Using Shell Commands in Playbooks

Andrew Mallett
Linux Author and Trainer

@theurbanpenguin www.theurbanpenguin.com

Overview

Ansible and Idempotency

- Ansible modules and idempotency
- Running shell commands when modules not available
  - shell
  - command
  - raw
  - script
  - win_command
- Using logic to control execution

Ansible Modules

Using Ansible modules is preferred rather than using native commands. A module will be idempotent, shell commands will not have the idempotency built in

Command vs Shell Modules

command

The Ansible command module allows execution of the native Linux commands without the need of creating a parent shell. This does not allow access to variables and stream operators

shell

The Ansible shell module opens the shell /bin/sh to execute the commands. This allows access to variables and stream operators but is considered less secure

```
$ MYVAR='myval;ls -l /etc/hosts'

$ ansible rhel -m shell -a "echo $MYVAR"

192.168.33.11 | CHANGED | rc=0 >>
myval
-rw-r--r--. 1 root root 188 Dec 29  2020 /etc/hosts

$ ansible rhel -m command -a "echo $MYVAR"

192.168.33.11 | CHANGED | rc=0 >>
myval;ls -l /etc/hosts
```

Use command Over shell

The command module is considered more secure as it can prevent injection of commands via variables and strategically place semi-colons

Demo

Protecting against command injection via variables:

- command module
- shell module

Ansible Raw Module

The command and shell modules still make use of Python on the managed device to execute the command or open the shell. If Python is not already installed on the managed device, the raw module executes the command directory via SSH and can allow for installation of Python, or even managing network devices that don't allow for Python

```
$ ansible all -i 18.172.45.3 -b -m raw -a "dnf install -y python3"
```

Using Ansible's raw Module

Using the raw module, we can execute commands on a host where SSH execution is allowed. Using AWS's CentOS image, Python is not installed. This means we first need to install Python before continuing to configure the system with Ansible

Demo

Using the Ansible raw module

```
#!/bin/sh
echo "Hello World!"

$ ansible all -m script -a "/home/vagrant/myscript.sh"
```

Using the Ansible script Module

Using the script module, we can create a script on the Ansible controller and have the script execute on the remote nodes, without the script needing to exist on the remote nodes. This is useful where the commands that need to be executed are more complex

Demo

In this demonstration we create local script and see it executed on each node

Idempotency and Ansible

Ansible module are designed to check if they need to execute before they do anything. Commands will not necessarily do this and can create errors. To maintain idempotency we need to build checks into Playbooks

```
$ mkdir -p ~/ansible/loop; cd ~/ansible/loop

$ vim loopdevice.yaml
- name: 'Manage Disk File'
  hosts: rhel
  become: true
  gather_facts: false
  tasks:
    - name: "Create Raw Disk File"
      command:
        cmd: 'fallocate -l 1G /root/disk0'
        creates: '/root/disk0'
```

Controlling Command Execution

Using the meta-parameter creates, the task only runs is the file does not exist. In this Playbook we create a raw file using fallocate. There is not a module to do this, so we use the command module. As the command fallocate creates a file we can easily check for this file before execution

Demo

Controlling Command Execution

The raw file can be used as a filesystem if we create a loop device file

Extending the Playbook

We can extend the Playbook. Using variables can help reduce the repetitions of the filename and device name. Add tasks will allow us to create the device and format it. Formatting is provided with a standard module

```yaml
- name: 'Manage Disk File'
  hosts: rhel
  become: true
  gather_facts: false
  vars:
    disk_file: '/root/disk0'
    loop_dev: '/dev/loop100'
  tasks:
    - name: "Create Raw Disk File"
      command:
        cmd: "fallocate -l 1G {{ disk_file }}"
        creates: "{{ disk_file }}"
    - name: "Create loop device"
      command:
        cmd: "losetup {{ loop_dev }} {{ disk_file }}"
        creates: "{{ loop_dev }}"
    - name: "Format disk file"
      filesystem:
        fstype: xfs
        dev: "{{ loop_dev }}"
```

Demo

Building Idempotent Playbooks using command modules

```
$ ansible-galaxy collection install ansible.windows

- name: Execute whoami
  ansible.windows.win_command: whoami
```

win_command

If we are managing Windows devices we can use the win_command or win_shell modules. These are not built-in to Ansible but can be added to the controller from Ansible Galaxy

Summary

Ansible Modules Running Native Commands:

- command
  - Does not create shell
- shell
  - Creates shell and possible injection attacks
- script
  - Runs script from controller across nodes
- raw
  - Natively run the remote command without need of Python
- creates: Meta parameter controls execution

Working with the Big Three