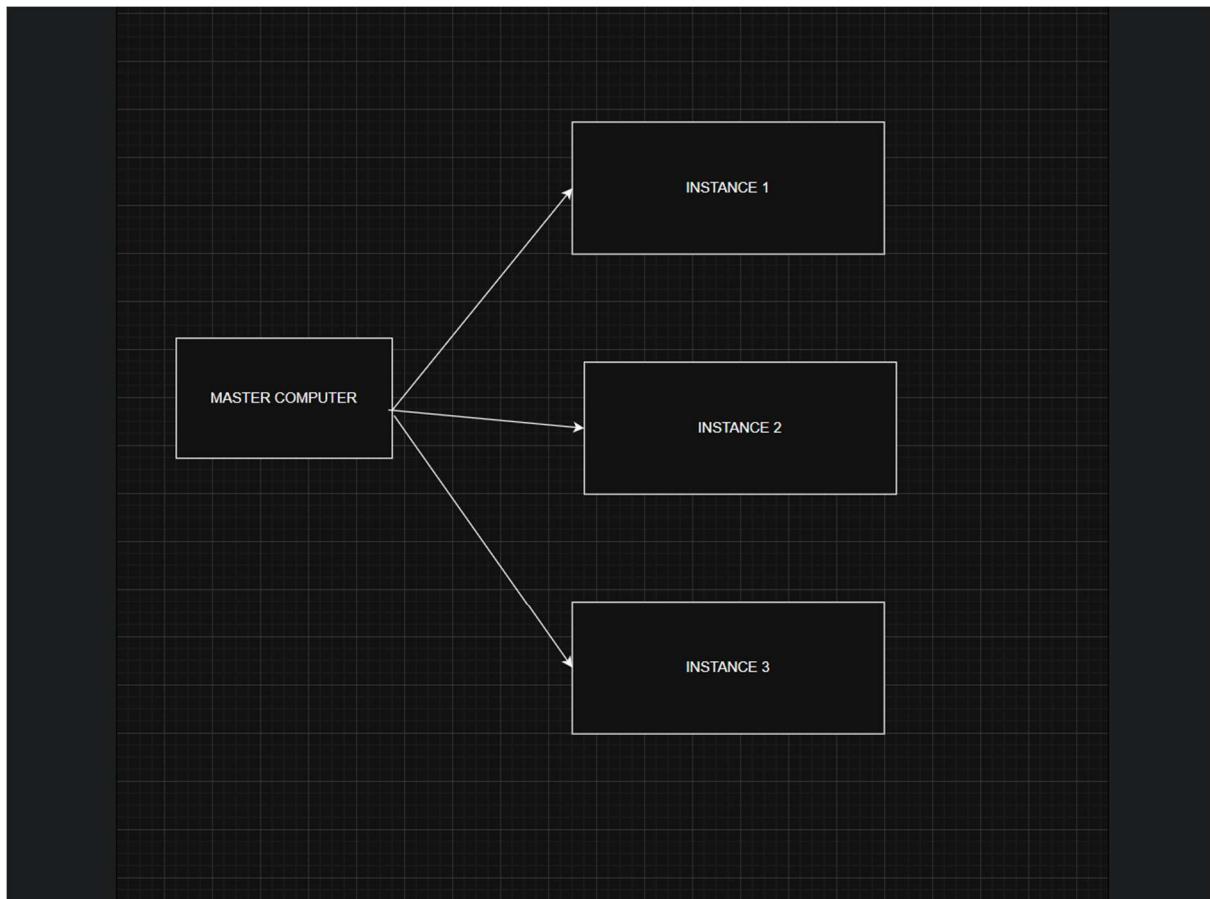


Name:Roshan Saral Kumar

## ANSIBLE PROJECT DEPLOYING AN APPLICATION USING PLAYBOOKS

ARCHITECTURE 1 FOR ANSIBLE:-



IN MASTER COMPUTER I HAVE INSTALLED THE ANSIBLE AND TRYING TO MAKE UPDATES AND CONFIGURATION MANAGEMENTS IN INSTANCES 1 , 2 AND 3.THE ARROWS REPRESENT HOW THE UPDATES ARE HAPPENING FOR DIFFERENT EC2 INSTANCES IN AWS.

WE ARE USING ANSIBLE WHICH IS AN OPEN SOURCE TOOL FOR CONFIGURATION MANAGEMENT AND IAAC(Infrastructure as code).

DIFFERENCE BETWEEN ANSIBLE AND TERRAFORM:-

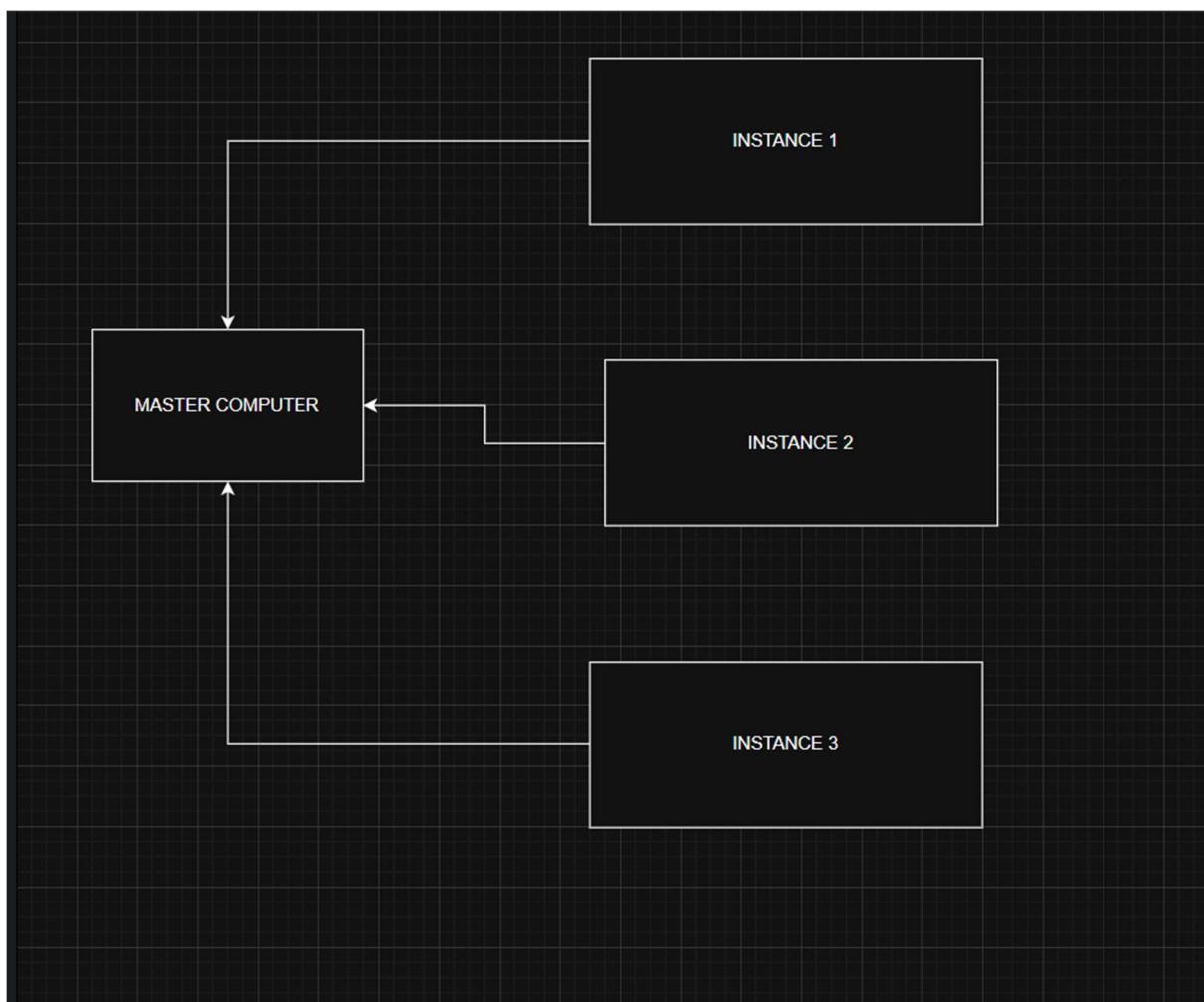
ANSIBLE	TERRAFORM
CONFIGURATION MANAGEMENT	INFRASTRUCTURE PROVISIONING
INSTALLS DEPENDENCIES	DOES NOT INSTALL DEPENDENCIES
USEFULL FOR PATCHING AND SOFTWARE UPDATES	NOT USEFUL FOR PATCHING AND SOFTWARE UPDATES
NOT USEFUL FOR BUILDING	VERY USEFULL FOR BUILDING

