Name → Siddharth Sutar

PRN → 22010517

Roll No → 323054

SUBJECT → CLOUD COMPUTING

CC Assignment 6

Write an ansible playbook to deploy Apache/Nginx Web server. (use roles)

Q1. What is YAML?

1. YAML (short for "YAML Ain't Markup Language") is a human-readable data serialization language that is often used for configuration files, data exchange between languages, and object representation in programming languages. YAML is designed to be easy to read and write for humans, making it a popular choice for configuration files in software development
2. YAML is a structured format that uses indentation and whitespace to organize data, instead of using braces or other characters like other programming languages. It uses a syntax that is easy to read and write, and can be used for complex data structures as well as simple key-value pairs
3. YAML

```
#This is a comment
fruits:
  - apple
  - banana
  - orange
vegetables:
  - carrot
  - tomato
```

4. In this example, we define a list of fruits and vegetables using YAML syntax. The # character denotes a comment in YAML, and the - character is used to define items in a list. The indentation level of each item determines the hierarchy and structure of the data

5. Data Types in YAML

a. LIST

```
# WAY 1 to write list

STUDENT1:
    name: Siddharth
    rollNo: 323054
    college: viit
    Subjects:
        - AMD
        - CC
        - DAA
        - SDA
        - EEF

# WAY 2 to write list

STUDENT2:
    name: Siddharth
    rollNo: 323054
    college: viit
    Subjects: [ AMD, CC, DAA, SDA, EEF ]
```

b. Strings

```
# 1
name: "Siddharth"
# 2
name2: 'Sidd'
# 3
name3: Siddharth
# String pipe
name4: |
This is first line
This is second line
This is third line
# Multiple lines are written but considered as single line
name5: >
This is first line
This is second line
This is third line
```

c. Numbers

```
num1: 23
```

```
num2: 123.1
num3: 0.1
```

d. Boolean

```
isTrue: true
isFalse: No
isFalse: off
```

e. Start & end

```
# Start
– – –

. . .
# End
```

Q2. What is ansible?

1. Ansible is an open-source automation tool that enables users to manage, configure, and deploy software applications and infrastructure across multiple machines or servers. It is designed to simplify the process of IT automation, making it easier to manage and maintain complex systems
2. Ansible works by defining and executing playbooks, which are sets of instructions or tasks that specify what actions need to be taken on a particular machine or group of machines. These playbooks can be written in YAML format and can include tasks such as installing software, configuring servers, and deploying applications
3. One of the key features of Ansible is its agentless architecture, which means that it does not require any software to be installed on the machines it is managing. Instead, it uses SSH and other standard protocols to communicate with remote machines, making it easy to get started and reducing the overall complexity of the system.
4. Overall, Ansible is a powerful automation tool that can help streamline IT operations, reduce manual labor, and improve system reliability and security
5. Basic Ansible concepts
    a. Inventory: The inventory is a list of servers or nodes that Ansible can manage. It's a file that contains a list of hostnames or IP addresses, organized into groups or categories. The inventory can be static or dynamic and can be defined in various formats like INI, YAML, or JSON
    b. Playbooks: Playbooks are files that define a series of tasks to be executed by Ansible. They are written in YAML format and describe the desired state of a system

c. Tasks: A task is a single action that Ansible can perform on a target node. A task can be anything from installing a package to configuring a service or updating a file
d. Modules: Ansible uses modules to perform specific actions on target nodes. Modules are small programs that can be executed on the target node to change the system's state. Ansible comes with many built-in modules, and you can also create custom modules to extend Ansible's functionality
e. Roles: A role is a collection of related tasks, files, and templates that can be reused across multiple playbooks. Roles can be thought of as reusable components that can simplify playbook development and make it easier to manage complex systems
f. Handlers: Handlers are similar to tasks, but they are only executed if a specific event occurs, such as a configuration file being updated. Handlers are defined in playbooks and can be triggered by specific events using the notify keyword
g. Variables: Ansible uses variables to store and retrieve values that are used throughout a playbook. Variables can be defined at various levels, including playbook level, role level, and host level
h. Templates: Templates are files that contain placeholders for dynamic data that is generated at runtime. Ansible can use templates to generate configuration files, scripts, and other files that need to be customized for each target node

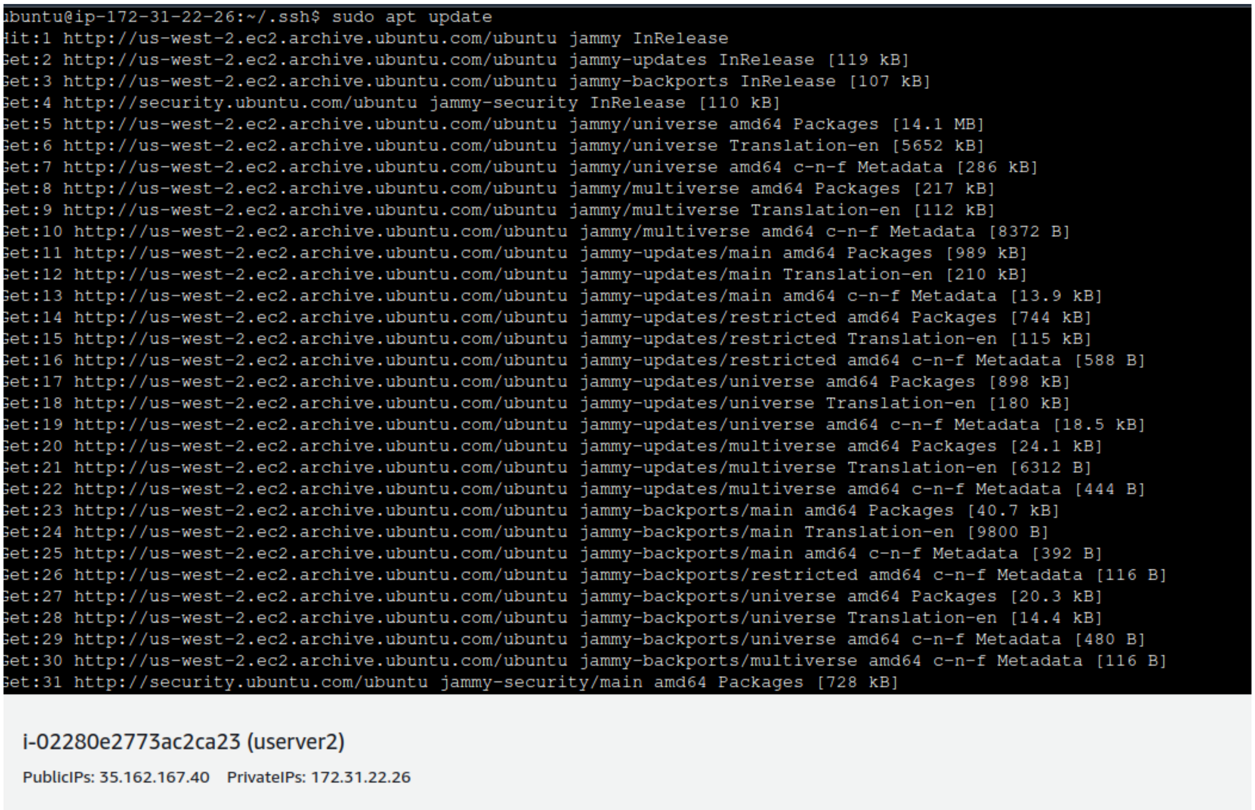
6. Overall, these are some of the key concepts in Ansible. Understanding these concepts is essential for working with Ansible effectively and creating robust, maintainable automation workflows

