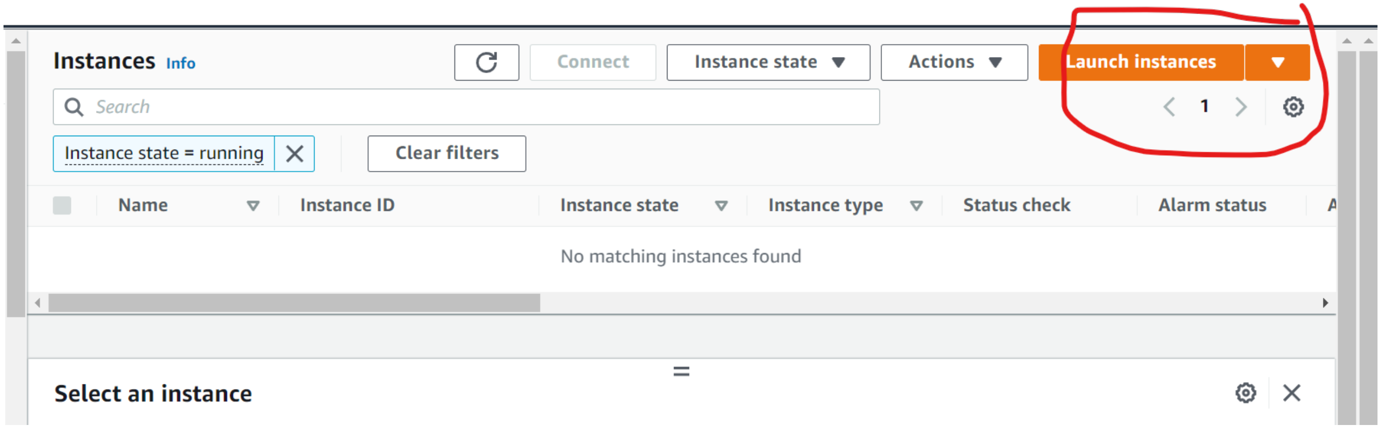
ANSIBLE LAB

To start ansible Lab we will require 3 Linux ec2-instance one for Ansible server and remaining two for nodes.

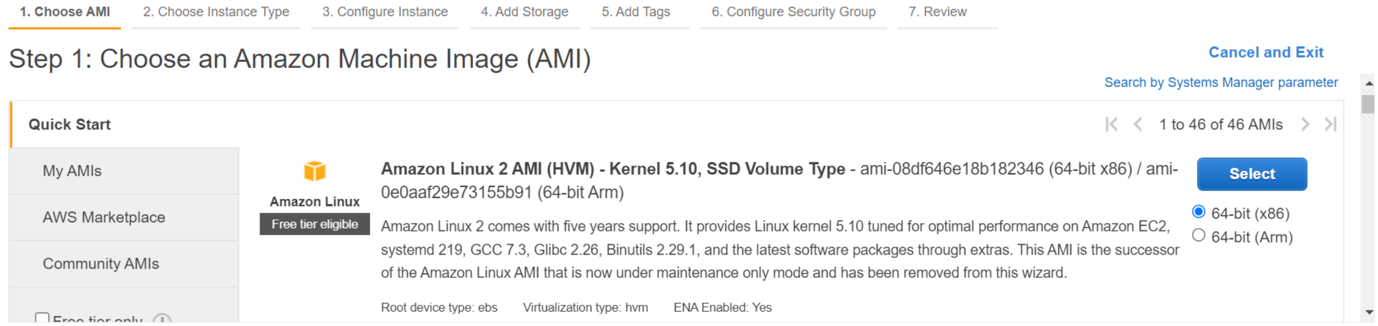
Steps:

(I) Login to aws account and create 3 linux ec2-instance in a go and in security policy attach http port 80 also.



Instance state = running ✕

No matching instances found



Step 1: Choose an Amazon Machine Image (AMI)

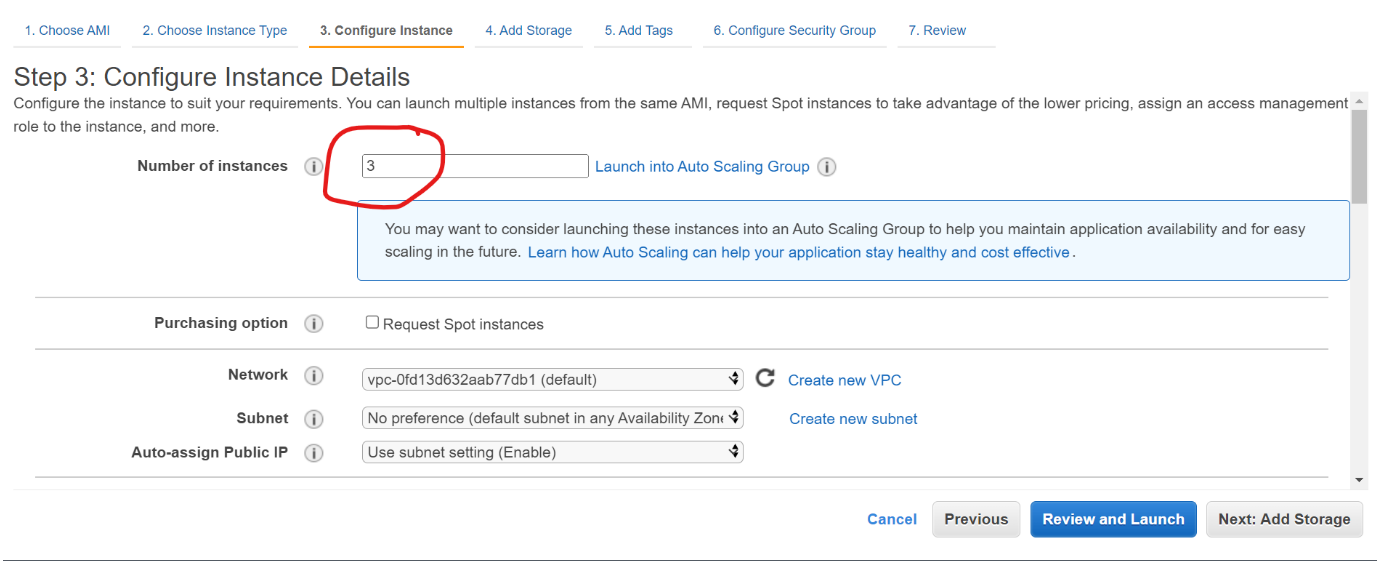
1 to 46 of 46 AMIs

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-08df646e18b182346 (64-bit x86) / ami-0e0aaf29e73155b91 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 3

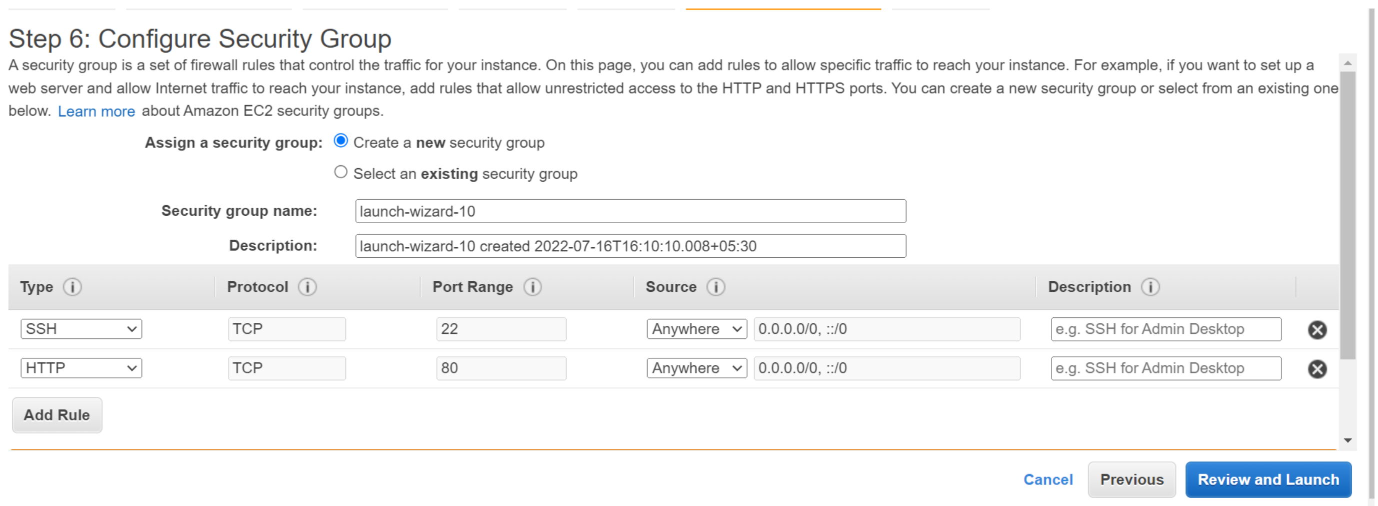
You may want to consider launching these instances into an Auto Scaling Group to help you maintain application availability and for easy scaling in the future. Learn how Auto Scaling can help your application stay healthy and cost effective.

Purchasing option: □ Request Spot instances

Network: vpc-0fd13d632aab77db1 (default)

Subnet: No preference (default subnet in any Availability Zone

Auto-assign Public IP: Use subnet setting (Enable)



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

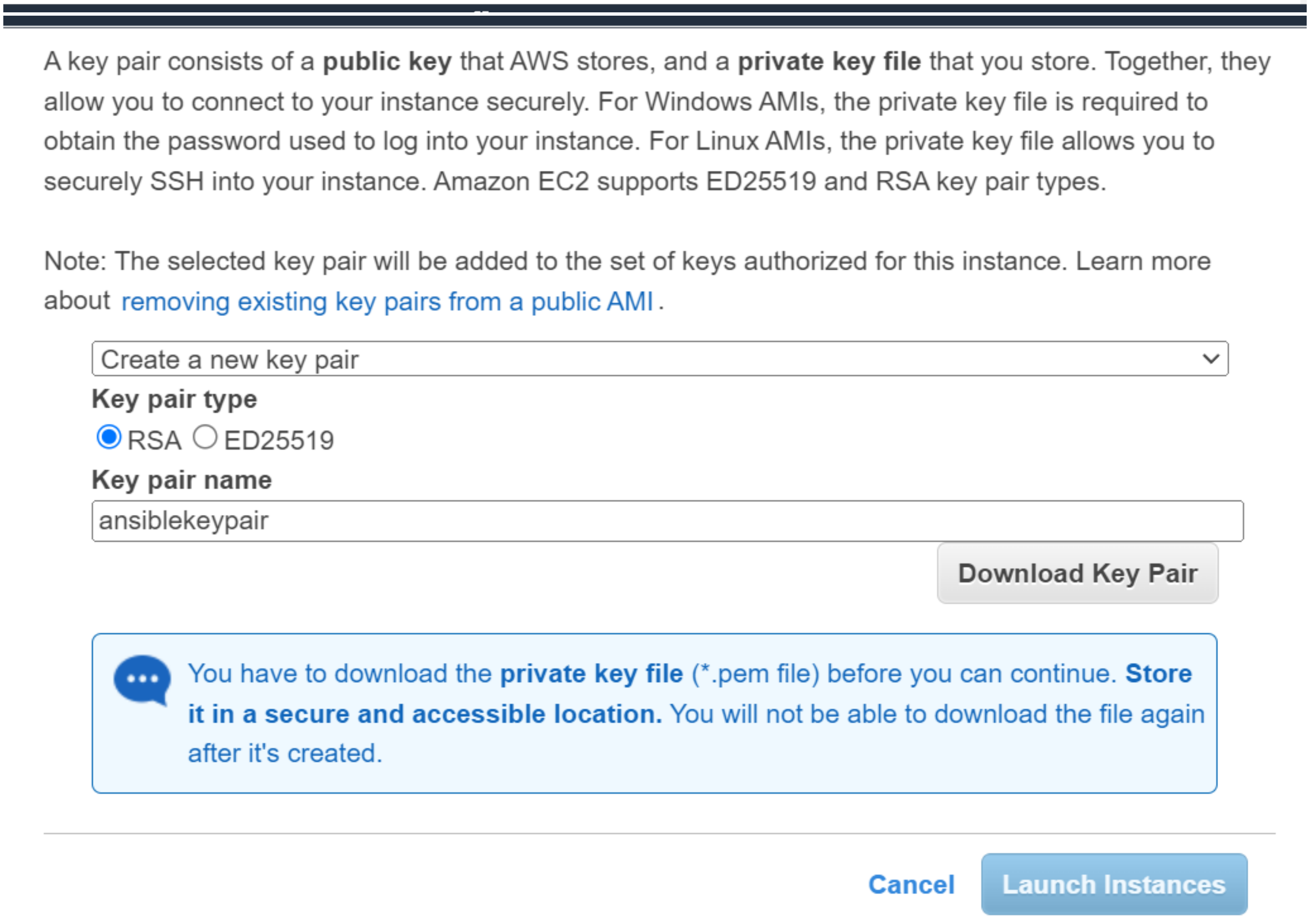
Assign a security group: ● Create a **new** security group

○ Select an **existing** security group

Security group name: launch-wizard-10

Description: launch-wizard-10 created 2022-07-16T16:10:10.008+05:30

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop



A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair

Key pair type

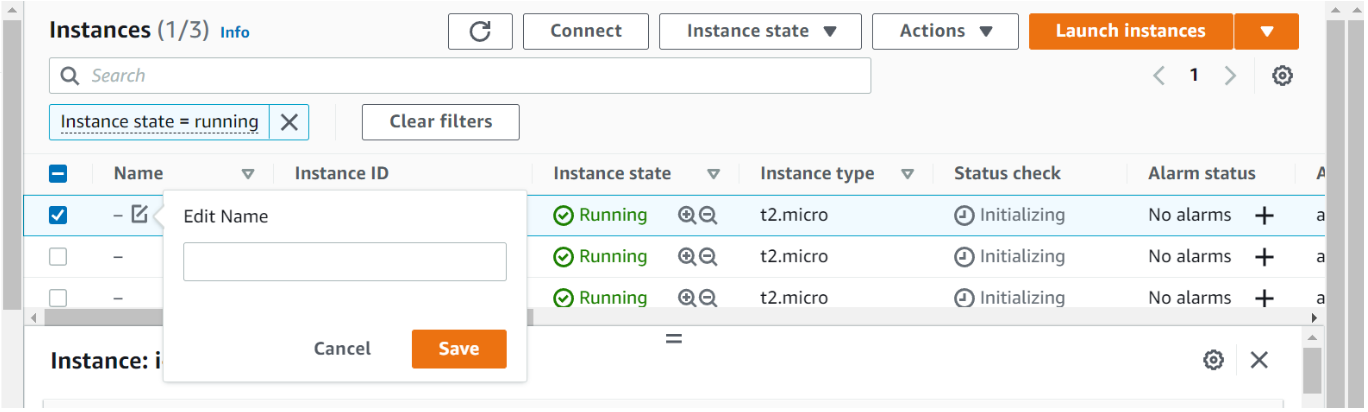
● RSA ○ ED25519

Key pair name

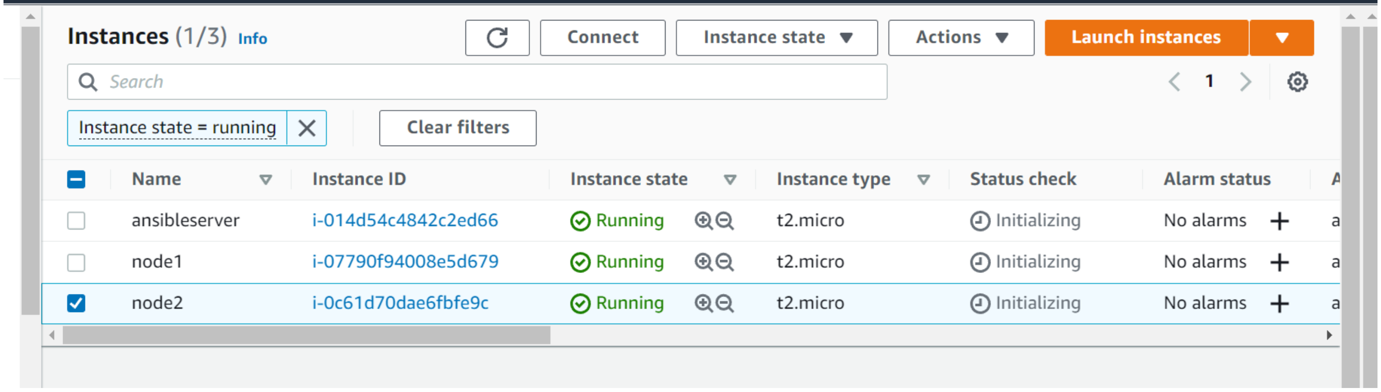
ansiblekeypair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

(ii) Rename the name like ansibleserver,node1 and node2

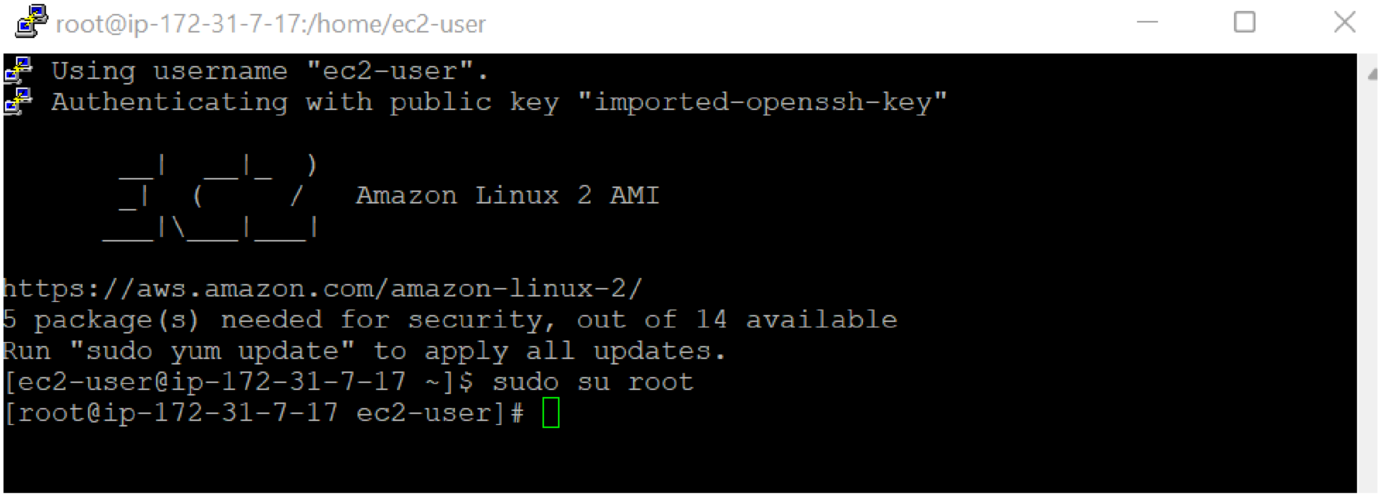


☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☑	–		Running	t2.micro	Initializing	No alarms
☐	–		Running	t2.micro	Initializing	No alarms
☐	–		Running	t2.micro	Initializing	No alarms



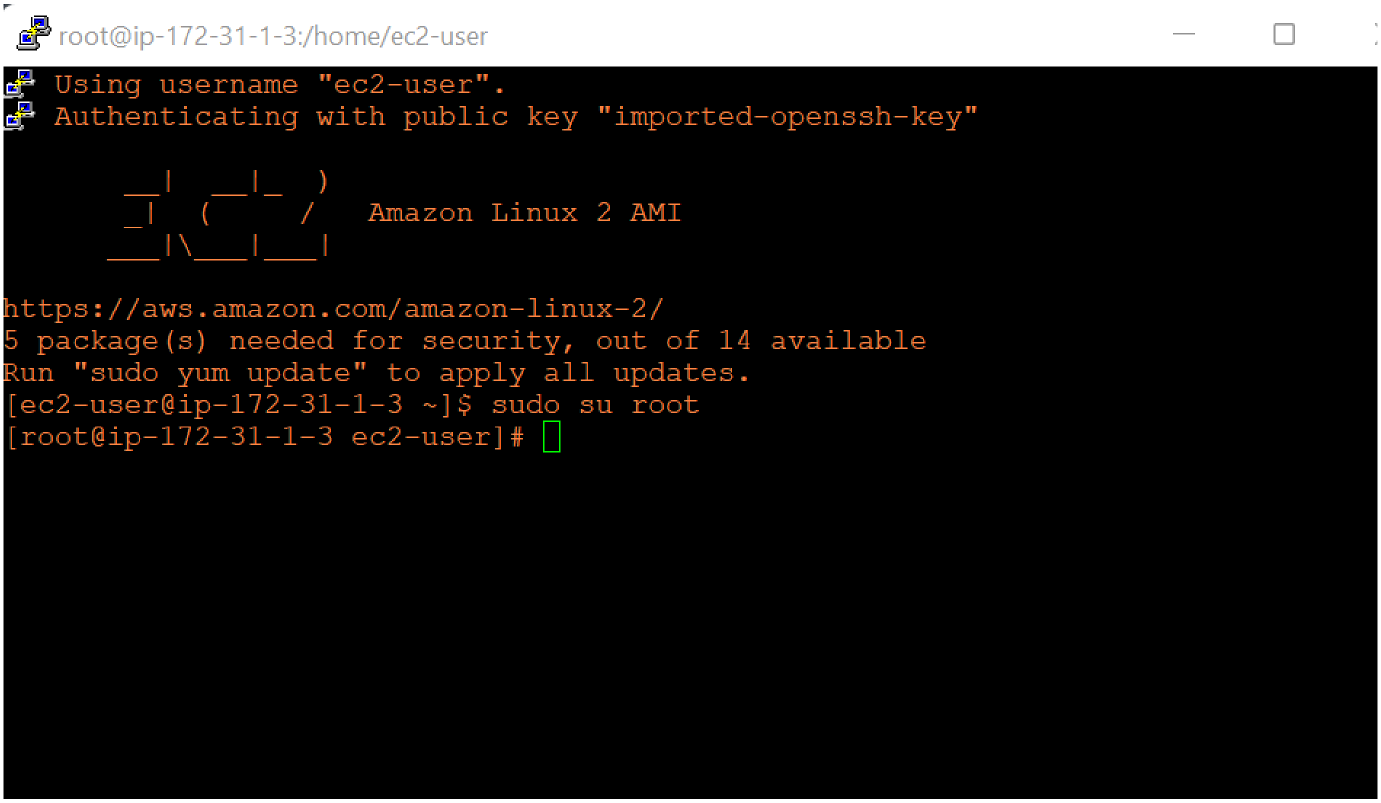
☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	ansibleserver	i-014d54c4842c2ed66	Running	t2.micro	Initializing	No alarms
☐	node1	i-07790f94008e5d679	Running	t2.micro	Initializing	No alarms
☑	node2	i-0c61d70dae6fbfe9c	Running	t2.micro	Initializing	No alarms

(iii) Now login to ansible server first. (use putty)



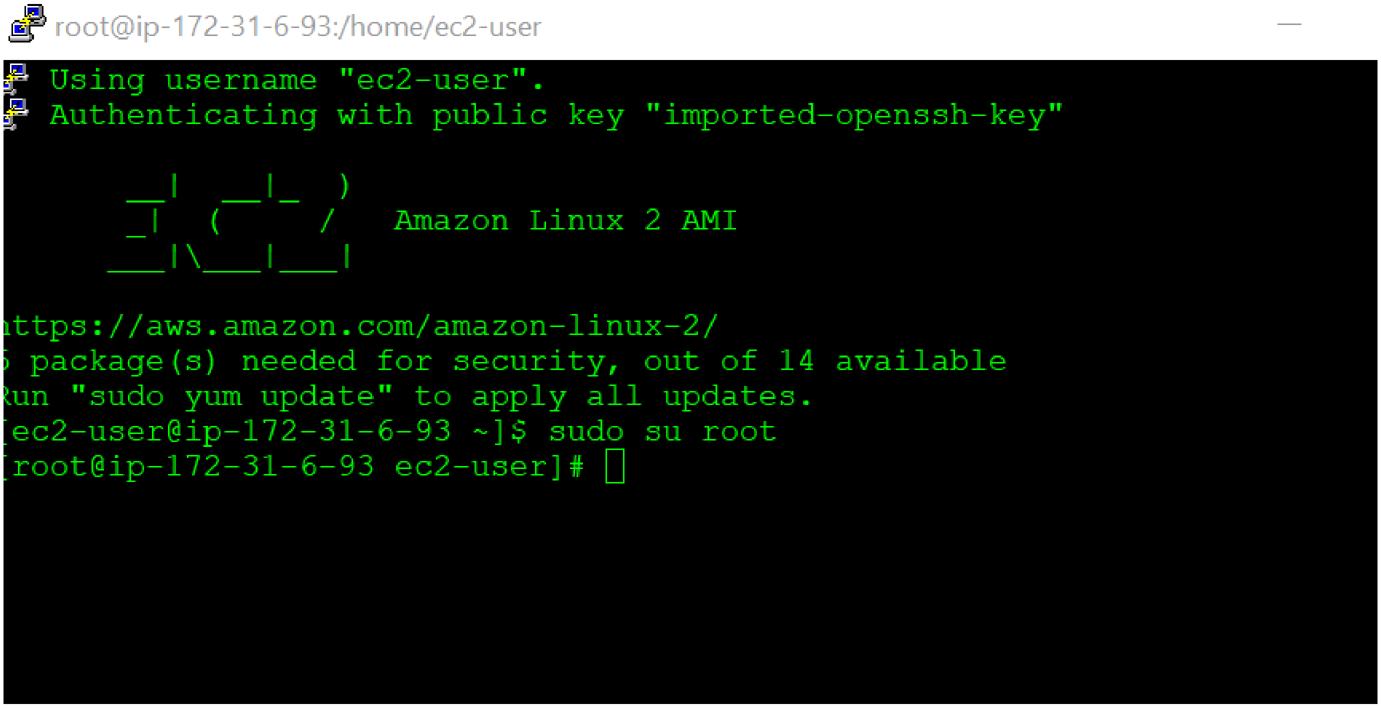
```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-7-17 ~]$ sudo su root
[root@ip-172-31-7-17 ec2-user]#
```

