Lesson 07 Demo 01

Creating Ansible Roles

Objective: To create and configure Ansible roles for managing configurations

Tools required: Ansible, Ansible Vault, and VS Code

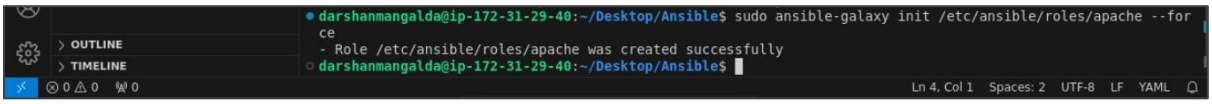
Prerequisites: None

Steps to be followed:

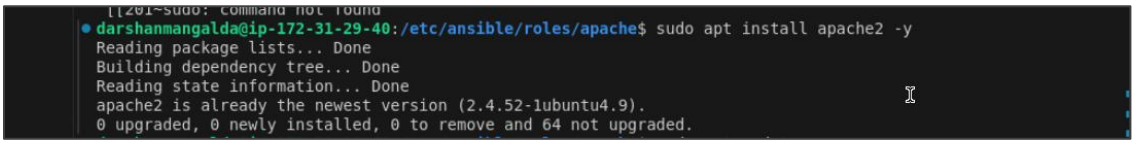
1. Initialize the roles
2. Create and define tasks
3. Organize task execution
4. Create the index.html file
5. Verify the configuration
6. Execute the playbook
7. Check the index.html file in the browser

Step 1: Initialize the roles

1.1 Run the following command to initialize the Apache role directory:
sudo ansible-galaxy init/etc/ansible/roles/apache --force

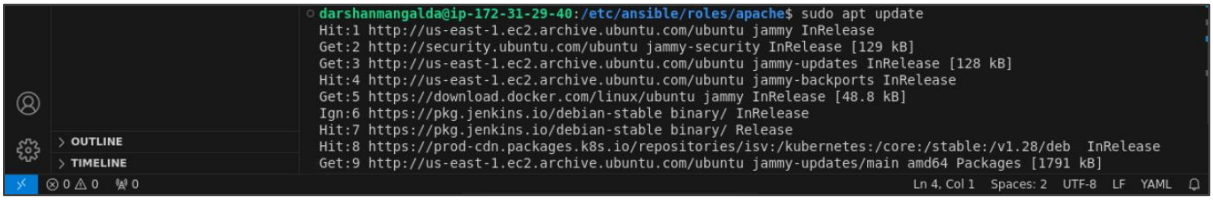


1.2 Execute the following command to install the Apache HTTP server:
sudo apt install apache2 -y



1.3 Run the following command to update the HTTP server:

sudo apt update



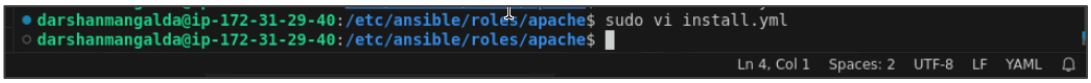
Step 2: Create and define tasks

2.1 Execute the following command to navigate to the Apache role directory:

cd /etc/ansible/roles/apache

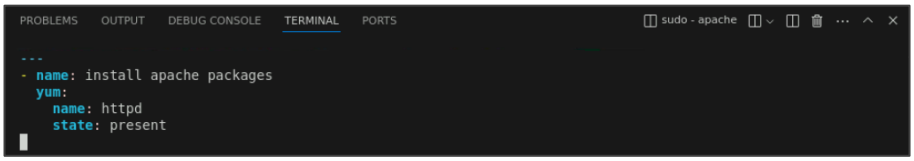
2.2 Run the following command to create an **install.yml** file:

sudo vi install.yml



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

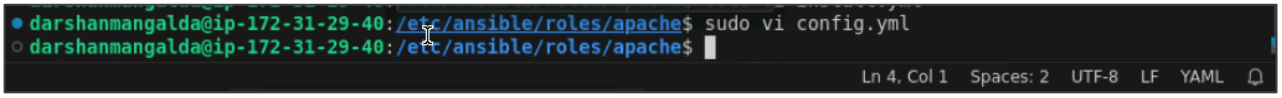
- name: install apache packages
yum:
name: httpd
state: present



This script installs the Apache HTTP server package using the **yum** package manager.

