Day 55: Understanding Configuration Management with Ansible

What's this Ansible?

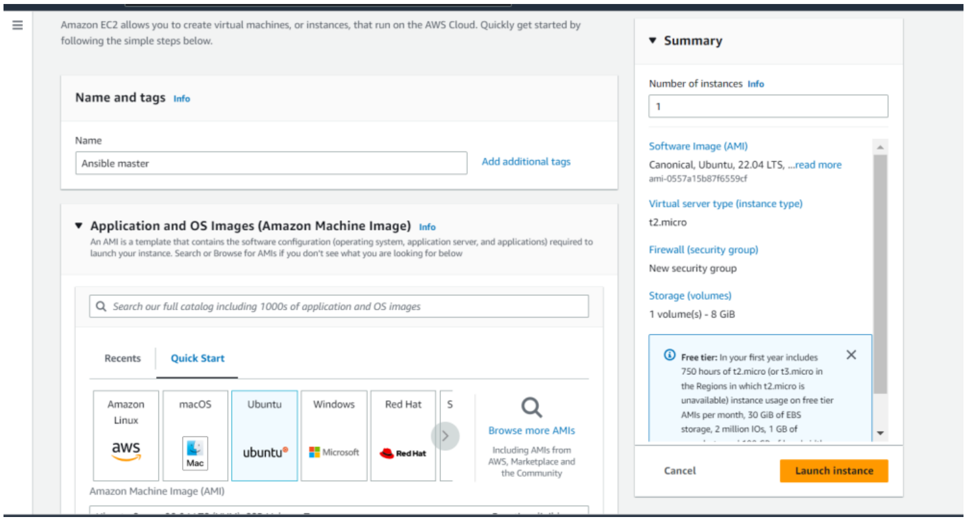
Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, interservice orchestration, and provisioning

Task-01

- 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.



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

Name and tags

Name: Ansible master

Application and OS Images (Amazon Machine Image)

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

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more ami-0557a15b87f6559cf

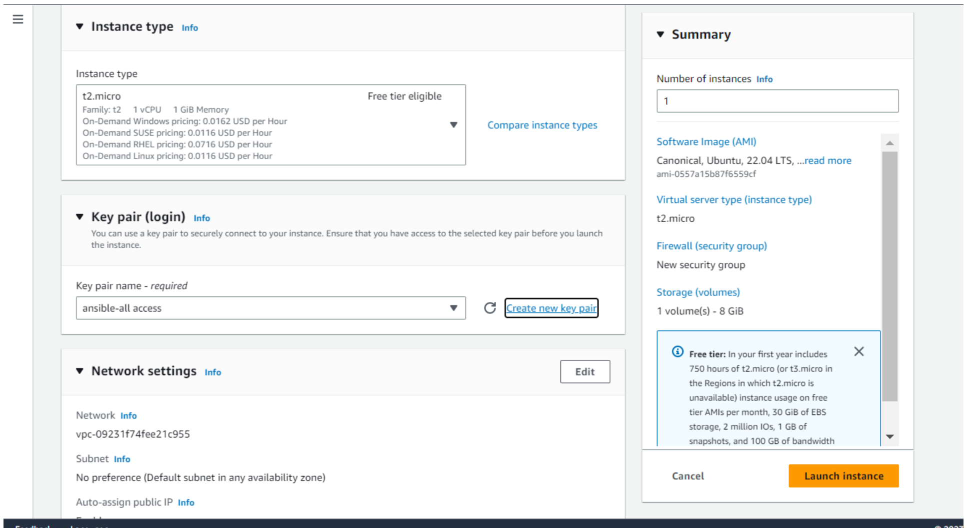
Virtual server type (instance type): t2.micro

