

CLOUD COMPUTING

ASSIGNMENT NO. 6

NAME : SAYALI PREMRAJ SONTAKKE
PRN : 22010259
ROLL NO. : 333052
BATCH : C3

TITLE : Write a ansible playbook to deploy Apache Web server.

THEORY :

1) What is YAML

- YAML is a human-readable data serialization language that is often used for writing configuration files. Depending on whom you ask, YAML stands for yet another markup language or YAML ain't markup language (a recursive acronym), which emphasizes that YAML is for data, not documents.
- YAML is a popular programming language because it is designed to be easy to read and understand. It can also be used in conjunction with other programming languages. Because of its flexibility and accessibility, YAML is used by the Ansible automation tool to create automation processes, in the form of Ansible Playbooks.
- YAML is designed to be easy to read and write by humans, with a minimal syntax that is both expressive and unambiguous. It is often used in conjunction with programming languages like Python, Ruby, and JavaScript, and is supported by many libraries and frameworks. One of the main advantages of YAML is that it is platform-independent and can be used on any operating system. It is also very flexible and can be customized to suit the needs of different projects and workflows.

2) Introduction to Ansible

- **Ansible** is simple open source IT engine which automates application deployment, intra service orchestration, cloud provisioning and many other IT tools.
- Ansible is easy to deploy because it does not use any agents or custom security infrastructure.
- Ansible uses playbook to describe automation jobs, and playbook uses very simple language i.e. **YAML** (It's a human-readable data serialization language & is commonly used for configuration files, but could be used in many applications where data is being stored) which is very easy for humans to understand, read and write. Hence the advantage is that even the IT infrastructure support guys can read and understand the playbook and debug if needed (YAML – It is in human readable form).
- Ansible is designed for multi-tier deployment. Ansible does not manage one system at time, it models IT infrastructure by describing all of your systems are interrelated. Ansible is completely agentless which means Ansible works by connecting your nodes through ssh(by default). But if you want other method for connection like Kerberos, Ansible gives that option to you.
- After connecting to your nodes, Ansible pushes small programs called as “Ansible Modules”. Ansible runs that modules on your nodes and removes them when finished. Ansible manages your inventory in simple text files (These are the hosts file). Ansible uses the hosts file where one can group the hosts and can control the actions on a specific group in the playbooks.
- Ansible is easy to learn and use, and has a large and active community that contributes to its development and provides support to users. It also integrates with many other tools and platforms, including Docker, AWS, and Kubernetes. One of the main advantages of Ansible is that it is agentless, meaning that it can manage remote machines without requiring any software to be installed on the target hosts. This makes it easy to manage large and complex infrastructures without having to manage individual agents or deal with complex installation and configuration processes. Overall, Ansible is a powerful and flexible automation tool that can help you streamline your IT operations and improve the efficiency of your workflows.

Implementation:

1. Architecture:

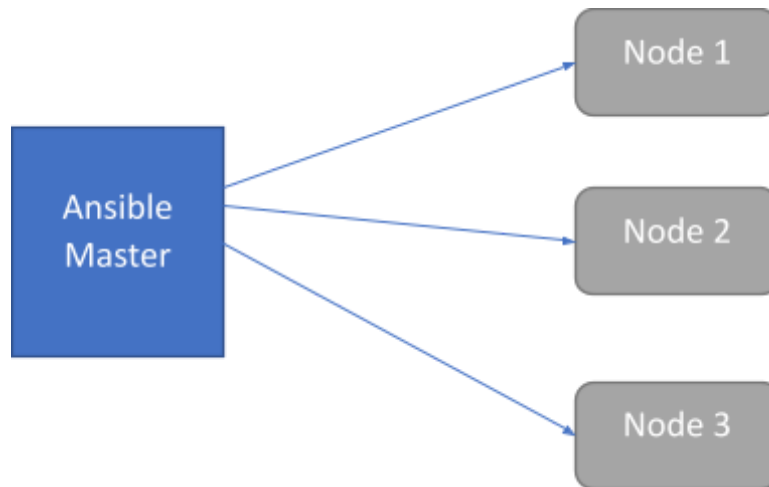


Figure1: Architecture Diagram

2. Steps

- Create 4 ec2 instances of **Ubuntu machine**.

