Community Experience Distilled

Mastering SaltStack

Take charge of SaltStack to automate and configure enterprise-grade environments

Foreword by Thomas S. Hatch - Founder and CTO, SaltStack

SALTSTACK

Joseph Hall

[PACKT] PUBLISHING open source* community experience distilled

Mastering SaltStack

Take charge of SaltStack to automate and configure enterprise-grade environments

Joseph Hall

[PACKT] open source
community experience distilled
PUBLISHING

BIRMINGHAM - MUMBAI

Mastering SaltStack

First published: August 2015

Production reference: 1130815

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

ISBN 978-1-78528-216-4

www.packtpub.com

Credits

Author
Joseph Hall

Reviewers
Pedro Algarvio
Darrel Clute
JiWei Liu
Hoàng Đình Quân
Peng Yao

Commissioning Editor
Kartikey Pandey

Acquisition Editor
Vivek Anantharaman

Content Development Editor
Merwyn Dsouza

Technical Editor
Siddhesh Patil

Copy Editors
Relin Hedly
Sonia Mathur

Project Coordinator
Neha Bhatnagar

Proofreader
Safis Editing

Indexer
Monica Mehta

Graphics
Disha Haria

Production Coordinator
Nilesh R. Mohite

Cover Work
Nilesh R. Mohite

Foreword

This is the Salt book I have been wanting to see for a long time. As the creator of Salt, I feel that many people don't know about the advanced and powerful capabilities of the Salt platform. Diving into the parts of Salt less trodden is where the truly amazing value of Salt comes into play. The more powerful aspects of Salt, such as how to use the Reactor for dynamic intelligent management, how to use salt-ssh to manage a wide variety of situations, and how to take Salt to the next level, are revealed in this book in an easy-to-understand way. I hope that this book will be a great help in bringing the great power of Salt to more people.

Joseph Hall is likely the best person to write this book. He is not only a close friend, but has also been involved with the Salt project from the very beginning, including the early design of the Salt States system. Joseph is the second person to write code for Salt (apart from me). He was the first engineer hired by SaltStack.

Thomas S. Hatch

Founder and CTO, SaltStack

About the Author

Joseph Hall has touched just about every area of the modern technology world from his beginnings as a support technician to being a web programmer, QA engineer, systems administrator, Linux instructor, and cloud engineer. He is currently a senior cloud and integrations engineer at SaltStack. Joseph enjoys working with some of the best minds in the business with his coworkers and SaltStack's partners. He is also a classically trained chef. Joseph's supreme pleasure lies in working with what he calls computational gastronomy.

I would like to thank my wife, Nat, for her support when I stayed up every night to write this book. I would also like to thank Tom Hatch for writing Salt and having the guts to turn it into the best company that I've ever worked for. My sincere thanks go out to Colton Meyers for hooking me up with Packt Publishing. I would also like to thank the Salt community for being awesome and helping make Salt what it is today.

About the Reviewers

Pedro Algarvio was a Sound Technician until May 2015. He likes to keep himself busy. Therefore, Pedro set out to make computers work for him. In his endeavor to deepen his knowledge of computers, he started with shell scripting and then moved on to learn Perl, and finally settled with Python. Pedro was involved with several open source projects. However, he credits Salt for giving him the opportunity to learn the most. He joined SaltStack in May 2015 and dedicated his time to improve the Salt software.

I would like to thank my wife for her continuous friendship, love, and support throughout. She even supported me when I took the risk of quitting my job to to pursue a skill, in which I had no experience. I would also like to thank my twins for allowing me some spare time to review Joseph's book. Further, I want to thank the Salt community for teaching me new things everyday. Last but not least, I would like to thank God.

I love learning new things. SaltStack gave the opportunity to learn more about Salt, which made me grow as a Python coder and enthusiast. I would like to thank SaltStack for their continuous belief and encouragement to the point of inviting me to join the team.

Darrel Clute is an IT infrastructure architect. Throughout his career, he has predominately focused on network engineering. Darrel has also spent an equal amount of time focusing on systems engineering, primarily with Unix-based systems. Apart from his job, he is also an advocate of the open source software. This is used in enterprise as well as for individuals. Darrel will never advocate the use of open source purely for the sake of it.

Outside his core competencies, he also has extensive experience with proprietary and open source virtualization platforms. His knowledge and experience with IaaS solutions—such as OpenStack—is constantly increasing as well. Additionally, with his systems engineering experience, Darrel has been exposed to and has supported various web and enterprise applications. His most recent exposure has been to PaaS solutions. This has led him to advocate that a public or private cloud offering is not complete without coupling IaaS and PaaS solutions designed and deployed in lockstep.

Beyond his core infrastructure, Darrel has recently been developing programing skills to augment his daily activities. Throughout his career, he has utilized various languages and utilities to automate infrastructure. Some of Darrel's programming has been with the use of Bash, net-snmp, sed, awk, and holistically Python.

JiWei Liu graduated in 2011. He is employed as a cloud computing and operation and maintenance engineer at Gamewave Group Limited (PRC), China's largest web game provider. It is also the leader of the Chinese web game industry and the professional provider of interactive entertainment services.

I would like to thank my sweetheart for her help and support in the process of writing this book.

Hoàng Đình Quân is a junior system administrator and network infrastructure engineer. He has extensive experience in designing, deploying, hardening, and maintaining Linux systems and the network infrastructure for small-and medium-sized companies.

Peng Yao is an operations engineer with more than 8 years of experience in infrastructure automation, data visualization, and cloud computing. He is the founder and coordinator of the China SaltStack user group.

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.

I would like to dedicate this book to the memory of Tim Hollinger. You were there with us in the early days of Salt, and you will not be forgotten. How I wish, how I wish you were here. Shine on, you crazy diamond.

