



Learn by doing: less theory, more results

Puppet 3

Start from scratch with the Puppet configuration management system, and learn how to fully utilize Puppet through simple, practical examples

Beginner's Guide

John Arundel

[PACKT] open source*
PUBLISHING community experience distilled

Puppet 3 Beginner's Guide

Start from scratch with the Puppet configuration management system, and learn how to fully utilize Puppet through simple, practical examples

John Arundel



BIRMINGHAM - MUMBAI

Puppet 3 Beginner's Guide

Copyright © 2013 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: April 2013

Production Reference: 1050413

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

ISBN 978-1-78216-124-0

www.packtpub.com

Cover Image by Faiz Fattohi (faizfattohi@gmail.com)

Credits

Author

John Arundel

Reviewers

Ugo Bellavance

Jason Slagle

Johan De Wit

Acquisition Editor

Joanne Fitzpatrick

Lead Technical Editor

Joanne Fitzpatrick

Technical Editors

Sharvari Baet

Kaustubh S. Mayekar

Project Coordinator

Anugya Khurana

Proofreader

Lawrence A. Herman

Indexer

Monica Ajmera Mehta

Graphics

Ronak Dhruv

Aditi Gajjar

Production Coordinator

Melwyn D'sa

Cover Work

Melwyn D'sa

About the Author

John Arundel is an infrastructure consultant who helps people make their computer systems more reliable, useful, and cost-effective and has fun doing it. He has what Larry Wall describes as the three great virtues of a programmer: laziness, impatience, and hubris.

Laziness, because he doesn't like doing work that a computer could do instead. Impatience, because he wants to get stuff done right away. Hubris, because he likes building systems that are as good as he can make them.

He was formerly a senior operations engineer at global telco Verizon, designing resilient, high-performance infrastructures for corporations such as Ford, McDonald's, and Bank of America. He now works independently, helping to bring enterprise-grade performance and reliability to clients with slightly smaller pockets but very big ideas.

He likes writing books, especially about Puppet. It seems that at least some people enjoy reading them, or maybe they just like the pictures. He also occasionally provides training and coaching on Puppet, which turns out to be far harder than simply doing the work himself.

Off the clock, he can usually be found driving a Land Rover up some mountain or other. He lives in a small cottage in Cornwall and believes, like Cicero, that if you have a garden and a library, then you have everything you need.

You can follow him on Twitter at [@bitfield](https://twitter.com/bitfield).

Thanks are due to my friend Luke Kanies, who created a configuration management tool that sucks less, and also to the many proofreaders and contributors to this book, including Andy Brockhurst, Tim Eilers, Martin Ellis, Adam Garside, Stefan Goethals, Jennifer Harbison, Kanthi Kiran, Cristian Leonte, Habeeb Rahman, John Smith, Sebastiaan van Steenis, Jeff Sussna, Nate Walck, Bryan Weber, and Matt Willsher.

About the Reviewers

Ugo Bellavance has done most of his studies in e-commerce, started using Linux at Red Hat 5.2, got Linux training from Savoir-Faire-Linux at the age of 20, and got his RHCE on RHEL 6 in 2011. He's been a consultant in the past, but he's now an employee for a provincial government agency for which he manages the infrastructure (servers, workstations, network, security, virtualization, SAN/NAS, PBX). He's a big fan of open-source software and its underlying philosophy. He's worked with Debian, Ubuntu, and SUSE, but what he knows best is RHEL-based distributions. He's known for his contributions to the MailScanner project (he has been a technical reviewer for the MailScanner book), but he also gave time to different open-source projects, such as mondorescue, OTRS, SpamAssassin, pfSense, and a few others.

I thank my lover, Lysanne, who accepted allowing me some free time slots for this review even with a 2-year-old and a 6-month-old to take care of. The presence of these 3 human beings in my life is simply invaluable.

I must also thank my friend Sébastien, whose generosity is only matched by his knowledge and kindness. I would never have reached that high in my career if it wasn't for him.

Jason Slagle is a 15-year veteran of Systems and Network administration. Having worked on everything from Linux systems to Cisco networks and SAN Storage, he is always looking for ways to make his work repeatable and automated. When he is not hacking at a computer for work or pleasure, he enjoys running, cycling, and occasionally geocaching.

