ANSHIKA SRIVASTAVA

ROLL NUMBER – R2142220907

SAP ID – 500107049

LAB EXERCISE 1

Lab Exercise 01

Creating Static Host Inventory

Objective: To create a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible

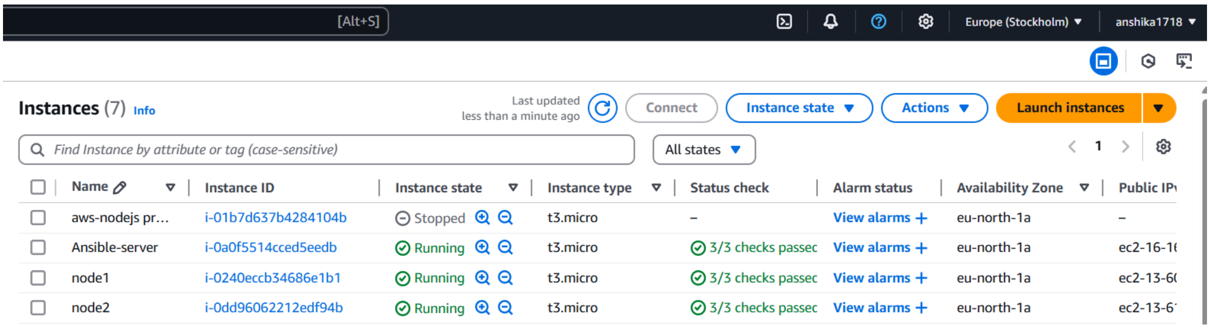
Tools required: Ubuntu OS

Prerequisites: You need to have Ansible installed to proceed with this demo

Steps to be followed:

1. Generate SSH key pair on the main node
2. Copy the SSH key to the two other nodes
3. Update the inventory or host file with the host IP address
4. Establish connectivity between the hosts specified in the host file and the Ansible server

First we need to start 3 Amazon Linux machines



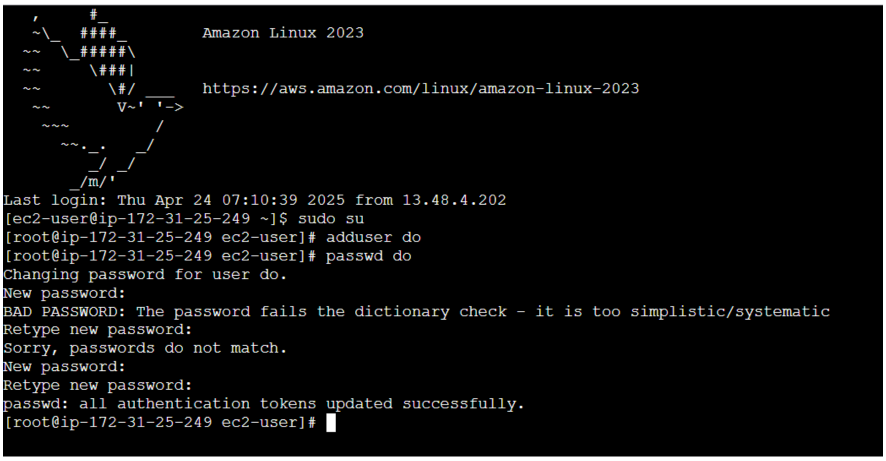
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
□	aws-nodejs pr…	i-01b7d637b4284104b	Stopped	t3.micro	–	View alarms +	eu-north-1a	–
□	Ansible-server	i-0a0f5514cced5eedb	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	ec2-16-16
□	node1	i-0240eccb34686e1b1	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	ec2-13-60
□	node2	i-0dd96062212edf94b	Running	t3.micro	3/3 checks passec	View alarms +	eu-north-1a	ec2-13-61

For all 3 machines -

Make user for all by using command – ***‘sudo su’***

Add user by using ***‘adduser<username>’***

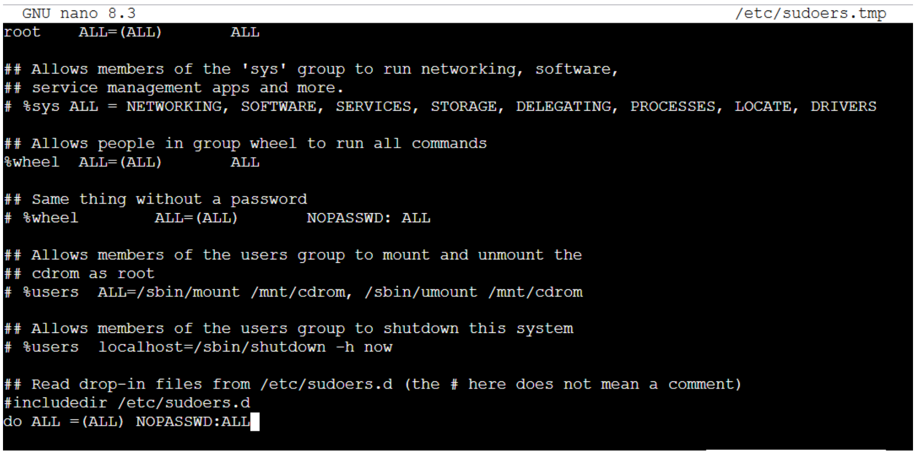
Password by adding ***‘passwd <username>***’ *--enterpassword*



```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Apr 24 07:10:39 2025 from 13.48.4.202
[ec2-user@ip-172-31-25-249 ~]$ sudo su
[root@ip-172-31-25-249 ec2-user]# adduser do
[root@ip-172-31-25-249 ec2-user]# passwd do
Changing password for user do.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-25-249 ec2-user]#
```

