PEARSON IT CERTIFICATION

Practice Tests

Flash Cards

Labs

Study Planner

Cert Guide

Advance your IT career with hands-on learning

Red Hat RHCSA 8 (EX200)

Second Edition

SANDER VAN VUGT

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Red Hat RHCSA™ 8 Cert Guide

Second Edition

EX200

Sander van Vugt

Red Hat RHCSA™ 8 Cert Guide: EX200, Second Edition

ISBN-13: 978-0-13-734162-7

ISBN-10: 0-13-734162-8

Library of Congress Control Number: 2021930496

ScoutAutomatedPrintCode

Editor-in-Chief
Mark Taub

Acquisitions Editor
Denise Lincoln

Development Editor
Ellie Bru

Managing Editor
Sandra Schroeder

Senior Project Editor
Tonya Simpson

Copy Editors
Bill McManus
Chuck Hutchinson

Indexer
Cheryl Ann Lenser

Proofreader
Donna E. Mulder

Technical Editors
William "Bo" Rothwell
John McDonough

Publishing Coordinator
Cindy Teeters

Cover Designer
Chuti Prasertsith

Compositor
codeMantra

Contents at a Glance

Table of Contents

About the Author

Sander van Vugt is an independent Linux trainer, author, and consultant living in the Netherlands. Sander is the author of the best-selling *Red Hat Certified System Administrator (RHCSA) Complete Video Course* and the *Red Hat Certified Engineer (RHCE) Complete Video Course*. He has also written numerous books about different Linux-related topics and many articles for Linux publications around the world. Sander has been teaching Red Hat, Linux+, and LFCS classes since 1994. As a consultant, he specializes in Linux high-availability solutions and performance optimization. You can find more information about Sander on his website at http://www.sandervanvugt.com.

For more information about RHCSA certification and additional resources, visit the author's Red Hat Certification page at http://www.rhatcert.com/.

Dedication

This book is dedicated to my parents. Dad, Mum, I've been able to get here thanks to you. This book is also dedicated to my in-laws. Papi, Mami, merci d'avoir fait une fille si formidable. Additionally, this book is dedicated to my wife. You, me, forever.

Acknowledgments

This book could not have been written without the help of all the people who contributed to it. I want to thank the people at Pearson, Denise Lincoln and Ellie Bru in particular. We've worked a lot together over the years, and this book is another milestone on our road to success!

About the Technical Reviewers

William "Bo" Rothwell, at the impressionable age of 14, crossed paths with a TRS-80 Micro Computer System (affectionately known as a Trash 80). Soon after, the adults responsible for Bo made the mistake of leaving him alone with the TRS-80. He immediately dismantled it and held his first computer class, showing his friends what made this "computer thing" work.

Since this experience, Bo's passion for understanding how computers work and sharing this knowledge with others has resulted in a rewarding career in IT training. His experience includes Linux, Unix, IT security, DevOps, and programming languages such as Perl, Python, Tcl, and Bash. Bo is the founder and lead instructor of One Course Source, an IT training organization.

Bo is an author of several books, including *Linux for Developers: Jumpstart Your Linux Programming Skills*, *Linux Essentials for Cybersecurity*, and *LPIC-2 Cert Guide*. He can be reached on LinkedIn: https://www.linkedin.com/in/bo-rothwell/.

John McDonough has more than 30 years of development experience. Currently, he is a developer advocate for Cisco DevNet. As a developer advocate, John writes code and creates DevNet Learning Labs about how to write code; writes blogs about writing code; and presents at Cisco Live, SXSW, AnsibleFest, and other industry events. John focuses on the Cisco computing systems products, Cisco UCS, and Cisco Intersight. John's career at Cisco has varied from product engineer to custom application developer, technical marketing engineer, and now a developer advocate.

We Want to Hear from You!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

We welcome your comments. You can email or write to let us know what you did or didn't like about this book—as well as what we can do to make our books better.

Please note that we cannot help you with technical problems related to the topic of this book.

When you write, please be sure to include this book's title and author as well as your name and email address. We will carefully review your comments and share them with the author and editors who worked on the book.

Email: community@informit.com

Introduction

Welcome to the *Red Hat RHCSA 8 Cert Guide*, Second Edition. The Red Hat exams are some of the toughest in the business, and this guide will be an essential tool in helping you prepare to take the Red Hat Certified System Administrator (RHCSA) exam.

As an instructor with more than 20 years of experience teaching Red Hat Enterprise Linux, I have taken the RHCSA exam (and the RHCE exam) numerous times so that I can keep current on the progression of the exam, what is new, and what is different. I share my knowledge with you in this comprehensive Cert Guide so that you get the guidance you need to pass the RHCSA exam.

The RHCSA exam was recently updated with materials about containers on RHEL 8. This book contains everything you need to know to pass the 2020 version of the RHCSA exam. As you will see, the Cert Guide covers every objective in the exam and comprises 28 chapters, more than 80 exercises, 4 practice exams, an extensive glossary, and hours of video training. This Cert Guide is the best resource you can get to prepare for and pass the RHCSA exam.

