Day 59: Ansible Project

Ansible playbooks are amazing, as you learned yesterday. What if you deploy a simple web app using ansible, sounds like a good project, right?

First, we need to set up the ansible master.

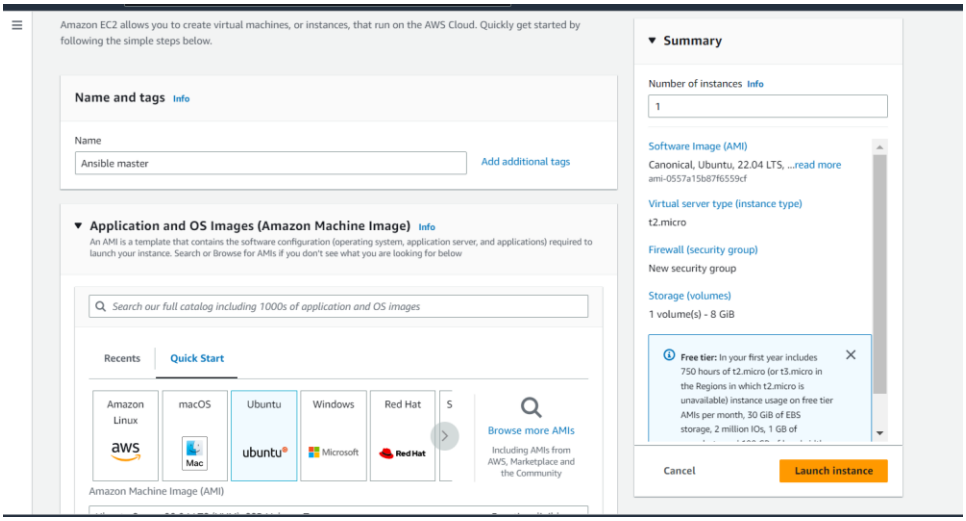
- Installation of Ansible on AWS EC2 (Master Node)