Installing Ansible

→ Put the commands below in order respectively

```
$ sudo apt update
```

On Next Page

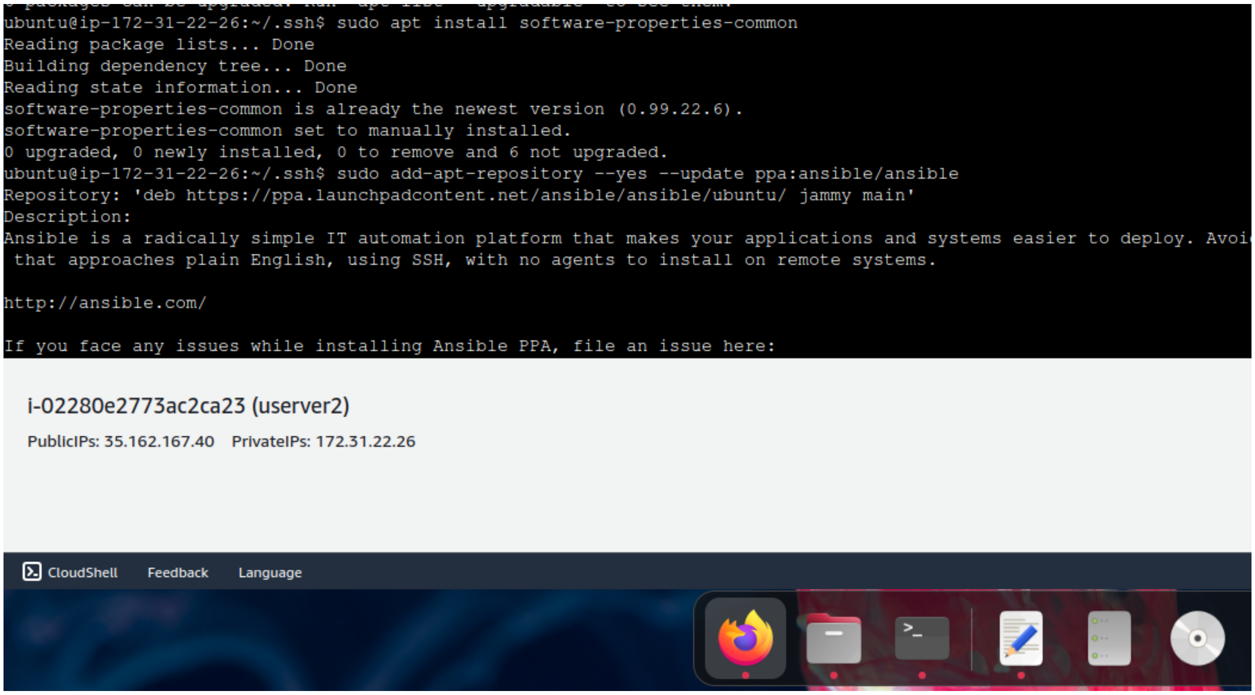


```
ubuntu@ip-172-31-22-26:~/.ssh$ sudo apt update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [989 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [210 kB]
Get:13 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.9 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [744 kB]
Get:15 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [115 kB]
Get:16 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [588 B]
Get:17 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [898 kB]
Get:18 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [180 kB]
Get:19 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.5 kB]
Get:20 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24.1 kB]
Get:21 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [6312 B]
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.7 kB]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [9800 B]
Get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [392 B]
Get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [20.3 kB]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [14.4 kB]
Get:29 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [480 B]
Get:30 http://us-west-2.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 [728 kB]
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40 PrivateIPs: 172.31.22.26

$ sudo apt install software-properties-common

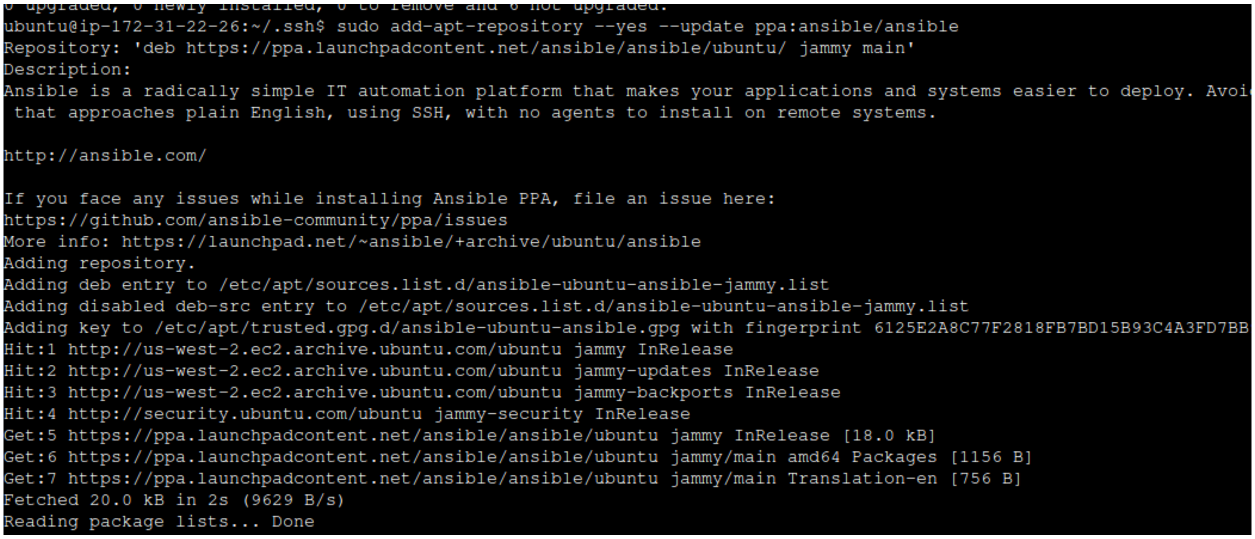


```
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-22-26:~/.ssh$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.6).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
ubuntu@ip-172-31-22-26:~/.ssh$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoi
 that approaches plain English, 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:
```

i-02280e2773ac2ca23 (userver2)

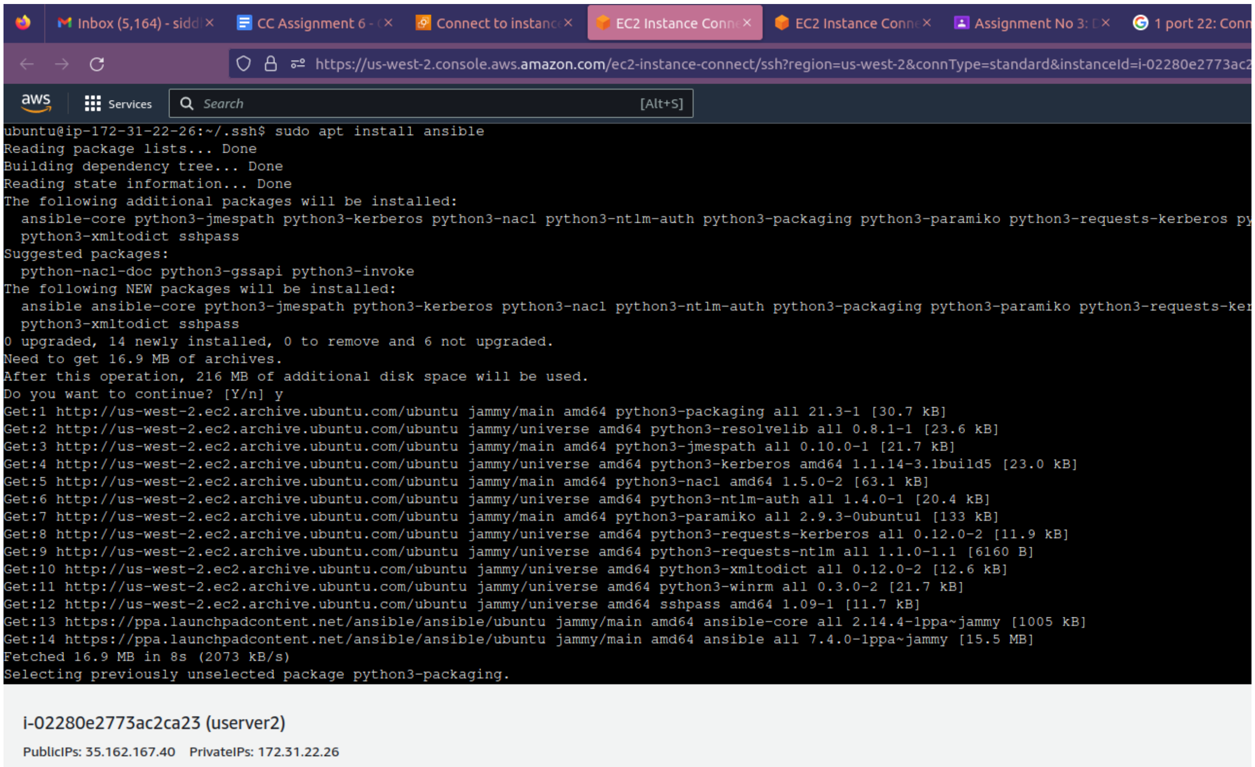
PublicIPs: 35.162.167.40 PrivateIPs: 172.31.22.26

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



