



Cert Guide

Advance your IT career with hands-on learning

Red Hat RHCE™ 8

(EX294)



Practice
Tests



Flash
Cards



Labs



Study
Planner



SANDER VAN VUGT

Red Hat RHCE™ 8 (EX294) Cert Guide

Sander van Vugt



Pearson
221 River Street
Hoboken, NJ 07030 USA

Red Hat RHCE 8 (EX294) Cert Guide

Copyright © 2021 by Pearson Education, Inc.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-0-13-687243-6

ISBN-10: 0-13-687243-3

Library of Congress Control Number: 2020941743

21 02

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.

Pearson IT Certification and Sander van Vugt have no affiliation with Red Hat, Inc. The RED HAT and RHCE trademarks are used for identification purposes only and are not intended to indicate affiliation with or approval by Red Hat, Inc.

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.

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 intlcs@pearson.com.

Editor-in-Chief

Mark Taub

Product Line Manager

Brett Bartow

Executive Editor

Denise Lincoln

Development Editor

Ellie Bru

Managing Editor

Sandra Schroeder

Senior Project Editor

Tonya Simpson

Copy Editor

Chuck Hutchinson

Indexer

Timothy Wright

Proofreader

Donna Mulder

Technical Editors

John McDonough

William “Bo” Rothwell

Publishing Coordinator

Cindy Teeters

Cover Designer

Chuti Prasertsith

Compositor

codeMantra

Contents at a Glance

Introduction xxii

Part I: Introduction to Ansible

- CHAPTER 1 Understanding Configuration Management 3
- CHAPTER 2 Installing Ansible 15
- CHAPTER 3 Setting Up an Ansible Managed Environment 31
- CHAPTER 4 Using Ad Hoc Commands 47
- CHAPTER 5 Getting Started with Playbooks 69
- CHAPTER 6 Working with Variables and Facts 97

Part II: Common Ansible Management Tasks

- CHAPTER 7 Using Task Control 131
- CHAPTER 8 Deploying Files 173
- CHAPTER 9 Using Ansible Roles 205
- CHAPTER 10 Using Ansible in Large Environments 229
- CHAPTER 11 Troubleshooting Ansible 251

Part III: Managing Systems with Ansible

- CHAPTER 12 Managing Software with Ansible 281
- CHAPTER 13 Managing Users 305
- CHAPTER 14 Managing Services and the Boot Process 333
- CHAPTER 15 Managing Storage 351
- CHAPTER 16 Final Preparation 383

Part IV: Practice Exams

- Practice Exam A 387
- Practice Exam B 391

Part V: Appendixes

- APPENDIX A Answers to the “Do I Know This Already?” Quizzes and Review Questions 395
- APPENDIX B Getting Started with Ansible Tower 415

APPENDIX C *Red Hat RHCE 8 (EX294) Cert Guide Exam Updates* 431

Glossary 433

Index 439

Online Elements:

Practice Exam C

Practice Exam D

APPENDIX D Memory Tables**APPENDIX E** Memory Tables Answer Key**APPENDIX F** Study Planner

Glossary

Table of Contents

Introduction xxii

Part I: Introduction to Ansible

Chapter 1 Understanding Configuration Management 3

“Do I Know This Already?” Quiz 3

Foundation Topics 6

Understanding Automation 6

What Is Automation? 6

Understanding the DevOps Way of Working 7

Understanding Infrastructure as Code 8

Other Automation Solutions 8

Understanding Ansible Essential Components 9

Ansible Is Python 9

Ansible Architecture 9

Understanding Ansible Tower 10

Understanding the Ansible Way 11

Understanding Ansible Use Cases 11

Using Ansible for Configuration Management 11

Using Ansible for Provisioning 12

Using Ansible for Continuous Delivery 12

Summary 12

Exam Preparation Tasks 12

Review All Key Topics 12

Define Key Terms 13

Chapter 2 Installing Ansible 15

“Do I Know This Already?” Quiz 15

Foundation Topics 18

Understanding an Ansible Environment 18

Understanding Controller Host Requirements 20

Understanding Installation Methods 20

Installing Ansible on RHEL 8 21

Installing Ansible on CentOS 8 21

Using python-pip to Install Ansible 21

Configuring Managed Hosts	22
Configuring the Ansible User	23
Setting Up SSH for the Ansible User	23
Configuring sudo for the Ansible User	25
Summary	26
Exam Preparation Tasks	26
Review All Key Topics	26
Memory Tables	27
Define Key Terms	27
Review Questions	27
End-of-Chapter Lab	28
Lab 2-1	28
Chapter 3 Setting Up an Ansible Managed Environment	31
“Do I Know This Already?” Quiz	31
Foundation Topics	34
Understanding Projects	34
Configuring Static Inventory	35
Listing Hosts	35
Inventory Host Groups	36
Using Inventory in Commands	37
Specifying Host Variables	38
Working with Dynamic Inventory	39
Working with Community-Provided Dynamic Inventory Scripts	39
Writing Your Own Inventory Scripts	40
Using Multiple Inventory Files	42
Managing Settings in ansible.cfg	42
Summary	44
Exam Preparation Tasks	44
Review All Key Topics	44
Memory Tables	44
Define Key Terms	44
Review Questions	45
End-of-Chapter Lab	45
Lab 3-1	45

Chapter 4 Using Ad Hoc Commands 47

- “Do I Know This Already?” Quiz 47
- Foundation Topics 50
- Understanding Ad Hoc Commands 50
- Working with Modules 53
 - Exploring Essential Modules 54
 - command* 54
 - shell* 55
 - raw* 55
 - copy* 55
 - yum* 56
 - service* 56
 - ping* 56
 - Browsing Available Modules 58
- Consulting Module Documentation 58
 - Using `ansible-doc` 58
 - Using <https://docs.ansible.com> 62
- Running Ad Hoc Commands from Shell Scripts 64
- Summary 66
- Exam Preparation Tasks 66
- Review All Key Topics 66
- Memory Tables 66
- Define Key Terms 66
- Review Questions 67
- End-of-Chapter Lab 67
- Lab 4-1 67

Chapter 5 Getting Started with Playbooks 69

- “Do I Know This Already?” Quiz 69
- Foundation Topics 72
- Exploring Your First Playbook 72
 - From Ad Hoc Commands to Playbook 72
 - Playbook Elements 73
 - Running the Playbook 74
 - Undoing Playbook Modifications 75

	Working with YAML	77
	Indentation	77
	Using Key-Value Pairs	78
	Understanding YAML Lists	79
	Using YAML Strings	79
	Verifying Syntax	80
	Performing a Playbook Dry Run	81
	Managing Multiplay Playbooks	83
	Multiplay Playbook Considerations	83
	Multiplay Playbook Example	84
	Increasing Output Verbosity	86
	Summary	92
	Exam Preparation Tasks	92
	Review All Key Topics	93
	Memory Tables	93
	Define Key Terms	93
	Review Questions	93
	End-of-Chapter Lab	94
	Lab 5-1	94
Chapter 6	Working with Variables and Facts	97
	“Do I Know This Already?” Quiz	97
	Foundation Topics	100
	Understanding the Use of Variables in Ansible Playbooks	100
	Working with Ansible Facts	101
	Gathering Facts	102
	Understanding How Facts Are Displayed	104
	Managing Fact Gathering	108
	Working with Custom Facts	109
	Working with Variables	111
	Defining Variables	112
	Using Include Files	112
	Managing Host and Group Variables	113
	Using Multivalued Variables	115
	Working with Magic Variables	117
	Understanding Variable Precedence	119

