | 14 |
| 3.3 Installing Red Hat Enterprise Linux 8 | 15 |
| 3.4 Partitioning a Disk for RHEL 8 | 19 |
| 3.5 The Physical Installation | 23 |
| 3.6 Final Configuration Steps..... | 24 |
| 3.7 Installing Updates..... | 25 |
| 3.8 Displaying Boot Messages..... | 25 |
| 3.9 Summary | 26 |
| 4. Dual Booting RHEL 8 with Windows | 27 |
| 4.1 Partition Resizing | 27 |
| 4.2 Editing the RHEL 8 Boot Menu | 29 |
| 4.3 Changing the Default Boot Option..... | 30 |
| 4.4 Accessing the Windows Partition from the Command-line | 32 |
| 4.5 Accessing the Windows Partition from the Desktop..... | 33 |
| 4.6 Summary | 36 |

Table of Contents

| | |
|--|-----------|
| 5. Allocating Windows Disk Partitions to RHEL 8 | 37 |
| 5.1 Unmounting the Windows Partition..... | 37 |
| 5.2 Deleting the Windows Partitions from the Disk | 37 |
| 5.3 Formatting the Unallocated Disk Partition | 40 |
| 5.4 Mounting the New Partition | 40 |
| 5.5 Editing the Boot Menu | 41 |
| 5.6 Summary | 41 |
| 6. A Guided Tour of the GNOME 3 Desktop | 43 |
| 6.1 Installing the GNOME Desktop | 43 |
| 6.2 An Overview of the GNOME 3 Desktop | 43 |
| 6.3 Launching Activities | 45 |
| 6.4 Managing Windows | 47 |
| 6.5 Using Workspaces | 48 |
| 6.6 Calendar and Notifications | 48 |
| 6.7 Desktop Settings | 49 |
| 6.8 Installing RHEL Desktop Software | 50 |
| 6.9 Beyond Basic Customization | 51 |
| 6.10 Summary | 52 |
| 7. An Overview of the Cockpit Web Interface | 53 |
| 7.1 An Overview of Cockpit..... | 53 |
| 7.2 Installing and Enabling Cockpit..... | 54 |
| 7.3 Accessing Cockpit | 54 |
| 7.4 Overview | 55 |
| 7.5 Logs | 56 |
| 7.6 Storage..... | 56 |
| 7.7 Networking..... | 56 |
| 7.8 Accounts | 57 |
| 7.9 Services | 57 |
| 7.10 Applications | 58 |
| 7.11 Podman Containers | 58 |
| 7.12 Diagnostic Reports..... | 59 |
| 7.13 Kernel Dump | 59 |
| 7.14 SELinux..... | 60 |
| 7.15 Virtual Machines | 60 |
| 7.16 Software Updates..... | 61 |
| 7.17 Subscriptions..... | 61 |
| 7.18 Terminal | 61 |
| 7.19 Connecting to Multiple Servers | 62 |
| 7.20 Enabling Stored Metrics | 64 |
| 7.21 Summary | 65 |

| | |
|---|-----------|
| 8. Using the Bash Shell on RHEL 8 | 67 |
| 8.1 What is a Shell? | 67 |
| 8.2 Gaining Access to the Shell | 67 |
| 8.3 Understanding Linux Pathnames..... | 68 |
| 8.4 Entering Commands at the Prompt..... | 69 |
| 8.5 Getting Information about a Command..... | 69 |
| 8.6 Bash Command-line Editing..... | 69 |
| 8.7 Working with the Shell History..... | 70 |
| 8.8 Filename Shorthand..... | 70 |
| 8.9 Filename and Path Completion..... | 71 |
| 8.10 Input and Output Redirection..... | 71 |
| 8.11 Working with Pipes in the Bash Shell..... | 72 |
| 8.12 Configuring Aliases | 72 |
| 8.13 Environment Variables | 72 |
| 8.14 Writing Shell Scripts | 74 |
| 8.15 Summary | 75 |
| 9. Managing RHEL 8 Users and Groups | 77 |
| 9.1 User Management from the Command-line | 77 |
| 9.2 User Management with Cockpit..... | 79 |
| 9.3 User Management using the Settings App | 80 |
| 9.4 Summary | 83 |
| 10. Understanding RHEL 8 Software Installation and Management..... | 85 |
| 10.1 Repositories..... | 85 |
| 10.2 The BaseOS Repository | 86 |
| 10.3 The AppStream Repository | 87 |
| 10.4 Summary | 90 |
| 11. Managing RHEL 8 systemd Units and Targets | 91 |
| 11.1 Understanding RHEL 8 systemd Targets..... | 91 |
| 11.2 Understanding RHEL 8 systemd Services | 91 |
| 11.3 RHEL 8 systemd Target Descriptions..... | 91 |
| 11.4 Identifying and Configuring the Default Target | 93 |
| 11.5 Understanding systemd Units and Unit Types..... | 94 |
| 11.6 Dynamically Changing the Current Target | 94 |
| 11.7 Enabling, Disabling and Masking systemd Units | 95 |
| 11.8 Working with systemd Units in Cockpit..... | 97 |
| 11.9 Summary | 98 |
| 12. RHEL 8 Network Management | 99 |
| 12.1 An Introduction to NetworkManager | 99 |
| 12.2 Installing and Enabling NetworkManager..... | 100 |

Table of Contents

| | |
|---|------------|
| 12.3 Basic nmcli Commands..... | 100 |
| 12.4 Working with Connection Profiles | 104 |
| 12.5 Interactive Editing..... | 106 |
| 12.6 Configuring NetworkManager Permissions..... | 108 |
| 12.7 Summary | 108 |
| 13. RHEL 8 Firewall Basics | 109 |
| 13.1 Understanding Ports and Services | 109 |
| 13.2 Securing Ports and Services..... | 109 |
| 13.3 Services and iptables Rules..... | 110 |
| 13.4 Well Known Ports and Services..... | 111 |
| 13.5 Summary | 114 |
| 14. Basic RHEL 8 Firewall Configuration with firewalld..... | 115 |
| 14.1 An Introduction to firewalld..... | 115 |
| 14.1.1 Zones | 115 |
| 14.1.2 Interfaces..... | 117 |
| 14.1.3 Services..... | 117 |
| 14.1.4 Ports..... | 117 |
| 14.2 Checking firewalld Status | 117 |
| 14.3 Configuring Firewall Rules with firewall-cmd..... | 118 |
| 14.3.1 Identifying and Changing the Default Zone..... | 118 |
| 14.3.2 Displaying Zone Information | 118 |
| 14.3.3 Adding and Removing Zone Services..... | 119 |
| 14.3.4 Working with Port-based Rules..... | 120 |
| 14.3.5 Creating a New Zone..... | 120 |
| 14.3.6 Changing Zone/Interface Assignments | 120 |
| 14.3.7 Masquerading | 120 |
| 14.3.8 Adding ICMP Rules | 121 |
| 14.3.9 Implementing Port Forwarding..... | 121 |
| 14.4 Managing firewalld from the Cockpit Interface..... | 122 |
| 14.5 Managing firewalld using firewall-config | 123 |
| 14.6 Summary | 124 |
| 15. Configuring SSH Key-based Authentication on RHEL 8..... | 125 |
| 15.1 An Overview of Secure Shell (SSH)..... | 125 |
| 15.2 SSH Key-based Authentication | 125 |
| 15.3 Setting Up Key-based Authentication | 126 |
| 15.4 Installing and Starting the SSH Service..... | 126 |
| 15.5 SSH Key-based Authentication from Linux and macOS Clients..... | 126 |
| 15.6 Managing Multiple Keys | 128 |
| 15.7 SSH Key-based Authentication from Windows 10 Clients | 129 |
| 15.8 SSH Key-based Authentication using PuTTY | 131 |

| | |
|---|------------|
| 15.9 Generating a Private Key with PuTTYgen | 132 |
| 15.10 Installing the Public Key for a Google Cloud Instance | 133 |
| 15.11 Summary | 134 |
| 16. RHEL 8 Remote Desktop Access with Vino | 137 |
| 16.1 Remote Desktop Access Types | 137 |
| 16.2 Secure and Insecure Remote Desktop Access | 137 |
| 16.3 Installing the GNOME Desktop Environment | 138 |
| 16.4 Firewall and SELinux Configuration | 139 |
| 16.5 Enabling Remote Desktop Access on RHEL 8 | 139 |
| 16.6 Connecting to the Shared Desktop | 141 |
| 16.7 Connecting from Non-Linux Clients | 143 |
| 16.8 Establishing a Secure Remote Desktop Session | 144 |
| 16.9 Establishing a Secure Tunnel on Windows using PuTTY | 146 |
| 16.10 Summary | 147 |
| 17. RHEL 8 Remote Desktop Access with VNC..... | 149 |
| 17.1 Installing VNC on RHEL 8 | 149 |
| 17.2 Configuring the VNC Server | 149 |
| 17.3 Connecting to a VNC Server | 151 |
| 17.4 Establishing a Secure Remote Desktop Session | 151 |
| 17.5 Shutting Down a Desktop Session | 152 |
| 17.6 Troubleshooting a VNC Connection | 152 |
| 17.7 Summary | 153 |
| 18. Displaying RHEL 8 Applications Remotely (X11 Forwarding) | 155 |
| 18.1 Requirements for Remotely Displaying RHEL 8 Applications | 155 |
| 18.2 Remotely Displaying a RHEL 8 Application | 156 |
| 18.3 Trusted X11 Forwarding | 156 |
| 18.4 Compressed X11 Forwarding | 156 |
| 18.5 Displaying Remote RHEL 8 Apps on Windows..... | 157 |
| 18.6 Summary | 160 |
| 19. Using NFS to Share RHEL 8 Files with Remote Systems | 161 |
| 19.1 Ensuring NFS Services are running on RHEL 8 | 161 |
| 19.2 Configuring the RHEL 8 Firewall to Allow NFS Traffic | 162 |
| 19.3 Specifying the Folders to be Shared | 162 |
| 19.4 Accessing Shared RHEL 8 Folders | 163 |
| 19.5 Mounting an NFS Filesystem on System Startup | 163 |
| 19.6 Unmounting an NFS Mount Point | 163 |
| 19.7 Accessing NFS Filesystems in Cockpit | 164 |
| 19.8 Summary | 165 |
| 20. Sharing Files between RHEL 8 and Windows Systems with Samba..... | 167 |

Table of Contents

| | |
|--|------------|
| 20.1 Accessing Windows Resources from the GNOME Desktop..... | 167 |
| 20.2 Samba and Samba Client..... | 168 |
| 20.3 Installing Samba on a RHEL 8 System | 168 |
| 20.4 Configuring the RHEL 8 Firewall to Enable Samba | 168 |
| 20.5 Configuring the <i>smb.conf</i> File..... | 169 |
| 20.5.1 Configuring the [global] Section..... | 169 |
| 20.5.2 Configuring a Shared Resource | 170 |
| 20.5.3 Removing Unnecessary Shares | 170 |
| 20.6 Configuring SELinux for Samba | 170 |
| 20.7 Creating a Samba User | 172 |
| 20.8 Testing the <i>smb.conf</i> File..... | 173 |
| 20.9 Starting the Samba and NetBIOS Name Services | 174 |
| 20.10 Accessing Samba Shares..... | 175 |
| 20.11 Accessing Windows Shares from RHEL 8 | 176 |
| 20.12 Summary | 178 |
| 21. An Overview of Virtualization Techniques | 179 |
| 21.1 Guest Operating System Virtualization | 179 |
| 21.2 Hypervisor Virtualization | 180 |
| 21.2.1 Paravirtualization | 181 |
| 21.2.2 Full Virtualization..... | 182 |
| 21.2.3 Hardware Virtualization | 182 |
| 21.3 Virtual Machine Networking..... | 183 |
| 21.4 Summary | 183 |
| 22. Installing KVM Virtualization on RHEL 8..... | 185 |
| 22.1 An Overview of KVM | 185 |
| 22.2 KVM Hardware Requirements..... | 185 |
| 22.3 Preparing RHEL 8 for KVM Virtualization | 186 |
| 22.4 Verifying the KVM Installation..... | 186 |
| 22.5 Summary | 188 |
| 23. Creating KVM Virtual Machines using Cockpit and virt-manager..... | 189 |
| 23.1 Installing the Cockpit Virtual Machines Module | 189 |
| 23.2 Creating a Virtual Machine in Cockpit | 189 |
| 23.3 Starting the Installation | 191 |
| 23.4 Working with Storage Volumes and Storage Pools..... | 193 |
| 23.5 Creating a Virtual Machine using virt-manager | 195 |
| 23.6 Starting the Virtual Machine Manager..... | 196 |
| 23.7 Configuring the KVM Virtual System | 197 |
| 23.8 Starting the KVM Virtual Machine | 200 |
| 23.9 Summary | 201 |
| 24. Creating KVM Virtual Machines with virt-install and virsh | 203 |