Goals and Methods

To learn the topics described in this book, it is recommended that you create your own testing environment. You cannot become an RHCSA without practicing a lot. Within the exercises included in every chapter of the book, you will find all the examples you need to understand what is on the exam and thoroughly learn the material needed to pass it. The exercises in the chapters provide step-by-step procedures that you can follow to find working solutions so that you can get real experience before taking the exam.

Each chapter also includes one or more end-of-chapter labs. These labs ask questions that are similar to the questions that you might encounter on the exam. Solutions are not provided for these labs, and that is on purpose, because you need to train yourself to verify your work before you take the exam. On the exam, you also have to be able to verify for yourself whether the solution is working as expected. Please be sure to also go to this book's companion website, which provides additional practice exams, appendixes, and video training—all key components to studying for and passing the exam.

To make working with the assignments in this book as easy as possible, the complete lab environment is Bring Your Own. In Chapter 1 you'll learn how to install CentOS or Red Hat Enterprise Linux 8 in a virtual machine, and that is all that is required to go through the labs. You don't need to import any virtual machines; just install your own virtual machine and you'll be ready to go!

This book contains everything you need to pass the exam, but if you want more guidance and practice, I have a number of video training titles available to help you study, including the following:

- *Red Hat Certified System Administrator (RHCSA) Complete Video Course*, 3rd Edition
- *Upgrading to Red Hat Enterprise Linux (RHEL) 8 LiveLessons*

Apart from these products, you might also appreciate my website, http://rhatcert.com. Through this website, I provide updates on anything that is useful to exam candidates. I recommend that you register on the website so that I can send you messages about important updates that I've made available. Also, you'll find occasional video updates on my YouTube channel, rhatcert. I hope that all these resources provide you with everything you need to pass the Red Hat Certified System Administrator exam in an affordable way. Good luck!

Who Should Read This Book?

This book is written as an RHCSA exam preparation guide. That means that you should read it if you want to increase your chances of passing the RHCSA exam. A secondary use of this book is as a reference guide for Red Hat system administrators. As an administrator, you'll like the explanations and procedures that describe how to get things done on Red Hat Enterprise Linux.

So, why should you consider passing the RHCSA exam? That question is simple to answer: Linux has become a very important operating system, and qualified professionals are in demand all over the world. If you want to work as a Linux professional and prove your skills, the RHCSA certificate really helps and is one of the most sought-after certificates in IT. Having this certificate dramatically increases your chances of becoming hired as a Linux professional.

How This Book Is Organized

This book is organized as a reference guide to help you prepare for the RHCSA exam. If you're new to the topics, you can just read it cover to cover. You can also read the individual chapters that you need to fine-tune your skills in this book. Every chapter starts with a "Do I Know This Already?" quiz that asks questions about ten topics that are covered in that chapter and provides a simple tool to check whether you're already familiar with the topics covered in the chapter.

The book also provides four RHCSA practice exams; these are an essential part of readying yourself for the real exam experience. You may be able to provide the right answer to the multiple-choice chapter questions, but that doesn't mean that you can

create the configurations when you take the exam. The companion files include two extra practice exams, two hours of video from the *Red Hat Certified System Administrator (RHCSA) Complete Video Course*, 3rd Edition, and additional appendixes. The following outline describes the topics that are covered in the chapters:

Part I: Performing Basic System Management Tasks

- **Chapter 1: Installing Red Hat Enterprise Linux:** In this chapter, you learn how to install Red Hat Enterprise Linux Server (RHEL). It also shows how to set up an environment that can be used for working on the labs and exercises in this book.
- **Chapter 2: Using Essential Tools:** This chapter covers some of the Linux basics, including working with the shell and Linux commands. This chapter is particularly important if you're new to working with Linux.
- **Chapter 3: Essential File Management Tools:** In this chapter, you learn how to work with tools to manage the Linux file system. This is an important skill because everything on Linux is very file system oriented.
- **Chapter 4: Working with Text Files:** In this chapter, you learn how to work with text files. The chapter teaches you how to create text files, but also how to look for specific content in the different text files.
- **Chapter 5: Connecting to Red Hat Enterprise Linux 8:** This chapter describes the different methods that can be used to connect to RHEL 8. It explains both local login and remote login and the different terminal types used for this purpose.
- **Chapter 6: User and Group Management:** On Linux, users are entities that can be used by people or processes that need access to specific resources. This chapter explains how to create users and make user management easier by working with groups.
- **Chapter 7: Permissions Management:** In this chapter, you learn how to manage Linux permissions through the basic read, write, and execute permissions, but also through the special permissions and access control lists.
- **Chapter 8: Configuring Networking:** A server is useless if it isn't connected to a network. In this chapter, you learn the essential skills required for managing network connections.

Part II: Operating Running Systems

- **Chapter 9: Managing Software:** Red Hat offers an advanced system for managing software packages. This chapter teaches you how it works.
- **Chapter 10: Managing Processes:** As an administrator, you need to know how to work with the different tasks that can be running on Linux. This

chapter shows how to do this, by sending signals to processes and by changing process priority.

