



Quick answers to common problems

Puppet 3 Cookbook

Build reliable, scalable, secure, and high-performance systems to fully utilize the power of cloud computing

John Arundel

[PACKT] open source*
PUBLISHING community experience distilled

Table of Contents

[Puppet 3 Cookbook](#)

[Credits](#)

[About the Author](#)

[About the Reviewers](#)

[www.PacktPub.com](#)

[Support files, eBooks, discount offers and more](#)

[Why Subscribe?](#)

[Free Access for Packt account holders](#)

[Preface](#)

[What this book covers](#)

[What you need for this book](#)

[Who this book is for](#)

[Conventions](#)

[Reader feedback](#)

[Customer support](#)

[Downloading the example code](#)

[Errata](#)

[Piracy](#)

[Questions](#)

[1. Puppet Infrastructure](#)

[Introduction](#)

[Linux distributions](#)

[Puppet versions](#)

[Installing Puppet](#)

[Getting ready...](#)

[How to do it...](#)

[Creating a manifest](#)

[How to do it...](#)

[Managing your manifests with Git](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Creating a decentralized Puppet architecture](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Writing a puppet script](#)

[How to do it...](#)

[How it works...](#)

[Running Puppet from cron](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Deploying changes with Rake](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Bootstrapping Puppet with Rake](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[Automatic syntax checking with Git hooks](#)

[How to do it...](#)

[How it works...](#)

[2. Puppet Language and Style](#)

[Introduction](#)

[Using community Puppet style](#)

[How to do it...](#)

[Indentation](#)

[Quoting](#)

[Variables](#)

[Parameters](#)

[Symlinks](#)

[There's more...](#)

[Checking your manifests with puppet-lint](#)

[Getting ready](#)

[How to do it...](#)

[There's more...](#)

[See also](#)

[Using modules](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Templates](#)

[Facts, functions, types, and providers](#)

[Autogenerating module layout](#)

[Third party modules](#)

[Module organization](#)

[See also](#)

[Using standard naming conventions](#)

[How to do it...](#)

[There's more...](#)

[Using inline templates](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Iterating over multiple items](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using hashes](#)

[Creating arrays with the split function](#)

[Writing powerful conditional statements](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Elsif branches](#)

[Comparisons](#)

[Combining expressions](#)

[See also](#)

[Using regular expressions in if statements](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Capturing patterns](#)

[Regular expression syntax](#)

[See also](#)

[Using selectors and case statements](#)

[How to do it...](#)

[How it works...](#)

[Selector](#)

[Case statement](#)

[There's more...](#)

[Regular expressions](#)

[Defaults](#)

[Using the in operator](#)

[How to do it...](#)

[There's more...](#)

[Using regular expression substitutions](#)

[How to do it...](#)

[How it works...](#)

[There's more](#)

[See also](#)

[3. Writing Better Manifests](#)

[Introduction](#)

[Using arrays of resources](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Using definitions](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using dependencies](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using tags](#)

[How to do it...](#)

[There's more...](#)

[Using run stages](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using node inheritance](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Passing parameters to classes](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using class inheritance and overriding](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Removing inherited attributes](#)

[Adding extra values using +>](#)

[Disabling resources](#)

[See also](#)

[Writing reusable, cross-platform manifests](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Getting information about the environment](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Importing dynamic information](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Passing arguments to shell commands](#)

[How to do it...](#)

[How it works...](#)

[4. Working with Files and Packages](#)

[Introduction](#)

[Making quick edits to config files](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using Augeas to automatically edit config files](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Building config files using snippets](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using ERB templates](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using array iteration in templates](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using GnuPG to encrypt secrets](#)

[Getting ready](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Installing packages from a third-party repository](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Building packages automatically from source](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Comparing package versions](#)

[How to do it...](#)

[How it works...](#)

[5. Users and Virtual Resources](#)

[Introduction](#)

[Using virtual resources](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing users with virtual resources](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Managing users' SSH access](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing users' customization files](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Efficiently distributing cron jobs](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using schedules to limit when resources can be applied](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using host resources](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using multiple file sources](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Distributing directory trees](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Cleaning up old files](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Auditing resources](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Temporarily disabling resources](#)

[How to do it...](#)

[How it works...](#)

[6. Applications](#)

[Introduction](#)

[Managing Apache servers](#)

[How to do it...](#)

[There's more...](#)

[Creating Apache virtual hosts](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Custom domains and docroots](#)

[Modifying all sites](#)

[See also](#)

[Creating Nginx virtual hosts](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing MySQL](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Creating MySQL databases](#)

[Managing Ruby](#)

[How to do it...](#)

[How it works...](#)

[ruby-build](#)

[rbenv](#)

[There's more...](#)

[7. Servers and Cloud Infrastructure](#)

[Introduction](#)

[Building high-availability services using Heartbeat](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing NFS servers and file shares](#)

[How to do it...](#)

[How it works...](#)

[Using HAProxy to load-balance multiple web servers](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing firewalls with iptables](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing EC2 instances](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Managing virtual machines with Vagrant](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[8. External Tools and the Puppet Ecosystem](#)

