

Name – Majahar Mahamud Kazi

Div – B

Batch – B2

Roll no. – 322036

PRN no. – 22110729

Assignment 6

Aim: Write an ansible-playbook to install nginx on target servers.

Theory:

A. What is YAML ?

YAML stands for "YAML Ain't Markup Language". It is a human-readable data serialization format used for configuring applications, storing data, and exchanging information between systems. It is often used in configuration files and is used as a replacement for XML or JSON. YAML uses indentation to define the structure of data and uses a minimalistic syntax that makes it easy to read and write by humans.

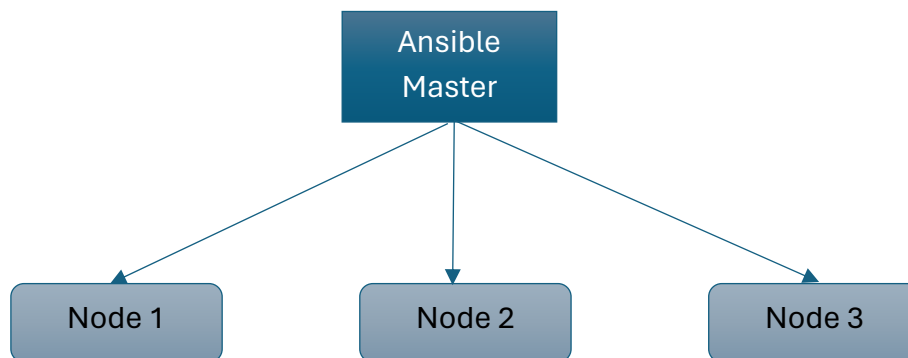
B. What is Ansible ?

Ansible is an open-source automation tool used for configuration management, application deployment, and orchestration. It uses a declarative language called YAML to define the configuration of systems and applications, and it executes tasks in parallel on multiple systems over SSH. Ansible is designed to be simple, easy to use, and highly scalable, and it does not require any agents or additional software to be installed on the target systems. Ansible can be used to automate tasks such as server provisioning, software installation, system updates, and application deployment, among others.



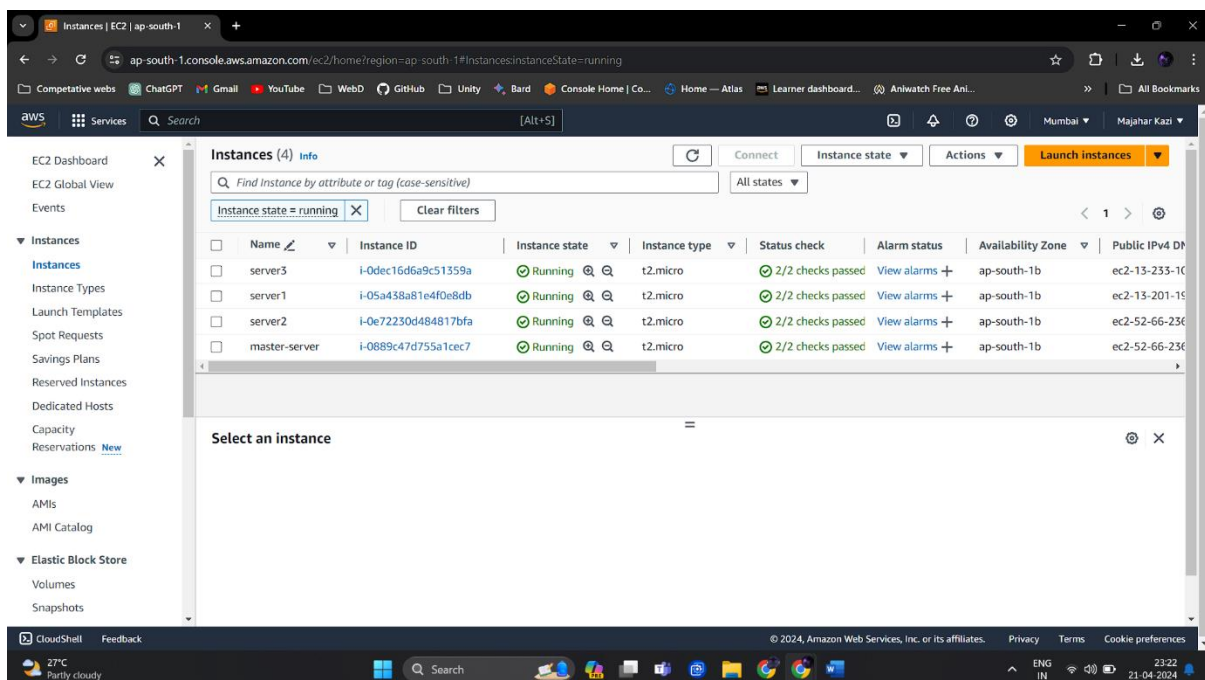
STEPWISE IMPLEMENTATION:

Architecture:



Steps:

Create 4 EC2 Instances on AWS



Here we have Created 1 Master & 3 Slaves Servers.

