



Quick answers to common problems

Chef Infrastructure Automation Cookbook

Second Edition

Over 80 recipes to automate your cloud and server infrastructure
with Chef and its associated toolset

Matthias Marschall

[PACKT]
PUBLISHING

Chef Infrastructure Automation Cookbook *Second Edition*

Over 80 recipes to automate your cloud and server infrastructure with Chef and its associated toolset

Matthias Marschall



BIRMINGHAM - MUMBAI

Chef Infrastructure Automation Cookbook

Second Edition

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: August 2013

Second edition: May 2015

Production reference: 1260515

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78528-794-7

www.packtpub.com

Credits

Author

Matthias Marschall

Project Coordinator

Izzat Contractor

Reviewers

Robert Curth

Kristian Hoffmann

Max Manders

Greg Swallow

Earl Waud

Proofreaders

Stephen Copestake

Safis Editing

Indexer

Priya Sane

Commissioning Editor

Ashwin Nair

Production Coordinator

Komal Ramchandani

Acquisition Editor

Vinay Argekar

Cover Work

Komal Ramchandani

Content Development Editor

Rohit Kumar Singh

Technical Editor

Naveenkumar Jain

Copy Editor

Adithi Shetty

About the Author

Matthias Marschall is a software engineer "Made in Germany" and the author of the *Chef Infrastructure Automation Cookbook* by Packt Publishing. His four children make sure that he feels comfortable and stays in control of chaotic situations. A lean and Agile engineering lead, he's passionate about continuous delivery, infrastructure automation, and all things DevOps.

In recent years, Matthias has helped build several web-based businesses, first with Java and then with Ruby on Rails. He quickly moved into system administration, writing his own configuration management tool before moving his whole infrastructure to Chef in its early days.

In 2008, he started a blog (<http://www.agileweboperations.com>) with Dan Ackerson. There, they shared their ideas about DevOps since the early days of the continually emerging movement. You can find him on Twitter at @mmarschall.

Matthias is the CTO of www.gutefrage.net GmbH that helps run Germany's biggest Q&A site among other high traffic sites. He holds a master's degree in computer science [Dipl.-Inf. (FH)] and teaches courses on Agile software development at the University of Augsburg.

When not writing or coding, Matthias enjoys drawing cartoons and playing Go. He lives near Munich, Germany.

Thanks go to my colleagues at gutefrage.net for all those valuable discussions.

I would also like to thank Adam Jacob, Joshua Timberman, and all the other great people at Chef, Inc. for your help with this book.

Special thanks go to my reviewers, Earl Waud, Greg Swallow, Max Manders, and Robert Curth, who made this book so much better.

About the Reviewers

Robert Curth is an engineer at gutefrage gruppe. In his current project, HELPSTER, Chef is used to automate the server setup. When Robert is not programming, he organizes company events and talks about how to live a good life on his blog at <http://rocu.de>.

I want to thank all the amazing authors of Chef cookbooks and tools. Chef has come a long way since the first edition of this book!

Thanks, Matthias, for updating this book. I love how elegant many of these recipes are. I hope you, dear reader, enjoy them as much as I did!

Kristian Hoffmann is the sort of twisted individual who likes hacking code (in Perl, if possible) and cars (the smaller and faster, the better) and solving problems that would lead other people to throw their hands up in despair. After some early experimentation with Linux (circa Slackware 3.0), his tech career started as a lowly tech at a local ISP. He went on to complete his bachelor's in computer science, marry a fellow technophile, and rise to the ranks of the president/CTO in his adolescent ISP hacking grounds. He now enjoys the most significant challenge of raising two hopelessly tech-bound children.

Max Manders is a recovering PHP developer and former sysadmin, who currently works as a systems developer and ops engineer helping to run the Operations Centre for Cloudreach, an Amazon Web Services Premier Consulting Partner. Max has put his past experiences and skills to good use to evangelize all things DevOps, working to master Ruby and advocating infrastructure-as-code as a Chef practitioner.