– Joseph Hall

Table of Contents

Preface

I'm very excited to have been given the chance to put this book together. From an idea in the brain of Tom Hatch to an award-winning open source project to the flagship product of an award-winning open source company, I've been given the rare opportunity to watch Salt grow. Salt has become an incredibly powerful framework, which I wish I had access to years ago.

Everyday, I learn something new about Salt. This book is a collection of a number of these things, which is aimed at the advanced user. Don't see it as the last word on any of the topics it covers. Instead, see it as a guide to using this tool to its fullest potential on your journey.

As you read through this book, I hope that the ideas and examples in it inspire you to update and innovate your infrastructure.

What this book covers

Chapter 1, *Reviewing a Few Essentials*, talks about how to review a few fundamental concepts to get into the right frame of mind. While many of the concepts should be familiar to the experienced user, you are likely to find plenty of new information as well.

Chapter 2, *Diving into Salt Internals*, jumps into the deeper workings behind Salt. It discusses the internal configuration, the loader system, renderers, and the state compiler.

Chapter 3, *Exploring Salt SSH*, explores how Salt SSH is a powerful tool. It's been getting a lot of love from the core developers lately. This is possibly the most complete discussion of Salt SSH.

Chapter 4, *Managing Tasks Asynchronously*, discusses how one of the most important concepts behind Salt is asynchronicity. This chapter lays down the fundamentals that will be referenced throughout the rest of the book.

Chapter 5, *Taking Salt Cloud to the Next Level*, goes deeper, exposing parts of Salt Cloud, which turn casual users into experts. No matter how much you've used Salt Cloud, there's a good chance you've only scratched the surface.

Chapter 6, *Using Salt with REST*, talks about how it's almost impossible to work with technology these days without depending on REST services. It uses Salt to tie these services to your infrastructure with ease.

Chapter 7, *Understanding the RAET Protocol*, teaches you the concepts behind RAET and how they impact upon you. RAET is still new, but it's already found its way into large organizations.

Chapter 8, *Strategies for Scaling*, talks about how to never assume that your infrastructure will stay small. It makes you think about how to scale your infrastructure properly before it's too late.

Chapter 9, *Monitoring with Salt*, discovers how Salt is a powerful monitoring tool if you know how to use it. It tells you how to integrate it with existing tools or use Salt alone.

Chapter 10, *Exploring Best Practices*, discusses the good and bad ways to use any tool. It teaches you the good ways to use Salt.

Chapter 11, *Troubleshooting Problems*, tells you where to look and how to find help when things go wrong.

What you need for this book

To follow the examples in this book, you should be running at least version 2015.5 of Salt. Only one machine is strictly necessary because both the salt-master and the salt-minion service can be run together, but Linux is currently required to run the salt-master service.

The examples in this book are targeted at Ubuntu Linux, except where stated otherwise.

Who this book is for

This book is ideal for professionals who have been managing groups of servers and want to learn how to add functionality and expand their toolset. This book explains some of the more advanced features of Salt. It explores how to use them to bring additional power to the fundamentals that the professionals have already been using.

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: "This function does little more than the `test.ping` command."

A block of code is set as follows:

```
nodegroups:
  webdev: 'I@role:web,G@cluster:dev'
  webqa: 'I@role:web,G@cluster:qa'
  webprod: 'I@role:web,G@cluster:prod'
```

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

```
# salt -S 192.168.0.0/24 test.ping
```

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: "Click the **Join This Group** link and you will be subscribed".

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
Reviewing a Few Essentials

Salt is a very powerful automation framework. Before we delve into the more advanced topics that this book covers, it may be wise to go back and review a few essentials. In this chapter, we will cover the following topics:

- Using remote execution
- Basic SLS file tree structure
- Using States for configuration management
- Basics of Grains, Pillars, and templates

This book assumes that you already have root access on a device with a common distribution of Linux installed. The machine used in the examples in this book is running Ubuntu 14.04, unless stated otherwise. Most examples should run on other major distributions, such as recent versions of Fedora, RHEL 5/6/7, or Arch Linux.

Executing commands remotely

The underlying architecture of Salt is based on the idea of executing commands remotely. This is not a new concept; all networking is designed around some aspect of remote execution. This could be as simple as asking a remote Web server to display a static Web page, or as complex as using a shell session to interactively issue commands against a remote server.

Under the hood, Salt is an example of one of the more complex types of remote execution. But whereas most Internet users are used to interacting with only one server at a time (so far as they are aware), Salt is designed to allow users to explicitly target and issue commands to multiple machines directly.

Master and Minions

Salt is based around the idea of a Master, which controls one or more Minions. Commands are normally issued from the Master to a target group of Minions, which then execute the tasks specified in the commands and then return any resulting data back to the Master.

Targeting Minions

The first facet of the `salt` command is targeting. A target must be specified with each execution, which matches one or more Minions. By default, the type of target is a *glob*, which is the style of pattern matching used by many command shells. Other types of targeting are also available, by adding a flag. For instance, to target a group of machines inside a particular subnet, the `-S` option is used:

```
# salt -S 192.168.0.0/24 test.ping
```

The following are most of the available target types, along with some basic usage examples. Not all target types are covered here; *Range*, for example, extends beyond the scope of this book. However, the most common types are covered.

Glob

This is the default target type for Salt, so it does not have a command line option. The Minion ID of one or more Minions can be specified, using shell wildcards if desired.

When the `salt` command is issued from most command shells, wildcard characters must be protected from shell expansion:

```
# salt '*' test.ping
# salt \* test.ping
```

When using Salt from an API or from other user interfaces, quoting and escaping wildcard characters is generally not required.

Perl Compatible Regular Expression (PCRE)

Short Option: `-E`

Long Option: `--pcre`