2.4 Execute the following command to create and edit a configuration file:

sudo vi config.yml



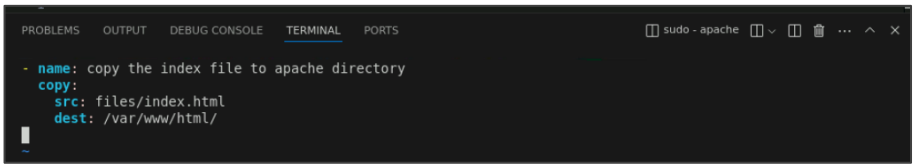
2.5 Add the following script to the **config.yml** file:

- name: copy the index file to apache directory

copy:

src: files/index.html

dest: /var/www/html/



This script copies the **index.html** file from the local files directory to the **/var/www/html/** directory on the server.

2.6 Run the following command to create and edit the service task file:

sudo vi service.yml

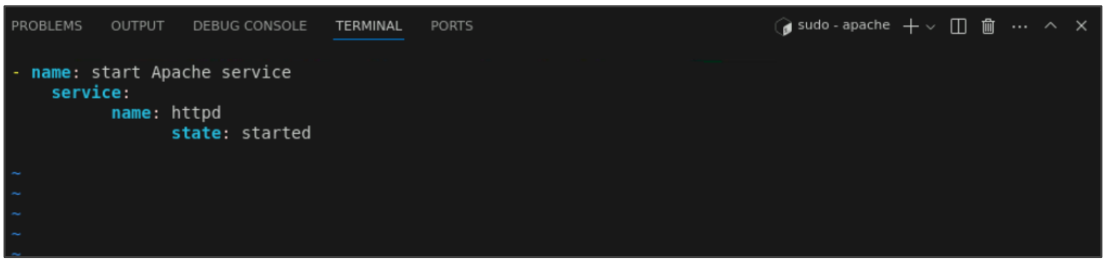
2.7 Add the following script to the **service.yml** file**:**

- name: start Apache service

service:

name: httpd

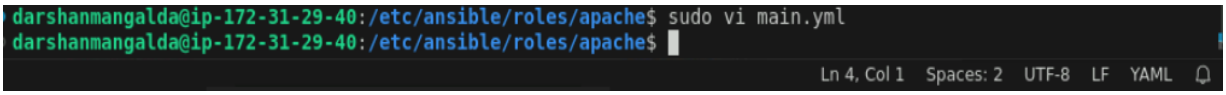
state: started



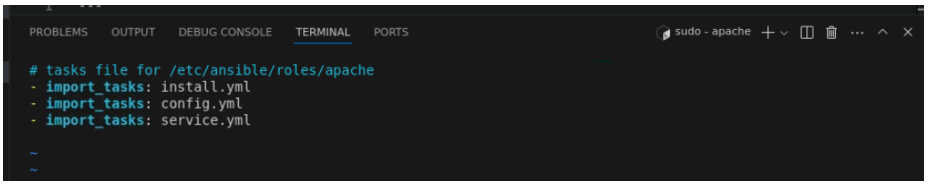
This Ansible task ensures that the **HTTPD** Apache service is started.

Step 3: Organize task execution

3.1 Run the following command to create the **main.yml** file:
sudo vi main.yml



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

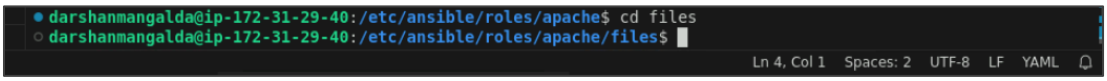


The code imports and runs tasks from install.yml, config.yml, and service.yml to install, configure, and manage the Apache server.

Step 4: Create the index.html file

4.1 Run the following command to exit from the current directory:
cd ..

4.2 Run the following command to navigate inside the **files** directory:
cd files

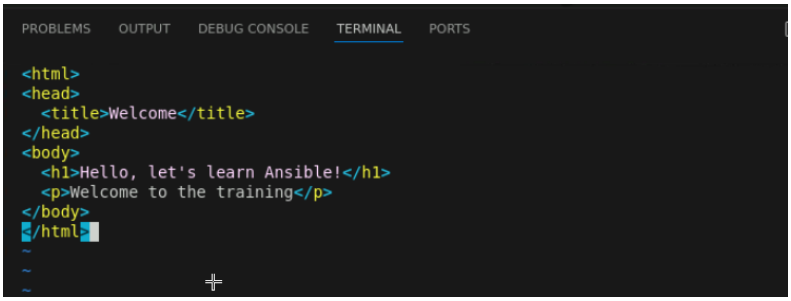


4.3 Run the following command to create an **index.html** file:

sudo vi index.html

4.4 Add the following script in the **index.html**:

```
<html>
<head>
 <title>Welcome</title>
</head>
<body>
 <h1>Hello, let's learn Ansible!</h1>
 <p>Welcome to the training</p>
</body>
</html>
```