| | |
|--|------------|
| 24.1 Running virt-install to build a KVM Guest System | 203 |
| 24.2 An Example RHEL 8 virt-install Command | 204 |
| 24.3 Starting and Stopping a Virtual Machine from the Command-Line | 205 |
| 24.4 Creating a Virtual Machine from a Configuration File..... | 205 |
| 24.5 Summary | 205 |
| 25. Creating a RHEL 8 KVM Networked Bridge Interface | 207 |
| 25.1 Getting the Current Network Manager Settings | 207 |
| 25.2 Creating a Network Manager Bridge from the Command-Line | 209 |
| 25.3 Declaring the KVM Bridged Network | 210 |
| 25.4 Using a Bridge Network in a Virtual Machine | 211 |
| 25.5 Creating a Bridge Network using nm-connection-editor..... | 212 |
| 25.6 Summary | 215 |
| 26. Managing KVM using the virsh Command-Line Tool..... | 217 |
| 26.1 The virsh Shell and Command-Line..... | 217 |
| 26.2 Listing Guest System Status | 218 |
| 26.3 Starting a Guest System | 219 |
| 26.4 Shutting Down a Guest System | 219 |
| 26.5 Suspending and Resuming a Guest System | 219 |
| 26.6 Saving and Restoring Guest Systems | 219 |
| 26.7 Rebooting a Guest System | 220 |
| 26.8 Configuring the Memory Assigned to a Guest OS | 220 |
| 26.9 Summary | 220 |
| 27. An Introduction to Linux Containers..... | 221 |
| 27.1 Linux Containers and Kernel Sharing..... | 221 |
| 27.2 Container Uses and Advantages..... | 222 |
| 27.3 RHEL 8 Container Tools | 223 |
| 27.4 Container Catalogs, Repositories and Registries | 223 |
| 27.5 Container Networking..... | 224 |
| 27.6 Summary | 224 |
| 28. Working with Containers on RHEL 8 | 225 |
| 28.1 Installing the Container Tools | 225 |
| 28.2 Logging in to the Red Hat Container Registry..... | 225 |
| 28.3 Pulling a Container Image | 225 |
| 28.4 Running the Image in a Container | 227 |
| 28.5 Managing a Container | 228 |
| 28.6 Saving a Container to an Image | 229 |
| 28.7 Removing an Image from Local Storage | 229 |
| 28.8 Removing Containers | 230 |
| 28.9 Building a Container with Buildah | 230 |
| 28.10 Building a Container from Scratch..... | 230 |

Table of Contents

| | |
|--|------------|
| 28.11 Container Bridge Networking..... | 231 |
| 28.12 Managing Containers in Cockpit..... | 233 |
| 28.13 Summary | 235 |
| 29. Setting Up a RHEL 8 Web Server | 237 |
| 29.1 Requirements for Configuring a RHEL 8 Web Server | 237 |
| 29.2 Installing the Apache Web Server Packages | 237 |
| 29.3 Configuring the Firewall | 238 |
| 29.4 Port Forwarding | 238 |
| 29.5 Starting the Apache Web Server | 238 |
| 29.6 Testing the Web Server | 238 |
| 29.7 Configuring the Apache Web Server for your Domain | 239 |
| 29.8 The Basics of a Secure Web Site..... | 240 |
| 29.9 Configuring Apache for HTTPS | 241 |
| 29.10 Obtaining an SSL Certificate | 241 |
| 29.11 Summary | 244 |
| 30. Configuring a RHEL 8 Postfix Email Server..... | 245 |
| 30.1 The structure of the Email System | 245 |
| 30.1.1 Mail User Agent | 245 |
| 30.1.2 Mail Transfer Agent | 245 |
| 30.1.3 Mail Delivery Agent | 246 |
| 30.1.4 SMTP | 246 |
| 30.1.5 SMTP Relay | 246 |
| 30.2 Configuring a RHEL 8 Email Server | 246 |
| 30.3 Postfix Pre-Installation Steps | 246 |
| 30.4 Firewall/Router Configuration..... | 247 |
| 30.5 Installing Postfix on RHEL 8 | 247 |
| 30.6 Configuring Postfix | 247 |
| 30.7 Configuring DNS MX Records | 249 |
| 30.8 Starting Postfix on a RHEL 8 System..... | 249 |
| 30.9 Testing Postfix..... | 249 |
| 30.10 Sending Mail via an SMTP Relay Server..... | 250 |
| 30.11 Summary | 251 |
| 31. Adding a New Disk Drive to a RHEL 8 System..... | 253 |
| 31.1 Mounted File Systems or Logical Volumes | 253 |
| 31.2 Finding the New Hard Drive | 253 |
| 31.3 Creating Linux Partitions | 254 |
| 31.4 Creating a File System on a RHEL 8 Disk Partition | 255 |
| 31.5 An Overview of Journaled File Systems..... | 255 |
| 31.6 Mounting a File System | 256 |
| 31.7 Configuring RHEL 8 to Automatically Mount a File System | 257 |

| | |
|--|------------|
| 31.8 Adding a Disk Using Cockpit | 257 |
| 31.9 Summary | 259 |
| 32. Adding a New Disk to a RHEL 8 Volume Group and Logical Volume | 261 |
| 32.1 An Overview of Logical Volume Management (LVM) | 261 |
| 32.1.1 Volume Group (VG)..... | 261 |
| 32.1.2 Physical Volume (PV) | 261 |
| 32.1.3 Logical Volume (LV) | 262 |
| 32.1.4 Physical Extent (PE) | 262 |
| 32.1.5 Logical Extent (LE) | 262 |
| 32.2 Getting Information about Logical Volumes | 262 |
| 32.3 Adding Additional Space to a Volume Group from the Command-Line | 264 |
| 32.4 Adding Additional Space to a Volume Group using Cockpit | 267 |
| 32.5 Summary | 268 |
| 33. Adding and Managing RHEL 8 Swap Space..... | 269 |
| 33.1 What is Swap Space? | 269 |
| 33.2 Recommended Swap Space for RHEL 8..... | 269 |
| 33.3 Identifying Current Swap Space Usage | 269 |
| 33.4 Adding a Swap File to a RHEL 8 System..... | 270 |
| 33.5 Adding Swap as a Partition | 271 |
| 33.6 Adding Space to a RHEL 8 LVM Swap Volume | 271 |
| 33.7 Adding Swap Space to the Volume Group | 272 |
| 33.8 Summary | 274 |
| 34. RHEL 8 System and Process Monitoring | 275 |
| 34.1 Managing Processes | 275 |
| 34.2 Real-time System Monitoring with htop..... | 279 |
| 34.3 Command-Line Disk and Swap Space Monitoring..... | 280 |
| 34.4 Summary | 281 |
| Index | 283 |

1. Introduction

Arguably one of the most highly regarded and widely used enterprise level operating systems available today is the Red Hat Enterprise Linux (typically shortened to RHEL and pronounced *rell*) distribution. Not only is it considered to be among the most stable and reliable operating systems, it is also backed by the considerable resources and technical skills of Red Hat, Inc. RHEL 8 Essentials is designed to provide detailed information on the installation, use and administration of the Red Hat Enterprise Linux 8 distribution. For beginners, the book covers topics such as operating system installation, the basics of the GNOME desktop environment, configuring email and web servers and installing packages and system updates using App Streams. Additional installation topics such as dual booting with Microsoft Windows are also covered, together with all important security topics such as configuring a firewall and user and group administration.

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

1.1 Superuser Conventions

RHEL 8, in common with Linux in general, has two types of user account, one being a standard user account with restricted access to many of the administrative files and features of the operating system, and the other a superuser (*root*) account with elevated privileges. Typically, a user can gain root access either by logging in as the root user, or using the *su* - command and entering the root password. In the following example, a user is gaining root access via the *su* - command:

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

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

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

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

```
[neil@demo-server ~]$ dnf update  
Not root, Subscription Management repositories not updated  
Error: This command has to be run under the root user.
```

Introduction

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

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

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

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

```
[sudo] password for neil:
```

```
Updating Subscription Management repositories.
```

```
.  
.
```

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

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

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

```
$ date
```

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

```
# dnf install openssh
```

1.2 Opening a Terminal Window

If you are running RHEL 8 with the GNOME desktop and need to access a command-prompt, you will need to open a terminal window. This can be achieved by clicking on the Activities button (marked A in Figure 1-1) followed by the Terminal icon located in the dock (B):

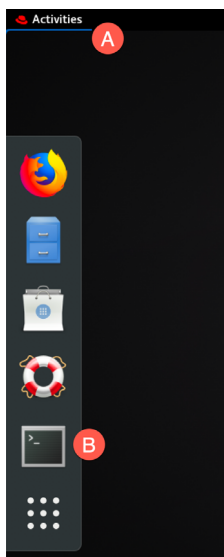


Figure 1-1

1.3 Editing Files

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

```
# nano <file>
```

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

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

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

```
GNU nano 2.9.3 /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nolog$
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management...:/run/systemd/netif:/usr$

[ Read 44 lines ]

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Figure 1-2

Introduction

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

```
# nano
```

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

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

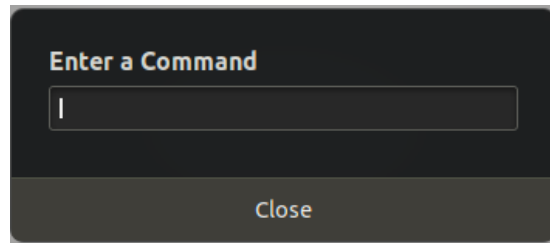


Figure 1-3

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

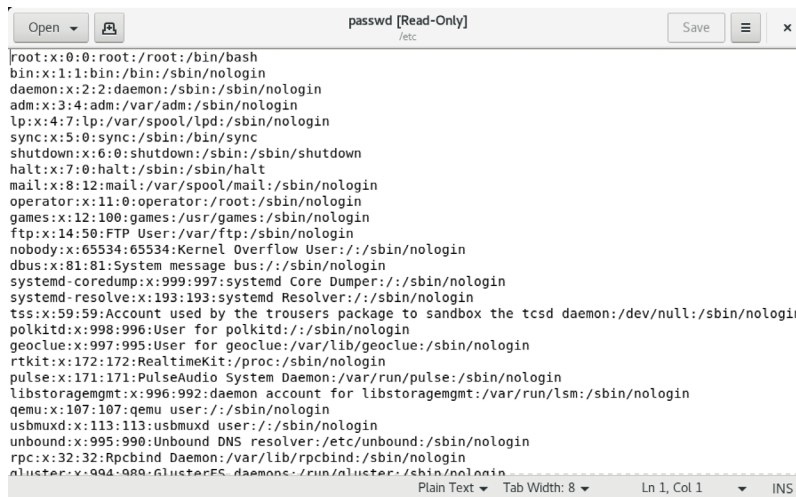


Figure 1-4

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/rhel8.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 Red Hat Linux

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

2.1 What exactly is Linux?

Linux is an operating system in much the same way that Windows is an operating system (and there are 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 the development of a new operating system named Multics. 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 designed for educational use with source code access provided to universities.

2.3 Who Created Linux?

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

A Brief History of Red Hat Linux

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 decided to begin 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 it seemed like the obvious solution was to combine his work with the kernel Linus had written to create a full operating system. This combination became known as GNU/Linux. Purists insist that Linux always be referred to as GNU/Linux (in fact, at one time, Richard Stallman refused to give press interviews to any publication which failed to refer to Linux as GNU/Linux). This is not unreasonable given that the GNU tools developed by the Free Software Foundation make up a significant and vital part of GNU/Linux. Unfortunately, most people and publications simply refer to Linux as Linux and this will probably always continue to be the case.

2.4 The Early Days of Red Hat

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

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

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

2.5 Red Hat Support

Early releases of Red Hat Linux were shipped to customers on floppy disks and CDs (this, of course, predated the widespread availability of broadband internet connections). When users encountered problems with the software they were only able to contact Red Hat by email. In fact, Bob Young often jokes that this was effective in limiting support requests since, by the time a customer realized they needed help, their computer was usually inoperative and therefore unavailable to be used to send an email message seeking assistance from Red Hat’s support

team. In later years Red Hat provided better levels of support tied to paid subscriptions and now provides a variety of support levels ranging from “self help” (no support) up to premium support.

2.6 Open Source

Red Hat Enterprise Linux 8 is the current commercial offering from Red Hat and is primarily targeted at corporate, mission critical installations. It is also the cornerstone of an expanding ecosystem of products and services offered by Red Hat. RHEL is an open source product in that you can download the source code free of charge and build the software yourself if you wish to do so (a task not to be undertaken lightly). If, however, you wish to download a pre-built, ready to install binary version of the software (either with or without support), you have to pay for it.

2.7 The Fedora Project

Red Hat also sponsors the Fedora Project, the goal of which is to provide access to a free Linux operating system (in both source and binary distributions) in the form of Fedora Linux. Fedora Linux also serves as a proving ground for many of the new features that are eventually adopted into the Red Hat Enterprise Linux operating system family. Red Hat Enterprise Linux 8.0, for example, was based to a large extent on Fedora 29.

2.8 CentOS - The Free Alternative

For users unable to afford a Red Hat Enterprise Linux subscription, another option is provided in the form of the CentOS operating system. The CentOS project, originally a community driven effort but now a collaboration with 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. Of course, while CentOS provides an operating system that is identical to RHEL in many ways, it does not include access to Red Hat technical support.

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.

3. Installing RHEL 8 on a Clean Disk Drive

There are now two ways in which a RHEL 8 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 that is already optimized for the cloud platform, and using that as the basis for the RHEL system, there is no need to perform a manual operating system installation in this situation.

If, on the other hand, you plan to install RHEL 8 on your own hardware, the first step on the path to learning about Red Hat Enterprise Linux 8 involves installing the operating system.

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

In this chapter we will be covering the clean disk approach to installation from local or remote installation media. Dual boot installation with a Windows 10 system will be covered in “*Dual Booting RHEL 8 with Windows*”.

3.1 Obtaining the Red Hat Enterprise Linux Installation Media

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

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

Members of the Red Hat Developer program are also provided with access to RHEL 8. Details on joining this program can be found on the Red Hat Developers web site:

<https://developers.redhat.com>

The installation media can be downloaded either as a DVD or boot ISO image. The DVD ISO image is self-contained including all of the packages necessary to install a RHEL system and is

Installing RHEL 8 on a Clean Disk Drive

named using the following convention:

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

For example, the RHEL 8.2 DVD image for 64-bit Intel systems is named as follows:

