VISHWAKARMA INSTITUTES VI	Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology **Department of** **Artificial Intelligence and Data Science**	
Name: Siddhesh Dilip Khairnar		
Class: TY	Division: B	Roll No: 372028
Semester: 5th	Academic Year: 2023-2024	
Subject Name & Code: Cloud Computing and Analytics (ADUA31203)		
Title of Assignment: Write ansible playbook to install nginx on target servers		

Assignment 6

Title: Write an ansible playbook to deploy NGINX Web server.

Theory:

1) What is YAML

YAML is a human-readable data serialization language that is often used for writing configuration files. Depending on whom you ask, YAML stands for yet another markup language or YAML isn’t markup language (a recursive acronym), which emphasizes that YAML is for data, not documents.

YAML is a popular programming language because it is designed to be easy to read and understand. It can also be used in conjunction with other programming languages. Because of its flexibility, and accessibility, YAML is used by Ansible® to create automation processes, in the form of Ansible Playbooks.

YAML syntax

YAML files use a .yml or .yaml extension, and follow specific syntax rules.

YAML has features that come from Perl, C, XML, HTML, and other programming languages. YAML is also a superset of JSON, so JSON files are valid in YAML.

There are no usual format symbols, such as braces, square brackets, closing tags, or quotation marks. And YAML files are simpler to read as they use Python-style indentation to determine the structure and indicate nesting. Tab characters are not allowed by design, to maintain portability across systems, so whitespaces—literal space characters—are used instead.

Comments can be identified with a pound or hash symbol (#). It’s always a best practice to use comments, as they describe the intention of the code. YAML does not support multi-line comment, each line needs to be suffixed with the pound character.

A common question for YAML beginners is “What do the 3 dashes mean?” 3 dashes (---) are used to signal the start of a document, while each document ends with three dots (...).

This is a very basic example of a YAML file:

```
#Comment: This is a supermarket list using YAML
#Note that - character represents the list
---
food:
  - vegetables: tomatoes #first list item
  - fruits: #second list item
      citrics: oranges
      tropical: bananas
      nuts: peanuts
      sweets: raisins
```

Note that the structure of a YAML file is a map or a list, and it follows a hierarchy depending on the indentation, and how you define your key values. Maps allow you to associate key-value pairs. Each key must be unique, and the order doesn't matter. Think of a Python dictionary or a variable assignment in a Bash script.

A map in YAML needs to be resolved before it can be closed, and a new map is created. A new map can be created by either increasing the indentation level or by resolving the previous map and starting an adjacent map.

A list includes values listed in a specific order and may contain any number of items needed. A list sequence starts with a dash (-) and a space, while indentation separates it from the parent. You can think of a sequence as a Python list or an array in Bash or Perl. A list can be embedded into a map.

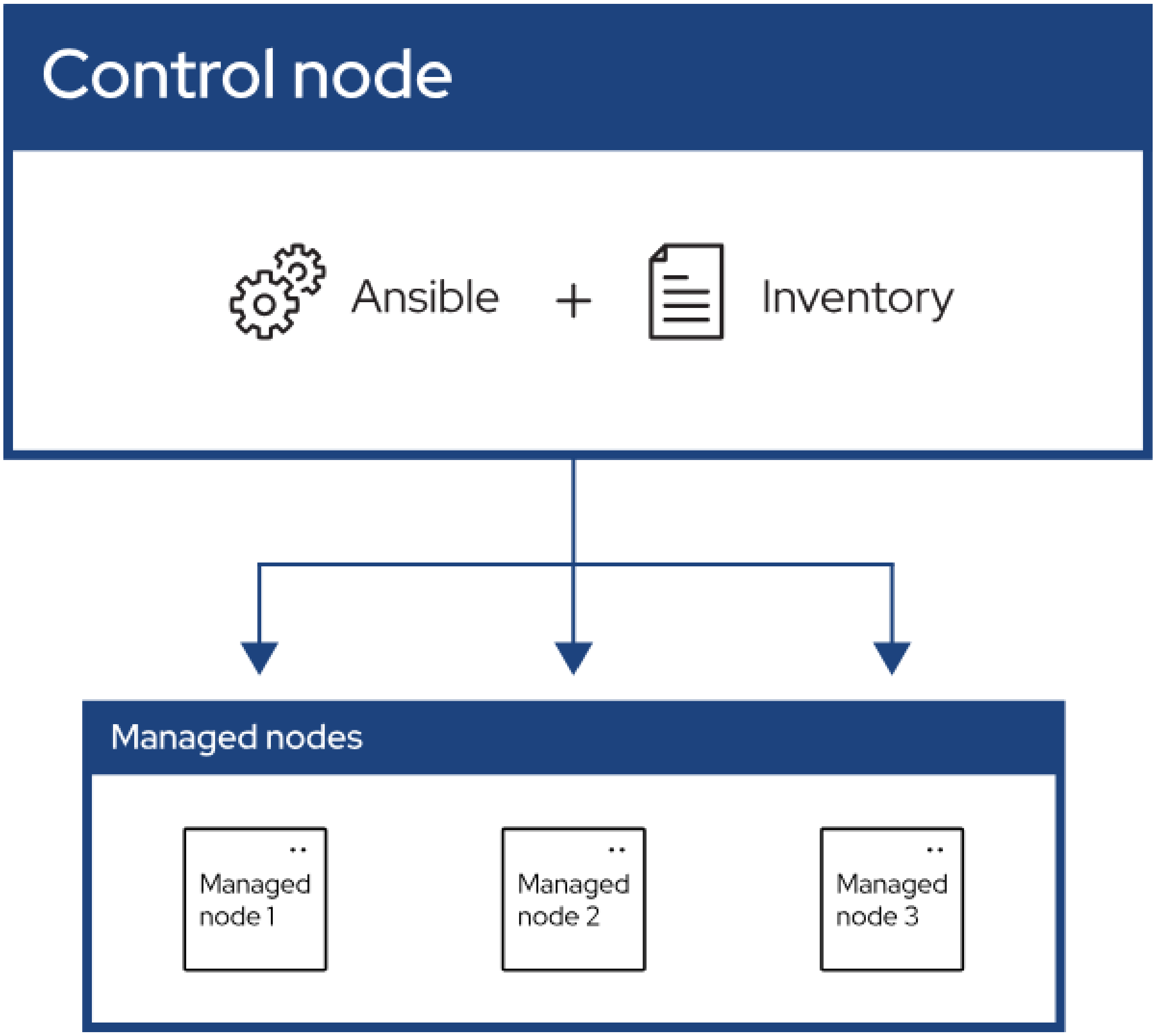
In the example provided above “vegetables” and “fruits” represent items that are part of the list named “food”.