```
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
ubuntu@ip-172-31-22-26:~/.ssh$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoi
 that approaches plain English, 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.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint 6125E2A8C77F2818FB7BD15B93C4A3FD7BB
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.0 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1156 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [756 B]
Fetched 20.0 kB in 2s (9629 B/s)
Reading package lists... Done
```

$ sudo apt install ansible



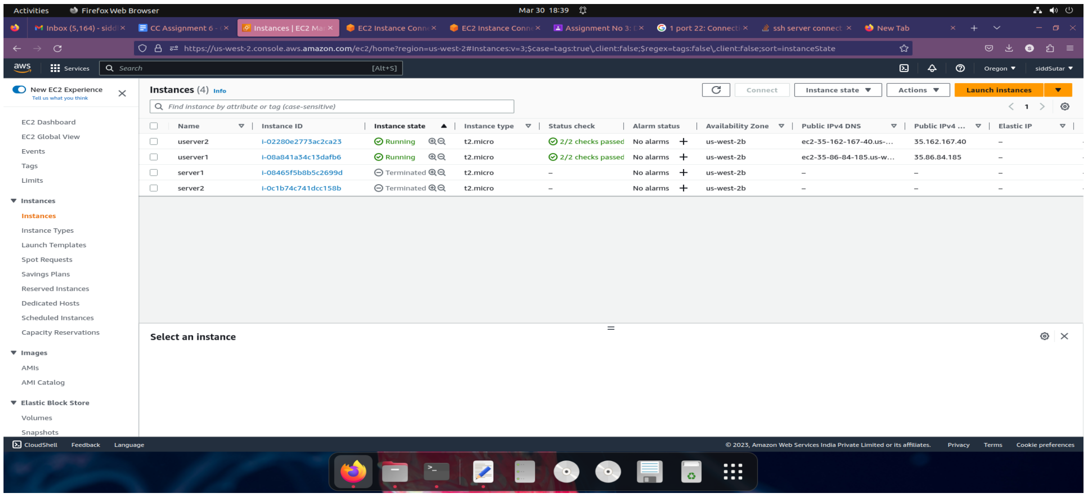
```
ubuntu@ip-172-31-22-26:~/.ssh$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests-kerberos py
  python3-xmltodict sshpass
Suggested packages:
  python-nacl-doc python3-gssapi python3-invoke
The following NEW packages will be installed:
  ansible ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth python3-packaging python3-paramiko python3-requests-ker
  python3-xmltodict sshpass
0 upgraded, 14 newly installed, 0 to remove and 6 not upgraded.
Need to get 16.9 MB of archives.
After this operation, 216 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-packaging all 21.3-1 [30.7 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-resolvelib all 0.8.1-1 [23.6 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-jmespath all 0.10.0-1 [21.7 kB]
Get:4 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-kerberos amd64 1.1.14-3.1build5 [23.0 kB]
Get:5 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-nacl amd64 1.5.0-2 [63.1 kB]
Get:6 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-ntlm-auth all 1.4.0-1 [20.4 kB]
Get:7 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 python3-paramiko all 2.9.3-0ubuntu1 [133 kB]
Get:8 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-kerberos all 0.12.0-2 [11.9 kB]
Get:9 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-requests-ntlm all 1.1.0-1.1 [6160 B]
Get:10 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-xmltodict all 0.12.0-2 [12.6 kB]
Get:11 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-winrm all 0.3.0-2 [21.7 kB]
Get:12 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 sshpass amd64 1.09-1 [11.7 kB]
Get:13 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible-core all 2.14.4-1ppa~jammy [1005 kB]
Get:14 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible all 7.4.0-1ppa~jammy [15.5 MB]
Fetched 16.9 MB in 8s (2073 kB/s)
Selecting previously unselected package python3-packaging.
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40 PrivateIPs: 172.31.22.26

Installing Apache Web Server using Ansible

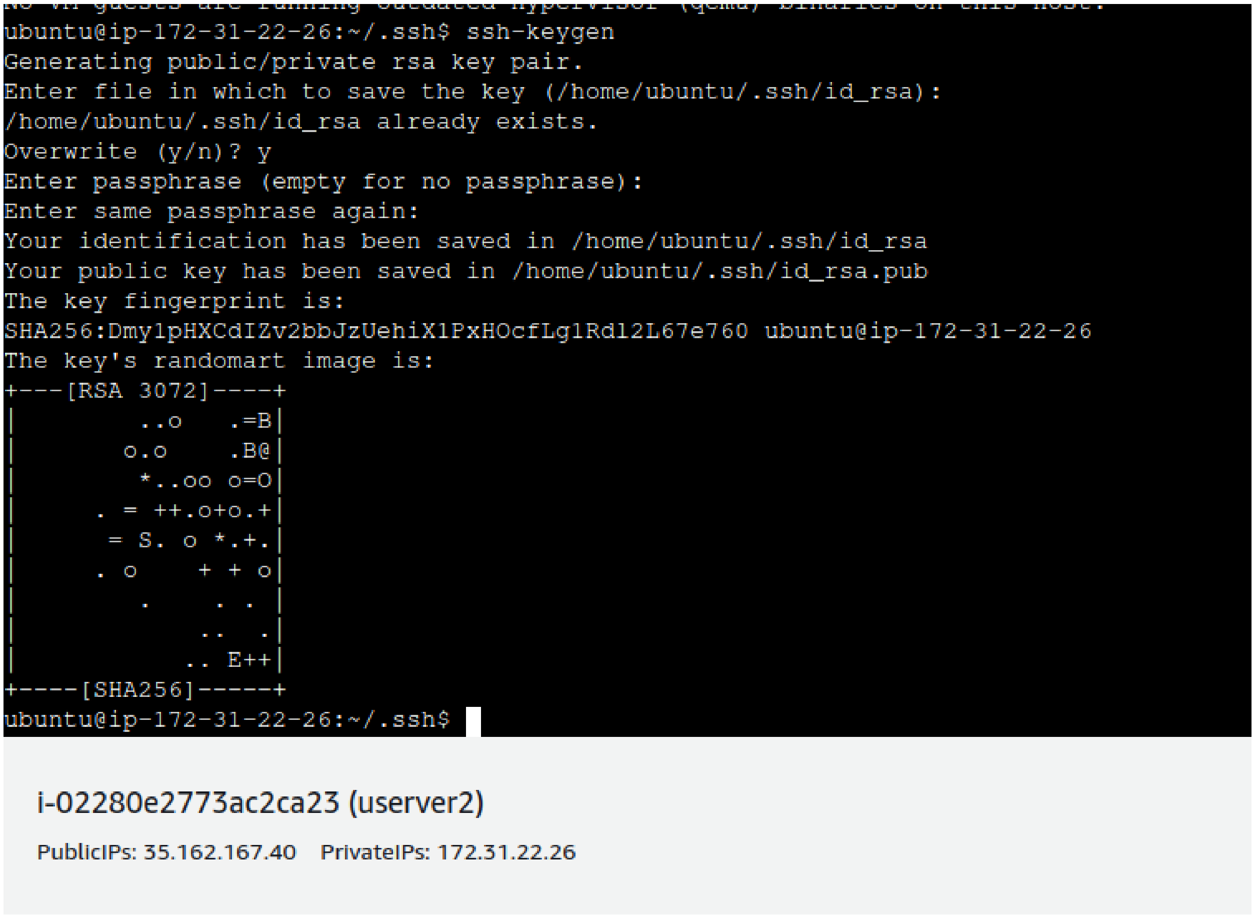
1. Create 2 Instances on AWS (Ansible is installed on userver2 in previous step)