Max is a cofounder and organizer of Whisky Web, a Scottish conference for the web development and ops community. When he's not writing code or tinkering with the latest and greatest monitoring and operations tools, Max enjoys the odd whisky and playing jazz and funk trombone. He lives in Edinburgh with his wife, Jo, and their cats, Ziggy and Maggie.

It's been an absolute pleasure to have the opportunity to provide a technical review of this book. I hope you enjoy reading it as much as I did! Thank you, Jo, for putting up with my mutterings and ignorance while I tinkered with the code in this book. And thank you, Shona, for sharing the load at work, affording me time to get this done!

Greg Swallow has been wrangling with all sorts of computers in the Indianapolis area for 20 years now, for folks like IN.gov, Expedient Data Centers, Salesforce, and Indigo BioAutomation. When he's not playing digital plumber, you can catch him on the roads and trails of Indiana, either on his bike or in his running shoes.

He has also reviewed *VMware vSphere 5.x Datacenter Design Cookbook* by Hersey Cartwright, Packt Publishing.

I would also like to thank Packt Publishing for offering me the opportunity to review this and other books. It's been fun!

Earl Waud is a Virtualization Development Professional with more than 9 years of focused industry experience creating innovative solutions for hypervisor provisioning, management, and automation. He is an expert in aligning engineering strategy with organizational vision and goals and delivering highly scalable and user-friendly virtualization environments.

With more than 20 years of experience developing customer-facing and corporate IT software solutions, Earl has a proven track record of delivering high-caliber and on-time technology solutions that significantly impact business results.

Earl lives in San Diego, California. He is blessed with a beautiful wife, Patti, and three amazing daughters, Madison, Daniella, and Alexis.

Thank you, my wonderful family, for allowing me to spend some of our precious family time to review this book. I love you and appreciate you, and I know I am truly blessed that you are my family.

Currently, Earl is a senior systems engineer with Intuit Inc., a company that creates business and financial management solutions that simplify the business of life for small businesses, consumers, and accounting professionals.

Earl can be reached online at <http://sandiegoearl.com>.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- ▶ Fully searchable across every book published by Packt
- ▶ Copy and paste, print, and bookmark content
- ▶ On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	v
Chapter 1: Chef Infrastructure	1
Introduction	2
Using version control	2
Installing the Chef development kit on your workstation	5
Using the hosted Chef platform	7
Managing virtual machines with Vagrant	10
Creating and using cookbooks	14
Inspecting files on your Chef server with knife	16
Defining cookbook dependencies	19
Managing cookbook dependencies with Berkshelf	21
Downloading and integrating cookbooks as vendor branches into your Git repository	25
Using custom knife plugins	30
Deleting a node from the Chef server	32
Developing recipes with local mode	33
Using roles	36
Using environments	37
Freezing cookbooks	41
Running Chef client as a daemon	43
Using chef-shell	44
Chapter 2: Evaluating and Troubleshooting Cookbooks and Chef Runs	47
Introduction	48
Testing your Chef cookbooks	48
Flagging problems in your Chef cookbooks	50
Test-driven development for cookbooks using ChefSpec	53

Integration testing your Chef cookbooks with Test Kitchen	58
Showing affected nodes before uploading cookbooks	63
Overriding a node's run list to execute a single recipe	64
Using why-run mode to find out what a recipe might do	66
Debugging Chef client runs	68
Inspecting the results of your last Chef run	70
Raising and logging exceptions in recipes	72
Diff-ing cookbooks with knife	74
Using community exception and report handlers	76
Creating custom handlers	78
Chapter 3: Chef Language and Style	83
Introduction	84
Using community Chef style	84
Using attributes to dynamically configure recipes	86
Using templates	89
Mixing plain Ruby with Chef DSL	92
Installing Ruby gems and using them in recipes	95
Using libraries	96
Using definitions	99
Creating your own Lightweight Resource Providers (LWRP)	101
Extending community cookbooks by using application wrapper cookbooks	106
Creating custom Ohai plugins	108
Creating custom knife plugins	112
Chapter 4: Writing Better Cookbooks	117
Introduction	118
Setting the environment variables	118
Passing arguments to shell commands	120
Overriding attributes	122
Using search to find nodes	125
Using data bags	128
Using search to find data bag items	131
Using encrypted data bag items	132
Accessing data bag values from external scripts	135
Getting information about the environment	138
Writing cross-platform cookbooks	139
Finding the complete list of operating systems you can use in cookbooks	142
Making recipes idempotent by using conditional execution	145