Run 'yum update -y' on all Instances & Install ansible on the master server.

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-1-165 ~]$ sudo -i
[root@ip-172-31-1-165 ~]# yum update -y
Last metadata expiration check: 0:05:52 ago on Sun Apr 21 17:50:46 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-1-165 ~]#
```

i-0889c47d755a1cec7 (master-server)
PublicIPs: 52.66.236.68 PrivateIPs: 172.31.1.165

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-12-16 ~]$ sudo -i
[root@ip-172-31-12-16 ~]# yum update -y
Last metadata expiration check: 0:07:04 ago on Sun Apr 21 17:48:21 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-12-16 ~]#
```

i-05a438a81e4f0e8db (server1)
PublicIPs: 13.201.192.132 PrivateIPs: 172.31.12.16

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-13-147 ~]$ sudo -i
[root@ip-172-31-13-147 ~]# yum update -y
Last metadata expiration check: 0:06:45 ago on Sun Apr 21 17:49:08 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-13-147 ~]#
```

i-0e72230d484817bfa (server2)
PublicIPs: 52.66.236.252 PrivateIPs: 172.31.13.147

```
[root@ip-172-31-7-57 ~]# yum update -y
Last metadata expiration check: 0:05:16 ago on Sun Apr 21 17:49:36 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-7-57 ~]#
```

i-0dec16d6a9c51359a (server3)
PublicIPs: 13.233.100.163 PrivateIPs: 172.31.7.57



Now copy the public key from the master server and paste it into the authorized_keys of the slave servers.

```
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-83-221:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:hhahoc7dmsc40sa+ucCzodfWudSSgwmfVbxdq3c0Y ubuntu@ip-172-31-83-221
The key's randomart image is:
+----[RSA 3072]-----+
|  o*+.. .o. .      |
| .==+.. .o        |
| .==+o . o E      |
| =..* . . o       |
| o..o . S o o     |
| .+o. o . +       |
| .+o+..           |
| o+.o.            |
| o ..             |
+----[SHA256]-----+
ubuntu@ip-172-31-83-221:~$ cd .ssh
ubuntu@ip-172-31-83-221:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-83-221:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-83-221:~/.ssh$
```

i-020bf2f9adb1ff21 (Ansible-Server1)
PublicIPs: 107.20.32.216 PrivateIPs: 172.31.83.221

```
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages (796 KB)
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en (156 KB)
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata (19024 B)
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages (830 KB)
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en (131 KB)
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata (604 B)
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages (727 KB)
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en (120 KB)
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata (14.2 KB)
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages (34.3 KB)
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en (6464 B)
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata (252 B)
Fetched 27.1 MB in 5s (5695 KB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
35 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-81-106:~$ sudo apt-get update
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
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-81-106:~$ cd .ssh
ubuntu@ip-172-31-81-106:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-81-106:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-81-106:~/.ssh$
```

i-0e760ee713c7d812b (Ansible-Server2)
PublicIPs: 5.87.13.163 PrivateIPs: 172.31.81.106

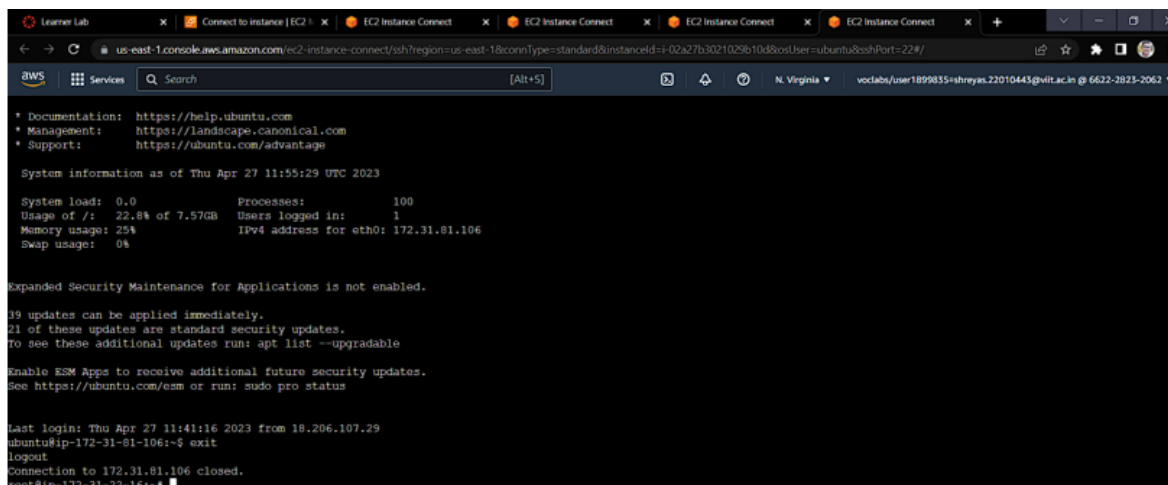


To check for ssh authentication:

ssh ubuntu@PrivateIP