He is currently employed by CNWR, Inc., an IT and Infrastructure consulting company in his home town of Toledo, Ohio. There he supports several larger customers in their quest to automate and improve their infrastructure and development operations.

I'd like to thank my wife, Heather, for being patient through the challenges of being married to a lifelong systems guy, and my new son, Jacob, for bringing a smile to my face on even the longest days.

Johan De Wit was an early Linux user and he still remembers those days building a 0.9x Linux kernel on his brand-new 486 computer that took a whole night, and always had a great love for the UNIX Operating System.

It is not surprising that he started a career as a UNIX system administrator.

Since 2009, he has been working as an open-source consultant at Open-Future, where he got the opportunity to work with Puppet. Right now, Puppet has become Johan's biggest interest, and recently he became a Puppet trainer.

Besides his work with Puppet, he spends a lot of his free time with his two lovely kids and his two Belgian draft horses, and if time and the weather permit, he likes to drive his chopper.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

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.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across 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 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 nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Introduction to Puppet	7
The problem	8
Configuration management	8
A day in the life of a sysadmin	8
Keeping the configuration synchronized	9
Repeating changes across many servers	10
Self-updating documentation	10
Coping with different platforms	10
Version control and history	11
Solving the problem	11
Reinventing the wheel	11
A waste of effort	12
Transferable skills	12
Configuration management tools	12
Infrastructure as code	13
Dawn of the devop	13
Job satisfaction	14
The Puppet advantage	14
Welcome aboard	15
The Puppet way	15
Growing your network	16
Cloud scaling	16
What is Puppet?	16
The Puppet language	16
Resources and attributes	17
Summary	18
Configuration management	18
What Puppet does	18

The Puppet advantage	19
Scaling	19
The Puppet language	19
Chapter 2: First steps with Puppet	21
What you'll need	22
Time for action – preparing for Puppet	22
Time for action – installing Puppet	23
Your first manifest	26
How it works	26
Applying the manifest	27
Modifying existing files	28
Exercise	28
Organizing your manifests	28
Time for action – creating a directory structure	29
Creating a nodes.pp file	29
Time for action – creating a node declaration	30
Summary	31
Installing Puppet	31
Manifests	31
Nodes	32
Chapter 3: Packages, Files, and Services	33
Packages	34
Time for action – installing Nginx	34
More about packages	36
Installing specific versions	36
Removing packages	37
Updating packages	37
Modules	38
Time for action – creating an Nginx module	38
Time for action – making a "puppet apply" command	40
Services	41
Time for action – adding the Nginx service	41
Requiring resources	43
More about services	44
Starting a service at boot time	44
Services that don't support "status"	45
Specifying how to start, stop, or restart a service	46
Files	46
Time for action – deploying a virtual host	46
Notifying other resources	49

The package–file–service pattern	49
Exercise	50
Summary	50
Packages	50
Modules	50
Services	51
Starting services at boot	51
Service status options	51
Service control commands	51
Resource dependencies	51
Files	52
Chapter 4: Managing Puppet with Git	53
What is version control?	54
Time for action – importing your manifests into Git	55
Time for action – committing and inspecting changes	56
How often should I commit?	60
Branching	60
Distributing Puppet manifests	61
Reliability	61
Scalability	61
Simplicity	61
Time for action – creating a master Git repo	62
Time for action – cloning the repo to a new machine	63
Time for action – adding a new node	65
Time for action – pushing changes to the master repo	65
Exercise	66
Pulling changes automatically	67
Time for action – automatic pull-and-apply script	67
Learning more about Git	68
Summary	68
Why version control?	69
Getting started with Git	69
Networking Puppet	69
Chapter 5: Managing users	71
Users	72
Security and access control	72
What Puppet can do	72
Time for action – creating a user	73
Removing user accounts	74

