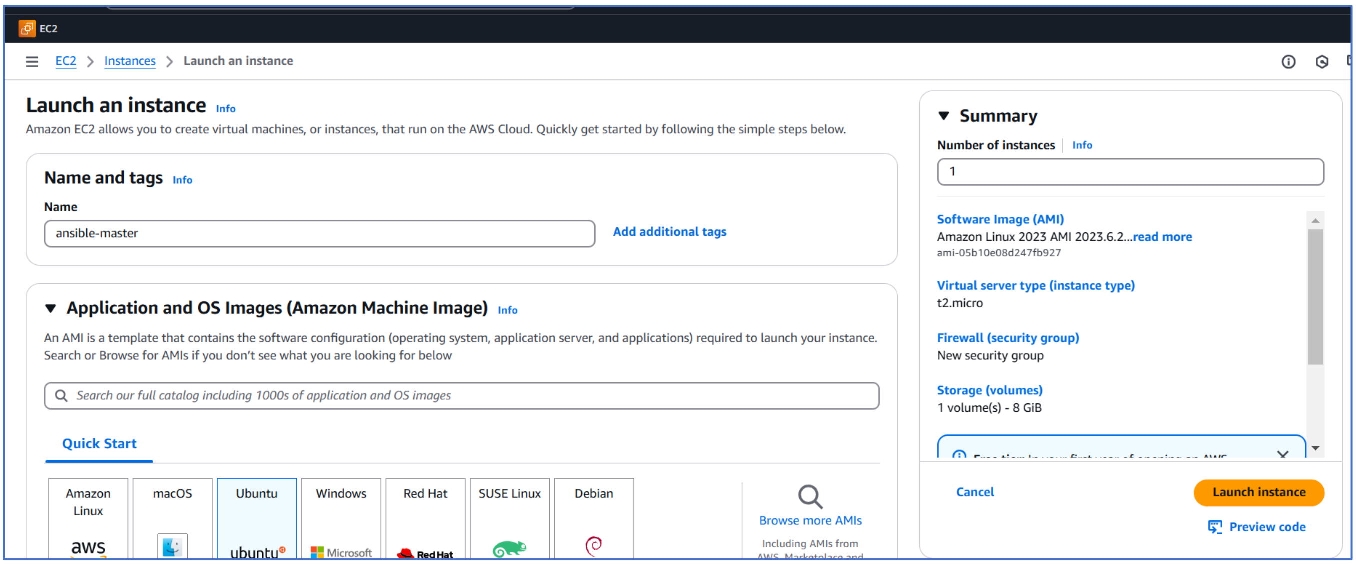
Ansible-Driven System Initialization and Environment Configuration

This project aims to automate the foundational setup of Ubuntu systems using Ansible, culminating in the deployment of a simple HTML webpage. We utilize Ansible to streamline essential tasks, from system updates and package installations to the installation of necessary web server components. This playbook incorporates conditional logic to achieve idempotency, minimizing unnecessary changes and guaranteeing a reliable, repeatable process. This documentation provides a step-by-step guide to:

- Automating system updates and essential package installations using Ansible.
- Installing and configuring a basic web server (e.g., Nginx or Apache).
- Deploying a static HTML page to the web server's document root.
- Implementing conditional logic for idempotent playbook execution.
- Ensuring consistent environment setup across multiple servers.

Launch an AWS EC2 instances

- **Enter the Name of the Instance**, eg: ansible-master
- Choose **Ubuntu Server 24.04 LTS (HVM)** under **Amazon Machine Image(AMI)**



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: ansible-master

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and

Summary

Number of instances Info: 1

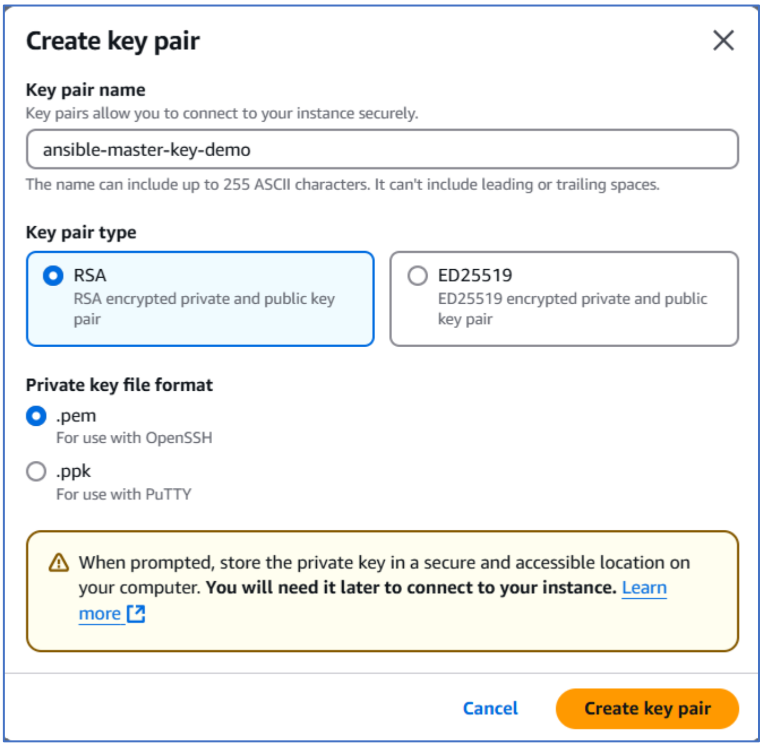
Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more ami-05b10e08d247fb927

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

- Choose **t2.micro** under **Instance type**.
- Under **Key pair (login)**, give your key pair name or create a new key pair eg: ansible-master-key-demo is my keypair.



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

ansible-master-key-demo

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

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

Private key file format

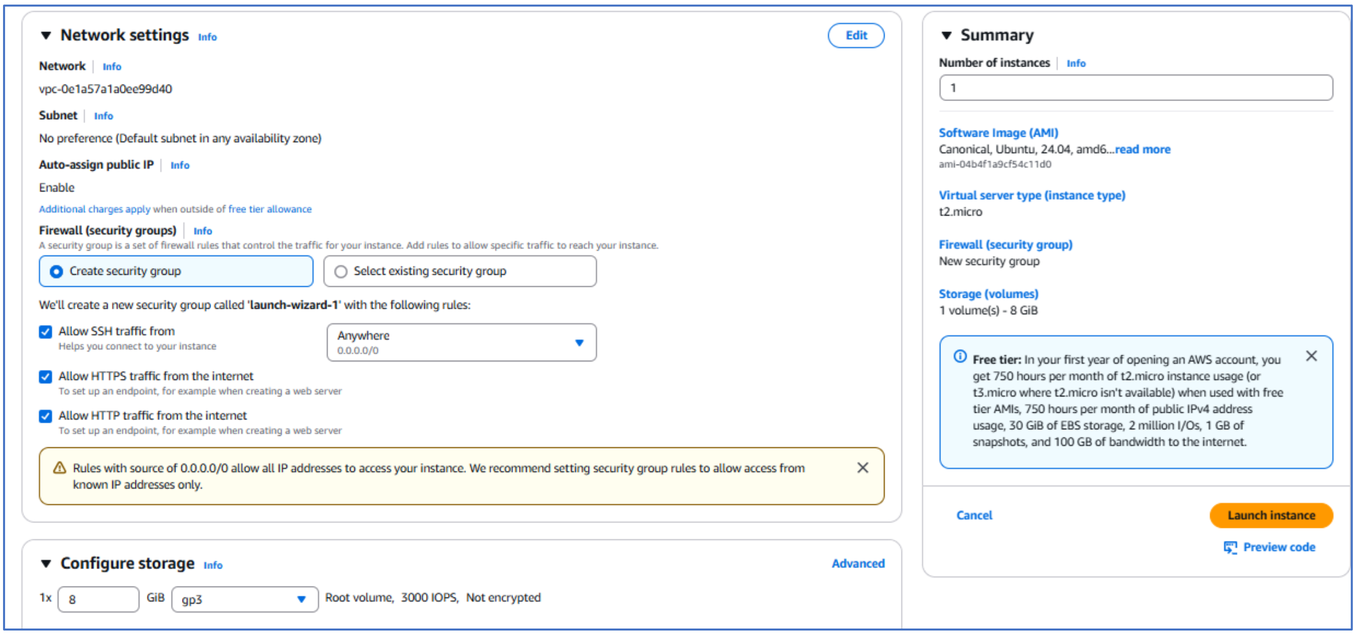
● .pem
For use with OpenSSH

○ .ppk
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

- In the network settings, choose the default VPC and subnet, and enable the option to auto-assign a public IP. For the firewall security groups, create a new security group and configure the following rules: allow **SSH** traffic from anywhere (port 22) to enable secure remote access, allow **HTTP** traffic from the internet (port 80) to support standard web traffic, and allow **HTTPS** traffic from the internet (port 443) for secure, encrypted web connections.

- Configure the storage settings by setting the root volume size to **8 GiB** to ensure sufficient space for the application and related dependencies.
- After finalizing the storage configuration and reviewing all settings, proceed to **launch the EC2 instance**.



Network settings

Network

vpc-0e1a57a1a0ee99d40

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0
Helps you connect to your instance

☑ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...read more

ami-04b4f1a9cf54c11d0

Virtual server type (instance type)

t2.micro

Firewall (security group)

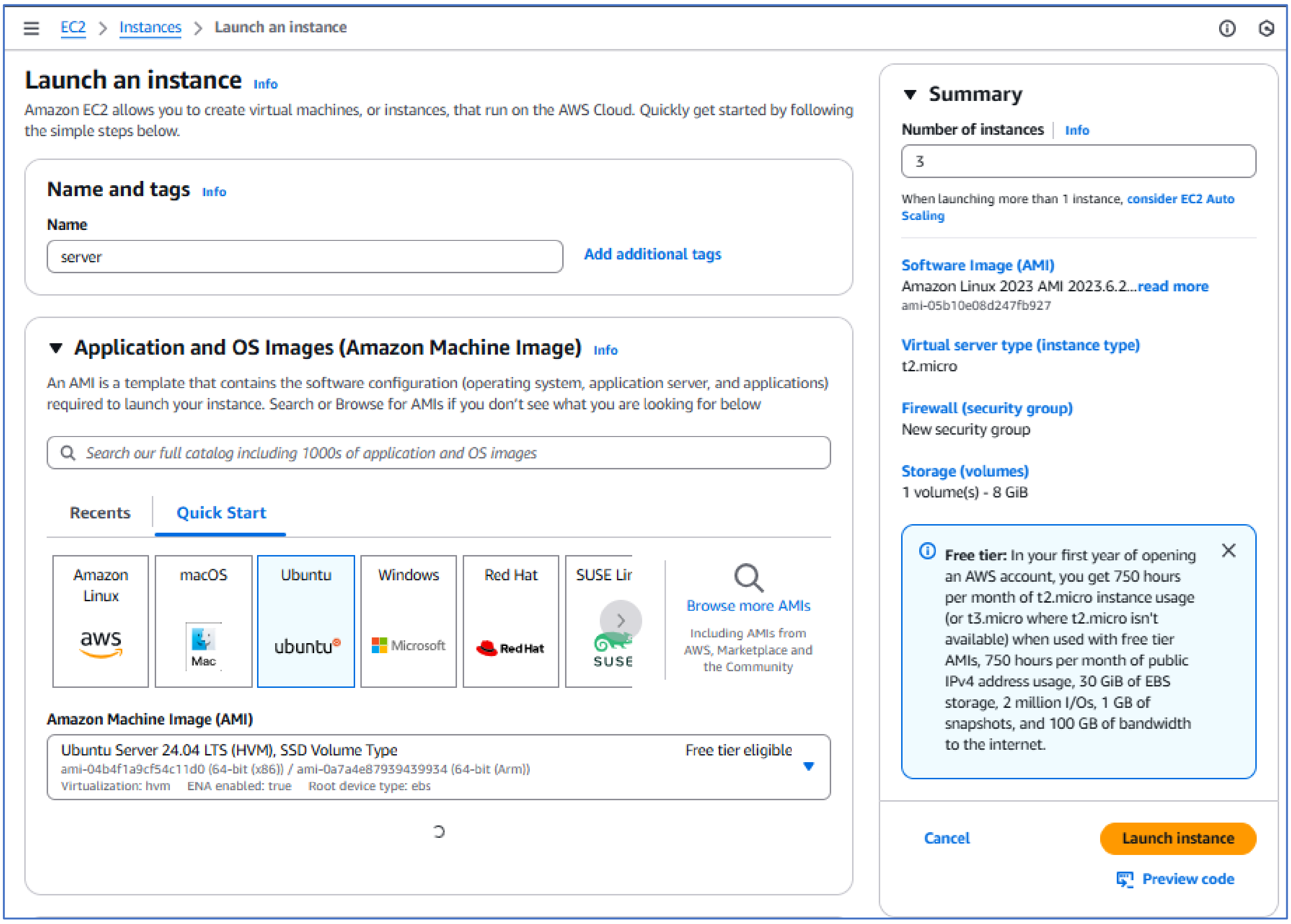
New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

- To demonstrate Ansible's capabilities in managing multiple servers, we will launch three additional EC2 instances. These instances will act as target servers, receiving configuration and installation instructions from the Ansible master instance. This allows us to automate the setup across a distributed environment.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: server

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Lir
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-04b4f1a9cf54c11d0 (64-bit (x86)) / ami-0a7a4e87939439934 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances Info: 3

When launching more than 1 instance, consider EC2 Auto Scaling

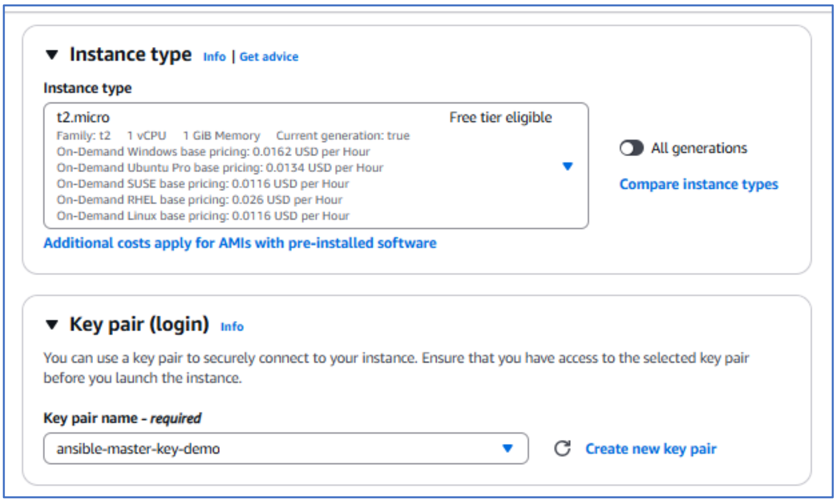
Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-05b10e08d247fb927

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.



Instance type Info | Get advice

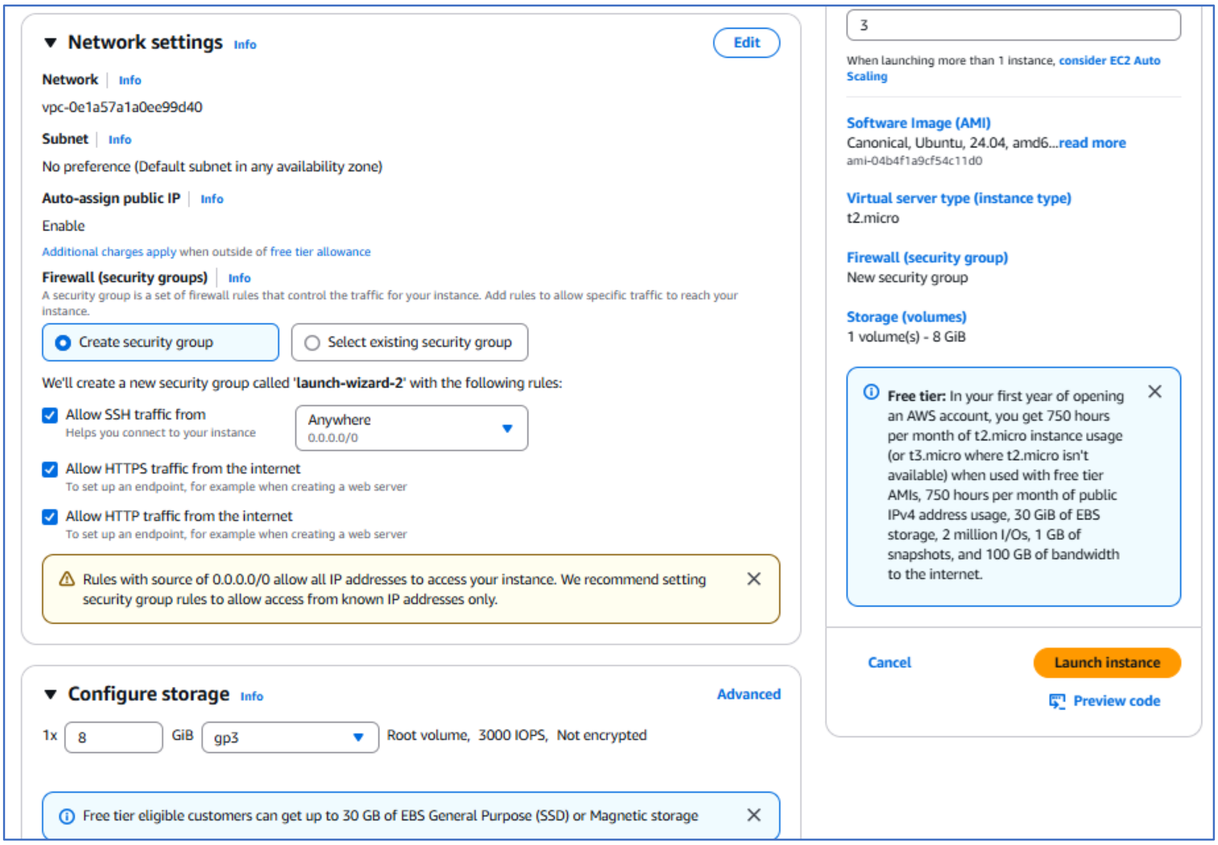
Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: ansible-master-key-demo



▼ Network settings Info

Edit

Network | Info

vpc-0e1a57a1a0ee99d40

Subnet | Info

No preference (Default subnet in any availability zone)

Auto-assign public IP | Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-2**' with the following rules:

☑ Allow SSH traffic from — Helps you connect to your instance — Anywhere 0.0.0.0/0

☑ Allow HTTPS traffic from the internet — To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet — To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

▼ Configure storage Info

Advanced

1x 8 GiB gp3 Root volume, 3000 IOPS, Not encrypted

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ✕

3

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...read more

ami-04b4f1a9cf54c11d0

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

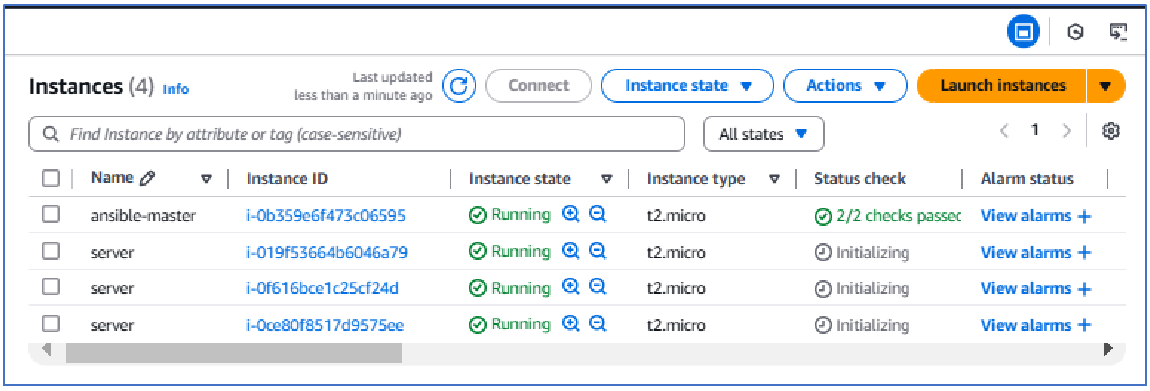
Storage (volumes)

1 volume(s) - 8 GiB

ⓘ **Free tier:** In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. ✕

Cancel Launch instance

Preview code



Instances (4) Info

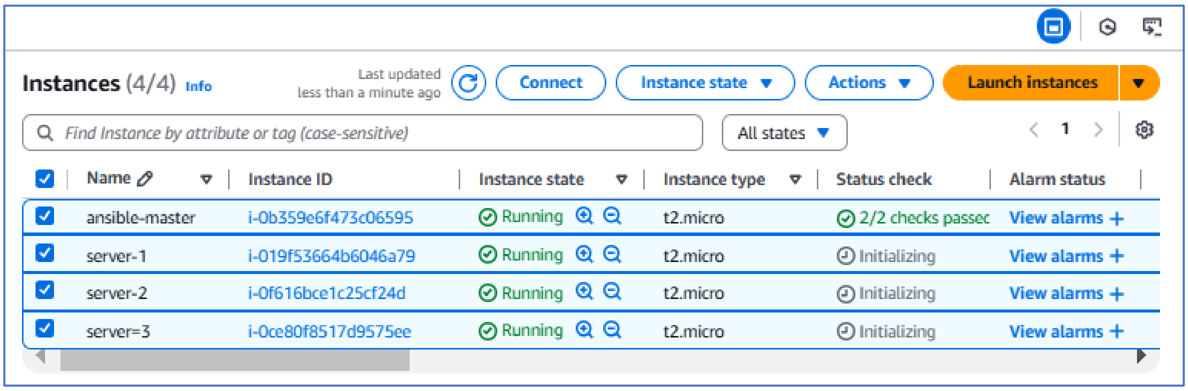
Last updated less than a minute ago

Connect Instance state ▼ Actions ▼ Launch instances ▼

Find Instance by attribute or tag (case-sensitive) All states ▼ 1

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	ansible-master	i-0b359e6f473c06595	Running	t2.micro	2/2 checks passec	View alarms +
□	server	i-019f53664b6046a79	Running	t2.micro	Initializing	View alarms +
□	server	i-0f616bce1c25cf24d	Running	t2.micro	Initializing	View alarms +
□	server	i-0ce80f8517d9575ee	Running	t2.micro	Initializing	View alarms +

- To ensure clarity and better organization, we manually renamed the three instances from **'server'** to **'server-1', 'server-2', and 'server-3'**, respectively.