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 IOs, 1 GB of

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

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

Network settings Info

Network Info: vpc-09231f74fee21c955

Subnet Info: No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Summary

Number of instances Info: 1

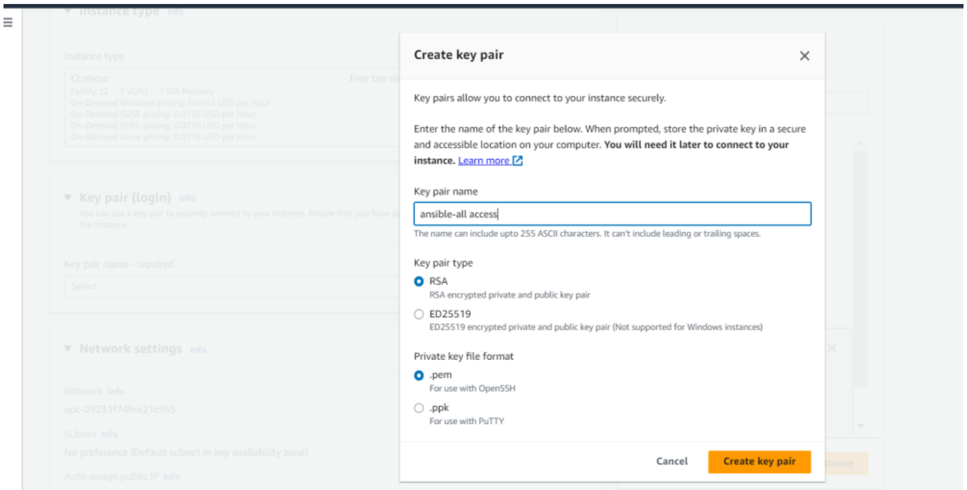
Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0557a15b87f6559cf

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth



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

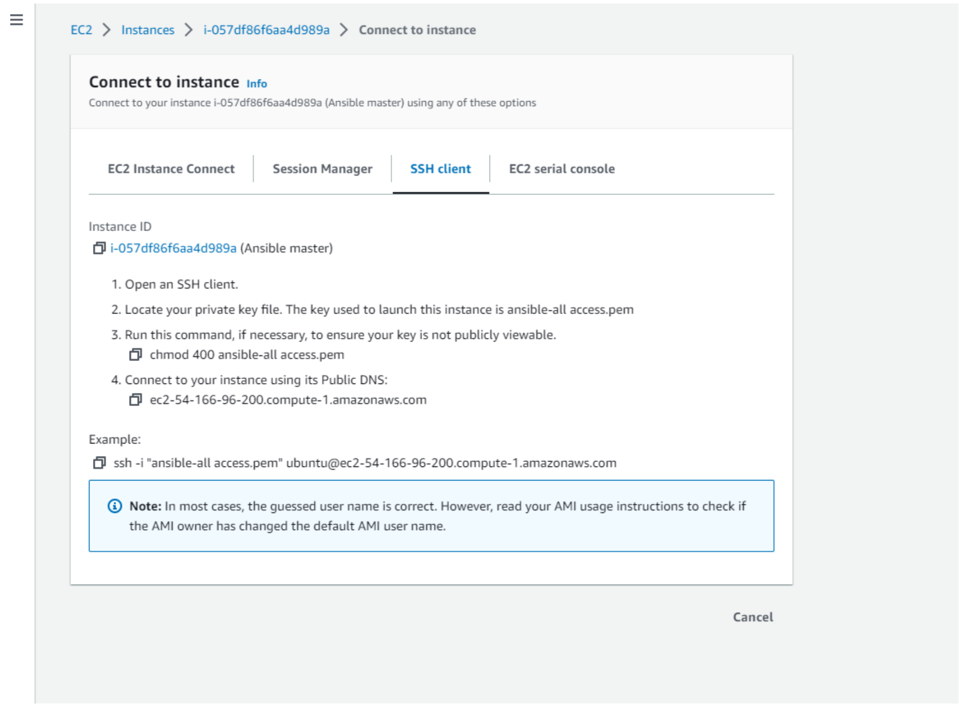
○ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

Connect to the EC2 instance using SSH Client.



Connect to instance Info

Connect to your instance i-057df86f6aa4d989a (Ansible master) using any of these options

Instance ID

i-057df86f6aa4d989a (Ansible master)

