Lesson 04 Demo 02

Configuring Apache Web Server Using Ansible

Objective: To configure and validate the setup of the Apache web server using Ansible on a local node machine for automated server management

Tools required: Linux terminal

Prerequisites: None

Steps to be followed:

1. Establish connectivity between the Ansible controller and node machine
2. Configure and validate the Apache web server setup

Step 1: Establish connectivity between the Ansible controller and node machine

1.1 Run the following command to generate a new SSH key pair using the RSA encryption algorithm:
ssh-keygen -t rsa

```
poojahksimplile@ip-172-31-79-37:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/poojahksimplile/.ssh/id_rsa):
Created directory '/home/poojahksimplile/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/poojahksimplile/.ssh/id_rsa
Your public key has been saved in /home/poojahksimplile/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2idghEiZ78rHsCaNAMoUm/E9ySn0GwYZ2tJRMWZYOj8 poojahksimplile@ip-172-31-79-
37
The key's randomart image is:
+---[RSA 3072]----+
|   .=*B.         |
| +**=..          |
| oO** +          |
|.+.o+@           |
|=  .oE* S        |
|o.. .o.+         |
|.+ =  . o .      |
|o * o    o       |
| o .             |
+----[SHA256]-----+
```

1.2 Run the following command to append the public key to the **authorized_keys** file:
cat .ssh/id_rsa.pub >> .ssh/authorized_keys

```
poojahksimplile@ip-172-31-79-37:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
poojahksimplile@ip-172-31-79-37:~$
```

1.3 Connect to localhost via SSH port 22 using the below command:
ssh localhost -p 22

```
poojahksimplile@ip-172-31-79-37:~$ ssh localhost -p 22
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1016-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Fri May  3 12:42:01 UTC 2024

  System load:  0.859375          Processes:                263
  Usage of /:   25.5% of 48.39GB  Users logged in:          0
  Memory usage: 24%               IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                IPv4 address for ens5:    172.31.79.37

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

59 updates can be applied immediately.
3 of these updates are standard security updates.
```

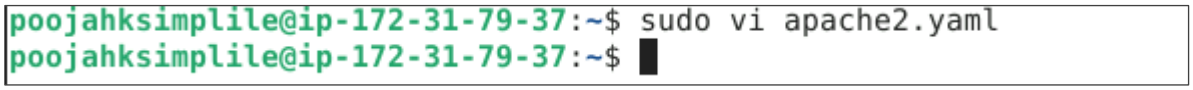
1.4 Run the following command to edit the Ansible **hosts** file:
sudo vi /etc/ansible/hosts

```
poojahksimplile@ip-172-31-79-37:~$ sudo vi /etc/ansible/hosts
poojahksimplile@ip-172-31-79-37:~$
```

