

How to Configure Ansible?

Step-1: Create 3 instances of t2.micro Amazon Linux VMs in AWS.

- a. 1 for - Control Node
- b. 2 for - Managed Nodes

Note: Connect to all 3 VMs using MobaXterm

create 3 EC2 instance

The screenshot shows the AWS EC2 'Launch an instance' wizard. In the 'Summary' section, the 'Number of instances' dropdown is set to 3. A large red circle highlights this field. The 'Software Image (AMI)' is set to 'Amazon Linux 2023.6.2...'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Firewall (security group)' dropdown is set to 'New security group'. At the bottom right, there are 'Cancel' and 'Launch instance' buttons, with the 'Launch instance' button being orange.

Here we created 3 ec2 instance successfully

1--> control node

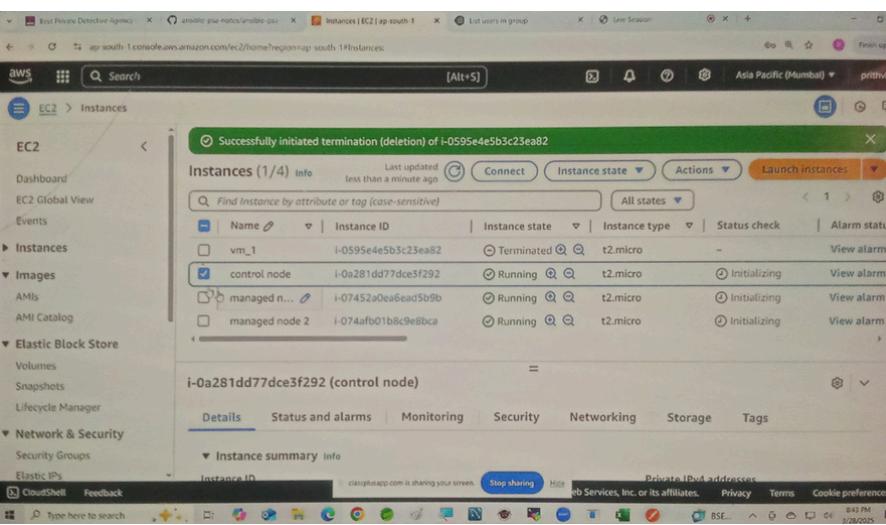
2--> managed node

The screenshot shows the AWS EC2 Instances page. A green success message at the top right says "Successfully initiated termination (deletion) of i-0595e4e5b3c23ea82". The main table lists four instances:

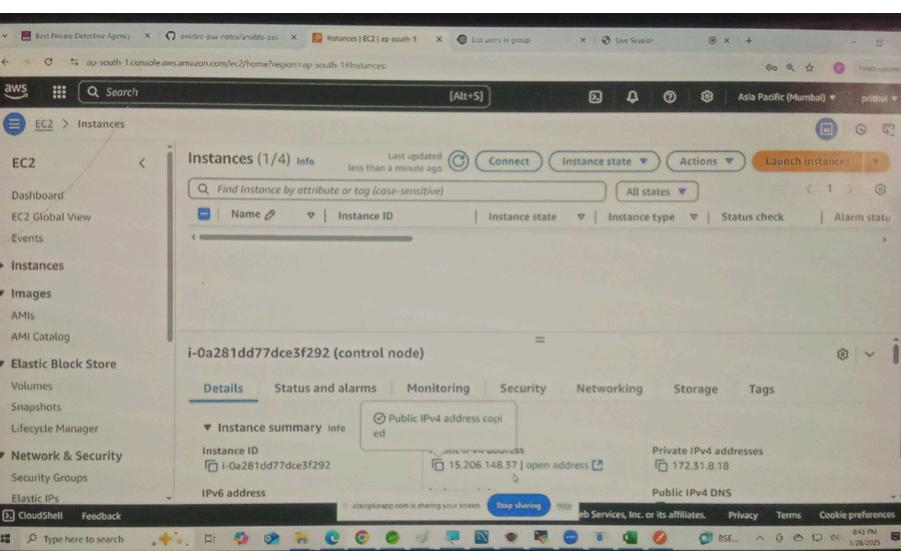
| Name | Instance ID | Instance state | Instance type | Status check | Alarm status |
|----------------|---------------------|----------------|---------------|--------------|--------------|
| vm_1 | i-0595e4e5b3c23ea82 | Terminated | t2.micro | - | View alarm |
| control node | i-0a281dd77dce3f292 | Running | t2.micro | Initializing | View alarm |
| managed node 1 | i-07452a0ea6ead5b9b | Running | t2.micro | Initializing | View alarm |
| managed node 2 | i-074afb01b8c9e8bca | Running | t2.micro | Initializing | View alarm |

A red box highlights the "control node" and "managed node 1" rows. A blue box highlights the entire table area. A red arrow points from the text "control node" in the previous section to the "control node" instance in the screenshot.

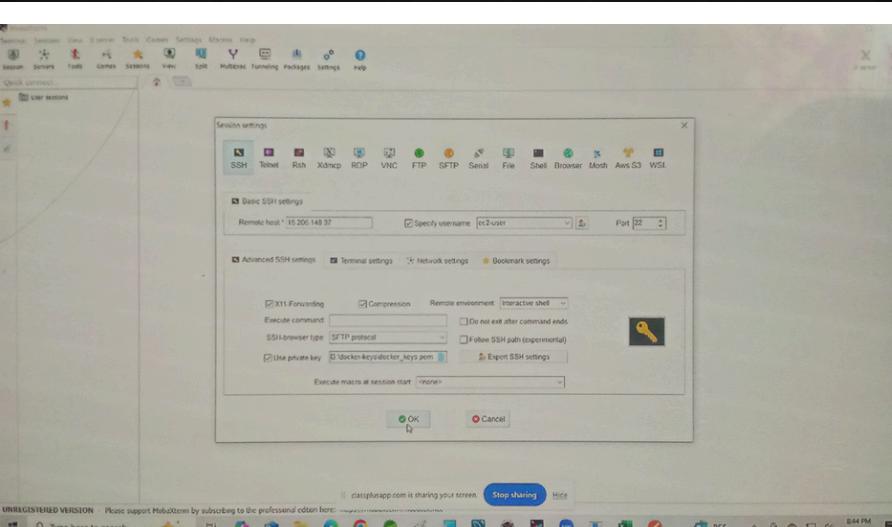
Note: Connect to all 3 VMs using MobaXterm



