Pulling Strings with Puppet

**Puppet** is an incredibly popular Ruby-based configuration management tool. Gaining its popularity with the open-source edition, it has become popular enough to also appear as a Puppet PE – Puppet Enterprise edition.

**Update February 2019:** I’m migrating book reviews from another website of mine, so this isn’t a recent review but still a great book I think you should read if you get a chance.

*Pulling Strings with Puppet* is now a classic: written 10 years ago, it’s still a really good introduction to configuration management with Puppet. James Turnbull has since written an
even better book (*Pro Puppet*) and I’ll probably find a few even more recent introductions to Puppet, but if you have *Pulling Strings with Puppet* on your colleague’s desk I suggest you borrow it.

**What is Puppet?**

Puppet is a framework for automatic configuration management of your systems. Originally oriented an Unix systems and servers specifically, it can now be used to manage quite a range of Unix-like systems as well as Windows environments.

Both open source and enterprise (paid) edition allow you to greatly optimize and automate the process of deploying and maintaining configurations of your environments via a scriptable core that can be configured to suit all of your needs.

Puppet has 400+ configurations already tested and available online on [Puppet Forge](http) – this is basically a collection of downloadable modules for automating all sorts of systems administration tasks.

The *Pulling Strings with Puppet* book gives a very good introduction into common tasks of configuration management and explains the multiple layers of Puppet-based solution very well: there’s a declarative language used for defining classes and modules, then there’s a transaction layer for creating and deploying updates and, finally, there is a resource abstraction layer – this piece of magic that makes it possible to use the same configuration stanza for deploying the same change to vastly different Unix-like distros.

**Deploying a Puppet infrastructure**

Puppet servers are called *puppet masters*. These are the servers which store all configurations along with list of Puppet clients (sometimes called *Puppet nodes*). Clients
communicate back to Puppet servers at regular intervals using RESTful API over HTTPS, checking in and downloading configurational updates if necessary.

Self-signed certificates are a requirement for all the clients and the way these are managed on the Puppet master side ensures that no unauthorized access can be gained easily enough. You will find a good enough description of how this works in practice so it will be very easy to get started if you’re new to Puppet.

The book goes into lots of detail when it comes to explaining the pre-requisites (Ruby, Facter) and installation process (compiling from sources or deploying from packages). The easiest way to deploy a Puppet client or server is probably to get Ruby installed and then deploy a Puppet Ruby gem.

I really liked the table with descriptions of all the common Puppet management commands – it’s a neat little reference giving you a good idea of what’s possible and where to look for more information once you become comfortable enough.

**Using Puppet for managing configurations**

It all starts with the [main] configuration namespace, and before you know it the book effortlessly takes you through configuring resources and attributes and deploying classes and modules. Puppet relies heavily on the Facter framework which is an abstract way of documenting, configuring and presenting useful configuration elements about your environments like OS version or name of your Linux distro. The book gives a very useful description of the approach to using facts and configurative definitions.

A good few pages are used to make sure you will have the full understanding of what variables are possible in Puppet and how variable scoping is probably different from most of the scripting and programming languages you already know.
You will learn about virtual resources and how they need to be realized before the changes are actually applied, and finally will be introduced to various default types available in a Puppet installation (cron/exec/file/filebucket/group/package/yumrepo and quite a few others).

**Sample Puppet configuration management environment**

Perhaps the most valuable element of the whole book is a complete description of nodes, classes, users and groups needed to deploy and support a typical LAMP environment. In addition to sample configurations for managing nodes and users, you are given full code for modules managing MySQL, Apache and Postfix.

The language used for composing classes and modules in Puppet is pretty straightforward, but the complete examples will help you not only learn the syntax but also pick up some of the best practices when it comes to starting even the simplest of Linux configuration management environments.

For the most technically curious minds there are hints for deploying custom set of Facter facts and even if that doesn’t impress enough you are given instructions for creating your own type for the most flexible resource/configuration management.

**Advanced Puppet usage**

The last few chapters in the Pulling Strings with Puppet book show you some more advanced challenges you’ll face if you chose to explore Puppet in depth.

In addition to getting plenty of hints for performance optimisation and scaling out of your Puppet infrastructure, you will find information about migrating nodes description
information into an external storage (scripts or indeed proper datastores like LDAP).

For scaling, you’ll get basic information about Mongrel (apparently it’s a Ruby friendly webserver you can use instead of Apache) but also full configuration examples for configuring Apache as a proxy and a load balancing solution pointing to multiple Puppet instances.

Even if you are well versed in mod_proxy or mod_ajp, reading the chapter about Puppet scalability with Mongrel and Apache proxy will be a time well spent — it doesn’t just show you what steps are needed for the desired configuration but gives you explanations of the typical challenges you’ll be trying to solve.

**Summary for the Pulling Strings with Puppet book**

I had thoroughly enjoyed this book the first time I had read it back in 2010 and I can still recommend it after reading it again in 2012. Majority of the topics in the book are still quite relevant and easily applicable even today, which is probably a testament to both the talents of the book author and the great planning/roadmapping of the founding geniuses behind the Puppet framework.
Pulling Strings with Puppet