Using Vault to Manage Sensitive Values	119
Understanding Vault	119
Managing Encrypted Files	120
Using Vault in Playbooks	120
Managing Files with Sensitive Variables	121
Capturing Command Output Using register	122
Summary	125
Exam Preparation Tasks	125
Review All Key Topics	125
Memory Tables	125
Define Key Terms	126
Review Questions	126
End-of-Chapter Labs	126
Lab 6-1	127
Lab 6-2	127

Part II: Common Ansible Management Tasks

Chapter 7 Using Task Control 131

“Do I Know This Already?” Quiz	131
Foundation Topics	134
Using Loops and Items	134
Working with Loops	134
Using Loops on Variables	135
Using Loops on Multivalued Variables	136
Understanding with_items	137
Using when to Run Tasks Conditionally	139
Working with when	139
Using Conditional Test Statements	141
Testing Multiple Conditions	144
Combining loop and when	145
Combining loop and register	147
Using Handlers	149
Working with Handlers	149
Understanding Handler Execution and Exceptions	154

Dealing with Failures	156
Understanding Task Execution	156
Managing Task Errors	156
Specifying Task Failure Conditions	158
Managing Changed Status	161
Using Blocks	163
Using Blocks with rescue and always Statements	164
Summary	169
Exam Preparation Tasks	169
Review All Key Topics	169
Memory Tables	170
Define Key Terms	170
Review Questions	170
End-of-Chapter Lab	171
Lab 7-1	171

Chapter 8 Deploying Files 173

“Do I Know This Already?” Quiz	173
Foundation Topics	176
Using Modules to Manipulate Files	176
File Module Manipulation Overview	176
Managing File Attributes	176
Managing File Contents	180
Creating and Removing Files	182
Moving Files Around	183
Managing SELinux Properties	187
Managing SELinux File Context	188
Applying Generic SELinux Management Tasks	189
Using Jinja2 Templates	194
Working with Simple Templates	194
Applying Control Structures in Jinja2 Using for	197
Using Conditional Statements with if	198
Using Filters	198
Summary	200

	Exam Preparation Tasks	200
	Review All Key Topics	200
	Memory Tables	201
	Define Key Terms	201
	Review Questions	201
	Exercise Answers	201
	End-of-Chapter Labs	202
	Lab 8-1: Generate an /etc/hosts File	202
	Lab 8-2: Manage a vsftpd Service	202
Chapter 9	Using Ansible Roles	205
	“Do I Know This Already?” Quiz	205
	Foundation Topics	208
	Using Ansible Roles	208
	Understanding Ansible Roles	208
	Understanding Role Location	209
	Using Roles from Playbooks	210
	Creating Custom Roles	210
	Managing Role Dependencies	213
	Understanding File Organization Best Practices	213
	Using Ansible Galaxy Roles	214
	Working with Galaxy	215
	Using the ansible-galaxy Command	216
	Managing Ansible Galaxy Roles	218
	Using RHEL System Roles	219
	Understanding RHEL System Roles	220
	Installing RHEL System Roles	220
	Using the RHEL SELinux System Role	221
	Using the RHEL TimeSync System Role	224
	Summary	225
	Exam Preparation Tasks	225
	Review All Key Topics	226
	Memory Tables	226
	Define Key Terms	226
	Review Questions	226

End-of-Chapter Labs 227

Lab 9-1 227

Lab 9-2 227

Chapter 10 Using Ansible in Large Environments 229

“Do I Know This Already?” Quiz 229

Foundation Topics 232

Advanced Inventory Usage 232

Working with Host Name Patterns 232

Configuring Dynamic Inventory 233

Using the ansible-inventory Command 235

Working with Multiple Inventory Files 236

Optimizing Ansible Processing 237

Managing Parallel Task Execution 237

Managing Serial Task Execution 238

Including and Importing Files 239

Importing Playbooks 240

Importing and Including Task Files 242

Using Variables When Importing and Including Files 243

Summary 247

Exam Preparation Tasks 247

Review All Key Topics 248

Define Key Terms 248

Review Questions 248

End-of-Chapter Lab 249

Lab 10-1 249

Chapter 11 Troubleshooting Ansible 251

“Do I Know this Already?” Quiz 251

Foundation Topics 254

Managing Ansible Errors and Logs 254

Using Check Mode 254

Understanding Output 256

Optimizing Command Output Error Formatting 259

Logging to Files 259

Running Task by Task 260

Using Modules for Troubleshooting and Testing	263
Using the Debug Module	263
Using the uri Module	263
Using the stat Module	265
Using the assert Module	268
Using Tags	271
Troubleshooting Common Scenarios	275
Analyzing Connectivity Issues	275
Analyzing Authentication Issues	276
Summary	277
Exam Preparation Tasks	277
Review All Key Topics	278
Memory Tables	278
Define Key Terms	278
Review Questions	278
End-of-Chapter Lab	279
Lab 11-1	279

Part III: Managing Systems with Ansible

Chapter 12 Managing Software with Ansible 281

“Do I Know This Already?” Quiz	281
Foundation Topics	284
Using Modules to Manage Packages	284
Configuring Repository Access	284
Managing Software with yum	285
Managing Package Facts	287
Using Modules to Manage Repositories and Subscriptions	289
Setting Up Repositories	290
Managing GPG Keys	291
Managing RHEL Subscriptions	292
Implementing a Playbook to Manage Software	297
Summary	301
Exam Preparation Tasks	302
Review All Key Topics	302

Memory Tables 302

Define Key Terms 302

Review Questions 302

End-of-Chapter Labs 303

Lab 12-1 303

Lab 12-2 303

Chapter 13 Managing Users 305

“Do I Know This Already?” Quiz 305

Foundation Topics 308

Using Ansible Modules to Manage Users and Groups 308

Modules Overview 308

Managing Users and Groups 308

Managing sudo 309

Managing SSH Connections 313

Understanding SSH Connection Management Requirements 313

Using the Lookup Plug-in 314

Setting Up SSH User Keys 314

Managing Encrypted Passwords 319

Understanding Encrypted Passwords 319

Generating Encrypted Passwords 319

Using an Alternative Approach 321

Managing Users Advanced Scenario Exercise 323

Summary 329

Exam Preparation Tasks 330

Review All Key Topics 330

Memory Tables 330

Define Key Terms 330

Review Questions 330

End-of-Chapter Lab 331

Lab 13-1 331

Chapter 14 Managing Services and the Boot Process 333

“Do I Know This Already?” Quiz 333

Foundation Topics 336

Managing Services	336
Managing Systemd Services	336
Managing cron Jobs	337
Managing at Jobs	338
Managing the Boot Process	341
Managing Systemd Targets	341
Rebooting Managed Hosts	342
Managing the Boot Process and Services Advanced Exercise	344
Summary	346
Exam Preparation Tasks	346
Review All Key Topics	346
Memory Tables	347
Define Key Terms	347
Review Questions	347
End-of-Chapter Lab	348
Lab 14-1	348
Chapter 15 Managing Storage	351
“Do I Know This Already?” Quiz	351
Foundation Topics	354
Discovering Storage-Related Facts	354
Using Storage-Related Facts	354
Using Storage-Related Facts in Conditional Statements	357
Managing Partitions and LVM	364
Creating Partitions	364
Managing Volume Groups and LVM Logical Volumes	366
Creating and Mounting File Systems	367
Configuring Swap Space	368
Configuring Storage Advanced Exercise	371
Summary	379
Exam Preparation Tasks	380
Review All Key Topics	380
Memory Tables	380
Define Key Terms	380