Chapter 5: Working with Files and Packages	147
Introduction	147
Creating configuration files using templates	148
Using pure Ruby in templates for conditionals and iterations	150
Installing packages from a third-party repository	153
Installing software from source	157
Running a command when a file is updated	161
Distributing directory trees	163
Cleaning up old files	166
Distributing different files based on the target platform	168
Chapter 6: Users and Applications	171
Introduction	171
Creating users from data bags	172
Securing the Secure Shell Daemon (SSHD)	176
Enabling passwordless sudo	179
Managing NTP	182
Managing nginx	185
Creating nginx virtual hosts	189
Creating MySQL databases and users	193
Managing WordPress sites	197
Managing Ruby on Rails applications	200
Managing Varnish	205
Managing your local workstation	208
Chapter 7: Servers and Cloud Infrastructure	213
Introduction	214
Creating your infrastructure using Chef Provisioning	214
Creating cookbooks from a running system with Blueprint	217
Running the same command on many machines at once	220
Setting up SNMP for external monitoring services	222
Deploying a Nagios monitoring server	224
Building high-availability services using heartbeat	228
Using HAProxy to load-balance multiple web servers	233
Using custom bootstrap scripts	236
Managing firewalls with iptables	238
Managing fail2ban to ban malicious IP addresses	241
Managing Amazon EC2 instances	244
Loading your Chef infrastructure from a file with spiceweasel and knife	248
Index	251

Preface

Irrespective of whether you're a systems administrator or developer, if you're sick and tired of repetitive manual work and don't know whether you may dare to reboot your server, it's time for you to get your infrastructure automated.

This book has all the required recipes to configure, deploy, and scale your servers and applications, irrespective of whether you manage five servers, 5,000 servers, or 500,000 servers.

It is a collection of easy-to-follow, step-by-step recipes showing you how to solve real-world automation challenges. Learn techniques from the pros and make sure you get your infrastructure automation project right the first time.

This book takes you on a journey through the many facets of Chef. It teaches you simple techniques as well as full-fledged real-world solutions. By looking at easily digestible examples, you'll be able to grasp the main concepts of Chef, which you'll need to automate your own infrastructure. Instead of wasting time trying to get the existing community cookbooks running in your environment, you'll get ready-made code examples to get you started.

After describing how to use the basic Chef tools, the book shows you how to troubleshoot your work and explains the Chef language. Then, it shows you how to manage users, applications, and your whole Cloud infrastructure. The book concludes by providing you with additional, indispensable tools, and giving you an in-depth look into the Chef ecosystem.

Learn the techniques of the pros by walking through a host of step-by-step guides to solve your real-world infrastructure automation challenges.

What this book covers

Chapter 1, Chef Infrastructure, helps you to get started with Chef. It explains some key concepts, such as cookbooks, roles, and environments, and shows you how to use some basic tools like the Chef development kit (ChefDK), such as Git, knife, chef shell, Vagrant, and Berkshelf.

Chapter 2, Evaluating and Troubleshooting Cookbooks and Chef Runs, is all about getting your cookbooks right. It covers logging and debugging as well as the why run mode, and shows you how to develop your cookbooks totally test driven.

Chapter 3, Chef Language and Style, covers additional Chef concepts, such as attributes, templates, libraries, and even Light Weight Resource Providers. It shows you how to use plain old Ruby inside your recipes and ends with writing your own Ohai and knife plugins.

Chapter 4, Writing Better Cookbooks, shows you how to make your cookbooks more flexible. It covers ways to override attributes, use data bags and search, and to make your cookbooks idempotent. Writing cross-platform cookbooks is covered as well.

Chapter 5, Working with Files and Packages, covers powerful techniques to manage configuration files, and install and manage software packages. It shows you how to install software from source and how to manage whole directory trees.

