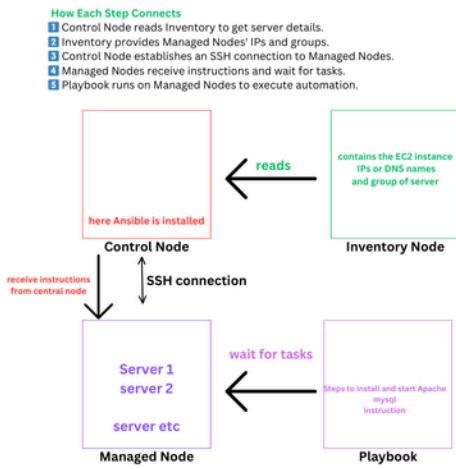


How to Configure Ansible?

overview what we are going todo



Architecture



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

- 1 for - Control Node
- 2 for - Managed Nodes

Note: Connect to all 3 VMs using MobaXterm

create 3 EC2 instance

The screenshot shows the AWS EC2 Instances Launch wizard. In the 'Summary' section, there is a red circle around the 'Number of instances' field which contains the value '3'. Below it, a note says 'When launching more than 1 instance, consider EC2 Auto Scaling'. Other visible fields include 'Software Image (AMI)', 'Virtual server type (instance type)' set to 't2.micro', and 'Firewall (security group)'. At the bottom right is a large orange 'Launch instance' button.

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 purple box highlights the table area, and three red arrows point to the "control node", "managed node 1", and "managed node 2" rows.

Note: Connect to all 3 VMs using MobaXterm

The screenshot shows the AWS EC2 Instances page. A green notification bar at the top indicates "Successfully initiated termination (deletion) of i-0595e4e5b3c23ea82". Below this, a table lists four instances. One instance, "control node" (i-0a281dd77dce3f292), is highlighted and has a checkmark next to it. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. The "control node" is listed as "Running" with "t2.micro" type and "Initializing" status.

step 1: select control node

This screenshot is similar to the one above, but the "control node" instance is now selected. A tooltip "Public IPv4 address copied" appears over the "Private IPv4 addresses" section, which lists "172.31.8.18". The "Public IPv4 DNS" section also displays "172.31.8.18".

step 2: copy private ip address

The screenshot shows the "Session settings" dialog box from MobaXterm. Under "Basic SSH settings", the "Remote host" field is set to "172.31.8.18". The "Specify username" dropdown is set to "ec2-user". The "Port" dropdown is set to "22". The "Advanced SSH settings" tab is selected, showing options like "X11 Forwarding", "Compression", and "SSH browser type: SFTP protocol". The "OK" button is visible at the bottom.

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

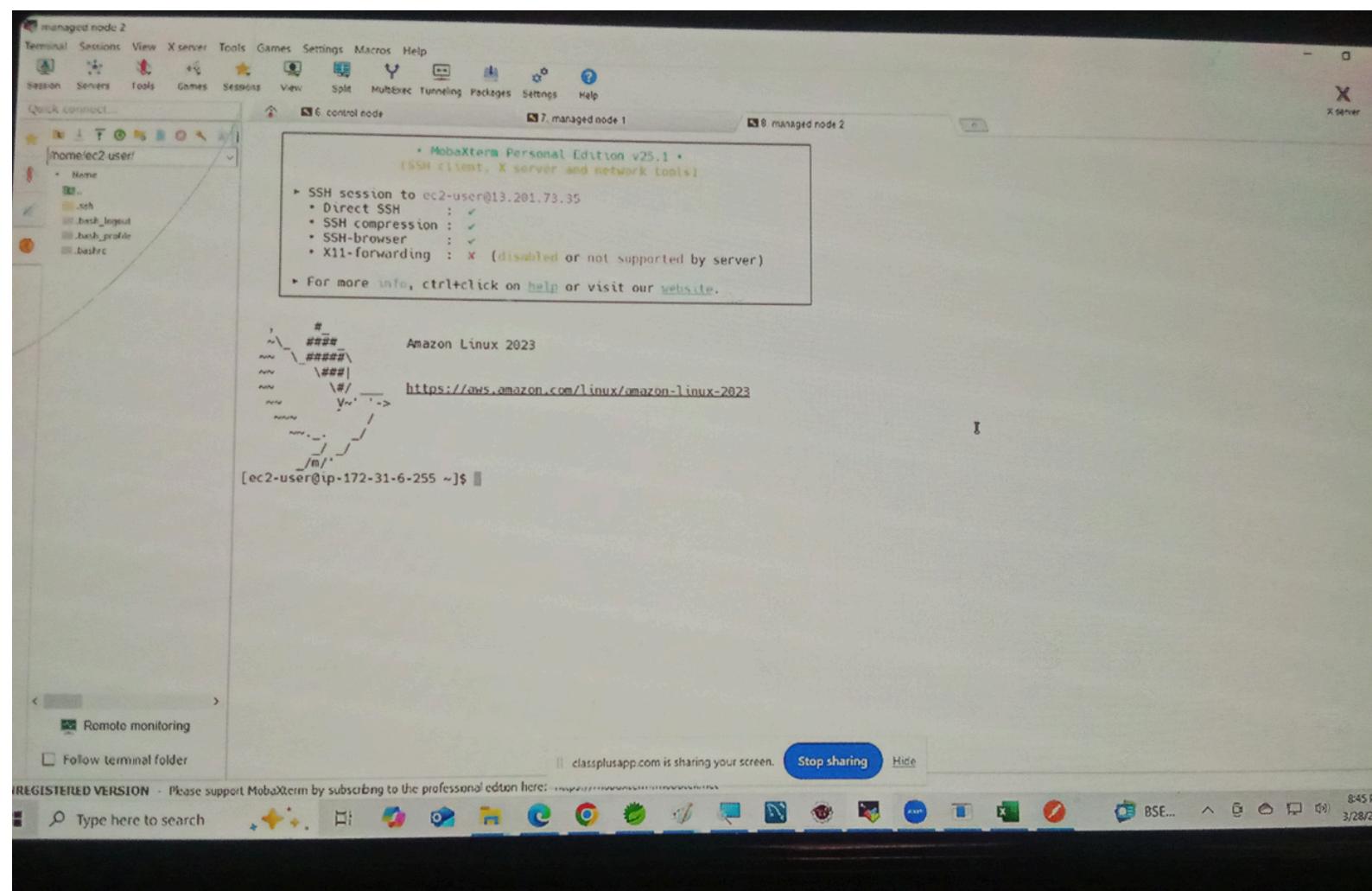
This screenshot shows the MobaXterm interface. A red arrow points from the previous "Session settings" dialog to the "Sessions" tab in the main window. A new session is being created, with a modal dialog asking "Please enter a new name for this session" and the text "control node" entered. The main terminal window shows a connection to "ec2-user@ip-172-31-8-10" on port 22, with the message "MobaXterm Personal Edition v20.1" displayed.

