SANDER VAN VUGT

Cert Guide

Learn, prepare, and practice for exam success

Red Hat RHCE®/RHCSA™ 7

Red Hat Enterprise Linux 7 (EX200 and EX300)

PEARSON IT CERTIFICATION

Red Hat® RHCSA™/RHCE® 7 Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300)

Sander van Vugt

PEARSON

800 East 96th Street
Indianapolis, Indiana 46240 USA

Red Hat RHCSA/RHCE 7 Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300)

Sander van Vugt

Published by:

Pearson IT Certification
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

First Printing August 2015

Library of Congress Control Number: 2015942167

ISBN-13: 978-0-7897-5405-9
ISBN-10: 0-7897-5405-3

Warning and Disclaimer

This book is designed to provide information about Red Hat RHCSA and Red Hat RHCE certification. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Pearson IT Certification cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the companion website, DVD, or programs accompanying it.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact international@pearsoned.com.

Publisher
Paul Boger

Associate Publisher
David Dusthimer

Acquisitions Editor
Denise Lincoln

Production Manager
Sandra Schroeder

Development Editor
Ellie Bru

Project Editor
Mandie Frank

Copy Editor
Keith Cline

Technical Editors
Frederik Vos
Ross Brunson
William "Bo" Rothwell

Publishing Coordinator
Vanessa Evans

Designer
Alan Clements

Composition
Nonie Ratcliff

Senior Indexer
Cheryl Lenser

Proofreader
The Wordsmithery LLC

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 also of the R*ed 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, SUSE, and LPI Linux 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 RHAT certification and additional resources, visit the author's Red Hat Certification page at http://www.rhatcert.com/.

About the Technical Reviewers

Frederik Vos is a senior technical trainer in Linux training and also in several virtualization solutions, such as VMware vSphere, XenServer, and KVM. For the past 8 years Frederik has working for XTG in Gouda, a training center specializing in virtualization solutions and Linux, as a teacher and Linux evangelist, responsible for the Linux portfolio at XTG.

He has specialized in data center infrastructures (several Linux distributions), hypervisors, networking, and storage solutions and cloud computing (Cloudstack, Cloudplatform, and OpenStack). He has a lot of knowledge as a teacher and also real-world experience as a system engineer and as a long-time (1997) Linux user.

Ross Brunson has more than 20 years of experience as a Linux and open source trainer, training manager, and technologist and is author of the popular LPIC-1 Exam Cram (Que Publishing), as well as the Pearson LPIC-1 Certification Guide.

Ross recently spent almost 5 years as the director of member services for the Linux Professional Institute, building the member program, conducting dozens of Train-the-Trainer sessions, and providing support for the worldwide Master Affiliate network spanning 100+ countries.

Ross holds a number of key IT certifications and is also author of several successful technical books, dozens of technical courses for major organizations (including the first LPI Certification Bootcamp). He is also skilled at both contributing to and building community around IT products.

He lives in Paradise Valley, Montana, with his family and enjoys traveling far and wide, winter sports, and photography.

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 TSR-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, and programming languages such as Perl, Python, Tcl, and Bash. Bo owns several IT certifications, including earning his original RHCE in 2003. He is a former RHCI (Red Hat Certified Instructor) and the founder and president of One Course Source, an IT training organization.

Dedication

This book is dedicated to my family: Florence, Franck, and Alex. Together we've made great accomplishments over the past year.

Acknowledgments

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

Next I want to thank my technical proofreaders. What has made this book special is that the first round of technical proofreading was completely done by volunteers. We started with 10 volunteers, but just 1 made it all the way to the end. Many thanks to Giles, the man behind the great and very useful website certdepot.net, the only one who reviewed all the chapters.

I also want to thank Jaques Weewer, Rob Mokkink, and all the other volunteer technical reviewers. You made many suggestions without which the book would not have the high quality that it has right now.

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[md]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: feedback@pearsonitcertification.com

Mail: Pearson IT Certification
ATTN: Reader Feedback
800 East 96th Street
Indianapolis, IN 46240 USA

Reader Services

Visit our website and register this book at www.pearsonitcertification.com/register for convenient access to any updates, downloads, or errata that might be available for this book.

Contents at a Glance

Contents

Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).

Introduction

Welcome to the Red Hat RHCSA/RHCE 7 Cert Guide. 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 both the Red Hat Certified System Administrator (RHCSA) and the Red Hat Systems Engineer (RHCE) exams.

As a Red Hat instructor with more than 20 years of experience, I have taken both the RHCSA and RHCE exams 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 both exams.

The RHCSA and RHCE exams were recently updated for Red Hat Enterprise Linux 7. This book contains all you need to know to pass these exams. As you will see, this Cert Guide covers every objective in both exams: 42 chapters, more than 100 exercises, 4 practice exams (2 RHCSA and 2 RHCE), interactive quizzes and exercises, 4 advanced command-line interface (CLI) simulations, 3 virtual machines, and hours of video training. This Red Hat RHCSA/RHCE 7 Cert Guide is the best resource you can get to prepare for and pass the exams.

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 or RHCE 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 procedure descriptions that you can work through to find working solutions so that you can get real experience before taking the tests.

Each chapter also includes an end-of-chapter lab. These labs ask questions that are very similar to the questions that you might encounter on the exam. There are no solutions for these labs provided, and that is on purpose, because you need to train yourself to verify your work before you take the exams. On the exam, you also have to be able to verify for yourself whether the solution is working as expected.

Before you begin, you can test your knowledge by taking the theoretical pre-assessment exams in Chapter 42. These exams help you determine what you know and what you do not know so that you can better prepare your study plan. When you feel ready to take the exams, take a look at the practice exams that come with this book, two RHCSA and two RHCE. These will help you test your skills and get comfortable with the exam content and how questions might be presented in the testing facility.

This book contains everything you need to pass the exams, 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
- Red Hat Certified Engineer (RHCE) Complete Video Course
- Red Hat Certified System Administrator (RHCSA) Exam Prep Video Workshop
- Red Hat Certified Engineer (RHCE) Exam Prep Video Workshop

Apart from these products, you might also appreciate my website 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 rhatcertification.com. I hope that all these resources provide you with everything you need to pass the Red Hat exams in an affordable way! Good luck!