```
rhel-8.2-x86_64-dvd.iso
```

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

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

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

3.2 Writing the ISO Installation Image to a USB Drive

These days it is more likely that an operating system installation will be performed from a USB drive than from a DVD. Having downloaded the ISO installation image for RHEL 8, 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 to identify 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:

```
[445597.988045] 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
```

From this output we can tell 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
```

Installing RHEL 8 on a Clean Disk Drive

```
/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) to complete 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 is ready to be used to install RHEL 8:

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

3.2.2 macOS

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

```
$ diskutil list
```

```
/dev/disk0 (internal, physical):
```

| #: | TYPE | NAME | SIZE | IDENTIFIER |
|----|-----------------------|-------|-----------|------------|
| 0: | GUID_partition_scheme | | *1.0 TB | disk0 |
| 1: | EFI | EFI | 209.7 MB | disk0s1 |
| 2: | Apple_APFS Container | disk2 | 1000.0 GB | disk0s2 |

```
/dev/disk1 (internal):
```

| #: | TYPE | NAME | SIZE | IDENTIFIER |
|----|-----------------------|-------|----------|------------|
| 0: | GUID_partition_scheme | | 28.0 GB | disk1 |
| 1: | EFI | EFI | 314.6 MB | disk1s1 |
| 2: | Apple_APFS Container | disk2 | 27.7 GB | disk1s2 |

