Task1

1. Create user for Ansible on all your virtual machines.

I use Vagrant file to create virtual machines. Since I use hyper-v as vm.provider I need to update hosts file and install ssh key manually because Vagrant does not work with Hyper-V networks

Node1: 192.168.0.108

Node2: 192.168.0.108

Control:

Commands:

Vagrant up

vagrant ssh control

sudo vim /etc/hosts # add node1 and node2 hosts

#update ssh keys

ssh-keygen -b 2048 -t rsa -f /home/vagrant/.ssh/id\_rsa -q -N ""

echo -e "Host node\*\n StrictHostKeyChecking no\n UserKnownHostsFile /dev/null\n User vagrant\n" > .ssh/config

chmod 600 .ssh/config

sshpass -p 'vagrant' ssh-copy-id -i /home/vagrant/.ssh/id\_rsa.pub vagrant@node1

sshpass -p 'vagrant' ssh-copy-id -i /home/vagrant/.ssh/id\_rsa.pub vagrant@node2

1. Install ansible to your control machine

sudo pip3 install ansible

1. Add your nodes to the inventory. You can use either default inventory location or create a you own file

Vim ansible.cfg

# Config #

|  |
| --- |
| [defaults] |
|  |

|  |
| --- |
| stdout\_callback = yaml |
|  |

|  |
| --- |
| connection = smart |
|  |

|  |
| --- |
| timeout = 60 |
|  |

|  |
| --- |
| deprecation\_warnings = False |
|  |

|  |
| --- |
| host\_key\_checking = False |
|  |

|  |
| --- |
| retry\_files\_enabled = False |
|  |

inventory = ./inventory/hosts.ini

##########

Mkdir playbooks

Cd playbooks

Mkdir inventory

Vim inventory/hosts

# hosts content #

[mgm]

control

[prod]

control

node2

[dev]

node1

[managed\_nodes]

node1

node2

####################

ansible managed\_nodes -m ping -i invertory/hosts.ini

1. With an ad-hoc command please make sure is everything runs correctly, for example print a hostnameof the nodes from the facts

ansible all -m setup -a”filter=’\* ansible\_hostname\*’”

1. Create a playbook to print a list of network interfaces for every virtual machine

Vim playbook-facts.yml

# Content #

---

- name: Return interfaces from facts

hosts: all

tasks:

- name: Debug Ansible Fact(s)

debug:

msg:

- "Interfaces: {{ ansible\_interfaces }}"

###########

Task 2

1.Create the role that makes following:

a.Install a httpd package

b.Split two hosts to different host groups: web1 and web2

c.Add a whoami.html file to default web-server location, that describes a server and server group and add a custom phrase “I’m tomato” for the first one and “I’m potato” for a second one (tip: web-serverdefault location is/var/www/html/)

d.Create a second file whoami2.html and include following lines to your file that matches a vegetable for your server, for example:i.nice potatoii.big potatoiii.cool potato

Please restart httpd process if only a ‘d’ has changes

httpd.yml

# Content #

---

###########################################################

############# Instal httpd service ########################

###########################################################

- name: Install httpd package

  yum:

    name: httpd

    state: latest

- name: run httpd service

  service:

    name: httpd

    state: started

    enabled: yes

- name: create new file (main file)

  template:

    src: whoami.html.j2

    dest: /var/www/html/whoami.html

  register: check\_main\_file

- name: create new file (second file)

  template:

    src: whoami2.html.j2

    dest: /var/www/html/whoami2.html

  register: check\_second\_file

- name: restart httpd

  service:

    name: httpd

    state: restart

  when: (check\_main\_file.changed == 'True' or check\_second\_file.changed == 'True')