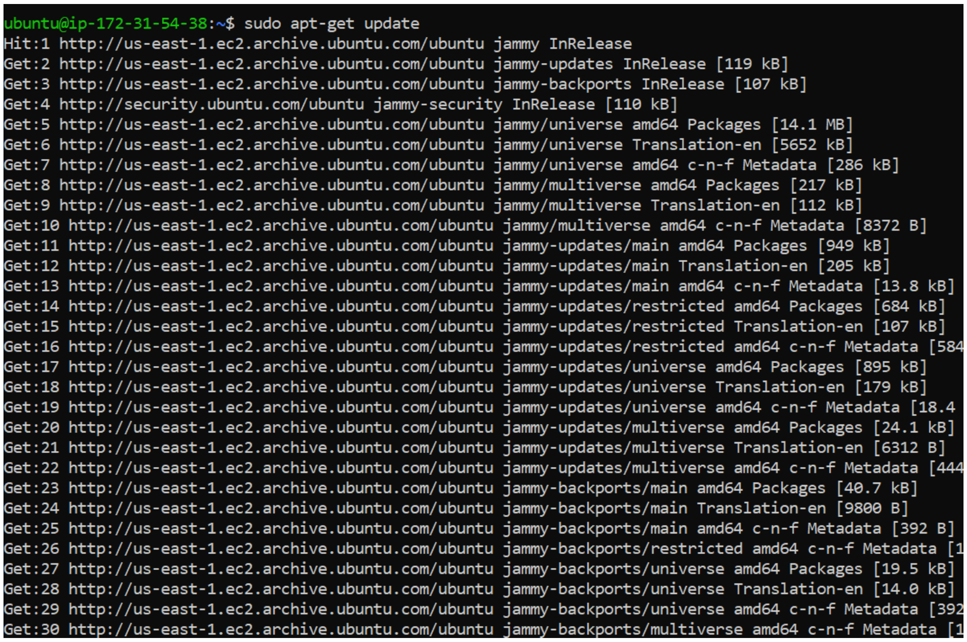
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ansible-all access.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 ansible-all access.pem
4. Connect to your instance using its Public DNS:
   ec2-54-166-96-200.compute-1.amazonaws.com

Example:

ssh -i "ansible-all access.pem" ubuntu@ec2-54-166-96-200.compute-1.amazonaws.com

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

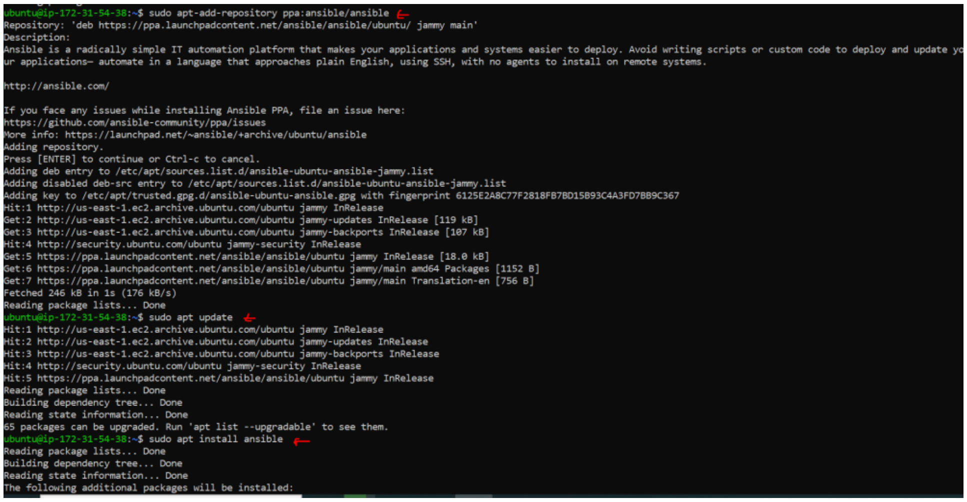
Update the Ansible master Server.



```
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
```