Instances (4/4) Info

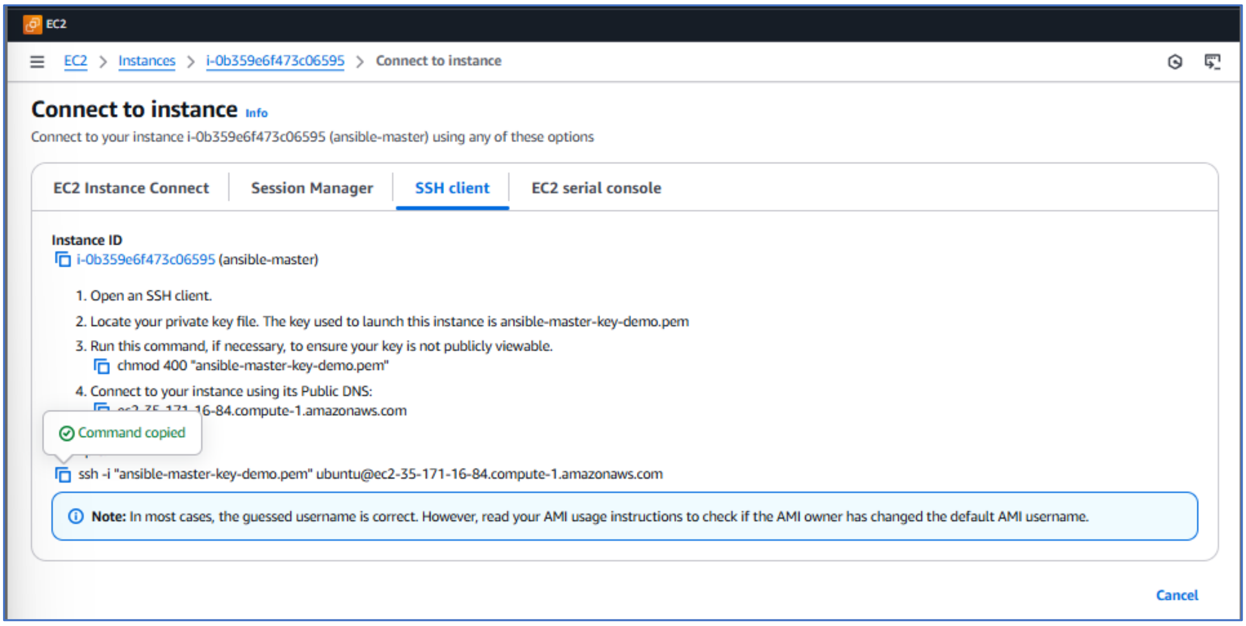
Last updated less than a minute ago

Connect Instance state ▼ Actions ▼ Launch instances ▼

Find Instance by attribute or tag (case-sensitive) All states ▼ 1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	ansible-master	i-0b359e6f473c06595	Running	t2.micro	2/2 checks passec	View alarms +
☑	server-1	i-019f53664b6046a79	Running	t2.micro	Initializing	View alarms +
☑	server-2	i-0f616bce1c25cf24d	Running	t2.micro	Initializing	View alarms +
☑	server=3	i-0ce80f8517d9575ee	Running	t2.micro	Initializing	View alarms +

- Next, we will establish an SSH connection to the `ansible-master` instance to proceed with the configuration



Connect to instance

Connect to your instance i-0b359e6f473c06595 (ansible-master) using any of these options

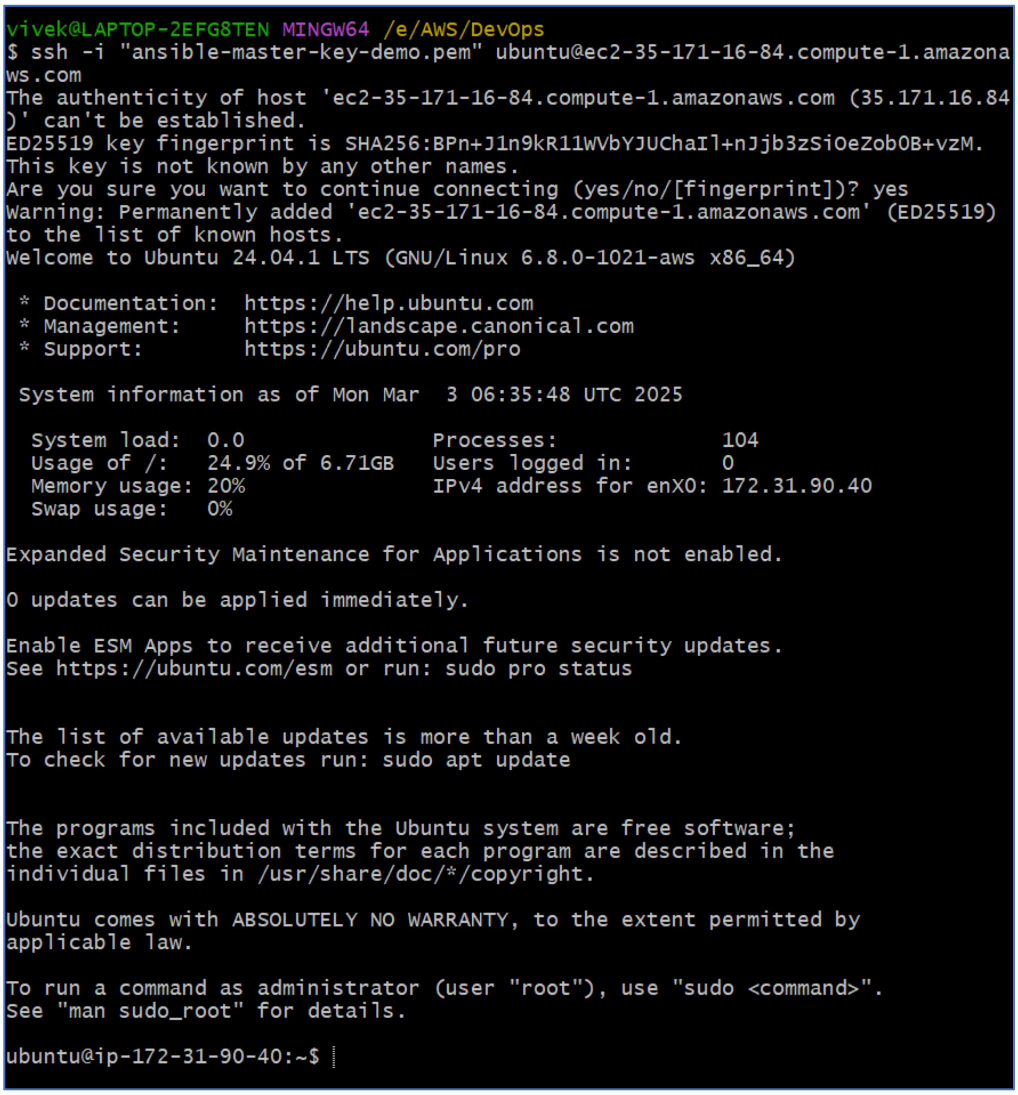
Instance ID

i-0b359e6f473c06595 (ansible-master)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ansible-master-key-demo.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "ansible-master-key-demo.pem"
4. Connect to your instance using its Public DNS:
   ec2-35-171-16-84.compute-1.amazonaws.com

ssh -i "ansible-master-key-demo.pem" ubuntu@ec2-35-171-16-84.compute-1.amazonaws.com

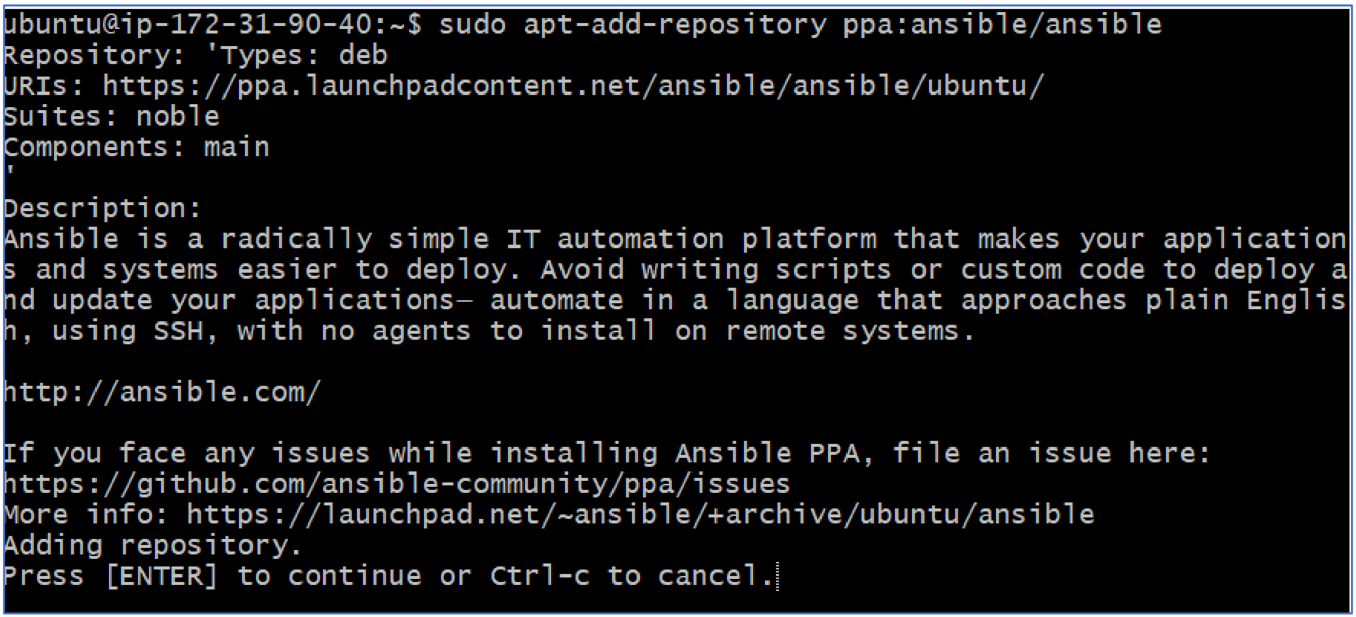
ⓘ **Note:** In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.



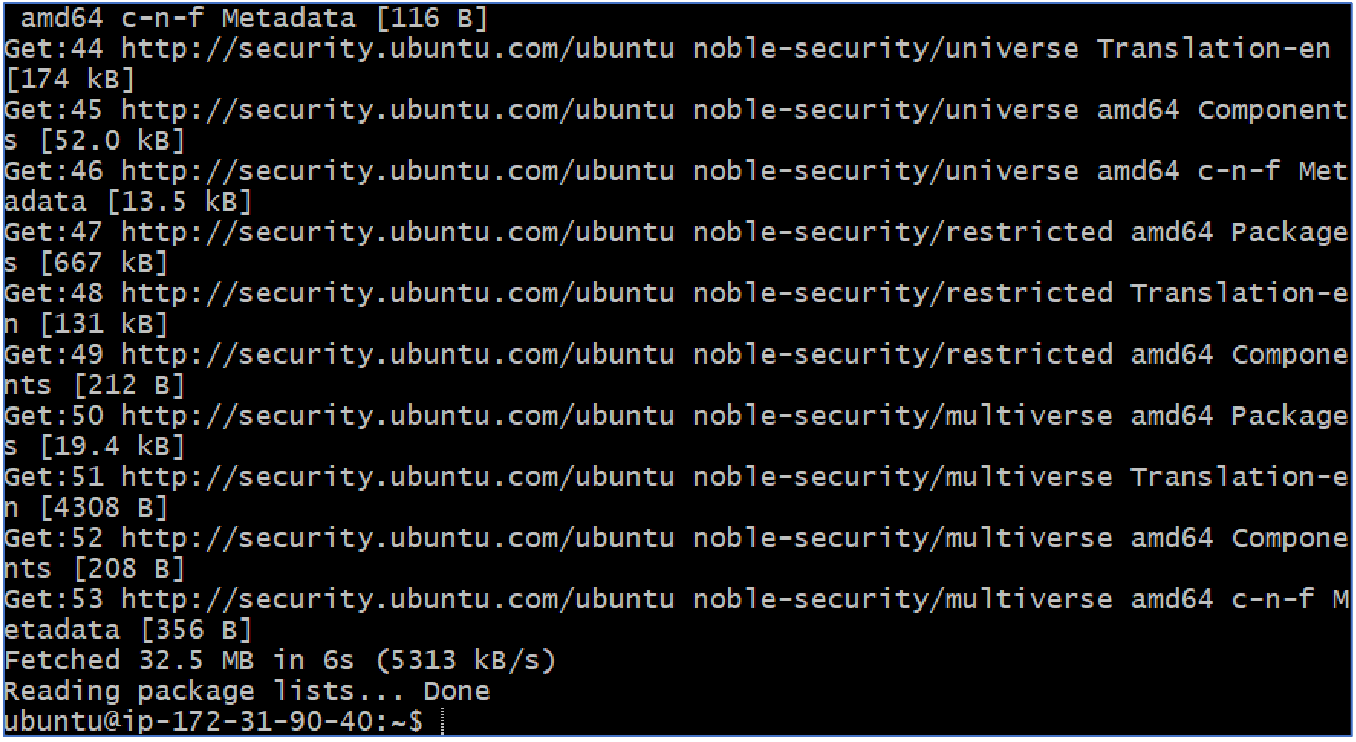
```
vivek@LAPTOP-2EFG8TEN MINGW64 /e/AWS/DevOps
$ ssh -i "ansible-master-key-demo.pem" ubuntu@ec2-35-171-16-84.compute-1.amazona
ws.com
The authenticity of host 'ec2-35-171-16-84.compute-1.amazonaws.com (35.171.16.84
)' can't be established.
ED25519 key fingerprint is SHA256:BPn+J1n9kR11WVbYJUChaIl+nJjb3zSiOeZobOB+vzM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-35-171-16-84.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)

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

 System information as of Mon Mar  3 06:35:48 UTC 2025

  System load:  0.0               Processes:             104
  Usage of /:   24.9% of 6.71GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for enX0: 172.31.90.40
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-172-31-90-40:~$
```

Set Up Ansible Master

- To install Ansible on the `ansible-master` instance, we need to add the Ansible PPA (Personal Package Archive) to the system's software repository list. This PPA provides the latest version of Ansible.
  We can achieve this by executing the following command:
  `sudo apt-add-repository ppa:ansible/ansible`
  This will ensure that we have access to the most up-to-date Ansible packages during the installation process.



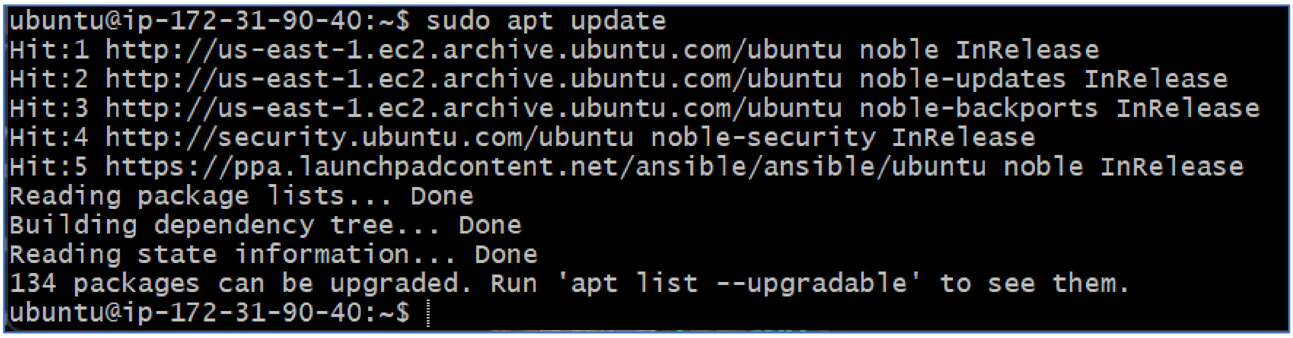
```
ubuntu@ip-172-31-90-40:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'Types: deb
URIs: https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/
Suites: noble
Components: main
'
Description:
Ansible is a radically simple IT automation platform that makes your application
s and systems easier to deploy. Avoid writing scripts or custom code to deploy a
nd update your applications— automate in a language that approaches plain Englis
h, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
```



```
 amd64 c-n-f Metadata [116 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en
[174 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Component
s [52.0 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Met
adata [13.5 kB]
Get:47 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Package
s [667 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-e
n [131 kB]
Get:49 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Compone
nts [212 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Package
s [19.4 kB]
Get:51 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-e
n [4308 B]
Get:52 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Compone
nts [208 B]
Get:53 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f M
etadata [356 B]
Fetched 32.5 MB in 6s (5313 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-90-40:~$
```

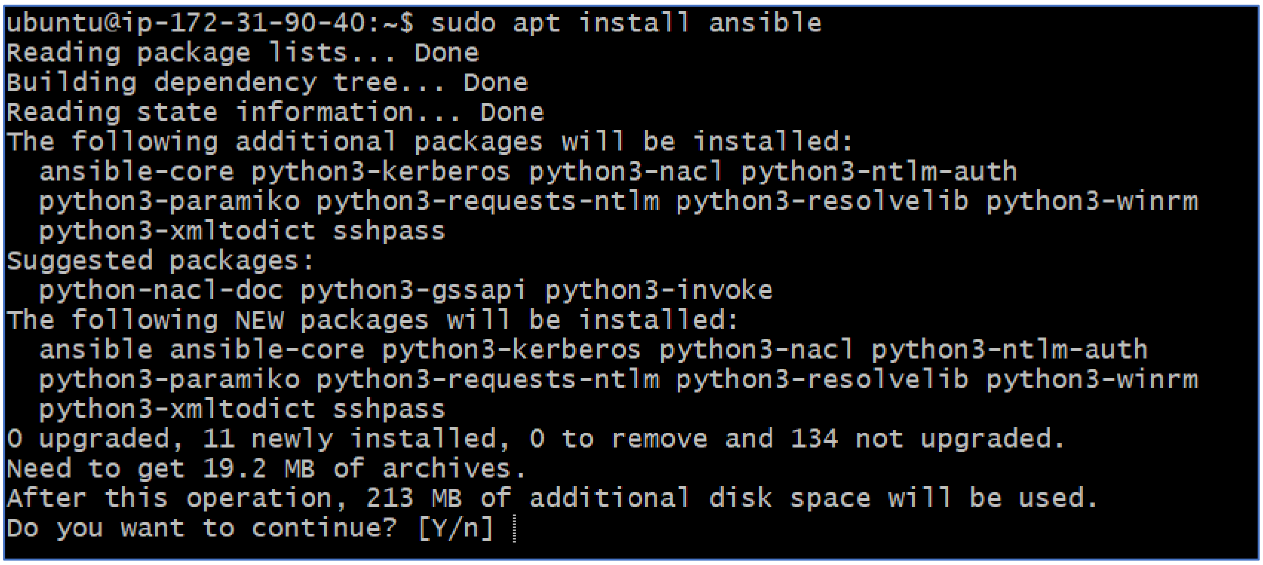
- After adding the Ansible PPA, we need to update the system's package list to include the new repository's information. This is done by running the command: `sudo apt update`

This ensures that the package manager has the latest information about available packages, including those from the newly added Ansible PPA.



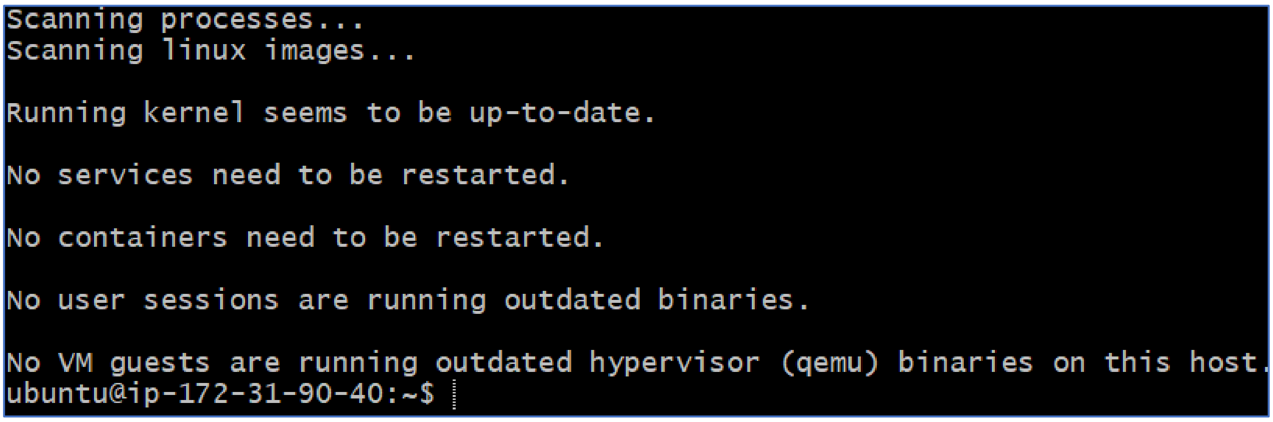
```
ubuntu@ip-172-31-90-40:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu 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
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-90-40:~$
```

- With the system's package list updated, we can now proceed to install Ansible. This is accomplished by executing the command:
  `sudo apt install ansible`
  This command instructs the package manager to download and install the Ansible package and its dependencies, making Ansible available for use on the `ansible-master` instance.



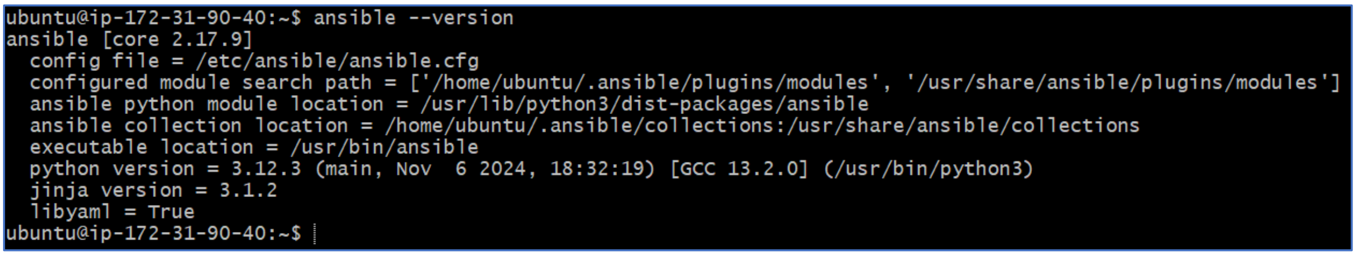
```
ubuntu@ip-172-31-90-40:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-kerberos python3-nacl python3-ntlm-auth
  python3-paramiko python3-requests-ntlm 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 134 not upgraded.
Need to get 19.2 MB of archives.
After this operation, 213 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

After initiating the Ansible installation with `sudo apt install ansible`, the system will prompt for confirmation. To proceed with the installation, type `Y` and press Enter.



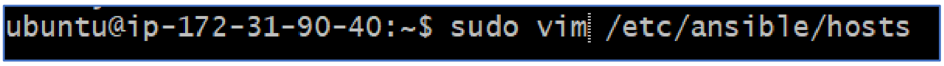
```
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-90-40:~$
```

- To verify the successful installation of Ansible and check the installed version, we can use the command: `ansible --version`
  This will display the version information of Ansible, confirming that it is correctly installed and ready for use.

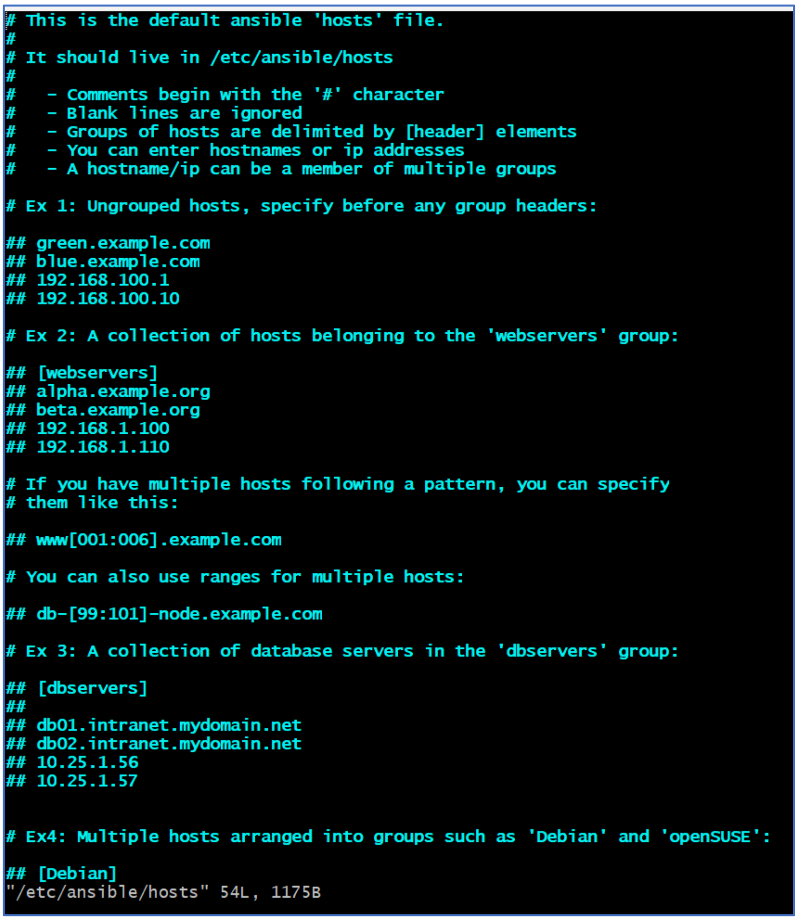


```
ubuntu@ip-172-31-90-40:~$ ansible --version
ansible [core 2.17.9]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/ubuntu/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Nov  6 2024, 18:32:19) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
ubuntu@ip-172-31-90-40:~$
```

- To configure Ansible for managing our target servers, we need to edit the Ansible inventory file, which contains information about the hosts Ansible will manage. This inventory file is typically located at **`/etc/ansible/hosts`**.
  We'll use the `vim` text editor to modify this file. To open the inventory file with `vim`, we'll use the command: `sudo vim /etc/ansible/hosts`
  This will allow us to add or modify entries for our target servers, enabling Ansible to connect to and manage them.