APPLICATIONS	APPLICATIONS AND INSTANCES IN AWS AND OTHER CLOUD PLATFORMS
--------------	---

DIFFERENCE BETWEEN CHEF AND ANSIBLE:-

CHEF	ANSIBLE
PULL BASED MECHANISM	PUSH BASED MECHANISM
IT TAKES OR PULLS UPDATES FROM MULTIPLE SERVERS AND UPDATES THE MASTER COMPUTER	IT PUSHES UPDATES TO MULTIPLE SERVERS AND INSTALLS DEPENDENCIES AND UPDATES PATCHES FOR MULTIPLE SERVERS

ARCHITECTURE 2 FOR CHEF:-



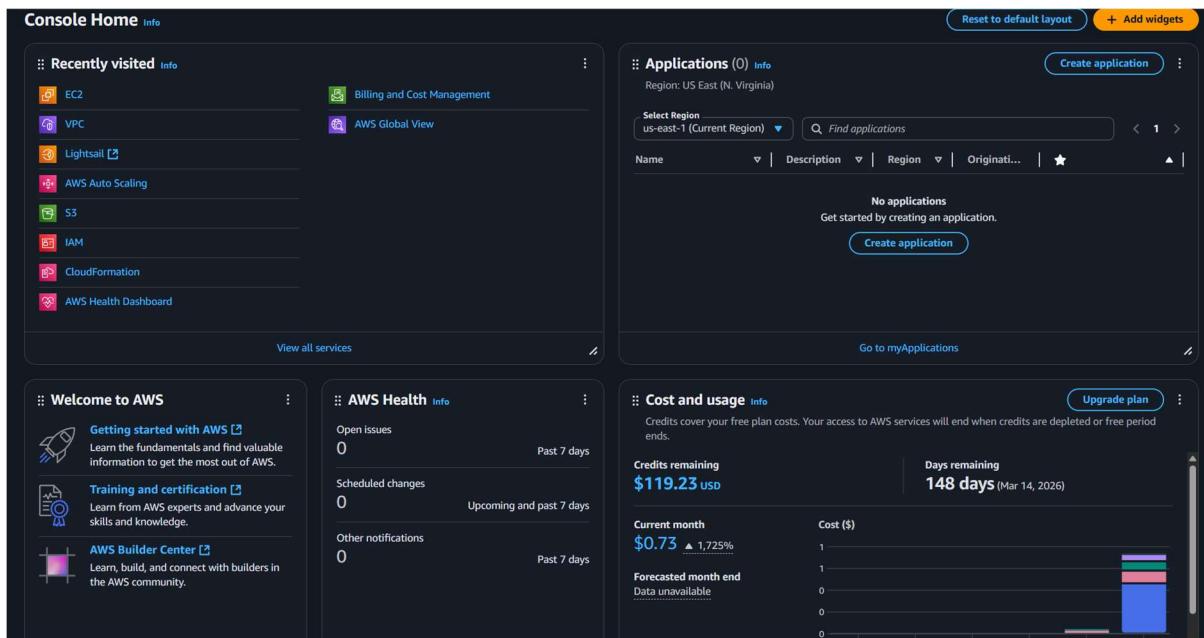
AS A DEVOPS ENGINEER TWO WAYS I CAN UPDATE THE SERVICES IS BY PUSH AND PULL ACTIONS

IF WE SEE THE ARCHITECTURES CAREFULLY THE ARROWS ARE CHANGING THE DIRECTIONS IN **ARCHITECTURE 1** IS TOWARDS THE INSTANCES WHICH MEANS THE MASTER COMPUTER WHERE ANSIBLE IS INSTALLED IS PUSHING THE UPDATES AND SOFTWARE PATCHES TO THE INSTANCES(**PUSH MECHANISM**). .THE DIRECTIONS IN

**ARCHITECTURE 2 IS FROM THE EC2 INSTANCES TOWARDS THE MASTER COMPUTER WHICH MEANS CHEF IS INSTALLED IN THE MASTER COMPUTER WHICH ACTS AS A PULL MECHANISM AND TAKES THE SOFTWARE UPDATES AND PATCHES FROM OTHER SERVERS AND UPDATES THE MASTER COMPUTER.**

WORKING OF ANSIBLE TO CREATE A PLAYBOOK IN AN EC2 INSTANCE:-

STEP 1 :- LOGIN TO AWS MANAGEMENT CONSOLE



STEP 2 :- SEARCH FOR EC2 INSTANCE AND CLICK ON LAUNCH EC2 INSTANCE.

The screenshot shows the AWS Services page with a search bar at the top containing "EC2". On the left, there's a sidebar with "Services" and various links like Features, Resources, Documentation, etc. The main content area has three sections: "Services", "Features", and "Resources". The "Services" section highlights the EC2 service with its icon and description: "Virtual Servers in the Cloud". The "Features" section includes "EC2 Instances" (CloudWatch feature), "EC2 Resource Health" (CloudWatch feature), and "Dashboard" (EC2 feature). Below these is a "Resources in us-east-1" search bar with a "Show more in Resource Explorer" link. A callout box provides information about cross-Region search, with a "Enable cross-Region search" button.

AFTER DOING THIS GIVE THE NAME FOR INSTANCE AS “Ansible-master” AND WE ARE USING OS AS UBUNTU. And using t3.micro as the instance type.

The screenshot shows the "Launch an instance" wizard. The first step, "Name and tags", has a "Name" field containing "Ansible-master". There's a "Add additional tags" button. The second step, "Application and OS Images (Amazon Machine Image)", shows a search bar and a grid of recent and quick start AMIs. One AMI, "Ubuntu Server 24.04 LTS (HVM), SSD Volume Type", is selected. It has a "Free tier eligible" badge. The "Description" section below it provides details about the selected AMI.

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type ami-0360c520857e3138f (64-bit (x86)) / ami-026fcd88446aa0bf (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible ▾		
<b>Description</b> Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ). Canonical, Ubuntu, 24.04, amd64 noble image			
<b>Architecture</b> 64-bit (x86) ▾	<b>AMI ID</b> ami-0360c520857e3138f	<b>Publish Date</b> 2025-08-21	<b>Username</b>   ⓘ ubuntu
<b>Verified provider</b>			

**▼ Instance type** [Info](#) | [Get advice](#)

<b>Instance type</b>	Free tier eligible
t3.micro	All generations
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour On-Demand Windows base pricing: 0.0196 USD per Hour	<a href="#">Compare instance types</a>

**Additional costs apply for AMIs with pre-installed software**

WE NEED TO SSH TO THE INSTANCES TO ACCESS THE TERMINAL OF THE INSTANCES. KEY VALUE PAIR HERE IS COMPULSORY BECAUSE IT IS USEFUL FOR PROVIDING AUTHENTICATION FOR THE OTHER SYSTEMS OR INSTANCES.

**Create key pair** X

**Key pair name**  
Key pairs allow you to connect to your instance securely.  
 Ansible-key-project-demo

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

<input checked="" type="radio"/> <b>RSA</b> RSA encrypted private and public key pair	<input type="radio"/> <b>ED25519</b> ED25519 encrypted private and public key pair
--	---

**Private key file format**

<input checked="" type="radio"/> <b>.pem</b> For use with OpenSSH	<input type="radio"/> <b>.ppk</b> For use with PuTTY
--	---

**⚠️** When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#) ↗

[Cancel](#) **Create key pair**

KEY VALUE PAIR CREATED. WE HAVE TO USE THE SAME KEY VALUE PAIR FOR ALL THE OTHER 3 INSTANCES CREATED AS SHOWN IN THE ARCHITECTURE 1 FOR ANSIBLE.

NOW AFTER THIS IAM ALLOWING EVEN THE HTTP AND HTTPS TRAFFIC FROM THE NETWORK SETTINGS.

