

O'REILLY®



# Learning Chef

---

A GUIDE TO CONFIGURATION MANAGEMENT AND AUTOMATION

Mischa Taylor & Seth Vargo



# **Learning Chef**

---

**Mischa Taylor**

**Seth Vargo**



# Preface

---

# What Is This Book?

---

*Learning Chef* is an introductory book on the Chef infrastructure automation platform. This is a book for beginners who are new to Chef, configuration management, and automation coding.

Using Chef, you can model the setup, packaging, and delivery of applications in your infrastructure as code. We'll show you how using code makes actions easily repeatable, while running commands by hand is not. Once you have this code blueprint, you can then build or rebuild your whole infrastructure's application stack in minutes or hours, instead of the days or weeks it would typically take doing by hand. In this book, we'll cover the basics of Chef, assuming you have no prior experience with infrastructure automation or coding.

First, we'll get started by showing you how to set up a Chef development environment on your own local machine. This is where you will write Chef code and verify that it works. Learning how to code takes a lot of hands-on practice, so we try to get you to dive in and start writing code very early in the book. Then we progress slowly, introducing one new concept along with its accompanying Chef tool in each chapter. Hands-on examples are provided to help cement the concepts in your mind and to give you practice coding.

After you've read *Learning Chef*, you will understand all the basic concepts and be ready to pick up and benefit from two more advanced books on Chef, such as the O'Reilly books *Test-Driven Infrastructure with Chef, 2nd Edition*, by Stephen Nelson-Smith, and *Customizing Chef*, by Jon Cowi

# Who Should Read This Book?

---

We wrote this book for both system administrators and software developers new to Chef and the concept of infrastructure automation. It is assumed that you have some familiarity with using the command line and performing basic system administration tasks.

You may run Linux, Mac OS X, or Windows on your local machine and follow the hands-on exercises in this book. When necessary, we'll provide separate instructions for each platform. Because there is currently no easy-to-use graphical integrated development environment (IDE) for Chef, your primary interface to Chef will be through the command line. Thankfully, the command line interface has very few platform-specific differences, except with the initial installation, so the choice to cover three operating system platforms in this book shouldn't be too distracting while broadening your choice of environments in which to work with Chef.

We do not assume that you have any experience with automation coding, but we do assume that you've written scripts before, such as shell scripts, batch files, or PowerShell scripts. You should be familiar with scripting in some form before trying to learn Chef coding.

## Why All the Culinary Terminology?

---

As you read this book, you might notice that the makers of Chef are fond of using culinary terms to describe infrastructure automation tools and concepts. The people at Chef Software discovered that words and phrases such as *cookbook* or *following a recipe* are good metaphors for abstract concepts such as *a collection of automation scripts* or *running code that consistently reproduces an infrastructure component*. The consistent use of cooking metaphors makes the topic of infrastructure automation more accessible to beginners and does not have the baggage of industry terms used in other contexts. Plus, it's a good way to market and brand Chef, as the cooking angle is unique and makes Chef memorable.



# Conventions Used in This Book

---

The following typographical conventions are used in this book:

## *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

## Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keyword

## Constant width bold

Shows commands or other text that should be typed by the user.

## *Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.

### **TIP**

This element signifies a tip or suggestion.

### **NOTE**

This element signifies a general note.

### **WARNING**

This element indicates a warning or caution.

# Using Code Examples

---

Supplemental material (code examples, exercises, etc.) is available for download at <http://learningchef.com>.

This book is here to help you get your job done. In general, if example code is offered with this book you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Learning Chef* by Mischa Taylor and Seth Vargo (O'Reilly). Copyright 2015 Mischa Taylor and Seth Vargo, 978-1-491-94493-6."

If you believe your use of code examples falls outside fair use or the permission given, feel free to contact us at [permissions@oreilly.com](mailto:permissions@oreilly.com).

## NOTE

Safari Books Online is an on-demand digital library that delivers expert **content** in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of **plans and pricing** for **enterprise**, **government**, **education**, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds **more**. For more information about Safari Books Online, please visit us **online**.

# How to Contact Us

---

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)

707-829-0515 (international or local)

707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at [http://bit.ly/learning\\_chef](http://bit.ly/learning_chef).

To comment or ask technical questions about this book, send email to [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com).

