Implementing Ansible for Configuration Management

Andrew Mallett

LINUX AUTHOR AND TRAINER

@theurbanpenguin www.theurbanpenguin.com

Objectives

Understanding Ansible

- Project owned by Red Hat
- Serverless / Clientless
  - SSH
  - WinRM
- Python based
  - Install from distribution repositories
  - Install from Python repositories
- Configuration / inventory
- Configuring nodes
  - Ad-hoc
  - Playbooks

```
elif _operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    #selection at the end -add back the deselected mirror modifier
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
    #mirror_ob.select = 0
```

Ansible

- Project now owned by Red Hat
- Python based
- Only needed to be installed on the node known as the controller
- Uses SSH to connect to Linux
- WinRM used to connect to Windows managed nodes
- Python is required on Linux managed nodes
- PowerShell on Windows nodes
- Playbooks created in YAML

No Server - No Client

With no long running service requirement for the Ansible controller or clients, ansible make for a very quick and easy setup. Being able to manage systems within minutes.

```
$ sudo apt update
$ sudo apt install ansible sshpass
$ ansible --version
$ ansible localhost -m ping
```

Installing Ansible

Ansible can be installed as package from standard repositories in Ubuntu, or in Red Hat based variants, from EPEL. It is also possible to install using pip/pip3 from the Python repos, giving you the latest version. We install Ansible on the Ubuntu host as it has a later version of Python. The sshpass utility allows for password caching.

```
$ ansible-doc -l
$ ansible-doc ping
$ ansible-doc -t keyword -l
```

Ansible Documentation

As well as docs.ansible.com, extensive help is available from the command line.

Demo

Working on the Ubuntu System:

- Install ansible and sshpass
- Check version
- Check ad-hoc commands with the built-in localhost node
- Use CLI documentation

```
$ mkdir ~/ansible ; cd ~/ansible
$ vim ansible.cfg
[defaults]
remote_user = vagrant
inventory = inventory
host_key_checking = false
$ ansible-config dump --only-changed
```

Configuring Ansible

Ansible works with the localhost without configuration, but it is normal to create a configuration file and inventory. First, let's look at the configuration.

```
$ vim inventory
[redhat]
192.168.56.11
[suse]
192.168.56.12
[ubuntu]
192.168.56.13 ansible_connection=local
$ ansible-inventory --list --yaml
```

Configuring an Ansible Inventory

The very basics of an inventory can be entirely created in the INI format. Other more complex inventories can be created to use YAML variables from directories.

```
$ cd ~/ansible
$ ansible all -km ping
$ ansible all -kbm package -a "name=tree state=present"
```

Testing Ansible Using Ad-Hoc Commands

We can now manage all hosts in the inventory. Although they are all Linux, they would use 3 different package managers, but we do not need to concern ourselves with that using Ansible.

-k: prompt for ssh password -b: elevate privileges -a: module arguments

Demo

Configuring Ansible on the Controller

- ansible.cfg
- inventory
- inventory variables
- testing ansible

```
$ cd ~/ansible ; vim playbook.yml
---
- name: my first play
  hosts: all
  become: true
  tasks:
  - name: my first task
    package:
      name: tree
      state: present
```

Simple Playbook

Whilst ad-hoc commands are good for quick configuration, they are not so good for repeatable configurations. Using playbooks, we can record the desired state of a system or systems and ensure this in a repeatable manner.

```
$ cd ~/ansible
$ ansible-playbook playbook.yml --check-syntax
$ ansible-playbook playbook.yml -C
$ ansible-playbook playbook.yml --list-tasks
$ ansible-playbook playbook.yml
```

Running Playbooks

We have the ansible-playbook command to execute the files. Various checks can be run on the file if required.

Demo

Writing YAML Ansible Playbooks:

- Create playbook
- Test playbook
- Execute playbook

Summary

Understanding Ansible

- Only need to install on controller
- No services
- Python based
- Use SSH to connect to Linux
- Create ansible.cfg in working directory
- Inventory is the managed nodes list
- Use ansible command for ad-hoc configuration
- Use ansible-playbook for repeatable configuration

Implementing Puppet for Configuration Management