step 4: give name as control node



AnyScanner

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
```

The screenshot shows a MobaXterm window titled "control node" with session icons for "control node", "managed node 1", and "managed node 2". The terminal window displays the following output:

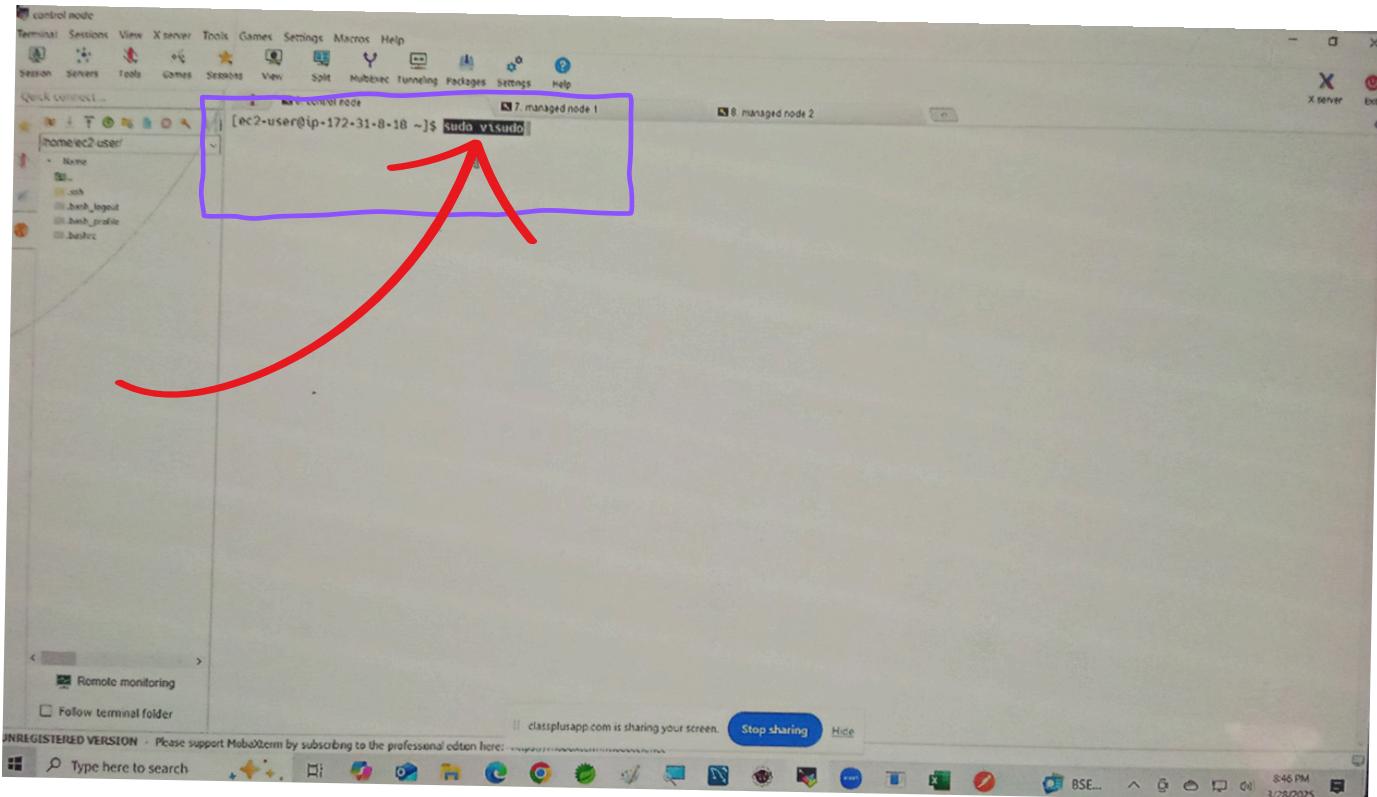
```
MobaXterm Personal Edition v25.1 •  
(SSH client, X server and network tools)  
SSH session to ec2-user@15.200.148.37  
• Direct SSH : ✓  
• SSH compression : ✓  
• SSH-browser : ✓  
• X11-forwarding : ✘ (disabled or not supported by server)  
For more info, ctrl+click on help or visit our website.  
#  
### Amazon Linux 2023  
###  
### https://aws.amazon.com/linux/amazon-linux-2023  
###  
[ec2-user@ip-172-31-8-18 ~]$ sudo useradd ansible  
sudo passwd ansible  
Changing password for user ansible.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[ec2-user@ip-172-31-8-18 ~]$
```

At the bottom of the terminal window, there is a message: "classplusapp.com is sharing your screen." with "Stop sharing" and "Hide" buttons. The taskbar at the bottom shows various application icons.

b) Configure user in sudoers file

sudo visudo

ansible ALL=(ALL) NOPASSWD: ALL