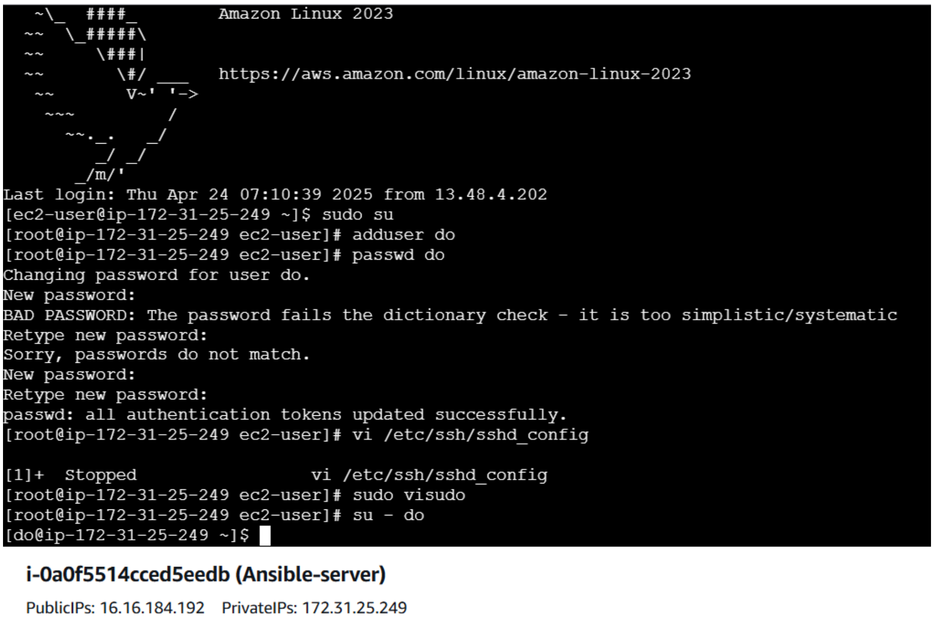
To **make as root user** open vi.sudo file command – *‘vi.sudo’*

‘do ALL=(ALL) NOPASSWD:ALL’



```
  GNU nano 8.3                                                          /etc/sudoers.tmp
root    ALL=(ALL)       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 unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
do ALL =(ALL) NOPASSWD:ALL
```

Now to **switch to user from root user** use *‘su – do’*

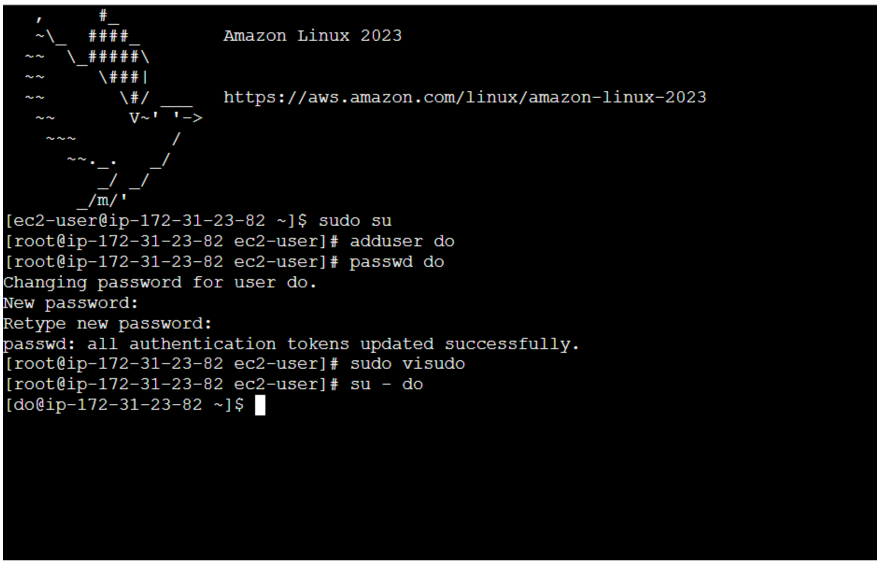


```
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Apr 24 07:10:39 2025 from 13.48.4.202
[ec2-user@ip-172-31-25-249 ~]$ sudo su
[root@ip-172-31-25-249 ec2-user]# adduser do
[root@ip-172-31-25-249 ec2-user]# passwd do
Changing password for user do.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-25-249 ec2-user]# vi /etc/ssh/sshd_config

[1]+  Stopped                 vi /etc/ssh/sshd_config
[root@ip-172-31-25-249 ec2-user]# sudo visudo
[root@ip-172-31-25-249 ec2-user]# su - do
[do@ip-172-31-25-249 ~]$
```

