Lesson-End Project

Initializing Ansible Roles and Inventory

Project agenda: To automate the deployment and management of an Apache web server using Ansible roles, securely handle sensitive information using Ansible Vault, and dynamically manage the inventory using a custom Python script for AWS EC2 instances

Description: As a DevOps engineer at a growing tech company, you are tasked with automating the deployment of Apache web servers. The company's infrastructure is dynamic, requiring frequent updates and the ability to manage sensitive configuration data securely. Your objectives are to create an Ansible role for installing and configuring Apache, use Ansible Vault to manage sensitive data like passwords, and implement a dynamic inventory to handle the constantly changing server environment.

Tools required: Python, Ansible, Ansible Vault, and VS Code

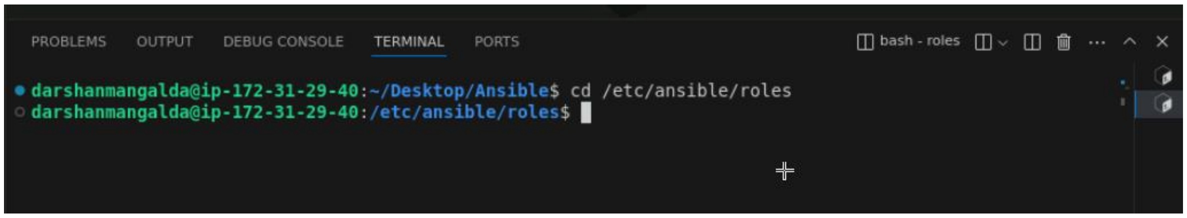
Prerequisites: None

Expected Deliverables: An Ansible playbook that automates Apache web server deployment and management, complete with detailed installation and execution documentation.

Steps to be followed:
1. Initialize the Ansible role for Apache
2. Define tasks for the Ansible roles
3. Configure and use Ansible Vault
4. Implement dynamic inventory

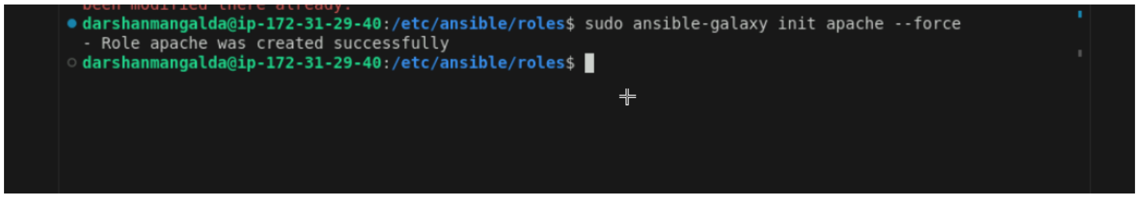
Step 1: Initialize the Ansible role for Apache

1.1 Run the following command to navigate to the **roles** directory:
cd /etc/ansible/roles



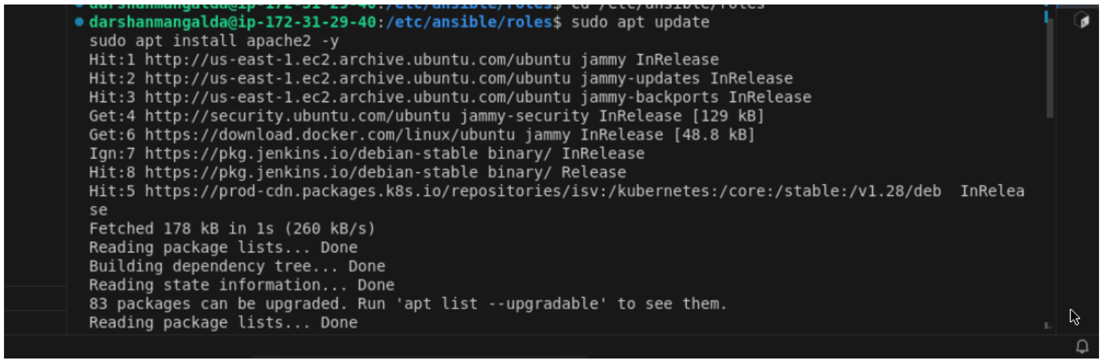
```
darshanmangalda@ip-172-31-29-40:~/Desktop/Ansible$ cd /etc/ansible/roles
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles$
```

