ANSIBLE setup, EC2 instance creation, .ssh connection b/w nodes,

Configuration management tools:-

- Puppet, Chef uses the Pull approach for communicating with Child nodes
- Ansible, Saltstalk uses the Push approach for communicating with Child nodes

Terms-: Nodes – Virtual Machines - Instances
Master Node - Parent Node - Ansible Controller Node
Worker Node – Ansible Child node

In this module, we will start with setting up the Ansible and creating ansible controller machine along with its worker nodes and setting up the .ssh connection between them successfully, to communicate via ansible modules. Also creating ansible .config and inventory file.

1. Setup of controller machine and worker nodes:
    1. First, We will create a EC2 instance for a controller machine. Its Steps are defined below in the later section.
       After setting up the controller machine, in the console window →
       Give your machine name as AnsibleControllerMachine through hostname command
       → Switch to AnsibleControllerMachine root section through #sudo su – command.
       → Install ansible on your AnsibleControllerMachine.
       # sudo apt-get install -y software-properties-common→# sudo apt-add-repository ppa:ansible/ansible→# sudo apt-get update → # sudo apt-get install -y ansible
       (*screenshot not included)
       → Create a new user Ansiuser and switch to ansiuser by #su ansiuser command. In this user directory, you will be creating all the Ansible playbooks, inventory, config files.

```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86_64)

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

  System information as of Fri Oct 20 00:13:29 UTC 2023

  System load:  0.0               Processes:             99
  Usage of /:   31.2% of 7.57GB   Users logged in:       0
  Memory usage: 31%               IPv4 address for eth0: 172.31.33.110
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

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

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

i-022206d3477c94ca3 (AnsibleControllerMachine)

PublicIPs: 65.2.191.20 PrivateIPs: 172.31.33.110

```
See "man sudo_root" for details.

ubuntu@ip-172-31-39-218:~$ hostname AnsibleControllerMachine
hostname: you must be root to change the host name
ubuntu@ip-172-31-39-218:~$ sudo su -
root@ip-172-31-39-218:~# hostname AnsibleControllerMachine
root@ip-172-31-39-218:~# sdo su -

Command 'sdo' not found, did you mean:

  command 'sds' from deb simh (3.8.1-6)
  command 'udo' from deb udo (6.4.1-5)
  command 'sdb' from deb sdb (1.2-2.1)
  command 'xdo' from deb xdo (0.5.7-1)
  command 'sudo' from deb sudo (1.8.31-1ubuntu1.5)
  command 'sudo' from deb sudo-ldap (1.8.31-1ubuntu1.5)
  command 'sdf' from deb sdf (2.001+1-7)
  command 'cdo' from deb cdo (1.9.9~rc1-1)
  command 'sdop' from deb sdop (0.81-1)
  command 'sdc' from deb hpsockd (0.17build3)
  command 'sdoc' from deb ruby-sdoc (1.0.0-1fakesync1)

Try: apt install <deb name>

root@ip-172-31-39-218:~# sudo su -
root@AnsibleControllerMachine:~#
```

```
root@AnsibleControllerMachine:~# adduser ansiuser
Adding user `ansiuser' ...
Adding new group `ansiuser' (1001) ...
Adding new user `ansiuser' (1001) with group `ansiuser' ...
Creating home directory `/home/ansiuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ansiuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@AnsibleControllerMachine:~# su ansiuser
```