The screenshot shows the 'Network settings' configuration page for an EC2 instance. It includes sections for Network, Subnet, Auto-assign public IP, and Firewall (security groups). The Firewall section is active, showing three checkboxes for allowing SSH, HTTPS, and HTTP traffic from anywhere. A note at the bottom of this section advises against using 0.0.0.0/0 and recommends setting security group rules for known IP addresses. Below this is the 'Configure storage' section, which shows a single volume of 8 GiB gp3 type, labeled as a Root volume with 3000 IOPS and Not encrypted. There is also an 'Add new volume' button.

NOW AFTER LAUNCHING OUR EC2 INSTANCE OUR ANSIBLE MASTER COMPUTER IS LAUNCHED.

The screenshot shows the 'Instances' page with a single entry for the 'Ansible-master' instance. The instance is listed as 'Running' with the ID i-04021edceaf72ce. It is an t3.micro instance in us-east-1b availability zone, with a public IPv4 address of 3.208.1.48. The page includes filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4, Elastic IP, and IPv6 IPs.

AFTER LAUNCHING THE MASTER COMPUTER I HAVE TO LAUNCH THE OTHER 3 INSTANCES BY SLECTING THE NUMBE ROF INSTANCES AS 3 AND TYPING SERVER.

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with 'HelloWorld' and a 'Create New Function' button. The main area has tabs for 'Code' (selected), 'Overview', 'Logs', 'Metrics', and 'Actions'. Under the 'Code' tab, there's a 'Lambda@Edge' section with a 'Create new Lambda@Edge function' button. Below that is a 'Lambda function' section with a 'Create new Lambda function' button. The bottom of the screen shows the AWS navigation bar.

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with 'HelloWorld' and a 'Create New Function' button. The main area has tabs for 'Code' (selected), 'Overview', 'Logs', 'Metrics', and 'Actions'. Under the 'Code' tab, there's a 'Lambda@Edge' section with a 'Create new Lambda@Edge function' button. Below that is a 'Lambda function' section with a 'Create new Lambda function' button. The bottom of the screen shows the AWS navigation bar.

FORTHOSE OTHER 3 INSTANCES ALSO WE ARE USING THE SAME KEY VALUE PAIR USED IN ANSIBLE MASTER COMPUTER THE ABOVE FIGURE SHOWS THAT AND WE ARE ALLOWING HTTP AND HTTPS TRAFFIC ALSO IN THE NETWORK SETTINGS.

WE ARE GOING TO LUNCH THE 3 SERVERS IN ONE GO.

<input type="checkbox"/>	server	i-095352b976ac6da49	<span>Running</span>	t3.micro	<span>Initializing</span>	<a href="#">View alarms</a>	us-east-1b	ec2-54-91-194-123.co...	54.91.194.123	-	-
<input type="checkbox"/>	server	i-06392da9362850d2	<span>Running</span>	t3.micro	<span>Initializing</span>	<a href="#">View alarms</a>	us-east-1b	ec2-18-209-46-0.compute...	18.209.46.0	-	-
<input type="checkbox"/>	server	i-09f99e26df29d1db0	<span>Running</span>	t3.micro	<span>Initializing</span>	<a href="#">View alarms</a>	us-east-1b	ec2-44-223-26-66.compute...	44.223.26.66	-	-
<input type="checkbox"/>	Ansible-master	i-04021edcceaf72ce	<span>Running</span>	t3.micro	<span>0/3 checks passed</span>	<a href="#">View alarms</a>	us-east-1b	ec2-3-208-1-48.compute...	3.208.1.48	-	-

3 MORE INSTANCES NAMED AS SERVER IS CREATED AND LAUNCHED.

I AM NOT GOING TO TOUCH SERVERS 1,2 AND 3 INSTEAD I WILL ONLY USE ANSIBLE MASTER COMPUTER TO UPDATE THE REMOTE SERVERS.

CONNECTING TO MY ANSIBLE-MASTER USING CONNECT PROVIDED BY AWS.

**Connect** Info

Connect to an instance using the browser-based client.

EC2 Instance Connect    Session Manager    SSH client    EC2 serial console

Instance ID  
 i-04021edcceaf72ce (Ansible-master)

Connection type

Connect using a Public IP  
Connect using a public IPv4 or IPv6 address

Connect using a Private IP  
Connect using a private IP address and a VPC endpoint

Public IPv4 address  
 3.208.1.48

IPv6 address

Username  
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu X

ⓘ Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

EC2 Instance Connect    Session Manager    **SSH client**    EC2 serial console

Instance ID  
 i-04021edcceaf72ce (Ansible-master)

1. Open an SSH client.  
2. Locate your private key file. The key used to launch this instance is Ansible-key-project-demo.pem  
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 chmod 400 "Ansible-key-project-demo.pem"  
4. Connect to your instance using its Public DNS:  
 ec2-3-208-1-48.compute-1.amazonaws.com

Example:  
 ssh -i "Ansible-key-project-demo.pem" ubuntu@ec2-3-208-1-48.compute-1.amazonaws.com

CLICK ON SSH CLIENT AND COPY THE EXAMPLE PORTION.

I HAVE OPENED MY WSL TERMINAL AND I AM INSIDE THE ANSIBLE MASTER COMPUTER.BYD DOING AN SSH.

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\rosha> WSL
roshan@Roshan-S:/mnt/c/Users/rosha$ cd Downloads
roshan@Roshan-S:/mnt/c/Users/rosha/Downloads$ sudo ssh -i "Ansible-key-project-demo.pem" ubuntu@ec2-3-208-1-48.compute-1.amazonaws.com
[sudo] password for roshan:
Sorry, try again.
[sudo] password for roshan:
The authenticity of host 'ec2-3-208-1-48.compute-1.amazonaws.com (3.208.1.48)' can't be established.
ECDSA key fingerprint is SHA256:06cWYeHHiTz3bZWSPWFkNHrhw8q2yS297Qyyqbuj4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-208-1-48.compute-1.amazonaws.com,3.208.1.48' (ECDSA) to the list of known hosts.
Welcomme to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