[Introduction](#)

[Creating custom facts](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Adding external facts](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Debugging external facts](#)

[Using external facts in Puppet](#)

[See also](#)

[Setting facts as environment variables](#)

[How to do it...](#)

[Importing configuration data with Hieradata](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Setting node-specific data with Hieradata](#)

[Looking up data with Hieradata](#)

[See also](#)

[Storing secret data with hieradata-gpg](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Generating manifests with puppet resource](#)

[How to do it...](#)

[There's more...](#)

[Generating manifests with other tools](#)

[Getting ready...](#)

[How to do it...](#)

[There's more...](#)

[Playing with Pysa](#)

[Conclusion](#)

[Testing your manifests with rspec-puppet](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Using public modules](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Using an external node classifier](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Creating your own resource types](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Documentation](#)

[Validation](#)

[Creating your own providers](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Creating your own functions](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[9. Monitoring, Reporting, and Troubleshooting](#)

[Introduction](#)

[Doing a dry run](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Logging command output](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Logging debug messages](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[Printing out variable values](#)

[Checking when a resource is applied](#)

[Generating reports](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Producing automatic HTML documentation](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Drawing dependency graphs](#)

[Getting ready...](#)

[How to do it...](#)

[How it works...](#)

[There's more...](#)

[See also](#)

[Understanding Puppet errors](#)

[How to do it...](#)

[See also](#)

[Inspecting configuration settings](#)

[How to do it...](#)

[How it works...](#)

[See also](#)

[Index](#)

Puppet 3 Cookbook

Puppet 3 Cookbook

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 authors, 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: October 2011

Second published: August 2013

Production Reference: 1190813

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78216-976-5

www.packtpub.com

Cover Image by Asher Wishkerman (<wishkerman@hotmail.com>)

Credits

Author

John Arundel

Reviewers

Dhruv Ahuja

Carlos Nilton Araújo Corrêa

Daniele Sluijters

Dao Thomas

Acquisition Editor

Kartikey Pandey

Lead Technical Editor

Madhuja Chaudhari

Technical Editors

Anita Nayak

Larissa Pinto

Project Coordinator

Kranti Berde

Proofreader

Lawrence A. Herman

Indexers

Hemangini Bari

Monica Ajmera Mehta

Graphics

Ronak Dhruv

Production Coordinator

Kyle Albuquerque

Cover Work

Kyle Albuquerque

About the Author

John Arundel is a devops consultant, which means he solves difficult problems for a living. (He doesn't get called in for easy problems.)

He has worked in the tech industry for 20 years, and during that time has done wrong (or seen done wrong) almost everything that you can do wrong with computers. That comprehensive knowledge of what not to do, he feels, is one of his greatest assets as a consultant. He is still adding to it.

He likes writing books, especially about Puppet (*The Puppet 3 Beginner's Guide* is available from the same publisher). It seems that at least some people enjoy reading them. He also provides training and coaching on Puppet, which it turns out is 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](#).

My thanks go to Rene Lehmann, Cristian Leonte, German Rodriguez, Keiran Sweet, Dean Wilson, and Dan White for their help with proofreading and suggestions. Special thanks are due to Lance Murray and Sebastiaan van Steenis, who diligently read and tested every chapter, and provided invaluable feedback on what did and didn't make sense.

About the Reviewers

Dhruv Ahuja is a Lead Engineer at a managed hosting provider. He specializes in infrastructure solutions design and configuration, with a keen eye on achieving mechanical sympathy. His first brush with Puppet was in 2011 when he developed a solution on dynamically scaling compute nodes for a multi-purpose grid platform. He also holds a master's degree in Advanced Software Engineering from King's College London, and won the Red Hat UK Channel Consultant of the Year award in 2012 for delivering progressive solutions. A long history in conventional software development and traditional systems administration equip him with aptness in both areas, and he bridges many infrastructural gaps in a well-delimited way. In this era of infrastructure as code, he believes that declarative abstraction is essential for a maintainable systems life-cycle process.

Carlos N. A. Corrêa is an IT Operations manager and consultant, and is also a Puppet enthusiast and an old-school Linux hacker. He has a Master's Degree in Systems Virtualization and holds CISSP and RHCE certifications. Backed by a 15-year career on systems administration, Carlos leads IT Operations teams for companies both in Brazil and Africa. He is also a part-time professor for graduate and undergraduate courses in Brazil. Carlos co-authored several research papers on network virtualization and OpenFlow, presented on peer-reviewed IEEE and ACM conferences worldwide.

I thank God for all the opportunities of hard work and all the lovely people I always found on my way. To the sweetest of them all, my wife Nanda, I give thanks for all the loving care and support that pushes me forward. And to my parents, Nilton and Zélia, for being such a big inspiration for all the things that I do.

Daniele Sluijters is a student of Informatics and has been working as a systems operator for a few years. Initially it all started out as a hobby, but eventually it turned into both his field of study and work. His primary focus in both work and study for the past years have been large(r) networks made up of mostly Unix systems offering services to the world disclosed over the internet and how to manage and secure both the systems, the services they provide and the networks they use. He has also worked on the book Zabbix Network Monitoring Essentials, Munin Plugin Starter

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>](mailto: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.