Instances (4)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
userver2	i-02280e2773ac2ca23	Running	t2.micro	2/2 checks passed	No alarms	us-west-2b	ec2-35-162-167-40.us-...	35.162.167.40	–
userver1	i-08a841a34c13dafb6	Running	t2.micro	2/2 checks passed	No alarms	us-west-2b	ec2-35-86-84-185.us-w...	35.86.84.185	–
server1	i-08465f5b8b5c2699d	Terminated	t2.micro	–	No alarms	us-west-2b	–	–	–
server2	i-0c1b74c741dcc158b	Terminated	t2.micro	–	No alarms	us-west-2b	–	–	–

2. Generate SSH using ssh-keygen from server 2 & copy it to auth keys in server 1

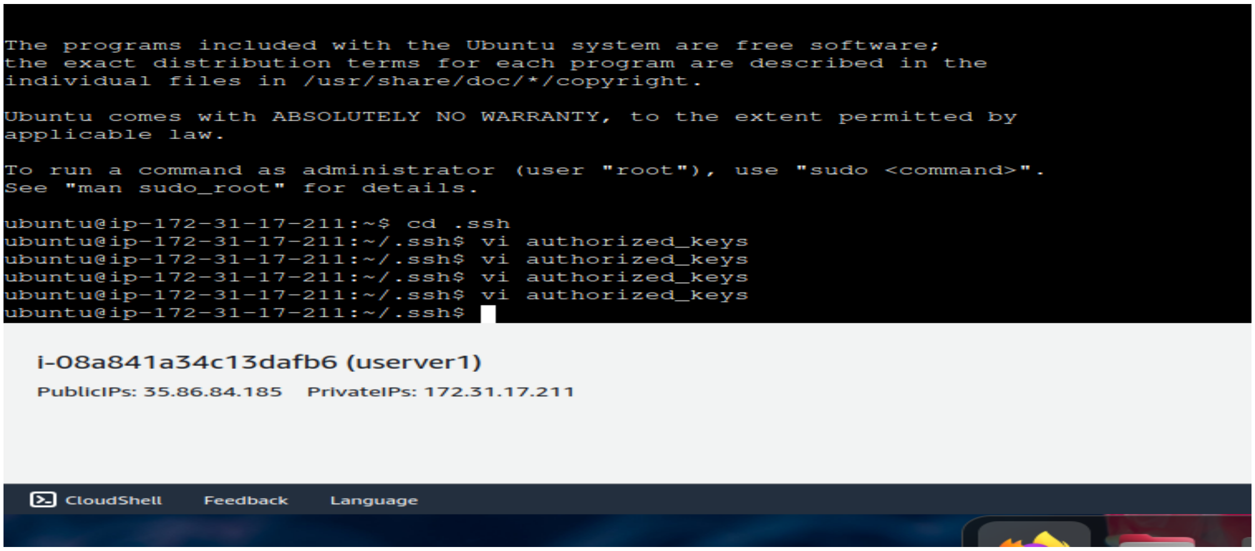


```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-22-26:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
/home/ubuntu/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (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:Dmy1pHXCdIZv2bbJzUehiX1PxHOcfLg1Rdl2L67e760 ubuntu@ip-172-31-22-26
The key's randomart image is:
+---[RSA 3072]----+
|        ..o   .=B|
|       o.o    .B@|
|        *..oo o=O|
|     . = ++.o+o.+|
|      = S. o *.+.|
|     . o    + + o|
|        .    . . |
|            ..  .|
|           .. E++|
+----[SHA256]-----+
ubuntu@ip-172-31-22-26:~/.ssh$
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

Keys are generated so let’s copy them to instances



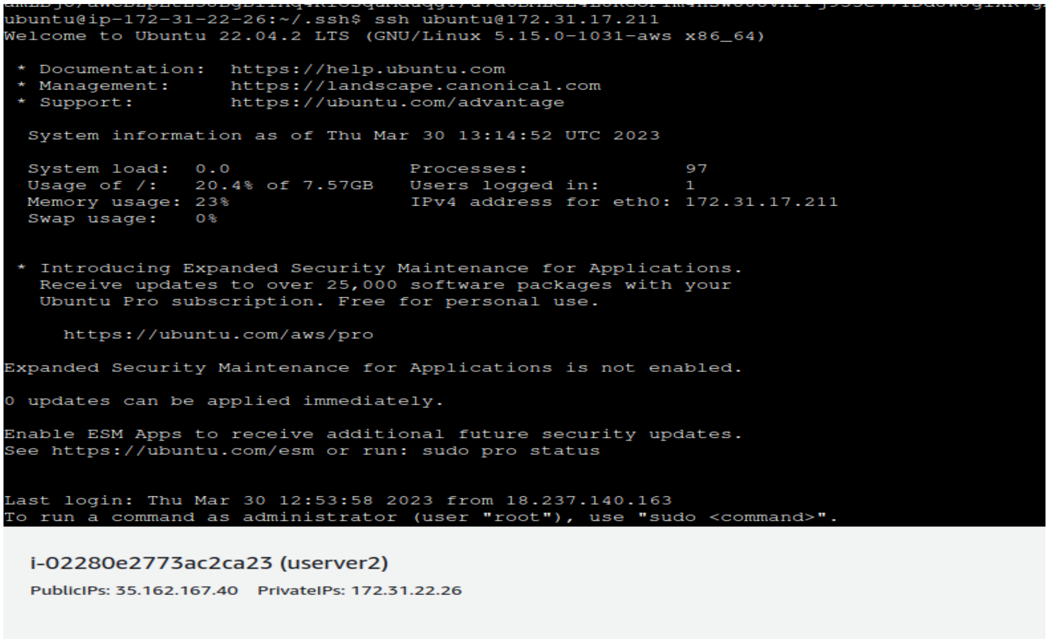
```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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-172-31-17-211:~$ cd .ssh
ubuntu@ip-172-31-17-211:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-17-211:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-17-211:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-17-211:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-17-211:~/.ssh$
```

i-08a841a34c13dafb6 (userver1)

PublicIPs: 35.86.84.185   PrivateIPs: 172.31.17.211

Keys have been copy using vim as you can see

3. Check whether userver2 connects to userver1 using ssh

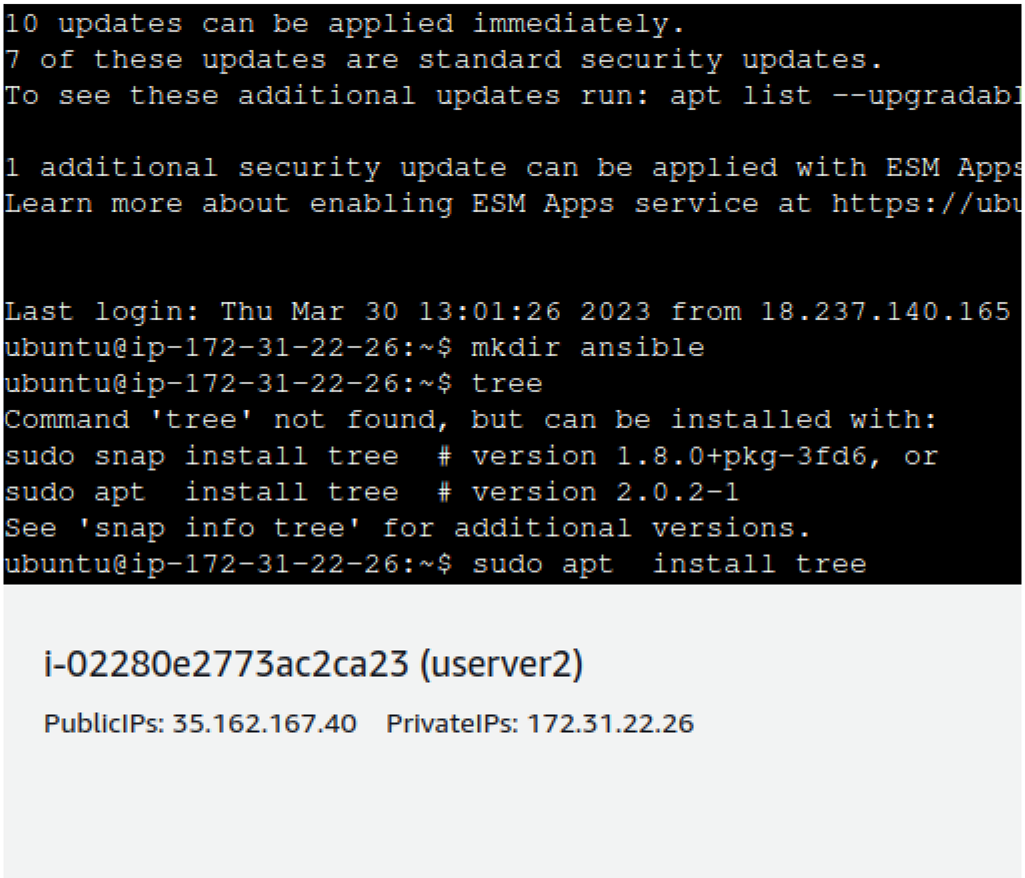