1.2 To initialize the role, run the following command:
sudo ansible-galaxy init apache --force



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles$ sudo ansible-galaxy init apache --force
- Role apache was created successfully
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles$
```

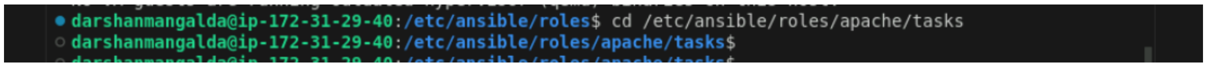
1.3 Run the following command to install the Apache HTTP server on the control node for local testing:
sudo apt update
sudo apt install apache2 -y



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles$ sudo apt update
sudo apt install apache2 -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Ign:7 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.28/deb  InRelea
se
Fetched 178 kB in 1s (260 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
83 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
```

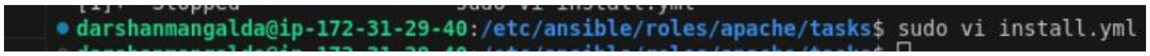
Step 2: Define tasks for the ansible roles

2.1 Run the following command to navigate to the **tasks** directory within the **Apache role** directory:
cd /etc/ansible/roles/apache/tasks



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles$ cd /etc/ansible/roles/apache/tasks
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$
```

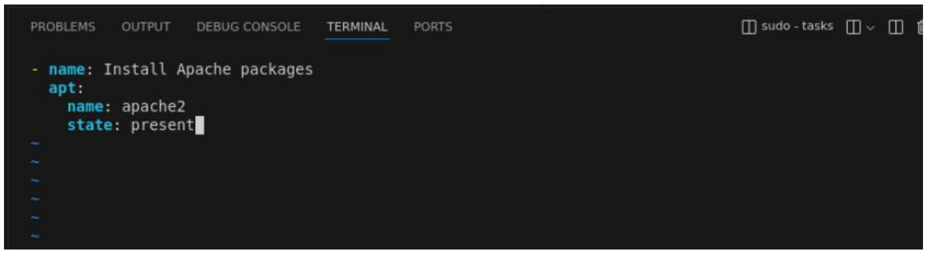
2.2 Execute the following command to create and edit the **install.yml** file:
sudo vi install.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$ sudo vi install.yml
```

2.3 Add the following script to the **install.yml** file:

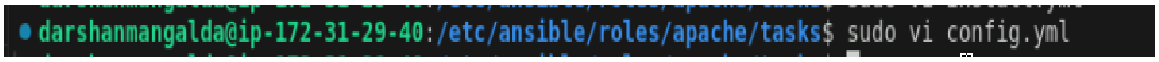
- name: Install Apache packages
  apt:
      name: apache2
      state: present



```
- name: Install Apache packages
  apt:
    name: apache2
    state: present
~
~
~
~
~
~
```

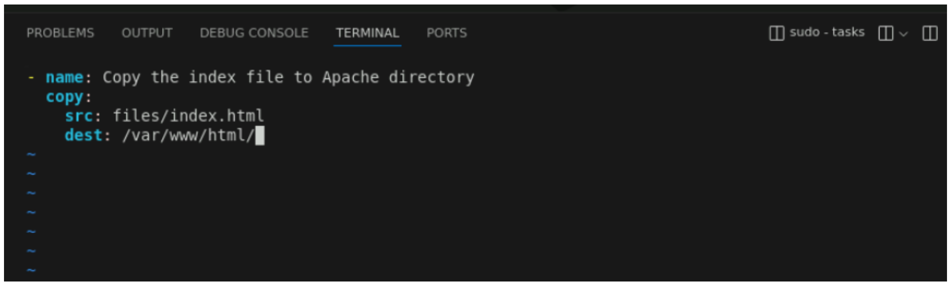
This script installs the Apache2 package using the apt package.

2.4 Execute the following command to create and edit the **config.yml** file:
sudo vi config.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$ sudo vi config.yml
```

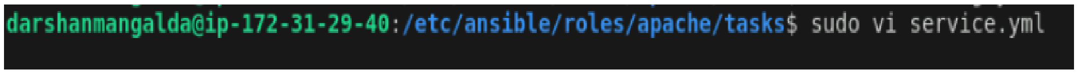
2.5 Add the following script to the **congif.yml** file:
- name: Copy the index file to Apache directory
  copy:
    src: files/index.html
    dest: /var/www/html/