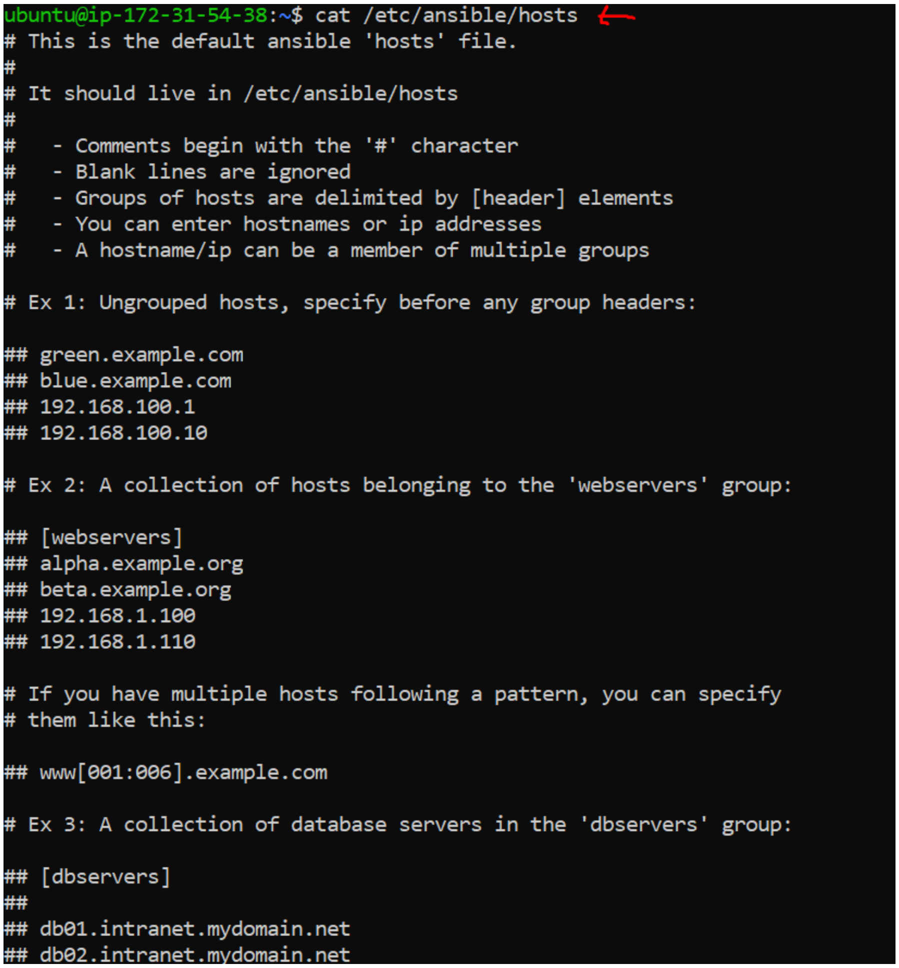
Use Above mentioned command to setup Ansible Mater node.

Run above all commands.