Who Should Read This Book?

This book is written as an RHCSA/RHCE exam preparation guide. That means that you should read it if you want to increase your chances of passing either the RHCSA or RHCE 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/RHCE exams? That question is simple to answer. Linux has become a very important operating system, and qualified professionals are sought after all over the world. If you want to work as a Linux professional, and prove your skills, the RHCSA or RHCE certificate really helps. Having these certificates 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 exams. If you're new to the topics, you can just read it cover to cover. It is a smart idea, though, to distinguish between the RHCSA part and the RHCE part of this book. Finish RHCSA before starting with RHCE, because it will be too much to learn all of it at once.

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. This quiz asks questions about 10 topics that are covered in each chapter and provides a simple tool to check whether you're already familiar with the topics covered in a chapter. Remember, though, the RHCSA and RHCE practice exams; these are an essential part of readying yourself for the real testing 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 tests. The companion files, included on the DVD and through the book's companion web page, also includes more than 40 interactive exercises to help you learn and retain the knowledge needed to pass the exam and 4 simulations that take you through complex CLI exercises so that you can feel sure you're ready not only for the exams but also to actually use Red Hat Linux.

The core chapters are organized in two parts. The first part, which includes Chapters 1 through 24, covers RHCSA topics; the second part, which consists of Chapters 25 through 40, covers RHCE objectives. All the objectives in both exams are covered in these chapters.

The following topics are covered in the chapters:

Part 1: RHCSA

Part 1-1: Performing Basic System Management Tasks

- Chapter 1: Installing Red Hat Enterprise Linux Server

  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 how to create text files, but also how to look for specific contents in the different text files.

- Chapter 5: Connecting to a Red Hat Enterprise Linux 7

  This chapter teaches about the different methods that can be used to connect to RHEL 7. It explains local login as well as remote log in, and the different terminal types used for this purpose as well.

- Chapter 6: User and Group Management

  On Linux, users are used as an entity 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: Configuring Permissions

  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 1-2: Operating Running Systems

- Chapter 9: 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 10: Working with Virtual Machines

  Red Hat Enterprise Linux includes KVM, a complete solution that allows you to run virtual machines on top of RHEL. This chapter explains how to manage virtual machines.

- Chapter 11: Managing Software

  Red Hat offers an advanced system for managing software packages. This chapter teaches you how it works.

- 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 partitions

  Storage management is an important skill of 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 LVM Logical Volumes

  Dividing disks in partitions isn't very flexible. If you need optimal flexibility, you need LVM logical volumes, which are used by default while installing Red Hat Enterprise Linux. This chapter shows how to create and manage those logical volumes.

Part 1-3: 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: Configuring a Basic Apache Server

  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 18: 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 19: Troubleshooting the Boot Procedure

  Sometimes a misconfiguration might cause your server no longer to boot properly. This chapter teaches you some of the techniques that can be applied when normal server startup is no longer possible.

Part 1-4: Managing Network Services

- Chapter 20: Using Kickstart

  If you want to install one server, you can go through a manual installation procedure. If you need to install many servers, you're better off using an installation server. This chapter teaches you how to set up such a server.

- Chapter 21: Managing SELinux

  Many Linux administrators only know how to switch it off, because SELinux is hard to manage and is often the reason 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 22: Configuring a Firewall

  Apart from SELinux, RHEL 7 comes with a firewall, 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 23: Configuring Remote Mounts and FTP

  While 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 also shows how to configure an FTP server.

- Chapter 24: 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.

Part 2: RHCE

Part 2-1: System Configuration and Management

- Chapter 25: Configuring External Authentication and Authorization

  If you have multiple servers to manage, it makes sense to use an external authentication and authorization server, such as a Lightweight Directory Access Protocol (LDAP) server that uses Kerberos for authorization. This chapter teaches you how to set up a server for usage of an existing LDAP server that uses Kerberized authorization. It also explains Kerberos protocol fundamentals.

- Chapter 26: Configuring an iSCSI SAN

  RHEL 7 includes everything that is needed to set up a storage-area network (SAN). This chapter explains how to set up the SAN itself, using the iSCSI target software, and how to connect to a SAN, using the iSCSI initiator software on the client server.

- Chapter 27: System Performance Reporting

  Your server might sometimes have problems replying to a user request adequately. If that happens, you need to be able to find out what is wrong with it. This chapter explains performance reporting and all the different tools available to do this in an efficient way.

- Chapter 28: System Optimization Basics

  If you've found that something is wrong with your server's performance, you need to optimize it. In this chapter, you learn how to optimize your server for specific workloads.

- Chapter 29: Configuring Advanced Log Features

  The rsyslog service used for logging on RHEL 7 contains some advanced features, such as working with modules and setting up a remote log server. This chapter explains how to use these features.

- Chapter 30: Configuring Routing and Advanced Networking

  To integrate a server in a datacenter, advanced network configurations are often needed. This chapter explains how to do so and includes configuring aggregated network interfaces using bonding or teaming, as well as routing and IPv6 configurations.

- Chapter 31: 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 2-2: System Security

- Chapter 32: Advanced Firewall Configuration

  In Chapter 22, you learned how to set up a firewalld based firewall, using the default components of this firewall. This chapter zooms in on some more advanced configurations, including port forwarding and rich rules.

- Chapter 33: Managing Advanced Apache Services

  The Apache web server offers many solutions to access web content. That includes using virtual servers, but also includes authentication and the use of Transport Layer Security (TLS) certificates. This chapter teaches you how to manage these advanced features.

- Chapter 34: Configuring DNS

  In this chapter, you learn how to set up a caching-only DNS name server, which is useful to handle DNS requests more efficiently.

- Chapter 35: Configuring a MariaDB Database

  As a Linux administrator, you'll have to deal with database management as well. That is, you do not have to become a skilled DBA, but at least you need to know how to manage database backups, set up a simple database and perform database queries. This chapter teaches how to do all this.