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

System information as of Sun Oct 19 04:37:05 UTC 2025

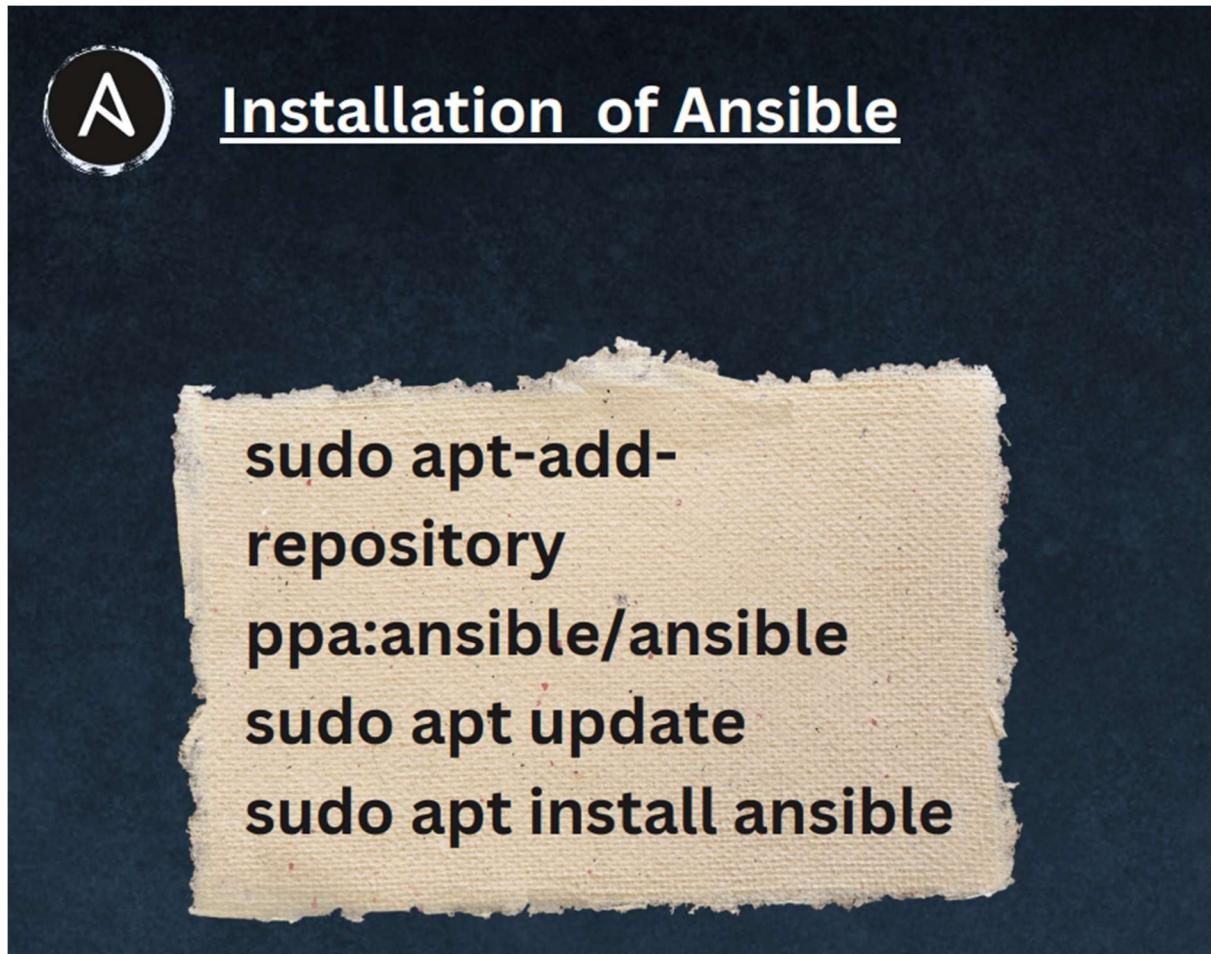
System load:  0.08      Temperature:        -273.1 °C
Usage of /:   25.6% of 6.71GB  Processes:          109
Memory usage: 24%           Users logged in:     0
Swap usage:   0%            IPv4 address for ens5: 172.31.31.208
```

THIS INDICATES IAM INSIDE THE PRIVATE IPV4 ADDRESS OF THE ANSIBLE MASTER COMPUTER.

```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

```
ubuntu@ip-172-31-31-208:~$
```

INSTALL ANSIBLE IN THE INSTANCE:-



ERROR FACED SO USING THE HIGHLIGHTED COMMAND INS TALLING THE PACKAGE.

```
ubuntu@ip-172-31-31-208:~$ sudo apt-addrepository  
ppa:ansible/ansible  
sudo: apt-addrepository: command not found  
-bash: ppa:ansible/ansible: No such file or directory  
ubuntu@ip-172-31-31-208:~$ sudo apt install software-properties-common -y
```

USING THIS TO ADD THE ANSIBLE REPOSITORY:-

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
```

UPDATING THE PACKAGES USING “sudo apt update” command..

```
ubuntu@ip-172-31-31-208:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu noble InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
53 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

AFTER CREATING REPO THEN UPDATING AND FINALYY INSTALLING ANSIBLE IN MY MASTER COMPUTER

```
ubuntu@ip-172-31-31-208:~$ 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-kerberos python3-nacl python3-ntlm-auth python3-paramiko python3-requests-ntlm
  python3-resolvelib python3-winrm python3-xmltodict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-kerberos python3-nacl python3-ntlm-auth python3-paramiko python3-requests-ntlm
  python3-resolvelib python3-winrm python3-xmltodict sshpass
0 upgraded, 11 newly installed, 0 to remove and 53 not upgraded.
Need to get 20.8 MB of archives.
After this operation, 235 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-resolvelib all 1.0.1-1 [25.7 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-kerberos amd64 1.1.14-3.1build9 [21.2 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-nacl amd64 1.5.0-4build1 [57.9 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 python3-ntlm-auth all 1.5.0-1 [21.3 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 python3-paramiko all 2.12.0-2ubuntu4.1 [13
```

THIS INDICATES THAT ANSIBLE IS INSTALLED.

```
ubuntu@ip-172-31-31-208:~$ ansible --version
ansible [core 2.18.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Jun 18 2025, 17:59:45) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-31-208:~$
```

AFTER THIS WE HAVE TO UPDATE THE HOST FILES WHICH CONTAINS THE INFORMATION OF OTHER INSTANCES IN ANSIBLE MASTER COMPUTER BY USING THIS COMMAND WE CAN CONFIGURE THE HOST FILE PRESENT IN “/etc/Ansible/hosts” path.

```
ubuntu@ip-172-31-31-208:~$ sudo vim /etc/ansible/hosts
```

```
# Ex 2: A collection of hosts belonging to the 'webservers' group:
[servers]
server_1 ansible_host=54.91.194.123
server_2 ansible_host=18.209.46.0
server_3 ansible_host=44.223.26.66
```

in the host file ima modifying the groups or iam configuring the ipv4 adreeses of ever y server by creating a servers grup enclosed in “[servers]”.

AFTER CONFIGURING THE OST FILE WE CAN MAKE A KEYS DIRECTORY TO STORE THE PRIVATE KEYS IN ORDER TO CONNECT THE OTHER INSTANCES. AND BE IN THE DIRECTORY OF “keys”.

```
ubuntu@ip-172-31-31-208:~$ sudo vim /etc/ansible/hosts
ubuntu@ip-172-31-31-208:~$ sudo vim /etc/ansible/hosts
ubuntu@ip-172-31-31-208:~$ mkdir keys
ubuntu@ip-172-31-31-208:~$ cd keys/
ubuntu@ip-172-31-31-208:~/keys$ ls
ubuntu@ip-172-31-31-208:~/keys$ |
```

AFTER THIS PLS COPY THE KEY VALUE PAIR FILE FOPR MASTER COMPUTER FROM THE LOCAL MACHINE TO THE PATH SPECIFIED BY PWD IN THE SERVER.

THE IMAGE BELOW GIVEN IS THE FILE IN MY LOCAL MACHINE

```
PS C:\Users\rosha> cd Downloads
PS C:\Users\rosha\Downloads> ls Ansible-key-project-demo.pem

Directory: C:\Users\rosha\Downloads

Mode                LastWriteTime         Length Name
----                -----          ----
-a----   19-10-2025      09:42           1678 Ansible-key-project-demo.pem
```

NOW IAM GOING TO TRANSFER HIS FILE FROM MY LOCAL MACHINE TO THE SERVER USING scp command just like the ssh command but it is used for copying files from local machine to the servers.

THIS IS THE COMAND USED TO INSTALL MY KEY PAIR VALUE FILE THAT IS .pem file to the server.

```
Install the latest PowerShell for new features and improvements: https://aka.ms/PSWindows

PS C:\Users\rosha> cd Downloads
PS C:\Users\rosha\Downloads> scp -i "Ansible-key-project-demo.pem" Ansible-key-project-demo.pem ubuntu@ec2-3-208-1-48.compute-1.amazonaws.com:/home/ubuntu/keys
The authenticity of host 'ec2-3-208-1-48.compute-1.amazonaws.com (3.208.1.48)' can't be established.
ED25519 key fingerprint is SHA256:Db2t/g+W7nycIPXYSH9lzXgJscov4RN39psEGIxeGms.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added 'ec2-3-208-1-48.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Ansible-key-project-demo.pem                                         100% 1678      6.4KB/s   00:00
PS C:\Users\rosha\Downloads> |
```

Command used is:-

```
scp -i "Ansible-key-project-demo.pem" Ansible-key-project-demo.pem ubuntu@ec2-3-208-1-48.compute-1.amazonaws.com:/home/ubuntu/keys
```

why keys because we have to move the private key to that folder to login to other servers.