```
ubuntu@ip-172-31-90-40:~$ sudo vim /etc/ansible/hosts
```

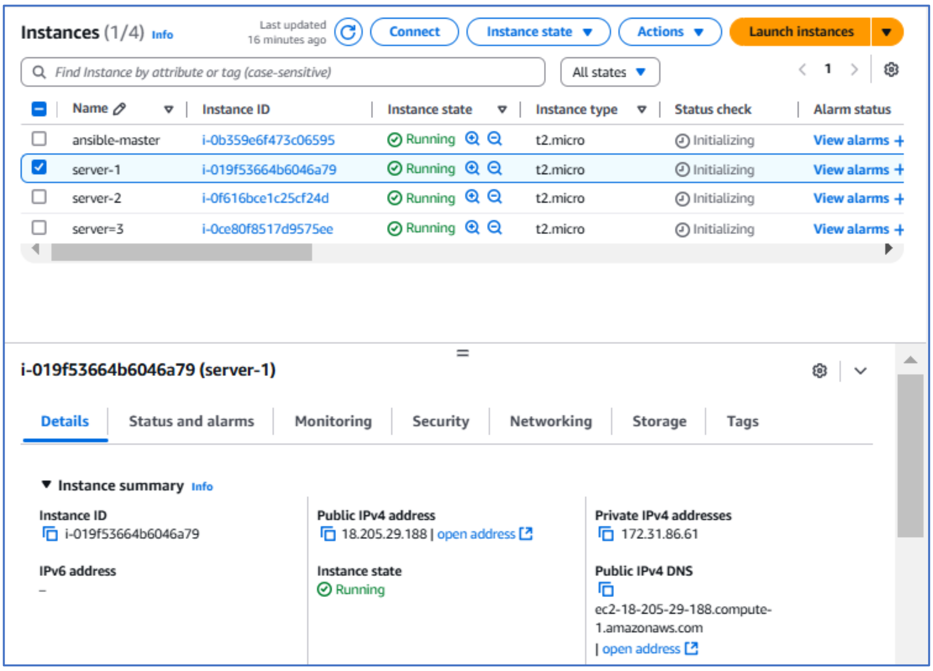


```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## 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:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# You can also use ranges for multiple hosts:

## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57


# Ex4: Multiple hosts arranged into groups such as 'Debian' and 'openSUSE':

## [Debian]
"/etc/ansible/hosts" 54L, 1175B
```

- Having launched our three EC2 instances, we need to obtain their public IP addresses.
  These IPs will then be entered into the `/etc/ansible/hosts` file under the group designation **`[servers]`.**
  Each server will be listed with its corresponding **`ansible_host`** parameter, which is set to its public IP.
  The configuration will appear as:

```
[servers]
server_1 ansible_host=44.201.246.198
server_2 ansible_host=3.82.112.155
server_3 ansible_host=54.211.111.210
```

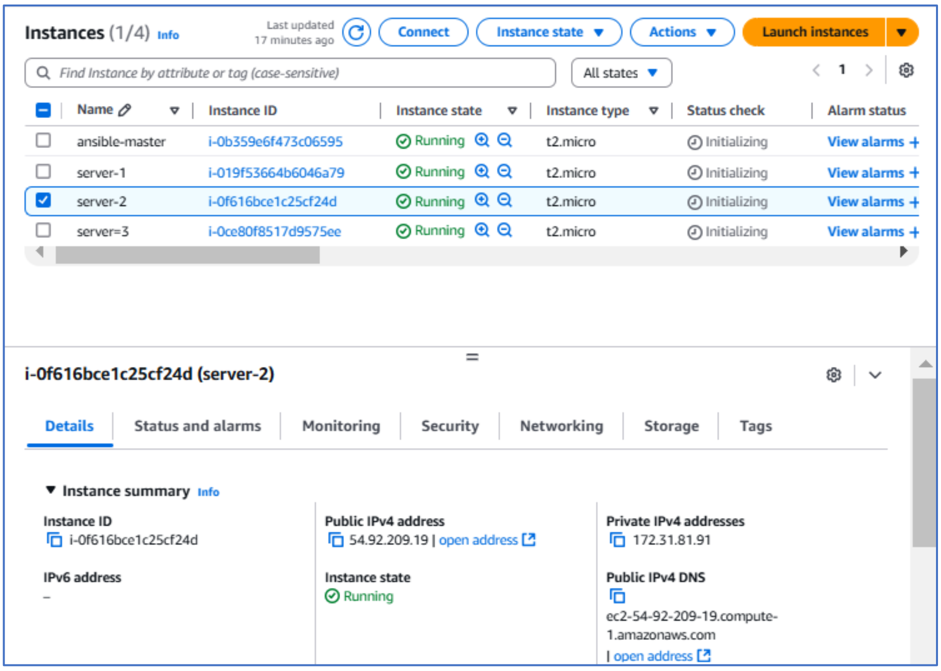
  This setup allows Ansible to recognize and connect to each of our target servers, enabling automated configuration and deployment.



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	ansible-master	i-0b359e6f473c06595	✓ Running	t2.micro	Initializing	View alarms +
☑	server-1	i-019f53664b6046a79	✓ Running	t2.micro	Initializing	View alarms +
□	server-2	i-0f616bce1c25cf24d	✓ Running	t2.micro	Initializing	View alarms +
□	server=3	i-0ce80f8517d9575ee	✓ Running	t2.micro	Initializing	View alarms +

i-019f53664b6046a79 (server-1)

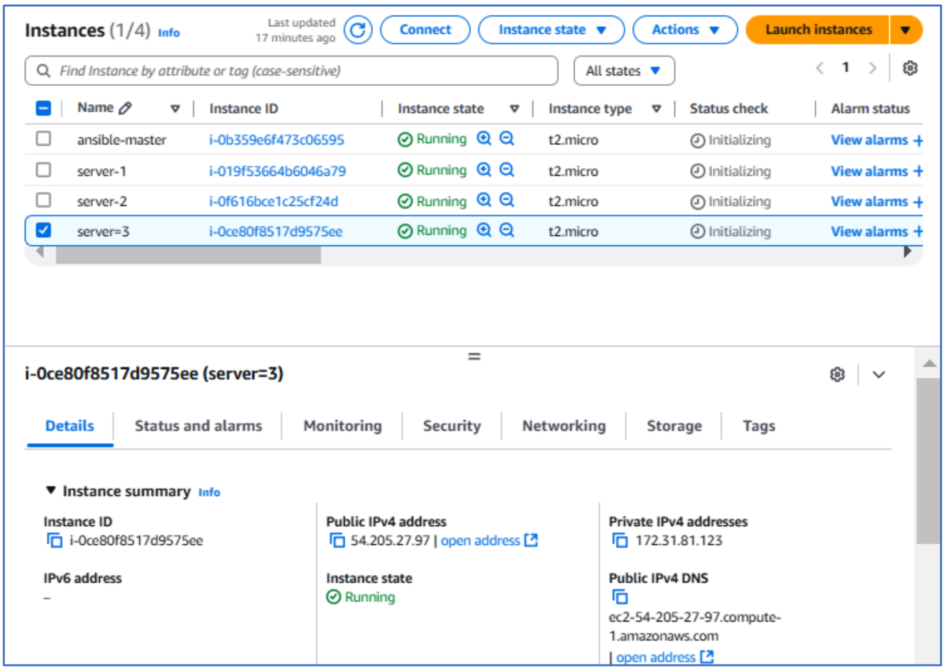
Instance ID: i-019f53664b6046a79
Public IPv4 address: 18.205.29.188 | open address
Private IPv4 addresses: 172.31.86.61
IPv6 address: –
Instance state: ✓ Running
Public IPv4 DNS: ec2-18-205-29-188.compute-1.amazonaws.com | open address



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	ansible-master	i-0b359e6f473c06595	✓ Running	t2.micro	Initializing	View alarms +
□	server-1	i-019f53664b6046a79	✓ Running	t2.micro	Initializing	View alarms +
☑	server-2	i-0f616bce1c25cf24d	✓ Running	t2.micro	Initializing	View alarms +
□	server=3	i-0ce80f8517d9575ee	✓ Running	t2.micro	Initializing	View alarms +

i-0f616bce1c25cf24d (server-2)

Instance ID: i-0f616bce1c25cf24d
Public IPv4 address: 54.92.209.19 | open address
Private IPv4 addresses: 172.31.81.91
IPv6 address: –
Instance state: ✓ Running
Public IPv4 DNS: ec2-54-92-209-19.compute-1.amazonaws.com | open address



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
□	ansible-master	i-0b359e6f473c06595	✓ Running	t2.micro	Initializing	View alarms +
□	server-1	i-019f53664b6046a79	✓ Running	t2.micro	Initializing	View alarms +
□	server-2	i-0f616bce1c25cf24d	✓ Running	t2.micro	Initializing	View alarms +
☑	server=3	i-0ce80f8517d9575ee	✓ Running	t2.micro	Initializing	View alarms +

i-0ce80f8517d9575ee (server=3)

▼ Instance summary Info

Instance ID
i-0ce80f8517d9575ee

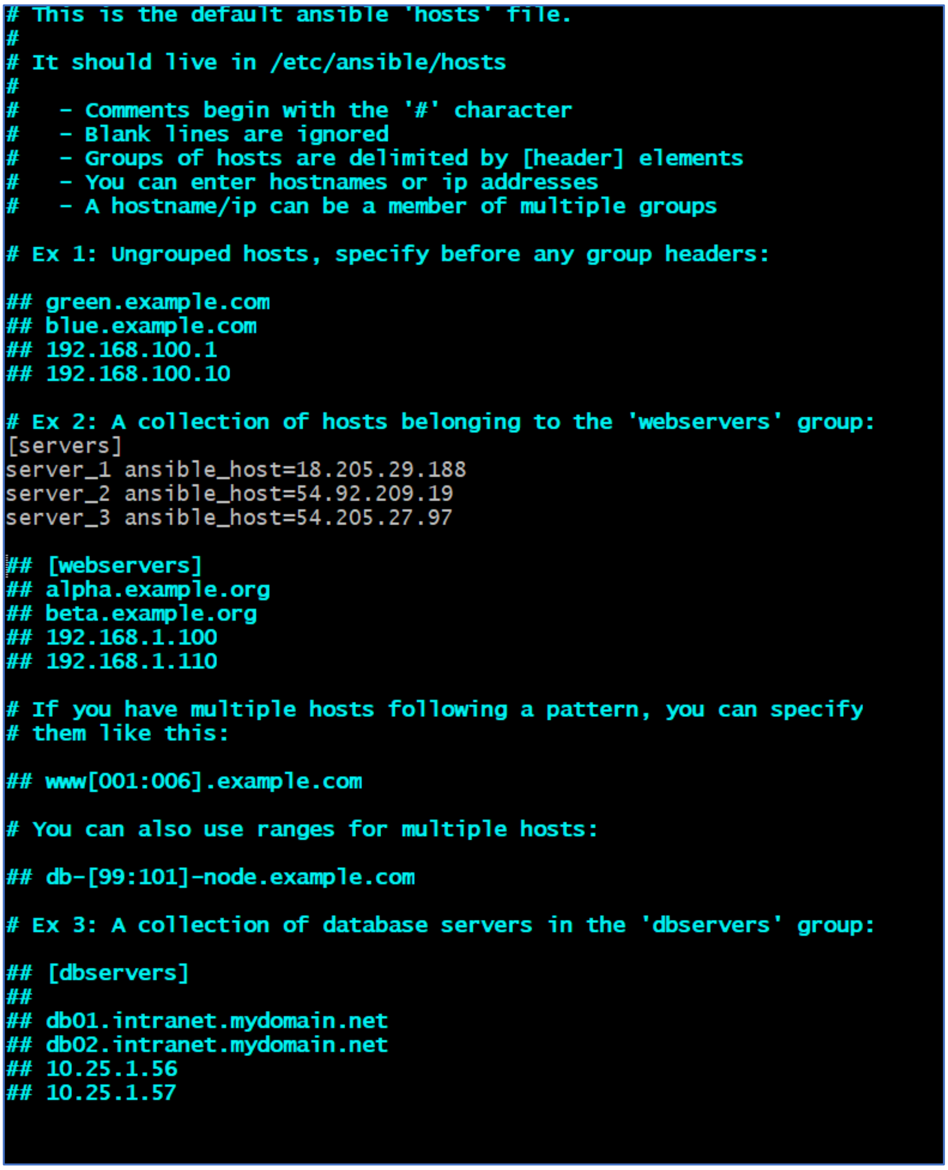
Public IPv4 address
54.205.27.97 | open address

Private IPv4 addresses
172.31.81.123

IPv6 address
–

Instance state
✓ Running

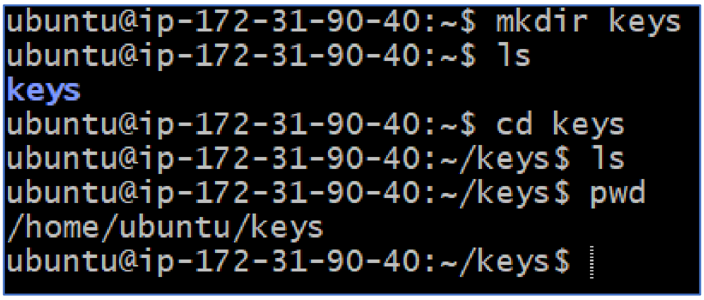
Public IPv4 DNS
ec2-54-205-27-97.compute-1.amazonaws.com
| open address



```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## 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=18.205.29.188
server_2 ansible_host=54.92.209.19
server_3 ansible_host=54.205.27.97

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# You can also use ranges for multiple hosts:

## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57
```

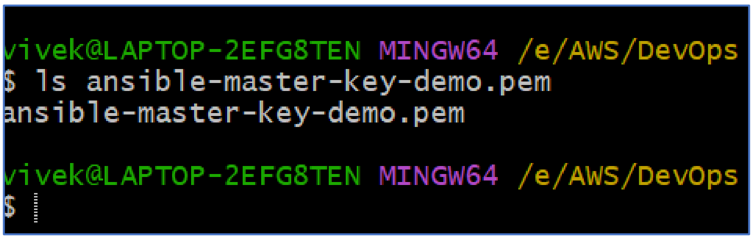
Establish Secure Communication

- To enable secure communication and management of our target servers, we need to provide the Ansible master instance with the necessary SSH key.
  This key is the same key pair that was generated during the creation of the EC2 instances. Since we have already saved this key pair locally, we need to securely copy it to the Ansible master instance.
  To do this, we first create a dedicated directory named **`keys`** within the home directory of the `ubuntu` user using the command: `mkdir keys`.
  We then navigate into this directory with the command: `cd keys`
  and confirm our location using `pwd`.
  The output should show that we are in the **`/home/ubuntu/keys`** directory.
  We will now securely copy the private key file into this directory, ensuring that the Ansible master can establish connections with the target servers (server-1, server-2, and server-3).



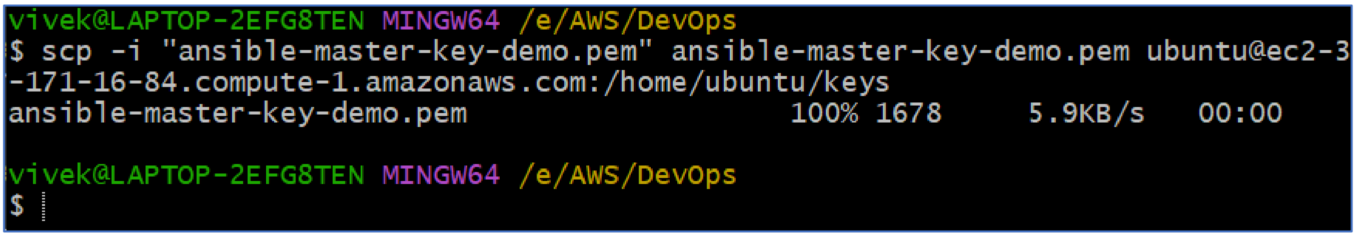
```
ubuntu@ip-172-31-90-40:~$ mkdir keys
ubuntu@ip-172-31-90-40:~$ ls
keys
ubuntu@ip-172-31-90-40:~$ cd keys
ubuntu@ip-172-31-90-40:~/keys$ ls
ubuntu@ip-172-31-90-40:~/keys$ pwd
/home/ubuntu/keys
ubuntu@ip-172-31-90-40:~/keys$
```

- Now, we navigate to the local directory where the SSH private key (**`ansible-master-key-demo.pem`**) is stored. In this case, it is located in the **`E:\AWS\DevOps`** directory, as indicated by the command prompt. We confirm the presence of the key file by running the command: `ls ansible-master-key-demo.pem`, which lists the file if it exists in the current directory.



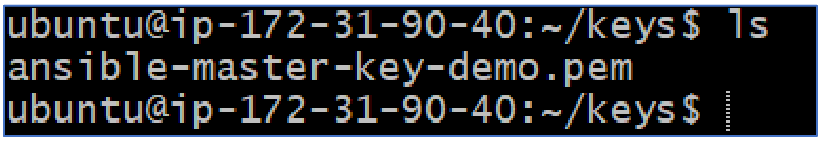
```
vivek@LAPTOP-2EFG8TEN MINGW64 /e/AWS/DevOps
$ ls ansible-master-key-demo.pem
ansible-master-key-demo.pem

vivek@LAPTOP-2EFG8TEN MINGW64 /e/AWS/DevOps
$
```

- Having already established the need for secure communication between our Ansible master instance and the target servers, we utilize the `scp` command to transfer the SSH private key (**`ansible-master-key-demo.pem`**) from our local machine to the `ansible-master` instance.
  This command allows for the secure copying of files over SSH, ensuring the key is transferred safely.
  We execute the command:
  `scp -i "ansible-master-key-demo.pem" ansible-master-key-demo.pem ubuntu@ec2-35-171-16-84.compute-1.amazonaws.com:/home/ubuntu/keys`
  specifying the key file to be used for authentication and the destination path on the `ansible-master` instance. The successful transfer is confirmed by the output message indicating 100% completion.



```
vivek@LAPTOP-2EFG8TEN MINGW64 /e/AWS/DevOps
$ scp -i "ansible-master-key-demo.pem" ansible-master-key-demo.pem ubuntu@ec2-3
-171-16-84.compute-1.amazonaws.com:/home/ubuntu/keys
ansible-master-key-demo.pem                   100% 1678     5.9KB/s   00:00

vivek@LAPTOP-2EFG8TEN MINGW64 /e/AWS/DevOps
$
```

- After successfully copying the **ansible-master-key-demo.pem** file to the **/home/ubuntu/keys** directory on the Ansible master instance, we verify its presence by listing the contents of the directory using the command: `ls`
  The output confirms that the key file is now present in the specified location, ensuring that Ansible can use it to authenticate with the target servers.



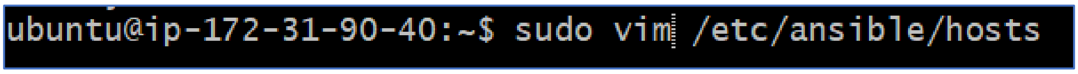
```
ubuntu@ip-172-31-90-40:~/keys$ ls
ansible-master-key-demo.pem
ubuntu@ip-172-31-90-40:~/keys$
```

Configure Ansible Inventory

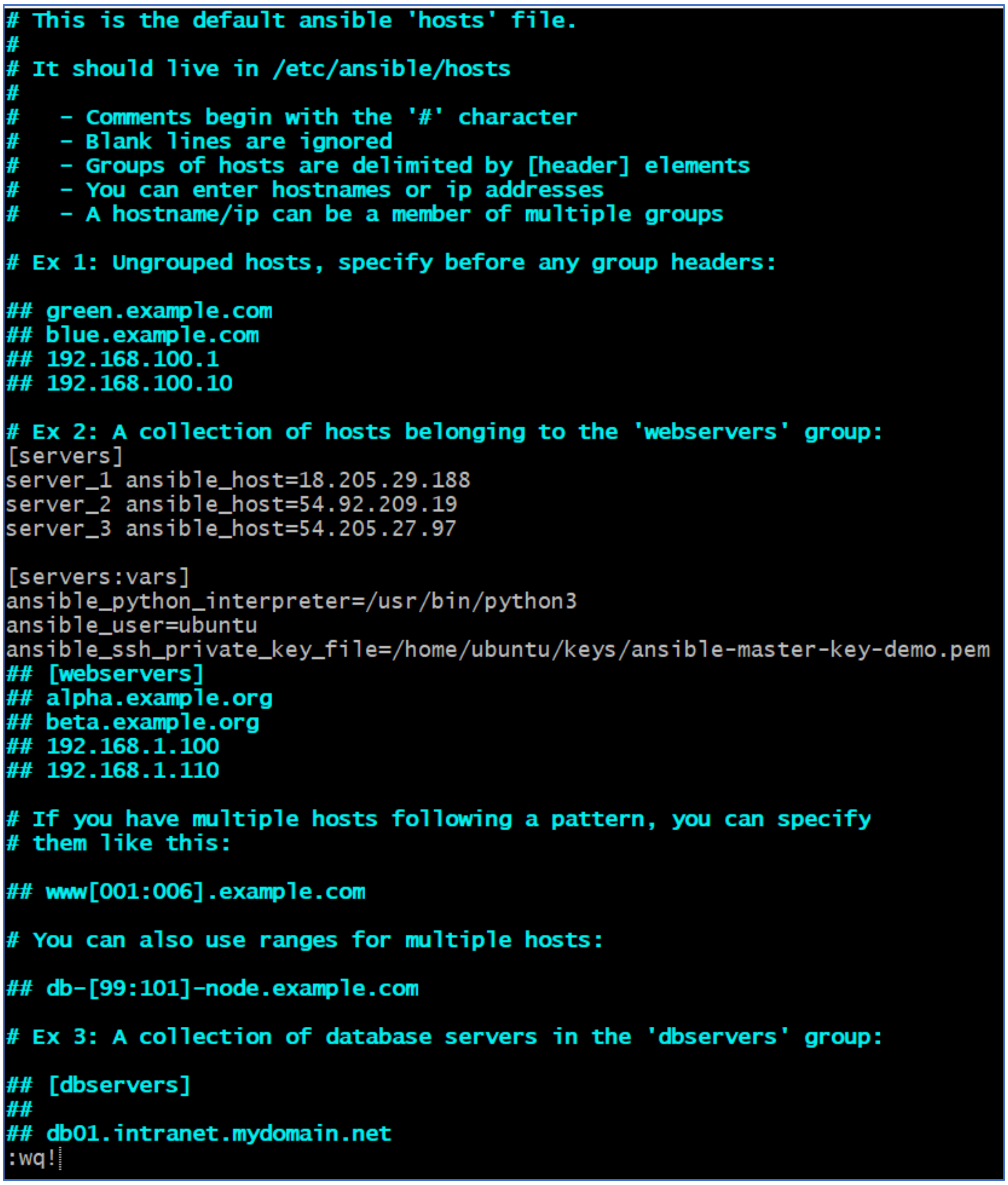
- To ensure Ansible can properly communicate with and manage our target servers, We further configure the **[servers]** group within the **/etc/ansible/hosts** file by adding a **[servers:vars]** section. This section allows us to define variables that apply specifically to the servers group.
  We define three key variables: **ansible_python_interpreter**, set to **/usr/bin/python3**, specifying the Python interpreter to be used on the target servers; **ansible_user**, set to **ubuntu**, indicating the username Ansible should use for connections; and
  ansible_ssh_private_key_file, which points to the location of the SSH private key file (**/home/ubuntu/keys/ansible-master-key-demo.pem**) used for authentication.