```
ubuntu@ip-172-31-22-26:~/.ssh$ ssh ubuntu@172.31.17.211
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

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

  System information as of Thu Mar 30 13:14:52 UTC 2023

  System load:  0.0               Processes:             97
  Usage of /:   20.4% of 7.57GB   Users logged in:       1
  Memory usage: 23%               IPv4 address for eth0: 172.31.17.211
  Swap usage:   0%


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

     https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

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


Last login: Thu Mar 30 12:53:58 2023 from 18.237.140.163
To run a command as administrator (user "root"), use "sudo <command>".
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

4. Make directory ansible on userver2 & write the playbook

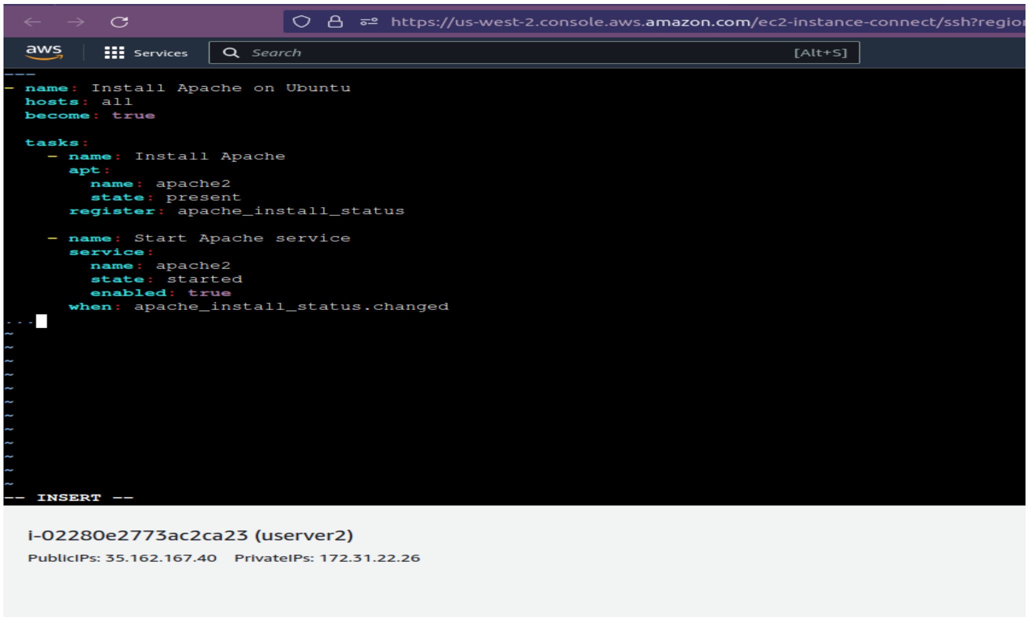


```
10 updates can be applied immediately.
7 of these updates are standard security updates.
To see these additional updates run: apt list --upgradabl

1 additional security update can be applied with ESM Apps
Learn more about enabling ESM Apps service at https://ubu


Last login: Thu Mar 30 13:01:26 2023 from 18.237.140.165
ubuntu@ip-172-31-22-26:~$ mkdir ansible
ubuntu@ip-172-31-22-26:~$ tree
Command 'tree' not found, but can be installed with:
sudo snap install tree  # version 1.8.0+pkg-3fd6, or
sudo apt  install tree  # version 2.0.2-1
See 'snap info tree' for additional versions.
ubuntu@ip-172-31-22-26:~$ sudo apt  install tree
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

YAML code (use command vim install-ansible.yaml)

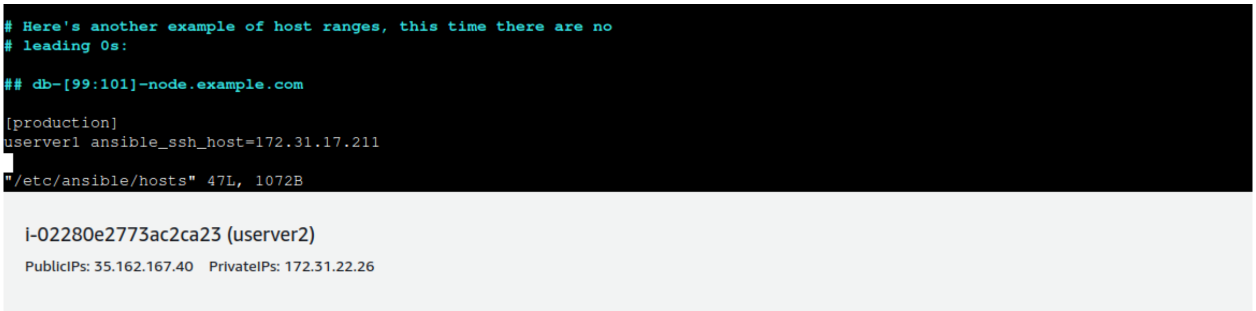

https://us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?regio
aws
Services
Search
[Alt+S]

```
---
- name: Install Apache on Ubuntu
  hosts: all
  become: true

  tasks:
    - name: Install Apache
      apt:
        name: apache2
        state: present
      register: apache_install_status

    - name: Start Apache service
      service:
        name: apache2
        state: started
        enabled: true
      when: apache_install_status.changed
...
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

5. Configure /etc/ansible/hosts (**To install on more servers just add their ip's in hosts & make sure to change the authorized_keys**)

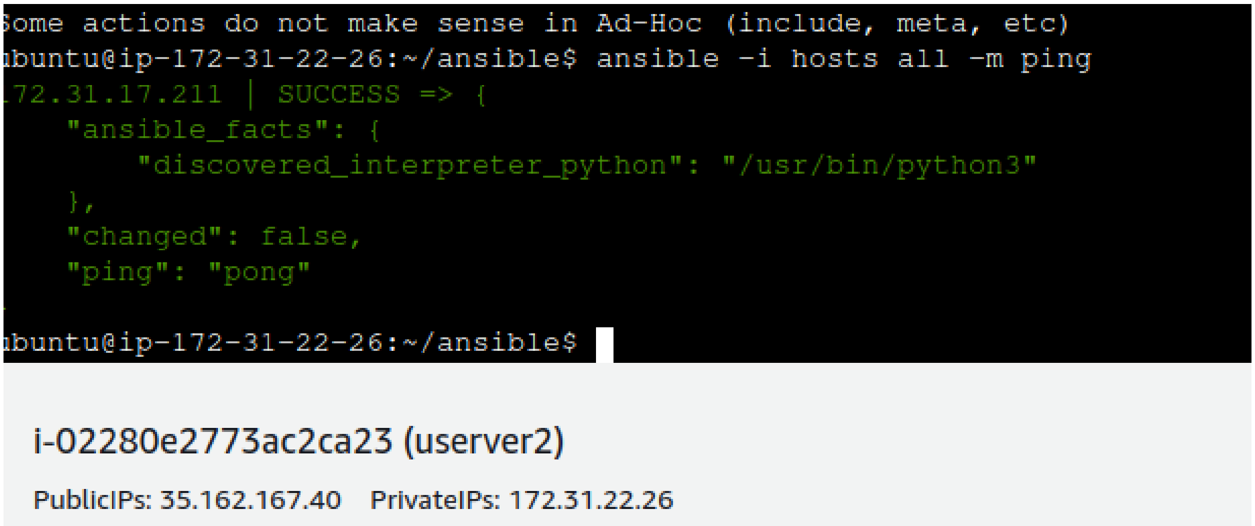


```
# Here's another example of host ranges, this time there are no
# leading 0s:

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

[production]
userver1 ansible_ssh_host=172.31.17.211

"/etc/ansible/hosts" 47L, 1072B
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

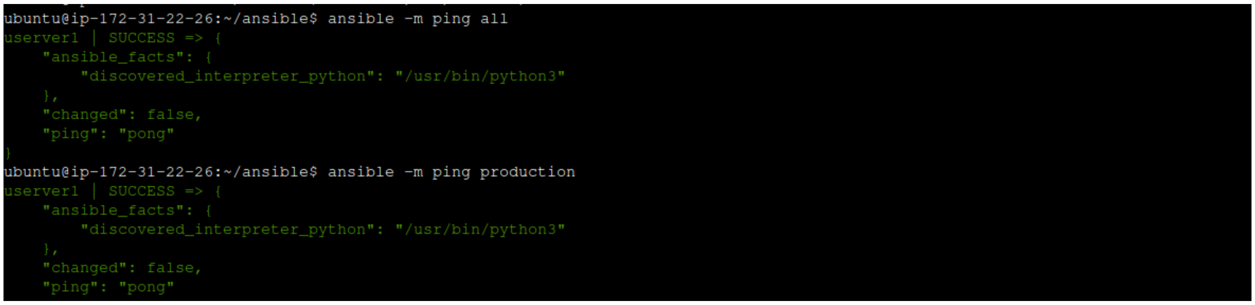
Let's verify hosts



