NWEA code sample solution

Prerequisites:

Working Puppet server and client

Virtual host allowed on DNS server

Puppet client fqdn: cody.home

Virtual host: doggie.cody.home

Configuration:

Create firewall rule:

sudo ufw allow 8888/tcp

On puppet server, install nginx module:

sudo puppet module install jfryman-nginx

Add code to site.pp on puppet server (see file)

Create index.html at /etc/puppet/modules/nginx-content/files on puppet server

Run “sudo puppet agent –test” on client.

Point browser at [http://doggie.cody.home:8888](http://doggie.cody.home:8888/)

Observe test output (see screen shot).

**Why did you choose this solution?**

My most recent experience with system management tools was with Puppet, so it was natural to leverage that expertise. The jfryman-nginx module was used because the “official” puppet module for nginx has been deprecated. This exercise was done on Linux Mint (Ubuntu) because I happened to have two laptops running Mint for another project. I have CentOS up on another machine, but it's an old version running legacy software (on top of EXSi) and I haven't stood up its replacement yet.

**What is the best/worst part of your solution?**

Best: The solution is very simple, only requiring a few lines of code. Changing the content served by nginx can be done on the Puppet server by altering the contents of /etc/puppet/modules/nginx-content/files. The solution lends itself to source code control, as all the files are in one place.

Worst: Port 8888 was opened by hand on the client. Although it was out of scope, I would have liked to configure iptables via puppet.

**Why would automating a task like this be helpful?**

Automation of this type easily lends itself to installing the same solution on multiple Puppet clients, or adding clients at a later time. (This is important in a webapps environment where servers may come and go in the cluster.) The need to replicate work is avoided. Configuration for multiple servers is in text files in a central location, making source code control straightforward. A solution of this type helps to insure that configuration is in sync across multiple servers.