```
[servers:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-master-key-
demo.pem
```

  With these variables in place, Ansible is equipped to connect to and manage the target servers securely and efficiently.

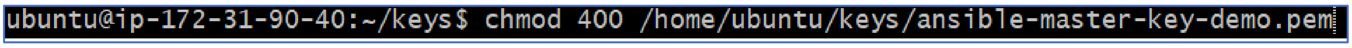


```
ubuntu@ip-172-31-90-40:~$ sudo vim /etc/ansible/hosts
```



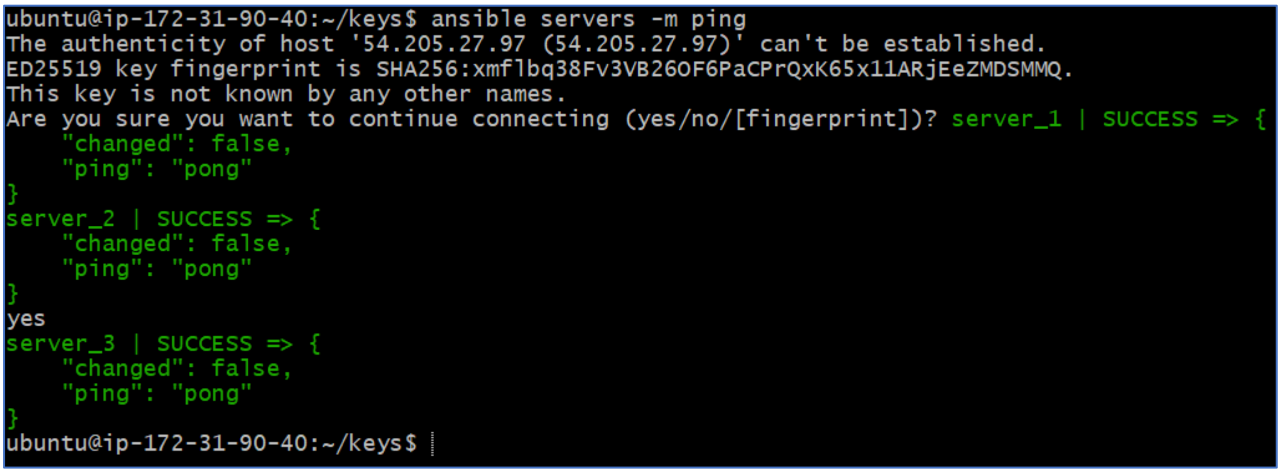
```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## 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=18.205.29.188
server_2 ansible_host=54.92.209.19
server_3 ansible_host=54.205.27.97

[servers:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-master-key-demo.pem
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# You can also use ranges for multiple hosts:

## db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
:wq!
```

- Protecting our SSH key is crucial for maintaining the security of our infrastructure. Therefore, we must adjust the permissions of the `ansible-master-key-demo.pem` file to restrict access.
  We achieve this using the `chmod` command, a powerful tool for managing file permissions in Linux.
  By executing: `chmod 400 /home/ubuntu/keys/ansible-master-key-demo.pem` we set the file permissions to `400`. This specific setting ensures that only the owner of the file (in this case, the `ubuntu` user) has read access, while preventing any other user from accessing the key's contents. This step is vital in safeguarding our Ansible setup and preventing unauthorized access to our servers.



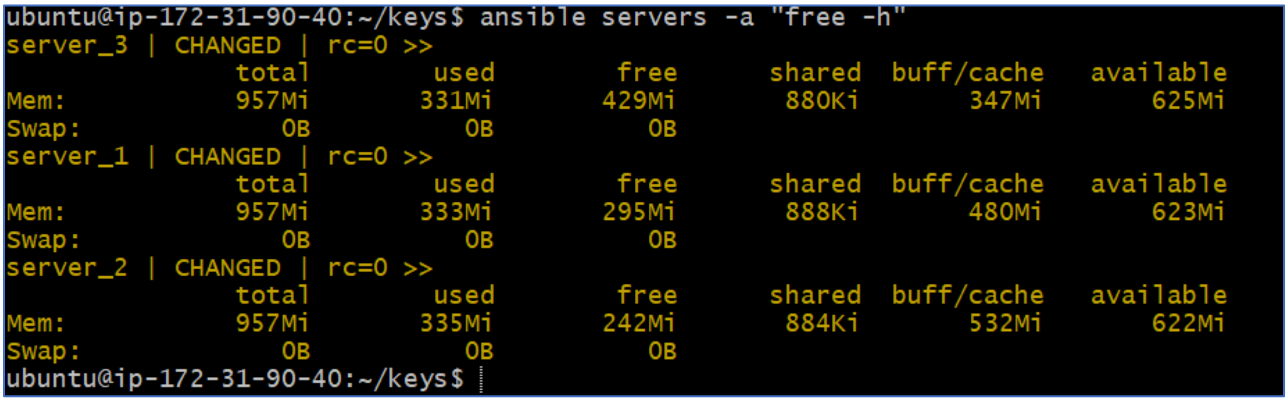
```
ubuntu@ip-172-31-90-40:~/keys$ chmod 400 /home/ubuntu/keys/ansible-master-key-demo.pem
```

- With our Ansible configuration in place, we can now test communication with our target servers.
  To achieve this, we use the command: `ansible servers -m ping`
  This command instructs Ansible to use the **`ping`** module to check connectivity with all servers listed in the **`servers`** group within our inventory file.
  A successful connection to each server will result in a **`SUCCESS`** message and a **`pong`** response, confirming that Ansible can communicate effectively with each instance.
  It's worth noting that during the first connection attempt,
  you might encounter a prompt asking **'Are you sure you want to continue connecting (yes/no/[fingerprint])?'**.
  This occurs because the servers are not yet listed in the **`known_hosts`** file. **Typing 'yes'** will add the servers to this file, preventing the prompt from appearing in future connection attempts.



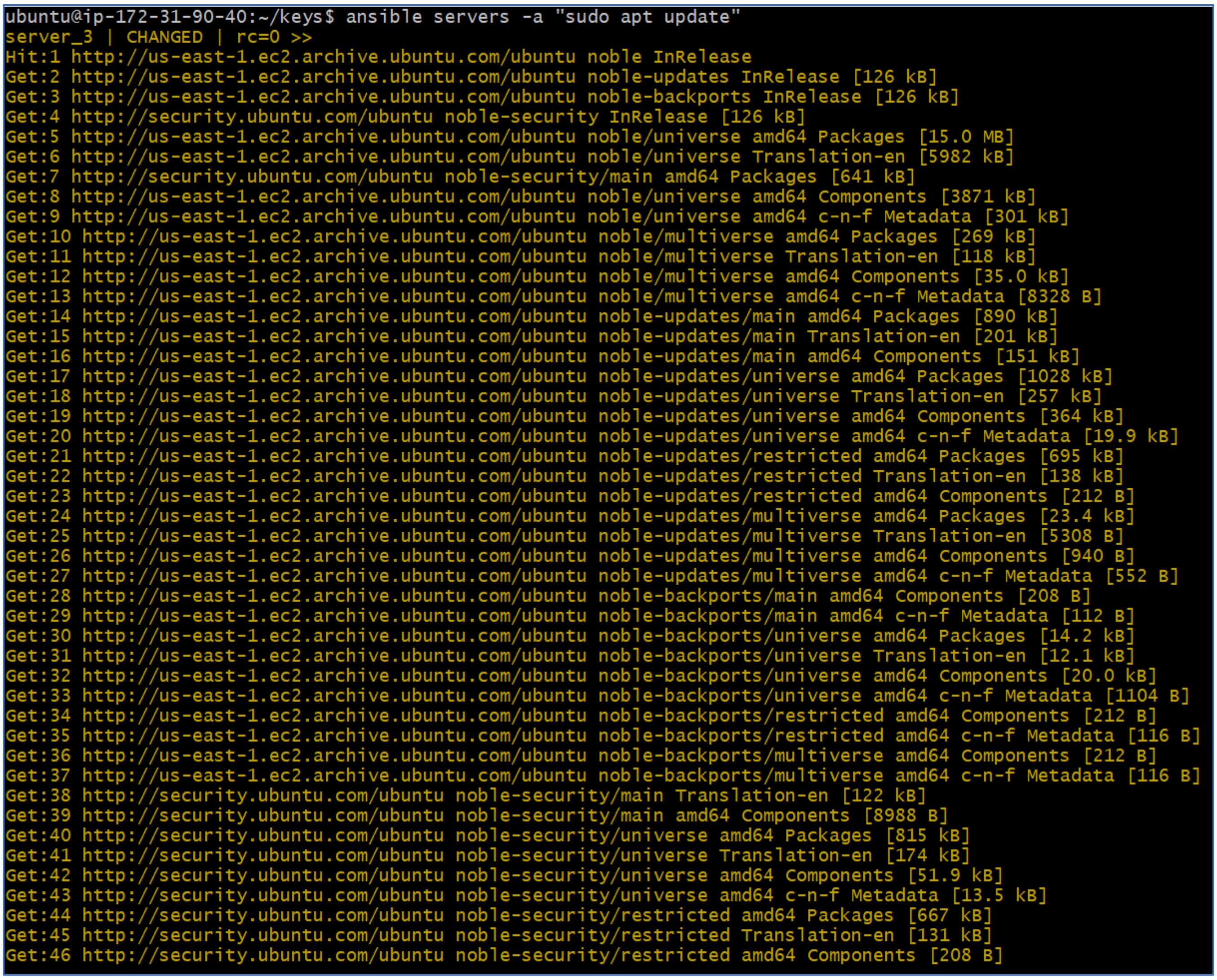
```
ubuntu@ip-172-31-90-40:~/keys$ ansible servers -m ping
The authenticity of host '54.205.27.97 (54.205.27.97)' can't be established.
ED25519 key fingerprint is SHA256:xmflbq38Fv3VB26OF6PaCPrQxK65x11ARjEeZMDSMMQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? server_1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
server_2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
yes
server_3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-90-40:~/keys$
```

- Having confirmed the connectivity between the Ansible master and our target servers, we can now utilize Ansible to gather information about those servers.
  To check the memory usage of each server,
  we execute the command: `ansible servers -a "free -h"`
  This instructs Ansible to run the `free -h` command on all servers in the **`servers`** group. The `free -h` command displays memory usage details in a human-readable format. The output presents the memory statistics for each server (server-1, server-2, and server-3), including total memory, used memory, free memory, and other relevant metrics. The **`CHANGED | rc=0`** in the output signifies that the command was executed successfully on each server, and we have retrieved the memory usage information.

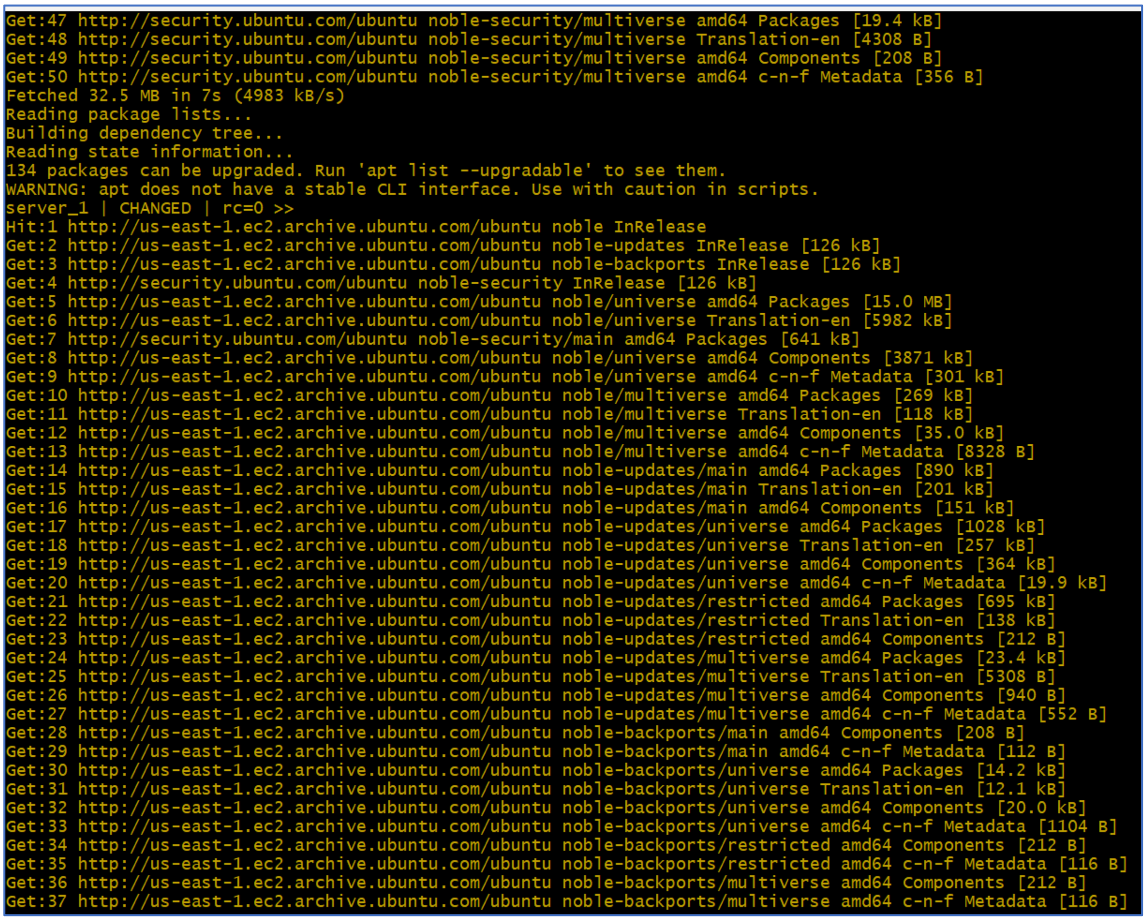


```
ubuntu@ip-172-31-90-40:~/keys$ ansible servers -a "free -h"
server_3 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           957Mi       331Mi       429Mi       880Ki       347Mi       625Mi
Swap:             0B          0B          0B
server_1 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           957Mi       333Mi       295Mi       888Ki       480Mi       623Mi
Swap:             0B          0B          0B
server_2 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:           957Mi       335Mi       242Mi       884Ki       532Mi       622Mi
Swap:             0B          0B          0B
ubuntu@ip-172-31-90-40:~/keys$
```

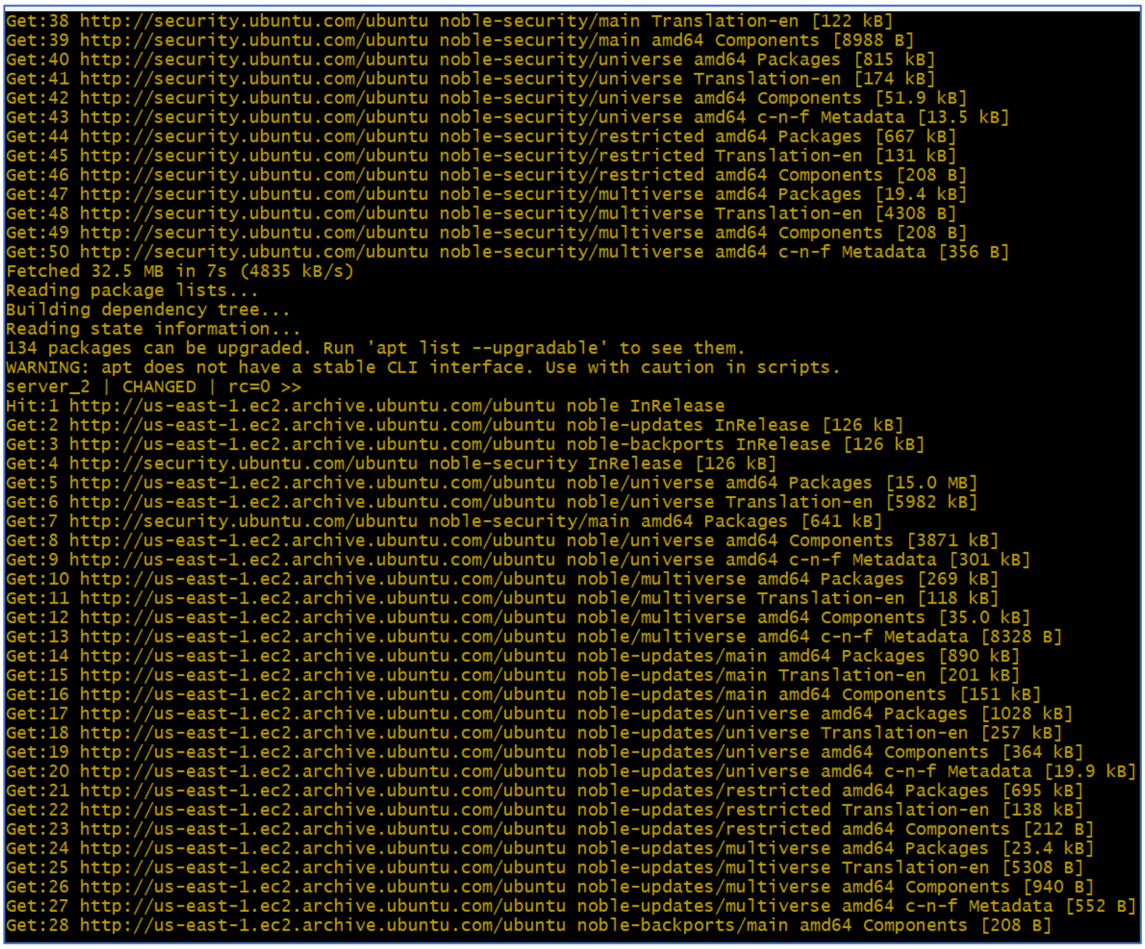
- To ensure our target servers have access to the latest software packages, we utilize Ansible to update their package lists.
  We execute the command: `ansible servers -a "sudo apt update"`
  which instructs Ansible to run the **`sudo apt update`** command on each server in the **`servers`** group. This command updates the package lists on each server, ensuring that the package manager has the most recent information about available software.
  The output shows the results of the update process on each server, including the retrieval of package lists from various repositories.
  The `CHANGED | rc=0` in the output indicates that the command was successfully executed on each server, and the package lists are now up to date.



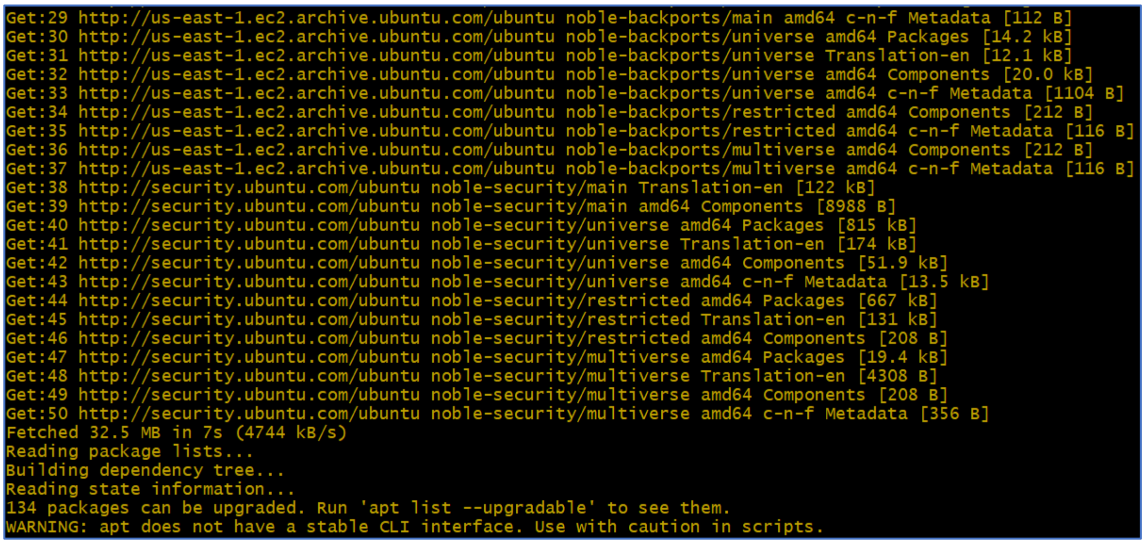
```
ubuntu@ip-172-31-90-40:~/keys$ ansible servers -a "sudo apt update"
server_3 | CHANGED | rc=0 >>
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [641 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [890 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [201 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1028 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [257 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [695 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [138 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [23.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5308 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.2 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [12.1 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [122 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8988 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [815 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [174 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [13.5 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [667 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [131 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
```



```
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [19.4 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4308 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 32.5 MB in 7s (4983 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
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://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [641 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [890 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [201 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1028 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [257 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [695 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [138 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [23.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5308 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.2 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [12.1 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
```

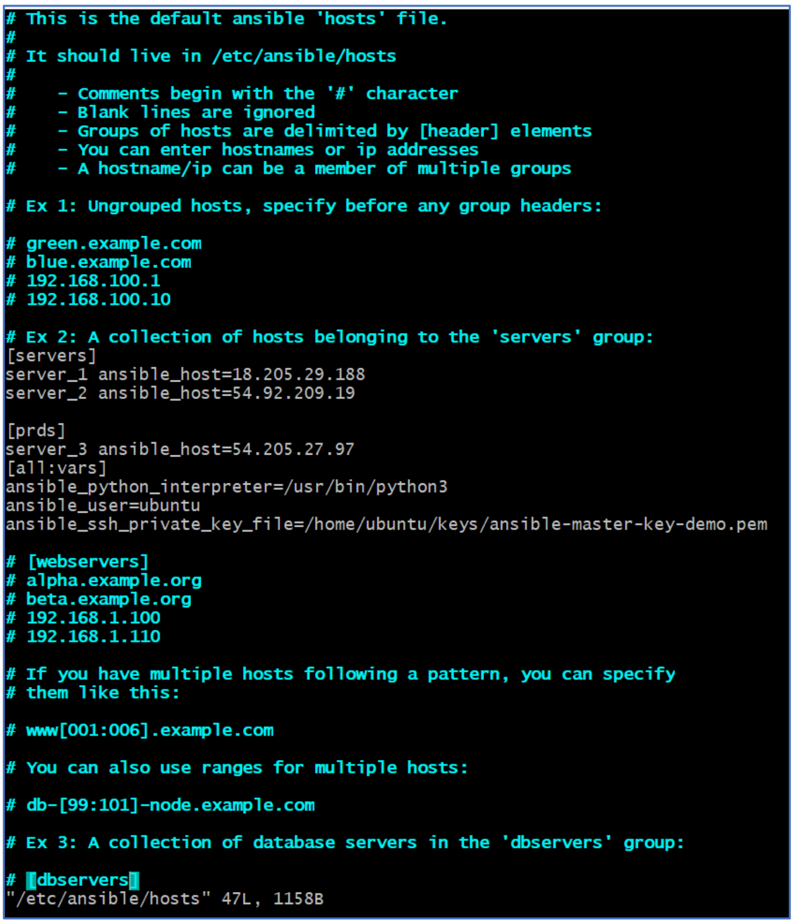


```
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [122 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8988 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [815 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [174 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [13.5 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [667 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [131 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [19.4 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4308 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 32.5 MB in 7s (4835 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
server_2 | 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://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [641 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [890 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [201 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1028 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [257 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [19.9 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [695 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [138 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [23.4 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5308 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [552 B]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
```



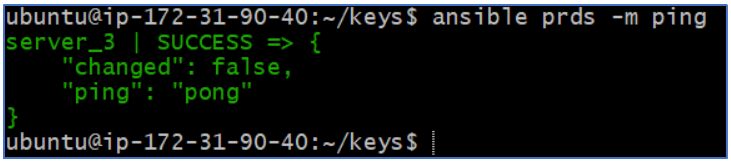
```
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [14.2 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [12.1 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [1104 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:38 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [122 kB]
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8988 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [815 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [174 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [51.9 kB]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [13.5 kB]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [667 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [131 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [19.4 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [4308 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [356 B]
Fetched 32.5 MB in 7s (4744 kB/s)
Reading package lists...
Building dependency tree...
Reading state information...
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
```

- To demonstrate Ansible's capability to manage multiple groups of servers, we will create a new group named `[prds]` in the `/etc/ansible/hosts` file. This new group will specifically host `server-3`, which was previously part of the `[servers]` group. This separation allows for more granular control and management, enabling us to apply specific configurations or actions to different groups of servers based on their roles or purposes.
  The updated `/etc/ansible/hosts` file will now include both the `[servers]` and `[prds]` groups, with `server-3` residing under the `[prds]` group while `server-1` and `server-2` reside under `[servers]` group.



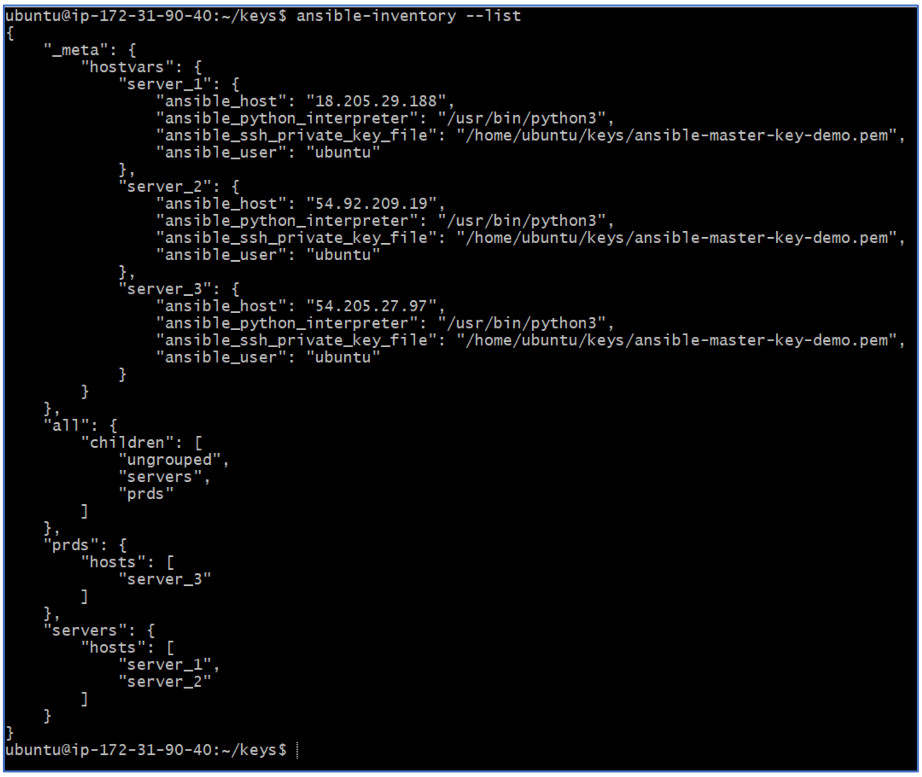
```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#    - Comments begin with the '#' character
#    - Blank lines are ignored
#    - Groups of hosts are delimited by [header] elements
#    - You can enter hostnames or ip addresses
#    - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

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

# Ex 2: A collection of hosts belonging to the 'servers' group:
[servers]
server_1 ansible_host=18.205.29.188
server_2 ansible_host=54.92.209.19

[prds]
server_3 ansible_host=54.205.27.97
[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/ubuntu/keys/ansible-master-key-demo.pem

# [webservers]
# alpha.example.org
# beta.example.org
# 192.168.1.100
# 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

# www[001:006].example.com

# You can also use ranges for multiple hosts:

# db-[99:101]-node.example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

# [dbservers]
"/etc/ansible/hosts" 47L, 1158B
```

- After reorganizing our server groups in the Ansible inventory, we need to confirm that Ansible can still effectively communicate with the servers in the new group.
  To verify this, we use the command: `ansible prds -m ping`
  targeting the `[prds]` group, which now contains `server-3`.
  This command instructs Ansible to send a ping request to `server-3` and report back on the connection status.
  A successful response, as indicated by the `SUCCESS` message and the `pong` output, confirms that Ansible can successfully reach and communicate with `server-3` in its new group. This verification step is crucial to ensure that our Ansible setup remains functional and capable of managing all servers, even after group modifications.



