Automating Linux and Unix System Administration

Second Edition

Nate Campi and Kirk Bauer

Apress®

Automating Linux and Unix System Administration, Second Edition

ISBN-13 (pbk): 978-1-4302-1059-7

ISBN-13 (electronic): 978-1-4302-1060-3

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

Lead Editor: Frank Pohlmann
Technical Reviewer: Mark Burgess
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Kylie Johnston
Copy Editors: Nina Goldschlager, Heather Lang
Associate Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositor: Linda Weidemann, Wolf Creek Press
Proofreader: Nancy Sixsmith
Indexer: Becky Hornyak
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

The source code for this book is available to readers at `http://www.apress.com`.

I dedicate this book to my dear grandmother Mary Lou.
Her influence makes everyone around her a better person,
and her presence lights up a room.
She is beautiful inside and out,
and she meets adversity with faith,
quiet dignity, and grace.

—Nate Campi

Contents at a Glance

Contents

About the Authors

NATE CAMPI is a UNIX and Linux system administrator by trade, currently working as a UNIX operations manager in San Francisco. His system administration experience is almost entirely with companies with large-scale web operations based on open source software. In his copious free time, he enjoys jogging, watching spaghetti westerns, experimenting with Linux systems, and spending time with his family.

KIRK BAUER has been involved in computer programming since 1985. He has been using and administering UNIX systems since 1994. Although his personal favorite UNIX variant is Linux, he has administered and developed on everything from FreeBSD to Solaris, AIX, and HP-UX. He is the author of various open source solutions such as Logwatch.

Kirk has been involved with software development and system/network administration since his first year at the Georgia Institute of Technology. He has done work for the Georgia Tech Research Institute, Fermi National Accelerator Laboratory, and DHL. In 2000, Kirk was one of the founders and the chief technology officer of TogetherWeb, which was purchased in 2003 by Proficient Systems. Kirk is now a systems engineer with F5 Networks.

Kirk graduated from Georgia Tech in 2001 with a bachelor's degree in computer engineering and is currently pursuing his MBA at Arizona State University. He lives in Peoria, Arizona, with his two dogs, and is looking forward to getting married to his lovely fiancée, Rachel.

About the Technical Reviewer

MARK BURGESS holds a first class honors degree in physics and a Ph.D. in theoretical physics from the University of Newcastle upon Tyne. After working as a physicist, he began to apply the methods of physics to the study of computers and eventually changed research fields to study the formalization of system administration. His current research interests include the behavior of computers as dynamic systems and applying ideas from physics to describe computer behavior. Mark is the author of the popular configuration management software package cfengine. He has received a number of awards including the SAGE 2003 Professional Contribution Award "for groundbreaking work in systems administration theory and individual contributions to the field." He currently holds the Professorship in Network and System Administration at Oslo University College.

Acknowledgments

Only two names are on the book cover, but many talented and dedicated people worked to make this book the best it could be.

We are very grateful to Paul W. Frields from Red Hat for Red Hat Enterprise Linux licenses. This book wouldn't have been possible without them. Mark Burgess lent his unique insight into both cfengine and the book writing process. Our editor Frank Pohlmann is incredibly skilled at finding the weak points in a description and forcing us to explain everything thoroughly. Thanks to our project manager Kylie Johnston; she is a consummate professional. Thanks to our copy editors Nina Goldschlager and Heather Lang, who are very talented and easy to work with. And thanks to our production editor Ellie Fountain.

We really need to thank our families for putting up with our mental absence while writing this book.

Finally, we'd like to thank the energy drink industry for enabling us to stay up late at night even when totally exhausted, go to work the next day feeling like we had been hit by a train, and do it all over again the very next night.

Introduction

The system administrator is one of the users of a system, and something more. The administrator wears many hats, as knowledgeable user of UNIX commands, as an operator of system hardware, and as a problem solver. The administrator is also called upon to be an arbitrator in human affairs. A multiuser computer is like a vast imaginary space where many people work and utilize the resources found there. The administrator must be the village elder in this space and settle the disputes that may arise with, hopefully, the wisdom of Solomon.

—Rebecca Thomas and Rik Farrow
(*UNIX Administration Guide for System V,*
Pearson PTR, 1989)

We find it interesting how little UNIX system administration has changed in the last twenty years. If you substitute "computer network" for "multiuser computer," this description still fits perfectly.

The main difference in UNIX system administration between 1989 and 2008 (besides ubiquitous networking) is the sheer number of systems that the average system administrator deals with. Automation is the primary tool to deal with the chaos that can result from so many systems. With it, you can deploy systems identically every time, restore systems to a known good state, and implement changes reliably across all systems (or only an appropriate subset).

We do not claim that the approaches, procedures, and tools used in this book are the only way to set up and maintain a UNIX-based environment. Instead, we walk you through the creation of an example environment, and during the process, help you gain a solid understanding of the basic principles of system automation. This way, you can decide for yourself how you want to set up your own UNIX-based environment.

This book *isn't* like most UNIX/Linux administration books, because it illustrates techniques and principles by building a real UNIX/Linux environment from scratch. We demonstrate that you can configure each host at your site, from installation through production service to system retirement, without logging in and making manual changes to the host. Instead, we'll configure the hosts via imaging systems designed for unattended installation, followed by management with an automation framework.