The screenshot shows the AWS Management Console interface for EC2 instances. The top navigation bar includes the AWS logo, 'Services', a search bar, and user information. The main content area is titled 'Instances (5)' and includes a search bar, a table of instances, and a 'Launch instances' button. The table lists five instances: 'ubuntu_1', 'master', 'ubuntu_2', and 'ubuntu_3', all in a 'Running' state. The footer contains links for 'CloudShell', 'Feedback', 'Language', 'Privacy', 'Terms', and 'Cookie preferences'.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
ubuntu_1	i-01063ea23a2316772	Running	t2.micro	2/2 checks passed	No alarms
master	i-01b3ee3a769c874c6	Running	t2.micro	2/2 checks passed	No alarms
ubuntu_2	i-0eb76a46a5ecbf95b	Running	t2.micro	2/2 checks passed	No alarms
ubuntu_3	i-05baa92b5741180ac	Running	t2.micro	2/2 checks passed	No alarms

- b) Connect to “Ansible-Master” server
- c) Write following commands

- 1) > sudo -i
- 2) > apt update

```
ubuntu@ip-172-31-89-14: ~/s × + ▾
ubuntu_server_1_key.pem  ubuntu_server_1_key.pem:Zone.Identifier
root@SANKET-SUPEKAR:/cloud_devOps/Ansible# ssh -i "ubuntu_server_1_key.pem" ubuntu@ec2-54-159-203-69
.compute-1.amazonaws.com
The authenticity of host 'ec2-54-159-203-69.compute-1.amazonaws.com (54.159.203.69)' can't be establ
ished.
ED25519 key fingerprint is SHA256:g0LGBKNHgXRrtLEAk0ORGBfVqEkFS4jwB7YsaNj2Vck.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
ubuntu@ip-172-31-89-14: ~/s × + ▾
ubuntu@ip-172-31-89-14:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [728 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [990 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [210 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.9
kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [744 k
B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [147 kB]
```

```
ubuntu@ip-172-31-89-14: ~/s × + ▾
ubuntu@ip-172-31-89-14:~$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.6).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
ubuntu@ip-172-31-89-14:~$ sudo apt-add-repository ppa:ansible/ansible
sudo: apt-add-repository: command not found
ubuntu@ip-172-31-89-14:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier
to deploy. Avoid writing scripts or custom code to deploy and update your applications- automate in
a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
```

3) Install ansible using command

❓ apt install ansible

```
ubuntu@ip-172-31-89-14: ~/s × + ▾
ubuntu@ip-172-31-89-14:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl
  python3-ntlm-auth python3-packaging python3-paramiko
  python3-requests-kerberos python3-requests-ntlm python3-resolvelib
  python3-winrm python3-xlrd python3-xmldict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  python3-ntlm-auth python3-paramiko
Selecting previously unselected package python3-ntlm-auth.
Preparing to unpack .../07-python3-ntlm-auth_1.4.0-1_all.deb ...
Unpacking python3-ntlm-auth (1.4.0-1) ...
Selecting previously unselected package python3-paramiko.
Preparing to unpack .../08-python3-paramiko_2.9.3-0ubuntu1_all.deb ...
```