Similarly login to node 1 and change its color to orange



```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-3 ~]$ sudo su root
[root@ip-172-31-1-3 ec2-user]#
```

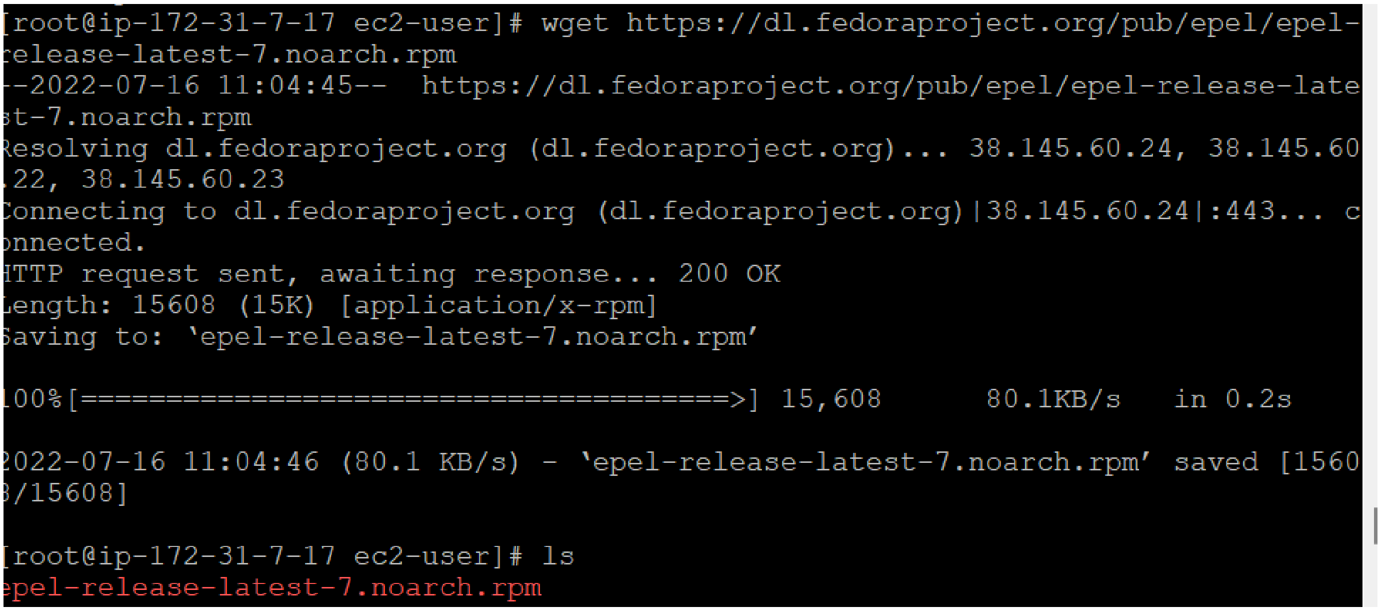
Similarly login to node 2 and change its color to green.



```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

ttps://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-93 ~]$ sudo su root
[root@ip-172-31-6-93 ec2-user]#
```

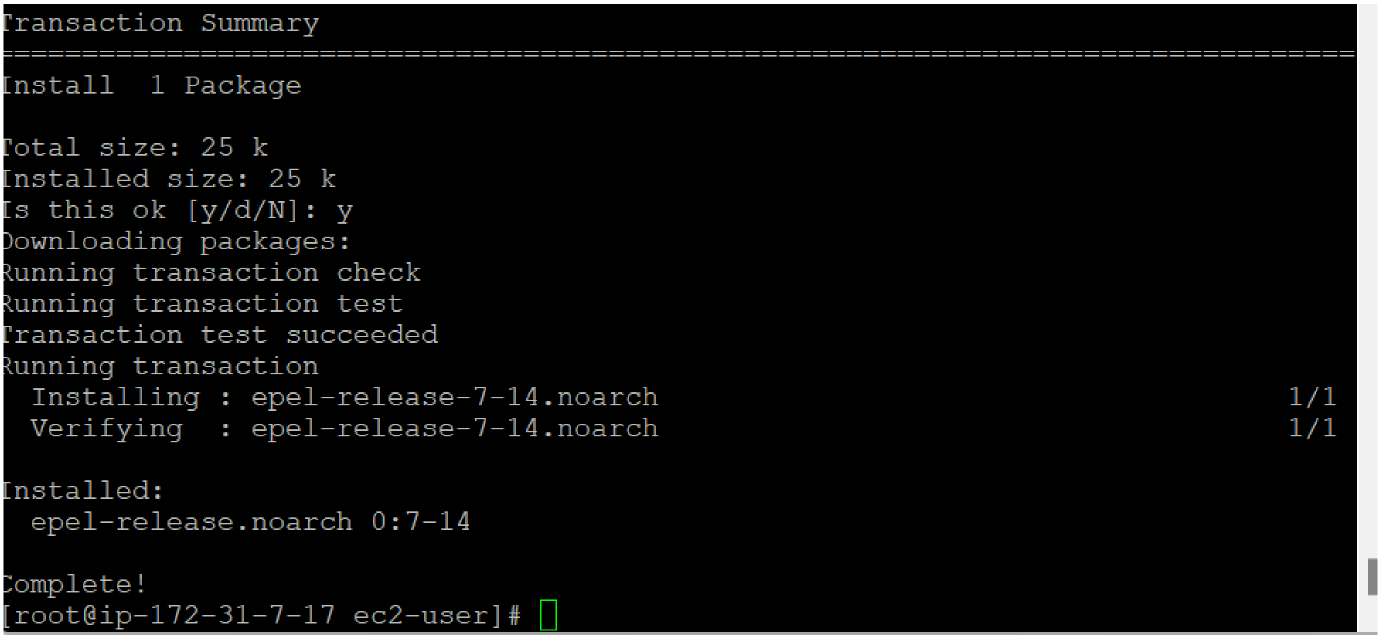
(iii) We have to download package epel-release first (do everything as a root user) in ansible server linux machine. {epel-extra package for enterprise linux}

```
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.
rpm
```



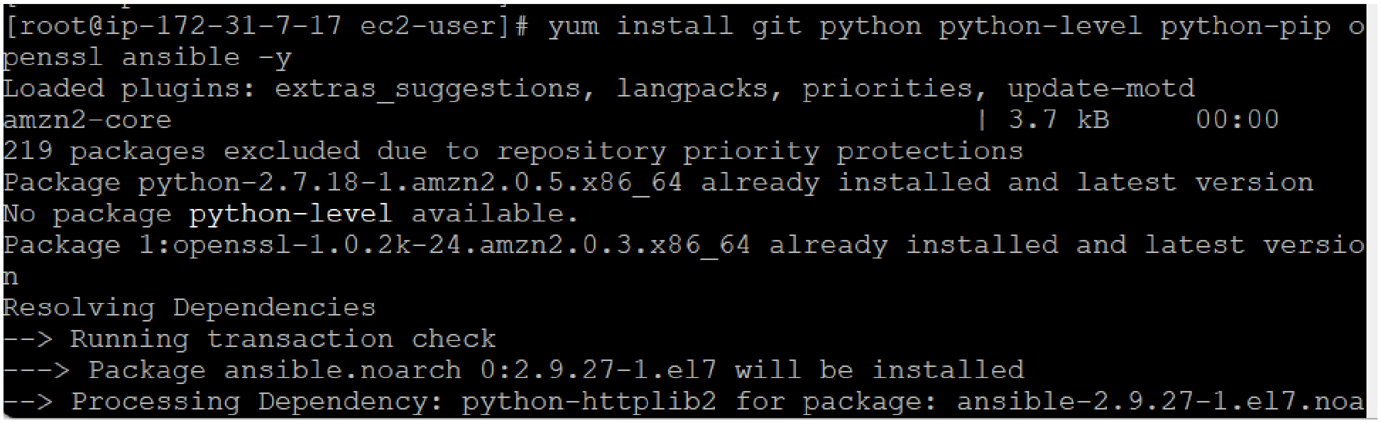
```
[root@ip-172-31-7-17 ec2-user]# wget https://dl.fedoraproject.org/pub/epel/epel-
release-latest-7.noarch.rpm
--2022-07-16 11:04:45--  https://dl.fedoraproject.org/pub/epel/epel-release-late
st-7.noarch.rpm
esolving dl.fedoraproject.org (dl.fedoraproject.org)... 38.145.60.24, 38.145.60
.22, 38.145.60.23
onnecting to dl.fedoraproject.org (dl.fedoraproject.org)|38.145.60.24|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
ength: 15608 (15K) [application/x-rpm]
aving to: 'epel-release-latest-7.noarch.rpm'

100%[=======================================>] 15,608      80.1KB/s   in 0.2s

2022-07-16 11:04:46 (80.1 KB/s) - 'epel-release-latest-7.noarch.rpm' saved [1560
8/15608]

[root@ip-172-31-7-17 ec2-user]# ls
epel-release-latest-7.noarch.rpm
```

```
yum install epel-release-latest-7.noarch.rpm
```

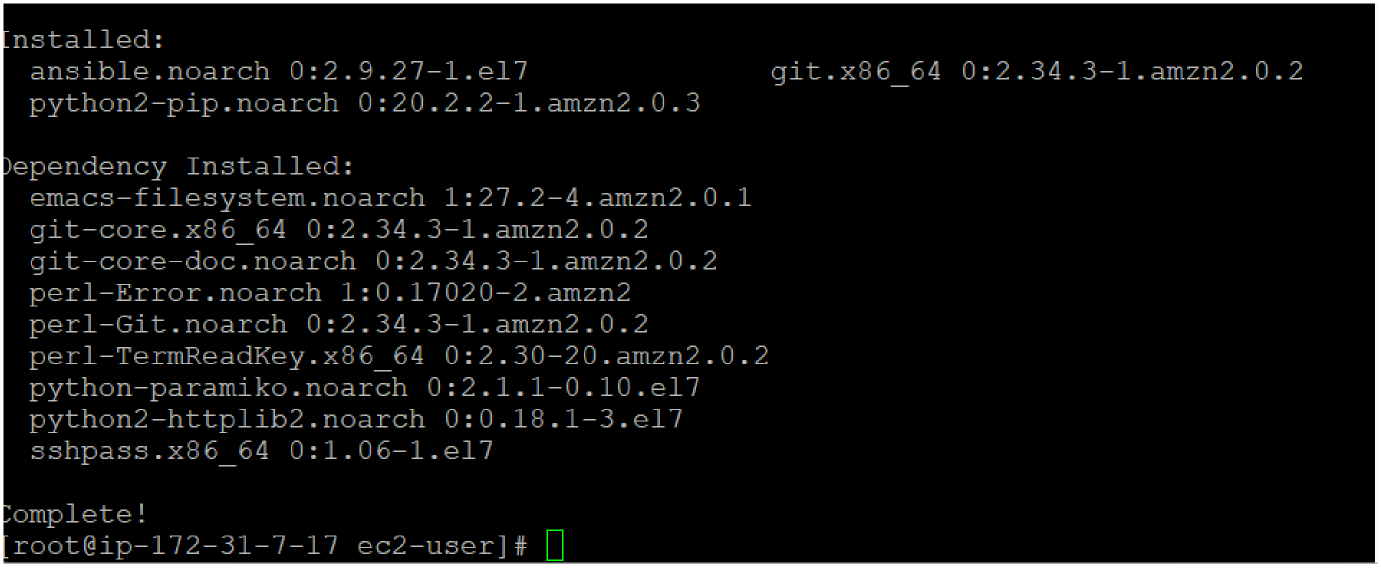


```
Transaction Summary
================================================================================
Install  1 Package

Total size: 25 k
Installed size: 25 k
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : epel-release-7-14.noarch                                     1/1
  Verifying  : epel-release-7-14.noarch                                     1/1

Installed:
  epel-release.noarch 0:7-14

Complete!
[root@ip-172-31-7-17 ec2-user]# 
```

```
yum install git python python-level python-pip openssl ansible -y
```



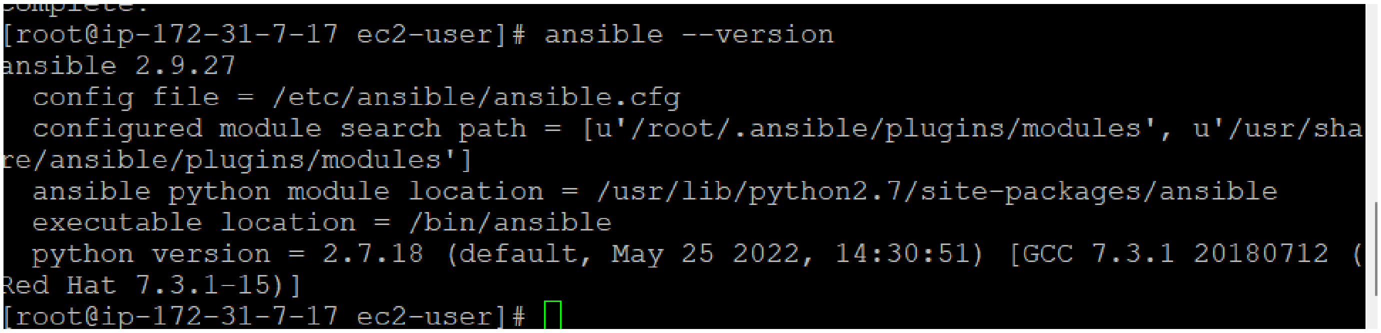
```
[root@ip-172-31-7-17 ec2-user]# yum install git python python-level python-pip o
penssl ansible -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                | 3.7 kB     00:00
219 packages excluded due to repository priority protections
Package python-2.7.18-1.amzn2.0.5.x86_64 already installed and latest version
No package python-level available.
Package 1:openssl-1.0.2k-24.amzn2.0.3.x86_64 already installed and latest versio
n
Resolving Dependencies
--> Running transaction check
---> Package ansible.noarch 0:2.9.27-1.el7 will be installed
--> Processing Dependency: python-httplib2 for package: ansible-2.9.27-1.el7.noa
```



```
Installed:
  ansible.noarch 0:2.9.27-1.el7              git.x86_64 0:2.34.3-1.amzn2.0.2
  python2-pip.noarch 0:20.2.2-1.amzn2.0.3

Dependency Installed:
  emacs-filesystem.noarch 1:27.2-4.amzn2.0.1
  git-core.x86_64 0:2.34.3-1.amzn2.0.2
  git-core-doc.noarch 0:2.34.3-1.amzn2.0.2
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-Git.noarch 0:2.34.3-1.amzn2.0.2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2
  python-paramiko.noarch 0:2.1.1-0.10.el7
  python2-httplib2.noarch 0:0.18.1-3.el7
  sshpass.x86_64 0:1.06-1.el7

Complete!
[root@ip-172-31-7-17 ec2-user]#
```

(iv) To confirm whether ansible is installed or not

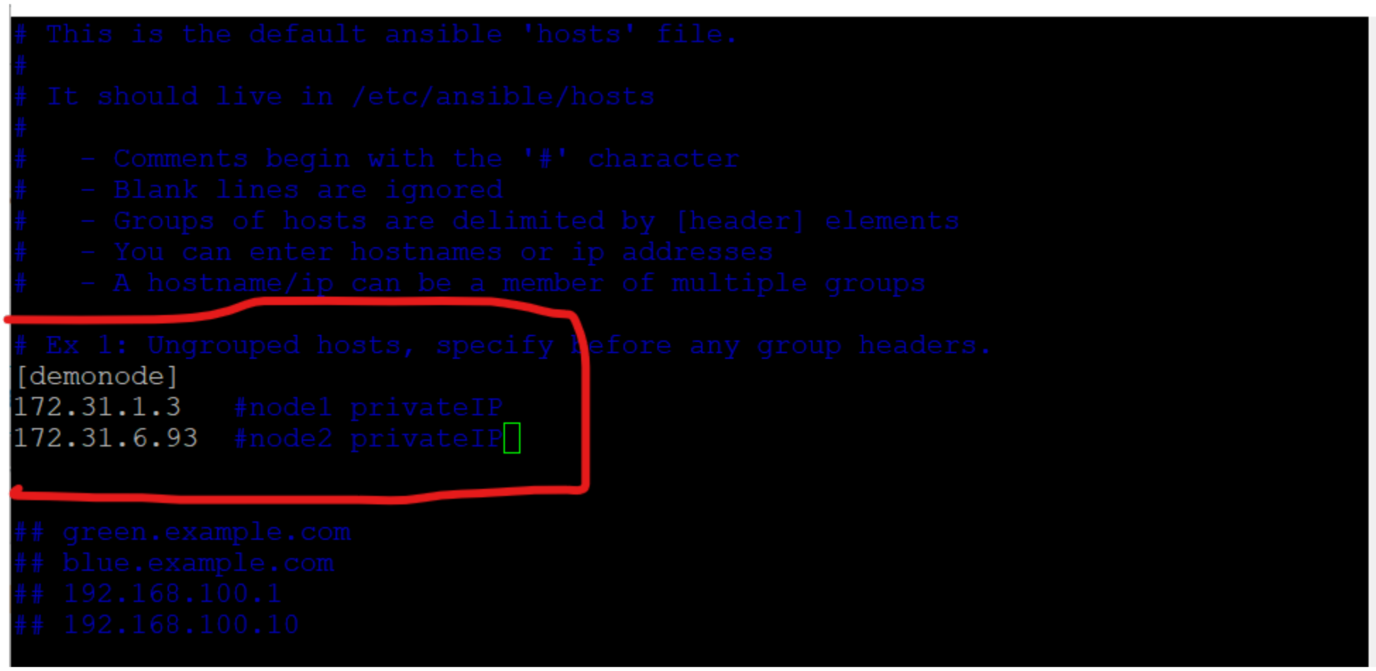
```
ansible --version
```



```
Complete!
[root@ip-172-31-7-17 ec2-user]# ansible --version
ansible 2.9.27
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/sha
re/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /bin/ansible
  python version = 2.7.18 (default, May 25 2022, 14:30:51) [GCC 7.3.1 20180712 (
Red Hat 7.3.1-15)]
[root@ip-172-31-7-17 ec2-user]#
```

(v) Update Private Ip of Node1 and Node2 in hosts file of ansible server , so that it will know that which node it have to configure.

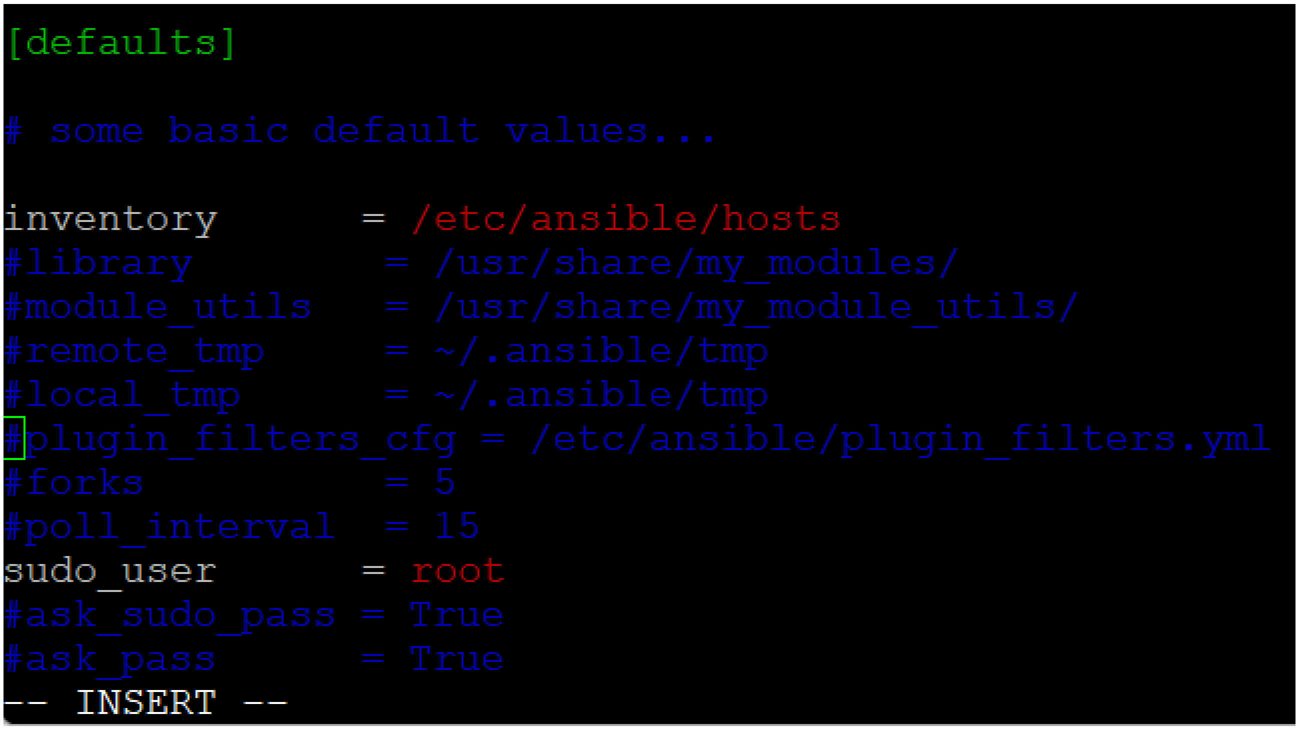
```
vi /etc/ansible/hosts
#create a group with name demonode
#add node1 privateIP
#add node2 privateIP
:wq (to save vi file)
```



```
# 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.
[demonode]
172.31.1.3    #node1 privateIP
172.31.6.93   #node2 privateIP

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
```

(VI) Now we have to update the configuration file of ansible

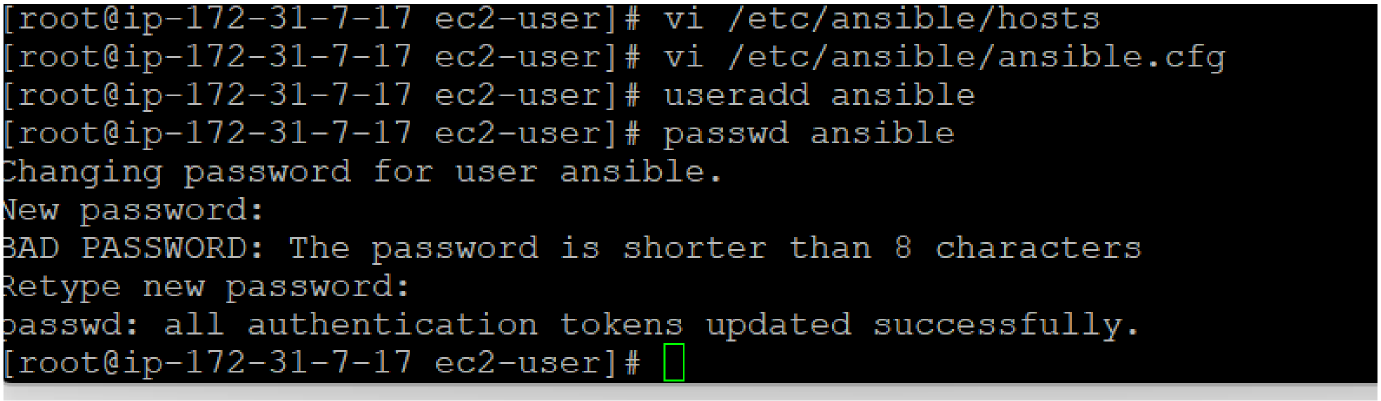
```
vi /etc/ansible/ansible.cfg
#uncomment inventory and sudo_user
```



```
[defaults]

# some basic default values...

inventory       = /etc/ansible/hosts
#library         = /usr/share/my_modules/
#module_utils    = /usr/share/my_module_utils/
#remote_tmp      = ~/.ansible/tmp
#local_tmp       = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks           = 5
#poll_interval   = 15
sudo_user       = root
#ask_sudo_pass = True
#ask_pass       = True
-- INSERT --
```

(vii) Now we will create a user with name “ansible” because we shouldn’t use root user in a production environment.