```
- name: Copy the index file to Apache directory
  copy:
    src: files/index.html
    dest: /var/www/html/
~
~
~
~
~
~
~
```

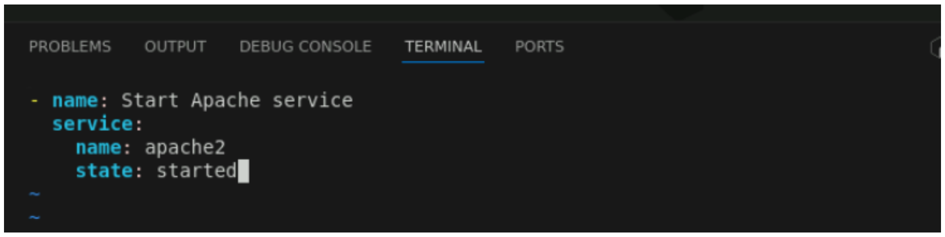
This script copies the **index.html** file to the Apache web server's document root directory.

2.6 Run the following command to create and edit the service task file:
sudo vi service.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$ sudo vi service.yml
```

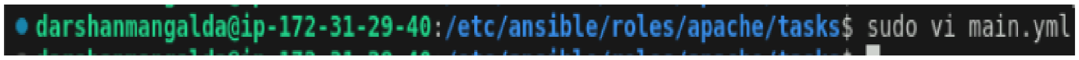
2.7 Add the following script to the **service.yml** file:
- name: Start Apache service
service:
name: apache2
state: started



```
- name: Start Apache service
  service:
    name: apache2
    state: started
~
~
```

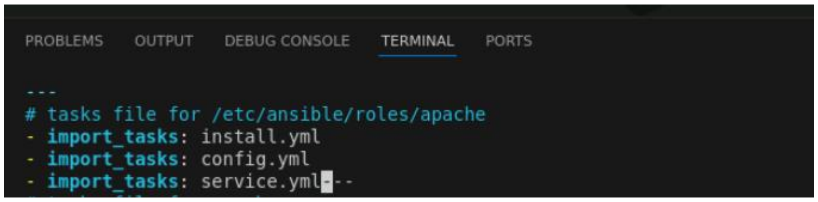
This script starts the Apache service and ensures it is running.

2.8 Execute the following command to create and edit the main.yml file:
sudo vi main.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$ sudo vi main.yml
```

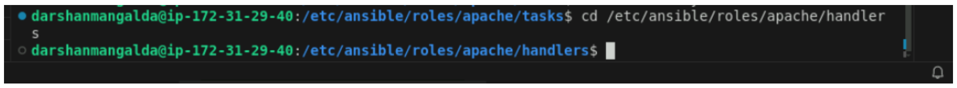
2.9 Add the following script to the **main.yml** file:
# tasks file for /etc/ansible/roles/apache
- import_tasks: install.yml
- import_tasks: config.yml
- import_tasks: service.yml



```
---
# tasks file for /etc/ansible/roles/apache
- import_tasks: install.yml
- import_tasks: config.yml
- import_tasks: service.yml---
```

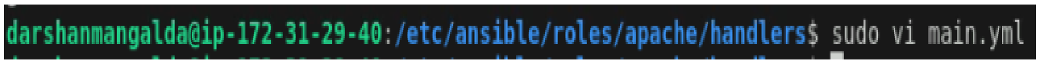
This script imports and runs the tasks defined in the install.yml, config.yml, and service.yml files for the Apache role.

2.10 Run the following command to navigate to the **handlers** directory:
cd /etc/ansible/roles/apache/handlers



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/tasks$ cd /etc/ansible/roles/apache/handler
s
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$
```

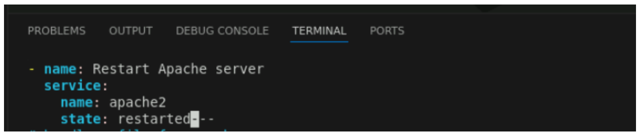
2.11 Execute the following command to edit the **main.yml** file:
sudo vi main.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo vi main.yml
```

2.12 Add the following script to the **main.yml** file:

```
- name: Restart Apache server
  service:
    name: apache2
    state: restarted
```



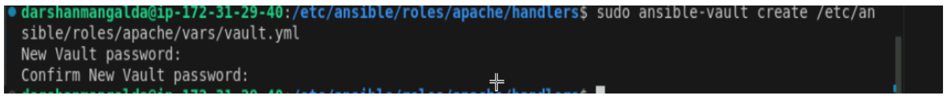
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
- name: Restart Apache server
  service:
    name: apache2
    state: restarted---
```

This handler will restart the Apache2 service when triggered by other tasks in the Ansible playbook.