1.5 Add the below two lines of the code inside the file as shown in the screenshot below:
[webservers]
localhost:22

Step 2: Configure and validate the Apache web server setup

2.1 Run the following command to edit the Apache configuration file:
sudo vi apache2.yaml

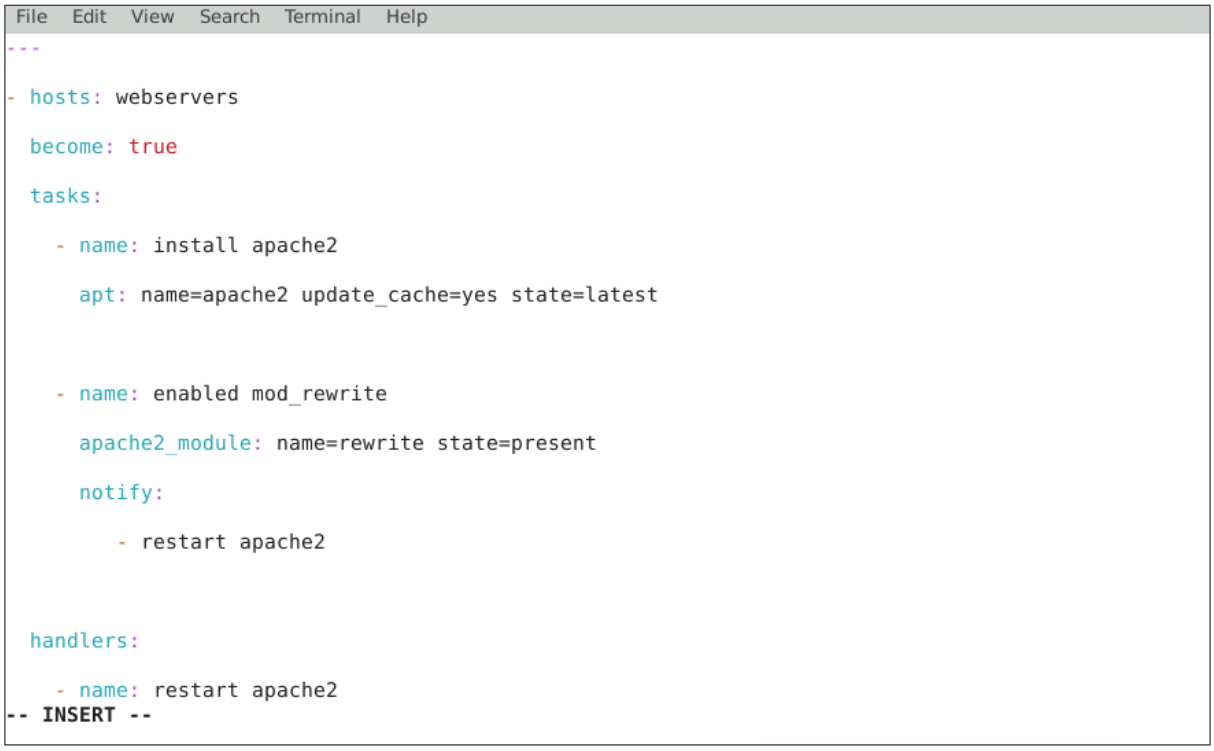


2.2 Enter the following script into the **apache2.yaml** file:

```
---
- hosts: webservers
  become: true
  tasks:
    - name: install apache2
      apt: name=apache2 update_cache=yes state=latest

    - name: enabled mod_rewrite
      apache2_module: name=rewrite state=present
      notify:
        - restart apache2

  handlers:
    - name: restart apache2
      service: name=apache2 state=restarted
```



2.3 Run the Apache installation playbook using the command below:
ansible-playbook apache2.yaml

```
poojahksimplile@ip-172-31-79-37:~$ ansible-playbook apache2.yaml

PLAY [webservers] ***************************************************************************************************

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

TASK [install apache2] **********************************************************************************************
changed: [localhost]

TASK [enabled mod_rewrite] ******************************************************************************************
changed: [localhost]

RUNNING HANDLER [restart apache2] ***********************************************************************************
changed: [localhost]

PLAY RECAP **********************************************************************************************************
localhost                  : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

poojahksimplile@ip-172-31-79-37:~$
```

2.4 Run the following Ansible ad hoc command to check the Apache server status:
ansible -m shell -a "service apache2 status" webservers

```
poojahksimplile@ip-172-31-79-37:~$ ansible -m shell -a "service apache2 status" webservers
localhost | CHANGED | rc=0 >>
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-05-03 12:51:49 UTC; 54s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 25059 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 25063 (apache2)
      Tasks: 55 (limit: 18607)
     Memory: 5.2M
        CPU: 31ms
     CGroup: /system.slice/apache2.service
             ├─25063 /usr/sbin/apache2 -k start
             ├─25064 /usr/sbin/apache2 -k start
             └─25065 /usr/sbin/apache2 -k start
poojahksimplile@ip-172-31-79-37:~$
```

By following these steps, you have successfully configured and validated the Apache web server setup using Ansible on a local node machine for automated server management.