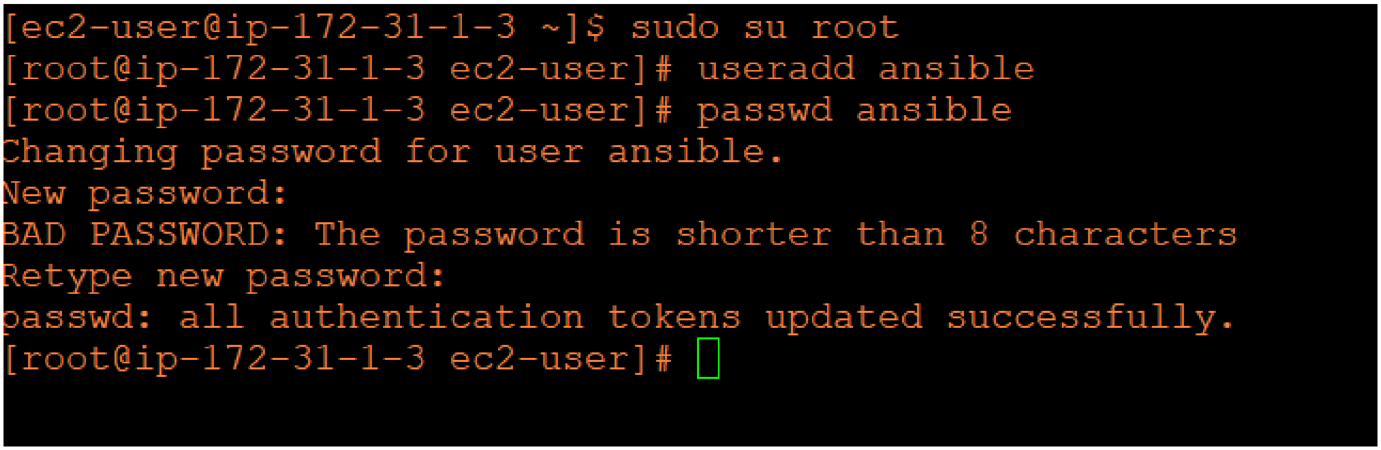
```
useradd ansible
passwd ansible
#new password
#retype new password
```



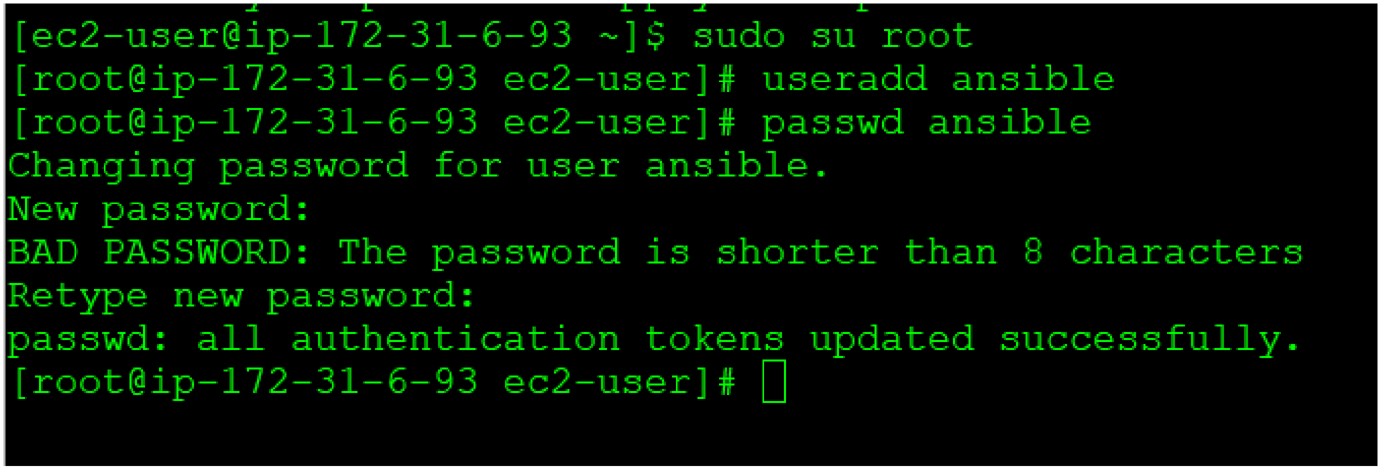
```
[root@ip-172-31-7-17 ec2-user]# vi /etc/ansible/hosts
[root@ip-172-31-7-17 ec2-user]# vi /etc/ansible/ansible.cfg
[root@ip-172-31-7-17 ec2-user]# useradd ansible
[root@ip-172-31-7-17 ec2-user]# passwd ansible
Changing password for user ansible.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-7-17 ec2-user]#
```

(viii) Now similary create a user in node1 and node2 also. For simplicity we will create user with same name “ansible” only in node1 and node2 also

```
useradd ansible
passwd ansible
#new password
#retype new password
```



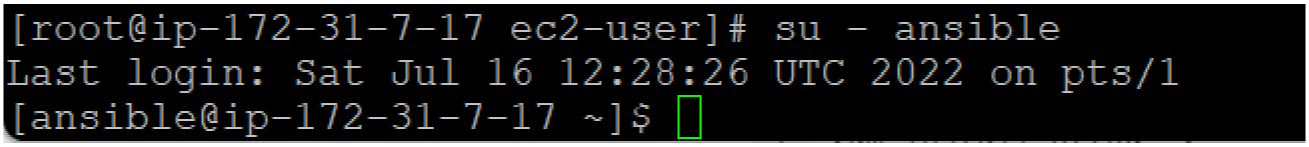
```
[ec2-user@ip-172-31-1-3 ~]$ sudo su root
[root@ip-172-31-1-3 ec2-user]# useradd ansible
[root@ip-172-31-1-3 ec2-user]# passwd ansible
Changing password for user ansible.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-1-3 ec2-user]#
```



```
[ec2-user@ip-172-31-6-93 ~]$ sudo su root
[root@ip-172-31-6-93 ec2-user]# useradd ansible
[root@ip-172-31-6-93 ec2-user]# 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.
[root@ip-172-31-6-93 ec2-user]#
```

(ix) Now switch to ansible user(newly created) in ansible server ec2-instance.

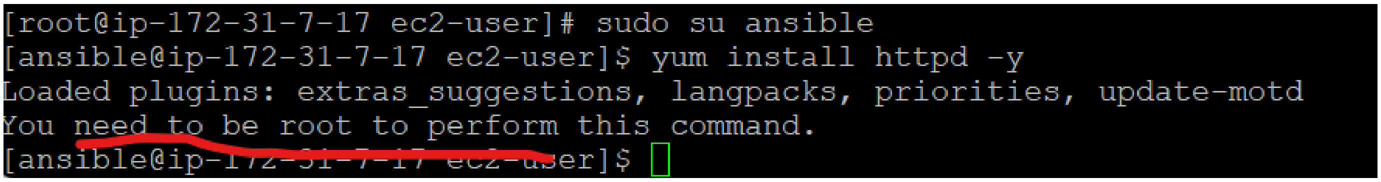
```
su - ansible
```



```
[root@ip-172-31-7-17 ec2-user]# su - ansible
Last login: Sat Jul 16 12:28:26 UTC 2022 on pts/1
[ansible@ip-172-31-7-17 ~]$
```

Try to run a command as a ansible user

```
yum install httpd -y
```

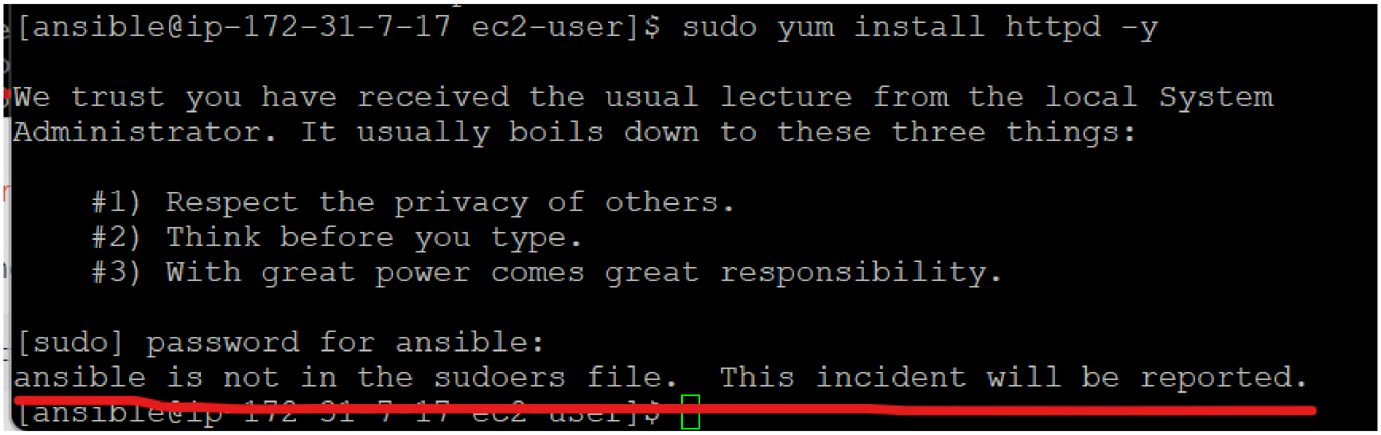


```
[root@ip-172-31-7-17 ec2-user]# sudo su ansible
[ansible@ip-172-31-7-17 ec2-user]$ yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ansible@ip-172-31-7-17 ec2-user]$
```

Need to be root to perform this command.

Now again try to run same command with sudo power.

```
sudo yum install httpd -y
```



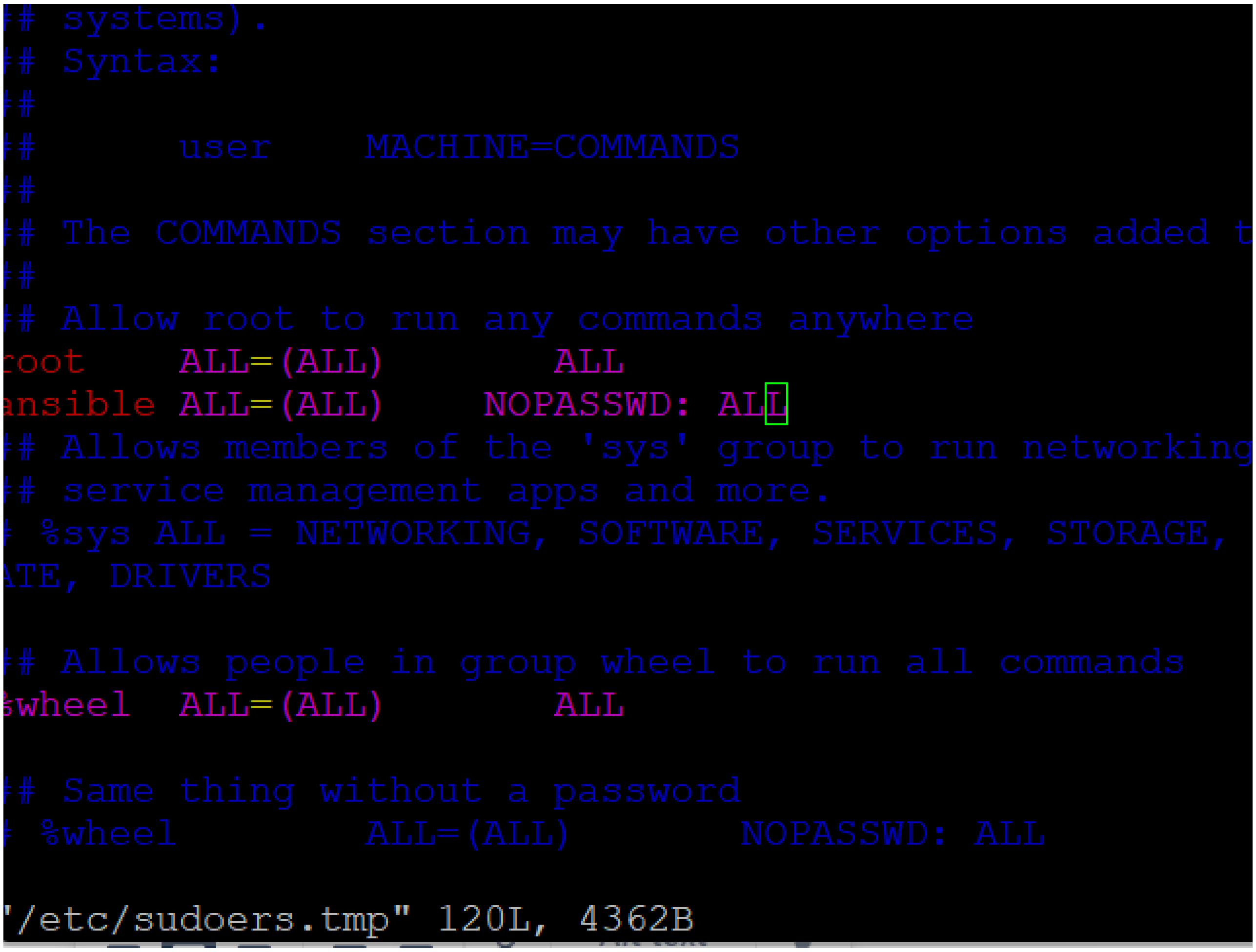
```
[ansible@ip-172-31-7-17 ec2-user]$ sudo yum install httpd -y

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for ansible:
ansible is not in the sudoers file.  This incident will be reported.
[ansible@ip-172-31-7-17 ec2-user]$
```

Ansible is not in the sudoers file.

We have to provide sudo power to our ansible user.

(x) Add ansible user in sudoers file

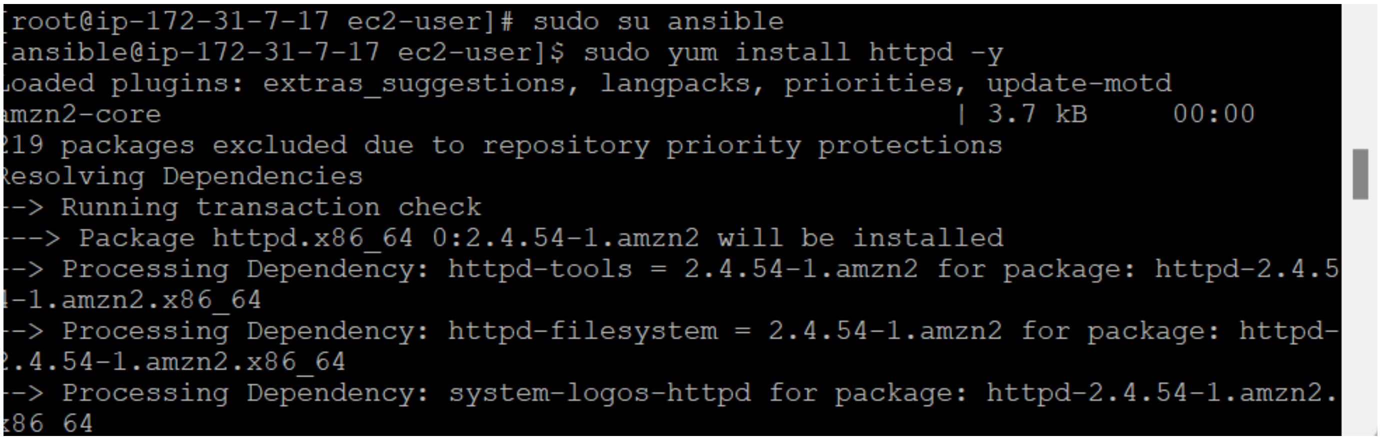
```
exit #switch again to root user from ansible user
visudo

#add this line after root
ansible ALL=(ALL)    NOPASSWD: ALL
```



```
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added t
##
## 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
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE,
ATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

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

"/etc/sudoers.tmp" 120L, 4362B
```

Now again switch to ansible user and again try to run command this time package will be installed.

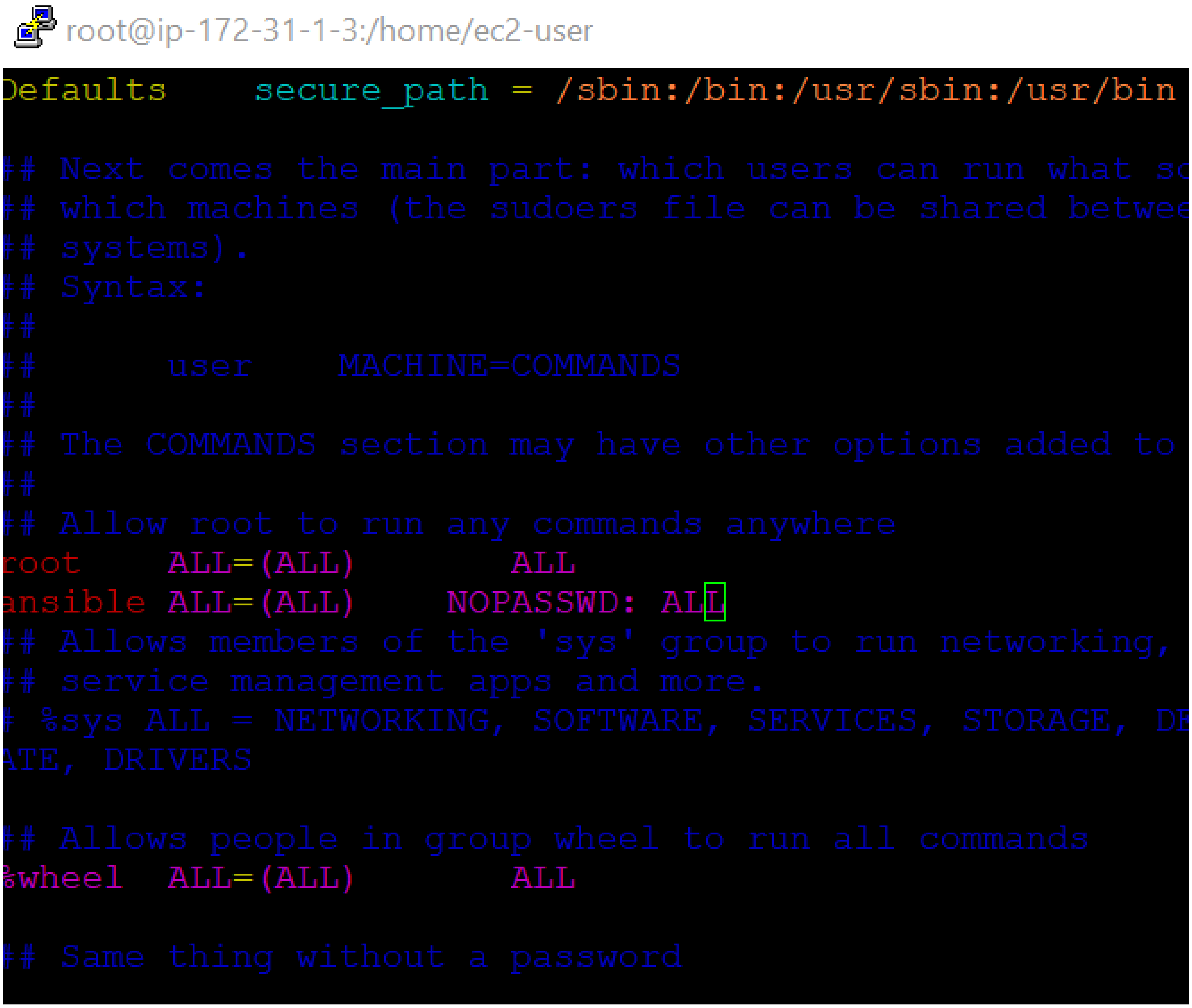
```
su - ansible
sudo yum install httpd -y
```



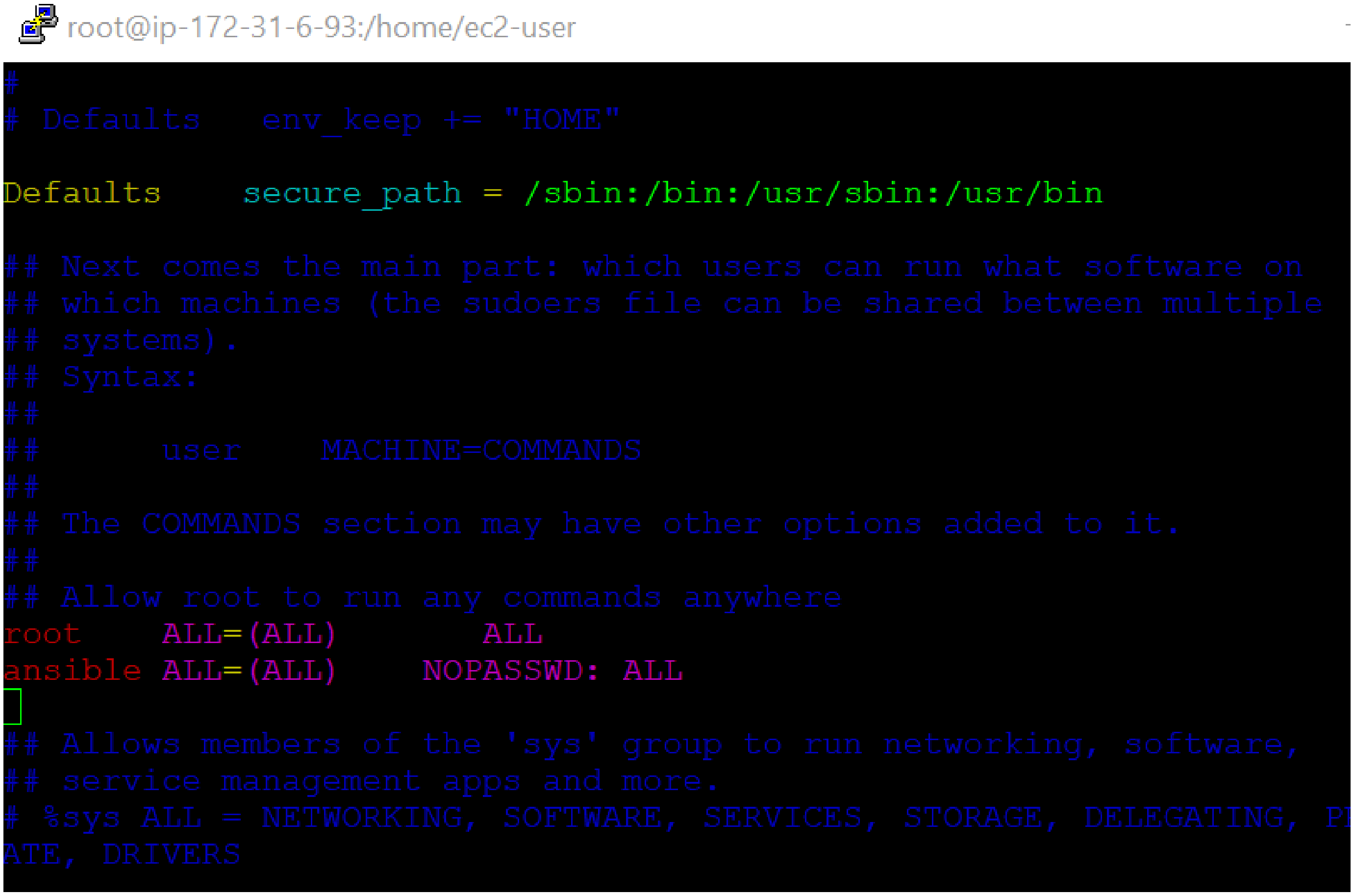
```
[root@ip-172-31-7-17 ec2-user]# sudo su ansible
[ansible@ip-172-31-7-17 ec2-user]$ sudo yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 3.7 kB     00:00
219 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.5
4-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-
2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.
x86_64
```

Now add ansible user in sudoer file in node1 and node2 also.

```
visudo

#add this line after root
ansible ALL=(ALL)    NOPASSWD: ALL
```



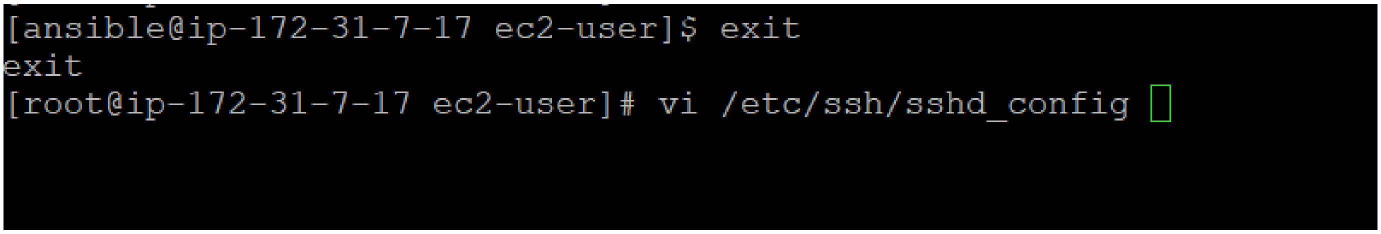
```
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what so
## which machines (the sudoers file can be shared betwee
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to
##
## 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,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DE
ATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
```



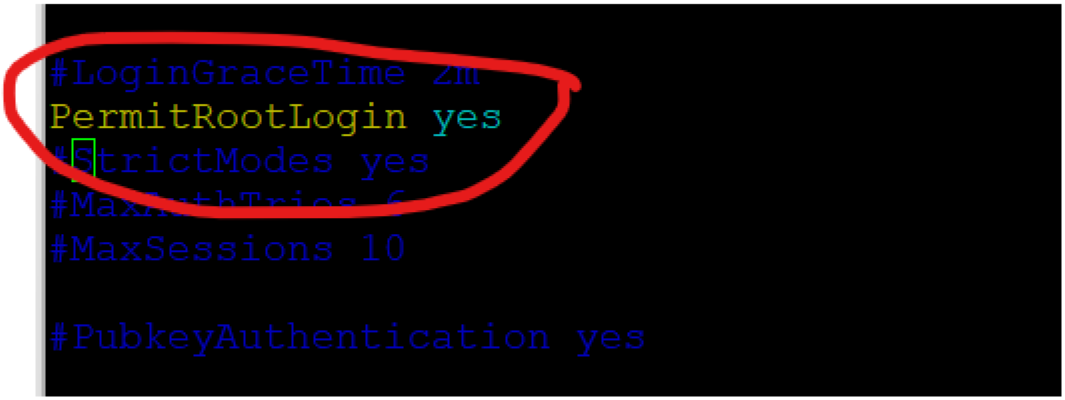
```
root@ip-172-31-6-93:/home/ec2-user
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## 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, P
ATE, DRIVERS
```

(xi) Edit ssh config file in ansible server ec2-instance and other two nodes also.