  sudo apt-add-repository ppa:ansible/ansible

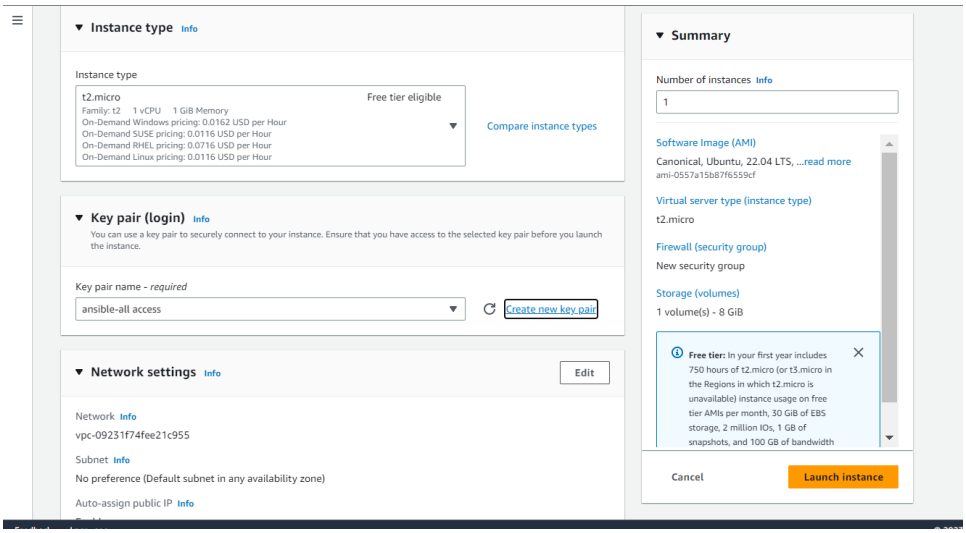
  sudo apt update

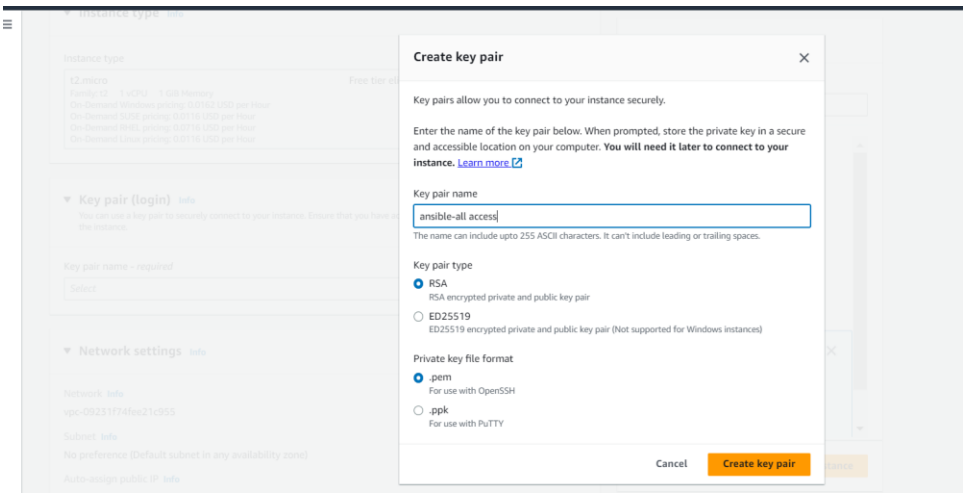
  sudo apt install ansible

Launch First EC2 Instance.

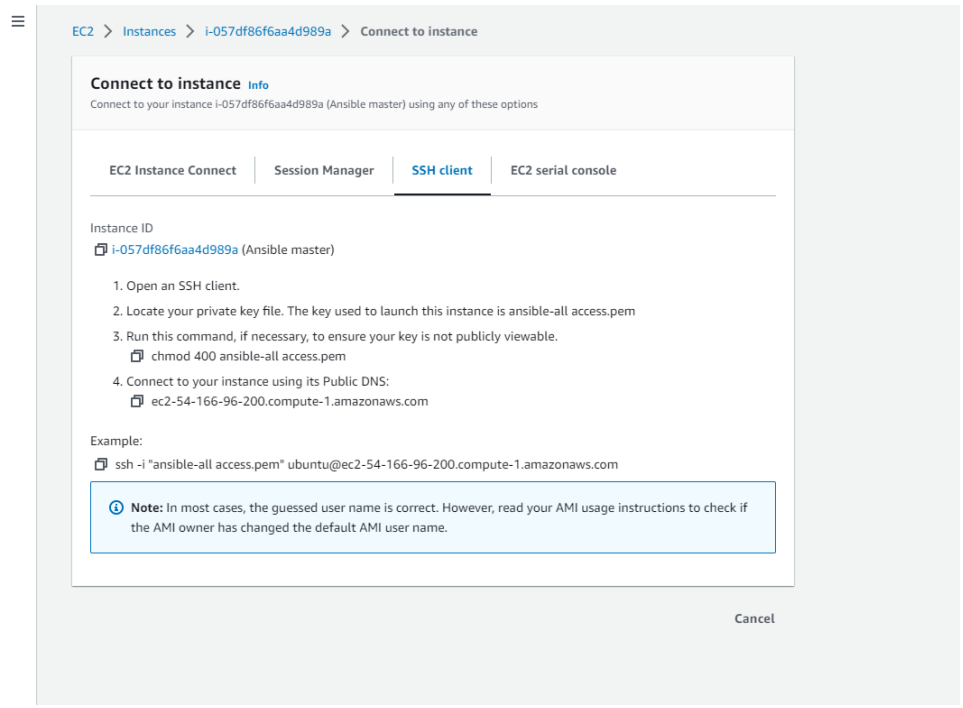


Create a new Key Pair.





Connect to the EC2 instance using SSH Client.

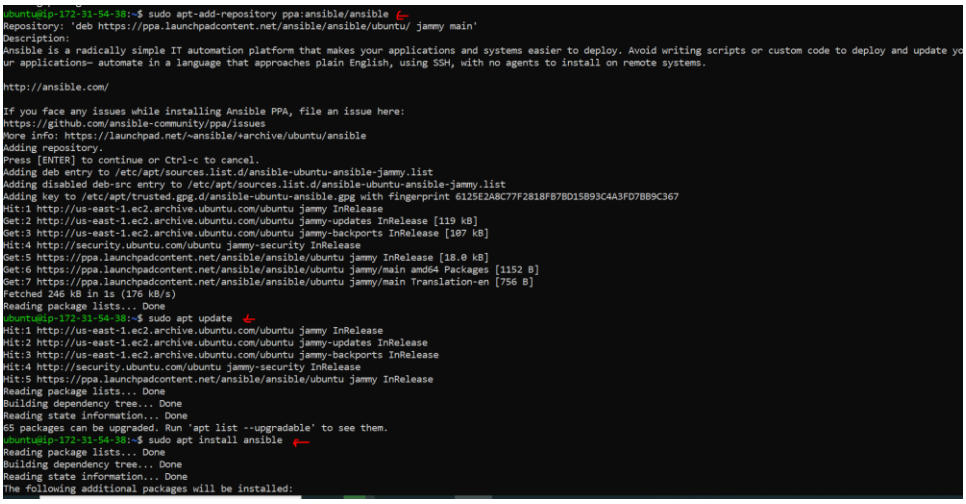


Update the Ansible master Server.

```
ubuntu@ip-172-31-54-38:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.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-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [949 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [205 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.8 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [684 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [107 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [584
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [895 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [179 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.4
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24.1 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [6312 B]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.7 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [9800 B]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [392 B]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [1
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [19.5 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [14.0 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [392
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [1
```