```
ubuntu@ip-172-31-90-40:~/keys$ ansible prds -m ping
server_3 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-90-40:~/keys$ 
```

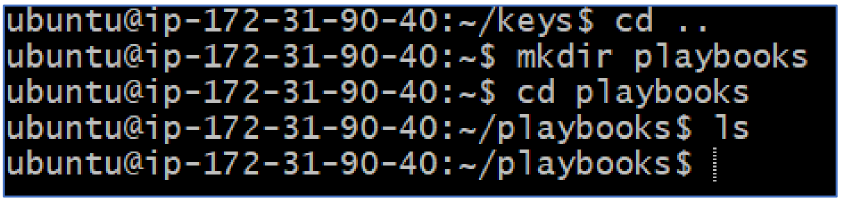
- To gain a comprehensive view of our Ansible inventory, we use the command:
  `ansible-inventory --list`
  This command generates a detailed output of the inventory in JSON format, encompassing all defined groups, hosts, and their associated variables.
  The output reveals the structure of our inventory, including the `_meta` section with host-specific variables like **`ansible_host`**, **`ansible_python_interpreter`**, **`ansible_ssh_private_key_file`**, and **`ansible_user`**.
  It also clearly shows the grouping of hosts under `all`, `prds`, and `servers`, highlighting the separation of `server-3` into the `prds` group.
  This command provides a valuable overview of our Ansible environment, confirming the organization and configuration of our managed servers.



```
ubuntu@ip-172-31-90-40:~/keys$ ansible-inventory --list
{
    "_meta": {
        "hostvars": {
            "server_1": {
                "ansible_host": "18.205.29.188",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/ansible-master-key-demo.pem",
                "ansible_user": "ubuntu"
            },
            "server_2": {
                "ansible_host": "54.92.209.19",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/ansible-master-key-demo.pem",
                "ansible_user": "ubuntu"
            },
            "server_3": {
                "ansible_host": "54.205.27.97",
                "ansible_python_interpreter": "/usr/bin/python3",
                "ansible_ssh_private_key_file": "/home/ubuntu/keys/ansible-master-key-demo.pem",
                "ansible_user": "ubuntu"
            }
        }
    },
    "all": {
        "children": [
            "ungrouped",
            "servers",
            "prds"
        ]
    },
    "prds": {
        "hosts": [
            "server_3"
        ]
    },
    "servers": {
        "hosts": [
            "server_1",
            "server_2"
        ]
    }
}
ubuntu@ip-172-31-90-40:~/keys$ 
```

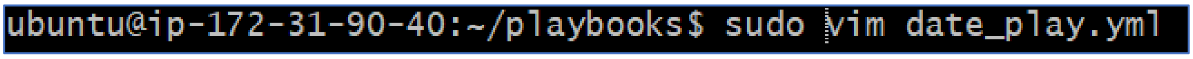
Create and Execute Ansible Playbooks

- To organize our Ansible playbooks, we will create a dedicated directory named **`playbooks`**.
  First, we navigate to the home directory of the `ubuntu` user from the current `/home/ubuntu/keys` directory by using the command: `cd ..`
  Then, we create the `playbooks` directory using the command: `mkdir playbooks`
  We then change our current directory to the newly created `playbooks` directory using the command: `cd playbooks`.
  Finally, we verify that we are in the correct directory by using `ls`, which shows that the directory is currently empty.



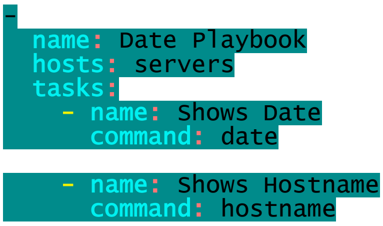
```
ubuntu@ip-172-31-90-40:~/keys$ cd ..
ubuntu@ip-172-31-90-40:~$ mkdir playbooks
ubuntu@ip-172-31-90-40:~$ cd playbooks
ubuntu@ip-172-31-90-40:~/playbooks$ ls
ubuntu@ip-172-31-90-40:~/playbooks$
```

- Now that we have organized our playbooks directory, we can begin creating our first Ansible playbook.
  We will use the `vim` text editor to create a file named **`date_play.yml`** within the `/home/ubuntu/playbooks` directory.
  To do this, we execute the command: `sudo vim date_play.yml`.
  This command opens the `date_play.yml` file in `vim` with superuser privileges, allowing us to edit and save the playbook. If the file doesn't already exist, `vim` will conveniently create it for us.



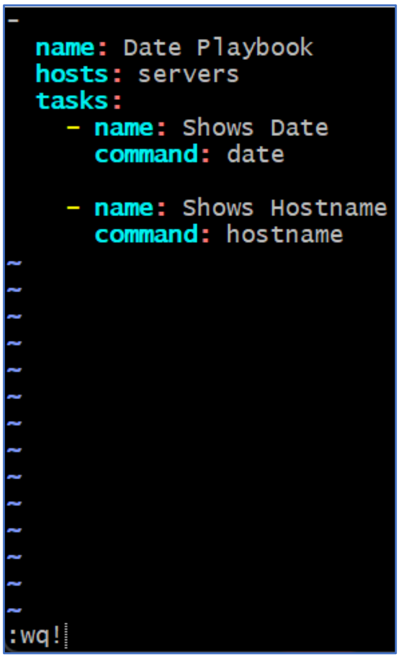
```
ubuntu@ip-172-31-90-40:~/playbooks$ sudo vim date_play.yml
```

- Now that we have created the `date_play.yml` playbook, we can execute it to see Ansible in action.



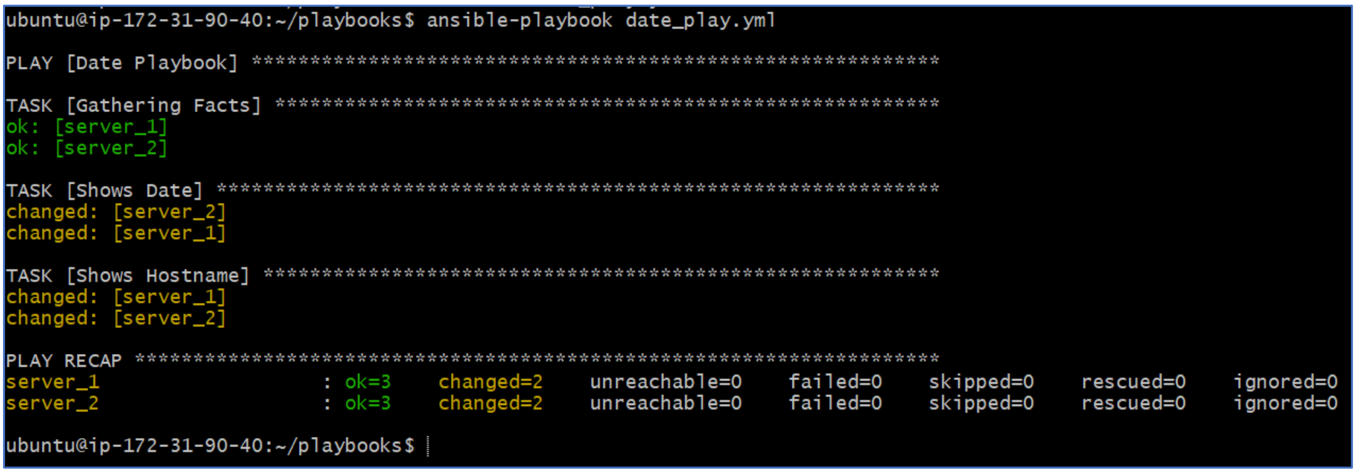
```
-
  name: Date Playbook
  hosts: servers
  tasks:
    - name: Shows Date
      command: date

    - name: Shows Hostname
      command: hostname
```

- This playbook, named 'Date Playbook', targets the `servers` group, which currently includes `server_1` and `server_2`.
  It comprises two main tasks: 'Shows Date', which executes the `date` command to display the current date and time on each server, and 'Shows Hostname', which executes the `hostname` command to display the hostname of each server.
  This simple playbook demonstrates the fundamental structure and functionality of Ansible playbooks, setting the stage for more complex automation tasks.
  To run the playbook, we use the command `ansible-playbook date_play.yml`. This instructs Ansible to read and execute the tasks defined in the playbook. The output will show the progress of the playbook execution, including details about each task and its status on the target servers.



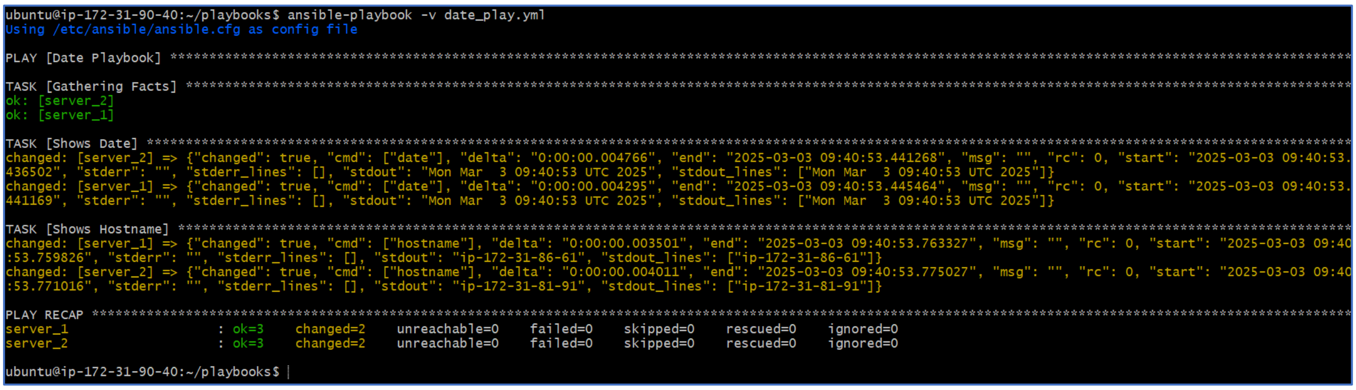
```
-
  name: Date Playbook
  hosts: servers
  tasks:
    - name: Shows Date
      command: date

    - name: Shows Hostname
      command: hostname
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

- To test the functionality of our `date_play.yml` playbook, we executed it using the command: `ansible-playbook date_play.yml`
  The output provides a detailed overview of the playbook's execution process. Initially, Ansible gathers facts about the target servers (`server_1` and `server_2`), ensuring it has the necessary information to proceed.
  Subsequently, the playbook executes the two defined tasks: 'Shows Date' and 'Shows Hostname'.
  The `changed` status for these tasks indicates that the `date` and `hostname` commands were successfully executed on both servers, producing the expected output – the current date and time, and the respective hostnames.
  The final 'PLAY RECAP' section confirms the overall success of the playbook, with all tasks completing without errors.
  This successful execution demonstrates the basic functionality of Ansible and its ability to automate tasks across multiple servers.



```
ubuntu@ip-172-31-90-40:~/playbooks$ ansible-playbook date_play.yml

PLAY [Date Playbook] *********************************************************

TASK [Gathering Facts] *******************************************************
ok: [server_1]
ok: [server_2]

TASK [Shows Date] ************************************************************
changed: [server_2]
changed: [server_1]

TASK [Shows Hostname] ********************************************************
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-172-31-90-40:~/playbooks$
```

- To gain a more detailed understanding of the playbook's execution, we ran the command: `ansible-playbook -v date_play.yml`
  The `-v` flag activates verbose output, providing granular information about each step.
  The verbose output includes specifics like the exact command executed, timestamps, and return codes, confirming successful execution and offering insights for debugging.
  This level of detail is valuable for troubleshooting and gaining a deeper understanding of the playbook's behavior.
  The final recap summarizes the successful completion of all tasks, reinforcing the effectiveness of our Ansible playbook.

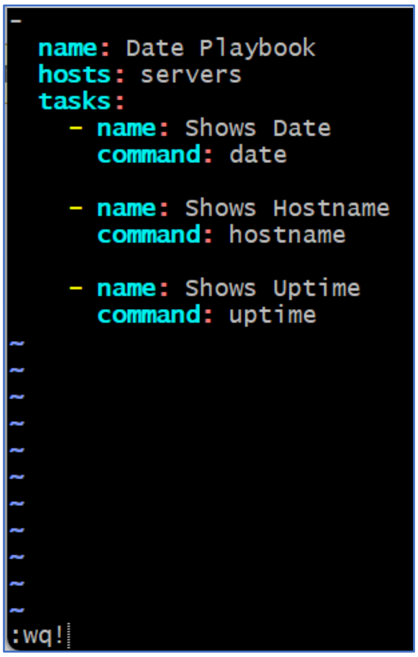


```
ubuntu@ip-172-31-90-40:~/playbooks$ ansible-playbook -v date_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Date Playbook] ***************************************************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************************
ok: [server_2]
ok: [server_1]

TASK [Shows Date] ******************************************************************************************************************************
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004766", "end": "2025-03-03 09:40:53.441268", "msg": "", "rc": 0, "start": "2025-03-03 09:40:53.
436502", "stderr": "", "stderr_lines": [], "stdout": "Mon Mar  3 09:40:53 UTC 2025", "stdout_lines": ["Mon Mar  3 09:40:53 UTC 2025"]}
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004295", "end": "2025-03-03 09:40:53.445464", "msg": "", "rc": 0, "start": "2025-03-03 09:40:53.
441169", "stderr": "", "stderr_lines": [], "stdout": "Mon Mar  3 09:40:53 UTC 2025", "stdout_lines": ["Mon Mar  3 09:40:53 UTC 2025"]}

TASK [Shows Hostname] **************************************************************************************************************************
changed: [server_1] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.003501", "end": "2025-03-03 09:40:53.763327", "msg": "", "rc": 0, "start": "2025-03-03 09:40
:53.759826", "stderr": "", "stderr_lines": [], "stdout": "ip-172-31-86-61", "stdout_lines": ["ip-172-31-86-61"]}
changed: [server_2] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.004011", "end": "2025-03-03 09:40:53.775027", "msg": "", "rc": 0, "start": "2025-03-03 09:40
:53.771016", "stderr": "", "stderr_lines": [], "stdout": "ip-172-31-81-91", "stdout_lines": ["ip-172-31-81-91"]}

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-172-31-90-40:~/playbooks$
```

- We have now expanded our `date_play.yml` playbook to include a new task, designed to retrieve the system uptime of our target servers.
  Using `vim`, we modified the playbook to add a task named 'Shows Uptime' which executes the `uptime` command on each server within the `servers` group.

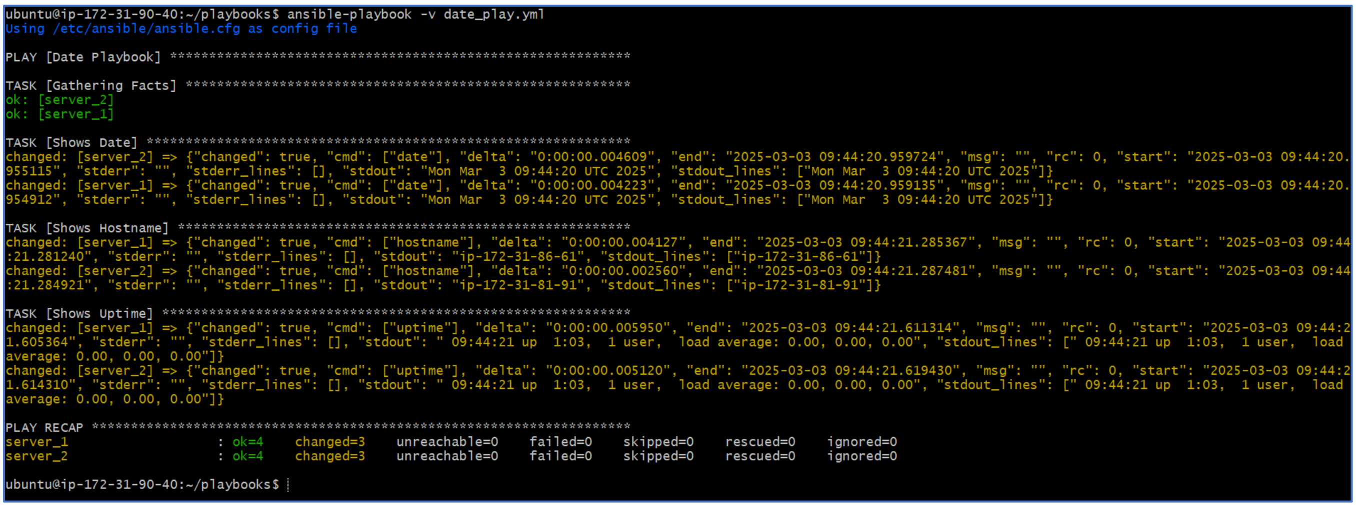
```
-
  name: Date Playbook
  hosts: servers
  tasks:
    - name: Shows Date
      command: date

    - name: Shows Hostname
      command: hostname

    - name: Shows Uptime
      command: uptime
```

  This addition allows us to gather more comprehensive system information with a single playbook execution.
  The updated playbook now includes three tasks: 'Shows Date', 'Shows Hostname', and 'Shows Uptime'.
  After adding the 'Shows Uptime' task, we saved and closed the `date_play.yml` file using the `:wq!` command in `vim`.
  This expanded playbook demonstrates Ansible's flexibility in managing multiple system tasks across our managed servers.



```
-
  name: Date Playbook
  hosts: servers
  tasks:
    - name: Shows Date
      command: date

    - name: Shows Hostname
      command: hostname

    - name: Shows Uptime
      command: uptime
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

- To incorporate the new 'Shows Uptime' task into our playbook execution, we ran the command: `ansible-playbook -v date_play.yml`.
  The verbose output, enabled by the `-v` flag, provided a detailed breakdown of each step.
  As before, Ansible successfully gathered facts about the servers and executed the 'Shows Date' and 'Shows Hostname' tasks, with the `changed` status confirming the successful execution of the `date` and `hostname` commands.
  The addition of the 'Shows Uptime' task introduced the `uptime` command, which also executed successfully, displaying the system uptime for each server.
  The verbose output provided detailed information about the execution of this new task, including timestamps and return codes, alongside the existing tasks.
  The `PLAY RECAP` section summarized the successful completion of all four tasks, with no errors or skipped tasks, demonstrating the expanded functionality of our Ansible playbook.

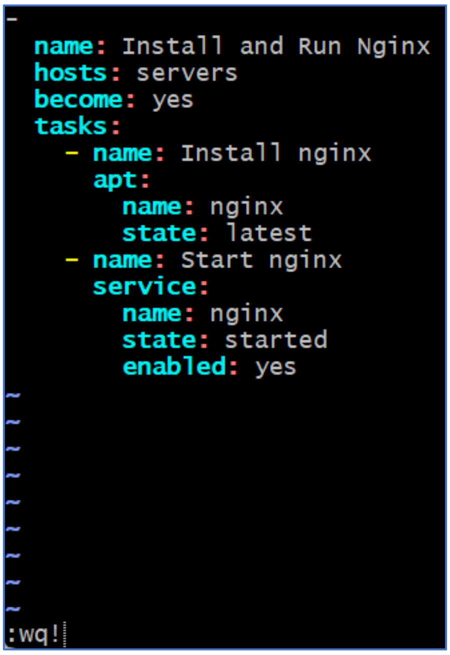


```
ubuntu@ip-172-31-90-40:~/playbooks$ ansible-playbook -v date_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Date Playbook] ***********************************************************

TASK [Gathering Facts] *********************************************************
ok: [server_2]
ok: [server_1]

TASK [Shows Date] **************************************************************
changed: [server_2] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004609", "end": "2025-03-03 09:44:20.959724", "msg": "", "rc": 0, "start": "2025-03-03 09:44:20.
955115", "stderr": "", "stderr_lines": [], "stdout": "Mon Mar  3 09:44:20 UTC 2025", "stdout_lines": ["Mon Mar  3 09:44:20 UTC 2025"]}
changed: [server_1] => {"changed": true, "cmd": ["date"], "delta": "0:00:00.004223", "end": "2025-03-03 09:44:20.959135", "msg": "", "rc": 0, "start": "2025-03-03 09:44:20.
954912", "stderr": "", "stderr_lines": [], "stdout": "Mon Mar  3 09:44:20 UTC 2025", "stdout_lines": ["Mon Mar  3 09:44:20 UTC 2025"]}

TASK [Shows Hostname] **********************************************************
changed: [server_1] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.004127", "end": "2025-03-03 09:44:21.285367", "msg": "", "rc": 0, "start": "2025-03-03 09:44
:21.281240", "stderr": "", "stderr_lines": [], "stdout": "ip-172-31-86-61", "stdout_lines": ["ip-172-31-86-61"]}
changed: [server_2] => {"changed": true, "cmd": ["hostname"], "delta": "0:00:00.002560", "end": "2025-03-03 09:44:21.287481", "msg": "", "rc": 0, "start": "2025-03-03 09:44
:21.284921", "stderr": "", "stderr_lines": [], "stdout": "ip-172-31-81-91", "stdout_lines": ["ip-172-31-81-91"]}

TASK [Shows Uptime] ************************************************************
changed: [server_1] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.005950", "end": "2025-03-03 09:44:21.611314", "msg": "", "rc": 0, "start": "2025-03-03 09:44:2
1.605364", "stderr": "", "stderr_lines": [], "stdout": " 09:44:21 up  1:03,  1 user,  load average: 0.00, 0.00, 0.00", "stdout_lines": [" 09:44:21 up  1:03,  1 user,  load
average: 0.00, 0.00, 0.00"]}
changed: [server_2] => {"changed": true, "cmd": ["uptime"], "delta": "0:00:00.005120", "end": "2025-03-03 09:44:21.619430", "msg": "", "rc": 0, "start": "2025-03-03 09:44:2
1.614310", "stderr": "", "stderr_lines": [], "stdout": " 09:44:21 up  1:03,  1 user,  load average: 0.00, 0.00, 0.00", "stdout_lines": [" 09:44:21 up  1:03,  1 user,  load
average: 0.00, 0.00, 0.00"]}

PLAY RECAP *********************************************************************
server_1                   : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2                   : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-90-40:~/playbooks$
```

- To further showcase Ansible's automation capabilities, we created a new playbook named **install_nginx_play.yml**.
  This playbook focuses on installing and running the Nginx web server on the servers in the `servers` group.
  Using `sudo vim install_nginx_play.yml`, we defined the playbook with the **become: yes** directive, allowing it to execute commands with root privileges.

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

  The playbook consists of **two tasks: 'Install nginx'**, which uses the **apt** module to install the latest version of Nginx, and **'Start nginx'**, which utilizes the **service** module to start the Nginx service, ensuring it's enabled and running.
  After defining these tasks, we saved and closed the **install_nginx_play.yml** file using the: `wq!` command in `vim`.
  This new playbook demonstrates Ansible's ability to handle software installation and service management, further expanding our automation toolkit.