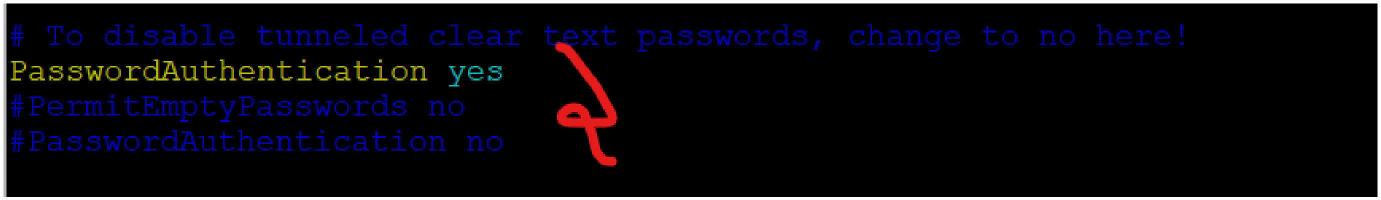
```
#login as root user
vi /etc/ssh/sshd_config
# uncomment permitrootlogin yes
#uncomment passwordauthentication yes
#comment passwordauthentication no

systemctl restart sshd
```



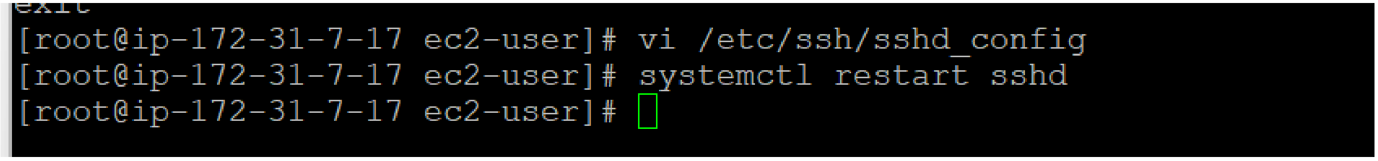
```
[ansible@ip-172-31-7-17 ec2-user]$ exit
exit
[root@ip-172-31-7-17 ec2-user]# vi /etc/ssh/sshd_config
```



```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

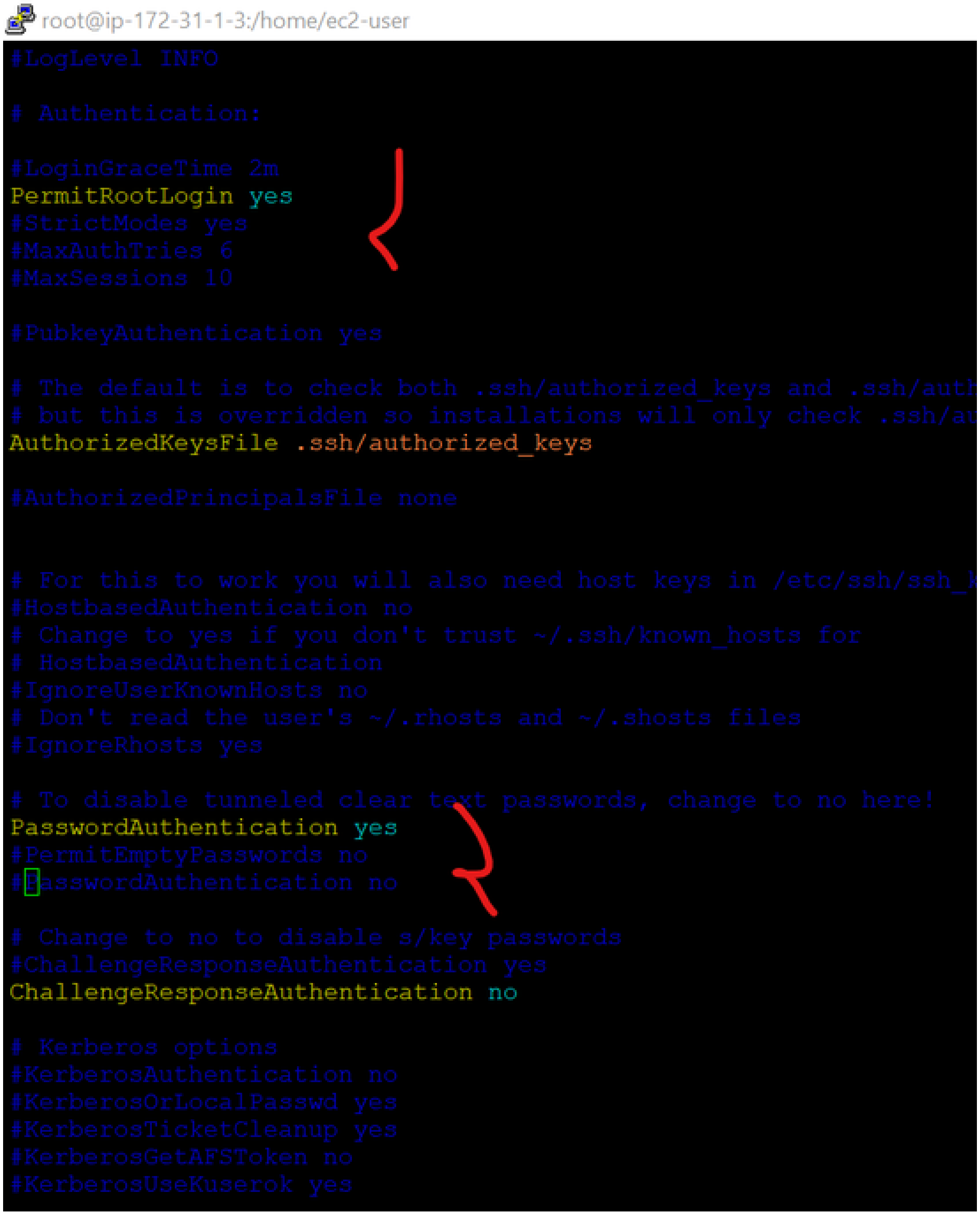


```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
#PasswordAuthentication no
```



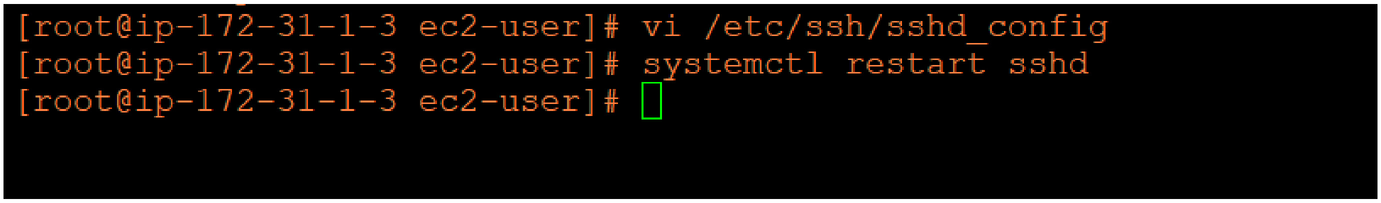
```
exit
[root@ip-172-31-7-17 ec2-user]# vi /etc/ssh/sshd_config
[root@ip-172-31-7-17 ec2-user]# systemctl restart sshd
[root@ip-172-31-7-17 ec2-user]# 
```

Now do same thing in other node also

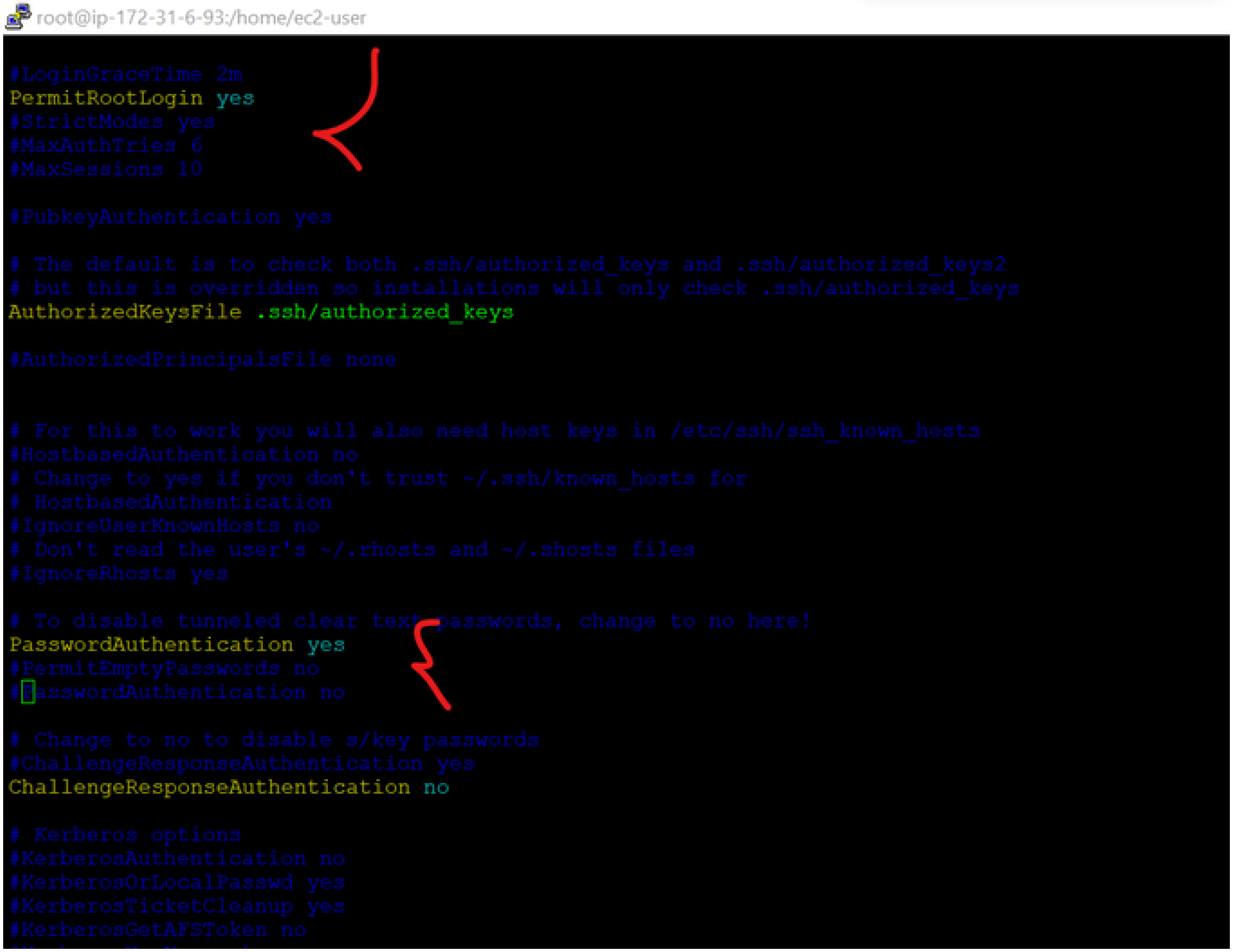


root@ip-172-31-1-3:/home/ec2-user

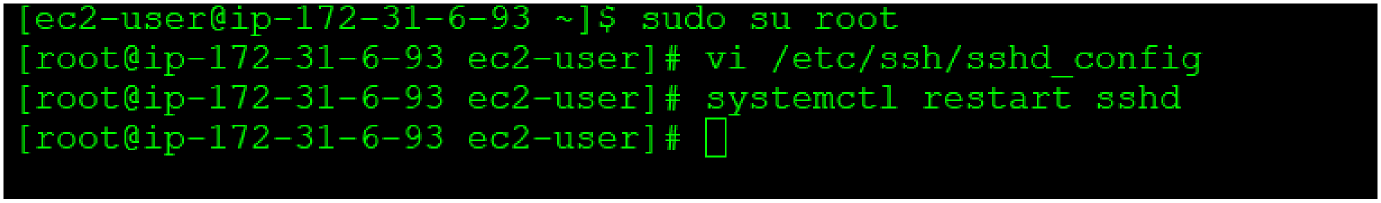
```
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/auth
# but this is overridden so installations will only check .ssh/au
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none


# For this to work you will also need host keys in /etc/ssh/ssh_k
#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
#PasswordAuthentication no

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

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



```
[root@ip-172-31-1-3 ec2-user]# vi /etc/ssh/sshd_config
[root@ip-172-31-1-3 ec2-user]# systemctl restart sshd
[root@ip-172-31-1-3 ec2-user]# 
```



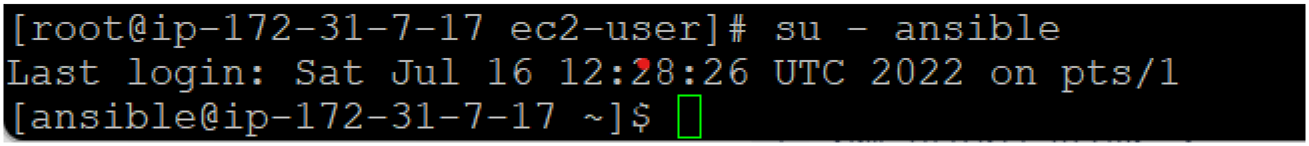
```
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile .ssh/authorized_keys

#AuthorizedPrincipalsFile none


# 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
#PasswordAuthentication no

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

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



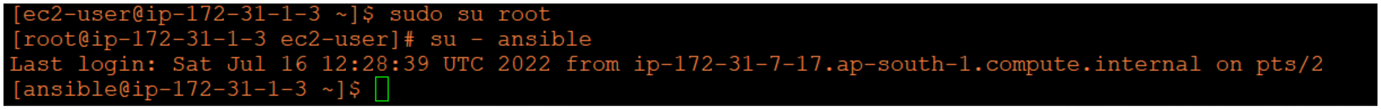
```
[ec2-user@ip-172-31-6-93 ~]$ sudo su root
[root@ip-172-31-6-93 ec2-user]# vi /etc/ssh/sshd_config
[root@ip-172-31-6-93 ec2-user]# systemctl restart sshd
[root@ip-172-31-6-93 ec2-user]#
```

(xii) Now login as ansible user in all the three ec2-instance.

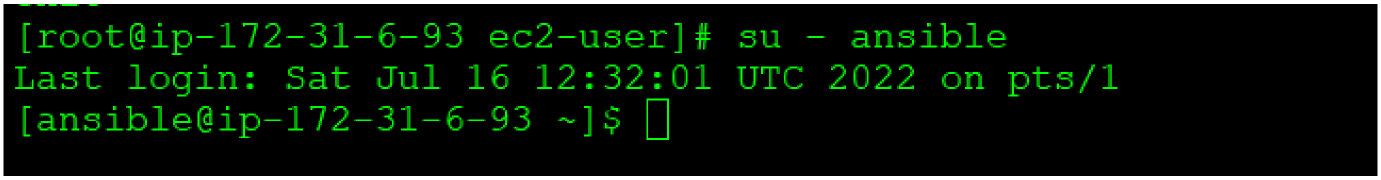
```
su - ansible
```



```
[root@ip-172-31-7-17 ec2-user]# su - ansible
Last login: Sat Jul 16 12:28:26 UTC 2022 on pts/1
[ansible@ip-172-31-7-17 ~]$
```

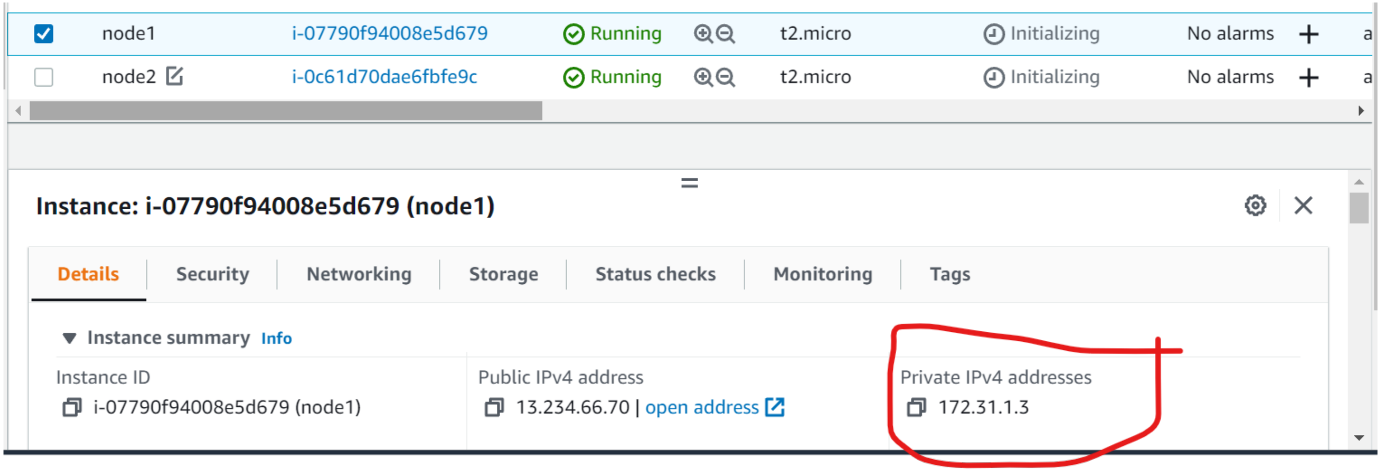


```
[ec2-user@ip-172-31-1-3 ~]$ sudo su root
[root@ip-172-31-1-3 ec2-user]# su - ansible
Last login: Sat Jul 16 12:28:39 UTC 2022 from ip-172-31-7-17.ap-south-1.compute.internal on pts/2
[ansible@ip-172-31-1-3 ~]$
```



```
[root@ip-172-31-6-93 ec2-user]# su - ansible
Last login: Sat Jul 16 12:32:01 UTC 2022 on pts/1
[ansible@ip-172-31-6-93 ~]$ 
```

(xii) Now try to do ssh to node1 and node2 from ansible server ec2-instance.



☑	node1	i-07790f94008e5d679	✓ Running	t2.micro	Initializing	No alarms	+	a
□	node2	i-0c61d70dae6fbfe9c	✓ Running	t2.micro	Initializing	No alarms	+	a

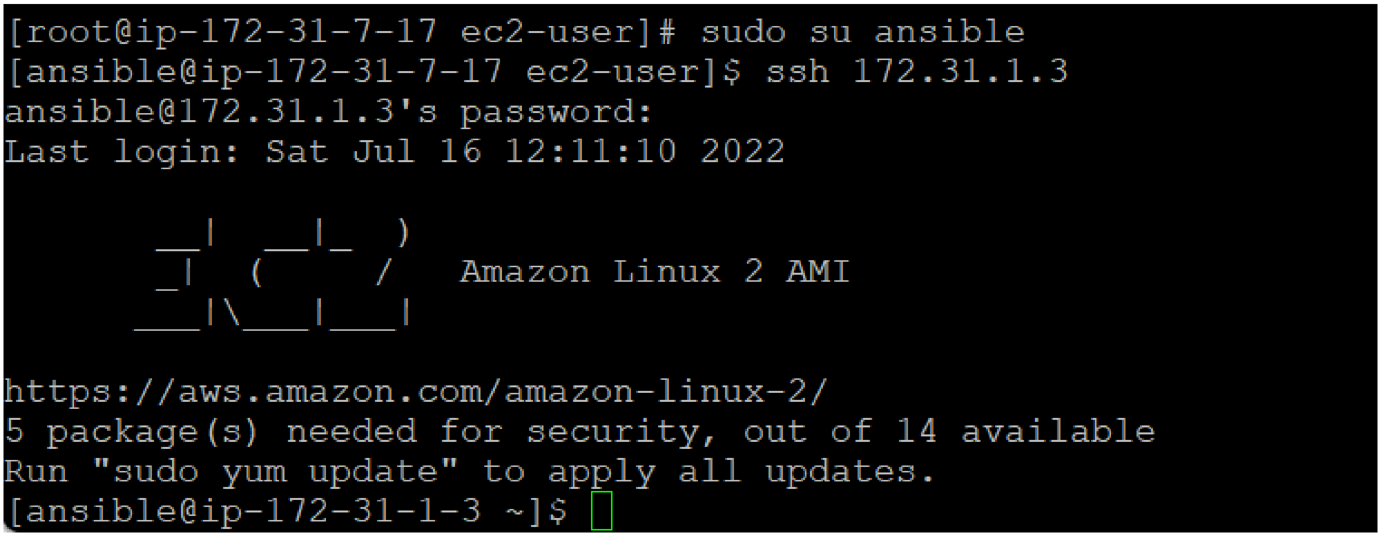
Instance: i-07790f94008e5d679 (node1)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

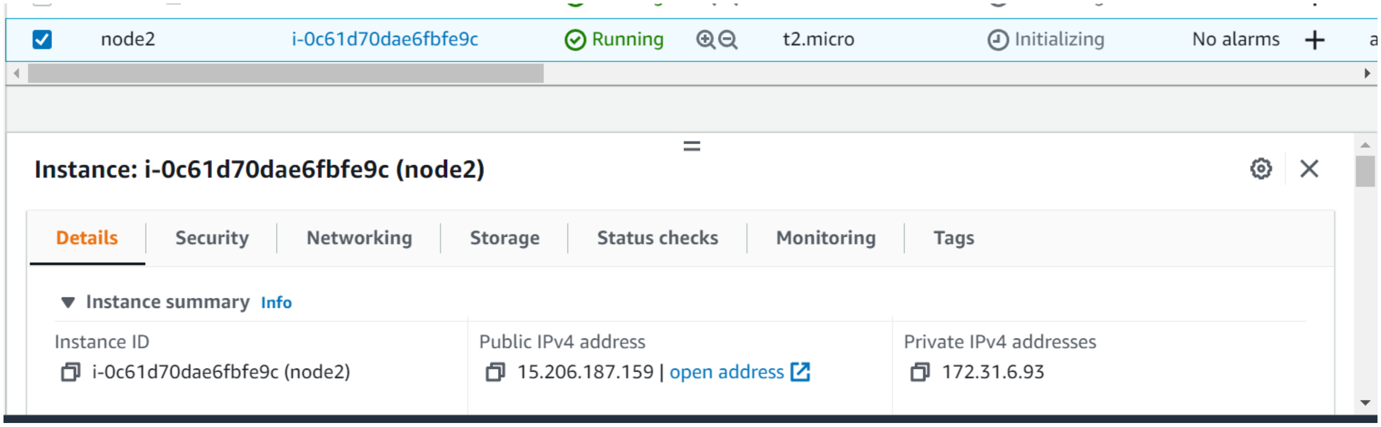
▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-07790f94008e5d679 (node1)	13.234.66.70 | open address	172.31.1.3

```
ssh 172.31.1.3 #Node1 IP
```



```
[root@ip-172-31-7-17 ec2-user]# sudo su ansible
[ansible@ip-172-31-7-17 ec2-user]$ ssh 172.31.1.3
ansible@172.31.1.3's password:
Last login: Sat Jul 16 12:11:10 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ansible@ip-172-31-1-3 ~]$ 
```



☑	node2	i-0c61d70dae6fbfe9c	✓ Running	t2.micro	Initializing	No alarms	+	a

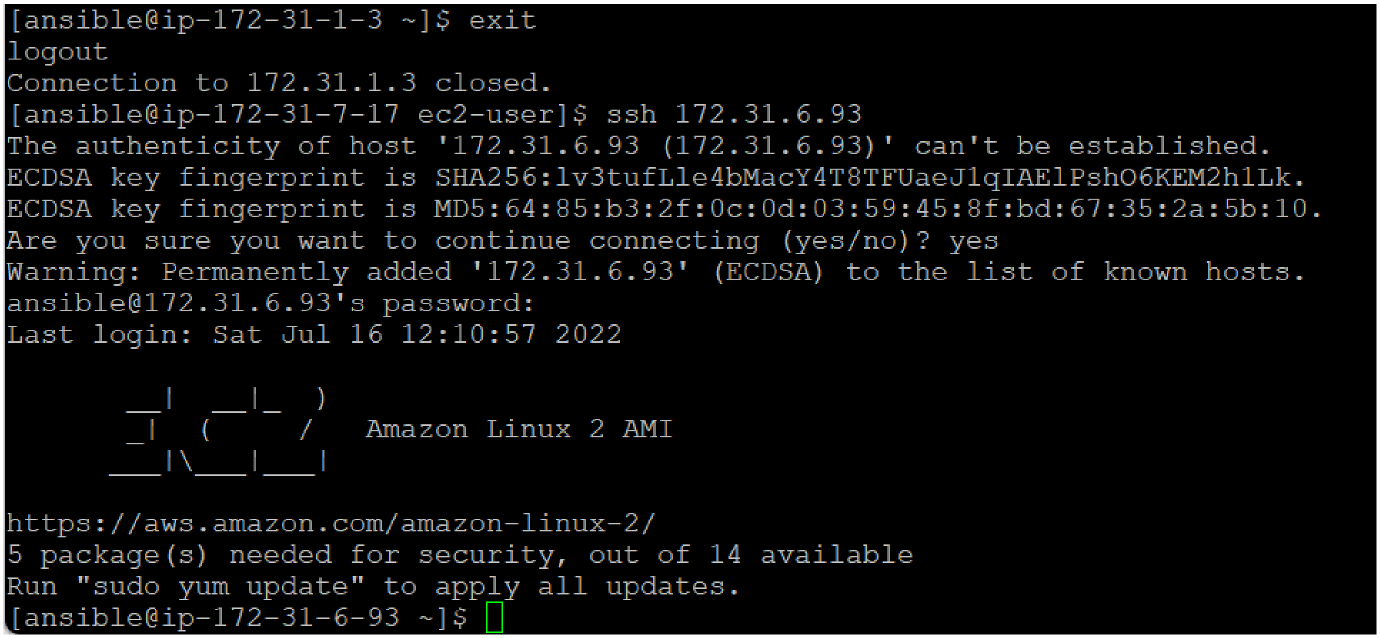
Instance: i-0c61d70dae6fbfe9c (node2)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

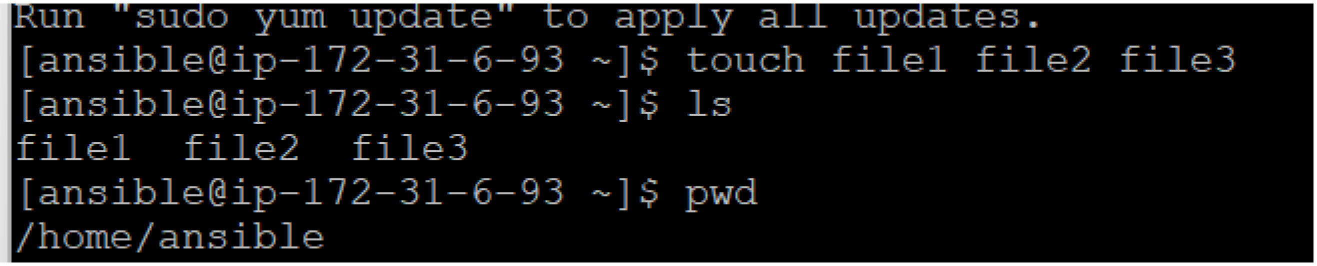
▼ Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0c61d70dae6fbfe9c (node2)	15.206.187.159 | open address	172.31.6.93

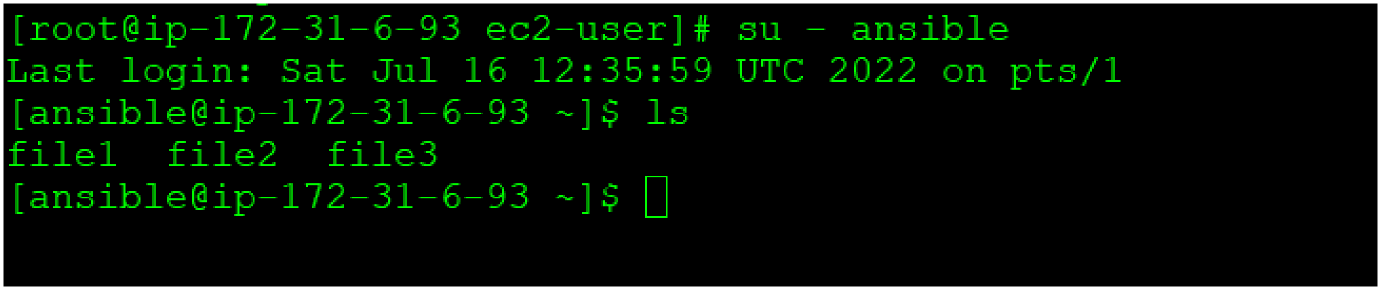
```
ssh 172.31.6.93
```



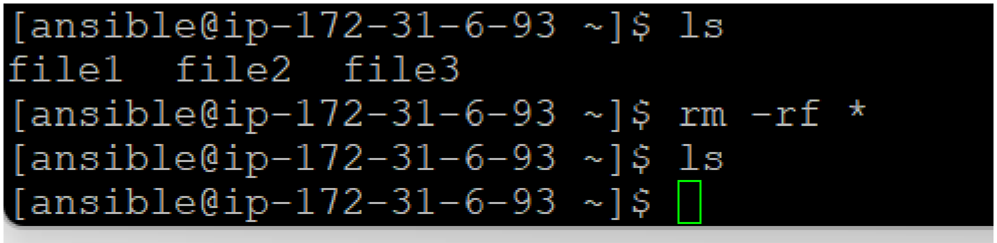
```
[ansible@ip-172-31-1-3 ~]$ exit
logout
Connection to 172.31.1.3 closed.
[ansible@ip-172-31-7-17 ec2-user]$ ssh 172.31.6.93
The authenticity of host '172.31.6.93 (172.31.6.93)' can't be established.
ECDSA key fingerprint is SHA256:lv3tufLle4bMacY4T8TFUaeJ1qIAElPshO6KEM2h1Lk.
ECDSA key fingerprint is MD5:64:85:b3:2f:0c:0d:03:59:45:8f:bd:67:35:2a:5b:10.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.31.6.93' (ECDSA) to the list of known hosts.
ansible@172.31.6.93's password:
Last login: Sat Jul 16 12:10:57 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ansible@ip-172-31-6-93 ~]$
```