```
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
ansible ALL=(ALL) NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys  ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

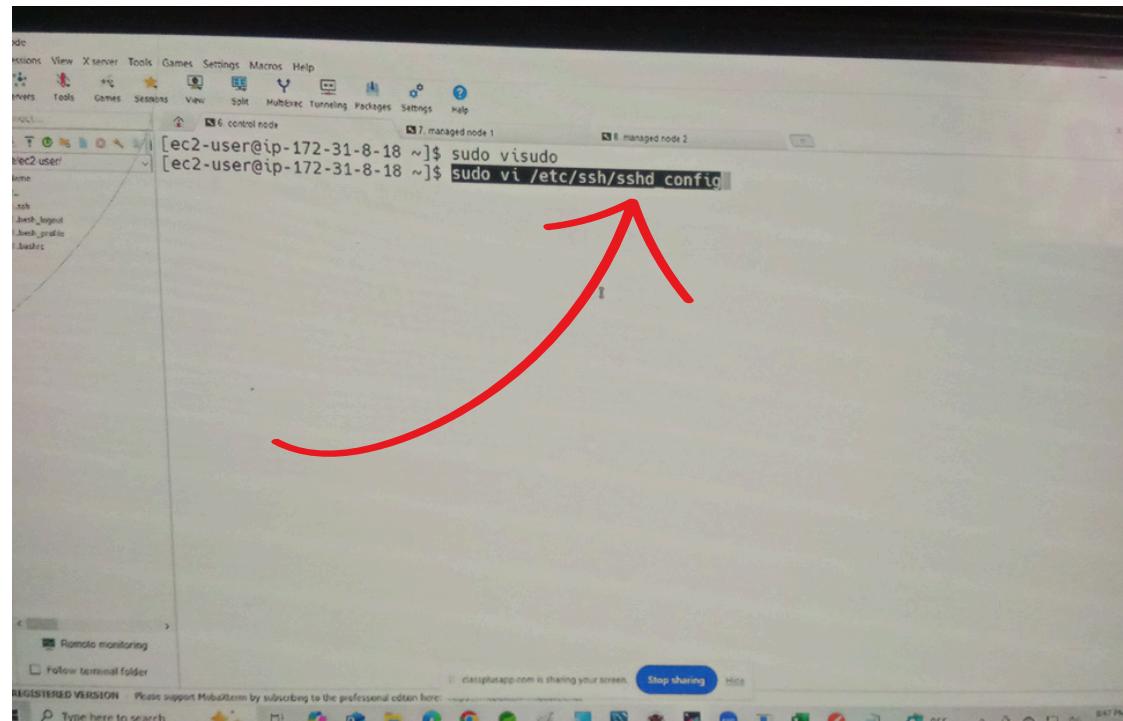
## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system

```

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
#GSSAPIEnablek5users 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`.

Session 6, titled "control node", shows a file browser window with the path `home/ec2 user/.ssh`. It lists files: `.ssh`, `.bash_logout`, `.bash_profile`, and `.bashrc`.

The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and several pinned applications.

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 a MobaXterm interface with multiple sessions. Session 6, titled "control node", is active and displays the following terminal session:

```
[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 "cd ~" command to the "pwd" command, highlighting the directory change.