```
/dev/disk2 (synthesized):
```

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

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

Installing RHEL 8 on a Clean Disk Drive

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

A number of free tools are available for Windows 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, launch the writer tool and select the *Custom image* option as highlighted in Figure 3-1.

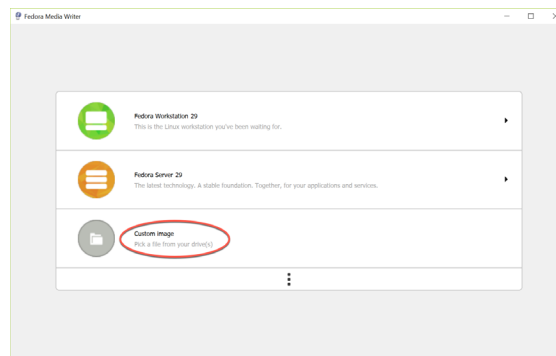


Figure 3-1

In the resulting file selection dialog, navigate to and select the RHEL 8 installation ISO image and click on the *Open* button. After selecting the image, a dialog will appear within which the image can be written to the USB drive. Select the target USB drive from the device menu before clicking on the *Write to Disk* button:

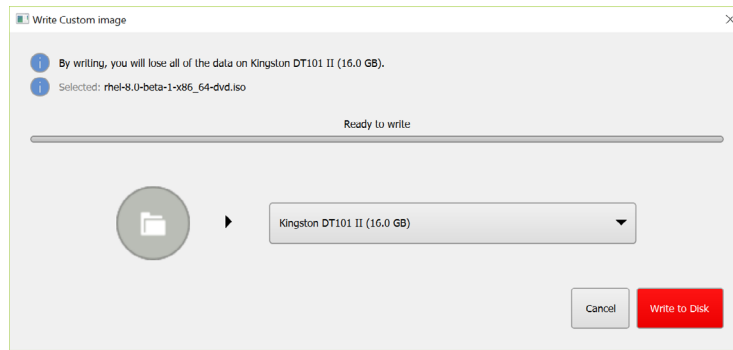


Figure 3-2

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

3.3 Installing Red Hat Enterprise Linux 8

Insert the RHEL 8 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:

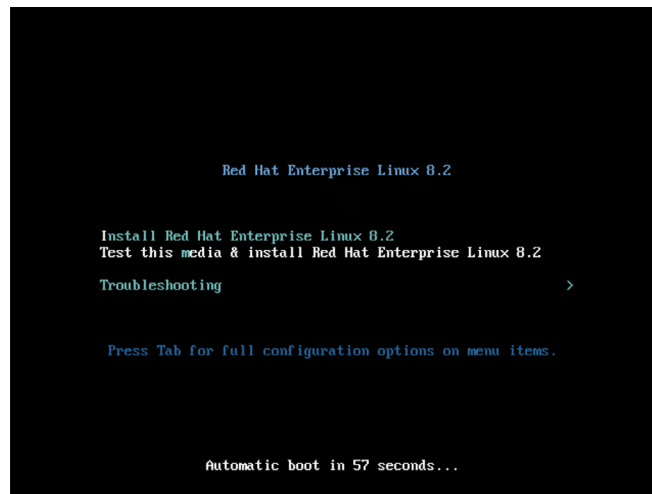


Figure 3-3

Use the arrow keys to navigate between the options and make a selection with the <Enter> key. If the *Troubleshooting* option is selected the screen shown in Figure 3-4 will appear including options to boot from the current operating system on the local drive (if one is installed), test the system memory, or rescue an installed RHEL 8 system. An option is also available to perform the installation in basic graphics mode for systems without a graphical console:

Installing RHEL 8 on a Clean Disk Drive

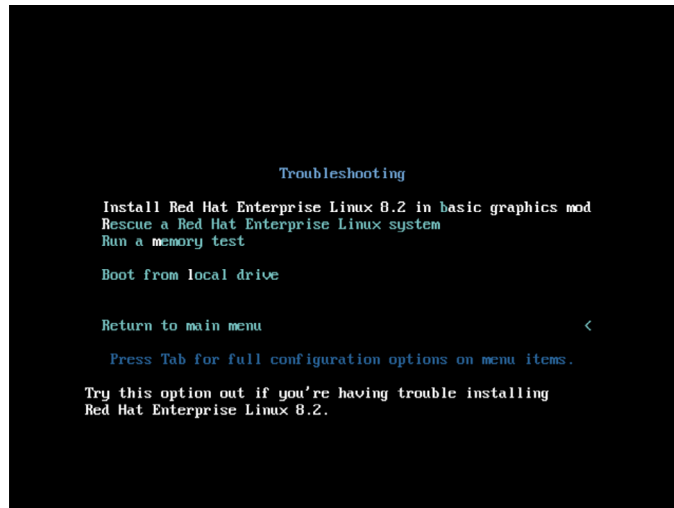


Figure 3-4

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

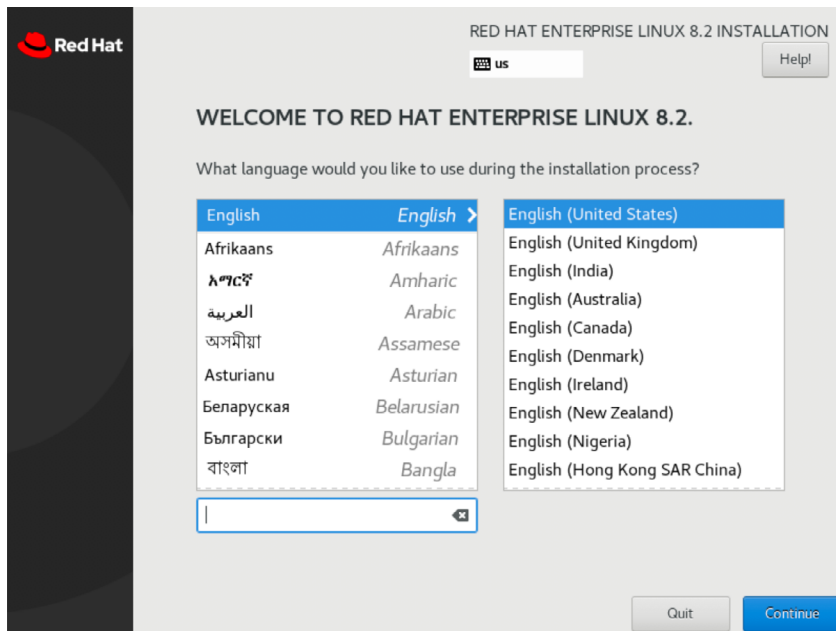


Figure 3-5

Select your preferred language on the first screen before clicking *Continue* to proceed to the main installation screen as shown in Figure 3-6:

Installing RHEL 8 on a Clean Disk Drive

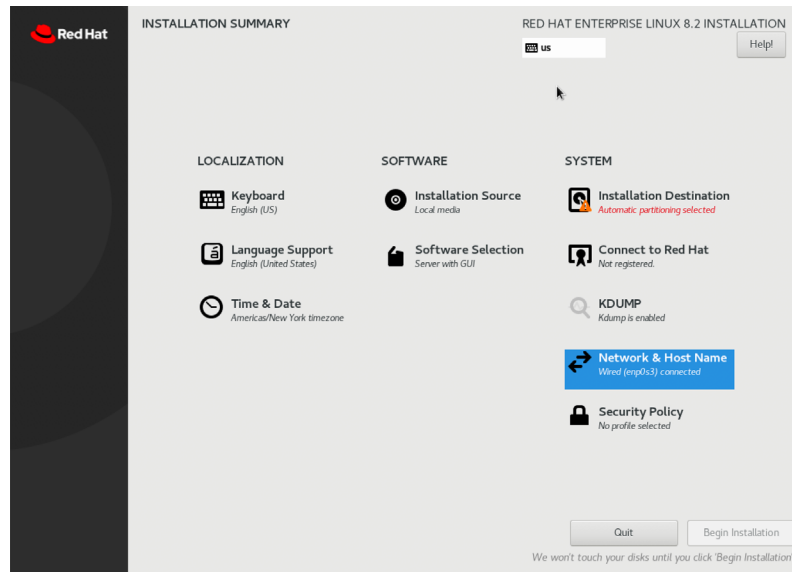


Figure 3-6

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

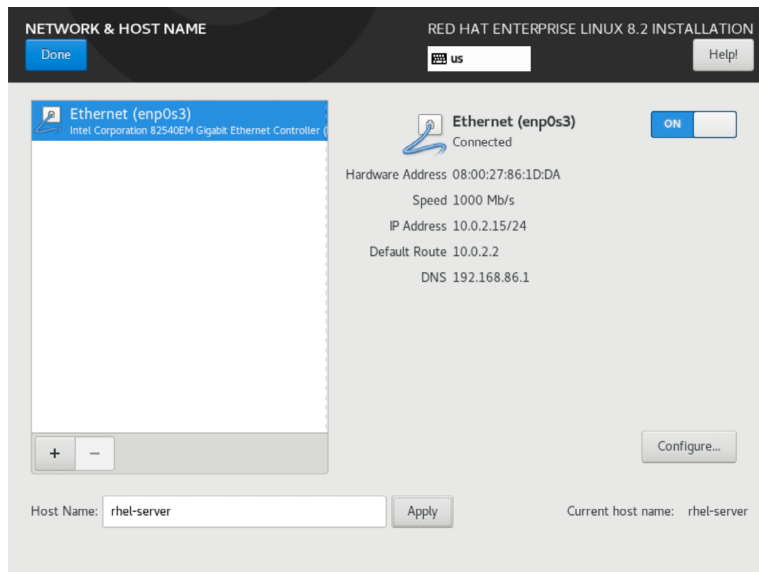


Figure 3-7

If your network connection requires additional settings, click on the *Configure...* button to access the advanced network settings screen illustrated in Figure 3-8:

Installing RHEL 8 on a Clean Disk Drive

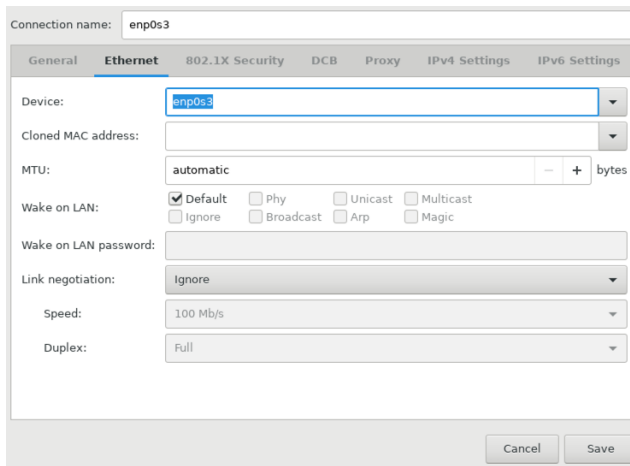


Figure 3-8

Once the host name 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 option 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 being performed from local media. To perform 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 to Share RHEL 8 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, a Server with GUI installation of RHEL 8 will be performed by the installer. To select a different installation configuration, choose the *Software Selection* option to display the screen shown below:

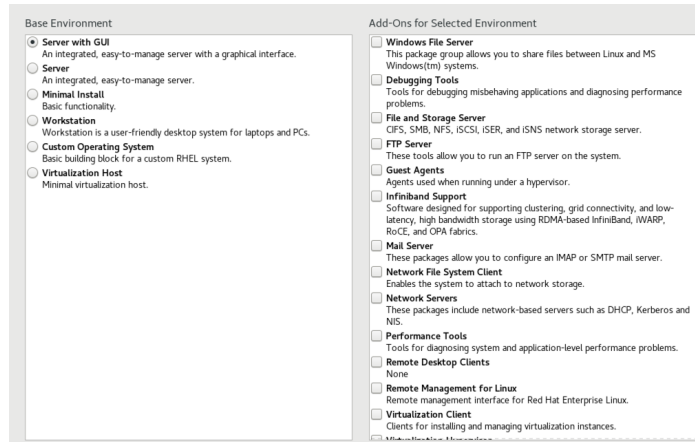


Figure 3-9

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 RHEL 8 system is intended to be used with a graphical desktop environment, choose either the *Workstation* or *Server with GUI* option. 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 needed.

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, it is not necessary to select a policy on this screen.

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 be useful in identifying 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 RHEL 8

When the *Installation Destination* option is selected, the installer will present a screen similar to the one illustrated in Figure 3-10 below:

Installing RHEL 8 on a Clean Disk Drive

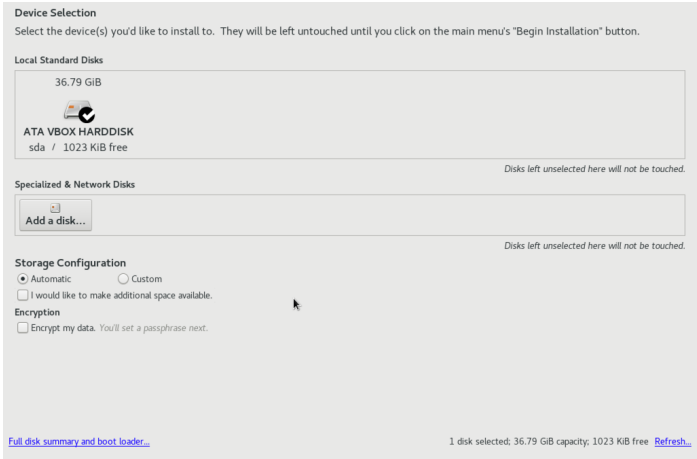


Figure 3-10

By default, the installer is configured to automatically install the operating system using the available space on the currently selected disk drive and to configure standard partition sizes. If the disk previously contained another operating system, these existing partitions will not be deleted, potentially leaving unused space on the drive after RHEL 8 has been installed. To remove any existing partitions so that they can be reclaimed and used by RHEL 8, enable the *I would like to make additional space available* option and click on the *Done* button to display the dialog shown in Figure 3-11.

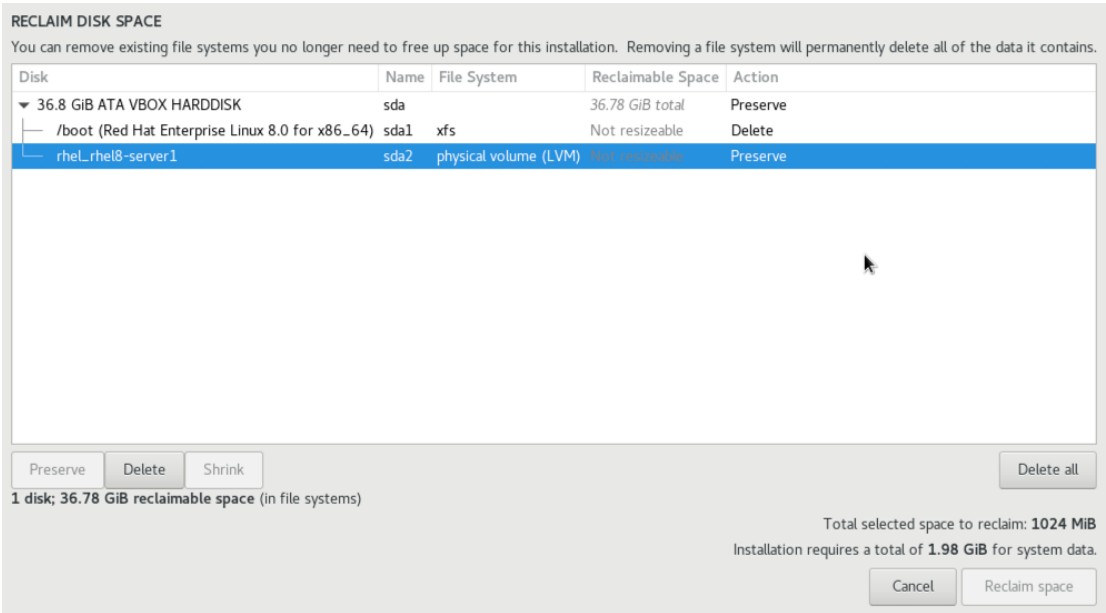


Figure 3-11

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 that are to be removed have been selected, click on the *Reclaim Space* button to perform the deletion. The reclaimed space will now be used automatically by the RHEL 8 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 3-12):

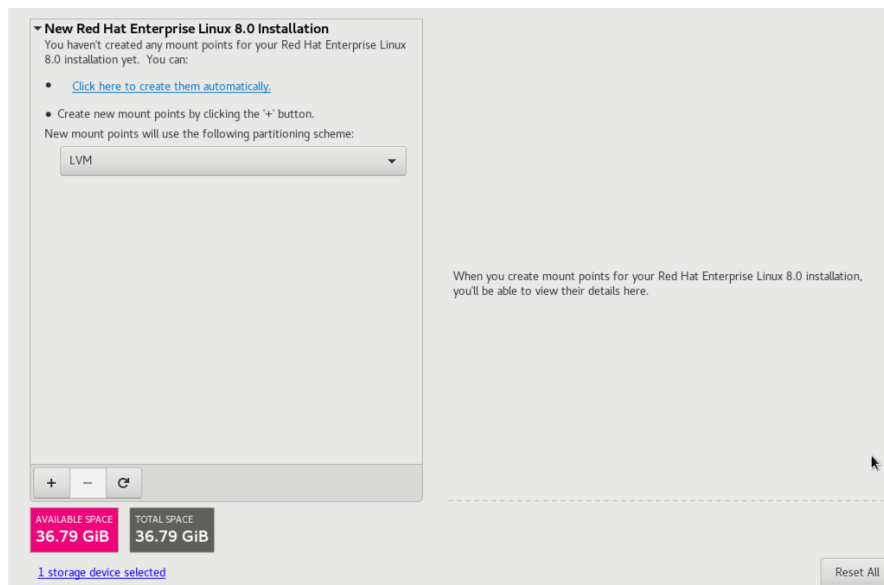


Figure 3-12

The manual partitioning screen provides the option to configure the disk using Logical Volume Management (LVM) or standard partitioning. LVM is the recommended option because it provides flexibility in terms of managing partition sizes once the system is up and running (LVM is covered in detail in the chapter entitled “*Adding a New Disk to a RHEL 8 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. To manually create partitions and allocate mount points, click on the + button to declare and assign each partition manually.

Another option is to select automatic partition creation, and then use the resulting screen to manually change the partition configuration as needed. Figure 3-13, 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 RHEL 8 on a Clean Disk Drive

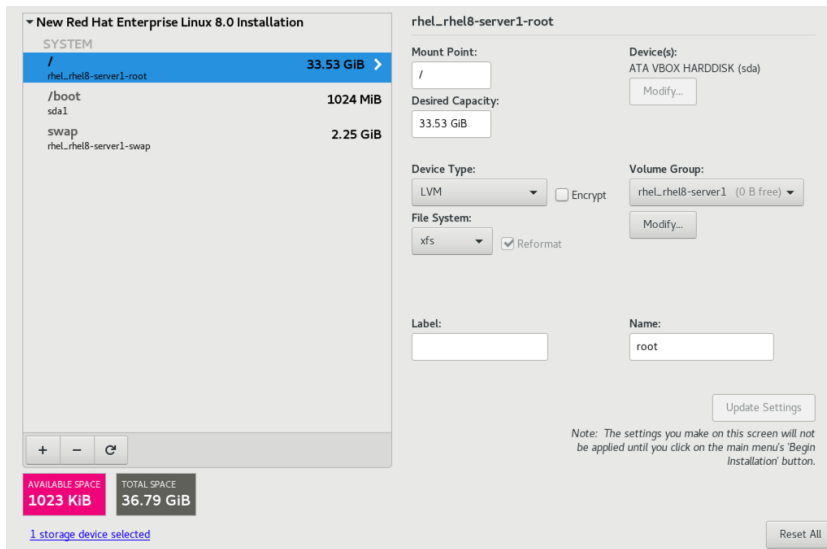


Figure 3-13

Once the disk has been configured, click on *Done* and review the summary of the changes to be made to the disk (Figure 3-14) before clicking on the *Accept Changes* button:

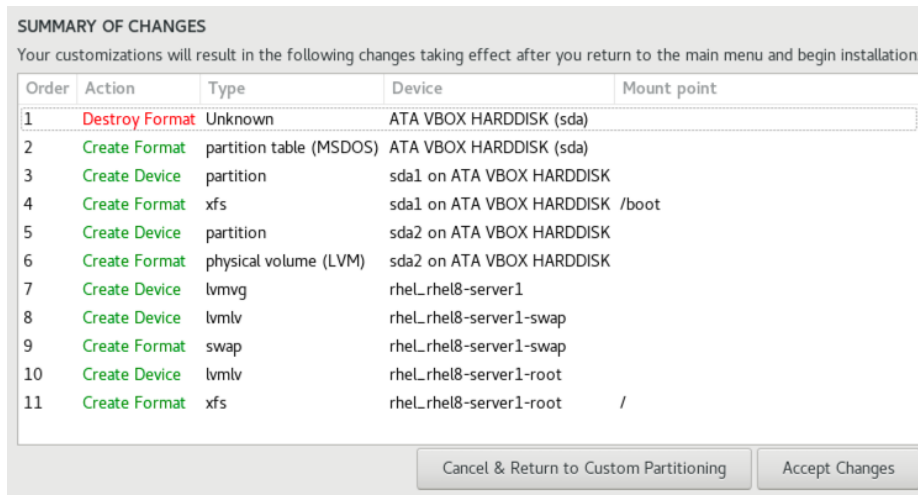
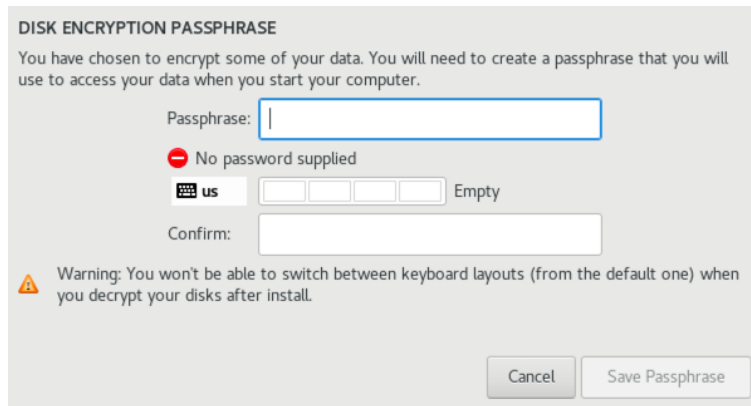


Figure 3-14


RHEL 8 also allows the contents of the disk 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 3-15:




DISK ENCRYPTION PASSPHRASE

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

Passphrase:

 No password supplied

 **us** Empty

Confirm:


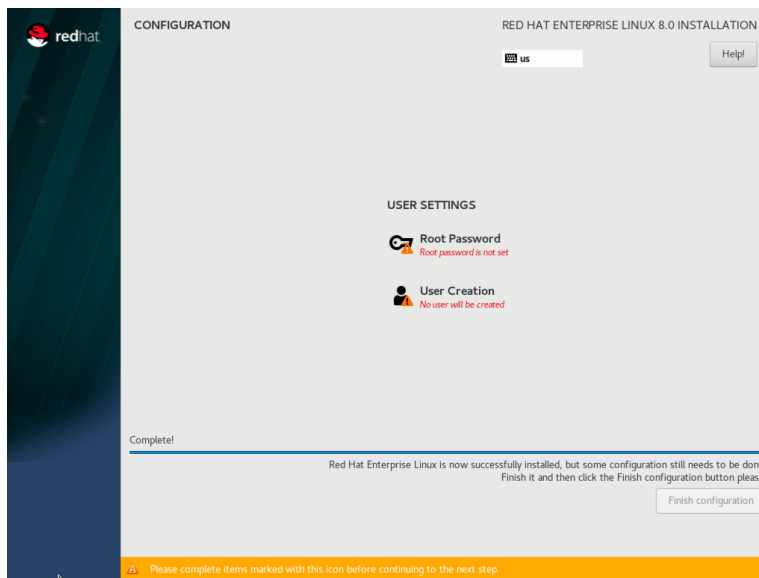
 Warning: You won't be able to switch between keyboard layouts (from the default one) when you decrypt your disks after install.



Figure 3-15

3.5 The Physical Installation


Having made the appropriate package selections, clicking *Begin Installation* will start the process of partitioning the disks and installing the packages that match the chosen installation settings. During this phase, a screen will appear (Figure 3-16) seeking the creation of a root password and an optional user account:




CONFIGURATION RED HAT ENTERPRISE LINUX 8.0 INSTALLATION

  **us**

USER SETTINGS

 **Root Password**
Root password is not set

 **User Creation**
No user will be created

Complete!

Red Hat Enterprise Linux is now successfully installed, but some configuration still needs to be done. Finish it and then click the Finish configuration button please.


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

Figure 3-16

The root, or super-user account, is a special user that has administrative privileges on the system. While you will generally use your own account to log into the system, you will need to gain root privileges in order to configure the system and to perform other administrative tasks. Enter a root password, create a user account and wait for the installation to complete.

Once all the system packages have been installed and configured, remove the installation media

Installing RHEL 8 on a Clean Disk Drive

and click *Reboot* to restart the system.

3.6 Final Configuration Steps

What appears on the console when the system has started up will depend on whether the *Workstation* or *Server with GUI* option was selected from the Software Selection installation screen. If either of these options was selected during installation, the GNOME Display Manager (GDM) login screen will appear. 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 that was created during the final steps of the installation process.

In the case of a *Workstation* or *Server with GUI* installation, the GNOME initial setup tool will launch providing configuration options such as language, keyboard layout, location services and connections to online accounts including Google and Facebook. After these settings have been configured (or skipped), the GNOME desktop welcome screen will appear containing links to tutorials on how to use the desktop environment.

Before any updates or additional packages can be installed, the system must be registered with Red Hat. This can be performed at the command prompt or, in the case of the desktop environment, a Terminal window. To launch a Terminal session in the desktop, click *Activities* in the menu bar (marked A in Figure 3-17), enter *Terminal* into the search field (B) and select the Terminal launch icon from the results list (C).

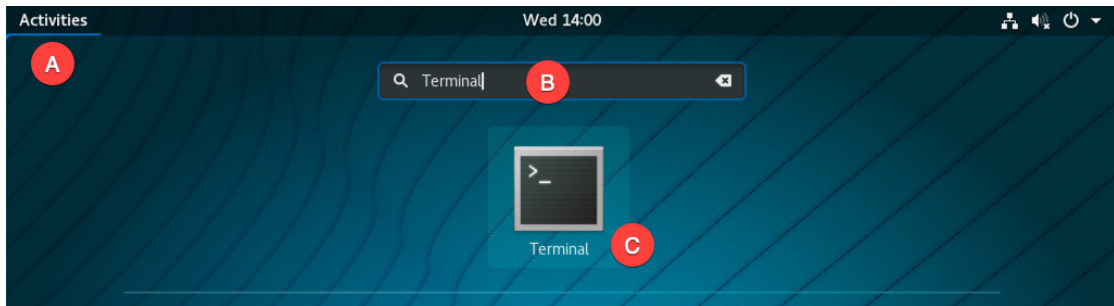


Figure 3-17

At the prompt, use the *su* command to gain root privileges:

```
$ su -
```

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

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

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


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

3.7 Installing Updates

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

Best practices dictate that the first step after installing RHEL is to make sure 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, simply run the following command:

```
# dnf check-update
```

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

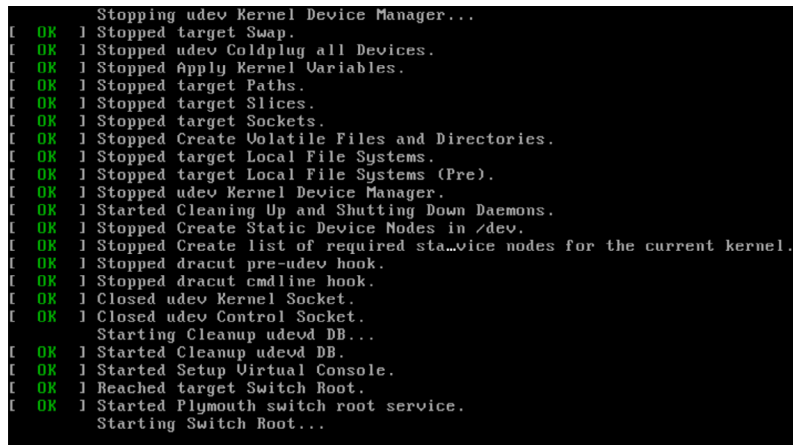
```
# dnf update
```

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

Once the update is complete the installation is essentially finished and RHEL 8 is ready for use.

3.8 Displaying Boot Messages

During the boot process, RHEL 8 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 3-18), simply press the keyboard Esc key while the system is starting:



```
Stopping udev Kernel Device Manager...
[ OK ] Stopped target Swap.
[ OK ] Stopped udev Coldplug all Devices.
[ OK ] Stopped Apply Kernel Variables.
[ OK ] Stopped target Paths.
[ OK ] Stopped target Slices.
[ OK ] Stopped target.Sockets.
[ OK ] Stopped Create Volatile Files and Directories.
[ OK ] Stopped target Local File Systems.
[ OK ] Stopped target Local File Systems (Pre).
[ OK ] Stopped udev Kernel Device Manager.
[ OK ] Started Cleaning Up and Shutting Down Daemons.
[ OK ] Stopped Create Static Device Nodes in /dev.
[ OK ] Stopped Create list of required static device nodes for the current kernel.
[ OK ] Started dracut pre-udev hook.
[ OK ] Stopped dracut cmdline hook.
[ OK ] Closed udev Kernel Socket.
[ OK ] Closed udev Control Socket.
Starting Cleanup udevd DB...
[ OK ] Started Cleanup udevd DB.
[ OK ] Started Setup Virtual Console.
[ OK ] Reached target Switch Root.
[ OK ] Started Plymouth switch root service.
Starting Switch Root...
```

Figure 3-18

Installing RHEL 8 on a Clean Disk Drive

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 display 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 next time the system starts:

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

3.9 Summary

The first step in working with RHEL 8 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 in the form of 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.

4. Dual Booting RHEL 8 with Windows

Red Hat Enterprise Linux 8, just like most Linux distributions, will happily co-exist on a hard disk drive with just about any version of Windows up to and including Windows 10. This is a concept known as dual-booting. Essentially, when you power up the system, you will be presented with a menu providing the option to boot either your RHEL 8 installation or Windows. Obviously you can only run one operating system at a time, but it is worth noting that the files on the Windows partition of your disk drive will be available to you from RHEL 8 regardless of whether your Windows partition was formatted using NTFS, FAT16 or FAT32. To run RHEL 8 and Windows concurrently, consider installing RHEL in virtualization environment such as VirtualBox.

This installation method involves shrinking the size of the existing Windows partitions and then installing RHEL 8 into the reclaimed space. This chapter will assume that RHEL 8 is being installed on a system currently running Windows 10.

4.1 Partition Resizing

In order to accommodate RHEL on a disk drive that already contains a Windows installation, the first step involves shrinking the Windows partition to make some room. The recommended course of action is to use the Windows Disk Management interface to reduce the size of the partition before attempting to install RHEL 8.

To access Disk Management on Windows 10, right-click on the Start menu and select *Disk Management* from the resulting menu as highlighted in Figure 4-1:

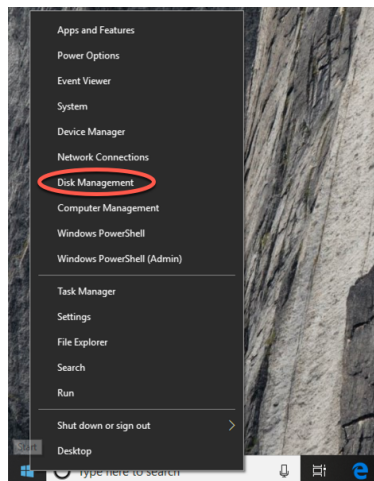


Figure 4-1

Dual Booting RHEL 8 with Windows

Once loaded, the Disk Management tool will display a graphical representation of the disk drives detected on the system:

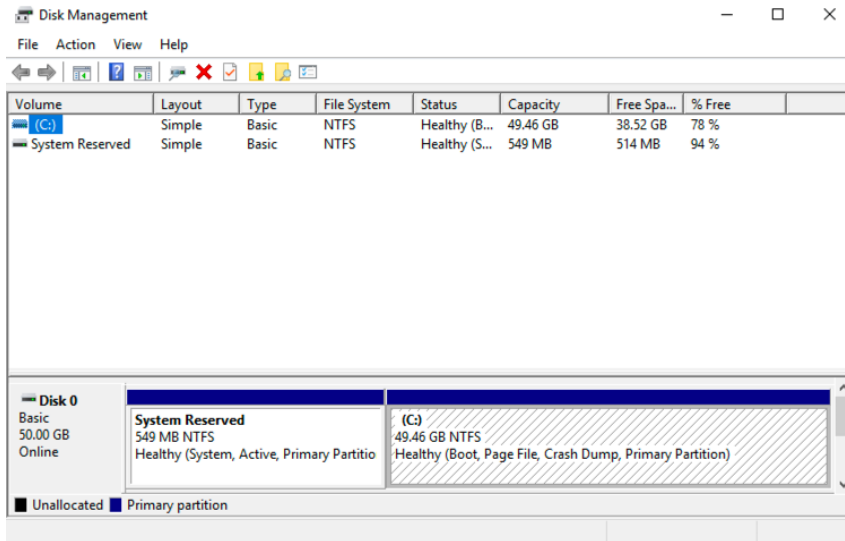


Figure 4-2

Right-click on the partition you wish to reduce in size and select *Shrink Volume...* from the popup menu. The tool will calculate the maximum amount by which the volume size can be reduced without data loss (a process that can take several minutes depending on the overall size of the partition). Once this analysis is complete, a dialog similar to the one in Figure 4-3 below will appear:

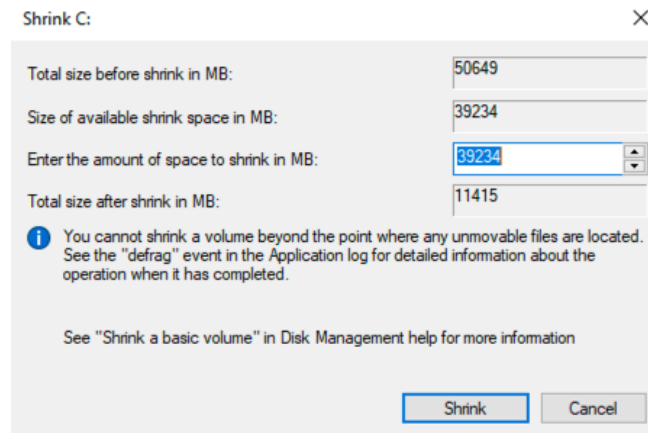


Figure 4-3

Specify a value in the *Enter amount of space to shrink in MB* field and click on the *Shrink* button to proceed. Once the resizing operation is complete, reboot using the RHEL 8 installation media (as outlined in *“Installing RHEL 8 on a Clean Disk Drive”*) and proceed with the installation making

use of the new free space. During the RHEL installation process, this can be achieved by selecting the *Installation Destination* option on the Installation Summary screen and making sure that the *Automatic* storage configuration option is selected. This will automatically install RHEL into the unallocated space that was created when the Windows partition was reduced in size.

4.2 Editing the RHEL 8 Boot Menu

Once installation of RHEL onto the disk is complete and the system has restarted, the standard RHEL boot menu will appear:

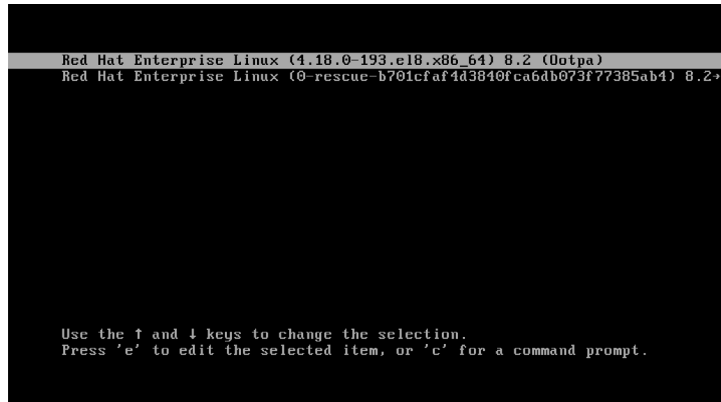


Figure 4-4

At this point, the boot menu only provides the option to boot the RHEL 8 system. Clearly an option also needs to be added so that the Windows system is still accessible. These steps will need to be taken from within the RHEL 8 system, so start the system and log in as root.

The first step is to identify the Windows boot partition on the disk drive using the *fdisk* command as follows:

```
# fdisk -l
```

```
.
.
```

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|----------|-----------|----------|-------|----|-----------------|
| /dev/sda1 | * | 2048 | 1126399 | 1124352 | 549M | 7 | HPFS/NTFS/exFAT |
| /dev/sda2 | | 1126400 | 53655551 | 52529152 | 25G | 7 | HPFS/NTFS/exFAT |
| /dev/sda3 | | 53655552 | 55752703 | 2097152 | 1G | 83 | Linux |
| /dev/sda4 | | 55752704 | 104857599 | 49104896 | 23.4G | 5 | Extended |
| /dev/sda5 | | 55754752 | 104857599 | 49102848 | 23.4G | 8e | Linux LVM |

In the above output */dev/sda3* through to */dev/sda5* are clearly the RHEL 8 partitions. The smaller */dev/sda1* partition is the Windows system partition, leaving */dev/sda2* as the Windows boot partition.

The RHEL 8 boot system works using partition indexes where the first partition is partition 0, the second is partition 1 and so on. As such, the Windows boot partition from the point of view of the boot menu is located on hard disk 0 at partition index location 1 and is defined in the boot

Dual Booting RHEL 8 with Windows

configuration as “hd0,1”.

The boot menu configuration settings are stored in the `/boot/grub2/grub.cfg` file which is generated using the `grub2-mkconfig` tool based on the content of the configuration files located in the `/etc/grub.d` directory. To add the Windows boot option to the configuration, edit the `/etc/grub.d/40_custom` file which should read as follows:

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
```