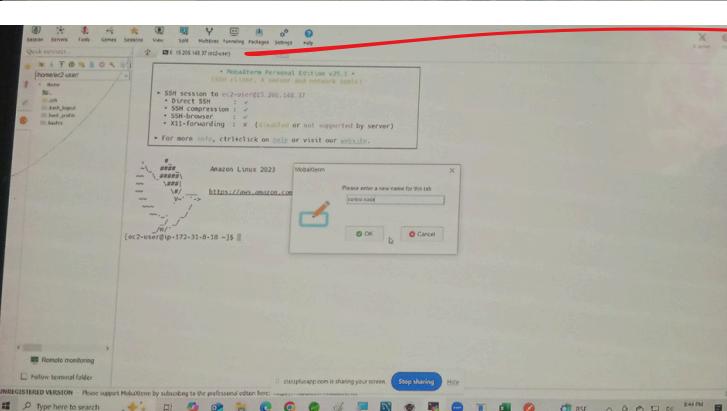
step 1: select control node



step 2: copy private ip address

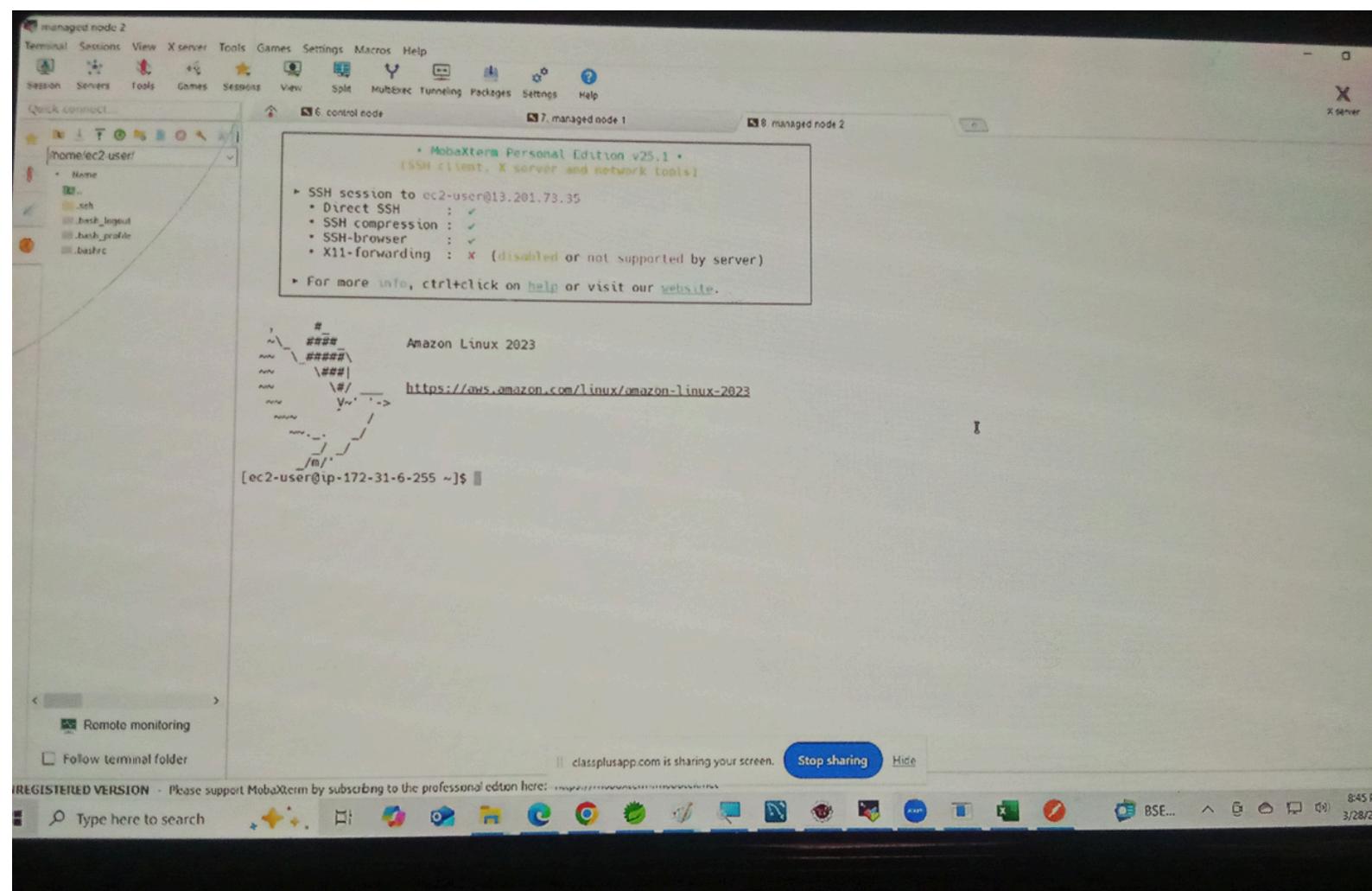


step 3: goto MobaXterm select session fill details as show in image



step 4: give name as control node

Now finally connected EC2 in your MobaXterm



Step-2: Do the following setup in all 3 machines

a) Create user:

```
sudo useradd ansible
```

```
sudo passwd ansible
```

b) Configure user in sudoers file

```
sudo visudo
```

```
ansible ALL=(ALL) NOPASSWD: ALL
```