YAML also contains scalars, which are arbitrary data (encoded in Unicode) that can be used as values such as strings, integers, dates, numbers, or booleans.

When creating a YAML file, you’ll need to ensure that you follow these syntax rules and that your file is valid. To achieve it, you can use a linter—an application that verifies the syntax of a file. The yamllint command can help to ensure you’ve created a valid YAML file before you hand it over to an application.

2) Introduction to Ansible

Ansible automates the management of remote systems and controls their desired state.



As shown in the preceding figure, most Ansible environments have three main components:

Control node

A system on which Ansible is installed. You run Ansible commands such as ansible or ansible-inventory on a control node.

Inventory

A list of managed nodes that are logically organized. You create an inventory on the control node to describe host deployments to Ansible.

Managed node

A remote system, or host, that Ansible controls.

Introduction to Ansible

Ansible provides open-source automation that reduces complexity and runs everywhere. Using Ansible lets you automate virtually any task. Here are some common use cases for Ansible:

- Eliminate repetition and simplify workflows
- Manage and maintain system configuration
- Continuously deploy complex software
- Perform zero-downtime rolling updates

Ansible uses simple, human-readable scripts called playbooks to automate your tasks. You declare the desired state of a local or remote system in your playbook. Ansible ensures that the system remains in that state.

As automation technology, Ansible is designed around the following principles:

Agentless architecture

Low maintenance overhead by avoiding the installation of additional software across IT infrastructure.

Simplicity

Automation playbooks use straightforward YAML syntax for code that reads like documentation. Ansible is also decentralized, using SSH existing OS credentials to access to remote machines.

Scalability and flexibility

Easily and quickly scale the systems you automate through a modular design that supports a large range of operating systems, cloud platforms, and network devices.

Idempotence and predictability

When the system is in the state your playbook describes, Ansible does not change anything, even if the playbook runs multiple times.

Implementation:

1. **Architecture:**

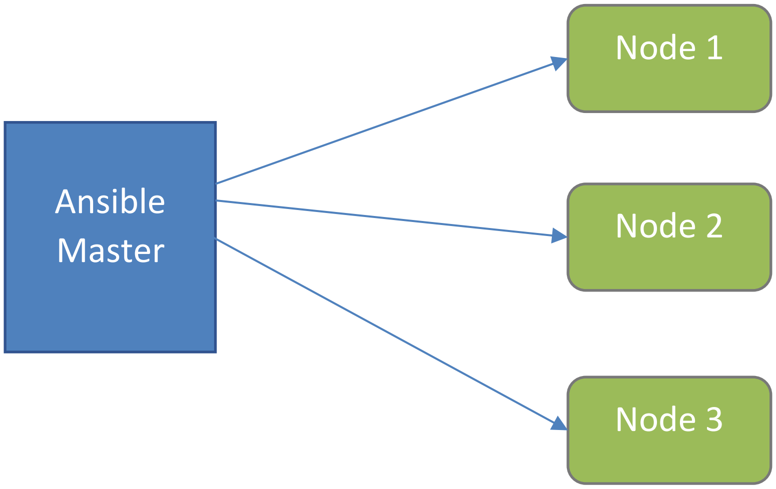
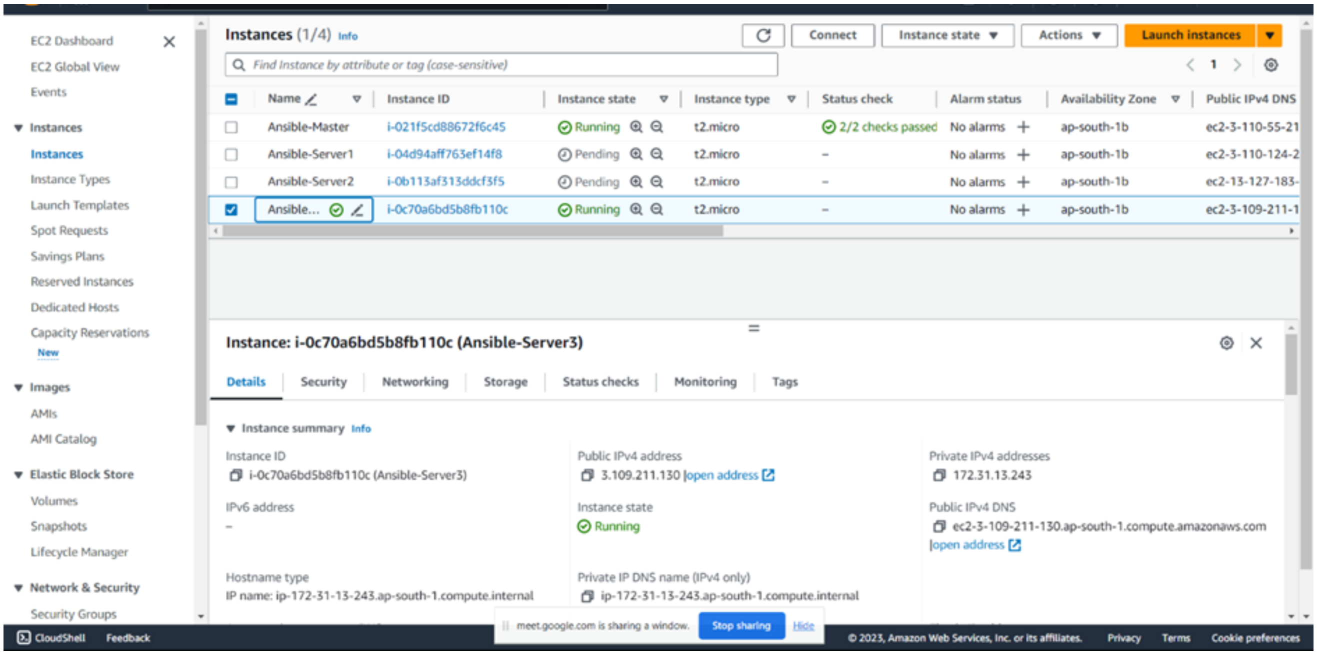


Figure1: Architecture Diagram

2. **Steps**

a) Create 4 ec2 instances of **Ubuntu machine.**



Instances (1/4) Info

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	Ansible-Master	i-021f5cd88672f6c45	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	ec2-3-110-55-21
☐	Ansible-Server1	i-04d94aff763ef14f8	Pending	t2.micro	–	No alarms	ap-south-1b	ec2-3-110-124-2
☐	Ansible-Server2	i-0b113af313ddcf3f5	Pending	t2.micro	–	No alarms	ap-south-1b	ec2-13-127-183-
☑	Ansible...	i-0c70a6bd5b8fb110c	Running	t2.micro	–	No alarms	ap-south-1b	ec2-3-109-211-1