Session 7, titled "managed node 1", and Session 8, titled "managed node 2", are also visible in the background.

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

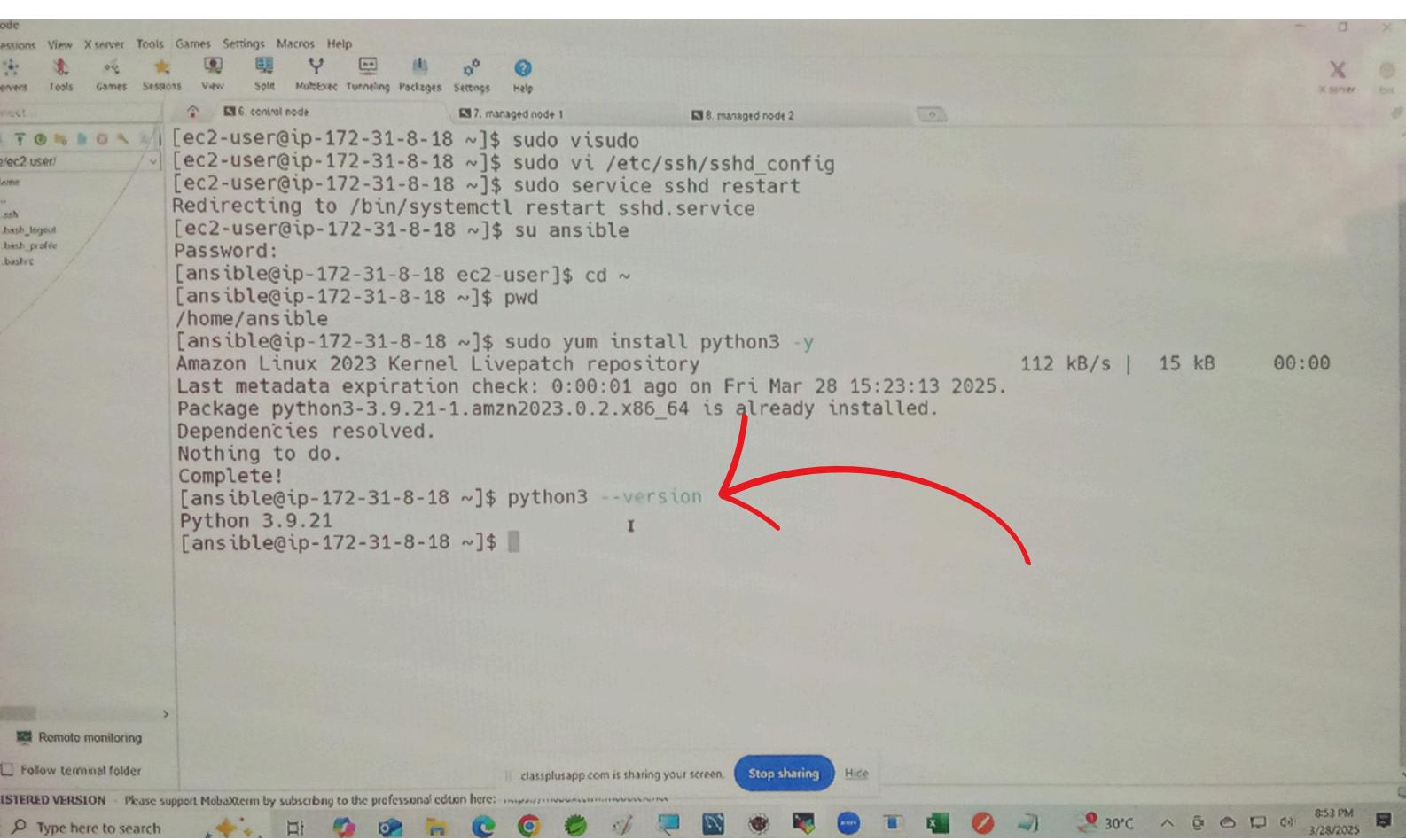
sudo yum install python3 -y

The screenshot shows a MobaXterm window with four tabs: control node, managed node 1, managed node 2, and managed node 3. The managed node 3 tab is active, displaying a terminal session. A red arrow points from the top right towards the sudo yum command in the session.

```
[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 ~]$
```

c) cross Check python is installed?