I have installed the file from my local system to the server

```
Last login: Sun Oct 19 05:21:23 2025 from 103.28.246.18
ubuntu@ip-172-31-31-208:~$ ls
keys
ubuntu@ip-172-31-31-208:~$ cd keys
ubuntu@ip-172-31-31-208:~/keys$ ls
Ansible-key-project-demo.pem
ubuntu@ip-172-31-31-208:~/keys$ |
```

After this we opened our /etc/ansible/hosts that is the path for the hostfile. I opened the host file and added more variables using [servers:vars] to ensure that my private key file is common for all servers in the servers group.so here the advantage is we can add additional dependencies or installations.

```
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:
[servers]
server_1 ansible_host=54.91.194.123
server_2 ansible_host=18.209.46.0
server_3 ansible_host=44.223.26.66

[servers]:vars
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key-project-demo.pem
```

AS u can see in the belo figure by doing like this in host file we can manage the configurations and the variables that iam assigning can be given to a particular group which is helpful to manage multiple servers and cloud environments easily.CHANGES ARE MADE IN THE HOST FILE.

```
# Ex 2: A collection of hosts belonging to the 'webservers' group:  
[servers]  
server_1 ansible_host=54.91.194.123  
server_2 ansible_host=18.209.46.0  
server_3 ansible_host=44.223.26.66  
  
[servers:vars]  
ansible_python_interpreter=/usr/bin/python3  
ansible_user=ubuntu  
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key-project-demo.pem
```

AFTER THIS CHANGE THE PERMISSONS OR OWNERSHIP using the “chmod 400 <path of key pair file.pem file>”

```
ubuntu@ip-172-31-31-208:~/keys$ chmod 400 /home/ubuntu/keys/Ansible-key-project-demo.pem  
ubuntu@ip-172-31-31-208:~/keys$ ls -l /home/ubuntu/keys/Ansible-key-project-demo.pem  
-r----- 1 ubuntu ubuntu 1678 Oct 19 05:33 /home/ubuntu/keys/Ansible-key-project-demo.pem  
ubuntu@ip-172-31-31-208:~/keys$ |
```

NOW AFTER THIS THE PERMISSIONS ARE CHANGED.

THE PREVIOUS STEP WAS VERY IMPORTANT BECAUSE WITHOUT THAT WE CANNOT REACH THE SERVERS AS SSH WILL BE TOO OPEN AND SHOWS ERRORS.

WE ARE TRYING TO PING TO SEE THE CONNECTIVITY BETWEEN THE SERVERS AND IT IS ALL SUCCESSFUL.

```
ubuntu@ip-172-31-31-208:~/keys$ ansible servers -m ping  
server_1 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
server_3 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
server_2 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}  
ubuntu@ip-172-31-31-208:~/keys$ |
```

WE CAN ALSO GET THE CPU USAGE OF ALL SERVERS by using -a that is adhoc command which is “free -h” for the group servers.

```
ubuntu@ip-172-31-31-208:~/keys$ ansible servers -a "free -h"
server_3 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    914Mi       365Mi     256Mi       2.7Mi     450Mi     548Mi
Swap:      0B         0B        0B
server_1 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    914Mi       345Mi     278Mi       2.7Mi     449Mi     568Mi
Swap:      0B         0B        0B
server_2 | CHANGED | rc=0 >>
      total        used        free      shared  buff/cache   available
Mem:    914Mi       363Mi     259Mi       2.7Mi     449Mi     550Mi
Swap:      0B         0B        0B
```

BY ALSO DUSING THE ADHOC COMMAND ansible servers -a “sudo apt update” we are able to update all servers from a single computer that is the master computer within seconds.

```
ubuntu@ip-172-31-31-208:~/keys$ ansible servers -a "sudo apt update"
server_1 | CHANGED | rc=0 >>
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1498 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [288 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1496 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1222 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [301 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [378 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.3 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2090 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [470 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [516 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.3 kB]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5564 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
```

NOWAFTER THIS IAM CHANGING MY HOST FILE TO HAVE TWO GROUPS ONE FOR DEV AND OTHER FOR PROD DOMAINS AND I HAVE CHANGED A SINGLE GROUP THAT IS [servers:vars] to [all:vars] which means applicable for every group .

```
# Ex 2: A collection of hosts belonging to the 'webservers' group:  
[servers]  
server_1 ansible_host=54.91.194.123  
server_2 ansible_host=18.209.46.0  
  
[prod]  
server_3 ansible_host=44.223.26.66  
  
[all:vars]  
ansible_python_interpreter=/usr/bin/python3  
ansible_user=ubuntu  
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key-project-demo.pem
```

when we are inside thisfile using vim and trying to modify we get this error like this when we are trying to save the file using :wq

```
[servers]  
server_1 ansible_host=54.91.194.123  
server_2 ansible_host=18.209.46.0  
server_3 ansible_host=44.223.26.66  
  
[servers:vars]  
ansible_python_interpreter=/usr/bin/python3  
ansible_user=ubuntu  
ansible_ssh_private_key_file=/home/ubuntu/keys/Ansible-key-project-demo.pem  
  
## [webservers]  
E45: 'readonly' option is set (add ! to override)
```

To avoide this use command :qa! To get out of the file and use “sudo vim /etc/ansible/hosts” this command. This is what we have to do to give permissions for read and write.

```
ubuntu@ip-172-31-31-208:~/keys$ vim /etc/ansible/hosts  
ubuntu@ip-172-31-31-208:~/keys$ sudo vim /etc/ansible/hosts
```

AS WE CAN SEE BELOW THE SERVER 3 IS GETTING CONNECTED IN PROD ENVIRONMENT OR GROUP.

```
ubuntu@ip-172-31-31-208:~/keys$ ansible prod -m ping  
server_3 | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

To check if ever yserver has the ansible master key from terminal we can use the command “ansible-inventory --list” command can be used.

```
ubuntu@ip-172-31-31-208:~/keys$ ansible-inventory --list
{
    "_meta": {
        "hostvars": {
            "server_1": {
                "ansible_host": "54.91.194.123",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key-project-demo.pem",
                "ansible_user": "ubuntu"
            },
            "server_2": {
                "ansible_host": "18.209.46.0",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key-project-demo.pem",
                "ansible_user": "ubuntu"
            },
            "server_3": {
                "ansible_host": "44.223.26.66",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/Ansible-key-project-demo.pem",
                "ansible_user": "ubuntu"
            }
        }
    }
}
```

WE CAN SEE THAT ALL THE SERVERS HAVE THAT IS ERVER 3 WHICH IS IN THE PROD ENVIRONMENT AND SERVER 1 AND SERVER 2 IN SERVERS ENVIRONMENT HAVE THE SAME .pem file that is key – value pair file so the servers we are able to ping in different environments in canse if I change the [all:vars] to [servers:vars] then the server 3 will not be reachable THAT IS IF WE MAKE A CHANGE IN THE HOST FILE.

.NOW CREATING ANSIBLE PLAYBOOKS:-

PLAYBOOKS ARE YML FILES USED FOR AUTOMATION OF SCRIPTS TO UPDATE MULTIPLE SERVERS CAN BE USED FOR PATCHING AND FOR INSTALLING MANY DEPENDENCIES.SO MANY TASK WILL BE GIVEN IN ORDER TO DO THE ACTIONS REQUIRED.

```
ubuntu@ip-172-31-31-208:~$ ls
keys
ubuntu@ip-172-31-31-208:~$ mkdir playbooks
ubuntu@ip-172-31-31-208:~$ ls
keys  playbooks
ubuntu@ip-172-31-31-208:~$ cd playbooks
ubuntu@ip-172-31-31-208:~/playbooks$ |
```

Creating playbooks in the instance by creating a directory in the master computer

NOW WE ARE CREATING A SIMPLE PLAYBOOK FOR GETTING THE SYSTEM DATE AND TIME USING FILE “date\_play.yml”.

```
ubuntu@ip-172-31-31-208:~$ ls
keys
ubuntu@ip-172-31-31-208:~$ mkdir playbooks
ubuntu@ip-172-31-31-208:~$ ls
keys  playbooks
ubuntu@ip-172-31-31-208:~$ cd playbooks
ubuntu@ip-172-31-31-208:~/playbooks$ vim date_play.yml|
```

This is how my yml format for playbook looks like and in this way we will be able to get the date for servers group 1 and 2 but not for server 3 as it is in the prod environment

```
- 
  name: Dates Playbook
  hosts: servers
  tasks:
    |
      - name: Show date
        command: date

      - name: Show date again
        command: date
    ~
    ~
    ~
    ~
    ~
```

Command used for running playbook given below:-

```
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook date_play.yml
```

OUTPUT IS LIKE THIS:-

```
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook date_play.yml
PLAY [Dates Playbook] ****
TASK [Gathering Facts] ****
ok: [server_1]
ok: [server_2]

TASK [Show date] ****
changed: [server_1]
changed: [server_2]

TASK [Show date again] ****
changed: [server_1]
changed: [server_2]

PLAY RECAP ****
server_1          : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2          : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ubuntu@ip-172-31-31-208:~/playbooks$
```

WE CAN SEE THAT AFTER RUNNING IT IS NOT SHOWING THE DATE BECAUSE WE HAVE TO ADD A VERBOSE TO IT SO THAT WE CAN SEE THE DATE FROM THE TERMINAL.

