

Scripting Linux Administration



Andrew Mallett

Linux Author and Trainer

@theurbanpenguin www.theurbanpenguin.com



Overview



We Don't Need Ansible - We Can Script IT!

- Well perhaps but...
- But what about differing operating systems
- What about idempotency and remind me, what is idempotency
- Playbooks are easier, believe me, the hard work has already been done
- But we can use Ansible in scripts
- We can also use Ansible in provisioning Vagrant virtual machines





Traditional Scripts

Administration of Linux has been maintained for many years with scripts running the automation



```
#!/bin/bash  
sudo timedatectl set-timezone 'Europe/London'  
sudo yum install tree
```

Simple Script

A simple script can easily install on a single distrubution

```
#!/bin/bash
if [ -f '/usr/bin/apt' ]; then
    sudo apt update -y
    sudo apt install -y tree
else
    sudo yum install -y tree
fi
```

Cross Distribution

Just dealing with two different Linux distributions adds an immense amount of complexity. This is just the script to install one package. We still must copy the script to each system and execute on each system

```
$ chmod u+x shell.sh
```

```
$ ./shell.sh
```

```
$ scp shell.sh tux@192.168.33.12:
```

```
$ scp shell.sh tux@192.168.33.13:
```

```
$ ssh tux@192.168.33.12 -C '~/shell.sh'
```

```
$ ssh tux@192.168.33.13 -C '~/shell.sh'
```

Executing the Script

Locally, it is easy to execute the script. For the remote systems we must copy the file across and then execute it

Demo



Working with BASH Scripts



Without building in more complexity, the installer will run each time no matter the state of the system




```
#!/bin/bash
cd ~/home/vagrant/
ansible -b all -m package -a 'name=tree state=present'
ansible -b all -m file -a \
'path=/etc/localtime state=link src=/usr/share/zoneinfo/Europe/London force=true'
```

Using Ansible in Scripts

If we do not like the idea of YAML and Playbooks, it is possible to use ad-hoc commands in scripts. Here we install tree and set the local timezone

Demo



Using Ansible within Scripts





Ansible and Vagrant

We have used Vagrant to deploy the virtual machines for the course. If we have Ansible installed on the host system, we can configure the systems using Ansible. For example, password-based authentication is not enabled for SSH, we can enable this



```
- name: create_user_tux
  user:
    name: tux
    state: present
    shell: /bin/bash
    password: "{{ 'Password1' | password_hash('sha512') }}"
    update_password: on_create
```

Creating User

We can create a dedicated user account

```
- name: allow_ssh_password_authentication
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^PasswordAuthentication '
    insertafter: '#PasswordAuthentication'
    line: 'PasswordAuthentication yes'
  notify: restart_sshd
```

Editing the SSHD Configuration

The **lineinfile** module allows simple editing of a file

```
- name: escalation_of_priviliges_for_tux
  copy:
    dest: /etc/sudoers.d/tux
    content: "tux ALL=(root) NOPASSWD: ALL"
```

Creating Sudoers File

We can create a custom sudoers entry for the newly created user

```
handlers:  
  - name: restart_sshd  
    service:  
      name: sshd  
      state: restarted
```

Handlers

Handlers, unlike tasks, only execute if called (notified) by a task. If the SSHD file is edited the service will restart

```
config.vm.define "rhel8" do |rhel8|  
  rhel8.vm.box = "generic/rhel8"  
  rhel8.vm.network "private_network", ip: "192.168.33.11"  
  rhel8.vm.hostname = "rhel8"  
  rhel8.vm.provision "ansible", playbook: "deploy.yaml"  
  rhel8.vm.provider "virtualbox" do |vb|  
    vb.memory = "1024"  
  end  
end  
  
$ vagrant up ubuntu --provision
```

Alter Vagrantfile

We can add, to each VM, the provision task. Even if the systems are provisioned, we can run Ansible when the system starts by using the `--provision` option

Demo



The power of Ansible

- Add Ansible provisioning to Vagrant
- Create Playbook to execute across the 3 Systems
- Create user, sudo rights, and SSHD configuration
- 3 different systems, one configuration



Summary



Scripting:

- BASH Scripts can automate administration
- We must build the logic
- Errors may occur if already configured
- Using Ansible in scripts
- Using Ansible in Vagrant





Using Shell Commands in Ansible