```
-
  name: Install and Run Nginx
  hosts: servers
  become: yes
  tasks:
    - name: Install nginx
      apt:
        name: nginx
        state: latest
    - name: Start nginx
      service:
        name: nginx
        state: started
        enabled: yes
~
~
~
~
~
~
~
~
~
:wq!
```

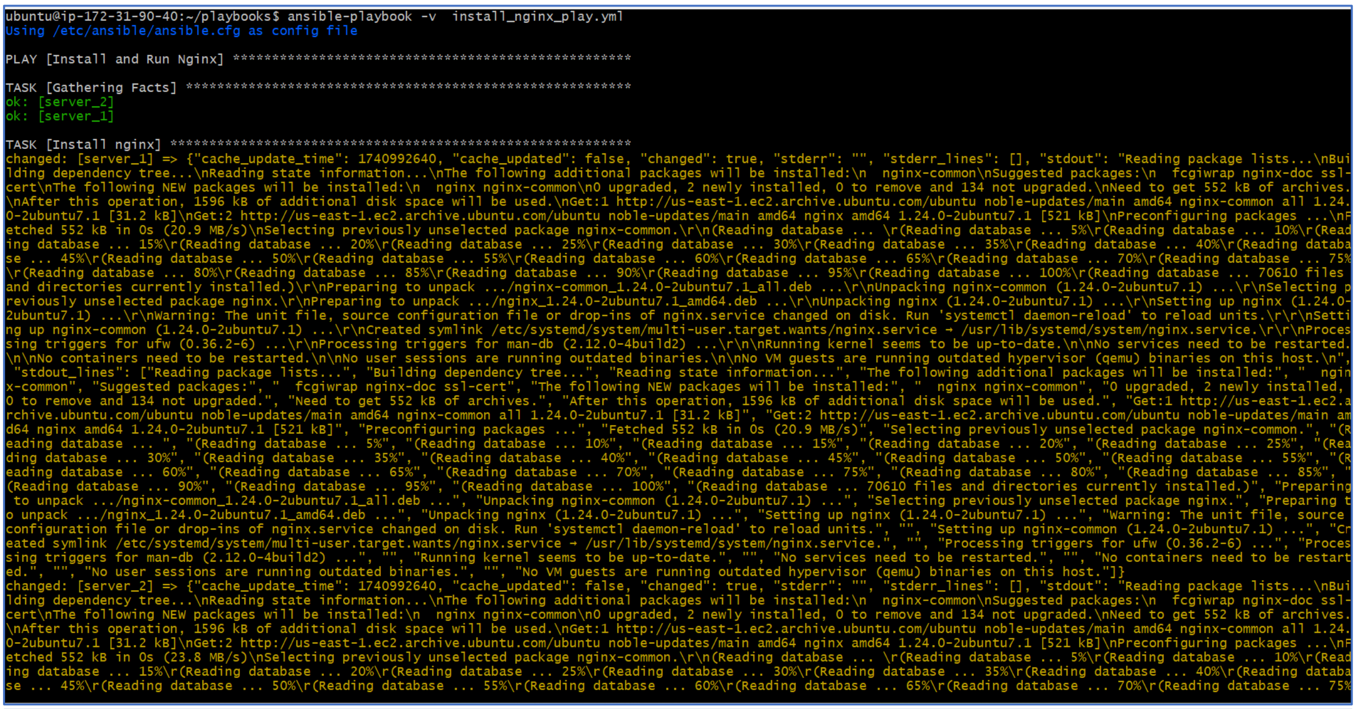
- To automate the installation and startup of Nginx on our servers, we executed the **install_nginx_play.yml** playbook using the command:
  `ansible-playbook -v install_nginx_play.yml`.
  The `-v` flag enabled verbose output, providing a detailed view of the playbook's execution.

The output confirms that Ansible successfully gathered facts about both `server_1` and `server_2`.

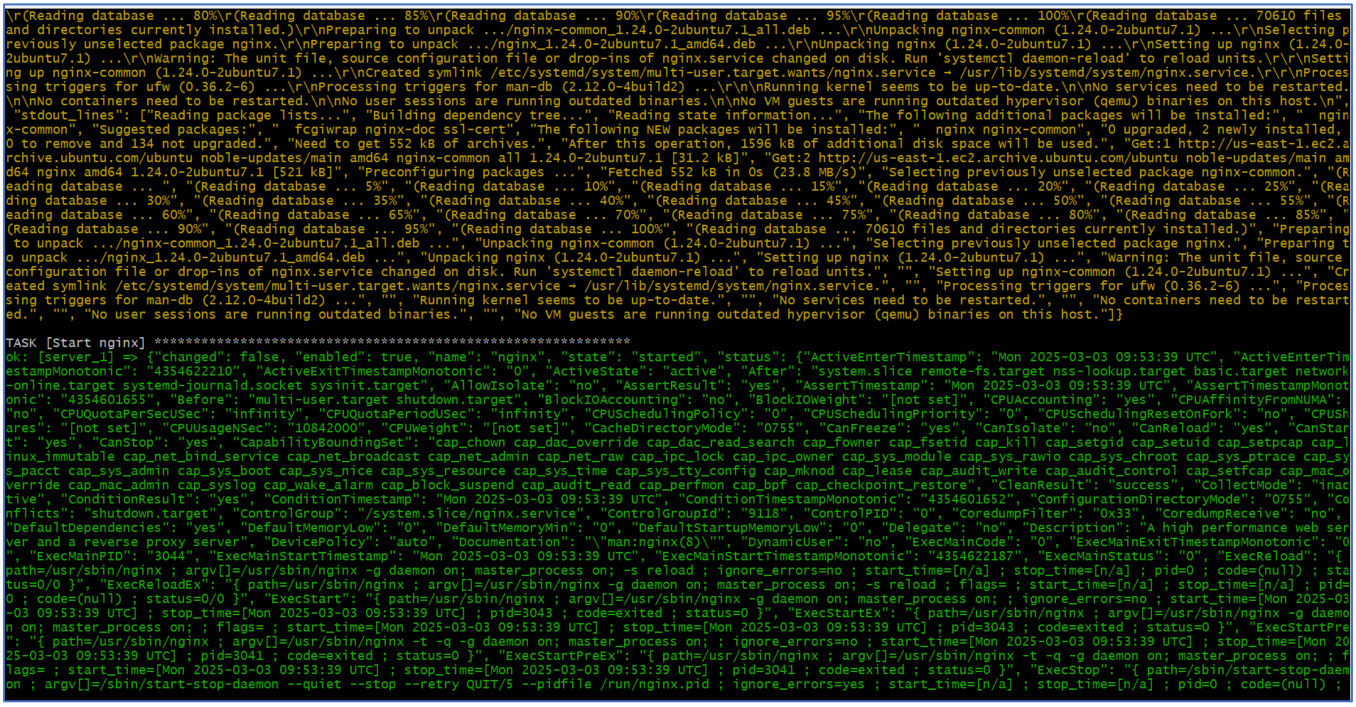
The 'Install nginx' task then proceeded to install Nginx using the `apt` module. The `changed` status indicates that Nginx was successfully installed on both servers, with the output detailing the package installation process.

Following this, the 'Start nginx' task used the `service` module to start the Nginx service on both servers, also resulting in a `changed` status.

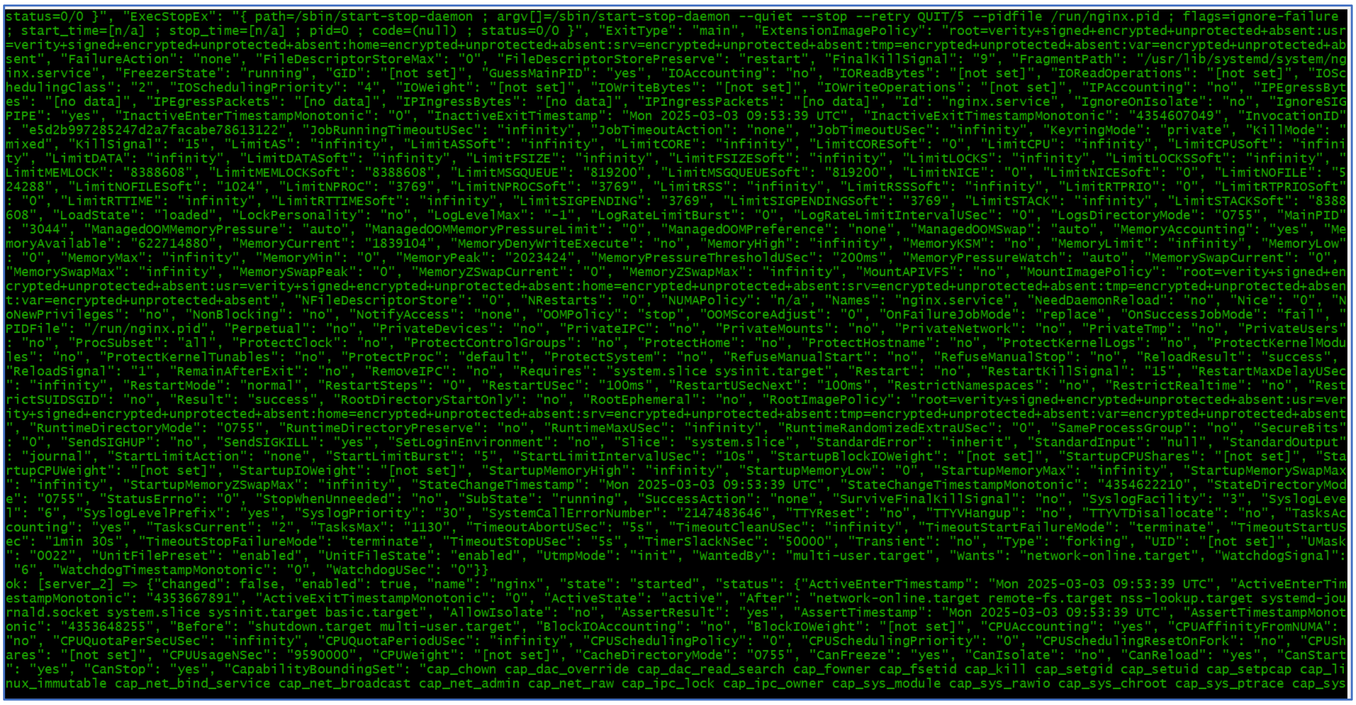
The verbose output provided detailed information about the service startup, ensuring that Nginx was not only installed but also properly started and enabled. The `PLAY RECAP` section confirmed the successful completion of all tasks, with no errors or skipped tasks.



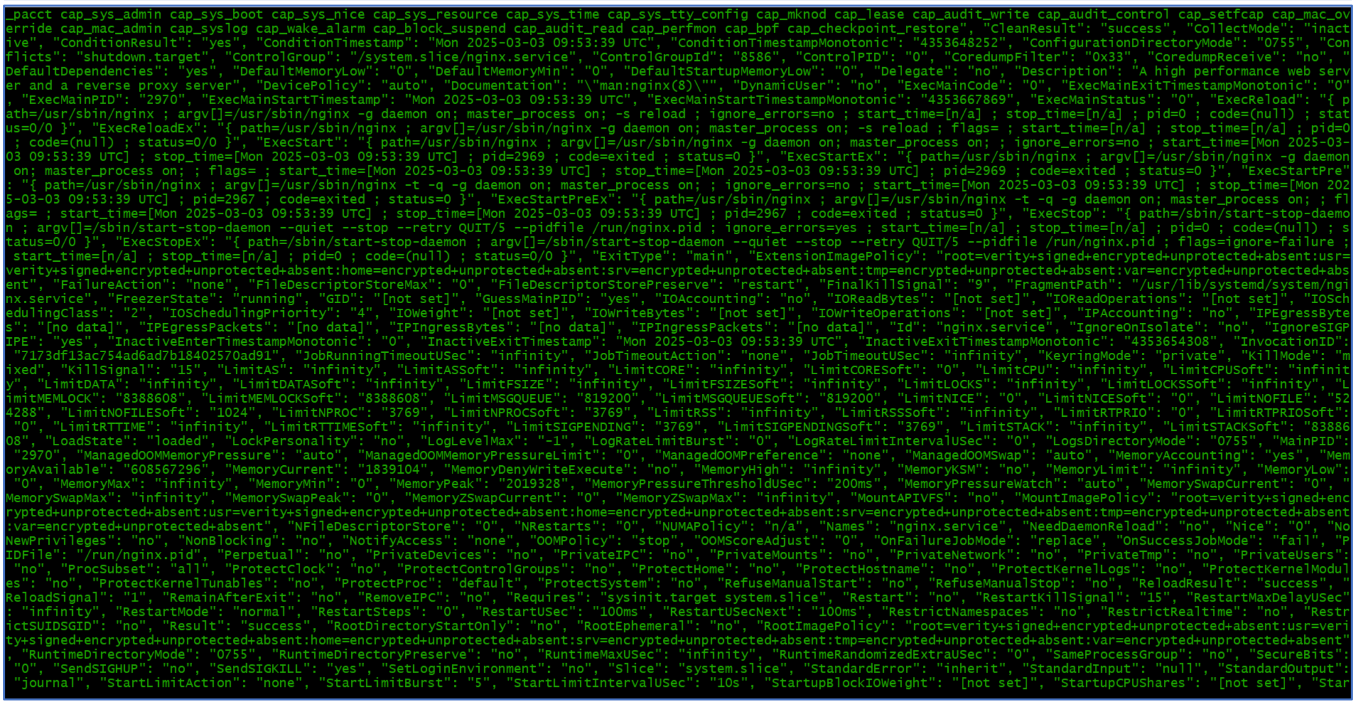
```
ubuntu@ip-172-31-90-40:~/playbooks$ ansible-playbook -v  install_nginx_play.yml
Using /etc/ansible/ansible.cfg as config file

PLAY [Install and Run Nginx] ***************************************************

TASK [Gathering Facts] *********************************************************
ok: [server_2]
ok: [server_1]

TASK [Install nginx] ***********************************************************
changed: [server_1] => {"cache_update_time": 1740992640, "cache_updated": false, "changed": true, "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBui
lding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n  nginx-common\nSuggested packages:\n  fcgiwrap nginx-doc ssl-
cert\nThe following NEW packages will be installed:\n  nginx nginx-common\n0 upgraded, 2 newly installed, 0 to remove and 134 not upgraded.\nNeed to get 552 kB of archives.
\nAfter this operation, 1596 kB of additional disk space will be used.\nGet:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.
0-2ubuntu7.1 [31.2 kB]\nGet:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]\nPreconfiguring packages ...\nF
etched 552 kB in 0s (20.9 MB/s)\nSelecting previously unselected package nginx-common.\r\n(Reading database ... \r(Reading database ... 5%\r(Reading database ... 10%\r(Read
ing database ... 15%\r(Reading database ... 20%\r(Reading database ... 25%\r(Reading database ... 30%\r(Reading database ... 35%\r(Reading database ... 40%\r(Reading databa
se ... 45%\r(Reading database ... 50%\r(Reading database ... 55%\r(Reading database ... 60%\r(Reading database ... 65%\r(Reading database ... 70%\r(Reading database ... 75%
\r(Reading database ... 80%\r(Reading database ... 85%\r(Reading database ... 90%\r(Reading database ... 95%\r(Reading database ... 100%\r(Reading database ... 70610 files 
and directories currently installed.)\r\nPreparing to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...\r\nUnpacking nginx-common (1.24.0-2ubuntu7.1) ...\r\nSelecting p
reviously unselected package nginx.\r\nPreparing to unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...\r\nUnpacking nginx (1.24.0-2ubuntu7.1) ...\r\nSetting up nginx (1.24.0-
2ubuntu7.1) ...\r\nWarning: The unit file, source configuration file or drop-ins of nginx.service changed on disk. Run 'systemctl daemon-reload' to reload units.\r\r\nSetti
ng up nginx-common (1.24.0-2ubuntu7.1) ...\r\nCreated symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.\r\r\nProces
sing triggers for ufw (0.36.2-6) ...\r\nProcessing triggers for man-db (2.12.0-4build2) ...\r\n\nRunning kernel seems to be up-to-date.\n\nNo services need to be restarted.
\n\nNo containers need to be restarted.\n\nNo user sessions are running outdated binaries.\n\nNo VM guests are running outdated hypervisor (qemu) binaries on this host.\n",
 "stdout_lines": ["Reading package lists...", "Building dependency tree...", "Reading state information...", "The following additional packages will be installed:", "  ngin
x-common", "Suggested packages:", "  fcgiwrap nginx-doc ssl-cert", "The following NEW packages will be installed:", "  nginx nginx-common", "0 upgraded, 2 newly installed, 
0 to remove and 134 not upgraded.", "Need to get 552 kB of archives.", "After this operation, 1596 kB of additional disk space will be used.", "Get:1 http://us-east-1.ec2.a
rchive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]", "Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main am
d64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]", "Preconfiguring packages ...", "Fetched 552 kB in 0s (20.9 MB/s)", "Selecting previously unselected package nginx-common.", "(R
eading database ... ", "(Reading database ... 5%", "(Reading database ... 10%", "(Reading database ... 15%", "(Reading database ... 20%", "(Reading database ... 25%", "(Rea
ding database ... 30%", "(Reading database ... 35%", "(Reading database ... 40%", "(Reading database ... 45%", "(Reading database ... 50%", "(Reading database ... 55%", "(R
eading database ... 60%", "(Reading database ... 65%", "(Reading database ... 70%", "(Reading database ... 75%", "(Reading database ... 80%", "(Reading database ... 85%", "
(Reading database ... 90%", "(Reading database ... 95%", "(Reading database ... 100%", "(Reading database ... 70610 files and directories currently installed.)", "Preparing
 to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...", "Unpacking nginx-common (1.24.0-2ubuntu7.1) ...", "Selecting previously unselected package nginx.", "Preparing t
o unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...", "Unpacking nginx (1.24.0-2ubuntu7.1) ...", "Setting up nginx (1.24.0-2ubuntu7.1) ...", "Warning: The unit file, source 
configuration file or drop-ins of nginx.service changed on disk. Run 'systemctl daemon-reload' to reload units.", "", "Setting up nginx-common (1.24.0-2ubuntu7.1) ...", "Cr
eated symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.", "", "Processing triggers for ufw (0.36.2-6) ...", "Proces
sing triggers for man-db (2.12.0-4build2) ...", "", "Running kernel seems to be up-to-date.", "", "No services need to be restarted.", "", "No containers need to be restart
ed.", "", "No user sessions are running outdated binaries.", "", "No VM guests are running outdated hypervisor (qemu) binaries on this host."]}
changed: [server_2] => {"cache_update_time": 1740992640, "cache_updated": false, "changed": true, "stderr": "", "stderr_lines": [], "stdout": "Reading package lists...\nBui
lding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n  nginx-common\nSuggested packages:\n  fcgiwrap nginx-doc ssl-
cert\nThe following NEW packages will be installed:\n  nginx nginx-common\n0 upgraded, 2 newly installed, 0 to remove and 134 not upgraded.\nNeed to get 552 kB of archives.
\nAfter this operation, 1596 kB of additional disk space will be used.\nGet:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.
0-2ubuntu7.1 [31.2 kB]\nGet:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]\nPreconfiguring packages ...\nF
etched 552 kB in 0s (23.8 MB/s)\nSelecting previously unselected package nginx-common.\r\n(Reading database ... \r(Reading database ... 5%\r(Reading database ... 10%\r(Read
ing database ... 15%\r(Reading database ... 20%\r(Reading database ... 25%\r(Reading database ... 30%\r(Reading database ... 35%\r(Reading database ... 40%\r(Reading databa
se ... 45%\r(Reading database ... 50%\r(Reading database ... 55%\r(Reading database ... 60%\r(Reading database ... 65%\r(Reading database ... 70%\r(Reading database ... 75%
```



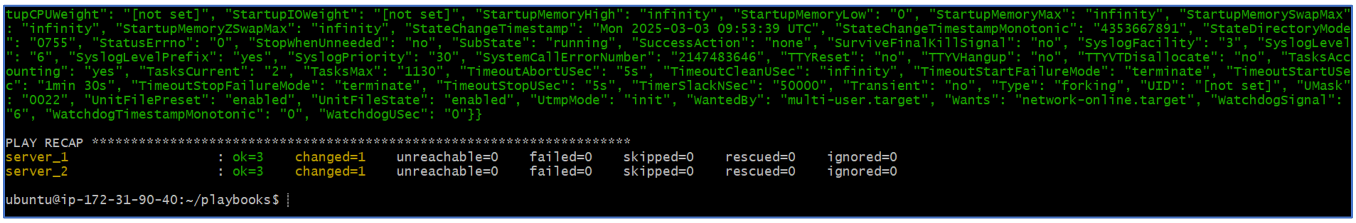
```
\r(Reading database ... 80%\r(Reading database ... 85%\r(Reading database ... 90%\r(Reading database ... 95%\r(Reading database ... 100%\r(Reading database ... 70610 files 
and directories currently installed.)\r\nPreparing to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...\r\nUnpacking nginx-common (1.24.0-2ubuntu7.1) ...\r\nSelecting p
reviously unselected package nginx.\r\nPreparing to unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...\r\nUnpacking nginx (1.24.0-2ubuntu7.1) ...\r\nSetting up nginx (1.24.0-
2ubuntu7.1) ...\r\nWarning: The unit file, source configuration file or drop-ins of nginx.service changed on disk. Run 'systemctl daemon-reload' to reload units.\r\r\nSetti
ng up nginx-common (1.24.0-2ubuntu7.1) ...\r\nCreated symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.\r\r\nProces
sing triggers for ufw (0.36.2-6) ...\r\nProcessing triggers for man-db (2.12.0-4build2) ...\r\n\nRunning kernel seems to be up-to-date.\n\nNo services need to be restarted.
\n\nNo containers need to be restarted.\n\nNo user sessions are running outdated binaries.\n\nNo VM guests are running outdated hypervisor (qemu) binaries on this host.\n",
 "stdout_lines": ["Reading package lists...", "Building dependency tree...", "Reading state information...", "The following additional packages will be installed:", "  ngin
x-common", "Suggested packages:", "  fcgiwrap nginx-doc ssl-cert", "The following NEW packages will be installed:", "  nginx nginx-common", "0 upgraded, 2 newly installed, 
0 to remove and 134 not upgraded.", "Need to get 552 kB of archives.", "After this operation, 1596 kB of additional disk space will be used.", "Get:1 http://us-east-1.ec2.a
rchive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]", "Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main am
d64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]", "Preconfiguring packages ...", "Fetched 552 kB in 0s (23.8 MB/s)", "Selecting previously unselected package nginx-common.", "(R
eading database ... ", "(Reading database ... 5%", "(Reading database ... 10%", "(Reading database ... 15%", "(Reading database ... 20%", "(Reading database ... 25%", "(Rea
ding database ... 30%", "(Reading database ... 35%", "(Reading database ... 40%", "(Reading database ... 45%", "(Reading database ... 50%", "(Reading database ... 55%", "(R
eading database ... 60%", "(Reading database ... 65%", "(Reading database ... 70%", "(Reading database ... 75%", "(Reading database ... 80%", "(Reading database ... 85%", "
(Reading database ... 90%", "(Reading database ... 95%", "(Reading database ... 100%", "(Reading database ... 70610 files and directories currently installed.)", "Preparing
 to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...", "Unpacking nginx-common (1.24.0-2ubuntu7.1) ...", "Selecting previously unselected package nginx.", "Preparing t
o unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...", "Unpacking nginx (1.24.0-2ubuntu7.1) ...", "Setting up nginx (1.24.0-2ubuntu7.1) ...", "Warning: The unit file, source 
configuration file or drop-ins of nginx.service changed on disk. Run 'systemctl daemon-reload' to reload units.", "", "Setting up nginx-common (1.24.0-2ubuntu7.1) ...", "Cr
eated symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.", "", "Processing triggers for ufw (0.36.2-6) ...", "Proces
sing triggers for man-db (2.12.0-4build2) ...", "", "Running kernel seems to be up-to-date.", "", "No services need to be restarted.", "", "No containers need to be restart
ed.", "", "No user sessions are running outdated binaries.", "", "No VM guests are running outdated hypervisor (qemu) binaries on this host."]}