```
ubuntu@ip-172-31-31-208:~/playbooks$ ls
date_play.yml
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook -v date_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Dates Playbook] ****
TASK [Gathering Facts] ****
ok: [server_1]
ok: [server_2]

TASK [Show date] ****
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004148", "end": "2025-10-19 07:10:54.666140", "msg": "", "rc": 0, "start": "2025-10-19 07:10:54.661992", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:10:54 UTC 2025", "stdout_lines": ["Sun Oct 19 07:10:54 UTC 2025"]}
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:01.005220", "end": "2025-10-19 07:10:55.671218", "msg": "", "rc": 0, "start": "2025-10-19 07:10:54.665998", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:10:55 UTC 2025", "stdout_lines": ["Sun Oct 19 07:10:55 UTC 2025"]}

TASK [Show date again] ****
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.003677", "end": "2025-10-19 07:10:55.996553", "msg": "", "rc": 0, "start": "2025-10-19 07:10:55.992876", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:10:55 UTC 2025", "stdout_lines": ["Sun Oct 19 07:10:55 UTC 2025"]}
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004340", "end": "2025-10-19 07:10:56.013934", "msg": "", "rc": 0, "start": "2025-10-19 07:10:56.009594", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:10:56 UTC 2025", "stdout_lines": ["Sun Oct 19 07:10:56 UTC 2025"]}
```

SO NOW IN THE ABOVE FIGURE U CAN CLEARLY SEE THAT THE DATE IS PRINTED .

NOW IAM CHANGING MY PLAYBOOK TO SHOW THE DATE AND UPTIME BY CHANGING THE yml file that is date\_play.yml

```
name: Dates Playbook
hosts: servers
tasks:
  - name: Show date
    command: date
  - name: Show UPTIME
    command: uptime|
```

THIS OUTPUT SHOWS US THE DATE AND UPTIME

```
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook -v date_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Dates Playbook] ****
TASK [Gathering Facts] ****
ok: [server_1]
ok: [server_2]

TASK [Show date] ****
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004144", "end": "2025-10-19 07:18:11.597988", "msg": "", "rc": 0, "start": "2025-10-19 07:18:11.593844", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:18:11 UTC 2025", "stdout_lines": ["Sun Oct 19 07:18:11 UTC 2025"]}
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004051", "end": "2025-10-19 07:18:11.596963", "msg": "", "rc": 0, "start": "2025-10-19 07:18:11.592912", "stderr": "", "stderr_lines": [], "stdout": "Sun Oct 19 07:18:11 UTC 2025", "stdout_lines": ["Sun Oct 19 07:18:11 UTC 2025"]}

TASK [Show UPTIME] ****
changed: [server_1] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.007515", "end": "2025-10-19 07:18:11.917956", "msg": "", "rc": 0, "start": "2025-10-19 07:18:11.910440", "stderr": "", "stderr_lines": [], "stdout": " 07:18:11 up 2:52, 1 user, load average: 0.00, 0.00, 0.00", "stdout_lines": [" 07:18:11 up 2:52, 1 user, load average: 0.00, 0.00, 0.00"]}
changed: [server_2] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.008852", "end": "2025-10-19 07:18:11.934942", "msg": "", "rc": 0, "start": "2025-10-19 07:18:11.926090", "stderr": "", "stderr_lines": [], "stdout": " 07:18:11 up 2:52, 1 user, load average: 0.00, 0.00, 0.00", "stdout_lines": [" 07:18:11 up 2:52, 1 user, load average: 0.00, 0.00, 0.00"]}

PLAY RECAP ****
server_1 : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
\
```

NEXT IS WE ARE TRYING TO INSTALL NGINX IN OUR PLAYBOOK:-

```
PLAY RECAP ****
server_1 : ok=3    changed=2    unreachable=0    failed=0
server_2 : ok=3    changed=2    unreachable=0    failed=0

ubuntu@ip-172-31-31-208:~/playbooks$ vim install_nginx_play.yml
```

IN install\_nginx\_play.yml file looks like this:-

```
- name: Install Nginx and start it
  hosts: servers
  become: yes
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: latest
        |
```

HERE “becomes:” makes us aas an admin or the root user to gain access for installations and dependencies “apt:” is a module used for running in ubuntu system.”state:” indicates the version of nginx we want that is the latest version in this case.

```
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook install_nginx_play.yml

PLAY [Install Nginx and start it] ****
TASK [Gathering Facts] ****
ok: [server_1]
ok: [server_2]

TASK [Install Nginx] ****
changed: [server_1]
changed: [server_2]

PLAY RECAP ****
server_1 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

BY DOING THIS NGINX IS INSTALLED FOR THE TWO SERVERS THAT IS SERVER 1 AND SERVER 2 IN SERVERS GROUP.WE HAVE JUST INSTALLED BUT NOT STARTED SO GO TO PLAYBOOK AND START THE NGINX.

```
- 
  name: Install Nginx and start it
  hosts: servers
  become: yes
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: latest
    - name: Start nginx
      service:
        name: nginx
        state: started
        enabled: yes
    |
```

HERE U CAN SEE THAT START IS A SERVICE SO we have written as “service:” and then specified the name , state and we want it enable dall the time so enabled: yes we have given.