```
Retype new password:
passwd: password updated successfully
Changing the user information for ansiuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@AnsibleControllerMachine:~# su ansiuser
ansiuser@AnsibleControllerMachine:/root$
```

2. Make sure you are On the root user. Install OpenSSH if not already installed.
   - ➔ From the root user privileges OR by using sudo command in front of #vim /etc/ssh/sshd_config you will be able to open the configuration file which handles all the ssh connections and privileges.
   - ➔ Here, you will be able to find a column of password authentication. Set it to YES after uncommenting if commented. It will give permissions to your machine to connect to other machines through .ssh key after authorizing with the password of ansiuser → save the file with :wq! Command
   - ➔ Now open the sudoers file with root user privileges. This file contains the permissions of users and other related configurations. Here, find the section of root user password-all and specify-(ansiuser ALL=NOPASSWD: ALL) below it. It will set up the all root privileges to our new ansiuser without been asked for the passwords every time we use the root privileges. Save the file.
   - ➔ Restart ssh server to apply all the configuration changes. You can use command like- # sudo systemctl restart ssh OR # sudo service ssh restart, based on your

linux distribution system.
Initial ansible, ssh permissions and ansiuser setup is completed now.

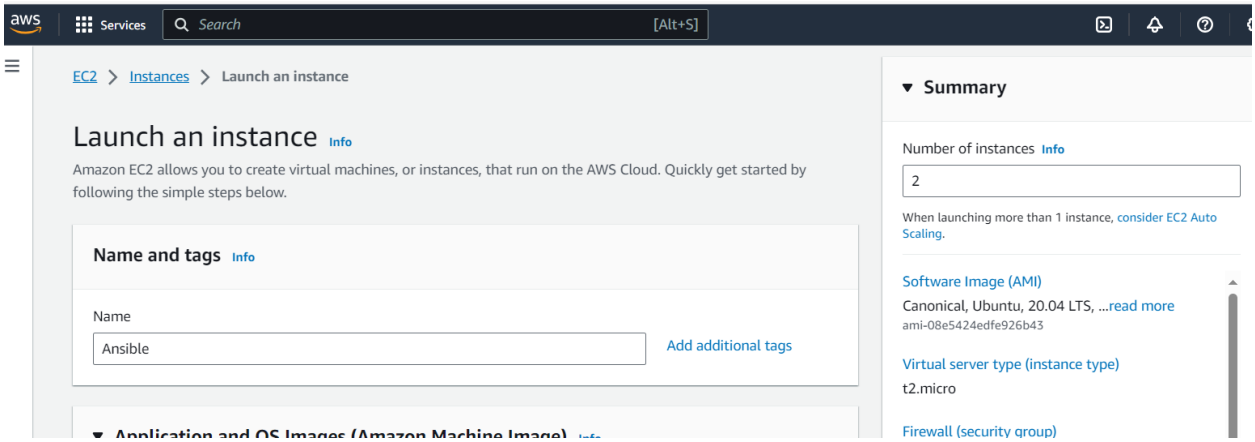
```
#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
:wq!
```

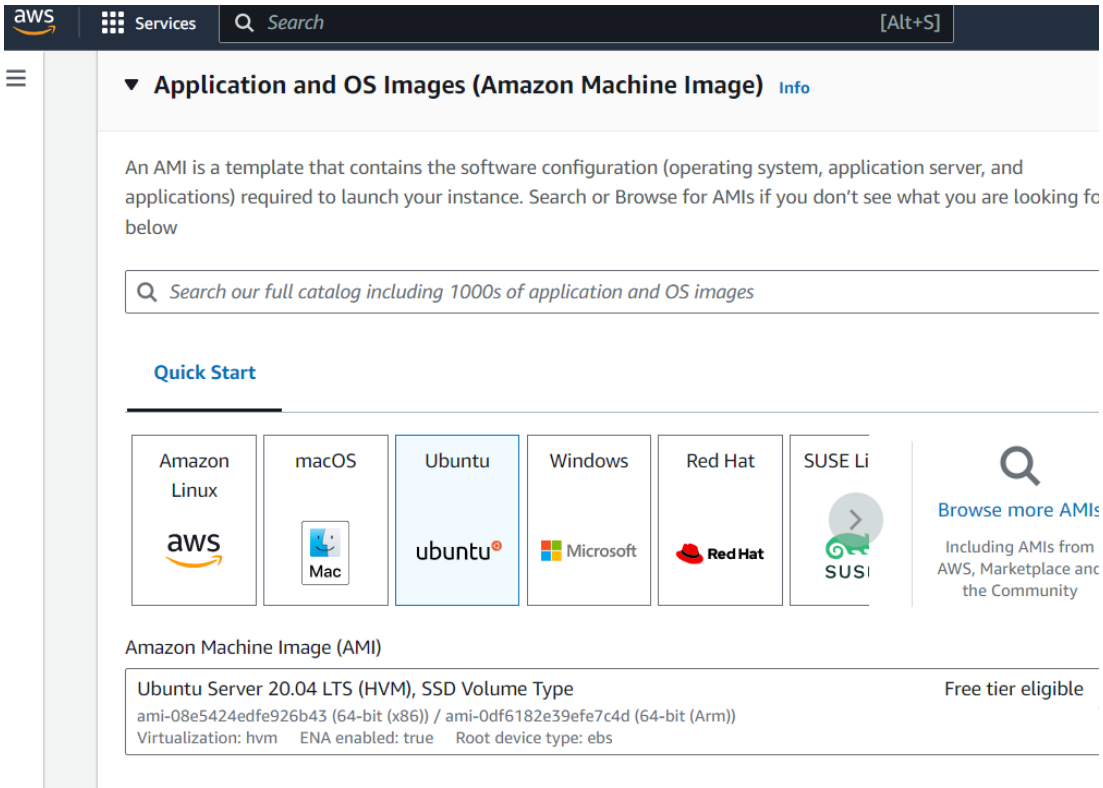
```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
ansiuser ALL=NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
-- INSERT -- W10: Warning: Changing a readonly file                                    21,5
```

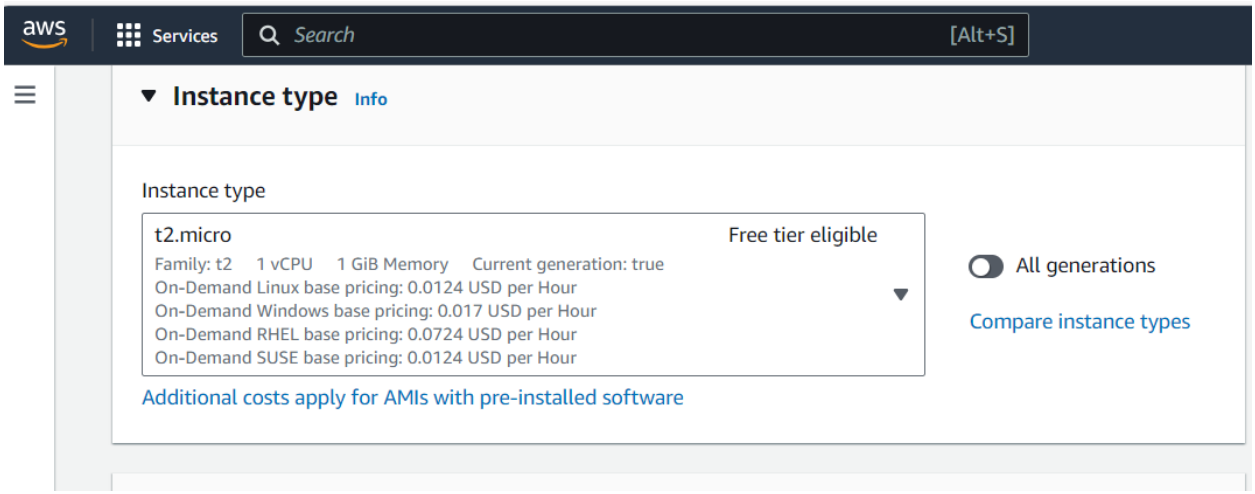
```
aws   Services   Search                                                    [Alt+S]
[sudo] password for ansiuser:
ansiuser is not in the sudoers file.  This incident will be reported.
ansiuser@AnsibleControllerMachine:/root$ sudo su -
[sudo] password for ansiuser:
ansiuser is not in the sudoers file.  This incident will be reported.
ansiuser@AnsibleControllerMachine:/root$ exit
exit
root@AnsibleControllerMachine:~# vim /etc/ssh/sshd_config
root@AnsibleControllerMachine:~# vim /etc/sudoers
root@AnsibleControllerMachine:~#   systemctl restart sshd