Access control	75
What is SSH?	75
Managing SSH keys	75
Time for action – adding an SSH authorized key	76
Generating new SSH keys	78
Special-purpose keys	78
Locking user accounts	78
Managing SSH configuration	79
Time for action – deploying an SSH configuration file	79
User privileges	80
sudo	81
Time for action – deploying a sudoers file	81
Summary	83
Security practices	83
User resources	83
Removing or locking accounts	84
Managing SSH keys	84
Configuring SSH	84
Managing privileges with sudo	85
Chapter 6: Tasks and templates	87
Running commands with exec resources	88
Time for action – running an arbitrary command	88
Running commands selectively	89
Triggering commands	90
Chaining commands	90
Command search paths	91
Scheduled tasks	92
Time for action – scheduling a backup	92
More scheduling options	94
Running jobs at regular intervals	94
Running a job as a specified user	94
Exercise	94
Distributing files	95
Time for action – using a recursive file resource	95
Using templates	97
Time for action – templating an Nginx virtual host	97
Inline templates	101
System facts	101
Doing the math	102
Putting it all together	102

Summary	103
Exec resources	103
Scheduled jobs	104
Recursive file resources	105
Templates	105
Chapter 7: Definitions and Classes	107
Grouping resources into arrays	108
Definitions	109
Passing parameters to definitions	111
Optional parameters	112
Time for action – creating a definition for Nginx websites	112
Multiple instances of definitions	115
Exercise	115
Classes	115
Defining classes	115
Putting classes inside modules	116
Declaring classes	116
What's the difference between a class and a definition?	117
Time for action – creating an NTP class	117
Summary	120
Arrays	120
Definitions	120
Classes	121
Chapter 8: Expressions and Logic	123
Conditionals	123
If statements	124
else and elsif	124
Unless statements	125
Case statements	125
The default case	127
Matching multiple cases	127
Selectors	127
Expressions	128
Comparisons	128
Equality	128
Magnitude	129
Substrings	129
Boolean operators	130
Combining Boolean operators	130
Arithmetic operators	130

Regular expressions	131
Operators	132
Syntax	132
Conditionals	133
Capture variables	133
Substitutions	134
Node definitions	135
Arrays and hashes	136
Grouping resources with arrays	136
Getting values out of arrays	137
Hashes	138
Multilevel hashes	138
Testing hash keys	139
Summary	139
Conditionals	139
Operators	140
Regular expressions	140
Text substitution	141
Arrays	141
Hashes	142
Chapter 9: Reporting and troubleshooting	143
Reporting	144
Summary reports	144
Enabling reports	145
What's in a report?	145
Time for action – generating a report	146
Using reports	150
Debug runs	150
Noop runs	151
Syntax checking	152
Debug output	152
Notify resources	153
Exec output	153
Specifying expected exit status	155
Monitoring	155
Managing monitoring with Puppet	155
What to monitor	156
Monitoring Puppet status	156
Problems with Puppet	157
Staying in sync	157

Errors	157
Compilation errors	158
Diagnosing errors	158
Missing file sources	158
Missing parent directory	159
Mistyped command line options	160
Summary	160
Reporting	160
Debug and dry-run modes	160
Printing messages	161
Monitoring Puppet	161
Common Puppet errors	161
Chapter 10: Moving on Up	163
Puppet style	164
Break out code into modules	164
Refactor common code into definitions	164
Keep node declarations simple	166
Use puppet-lint	167
Make comments superfluous	168
Puppet learning resources	169
Reference	169
Resource types	169
Language and syntax	170
Facts	170
Style	170
Modules and code	171
Puppet Forge	171
The Puppet Cookbook	171
Projects	172
Puppet everywhere	173
User accounts	173
System toolbox	173
Time sync	173
Monitoring server	174
Puppetize your key services	174
Automate backups	175
Set up staging servers	175
Automate everything	175
Last word	176
Index	179

Preface

If you work with computer systems, then you know how time-consuming it can be to install and configure software, to do administration tasks such as backups and user management, and to keep the machines up to date with security patches and new releases. Maybe you've already come up with some written procedures, shell scripts, and other ways to document your work and make it more automated and reliable.

Perhaps you've read about how Puppet can help with this, but aren't sure how to get started. The online documentation is great for reference, but doesn't really explain the whole thing from scratch. Many of the books and tutorials available spend a lot of time explaining how to set up your Puppet server and infrastructure before ever getting to the point where you can use Puppet to actually do something.

In my work as an infrastructure consultant I do a good deal of Puppet training, mostly for absolute beginners, and I've found that the most effective and fun way to do this is to get into some real work right away. In the first five minutes, I have people making changes to their systems using Puppet. If there was a fire alarm and we had to terminate the class after that first five minutes, they would still go away knowing something useful that could help them in their jobs.