- **Chapter 11: Working with Systemd:** Systemd is the standard manager of services and more in RHEL 8. In this chapter, you learn how to manage services using Systemd.
- **Chapter 12: Scheduling Tasks:** In this chapter, you learn how to schedule a task for execution on a moment that fits you best.
- **Chapter 13: Configuring Logging:** As an administrator, you need to know what's happening on your server. The rsyslogd and journald services are used for this purpose. This chapter explains how to work with them.
- **Chapter 14: Managing Storage:** Storage management is an important skill to master as a Linux administrator. This chapter explains how hard disks can be organized in partitions and how these partitions can be mounted in the file system.
- **Chapter 15: Managing Advanced Storage:** Dividing disks in partitions isn't very flexible. If you need optimal flexibility, you need LVM logical volumes, which are used by default while you're installing Red Hat Enterprise Linux. This chapter shows how to create manage those logical volumes. You'll also learn how to work with the Stratis and VDO storage techniques.

Part III: Performing Advanced System Administration Tasks

- **Chapter 16: Basic Kernel Management:** The kernel is the part of the operating system that takes care of handling hardware. This chapter explains how that works and what an administrator can do to analyze the current configuration and manage hardware devices in case the automated procedure doesn't work well.
- **Chapter 17: Managing and Understanding the Boot Procedure:** Many things are happening when a Linux server boots. This chapter describes the boot procedure in detail and zooms in on vital aspects of the boot procedure, including the GRUB 2 boot loader and the Systemd service manager.
- **Chapter 18: Essential Troubleshooting Skills:** Sometimes a misconfiguration can cause your server to no longer boot properly. This chapter teaches you some of the techniques that can be applied when normal server startup is no longer possible.
- **Chapter 19: An Introduction to Bash Shell Scripting:** Some tasks are complex and need to be performed repeatedly. Such tasks are ideal candidates for optimization through shell scripts. In this chapter, you learn how to use conditional structures in shell scripts to automate tasks efficiently.

Part IV: Managing Network Services

- **Chapter 20: Configuring SSH:** Secure Shell (SSH) is one of the fundamental services that is enabled on RHEL 8 by default. Using SSH allows you to connect to a server remotely. In this chapter, you learn how to set up an SSH server.
- **Chapter 21: Managing Apache HTTP Services:** Apache is the most commonly used service on Linux. This chapter shows how to set up Apache web services, including the configuration of Apache virtual hosts.
- **Chapter 22: Managing SELinux:** Many Linux administrators only know how to switch it off, because SELinux is hard to manage and is often why services cannot be accessed. In this chapter, you learn how SELinux works and what to do to configure it so that your services are still working and will be much better protected against possible abuse.
- **Chapter 23: Configuring a Firewall:** Apart from SELinux, RHEL 8 comes with a firewall as one of the main security measures, which is implemented by the firewalld service. In this chapter, you learn how this service is organized and what you can do to block or enable access to specific services.
- **Chapter 24: Accessing Network Storage:** While you're working in a server environment, managing remote mounts is an important skill. A remote mount allows a client computer to access a file system offered through a remote server. These remote mounts can be made through a persistent mount in /etc/fstab, or by using the automount service. This chapter teaches how to set up either of them and shows how to configure an FTP server.
- **Chapter 25: Configuring Time Services:** For many services, such as databases and Kerberos, it is essential to have the right time. That's why as an administrator you need to be able to manage time on Linux. This chapter teaches you how.
- **Chapter 26: Managing Containers:** Containers have revolutionized datacenter IT. Where services not so long ago were running directly on top of the server operating system, nowadays services are often offered as containers. Red Hat Enterprise Linux 8 includes a complete platform to run containers. In this chapter, you learn how to work with it.
- **Chapter 27: Final Preparation:** In this chapter, you get some final exam preparation tasks. It contains many tips that help you maximize your chances of passing the RHCSA exam.
- **Chapter 28: Theoretical Pre-Assessment Exam:** This chapter provides an RHCSA theoretical pre-assessment exam to help you assess your skills and determine the best route forward for studying for the exam.

Part V: RHCSA RHEL 8 Practice Exams

This part supplies two RHCSA practice exams so that you can test your knowledge and skills further before taking the exams. Two additional exams are on the companion website.

How to Use This Book

To help you customize your study time using this book, the core chapters have several features that help you make the best use of your time:

- **"Do I Know This Already?" Quizzes:** Each chapter begins with a quiz that helps you determine the amount of time you need to spend studying that chapter and the specific topics that you need to focus on.
- **Foundation Topics:** These are the core sections of each chapter. They explain the protocols, concepts, and configuration for the topics in that chapter.
- **Exam Preparation Tasks:** Following the "Foundation Topics" section of each chapter, the "Exam Preparation Tasks" section lists a series of study activities that you should complete. Each chapter includes the activities that make the most sense for studying the topics in that chapter. The activities include the following:
  - **Review All Key Topics:** The Key Topic icon is shown next to the most important items in the "Foundation Topics" section of the chapter. The Review All Key Topics activity lists the key topics from the chapter and their corresponding page numbers. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic.
  - **Complete Tables and Lists from Memory:** To help you exercise your memory and memorize some facts, many of the more important lists and tables from the chapter are included in a document on the companion website. This document offers only partial information, allowing you to complete the table or list.
  - **Define Key Terms:** This section lists the most important terms from the chapter, asking you to write a short definition and compare your answer to the glossary at the end of this book.
- **Review Questions:** These questions at the end of each chapter measure insight into the topics that were discussed in the chapter.
- **End-of-Chapter Labs:** Real labs give you the right impression of what an exam assignment looks like. The end-of-chapter labs are your first step in finding out what the exam tasks really look like.