```
i-020bf2f9adbd1ff21 (Ansible-Server1)
PublicIPs: 107.20.32.216 PrivateIPs: 172.31.83.221

i-0e760ee713c7d812b (Ansible-Server2)
PublicIPs: 3.87.13.163 PrivateIPs: 172.31.81.106
```



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

System information as of Thu Apr 27 11:55:29 UTC 2023

System load: 0.0 Processes: 100
Usage of /: 22.8% of 7.57GB Users logged in: 1
Memory usage: 25% IPv4 address for eth0: 172.31.81.106
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

39 updates can be applied immediately.
21 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

Last login: Thu Apr 27 11:41:16 2023 from 18.206.107.29
ubuntu@ip-172-31-81-106:~$ exit
logout
Connection to 172.31.81.106 closed.
root@ip-172-31-22-16:~#
```

```
i-02a27b3021029b10d (Ansible-Master)
PublicIPs: 54.87.55.46 PrivateIPs: 172.31.22.16
```

Here we have cross verified the ssh-authentication for server-2 Use the exit command to close the ssh connection

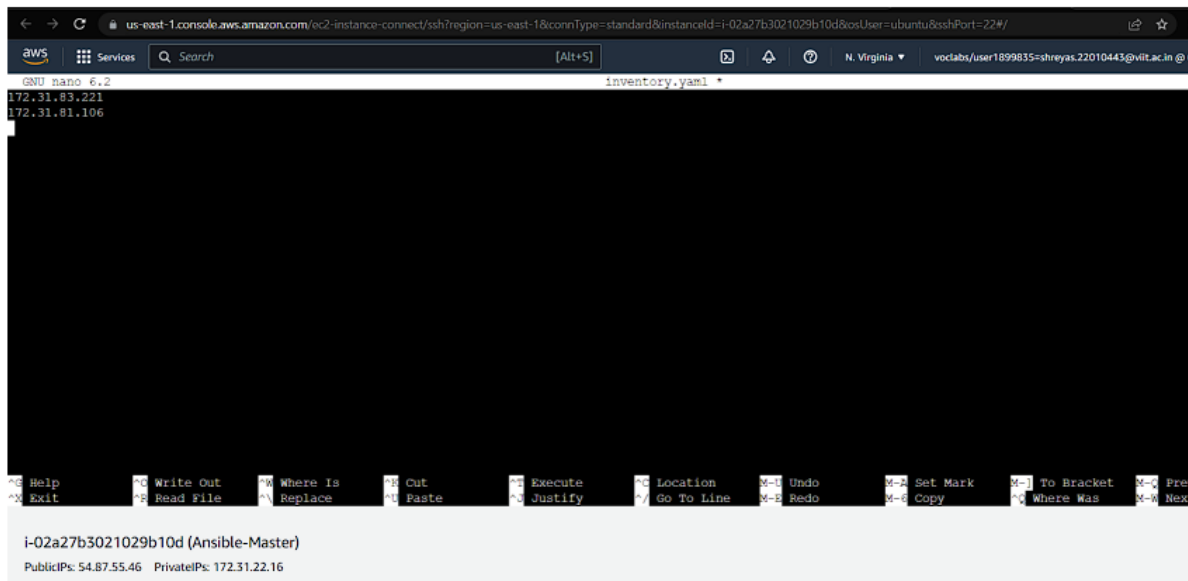
make a directory named ansible using the mkdir ansible