i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

Similarly, for node 1 –

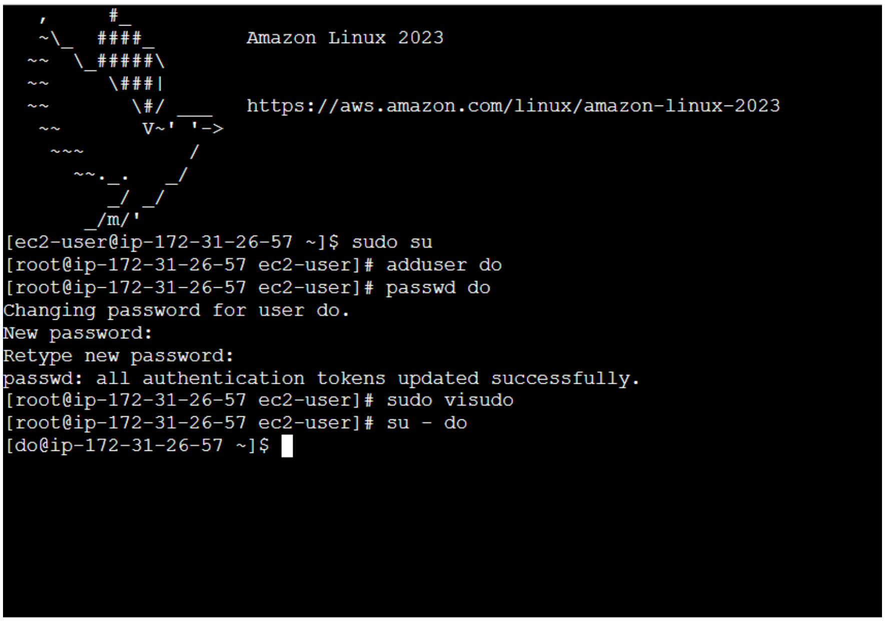


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-23-82 ~]$ sudo su
[root@ip-172-31-23-82 ec2-user]# adduser do
[root@ip-172-31-23-82 ec2-user]# passwd do
Changing password for user do.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-23-82 ec2-user]# sudo visudo
[root@ip-172-31-23-82 ec2-user]# su - do
[do@ip-172-31-23-82 ~]$
```

i-0240eccb34686e1b1 (node1)

PublicIPs: 13.60.238.244   PrivateIPs: 172.31.23.82

Node 2 –

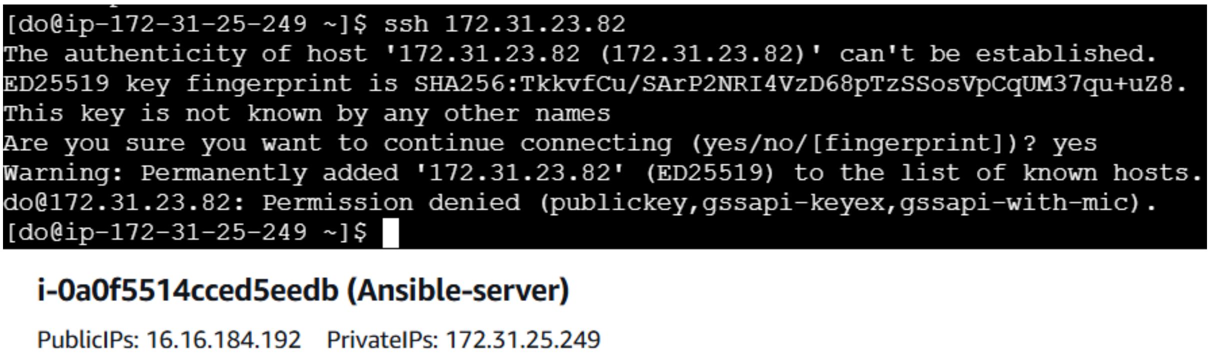


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-26-57 ~]$ sudo su
[root@ip-172-31-26-57 ec2-user]# adduser do
[root@ip-172-31-26-57 ec2-user]# passwd do
Changing password for user do.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-26-57 ec2-user]# sudo visudo
[root@ip-172-31-26-57 ec2-user]# su - do
[do@ip-172-31-26-57 ~]$
```

i-0dd96062212edf94b (node2)

PublicIPs: 13.61.181.231   PrivateIPs: 172.31.26.57

Now trying to ssh into node 1 from Ansible server



```
[do@ip-172-31-25-249 ~]$ ssh 172.31.23.82
The authenticity of host '172.31.23.82 (172.31.23.82)' can't be established.
ED25519 key fingerprint is SHA256:TkkvfCu/SArP2NRI4VzD68pTzSSosVpCqUM37qu+uZ8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.23.82' (ED25519) to the list of known hosts.
do@172.31.23.82: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[do@ip-172-31-25-249 ~]$ 
```

i-0a0f5514cced5eedb (Ansible-server)

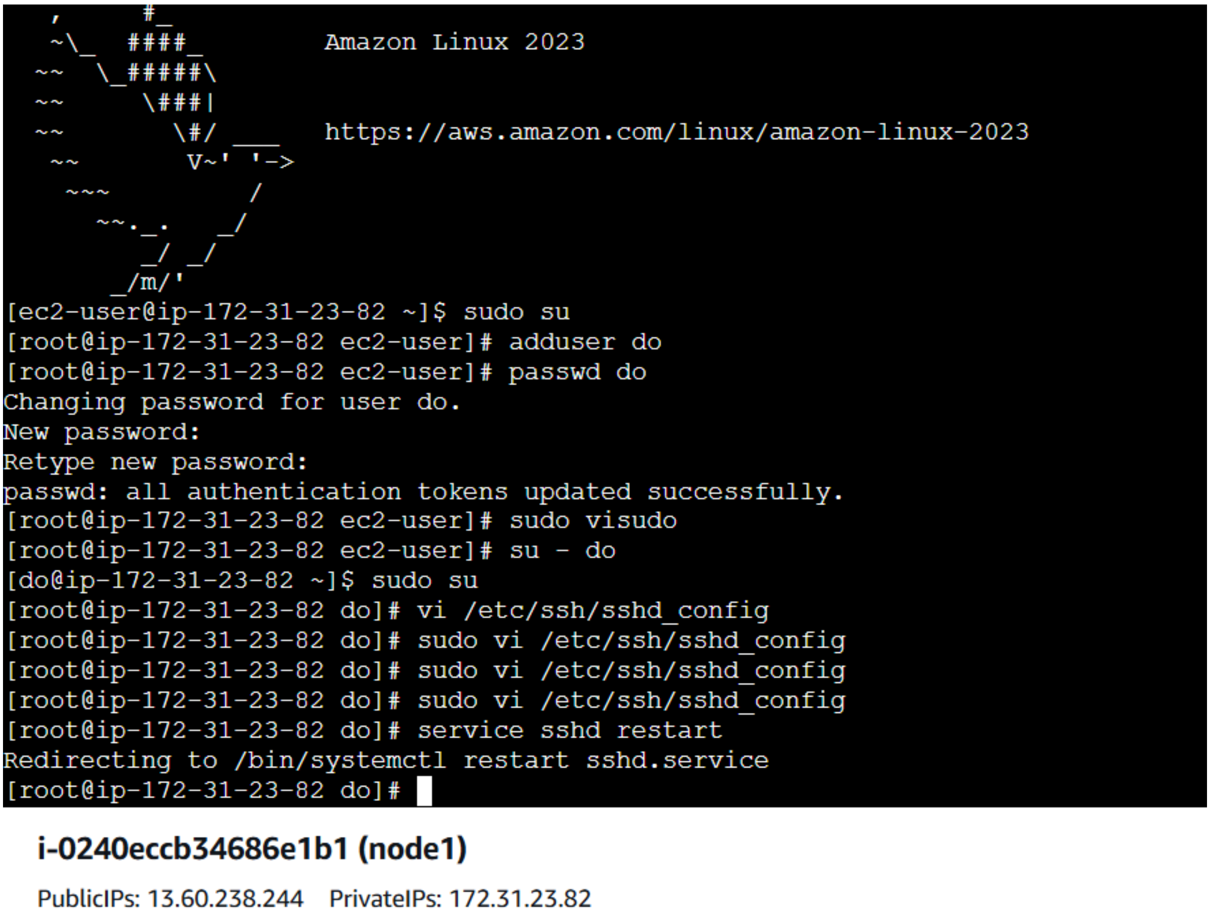
PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