Instance: i-0c70a6bd5b8fb110c (Ansible-Server3)

Details

Instance summary Info

Instance ID
i-0c70a6bd5b8fb110c (Ansible-Server3)

IPv6 address
–

Hostname type
IP name: ip-172-31-13-243.ap-south-1.compute.internal

Public IPv4 address
3.109.211.130 |open address

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-13-243.ap-south-1.compute.internal

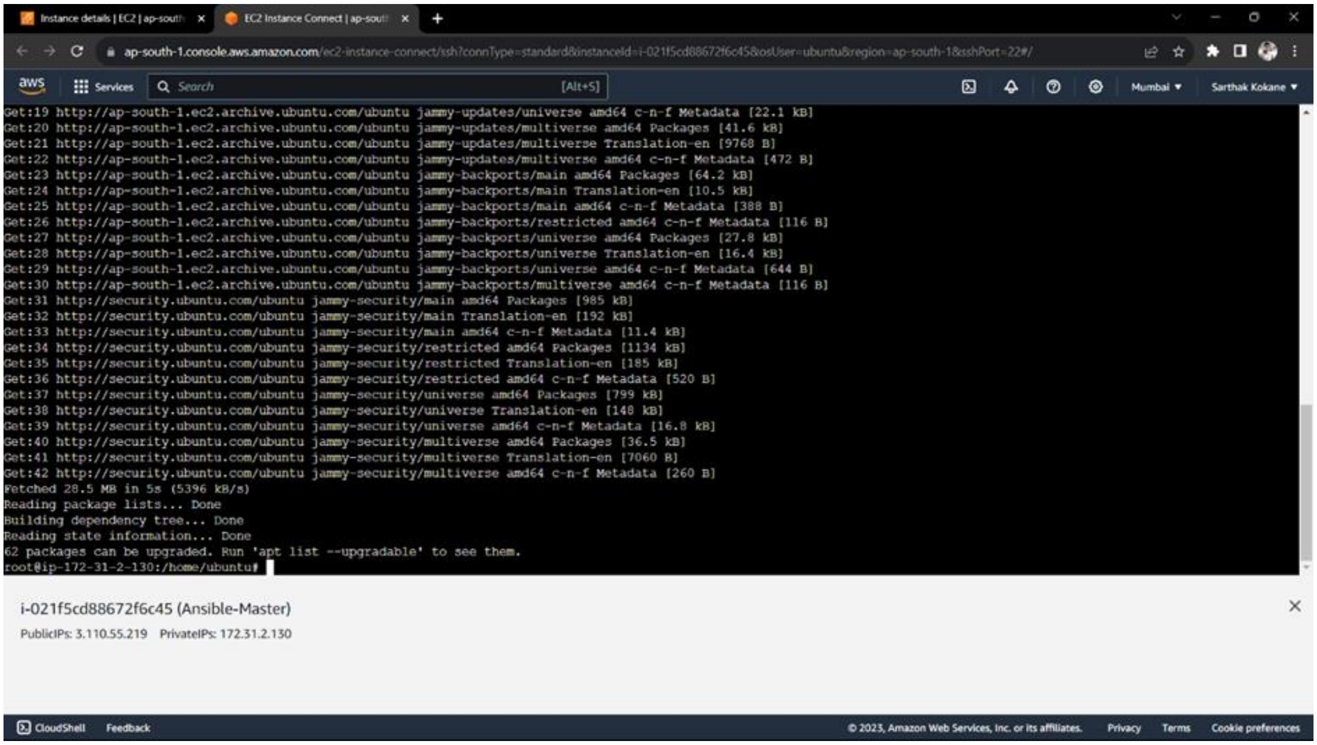
Private IPv4 addresses
172.31.13.243

Public IPv4 DNS
ec2-3-109-211-130.ap-south-1.compute.amazonaws.com
|open address

b) Connect to **"Ansible-Master"** server

c) Run following commands.

- **sudo -i**
- **apt update**

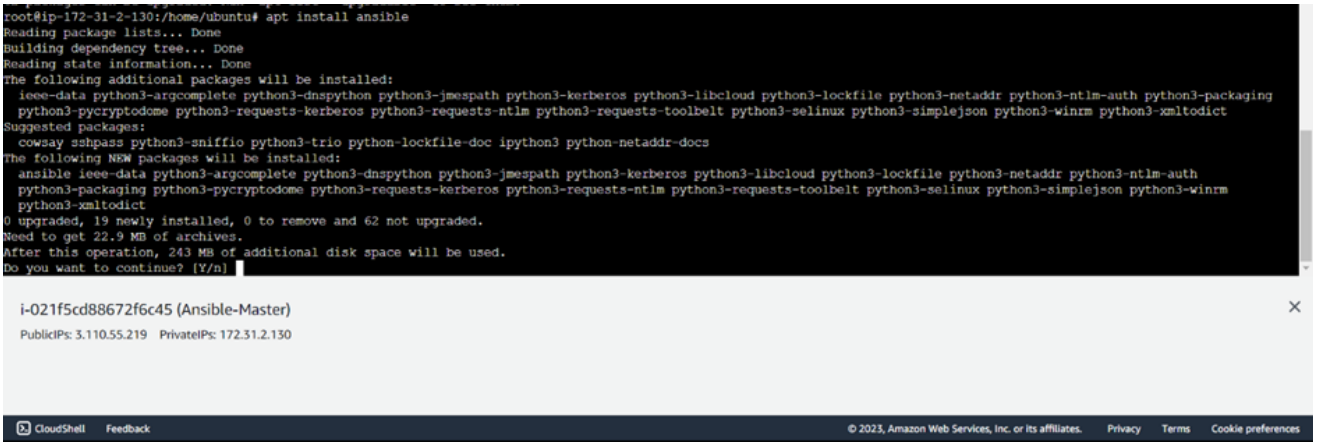

Instance details | EC2 | ap-south
EC2 Instance Connect | ap-south
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-021f5cd88672f6c45&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Sarthak Kokane
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [41.6 kB]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [9768 B]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [64.2 kB]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:29 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:30 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [985 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [192 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1134 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [185 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [520 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [799 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [148 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 28.5 MB in 5s (5396 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
62 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-2-130:/home/ubuntu#
```

i-021f5cd88672f6c45 (Ansible-Master)

PublicIPs: 3.110.55.219 PrivateIPs: 172.31.2.130

- Install ansible using command.
  apt install ansible