Edit the file to add the following entry to the end of the `40_custom` file:

```
menuentry "Windows 10" {
    set root=(hd0,1)
    chainloader +1
}
```

Save the file and generate the boot menu as follows:

```
# grub2-mkconfig --output=/boot/grub2/grub.cfg
Generating grub configuration file ...
done
```

On the next reboot, Windows will now be included as a boot option as shown in Figure 4-5:

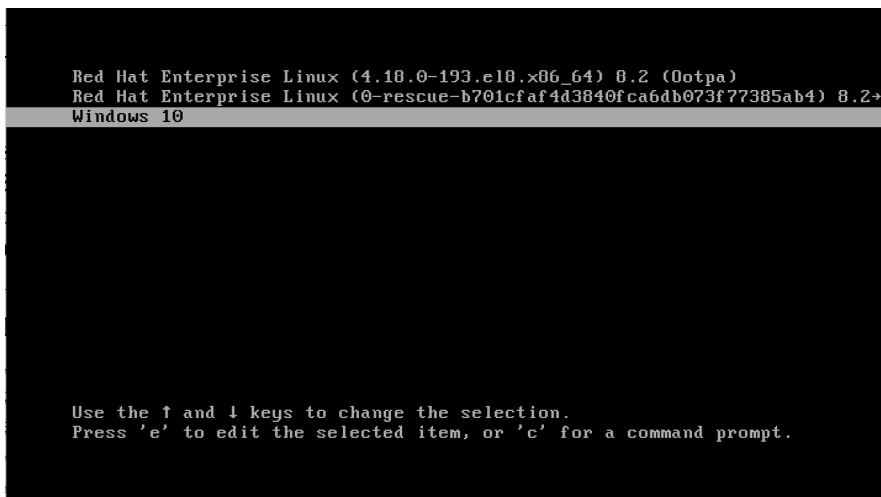


Figure 4-5

4.3 Changing the Default Boot Option

When the system starts, the boot options screen will appear and wait 5 seconds for the user to make an operating system choice. If no selection has been made before the timeout elapses, the default operating system will be started. On a newly configured system, the default operating

system will be the standard (non-rescue) RHEL 8 image. This default can, however, be changed from within RHEL.

A range of boot configuration options (including the 5 second timeout and the boot RHGB settings outlined in “*Installing RHEL 8 on a Clean Disk Drive*”) are declared in the `/etc/default/grub` file which reads as follows on a new installation:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root
rd.lvm.lv=cl/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

The first step in changing the default boot system is to declare the `GRUB_SAVEDefault` setting within this file:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_SAVEDefault=true
.
.
```

This setting allows a new default value to be saved within the boot configuration. Next, run the `grub2-set-default` command to change the default setting using a numbering system that counts the first option as 0. For example, if the Windows 10 option is position 3 in the menu, the command to make Windows 10 the default boot option would read as follows:

```
# grub2-set-default 2
```

Check that the new setting has taken effect by running the following command:

```
# grub2-editenv list
saved_entry=2
kernelopts=root=/dev/mapper/cl-root ro crashkernel=auto resume=/dev/mapper/cl-
swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet
boot_success=0
boot_indeterminate=0
```

Note that the `saved_entry` value is now set to 2. After changing the default, regenerate the boot configuration file as follows:

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

Reboot the system and verify that the boot menu defaults to the Windows 10 option and that Windows loads after the timeout expires.

Dual Booting RHEL 8 with Windows

4.4 Accessing the Windows Partition from the Command-line

When running RHEL in a dual boot configuration it is still possible to access files located on the Windows partition. This can be achieved by manually mounting the partition from the command-line. Before doing so, however, some additional packages need to be installed on the system. First the *fuse* kernel module needs to be downloaded and installed:

```
# dnf install fuse
# modprobe fuse
```

Next, the Fuse NTFS driver needs to be installed. Unfortunately, this package is not included in the standard RHEL 8 repositories so the Extra Packages for Enterprise Linux (EPEL) repository needs to be added to the system as follows:

```
# dnf install wget
# wget http://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
# rpm -ivh epel-release-latest-8.noarch.rpm
```

With the EPEL repository added, the driver can now be installed:

```
# dnf install fuse-ntfs-3g
```

Once the requisite packages are installed, the next step is to create a directory to use as the mount point for our Windows partition. In this example we will create a directory named */mnt/windows*:

```
# mkdir /mnt/windows
```

In order to identify the device name that has been assigned to the Windows partition, use the *fdisk* command as follows:

```
# fdisk -l
.
.

```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|------|----------|----|-----------|
| /dev/sda1 | * | 1 | 13 | 102400 | 7 | HPFS/NTFS |
| /dev/sda2 | | 13 | 1400 | 11141120 | 7 | HPFS/NTFS |
| /dev/sda3 | | 1400 | 1464 | 512000 | 83 | Linux |
| /dev/sda4 | | 1464 | 2611 | 9214976 | 5 | Extended |
| /dev/sda5 | | 1464 | 2611 | 9213952 | 8e | Linux LVM |

In the above output, the main Windows partition containing the files we need access to is represented by */dev/sda2*. Next, we need to run the *mount* command (assuming the Windows partition is */dev/sda2* and NTFS format) as follows:

```
# mount /dev/sda2 /mnt/windows
```

Check that the mount was successful by listing the contents of the top level directory of the mount point:

```
# ls /mnt/windows
'$Recycle.Bin'          ProgramData              swapfile.sys
'Documents and Settings' 'Program Files'          'System Volume Information'
pagefile.sys            'Program Files (x86)'    Users
```


To automate the mount each time the system is booted, simply add the appropriate mount line to the */etc/fstab* file:

```
/dev/sda2 /mnt/windows ntfs defaults 0 0
```

To unmount the Windows file system at any time:

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

4.5 Accessing the Windows Partition from the Desktop

The first step in mounting a partition using the desktop environment is to launch the disks utility. With the desktop loaded, right-click on the desktop background and select Open Terminal from the resulting menu. Within the terminal window, gain root privileges before launching the Disks utility as follows:

```
$ gnome-disks
```

Once the disks tool has launched, a window similar to that shown in Figure 4-6 will appear:

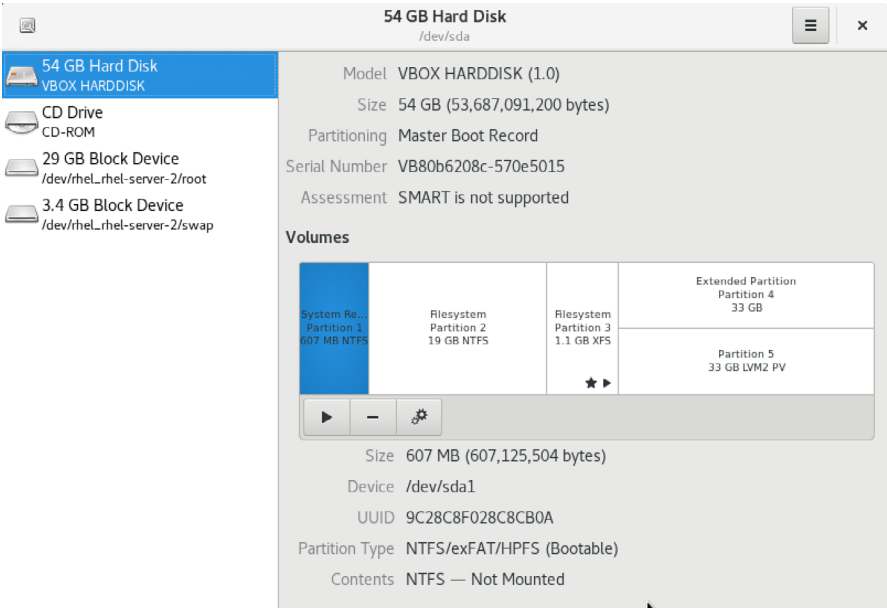


Figure 4-6

To view information about a partition, select it from the graphical representation in the Volumes section of the dialog. In the above example, the Windows system partition is selected and highlighted. To mount the partition so that the contents can be accessed from within RHEL 8, select the partition and click on the options button indicated in Figure 4-7 to display the menu:

Dual Booting RHEL 8 with Windows

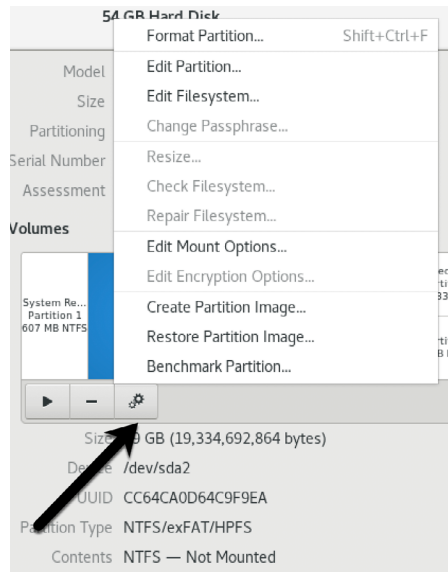


Figure 4-7

From the menu, select *Edit Mount Options...* to display the dialog shown in Figure 4-8. Turn off the User Session Defaults switch and make sure that the *Show in user interface* option is enabled. Within the Mount Point text field, change the path to `/mnt/windows`. If the partition needs to be automatically mounted each time the system reboots, also enable the *Mount at system startup* option:

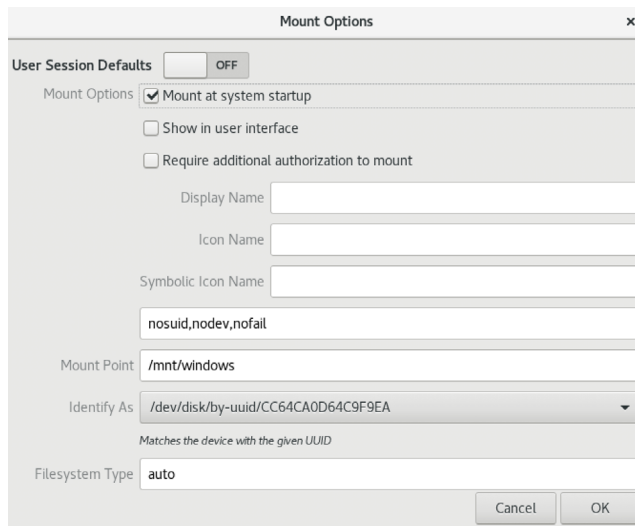


Figure 4-8

With the changes made, click on the OK button to return to the main dialog. To mount the partition, simply click on the mount button highlighted in Figure 4-9 below:

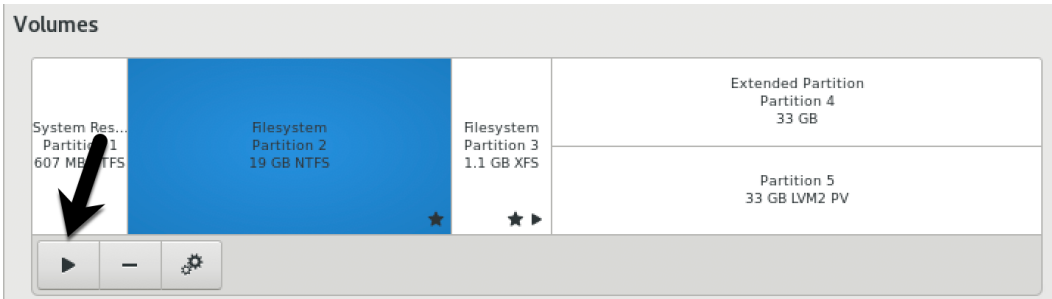


Figure 4-9

To locate the mount point, open the desktop Files app as highlighted in Figure 4-10:



Figure 4-10

Within the app, click on the Other Locations entry in the right-hand column (marked A in Figure 4-11 below) followed by the mounted volume from the list (B):

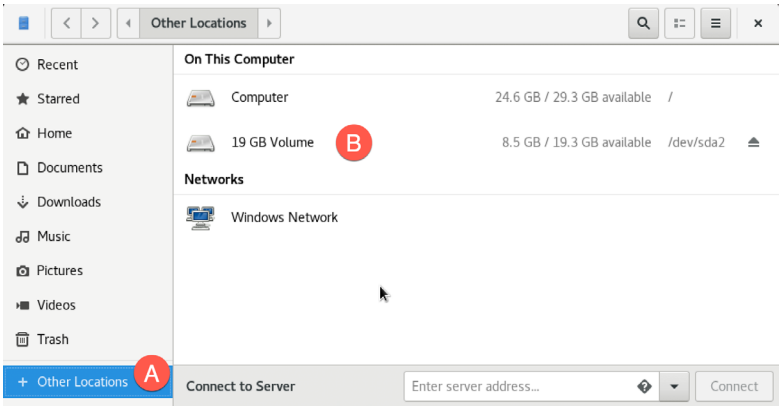


Figure 4-11

Dual Booting RHEL 8 with Windows

Once selected, the files and folders within the Windows partition will be listed:

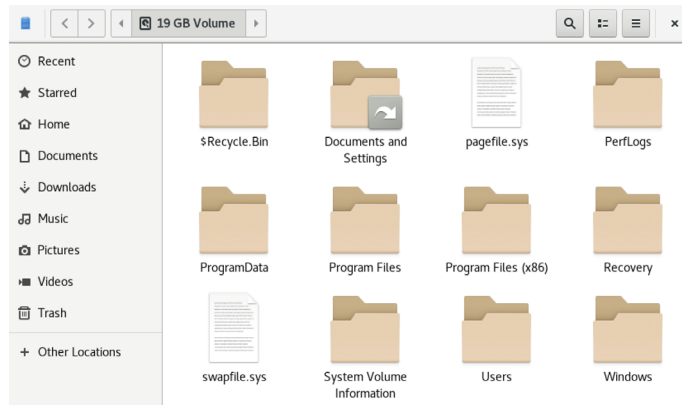


Figure 4-12

To unmount the partition, click on the mount button (which will have changed to a stop button) as shown in Figure 4-9 above.

4.6 Summary

RHEL 8 can safely co-exist on the same disk drive as a Windows operating system by creating a dual boot environment. This involves shrinking the amount of space occupied by the Windows system to make room for RHEL 8 before performing the installation. Once RHEL has been installed, the boot menu configuration must be modified to include the option to boot from Windows. To access the Windows filesystem from within RHEL, the Fuse NTFS driver needs to be installed and used to mount the Windows partitions using either the command-line or the `gnome-disks` tool.

5. Allocating Windows Disk Partitions to RHEL 8

In the previous chapter we looked at how to install Red Hat Enterprise Linux 8 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 RHEL 8 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 RHEL 8 with Windows*” chapter were followed to mount the Windows partition from within RHEL 8, 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 RHEL is to delete that partition. Before doing so, however, it is imperative that any data you need to keep is backed up from both the Windows and RHEL partitions. Having done that, it is safe to proceed with this chapter.

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