Permission denied. So,make settings in node 1 and node 2 to allow.

Go to root user of node 1 and node 2 by *‘sudo su’*

Navigate to *‘vi /etc/ssh/sshd_config’* and uncomment permit root login and comment PasswordAuthentication

Restart the service using *‘service sshd restart’*

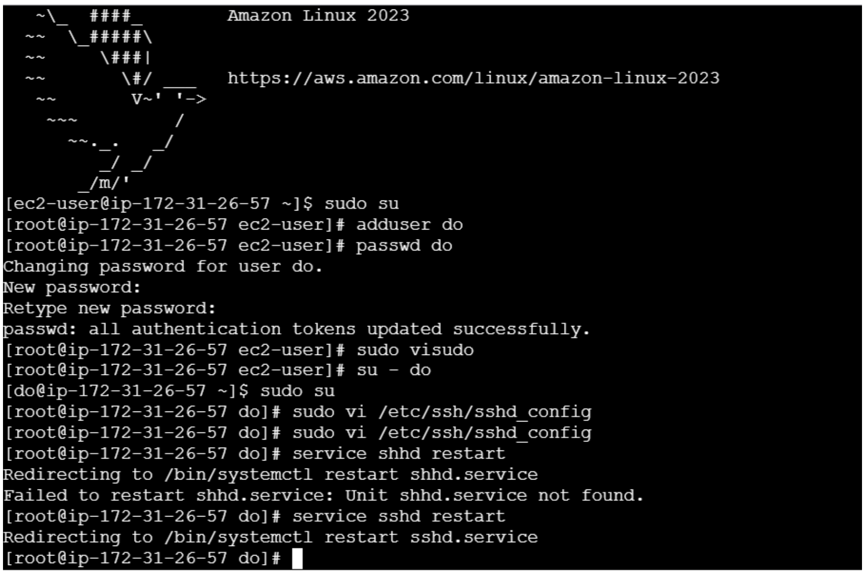


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-23-82 ~]$ sudo su
[root@ip-172-31-23-82 ec2-user]# adduser do
[root@ip-172-31-23-82 ec2-user]# passwd do
Changing password for user do.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-23-82 ec2-user]# sudo visudo
[root@ip-172-31-23-82 ec2-user]# su - do
[do@ip-172-31-23-82 ~]$ sudo su
[root@ip-172-31-23-82 do]# vi /etc/ssh/sshd_config
[root@ip-172-31-23-82 do]# sudo vi /etc/ssh/sshd_config
[root@ip-172-31-23-82 do]# sudo vi /etc/ssh/sshd_config
[root@ip-172-31-23-82 do]# sudo vi /etc/ssh/sshd_config
[root@ip-172-31-23-82 do]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-23-82 do]# 
```

i-0240eccb34686e1b1 (node1)

PublicIPs: 13.60.238.244   PrivateIPs: 172.31.23.82

Repeat the process for node 2

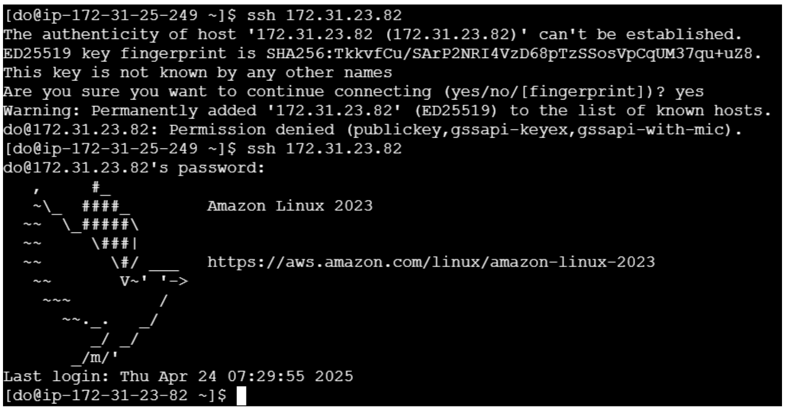