Review Questions 380

End-of-Chapter Lab 381

Lab 15-1 381

Chapter 16 Final Preparation 383

Generic Tips 383

Verifying Your Readiness 383

Registering for the Exam 384

On Exam Day 384

During the Exam 385

The Nondisclosure Agreement 386

Part IV: Practice Exams

Practice Exam A 387

Practice Exam B 391

Part V: Appendixes

Appendix A Answers to the “Do I Know This Already?” Quizzes and Review Questions 395

Appendix B Getting Started with Ansible Tower 415

Appendix C *Red Hat RHCE 8 (EX294) Cert Guide* Exam Updates 431

Glossary 433

Index 439

Online Elements:

Practice Exam C

Practice Exam D

Appendix D Memory Tables

Appendix E Memory Tables Answer Key

Appendix F Study Planner

Glossary

About the Author

Sander van Vugt has been teaching Linux classes since 1995 and has written more than 60 books about different Linux-related topics, including the best-selling *RHCSA-RHCE 7 Cert Guide* and the *RHCSA 8 Cert Guide: EX200*. Sander is also the author of more than 25 video courses, including his RHCSA and RHCE Complete Video Courses, *Hands-On Ansible LiveLessons*, and many other titles. He teaches courses for customers around the world and is also a regular speaker at major conferences related to open-source software. Sander is also the founder of the Living Open Source Foundation, a nonprofit organization that teaches open-source courses in African countries.

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 years, and this book is another milestone on our road to success! It has been fantastic how you both have helped me to realize this book in just two months!

Next, I want to thank my technical reviewers. Big thanks to Bo and John! Thanks to your great feedback, I’ve been able to apply important improvements to the contents of this book. Also, a special thanks to Ettiene Esterhuizen from New Zealand and Santos Venter Chibenga and Robert Charles Muchendu from the African Living Open Source Community, who helped me as volunteer reviewers. And last but not least, thanks to my fellow instructor and colleague Pascal van Dam, who helped me make some important last-minute improvements.

About the Technical Reviewers

John McDonough has more than 30 years of development experience; currently, John 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.

William “Bo” Rothwell crossed paths with a TRS-80 Micro Computer System (affectionately known as a “Trash 80”) at the impressionable age of 14. 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, and programming languages such as Perl, Python, Tcl, and BASH. He is the founder and president of One Course Source, an IT training organization.

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 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

Reader Services

Register your copy of *Red Hat RHCE 8 (EX294) Cert Guide* at www.pearsonitcertification.com for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.pearsonitcertification.com/register and log in or create an account.* Enter the product ISBN 9780136872436 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Introduction

Welcome to the *Red Hat RHCE 8 (EX294) Cert Guide*! With the release of Red Hat Enterprise Linux 8, Red Hat has decided to take a completely new direction for the RHCE exam. The exam is now completely about managing configurations with Ansible. This is a great choice because in the current IT landscape the days of the system administrator who applies specialized skills to tune individual servers is over. Today the work is all about automation, and Ansible has rapidly become one of the most important solutions to do so.

As a Linux instructor with more than 25 years of experience, I have been certified for both the RHCSA and RHCE exams for every RHEL version since RHEL 4. Taking the exams myself has helped me keep current on the progression of the exam, what is new, and what is different. I am thrilled to be able to share my knowledge with you in this comprehensive Cert Guide so you can get the guidance you need to pass your RHCE RHEL 8 EX294 exam.

As you will see, this Cert Guide covers every objective in the updated RHCE exam, with 16 chapters, more than 40 exercises, 4 practice exams (2 printed in the book and 2 on the companion website), and 1 hour of video training. This *Red Hat RHCE 8 (EX294) 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, I recommend that you create your own testing environment, which is explained in Chapter 2, “Installing Ansible.” You cannot become an RHCE without practicing a lot. To get familiar with the topics in the chapters, here is what I recommend:

- Read the explanation in the chapters and study the code examples that are provided in the listings. For your convenience, the listings are also provided in the book GitHub repository at <https://github.com/sandervanvugt/rhce8-book>. Study the examples and try to understand what they do.
- Walk through all of the numbered exercises in the book. The numbered exercises provide step-by-step instructions, and you should follow along with all of them, to walk through configuration tasks and learn how to manage specific features.
- At the end of each chapter, there’s an end-of-chapter lab. This lab is much like the lab assignments that you will find on the exam.

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. Although you may feel familiar with some topics, it's a good idea to work through all of the exercises in the book. The RHCE exam is hands-on, which can be a lot of pressure on test day. The exercises in each chapter help provide the practice you need to make sure you have the experience you need to not make small errors and mistakes while taking the exam. The exercises are the best way to make sure you work through common errors and learn from your mistakes before you take the test.

Each chapter also includes an end-of-chapter lab. These labs ask questions that are similar to the questions that you might encounter on the exam so you can use them to practice. I have purposely excluded solutions for these labs for a few reasons: (1) you need to train yourself to verify your work before test day because you will be expected to do this on the exam; (2) while taking the test, you will be required to verify for yourself whether your solution is working as expected; and (3) most labs have multiple solutions and I don't want to suggest that my solution is the right one and yours is wrong because it takes a different approach. Your solution is as good as mine, as long as it accomplishes what was asked for in the exercise.

Other Resources

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 Engineer (RHCE) 3/ed Complete Video Course*
- *Hands-on Ansible LiveLessons*

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: [rhatcert](https://www.youtube.com/rhatcert). 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 RHCE exam preparation guide. That means that you should read it if you want to increase your chances of passing the RHCE exam.

I have also written this book to help you become familiar with Ansible. So even if you're not interested in the RHCE EX294 exam at all, this book will teach you everything you need to know to get your Ansible career up and running.

So, why should you consider passing the RHCE exam? 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 RHCE certificate really helps. Having these certificates dramatically increases your chances of becoming hired as a Linux professional. Notice that in order to get RHCE certified, you must hold a current RHCSA certification. You can take the RHCE EX294 exam before you are RHCSA certified, but you can call yourself an RHCE only if you have passed both the RHCSA exam and the RHCE exam.

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. 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. These quizzes do not represent the types of questions you will get on the real exam though.

The best exam preparation is offered in the RHCE practice exams; these are an essential part of readying yourself for the real testing experience. You might 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. We have included two practice exams in the printed book. The book's companion website then includes two additional practice exams as well as flashcards created from the book's glossary so you can further test your knowledge and skills. You will also find one hour of video from my *Red Hat Certified Engineer (RHCE) 3/ed Complete Video Course*.