root@AnsibleControllerMachine:~# su ansiuser
```

3. Now, you will be needing to create the worker nodes. We will use AWS EC2 machines to create 2 worker nodes. Steps are specified below in screenshots.
4. After connecting to both the nodes, you will need to repeat all the above steps for both the worker nodes, from giving hostname AnsibleNode1/2 → Installing Ansible → making ansiuser → setting up all the ssh permissions and sudoers permissions → to restarting sshd

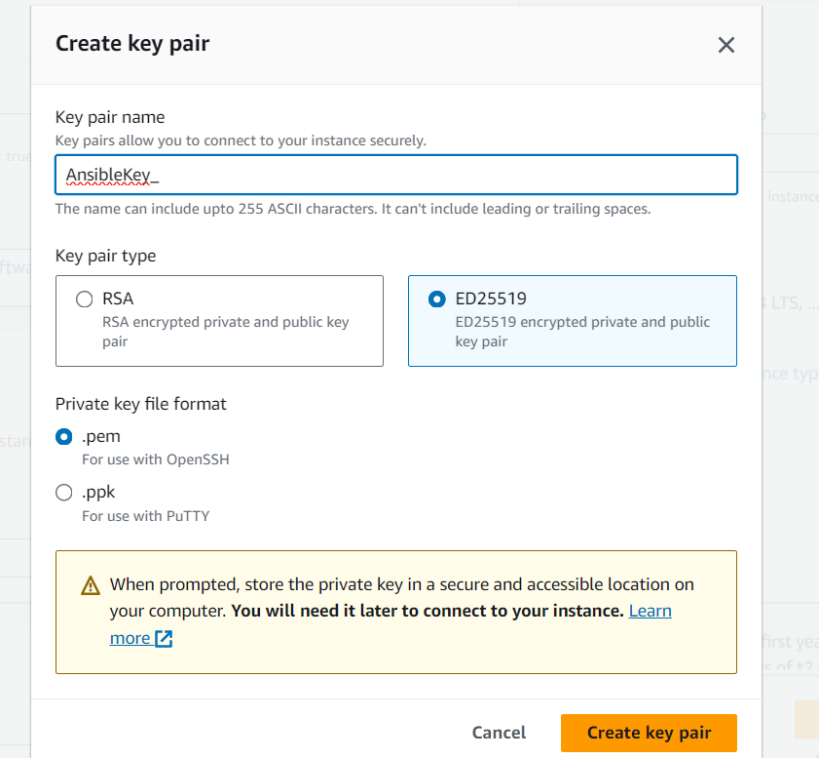
Give instance name and count on the right



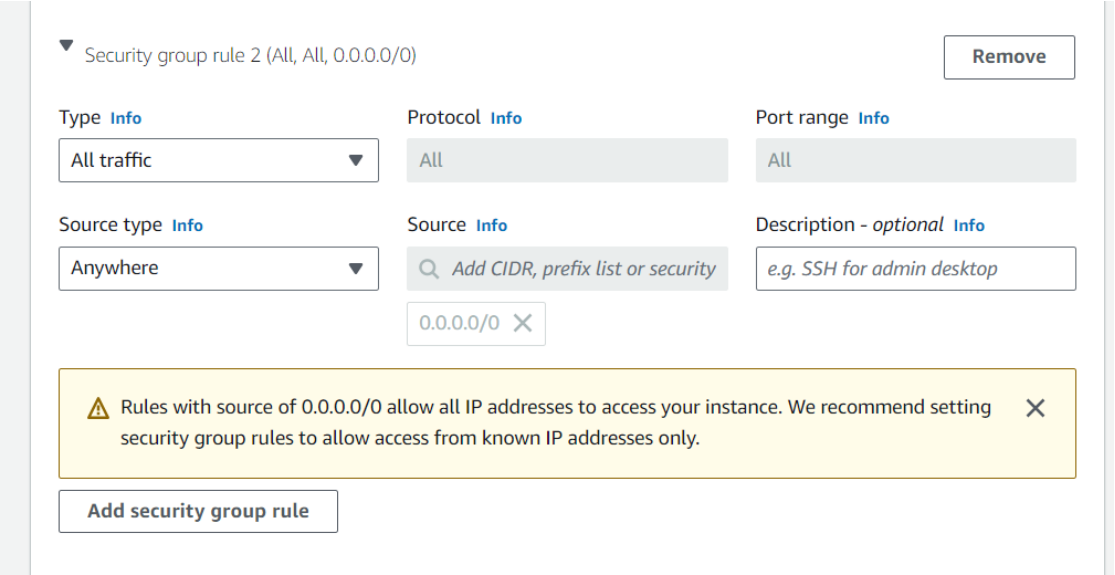




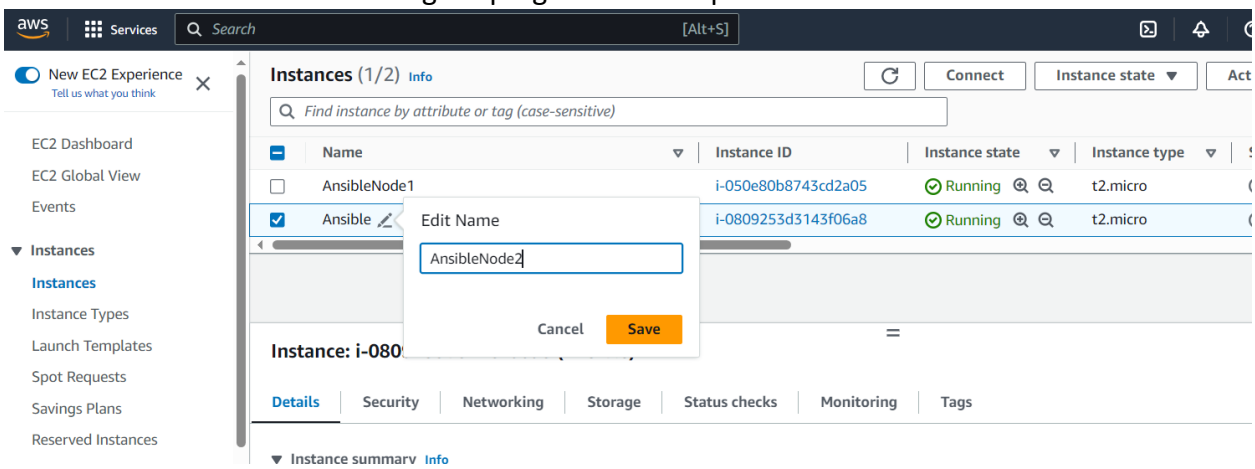
Create Key pair and download the same, which will be used if you want to run this instance in your local system CLI instead of AWS console. Scenario shown later in below sections



In network security group section, add a security group with following permissions



Launch the instance. 2 instances will be created with the same name. Rename them and connect each of them through top right connect option.



After connecting successfully, you will be able to see the console window of your instances OR worker nodes successfully

aws Services Search [Alt+S]