```
Run "sudo yum update" to apply all updates.
[ansible@ip-172-31-6-93 ~]$ touch file1 file2 file3
[ansible@ip-172-31-6-93 ~]$ ls
file1  file2  file3
[ansible@ip-172-31-6-93 ~]$ pwd
/home/ansible
```



```
[root@ip-172-31-6-93 ec2-user]# su - ansible
Last login: Sat Jul 16 12:35:59 UTC 2022 on pts/1
[ansible@ip-172-31-6-93 ~]$ ls
file1  file2  file3
[ansible@ip-172-31-6-93 ~]$
```



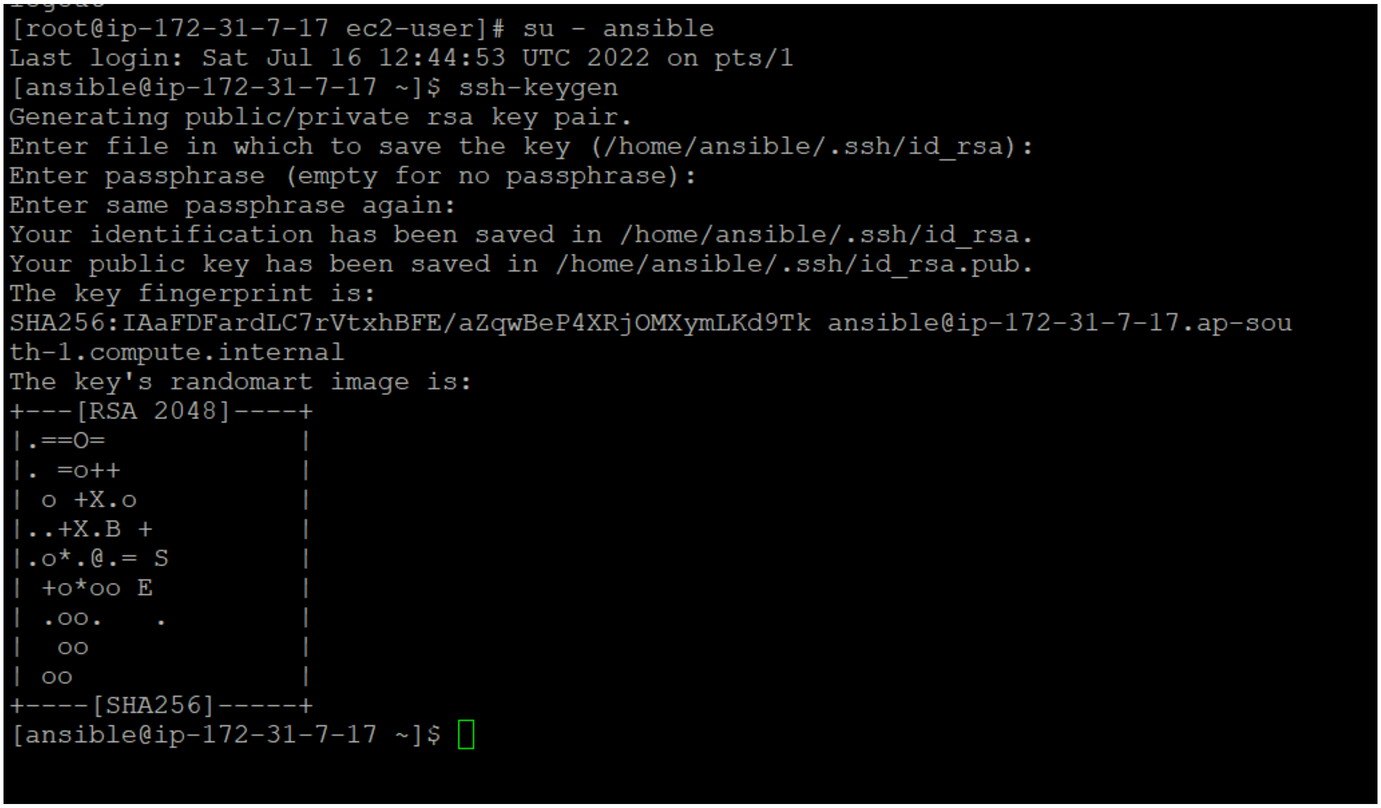
```
[ansible@ip-172-31-6-93 ~]$ ls
file1  file2  file3
[ansible@ip-172-31-6-93 ~]$ rm -rf *
[ansible@ip-172-31-6-93 ~]$ ls
[ansible@ip-172-31-6-93 ~]$
```

(xIII) Now when we are doing ssh connection to any of the node it always ask for password, so if we have 100 node then we have to give password 100 times each for one node.

So we have to do some changes so when we do ssh it should not ask password.

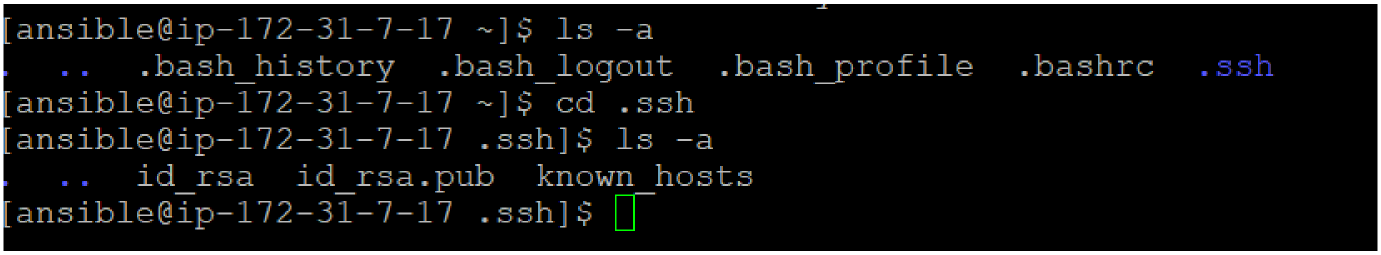
First login to ansible server ec2-instance as ansible user.

```
su - ansible
ssh-keygen
#press enter three times
```



```
[root@ip-172-31-7-17 ec2-user]# su - ansible
Last login: Sat Jul 16 12:44:53 UTC 2022 on pts/1
[ansible@ip-172-31-7-17 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
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:IAaFDFardLC7rVtxhBFE/aZqwBeP4XRjOMXymLKd9Tk ansible@ip-172-31-7-17.ap-sou
th-1.compute.internal
The key's randomart image is:
+---[RSA 2048]----+
|.==O=            |
|. =o++           |
| o +X.o          |
|..+X.B +         |
|.o*.@.= S        |
| +o*oo E         |
| .oo.   .        |
|  oo             |
| oo              |
+----[SHA256]-----+
[ansible@ip-172-31-7-17 ~]$
```

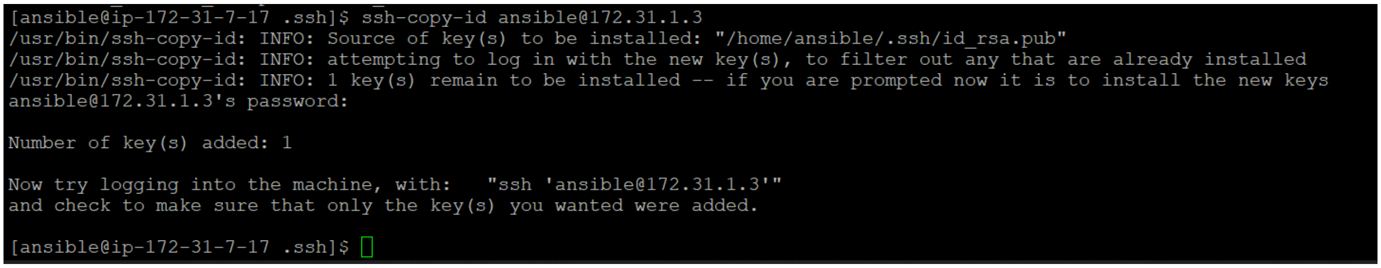
```
ls -a
cd .ssh
ls -a
```



```
[ansible@ip-172-31-7-17 ~]$ ls -a
.  ..  .bash_history  .bash_logout  .bash_profile  .bashrc  .ssh
[ansible@ip-172-31-7-17 ~]$ cd .ssh
[ansible@ip-172-31-7-17 .ssh]$ ls -a
.  ..  id_rsa  id_rsa.pub  known_hosts
[ansible@ip-172-31-7-17 .ssh]$
```

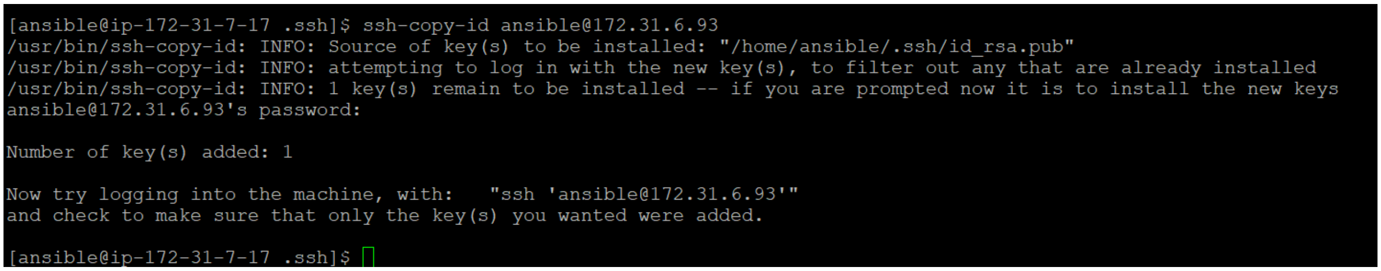
Now we have to copy these files to both node1 and node2.

```
ssh-copy-id ansible@172.31.1.3
```



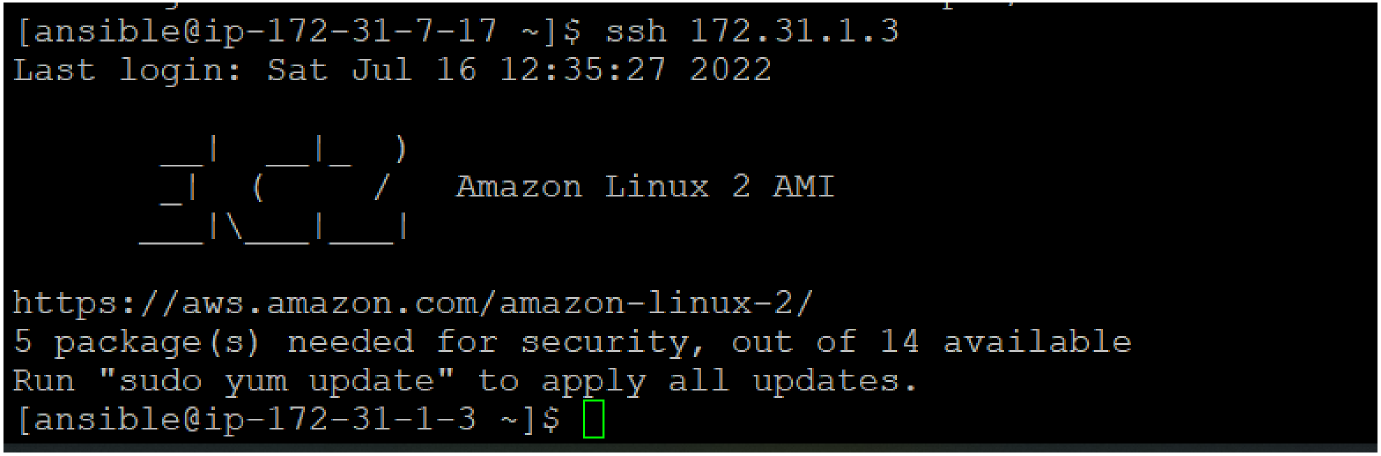
```
[ansible@ip-172-31-7-17 .ssh]$ ssh-copy-id ansible@172.31.1.3
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@172.31.1.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@172.31.1.3'"
and check to make sure that only the key(s) you wanted were added.

[ansible@ip-172-31-7-17 .ssh]$
```

```
ssh-copy-id ansible@172.31.6.93
```

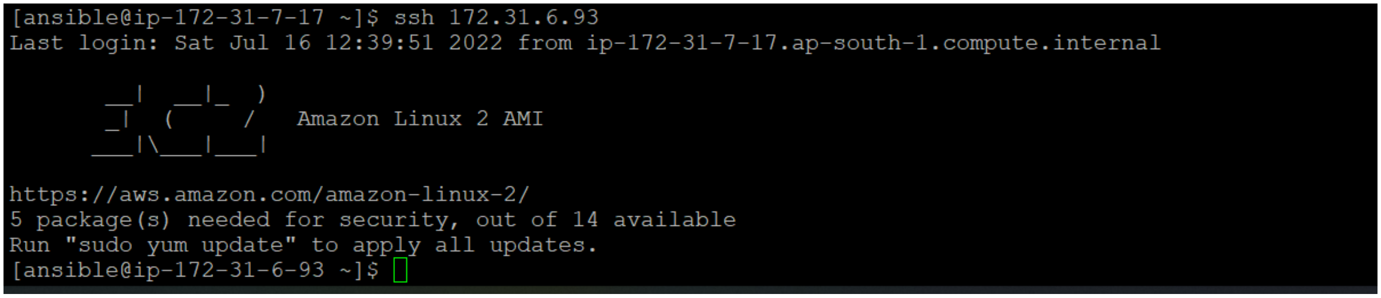


```
[ansible@ip-172-31-7-17 .ssh]$ ssh-copy-id ansible@172.31.6.93
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@172.31.6.93's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ansible@172.31.6.93'"
and check to make sure that only the key(s) you wanted were added.

[ansible@ip-172-31-7-17 .ssh]$
```

Now try to do ssh to node1 or node2, you will see that now it will not ask for passwords.



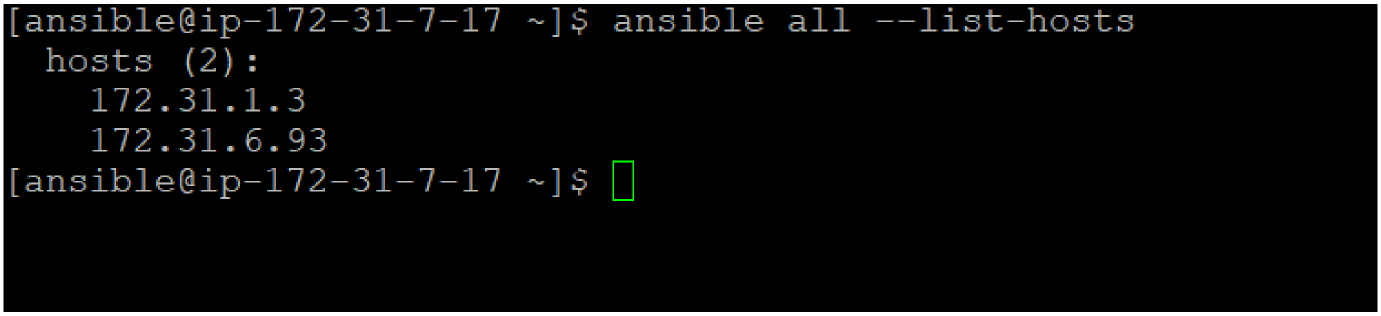
```
[ansible@ip-172-31-7-17 ~]$ ssh 172.31.1.3
Last login: Sat Jul 16 12:35:27 2022

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ansible@ip-172-31-1-3 ~]$
```



```
[ansible@ip-172-31-7-17 ~]$ ssh 172.31.6.93
Last login: Sat Jul 16 12:39:51 2022 from ip-172-31-7-17.ap-south-1.compute.internal

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ansible@ip-172-31-6-93 ~]$
```

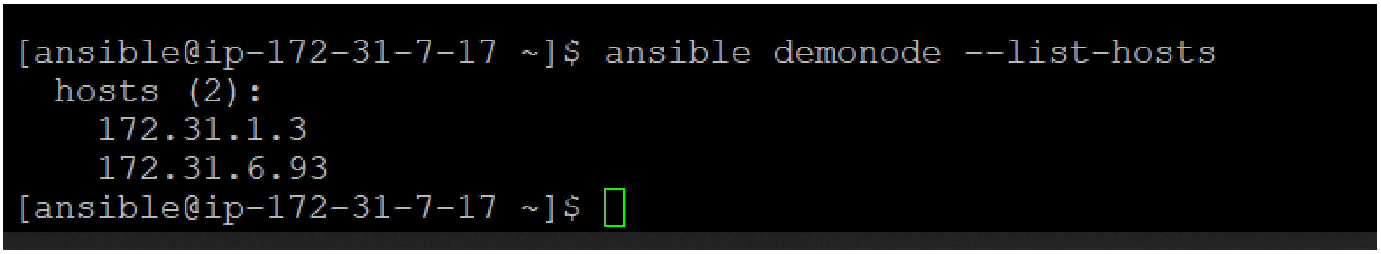
(XIV) Now go to ansible server

```
ansible all --list-hosts
```



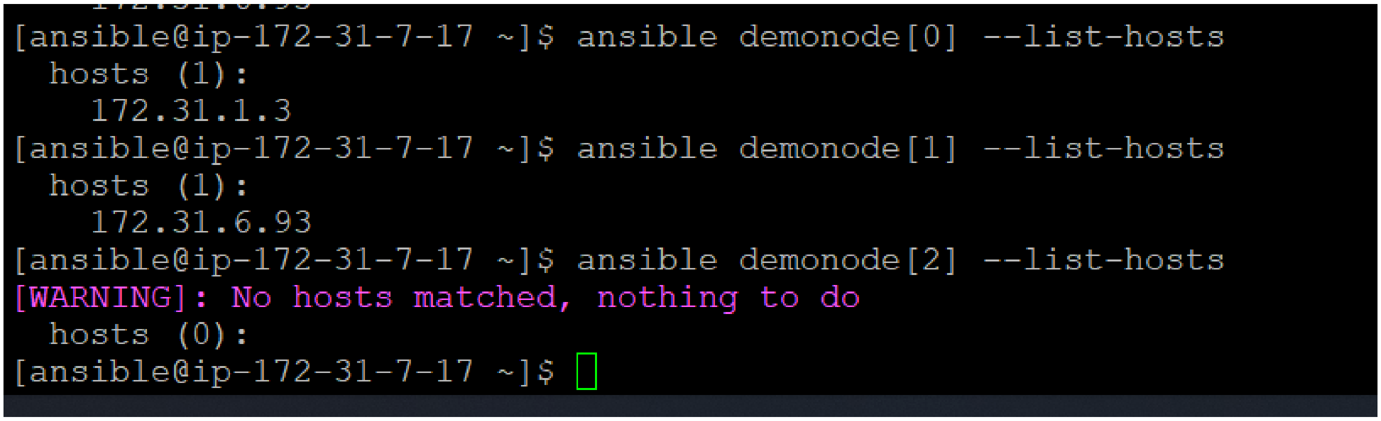
```
[ansible@ip-172-31-7-17 ~]$ ansible all --list-hosts
  hosts (2):
    172.31.1.3
    172.31.6.93
[ansible@ip-172-31-7-17 ~]$
```

```
ansible demonode --list-hosts
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode --list-hosts
  hosts (2):
    172.31.1.3
    172.31.6.93
[ansible@ip-172-31-7-17 ~]$
```

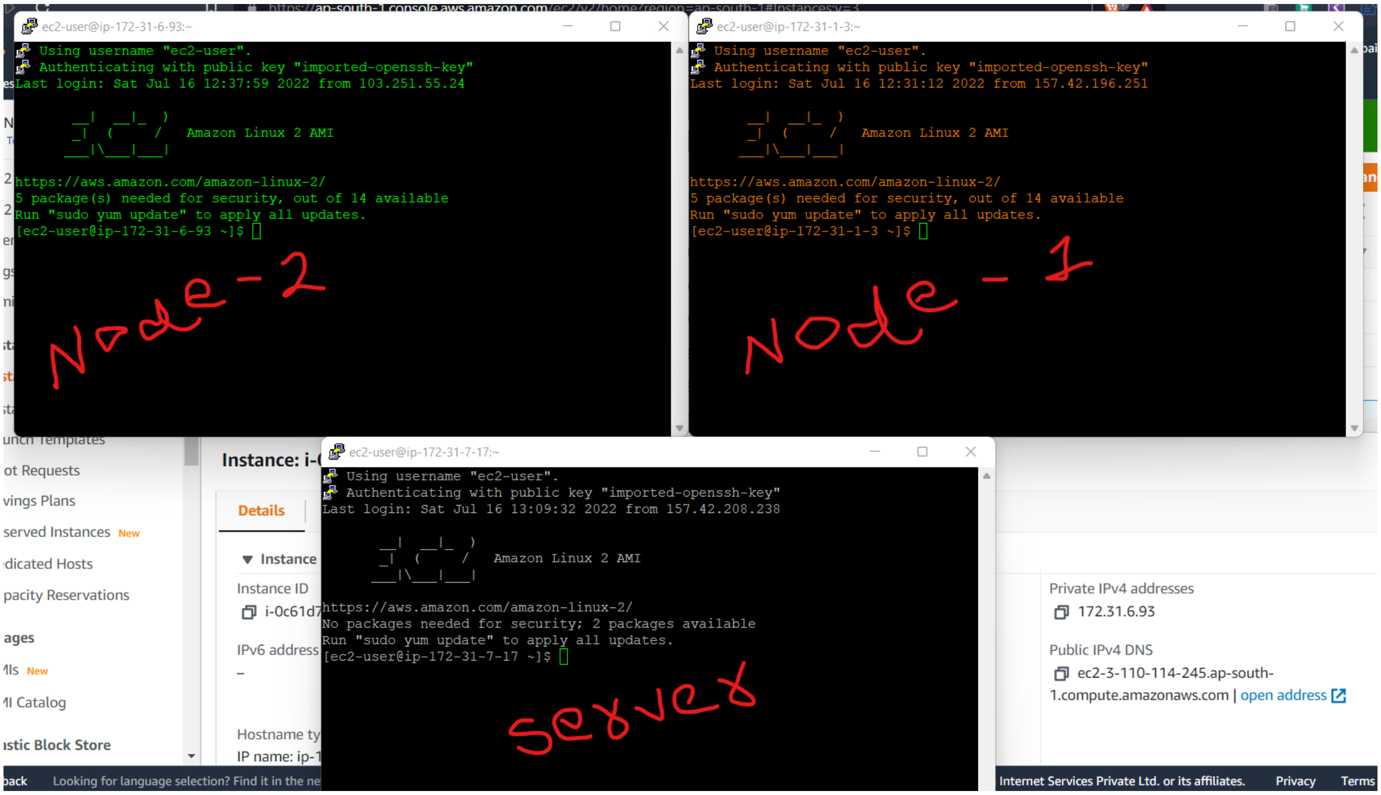
```
ansible demonode[0] --list-hosts
ansible demonode[1] --list-hosts
ansible demonode[2] --list-hosts
```