Other Features

In addition to the features in each of the core chapters, this book, as a whole, has additional study resources on the companion website, including the following:

- **Four practice exams:** *Red Hat RHCSA 8 Cert Guide*, Second Edition, comes with four practice exams. You will find two in the book and two additional exams on the companion website; these are provided as PDFs so you can get extra practice testing your skills before taking the exam in the testing facility.
- **More than two hours of video training:** The companion website contains more than two hours of instruction from the best-selling *Red Hat Certified System Administrator (RHCSA) Complete Video Course*, 3rd Edition.

Exam Objective to Chapter Mapping

Table 1 details where every objective in the RHCSA exam is covered in this book so that you can more easily create a successful plan for passing the exam.

Table 1 Coverage of RHCSA Objectives

Objective	Chapter Title	Chapter
Access a shell prompt and issue commands with correct syntax	Using Essential Tools	2
Use input-output redirection (>, >>, |, 2>, etc.)	Using Essential Tools	2
Use grep and regular expressions to analyze text	Working with Text Files	4
Access remote systems using SSH	Connecting to Red Hat Enterprise Linux 8	5
Log in and switch users in multiuser targets	Connecting to Red Hat Enterprise Linux 8	5
Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2	Essential File Management Tools	3
Create and edit text files	Working with Text Files	4
Create, delete, copy, and move files and directories	Essential File Management Tools	3
Create hard and soft links	Essential File Management Tools	3

Objective	Chapter Title	Chapter
List, set, and change standard ugo/rwx permissions	Permissions Management	7
Locate, read, and use system documentation including man, info, and files in /usr/share/doc	Using Essential Tools	2
Boot, reboot, and shut down a system normally	Connecting to Red Hat Enterprise Linux 8	5
Boot systems into different targets manually	Essential Troubleshooting Skills	18
Interrupt the boot process in order to gain access to a system	Essential Troubleshooting Skills	18
Identify CPU/memory-intensive processes and kill processes	Managing Processes	10
Adjust process scheduling	Managing Processes	10
Locate and interpret system log files and journals	Configuring Logging	13
Preserve system journals	Configuring Logging	13
Start, stop, and check the status of network services	Configuring Networking	8
Securely transfer files between systems	Connecting to Red Hat Enterprise Linux 8	5
List, create, and delete partitions on MBR and GPT disks	Managing Storage	14
Create and remove physical volumes	Managing Advanced Storage	15
Assign physical volumes to volume groups	Managing Advanced Storage	15
Create and delete logical volumes	Managing Advanced Storage	15
Configure systems to mount file systems at boot by universally unique ID (UUID) or label	Managing Storage	14
Add new partitions and logical volumes, and swap to a system nondestructively	Managing Storage	14
Create, mount, unmount, and use vfat, ext4, and xfs file systems	Managing Storage	14

Objective	Chapter Title	Chapter
Mount and unmount network file systems using NFS	Accessing Network Storage	24
Extend existing logical volumes	Managing Advanced Storage	15
Create and configure set-GID directories for collaboration	Permissions Management	7
Configure disk compression	Managing Advanced Storage	15
Manage layered storage	Managing Advanced Storage	15
Diagnose and correct file permission problems	Permissions Management	7
Schedule tasks using at and cron	Scheduling Tasks	12
Start and stop services and configure services to start automatically at boot	Working with Systemd	11
Configure systems to boot into a specific target automatically	Managing and Understanding the Boot Procedure	17
Configure time service clients	Configuring Time Services	25
Install and update software packages from Red Hat Network, a remote repository, or from the local file system	Managing Software	9
Work with package module streams	Managing Software	9
Modify the system bootloader	Managing and Understanding the Boot Procedure	17
Configure IPv4 and IPv6 addresses	Configuring Networking	8
Configure hostname resolution	Configuring Networking	8
Configure network services to start automatically at boot	Configuring Networking	8
Restrict network access using firewall-cmd/firewall	Configuring a Firewall	23
Create, delete, and modify local user accounts	User and Group Management	6
Change passwords and adjust password aging for local user accounts	User and Group Management	6
Create, delete, and modify local groups and group memberships	User and Group Management	6
Configure superuser access	User and Group Management	6

Objective	Chapter Title	Chapter
Manage security		
Configure firewall settings using firewall-cmd/firewalld	Configuring a Firewall	23
Configure key-based authentication for SSH	Configuring SSH	20
Set enforcing and permissive modes for SELinux	Managing SELinux	22
List and identify SELinux file and process context	Managing SELinux	22
Restore default file contexts	Managing SELinux	22
Use Boolean settings to modify system SELinux settings	Managing SELinux	22
Diagnose and address routine SELinux policy violations	Managing SELinux	22
Manage containers	Managing Containers	26
Create simple shell scripts	An Introduction to Bash Shell Scripting	19

Where Are the Companion Content Files?

Register this print version of *Red Hat RHCSA 8 Cert Guide*, Second Edition, to access the bonus content online.