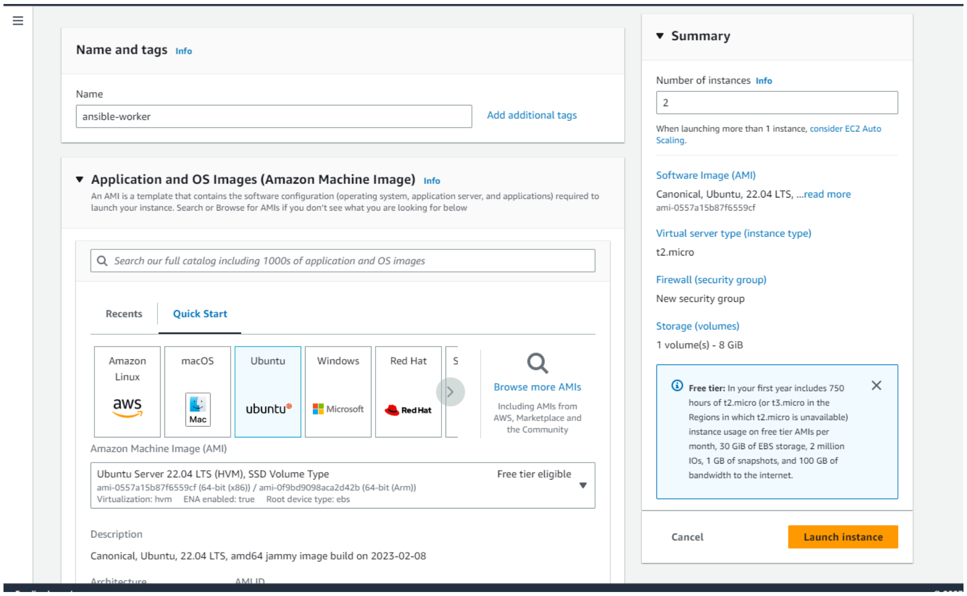
```
ubuntu@ip-172-31-54-38:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update yo
ur 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
More 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 6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
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 [18.0 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 (176 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 '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.



Name and tags Info

Name: ansible-worker

Add additional tags

Application and OS Images (Amazon Machine Image) Info

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

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0557a15b87f6559cf (64-bit (x86)) / ami-0f9bd9098aca2d42b (64-bit (Arm))
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-02-08

Architecture   AMI ID

Summary

Number of instances Info: 2

When launching more than 1 instance, consider EC2 Auto Scaling.

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0557a15b87f6559cf

Virtual server type (instance type): t2.micro

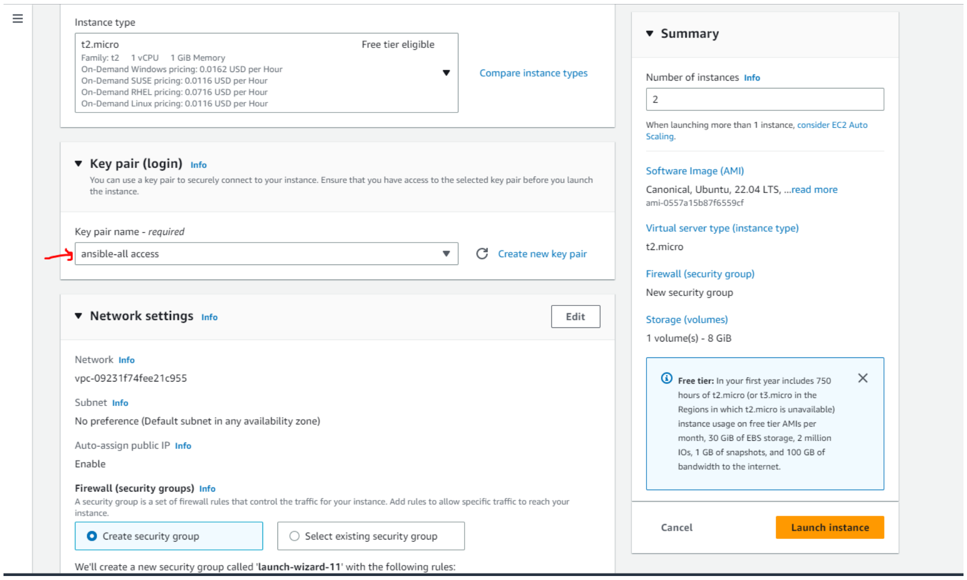
Firewall (security group): New security group

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance

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



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 — Edit

Network Info: vpc-09231f74fee21c955

Subnet Info: No preference (Default subnet in any availability zone)

Auto-assign public IP Info: Enable

Firewall (security groups) Info

