

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.

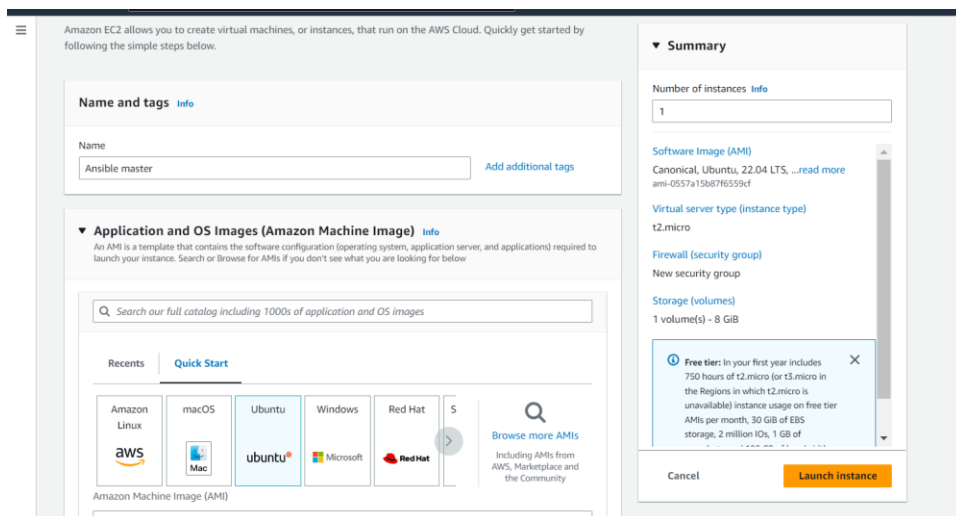
- Installation of Ansible on AWS EC2 (Master Node)

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

```
sudo apt update
```

```
sudo apt install ansible
```

Launch First EC2 Instance.



Create a new Key Pair.

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

▼

Compare instance types

▼ 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-all access

↻ Create new key pair

▼ Network settings [Info](#)

Network [Info](#)

vpc-09231f74fee21c955

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Edit

▼ Summary

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0557a15b87f6559cf

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 includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GiB of bandwidth

Cancel

Launch instance

▼ Instance type [Info](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory
On-Demand Windows pricing: 0.0162 USD per Hour
On-Demand SUSE pricing: 0.0116 USD per Hour
On-Demand RHEL pricing: 0.0716 USD per Hour
On-Demand Linux pricing: 0.0116 USD per Hour

▼

Compare instance types

▼ 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

Select

▼ Network settings [Info](#)

Network [Info](#)

vpc-09231f74fee21c955

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. 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](#)

Key pair name

ansible-all access

The name can include upto 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 (Not supported for Windows instances)

Private key file format

☒ .pem

For use with OpenSSH

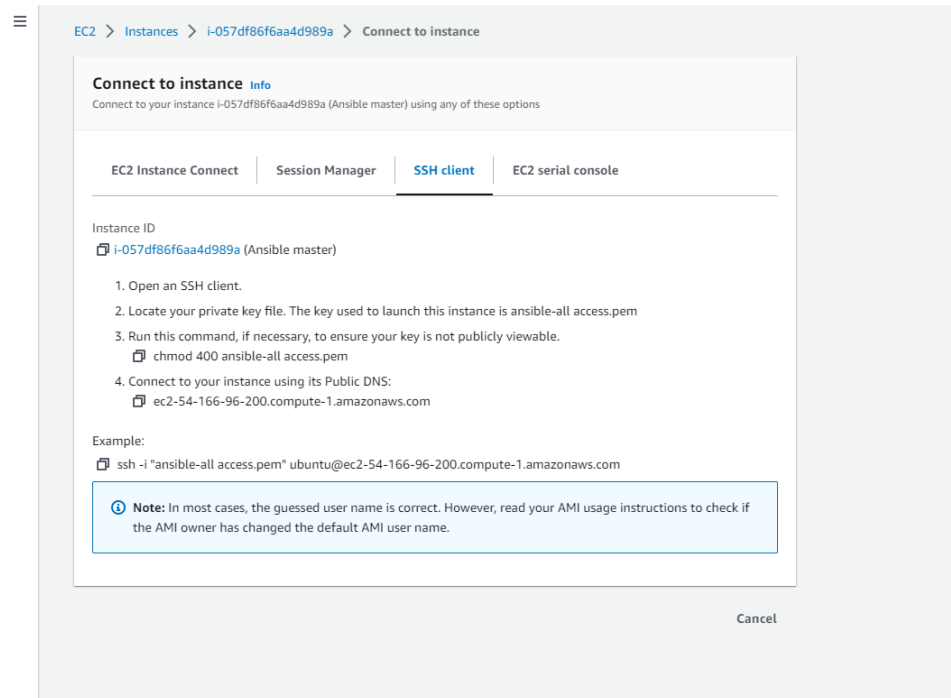
☐ .ppk

For use with PuTTY

Cancel

Create 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 B]
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 kB]
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 B]
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 B]
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 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [1 B]
```

Use Above mentioned command to setup Ansible Mater node.

Run above all commands.