```
  Memory usage: 22%               IPv4 address for eth0: 172.31.7.129
  Swap usage:   0%

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


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-172-31-7-129:~$
```

i-050e80b8743cd2a05 (AnsibleNode1)

aws Services Search [Alt+S]

```
  Memory usage: 23%               IPv4 address for eth0: 172.31.8.49
  Swap usage:   0%

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


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-172-31-8-49:~$
```

i-0809253d3143f06a8 (AnsibleNode2)

PublicIPs: 3.110.223.46 PrivateIPs: 172.31.8.49

```
Is the information correct? [Y/n]
root@AnsibleNode2:~# Y
Y: command not found
root@AnsibleNode2:~# su ansiuser
ansiuser@AnsibleNode2:/root$ exit
exit
root@AnsibleNode2:~# vim /etc/ssh/sshd_config
root@AnsibleNode2:~# vim /etc/sudoers
root@AnsibleNode2:~# systemctl restart sshd
root@AnsibleNode2:~#
```

i-0809253d3143f06a8 (AnsibleNode2)

PublicIPs: 3.110.223.46 PrivateIPs: 172.31.8.49

```
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@AnsibleNode1:~# vim /etc/ssh/sshd_config
root@AnsibleNode1:~# vim /etc/sudoers
root@AnsibleNode1:~# systemctl restart sshd
root@AnsibleNode1:~#
```

i-050e80b8743cd2a05 (AnsibleNode1)

PublicIPs: 3.110.189.85 PrivateIPs: 172.31.7.129

2. Connecting Master and Worker Nodes from SSH key
    1. These steps will only be executed on the Master Node or the Controller EC2 machine-
    2. Generate SSH Key pair on master node in ansiuser.
    3. Copy the ssh-key generated in the both Worker node through the master node via command - #ssh-copy-id -i ansiuser@privateipWorker1 , #ssh-copy-id -i ansiuser@privateipWorker2
    (please note* If your controller machine is not in the AWS network but another network or your local system, you will be using the Public IP address of the worker nodes throughout the process)

4. Now go to directory for ansible in- /etc/ansible and find the hosts file there, in which you will add both the created worker nodes Private IP addresses as shown. Define each node IP address's name to be called through that name in the module functions instead of writing IP addresses again. After adding IP addresses execute the ping module command.
5. Ping module checks if the worker nodes are connected or not, by responding pong with response if connected successfully. If not connected, it will show the error.