We wrote this book, because we felt that it is important to demonstrate that an entire site can be managed using automation. Our goal is to be able to quickly, easily, and reliably restore hosts to service after complete system failure. The host might have failed

due to hardware issues; an entire geographic region might be unreachable due to natural disaster, or you might simply have purchased updated hardware on which to run that particular host and need to upgrade. The point of our approach is to configure a host only once and, from that point on, allow an automation system to do that work for you.

Whether you choose to use our exact setup or something completely different, you'll have gained knowledge and experience by going though the process with us in our example environment. Our promise to you is that if you need to configure a new UNIX-based infrastructure from scratch (and you're able or allowed to use the operating systems and software we demonstrate), you can use this book to create a fully functional and scalable new infrastructure. Every service and piece of architecture that our new environment needs is set up using automation.

This book moves fast and will be best utilized if you follow along with the examples and implement the described steps on systems of your own. In addition, download the code and configuration files from the Source Code page of the Apress web site (`http://www.apress.com`).

Who This Book Is For

This book is written for the experienced system administrator. We have made every attempt to refer you to appropriate external sources when we weren't able to delve into great detail on a service or protocol that we were automating. In addition, little explanation is given to the usage of basic UNIX/Linux commands and shell scripts. You don't, however, have to be an advanced system administrator. We feel that a system administrator with only one or two years of full-time on-the-job experience is more than ready to utilize the concepts and tools in this book.

How This Book Is Structured

The book begins with four introductory chapters that you should be very familiar with before you move on to later, more detailed chapters. The later chapters, starting with Chapter 5, build a new UNIX environment: we set up an automation system; automate installation systems; and enhance the site with real applications, monitoring, reporting, and security.

Chapter 1, "Introducing the Basics of Automation," covers the reasons for and benefits of automation, as well as the methodology behind it. Also, the `sudo` utility is introduced and explained.

Chapter 2, "Applying Practical Automation," covers the steps behind automating a common procedure—adding a new user account. During the process, the core tenets of automation are covered.

Chapter 3, "Using SSH to Automate System Administration Securely," covers the basics of using secure shell (SSH), discusses SSH security concerns, describes how to set up public key authentication in SSH, and delves into various other related topics such as SSH log analysis.

Chapter 4, "Configuring Systems with cfengine," explains the concepts behind cfengine, as well as the various cfengine daemons and utilities. A full discussion takes place of the common configuration settings in the main cfengine configuration file. The requirements for a minimal cfengine architecture with two hosts are fully explored.

Chapter 5, "Bootstrapping a New Infrastructure," covers the cfengine configuration for a new, automated UNIX/Linux environment. A "master" cfengine host is set up, with all the required configuration files to manage new Red Hat Linux, Debian Linux, and Solaris hosts. This is the first step in building a UNIX/Linux environment from scratch using automation.

Chapter 6, "Setting Up Automated Installation," demonstrates the automated installation of Red Hat Linux using Kickstart, Debian Linux using Fully Automatic Installation (FAI), and Sun Solaris using Jumpstart. The hosts deployed in this chapter continue to be used in the later development of our example UNIX/Linux infrastructure.

Chapter 7, "Automating a New System Infrastructure," covers the automation of these services and procedures in our new infrastructure: the Network Time Protocol (NTP), Domain Name System (DNS), standardized local account files and new user accounts, mail routing, and home directories mounted with the Network File System (NFS).

Chapter 8, "Deploying Your First Application," covers the deployment and configuration of the Apache web server, demonstrating various ways to automate the distribution of both the web server daemon binaries and the web content. Along the way, you learn about sharing data with NFS, rsync, `scp`, cfengine data copies, and Subversion.

Chapter 9, "Generating Reports and Analyzing Logs," covers automated syslog and cfengine log analysis and reporting in our new infrastructure.

Chapter 10, "Monitoring," uses cfengine to automate the deployment and configuration of Ganglia and Nagios in our example environment.

Chapter 11, "Infrastructure Enhancement," uses cfengine to manage version control with Subversion, including branching the cfengine configuration tree to create testing and development environments. Also, backups are handled, in a very simple way.

Chapter 12, "Improving System Security," covers the implementation of security enhancements with cfengine. Measures undertaken include removing the SUID bit from root-owned binaries, protecting system accounts, applying UNIX/Linux patches and vendor updates, shutting down unneeded daemons, adding host-based firewalls, and more.

Appendix A, "Introducing the Basic Tools," provides a basic introduction to the tools used throughout this book and provides a good starting point for understanding and utilizing the examples presented in this text. This appendix covers the following tools: bash, Perl, `grep`, `sed`, and AWK.

Appendix B, "Writing cfengine Modules," covers extending cfengine through modules. This is a quick but thorough introduction using examples.

Downloading the Code

The source code for this book is available to readers at `http://www.apress.com` in the Source Code section of this book's home page. Please feel free to visit the Apress web site and download all the code there. You can also check for errata and find related titles from Apress.

Contacting the Authors

We have gone through several stages of proofreading and error checking during the production of this book in an effort to reduce the number of errors. We have also tried to make the examples and the explanations as clear as possible.

There may, however, still be errors and unclear areas in this book. If you have questions or find any of these errors, please feel free to contact us at `nate@campin.net`. You can also visit the Apress web site at `http://www.apress.com` to download code from the book and see any available errata.