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

● Create security group   ○ Select existing security group

We'll create a new security group called '**launch-wizard-11**' with the following rules:

Summary

Number of instances Info: 2

When launching more than 1 instance, consider EC2 Auto Scaling.

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0557a15b87f6559cf

Virtual server type (instance type): t2.micro

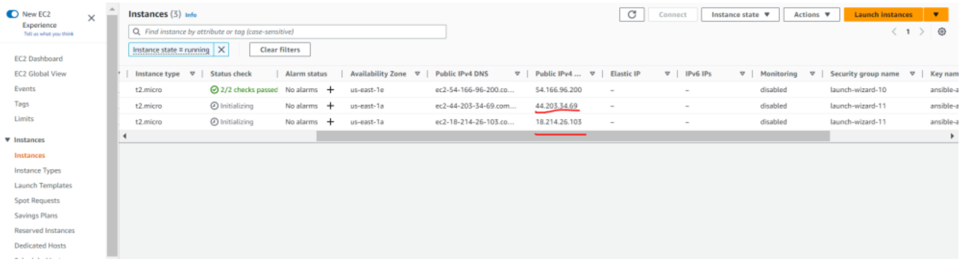
Firewall (security group): New security group

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 IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel | Launch instance

Copy both Ansible Worker Ip Address.



Instances (3) Info

Find instance by attribute or tag (case-sensitive)

Instance state = running | Clear filters

Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key nam
t2.micro	2/2 checks passed	No alarms +	us-east-1e	ec2-54-166-96-200.co...	54.166.96.200	–	–	disabled	launch-wizard-10	ansible-a
t2.micro	Initializing	No alarms +	us-east-1a	ec2-44-203-34-69.com...	44.203.34.69	–	–	disabled	launch-wizard-11	ansible-a
t2.micro	Initializing	No alarms +	us-east-1a	ec2-18-214-26-103.co...	18.214.26.103	–	–	disabled	launch-wizard-11	ansible-a

Ansible Default Special Variables

Task-02

- 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

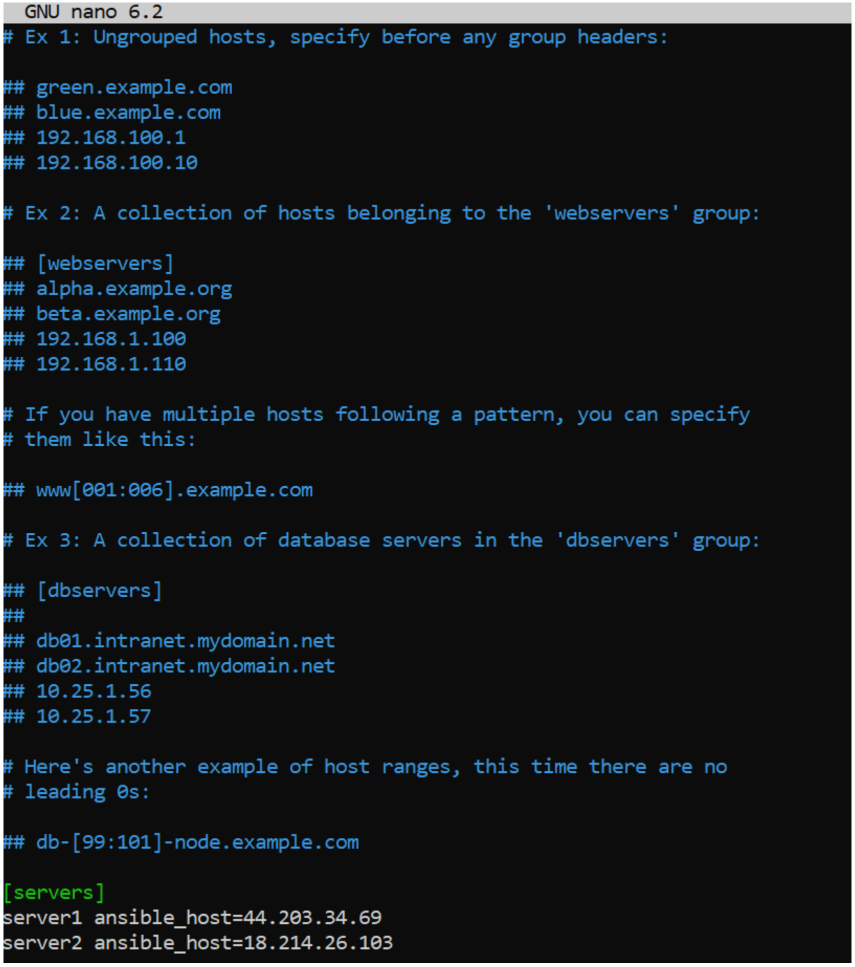
Sudo nano /etc/ansible/hosts

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

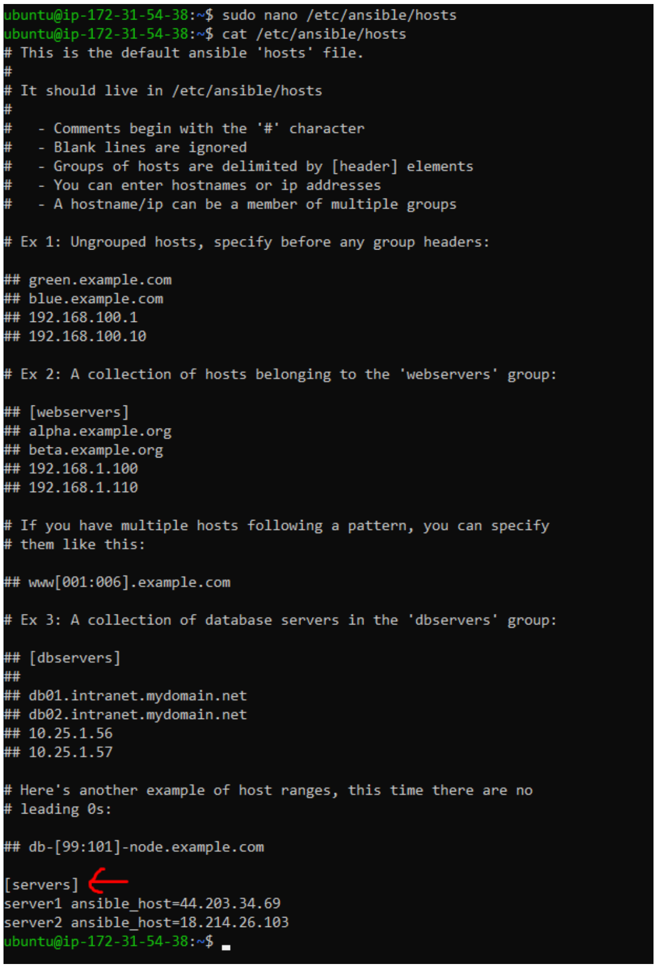
Explanation: -

Servers: - This is a Group Name

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