SO NOW ALL SERVERS HAVE NGINX INSTALLED AND NGINX IS RUNNING IN ALL SERVERS. TO KNOW OR VERIFY COPY THE PUBLIC IPV4 ADREES OF SERVER 1 or 2 and paste it in the we browser we will ge tit but not for server 3 because it is in a different environment or group known as “prod”.

```
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook install_nginx_play.yml
PLAY [Install Nginx and start it] ****
TASK [Gathering Facts] ****
ok: [server_1]
ok: [server_2]

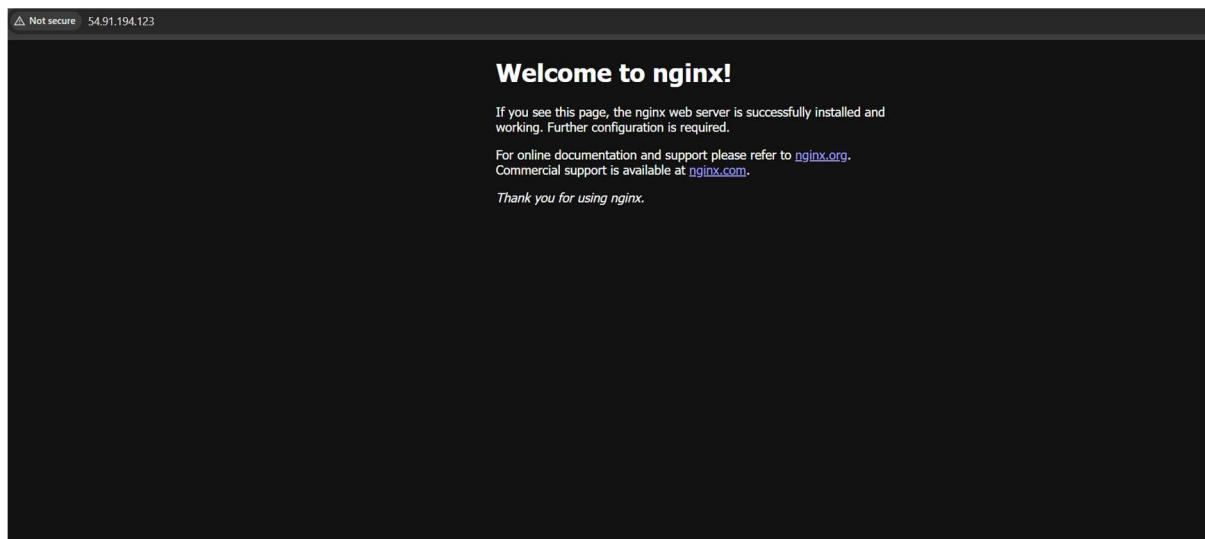
TASK [Install Nginx] ****
ok: [server_1]
ok: [server_2]

TASK [Start nginx] ****
ok: [server_2]
ok: [server_1]

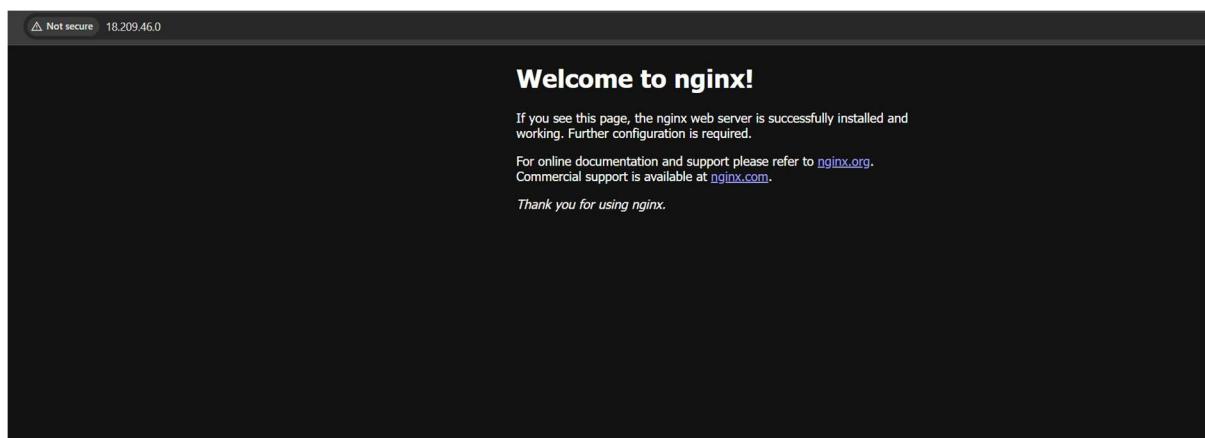
PLAY RECAP ****
server_1          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

OUTPUT:-

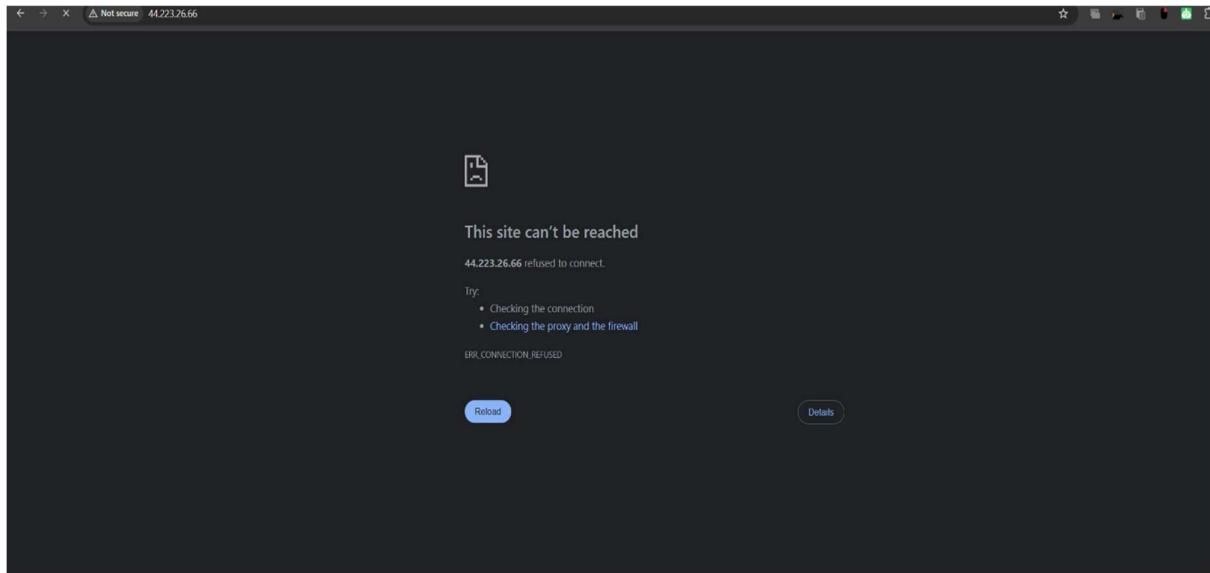
SERVER 1:-



SERVER 2:-



SERVER 3 DIFFERENT ENVIRONMENT SO WILL NOT RUN NGINX AND WILL NOT PRINT ANYTHING IN THE WEB BROWSER.



NOW FOR DEPLOYING A STATIC WEBSITE CREATE A NEW YML FILE AND WRITE THE PIECE OF CODE. SRC IS THE INDEX.HTML FILE DESTINATION IS WHERE ANY CHANGES TO THE HTML FILE MADE IS REFLECTED.SO we are copying the source to shte destination.

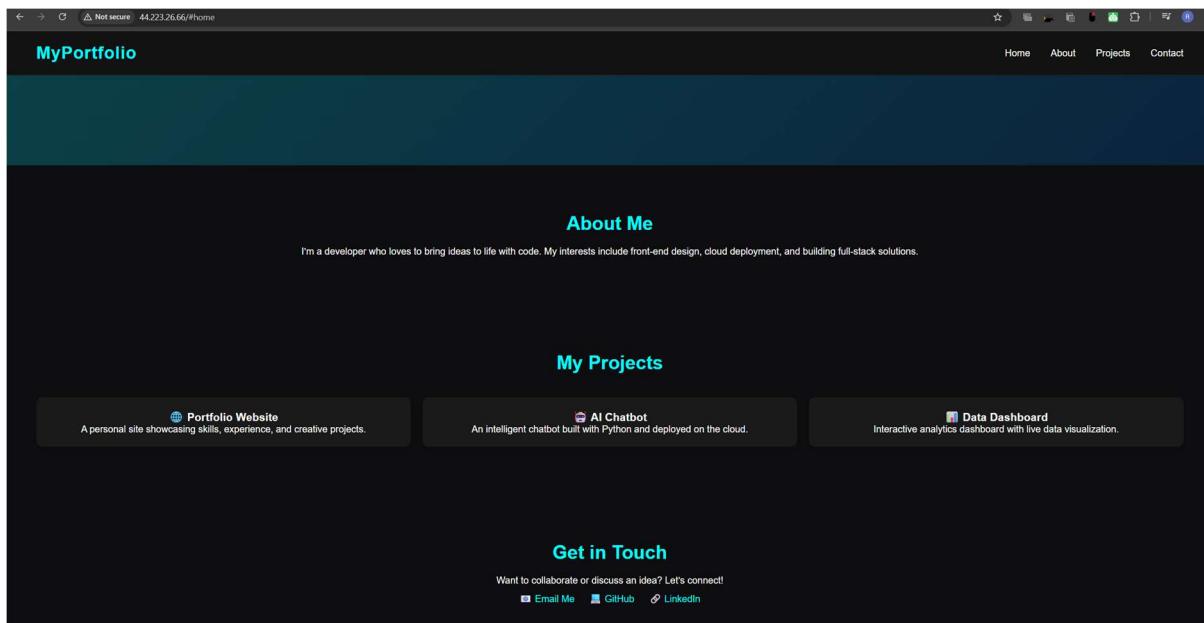
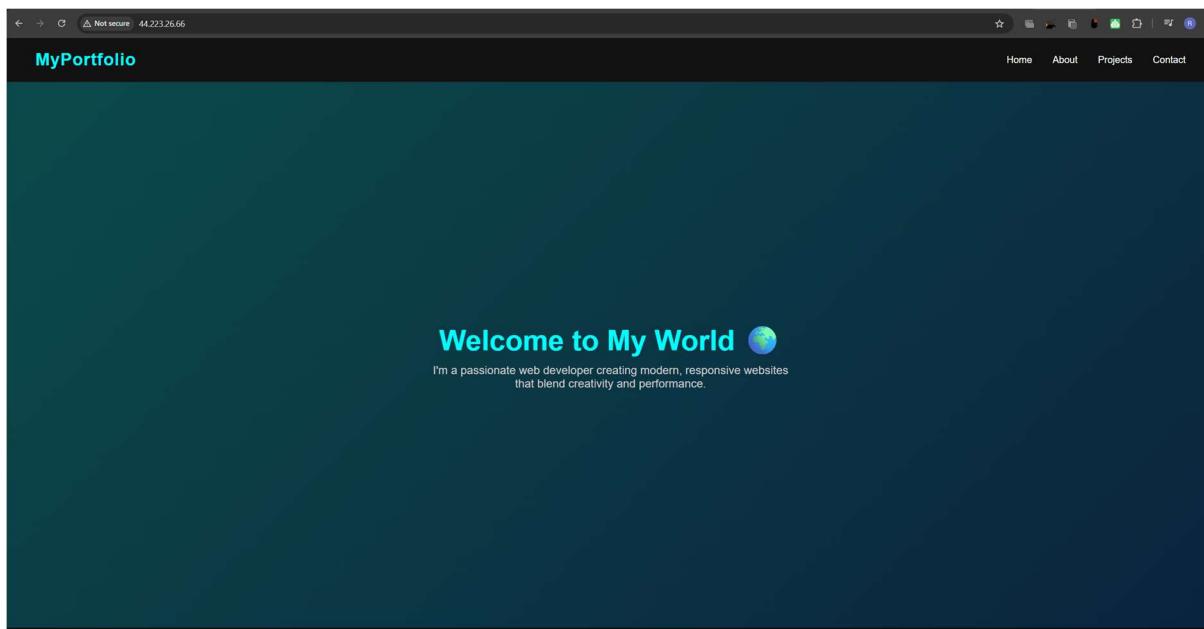
```
-  
  name: Install nginx and serve static website  
  hosts: prod  
  become: yes  
  tasks:  
    - name: Install nginx  
      apt:  
        name: nginx  
        state: latest  
  