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

# Acknowledgments

---

*From Mischa:* I would like to thank all the people who provided input and feedback on our book during the writing process. Thanks to my sister, Dr. Jane Maris Sinagub, who provided moral support and daily encouragement.

I want to give heartfelt thanks to all the people who helped contribute to this book, including but not limited to: Alex Vinyar, Alyssa Nabors, Anthony Stonebarger, Daniel DeLeo, Deluan Quintao, Eric Helgeson, Gene Harris, Glenna Gorlick, Jason Steele, Jennifer Davis, Jo Rhett, John Keiser, Jennifer Davis, John Fitzpatrick, Jon Cowie, Julian Dunn, Katherine Daniels, Kelly Setzer, Kimberly Lanning, Landon Medlock, Lejo Varughese, Mandi Walls, Michael Goetz, Michael Vitale, Nathen Harvey, Patricia Fernandes, Rhiannon Portwood, Sascha Bates, Sean Carolan, Serdar Sutay, Shane Robinson, Steve Taylor, and Thomas Petchel. Thanks to Chef Software for making Chef Fundamentals Training materials Creative Commons Attribution Share Alike licensed. Some of the diagrams and examples were reused in this book. Thanks as well to Courtney Nash, Brian Anderson, and all the marvelous people at O'Reilly. Your amazing Atlas authoring system made the writing process more delightful.

Special thanks go out to the following people: Mark Burgess for developing the theory to make Chef possible, John Keiser for providing the “written in stone” example, Jennifer Davis for providing valuable input on how to structure the initial chapters in the book, John Fitzpatrick for testing the installation examples and providing feedback, Nathen Harvey for helping with the introductory material, Sascha Bates for providing clear guidance on what beginners should first learn about Chef to be effective, Adam Jacob and Sean O'Meara for educating me on the theory behind the configuration management philosophy from which Chef was born, and Steve Taylor and Mark Andersen for providing training cohorts on which to try this book's material.

And finally I would like to thank my coauthor Seth Vargo, for being so gracious in allowing me to come on board to help finish the book.

*From Seth:* I would like to thank everyone who helped make this book possible. I would be remiss if I did not thank Nathen Harvey, Ramez Mourad, and Jake Vanderdray for introducing me to Chef and the Chef community. Without their knowledge and encouragement, I would not be here today.

In addition to everyone Mischa mentioned, I would like to extend my gratitude to my team, Seth Chisamore and Yvonne Lam, for their continued support throughout this endeavor; my roommate and best friend Joe Frick for supporting me as I quietly mumbled things at my laptop screen at all hours of the night; Prof. Larry Heimann and Prof. Jeria Quesenberry from Carnegie Mellon for giving me the experience and opportunity to be an educator; Stafford Brunk for forcing me to think outside of the box; and my parents Richard Stormer, Robbin Stormer, Robert Vargo, and Dara Vargo for being incredibly supportive while continually reminding me to finish this book.

Thanks as well to Courtney, Brian, Sonia, and the entire O'Reilly crew for making this an absolutely amazing experience. I would like to extend a special thank you to my coauthor Mischa Taylor for his amazing contributions and diligence in finishing this book. I could not have done it without you!



# Chapter 1. Configuration Management and Chef

Chef is a configuration management tool for information technology (IT) professionals, like you. Because there are a wide variety of definitions for the term *configuration management*, let's take a moment to explain what *configuration management* means in the context of this book and why you need a configuration management tool. We'll also cover what Chef is, and why you need it as well.

# What Is Configuration Management?

---

With respect to IT, *configuration management* covers the set of engineering practices for managing the following entities involved in delivering software applications to consumers:

- Hardware
- Software
- Infrastructure
- People
- Process

Configuration management came about to address the fundamental challenges involved in doing group work. Managing change when you are a lone system administrator with a handful of servers to manage is relatively straightforward. Trying to coordinate the work of multiple system administrators and developers involving hundreds, or even thousands, of servers and applications to support a large customer base is complex and typically requires the support of a tool.

A modern IT configuration management tool usually involves an implementation inspired by the automation and policy-based theory originally developed by Mark Burgess. He developed the following core ideas of this theory for automating IT when he was a professor at Oslo University College in the late 1990s and early 2000s:

- Changes must be handled in a systematic fashion to ensure that a system is configured in a correct and reliable manner.
- There must be some form of autonomy in the system so that it can automatically detect faults and repair them without being explicitly told to do so.