c) Update sshd config file

```
sudo vi /etc/ssh/sshd_config
```

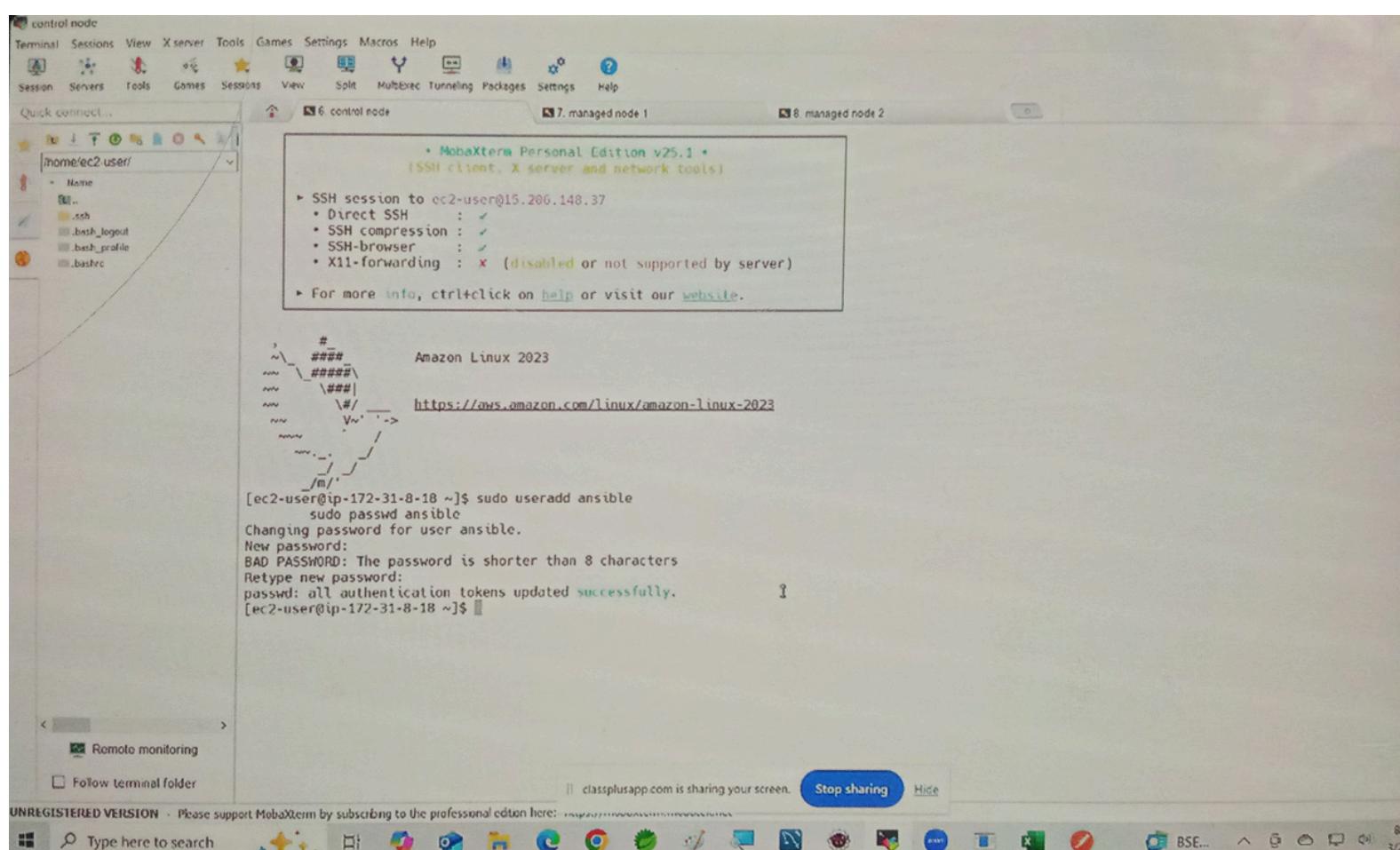
-> change PasswordAuthentication "no" and PermitEmptyPasswords "yes"

d) Restart the server

```
sudo service sshd restart
```

a) Create user:

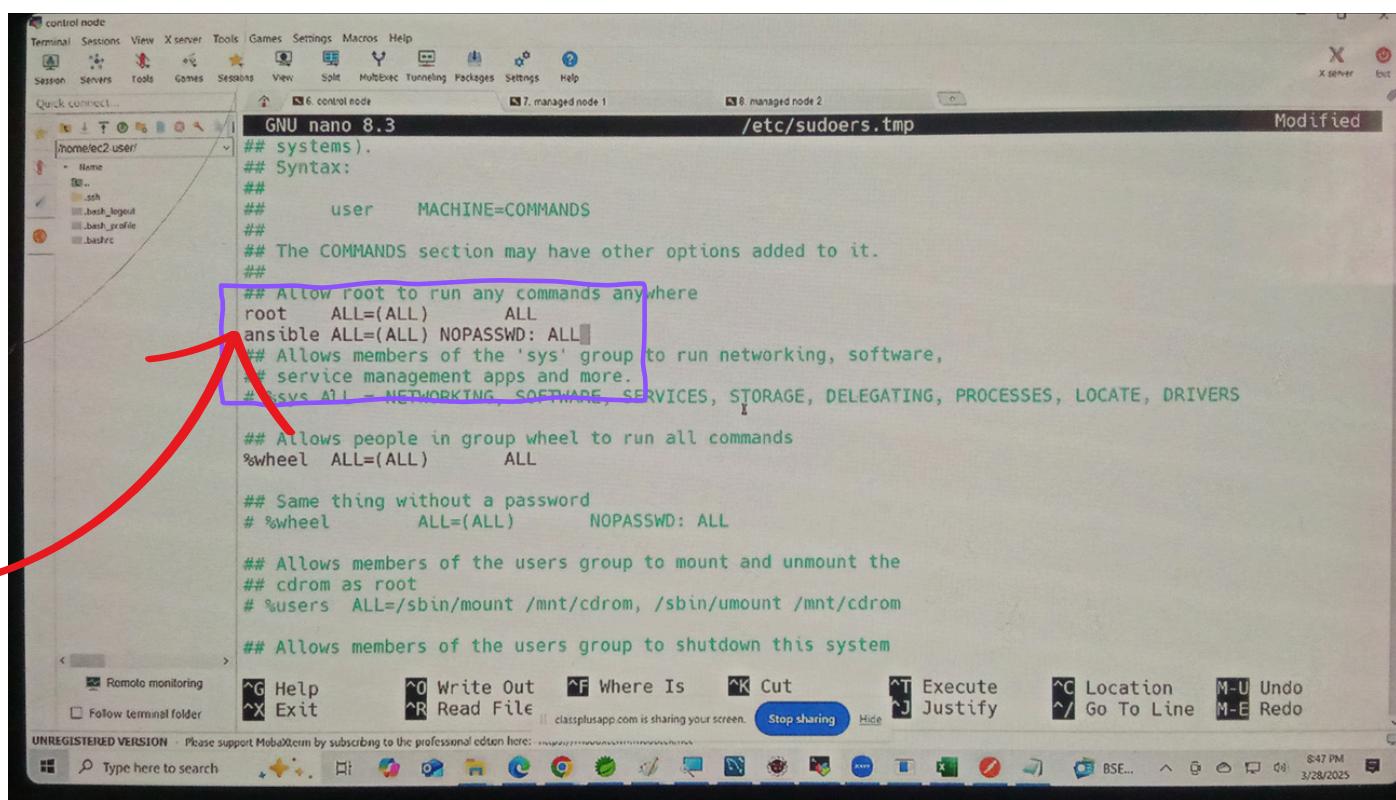
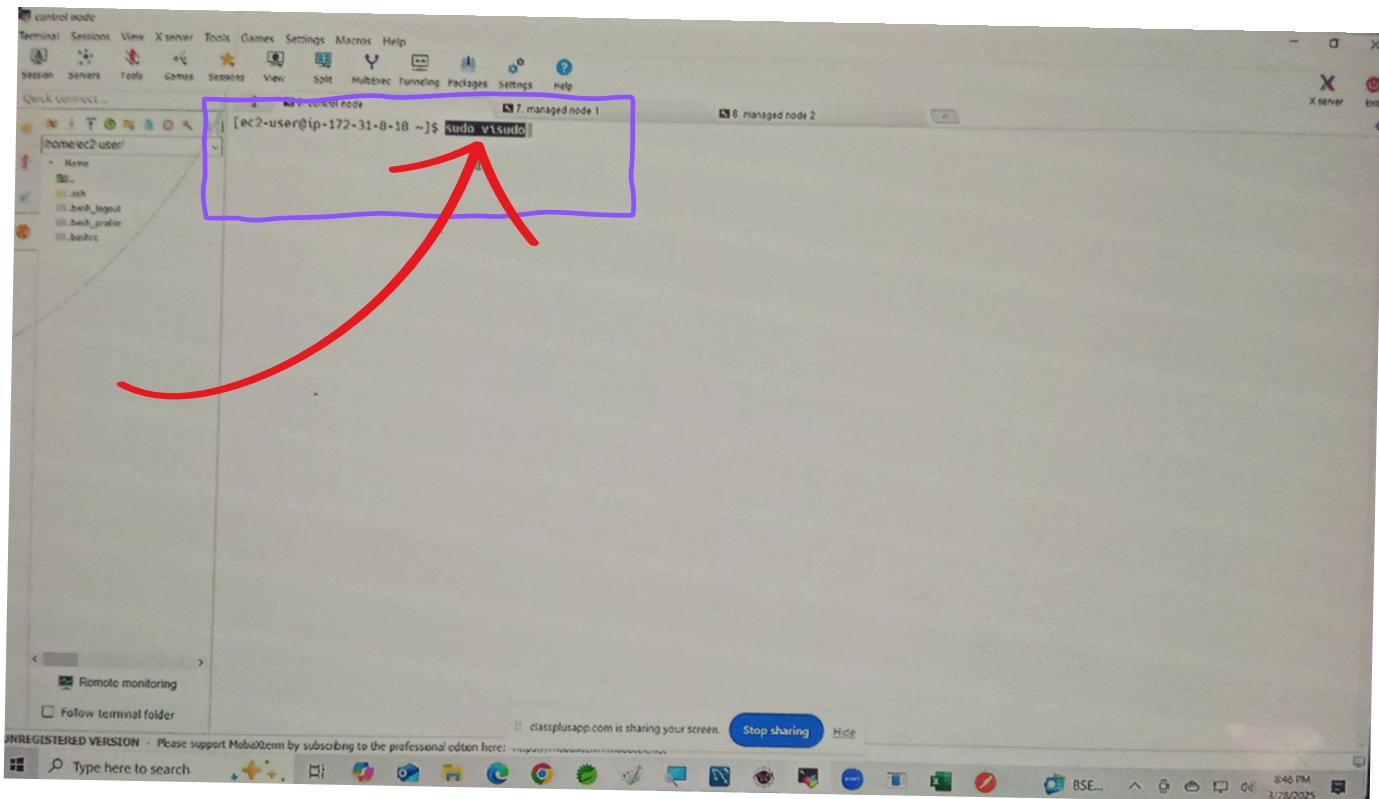
```
sudo useradd ansible  
sudo passwd ansible
```