```
# fdisk -l
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe3673009
```

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|---------|----------|----------|------|----|-----------------|
| /dev/sda1 | * | 2048 | 1126399 | 1124352 | 549M | 7 | HPFS/NTFS/exFAT |
| /dev/sda2 | | 1126400 | 53655551 | 52529152 | 25G | 7 | HPFS/NTFS/exFAT |

Allocating Windows Disk Partitions to RHEL 8

```
/dev/sda3      53655552  55752703  2097152      1G 83 Linux
/dev/sda4      55752704 104857599 49104896 23.4G  5 Extended
/dev/sda5      55754752 104857599 49102848 23.4G 8e Linux LVM
```

In the above example output the system contains one physical disk drive referenced by device name `/dev/sda`. On that disk drive are five partitions accessed via the device names `/dev/sda1` through `/dev/sda5` respectively. Based on the values in the System column, there are two NTFS partitions. The first is the Windows system partition while the second, much larger, NTFS partition is the Windows boot partition containing the Windows operating system and user data. On some systems, an additional Windows Recovery partition may be listed and may also be deleted to free up space for RHEL 8. If such a partition exists, it will likely be listed as follows::

```
/dev/sda3      967012352 976771071   9758720    4.7G 27 Hidden NTFS WinRE
```

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

```
# fdisk /dev/sda
```

```
Welcome to fdisk (util-linux 2.32.1).
```

```
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/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
```

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

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

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

```
Disklabel type: dos
```

```
Disk identifier: 0xe3673009
```

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|----------|-----------|----------|-------|----|-----------------|
| /dev/sda1 | * | 2048 | 1126399 | 1124352 | 549M | 7 | HPFS/NTFS/exFAT |
| /dev/sda2 | | 1126400 | 53655551 | 52529152 | 25G | 7 | HPFS/NTFS/exFAT |
| /dev/sda3 | | 53655552 | 55752703 | 2097152 | 1G | 83 | Linux |
| /dev/sda4 | | 55752704 | 104857599 | 49104896 | 23.4G | 5 | Extended |
| /dev/sda5 | | 55754752 | 104857599 | 49102848 | 23.4G | 8e | Linux LVM |

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

Currently, the Windows system partition is listed as being the bootable partition. Since we will be deleting this partition, the Linux boot partition needs to be marked as bootable. In the above configuration, this is represented by `/dev/sda3`. Remaining within the *fdisk* tool, make this the bootable partition as follows:

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

```
Partition number (1,3-5, default 5): 3
```

The bootable flag on partition 3 is enabled now.

Before proceeding, make a note of the start and end addresses of the partitions we will be deleting (in other words the start of `/dev/sda1` and the sector before the start of `/dev/sda3`).

At the command prompt, delete the Windows partitions (these being partitions 1 and 2 on our example system):

```
Command (m for help): d
Partition number (1-5, default 5): 1
```

Partition 1 has been deleted.

```
Command (m for help): d
Partition number (2-5, default 5): 2
```

Partition 2 has been deleted.

Now that we have deleted the Windows partitions we need to create the new RHEL partition in the vacated disk space. The partition number must match the number of the partition removed (in this case 1) and is going to be a primary partition. It will also be necessary to enter the Start and End sectors of the partition exactly 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 type
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p): p
Partition number (1,2, default 1): 1
First sector (2048-104857599, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-53655551, default 53655551):
```

Created a new partition 1 of type 'Linux' and of size 25.6 GiB.

Partition #1 contains a ntfs signature.

Do you want to remove the signature? [Y]es/[N]o: y

The signature will be removed by a write command.

Having made these changes the next step is to check that the settings are correct (taking this opportunity to double check that the Linux boot partition is bootable):

```
Command (m for help): p
Disk /dev/sda: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

Allocating Windows Disk Partitions to RHEL 8

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe3673009
```

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|----------|-----------|----------|-------|----|-----------|
| /dev/sda1 | | 2048 | 53655551 | 53653504 | 25.6G | 83 | Linux |
| /dev/sda3 | * | 53655552 | 55752703 | 2097152 | 1G | 83 | Linux |
| /dev/sda4 | | 55752704 | 104857599 | 49104896 | 23.4G | 5 | Extended |
| /dev/sda5 | | 55754752 | 104857599 | 49102848 | 23.4G | 8e | Linux LVM |

Filesystem/RAID signature on partition 1 will be wiped.

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

In order to make the new partition suitable for use by RHEL 8, it needs to have a file system created on it. The recommended file system type for the current release of RHEL is XFS which will be covered in greater detail in the chapter entitled “*Adding a New Disk Drive to a RHEL 8 System*”. Creation of the file system is performed using the *mkfs.xfs* command as follows:

```
# mkfs.xfs -f /dev/sda1
meta-data=/dev/sda1             isize=512    agcount=4, agsize=1676672 blks
                               =               sectsz=512   attr=2, projid32bit=1
                               =               crc=1        finobt=1, sparse=1, rmapbt=0
                               =               reflink=1
data      =                     bsize=4096   blocks=6706688, imaxpct=25
                               =               sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log       bsize=4096   blocks=3274, 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 RHEL 8 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/sda1` device to match your environment):

```
/dev/sda1 /data xfs defaults 0 0
```

Finally, we can manually mount the new partition (note that on subsequent reboots this will not be necessary as the partition will automount as a result of 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/sda1       26G   215M   26G   1% /data
```

5.5 Editing the Boot Menu

The next step is to modify the RHEL boot menu. Since this was originally a dual boot system, the menu is configured to provide the option of booting either Windows or RHEL. Now that the Windows partition is gone, we need to remove this boot option. Start by editing the `/etc/grub.d/40_custom` file and removing the Windows menu entry:

```
#!/bin/sh
exec tail -n +3 $0

# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.

menuentry "Windows 10" {
    set root=(hd0,1)
    chainloader +1
}
```

Save the file and use the `grub2-mkconfig` tool to generate the `/boot/grub2/grub.cfg` file as follows:

```
# grub2-mkconfig --output=/boot/grub2/grub.cfg
Generating grub configuration file ...
done
```

Next time the system restarts, the Windows 10 option will no longer be provided by the boot menu.

5.6 Summary

The Windows partitions in a dual boot configuration can be removed at any time to free up space for a RHEL 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 RHEL system. The final task is to remove the Windows option from the boot menu configuration. These tasks can be performed either from the command-line using `fdisk`, or from within the desktop environment using the `gnome-disks` utility.

6. A Guided Tour of the GNOME 3 Desktop

Red Hat Enterprise Linux 8 includes the GNOME 3 desktop environment. Although lacking the complexity of Windows and macOS desktops, GNOME 3 provides an uncluttered and intuitive desktop environment that provides all of the essential features of a windowing environment with the added advantage that it can be learned quickly.

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

6.1 Installing the GNOME Desktop

If either the Workstation or Server with GUI software configuration was selected during the RHEL 8 installation process, the GNOME desktop will already be installed and will automatically launch each time the system starts.

If any other software configuration was selected during the RHEL 8 installation process, the GNOME desktop will not have been included in the packages installed on the system. On server-based systems without a display attached, the idea of installing a graphical desktop environment may seem redundant. It is worth noting, however, that remote access to the GNOME desktop is also possible so, even on so called *headless servers* (i.e. servers lacking a monitor, keyboard and mouse) it may still be beneficial to install the GNOME desktop packages. The topic of establishing remote desktop access will be covered in detail starting with the “*RHEL 8 Remote Desktop Access with Vino*” chapter of this book.

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

```
# dnf groupinstall "Workstation"
```

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

```
$ startx
```

6.2 An Overview of the GNOME 3 Desktop

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

A Guided Tour of the GNOME 3 Desktop

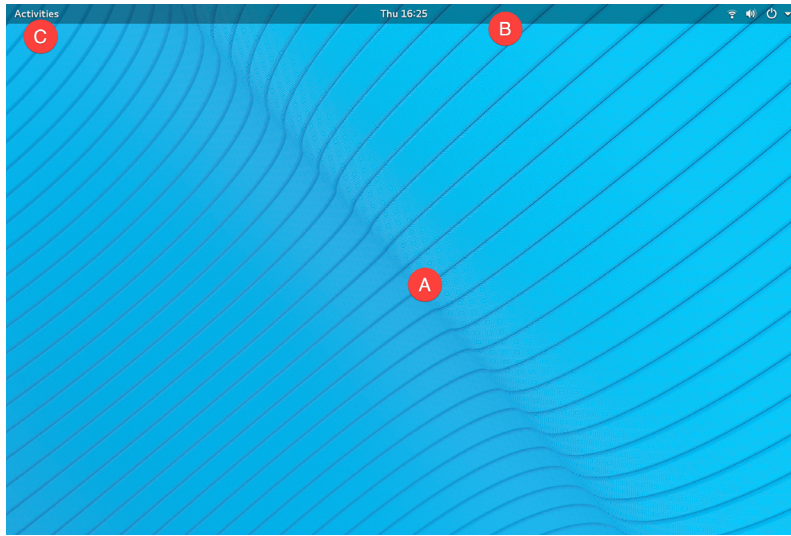


Figure 6-1

The main desktop area (marked A) is where windows will appear when applications and utilities are launched. Unlike other desktop environments, it is not possible to drag and drop files or applications onto the desktop, providing a clean and uncluttered workspace.

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

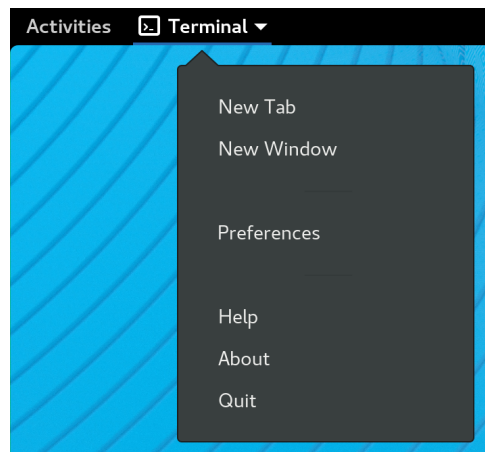


Figure 6-2

6.3 Launching Activities

Applications and utilities are launched using the Activities overview dashboard (referred to as the *dash*) which may be displayed either by clicking on the Activities button in the top bar or pressing the *special key* on the keyboard. On Windows keyboards this is the Windows key, on macOS the Command key and on Chromebooks the key displaying a magnifying glass.

When displayed, the dash will appear as shown in Figure 6-3 below:

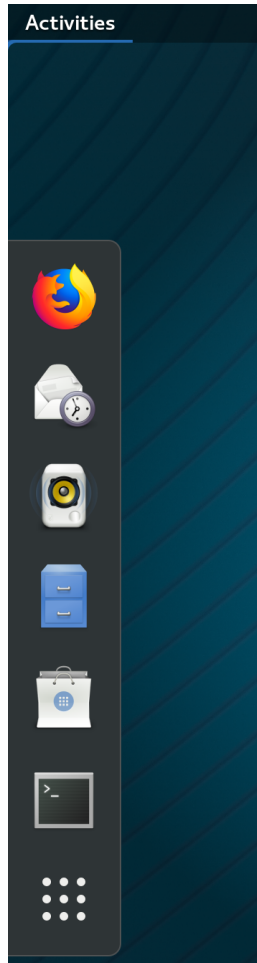


Figure 6-3

By default the dash will display an icon for a predefined set of commonly used applications and will also include an icon for any applications that are currently running. If the application is currently running it will appear with a bar marker beneath the icon.

