Guided Practice: Using Ansible Roles

Introduction

Ansible’s strengths include the ability to facilitate the configuration of UNIX, Linux, Windows, and Mac operating systems that include groups that may be dependent on other groups which can involve the secure backing up of data, configuring firewalls, creating users and groups, and replication.

Outcome

In this Guided Practice, you will create a modular, reusable library of code for application configurations using ansible-galaxy to set up roles to install Apache.

Resources Needed

- For this Guided Practice, we will use the CentOS 8 and the Ubuntu 2020 server in the VCastle pod configured for this class.

Level of Difficulty

Medium

Deliverables

Deliverables are marked with a red border around the screenshot.

General Considerations

You should be familiar with Linux and Windows networking. Secure Shell (ssh) is installed and configured on the Ubuntu server from a prior Guided Practice. Ansible is installed and configured on your CentOS Computer, and you have completed all the previous exercises.

Instructions

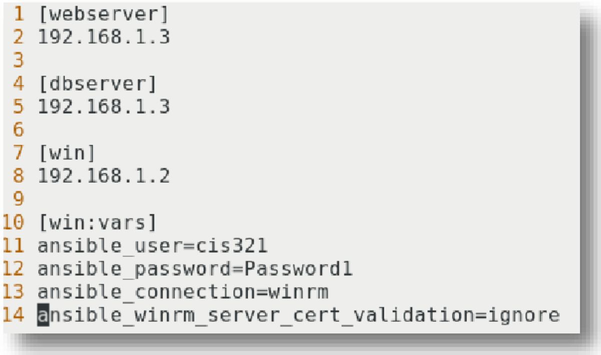
⚠ *Important*: *Please Note*

All of your screenshots should include information that shows your login information.

Edit the Ansible Inventory (host) File

First, you will edit the Ansible Inventory (host) file to reflect your network architecture, if necessary.

1. View the inventory (hosts) file, and ensure your inventory file reflects the architecture of your network as shown below.

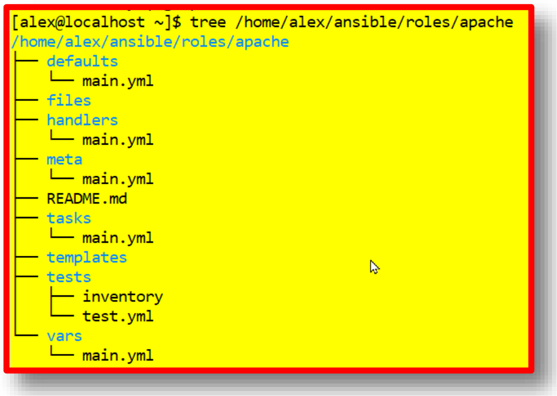


```
[webserver]
192.168.1.3

[dbserver]
192.168.1.3

[win]
192.168.1.2

[win:vars]
ansible_user=cis321
ansible_password=Password1
ansible_connection=winrm
ansible_winrm_server_cert_validation=ignore
```

Set Up the Directory Structure

2. On the CentOS server, type the following commands in a shell (substituting the appropriate username for **alex** for all steps):

```
ansible-galaxy init /home/alex/ansible/roles/apache --offline
tree /home/alex/ansible/roles/apache
```



```
[alex@localhost ~]$ tree /home/alex/ansible/roles/apache
/home/alex/ansible/roles/apache
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

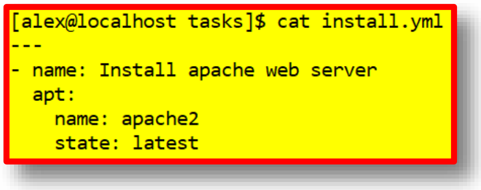
Take a screenshot that resembles the one above, and paste it in your Lab Report.

Create the Required Files to Install Apache Using Roles

Note: If Apache has been previously installed, this playbook will run without making unnecessary changes.

3. In your shell, type the following:

```
cd /home/alex/ansible/roles/apache/tasks/
vim install.yml
i
---
- name: Install apache web server
  apt:
    name: apache2
    state: latest
<ESC>:wq and press <ENTER>
```

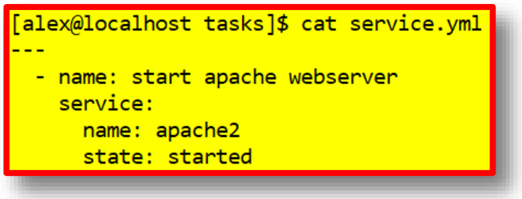


```
[alex@localhost tasks]$ cat install.yml
---
- name: Install apache web server
  apt:
    name: apache2
    state: latest
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

4. In your shell, type:

```
vim service.yml
i
---
  - name: start apache webserver
    service:
      name: apache2
      state: started
      enabled: true
<ESC>:wq and press <ENTER>
```

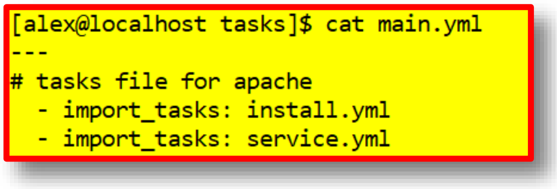


```
[alex@localhost tasks]$ cat service.yml
---
  - name: start apache webserver
    service:
      name: apache2
      state: started
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

5. In your shell, type:

```
vim main.yml
i
---
# tasks file for apache
  - import_tasks: install.yml
  - import_tasks: service.yml
<ESC>:wq and press <ENTER>
```



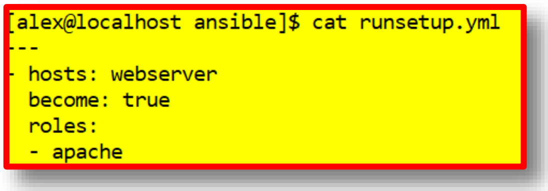
```
[alex@localhost tasks]$ cat main.yml
---
# tasks file for apache
  - import_tasks: install.yml
  - import_tasks: service.yml
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

Create a Playbook to Install APACHE Using Your Previously Created Structure

6. In your shell, type:

```
cd /home/alex/ansible/
vim runsetup.yml
i
---
- hosts: webserver
  become: true
  roles:
  - apache
<ESC>:wq and press <ENTER>
```

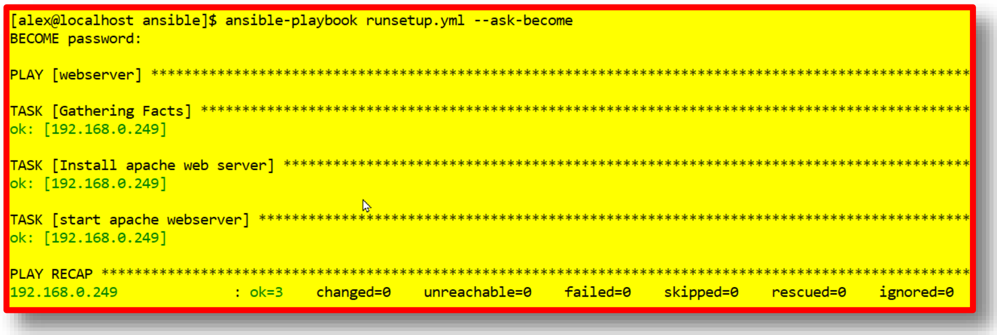


```
[alex@localhost ansible]$ cat runsetup.yml
---
- hosts: webserver
  become: true
  roles:
  - apache
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

7. In your shell type:

```
ansible-playbook runsetup.yml --ask-become
```



```
[alex@localhost ansible]$ ansible-playbook runsetup.yml --ask-become
BECOME password:

PLAY [webserver] **************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [192.168.0.249]

TASK [Install apache web server] **********************************************************************************
ok: [192.168.0.249]

TASK [start apache webserver] *************************************************************************************
ok: [192.168.0.249]

PLAY RECAP ********************************************************************************************************
192.168.0.249              : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Take a screenshot that resembles the one above, and paste it in your Lab Report.

Guided Practice Questions

In your **Guided Practice Lab Report**, answer the following questions about this learning activity.

1. How do roles facilitate adding new applications or configurations (installing emacs for example)?