```
root@ip-172-31-2-130:/home/ubuntu# apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-dnspython python3-jmespath python3-kerberos python3-libcloud python3-lockfile python3-netaddr python3-ntlm-auth python3-packaging
  python3-pycryptodome python3-requests-kerberos python3-requests-ntlm python3-requests-toolbelt python3-selinux python3-simplejson python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python3-sniffio python3-trio python-lockfile-doc ipython3 python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-dnspython python3-jmespath python3-kerberos python3-libcloud python3-lockfile python3-netaddr python3-ntlm-auth
  python3-packaging python3-pycryptodome python3-requests-kerberos python3-requests-ntlm python3-requests-toolbelt python3-selinux python3-simplejson python3-winrm
  python3-xmltodict
0 upgraded, 19 newly installed, 0 to remove and 62 not upgraded.
Need to get 22.9 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

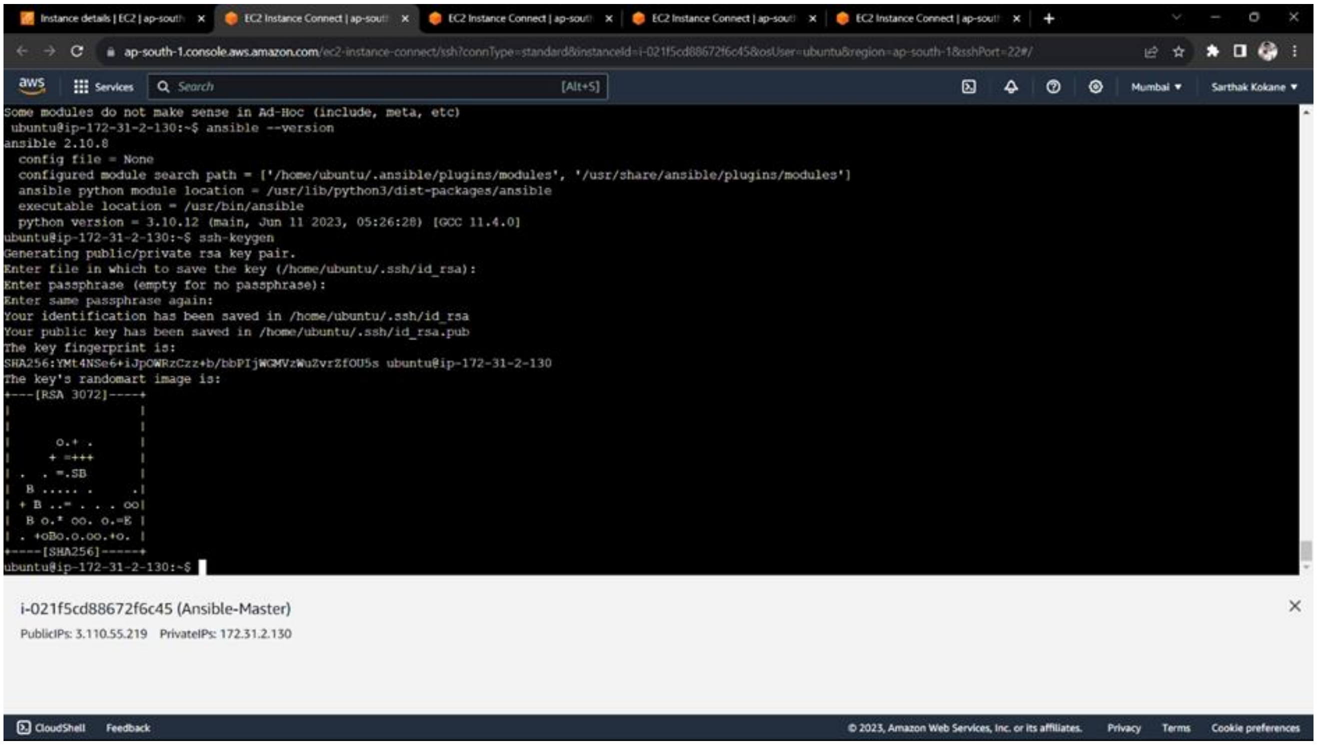
i-021f5cd88672f6c45 (Ansible-Master)

PublicIPs: 3.110.55.219 PrivateIPs: 172.31.2.130

- Check the version of ansible using command
  ansible --version
- Update remaining all hosts i.e. Ansible-Server1, Ansible-Server2, Ansible-Server3 using command
  sudo apt-get update

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

ssh-keygen


Instance details | EC2 | ap-south
EC2 Instance Connect | ap-south
EC2 Instance Connect | ap-south
EC2 Instance Connect | ap-south
EC2 Instance Connect | ap-south
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-021f5cd88672f6c45&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Sarthak Kokane
CloudShell
Feedback
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

```
Some modules do not make sense in Ad-Hoc (include, meta, etc)
 ubuntu@ip-172-31-2-130:~$ ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0]
ubuntu@ip-172-31-2-130:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:YMt4NSe6+iJpOWRzCzz+b/bbPIjWGMVzWuZvrZfOU5s ubuntu@ip-172-31-2-130
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|      o.+ .      |
|     + =+++      |
| .  . =.SB       |
|  B ..... .     .|
| + B ..= . . . oo|
|  B o.* oo. o.=E |
| . +oBo.o.oo.+o. |
+----[SHA256]-----+
ubuntu@ip-172-31-2-130:~$
```

i-021f5cd88672f6c45 (Ansible-Master)