```
Some actions do not make sense in Ad-Hoc (include, meta, etc)
ubuntu@ip-172-31-22-26:~/ansible$ ansible -i hosts all -m ping
172.31.17.211 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-22-26:~/ansible$
```

i-02280e2773ac2ca23 (userver2)

PublicIPs: 35.162.167.40   PrivateIPs: 172.31.22.26

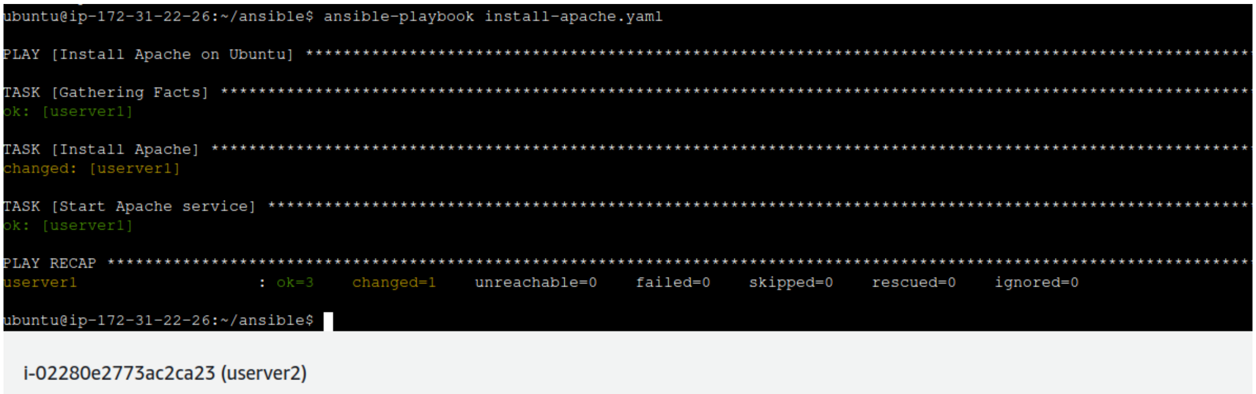


```
ubuntu@ip-172-31-22-26:~/ansible$ ansible -m ping all
userver1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-22-26:~/ansible$ ansible -m ping production
userver1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
```

6. Run the playbook using the command below

   ansible-playbook install-apache.yaml

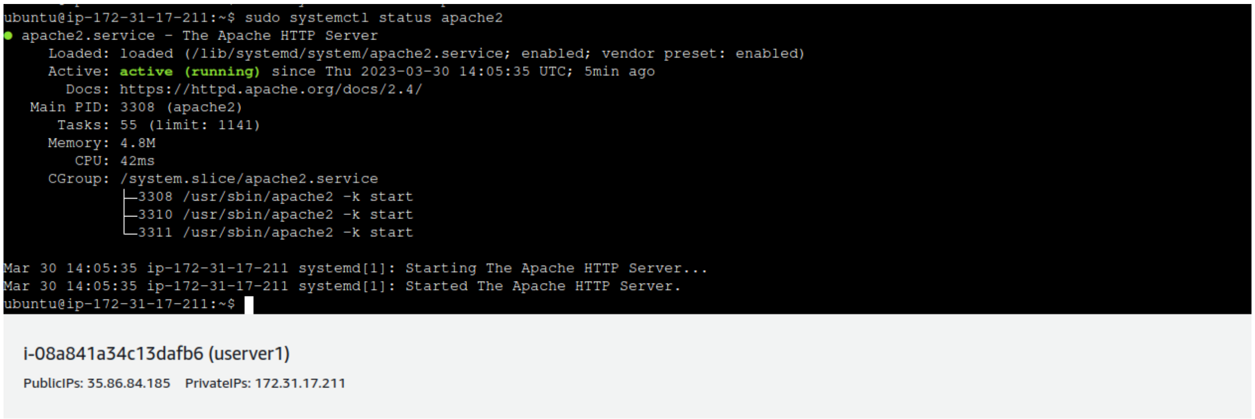
   Playing the playbook



```
ubuntu@ip-172-31-22-26:~/ansible$ ansible-playbook install-apache.yaml

PLAY [Install Apache on Ubuntu] ***************************************************************************************************

TASK [Gathering Facts] ************************************************************************************************************
ok: [userver1]

TASK [Install Apache] *************************************************************************************************************
changed: [userver1]

TASK [Start Apache service] *******************************************************************************************************
ok: [userver1]

PLAY RECAP ************************************************************************************************************************
userver1                   : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-22-26:~/ansible$
```

i-02280e2773ac2ca23 (userver2)

7. Check on userver1



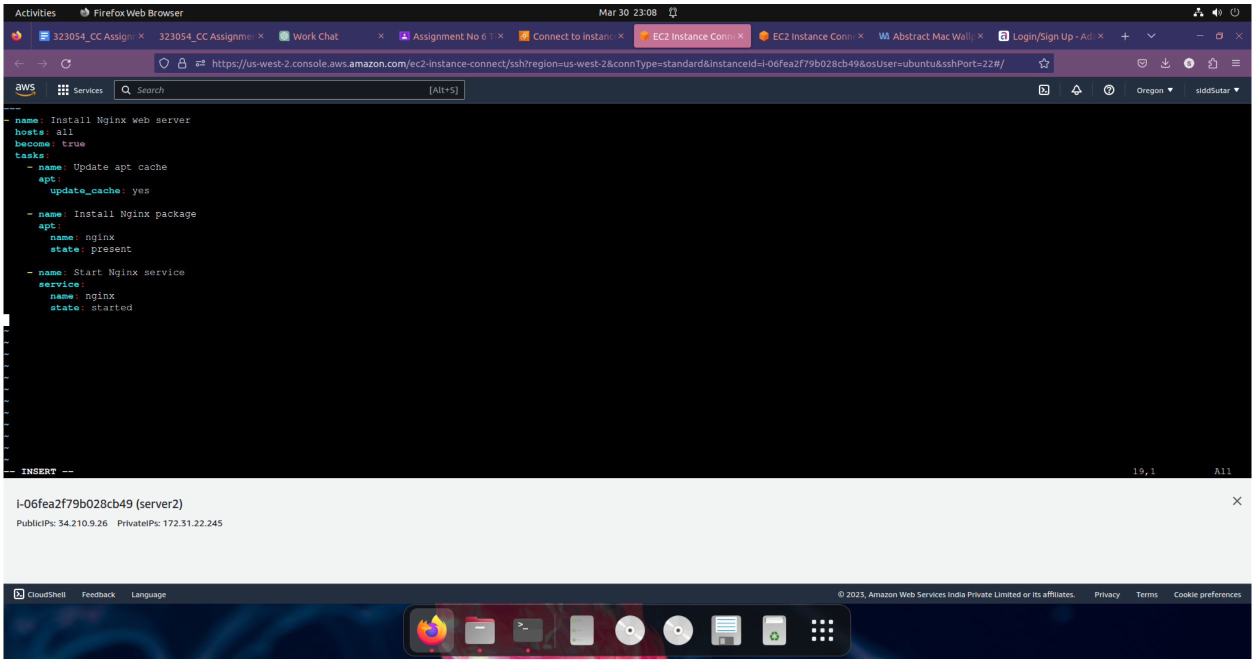
```
ubuntu@ip-172-31-17-211:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-03-30 14:05:35 UTC; 5min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3308 (apache2)
      Tasks: 55 (limit: 1141)
     Memory: 4.8M
        CPU: 42ms
     CGroup: /system.slice/apache2.service
             ├─3308 /usr/sbin/apache2 -k start
             ├─3310 /usr/sbin/apache2 -k start
             └─3311 /usr/sbin/apache2 -k start