`python3 --version`



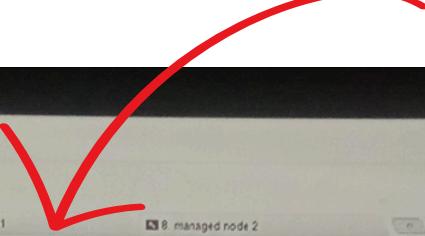
A screenshot of a terminal window titled "control node" showing the following command and its output:

```
[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

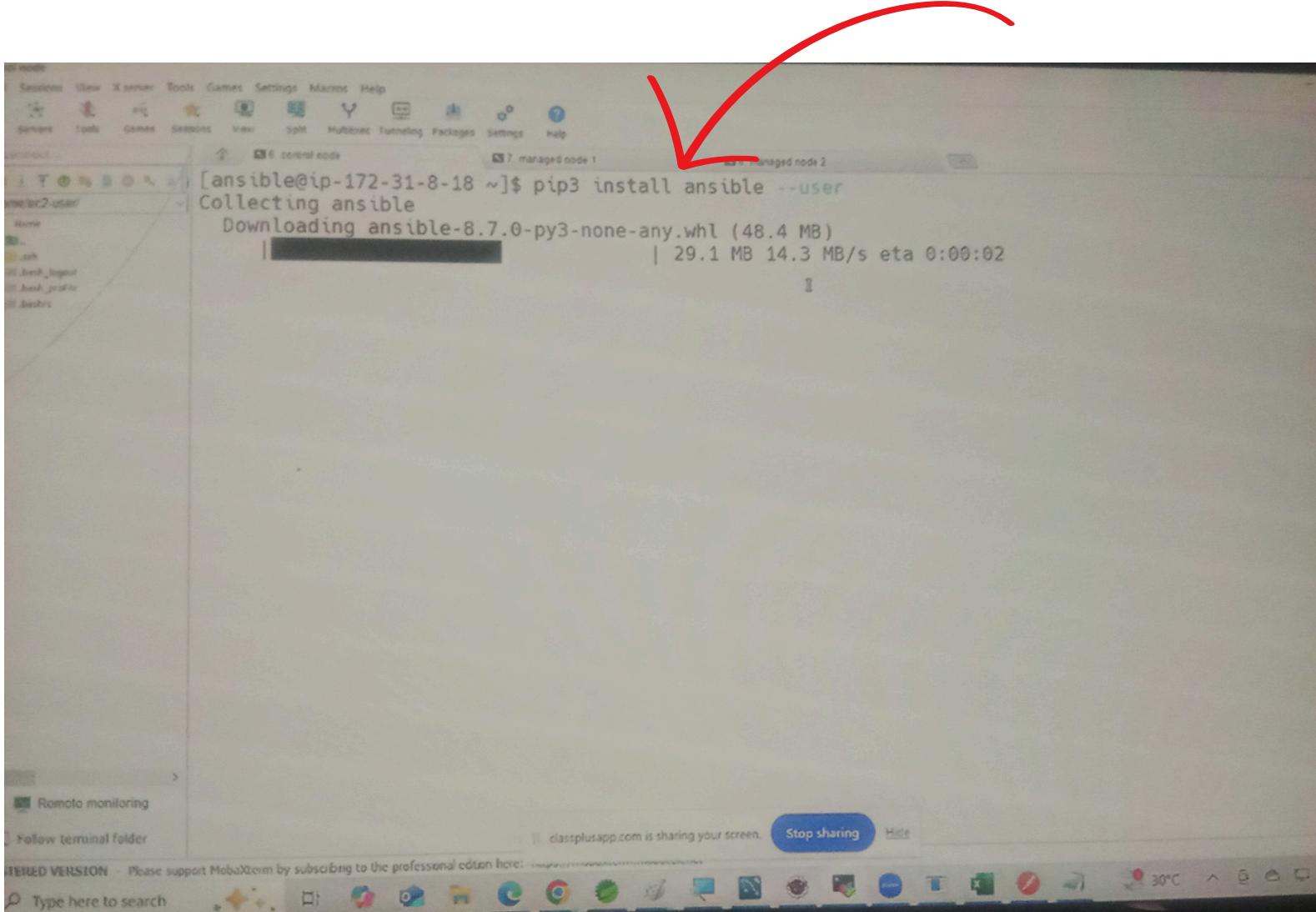
Transaction Summary
=====
Install 2 Packages

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
```

e) Install Ansible using Python PIP

pip3 install ansible --user



A screenshot of a MobaXterm terminal window. The terminal interface includes a menu bar with Sessions, View, Xserver, Tools, Games, Settings, Macros, Help, and several icons for file operations like Open, Save, Print, and Settings. Below the menu is a toolbar with buttons for Servers, Tools, Games, Sessions, View, Split, Multisession, Packages, Settings, and Help. The main terminal area shows a command-line session:

```
[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
```