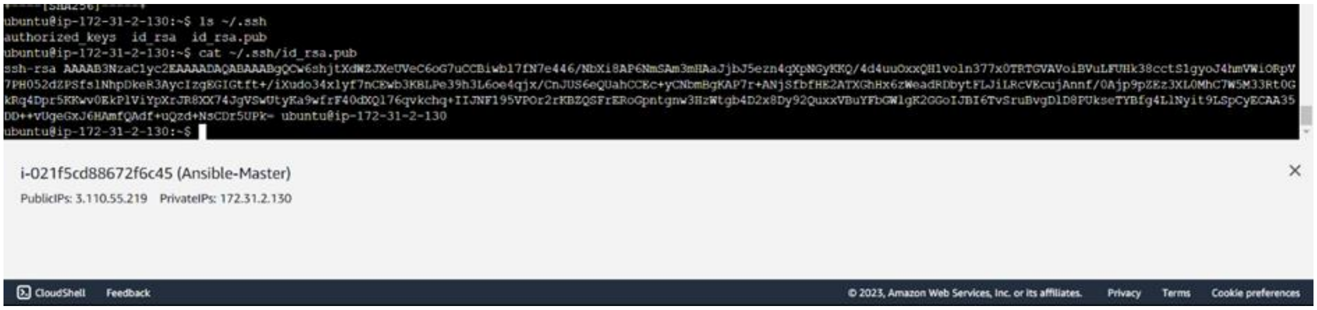
PublicIPs: 3.110.55.219   PrivateIPs: 172.31.2.130

e) copy the public key which is in .ssh folder into "authorized keys" on ansible-server1

commands:

ls ~/.ssh

cat ~/.ssh/id_rsa.pub



```
+----[SHA256]-----+
ubuntu@ip-172-31-2-130:~$ ls ~/.ssh
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-2-130:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCw6shjtXdWZJXeUVeC6oG7uCCBiwbl7fN7e446/NbXi8AP6NmSAm3mHAaJjbJ5ezn4qXpNGyKKQ/4d4uuOxxQHlvoln377x0TRTGVAVoiBVuLFUHk38cctSlgyoJ4hmVWiORpV
7PH052dZPSfslNhpDkeR3AycIzgEGIGtft+/iXudo34xlyf7nCEwb3KBLPe39h3L6oe4qjx/CnJUS6eQUahCCEc+yCNbmBqKAP7r+ANjSfbfHE2ATXGhHx6zWeadRDbytFLJiLRcVEcujAnnf/0Ajp9pZEz3XL0MhC7W5M33Rt0G
kRq4Dpr5KKwv0EkPlViYpXrJR8XX74JgVSwUtyKa9wfrF40dXQl76qvkchq+IIJNF195VPOr2rKBZQSFrERoGpntgnw3HzWtgb4D2x8Dy92QuxxVBuYFbGWlgK2GGoIJBI6TvSruBvgDlD8PUkseTYBfg4LlNyit9LSpCyECAA35
DD++vUgeGxJ6HAmfQAdf+uQzd+NsCDr5UPk= ubuntu@ip-172-31-2-130
ubuntu@ip-172-31-2-130:~$
```

i-021f5cd88672f6c45 (Ansible-Master)

PublicIPs: 3.110.55.219   PrivateIPs: 172.31.2.130

f) Connect to ansible-server1 and again give command

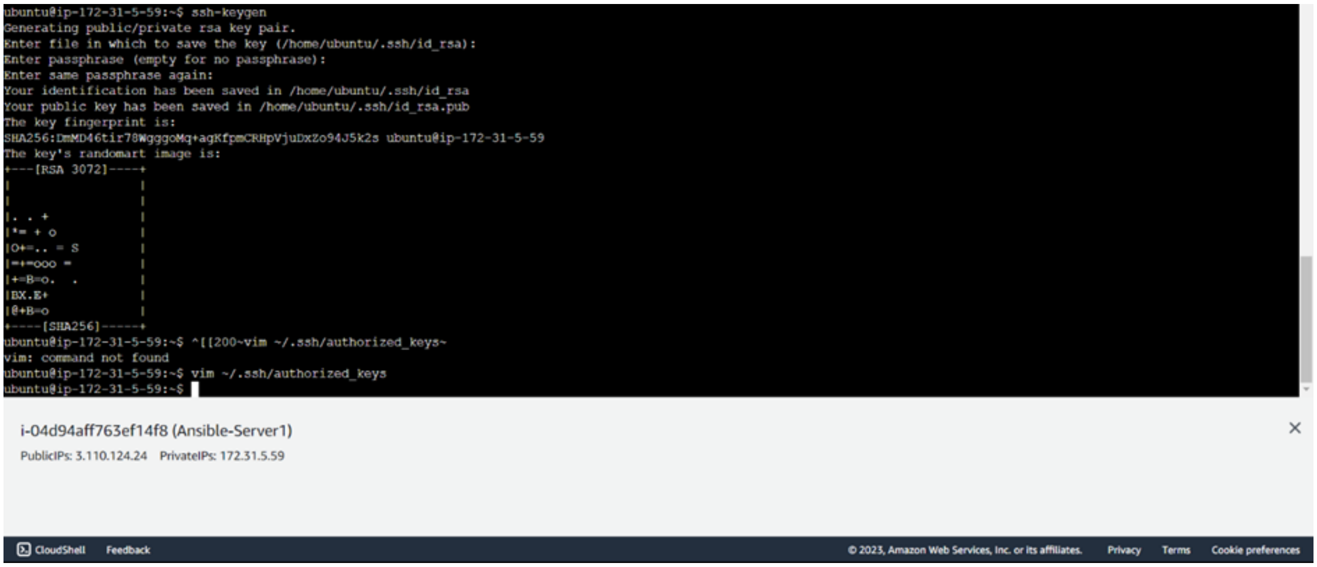
ssh-keygen

It will create the same files on ansible-server1

Now,

vim ~/.ssh/authorized_keys

copy the public key

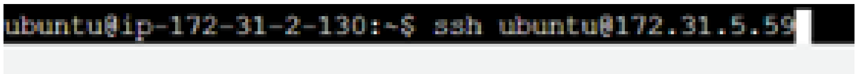


```
ubuntu@ip-172-31-5-59:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DmMD46tir78WgggoMq+agKfpmCRHpVjuDxZo94J5k2s ubuntu@ip-172-31-5-59
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|. . +            |
|*= + o           |
|O+=.. = S        |
|=+=ooo =         |
|+=B=o.  .        |
|BX.E+            |
|@+B=o            |
+----[SHA256]-----+
ubuntu@ip-172-31-5-59:~$ ^[[200~vim ~/.ssh/authorized_keys~
vim: command not found
ubuntu@ip-172-31-5-59:~$ vim ~/.ssh/authorized_keys
ubuntu@ip-172-31-5-59:~$
```

i-04d94aff763ef14f8 (Ansible-Server1)

PublicIPs: 3.110.124.24 PrivateIPs: 172.31.5.59

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



```
ubuntu@ip-172-31-2-130:~$ ssh ubuntu@172.31.5.59
```

Create a playbook on Ansible-master

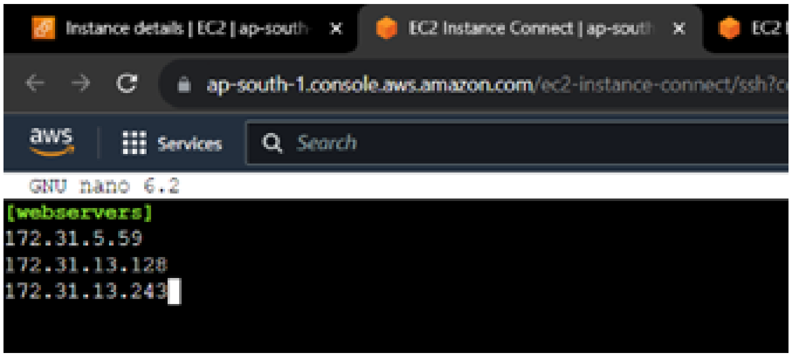
Step 1:- Connect to "Ansible-Master"

Step 2:- Create a new folder "ansible-project" using command

Step 3:

- cd ansible-project
- nano inventory

1. Write the private IP of "Ansible-server1" into inventory.
2. Write the private IP of "Ansible-server2" into inventory.
3. Write the private IP of "Ansible-server3" into inventory.