Chapter 6, Users and Applications, shows you how to manage user accounts, securing SSH and configuring sudo. Then, it walks you through installing complete applications, such as nginx, MySQL, WordPress, Ruby on Rails, and Varnish. It ends by showing you how to manage your own OS X workstation with Chef.

Chapter 7, Servers and Cloud Infrastructure, deals with networking and applications spanning multiple servers. You'll learn how to create your whole infrastructure using Chef provisioning. Then it shows you how to set up high-availability services and load-balancers, and how to monitor your whole infrastructure with Nagios. Finally, it'll show you how to manage your Amazon EC2 Cloud with Chef.

What you need for this book

To run the examples in this book, you'll need a computer running OS X or Ubuntu Linux 14.04. The examples will use Sublime Text (<http://www.sublimetext.com/>) as the editor. Make sure you configured Sublime text command-line tool `subl` to follow along smoothly.

It helps if you have Ruby 2.1.0 with bundler (<http://bundler.io/>) installed on your system as well.

Who this book is for

This book is for system engineers and administrators who have a fundamental understanding of information management systems and infrastructure. It helps if you've already played around with Chef; however, this book covers all the important topics you will need to know. If you don't want to dig through a whole book before you can get started, this book is for you, as it features a set of independent recipes you can try out immediately.

Sections

In this book, you will find several headings that appear frequently (Getting ready, How to do it..., How it works..., There's more..., and See also).

To give clear instructions on how to complete a recipe, we use these sections as follows:

Getting ready

This section tells you what to expect in the recipe and describes how to set up any software or preliminary settings required for the recipe.

How to do it...

This section contains the steps required to follow the recipe.

How it works...

This section usually consists of a detailed explanation of what happened in the previous section.

There's more...

This section consists of additional information about the recipe in order to make the reader more knowledgeable about the recipe.

See also

This section provides helpful links to other useful information for the recipe.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "The omnibus installer will download Ruby and all required Ruby gems into `/opt/chef/embedded`."

A block of code is set as follows:

```
name "web_servers"
description "This role contains nodes, which act as web servers"
run_list "recipe[ntp]"
default_attributes 'ntp' => {
  'ntpdate' => {
    'disable' => true
  }
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
name "web_servers"
description "This role contains nodes, which act as web servers"
run_list "recipe[ntp]"
default_attributes 'ntp' => {
  'ntpdate' => {
    'disable' => true
  }
}
```