#########

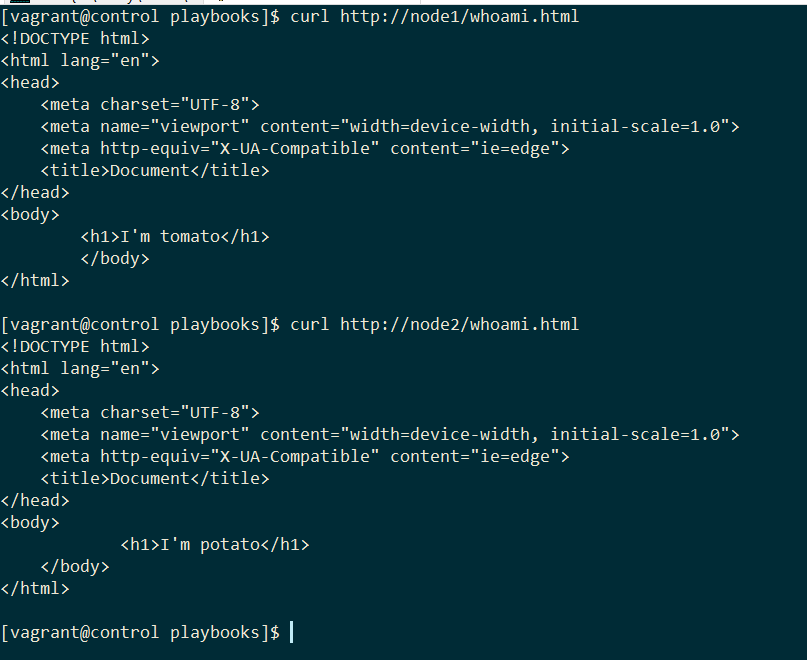
Add 2 template files to templates folder

2.Add your role to playbook and run.

3.Ensure all done correctly by running curl like:

curlhttp://node1/whoami.html

curlhttp://node1/whoami2.htmlHintUse control loop for creating entries in files



Task 3

You are the systems engineer in Company and you are administering the server fleet. Your manager needs a server uptime report.

Create a custom module to report server uptime and then create a report file on control server:

Hostname:

Server uptime:

Ensure that your control node is absent in your report.

HintUse ‘delegate\_to’ option to create a local file for report

Create new role

Add tasks in uptime.yml

---

###########################################################

####################### Get uptime ########################

###########################################################

- name: get uptime

host\_uptime:

register: result

- name: print report

debug: msg="{{ ansible\_fqdn }} {{result.uptime}} seconds"

- name: print report control

debug: msg="{{ ansible\_fqdn }} {{result.uptime}} seconds"

delegate\_to: control

- name: generate report

template:

src: report.txt.j2

dest: /home/vagrant/playbooks/report.txt

delegate\_to: control

Add report.txt.j2 template file for report

{{ ansible\_fqdn }} {{result.uptime}} seconds

Add module file host\_uptime.py

#!/usr/bin/env python

from ansible.module\_utils.basic import AnsibleModule

def get\_uptime():

    with open('/proc/uptime', 'r') as f:

        uptime\_seconds = float(f.readline().split()[0])

    return uptime\_seconds

def main():

  module = AnsibleModule(

    argument\_spec=dict(),

    supports\_check\_mode=False

  )

  result = dict()

  result["changed"] = True

  result["uptime"] = get\_uptime()

  module.exit\_json(\*\*result)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

Task 4

You are the systems engineer, and you need to create a several users on your server fleet. You have a list of passwords and users.

Create playbook to add 5 users (user1, user2...etc.) without any clear-text password compromising -even in ansible output. With minimal efforts.

Hint: Use no\_log option to hide output from the terminal

Create encrypted file using ansible-vault

#passwords content#

passwords:

- 1

- 2

- 3

- 4

- 5

#########

Put it in group\_vars folder

Create .vault-pass file

Add vault\_password\_file = .vault-pass to ansible.cfg

Create new playbook

# task4.yml content #

---

## PLAY 4

#######################

- name: task4

hosts:

- web1

- web2

become: yes

vars\_files:

- ./group\_vars/passwords

- ./group\_vars/users.yml

roles:

- users

#######

Create new users role and add next task

# users.yml content #

---

###########################################################

##################### Create users ########################

###########################################################

- name : Create users

no\_log: True

user:

name: "{{ item.0 }}"

password: "{{ item.1 }}"

with\_together:

- "{{ users }}"

- "{{ passwords }}"

#################