This print version of this title comes with a website of companion content. You have online access to these files by following these steps:

1. Go to www.pearsonITcertification.com/register and log in or create a new account.
2. Enter the ISBN: **9780137341627**.
3. Answer the challenge question as proof of purchase.
4. Click the **Access Bonus Content** link in the Registered Products section of your account page to be taken to the page where your downloadable content is available.

Please note that many of the companion content files can be very large, especially image and video files.

If you are unable to locate the files for this title by following the steps, please visit www.pearsonITcertification.com/contact and select the Site Problems/Comments option. A customer service representative will assist you.

This book also includes an exclusive offer for 70% off the *Red Hat Certified System Administrator (RHCSA) Complete Video Course*, 3rd Edition.

Figure Credits

Chapter opener images by Charlie Edwards/ Photodisc/Getty Images

Figures 1-1 through 1-15 © 2019 Red Hat, Inc

Figures 2-1 and 2-2 © 2019 Red Hat, Inc

Figures 5-1 and 5-2 © 2019 Red Hat, Inc

Figures 8-1 through 8-5 © 2019 Red Hat, Inc

Figure 10-1 © 2019 Red Hat, Inc

Figures 18-1 through 18-5 © 2019 Red Hat, Inc

This page intentionally left blank

Managing and Understanding the Boot Procedure

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the "Exam Preparation Tasks" section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 17-1 lists the major headings in this chapter and their corresponding "Do I Know This Already?" quiz questions. You can find the answers in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and 'Review Questions.'"

Table 17-1 "Do I Know This Already?" Section-to-Question Mapping

Foundation Topics Section	Questions
Working with Systemd	1–7
Working with GRUB 2	8–10

1. Which of the following is the most efficient way to define a system want?
   a. Use the **systemctl enable** command.
   b. Define the want in the unit file [Service] section.
   c. Create a symbolic link in the /usr/lib/system/system directory.
   d. Create a symbolic link in the unit wants directory in the /etc/system/system directory.

2. Which target is considered the normal target for servers to start in?
   a. graphical.target
   b. server.target
   c. multi-user.target
   d. default.target

3. Which of the following is *not* an example of a system target?
   a. rescue.target
   b. restart.target
   c. multi-user.target
   d. graphical.target

4. Where do you define which target a unit should be started in if it is enabled?
   a. The target unit file
   b. The wants directory
   c. The systemctl.conf file
   d. The [Install] section in the unit file

5. To allow targets to be isolated, you need a specific statement in the target unit file. Which of the following describes that statement?
   a. **AllowIsolate**
   b. **Isolate**
   c. **SetIsolate**
   d. **Isolated**

6. An administrator wants to change the current multi-user.target to the rescue.target. Which of the following should she do?
   a. Use **systemctl isolate rescue.target**
   b. Use **systemctl start rescue.target**
   c. Restart the system, and from the GRUB boot prompt specify that rescue.target should be started
   d. Use **systemctl enable rescue.target --now**

7. To which System V runlevel does multi-user.target correspond?
   a. 2
   b. 3
   c. 4
   d. 5

8. What is the name of the file where you should apply changes to the GRUB 2 configuration?
   a. /boot/grub/menu.lst
   b. /boot/grub2/grub.cfg
   c. /etc/sysconfig/grub
   d. /etc/default/grub

9. After applying changes to the GRUB 2 configuration, you need to write those changes. Which of the following commands will do that for you?
   a. **grub2 -o /boot/grub/grub.cfg**
   b. **grub2-mkconfig > /boot/grub2/grub.cfg**
   c. **grub2 > /boot/grub2/grub.cfg**
   d. **grub2-install > /boot/grub2/grub.cfg**

10. What is the name of the GRUB2 configuration file that is generated on a UEFI system?
    a. /boot/efi/redhat/grub.cfg
    b. /boot/efi/EFI/redhat/grub.cfg
    c. /boot/EFI/grub.cfg
    d. /boot/EFI/efi/grub.cfg

Foundation Topics

Managing Systemd Targets

Systemd is the service in Red Hat Enterprise Linux 8 that is responsible for starting all kinds of things. Systemd goes way beyond starting services; other items are started from Systemd as well. In Chapter 11, "Working with Systemd," you learned about the Systemd fundamentals; this chapter looks at how Systemd targets are used to boot your system into a specific state.

Understanding Systemd Targets

A Systemd *target* is basically just a group of units that belong together. Some targets are just that and nothing else, whereas other targets can be used to define the state a system is booting in, because these targets have one specific property that regular targets don't have: they can be isolated. Isolatable targets contain everything a system needs to boot or change its current state. Four targets can be used while booting:

- **emergency.target:** In this target only a minimal number of units are started, just enough to fix your system if something is seriously wrong. You'll find that it is quite minimal, as some important units are not started.
- **rescue.target:** This target starts all units that are required to get a fully operational Linux system. It doesn't start nonessential services though.
- **multi-user.target:** This target is often used as the default target a system starts in. It starts everything that is needed for full system functionality and is commonly used on servers.
- **graphical.target:** This target also is commonly used. It starts all units that are needed for full functionality, as well as a graphical interface.

Working with Targets

Working with targets may seem complicated, but it is not. It drills down to three common tasks:

- Adding units to be automatically started
- Setting a default target
- Running a nondefault target to enter troubleshooting mode

In Chapter 11 you learned how to use the **systemctl enable** and **systemctl disable** commands to add services to or remove services from targets. In this chapter you learn how to set a default target and how to run a nondefault target to enter troubleshooting mode. But first let's take a closer look at the working of targets under the hood.