```
root@AnsibleControllerMachine:~# su ansiuser
ansiuser@AnsibleControllerMachine:/root$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansiuser/.ssh/id_rsa):
Created directory '/home/ansiuser/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansiuser/.ssh/id_rsa
Your public key has been saved in /home/ansiuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:3jIGOQyFxHtZNMx/iwAuDrIdJo+u6/XnFKtBlBZpyzE ansiuser@AnsibleControllerMachine
The key's randomart image is:
+---[RSA 3072]----+
|   o.oo+o        |
|    oE+ +.       |
|    +*++ .       |
|o + ==+.. . .    |
| O + +=.S. o .   |
|o o o  +o.. .    |
|.  . . o= .      |
| .. . +o o       |
|=o   oo.         |
+----[SHA256]-----+
ansiuser@AnsibleControllerMachine:/root$ ssh-copy-id -i ansiuser@172.31.7.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansiuser/.ssh/id_rsa.pub"
```

i-022206d3477c94ca3 (AnsibleControllerMachine)

PublicIPs: 65.2.191.20 PrivateIPs: 172.31.33.110

```
+----[SHA256]-----+
ansiuser@AnsibleControllerMachine:/root$ ssh-copy-id -i ansiuser@172.31.7.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansiuser/.ssh/id_rsa.pub"
The authenticity of host '172.31.7.129 (172.31.7.129)' can't be established.
ECDSA key fingerprint is SHA256:c+UjdhRRmPJmaU5wjvxkXLL0hMSNIgsCkWrHhHJTEKM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansiuser@172.31.7.129's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansiuser@172.31.7.129'"
and check to make sure that only the key(s) you wanted were added.
```

```
ansiuser@AnsibleControllerMachine:/root$ cd /etc/ansible/
ansiuser@AnsibleControllerMachine:/etc/ansible$ ls
ansible.cfg  hosts  roles
ansiuser@AnsibleControllerMachine:/etc/ansible$ sudo vim hosts
```

```
## [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
[worker1]
172.31.7.129

[worker2]
172.31.8.49

[webserver]
172.31.7.129
172.31.8.49
"hosts" [readonly] 52L, 1101C
```

i-022206d3477c94ca3 (AnsibleControllerMachine)

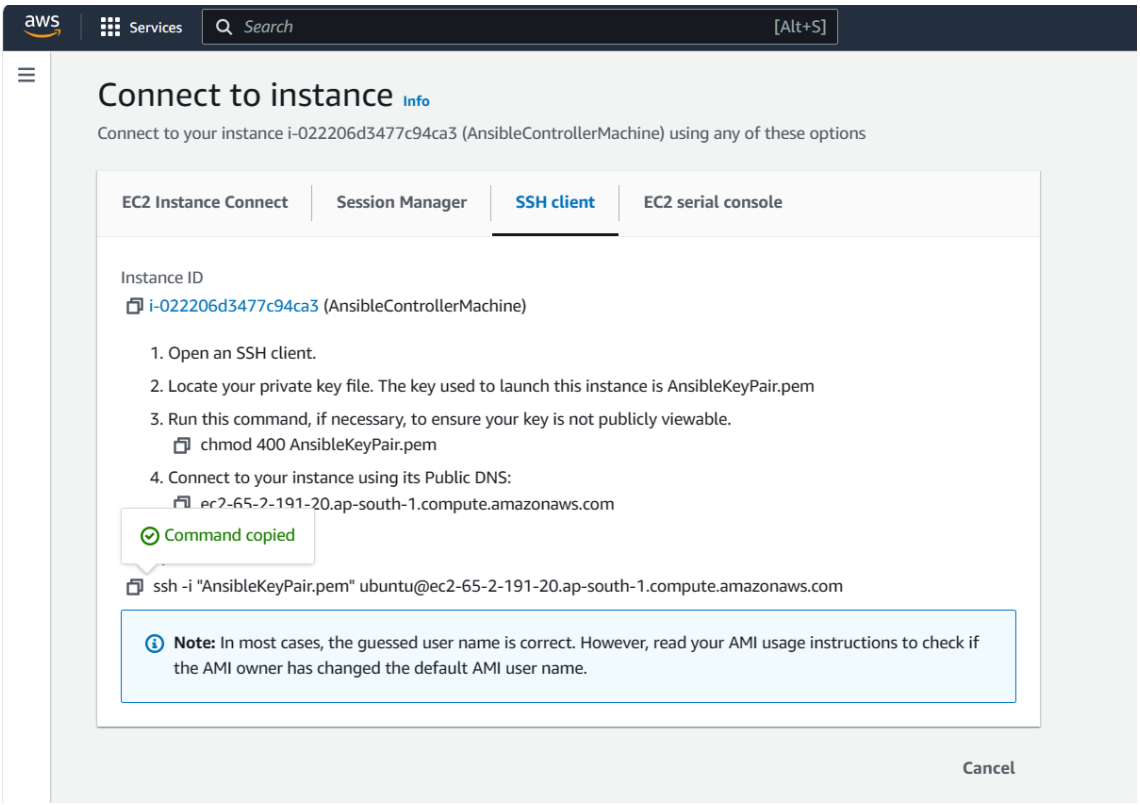
PublicIPs: 65.2.191.20 PrivateIPs: 172.31.33.110