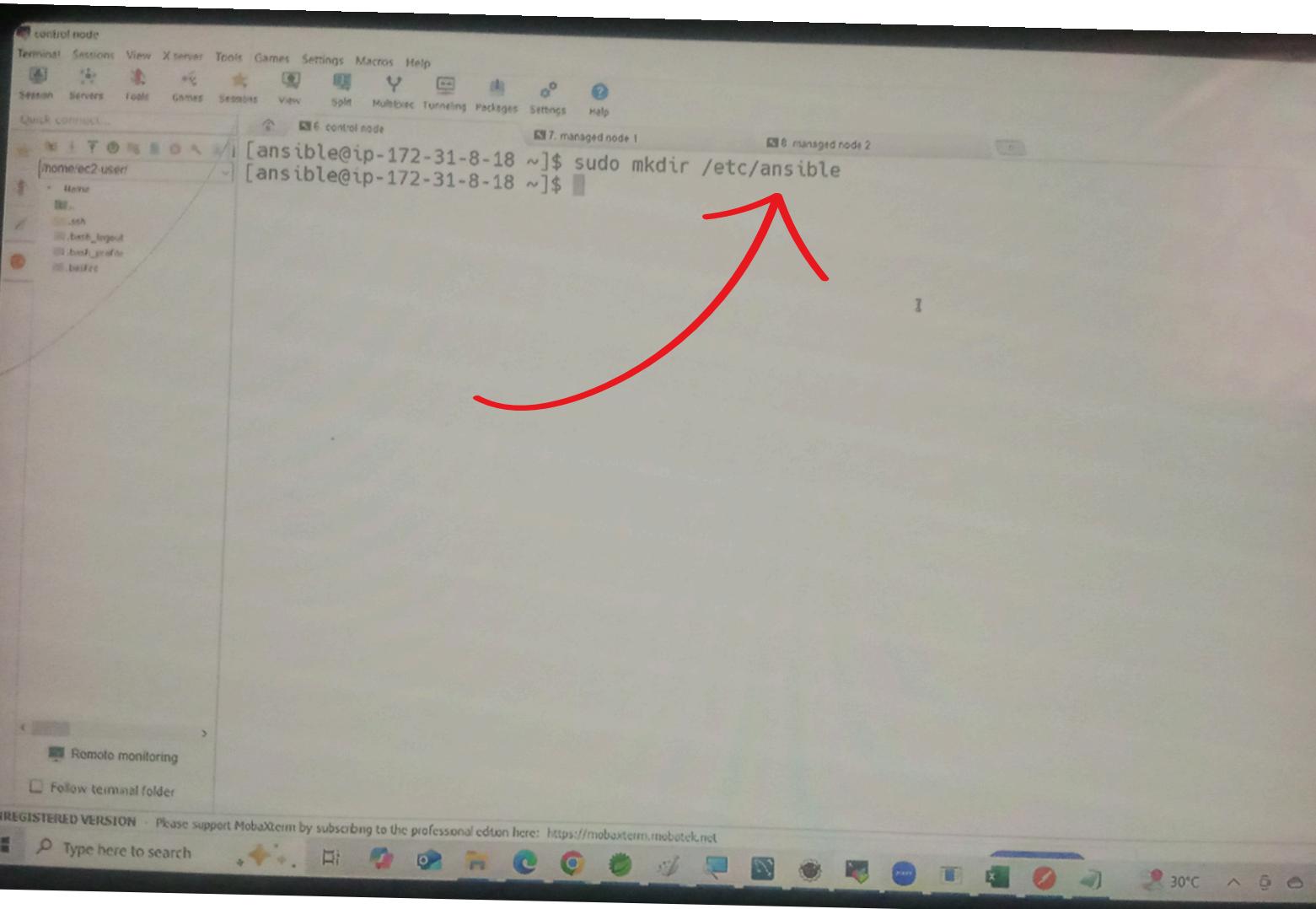
A red arrow points from the top right towards the progress bar of the download command. The status bar at the bottom of the terminal window indicates "classplusapp.com is sharing your screen." and includes links for "Stop sharing" and "Hide". The taskbar at the bottom of the screen shows various application icons and the system clock.