- Chapter 36: Configuring NFS

  The Network File System (NFS) protocol is used to share files between Linux servers or between Linux servers and clients. This chapter teaches you how to set up Domain Name System (DNS), including advanced setups such as Kerberized NFS servers.

- Chapter 37: Configuring Samba File Services

  The Samba file server offers a solution to share directories on Linux to make them accessible for Windows clients. This chapter shows you how to set up a Samba server, and also discusses some of the advanced methods that Samba shares can be integrated in the client file system.

- Chapter 38: Setting Up an SMTP Server

  A Linux server occasionally needs to send email messages to other servers. This chapter shows how to set up a simple configuration for sending email using other mail servers as a relay host.

- Chapter 39: Configuring SSH

  The Secure Shell (SSH) service is used for remote access, but it can do so much more. In this chapter, you learn how to optimize the SSH service through its many parameters in the configuration files. You also learn how to set up SSH port forwarding.

- Chapter 40: Managing Time Synchronization

  As discussed earlier, time is a critical factor for many services to work successfully. This chapter explains how to manage time synchronization by using the Network Time Protocol (NTP).

Part 3: Final Preparation

- Chapter 41: Final Preparation

  In this chapter, you get some final exam preparation tasks. It contains some test exams and many tips that help you maximize your chances of passing the exam.

- Chapter 42: Theoretical Pre-Assessment Exams

  In this chapter, you'll get an RHCSA Theoretical Pre-Assessment Exam and an RHCE Theoretical Pre-Assessment Exam, so you can pre-assess your skills and determine the best route forward for studying for the exams.

- Practice Exams: This section supplies two RHCSA Practice Exams and two RHCE Practice Exams, so you can test your knowledge and skills before taking the exams. These exams are also available on the book's companion website and DVD as PDF files.

Chapter Features

To help you customize your study time using these books, 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.

- **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:** At the end of the "Foundation Topics" section of each chapter, the "Exam Preparation Tasks" section lists a series of study activities that should be done at the end of the chapter. Each chapter includes the activities that make the most sense for studying the topics in that chapter. The activities include the following:

- **Review Key Topics:** The Key Topic icon is shown next to the most important items in the "Foundation Topics" section of the chapter. The Key Topics Review 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 lists of facts, many of the more important lists and tables from the chapter are included in a document on the DVD and companion website. This document lists 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:** Questions at the end of each chapter that measure insight in the topics that were discussed in the chapter.
- **End-of-Chapter Labs:** Real labs that give you the right impression on 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 DVD and companion website, including the following:

- **Four practice exams:** The companion website and DVD contains the four practice exams, two RHCSA and two RHCE, provided in the book as PDFs so that you can readily test your skills before taking the exams in the testing facility.
- **Interactive exercises and quizzes:** The companion website and DVD contains more than 40 interactive hands-on exercises and 40 interactive quizzes so that you can test your knowledge on the spot.
- **Four advanced CLI simulations:** The companion website and DVD contains four advanced CLI simulations—two RHCSA and two RHCE—that allow you to walk through multistep CLI scenarios in a simulated environment.
- **Glossary quizzes:** The companion website and DVD contains interactive quizzes that allow you to test yourself on every glossary term in the book.

- **More than 2.5 hours of video training:** The companion website and DVD contains 30 minutes of unique test-prep videos plus more than 2 hours of instruction from the best-selling RHCSA and RHCE Complete Video Course series.
- **Virtual Machines:** The companion website and DVD contains three virtual machines so that you can easily get access to an environment where you can work on the labs and exercises in this book.

Book Organization, Chapters, and Appendixes

I have also included two tables that detail where every objective in the the RHCSA and RHCE exams is covered in this book so that you can more easily create a successful plan for passing the tests.

Table 1 RHCSA Objectives

Objective	Chapter Title	Chapter	Page
Understand and use essential tools			
Access a shell prompt and issue commands with correct syntax	Using Essential Tools	2	33
Use input-output redirection (>, >>, |, 2>, etc.)	Using Essential Tools	2	33
Use grep and regular expressions to analyze text	Working with Text Files	4	85
Access remote systems using ssh	Connecting to an RHEL Server	5	103
Log in and switch users in multiuser targets	Connecting to an RHEL Server	5	103
Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2	Essential File Management Tools	3	57
Create and edit text files	Working with Text Files	4	85
Create, delete, copy, and move files and directories	Essential File Management Tools	3	57
Create hard and soft links	Essential File Management Tools	3	57
List, set, and change standard ugo/rwx permissions	Permissions Management	7	151

Objective	Chapter Title	Chapter	Page
Locate, read, and use system documentation including man, info, and files in /usr/share/doc Note: Red Hat may use applications during the exam that are not included in Red Hat Enterprise Linux for the purpose of evaluating candidate's abilities to meet this objective.	Using Essential Tools	2	33
Operate running systems			
Boot, reboot, and shut down a system normally	Connecting to an RHEL Server	5	103
Boot systems into different targets manually	Essential Book Procedure Troubleshooting	19	429
Interrupt the boot process in order to gain access to a system	Essential Book Procedure Troubleshooting	19	429
Identify CPU/memory intensive processes, adjust process priority with renice, and kill processes	Process Management	9	205
Locate and interpret system log files and journals	Configuring Logging	13	295
Access a virtual machine's console	Working with Virtual Machines	10	225
Start and stop virtual machines	Working with Virtual Machines	10	225
Start, stop, and check the status of network services	Configuring Networking	8	177
Securely transfer files between systems	Connecting to an RHEL Server	5	103
Configure local storage			
List, create, and delete partitions on MBR and GPT disks	Managing Partitions	14	319
Create and remove physical volumes, assign physical volumes to volume groups, and create and delete logical volumes	Managing LVM Logical Volumes	15	349
Configure systems to mount file systems at boot by Universally Unique ID (UUID) or label	Managing Partitions	14	319