The following topics are covered in the chapters:

- **Chapter 1, “Understanding Configuration Management”:** In this chapter, you learn about Ansible as a solution. The chapter explains what can be done with Ansible and how Ansible relates to other solutions for configuration management.
- **Chapter 2, “Installing Ansible”:** This chapter covers installation of Ansible. You learn what is needed to set up the Ansible control node, as well as the other parts of the Ansible software.

- **Chapter 3, “Setting Up an Ansible Managed Environment”:** In this chapter you learn how to get started with node management. The chapter explains what is needed on the managed nodes as well as the essential Ansible configuration files that are required to reach out to the managed nodes.
- **Chapter 4, “Using Ad Hoc Commands”:** In this chapter you learn about Ansible modules. Modules are the heart of Ansible; they provide solutions for everything that Ansible can do, and the easiest way to use these modules is in ad hoc commands. In this chapter you learn how to work with them.
- **Chapter 5, “Getting Started with Playbooks”:** This chapter provides an introduction to working with playbooks. You learn about YAML, the language used to write playbooks, and how to structure a playbook using plays and tasks.
- **Chapter 6, “Working with Variables and Facts”:** In Ansible, variables can be used to provide dynamic values to specific configuration items. Using variables enables you to separate the static code in a playbook with host-specific information. In this chapter you learn how to work with variables as well as Ansible facts, which are variables that are automatically set for managed nodes.
- **Chapter 7, “Using Task Control”:** To make Ansible smart, you must apply task control. Using task control enables you to run tasks conditionally, and that can be done in many ways. You learn how to use tests, to test for a specific condition, as well as loops that allow you to evaluate a range of items, and handlers, which allow for task execution only if another task was executed successfully.
- **Chapter 8, “Deploying Files”:** Ansible is used for configuration management, and configuration on Linux is stored in files. Hence, managing files is a key skill in Ansible. In this chapter you learn how to use modules to modify files and how to use templates to automatically set up configuration files with specific parameters obtained from facts or variables.
- **Chapter 9, “Using Ansible Roles”:** When you are working with Ansible, it’s good if code can be reused. That is what Ansible roles are all about. In this chapter you learn how to work with roles, which are provided through Ansible Galaxy, or as RHEL system roles.
- **Chapter 10, “Using Ansible in Large Environments”:** When working with Ansible in large environments, you should know about a few specific techniques. These techniques are covered in this chapter. You learn how to optimize Ansible by modifying the number of concurrent tasks that can be executed. You also learn how to work with includes and imports, which allow you to set up modular playbooks.

- **Chapter 11, “Troubleshooting Ansible”:** In some cases your playbook might not give you the desired result. Then you need to start troubleshooting. This chapter contains not only all you need to know about troubleshooting, including some best practices while developing playbooks, but also information about modules that can be used to make troubleshooting easier.
- **Chapter 12, “Managing Software with Ansible”:** This is the first chapter about specific common tasks that you can perform with Ansible. In this chapter you learn how to set up repositories and how to manage software packages with Ansible.
- **Chapter 13, “Managing Users”:** To do anything on Linux, you need user accounts. In this chapter you learn all that is needed to create user accounts, including setting encrypted passwords.
- **Chapter 14, “Managing Services and the Boot Process”:** Occasionally, you might want to run scheduled jobs. These jobs will be executed at a specific time, using either cron or at. In this chapter you learn how to do that, and you also learn how to manage the systemd default target.
- **Chapter 15, “Managing Storage”:** Setting up storage is a key task when working with Linux. In this chapter you learn how to automate storage configuration with Ansible. You also learn how to discover disk devices available on your managed systems and how to set them up, using partitions, logical volumes, filesystems, and mounts.
- **Chapter 16, “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.

How to Use This Book

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 measure insight in the topics that were discussed in the chapter.
- **End-of-Chapter Labs:** These real labs 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 companion website, including the following:

- **Four practice exams:** The companion website contains the four practice exams: two provided in the book and two available on the companion website.
- **Flashcards:** The companion website contains interactive flashcards created from the glossary terms in the book so you can better learn key terms and test your knowledge.
- **More than one hour of video training:** The companion website contains more than one hour of video training from the best-selling *Red Hat Certified Engineer (RHCE) 3/ed Complete Video Course*.

Book Organization, Chapters, and Appendixes

I have also included a table that details where every objective in the RHCE exam is covered in this book so that you can more easily create a successful plan for passing the tests.

Table 1 RHCE Objectives

Objective	Chapter Title	Chapter	Page
Understand core components of Ansible: Inventories	Setting Up an Ansible Managed Environment	3	31
Understand core components of Ansible: Modules	Using Ad Hoc Commands	4	47
Understand core components of Ansible: Variables	Working with Variables and Facts	6	97
Understand core components of Ansible: Facts	Working with Variables and Facts	6	97
Understand core components of Ansible: Plays	Getting Started with Playbooks	5	69
Understand core components of Ansible: Playbooks	Getting Started with Playbooks	5	69
Understand core components of Ansible: Configuration files	Setting Up an Ansible Managed Environment	3	31
Understand core components of Ansible: Use provided documentation	Using Ad Hoc Commands	4	47
Install and configure an Ansible control node: Install required packages	Installing Ansible	2	15
Install and configure an Ansible control node: Create a static host inventory file	Setting Up an Ansible Managed Environment	3	31
Install and configure an Ansible control node: Create a configuration file	Setting Up an Ansible Managed Environment	3	31
Install and configure an Ansible control node: Create and use static inventories	Setting Up an Ansible Managed Environment	3	31
Install and configure an Ansible control node: Manage parallelism	Using Ansible in Large Environments	10	229
Configure Ansible managed nodes: Create and distribute SSH keys to managed nodes	Installing Ansible	2	15
Configure Ansible managed nodes: Configure privilege escalation on managed nodes	Installing Ansible	2	15

Objective	Chapter Title	Chapter	Page
Configure Ansible managed nodes: Validate a working configuration using ad hoc Ansible commands	Using Ad Hoc Commands	4	47
Script administration tasks: Create simple shell scripts	Using Ad Hoc Commands	4	47
Script administration tasks: Create simple shell scripts that run ad hoc Ansible commands	Using Ad Hoc Commands	4	47
Create Ansible plays and playbooks: Know how to work with commonly used Ansible modules	Using Ad Hoc Commands	4	47
Create Ansible plays and playbooks: Use variables to retrieve the results of running a command	Working with Variables and Facts	6	97
Create Ansible plays and playbooks: Use conditionals to control play execution	Using Task Control	7	131
Create Ansible plays and playbooks: Configure error handling	Using Task Control	7	131
Create Ansible plays and playbooks: Create playbooks to configure systems to a specified state	Getting Started with Playbooks	5	69
Use Ansible modules for system administration tasks that work with: Software packages and repositories	Managing Software with Ansible	12	281
Use Ansible modules for system administration tasks that work with: Services	Managing Processes and Tasks	14	333
Use Ansible modules for system administration tasks that work with: Firewall rules	Getting Started with Playbooks	5	69
Use Ansible modules for system administration tasks that work with: File systems	Managing Storage	15	351
Use Ansible modules for system administration tasks that work with: Storage devices	Managing Storage	15	351
Use Ansible modules for system administration tasks that work with: File content	Deploying Files	8	173
Use Ansible modules for system administration tasks that work with: Archiving	Managing Storage	15	351

Objective	Chapter Title	Chapter	Page
Use Ansible modules for system administration tasks that work with: Scheduled tasks	Managing Processes and Tasks	14	333
Use Ansible modules for system administration tasks that work with: Security	Managing Users	13	305
Use Ansible modules for system administration tasks that work with: Users and Groups	Managing Users	13	305
Work with roles: Create roles	Using Ansible Roles	9	205
Work with roles: Download roles from an Ansible Galaxy and use them	Using Ansible Roles	9	205
Use advanced Ansible features: Create and use templates to create customized configuration files	Deploying Files	8	173
Use advanced Ansible features: Use Ansible Vault in playbooks to protect sensitive data	Working with Variables and Facts	6	97

Where Are the Companion Content Files?