Note: Press esc, then type :wq to save and quit

Step 5: Verify the configuration

5.1 Run the following command to exit from the current directory:

cd ..

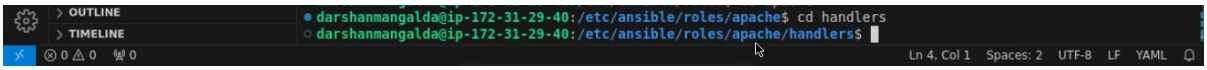
5.2 Run the following command to navigate to the **apache** directory:

cd apache

Step 6: Execute the playbook

6.1 Run the following command to exit from the current directory:
cd ..

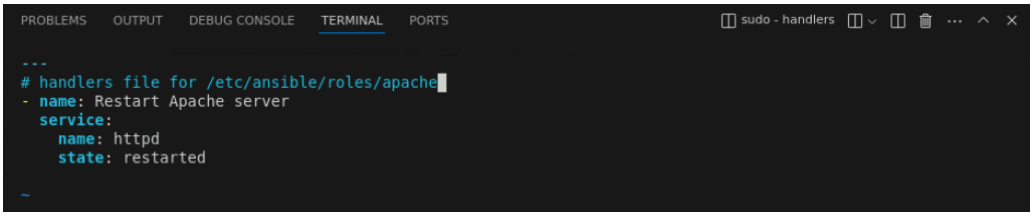
6.2 Execute the following command to navigate inside the **handlers** directory:
cd handlers



6.3 Run the following command to edit the **main.yml** file:
sudo vi main.yml

6.4 Add the following script in the **main.yml** file:

```
# handlers file for /etc/ansible/roles/apache
  - name: Restart Apache server
    service:
      name: httpd
      state: restarted
```



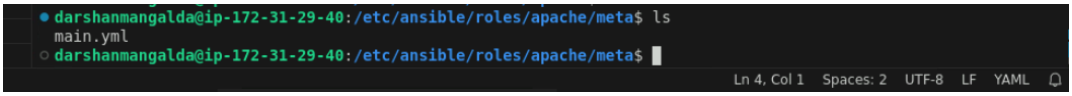
The code restarts the Apache server using the **httpd** service.

> **Note:** You have already created the service.yml file in the task. In the later part of the lab, we will remove the service.yml file and handlers will be used.

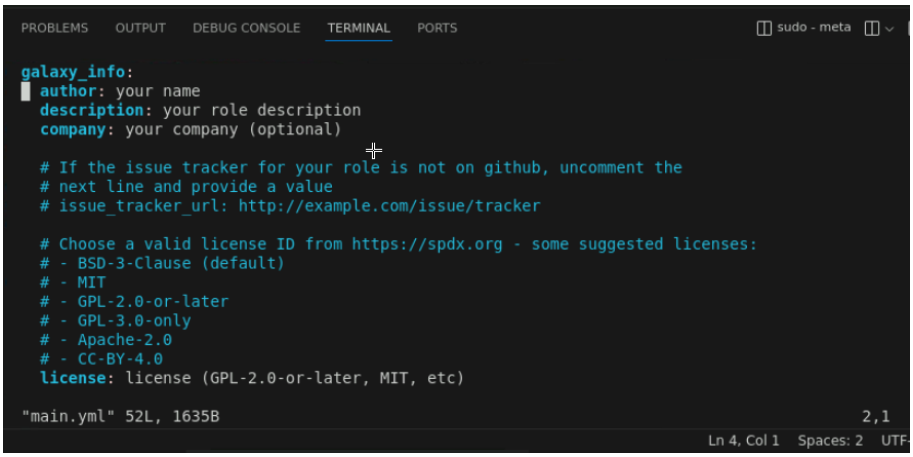
6.5 Run the following command to exit from the current directory:
cd ..