Objective	Chapter Title	Chapter	Page
Add new partitions and logical volumes, and swap to a system non-destructively	Managing Partitions	14	319
Create and configure file systems			
Create, mount, unmount, and use vfat, ext4, and xfs file systems	Managing Partitions	14	319
Mount and unmount CIFS and NFS network file systems	Configuring Remote Mounts and FTP	23	515
Extend existing logical volumes	Managing LVM Logical Volumes	15	349
Create and configure set-GID directories for collaboration	Permissions Management	7	151
Create and manage access control lists (ACLs)	Permissions Management	7	151
Diagnose and correct file permission problems	Permissions Management	7	151
Deploy, configure, and maintain systems			
Configure networking and hostname resolution statically or dynamically	Configuring Networking	8	177
Schedule tasks using at and cron	Scheduling Tasks	12	281
Start and stop services and configure services to start automatically at boot	Managing and Understanding the Boot Procedure	18	405
Configure systems to boot into a specific target automatically	Managing and Understanding the Boot Procedure	18	405
Install Red Hat Enterprise Linux automatically using Kickstart	Using Kickstart	20	451
Configure a physical machine to host virtual guests	Working with Virtual Machines	10	225
Install Red Hat Enterprise Linux systems as virtual guests	Working with Virtual Machines	10	225
Configure systems to launch virtual machines at boot	Working with Virtual Machines	10	225
Configure network services to start automatically at boot	Configuring Networking	8	177
Configure a system to use time services	Configuring Time Services	24	539

Objective	Chapter Title	Chapter	Page
Install and update software packages from Red Hat Network, a remote repository, or from the local file system	Installing Software Packages	11	249
Update the kernel package appropriately to ensure a bootable system	Basic Kernel Management	16	369
Modify the system bootloader	Managing and Understanding the Boot Procedure	18	405
Manage users and groups			
Create, delete, and modify local user accounts	User and Group Management	6	123
Change passwords and adjust password aging for local user accounts	User and Group Management	6	123
Create, delete, and modify local groups and group memberships	User and Group Management	6	123
Configure a system to use an existing authentication service for user and group information	User and Group Management	6	123
Manage security			
Configure firewall settings using firewall-config, firewall-cmd, or iptables	Configuring a Firewall	22	499
Configure key-based authentication for SSH	Connecting to an RHEL Server	5	103
Set enforcing and permissive modes for SELinux	Managing SELinux	21	473
List and identify SELinux file and process context	Managing SELinux	21	473
Restore default file contexts	Managing SELinux	21	473
Use boolean settings to modify system SELinux settings	Managing SELinux	21	473
Diagnose and address routine SELinux policy violations	Managing SELinux	21	473

Table 2 RHCE Objectives

Objective	Chapter Title	Chapter	Page
System configuration and management			
Use network teaming or bonding to configure aggregated network links between two Red Hat Enterprise Linux systems	Configuring Routing and Advanced Networking	30	655
Configure IPv6 addresses and perform basic IPv6 troubleshooting	Configuring Routing and Advanced Networking	30	655
Route IP traffic and create static routes	Configuring Routing and Advanced Networking	30	655
Use firewalld and associated mechanisms such as rich rules, zones and custom rules, to implement packet filtering and configure Network Address Translation (NAT)	Managing Linux-Based Firewalls	32	701
Use /proc/sys and sysctl to modify and set kernel runtime parameters	System Optimization Basics	28	627
Configure a system to authenticate using Kerberos	Configuring External Authentication and Authorization	25	557
Configure a system as either an iSCSI target or initiator that persistently mounts an iSCSI target	Configuring an iSCSI SAN	26	577
Produce and deliver reports on system utilization (processor, memory, disk, and network)	System Performance Reporting	27	607
Use shell scripting to automate system maintenance tasks	An Introduction to Bash Shell Scripting	31	683
Network services			
Install the packages needed to provide the service	Installing Software Packages	11, 25, 26, 31, 33, 34, 35, 36, 37, 38	249, 557, 577, 683, 719, 781, 801, 825
Configure SELinux to support the service	Managing SELinux	21, 33, 35, 36, 37, 39	473, 719, 759, 781, 801, 845
Use SELinux port labeling to allow services to use non-standard ports	Managing SELinux	2, 33, 391	33, 719, 845

Objective	Chapter Title	Chapter	Page
Configure the service to start when the system is booted	Managing and Understanding the Boot Procedure	18, 25, 26, 33, 34, 35, 36, 37, 38	405, 557, 577, 719, 741, 759, 781, 801, 825
Configure the service for basic operation	Sander please insert chapter names and numbers	25, 26, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40	557, 577, 641, 655, 683, 701, 719, 741, 759, 781, 801, 825, 845, 859
Configure host-based and user-based security for the service	Sander please insert chapter names and numbers	25, 26, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40	557, 577, 641, 655, 683, 701, 719, 741, 759, 781, 801, 825, 845, 859
HTTP/HTTPS			
Configure a virtual host	Managing Advanced Apache Services	33	719
Configure private directories	Managing Advanced Apache Services	33	719
Deploy a basic CGI application	Managing Advanced Apache Services	33	719
Configure group-managed content	Managing Advanced Apache Services	33	719
Configure TLS security	Managing Advanced Apache Services	33	719
DNS			
Configure a caching-only name server	Configuring DNS	34	741
Troubleshoot DNS client issues	Configuring DNS	34	741
NFS			
Provide network shares to specific clients	Configuring NFS	36	781
Provide network shares suitable for group collaboration	Configuring NFS	36	781
Use Kerberos to control access to NFS network shares	Configuring NFS	36	781
SMB			
Provide network shares to specific clients	Configuring Samba File Services	37	801

Objective	Chapter Title	Chapter	Page
Provide network shares suitable for group collaboration	Configuring Samba File Services	37	801
SMTP			
Configure a system to forward all email to a central mail server	Setting up an SMTP Server	38	825
SSH			
Configure key-based authentication	Configuring SSH	39	845
Configure additional options described in documentation	Configuring SSH	39	845
NTP			
Synchronize time using other NTP peers	Managing Time Synchronization	40	859
Database services			
Install and configure MariaDB	Configuring a MariaDB Database	35	759
Backup and restore a database	Configuring a MariaDB Database	35	759
Create a simple database schema	Configuring a MariaDB Database	35	759
Perform simple SQL queries against a database	Configuring a MariaDB Database	35	759

About the Virtual Machines

On the companion website and DVD that comes with this book, you can find a set of virtual machines that you can use to work through the labs in this book. Alternatively, you can set up an IPA server as instructed in Appendix D of this book and install your own environment according to the instructions in Chapter 1.