Use Above mentioned command to setup Ansible Mater node.

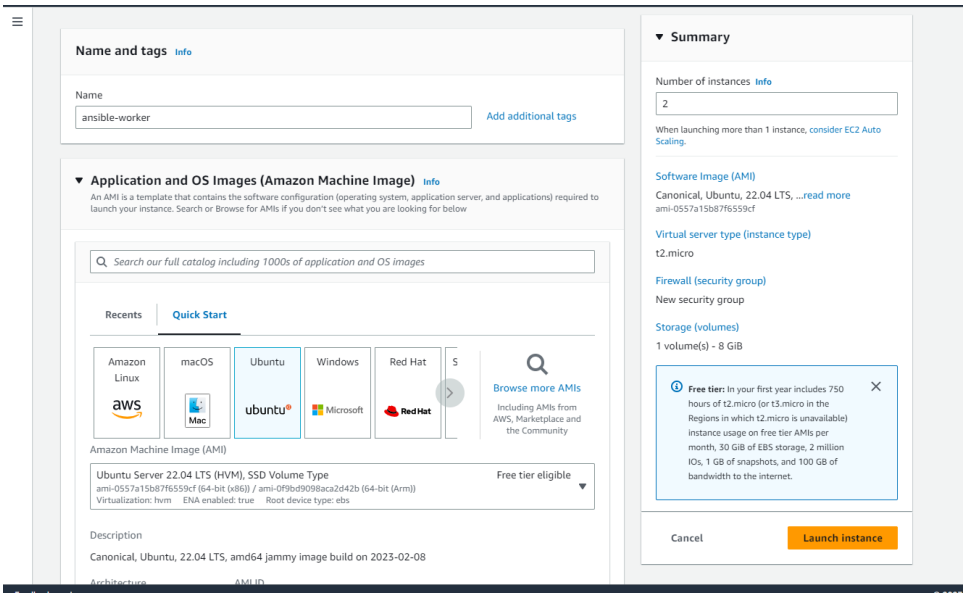
Run above all commands.



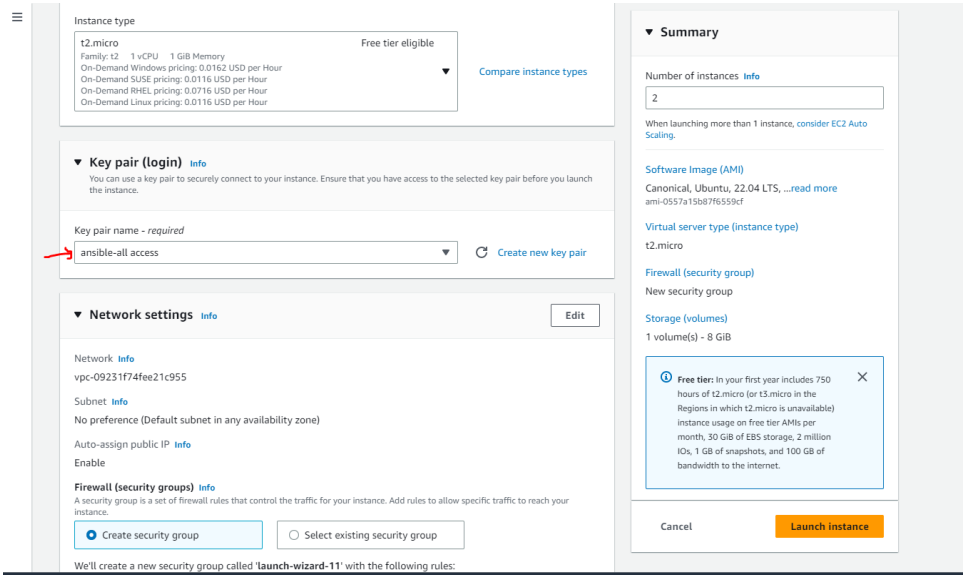
Hit below command to verify, whether our Ansible master node is created successfully or not.

```
ubuntu@ip-172-31-54-38:~$ cat /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

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

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

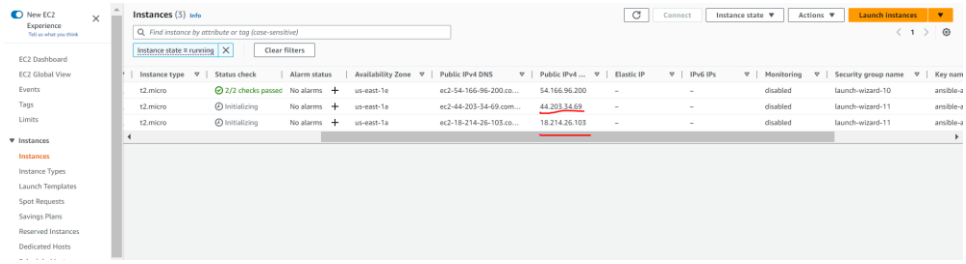
Launch 2 Ansible Worker node.



Use the **Same Key-pair**, you used previously in Ansible Master.



Copy both Ansible Worker Ip Address.



[Ansible Default Special Variables]

- read more about Hosts file sudo nano /etc/ansible/hosts ansible-inventory --list –y

The simplest inventory is a single file with a list of hosts and groups. The default location for this file is /etc/ansible/hosts. You can specify a different inventory file at the command line using the -i <path> option or in configuration using inventory.

Go to Ansible inventory location

Sudo nano /etc/ansible/hosts