Register this print version of *Red Hat RHCE 8 (EX294) Cert Guide* to access the bonus content online.

This print version of this title comes with 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: **9780136872436**.
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, please visit www.pearsonITcertification.com/contact and select the Site Problems/Comments option. Our customer service representatives will assist you.

Credits

Chapter opener images by Charlie Edwards/Photodisc/Getty Images

Chapter 1 quote, “a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality,” © Len Bass, Ingo Weber, and Liming Zhu, *DevOps: A Software Architect’s Perspective*, Boston, MA: Addison-Wesley Professional, 2015.

Chapter 4 quote, “Your work will be evaluated by applying the playbooks created during the exam against freshly installed systems and verifying that those systems and services work as specified,” © 2020 Red Hat, Inc.

Cover image: Branislav Nenin/Shutterstock



This chapter covers the following subjects:

- Understanding Automation
- Understanding Ansible Essential Components
- Understanding Ansible Use Cases

Understanding Configuration Management

The following RHCE exam objectives are covered in this chapter:

- Understand Core Components of Ansible

“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 1-1 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 1-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Understanding Automation	1–5
Understanding Ansible Essential Components	6–9
Understanding Ansible Use Cases	10

1. Which of the following are disadvantages of using shell scripts for automation? (Choose two.)
 - a. They don’t work on any target managed operating system.
 - b. They require advanced skills.
 - c. It is difficult to guarantee they will always produce the same result if the configuration changes.
 - d. Using shell scripts makes sense only when they are used with root privileges.

2. In DevOps, the application life cycle is managed by focusing on different key aspects. Which of the following key aspects comes after releasing?
 - a. Testing
 - b. Configuring
 - c. Monitoring
 - d. Packaging
3. Which of the following are advantages of using a CVS to manage the machine-readable configuration files that are used in infrastructure as code?
 - a. It is easy to reproduce.
 - b. It makes upgrades easy.
 - c. It makes rollbacks easy.
 - d. All the above are true.
4. Ansible is an automation tool. Other automation solutions exist as well. Which of the following is not one of them?
 - a. Puppet
 - b. SaltStack
 - c. Satellite
 - d. Chef
5. When you compare Ansible to competing solutions such as Puppet, SaltStack, and Chef, Ansible offers two significant benefits. Which are these?
 - a. Speed
 - b. Easy configuration
 - c. Agentless operation
 - d. Price
6. Ansible provides different solutions to access remote hosts. Which of the following is not one of them?
 - a. Agent
 - b. API access
 - c. winRM
 - d. SSH

7. What is the name of the free open-source project that enables you to manage Ansible from a web interface?
 - a. Ansible Galaxy
 - b. OKD
 - c. AWX
 - d. Ansible Tower
8. Which of the following is the best description of the declarative approach?
 - a. In the declarative approach you run commands to get a specific result.
 - b. The declarative approach in Ansible is implemented by using scripts.
 - c. The declarative approach enables you to focus on changes that are required to reach the desired state.
 - d. The declarative approach uses playbooks.
9. Ansible can be used for provisioning. Which of the following is not a common provisioning scenario that Ansible is used for?
 - a. Deploying instances in cloud
 - b. Provisioning virtual machines
 - c. PXE-booting bare-metal servers
 - d. Deploying containers
10. Ansible can be used for different purposes. Which of the following can be considered the core function of Ansible?
 - a. Configuration management
 - b. Application management
 - c. Provisioning
 - d. Continuous delivery

Foundation Topics

Understanding Automation

Ansible is often referred to as a configuration management solution. That description doesn't do justice to all that Ansible can do. Ansible is more a solution for automation, allowing system administrators to take care of multiple systems in an efficient way. In this section you learn about all that Ansible can do as an automation tool. We also take a quick look at other automation solutions.

What Is Automation?

In the years of the system administrator, companies used servers. These servers performed a wide range of different tasks, and to ensure that every server was doing what it needed to be doing, a system administrator was needed. System administrators typically had advanced skills in managing different parts of the operating system that ran on their servers.

Even though the years of the system administrator were glorious, and many gurus worked in IT departments, from a company perspective, this scenario was not ideal. First, because system administrator skills are specific to that person, if that person goes away, forgets about brilliant solutions applied earlier, or just has a bad day, things might go wrong.

Another part that was not ideal was that the system administrator typically took care of individual servers, and with the development of IT in recent years, companies have gone from a handful of servers to data centers and cloud environments with hundreds if not thousands of servers. So a more efficient approach was needed.

A first attempt in many sites was the use of shell scripts. Based on the deep knowledge of many system administrators, shell scripts can be used in a flexible way to automate a wide range of tasks on many servers. Using shell scripts, however, does come with some disadvantages:

- Shell scripts cannot be used on a wide range of different devices that need management.
- It is difficult to write shell scripts in a way that will always produce the same result in every situation.

Because of these differences, and also because of changes in the way companies consume IT, a new approach was needed.

Understanding the DevOps Way of Working

Throughout the years the way IT is consumed has changed. In the past, IT was used to provide great services to end users who just had to deal with them. Now the landscape has changed to an environment in which IT is everywhere, and multiple applications can provide a solution to the same IT problem. The years of the system administrator slowly came to an end, and the system administrator's role needed to come closer to that of the developers.

In this new way of working, the developers take care of building applications, and system administrators take care of implementing the code as a working application. Because this change required a deep cooperation between the developer and the system administrator, a new role was created: the role of the DevOps. The term *DevOps* is a contraction of *developer* and *operator*. In this role, tasks performed by the developer and the system administrator come together. A common definition of DevOps is “a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality” (Len Bass, Ingo Weber, and Liming Zhu, *DevOps: A Software Architect's Perspective*, Boston, MA: Addison-Wesley Professional, 2015).

With this new role, the “DevOps way of working” was introduced. The exact definition is not always the same, but in general, it comes down to managing the entire application life cycle, which consists of the following elements:

Key Topic

- **Coding:** Developing and reviewing application source code
- **Building:** Using continuous integration to include changes in the source code and convert to a working application
- **Testing:** Using a toolchain that takes care of testing the application and making sure that feedback is provided on business risks, if there are any
- **Packaging:** Delivering the code to its end users by bundling it into packages and offering these packages in a repository
- **Releasing:** Approving, managing, and automating new software releases
- **Configuring:** Managing the infrastructure to support the new code
- **Monitoring:** Keeping an eye on application performance and the way it is experienced by the end users

To manage these different elements in the application life cycle, new tools were introduced. Ansible is one of these tools, with a strong focus on managing the configuration of the managed environment according to the infrastructure as code approach.