b) Configure user in sudoers file

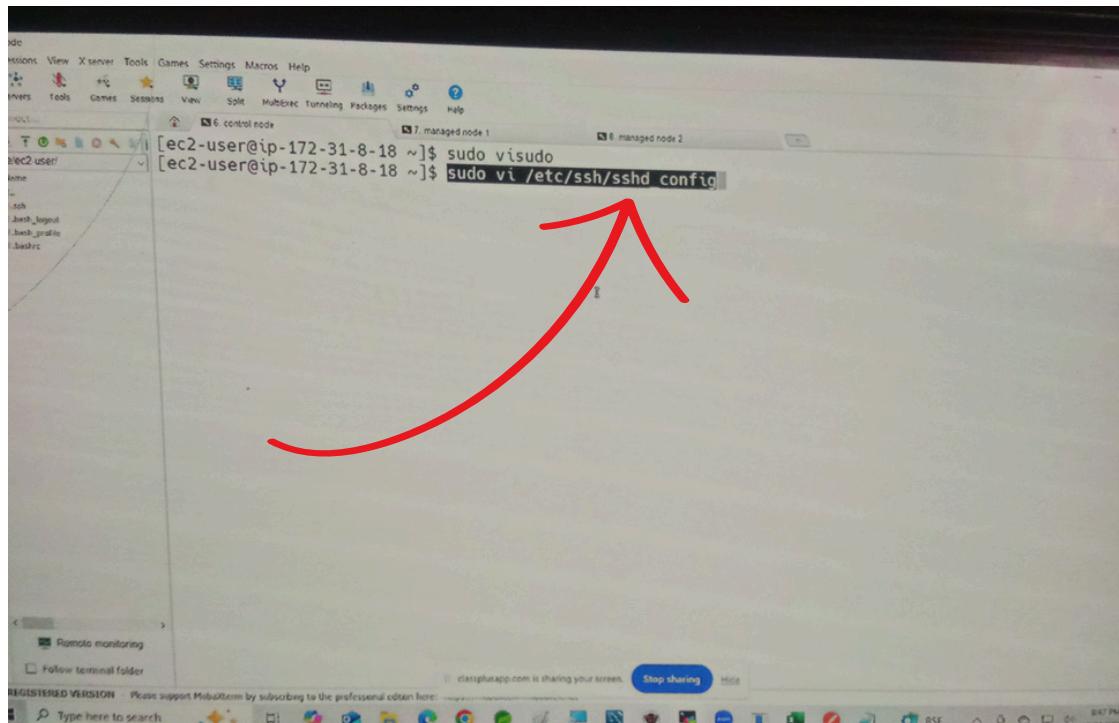
sudo visudo

ansible ALL=(ALL) NOPASSWD: ALL



c) Update sshd config file

`sudo vi /etc/ssh/sshd_config`



-> change PasswordAuthentication "no" and
PermitEmptyPasswords "yes"

```
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords yes

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

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

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
#GSSAPIEnablekSusers no

-- INSERT --
```

d) Restart the server

`sudo service sshd restart`

The screenshot shows a MobaXterm interface with multiple sessions. Session 7, titled "managed node 1", contains the following terminal history:

```
[ec2-user@ip-172-31-8-18 ~]$ sudo visudo
[ec2-user@ip-172-31-8-18 ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@ip-172-31-8-18 ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[ec2-user@ip-172-31-8-18 ~]$
```

A large red arrow points from the top left towards the command `sudo service sshd restart`.

At the bottom of the window, there is a sharing notice: "classplusapp.com is sharing your screen. Stop sharing Hide".

The taskbar at the bottom of the screen shows various application icons.