```
    172.31.6.93
[ansible@ip-172-31-7-17 ~]$ ansible demonode[0] --list-hosts
  hosts (1):
    172.31.1.3
[ansible@ip-172-31-7-17 ~]$ ansible demonode[1] --list-hosts
  hosts (1):
    172.31.6.93
[ansible@ip-172-31-7-17 ~]$ ansible demonode[2] --list-hosts
[WARNING]: No hosts matched, nothing to do
  hosts (0):
[ansible@ip-172-31-7-17 ~]$
```

AD-HOC COMMAND

First login to all ec2-instance(ansibleserver,node1,node2 using putty)

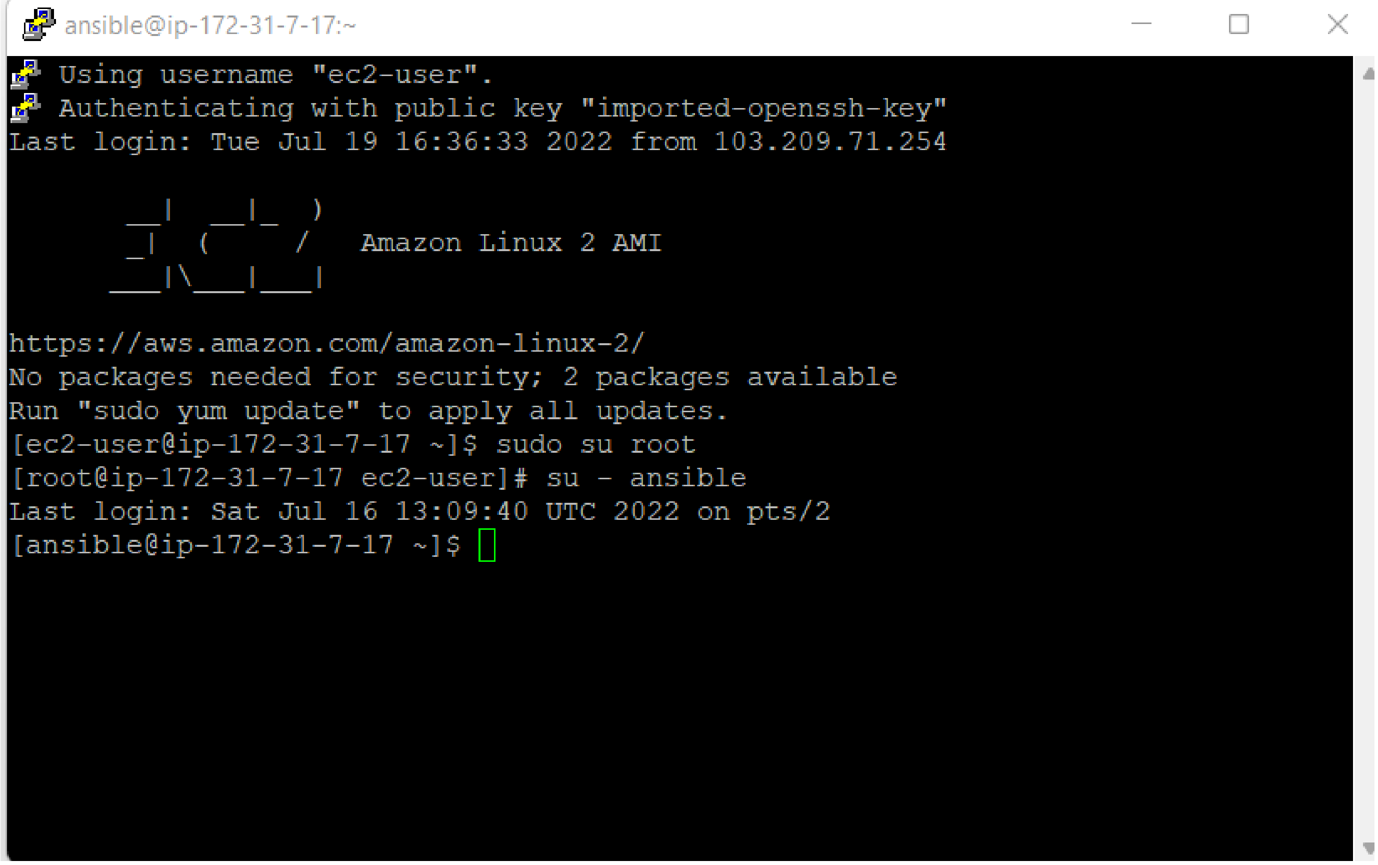


```
ec2-user@ip-172-31-6-93:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sat Jul 16 12:37:59 2022 from 103.251.55.24

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-93 ~]$
```

```
ec2-user@ip-172-31-1-3:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sat Jul 16 12:31:12 2022 from 157.42.196.251

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-3 ~]$
```

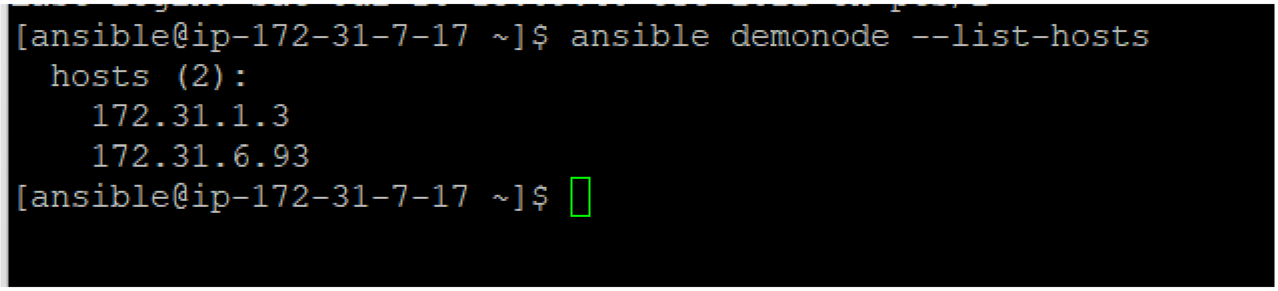
```
ec2-user@ip-172-31-7-17:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sat Jul 16 13:09:32 2022 from 157.42.208.238

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-7-17 ~]$
```

Run this command on ansible server as a ansible user



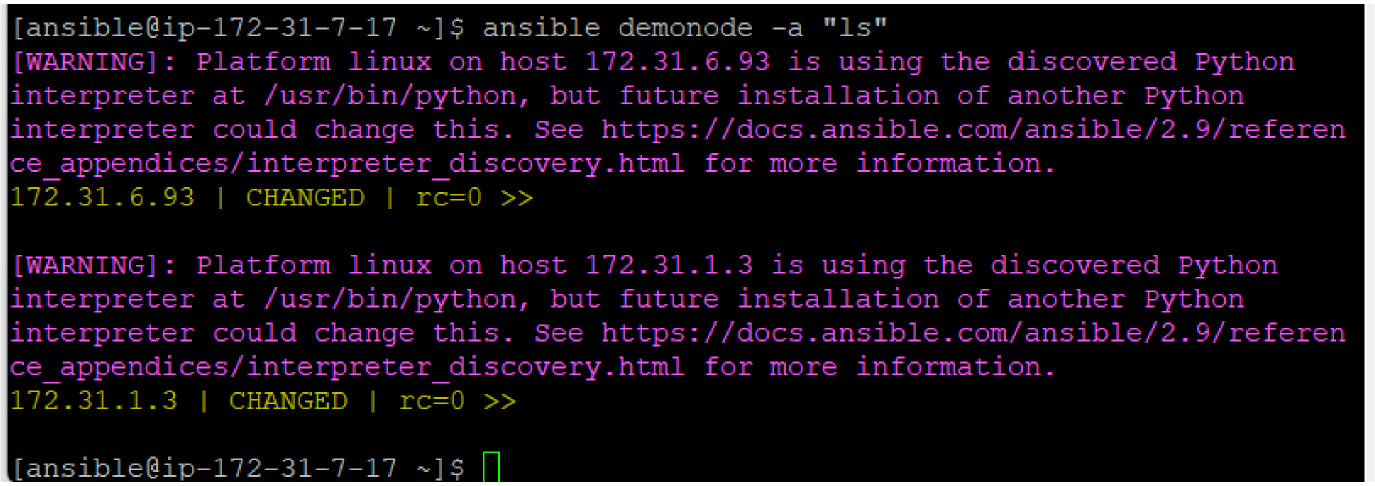
```
ansible@ip-172-31-7-17:~
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Tue Jul 19 16:36:33 2022 from 103.209.71.254

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-7-17 ~]$ sudo su root
[root@ip-172-31-7-17 ec2-user]# su - ansible
Last login: Sat Jul 16 13:09:40 UTC 2022 on pts/2
[ansible@ip-172-31-7-17 ~]$
```

```
ansible demonode --list-hosts
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode --list-hosts
  hosts (2):
    172.31.1.3
    172.31.6.93
[ansible@ip-172-31-7-17 ~]$
```

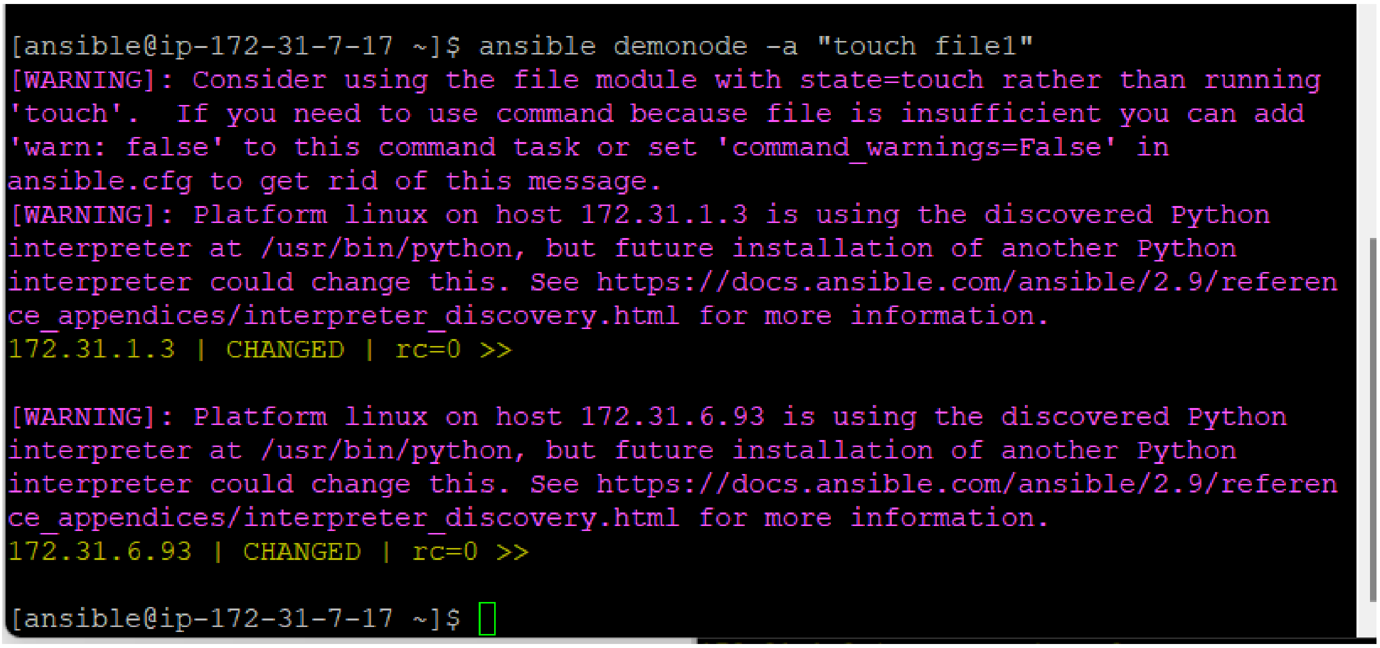
```
ansible demonode -a "ls"
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -a "ls"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED | rc=0 >>

[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.1.3 | CHANGED | rc=0 >>

[ansible@ip-172-31-7-17 ~]$
```

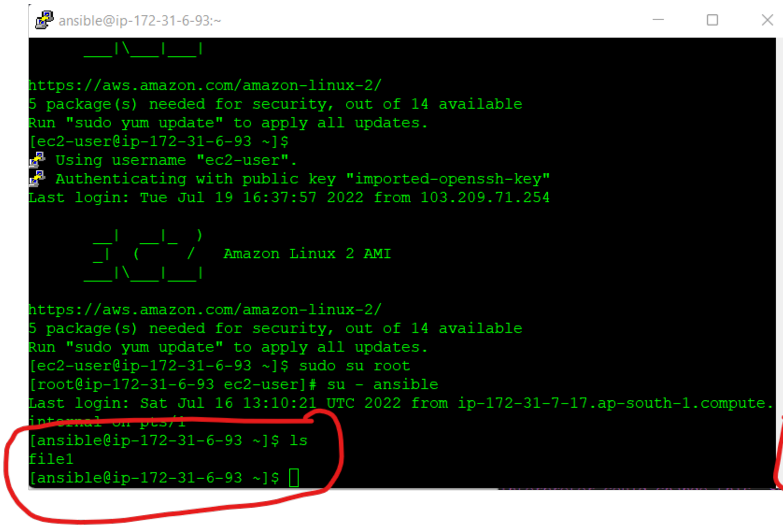
To create a file

```
ansible demonode -a "touch file1"
```



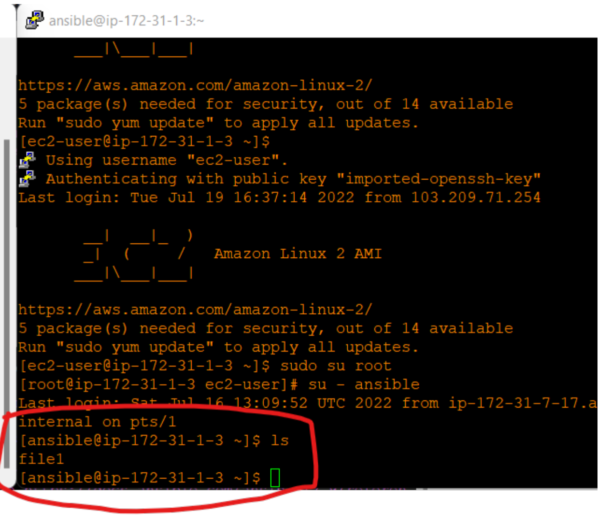
```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -a "touch file1"
[WARNING]: Consider using the file module with state=touch rather than running
'touch'.  If you need to use command because file is insufficient you can add
'warn: false' to this command task or set 'command_warnings=False' in
ansible.cfg to get rid of this message.
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.1.3 | CHANGED | rc=0 >>

[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED | rc=0 >>

[ansible@ip-172-31-7-17 ~]$
```

To check whether file is created in node and node2



ansible@ip-172-31-6-93:~

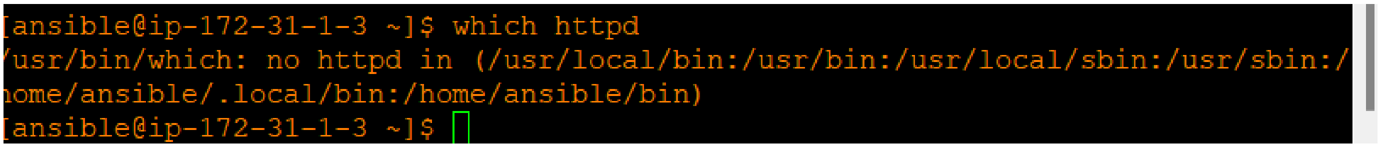
```
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-93 ~]$
 Using username "ec2-user".
 Authenticating with public key "imported-openssh-key"
Last login: Tue Jul 19 16:37:57 2022 from 103.209.71.254

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-93 ~]$ sudo su root
[root@ip-172-31-6-93 ec2-user]# su - ansible
Last login: Sat Jul 16 13:10:21 UTC 2022 from ip-172-31-7-17.ap-south-1.compute.
internal on pts/1
[ansible@ip-172-31-6-93 ~]$ ls
file1
[ansible@ip-172-31-6-93 ~]$
```



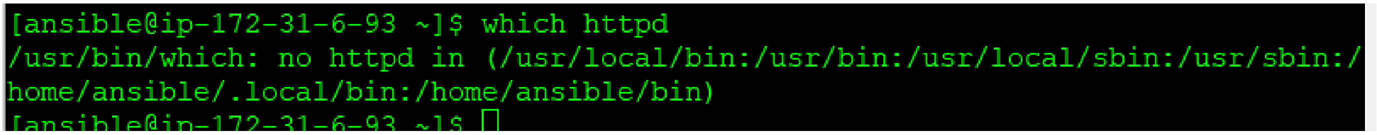
ansible@ip-172-31-1-3:~

```
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-3 ~]$
 Using username "ec2-user".
 Authenticating with public key "imported-openssh-key"
Last login: Tue Jul 19 16:37:14 2022 from 103.209.71.254

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
5 package(s) needed for security, out of 14 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-1-3 ~]$ sudo su root
[root@ip-172-31-1-3 ec2-user]# su - ansible
Last login: Sat Jul 16 13:09:52 UTC 2022 from ip-172-31-7-17.a
internal on pts/1
[ansible@ip-172-31-1-3 ~]$ ls
file1
[ansible@ip-172-31-1-3 ~]$
```

To install httpd package

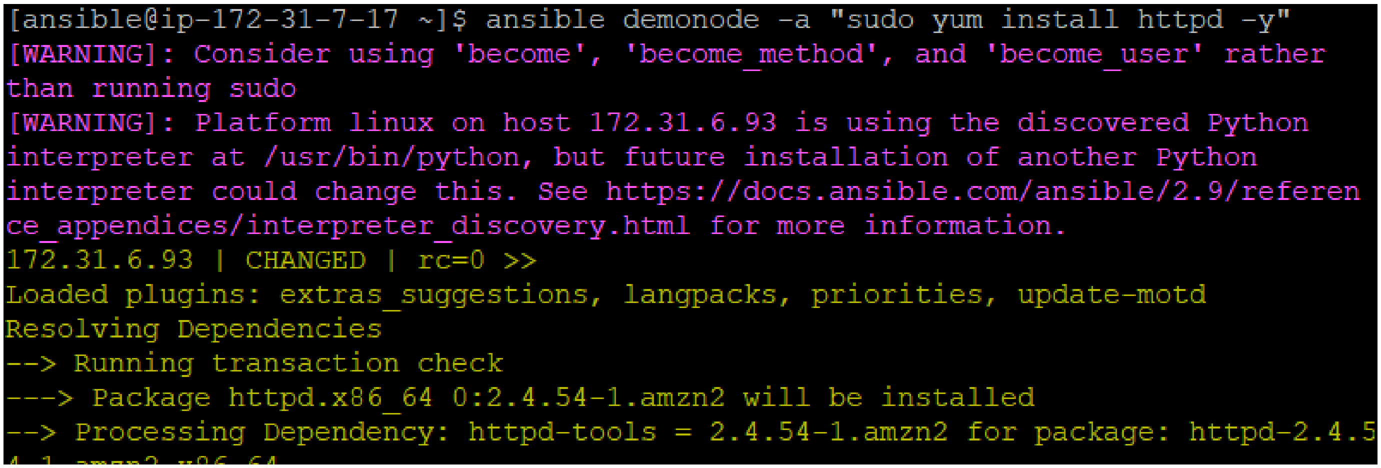


```
[ansible@ip-172-31-1-3 ~]$ which httpd
/usr/bin/which: no httpd in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/
ome/ansible/.local/bin:/home/ansible/bin)
[ansible@ip-172-31-1-3 ~]$
```

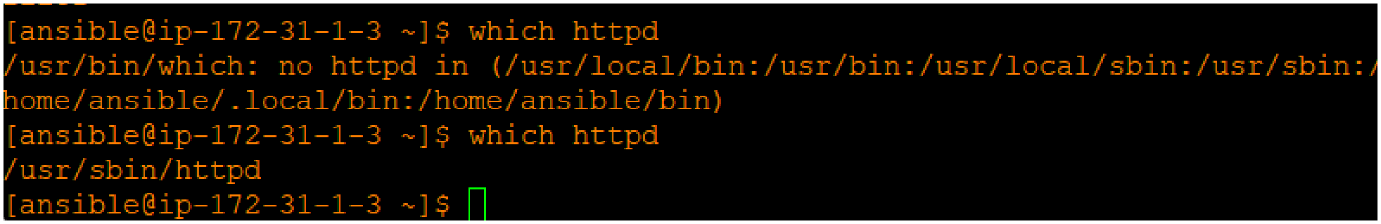


```
[ansible@ip-172-31-6-93 ~]$ which httpd
/usr/bin/which: no httpd in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/
home/ansible/.local/bin:/home/ansible/bin)
[ansible@ip-172-31-6-93 ~]$
```

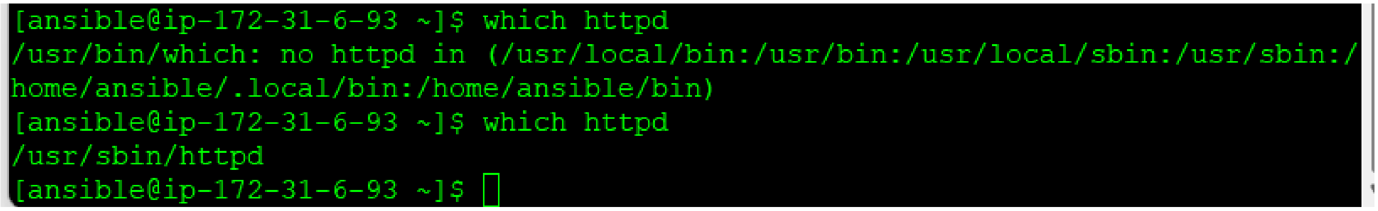
```
ansible demonode -a "sudo yum install httpd -y"
ansible demonode -ba "yum install httpd -y"
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -a "sudo yum install httpd -y"
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED | rc=0 >>
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.5
4-1.amzn2.x86_64
```



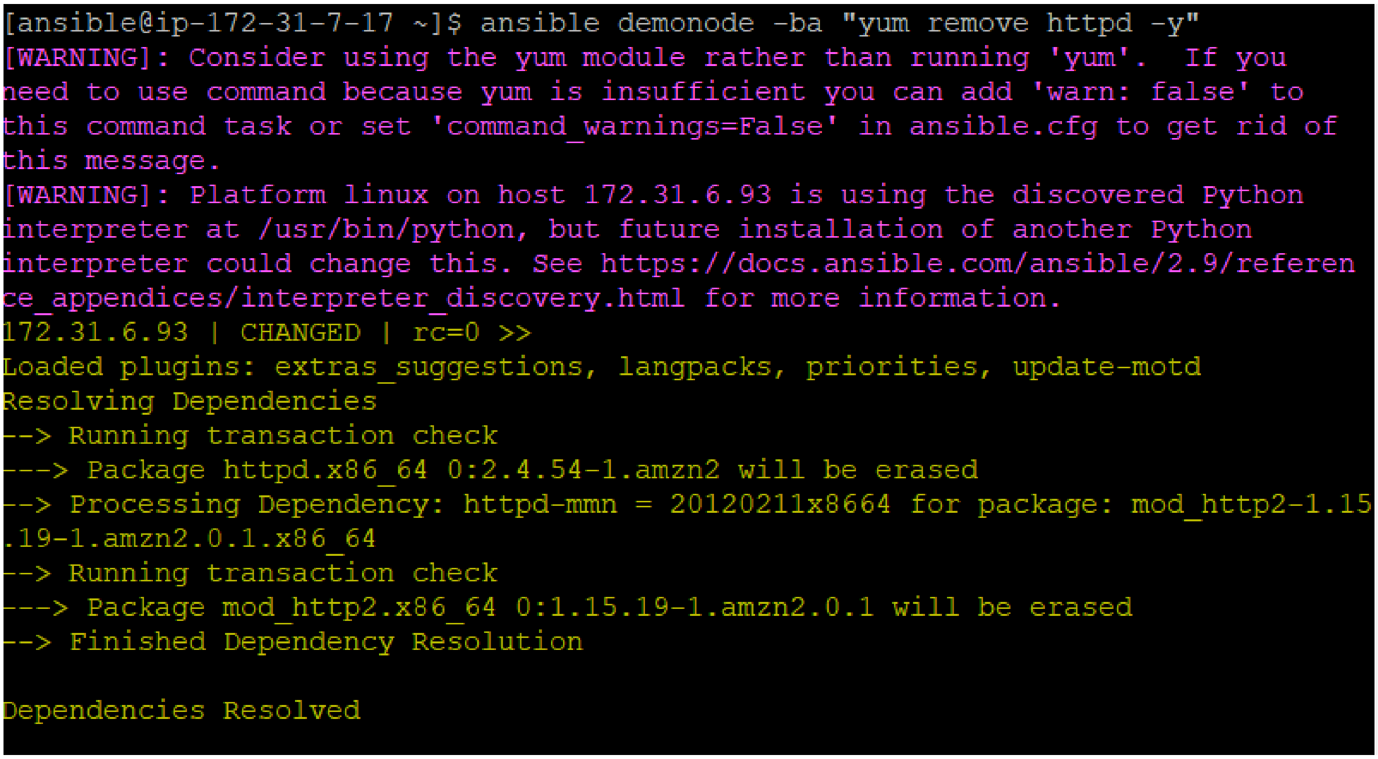
```
[ansible@ip-172-31-1-3 ~]$ which httpd
/usr/bin/which: no httpd in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/
home/ansible/.local/bin:/home/ansible/bin)
[ansible@ip-172-31-1-3 ~]$ which httpd
/usr/sbin/httpd
[ansible@ip-172-31-1-3 ~]$
```



```
[ansible@ip-172-31-6-93 ~]$ which httpd
/usr/bin/which: no httpd in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/
home/ansible/.local/bin:/home/ansible/bin)
[ansible@ip-172-31-6-93 ~]$ which httpd
/usr/sbin/httpd
[ansible@ip-172-31-6-93 ~]$
```

To remove httpd package