Key Topic

Understanding Target Units

Behind a target there is some configuration. This configuration consists of two parts:

- The target unit file
- The "wants" directory, which contains references to all unit files that need to be loaded when entering a specific target

Targets by themselves can have dependencies to other targets, which are defined in the target unit file. Example 17-1 shows the definition of the multi-user.target file, which defines the normal operational state of a RHEL server.

Example 17-1 The multi-user.target File

```
[root@localhost ~]# systemctl cat multi-user.target
# /usr/lib/systemd/system/multi-user.target
#  SPDX-License-Identifier: LGPL-2.1+
#
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as
     published by
#  the Free Software Foundation; either version 2.1 of the License,
     or
#  (at your option) any later version.

[Unit]
Description=Multi-User System
Documentation=man:systemd.special(7)
Requires=basic.target
Conflicts=rescue.service rescue.target
After=basic.target rescue.service rescue.target
AllowIsolate=yes
```

You can see that by itself the target unit does not contain much. It just defines what it requires and which services and targets it cannot coexist with. It also defines load ordering, by using the **After** statement in the [Unit] section. The target file does not contain any information about the units that should be included; that is in the individual unit files and the wants (explained in the upcoming section "Understanding Wants").

Systemd targets look a bit like runlevels used in previous versions of RHEL, but targets are more than that. A target is a group of units, and there are multiple different targets. Some targets, such as the multi-user.target and the graphical.target, define a specific state that the system needs to enter. Other targets just bundle a group of units together, such as the nfs.target and the printer.target. These targets are included from other targets, such as multi-user.target or graphical.target.

Understanding Wants

Understanding the concept of a want simply requires understanding the verb *want* in the English language, as in "I want a cookie." Wants in Systemd define which units Systemd wants when starting a specific target. Wants are created when Systemd units are enabled using **systemctl enable**, and this happens by creating a symbolic link in the /etc/systemd/system directory. In this directory, you'll find a subdirectory for every target, containing wants as symbolic links to specific services that are to be started.

Managing Systemd Targets

As an administrator, you need to make sure that the required services are started when your server boots. To do this, use the **systemctl enable** and **systemctl disable** commands. You do not have to think about the specific target a service has to be started in. Through the [Install] section in the service unit file, the services know for themselves in which targets they need to be started, and a want is created automatically in that target when the service is enabled. The following procedure walks you through the steps of enabling a service:

1. Type **yum install -y vsftpd**, followed by **systemctl status vsftpd**. If the service has not yet been enabled, the Loaded line will show that it currently is disabled:

```
[root@server202 ~]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
    Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled)
    Active: inactive (dead)
```

2. Type **ls /etc/systemd/system/multi-user.target.wants**. You'll see symbolic links that are taking care of starting the different services on your machine. You can also see that the vsftpd.service link does not exist.

3. Type **systemctl enable vsftpd**. The command shows you that it is creating a symbolic link for the file /usr/lib/systemd/system/vsftpd.service to the directory /etc/systemd/system/multi-user.target.wants. So basically, when you enable a Systemd unit file, in the background a symbolic link is created.

TIP On the RHCSA exam, you are likely to configure a couple of services. It is a good idea to read through the exam questions, identify the services that need to be enabled, and enable them all at once to make sure that they are started automatically when you restart. This prevents your being so focused on configuring the service that you completely forget to enable it as well.

Isolating Targets

As already discussed, on Systemd machines there are several targets. You also know that a target is a collection of units. Some of those targets have a special role because they can be isolated. These are also the targets that you can set as the targets to get into after system start.

By isolating a target, you start that target with all of its dependencies. Only targets that have the **isolate** option enabled can be isolated. We'll explore the **systemctl isolate** command later in this section. Before doing that, let's take a look at the default targets on your computer.

To get a list of all targets currently loaded, type **systemctl --type=target**. You'll see a list of all the targets currently active. If your server is running a graphical environment, this will include all the dependencies required to install the graphical.target also. However, this list shows only the active targets, not all the targets. Type **systemctl --type=target --all** for an overview of all targets that exist on your computer. You'll now see inactive targets also (see Example 17-2).