```
  ~\_  ####_        Amazon Linux 2023
 ~~  \_#####\
 ~~     \###|
 ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
  ~~       V~' '->
   ~~~         /
     ~~._.   _/
        _/ _/
      _/m/'
[ec2-user@ip-172-31-26-57 ~]$ sudo su
[root@ip-172-31-26-57 ec2-user]# adduser do
[root@ip-172-31-26-57 ec2-user]# passwd do
Changing password for user do.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-26-57 ec2-user]# sudo visudo
[root@ip-172-31-26-57 ec2-user]# su - do
[do@ip-172-31-26-57 ~]$ sudo su
[root@ip-172-31-26-57 do]# sudo vi /etc/ssh/sshd_config
[root@ip-172-31-26-57 do]# sudo vi /etc/ssh/sshd_config
[root@ip-172-31-26-57 do]# service shhd restart
Redirecting to /bin/systemctl restart shhd.service
Failed to restart shhd.service: Unit shhd.service not found.
[root@ip-172-31-26-57 do]# service sshd restart
Redirecting to /bin/systemctl restart sshd.service
[root@ip-172-31-26-57 do]#
```

i-0dd96062212edf94b (node2)

PublicIPs: 13.61.181.231 PrivateIPs: 172.31.26.57

Now again from Ansible server trying to ‘ssh <private ip of node 1>’

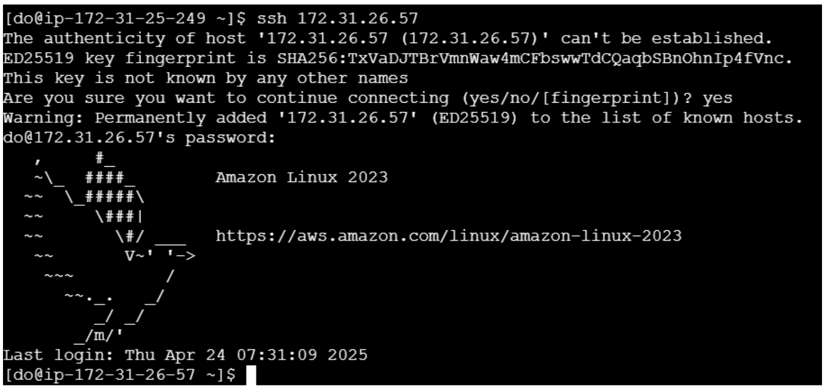


```
[do@ip-172-31-25-249 ~]$ ssh 172.31.23.82
The authenticity of host '172.31.23.82 (172.31.23.82)' can't be established.
ED25519 key fingerprint is SHA256:TkkvfCu/SArP2NRI4VzD68pTzSSosVpCqUM37qu+uZ8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.23.82' (ED25519) to the list of known hosts.
do@172.31.23.82: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[do@ip-172-31-25-249 ~]$ ssh 172.31.23.82
do@172.31.23.82's password:
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Apr 24 07:29:55 2025
[do@ip-172-31-23-82 ~]$
```

i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192 PrivateIPs: 172.31.25.249

Similarly ‘ssh <ip node 2>’



```
[do@ip-172-31-25-249 ~]$ ssh 172.31.26.57
The authenticity of host '172.31.26.57 (172.31.26.57)' can't be established.
ED25519 key fingerprint is SHA256:TxVaDJTBrVmnWaw4mCFbswwTdCQaqbSBnOhnIp4fVnc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.26.57' (ED25519) to the list of known hosts.
do@172.31.26.57's password:
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Apr 24 07:31:09 2025
[do@ip-172-31-26-57 ~]$
```

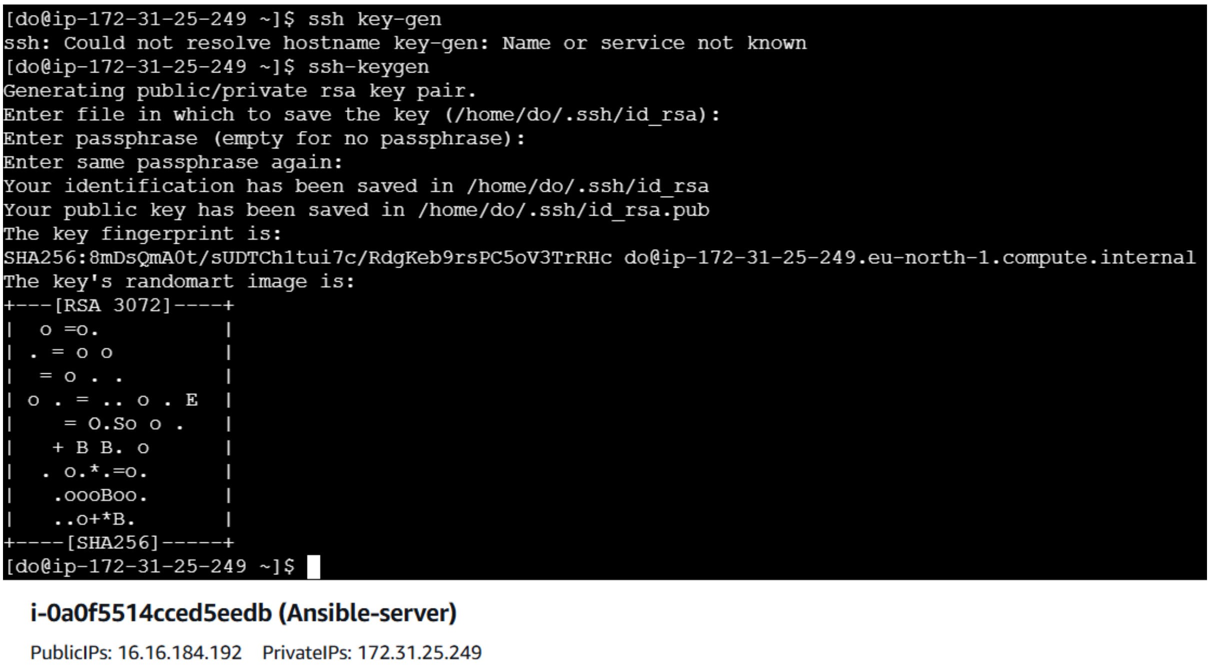
i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192 PrivateIPs: 172.31.25.249

This technique works only for 1 machine at a time, but we need parallel firing of commands for multiple machines.