TASK [Start nginx] *************************************************************
ok: [server_1] => {"changed": false, "enabled": true, "name": "nginx", "state": "started", "status": {"ActiveEnterTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ActiveEnterTim
estampMonotonic": "4354622210", "ActiveExitTimestampMonotonic": "0", "ActiveState": "active", "After": "system.slice remote-fs.target nss-lookup.target basic.target network
-online.target systemd-journald.socket sysinit.target", "AllowIsolate": "no", "AssertResult": "yes", "AssertTimestamp": "Mon 2025-03-03 09:53:39 UTC", "AssertTimestampMonot
onic": "4354601655", "Before": "multi-user.target shutdown.target", "BlockIOAccounting": "no", "BlockIOWeight": "[not set]", "CPUAccounting": "yes", "CPUAffinityFromNUMA":
"no", "CPUQuotaPerSecUSec": "infinity", "CPUQuotaPeriodUSec": "infinity", "CPUSchedulingPolicy": "0", "CPUSchedulingPriority": "0", "CPUSchedulingResetOnFork": "no", "CPUSh
ares": "[not set]", "CPUUsageNSec": "10842000", "CPUWeight": "[not set]", "CacheDirectoryMode": "0755", "CanFreeze": "yes", "CanIsolate": "no", "CanReload": "yes", "CanStar
t": "yes", "CanStop": "yes", "CapabilityBoundingSet": "cap_chown cap_dac_override cap_dac_read_search cap_fowner cap_fsetid cap_kill cap_setgid cap_setuid cap_setpcap cap_l
inux_immutable cap_net_bind_service cap_net_broadcast cap_net_admin cap_net_raw cap_ipc_lock cap_ipc_owner cap_sys_module cap_sys_rawio cap_sys_chroot cap_sys_ptrace cap_sy
s_pacct cap_sys_admin cap_sys_boot cap_sys_nice cap_sys_resource cap_sys_time cap_sys_tty_config cap_mknod cap_lease cap_audit_write cap_audit_control cap_setfcap cap_mac_o
verride cap_mac_admin cap_syslog cap_wake_alarm cap_block_suspend cap_audit_read cap_perfmon cap_bpf cap_checkpoint_restore", "CleanResult": "success", "CollectMode": "inac
tive", "ConditionResult": "yes", "ConditionTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ConditionTimestampMonotonic": "4354601652", "ConfigurationDirectoryMode": "0755", "Co
nflicts": "shutdown.target", "ControlGroup": "/system.slice/nginx.service", "ControlGroupId": "9118", "ControlPID": "0", "CoredumpFilter": "0x33", "CoredumpReceive": "no",
"DefaultDependencies": "yes", "DefaultMemoryLow": "0", "DefaultMemoryMin": "0", "DefaultStartupMemoryLow": "0", "Delegate": "no", "Description": "A high performance web ser
ver and a reverse proxy server", "DevicePolicy": "auto", "Documentation": "\"man:nginx(8)\"", "DynamicUser": "no", "ExecMainCode": "0", "ExecMainExitTimestampMonotonic": "0
", "ExecMainPID": "3044", "ExecMainStartTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ExecMainStartTimestampMonotonic": "4354622187", "ExecMainStatus": "0", "ExecReload": "{
path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; -s reload ; ignore_errors=no ; start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; sta
tus=0/0 }", "ExecReloadEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; -s reload ; flags= ; start_time=[n/a] ; stop_time=[n/a] ; pid=
0 ; code=(null) ; status=0/0 }", "ExecStart": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; ; ignore_errors=no ; start_time=[Mon 2025-03
-03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=3043 ; code=exited ; status=0 }", "ExecStartEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemo
n on; master_process on; ; flags= ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=3043 ; code=exited ; status=0 }", "ExecStartPre
": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -t -q -g daemon on; master_process on; ; ignore_errors=no ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 20
25-03-03 09:53:39 UTC] ; pid=3041 ; code=exited ; status=0 }", "ExecStartPreEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -t -q -g daemon on; master_process on; ; f
lags= ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=3041 ; code=exited ; status=0 }", "ExecStop": "{ path=/sbin/start-stop-daem
on ; argv[]=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid ; ignore_errors=yes ; start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ;
```



```
status=0/0 }", "ExecStopEx": "{ path=/sbin/start-stop-daemon ; argv[]=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid ; flags=ignore-failure
; start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }", "ExitType": "main", "ExtensionImagePolicy": "root=verity+signed+encrypted+unprotected+absent:usr
=verity+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absent:var=encrypted+unprotected+ab
sent", "FailureAction": "none", "FileDescriptorStoreMax": "0", "FileDescriptorStorePreserve": "restart", "FinalKillSignal": "9", "FragmentPath": "/usr/lib/systemd/system/ng
inx.service", "FreezerState": "running", "GID": "[not set]", "GuessMainPID": "yes", "IOAccounting": "no", "IOReadBytes": "[not set]", "IOReadOperations": "[not set]", "IOSc
hedulingClass": "2", "IOSchedulingPriority": "4", "IOWeight": "[not set]", "IOWriteBytes": "[not set]", "IOWriteOperations": "[not set]", "IPAccounting": "no", "IPEgressByt
es": "[no data]", "IPEgressPackets": "[no data]", "IPIngressBytes": "[no data]", "IPIngressPackets": "[no data]", "Id": "nginx.service", "IgnoreOnIsolate": "no", "IgnoreSIG
PIPE": "yes", "InactiveEnterTimestampMonotonic": "0", "InactiveExitTimestamp": "Mon 2025-03-03 09:53:39 UTC", "InactiveExitTimestampMonotonic": "4354607049", "InvocationID"
: "e5d2b997285247d2a7facabe78613122", "JobRunningTimeoutUSec": "infinity", "JobTimeoutAction": "none", "JobTimeoutUSec": "infinity", "KeyringMode": "private", "KillMode": "
mixed", "KillSignal": "15", "LimitAS": "infinity", "LimitASSoft": "infinity", "LimitCORE": "infinity", "LimitCORESoft": "0", "LimitCPU": "infinity", "LimitCPUSoft": "infini
ty", "LimitDATA": "infinity", "LimitDATASoft": "infinity", "LimitFSIZE": "infinity", "LimitFSIZESoft": "infinity", "LimitLOCKS": "infinity", "LimitLOCKSSoft": "infinity", "
LimitMEMLOCK": "8388608", "LimitMEMLOCKSoft": "8388608", "LimitMSGQUEUE": "819200", "LimitMSGQUEUESoft": "819200", "LimitNICE": "0", "LimitNICESoft": "0", "LimitNOFILE": "5
24288", "LimitNOFILESoft": "1024", "LimitNPROC": "3769", "LimitNPROCSoft": "3769", "LimitRSS": "infinity", "LimitRSSSoft": "infinity", "LimitRTPRIO": "0", "LimitRTPRIOSoft"
: "0", "LimitRTTIME": "infinity", "LimitRTTIMESoft": "infinity", "LimitSIGPENDING": "3769", "LimitSIGPENDINGSoft": "3769", "LimitSTACK": "infinity", "LimitSTACKSoft": "8388
608", "LoadState": "loaded", "LockPersonality": "no", "LogLevelMax": "-1", "LogRateLimitBurst": "0", "LogRateLimitIntervalUSec": "0", "LogsDirectoryMode": "0755", "MainPID"
: "3044", "ManagedOOMMemoryPressure": "auto", "ManagedOOMMemoryPressureLimit": "0", "ManagedOOMPreference": "none", "ManagedOOMSwap": "auto", "MemoryAccounting": "yes", "Me
moryAvailable": "622714880", "MemoryCurrent": "1839104", "MemoryDenyWriteExecute": "no", "MemoryHigh": "infinity", "MemoryKSM": "no", "MemoryLimit": "infinity", "MemoryLow"
: "0", "MemoryMax": "infinity", "MemoryMin": "0", "MemoryPeak": "2023424", "MemoryPressureThresholdUSec": "200ms", "MemoryPressureWatch": "auto", "MemorySwapCurrent": "0",
"MemorySwapMax": "infinity", "MemorySwapPeak": "0", "MemoryZSwapCurrent": "0", "MemoryZSwapMax": "infinity", "MountAPIVFS": "no", "MountImagePolicy": "root=verity+signed+en
crypted+unprotected+absent:usr=verity+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absen
t:var=encrypted+unprotected+absent", "NFileDescriptorStore": "0", "NRestarts": "0", "NUMAPolicy": "n/a", "Names": "nginx.service", "NeedDaemonReload": "no", "Nice": "0", "N
oNewPrivileges": "no", "NonBlocking": "no", "NotifyAccess": "none", "OOMPolicy": "stop", "OOMScoreAdjust": "0", "OnFailureJobMode": "replace", "OnSuccessJobMode": "fail", "
PIDFile": "/run/nginx.pid", "Perpetual": "no", "PrivateDevices": "no", "PrivateIPC": "no", "PrivateMounts": "no", "PrivateNetwork": "no", "PrivateTmp": "no", "PrivateUsers"
: "no", "ProcSubset": "all", "ProtectClock": "no", "ProtectControlGroups": "no", "ProtectHome": "no", "ProtectHostname": "no", "ProtectKernelLogs": "no", "ProtectKernelModu
les": "no", "ProtectKernelTunables": "no", "ProtectProc": "default", "ProtectSystem": "no", "RefuseManualStart": "no", "RefuseManualStop": "no", "ReloadResult": "success",
"ReloadSignal": "1", "RemainAfterExit": "no", "RemoveIPC": "no", "Requires": "system.slice sysinit.target", "Restart": "no", "RestartKillSignal": "15", "RestartMaxDelayUSec
": "infinity", "RestartMode": "normal", "RestartSteps": "0", "RestartUSec": "100ms", "RestartUSecNext": "100ms", "RestrictNamespaces": "no", "RestrictRealtime": "no", "Rest
rictSUIDSGID": "no", "Result": "success", "RootDirectoryStartOnly": "no", "RootEphemeral": "no", "RootImagePolicy": "root=verity+signed+encrypted+unprotected+absent:usr=ver
ity+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absent:var=encrypted+unprotected+absent
", "RuntimeDirectoryMode": "0755", "RuntimeDirectoryPreserve": "no", "RuntimeMaxUSec": "infinity", "RuntimeRandomizedExtraUSec": "0", "SameProcessGroup": "no", "SecureBits
": "0", "SendSIGHUP": "no", "SendSIGKILL": "yes", "SetLoginEnvironment": "no", "Slice": "system.slice", "StandardError": "inherit", "StandardInput": "null", "StandardOutput
": "journal", "StartLimitAction": "none", "StartLimitBurst": "5", "StartLimitIntervalUSec": "10s", "StartupBlockIOWeight": "[not set]", "StartupCPUShares": "[not set]", "Sta
rtupCPUWeight": "[not set]", "StartupIOWeight": "[not set]", "StartupMemoryHigh": "infinity", "StartupMemoryLow": "0", "StartupMemoryMax": "infinity", "StartupMemorySwapMax
": "infinity", "StartupMemoryZSwapMax": "infinity", "StateChangeTimestamp": "Mon 2025-03-03 09:53:39 UTC", "StateChangeTimestampMonotonic": "4354622210", "StateDirectoryMod
e": "0755", "StatusErrno": "0", "StopWhenUnneeded": "no", "SubState": "running", "SuccessAction": "none", "SurviveFinalKillSignal": "no", "SyslogFacility": "3", "SyslogLeve
l": "6", "SyslogLevelPrefix": "yes", "SyslogPriority": "30", "SystemCallErrorNumber": "2147483646", "TTYReset": "no", "TTYVHangup": "no", "TTYVTDisallocate": "no", "TasksAc
counting": "yes", "TasksCurrent": "2", "TasksMax": "1130", "TimeoutAbortUSec": "5s", "TimeoutCleanUSec": "infinity", "TimeoutStartFailureMode": "terminate", "TimeoutStartUS
ec": "1min 30s", "TimeoutStopFailureMode": "terminate", "TimeoutStopUSec": "5s", "TimerSlackNSec": "50000", "Transient": "no", "Type": "forking", "UID": "[not set]", "UMask
": "0022", "UnitFilePreset": "enabled", "UnitFileState": "enabled", "UtmpMode": "init", "WantedBy": "multi-user.target", "Wants": "network-online.target", "WatchdogSignal":
 "6", "WatchdogTimestampMonotonic": "0", "WatchdogUSec": "0"}}
ok: [server_2] => {"changed": false, "enabled": true, "name": "nginx", "state": "started", "status": {"ActiveEnterTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ActiveEnterTim
estampMonotonic": "4353667891", "ActiveExitTimestampMonotonic": "0", "ActiveState": "active", "After": "network-online.target remote-fs.target nss-lookup.target systemd-jou
rnald.socket system.slice sysinit.target basic.target", "AllowIsolate": "no", "AssertResult": "yes", "AssertTimestamp": "Mon 2025-03-03 09:53:39 UTC", "AssertTimestampMonot
onic": "4353648255", "Before": "shutdown.target multi-user.target", "BlockIOAccounting": "no", "BlockIOWeight": "[not set]", "CPUAccounting": "yes", "CPUAffinityFromNUMA":
"no", "CPUQuotaPerSecUSec": "infinity", "CPUQuotaPeriodUSec": "infinity", "CPUSchedulingPolicy": "0", "CPUSchedulingPriority": "0", "CPUSchedulingResetOnFork": "no", "CPUSh
ares": "[not set]", "CPUUsageNSec": "9590000", "CPUWeight": "[not set]", "CacheDirectoryMode": "0755", "CanFreeze": "yes", "CanIsolate": "no", "CanReload": "yes", "CanStart
": "yes", "CanStop": "yes", "CapabilityBoundingSet": "cap_chown cap_dac_override cap_dac_read_search cap_fowner cap_fsetid cap_kill cap_setgid cap_setuid cap_setpcap cap_li
nux_immutable cap_net_bind_service cap_net_broadcast cap_net_admin cap_net_raw cap_ipc_lock cap_ipc_owner cap_sys_module cap_sys_rawio cap_sys_chroot cap_sys_ptrace cap_sys
```



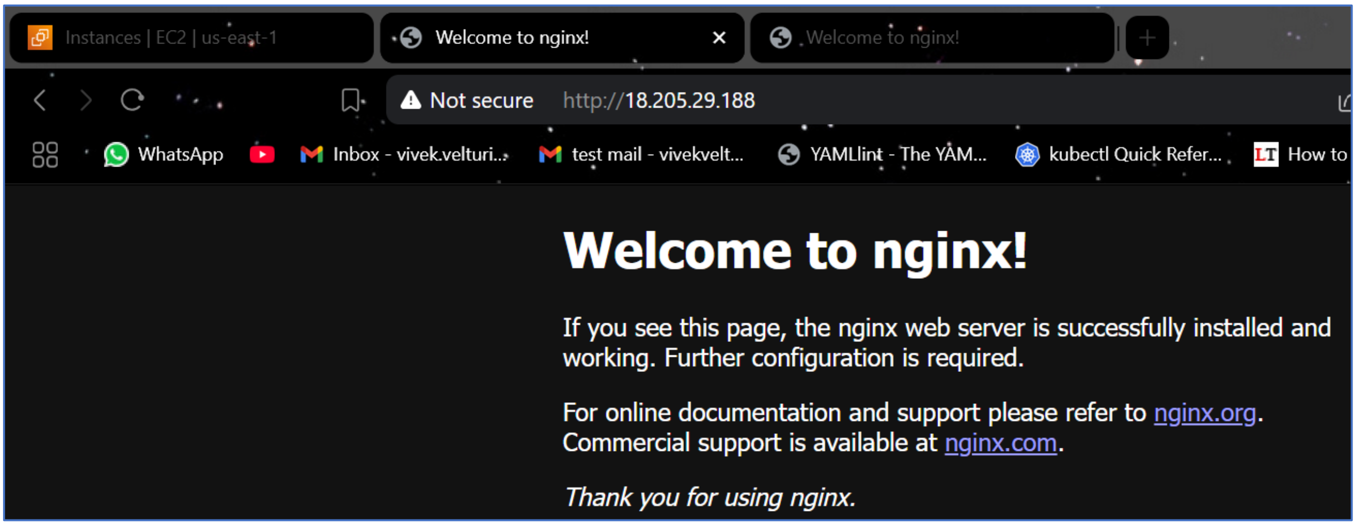
```
_pacct cap_sys_admin cap_sys_boot cap_sys_nice cap_sys_resource cap_sys_time cap_sys_tty_config cap_mknod cap_lease cap_audit_write cap_audit_control cap_setfcap cap_mac_ov
erride cap_mac_admin cap_syslog cap_wake_alarm cap_block_suspend cap_audit_read cap_perfmon cap_bpf cap_checkpoint_restore", "CleanResult": "success", "CollectMode": "inact
ive", "ConditionResult": "yes", "ConditionTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ConditionTimestampMonotonic": "4353648252", "ConfigurationDirectoryMode": "0755", "Con
flicts": "shutdown.target", "ControlGroup": "/system.slice/nginx.service", "ControlGroupId": "8586", "ControlPID": "0", "CoredumpFilter": "0x33", "CoredumpReceive": "no", "
DefaultDependencies": "yes", "DefaultMemoryLow": "0", "DefaultMemoryMin": "0", "DefaultStartupMemoryLow": "0", "Delegate": "no", "Description": "A high performance web serv
er and a reverse proxy server", "DevicePolicy": "auto", "Documentation": "\"man:nginx(8)\"", "DynamicUser": "no", "ExecMainCode": "0", "ExecMainExitTimestampMonotonic": "0"
, "ExecMainPID": "2970", "ExecMainStartTimestamp": "Mon 2025-03-03 09:53:39 UTC", "ExecMainStartTimestampMonotonic": "4353667869", "ExecMainStatus": "0", "ExecReload": "{ p
ath=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; -s reload ; ignore_errors=no ; start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; stat
us=0/0 }", "ExecReloadEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; -s reload ; flags= ; start_time=[n/a] ; stop_time=[n/a] ; pid=0
 ; code=(null) ; status=0/0 }", "ExecStart": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon on; master_process on; ; ignore_errors=no ; start_time=[Mon 2025-03-
03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=2969 ; code=exited ; status=0 }", "ExecStartEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -g daemon
 on; master_process on; ; flags= ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=2969 ; code=exited ; status=0 }", "ExecStartPre"
: "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -t -q -g daemon on; master_process on; ; ignore_errors=no ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 202
5-03-03 09:53:39 UTC] ; pid=2967 ; code=exited ; status=0 }", "ExecStartPreEx": "{ path=/usr/sbin/nginx ; argv[]=/usr/sbin/nginx -t -q -g daemon on; master_process on; ; fl
ags= ; start_time=[Mon 2025-03-03 09:53:39 UTC] ; stop_time=[Mon 2025-03-03 09:53:39 UTC] ; pid=2967 ; code=exited ; status=0 }", "ExecStop": "{ path=/sbin/start-stop-daemo
n ; argv[]=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid ; ignore_errors=yes ; start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; s
tatus=0/0 }", "ExecStopEx": "{ path=/sbin/start-stop-daemon ; argv[]=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid ; flags=ignore-failure ;
 start_time=[n/a] ; stop_time=[n/a] ; pid=0 ; code=(null) ; status=0/0 }", "ExitType": "main", "ExtensionImagePolicy": "root=verity+signed+encrypted+unprotected+absent:usr=
verity+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absent:var=encrypted+unprotected+abs
ent", "FailureAction": "none", "FileDescriptorStoreMax": "0", "FileDescriptorStorePreserve": "restart", "FinalKillSignal": "9", "FragmentPath": "/usr/lib/systemd/system/ngi
nx.service", "FreezerState": "running", "GID": "[not set]", "GuessMainPID": "yes", "IOAccounting": "no", "IOReadBytes": "[not set]", "IOReadOperations": "[not set]", "IOSch
edulingClass": "2", "IOSchedulingPriority": "4", "IOWeight": "[not set]", "IOWriteBytes": "[not set]", "IOWriteOperations": "[not set]", "IPAccounting": "no", "IPEgressByte
s": "[no data]", "IPEgressPackets": "[no data]", "IPIngressBytes": "[no data]", "IPIngressPackets": "[no data]", "Id": "nginx.service", "IgnoreOnIsolate": "no", "IgnoreSIGP
IPE": "yes", "InactiveEnterTimestampMonotonic": "0", "InactiveExitTimestamp": "Mon 2025-03-03 09:53:39 UTC", "InactiveExitTimestampMonotonic": "4353654308", "InvocationID":
 "7173df13ac754ad6ad7b18402570ad91", "JobRunningTimeoutUSec": "infinity", "JobTimeoutAction": "none", "JobTimeoutUSec": "infinity", "KeyringMode": "private", "KillMode": "m
ixed", "KillSignal": "15", "LimitAS": "infinity", "LimitASSoft": "infinity", "LimitCORE": "infinity", "LimitCORESoft": "0", "LimitCPU": "infinity", "LimitCPUSoft": "infinit
y", "LimitDATA": "infinity", "LimitDATASoft": "infinity", "LimitFSIZE": "infinity", "LimitFSIZESoft": "infinity", "LimitLOCKS": "infinity", "LimitLOCKSSoft": "infinity", "L
imitMEMLOCK": "8388608", "LimitMEMLOCKSoft": "8388608", "LimitMSGQUEUE": "819200", "LimitMSGQUEUESoft": "819200", "LimitNICE": "0", "LimitNICESoft": "0", "LimitNOFILE": "52
4288", "LimitNOFILESoft": "1024", "LimitNPROC": "3769", "LimitNPROCSoft": "3769", "LimitRSS": "infinity", "LimitRSSSoft": "infinity", "LimitRTPRIO": "0", "LimitRTPRIOSoft":
 "0", "LimitRTTIME": "infinity", "LimitRTTIMESoft": "infinity", "LimitSIGPENDING": "3769", "LimitSIGPENDINGSoft": "3769", "LimitSTACK": "infinity", "LimitSTACKSoft": "83886
08", "LoadState": "loaded", "LockPersonality": "no", "LogLevelMax": "-1", "LogRateLimitBurst": "0", "LogRateLimitIntervalUSec": "0", "LogsDirectoryMode": "0755", "MainPID":
 "2970", "ManagedOOMMemoryPressure": "auto", "ManagedOOMMemoryPressureLimit": "0", "ManagedOOMPreference": "none", "ManagedOOMSwap": "auto", "MemoryAccounting": "yes", "Mem
oryAvailable": "608567296", "MemoryCurrent": "1839104", "MemoryDenyWriteExecute": "no", "MemoryHigh": "infinity", "MemoryKSM": "no", "MemoryLimit": "infinity", "MemoryLow":
 "0", "MemoryMax": "infinity", "MemoryMin": "0", "MemoryPeak": "2019328", "MemoryPressureThresholdUSec": "200ms", "MemoryPressureWatch": "auto", "MemorySwapCurrent": "0", "
MemorySwapMax": "infinity", "MemorySwapPeak": "0", "MemoryZSwapCurrent": "0", "MemoryZSwapMax": "infinity", "MountAPIVFS": "no", "MountImagePolicy": "root=verity+signed+enc
rypted+unprotected+absent:usr=verity+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absent
:var=encrypted+unprotected+absent", "NFileDescriptorStore": "0", "NRestarts": "0", "NUMAPolicy": "n/a", "Names": "nginx.service", "NeedDaemonReload": "no", "Nice": "0", "No
NewPrivileges": "no", "NonBlocking": "no", "NotifyAccess": "none", "OOMPolicy": "stop", "OOMScoreAdjust": "0", "OnFailureJobMode": "replace", "OnSuccessJobMode": "fail", "P
IDFile": "/run/nginx.pid", "Perpetual": "no", "PrivateDevices": "no", "PrivateIPC": "no", "PrivateMounts": "no", "PrivateNetwork": "no", "PrivateTmp": "no", "PrivateUsers":
 "no", "ProcSubset": "all", "ProtectClock": "no", "ProtectControlGroups": "no", "ProtectHome": "no", "ProtectHostname": "no", "ProtectKernelLogs": "no", "ProtectKernelModul
es": "no", "ProtectKernelTunables": "no", "ProtectProc": "default", "ProtectSystem": "no", "RefuseManualStart": "no", "RefuseManualStop": "no", "ReloadResult": "success", "
ReloadSignal": "1", "RemainAfterExit": "no", "RemoveIPC": "no", "Requires": "sysinit.target system.slice", "Restart": "no", "RestartKillSignal": "15", "RestartMaxDelayUSec"
: "infinity", "RestartMode": "normal", "RestartSteps": "0", "RestartUSec": "100ms", "RestartUSecNext": "100ms", "RestrictNamespaces": "no", "RestrictRealtime": "no", "Restr
ictSUIDSGID": "no", "Result": "success", "RootDirectoryStartOnly": "no", "RootEphemeral": "no", "RootImagePolicy": "root=verity+signed+encrypted+unprotected+absent:usr=veri
ty+signed+encrypted+unprotected+absent:home=encrypted+unprotected+absent:srv=encrypted+unprotected+absent:tmp=encrypted+unprotected+absent:var=encrypted+unprotected+absent"
, "RuntimeDirectoryMode": "0755", "RuntimeDirectoryPreserve": "no", "RuntimeMaxUSec": "infinity", "RuntimeRandomizedExtraUSec": "0", "SameProcessGroup": "no", "SecureBits":
 "0", "SendSIGHUP": "no", "SendSIGKILL": "yes", "SetLoginEnvironment": "no", "Slice": "system.slice", "StandardError": "inherit", "StandardInput": "null", "StandardOutput":
 "journal", "StartLimitAction": "none", "StartLimitBurst": "5", "StartLimitIntervalUSec": "10s", "StartupBlockIOWeight": "[not set]", "StartupCPUShares": "[not set]", "Star
```