Step-3: Install Ansible in Control Node

a) Switch to Ansible user

```
sudo su ansible
```

```
cd ~
```

b) Install Python because ansible is developed using python & to run ansible we require python software

```
sudo yum install python3 -y
```

c) cross Check python is installed?

```
python3 --version
```

d) Install PIP (package manager) for downloading softwares and intalling it

```
sudo yum -y install python3-pip
```

e) Install Ansible using Python PIP

```
pip3 install ansible --user
```

f) Verify ansible version

```
ansible --version
```

g) Create ansible folder under /etc

```
sudo mkdir /etc/ansible
```

a) Switch to Ansible user

sudo su ansible

cd ~

The screenshot shows the MobaXterm interface with four sessions open:

- control node**: Session 1, showing the user's home directory.
- 6 control node**: Session 2, showing the user's home directory.
- 7 managed node 1**: Session 3, showing the user's home directory.
- 8 managed node 2**: Session 4, showing the user's home directory.

In session 6 (control node), the terminal history is as follows:

```
[ec2-user@ip-172-31-8-18 ~]$ sudo visudo
[ec2-user@ip-172-31-8-18 ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@ip-172-31-8-18 ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[ec2-user@ip-172-31-8-18 ~]$ su ansible
Password:
[ansible@ip-172-31-8-18 ec2-user]$ cd ~
[ansible@ip-172-31-8-18 ~]$ pwd
/home/ansible
[ansible@ip-172-31-8-18 ~]$
```

A red arrow points from the word "ansible" in the password prompt to the "su" command in the terminal history.

b) Install Python because ansible is developed using python & to run ansible we require python software

sudo yum install python3 -y

```
[ec2-user@ip-172-31-8-18 ~]$ sudo visudo
[ec2-user@ip-172-31-8-18 ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@ip-172-31-8-18 ~]$ sudo service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[ec2-user@ip-172-31-8-18 ~]$ su ansible
Password:
[ansible@ip-172-31-8-18 ec2-user]$ cd ~
[ansible@ip-172-31-8-18 ~]$ pwd
/home/ansible
[ansible@ip-172-31-8-18 ~]$ sudo yum install python3 -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Fri Mar 28 15:23:13 2025.
Package python3-3.9.21-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ansible@ip-172-31-8-18 ~]$
```

A red arrow points from the top right towards the terminal command `sudo yum install python3 -y`.

c) cross Check python is installed?

`python3 --version`

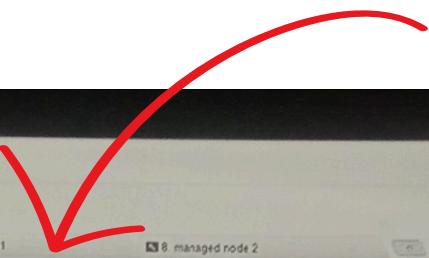
A screenshot of a terminal window titled "control node". The terminal shows the following command execution:

```
[ec2-user@ip-172-31-8-18 ~]$ sudo visudo  
[ec2-user@ip-172-31-8-18 ~]$ sudo vi /etc/ssh/sshd_config  
[ec2-user@ip-172-31-8-18 ~]$ sudo service sshd restart  
Redirecting to /bin/systemctl restart sshd.service  
[ec2-user@ip-172-31-8-18 ~]$ su ansible  
Password:  
[ansible@ip-172-31-8-18 ec2-user]$ cd ~  
[ansible@ip-172-31-8-18 ~]$ pwd  
/home/ansible  
[ansible@ip-172-31-8-18 ~]$ sudo yum install python3 -y  
Amazon Linux 2023 Kernel Livepatch repository  
Last metadata expiration check: 0:00:01 ago on Fri Mar 28 15:23:13 2025.  
Package python3-3.9.21-1.amzn2023.0.2.x86_64 is already installed.  
Dependencies resolved.  
Nothing to do.  
Complete!  
[ansible@ip-172-31-8-18 ~]$ python3 --version  
Python 3.9.21  
[ansible@ip-172-31-8-18 ~]$
```

A red arrow points from the word "Complete!" in the terminal output to the "Stop sharing" button in the top right corner of the terminal window.

d) Install PIP (package manager) for downloading softwares and intalling it

```
sudo yum -y install python3-pip
```



```
[ansible@ip-172-31-8-18 ~]$ sudo yum -y install python3-pip
Last metadata expiration check: 0:00:51 ago on Fri Mar 28 15:23:13 2025.
Dependencies resolved.

=====
Package          Architecture Version      Repository  Size
=====
Installing:
python3-pip      noarch      21.3.1-2.amzn2023.0.10   amazonlinux 1.8 M
Installing weak dependencies:
libxcrypt-compat x86_64      4.4.33-7.amzn2023    amazonlinux 92 k

Transaction Summary
=====
Install 2 Packages

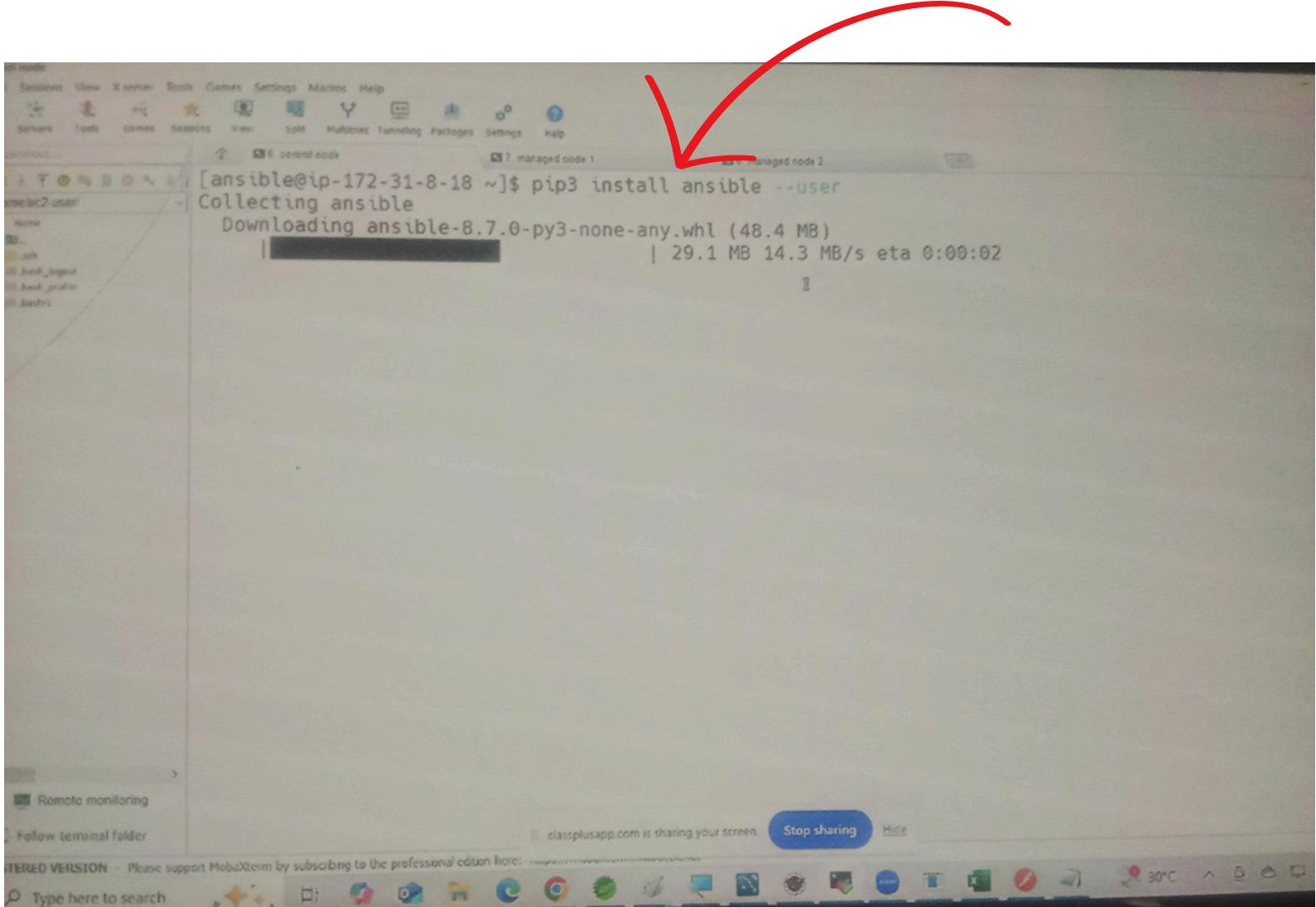
Total download size: 1.9 M
Installed size: 11 M
Downloading Packages:
(1/2): libxcrypt-compat-4.4.33-7.amzn2023.x86_64.rpm           2.3 MB/s | 92 kB    00:00
(2/2): python3-pip-21.3.1-2.amzn2023.0.10.noarch.rpm            28 MB/s | 1.8 MB   00:00
Total                                         19 MB/s | 1.9 MB   00:00

Running transaction check
Transaction check succeeded.
Running transaction test

classplusapp.com is sharing your screen. Stop sharing Hide
```