```
[servers]
Server1 ansible_host
<Ansible_Worker1_ip_address>
Server2 ansible_host
<Ansible_Worker2_ip_address>
```

Explanation: -

Servers: - This is a Group Name

ansible_host : Default Ansible variable. Don’t change, use as it is

```
GNU nano 6.2

## 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

# 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

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

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

[servers]
server1 ansible_host=35.172.234.42
server2 ansible_host=3.237.92.220

[servers:vars]
ansible_ssh_private_key_file =/home/ubuntu/.ssh/ansible-all-access.pem
```

```
#   - 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

# 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

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

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

[servers]
server1 ansible_host=35.172.234.42
server2 ansible_host=3.237.92.220

[servers:vars]
ansible_ssh_private_key_file =/home/ubuntu/.ssh/ansible-all-access.pem
```

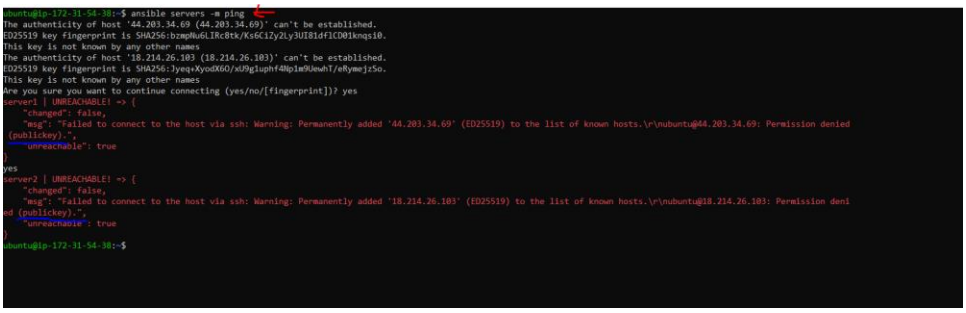
- Setting up 2 more EC2 instances with same Private keys as the previous instance (Node)

Create a Connection between ansible master and ansible worker using ping command.

ansible <group_name> -m ping

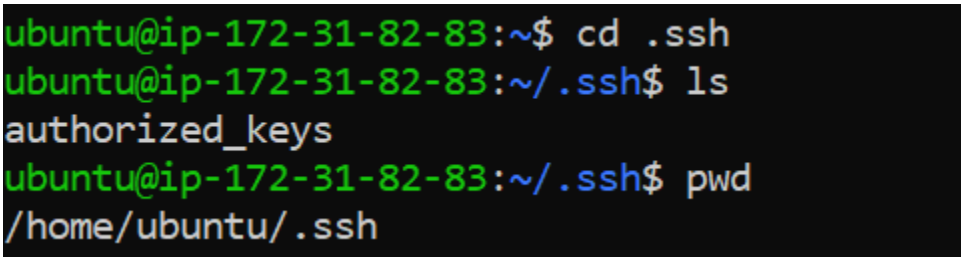
-m means module

See You Got Authentication Error.



The Above Error occurs because of the public key is not available on ansible worker.

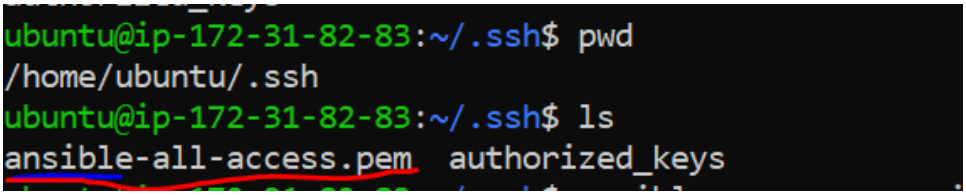
Goto .ssh path and copy the path



Using **SCP command**, send local public key to ansible master remote path

See, in ansible master server.

Public is Received.



Ok try once.

Again, you are facing the same issue, and this time the reason is different.

The reason is the current user has don't have any permission to execute or use the public key.

Use **chmod** command to give permission.

After giving this permission, you have successfully connected to the ansible worker.

