Puppet, a configuration management tool

Configuration management tool to facilitate managing our (Linux) cluster head nodes, other Linux servers, and our Linux desktops. Works with other OSes (MacOS, Windows), but we'll first focus on Linux.

Puppet

http://docs.puppetlabs.com/#puppetpuppet

Watch the under 4 minute video on this page:

http://info.puppetlabs.com/download-learning-puppet-VM.html

Excerpts from "Puppet 3 Beginner's Guide", by John Arundel (April 17, 2013):

Configuration management tools: This is roughly the situation we have now. Several different CM systems have been developed over the years, with new ones coming along all the time, but only a few have achieved significant market share. At the time of writing, at least for UNIX-like systems, these CM systems are Puppet, Chef, and CFEngine. There really isn't much to choose between these different systems. They all solve more or less the same problems - the ones we saw earlier in this chapter - in more or less the same way. Some people prefer the Puppet way of doing things; some people are more comfortable with Chef, and so on. But essentially, these, and many other CM systems, are all great solutions to the CM problem, and it's not very important which one you choose as long as you choose one.

Access the full book for free via CU Library's Safari subscription:

http://proquest.safaribooksonline.com/book/operating-systems-and-server-administration/virtualization/9781782161240/puppet-3-beginner-s-guide/pr06_html?uicode=cornell

NOTE: Read many ebooks for free using CU Library's access. Get CUL Passkey to facilitate access to cited books from off-campus. See details on this page's parent page.

Infrastructure as code: Once we start writing programs to configure machines, we get some benefits right away. We can adopt the tools and techniques that regular programmers - who write code in Ruby or Java, for example - have used for years:

- Powerful editing and refactoring tools
- Version control
- Tests
- Pair programming
- Code reviews

This can make us more agile and flexible as system administrators, able to deal with fast-changing requirements and deliver things quickly to the business. We can also produce higher-quality, more reliable work.

Buzzword compliance

Declarative system

Puppet is said to be a declarative system. See "idempotence", below.

Imperative system

Chef is said to be an imperative system.

Idempotence

Denoting an element of a set that is unchanged in value when multiplied or otherwise operated on by itself.

https://www.google.com/search?q=define+idempotent

http://en.wikipedia.org/wiki/Idempotence

Paraphrasing ansibleworks.com's web site: An “idempotent” resource model describes the desired state of computer systems and services, not the paths to get them to this state. Thus, no matter what state a system is in, this type of system understands how to transform it to the desired state (some also support a “dry run” mode to preview needed changes). This allows reliable and repeatable IT infrastructure configuration, avoiding the potential failures from scripting and script-based solutions that describe explicit and often irreversible actions rather than the end goal.

DevOps

- http://en.wikipedia.org/wiki/DevOps
- From "Puppet 3 Beginner's Guide", cited above:
    - **Devops** write code, herd servers, build apps, scale systems, analyze outages, and fix bugs. With the advent of CM (configuration management) systems, devs and ops are now all just people who work with code.

Systems management

One of the functions of systems management is configuration management.

- http://en.wikipedia.org/wiki/Systems_management

Artisan server crafting

From "Puppet 3 Beginner's Guide", cited above:
So the first problem with building servers by hand (**artisan server crafting**, as it's been called) is that it's complicated and tedious and it takes a long time. There's another problem. The next time you need to build an identical server, how do you do it, especially since your painstaking notes will no longer be up to date with reality.

Other configuration management tools (vs. Puppet)

- http://java.dzone.com/articles/comparing-flavors-config
- http://bitfieldconsulting.com/puppet-vs-chef
- http://www.opscode.com/chef/#which-chef
- https://cfengine.com/what-is-cfengine
- https://github.com/benhoskings/babushka
- http://www.ansibleworks.com/configuration-management/
- http://en.wikipedia.org/wiki/Configuration_management
    - http://en.wikipedia.org/wiki/Comparison_of_open_source_configuration_management_software

From one person's comment to the "versus" article<http://bitfieldconsulting.com/puppet-vs-chef> above:
Chef will draw in Ruby developers because it’s not declarative, and because it’s easy.
My experience is that most developers don’t do declarative systems. Everyday languages are imperative, and when you’re a developer looking to get something deployed quickly, you’re most likely to pick the tool that suits your world view.
Systems Administrators tend to use more declarative tools (make, etc.)
Developers and Systems Administrators also have a divergent set of incentives. Developers are generally rewarded for delivering systems quickly, and SA’s are rewarded for stability. IMHO, Chef is a tool to roll out something quickly, and Puppet is the one to manage the full lifecycle. That’s why I think Chef makes a good fit for cloud deployment because Vm instances have a short lifespan.

- Chef will draw in Ruby developers because it’s not declarative, and because it’s easy.
  My experience is that most developers don’t do declarative systems. Everyday languages are imperative, and when you’re a developer looking to get something deployed quickly, you’re most likely to pick the tool that suits your world view.
  Systems Administrators tend to use more declarative tools (make, etc.)
  Developers and Systems Administrators also have a divergent set of incentives. Developers are generally rewarded for delivering systems quickly, and SA’s are rewarded for stability. IMHO, Chef is a tool to roll out something quickly, and Puppet is the one to manage the full lifecycle. That’s why I think Chef makes a good fit for cloud deployment because Vm instances have a short lifespan.