```
ansible demonode -ba "yum remove httpd -y"
```

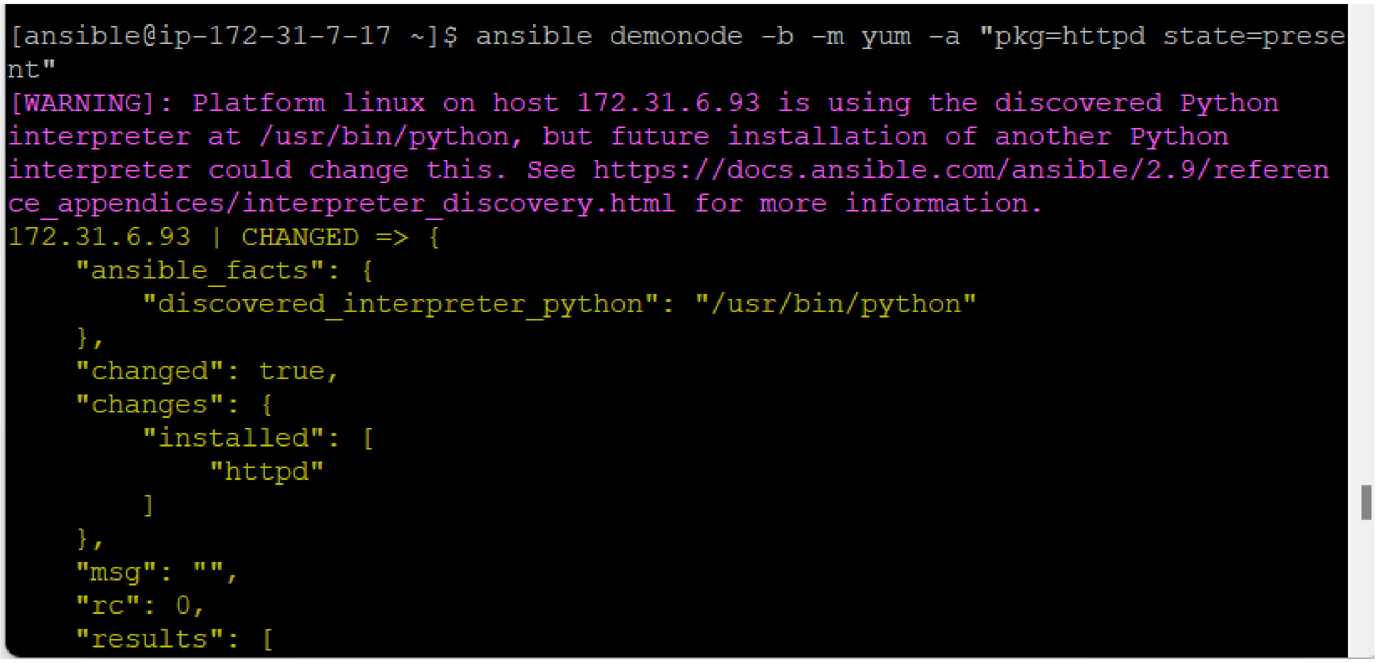


```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -ba "yum remove httpd -y"
[WARNING]: Consider using the yum module rather than running 'yum'.  If you
need to use command because yum is insufficient you can add 'warn: false' to
this command task or set 'command_warnings=False' in ansible.cfg to get rid of
this message.
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED | rc=0 >>
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.54-1.amzn2 will be erased
--> Processing Dependency: httpd-mmn = 20120211x8664 for package: mod_http2-1.15
.19-1.amzn2.0.1.x86_64
--> Running transaction check
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved
```

Ansible Modules

To install httpd package

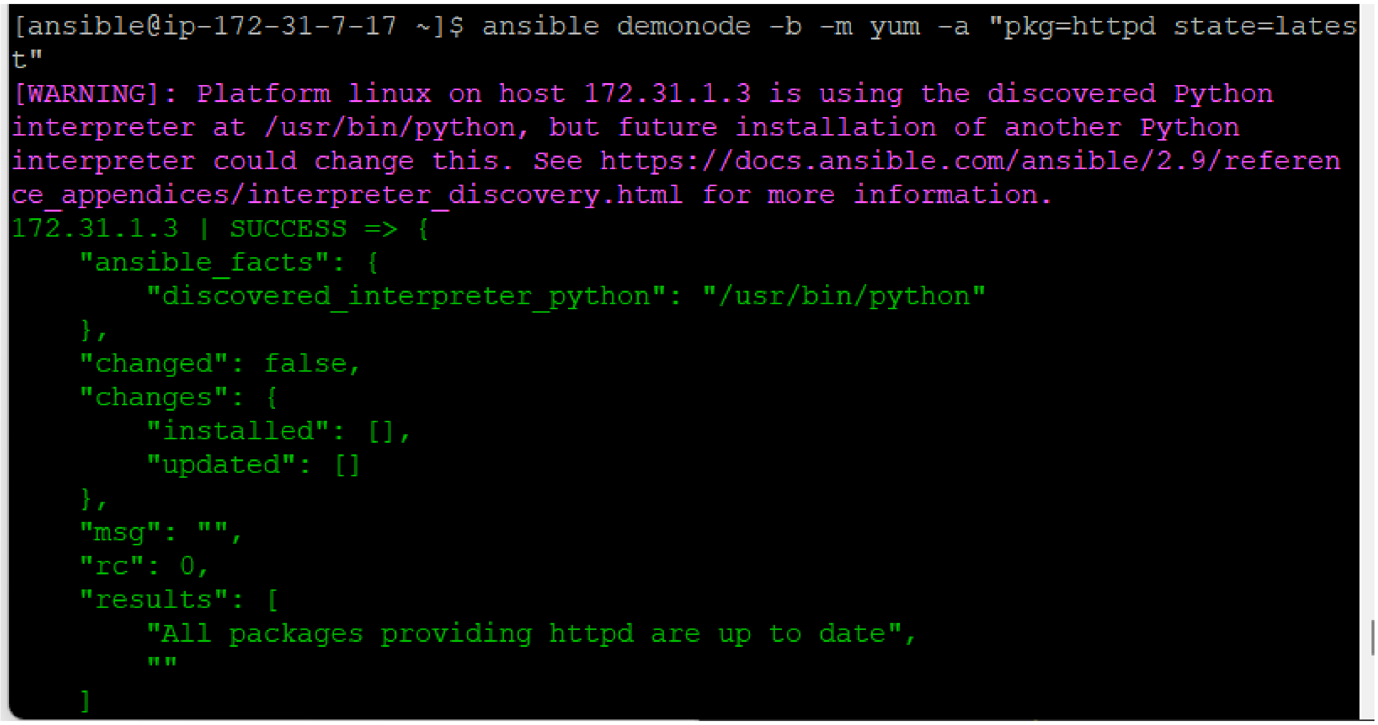
```
ansible demonode -b -m yum -a "pkg=httpd state=present"
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m yum -a "pkg=httpd state=prese
nt"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "changes": {
        "installed": [
            "httpd"
        ]
    },
    "msg": "",
    "rc": 0,
    "results": [
```

To update httpd package

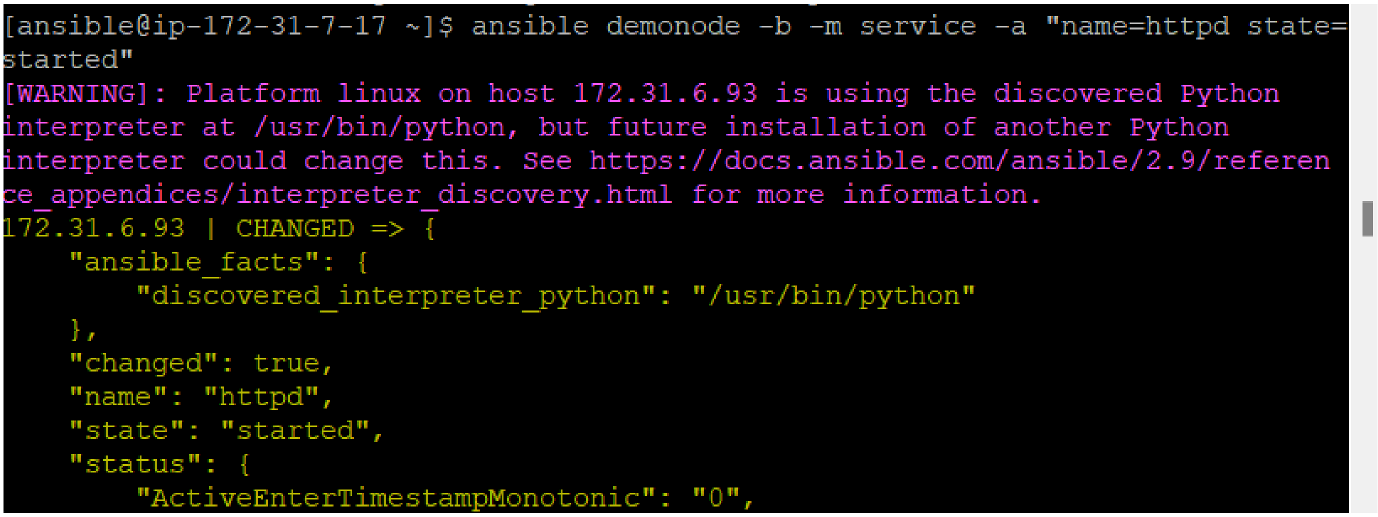
```
ansible demonode -b -m yum -a "pkg=httpd state=latest"
```



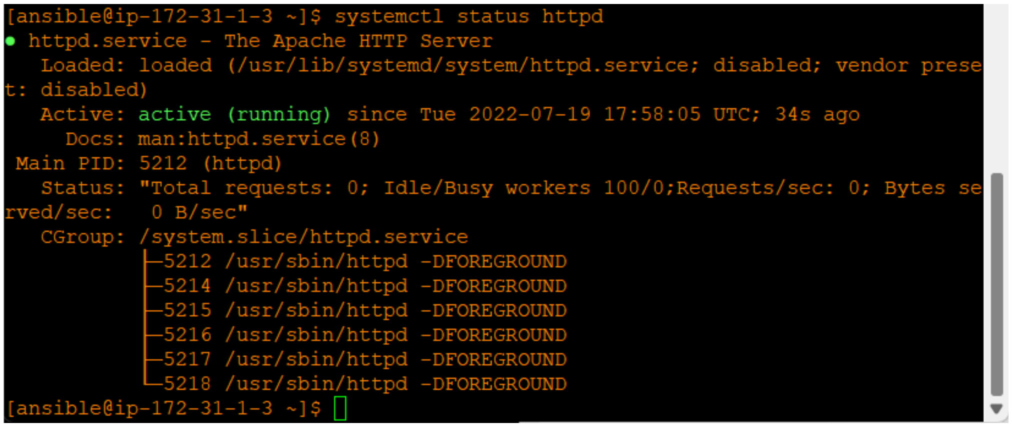
```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m yum -a "pkg=httpd state=lates
t"
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.1.3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "changes": {
        "installed": [],
        "updated": []
    },
    "msg": "",
    "rc": 0,
    "results": [
        "All packages providing httpd are up to date",
        ""
    ]
```

To start the httpd service

```
ansible demonode -b -m service -a "name=httpd state=started"
```



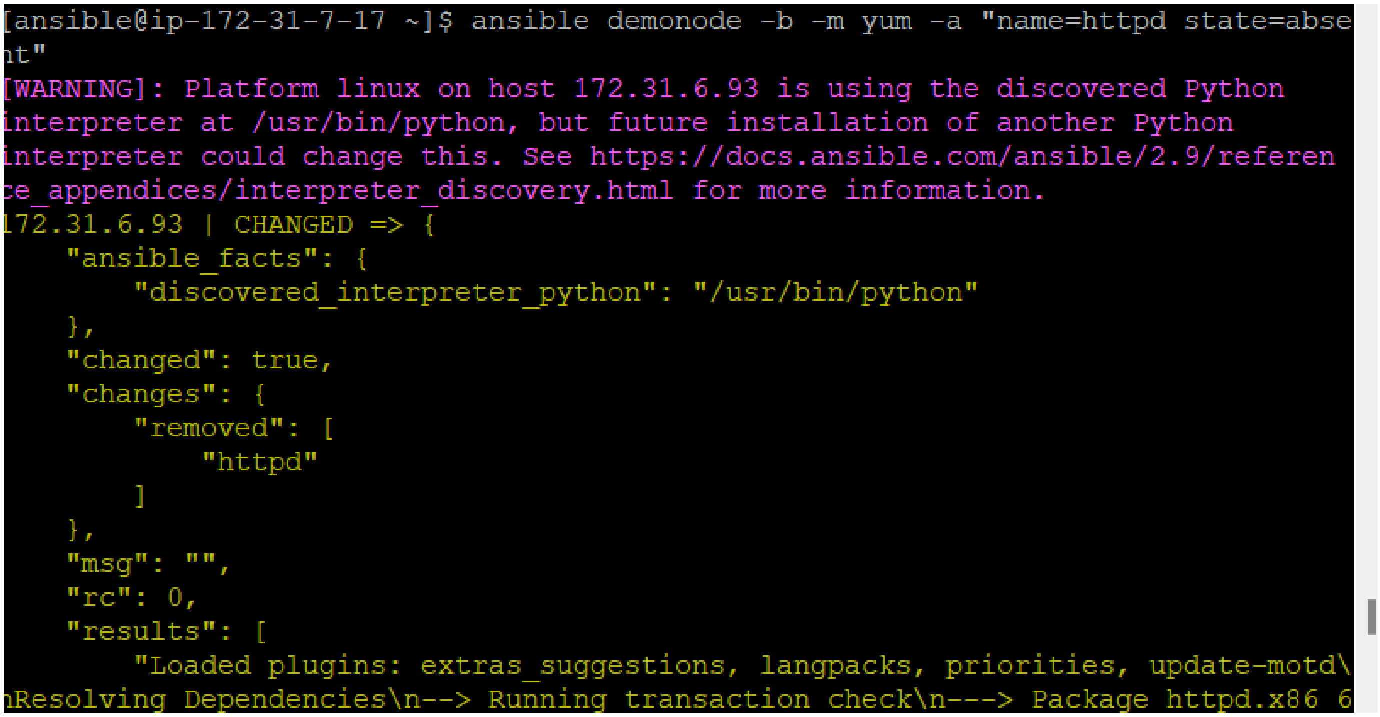
```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m service -a "name=httpd state=
started"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "name": "httpd",
    "state": "started",
    "status": {
        "ActiveEnterTimestampMonotonic": "0",
```



```
[ansible@ip-172-31-1-3 ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since Tue 2022-07-19 17:58:05 UTC; 34s ago
     Docs: man:httpd.service(8)
 Main PID: 5212 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes se
rved/sec:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─5212 /usr/sbin/httpd -DFOREGROUND
           ├─5214 /usr/sbin/httpd -DFOREGROUND
           ├─5215 /usr/sbin/httpd -DFOREGROUND
           ├─5216 /usr/sbin/httpd -DFOREGROUND
           ├─5217 /usr/sbin/httpd -DFOREGROUND
           └─5218 /usr/sbin/httpd -DFOREGROUND
[ansible@ip-172-31-1-3 ~]$
```

To remove httpd package

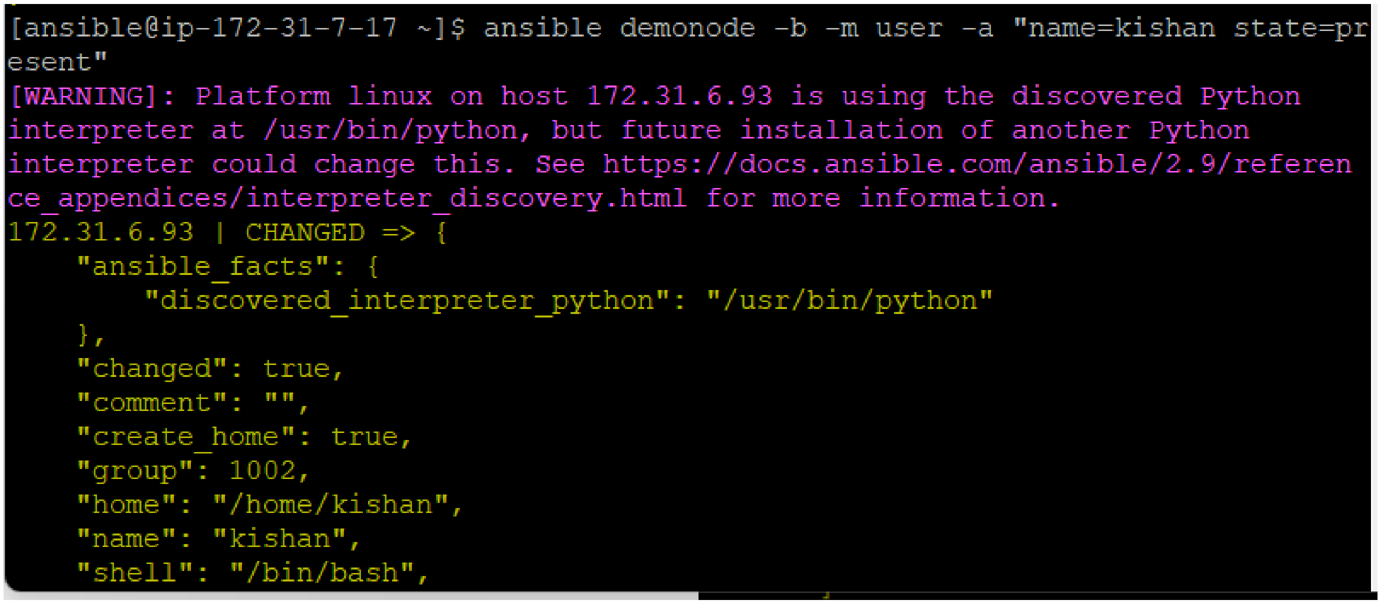
```
ansible demonode -b -m yum -a "name=httpd state=absent"
```



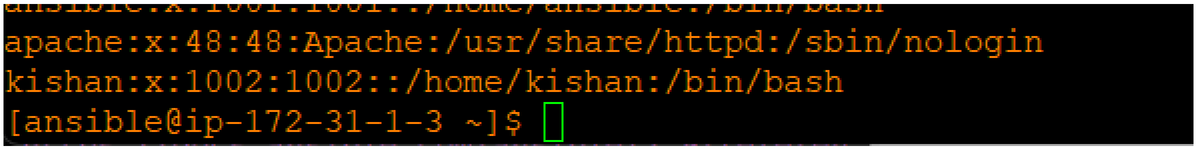
```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m yum -a "name=httpd state=abse
nt"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "changes": {
        "removed": [
            "httpd"
        ]
    },
    "msg": "",
    "rc": 0,
    "results": [
        "Loaded plugins: extras_suggestions, langpacks, priorities, update-motd\
nResolving Dependencies\n--> Running transaction check\n---> Package httpd.x86 6
```

To create a new user

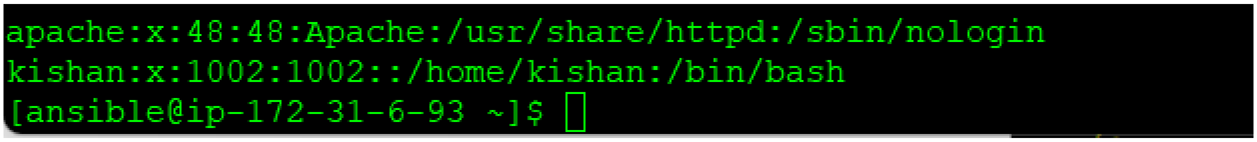
```
ansible demonode -b -m user -a "name=kishan state=present"
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m user -a "name=kishan state=pr
esent"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1002,
    "home": "/home/kishan",
    "name": "kishan",
    "shell": "/bin/bash",
```



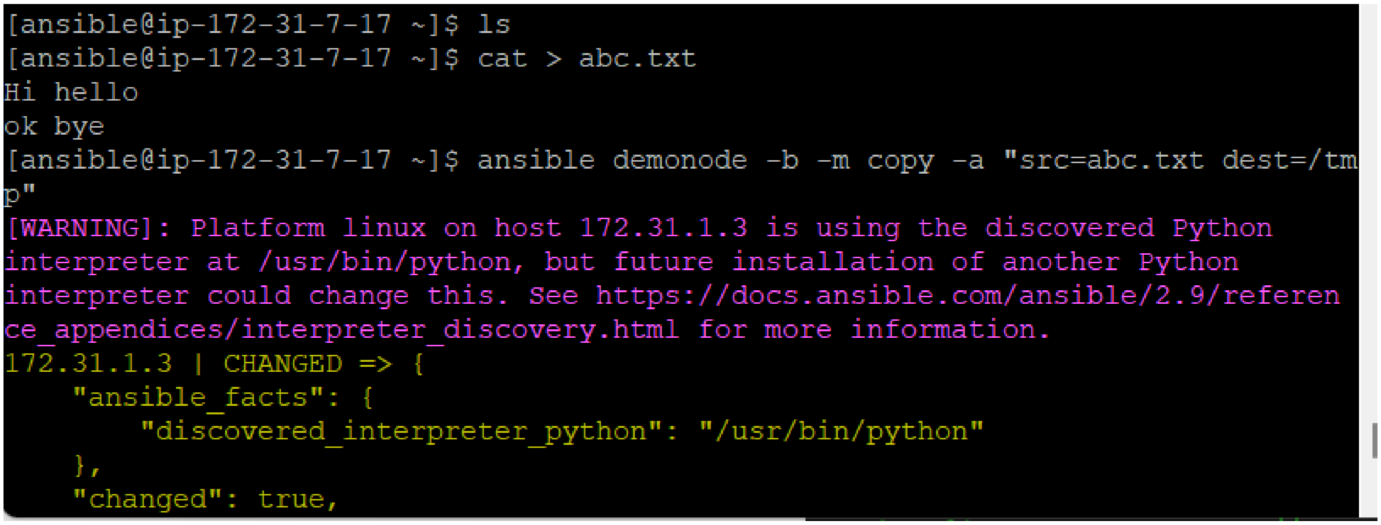
```
ansible:x:1001:1001::/home/ansible:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
kishan:x:1002:1002::/home/kishan:/bin/bash
[ansible@ip-172-31-1-3 ~]$
```



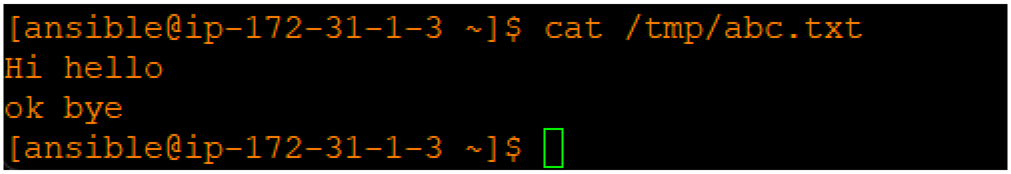
```
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
kishan:x:1002:1002::/home/kishan:/bin/bash
[ansible@ip-172-31-6-93 ~]$
```

To copy a file

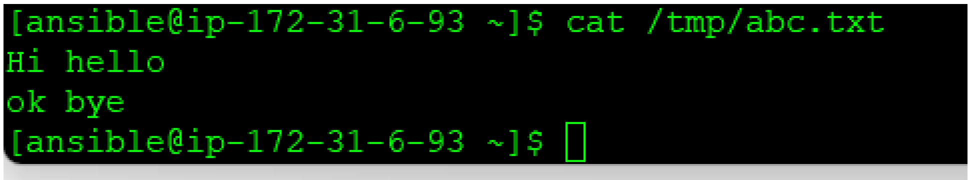
```
cat > abc.txt
hi hello
ok bye

ansible demonode -b -m copy -a "src=abc.txt dest=/tmp"
```



```
[ansible@ip-172-31-7-17 ~]$ ls
[ansible@ip-172-31-7-17 ~]$ cat > abc.txt
Hi hello
ok bye
[ansible@ip-172-31-7-17 ~]$ ansible demonode -b -m copy -a "src=abc.txt dest=/tm
p"
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.1.3 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
```



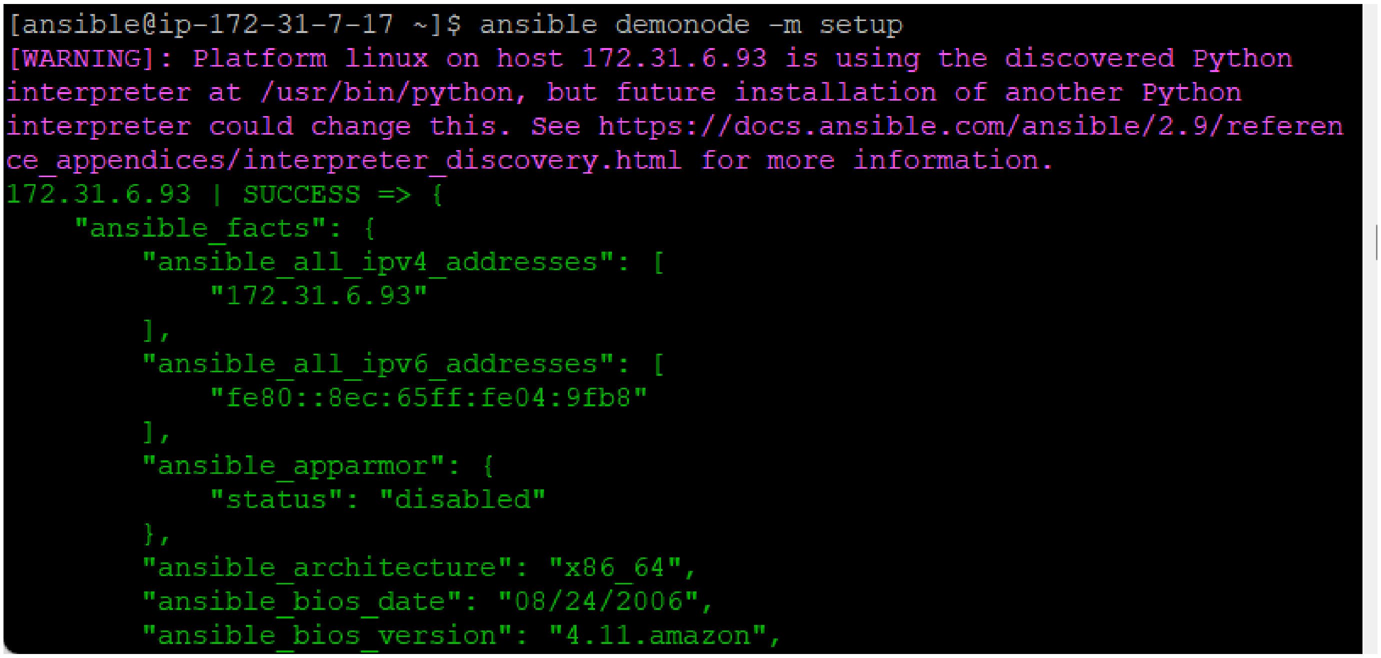
```
[ansible@ip-172-31-1-3 ~]$ cat /tmp/abc.txt
Hi hello
ok bye
[ansible@ip-172-31-1-3 ~]$
```