Some categories in the DevOps approach are more important than others. The most important elements are continuous integration, with solutions such as Jenkins and GitLab, but also OpenShift and even Ansible. The other main component is infrastructure as code, where Ansible, Puppet, and Terraform are important solutions.

Understanding Infrastructure as Code

The essence in infrastructure as code is that machine-readable code (the automation language) is used to describe the state the managed IT infrastructure needs to be in. This is referred to as the *desired state*. This code is next applied to the infrastructure to ensure that it actually is in that state.

In this approach, the machine-readable code files, which basically are simple text files, should be managed like software code, using a version control system, or Concurrent Version System (CVS). That means the tools that are common to the developer are implemented to manage the infrastructure as code. Commonly, Git repositories are used for this purpose.

Putting these files in a CVS makes managing it easy. This approach provides some benefits, such as easy management of change history, upgrades, and rollback. Infrastructure as code is the place where the developer meets the operator in DevOps. Developers can easily review changes, and operators can ensure that the systems are in the state that developers expect.

Other Automation Solutions

To provide automation of configuration management, Ansible is one of the most common solutions. Even if it seems to be currently the most-used configuration management solution, it's not the only one. Other common solutions include Puppet, Chef, and SaltStack.

Like Ansible, Puppet is one of the most important automation solutions. There are a few reasons why Ansible is taking over market share from Puppet though. One of the reasons is YAML. Ansible configurations are written in YAML, which is an easy-to-use and easy-to-understand language. Puppet uses its own language, which is just not as easy. Another major difference is that Ansible uses a push approach, where configurations are sent from the controller node to the managed nodes. Puppet uses a pull approach as its main strategy, where managed nodes use an agent to connect to the Puppet control node to fetch their desired state.

Chef is built as a client/server solution, where the server parts run on the control node machine and the client parts are implemented as an agent on the managed machines. Chef provides its configuration in Ruby DSL, whereas Ansible uses playbooks written in YAML. As a result, Ansible is easier to learn because YAML is a much more accessible data format.

SaltStack is another important alternative to Ansible. The main difference between Ansible and SaltStack is the performance. SaltStack uses the ZeroMQ message queue to realize communication between the SaltStack minions and the control node, and that seems to be faster. Saltstack can work with different languages for configuration files. It uses an agent, which makes implementing SaltStack more complex.

Understanding Ansible Essential Components

Now that you know a bit about Ansible and how it works, let's look at the different components used in Ansible. In this section you learn about the role of Python, the Ansible architecture, the Ansible Tower management platform, and how to manage systems the Ansible way.

Ansible Is Python

There are many programming and scripting languages in use in IT. In open source, the last few decades have seen the rise of the Python scripting language. Python has become the foundation of different solutions, such as Ansible and OpenStack. The reason is that Python is relatively easy to learn. The focus in Python is on readability of code, while at the same time Python makes it possible to do things in an easy way.

Ansible is written in Python, and most components that are used in Ansible are written in Python as well. The default Ansible version that is installed on Red Hat Enterprise Linux 7 is based on Python 2.7; the Ansible release that is used in RHEL 8 is based on Python 3.6. There is no direct relation between an Ansible version and a Python version. Recent versions of Ansible can call either Python 2.x or Python 3.x scripts, but Python 3.x is the better option nowadays because Python 2 is past its end of support life.

The fact that Ansible is written in Python makes it easier to integrate Ansible with custom scripts because Python is a very common and widely known scripting language. This doesn't mean you have to know Python to work with Ansible though. It's true that if you understand the workings of Python it's easier to explain specific behavior in Ansible, but it's perfectly possible to be an expert in Ansible without even knowing how to write a Hello World script in Python.

Ansible Architecture

There are two main node roles in Ansible. The controller node is the node that runs the Ansible software and from which the operator issues Ansible commands. The controller node can be a server running Linux, an operator laptop, or a system running Ansible Tower. The only requirement is that the controller node needs to be Linux.

From the controller node, the managed nodes are addressed. On the controller node, an inventory is maintained to know which managed nodes are available. Ansible doesn't require the use of any agents. That means it can reach out to managed nodes without a need to install anything. To do so, Ansible uses native remote access solutions that are provided by the managed node. On Linux, remote access is realized by using SSH; on Windows, it is realized by using Windows Remote Management (WinRM); and on network devices, it can be provided by using SSH or API access.

To configure the managed nodes, Ansible uses playbooks. A playbook is written in YAML and contains one or more plays. Each play consists of one or more tasks that are executed on the managed nodes.

To implement the tasks, Ansible uses modules. Modules are the pieces of code that do the actual work on the managed nodes, and many modules are available—more than 3000 already, and the number is increasing. Ansible also provides plug-ins. Ansible plug-ins are used to extend Ansible functionality with additional features.

Ansible playbooks should be developed to be idempotent. That means a playbook will always produce the same results, even if it is started multiple times on the same node. As a part of the idempotency, playbooks should also be self-containing and not depend on any other playbooks to be successful.

Understanding Ansible Tower

Ansible can be used in two different ways: Ansible Engine or Ansible Tower. Ansible Engine is the command-line version of Ansible, where modules and plug-ins are used to offer Ansible functionality. Ansible Engine is the solution of choice for people who like to work from the command line in a medium- to mid-sized environment.

Apart from Ansible Engine, there is Ansible Tower, which is based on the AWX open-source solution. It provides a web-based interface to manage Ansible. Ansible Tower adds different features to Ansible Engine, such as

Key Topic

- Web management interface
- Role-based access control
- Job scheduling
- Enhanced security
- Centralized logging

Because the RHCE EX294 exam is about Ansible Engine, you won't find much information about Ansible Tower in this book.

Understanding the Ansible Way

While working with Ansible, you need to make choices on how to approach specific tasks. In many cases, many solutions are available. If, however, you choose to work the Ansible way, making the right solution becomes a lot easier. The Ansible way is focused around the following rules:

Key Topic

- **Keep it simple:** At its launch, Ansible was positioned as a solution that is simpler than the others. That goes for the playbooks and other solutions you'll develop as well. Keep it simple, and it will be easier for others to understand what you had in mind.
- **Make it readable:** As with anything in IT, you can make it very complex and use compact structures to ensure that nobody understands what you were trying to do. That approach doesn't make sense. You should keep it readable, and that starts with your development of Ansible playbooks.
- **Use a declarative approach:** In Ansible, it's all about the desired state. The purpose of Ansible is to bring managed assets into the desired state, regardless of the current state, and make only the modifications that are necessary. The desired state is implemented in playbooks, and using playbooks to make the current state match the desired state is what is known as the declarative approach.
- **Use specific solutions:** On many occasions, you'll find that multiple solutions are available to reach a specific desired state. For instance, you can use the `command` module to run arbitrary commands, making it possible to accomplish almost anything. You shouldn't, though. To make sure that you get the desired result, use the most specific solution. So if, for instance, a `user` module allows you to create users, use that module and don't use the Linux `useradd` command with the `command` module.