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 multiple sessions listed at the top: 6 control node, 7 managed node 1, 8 managed node 2, and 9. The current session is 7 managed node 1. 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 ~]$
```

A red arrow points from the text "we already in ansible user no need to switch again" to the line "sudo su ansible". Another red arrow points from the text "Generating public/private rsa key pair." to the command "ssh-keygen".

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

AWS EC2 Instances page showing three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
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

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/cCBxPJHAGh/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+VMPSXv8h9WwUV2QLyzJ0AqavJ0L3FPxo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

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

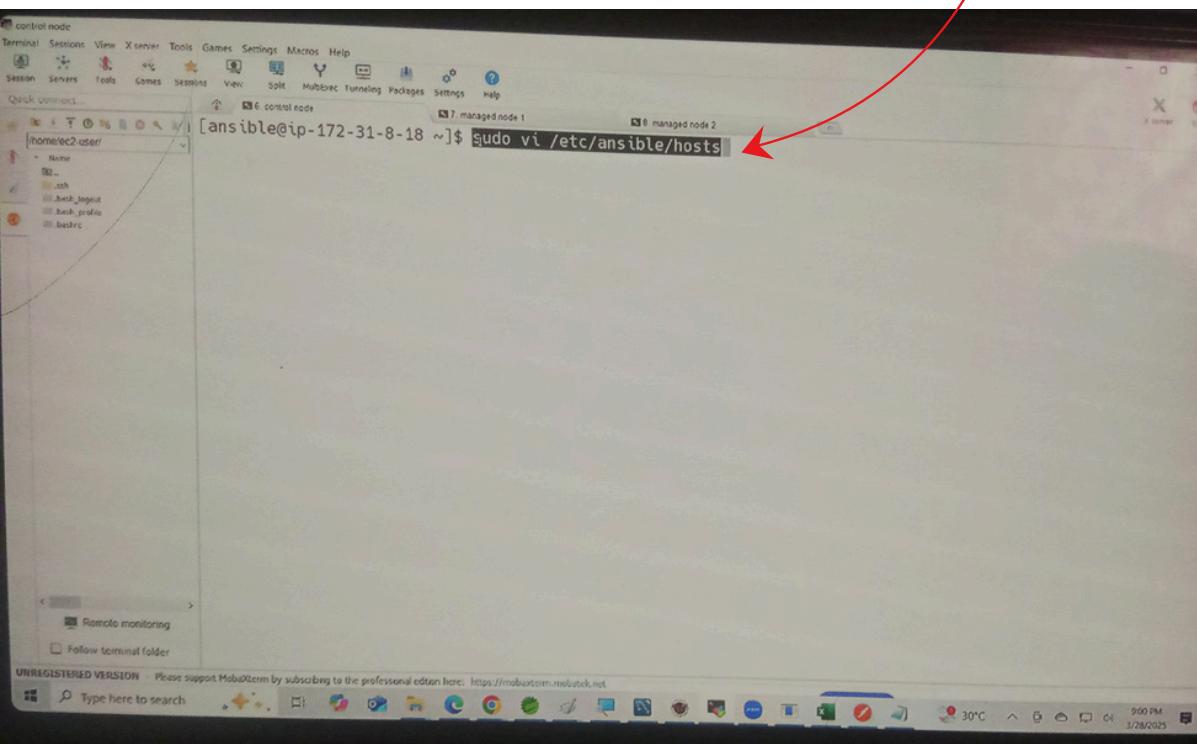
```
sudo vi /etc/ansible/hosts
```

[webservers]

192.31.0.247

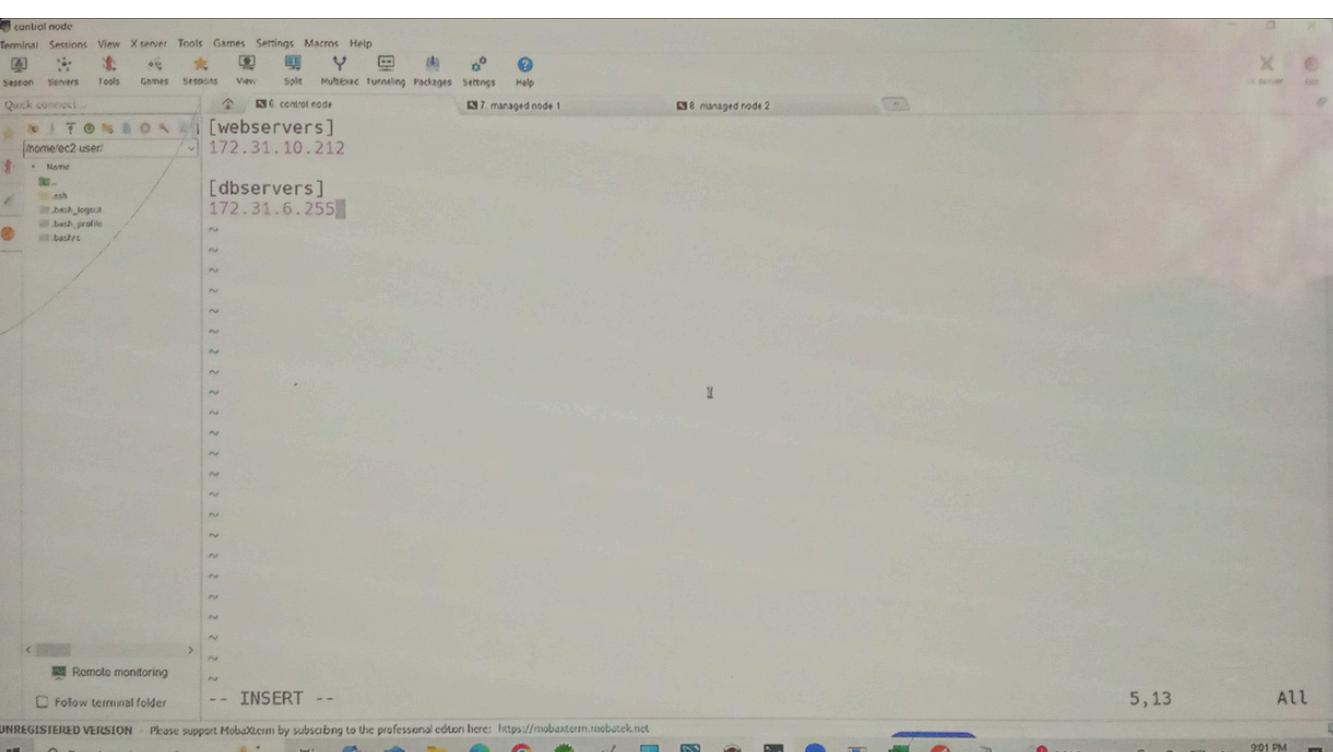
[dbservers]

192.31.0.17



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

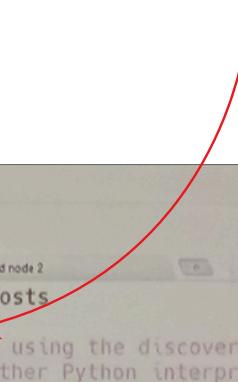
add here only managed node ip address



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

901 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
```