To use the virtual machines, you need VMware Workstation/Player or Fusion. Using the VMs if one of these virtualization platforms is installed on your computer is easy—just doubleclick the files that you'll find in the archive after uncompressing it. Notice that you'll need the following to use these virtual machines:

- A total of 30 GB of disk space
- A total of 3 GB of available RAM

Where are the Companion Content Files?

Register this print version of Red Hat RHCSA/RHCE 7 Cert Guide to access the Bonus content online.

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

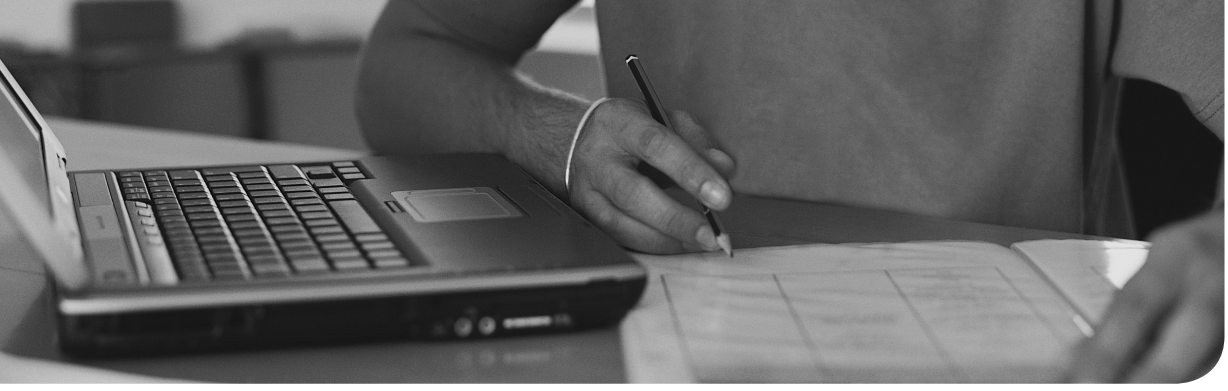
1. Go to www.pearsonITcertification.com/register and log in or create a new account.
2. Enter the ISBN: 9780789754059
3. Answer the challenge question as proof of purchase.
4. Click on the "Access Bonus Content" link in the Registered Products section of your account page, to be taken to the page where your downloadable content is available.

Please note that many of our 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 at left, please visit www.pearsonITcertification.com/ contact and select the "Site Problems/ Comments" option. Our customer service representatives will assist you.

This book also includes an exclusive offer for 70% off the Premium Edition eBook and Practice Tests edition of this title. Please see the coupon code included with the DVD for information on how to purchase the Premium Edition.

This page intentionally left blank



The following topics are covered in this chapter:

- Working with Systemd
- Working with GRUB 2

The following RHCSA exam objectives are covered in this chapter:

- Start and stop services and configure services to start automatically at boot
- Configure systems to boot into a specific target automatically
- Modify the system bootloader

Managing and Understanding the Boot Procedure

In this chapter, you learn how the boot procedure on Red Hat Enterprise Linux is organized. We first go through a section about systemd, the overall service that takes care of starting everything on your server. In this section, you also learn how systemd targets are used to group systemd units and come to a final operational environment.

The second part of this chapter discusses GRUB2 and how to apply changes to the GRUB 2 boot loader. Troubleshooting is not a topic in this chapter; it is covered in Chapter 19, "Troubleshooting 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 18.1 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 18.1 "Do I Know This Already?" Section-to-Question Mapping

Foundation Topics Section	Questions
Working with Systemd	1–7, 10
Working with GRUB 2	8, 9

1. Which command shows all service unit files on your system that are currently loaded?
   a. **systemctl --type=service**
   b. **systemctl --type=service --all**
   c. **systemctl --list-services**
   d. **systemctl --show-units | grep services**

2. Which statement about systemd wants is *not* true?
   a. You can create wants by using the **systemctl enable** command.
   b. The target to which a specific want applies is agnostic of the associated wants.
   c. Wants are always administered in the /usr/lib/systemd/system directory.
   d. Each service knows to which target its wants should be added.

3. What is the best solution to avoid conflicts between incompatible units?
   a. Nothing, the unit files have defined for themselves which units they are not compatible with.
   b. Disable the service using **systemctl disable**.
   c. Unmask the service using **systemctl unmask**.
   d. Mask the service using **systemctl mask**.

4. Which of the following is not a valid status for systemd services?
   a. Running(active)
   b. Running(exited)
   c. Running(waiting)
   d. Running(dead)

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. Which of the following is not a valid systemd unit type?
   a. service
   b. udev
   c. mount
   d. socket

7. You want to find out which other systemd units have dependencies to this specific unit. Which command would you use?
   a. **systemd list-dependencies --reverse**
   b. **systemctl list-dependencies --reverse**
   c. **systemctl status my.unit --show-deps**
   d. **systemd status my.unit --show-deps -r**

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. Which of the following is *not* a valid command while working with units in systemctl?
    a. **systemctl unit start**
    b. **systemctl status -l unit**
    c. **systemctl mask unit**
    d. **systemctl disable unit**

Foundation Topics

Working with Systemd

Systemd is the new service in Red Hat Enterprise Linux 7 that is responsible for starting all kinds of things. Systemd goes way beyond starting services; other items are started from systemd as well. In this chapter, you learn how systemd is organized and what items are started from systemd.

Understanding Systemd

To describe it in a generic way, the systemd System and Service Manager is used to start stuff. The stuff is referred to as *units*. Units can be many things. One of the most important unit types is the service. Typically, services are processes that provide specific functionality and allow connections from external clients coming in. Apart from services, other unit types exist, such as sockets, mounts, and others. To display a list of available units, type **systemctl -t help** (see Listing 18.1).