Understanding Ansible Use Cases

The core of Ansible is configuration management. The Ansible modules and plug-ins cover a wide range of functions, which means that Ansible can be used for configuration management and beyond. Here are some common use cases.

Using Ansible for Configuration Management

Many people know Ansible only as a configuration management solution, and there's a reason for that. Ansible started as a solution for configuration management, and that is what it still is used for in most cases. In configuration management, Ansible is used to manage configuration files, install software, create users, and perform similar tasks to guarantee that the managed systems all are in the desired state.

Using Ansible for Provisioning

Another common Ansible usage scenario is to deploy and install systems (provisioning). Provisioning is particularly common in virtual and cloud environments, where in the end a new machine is just a configuration file that needs to be pushed to the managed machine and started from there. Ansible does not offer the functionality to PXE-boot and kickstart a bare-metal server but is used in combination with solutions that can take care of that as well. While exploring the different modules that are available, you'll notice that a wide range of modules is provided to work with Ansible in different cloud environments.

Using Ansible for Continuous Delivery

Continuous integration/continuous delivery (CI/CD) makes sure that source code can easily be developed and updated, and the results are easily provisioned as a new version of an application. Ansible cannot take care of the entire CI/CD procedure itself, but Ansible playbooks can play an important role in the CD part of the CI/CD pipeline.

Summary

In this chapter you learned about configuration management in general and the role of Ansible in configuration management solutions. You also read what makes Ansible unique and in which typical use cases Ansible is used. In the next chapter you'll learn how to build your own management infrastructure based on Ansible.

Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 16, “Final Preparation,” and the exam simulation questions on the companion website.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 1-2 lists a reference of these key topics and the page numbers on which each is found.



Table 1-2 Key Topics for Chapter 1

Key Topic Element	Description	Page Number
List	Application life cycle	7
List	Features Ansible Tower adds to Ansible Engine	10
List	Rules of the Ansible way	11

Define Key Terms

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

automation language, CI/CD, current state, CVS, declarative approach, desired state, DevOps, Git repository, idempotent, infrastructure as code, pipelines, provisioning, Python, version control system



This chapter covers the following subjects:

- Understanding an Ansible Environment
- Understanding Controller Host Requirements
- Understanding Installation Methods
- Configuring Managed Hosts
- Configuring the Ansible User

Installing Ansible

The following RHCE exam objectives are covered in this chapter:

- Install and configure an Ansible control node
 - Install required packages
- Configure Ansible managed nodes
 - Create and distribute SSH keys to managed nodes
 - Configure privilege escalation on managed nodes

“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 2-1 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 2-1 “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Understanding an Ansible Environment	1, 2
Understanding Controller Host Requirements	3, 5, 7
Understanding Installation Methods	6
Configuring Managed Hosts	4, 8, 9
Configuring the Ansible User	10

1. Which of the following are required on Ansible managed nodes? (Choose all that apply.)
 - a. An Ansible agent
 - b. Python
 - c. Root access
 - d. SSH access
2. Which statements about Python in an Ansible environment are true? (Choose two.)
 - a. Python needs to be installed only on the control node.
 - b. Python needs to be installed on the managed nodes.
 - c. Python 3 is the default version in RHEL 8.
 - d. While installing Ansible on CentOS 8, you can select which Python version you want to use.
3. While you are setting up an Ansible managed environment, there are a few requirements. Which of the following is not a mandatory requirement?
 - a. Set up a dedicated user account.
 - b. Configure SSH key-based login.
 - c. Install Python on managed hosts.
 - d. Install Python on the Ansible control node.
4. Which statement about sudo is not true?
 - a. On managed hosts, the Ansible user must be able to escalate permissions.
 - b. Setting up sudo on the control node is not required.
 - c. Setting up sudo is not required on all types of managed nodes.
 - d. Privilege escalation should be set up to run tasks as root.
5. To set up Ansible to learn for EX294, there are different requirements. Which of the following is not one of them?
 - a. RHEL or CentOS 8.x
 - b. 1 GB of RAM or more
 - c. 20 GB or more disk space
 - d. 1 CPU

6. Ansible can be installed in different ways. Which of the following is not one of them?
 - a. Use RHEL subscription manager to install from the RHEL Ansible repository.
 - b. Use EPEL on either RHEL or CentOS 8.
 - c. Use the python-pip installer.
 - d. Use the Java installer.
7. Executing tasks as a remote user can be secured in different ways. Which of the following is not one of them?
 - a. Protect the SSH private key with a passphrase.
 - b. Use password-based SSH login.
 - c. Disallow root execution through sudo.
 - d. Make password use mandatory while using sudo.
8. Which of the following is not a required setup task while installing Ansible on CentOS 8?
 - a. Add the EPEL repository.
 - b. Install Python 3.
 - c. Configure sudo.
 - d. Enable SSH access.
9. Which firewall ports need to be open on Ansible-managed hosts? (Choose all that apply.)
 - a. 22
 - b. 80
 - c. 443
 - d. 2022
10. After you use ssh-keygen to create the SSH public/private key pair, what is the next step?
 - a. Configure sudo access.
 - b. Use ssh-copy-id to copy the private key to the managed machines.
 - c. Copy the public key to the remote hosts.
 - d. Install Python 3 on the managed hosts.

Foundation Topics

In this chapter you learn how to set up Ansible. This process involves setting up not just the controller host, from which Ansible tasks will be executed on the managed hosts and devices, but also the managed hosts. Because this book was written with the RHCE EX294 in mind, we focus on setting up an environment to manage RHEL/CentOS 8 Linux hosts.

Understanding an Ansible Environment

Ansible is about managing a wide range of different device types. To do so, you need a host that plays the role of the manager. In this book, we call that the controller node. The controller node runs the Ansible software; the managed nodes run nothing in particular. The only requirement on managed nodes is that Ansible should be able to contact them. On Linux and many network devices, that means you need to run Secure Shell (SSH). On Windows that means you need to configure Remote Management or Secure Shell for Windows. To work with Ansible, you don't need to run or configure any agent on managed devices.

To set up this book's test environment, you need a minimum of three nodes. The requirements of these nodes are listed in Table 2-2. One of these nodes is used as the controller node; the other nodes are used as managed nodes. Notice that in this book I've chosen not to manage the controller node with Ansible also. The controller node is an isolated environment to reduce the risk of things going wrong and the controller node accidentally getting changed so that it no longer works.

Notice that the specific operating system version is not very important, which is why Table 2-2 mentions RHEL or CentOS 8.x. While writing this book, I installed on 8.1. By the time you read this, a newer version of the operating system will be available. Even if you are installing Ansible on RHEL or CentOS 8.8, it should not matter. The underlying operating system doesn't change how Ansible approaches things. What matters is the Ansible version that is used, and for RHCE EX294, that is Ansible 2.x—version 2.8 or later, to be even more specific.