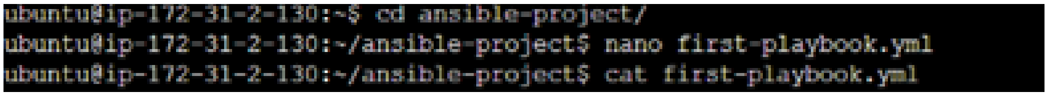
Instance details | EC2 | ap-south
EC2 Instance Connect | ap-south
EC2
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?c
aws
Services
Search

```
GNU nano 6.2
[webservers]
172.31.5.59
172.31.13.128
172.31.13.243
```

4. Save and exit.

Task: Install Nginx and Start Nginx

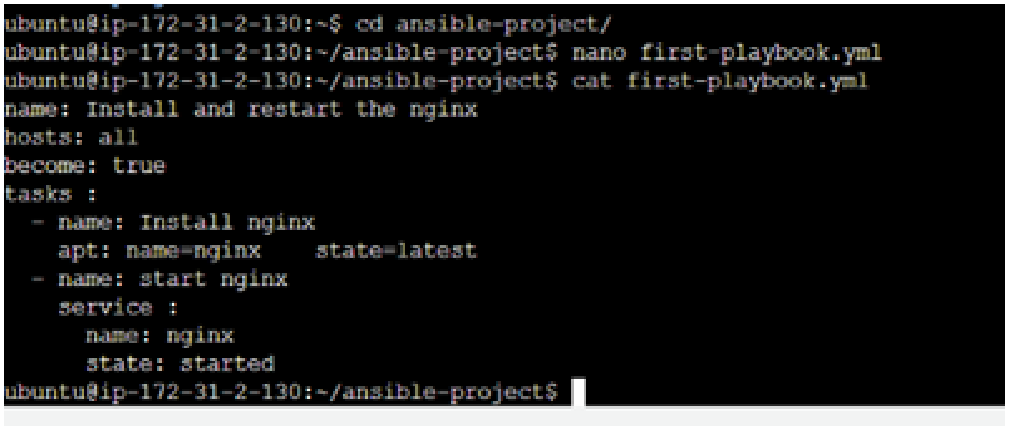
Step 1: Create a new file called “first-playbook.yml”



```
ubuntu@ip-172-31-2-130:~$ cd ansible-project/
ubuntu@ip-172-31-2-130:~/ansible-project$ nano first-playbook.yml
ubuntu@ip-172-31-2-130:~/ansible-project$ cat first-playbook.yml
```

Code:

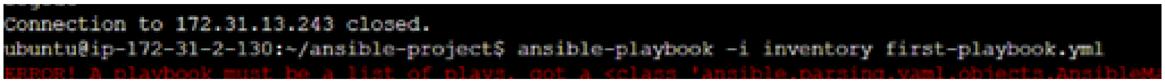
```
- name: Install and restart the nginx
  hosts: all
  become: true
  tasks:
    - name: install nginx
      apt: name=nginx    state=latest
    - name: start nginx
      service:
        name: nginx
        state: started
```



```
ubuntu@ip-172-31-2-130:~$ cd ansible-project/
ubuntu@ip-172-31-2-130:~/ansible-project$ nano first-playbook.yml
ubuntu@ip-172-31-2-130:~/ansible-project$ cat first-playbook.yml
name: Install and restart the nginx
hosts: all
become: true
tasks :
  - name: Install nginx
    apt: name=nginx    state=latest
  - name: start nginx
    service :
      name: nginx
      state: started
ubuntu@ip-172-31-2-130:~/ansible-project$
```

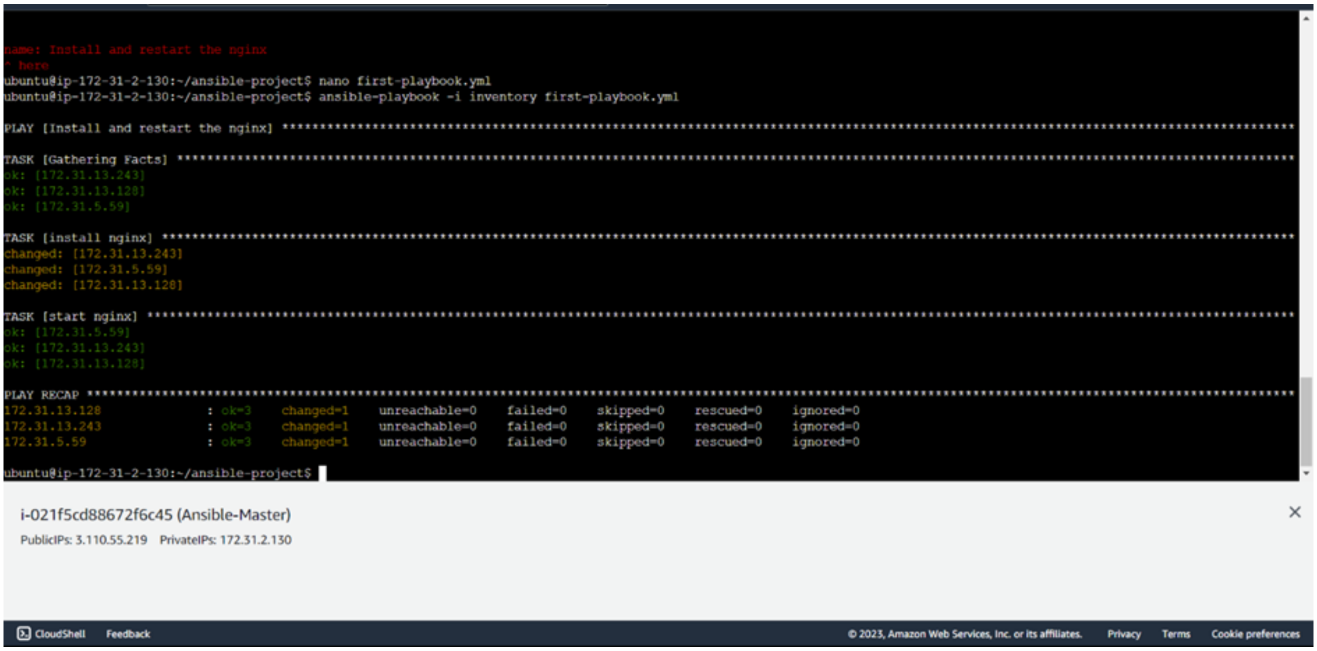
Execute the playbook by using command:

ansible-playbook -i inventory first-playbook.yml



```
Connection to 172.31.13.243 closed.
ubuntu@ip-172-31-2-130:~/ansible-project$ ansible-playbook -i inventory first-playbook.yml
ERROR! A playbook must be a list of plays, got a <class 'ansible.parsing.yaml.objects.AnsibleM
```

Output:



```
name: Install and restart the nginx
^ here
ubuntu@ip-172-31-2-130:~/ansible-project$ nano first-playbook.yml
ubuntu@ip-172-31-2-130:~/ansible-project$ ansible-playbook -i inventory first-playbook.yml

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

TASK [Gathering Facts] *********************************************************************************************
ok: [172.31.13.243]
ok: [172.31.13.128]
ok: [172.31.5.59]

TASK [install nginx] ***********************************************************************************************
changed: [172.31.13.243]
changed: [172.31.5.59]
changed: [172.31.13.128]

TASK [start nginx] *************************************************************************************************
ok: [172.31.5.59]
ok: [172.31.13.243]
ok: [172.31.13.128]

PLAY RECAP *********************************************************************************************************
172.31.13.128              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.13.243              : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.5.59                : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-2-130:~/ansible-project$
```

i-021f5cd88672f6c45 (Ansible-Master)

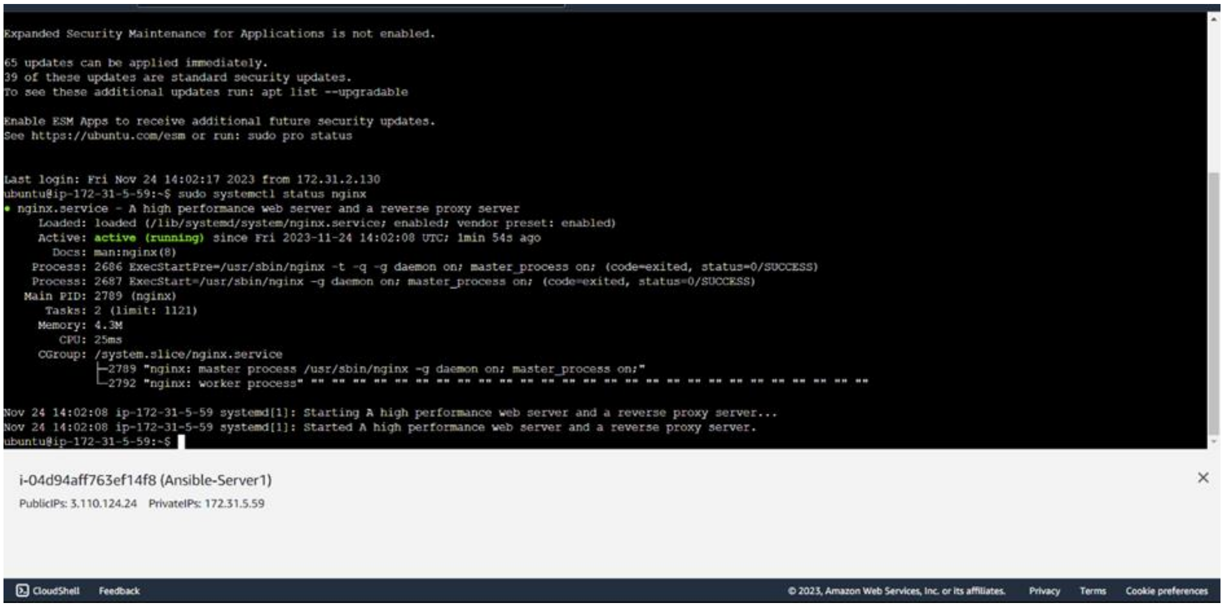
PublicIPs: 3.110.55.219 PrivateIPs: 172.31.2.130

Verify the output:

Step 1: Connect to any Ansible-Server

Step 2: Run the command: **sudo systemctl status nginx**

- *Ansible-Server1*

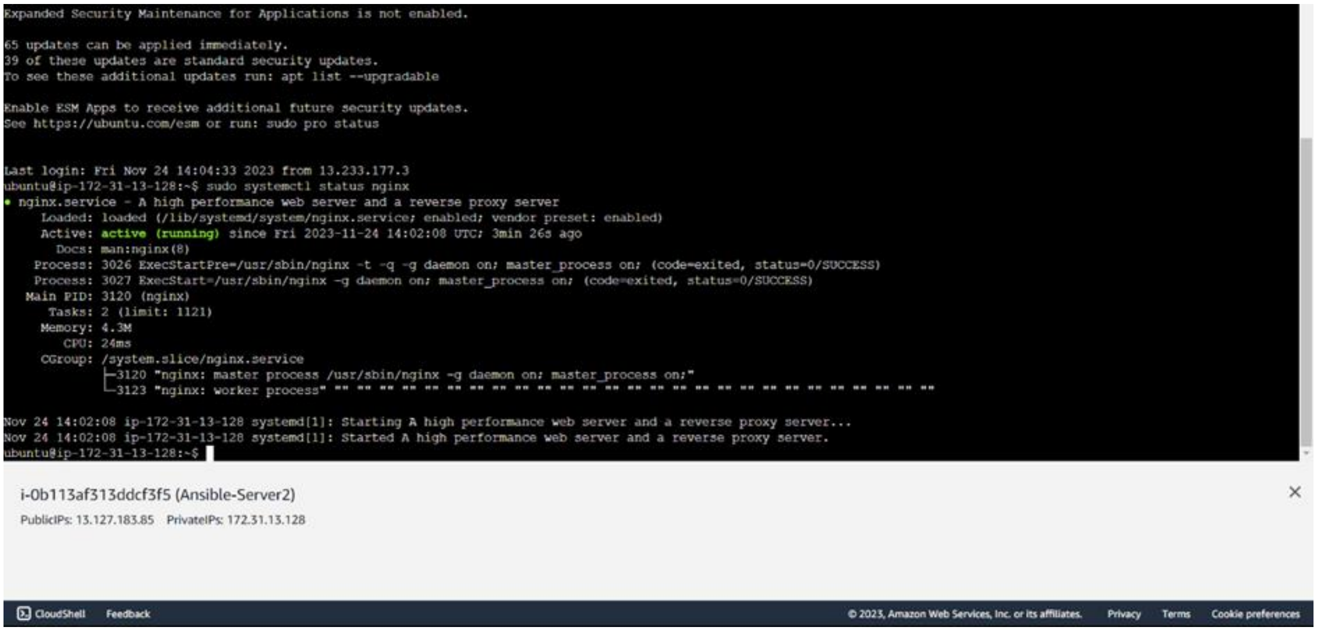


```
Expanded Security Maintenance for Applications is not enabled.

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

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


Last login: Fri Nov 24 14:02:17 2023 from 172.31.2.130
ubuntu@ip-172-31-5-59:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-11-24 14:02:08 UTC; 1min 54s ago
       Docs: man:nginx(8)
    Process: 2686 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2687 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2789 (nginx)
      Tasks: 2 (limit: 1121)
     Memory: 4.3M
        CPU: 25ms
     CGroup: /system.slice/nginx.service
             ├─2789 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2792 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Nov 24 14:02:08 ip-172-31-5-59 systemd[1]: Starting A high performance web server and a reverse proxy server...
Nov 24 14:02:08 ip-172-31-5-59 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-5-59:~$
```