a) Generate a ssh key on Ansible-master using command

ssh-keygen

```
ubuntu@ip-172-31-89-14: ~/.ssh x + v
ubuntu@ip-172-31-89-14:~$ cd .ssh
ubuntu@ip-172-31-89-14:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-89-14:~/.ssh$ cd ..
ubuntu@ip-172-31-89-14:~$ ssh ubuntu@3.92.183.49
The authenticity of host '3.92.183.49 (3.92.183.49)' can't be established.
ED25519 key fingerprint is SHA256:iuBURHnTjFVS5V+gZe8cgSODLCMqoxFo9AAyngwzvkc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.92.183.49' (ED25519) to the list of known hosts.
ubuntu@3.92.183.49: Permission denied (publickey).
ubuntu@ip-172-31-89-14:~$ cd .ssh
ubuntu@ip-172-31-89-14:~/.ssh$ ls
authorized_keys  known_hosts
ubuntu@ip-172-31-89-14:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
```

```
ubuntu@ip-172-31-89-14: ~/.ssh x + v
ubuntu@ip-172-31-89-14:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
ubuntu@ip-172-31-89-14:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCYRXtfn6z10gB5SyyM3k5zhSyF3X6AnKH+Uz1jRGDRUtQVT
Rm7+39F+AX5ZA686qan+mx18oFFs7mKN+OWUSV96iB06Q0rgvXuaFcgy9+i5h6rHtA8v48efQiFZMRXKksrwQ
DTi90hkFQ5kNspGl830hna4dq+I/D6bgJyY45mMrx08KG0I3GfZ70r5zExLMVUVb9es9Gg8FcGpIXxUIaBNfS
2lKBGwyt6X6IGYeMnJ0iNYpvL6SnZhJeeT2/XaptyhI9EntudfzE0K2sBV10/ZduuKnDvTecbHjcY3HkodPE3
RjC3p9jhgXqRbBhpRZsxYCxKQYlyrAKcge06NUt0QMLL2Plab9mfLUwsD7wc2NezHRtVSAzaEhcHuwmeh672U
0ttgWxfhqjy8j+TzdCsoeWBIYw62oXAsmjNjbWeCQEpN0FJmRTE5mswq46WI1HjhkUhpNx+C9WR/kotm5Nb5h
QzdAygEI704oKQqV95vIu99BDYnKzjNamz7dLttSU= ubuntu@ip-172-31-89-14
```

- a) Now login to Ansible-master and try to connect to ansible server using command
- > ssh ubuntu@private-ip


```
ubuntu@ip-172-31-89-14: ~/s × + ▾
ubuntu@ip-172-31-89-14:~/.ssh$ ssh ubuntu@3.92.183.49
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

System information as of Sat Apr  8 08:41:54 UTC 2023

System load:  0.0          Processes:           99
Usage of /:   22.7% of 7.57GB Users logged in:        1
Memory usage: 25%          IPv4 address for eth0: 172.31.83.59
Swap usage:   0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

15 updates can be applied immediately.
12 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

```
ubuntu@ip-172-31-83-59: ~/s × + ▾
root@SANKET-SUPEKAR:/# cd cloud_devOps
root@SANKET-SUPEKAR:/cloud_devOps# cd Ansible
root@SANKET-SUPEKAR:/cloud_devOps/Ansible# ssh -i "ubuntu_server_1_key.pem" ubuntu@ec2-3-92-183-49.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

- a) copy the public key which is in .ssh folder into “authorized keys” on ansible-server1

commands:

- 1) ls ~/.ssh
- 2) cat ~/.ssh/id_rsa.pub

- b) connect to ansible-server1 and again give command

> ssh-keygen

It will create the same files on ansible-server1

Now,

> vim ~/.ssh/authorized_keys

and Copy the public key

```
ubuntu@ip-172-31-83-59: ~/.ssh$ python3 --version
Python 3.10.6
ubuntu@ip-172-31-83-59: ~/.ssh$ cd .ssh
ubuntu@ip-172-31-83-59: ~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-83-59: ~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-83-59: ~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDJxmZ8H1C7FJqGjTVr0BCGYrqgB7KBnYyenTWQCmyc5gt+xPX/
6yh6biManD15LDG0eYuQggakxzp9J3Ax89vp6oJj01cmca0VBZ41d8XEQ8FMv+9g4yLFngkCF5Rs9BAqMo4EBsWb
dPLUlojwQD4BMNm3ASmMhJ8CrBZxcvk3FJzdEZudsH2Fgs47dI5oLGhs/dGi9thz0M+viEaMEqhkh07+84QcTr9
JhBBPePRLi6iVqF+j0HKA1C3ZFUQHk9eFqvYS7LYErz4XzUfKwivumWqwbB1tMSVL00sdPLNEy9gN6w810X6n+/6
32FU8hDFbWh+mJSqJlqion6cylEN ubuntu_server_1_key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQCVRXtfn6z10gB5SyyM3k5zhisyF3X6AnKH+Uz1jRGDRUtQVTRm7
+39F+AX5ZA686qan+mx18oFfs7mKN+OWUSV96iB06Q0rgvXuaFcgy9+i5h6rHtA8v48efQiFZMRXKksrwQDTi90h
kFQ5kNspGL830hna4dq+I/D6bgJyY45mMrx08KG0I3GfZ70r5zExLMVUVb9es9Gg8FcGpIXxUIaBNfS2lKBGwyt6
X6IGYeMnJ0iNypvL6SnZhJeeT2/XaptyhI9EntudfzE0K2sBV10/ZduuKnDvTecbHjcY3HkodPE3RjC3p9jhgxr
bBhpRZsxYCxKQYlyrAKcge06NUT0QMLL2PLaB9mfLUwsD7wc2NezHRtVSAzaEhcHuwmeh672U0ttgWxfhqjy8j+T
zdCsoeWBIYw62oXAsmjNjbWeCQEpN0FJmRTE5mswq46WI1HjhkUhpNx+C9WR/kotm5Nb5hQzdAygEI704oKQQV95
vIu99BDYnKzjNamz7dLttSU= ubuntu@ip-172-31-89-14
ubuntu@ip-172-31-83-59: ~/.ssh$
```

```
ubuntu@ip-172-31-89-14: ~/.ssh$ ansible -m ping all
ubuntu_1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu_3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu_2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
ubuntu@ip-172-31-89-14: ~/.ssh$
```