Also note that for the node setup in this book, I used a virtual machine installation from the CentOS and Red Hat Enterprise Linux ISO files. It is also possible to use different environments for creating the setup described in this chapter, such as cloud instances for either RHEL or CentOS. While you work with Cloud instances, the operating system will be configured slightly differently, according to the cloud platform that is used. I do not try to cover all these differences in this chapter because there are just too many platforms to consider. Everything in this book is based on virtual machines because that is what you'll work with in the exam. For that reason I recommend you set up your test environment in a virtual machine and not in cloud.

TIP In a production environment, you probably want to manage the controller node with Ansible as well. You would just set up a set of playbooks that allows you to easily configure, update, and manage the controller node.

Table 2-2 Test Environment Node Requirements

Item	Requirement
Operating system	RHEL 8.x or CentOS 8.x
RAM	1 GB or more
Disk space	20 GB or more
Installation profile	Minimal installation

Exercise 2-1 Setting Up the Test Network

1. Set up a test network, consisting of three virtual machines, using the specifications mentioned in Table 2-2. Ensure that the following names are used:
 - control.example.com: the Ansible control machine
 - ansible1.example.com: the first managed node
 - ansible2.example.com: the second managed node
2. Provide fixed IP addresses in each of the virtual machines. Use IP addresses that match your current network environment. The IP addresses should allow your machines to connect to the Internet.
3. While installing, create a user account with the name `ansible`.
4. Edit the `/etc/hosts` file on each of the three virtual machines such that you can ping each of the other virtual machines. See Listing 2-1 for an example of the `/etc/hosts` contents.

Listing 2-1 Sample `/etc/hosts` Contents

```
127.0.0.1 localhost localhost.localdomain localhost4 \
localhost4.localdomain4
::1  localhost localhost.localdomain localhost6 \
localhost6.localdomain6
192.168.4.200 control.example.com control
192.168.4.201 ansible1.example.com ansible1
192.168.4.202 ansible2.example.com ansible2
```

Understanding Controller Host Requirements

To install Ansible on the controller node, you need a few items:

Key Topic

- Python 3.x
- An SSH client
- Access to an Ansible Repository
- A dedicated user account that is configured with SSH and sudo permissions on managed hosts

Ansible is written in Python, and as a result Python has to be installed on the Ansible controller node as well as the Ansible managed nodes. On older versions of RHEL, this was something you needed to do separately. In RHEL 8, a default Python stack is installed automatically, so you don't have to do anything about that anymore. Notice that on some public cloud instances, a Python stack is not installed by default, and you might still have to manually install it.

Python is needed because, for managing managed hosts, Ansible generates Python scripts. These Python scripts are next executed, using **ssh** on the remote host. Managed network devices are an exception though, as network devices don't typically run a Python stack. But managing network devices with Ansible is outside the scope of the RHCE EX294 exam anyway.

Currently, there are two major versions of Python. Python 3.x slowly has become the standard in recent years, but Python 2.x is still used on many sites. Transitioning from Python 2.x to Python 3.x is not always easy because the scripts are not compatible. On RHEL 8, Ansible uses Python 3 as the default stack though. And even if it is technically possible to use Python 2.x on RHEL 8, this is not something you have to master for the RHCE 8 exam. You shouldn't want to do this anyway. Python 2.x is old and becoming obsolete; Python 3.x is what matters.

Understanding Installation Methods

Ansible is open-source software, and as a result, Ansible can be installed on many platforms, each of which comes with its own installation method. In this book we cover installation of Ansible on either Red Hat Enterprise Linux 8 or CentOS 8. On both platforms you can install Ansible from the distribution repositories or using Python pip. In the following sections you read how to do this.

NOTE Use the instructions that correspond to the operating system and installation method that you want to use. So do not perform Exercises 2-2, 2-3, and 2-4 but pick the one that applies to your operating system platform.

Installing Ansible on RHEL 8

Obviously, to take the RHCE 8 exam, you run all assignments on Red Hat Enterprise Linux. Using Ansible on Red Hat Enterprise Linux does require you to have a valid license though. Free developer licenses can be requested from <https://developers.redhat.com>, and using such a license allows you to install RHEL 8 as well as Ansible for free. After you set up the base RHEL 8 environment, the rest of the setup procedure is easy; you just have to add a repository and can then install Ansible. Exercise 2-2 outlines the steps that need to be performed on a pre-installed RHEL 8 machine.

Exercise 2-2 Installing Ansible on RHEL 8

1. On the RHEL 8 control node, open a root shell and type **subscription-manager repos --list**. This shows you a list of currently configured repositories. You should see the standard RHEL 8 repositories.
2. Type **subscription-manager repos --enable=ansible-2-for-rhel-8-x86_64-rpms** to add the Ansible 2.x repository.
3. Use **yum install ansible** to install the Ansible software.
4. Use **ansible --version** to verify that the Ansible software has been installed.
5. Type **rpm -qa | grep python** to verify that Python 3 is also installed.

Installing Ansible on CentOS 8

If you are using CentOS 8, the Ansible software can be installed from the EPEL repository. The EPEL repository contains Extra Packages for Enterprise Linux. Ansible is among the EPEL packages. After enabling access to the EPEL repository, you can install Ansible using a simple **yum install -y ansible**. Exercise 2-3 guides you through this procedure.

Exercise 2-3 Installing Ansible on CentOS 8

1. On the CentOS 8 control node, open a root shell.
2. Type **yum install -y epel-release** to add the EPEL repository.
3. Type **yum install -y ansible** to install the Ansible software.
4. Type **ansible --version** to verify that the Ansible software has been installed.

Using python-pip to Install Ansible

Ansible is completely written in the scripting language Python, and Python includes its own package manager: python-pip. As a result, for software that is written in

Python, you can use either the package manager of your Linux repository or the `python-pip` software manager. Because Ansible is written in Python, this applies to Ansible as well.

The advantage of using `python-pip` is that it is an advanced method of managing software, often giving you access to the most recent version of the software before it is available in the repositories of your distribution. The disadvantage is that the software isn't controlled from your distribution, which makes it harder to manage software updates from a generic interface. In Exercise 2-4 you learn how to install Ansible from the `python-pip` installer.

Exercise 2-4 Using `python-pip` to Install Ansible

1. From a root shell on your CentOS or RHEL 8 control node, type **`yum install -y python3-pip`** to install the Python 3 `pip` installer.
2. Type **`alternatives --set python /usr/bin/python3`**. This creates a symbolic link with the name `python` that allows you to start the `python3` binary.
3. Exit the root shell, and open a shell as your Ansible user. From the ansible user shell, type **`pip3 install ansible --user`**.
4. Once the installation completes, type **`ansible --version`** to verify that Ansible has been installed.

Configuring Managed Hosts

Key Topic

Managed hosts in an Ansible environment don't need a lot. Opposed to what is needed in some other configuration management solutions, no agents need to be installed on the managed hosts. You only have to enable SSH access and ensure that Python is available on the managed hosts.

If you want to manage non-Linux nodes, notice that additional steps might be required. To manage Windows with Ansible, for instance, you must enable a Windows-native remote access mechanism, like Windows Remote Management (WinRM) or the Windows SSH daemon, and if you want to manage network devices, you might have to use device-specific requirements. This is not relevant for RHCE EX294 though, where you only have to manage Linux machines with Ansible. Exercise 2-5 guides you through the procedure of setting up Linux-based managed nodes.