```
ubuntu@ip-172-31-82-83:~$ ansible servers -m ping
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Create a Simple Ansible-playbook to install nginx and run the Web application

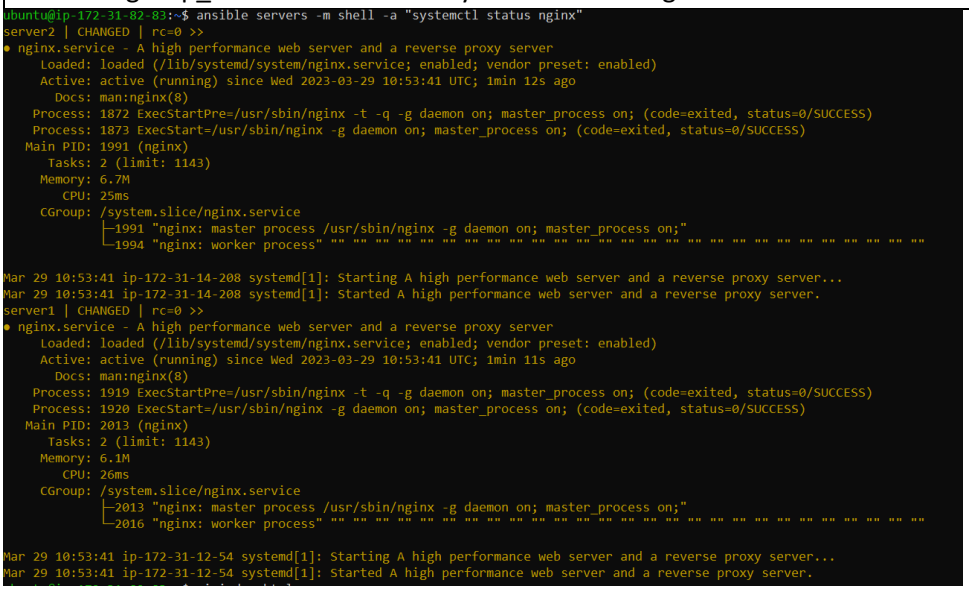
```
---
- name: create naginx and run a simple web application
  hosts: servers
  become: yes
  tasks:
    - name: install nginx
      apt:
        name: nginx
        state: latest
    - name: start nginx
      service:
        name: nginx
        state: started
    - name: copy index file to nginx default path
      ansible.builtin.copy:
        src: ./index.html
        dest: /var/www/html/
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

Create a custom index.html

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Example</title>
    </head>
    <body>
        <p>This is an example of a simple HTML page with one paragraph.</p>
    </body>
</html>
```

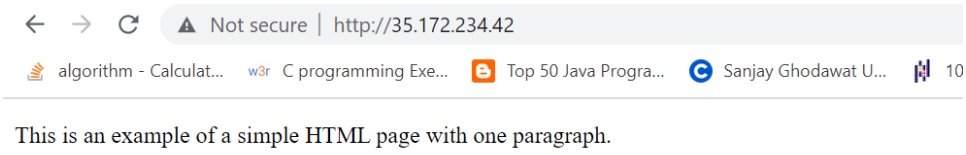
Check Your Nginx is running on distributed server, using below command

Ansible <group_name> -m shell –a “systemctl status nginx”



Copy the ansible-worker public Ip address and hit on browser

Your Webapp is running



This is an example of a simple HTML page with one paragraph.

happy learning...