```
# 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=44.203.34.69
server2 ansible_host=18.214.26.103
```



```
ubuntu@ip-172-31-54-38:~$ sudo nano /etc/ansible/hosts
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
## 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=44.203.34.69
server2 ansible_host=18.214.26.103
ubuntu@ip-172-31-54-38:~$
```

Task-03

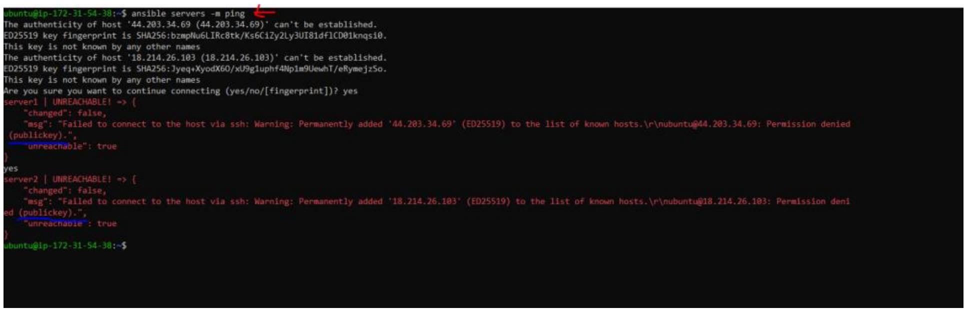
- Setup 2 more EC2 instances with same Private keys as the previous instance (Node)
- Copy the private key to master server where Ansible is setup
- Try a ping command using ansible to the Nodes.

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

ansible <group_name> -m ping

-m means module

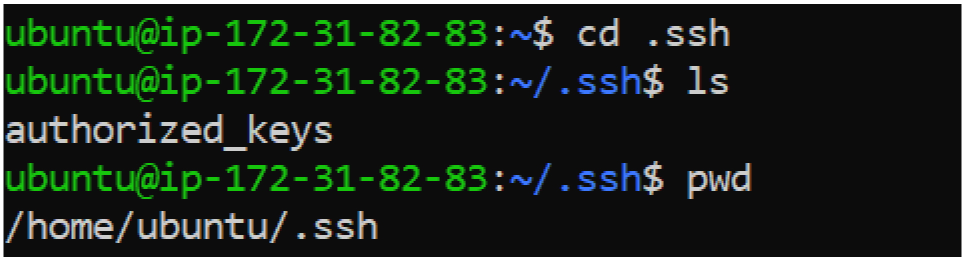
See You Got Authentication Error.



```
ubuntu@ip-172-31-54-38:~$ 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:bzmpNu6LIRc8tk/Ks6CiZy2Ly3UI81dflCD01knqsi0.
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:Jyeq+XyodX6O/xU9g1uphf4Np1m9UewhT/eRymejzSo.
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.\r\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.\r\nubuntu@18.214.26.103: Permission deni
ed (publickey).",
    "unreachable": true
}
ubuntu@ip-172-31-54-38:~$
```

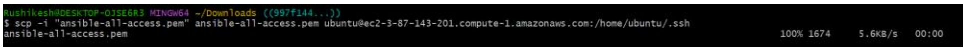
The Above Error occurs because of the public key is not available on ansible worker.

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
```

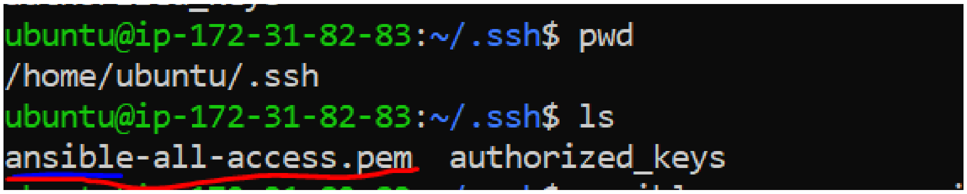
Using **SCP command**, send local public key to ansible master remote path



```
Rushikesh@DESKTOP-OJSE6R3 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.

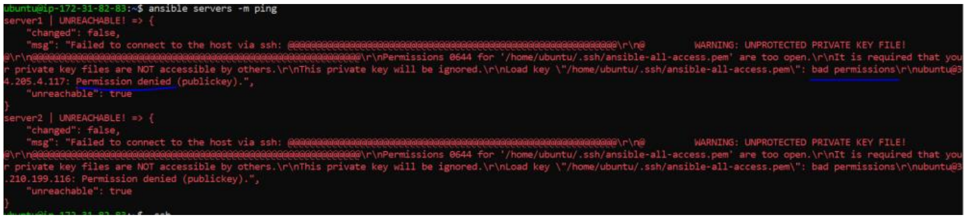