Key Topic

Lising 18.1 Unit Types in Systemd

```
[root@server1 ~]# systemctl -t help
Available unit types:
service
socket
target
device
mount
automount
snapshot
timer
swap
path
slice
scope
```

TIP For RHCSA, you need to know how to work with services. The other unit types do not matter that much.

Understanding Service Units

The major benefit of working with systemd, as compared to previous methods Red Hat used for managing services, is that it provides a uniform interface to start units. This interface is defined in the unit file. The system default unit files are in /usr/lib/systemd/system. System-specific modifications (overriding the defaults) are in /etc/systemd/system. Also, the runtime configuration that is generated automatically is stored in /run/systemd/system. Listing 18.2 gives an example of the vsftpd.service unit file.

Listing 18.2 Example of the Vsftpd Unit File

```
[Unit]
Description=Vsftpd ftp daemon
After=network.target

[Service]
Type=forking
ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

[Install]
WantedBy=multi-user.target
```

From this unit file example, you can see that it is relatively easy to understand. Any systemd service unit file consists of three sections. (You'll find different sections in other types of unit files.)

- **[Unit]**, which describes the unit and defines dependencies. This section also contains the important **After** statement, and optionally the **Before** statement. These statements define dependencies between different units. The Before statement relates to another unit that is started after this unit. The after unit refers to a unit that needs to be started before this unit can be started.

Key Topic

- **[Service]**, in which there is a description on how to start and stop the service and request status installation. Normally, you can expect an ExecStart line, which indicates how to start the unit, or an ExecStop line, which indicates how to stop the unit.
- **[Install]**, in which the wants are taken care of. You'll read more about this in the next section, "Understanding Target Units."

Listing 18.3 shows another example of a unit file. This time it is the tmp.mount unit.

Listing 18.3 Example of a Mount Unit File

```
[Unit]
Description=Temporary Directory
Documentation=man:hier(7)
Documentation=http://www.freedesktop.org/wiki/Software/systemd/
  APIFileSystems
DefaultDependencies=no
Conflicts=umount.target
Before=local-fs.target umount.target

[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,strictatime

# Make 'systemctl enable tmp.mount' work:
[Install]
WantedBy=local-fs.target
```

The tmp.mount unit file in Listing 18.3 shows some interesting additional information. In the Unit section, you can see the **Conflicts** statement. This is used to list units that cannot be used together with this unit. Use this for mutually exclusive units. Next, there is the Mount section, which is specific for this unit type and defines where exactly the mount has to be performed. You'll recognize the arguments that are typically used in any **mount** command. Last, there is the WantedBy section, which defines where the unit has to be started.

Another type of unit that is interesting to look at is the socket. A socket creates a method for applications to communicate with one another. Some services create their own sockets while starting, whereas other services need a socket unit file to create sockets for them. It is also the other way around: Every socket needs a corresponding service file. The socket file example in Listing 18.4 shows how this happens for virtlockd, a systemd socket that tracks activity for virtual machines.

Listing 18.4 Socket Unit File Example

```
[root@server202 system]# cat virtlockd.socket
[Unit]
Description=Virtual machine lock manager socket

[Socket]
ListenStream=/var/run/libvirt/virtlockd-sock

[Install]
WantedBy=multi-user.target
```

When working with systemd unit files, you risk getting overwhelmed with options. Every unit file can be configured with different options. To figure out which options are available for a specific unit, use the **systemctl show** command. For instance, the **systemctl show sshd** command shows all systemd options that can be configured in the sshd.service unit, including their current default values. Listing 18.5 shows the output of this command.

Listing 18.5 Showing Available Options with **systemctl show**

```
Id=sshd.service
Names=sshd.service
Requires=basic.target
Wants=sshd-keygen.service system.slice
WantedBy=multi-user.target
ConsistsOf=sshd-keygen.service
Conflicts=shutdown.target
ConflictedBy=sshd.socket
Before=shutdown.target multi-user.target
After=network.target sshd-keygen.service systemd-journald.socket
basic.target system.slice
Description=OpenSSH server daemon
LoadState=loaded
ActiveState=active
SubState=running
FragmentPath=/usr/lib/systemd/system/sshd.service
UnitFileState=enabled
InactiveExitTimestamp=Sat 2015-05-02 11:06:02 EDT
InactiveExitTimestampMonotonic=2596332166
```

```
ActiveEnterTimestamp=Sat 2015-05-02 11:06:02 EDT
ActiveEnterTimestampMonotonic=2596332166
ActiveExitTimestamp=Sat 2015-05-02 11:05:22 EDT
ActiveExitTimestampMonotonic=2559916100
InactiveEnterTimestamp=Sat 2015-05-02 11:06:02 EDT
InactiveEnterTimestampMonotonic=2596331238
CanStart=yes
CanStop=yes
CanReload=yes
CanIsolate=no
StopWhenUnneeded=no
RefuseManualStart=no
RefuseManualStop=no
AllowIsolate=no
DefaultDependencies=yes
OnFailureIsolate=no
IgnoreOnIsolate=no
```

Understanding Target Units

The unit files are used to build the functionality that is needed on your server. To make it possible to load them in the right order and at the right moment, a specific type of unit is used: the target unit. A simple definition of a target unit is "a group of units." Some targets are used as the equivalents to the old run levels, which on earlier versions of RHEL were used to define the state a server should be started in. A run level was a collection of services that were needed for a server to be started in multi-user mode or in graphical mode. Targets go beyond that. A good starting point to understanding targets is to see them as a group of units.

Targets by themselves can have dependencies to other targets, which are defined in the target unit. Let's take a look at Listing 18.6, where you can see the definition of the multi-user.target, which defines the normal operational state of an RHEL server.

Listing 18.6 The Multi-user.target File

```
[root@server202 system]# cat multi-user.target
...

[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

[Install]
Alias=default.target
```

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. And you can see that in the Install section it is defined as the default.target, so this is what your server starts by default. 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").

Even if a systemd target looks a bit like the old run levels, it is 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, like the multi-user or graphical targets.

Understanding Wants

To understand the concept of a want, let's start looking at 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, 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 Units Through Systemd

As an administrator, you need to manage systemd units. It starts by starting and stopping units. You use the **systemctl** command to do that. In Exercise 18.1, you walk start, stop, and manage a unit. After you have configured a unit so that it can be started without problems, you need to make sure that it restarts automatically upon reboot. You do this by enabling or disabling the unit.