Example 17-2 Showing System Targets

```
root@localhost ~]# systemctl --type=target --all
  UNIT                    LOAD      ACTIVE   SUB    DESCRIPTION
  basic.target            loaded    active   active Basic System
  bluetooth.target        loaded    active   active Bluetooth
  cryptsetup.target       loaded    active   active Local Encrypted
                                                    Volumes
  dbus.target             not-found inactive dead   dbus.target
  emergency.target        loaded    inactive dead   Emergency Mode
  getty-pre.target        loaded    active   active Login Prompts
                                                      (Pre)
  getty.target            loaded    active   active Login Prompts
```

```
graphical.target           loaded    active   active Graphical
                                                    Interface
 initrd-fs.target          loaded    inactive dead   Initrd File
                                                    Systems
 initrd-root-device.target loaded    inactive dead   Initrd Root
                                                    Device
 initrd-root-fs.target     loaded    inactive dead   Initrd Root File
                                                    System
 initrd-switch-root.target loaded    inactive dead    Switch Root
 initrd.target             loaded    inactive dead    Initrd Default
                                                    Target
 local-fs-pre.target       loaded    active   active Local File
                                                    Systems (Pre)
 local-fs.target           loaded    active   active Local File
                                                    Systems
 multi-user.target         loaded    active   active Multi-User
                                                    System
 network-online.target     loaded    active   active Network is
                                                    Online
 network-pre.target        loaded    active   active Network (Pre)
 network.target            loaded    active   active Network
 nfs-client.target         loaded    active   active NFS client
                                                    services
 nss-lookup.target         loaded    inactive dead     Host and Network
                                                    Name Lookups
 nss-user-lookup.target    loaded    active   active User and Group
                                                    Name Lookups
 paths.target              loaded    active   active Paths
 remote-fs-pre.target      loaded    active   active Remote File
                                                    Systems (Pre)
 remote-fs.target          loaded    active   active Remote File
                                                    Systems
 rescue.target             loaded    inactive dead   Rescue Mode
 rpc_pipefs.target         loaded    active   active rpc_pipefs.
                                                    target
 rpcbind.target            loaded    active   active RPC Port Mapper
 shutdown.target           loaded    inactive dead   Shutdown
 slices.target             loaded    active   active Slices
 sockets.target            loaded    active   active Sockets
 sound.target              loaded    active   active Sound Card
 sshd-keygen.target        loaded    active   active sshd-keygen.
                                                    target
 swap.target               loaded    active   active Swap
 sysinit.target            loaded    active   active System
                                                    Initialization
```

Of the targets on your system, a few have an important role because they can be started (isolated) to determine the state your server starts in. These are also the targets that can be set as the default targets. These targets also roughly correspond to runlevels used on earlier versions of RHEL. These are the following targets:

poweroff.target	runlevel 0
rescue.target	runlevel 1
multi-user.target	runlevel 3
graphical.target	runlevel 5
reboot.target	runlevel 6

If you look at the contents of each of these targets, you'll also see that they contain the AllowIsolate=yes line. That means that you can switch the current state of your computer to either one of these targets using the **systemctl isolate** command. Exercise 17-1 shows you how to do this.

Key Topic

Exercise 17-1 Isolating Targets

1. From a root shell, go to the directory /usr/lib/systemd/system. Type **grep Isolate *.target**. This command shows a list of all targets that allow isolation.
2. Type **systemctl isolate rescue.target**. This command switches your computer to rescue.target. You need to type the root password on the console of your server to log in.
3. Type **systemctl isolate reboot.target**. This command restarts your computer.

Setting the Default Target

Setting the default target is an easy procedure that can be accomplished from the command line. Type **systemctl get-default** to see the current default target and use **systemctl set-default** to set the desired default target.

To set the graphical.target as the default target, you need to make sure that the required packages are installed. If this is not the case, you can use the **yum group list** command to show a list of all RPM package groups. The "server with gui" and "GNOME Desktop" package groups both apply. Use **yum group install "server with gui"** to install all GUI packages on a server where they have not been installed yet.

Working with GRUB 2

The GRUB 2 boot loader is one of the first things that needs to be working well to boot a Linux server. As an administrator, you will sometimes need to apply modifications to the GRUB 2 boot loader configuration. This section explains how to do so. The RHEL 8 boot procedure is discussed in more detail in Chapter 18, where troubleshooting topics are covered as well.

Understanding GRUB 2

The GRUB 2 boot loader makes sure that you can boot Linux. GRUB 2 is installed in the boot sector of your server's hard drive and is configured to load a Linux kernel and the initramfs:

Key Topic

- The kernel is the heart of the operating system, allowing users to interact with the hardware that is installed in the server.
- The initramfs contains drivers that are needed to start your server. It contains a mini file system that is mounted during boot. In it are kernel modules that are needed during the rest of the boot process (for example, the LVM modules and SCSI modules for accessing disks that are not supported by default).

Normally, GRUB 2 works just fine and does not need much maintenance. In some cases, though, you might have to change its configuration. To apply changes to the GRUB 2 configuration, the starting point is the /etc/default/grub file, which has options that tell GRUB what to do and how to do it. Example 17-3 shows the contents of this file after an installation with default settings of RHEL 8.