```
[ansible@ip-172-31-6-93 ~]$ cat /tmp/abc.txt
Hi hello
ok bye
[ansible@ip-172-31-6-93 ~]$
```

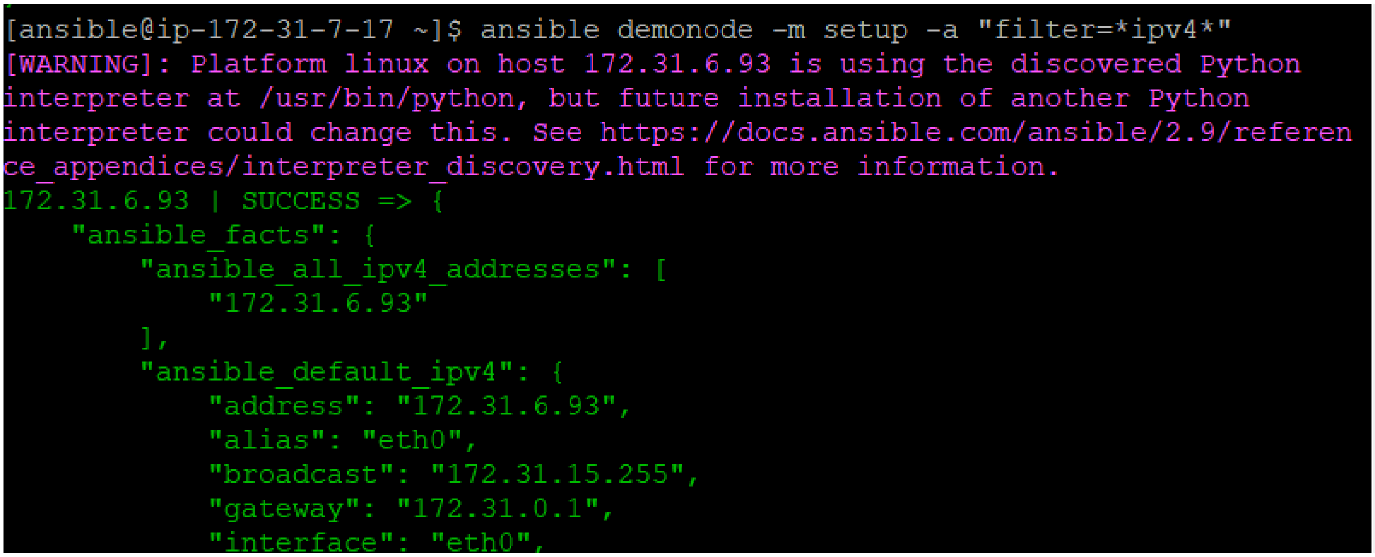
To see setup details

```
ansible demonode -m setup
```



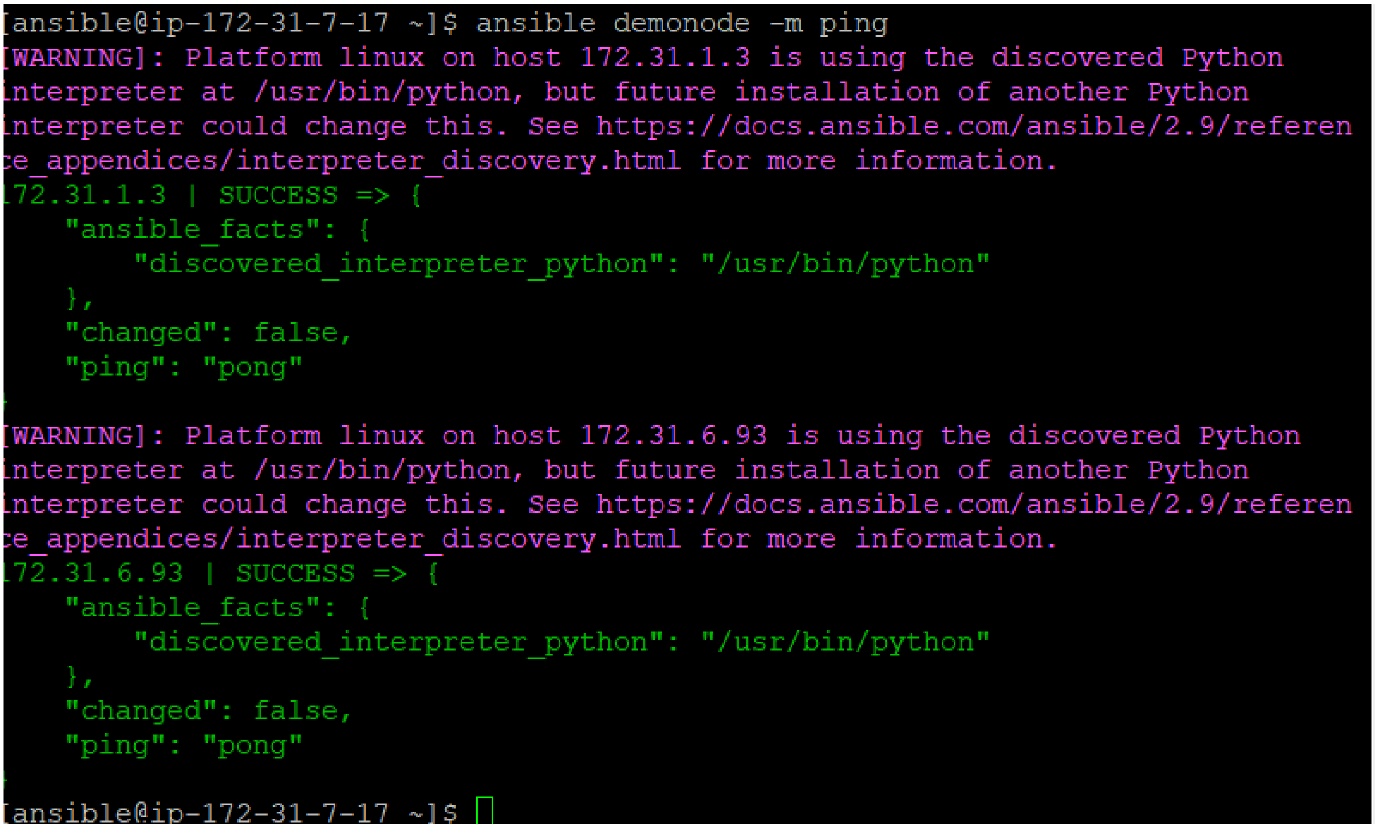
```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -m setup
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.31.6.93"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::8ec:65ff:fe04:9fb8"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "08/24/2006",
        "ansible_bios_version": "4.11.amazon",
```

```
ansible demonode -m setup -a "filter=*ipv4*"
```



```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -m setup -a "filter=*ipv4*"
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.31.6.93"
        ],
        "ansible_default_ipv4": {
            "address": "172.31.6.93",
            "alias": "eth0",
            "broadcast": "172.31.15.255",
            "gateway": "172.31.0.1",
            "interface": "eth0",
```

```
ansible demonode -m ping
```

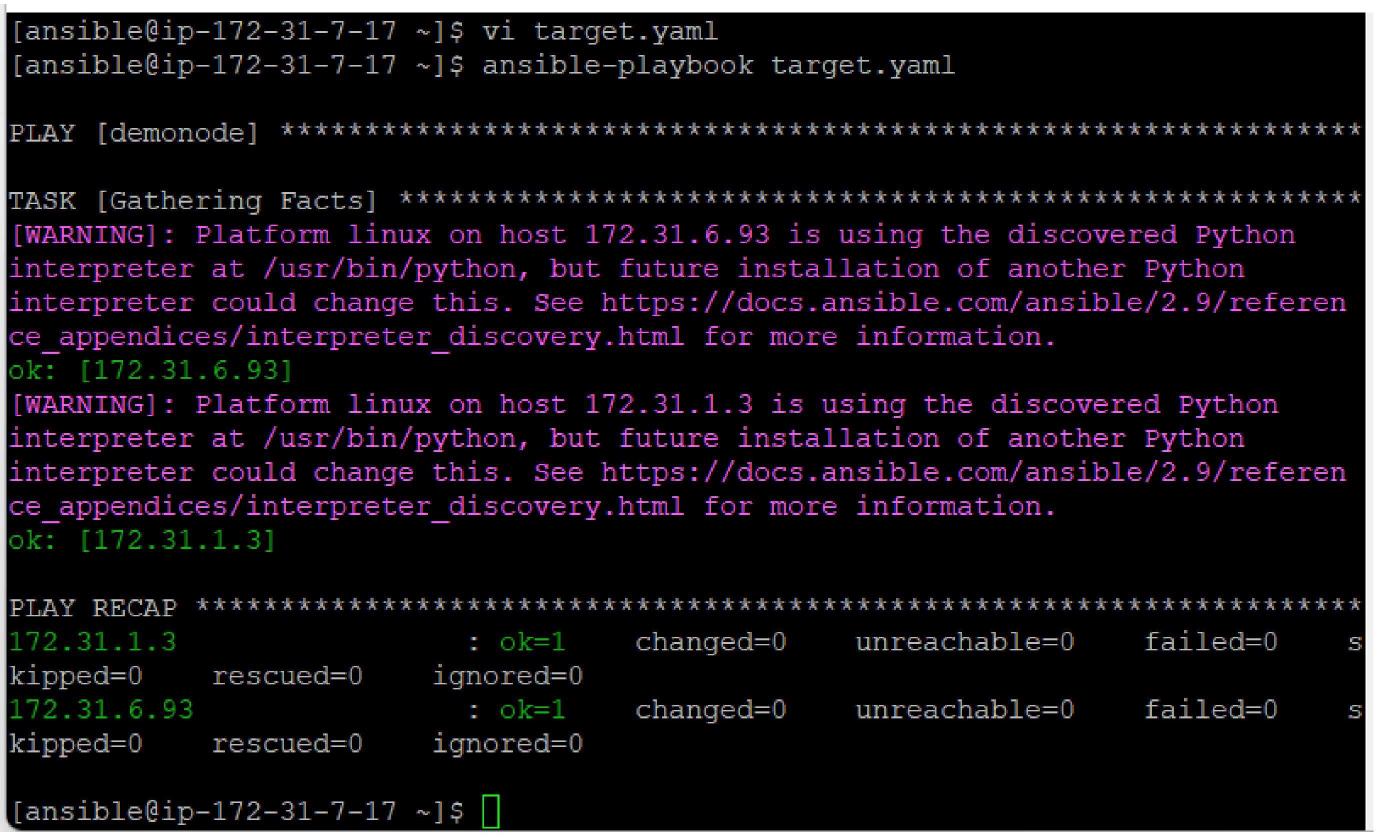


```
[ansible@ip-172-31-7-17 ~]$ ansible demonode -m ping
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.1.3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.6.93 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[ansible@ip-172-31-7-17 ~]$
```

To write a playbook

1. simple playbook to get private ip of node

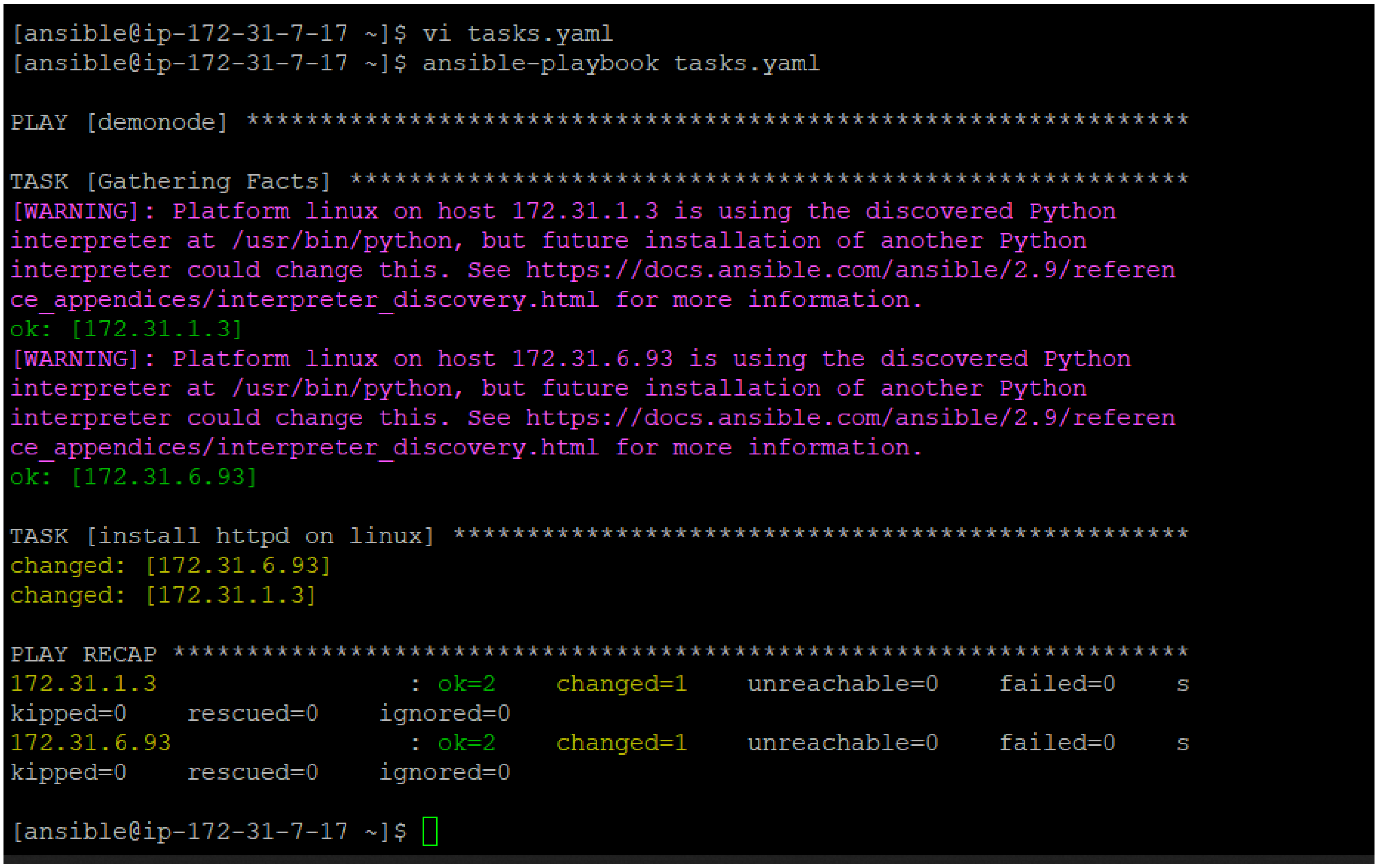
```
vi target.yaml

- hosts: demonode
  user: ansible
  become: yes
  connection: ssh
  gather_facts: true
```



```
[ansible@ip-172-31-7-17 ~]$ vi target.yaml
[ansible@ip-172-31-7-17 ~]$ ansible-playbook target.yaml

PLAY [demonode] ****************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.6.93]
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.1.3]

PLAY RECAP *********************************************************************
172.31.1.3                 : ok=1    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.6.93                : ok=1    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[ansible@ip-172-31-7-17 ~]$
```

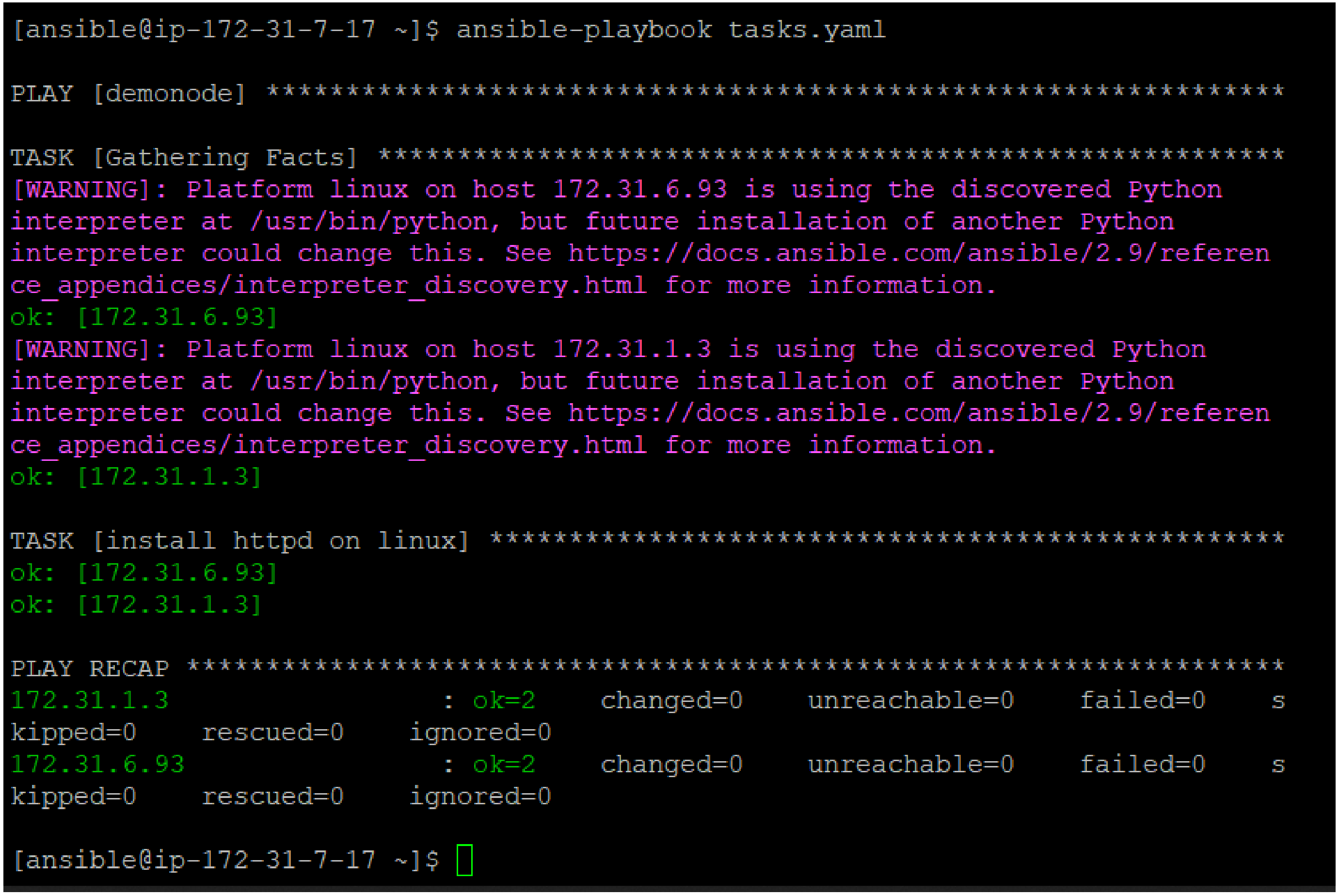
2. playbook to install httpd

```
vi task.yaml

- hosts: demonode
  user: ansible
  become: yes
  connection: ssh
  tasks:
  - name: install httpd on linux
    action: yum name=httpd state=present
```



```
[ansible@ip-172-31-7-17 ~]$ vi tasks.yaml
[ansible@ip-172-31-7-17 ~]$ ansible-playbook tasks.yaml

PLAY [demonode] ****************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.1.3]
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.6.93]

TASK [install httpd on linux] **************************************************
changed: [172.31.6.93]
changed: [172.31.1.3]

PLAY RECAP *********************************************************************
172.31.1.3                 : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.6.93                : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[ansible@ip-172-31-7-17 ~]$
```

Try to run it again

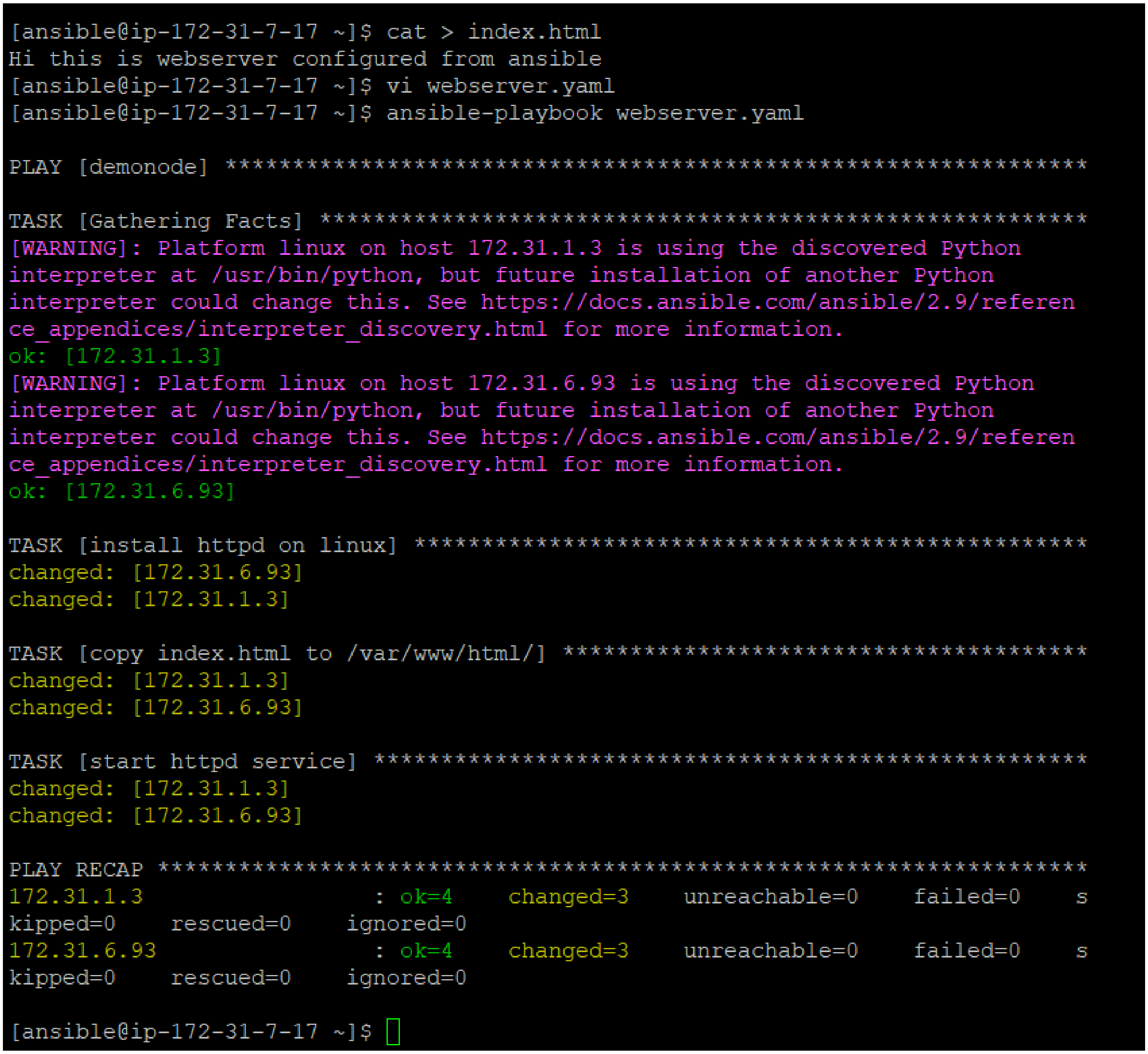


```
[ansible@ip-172-31-7-17 ~]$ ansible-playbook tasks.yaml

PLAY [demonode] ****************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.6.93]
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.1.3]

TASK [install httpd on linux] **************************************************
ok: [172.31.6.93]
ok: [172.31.1.3]

PLAY RECAP *********************************************************************
172.31.1.3                 : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.6.93                : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[ansible@ip-172-31-7-17 ~]$
```

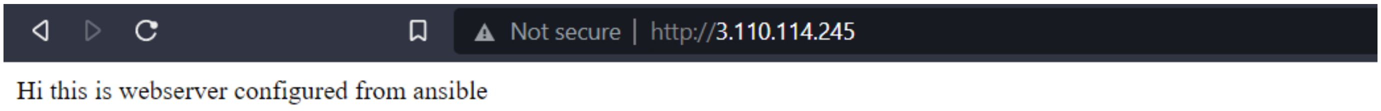
To create a webserver from playbook

```
cat > index.html
Hi this is webserver configured from ansible
```

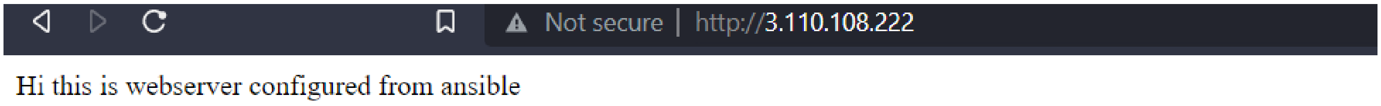
```
vi webserver.yaml

- hosts: demonode
  user: ansible
  become: yes
  connection: ssh
  tasks:
  - name: install httpd on linux
    action: yum name=httpd state=present
  - name: copy index.html to /var/www/html/
    action: copy src=index.html dest=/var/www/html
  - name: start httpd service
    action: service name=httpd state=started
```



```
[ansible@ip-172-31-7-17 ~]$ cat > index.html
Hi this is webserver configured from ansible
[ansible@ip-172-31-7-17 ~]$ vi webserver.yaml
[ansible@ip-172-31-7-17 ~]$ ansible-playbook webserver.yaml

PLAY [demonode] ****************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 172.31.1.3 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.1.3]
[WARNING]: Platform linux on host 172.31.6.93 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.6.93]

TASK [install httpd on linux] **************************************************
changed: [172.31.6.93]
changed: [172.31.1.3]

TASK [copy index.html to /var/www/html/] ***************************************
changed: [172.31.1.3]
changed: [172.31.6.93]

TASK [start httpd service] *****************************************************
changed: [172.31.1.3]
changed: [172.31.6.93]

PLAY RECAP *********************************************************************
172.31.1.3                 : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.6.93                : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[ansible@ip-172-31-7-17 ~]$
```



Hi this is webserver configured from ansible



Hi this is webserver configured from ansible

over…………………