Examples of modern IT configuration management tools are CFEngine, Puppet, the Desired State Configuration engine in Microsoft Windows, Ansible, SaltStack, and of course, Chef.



# Why You Need a Configuration Management Tool to Automate IT

There are a number of reasons why automated configuration management tools play a vital role in managing complex enterprise infrastructures. Here are four of the most popular reasons:

- **Consistency.** If your infrastructure is being configured manually, how do you know your servers are being set up in a consistent manner? Further, how do you know these changes are being performed in a way that meets your compliance and security requirements? (For instance, are administrators logging changes in the appropriate systems?)  
Make life easier for your system administrators by automating repeated tasks with a configuration management tool. When repeated tasks are tedious, humans are alarmingly bad at performing them consistently. Automate tedious administration tasks with a configuration management tool so your staff can focus on other important things that humans do best.
- **Efficient change management.** Whenever infrastructure is built manually without the aid of a configuration management tool, people tend to fear change. Over time, servers that are maintained by hand tend to become fragile environments that are hard to understand and modify.  
In these situations, organizations tend to develop a lot of processes for managing changes, usually with the sole intent on minimizing change or even delaying it as long as possible. This tends to delay introducing new features your customers need.  
When servers can be reproduced easily in a repeatable fashion, fewer processes are needed to manage change. Small change batches can be performed on a regular basis, such as daily, or even several times a day.
- **Simplicity in rebuild.** When servers are built manually, it's typically not easy to rebuild them from scratch. What would happen if you suddenly lost your servers in a catastrophic event? How quickly could you restore service if disaster struck?  
Automated deployments using a configuration management tool help quickly restore service. Rather than bothering to upgrade or patch applications, which can be inherently fragile operations, system administrators can build a new, upgraded system in an automated fashion and throw the old one away, returning it to the server pool. When rebuilds are easy, system administrators gain confidence to make changes to infrastructure more rapidly.
- **Visibility.** Configuration management tools include auditing and reporting capabilities. Monitoring the work performed by one system administrator doesn't require a sophisticated tool. But trying to understand what is going on with a team of, say, 10 system administrators and 10 software developers deploying software changes many times per day? You need a configuration tool.  
When infrastructure changes are handled by automated systems, changes can be automatically logged in all relevant tracking systems to raise visibility on the meaningful work your teams are doing.

# What Is Chef?

---

Chef is an automation platform that configures and manages your infrastructure whether it is on-premises or in the cloud. You can deploy to the infrastructure type that makes the most sense for your business. You can use Chef to speed up application deployment, even creating a continual deployment pipeline. The key to Chef's power is that it turns infrastructure into code.

Infrastructure as code means that your computing environment has some of the same attributes as your application:

- Your infrastructure is versionable.
- Your infrastructure is repeatable.
- Your infrastructure is testable.

