BCNET

Shared IT Services for Higher Education & Research

Conference 2018

Welcome!

Using Ansible to Provision Web Servers and Install Wordpress

About Me + Resources

Scott Robarts
Web Server Administrator, Capilano University
scottrobarts@capilanou.ca

Github: srobarts
Twitter: srobarts
Instagram: srobarts

Notes and Resources:

https://github.com/srobarts/bcnet-ansible-presentation

What is Ansible?

"Configuration management for humans"

Key pluses:

1. Easy install
2. SSH for remote management – simple, built-in, fast
3. 300+ built in modules
4. Low infrastructure – just Ansible, SSH, and your playbooks
5. Agent free

What is Ansible, continued …

Free and open-source (GNU Public License)

Purchased by Redhat in 2015

Written in Python. (Also some Powershell, for Windows)

Command line based, but also there are GUI tools:

- Ansible Tower – paid – www.ansible.com
- AWX – open source - https://github.com/ansible/awx

More information in general: www.ansible.com

Ansible's competitors …

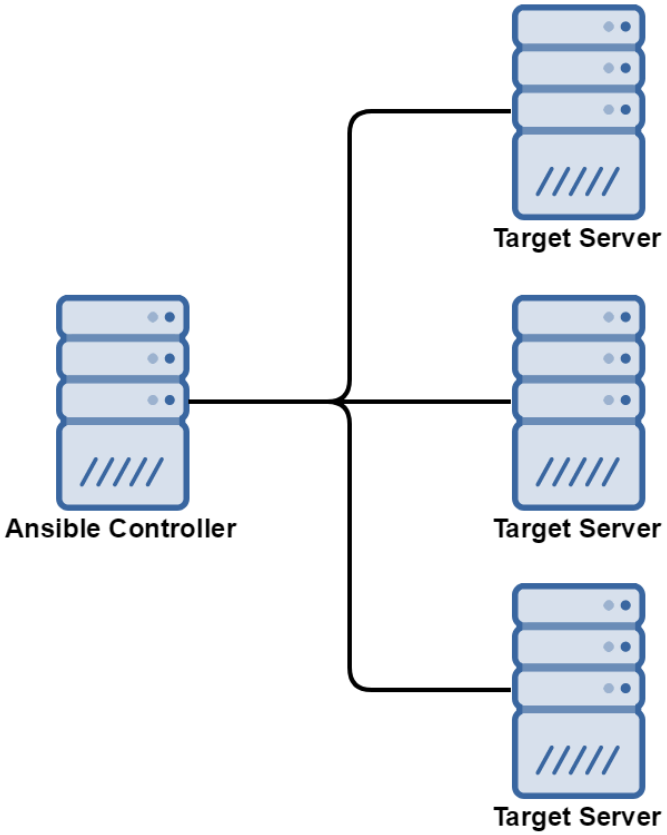
Both Chef and Puppet do similar things to Ansible.

Some advantages – i.e. they monitor state of servers,
to maintain desired state configuration

Disadvantages:

- Require an agent to be installed on servers
- Are more complex
- May be open-source, but more advanced config costs $$

Ansible architecture



Controller can be a dedicated server, or your laptop,
It only needs Ansible installed

Target Servers can be:

- Web servers
- Database servers
- Network devices
- Linux servers
- Windows servers

Connection is via SSH,
or WinRM for Windows

Ansible does not need to be installed on target servers

Ansible installation

Debian/Ubuntu:

```
sudo apt-get-repository ppa:ansible/ansible
sudo apt-get update
sudo apt-get install ansible
```

Redhat/CentOS:

```
sudo yum install ansible
```

Mac:

```
brew install ansible
```

Four main parts:

1. **Inventory:** Describe & list your infrastructure
2. **Ad-Hoc Commands**: One-off tasks
3. **Playbooks**: Task orchestration, “infrastructure as code”
4. **Roles**: Configuration encapsulation

Inventory (the hosts file)

Used for describing, listing and group your infrastructure.

Located by default at /etc/ansible/hosts

HOSTS DEMO

Connecting to target servers

Ansible User

- By default Ansible will use SSH
- Best practice for security is to disable SSH login of root user
- Instead create an Ansible user on servers
- Ansible user will be able to login and sudo

Key-based SSH Login

- We need to create a private and public key on our Ansible controller
- Then share the public key with our target servers

Idempotence

- A key strength of Ansible
- Run commands over and over again, without doing things over and over again
- Ansible checks **Facts** about the server, **before running Tasks**
- Facts are used to find the state of the server
- **Desired State Configuration**

Snippet:


```
"name=vim state=present"
```

– We tell Ansible that we want VIM to be present on a server, not that we want to install it.

Basic Playbooks

Running ad-hoc commands is not very powerful

Better to group Tasks into a playbook

Playbooks encapsulate Tasks, Handlers, Files, Templates.

<<Playbook Example>>

Playbooks - Roles

Roles are a way of encapsulating playbook functions

In the examples I will show (for installing Wordpress), we will have the following roles:

Common
Apache
PHP
MySQL
Wordpress

Playbooks - concepts

Handlers:

- Basically a task, and can do everything a task can do, but will only be run when called by another task
  << example >>

Variables:

- Ansible allows you to use variables in playbooks. In this way we can have one location to maintain variables, to be used across playbooks.
  << example >>

Playbooks - concepts

Templates:

- Ansible allows you to create templates using the Jinja2 templating engine. These templates should have the .j2 extension.

  << example >>

Advanced Playbooks

Advanced Playbook Concepts and Structure

Playbook folders:

(you can have some, or all of these)

- files/
- handlers/
- meta/
- templates/
- tasks/
- vars/

WP-CLI

WP-CLI is awesome (if you're a geek)

WP-CLI is a command line interface for Wordpress.

A few things you can do:

- Update Wordpress
- Install and active plugins
- Install and activate themes
- Manage users
- Administer Wordpress multisite
- Scaffold new sites
- Work with media
- Perform basic database operations

WP-CLI Example Commands

wp install plugin user-switching –activate

wp install theme twenty-sixteen –activate

wp theme list --status=inactive

<< demo >>

Pulling it all together

One playbook to install LAMP stack and Wordpress.

Playbook will use **WP-CLI** to handle some Wordpress related work.

Playbook will be divided into **Roles** to encapsulate tasks.

Variables will be used to share values across playbook.

Templates will be used to scaffold configuration files.