e) Install Ansible using Python PIP

pip3 install ansible --user



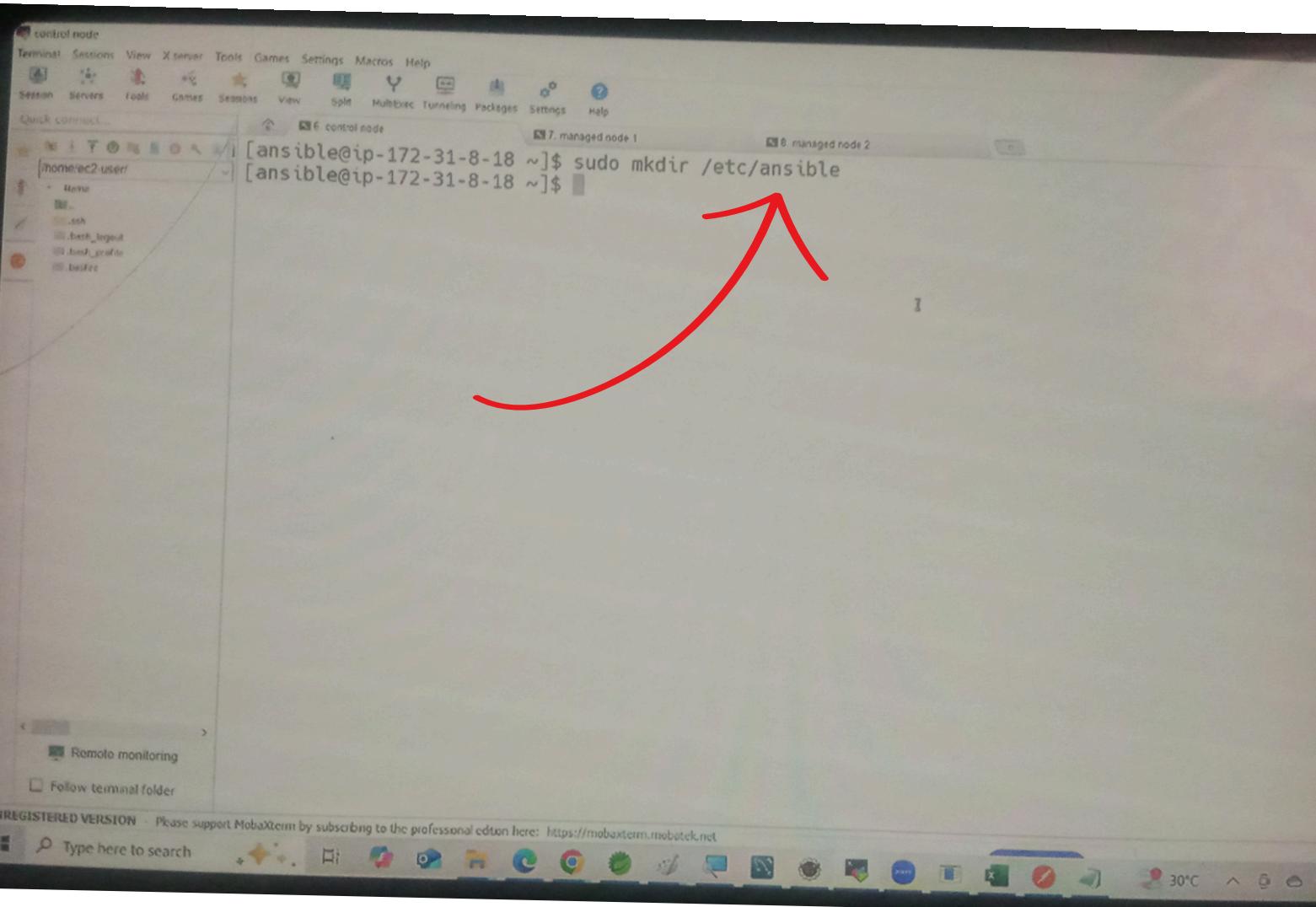
```
[ansible@ip-172-31-8-18 ~]$ pip3 install ansible --user
Collecting ansible
  Downloading ansible-8.7.0-py3-none-any.whl (48.4 MB)
    |██████████| 29.1 MB 14.3 MB/s eta 0:00:02
```

f) Verify ansible version

ansible --version

g) Create ansible folder under /etc

sudo mkdir /etc/ansible



Step-4: Generate SSH Key In your Control Node and Copy that SSH key into Managed Nodes

a) Switch to ansible user

```
sudo su ansible
```

b) Generate ssh key using below command

```
ssh-keygen
```

c) Copy it to Managed Nodes as ansible user

```
ssh-copy-id ansible@<ManagedNode-Private-IP-address>
```

Note: Repeat above command by updating HOST IP for all the managed Servers.