**Figure 1-1** presents an overview of the major components of Chef.

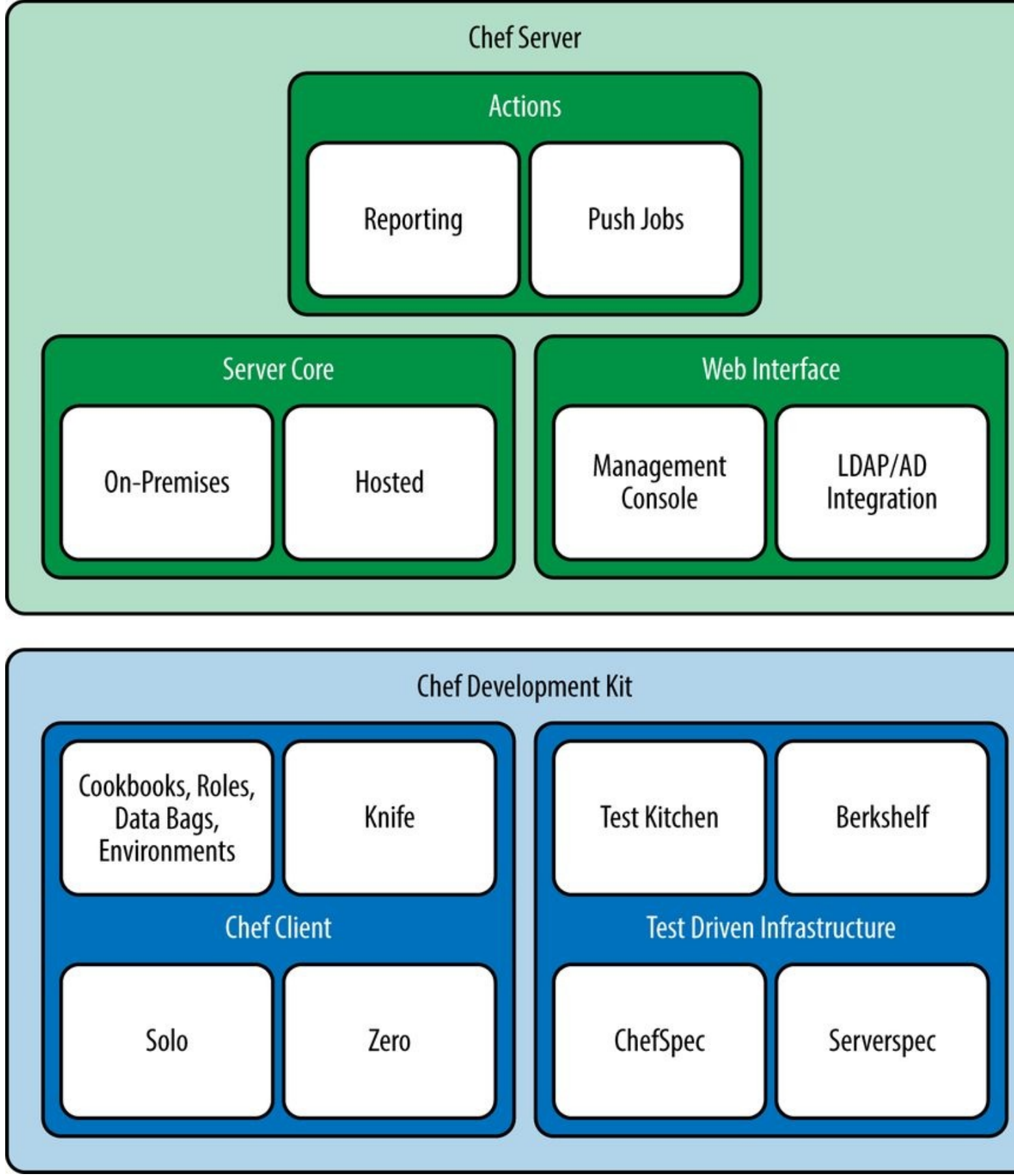


Figure 1-1. Chef architecture

The components of the Chef Development Kit help support you as you write Chef code on your development workstation. Chef Server provides additional components to help scale your configuration management capabilities to hundreds or thousands of servers, and beyond.

# Why Chef Might Be a Good Tool for Your Enterprise

---

When Adam Jacob first created Chef in 2009, he had three key insights to address the shortcomings he saw in other configuration management tools:

1. A configuration management tool should easily enable web IT, providing first-class support for managing cloud infrastructure.
2. Everyone's infrastructure is unique. Complex, enterprise infrastructures benefit greatly from being able to model their IT infrastructure and application delivery process as code.
3. Great tools and ideas also come out of a vibrant and involved user community. You can't do it alone.

Over time with community support, Chef has evolved to have a powerful set of features that make it unique among configuration management tools:

## Extreme scalability

Customers such as Facebook **use Chef to manage tens of thousands of servers** using only a handful of employees.

## Power

Chef is built on top of the Ruby programming language. When you need it, you have full access to the power of Ruby to customize Chef.

## Choice

You are not locked into one way of using Chef. Chef can operate in a distributed standalone mode or in a centralized mode requiring a server. There are also options to use pull or push models (or both) for deployment.

## Open

Chef is open source and supported by a vibrant community of system administrators and developers. Chef open source has been used to power **products from Dell, from Facebook, and from Amazon Web Services.**

## Visible

As a premium feature, the Chef Analytics Platform provides powerful enhancements to Chef integrated with the tool, so that you can be notified when important changes are made as a way to enforce compliance.

Chef's unique approach gives you tremendous flexibility. You don't need to struggle to conform to Chef. Chef adapts to you and your environment. You can deploy to the cloud or local infrastructure. You can describe any resource you have in code no matter how much it differs from a standard configuration.