Preface

A revolution is underway in the field of IT operations. The new generation of configuration management tools can build servers in seconds and automate your entire network. Tools such as Puppet are essential to take full advantage of the power of cloud computing, and build reliable, scalable, secure, high-performance systems.

This book takes you beyond the basics and explores the full power of Puppet, showing you in detail how to tackle a variety of real-world problems and applications. At every step, it shows you exactly what commands you need to type and includes complete code samples for every recipe.

It takes the reader from rudimentary knowledge of Puppet to a more complete and expert understanding of Puppet's latest and most advanced features, community best practices, writing great manifests, scaling and performance, and how to extend Puppet by adding your own providers and resources.

This book also includes real examples from production systems and techniques that are in use in some of the world's largest Puppet installations. It will show you different ways to do things using Puppet and point out some of the pros and cons of these approaches.

The book is structured so that you can dip in at any point and try out a recipe without having to work your way through from cover to cover. You'll find links and references to more information on every topic, so that you can explore further for yourself. Whatever your level of Puppet experience, there's something for you, from simple workflow tips to advanced, high-performance Puppet architectures.

I've tried hard to write the kind of book that would be useful to me in my day-to-day work as a devops consultant. I hope it will inspire you to learn, to experiment, and to come up with your own new ideas in this exciting and fast-moving field.

What this book covers

You'll find the following chapters in this book:

[Chapter 1](#), *Puppet Infrastructure*, shows how to set up Puppet for the first time, including instructions on installing Puppet, creating your first manifests, using version control with Puppet, building a distributed Puppet architecture based on Git, writing a script to apply Puppet manifests, running Puppet automatically, using Rake to bootstrap machines and deploy changes, and using Git hooks to automatically syntax-check your manifests.

[Chapter 2](#), *Puppet Language and Style*, covers aspects of writing good Puppet code, including using Puppet community style, checking your manifests with `puppet-lint`, structuring your manifests with modules, using standard naming and style conventions, using inline templates using iteration, conditional statements, and regular expressions, using selectors and case statements, and string operations.

[Chapter 3](#), *Writing Better Manifests*, goes into detail on specific features of Puppet that you can use to improve your code quality and usability, including arrays, definitions, ordering your resources with dependencies, inheriting from nodes and classes, passing parameters to classes, overriding parameters, reading information from the environment, writing reusable manifests, and using tags and run stages.

[Chapter 4](#), *Working with Files and Packages*, deals with some of the most common sysadmin tasks, including managing config files, using Augeas, generating files from snippets and templates, managing third-party package repositories, using GnuPG to encrypt secret data in Puppet, and building packages from source.

[Chapter 5](#), *Users and Virtual Resources*, explains what virtual resources are and how they can help you manage different combinations of users and packages on different machines, and shows you how to use Puppet's resource scheduling and auditing features.

[Chapter 6](#), *Applications*, focuses on some specific applications that you may need to manage with Puppet, including complete recipes for Apache and Nginx, MySQL, and Ruby.

[Chapter 7](#), *Servers and Cloud Infrastructure*, extends the power of Puppet to manage virtual machines, both on the cloud and on your desktop, with recipes for Vagrant and EC2 instances. It also shows you how to set up load balancing with HAProxy, firewalls with `iptables`, network filesystems with NFS, and high-availability services with Heartbeat.

[Chapter 8](#), *External Tools and the Puppet Ecosystem*, looks at some of the tools that have grown up around Puppet, including Hiera, Facter, and `rspec-puppet`. It also introduces you to some advanced topics including writing your own resource types, providers, and external node classifiers.

[Chapter 9](#), *Monitoring, Reporting, and Troubleshooting*, covers ways that Puppet can report

information about what it's doing, and the status of your system. This includes reports, log, and debug messages, dependency graphing, testing and dry-running your manifests, and a guide to some of Puppet's more common error messages.

What you need for this book

To run the examples in this book, you will need a computer with Ubuntu Linux 12.04 and an Internet connection.

Who this book is for

The book assumes that the reader has a little experience of Linux systems administration, including familiarity with the command line, file system, and text editing. No programming experience is required.

sample content of Puppet 3 Cookbook

- [click Tzili online](#)
- [Round Ireland in Low Gear pdf, azw \(kindle\), epub, doc, mobi](#)
- [read online Pro JavaScript Techniques \(2nd Edition\) for free](#)
- [read online Sew with Confidence: A Beginner's Guide to Basic Sewing pdf, azw \(kindle\), epub, doc, mobi](#)

- <http://weddingcellist.com/lib/Main-Street--Barnes---Noble-Classics-Series-.pdf>
- <http://berttrotman.com/library/Alcatraz--History-and-Design-of-an-Icon.pdf>
- <http://chelseaprintandpublishing.com/?freebooks/End-of-the-World-in-Breslau.pdf>
- <http://korplast.gr/lib/Sew-with-Confidence--A-Beginner-s-Guide-to-Basic-Sewing.pdf>