**Ansible Essentials Key Takeaways**

*Setting up the Test Environment:*

* Vagrant Commands:
  + vagrant up (to bring up a vagrant machine when a VagrantFile already exists)
  + vagrant ssh (to ssh into a vagrant machine. Note: supply box name when VagrantFile contains more than one VM)
  + vagrant suspend (to power down a vagrant machine)
  + vagrant destroy –f (to terminate a vagrant machine without additional prompts)
* Configuration for Ansible:
  + Generate an ssh key to use for authentication. Our training is using an insecure private key generated by Vagrant creator.
  + chmod 400 /path/to/file
  + Set private\_key\_file = /path/to/file in ansible.cfg for private key authentication.
  + Uncomment host\_key\_checking = False in ansible.cfg to connect to machines without prompts if you haven’t previously connected to them.
* Inventory:
  + Host file in ini format with connecting user variable:

[wordpress-server]

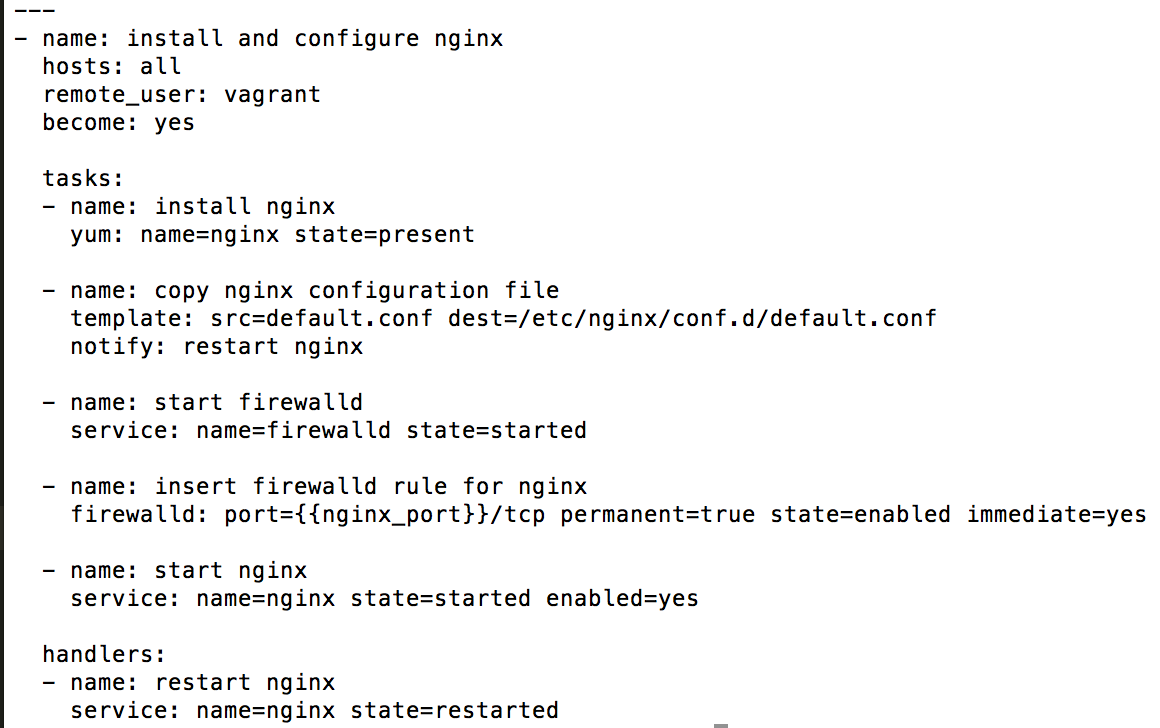
10.42.0.6 ansible\_user=vagrant

*Ad-Hoc Commands:*

* ansible all –i hosts –m ping (to test connectivity)
* ansible all –i hosts –m setup (to gather facts)
* ansible all –i hosts –m yum –a “name=nginx state=present” –b (to install nginx on target machines)

*Basic Playbook:*

* Install and configure nginx playbook:

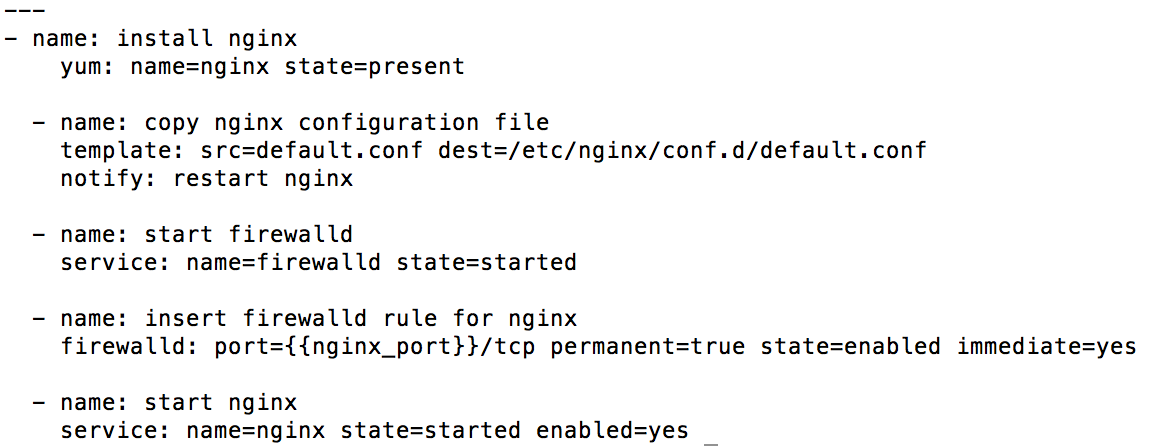


*Creating Role Structure:*

* mkdir roles (create a roles directory to hold our roles)
* cd roles/
* ansible-galaxy init nginx (ansible galaxy command that creates the roles directories automatically).

*Breaking Basic Playbook into a Role:*

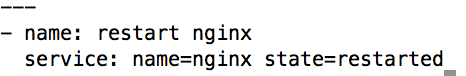
* Nginx tasks file:



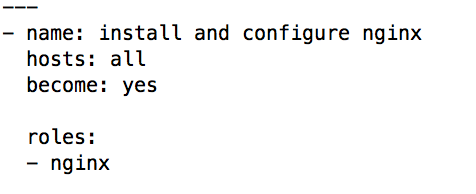
* Nginx template file:



* Nginx handlers file:



* Site.yml:



*Ansible-playbook:*

* ansible-playbook –i hosts site.yml