```

ubuntu@ip-172-31-54-38:~$ sudo apt-add-repository 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. Avoid writing scripts or custom code to deploy and update your applications—automate in a language 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
None info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Press [ENTER] to continue or Ctrl-C to cancel.
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 6125E2ABC77F2818F878015893CA43FD7889C367
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]
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease [18.9 kB]
Get:6 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 Packages [1152 B]
Get:7 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main Translation-en [756 B]
Fetched 246 kB in 1s (206 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-54-38:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
65 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-54-38:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:

```

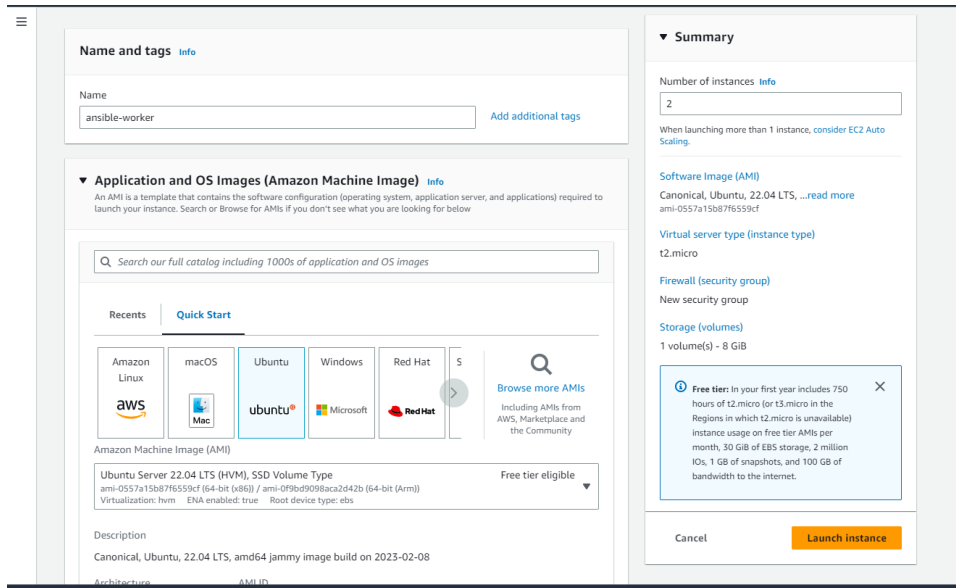
Hit below command to verify, whether our Ansible master node is created successfully or not.

```

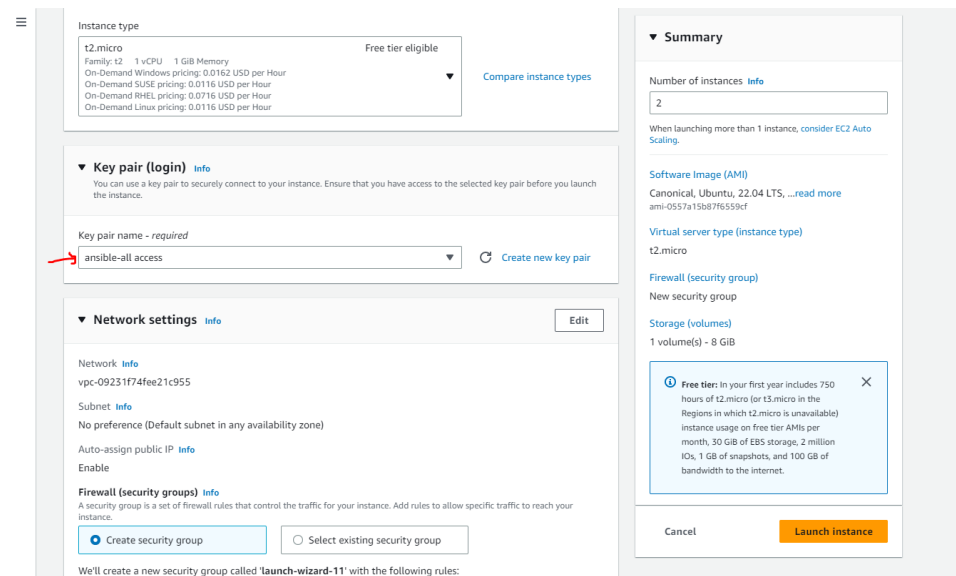
ubuntu@ip-172-31-54-38:~$ cat /etc/ansible/hosts
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - 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 'webserver' group:
## [webserver]
## 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 'dbserver' group:
## [dbserver]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net

```

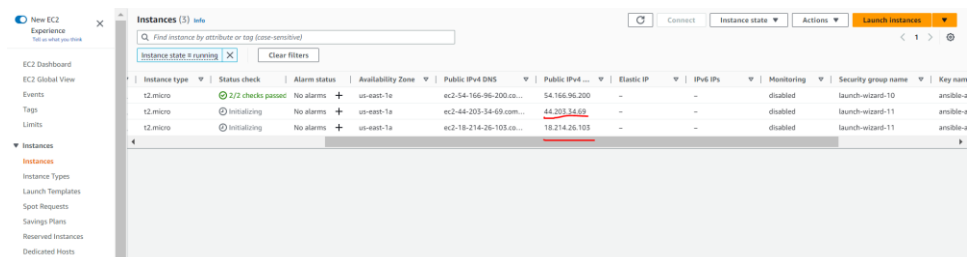
Launch 2 Ansible Worker node.



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 0.2

## 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.