Step 1: Generate SSH key pair on the main node

1.1 Use the following command to generate the SSH key on the Ansible server:
ssh-keygen



```
[do@ip-172-31-25-249 ~]$ ssh key-gen
ssh: Could not resolve hostname key-gen: Name or service not known
[do@ip-172-31-25-249 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/do/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/do/.ssh/id_rsa
Your public key has been saved in /home/do/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8mDsQmA0t/sUDTCh1tui7c/RdgKeb9rsPC5oV3TrRHc do@ip-172-31-25-249.eu-north-1.compute.internal
The key's randomart image is:
+---[RSA 3072]----+
|  o =o.          |
| . = o o         |
|  = o . .        |
| o . = .. o . E  |
|    = O.So o .   |
|   + B B. o      |
|  . o.*.=o.      |
|   .oooBoo.      |
|   ..o+*B.       |
+----[SHA256]-----+
[do@ip-172-31-25-249 ~]$
```

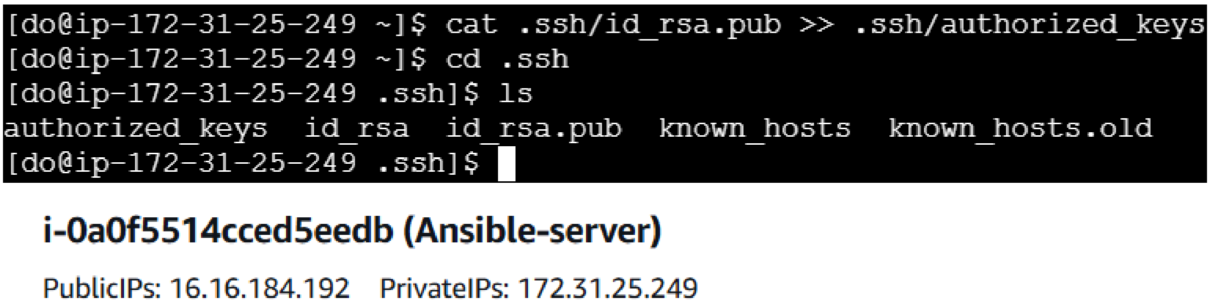
i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

Step 2: Copy the SSH key to the other two nodes

2.1 Use the following command to copy the public key to a file named **authorized_keys** in localhost:
cat .ssh/id_rsa.pub >> .ssh/authorized_keys

2.2 Run the following command to go to the **.ssh** directory of the Ansible server:
cd .ssh



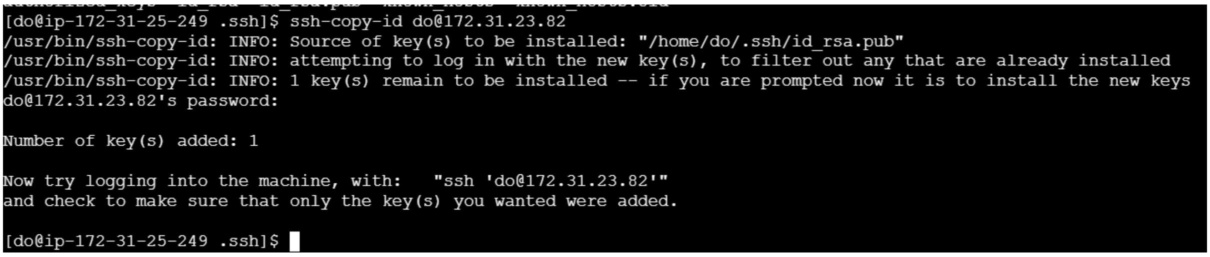
```
[do@ip-172-31-25-249 ~]$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
[do@ip-172-31-25-249 ~]$ cd .ssh
[do@ip-172-31-25-249 .ssh]$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts  known_hosts.old
[do@ip-172-31-25-249 .ssh]$
```

i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

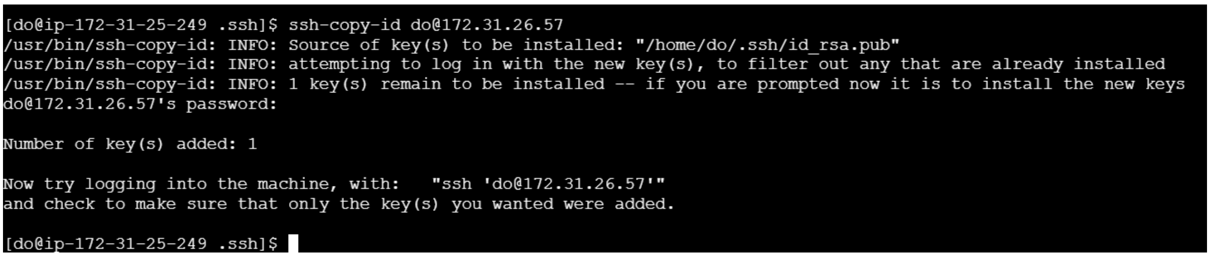
2.3 Run the following command to copy the public key to another node that will connect to the Ansible server:
ssh-copy-id username@ip -p 22