Create playbook

in control node

- check **pwd** if it is not ansible
- use **cd ~** it will point ansible **pwd** as shown below
- then create playbook **vi name.yml**

image 1

A screenshot of a terminal window titled "3.10.53.31 (ec2-user)". The terminal shows the following command sequence:

```
[ansible@ip-172-31-8-18 ec2-user]$ pwd
/home/ec2-user
[ansible@ip-172-31-8-18 ec2-user]$ cd ~
[ansible@ip-172-31-8-18 ~]$ pwd
/home/ansible
[ansible@ip-172-31-8-18 ~]$ vi book01.yml
```

The terminal window has a red box around the command "cd ~" and another red box around the command "vi book01.yml".

now you write inside vi name.yml

A screenshot of a terminal window titled "3.10.53.31 (ec2-user)". The terminal shows the following YAML configuration:

```
hosts: all
  tasks:
    - name: ping all managed nodes
      ping:
```

The entire configuration block is highlighted with a red box.

- hosts: all
 tasks:
 - name: ping all managed nodes
 ping:
...

Now run your playbook

A screenshot of a terminal window titled "3.10.53.31 (ec2-user)". The terminal shows the command "ansible-playbook book-01.yml" being run. The output shows the playbook execution:

```
PLAY [all] ****
TASK [Gathering Facts] ****
[WARNING]: Platform linux on host 172.31.6.255 is using the discovered Python interpreter at /usr/bin/python2.9, 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.
ok: [172.31.6.255]
[WARNING]: Platform linux on host 172.31.10.212 is using the discovered Python interpreter at /usr/bin/python2.9, 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.
ok: [172.31.10.212]

TASK [ping all managed nodes] ****
ok: [172.31.6.255]
ok: [172.31.10.212]

PLAY RECAP ****
172.31.10.212 : ok=2    changed=0   unreachable=0   failed=0    skipped=0   rescued=0
172.31.6.255   : ok=2    changed=0   unreachable=0   failed=0    skipped=0   rescued=0
ok: [172.31.6.255]
```

ansible-playbook <playbook-yml-file>

once you run you will see all the pings what we written playbook

You can write your own playbook and as we did earlier

Example 2:

```
---  
- hosts: all  
  tasks:  
    - name: create a file  
      file:  
        path: /home/ansible/t1.txt  
        state: touch
```

...

Example 3:

```
---  
- hosts: all  
  tasks:  
    - name: copy content to file  
      copy: content="Hello world\n" dest="/home/ansible/t1.txt"
```

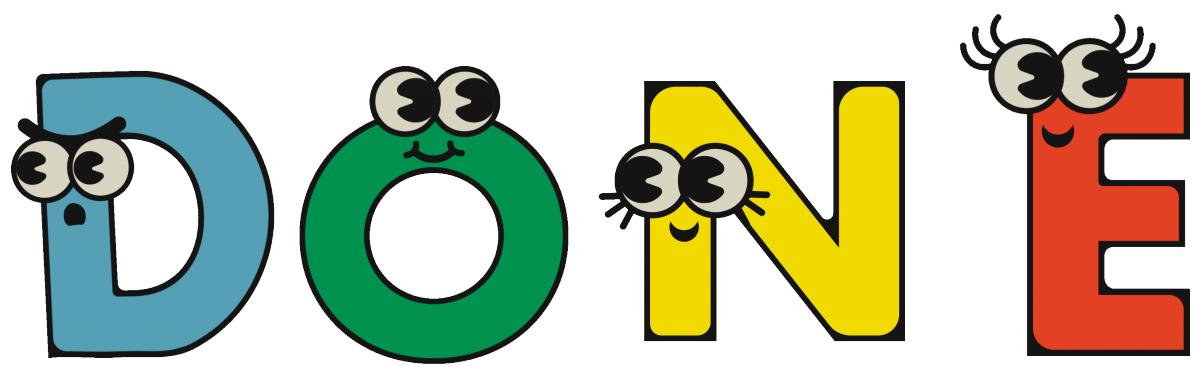
...

Example 4:

```
---  
- hosts: webservers  
  become: true #use it if you need sudo privileges  
  tasks:  
    - name: install httpd package  
      yum:  
        name: httpd  
        state: latest  
    - name: copy index.html file  
      copy:  
        src: index.html  
        dest: /var/www/html/index.html  
    - name: start httpd service  
      service:  
        name: httpd  
        state: started
```

...

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.