```

ubuntu@ip-172-31-54-30:~$ ansible servers -m ping
The authenticity of host '44.203.34.69 (44.203.34.69)' can't be established.
ED25519 key fingerprint is SHA256:bzuph0dLlR0ts/Ks6C12y2j3UfE1eF1CD01kmp10.
This key is not known by any other names
The authenticity of host '18.214.26.103 (18.214.26.103)' can't be established.
ED25519 key fingerprint is SHA256:jyeqXyosX00/s0p9luhf40p1eR0whf/ekymcj5o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
server1 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Warning: Permanently added '44.203.34.69' (ED25519) to the list of known hosts.\n\nubuntu@44.203.34.69: Permission denied (publickey).",
  "unreachable": true
}
yes
server2 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Warning: Permanently added '18.214.26.103' (ED25519) to the list of known hosts.\n\nubuntu@18.214.26.103: Permission denied (publickey).",
  "unreachable": true
}
ubuntu@ip-172-31-54-30:~$

```

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

Goto `.ssh` path and copy the path

```

ubuntu@ip-172-31-82-83:~$ cd .ssh
ubuntu@ip-172-31-82-83:~/ssh$ ls
authorized_keys
ubuntu@ip-172-31-82-83:~/ssh$ pwd
/home/ubuntu/.ssh

```

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

```

Rushikesh@DESKTOP-02SE6R3 MINGW64 ~/Downloads ((997F144...))
$ scp -i "ansible-all-access.pem" ansible-all-access.pem ubuntu@ec2-3-87-143-201.compute-1.amazonaws.com:/home/ubuntu/.ssh
ansible-all-access.pem 100% 1674 5.6KB/s 00:00

```

See, in ansible master server.

Public is Received.

```

ubuntu@ip-172-31-82-83:~/ssh$ pwd
/home/ubuntu/.ssh
ubuntu@ip-172-31-82-83:~/ssh$ ls
ansible-all-access.pem  authorized_keys

```

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.

```

ubuntu@ip-172-31-82-83:~$ ansible servers -m ping
server1 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: WARNING: UNPROTECTED PRIVATE KEY FILE!\n\nPermissions 0644 for '/home/ubuntu/.ssh/ansible-all-access.pem' are too open.\n\nIt is required that you\nprivate key files are NOT accessible by others.\n\nThis private key will be ignored.\n\nLoad key '/home/ubuntu/.ssh/ansible-all-access.pem': bad permissions\n\nubuntu@44.203.34.69: Permission denied (publickey).",
  "unreachable": true
}
server2 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: WARNING: UNPROTECTED PRIVATE KEY FILE!\n\nPermissions 0644 for '/home/ubuntu/.ssh/ansible-all-access.pem' are too open.\n\nIt is required that you\nprivate key files are NOT accessible by others.\n\nThis private key will be ignored.\n\nLoad key '/home/ubuntu/.ssh/ansible-all-access.pem': bad permissions\n\nubuntu@18.214.26.103: Permission denied (publickey).",
  "unreachable": true
}
ubuntu@ip-172-31-82-83:~$

```

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

```
<!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"

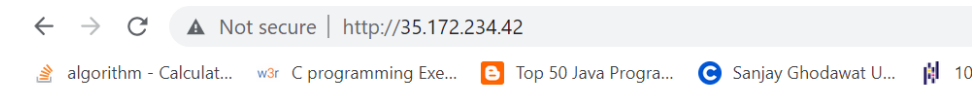
```
ubuntu@ip-172-31-82-83:~$ ansible servers -m shell -a "systemctl status nginx"
server2 | CHANGED | rc=0 >>
• nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-03-29 10:53:41 UTC; 1min 12s ago
  Docs: man:nginx(8)
  Process: 1872 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 1873 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 1991 (nginx)
  Tasks: 2 (limit: 1143)
  Memory: 6.7M
  CPU: 25ms
  CGroup: /system.slice/nginx.service
          └─1991 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1994 "nginx: worker process"

Mar 29 10:53:41 ip-172-31-14-208 systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 29 10:53:41 ip-172-31-14-208 systemd[1]: Started A high performance web server and a reverse proxy server.
server1 | CHANGED | rc=0 >>
• 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 Wed 2023-03-29 10:53:41 UTC; 1min 11s ago
  Docs: man:nginx(8)
  Process: 1919 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 1920 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 2013 (nginx)
  Tasks: 2 (limit: 1143)
  Memory: 6.1M
  CPU: 26ms
  CGroup: /system.slice/nginx.service
          └─2013 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─2016 "nginx: worker process"

Mar 29 10:53:41 ip-172-31-12-54 systemd[1]: Starting A high performance web server and a reverse proxy server...
Mar 29 10:53:41 ip-172-31-12-54 systemd[1]: Started A high performance web server and a reverse proxy server.
```

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...