6.6 Execute the following command to navigate into the **meta** directory:
cd meta

6.7 Run the **ls** command to list the contents in the directory



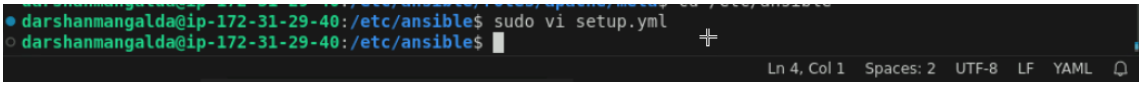
6.8 Execute the following command to view the **main.yml** file:
sudo vi main.yml



6.9 Execute the following command to navigate to the **ansible** directory:
cd /etc/ansible

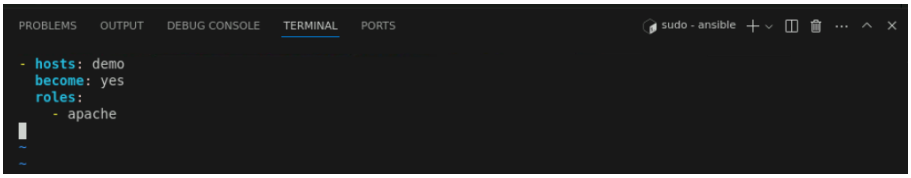
6.10 Run the following command to edit the **setup.yml** file:

sudo vi setup.yml



6.11 Add the following script in **setup.yml**:

```
- hosts: demo
  become: yes
  roles:
    - apache
```

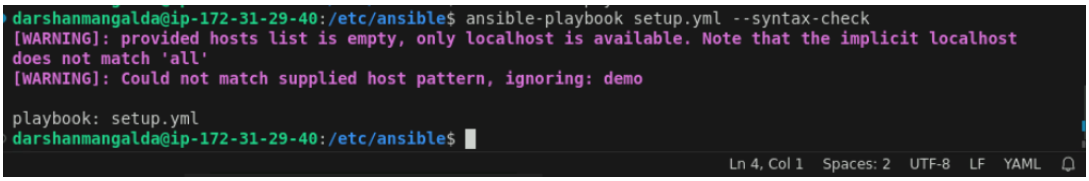


This code runs the Apache role on all hosts in the demo group with elevated privileges.

Note: If you have logged in as the root user (which means you have used 'sudo su'), you need to first run the 'exit' command before you can run the 'ansible-playbook' command.

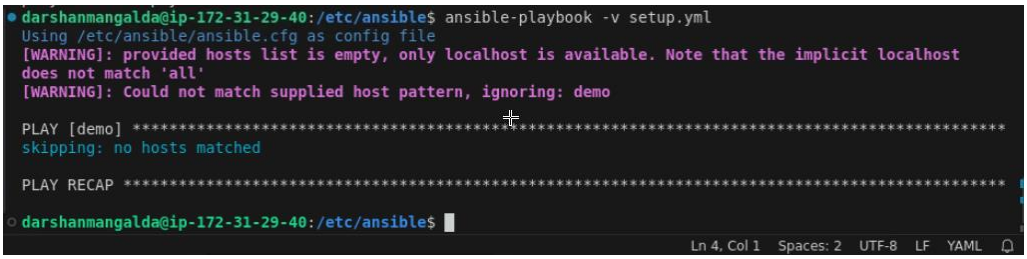
6.12 Run the following command to check the syntax of the **setup.yml** playbook:

ansible-playbook setup.yml --syntax-check



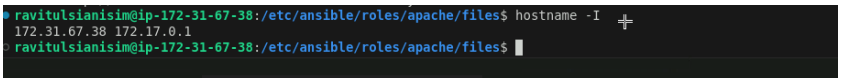
This means there are no errors, and we can now execute the YAML file.

6.13 Run the following command to execute the **setup.yml** playbook:
ansible-playbook -v setup.yml



Step 7: Check the index.html file in the browser

7.1 Run the following command to check the IP address of the server:
hostname -I



7.2 Open a web browser and navigate to the IP address of the server. If the IP address is **172.31.67.38**, enter the following URL in the browser:

By following the above steps, you have successfully created and executed an Ansible role to install and configure the Apache HTTP server. This process included initializing roles, defining and organizing tasks, and deploying the configuration. You can now verify the setup by accessing the index.html file through your server's IP address in a web browser. This approach ensures clarity, modularity, and ease of management for your infrastructure using Ansible roles.