With Chef there is no reason to start from scratch; many standard infrastructure configurations and tasks are already described in Chef cookbooks are available for free on the **Chef Supermarket** site.

Once you master Chef, you can use it to

- Fully automate deployments, including internal development and end-user systems
- Automate scaling of infrastructure
- Make your infrastructure self-healing

As an example, Tom Hallet used Chef to create a tool called SoloWizard, which he uses to automate

deployments of his Mac OS X development machines. SoloWizard is based on the work Pivotal Labs has done to promote the use of automation for developer and end-user systems. As you can see from [Figure 1-2](#), SoloWizard lets you create a new development environment with a single command. You can even personalize the output script to meet your needs by making choices on a simple website. The tool is publicly available at [the SoloWizard site](#).

# Where Do We Go From Here?

---

In the first half of this book, we'll cover all the essential components of client-side Chef, showing you how to make your infrastructure versionable and repeatable with code. We'll cover:

- Test Kitchen
- Chef Solo/Chef Local
- Cookbooks

We'll also give you a firm foundation in the basics of infrastructure coding with Chef by providing lots of hands-on examples.

In the second half of this book, we'll introduce you to the essential topics related to Chef Server, showing you how to make your configuration management abilities scale as your infrastructure increases in complexity and scope:

- On-Premises Chef Server
- Chef Zero
- Roles, Data Bags, and Environments
- Knife

Tools covered in this book are freely available as open source downloads. Some paid tools will be mentioned, but are not required to learn or use Chef.

In the next chapter, we'll start your adventure with Chef by walking you through the Chef Development Kit installation process, so that you can get started right away writing Chef code.

SoloWizard - Configurable x

www.solowizard.com

Fork me on GitHub

# SoloWizard

Who says you can't bootstrap your machine with a one-liner?

---

**tl;dr: You are a few clicks away from a configurable chef script for your OS-X development machine.**

Many teams use chef to manage their production machines, but developers often build their development boxes by hand. SoloWizard makes it painless to create a configurable chef solo script to get your development machine humming: mysql, sublime text, .bash\_profile tweaks to OS-X settings - it's all there!

Dependencies: Xcode and OS-X Mountain Lion ([read the faq for more details](#))

Solo Wizard Demo - Configurable Chef Scripts

1. Make sure you don't have any recipes (checked which aren't present on the "New Solo Wizard" page) [View Demo](#)

2. Create a Solo Wizard script for the recipe

3. Download the Solo Wizard chef script (download)

4. Add the following line right below the line which has "solo"

5. Run the script and save for the next!

**create your personal Solo Wizard script:**

Select which recipes you'd like to use and then click the button - we will build a chef script for you, which you can bookmark or share with others.

---

**Development Stack**

- activemq
- erlang
- imagemagick
- java
- memcached
- nginx
- node\_js
- qt
- rabbitmq
- redis
- selenium\_webdriver
- snmpd
- solr

---

**Databases**

- mongodb
- mysql
- postgres

---

**Command Line Tools**

- ack
- ec2\_api\_tools

Figure 1-2. SoloWizard bootstraps Mac OS X development workstations





- [\*Hactivism and Cyberwars: Rebels with a Cause? book\*](#)
- [click Cody's Collection of Popular SAS Programming Tasks and How to Tackle Them](#)
- [download \*The Long and Faraway Gone pdf, azw \(kindle\)\*](#)
- [Frommer's Best Day Trips From London \(Frommer's Color Complete\) \(5th Edition\) pdf, azw \(kindle\), epub, doc, mobi](#)
- [\*\*Introducing Psychoanalysis: A Graphic Guide pdf, azw \(kindle\)\*\*](#)
  
- <http://patrickvincitore.com/?ebooks/The-Story--A-Reporter-s-Journey.pdf>
- <http://academialanguagebar.com/?ebooks/Bones.pdf>
- <http://junkrobots.com/ebooks/Fig-Heaven--70-Recipes-for-the-World-s-Most-Luscious-Fruit.pdf>
- <http://www.netc-bd.com/ebooks/The-Complete-Book-of-Rod-Building-and-Tackle-Making.pdf>
- <http://cavalldecartro.highlandagency.es/library/Introducing-Psychoanalysis--A-Graphic-Guide.pdf>