```
ubuntu@ip-172-31-82-83:~/.ssh$ pwd
/home/ubuntu/.ssh
ubuntu@ip-172-31-82-83:~/.ssh$ ls
ansible-all-access.pem  authorized_keys
```

Ok try once.

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

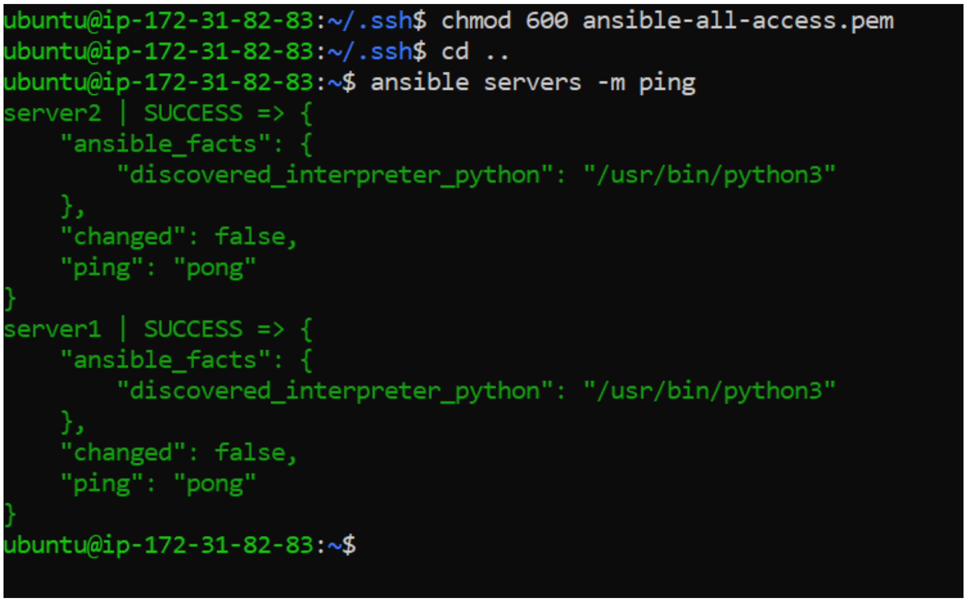
The reason is, the current user has don’t have any permission to execute or use the public key.



```
ubuntu@ip-172-31-82-83:~$ ansible servers -m ping
server1 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n@         WARNING: UNPROTECTED PRIVATE KEY FILE!
@\r\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\nPermissions 0644 for '/home/ubuntu/.ssh/ansible-all-access.pem' are too open.\r\nIt is required that you
r private key files are NOT accessible by others.\r\nThis private key will be ignored.\r\nLoad key \"/home/ubuntu/.ssh/ansible-all-access.pem\": bad permissions\r\nubuntu@3
4.205.4.117: Permission denied (publickey).",
    "unreachable": true
}
server2 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\n@         WARNING: UNPROTECTED PRIVATE KEY FILE!
@\r\n@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\r\nPermissions 0644 for '/home/ubuntu/.ssh/ansible-all-access.pem' are too open.\r\nIt is required that you
r private key files are NOT accessible by others.\r\nThis private key will be ignored.\r\nLoad key \"/home/ubuntu/.ssh/ansible-all-access.pem\": bad permissions\r\nubuntu@3
.210.199.116: Permission denied (publickey).",
    "unreachable": true
}
```

Use **chmod** command to give permission.

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



```
ubuntu@ip-172-31-82-83:~/.ssh$ chmod 600 ansible-all-access.pem
ubuntu@ip-172-31-82-83:~/.ssh$ cd ..
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"
}
ubuntu@ip-172-31-82-83:~$
```

Write a blog on this topic with screenshots in the most creative way and post it on linkedIn :)

happy learning...