For node 1 -



```
[do@ip-172-31-25-249 .ssh]$ ssh-copy-id do@172.31.23.82
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/do/.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
do@172.31.23.82's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'do@172.31.23.82'"
and check to make sure that only the key(s) you wanted were added.

[do@ip-172-31-25-249 .ssh]$
```

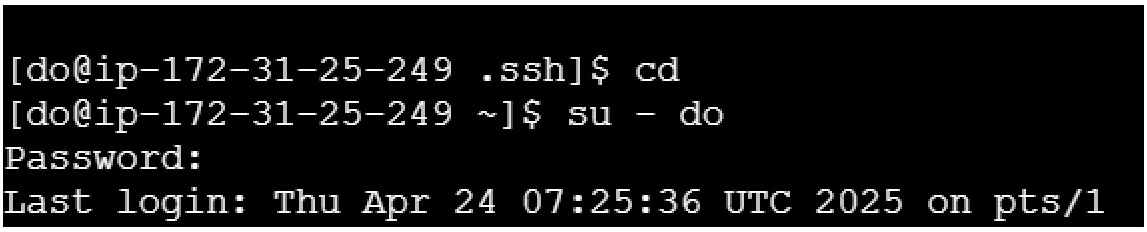
For node 2 -



```
[do@ip-172-31-25-249 .ssh]$ ssh-copy-id do@172.31.26.57
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/do/.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
do@172.31.26.57's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'do@172.31.26.57'"
and check to make sure that only the key(s) you wanted were added.

[do@ip-172-31-25-249 .ssh]$
```

Note: You must use a **username@ip** with your node and IP username, which are provided in the lab credential.

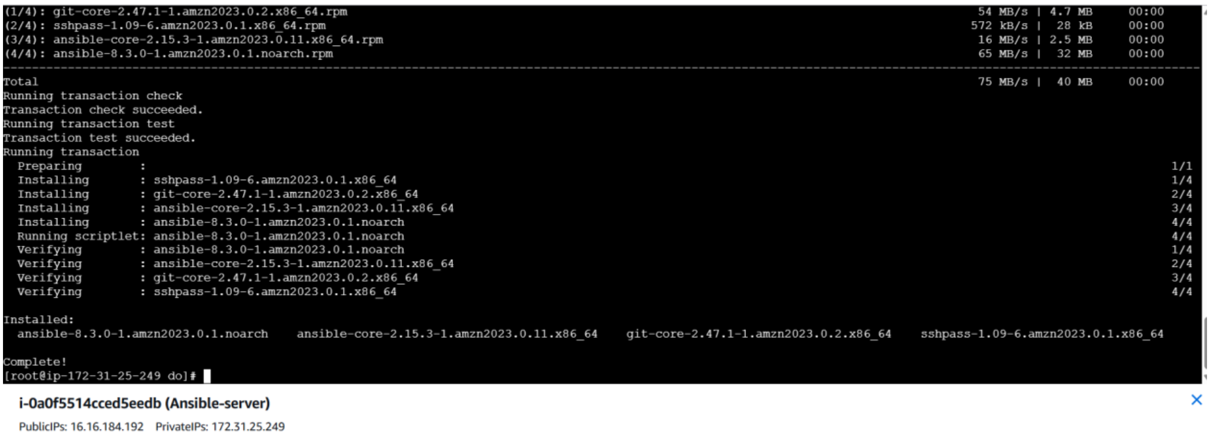
2.4 Execute the following command to exit the **.ssh** directory of the Ansible server:
cd



```
[do@ip-172-31-25-249 .ssh]$ cd
[do@ip-172-31-25-249 ~]$ su - do
Password:
Last login: Thu Apr 24 07:25:36 UTC 2025 on pts/1
```

Install Ansible on ansible server

Use commands ‘sudo yum update -y’ and ‘sudo yum install ansible -y’

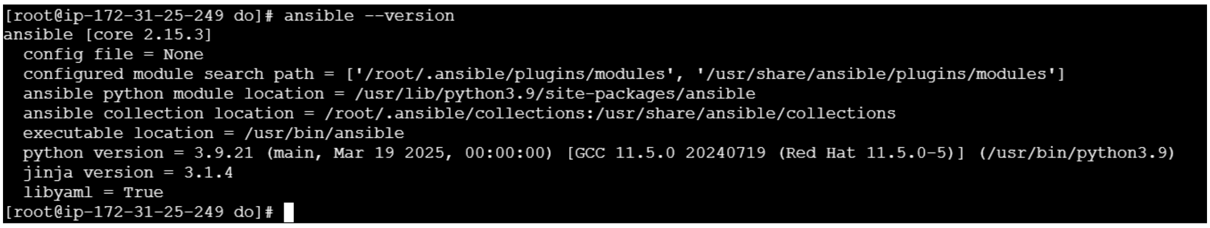


```
(1/4): git-core-2.47.1-1.amzn2023.0.2.x86_64.rpm                                                 54 MB/s | 4.7 MB     00:00
(2/4): sshpass-1.09-6.amzn2023.0.1.x86_64.rpm                                                   572 kB/s |  28 kB     00:00
(3/4): ansible-core-2.15.3-1.amzn2023.0.11.x86_64.rpm                                            16 MB/s | 2.5 MB     00:00
(4/4): ansible-8.3.0-1.amzn2023.0.1.noarch.rpm                                                   65 MB/s |  32 MB     00:00
------------------------------------------------------------------------------------------------------------------------------
Total                                                                                            75 MB/s |  40 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                                       1/1
  Installing       : sshpass-1.09-6.amzn2023.0.1.x86_64                                                                    1/4
  Installing       : git-core-2.47.1-1.amzn2023.0.2.x86_64                                                                 2/4
  Installing       : ansible-core-2.15.3-1.amzn2023.0.11.x86_64                                                            3/4
  Installing       : ansible-8.3.0-1.amzn2023.0.1.noarch                                                                   4/4
  Running scriptlet: ansible-8.3.0-1.amzn2023.0.1.noarch                                                                   4/4
  Verifying        : ansible-8.3.0-1.amzn2023.0.1.noarch                                                                   1/4
  Verifying        : ansible-core-2.15.3-1.amzn2023.0.11.x86_64                                                            2/4
  Verifying        : git-core-2.47.1-1.amzn2023.0.2.x86_64                                                                 3/4
  Verifying        : sshpass-1.09-6.amzn2023.0.1.x86_64                                                                    4/4

Installed:
  ansible-8.3.0-1.amzn2023.0.1.noarch    ansible-core-2.15.3-1.amzn2023.0.11.x86_64    git-core-2.47.1-1.amzn2023.0.2.x86_64    sshpass-1.09-6.amzn2023.0.1.x86_64

Complete!
[root@ip-172-31-25-249 do]#
```