Mar 30 14:05:35 ip-172-31-17-211 systemd[1]: Starting The Apache HTTP Server...
Mar 30 14:05:35 ip-172-31-17-211 systemd[1]: Started The Apache HTTP Server.
ubuntu@ip-172-31-17-211:~$
```

i-08a841a34c13dafb6 (userver1)

PublicIPs: 35.86.84.185 PrivateIPs: 172.31.17.211

Installing NGINX web server

1. Considering above steps are followed for two new instances (server 1 & server 2) so we will be just writing new playbook script for purpose (server 2 is master & server1 is slave just like before) we will start from saving the yaml script

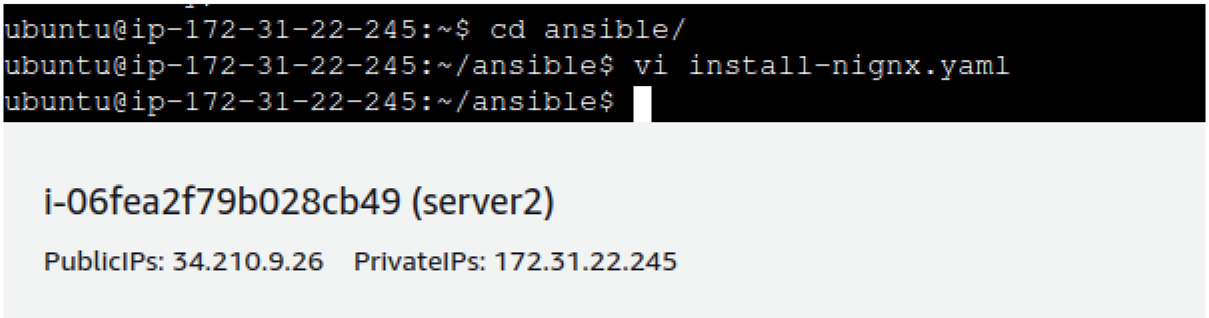


```
---
- name: Install Nginx web server
  hosts: all
  become: true
  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Install Nginx package
      apt:
        name: nginx
        state: present

    - name: Start Nginx service
      service:
        name: nginx
        state: started

~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                                    19,1          All
```

i-06fea2f79b028cb49 (server2)

PublicIPs: 34.210.9.26 PrivateIPs: 172.31.22.245

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

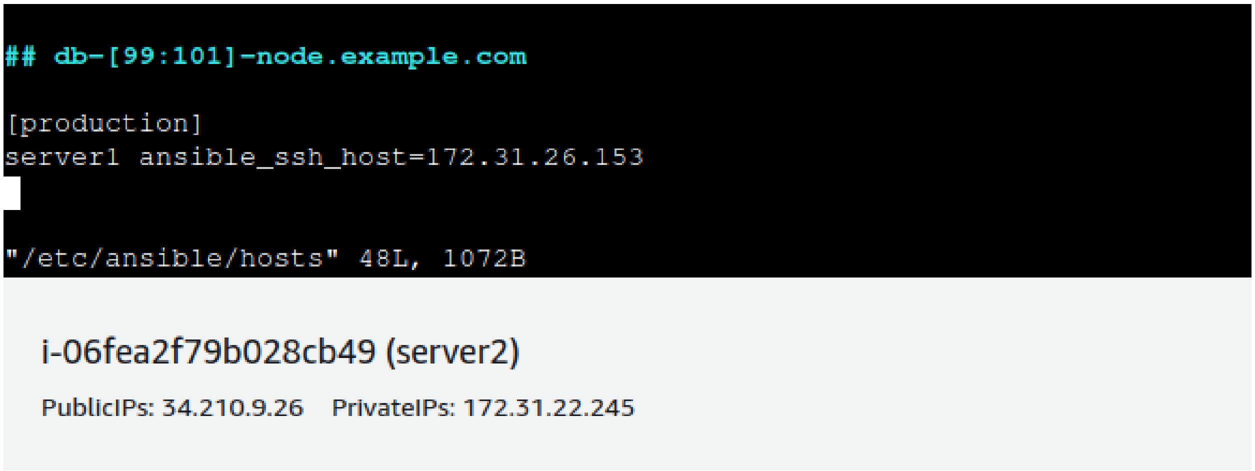


```
ubuntu@ip-172-31-22-245:~$ cd ansible/
ubuntu@ip-172-31-22-245:~/ansible$ vi install-nignx.yaml
ubuntu@ip-172-31-22-245:~/ansible$ 
```

i-06fea2f79b028cb49 (server2)

PublicIPs: 34.210.9.26   PrivateIPs: 172.31.22.245

2. Configure /etc/ansible/hosts (**To install on more servers just add their ip's in hosts & make sure to change the authorized_keys**)

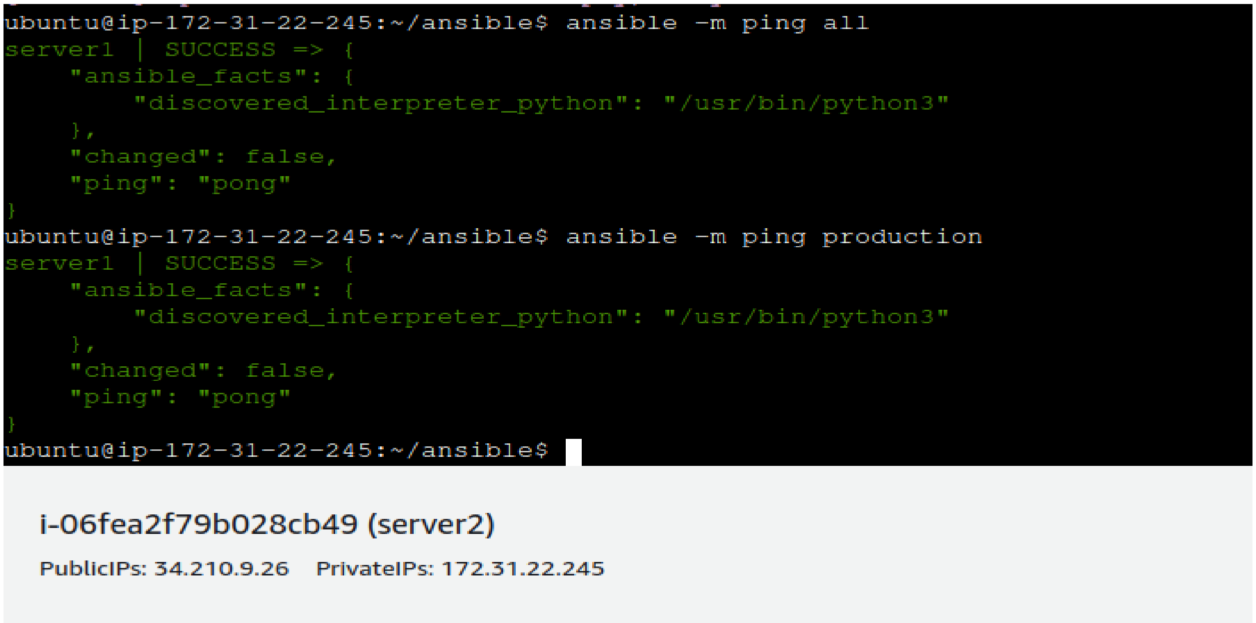