I've taken the same approach in this book. Without going into lots of theory or background detail, I'll show you how to do useful things with Puppet right away: install packages and config files, create users, set up scheduled jobs, and so on. Every exercise deals with something real and practical that you're likely to need in your work, and you'll see the complete Puppet code to make it happen, along with step-by-step instructions for what to type and what output you'll see.

After each exercise, I'll explain in detail what each line of code does and how it works, so that you can adapt it to your own purposes, and feel confident that you understand everything that's happened. By the end of the book, you will have all the skills you need to do real, useful, everyday work with Puppet.

So let's get started.

What this book covers

Chapter 1, Introduction to Puppet, explains the problem of configuration management and why traditional manual approaches to them don't scale. It shows how Puppet deals with these problems efficiently, and introduces the basic architecture of Puppet.

Chapter 2, First Steps with Puppet, guides you through installing Puppet for the first time, creating a simple manifest, and applying it to a machine. You'll see how to use the Puppet language to describe and modify resources, such as a text file.

Chapter 3, Packages, Files, and Services, shows you how to use these key resource types, and how they work together. We'll work through a complete and useful example based on the Nginx web server.

Chapter 4, Managing Puppet with Git, describes a simple and powerful way to connect machines together using Puppet, and to distribute your manifests and work on them collaboratively using the version control system Git.

Chapter 5, Managing Users, outlines some good practices for user administration and shows how to use Puppet to implement them. You'll also see how to control access using SSH and manage user privileges using `sudo`.

Chapter 6, Tasks and Templates, covers more key aspects of automation: scheduling tasks, and building configuration files from dynamic data using Puppet's template mechanism.

Chapter 7, Definitions and Classes, builds on previous chapters by showing you how to organize Puppet code into reusable modules and objects. We'll see how to create definitions and classes, and how to pass parameters to them.

Chapter 8, Expressions and Logic, delves into the Puppet language and shows how to control flow using conditional statements and logical expressions, and how to build arithmetic and string expressions. It also covers operators, arrays, and hashes.

Chapter 9, Reporting and Troubleshooting, looks at the practical side of working with Puppet: how to diagnose and solve common problems, debugging Puppet's operations, and understanding Puppet error messages.

Chapter 10, Moving on Up, shows you how to make your Puppet code more elegant, more readable, and more maintainable. It offers some links and suggestions for further reading, and outlines a series of practical projects that will help you deliver measurable business value using Puppet.

What you need for this book

You'll need a computer system (preferably, but not essentially, Ubuntu Linux-based) and access to the Internet. You won't need to be a UNIX expert or an experienced sysadmin; I'll assume you can log in, run commands, and edit files, but otherwise I'll explain everything you need as we go.

Who this book is for

This book is aimed at system administrators, developers, and others who need to do system administration, who have grasped the basics of working with the command line, editing files, and so on, but want to learn how to use Puppet to get more done, and make their lives easier.

Conventions

In this book, you will find several headings appearing frequently.

To give clear instructions on how to complete a procedure or task, we use:

Time for action – heading

- 1.** Action 1
- 2.** Action 2
- 3.** Action 3

Instructions often need some extra explanation to make sense, so they are followed with:

What just happened?

This heading explains the working of tasks or instructions that you have just completed.

You will also find some other learning aids in the book, including:

Pop quiz – heading

These are short multiple-choice questions intended to help you test your own understanding.

Have a go hero – heading

These practical challenges give you ideas for experimenting with what you have learned.

You will also find a number of styles of text 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: "To have Puppet read a manifest file and apply it to the server, use the `puppet apply` command."

A block of code is set as follows:

```
file { '/tmp/hello':  
  content => "Hello, world\n",  
}
```

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

```
file { '/tmp/hello':  
  content => "Hello, world\n",  
}
```

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

```
ubuntu@demo:~$ puppet apply site.pp  
Notice: /Stage[main]/Node[demo]/File[/tmp/hello]/ensure: defined content  
as '{md5}bc6e6f16b8a077ef5fbc8d59d0b931b9'  
Notice: Finished catalog run in 0.05 seconds
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "On the **Select Destination Location** screen, click on **Next** to accept the default destination."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.