To launch an application, simply click on the icon in the dash.

To find an application not included on the dash, one option is to select the bottom most icon (the

A Guided Tour of the GNOME 3 Desktop

square comprising nine dots) to display a browsable list of applications as shown in Figure 6-4:



Figure 6-4

Note that the list can be filtered to display all applications or only those used frequently by selecting the buttons at the bottom center of the screen. It is also important to be aware that some entries in the list are actually folders holding additional applications.

An alternative to browsing the applications is to perform a search using the search bar which appears when the dash is displayed as shown in Figure 6-5:



Figure 6-5

As text is typed into the search box, the list of possible matches will be refined.

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



Figure 6-6

To remove an app from the dash, right-click on the icon in the dash and select *Remove from Favorites*.

6.4 Managing Windows

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

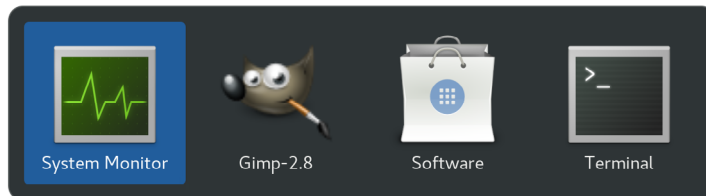


Figure 6-7

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

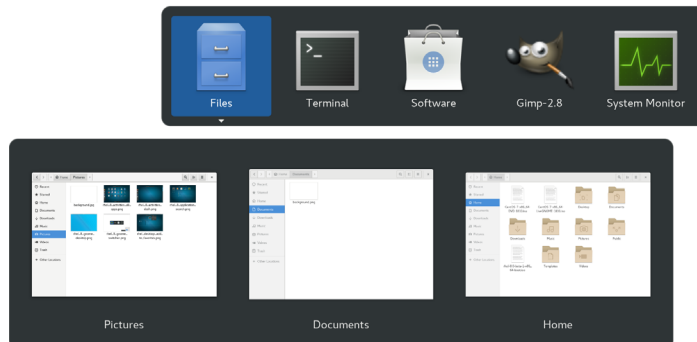


Figure 6-8

A Guided Tour of the GNOME 3 Desktop

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

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

6.5 Using Workspaces

The area of the screen where the application windows appear is referred to as the *workspace* and GNOME 3 allows multiple workspaces to be configured. To create a new workspace, display the Activities overview and move the mouse pointer to the far right of the screen to display the workspaces panel:

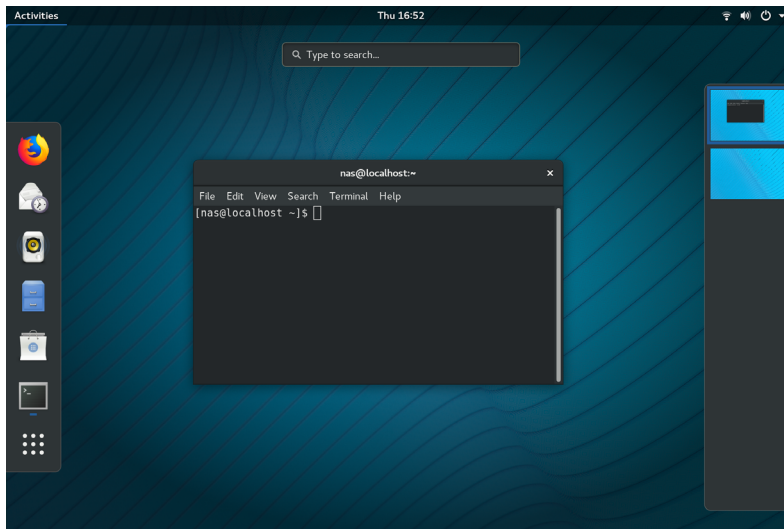


Figure 6-9

To switch to a different panel, simply select it from the list. To move a window from one workspace to another, display the workspaces panel and drag and drop the application window (either the actual window from the current workspace or the thumbnail window in the workspaces panel) onto the destination workspace. When a window is added to a blank workspace, another blank workspace is added to the workspace panel, allowing multiple workspaces to be created.

To remove a workspace either close all the windows on that workspace, or move them to another workspace.

6.6 Calendar and Notifications

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

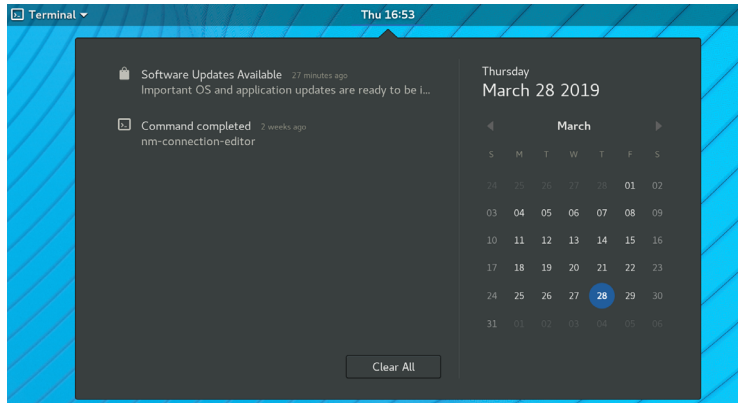


Figure 6-10

6.7 Desktop Settings

To access the Settings application, click on the down arrow on the far right of the top bar and select the button with the tools icon as highlighted in Figure 6-11:

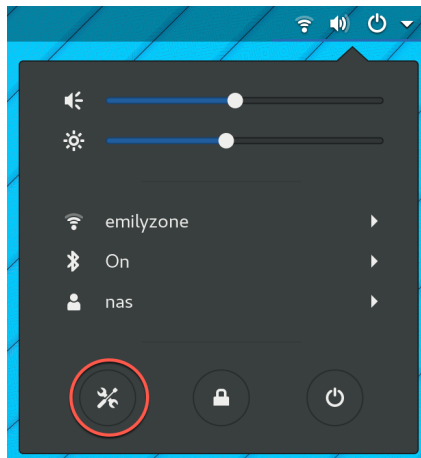


Figure 6-11

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

A Guided Tour of the GNOME 3 Desktop

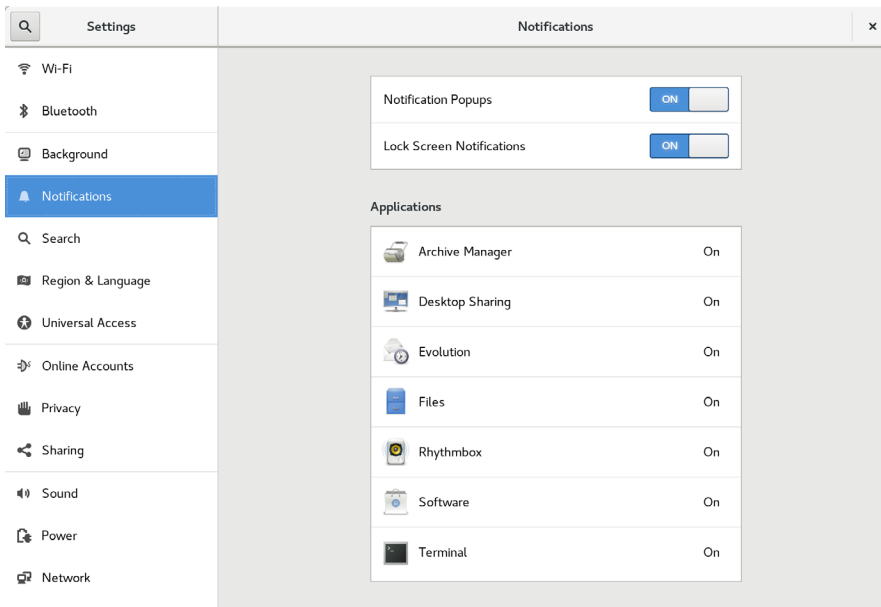


Figure 6-12

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

6.8 Installing RHEL Desktop Software

In common with other operating systems such as macOS, Windows, iOS and Android, RHEL has an “app store” in the form of the Software tool. An icon is usually placed in the dash as indicated in Figure 6-13:



Figure 6-13

Once loaded, the tool provides a list of applications available for installation on the system grouped together based on categories, recommendations and editor's picks. The library of available applications may also be searched to find a specific item:

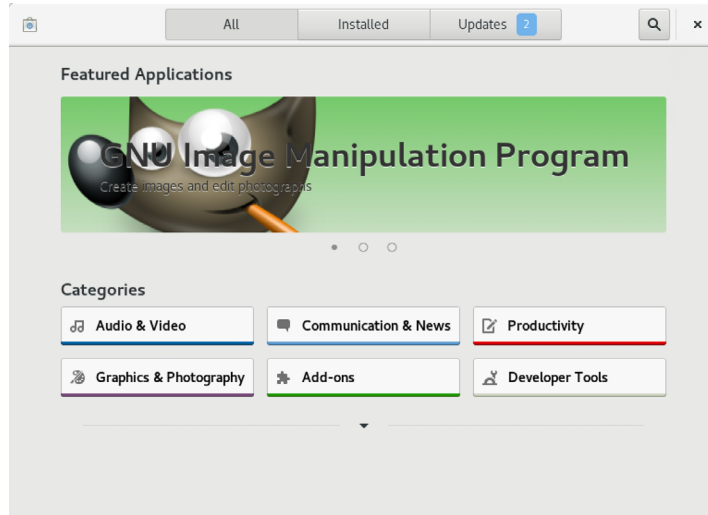


Figure 6-14

To install an application, simply select it in the Software window and click on the *Install* button:

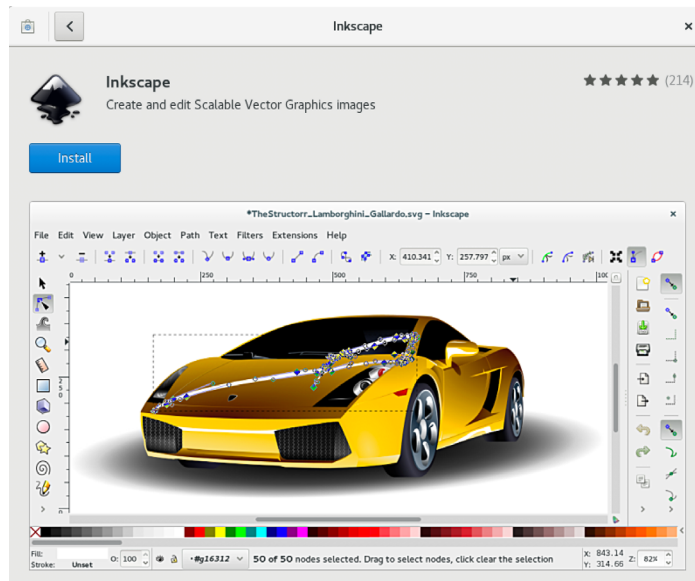


Figure 6-15

6.9 Beyond Basic Customization

The GNOME 3 desktop is, by design, a clean and uncluttered environment with minimal customization options. That does not mean, however, that it is not possible to make additional

A Guided Tour of the GNOME 3 Desktop

changes to the desktop. In fact, the GNOME Project has developed a tool called GNOME Tweaks for this very purpose. Use the following commands to install and run this tool:

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

Once GNOME Tweaks has loaded, the interface shown in Figure 6-16 will appear:

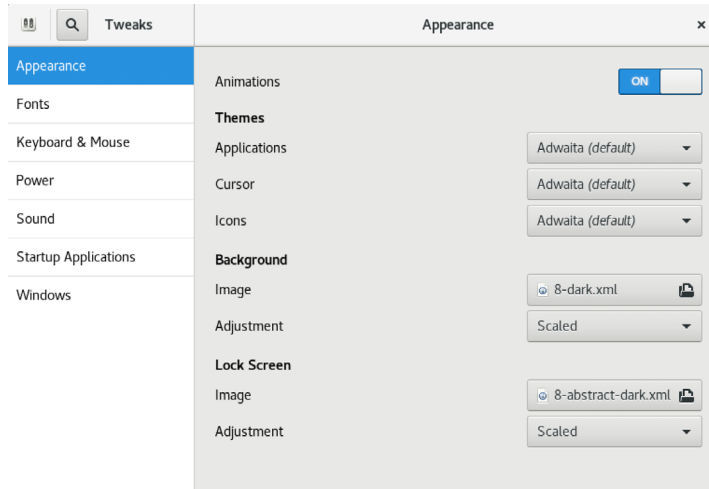


Figure 6-16

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

6.10 Summary

RHEL 8 includes the GNOME 3 desktop environment which may either be included during the initial installation or installed later using the *dnf* group package installation feature. Unlike most other desktop environments, GNOME 3 is intended to provide a clean and easy to use windowing user interface. Key areas of the GNOME 3 desktop include the top bar, Activities overview and dash. In addition, GNOME 3 supports multiple workspaces keeping running applications organized and the screen uncluttered. A variety of configuration options is also available within the Settings app including desktop background settings, audio, network configuration and WiFi network selection. Additional configuration options may be accessed by installing the GNOME Tweaks package.