TIP Memorizing all the different arguments that can be used with the **systemctl** command might seem hard, but you don't have to do that. Instead, just type **systemctl** and press the **Tab** key twice to use command autocompletion. This will show you all available commands.

Key Topic

Exercise 18.1 Managing Units with systemctl

1. Type **yum -y install vsftpd** to install the Very Secure FTP service.
2. Type **systemctl start vsftpd**. This activates the FTP server on your machine.
3. Type **systemctl status vsftpd**. You'll get an output as in Listing 18.7 and see that the vsftpd service is currently operational. You can also see in the Loaded line that it is currently disabled, which means that it will not be activated on a system restart.
4. Type **systemctl enable vsftpd**. This creates a symbolic link in the wants directory for the multi-user target to ensure that the service gets back after a restart.
5. Type **systemctl status vsftpd** again. You'll now see that the unit file has changed from being disabled to enabled.

Listing 18.7 Requesting Current Unit Status with **systemctl status**

```
[root@server202 system]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled)
   Active: active (running) since Sun 2014-09-28 08:42:59 EDT; 2s ago
  Process: 34468 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
(code=exited, status=0/SUCCESS)
 Main PID: 34469 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─34469 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Sep 28 08:42:59 server202.example.com systemd[1]: Starting Vsftpd ftp
daemon...
Sep 28 08:42:59 server202.example.com systemd[1]: Started Vsftpd ftp
daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

When requesting the current status of a systemd unit as in Listing 18.6, you can see different kinds of information about it. Table 18.2 shows the different kinds of information that you can get about unit files when using the **systemctl status** command:

Table 18.2 Systemd Status Overview

Status	Description
Loaded	The unit file has been processed and the unit is active.
Active(running)	Running with one or more active processes.
Active(exited)	Successfully completed a one-time configuration.
Active(waiting)	Running and waiting for an event.
Inactive	Not running.
Enabled	Will be started at boot time.
Disabled	Will not be started at boot time.
Static	This unit can not be enabled but may be started by another unit automatically.

As an administrator, you'll also often need to get a current overview of the current status of systemd unit files. Different commands can help you to get this insight, some of which are shown in Table 18.3.

Table 18.3 Systemctl Unit Overview Commands

Command	Description
systemctl --type=service	Shows only service units
systemctl list-units --type=service	Shows all active service units (same result as the previous command)
systemctl list-units --type=service --all	Shows inactive service units as well as active service units
systemctl --failed --type=service	Shows all services that have failed
systemctl status -l your.service	Shows detailed status information about services

Managing Dependencies

Systemctl units in many cases have dependencies. Some units will be started as a dependency of other units, and an event where one specific unit is requested may

trigger the start of another unit. An example is the cups.service service. This service can be started by itself, but it can also be started by activity on the cups.path and cups.socket units, which may trigger the service to be started again. As an administrator, you can request a list of unit dependencies. Type **systemctl list-dependencies** followed by a unit name to find out which dependencies it has, and add the **--reverse** option to find out which units are dependent of this unit. Listing 18.8 shows an example of this command.

Listing 18.8 Showing Unit Dependencies

```
[root@server1 ~]# systemctl list-dependencies vsftpd
   vsftpd.service
├─system.slice
└─basic.target
  ├─alsa-restore.service
  ├─alsa-state.service
  ├─firewalld.service
  ├─microcode.service
  ├─rhel-autorelabel-mark.service
  ├─rhel-autorelabel.service
  ├─rhel-configure.service
  ├─rhel-dmesg.service
  ├─rhel-loadmodules.service
  ├─paths.target
  ├─slices.target
  | ├─-.slice
  | └─system.slice
  ├─sockets.target
  | ├─avahi-daemon.socket
  | ├─cups.socket
  | ├─dbus.socket
  | ├─dm-event.socket
  | ├─iscsid.socket
  | ├─iscsiuio.socket
  | ├─lvm2-lvmetad.socket
  | ├─rpcbind.socket
  | ├─systemd-initctl.socket
  | ├─systemd-journald.socket
  | ├─systemd-shutdownd.socket
  | ├─systemd-udevd-control.socket
  | └─systemd-udevd-kernel.socket
```

```
├─sysinit.target
│ ├─dev-hugepages.mount
│ ├─dev-mqueue.mount
│ ├─dmraid-activation.service
│ ├─iscsi.service
```

Apart from dependencies, some units have conflicts with other units. Examples of these include the following:

- Mount and umount units that cannot be loaded together
- The network and NetworkManager service
- The iptables and the firewalld service
- The cronyd and ntpd service

If units have conflicts with other units, this is described in the unit file. As an administrator, you can also make sure that conflicting units will never be loaded at the same time on the same system. To do this, you can use the **systemctl mask** command, which basically makes a unit no longer a candidate for being started. Apply the following procedure to find out how it works:

1. Open a root shell and type **systemctl status firewalld**. Next type **systemctl status iptables**. If one of the services is active, do not load it again in the next step.
2. Type **systemctl start firewalld** and **systemctl start iptables** to load both services. You will see that iptables refuses to start; this is because the firewalld service is already activated.
3. Type **cat /usr/lib/systemd/system/firewalld.service**. Notice the conflicts setting. Type **cat /usr/lib/systemd/system/iptables.service**. Notice that this unit does not have a conflicts line.
4. Unload both services by using **systemctl stop firewalld** followed by **systemctl stop iptables**. Notice that it is not really necessary to stop the iptables service because it has failed to load, but we really need to make sure that it is not loaded at all before continuing.
5. Type **systemctl mask iptables** and look at what is happening: A symbolic link to /dev/null is created for /etc/systemd/system/iptables.service (as you can see in the output of the following command example). Because the unit files in

/etc/systemd have precedence over the files in /usr/lib/systemd, this makes it impossible to start the iptables service by accident:

```
[root@server202 system]# systemctl mask iptables
ln -s '/dev/null' '/etc/systemd/system/iptables.service'
```

6. Type **systemctl start iptables**. You'll see an error message indicating that this service is masked and for that reason cannot be started.
7. Type **systemctl enable iptables**. Notice that no error message is shown and it looks as if it is working all right. Restart your server using **systemctl reboot** (or just **reboot**).
8. After restart, type **systemctl status -l iptables**. You'll see that it is inactive and that the loaded status is indicated as masked:

```
[root@server202 ~]# systemctl status -l iptables
iptables.service
   Loaded: masked (/dev/null)
   Active: inactive (dead)
```

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. The services know for themselves in which targets they need to be started and a want is created automatically in that target. The following procedure walks you through the steps of enabling a service:

1. Type **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, on the background a symbolic link is created.

TIP On both the RHCSA and the RHCE exams, 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 a couple of 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. By isolating a target, you start that target with all of its dependencies. Not all targets can be isolated, but only targets that have the isolate option enabled. We'll explore the **systemctl isolate** command in a while. 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 does not show all the targets, but only the active targets. Type **systemctl --type=target --all** for an overview of all targets that exist on your computer. You'll now see inactive targets also.

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 target. These targets also roughly correspond to run levels as they were 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 18.2 shows you how to do this.

Key Topic

Exercise 18.2 Isolating Targets

1. From a root shell, go to the directory /usr/lib/systemd/system. Type **grep Isolate *.target**. This shows a list of all targets that allow isolation.
2. Type **systemctl isolate rescue.target**. This 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 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 7 boot procedure is discussed in more detail in Chapter 19, 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. In this file, you'll find options that tell GRUB what to do and how to do it. Listing 18.9 shows the contents of this file after an installation with default settings of CentOS 7.

Listing 18.9 Contents of the /etc/default/grub File

```
[root@server202 grub.d]# 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="rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-
  sun16 rd.lvm.lv=centos/root crashkernel=auto  vconsole.keymap=us
  rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

Key Topic

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 it. You also will not need to modify anything if you want to make it possible 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.

Based on the configuration files mentioned previously, the main configuration file /boot/grub2/grub.cfg is created. Even if this looks like a configuration file that can be manually modified, you should never do that, because it will get overwritten at some point. This will happen, for instance, after updating the kernel. The RPM from which the kernel is updated will run a post-installation script that regenerates

the kernel. In the next section, you learn how to make changes to the GRUB 2 configuration.

If you enter the GRUB 2 boot prompt to add kernel startup parameters, the contents of the /boot/grub2/grub.cfg file display. From here, you add one-time-only startup options. Listing 18.10 shows the relevant part of the grub.cfg file that takes care of loading the Linux kernel. In this listing, you see the part of the configuration file that takes care of loading the default kernel. Notice the line that starts with linux16; this line specifies all kernel boot parameters.

Listing 18.10 Partial Contents of the /boot/grub2/grub.cfg Configuration File

```
menuentry 'CentOS Linux (3.10.0-229.1.2.el7.x86_64) 7 (Core)' --class
centos --class gnu-linux --class gnu --class os --unrestricted
$menuentry_id_option 'gnulinux-3.10.0-123.el7.x86_64-advanced-50faa2a1-
01d3-430b-8114-4a98daf5bdb9' {
      load_video
      set gfxpayload=keep
      insmod gzio
      insmod part_msdos
      insmod xfs
      set root='hd0,msdos1'
      if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1'
057ba3d8-bfe7-4676-bb99-79e9980a1966
     else
       search --no-floppy --fs-uuid --set=root 057ba3d8-bfe7-4676-bb99-
79e9980a1966
     fi
     linux16 /vmlinuz-3.10.0-229.1.2.el7.x86_64 root=/dev/mapper/
centos-root ro rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-sun16
rd.lvm.lv=centos/root crashkernel=auto  vconsole.keymap=us rhgb quiet
LANG=en_US.UTF-8
     initrd16 /initramfs-3.10.0-229.1.2.el7.x86_64.img
}
```

One of the most important differences between GRUB 2 and its previous version is the availability of GRUB 2 modules. In GRUB 2, a large number of modules are available. By default, you can find them in /boot/grub2/i386-pc. The modules determine what you can and what you cannot do from the GRUB 2 boot loader. If some hardware or file system is not supported in what you want to do, check here to make

sure that a supporting GRUB 2 module is available. In Listing 18.10, you can see examples of the code used to load specific GRUB 2 modules.

Modifying Default GRUB 2 Boot Options

To apply modifications to the GRUB 2 boot loader, the file /etc/default/grub is your entry point; do not change the contents of the /boot/grub2/grub.cfg configuration file directly. 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.

In Exercise 18.3, you learn how to apply modifications to GRUB 2.

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 18.3 Applying Modifications to GRUB2

In this exercise you'll apply some changes to the GRUB2 boot configuration and write them to the /boot/grub2/grub.cfg configuration file.

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 also how units can be configured for automatic start with the use of targets. You also 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 Prep Tasks

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 18.4 lists a reference of these key topics and the page numbers on which each is found.

Table 18.4 Key Topics for Chapter 18

Key Topic Element	Description	Page Number
Listing 18.1	Unit types in systemd	408
List	Three sections of a systemd unit file	409
Paragraph	Understanding Target Units	412
Exercise 18.1	Managing units with **systemctl**	414

Key Topic Element	Description	Page Number
Table 18.3	Systemctl unit overview commands	415
Section	Managing systemd targets	418
Exercise 18.2	Isolating targets	420
Bullet list	Explanation of the role of kernel and initramfs	420
Listing 18.9	Contents of the /etc/default/grub file	421
Exercise 18.3	Applying modifications to GRUB 2	424

Complete Tables and Lists from Memory

Print a copy of Appendix B, "Memory Tables" (found on the disc), or at least the section for this chapter, and complete the tables and lists from memory. Appendix C, "Memory Tables Answer Key," also on the disc, includes completed tables and lists to check your work.

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

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 units and the GRUB 2 boot loader. Before continuing, 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 18.1

Make sure that the firewalld service is started on boot. Also make sure that the iptables service can never be started at the same time.

Lab 18.2

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

This page intentionally left blank