a) Switch to ansible user
sudo su ansible

we already in ansible user
no need to switch again

b) Generate ssh key using below command
ssh-keygen

The screenshot shows a terminal window in MobaXterm with three tabs: "control node", "managed node 1", and "managed node 2". The "control node" tab is active, displaying the output of the ssh-keygen command. A red arrow points from the text "Generating public/private rsa key pair." to the command itself. The terminal output is as follows:

```
[ansible@ip-172-31-8-18 ~]$ sudo mkdir /etc/ansible
[ansible@ip-172-31-8-18 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
Created directory '/home/ansible/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:aGBnGyANeeBDY8vzW6zISylrbxLrWM3QUb46c30Wzws ansible@ip-172-31-8-18.ap-south-1.compute.internal
The key's randomart image is:
+---[RSA 3072]---+
| = . |
| = B |
| X o |
| . O + |
| . o X S . |
| . = O O + |
| oo@ + + E o |
| O++.= = . . |
| .OO+o . . |
+---[SHA256]---+
[ansible@ip-172-31-8-18 ~]$
```

At the bottom of the screen, there is a taskbar with various icons and system status information.

go to managed node in aws account and copy private ip address

The screenshot shows the AWS EC2 Instances page. On the left sidebar, under the 'Instances' section, there is a link labeled 'Elastic Block Store'. A red arrow points from this link to the instance details for 'managed node 1'. In the instance details, there is a message 'Private IPv4 address copied' with a copy icon. Another red arrow points from this message to the private IP address field, which contains '172.31.10.212'.

copy both managed node 1 and 2 private ip address and copy ip address goto next step

c) Copy it to Managed Nodes as ansible user

ssh-copy-id ansible@<ManagedNode-Private-IP-address>

copying managed node 1 ip address

```
[ansible@ip-172-31-8-18 ~]$ ssh-copy-id ansible@172.31.10.212
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
The authenticity of host '172.31.10.212 (172.31.10.212)' can't be established.
ED25519 key fingerprint is SHA256:RUWIwyvM/2FNRG7kVRv/cCBxPJAHg/rti4xXvy+oPw.
This key is not known by any other names
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 t
he new keys
ansible@172.31.10.212's password:
Permission denied, please try again.
ansible@172.31.10.212's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@172.31.10.212'" and check to make sure that only the key(s) you wanted were added.

[ansible@ip-172-31-8-18 ~]$ ssh-copy-id ansible@172.31.6.255
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
The authenticity of host '172.31.6.255 (172.31.6.255)' can't be established.
ED25519 key fingerprint is SHA256:dsrw0KFYkt+VMP5Xv8h9WwUV20LyzJ0AqavJ0L3FPxo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

copying managed node 2 ip address

Note: Repeat above command by updating HOST IP for all the managed Servers.

Step-4: Update Host Inventory in Ansible Server to add managed node servers details

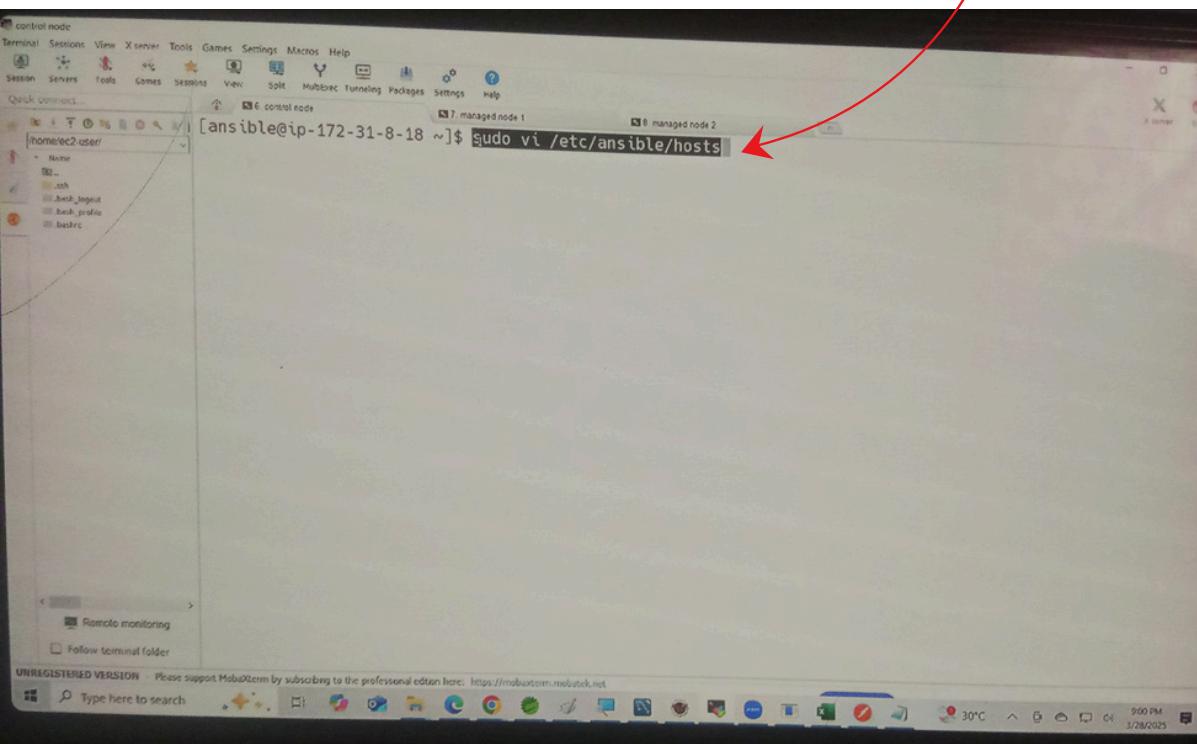
```
sudo vi /etc/ansible/hosts
```

[webservers]

192.31.0.247

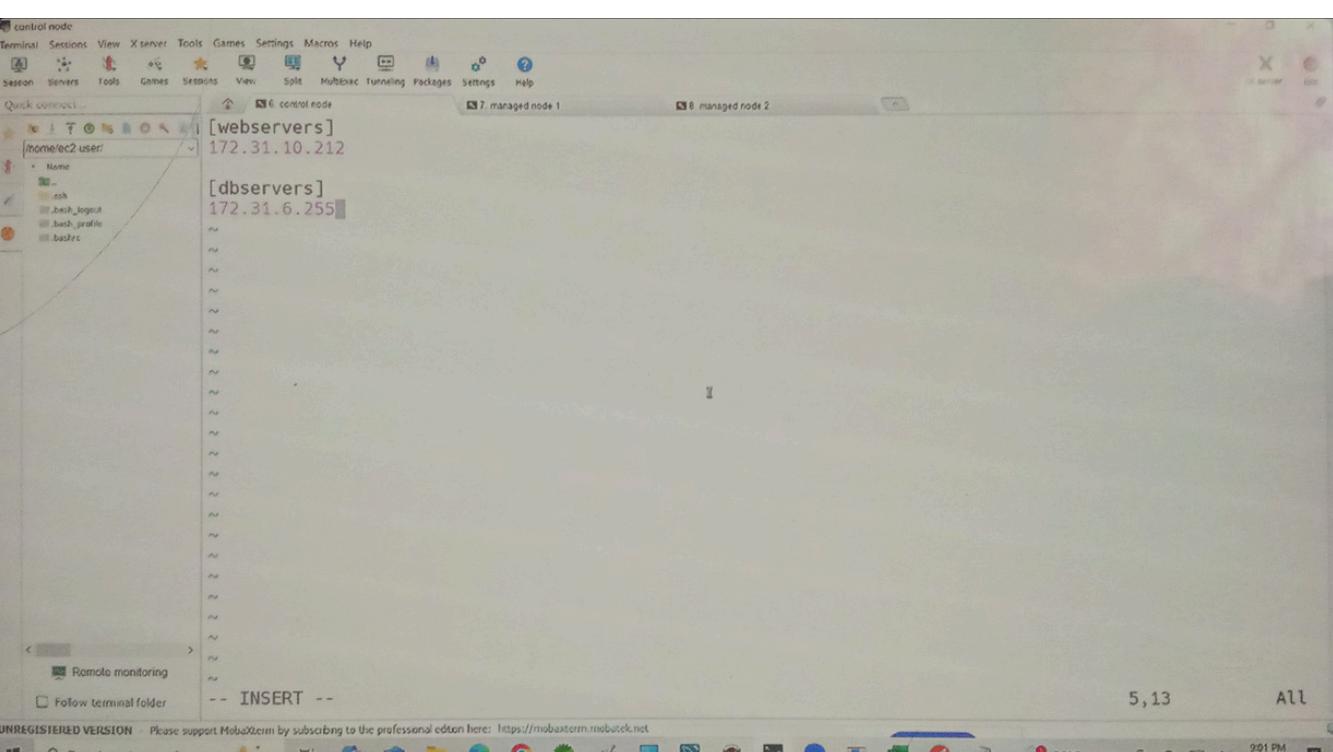
[dbservers]

192.31.0.17



Add here those ip address where we copied from aws ec2 managed node

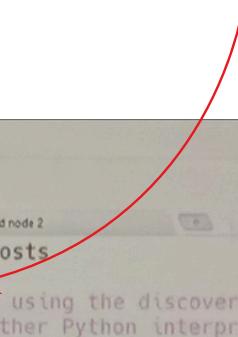
add here only managed node ip address



Step-6: Test Connectivity

ansible all -m ping

ansible all -m ping