i-04d94aff763ef14f8 (Ansible-Server1)

PublicIPs: 3.110.124.24 PrivateIPs: 172.31.5.59

- *Ansible-Server2*

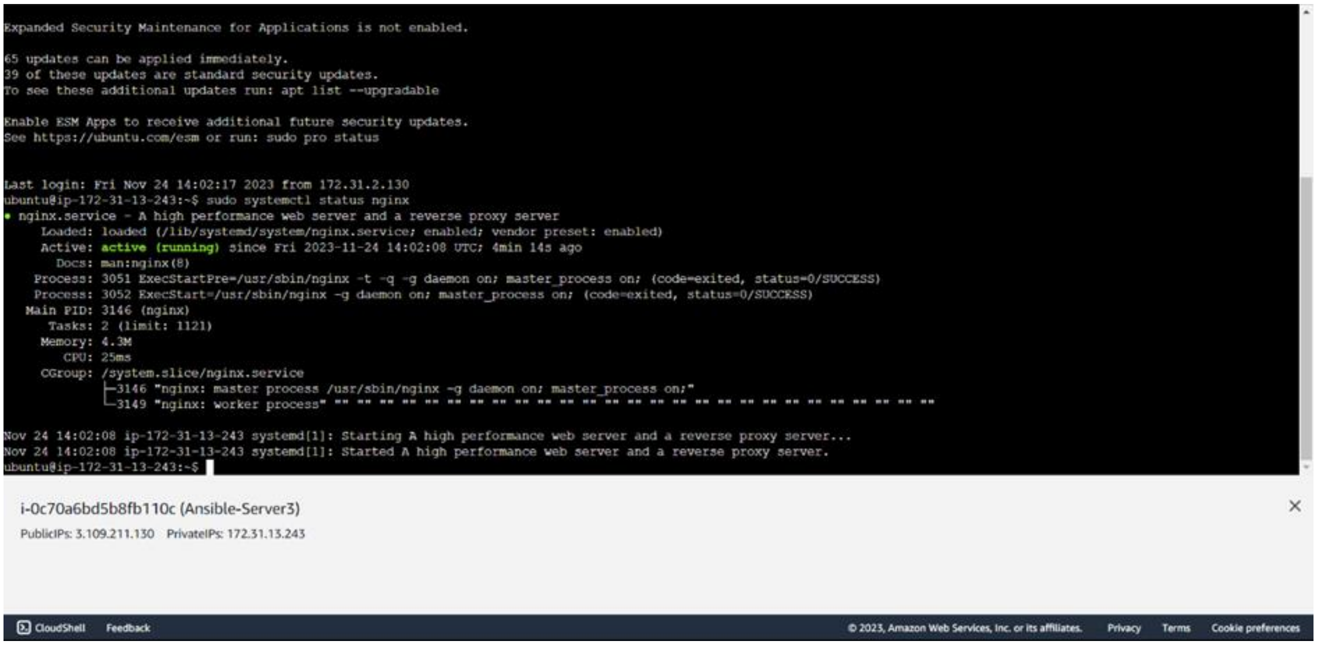


```
Expanded Security Maintenance for Applications is not enabled.

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

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


Last login: Fri Nov 24 14:04:33 2023 from 13.233.177.3
ubuntu@ip-172-31-13-128:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-11-24 14:02:08 UTC; 3min 26s ago
       Docs: man:nginx(8)
    Process: 3026 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3027 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3120 (nginx)
      Tasks: 2 (limit: 1121)
     Memory: 4.3M
        CPU: 24ms
     CGroup: /system.slice/nginx.service
             ├─3120 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3123 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Nov 24 14:02:08 ip-172-31-13-128 systemd[1]: Starting A high performance web server and a reverse proxy server...
Nov 24 14:02:08 ip-172-31-13-128 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-13-128:~$
```

i-0b113af313ddcf3f5 (Ansible-Server2)

PublicIPs: 13.127.183.85   PrivateIPs: 172.31.13.128

- *Ansible-Server3*



```
Expanded Security Maintenance for Applications is not enabled.

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

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


Last login: Fri Nov 24 14:02:17 2023 from 172.31.2.130
ubuntu@ip-172-31-13-243:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-11-24 14:02:08 UTC; 4min 14s ago
       Docs: man:nginx(8)
    Process: 3051 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3052 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3146 (nginx)
      Tasks: 2 (limit: 1121)
     Memory: 4.3M
        CPU: 25ms
     CGroup: /system.slice/nginx.service
             ├─3146 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3149 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Nov 24 14:02:08 ip-172-31-13-243 systemd[1]: Starting A high performance web server and a reverse proxy server...
Nov 24 14:02:08 ip-172-31-13-243 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-13-243:~$
```

i-0c70a6bd5b8fb110c (Ansible-Server3)

PublicIPs: 3.109.211.130   PrivateIPs: 172.31.13.243

Conclusion: Thus, we have successfully used Ansible playbooks to install nginx on 3 target servers.