    - name: Start nginx  
      service:  
        name: nginx  
        state: started  
        enabled: yes  
  
    - name: Deploy webpage  
      copy:  
        src: index.html  
        dest: /var/www/html|
```

CREATE AN INDEX.HTML PAGE SEPERATELY AND RUN THE playbook command to deploy the static website her iam deploying my portfolio which is a static website.

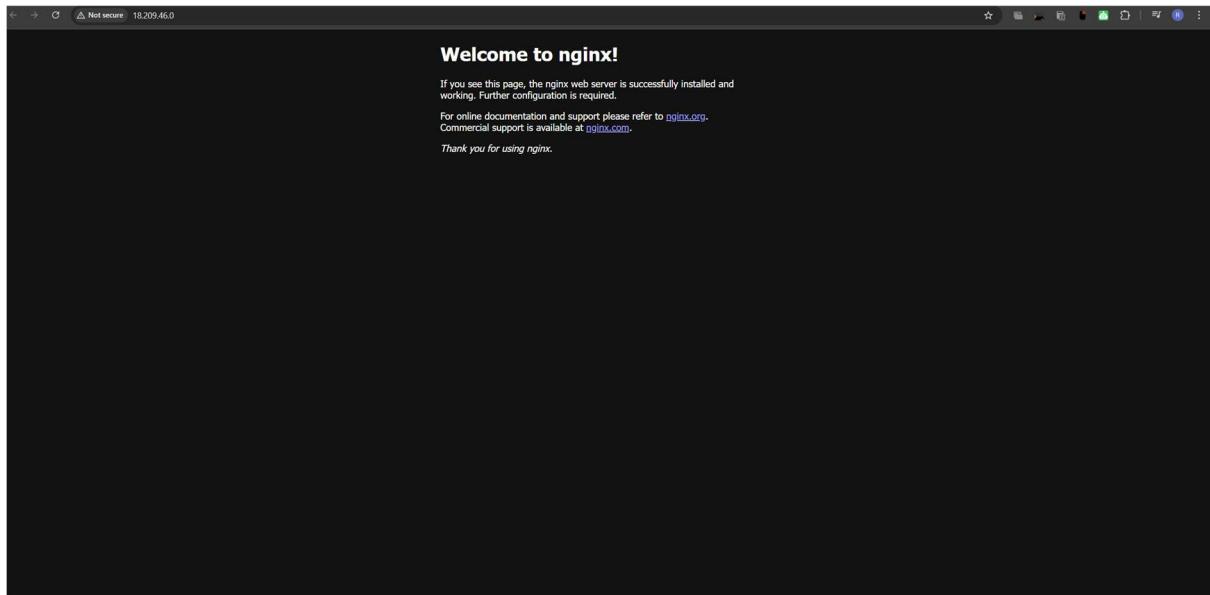
```
ubuntu@ip-172-31-31-208:~/playbooks$ ls  
date_play.yml  deploy_static_page_play.yml  index.html  install_nginx_play.yml  
ubuntu@ip-172-31-31-208:~/playbooks$ ansible-playbook  deploy_static_page_play.yml
```

DEPLOYING MY STATIC WEBSITE IN SERVER 3 WHICH IS IN THE PROD GROUP  
WEBSITE CANNOT BE SEEN IN SERVER 1 AND SERVER 2

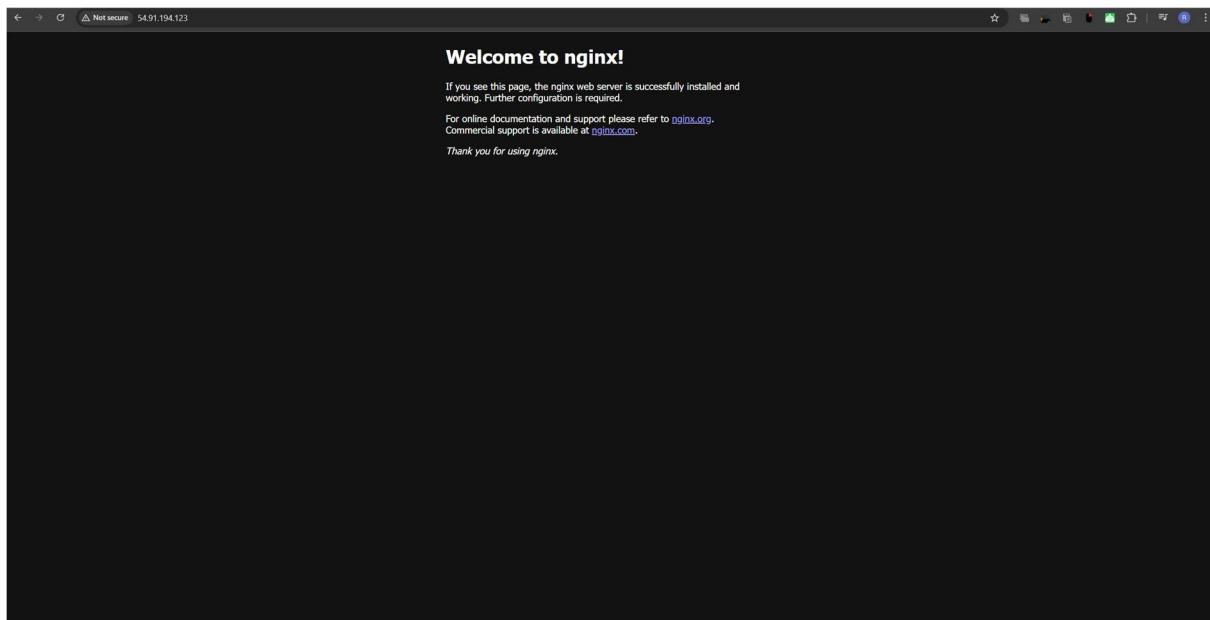
SERVER 3 OUTPUT:-



## SERVER 2 OUTPUT:-



## SERVER 1 OUTPUT:-



CONCLUSION: ANSIBLE IS VERY USEFUL IF WE WANT TO HANDLE MULTIPLE ENVIRONMENTS OR GROUPS AND HANDLE DEPENDENCIES AND INSTLLATIONS FOR AROUND 1BIILON SERVERS APPROX SO AS A SYSTEM ADMIN OR A DEVOPS ENGINEER MY TASK BECOMES EASIER