Key Topic

Example 17-3 Contents of the /etc/default/grub File

```
[root@localhost ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/rhel-swap
  rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

As you can see, the /etc/default/grub file does not contain much information. The most important part that it configures is the GRUB_CMDLINE_LINUX option. This line contains boot arguments for the kernel on your server.

TIP For the RHCSA exam, make sure that you understand the contents of the /etc/default/grub file. That is the most important part of the GRUB 2 configuration anyway.

Apart from the configuration in /etc/default/grub, there are a few configuration files in /etc/grub.d. In these files, you'll find rather complicated shell code that tells GRUB what to load and how to load it. You typically do not have to modify these files. You also do not need to modify anything if you want the capability to select from different kernels while booting. GRUB 2 picks up new kernels automatically and adds them to the boot menu automatically, so nothing has to be added manually.

Understanding GRUB 2 Configuration Files

Based on the configuration files mentioned previously, the main configuration file is created. If your system is a BIOS system, the name of the file is /boot/grub2/grub.cfg. On a UEFI system the file is written to /boot/efi/EFI/redhat on RHEL and /boot/efi/EFI/centos on CentOS. After making modifications to the GRUB 2 configuration, you'll need to regenerate the relevant configuration file, which is why you should know the name of the file that applies to your system architecture. Do *not* edit it, as this file is automatically generated.

Modifying Default GRUB 2 Boot Options

To apply modifications to the GRUB 2 boot loader, the file /etc/default/grub is your entry point. The most important line in this file is GRUB_CMDLINE_LINUX, which defines how the Linux kernel should be started. In this line, you can apply permanent fixes to the GRUB 2 configuration. Some likely candidates for removal are the options **rhgb** and **quiet**. These options tell the kernel to hide all output while booting. That is nice to hide confusing messages for end users, but if you are a server administrator, you probably just want to remove these options.

TIP On the exam, you want to know immediately if something does not work out well. To accomplish this, it is a good idea to remove the **rhgb** and **quiet** boot options. Without these you will not have to guess why your server takes a long time after a restart; you'll just be able to see.

Another interesting parameter is GRUB_TIMEOUT. This defines the amount of time your server waits for you to access the GRUB 2 boot menu before it continues booting automatically. If your server runs on physical hardware that takes a long time to get through the BIOS checks, it may be interesting to increase this time a bit.

While working with GRUB 2, you need to know a bit about kernel boot arguments. There are many of them, and most of them you'll never use, but it is good to know where you can find them. Type **man 7 bootparam** for a man page that contains an excellent description of all boot parameters that you may use while starting the kernel.

To write the modified configuration to the appropriate files, you use the **grub2-mkconfig** command and redirect its output to the appropriate configuration file. On a BIOS system, the command would be **grub2-mkconfig -o /boot/grub2/grub.cfg**, and on a UEFI system the command would be **grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg**.

In Exercise 17-2, you learn how to apply modifications to the GRUB 2 configuration and write them to the /boot/grub2/grub.cfg configuration file.

TIP You should know how to apply changes to the GRUB configuration, but you should also know that the default GRUB 2 configuration works fine as it is for almost all computers. So, you will probably never have to apply any changes at all!

Key Topic

Exercise 17-2 Applying Modifications to GRUB 2

1. Open the file /etc/default/grub with an editor and remove the **rhgb** and **quiet** options from the GRUB_CMDLINE_LINUX line.
2. From the same file, set the GRUB_TIMEOUT parameter to 10 seconds. Save changes to the file and close the editor.
3. From the command line, type **grub2-mkconfig > /boot/grub2/grub.cfg** to write the changes to GRUB 2. (Note that instead of using the redirector > to write changes to the grub.cfg file, you could use the **-o** option. Both methods have the same result.)
4. Reboot and verify that while booting you see boot messages scrolling by.

Summary

In this chapter you learned how Systemd and GRUB 2 are used to bring your server into the exact state you desire at the end of the boot procedure. You also learned how Systemd is organized, and how units can be configured for automatic start with the use of targets. Finally, you read how to apply changes to the default GRUB 2 boot loader. In the next chapter, you learn how to troubleshoot the boot procedure and fix some common problems.

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have several choices for exam preparation: the end-of-chapter labs; the memory tables in Appendix B; Chapter 27, "Final Preparation"; and the practice exams.

Review All Key Topics

Review the most important topics in the chapter, noted with the Key Topic icon in the outer margin of the page. Table 17-2 lists a reference of these key topics and the page number on which each is found.

Key Topic

Table 17-2 Key Topics for Chapter 17

Key Topic Element	Description	Page
Section	Understanding target units	389
Section	Managing Systemd targets	390
Exercise 17-1	Isolating targets	393
List	Explanation of the role of kernel and initramfs	394
Example 17-3	Contents of the /etc/default/grub file	394
Exercise 17-2	Applying modifications to GRUB 2	396

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

unit, wants, target, Systemd, dependencies, initramfs, kernel, boot loader, GRUB

Review Questions

The questions that follow are meant to help you test your knowledge of concepts and terminology and the breadth of your knowledge. You can find the answers to these questions in Appendix A.

1. What is a unit?
2. Which command enables you to make sure that a target is no longer eligible for automatic start on system boot?

3. Which configuration file should you modify to apply common changes to GRUB 2?
4. Which command should you use to show all service units that are currently loaded?
5. How do you create a want for a service?
6. How do you switch the current operational target to the rescue.target?
7. Why can it happen that you get the message that a target cannot be isolated?
8. You want to shut down a Systemd service, but before doing that you want to know which other units have dependencies to this service. Which command would you use?
9. What is the name of the GRUB 2 configuration file where you apply changes to GRUB 2?
10. After applying changes to the GRUB 2 configuration, which command should you run?

End-of-Chapter Labs

You have now learned how to work with Systemd targets and the GRUB 2 boot loader. Before you continue, it is a good idea to work on some labs that help you ensure that you can apply the skills that you acquired in this chapter.

Lab 17.1

1. Set the default target to multi-user.target.
2. Reboot to verify this is working as expected.

Lab 17.2

1. Change your GRUB 2 boot configuration so that you will see boot messages upon startup.

This page intentionally left blank

Index

SYMBOLS

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z