Step 3: Configure and use Ansible Vault

3.1 Run the following command to create a new Vault file for securely storing sensitive data:
sudo ansible-vault create /etc/ansible/roles/apache/vars/vault.yml



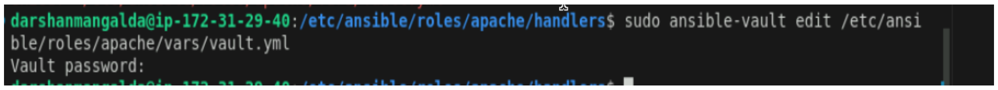
```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo ansible-vault create /etc/an
sible/roles/apache/vars/vault.yml
New Vault password:
Confirm New Vault password:
```

Add sensitive information (such as passwords) to **vault.yml**.

Note: It will prompt you to create a password. Enter and confirm your password.

3.2 Execute the following command to edit the vault file:

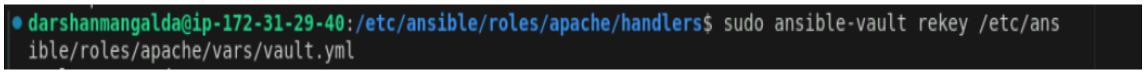
sudo ansible-vault edit /etc/ansible/roles/apache/vars/vault.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo ansible-vault edit /etc/ansi
ble/roles/apache/vars/vault.yml
Vault password:
```

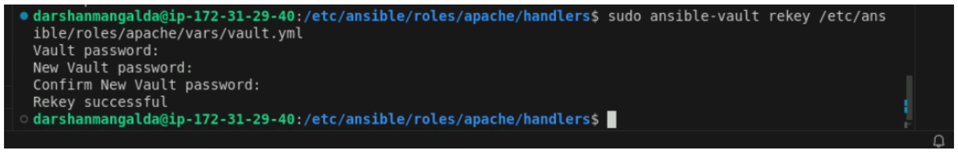
3.3 Use the following command to change the password:

sudo ansible-vault rekey /etc/ansible/roles/apache/vars/vault.yml



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo ansible-vault rekey /etc/ans
ible/roles/apache/vars/vault.yml
```

3.4 Enter the vault password, the new password, and confirm the new password.

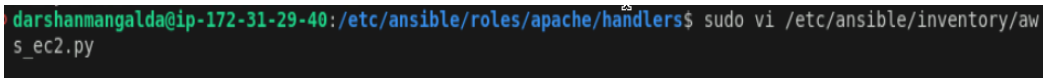


```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo ansible-vault rekey /etc/ans
ible/roles/apache/vars/vault.yml
Vault password:
New Vault password:
Confirm New Vault password:
Rekey successful
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$
```

Step 4: Implement Dynamic Inventory

4.1 Run the following command to create a Dynamic Inventory Script for AWS:

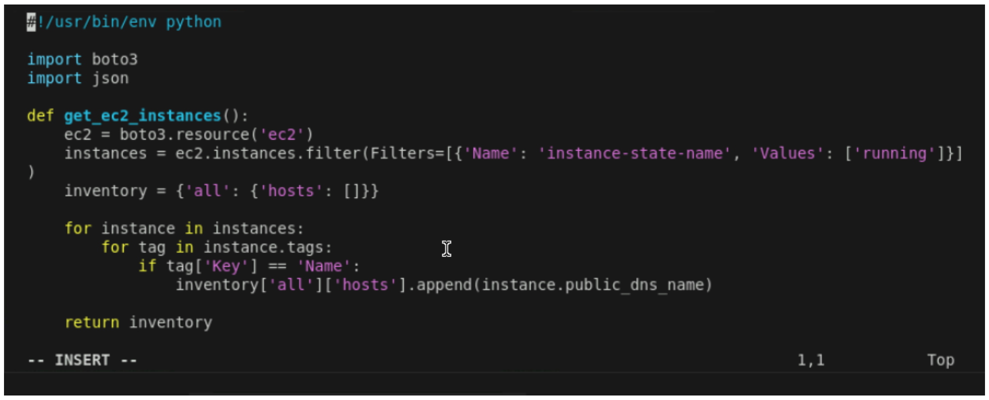
sudo vi /etc/ansible/inventory/aws_ec2.py