```
[ansible@ip-172-31-8-18 ~]$ sudo vi /etc/ansible/hosts
[ansible@ip-172-31-8-18 ~]$ ansible all -m ping
[WARNING]: Platform linux on host 172.31.10.212 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.10.212 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.6.255 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of
that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.6.255 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[ansible@ip-172-31-8-18 ~]$
```

REGISTERED VERSION · Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

Type here to search

9:01 PM 3/28/2025



How to Configure Ansible?

#####
Step-1: Create 3 instances of t2.micro Amazon Linux VMs in AWS.

- a. 1 for - Control Node
- b. 2 for - Managed Nodes

Note: Connect to all 3 VMs using MobaXterm

Step-1: Do the following setup in all 3 machines

- a) Create user:

```
sudo useradd ansible  
sudo passwd ansible
```

- b) Configure user in sudoers file

```
sudo visudo  
ansible ALL=(ALL) NOPASSWD: ALL
```

- c) Update sshd config file

```
sudo vi /etc/ssh/sshd_config
```

-> change PasswordAuthentication "no" and PermitEmptyPasswords "yes"

- d) Restart the server

```
sudo service sshd restart
```

Step-2: Install Ansible in Control Node

- a) Switch to Ansible user

```
sudo su ansible  
cd ~
```

- b) Install Python because ansible is developed using python & to run ansible we require python software

```
sudo yum install python3 -y
```

- c) Check python is installed?

```
python3 --version
```

- d) Install PIP (package manager) for downloading softwares and installing it

```
sudo yum -y install python3-pip
```

- e) Install Ansible using Python PIP

```
pip3 install ansible --user
```

- f) Verify ansible version

```
ansible --version
```

- g) Create ansible folder under /etc

```
sudo mkdir /etc/ansible
```

Step-3: Generate SSH Key In your Control Node and Copy that SSH key into Managed Nodes

- a) Switch to ansible user

```
sudo su ansible
```

- b) Generate ssh key using below command

```
ssh-keygen
```

- c) Copy it to Managed Nodes as ansible user

```
ssh-copy-id ansible@<ManagedNode-Private-IP-address>
```

Note: Repeat above command by updating HOST IP for all the managed Servers.

Step-4: Update Host Inventory in Ansible Server to add managed node servers details

```
sudo vi /etc/ansible/hosts
```

```
[webservers]
```

```
192.31.0.247
```

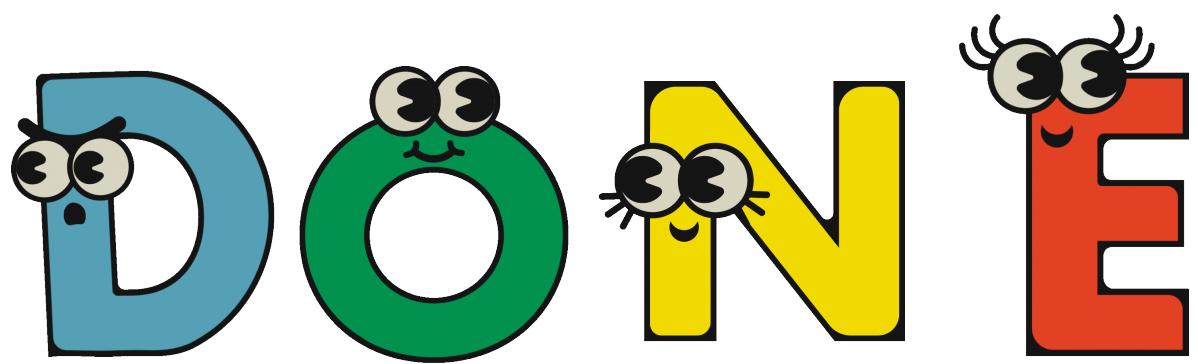
```
[dbservers]
```

```
192.31.0.17
```

Step-5: Test Connectivity

```
ansible all -m ping
```

D **O** **N** **E**

The word "DONE" is written in large, bold, colorful letters. Each letter has a cartoonish face with two large, round eyes and a small smile. The 'D' is blue, the 'O' is green, the 'N' is yellow, and the 'E' is red. The letters are arranged horizontally, centered on the page.