INTRODUCTION TO ANSIBLE

Tomas Schlosser

ABOUT ME

- Senior Quality Engineer in Red Hat
- Work as xPaaS QE technical lead
- Working extensively with Ansible for over a year
    - authored OpenShift 2 installer (internal)
    - authored OpenShift 3 installer extending official one (internal)

CONTENT

- What is Ansible
- Requirements
- Inventory
- Ad-hoc commands
  - with demo
- Basic playbooks
  - with demo
- Complex playbooks
  - with demo
- Best practices
- Links

WHAT IS ANSIBLE

Ansible is an extra-simple tool/framework/API for doing 'remote things' over SSH.

Ansible manual page

- Powerful configuration/deployment/orchestration tool
- Uses declarative approach
- Reusable building blocks (modules)
- Idempotent core modules

REQUIREMENTS

- On machine running Ansible
  - Ansible
  - Python 2.6 or 2.7
  - OpenSSH client
- On managed machine
  - Running SSH server
  - Python 2.4 or higher (not Python 3)
    - python-simplejson
    - libselinux-python

INVENTORY

- List of managed machines
- Can be split into groups
- Can contain variables for each node
- Dynamic inventory (Cobbler, EC2)

```
# list of managed servers
172.17.0.3
node1.example.com

[master]
master.example.com  node_username=admin

[nodes]
node[1:5].example.com
172.17.0.4  node_name=node6

[nodes:vars]
node_username=node
```

AD-HOC COMMANDS

- Useful for simple commands

```
# print information about all machines
ansible -i inventory all -m setup

# ping the machines in group nodes
ansible -i inventory nodes -m ping

# update all machines (requires dnf on each machine)
ansible -i inventory all -m dnf -a "name=* state=latest"

# update all machines (Ansible 2.0 feature)
ansible -i inventory all -m package -a "name=* state=latest"

# reboot node machines
ansible -i inventory nodes -a "shutdown --reboot now"
```

BASIC PLAYBOOKS

- Used with multi-step tasks

```
---
- hosts: master
  remote_user: root
  tasks:
    - name: Install java
      dnf: name=java-1.8.0-openjdk-devel state=present
    - name: Download Wildfly
      get_url: url=http://server.com/wildfly/wildfly-9.0.2.Final.zip
               dest=/tmp/wildfly9.zip
    - name: Unzip Wildfly
      unarchive: src=/tmp/wildfly9.zip dest=/opt copy=no
                 creates=/opt/wildfly-9.0.2.Final
```

COMPLEX PLAYBOOKS

- Use for complex tasks
- More structured
  - directories for roles with files and templates
- Roles can be shared between playbooks

```
playbook.yaml
roles/
|- role-name/
|   |- tasks/
|   |   \- main.yaml
|   |- files/
|   |- templates/
|   |- handlers/
|   |- vars/
|   |- defaults/
|   \- meta/
\- role2/
    \- tasks/
        \- main.yaml
```

BEST PRACTICES

- Create idempotent (replayable) playbooks
- Use comments, include sample inventory file
- Use tasks in playbook only for simple tasks, for everything else, separate the tasks to roles
- Keep it simple

You can find more tips in Ansible documentation.

LINKS

- Ansible documentation (especially module index)
  - http://docs.ansible.com/ansible/index.html
  - http://docs.ansible.com/ansible/modules_by_category.html
- This demo
  - https://github.com/tomason/ansible-demo
- This presentation
  - http://redhat.slides.com/tschloss-1/introduction-to-ansible?token=g6B1P4jx