Create two yaml files namely

1. inventory.yaml
2. playbook.yaml

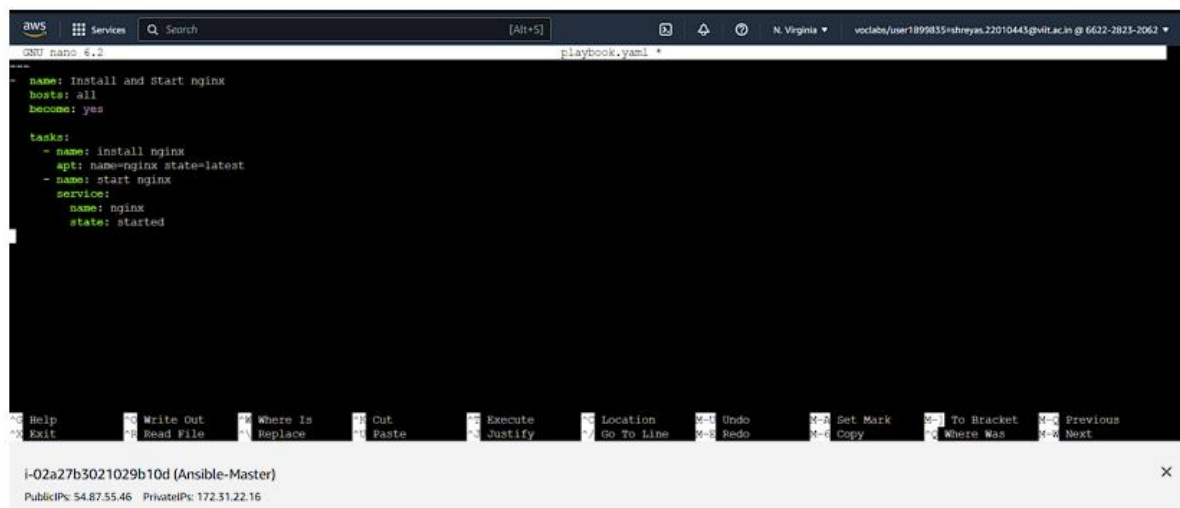
The inventory.yaml file must consist of the the private IP addresses of the slave servers





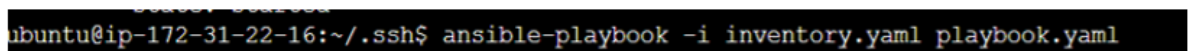
```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-02a27b3021029b10d&osUser=ubuntu&sshPort=22#/  
GNU nano 6.2 inventory.yaml *  
172.31.83.221  
172.31.81.106  
  
i-02a27b3021029b10d (Ansible-Master)  
PublicIPs: 54.87.55.46 PrivateIPs: 172.31.22.16
```

The `playbook.yaml` file must contain the ansible script which is to be executed.



```
GNU nano 6.2 playbook.yaml *  
- name: Install and Start nginx  
  hosts: all  
  become: yes  
  
  tasks:  
    - name: install nginx  
      apt: name=nginx state=latest  
    - name: start nginx  
      service:  
        name: nginx  
        state: started  
  
i-02a27b3021029b10d (Ansible-Master)  
PublicIPs: 54.87.55.46 PrivateIPs: 172.31.22.16
```

Run the `playbook.yaml` & `inventory.yaml` files



```
ubuntu@ip-172-31-22-16:~/.ssh$ ansible-playbook -i inventory.yaml playbook.yaml
```



Output:

```
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user1899835@shreyas.22010443@vill.ac.in @ 6622-2823-2062

ubuntu@ip-172-31-22-16:~/.ssh$ cat playbook.yaml
---
- name: Install and Start nginx
  hosts: all
  become: yes

  tasks:
    - name: install nginx
      apt: name=nginx state=latest
    - name: start nginx
      service:
        name: nginx
        state: started
ubuntu@ip-172-31-22-16:~/.ssh$ ansible-playbook -i inventory.yaml playbook.yaml
[WARNING]: Unable to parse /home/ubuntu/.ssh/inventory.yaml as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [install and Start nginx] *****
skipping: no hosts matched

PLAY RECAP *****
ubuntu@ip-172-31-22-16:~/.ssh$
```

i-02a27b3021029b10d (Ansible-Master)
PublicIPs: 54.87.55.46 PrivateIPs: 172.31.22.16

Conclusion:

We have successfully written an ansible-playbook to install nginx on target servers. We have also explored domains like ansible and yaml.

Author – Majahar Kazi

Sunday 21 April 2024 23:54:40 PM IST