```
tupCPUWeight": "[not set]", "StartupIOWeight": "[not set]", "StartupMemoryHigh": "infinity", "StartupMemoryLow": "0", "StartupMemoryMax": "infinity", "StartupMemorySwapMax"
: "infinity", "StartupMemoryZSwapMax": "infinity", "StateChangeTimestamp": "Mon 2025-03-03 09:53:39 UTC", "StateChangeTimestampMonotonic": "4353667891", "StateDirectoryMode
": "0755", "StatusErrno": "0", "StopWhenUnneeded": "no", "SubState": "running", "SuccessAction": "none", "SurviveFinalKillSignal": "no", "SyslogFacility": "3", "SyslogLevel
": "6", "SyslogLevelPrefix": "yes", "SyslogPriority": "30", "SystemCallErrorNumber": "2147483646", "TTYReset": "no", "TTYVHangup": "no", "TTYVTDisallocate": "no", "TasksAcc
ounting": "yes", "TasksCurrent": "2", "TasksMax": "1130", "TimeoutAbortUSec": "5s", "TimeoutCleanUSec": "infinity", "TimeoutStartFailureMode": "terminate", "TimeoutStartUSe
c": "1min 30s", "TimeoutStopFailureMode": "terminate", "TimeoutStopUSec": "5s", "TimerSlackNSec": "50000", "Transient": "no", "Type": "forking", "UID": "[not set]", "UMask"
: "0022", "UnitFilePreset": "enabled", "UnitFileState": "enabled", "UtmpMode": "init", "WantedBy": "multi-user.target", "Wants": "network-online.target", "WatchdogSignal":
"6", "WatchdogTimestampMonotonic": "0", "WatchdogUSec": "0"}}

PLAY RECAP *********************************************************************
server_1                   : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
server_2                   : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-90-40:~/playbooks$ 
```

This execution demonstrates Ansible's ability to automate software installation and service management, ensuring Nginx is consistently deployed and running across our managed servers.

- To confirm the successful installation and operation of Nginx on `server_1` and `server_2`, we accessed the public IP addresses of these servers via a web browser. Upon navigating to the respective IP addresses, we observed the default Nginx welcome page.

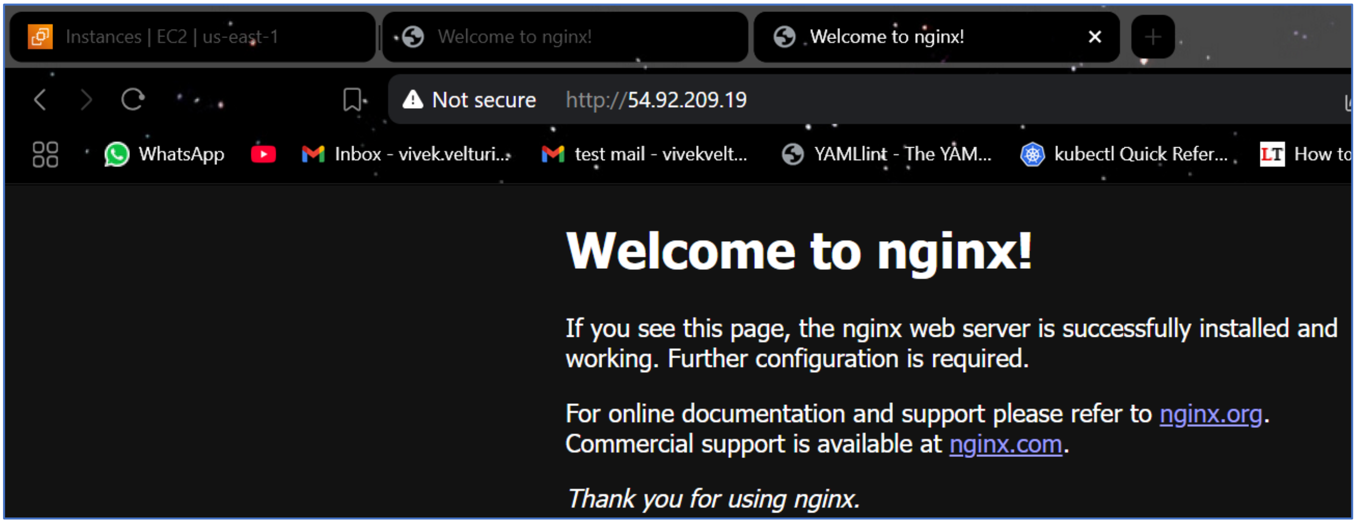


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

This page indicates that the Nginx web server is running and accessible, validating the successful execution of our Ansible playbook. This visual confirmation ensures that Nginx has been properly installed and started on both servers, fulfilling the intended outcome of our automation task.

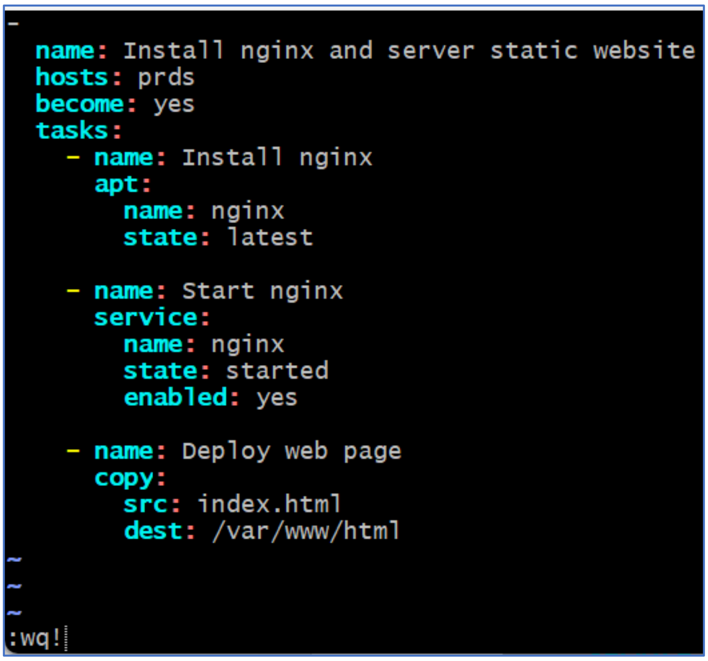
- To further demonstrate Ansible's capabilities, we created a new playbook named **`deploy_static_html_page_play.yml`**.
  This playbook targets the `prds` group, which currently contains **`server_3`**, and aims to install Nginx and deploy a static website.
  Using `sudo vim deploy_static_html_page_play.yml`,

```
-
  name: Install nginx and server static website
  hosts: prds
  become: yes
  tasks:
    - name: Install nginx
      apt:
        name: nginx
        state: latest

    - name: Start nginx
      service:
        name: nginx
        state: started
        enabled: yes

    - name: Deploy web page
      copy:
        src: index.html
        dest: /var/www/html
```

we defined the playbook with `become: yes` to allow commands to run with root privileges.

The playbook includes three tasks: 'Install nginx' to install the latest version of Nginx using the **`apt`** module, **'Start nginx'** to start the Nginx service using the **`service`** module, and **'Deploy web page'** to copy a local **`index.html`** file to the **`/var/www/html`** directory on the server using the **`copy`** module.

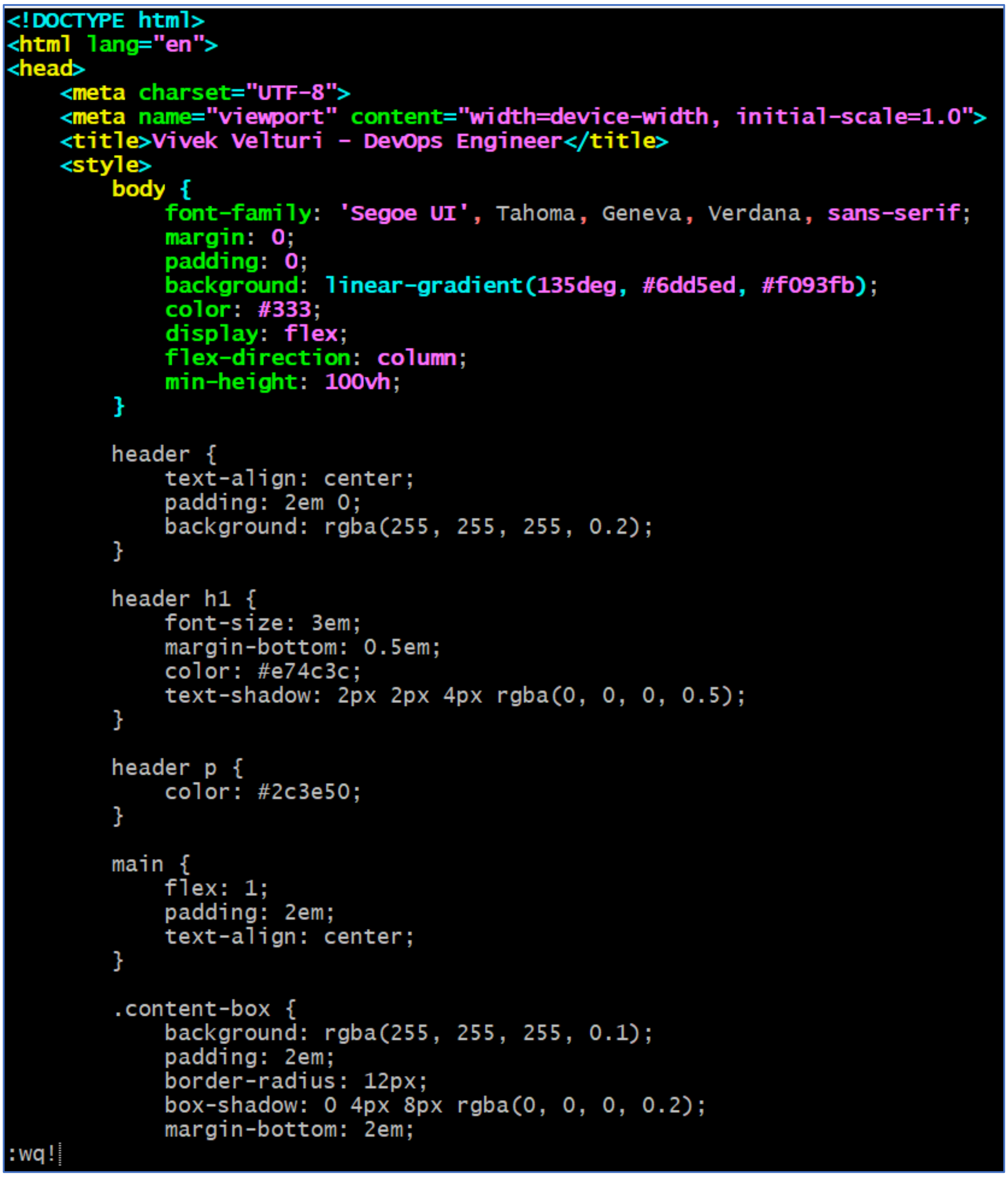
After defining these tasks, we saved and closed the `deploy_static_html_page_play.yml` file using the: `wq!` command in `vim`.



```
-
  name: Install nginx and server static website
  hosts: prds
  become: yes
  tasks:
    - name: Install nginx
      apt:
        name: nginx
        state: latest

    - name: Start nginx
      service:
        name: nginx
        state: started
        enabled: yes

    - name: Deploy web page
      copy:
        src: index.html
        dest: /var/www/html
~
~
~
:wq!
```

This playbook showcases Ansible's ability to handle a series of tasks, including software installation, service management, and file deployment, further illustrating its potential for automating complex deployments.

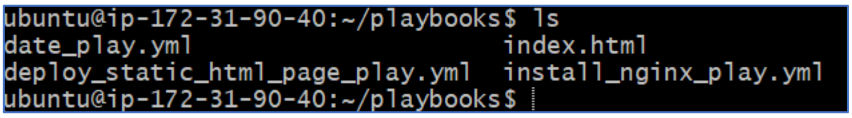
- To prepare for the deployment of our static website, **we created an `index.html` file** and populated it with the necessary HTML code.
  We used the command `sudo vim index.html` to open the file in `vim` with superuser privileges, allowing us to edit and save the file.
  Within this file, **we pasted the HTML code for our static web page**, ensuring it's ready for deployment to our target server.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vivek Velturi - DevOps Engineer</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            margin: 0;
            padding: 0;
            background: linear-gradient(135deg, #6dd5ed, #f093fb);
            color: #333;
            display: flex;
            flex-direction: column;
            min-height: 100vh;
        }

        header {
            text-align: center;
            padding: 2em 0;
            background: rgba(255, 255, 255, 0.2);
        }

        header h1 {
            font-size: 3em;
            margin-bottom: 0.5em;
            color: #e74c3c;
            text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
        }

        header p {
            color: #2c3e50;
        }

        main {
            flex: 1;
            padding: 2em;
            text-align: center;
        }

        .content-box {
            background: rgba(255, 255, 255, 0.1);
            padding: 2em;
            border-radius: 12px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            margin-bottom: 2em;
:wq!
```

This step sets the stage for the final deployment process, where Ansible will copy this `index.html` file to the web server's document root, making our static website accessible.

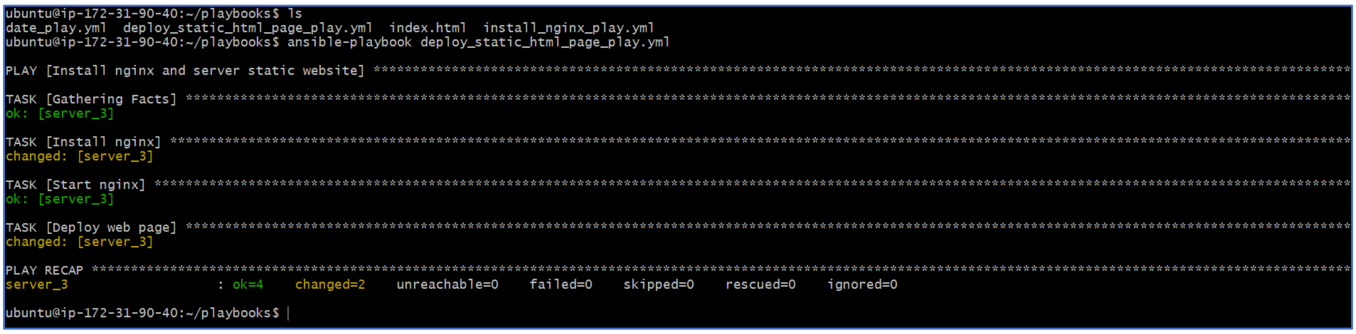
- After creating and populating the `index.html` file, we verified its presence in the `/home/ubuntu/playbooks` directory using the `ls` command.



```
ubuntu@ip-172-31-90-40:~/playbooks$ ls
date_play.yml                     index.html
deploy_static_html_page_play.yml  install_nginx_play.yml
ubuntu@ip-172-31-90-40:~/playbooks$
```

The output confirms that `index.html` is now present alongside our other Ansible playbooks. This step ensures that the file is correctly located and ready for deployment to our target server as part of the `deploy_static_html_page_play.yml` playbook execution.

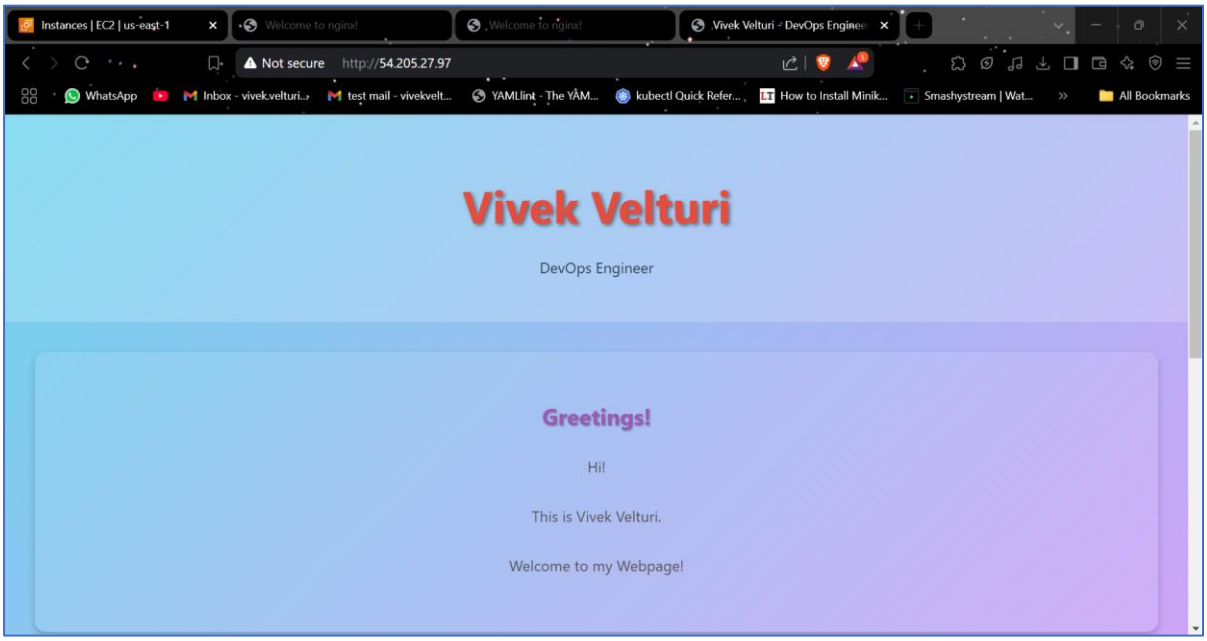
- To deploy our static website and ensure Nginx was properly installed and configured, we executed the `deploy_static_html_page_play.yml` playbook using the command: `ansible-playbook deploy_static_html_page_play.yml`.



```
ubuntu@ip-172-31-90-40:~/playbooks$ ls
date_play.yml  deploy_static_html_page_play.yml  index.html  install_nginx_play.yml
ubuntu@ip-172-31-90-40:~/playbooks$ ansible-playbook deploy_static_html_page_play.yml

PLAY [Install nginx and server static website] *****************************************************************************************************************************

TASK [Gathering Facts] *****************************************************************************************************************************************************
ok: [server_3]

TASK [Install nginx] *******************************************************************************************************************************************************
changed: [server_3]

TASK [Start nginx] *********************************************************************************************************************************************************
ok: [server_3]

TASK [Deploy web page] *****************************************************************************************************************************************************
changed: [server_3]

PLAY RECAP *****************************************************************************************************************************************************************
server_3                   : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-90-40:~/playbooks$ |
```

The output indicated the successful execution of each task, including gathering facts, installing Nginx, starting the Nginx service, and deploying the `index.html` file.
The `changed` status for the installation and deployment tasks confirmed that changes were made to the server environment.
The `PLAY RECAP` section summarized the successful completion of all tasks, with no errors or skipped tasks.
This successful execution demonstrated Ansible's ability to automate the entire deployment process, ensuring that our static website was correctly installed and configured on the target server.

- To verify the successful deployment of our static HTML page, we accessed the public IP address of `server_3` via a web browser.
Upon navigating to the address, we observed the content of our `index.html` file, displaying the static webpage as expected.



Vivek Velturi

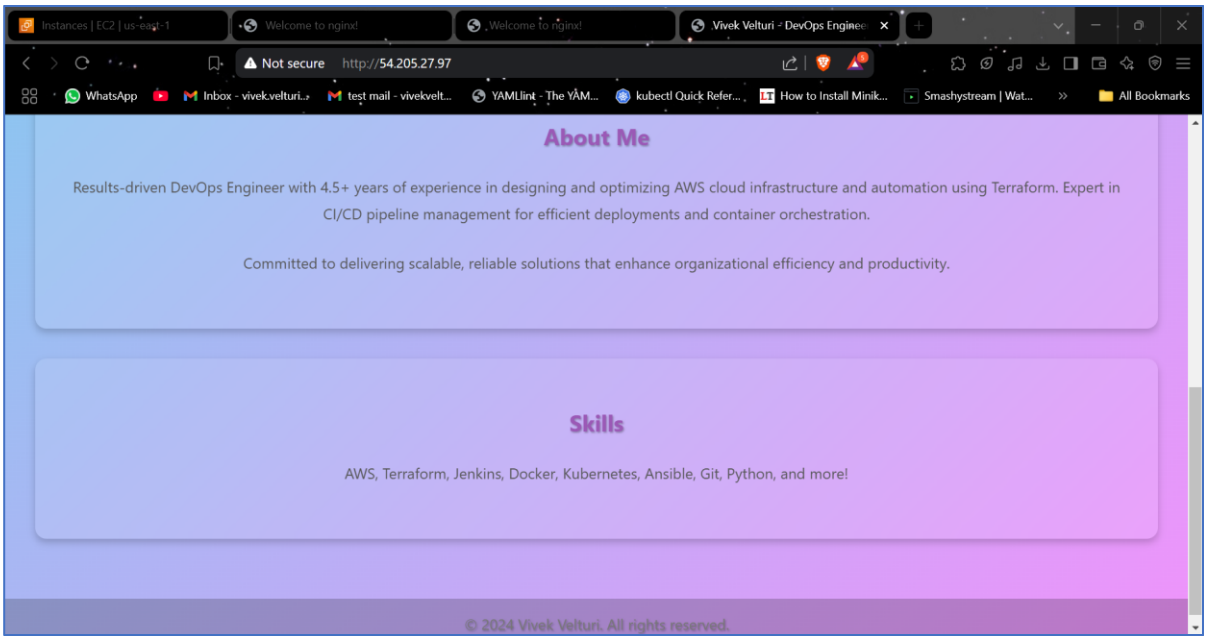
DevOps Engineer

Greetings!

Hi!

This is Vivek Velturi.

Welcome to my Webpage!



About Me

Results-driven DevOps Engineer with 4.5+ years of experience in designing and optimizing AWS cloud infrastructure and automation using Terraform. Expert in CI/CD pipeline management for efficient deployments and container orchestration.

Committed to delivering scalable, reliable solutions that enhance organizational efficiency and productivity.

Skills

AWS, Terraform, Jenkins, Docker, Kubernetes, Ansible, Git, Python, and more!

© 2024 Vivek Velturi. All rights reserved.

This confirmed that Nginx was correctly serving our static website, validating the successful execution of our Ansible playbook. The deployed page was now accessible, demonstrating Ansible's ability to automate the deployment of static websites and ensuring our content was live on the target server.

- Through this exercise, we have successfully demonstrated the capabilities of Ansible in automating various tasks across multiple servers. From installing and configuring Nginx to deploying a static website, Ansible has proven to be a powerful tool for managing and orchestrating IT infrastructure. By streamlining these processes, Ansible enables greater efficiency, reduces the risk of human error, and ensures consistency across deployments. This hands-on experience has provided valuable insights into the benefits and practical applications of Ansible in real-world scenarios.