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/roles/apache/handlers$ sudo vi /etc/ansible/inventory/aw
s_ec2.py
```

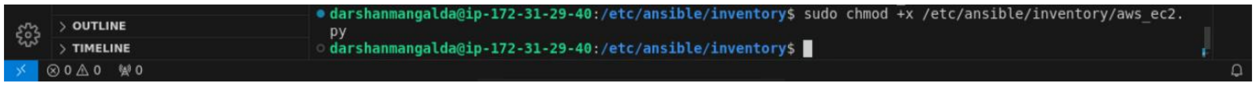
4.2 Add the following python script to the **aws_ec2.py** file:

```
#!/usr/bin/env python

import boto3
import json

def get_ec2_instances():
   ec2 = boto3.resource('ec2')
   instances = ec2.instances.filter(Filters=[{'Name': 'instance-state-name', 'Values':
['running']}])
   inventory = {'all': {'hosts': []}}

   for instance in instances:
      for tag in instance.tags:
           if tag['Key'] == 'Name':
               inventory['all']['hosts'].append(instance.public_dns_name)

   return inventory

if __name__ == "__main__":
   inventory = get_ec2_instances()
   print(json.dumps(inventory))
```



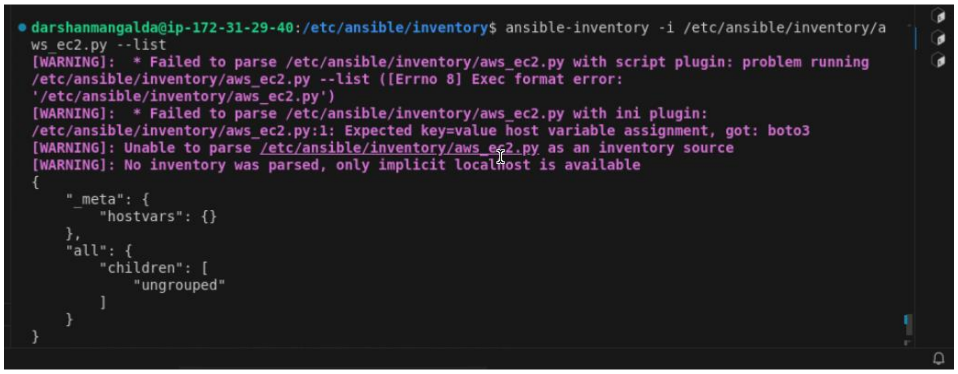
```
#!/usr/bin/env python

import boto3
import json

def get_ec2_instances():
    ec2 = boto3.resource('ec2')
    instances = ec2.instances.filter(Filters=[{'Name': 'instance-state-name', 'Values': ['running']}]
)
    inventory = {'all': {'hosts': []}}

    for instance in instances:
        for tag in instance.tags:
            if tag['Key'] == 'Name':
                inventory['all']['hosts'].append(instance.public_dns_name)

    return inventory

-- INSERT --                                                                    1,1          Top
```

4.3 Enter the following command to make the script executable:
sudo chmod +x /etc/ansible/inventory/aws_ec2.py



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/inventory$ sudo chmod +x /etc/ansible/inventory/aws_ec2.
py
darshanmangalda@ip-172-31-29-40:/etc/ansible/inventory$
```

4.4 Run the following command to test the Dynamic Inventory:
ansible-inventory -i /etc/ansible/inventory/aws_ec2.py –list



```
darshanmangalda@ip-172-31-29-40:/etc/ansible/inventory$ ansible-inventory -i /etc/ansible/inventory/a
ws_ec2.py --list
[WARNING]:  * Failed to parse /etc/ansible/inventory/aws_ec2.py with script plugin: problem running
/etc/ansible/inventory/aws_ec2.py --list ([Errno 8] Exec format error:
'/etc/ansible/inventory/aws_ec2.py')
[WARNING]:  * Failed to parse /etc/ansible/inventory/aws_ec2.py with ini plugin:
/etc/ansible/inventory/aws_ec2.py:1: Expected key=value host variable assignment, got: boto3
[WARNING]: Unable to parse /etc/ansible/inventory/aws_ec2.py as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
{
    "_meta": {
        "hostvars": {}
    },
    "all": {
        "children": [
            "ungrouped"
        ]
    }
}
```

By following these steps, you have successfully created an Ansible role for installing and configuring Apache, securely managing sensitive data using Ansible Vault, dynamically managing the inventory of servers, and then executing it using an Ansible playbook.