Any command-line input or output is written as follows:

```
mma@laptop:~/chef-repo $ knife role from file web_servers.rb
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Open `http://requestb.in` in your browser and click on **Create a RequestBin**."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail feedback@packtpub.com, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

1

Chef Infrastructure

"What made Manhattan Manhattan was the underground infrastructure, that engineering marvel."

Andrew Cuomo

A well-engineered infrastructure builds the basis for successful companies. In this chapter, we will see how to set up the infrastructure around Chef as the basis of your infrastructure as code. We'll cover the following recipes in this chapter:

- ▶ Using version control
- ▶ Installing the Chef development kit on your workstation
- ▶ Using the hosted Chef platform
- ▶ Managing virtual machines with Vagrant
- ▶ Creating and using cookbooks
- ▶ Inspecting files on your Chef server with knife
- ▶ Defining cookbook dependencies
- ▶ Managing cookbook dependencies with Berkshelf
- ▶ Downloading and integrating cookbooks as vendor branches into your Git repository
- ▶ Using custom knife plugins
- ▶ Deleting a node from the Chef server
- ▶ Developing recipes with local mode
- ▶ Using roles
- ▶ Using environments
- ▶ Freezing cookbooks
- ▶ Running Chef client as a daemon
- ▶ Using chef-shell

Introduction

This chapter will cover the basics of Chef, including common terminology, workflow practices, and various tools that work in accordance with Chef. We will explore version control using Git, walk through working with community cookbooks, and running those cookbooks on your own servers, so that you can configure them in the way you need them.

First, let's talk about some important terms used in the Chef universe.

A cookbook is a collection of all the components needed to change something on a server, such as installing MySQL, the most important one being recipes, which tell Chef which resources you want to configure on your host.

You need to deploy cookbooks to the nodes that you want to change. Chef offers multiple ways for this task. Most probably, you'll use a central **Chef server**. You can either run your own server or sign up for **hosted Chef**.

The Chef server is the central registry, where each node needs to be registered. The Chef server distributes the cookbooks you uploaded to it, to your nodes.

Knife is Chef's command-line tool to interact with the Chef server. You run it on your local workstation and use it to upload cookbooks and manage other aspects of Chef.

On your nodes, you need to install **Chef client**—the part that retrieves the cookbooks from the Chef server and executes them on the node.

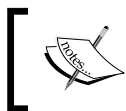
In this chapter, we'll see the basic infrastructure components of your Chef setup at work and learn how to use the basic tools. Let's get started by taking a look at how to use Git as a version control system for your cookbooks.

Using version control

Do you manually back up every file before you change it? And do you invent creative file name extensions such as `_me` and `_you` when you try to collaborate a file? If you answer yes to any of these, it's time to rethink your processes.

A **version control system (VCS)** helps you stay sane when dealing with important files and collaborating with them.

Using version control is a fundamental part of any infrastructure automation. There are multiple solutions (some free, some paid) to manage source version control, including Git, SVN, Mercurial, and Perforce. Due to its popularity among the Chef community, we will be using Git. However, you could easily use any other version control system with Chef.



Don't even think about building your infrastructure as code without using a version control system to manage it!

Getting ready

You'll need Git installed on your local workstation. Either use your operating system's package manager (such as Apt on Ubuntu or Homebrew on OS X), or simply download the installer from www.git-scm.org.

Git is a distributed version control system. This means that you don't necessarily need a central host to store your repositories. However, in practice, using GitHub as your central repository has proven to be very helpful. In this book, I'll assume that you're using GitHub. Therefore, you need to go to www.github.com and create an (free) account to follow the instructions given in this book. Make sure that you upload your Secure Shell (SSH) key by following the instructions at <https://help.github.com/articles/generating-ssh-keys>, so that you're able to use the SSH protocol to interact with your GitHub account.

As soon as you have created your GitHub account, you should create your repository by visiting <https://github.com> while you're still logged in and using `chef-repo` as the repository name.

Make sure you have `wget` installed on your local workstation, in order to be able to download the required files from public servers.

How to do it...

Before you can write any cookbooks, you need to set up your initial Git repository on your development box. Chef Software, Inc. provides an empty Chef repository to get you started. Let's see how you can set up your own Chef repository with Git, using Chef's skeleton.

1. Download Chef's skeleton repository as a tarball:

```
mma@laptop $ wget http://github.com/chef/chef-repo/tarball/master
...TRUNCATED OUTPUT...
2014-11-30 22:00:43 (1.30 MB/s) - 'master' saved [9309/9309]
```

2. Extract the downloaded tarball:

```
mma@laptop $ tar xzvf master
```

3. Rename the directory:

```
mma@laptop:~ $ mv opscience-chef-repo-* chef-repo
```

4. Change to your newly created Chef repository:

```
mma@laptop:~ $ cd chef-repo/
```
5. Initialize a fresh Git repository:

```
mma@laptop:~/chef-repo $ git init .
```

Initialized empty Git repository in /Users/mma/work/chef-repo/.git/
6. Connect your local repository to your remote repository on github.com. Make sure to replace mmarschall with your own GitHub username:

```
mma@laptop:~/chef-repo $ git remote add origin git@github.com:mmarschall/chef-repo.git
```
7. Configure Git with your user name and e-mail address:

```
mma@laptop:~/chef-repo $ git config --global user.email "you@example.com"
```

```
mma@laptop:~/chef-repo $ git config --global user.name "Your Name"
```
8. Add and commit Chef's default directory structure:

```
mma@laptop:~/chef-repo $ git add .
```

```
mma@laptop:~/chef-repo $ git commit -m "initial commit"
```

```
[master (root-commit) 6148b20] initial commit
 11 files changed, 545 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
...TRUNCATED OUTPUT...
 create mode 100644 roles/README.md
```
9. Push your initialized repository to GitHub. This makes it available to all your co-workers to collaborate on:

```
mma@laptop:~/chef-repo $ git push -u origin master
```

```
...TRUNCATED OUTPUT...
To git@github.com:mmarschall/chef-repo.git
 * [new branch]      master -> master
```

How it works...

You have downloaded a tarball containing Chef's skeleton repository. Then, you initialized chef-repo and connected it to your own repository on GitHub.

After that, you added all the files from the tarball to your repository and committed them. This makes Git track your files and the changes you make later.

Finally, you pushed your repository to GitHub, so that your co-workers can use your code too.

There's more...

Let's assume you're working on the same `chef-repo` repository, together with your co-workers. They cloned your repository, added a new cookbook called `other_cookbook`, committed their changes locally, and pushed their changes back to GitHub. Now, it's time for you to get the new cookbook downloaded on to your own laptop.

Pull your co-workers' changes from GitHub. This will merge their changes into your local copy of the repository. Use the `pull` subcommand:

```
mma@laptop:~/chef-repo $ git pull --rebase

From github.com:mmarschall/chef-repo
* branch            master      -> FETCH_HEAD
...TRUNCATED OUTPUT...
create mode 100644 cookbooks/other_cookbook/recipes/default.rb
```

In case of any conflicting changes, Git will help you merge and resolve them.

See also

- ▶ Learn about Git basics at <http://git-scm.com/videos>
- ▶ Walk through the basic steps using GitHub at <https://help.github.com/categories/54/articles>
- ▶ You'll use more Git features in the *Downloading and integrating cookbooks as vendor branches into your Git repository* recipe in this chapter

Installing the Chef development kit on your workstation

If you want to use Chef, you'll need to install the **Chef development kit (DK)** on your local workstation first. You'll have to develop your configurations locally and use Chef to distribute them to your Chef server.

Chef provides a fully packaged version, which does not have any external prerequisites. This fully packaged Chef is called the **omnibus installer**. We'll see how to use it in this section.

How to do it...

Let's see how to install the Chef DK on your local workstation using Chef's omnibus installer:

1. Download the Chef DK for your specific workstation platform from <https://downloads.chef.io/chef-dk/> and run the installer.
2. Verify that Chef installed all the required components:

```
mma@laptop:~ $ chef verify
...TRUNCATED OUTPUT...
Verification of component 'rubocop' succeeded.
Verification of component 'kitchen-vagrant' succeeded.
Verification of component 'chefspec' succeeded.
Verification of component 'berkshelf' succeeded.
Verification of component 'fauxhai' succeeded.
Verification of component 'test-kitchen' succeeded.
Verification of component 'package installation' succeeded.
Verification of component 'chef-dk' succeeded.
Verification of component 'knife-spork' succeeded.
Verification of component 'chef-client' succeeded.
```

3. Add the newly installed Ruby to your path:

```
mma@laptop:~ $ echo 'export PATH="/opt/chefdk/embedded/bin:$PATH"'
>> ~/.bash_profile && source ~/.bash_profile
```



You may not want to use (and don't have to use) ChefDK's Ruby, especially if you are a Rails Developer.

If you're happily using your Ruby `rvm`, or `rbenv` environment, you can continue to do so. Just ensure that the ChefDK-provided applications appear first in your `PATH`, before any `gem`-installed versions, and you're good to go.

How it works...

The omnibus installer will download Ruby and all required Ruby gems into `/opt/chefdk`.

See also

- ▶ Find detailed instructions for OS X, Linux, and Windows at <https://learn.chef.io>
- ▶ Find ChefDK on GitHub at <https://github.com/opscode/chef-dk>

Using the hosted Chef platform

If you want to get started with Chef right away (without the need to install your own Chef server) or want a third party to give you a **Service Level Agreement (SLA)** for your Chef server, you can sign up for hosted Chef by Chef Software, Inc. Chef Software, Inc. operates Chef as a cloud service. It's quick to set up and gives you full control, using users and groups to control the access permissions to your Chef setup. We'll configure knife, Chef's command-line tool to interact with hosted Chef, so that you can start managing your nodes.

Getting ready

Before being able to use hosted Chef, you need to sign up for the service. There is a free account for up to five nodes.

Visit <http://manage.chef.io/signup> and register for a free trial or a free account.

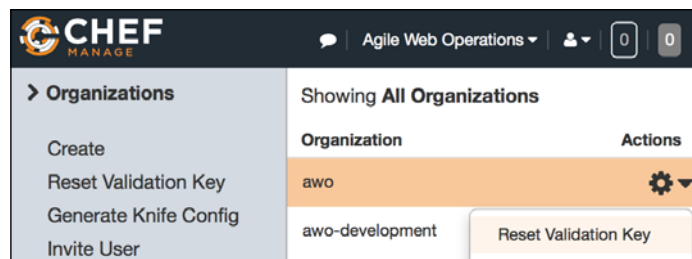
I registered as the user `webops` with an organization short name of `awo`.

After registering your account, it is time now to prepare your organization to be used with your `chef-repo` repository.

How to do it...

Carry out the following steps in order to interact with the hosted Chef:

1. Create the configuration directory for your Chef client on your local workstation:
`mma@laptop:~/chef-repo $ mkdir .chef`
2. Navigate to <http://manage.chef.io/organizations>. After logging in, you can start downloading your validation keys and configuration file.
3. Select your organization to be able to see its contents using the web UI.



Regenerate the validation key for your organization and save it as `<your-organization-short-name>-validator.pem` in the `chef` directory inside your `chef-repo` repository.



4. Generate the knife config and put the downloaded `knife.rb` into the `.chef` directory inside your `chef-repo` directory, as well. Make sure you have downloaded your user's private key from <https://www.chef.io/account/password> and replace `webops` with the username you chose for hosted Chef, and `awo` with the short name you chose for your organization:

```
current_dir = File.dirname(__FILE__)
log_level          :info
log_location       STDOUT
node_name          "webops"
client_key          "#{current_dir}/webops.pem"
validation_client_name "awo-validator"
validation_key      "#{current_dir}/awo-validator.pem"
chef_server_url     "https://api.chef.io/organizations/awo"
cache_type         'BasicFile'
cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )
cookbook_path       ["#{current_dir}/../cookbooks"]
```

[

Take a look at the following code:

```
.chef/*.pem
.chef/encrypted_data_bag_secret
```

You should add the preceding code to your `.gitignore` file inside `chef-repo` to avoid your credentials from ending up in your Git repository.

]

5. Use knife to verify that you can connect to your hosted Chef organization. It should only have your validator client, so far. Instead of `awo`, you'll see your organization's short name:

```
mma@laptop:~/chef-repo $ knife client list
awo-validator
```

How it works...

Hosted Chef uses two private keys (called validators):

- ▶ one for the organization
- ▶ one for every user.

You need to tell knife where it can find these two keys in your `knife.rb` file.

The following two lines of code in your `knife.rb` file tell the knife about which organization to use and where to find its private key. The `validation_key` is used to allow new clients to authenticate the Chef server before getting their own Client key:

```
validation_client_name  "awo-validator"
validation_key           "#{current_dir}/awo-validator.pem"
```

The following line of code in your `knife.rb` file tells the knife where to find your users' private key. It is used by your local workstation to authenticate the Chef server:

```
client_key              "#{current_dir}/webops.pem"
```

Also, the following line of code in your `knife.rb` file tells knife that you are using hosted Chef. You will find your organization name as the last part of the URL:

```
chef_server_url         "https://api.chef.io/organizations/awo"
```

Using the `knife.rb` file and your two validators knife, you can now connect to your organization hosted by Chef Software, Inc.

You do not need your own self-hosted Chef server, nor do you need to use Chef client local mode in this setup.

There's more...

This setup is good for you if you do not want to worry about running, scaling, and updating your own Chef server and if you're happy with saving all your configuration data in the Cloud (under the control of Chef Software, Inc.).



If you need to have all your configuration data within your own network boundaries, you can install Chef server on premises by choosing "ON PREMISES CHEF" at <https://www.chef.io/chef/choose-your-version/> or install the Open Source version of Chef server directly from GitHub at <https://github.com/chef/chef>.