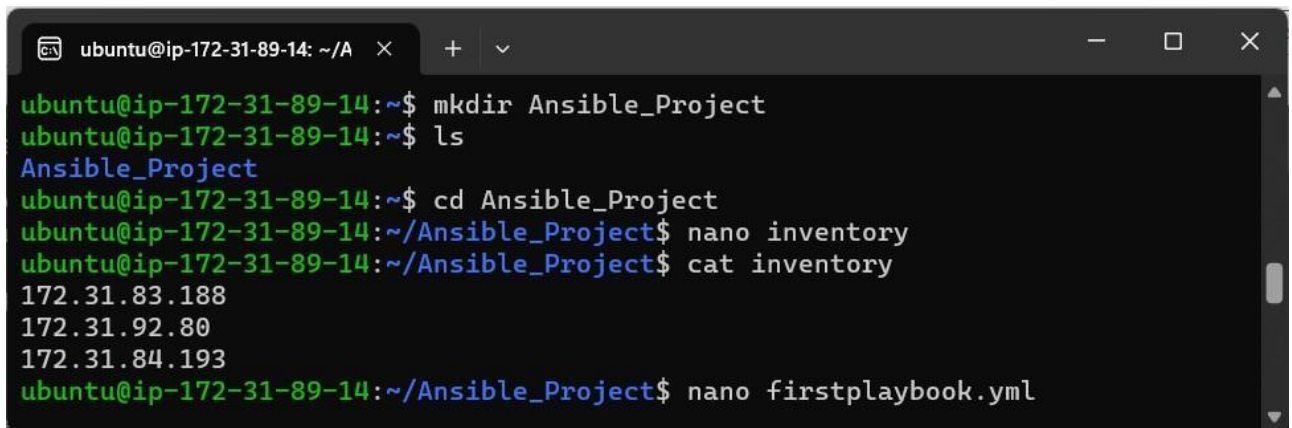
Create a playbook on Ansible-master

Step 1:- connect to “Ansible-Master”

Step 2:- create a new folder “ansible-project” using command

Step 3:

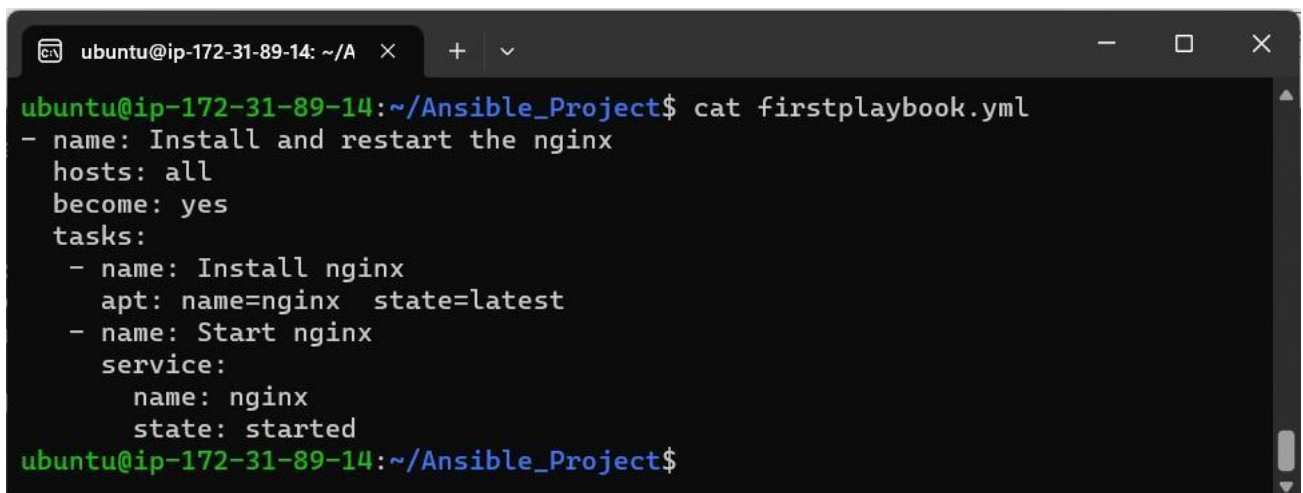
- a) > cd ansible-project
- b) > nano inventory
- c) > write a private IP of “Ansible-server1” into inventory
- d) > write a private IP of “Ansible-server2” into inventory
- e) > write a private IP of “Ansible-server3” into inventory



```
ubuntu@ip-172-31-89-14: ~/A × + v
ubuntu@ip-172-31-89-14:~$ mkdir Ansible_Project
ubuntu@ip-172-31-89-14:~$ ls
Ansible_Project
ubuntu@ip-172-31-89-14:~$ cd Ansible_Project
ubuntu@ip-172-31-89-14:~/Ansible_Project$ nano inventory
ubuntu@ip-172-31-89-14:~/Ansible_Project$ cat inventory
172.31.83.188
172.31.92.80
172.31.84.193
ubuntu@ip-172-31-89-14:~/Ansible_Project$ nano firstplaybook.yml
```

Task: Install Nginx and Start Nginx

Step 1: Create a new file called “first-playbook.yml”



```
ubuntu@ip-172-31-89-14: ~/A × + v
ubuntu@ip-172-31-89-14:~/Ansible_Project$ cat firstplaybook.yml
- name: Install and restart the nginx
  hosts: all
  become: yes
  tasks:
    - name: Install nginx
      apt: name=nginx state=latest
    - name: Start nginx
      service:
        name: nginx
        state: started
ubuntu@ip-172-31-89-14:~/Ansible_Project$
```

Execute the playbook by using command:

Ansible-playbook -i inventory first-playbook.yml

```
ubuntu@ip-172-31-89-14: ~/Ansible_Project$ ansible-playbook -i inventory firstplaybook.yml

PLAY [Install and restart the nginx] *****

TASK [Gathering Facts] *****
The authenticity of host '172.31.84.193 (172.31.84.193)' can't be established.
ED25519 key fingerprint is SHA256:sPBy2x/zgrn6dTK8wHTWZpJL6VOMsI5ERjqXfpM26XU.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? ok: [172.31.83.188]
ok: [172.31.92.80]
```

Verify the output:

Step 1: connect to any ansible-server1

Step run the command: **sudo systemctl status nginx**

```
ubuntu@ip-172-31-92-80: ~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-04-08 10:22:16 UTC; 18s ago
     Docs: man:nginx(8)
  Process: 2313 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 2314 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 2407 (nginx)
    Tasks: 2 (limit: 1141)
   Memory: 6.1M
      CPU: 24ms
  CGroup: /system.slice/nginx.service
          └─2407 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2410 "nginx: worker process"

Apr 08 10:22:16 ip-172-31-92-80 systemd[1]: Starting A high performance web server and a reverse proxy server...
Apr 08 10:22:16 ip-172-31-92-80 systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-16/16 (END)
```

CONCLUSION : In this assignment, We learned how to write **ansible** playbook to deploy **Apache Web server**.