```
## db-[99:101]-node.example.com

[production]
server1 ansible_ssh_host=172.31.26.153


"/etc/ansible/hosts" 48L, 1072B
```

i-06fea2f79b028cb49 (server2)

PublicIPs: 34.210.9.26   PrivateIPs: 172.31.22.245

Verifying whether hosts file is done correctly or not

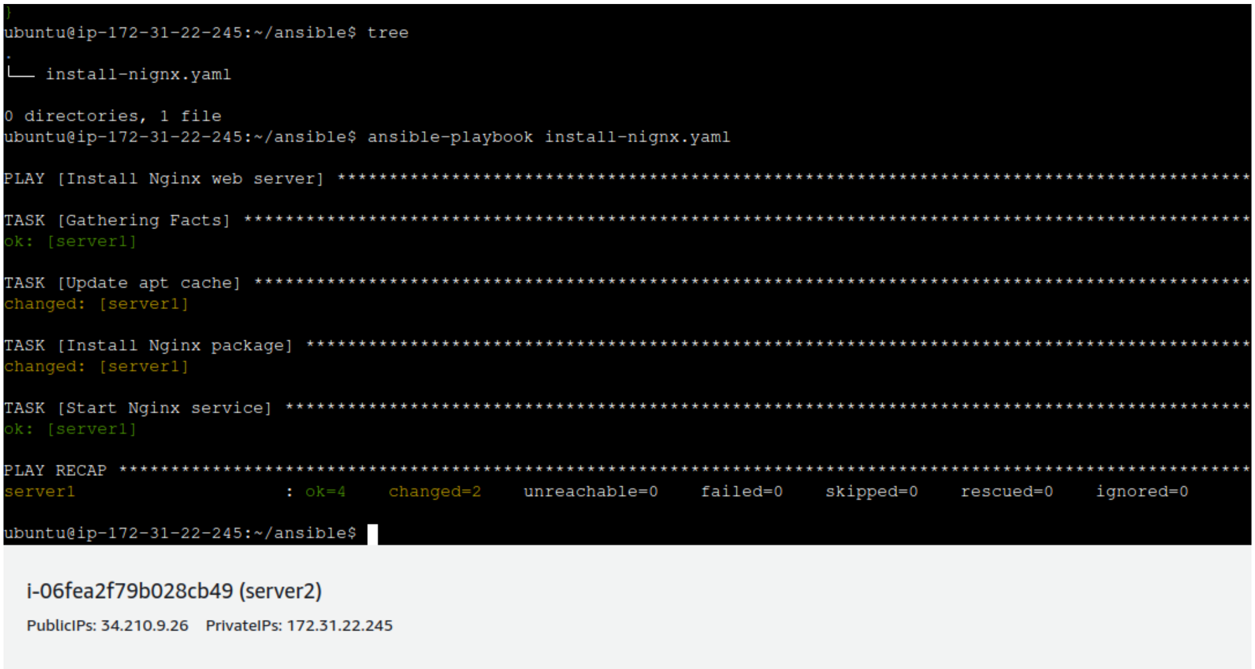


```
ubuntu@ip-172-31-22-245:~/ansible$ ansible -m ping all
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-22-245:~/ansible$ ansible -m ping production
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-22-245:~/ansible$ 
```

i-06fea2f79b028cb49 (server2)

PublicIPs: 34.210.9.26   PrivateIPs: 172.31.22.245

3. Playing the playbook

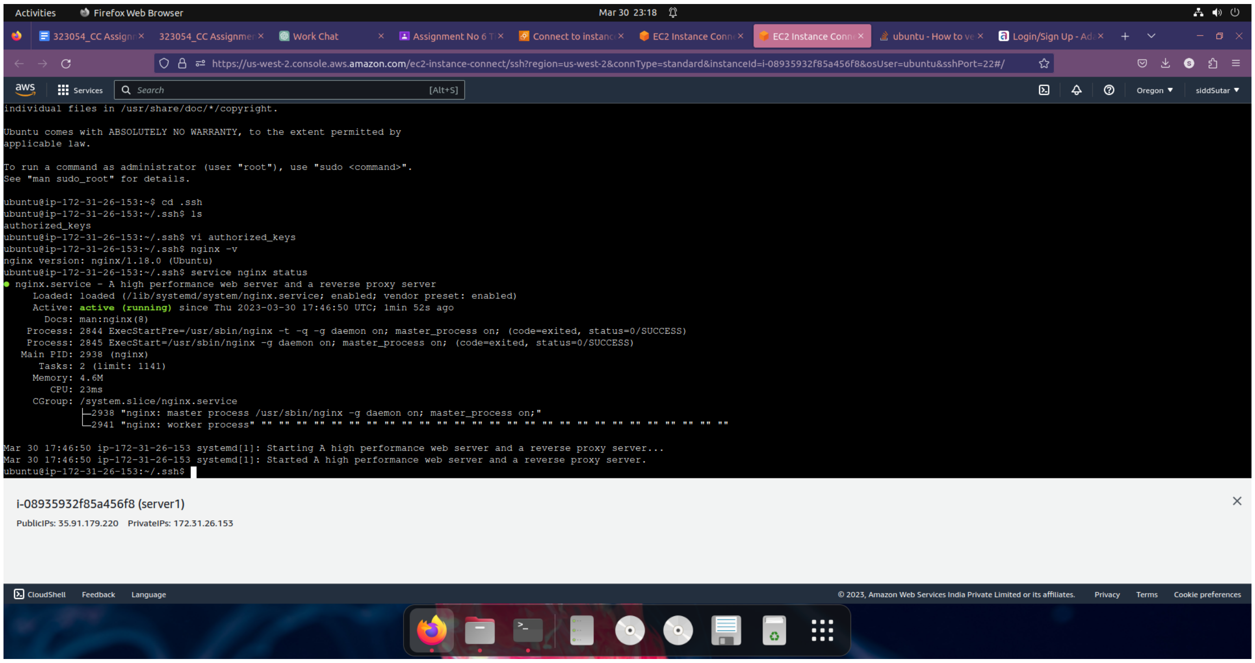


```
ubuntu@ip-172-31-22-245:~/ansible$ tree
.
└── install-nignx.yaml

0 directories, 1 file
ubuntu@ip-172-31-22-245:~/ansible$ ansible-playbook install-nignx.yaml

PLAY [Install Nginx web server] ***********************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [server1]

TASK [Update apt cache] *******************************************************************************************
changed: [server1]

TASK [Install Nginx package] **************************************************************************************
changed: [server1]

TASK [Start Nginx service] ****************************************************************************************
ok: [server1]

PLAY RECAP ********************************************************************************************************
server1                    : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-22-245:~/ansible$
```

i-06fea2f79b028cb49 (server2)

PublicIPs: 34.210.9.26   PrivateIPs: 172.31.22.245

4. Verify whether NGINX is started or not



```
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-172-31-26-153:~$ cd .ssh
ubuntu@ip-172-31-26-153:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-26-153:~/.ssh$ vi authorized_keys
ubuntu@ip-172-31-26-153:~/.ssh$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
ubuntu@ip-172-31-26-153:~/.ssh$ service nginx status
● 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 Thu 2023-03-30 17:46:50 UTC; 1min 52s ago
       Docs: man:nginx(8)
    Process: 2844 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2845 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2938 (nginx)
      Tasks: 2 (limit: 1141)
     Memory: 4.6M
        CPU: 23ms
     CGroup: /system.slice/nginx.service
             ├─2938 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2941 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""

Mar 30 17:46:50 ip-172-31-26-153 systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 30 17:46:50 ip-172-31-26-153 systemd[1]: Started A high performance web server and a reverse proxy server.
ubuntu@ip-172-31-26-153:~/.ssh$
```

i-08935932f85a456f8 (server1)

PublicIPs: 35.91.179.220   PrivateIPs: 172.31.26.153

CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

Conclusion:

→ Ansible along with it’s concepts is understood