```
ansiuser@AnsibleControllerMachine:/etc/ansible$ sudo vim hosts
ansiuser@AnsibleControllerMachine:/etc/ansible$ ansible worker2 -m ping
172.31.8.49 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ansiuser@AnsibleControllerMachine:/etc/ansible$ ansible webserver -m ping
172.31.8.49 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.7.129 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

3. Additonal Important Scenarios

    1. Create the file named MyInventory outside the current directory /etc/ansible to /home/ansiuser, to enter both the IP addresses in that file, and call the ping module from that directory defining the path of inventory file.

- Prior to that, we will execute this controller machine from our local system instead of aws cli. By the help of ssh local system url provided by the aws, execute the above steps from the local system itself.
  - ➔ Copy the ssh-client url of Controller machine from the AWS connect module as show in in figure



  - ➔ While creating the EC2 machine, the private key .pem file must be downloaded in your system , and would be present in the downloads folder of your local system. So open the CLI (git-bash for windows) and go to downloads folder and paste the ssh client url there, so that ssh can access the private key .pem file
  - ➔ You will now be connected to your ec2 machine and can operate your machine from your local system instead of aws console.

```
ubuntu@AnsibleControllerMachine: ~

admin@DESKTOP-9UJRCUE MINGW64 ~
$ cd Downloads/

admin@DESKTOP-9UJRCUE MINGW64 ~/Downloads
$ ssh -i "AnsibleKeyPair.pem" ubuntu@ec2-65-2-191-20.ap-south-1.compute.amazonaw
s.com
The authenticity of host 'ec2-65-2-191-20.ap-south-1.compute.amazonaws.com (65.2
.191.20)' can't be established.
ED25519 key fingerprint is SHA256:l7gbg62vCL8fiDNPMrZwxH4HksRx38FC4VMk4IW2QWk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-2-191-20.ap-south-1.compute.amazonaws.com' (E
D25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1036-aws x86_64)

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

  System information as of Fri Oct 20 00:52:50 UTC 2023

  System load:  0.01              Processes:             106
  Usage of /:   31.2% of 7.57GB   Users logged in:       1
```

2. Now, we will switch to the home directory of ansiuser and create MyInventory file there filling in all the details.
   Now, call the ping module from the same directory defining the path to your inventory file.

```
/
ansiuser@AnsibleControllerMachine:/$ cd /home/ansiuser/
ansiuser@AnsibleControllerMachine:~$ pwd
/home/ansiuser
ansiuser@AnsibleControllerMachine:~$ ls
ansiuser@AnsibleControllerMachine:~$ vim MyInventory
```

```
ansiuser@AnsibleControllerMachine: ~
[webserver]
172.31.7.129
172.31.8.49
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

```
ansiuser@AnsibleControllerMachine:~$ ansible -i /home/ansiuser/MyInventory webserver -m ping
172.31.7.129 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.8.49 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ansiuser@AnsibleControllerMachine:~$
```

3. Ansible Configuration File :
   In the ansiuser home directory when you will command #ls to find - MyInventory file present there which consists the IP address of both the worker nodes. Now, we will be creating the Ansible Configuration file in the same directory. With extension .cfg and fill the inventory path details in the configuration file.
   Now we will execute the ping module command without defining the path to the inventory file because it has already been defined into the config file. The ping module will successfully be running.

```
}
ansiuser@AnsibleControllerMachine:~$ sudo vim ansible.cfg
```

ansiuser@AnsibleControllerMachine: ~

```
[defaults]
inventory = /home/ansiuser/MyInventory
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

```
ansiuser@AnsibleControllerMachine:~$ sudo vim ansible.cfg
ansiuser@AnsibleControllerMachine:~$ ansible webserver -m ping
172.31.8.49 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
172.31.7.129 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```