i-0a0f5514cced5eedb (Ansible-server)

PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

To check Ansible version use *'ansible --version'*

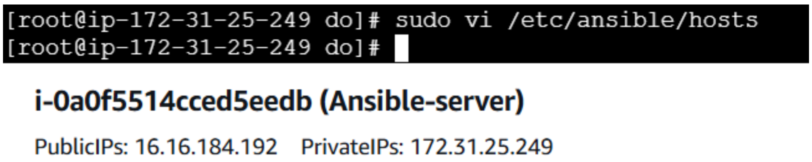


```
[root@ip-172-31-25-249 do]# ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.21 (main, Mar 19 2025, 00:00:00) [GCC 11.5.0 20240719 (Red Hat 11.5.0-5)] (/usr/bin/python3.9)
  jinja version = 3.1.4
  libyaml = True
[root@ip-172-31-25-249 do]#
```

Step 3: Update the inventory or host file with the host IP address

3.1 Use the following command to open the Ansible inventory file and add the host localhost to it:

sudo vi /etc/ansible/hosts



```
[root@ip-172-31-25-249 do]# sudo vi /etc/ansible/hosts
[root@ip-172-31-25-249 do]#
```

i-0a0f5514cced5eedb (Ansible-server)

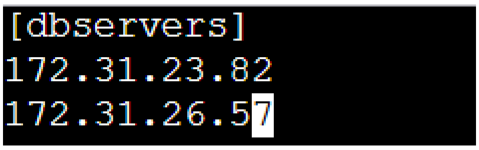
PublicIPs: 16.16.184.192   PrivateIPs: 172.31.25.249

3.2 When the file opens, add the three lines of code below to the end of the file:

[dbbservers]

localhost:22

172.31.5.76:22



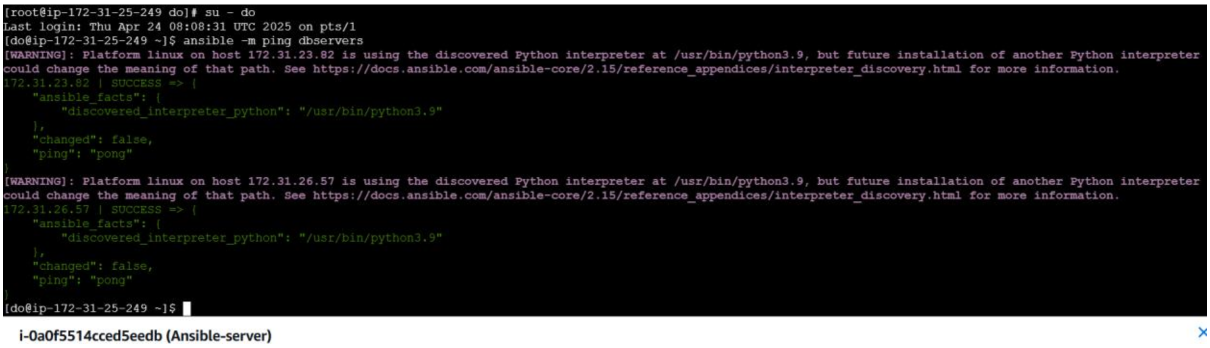
```
[dbservers]
172.31.23.82
172.31.26.57
```

Note: Press **esc**, then write **:wq** and press **enter** to save the file.

Step 4: Establish connectivity between the hosts specified in the host file and
the Ansible server

4.1 Run the following command to copy the public key to another node that will connect to the Ansible server:

ansible -m ping dbbservers

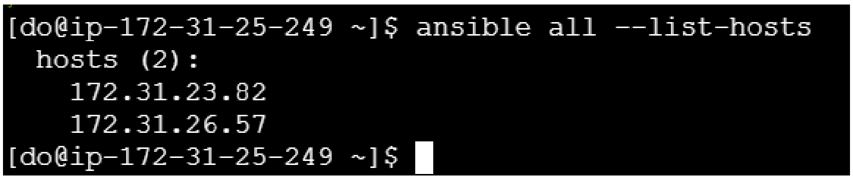


```
[root@ip-172-31-25-249 do]# su - do
Last login: Thu Apr 24 08:08:31 UTC 2025 on pts/1
[do@ip-172-31-25-249 ~]$ ansible -m ping dbservers
[WARNING]: Platform linux on host 172.31.23.82 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter
could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
172.31.23.82 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.26.57 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.
172.31.26.57 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
[do@ip-172-31-25-249 ~]$
```

i-0a0f5514cced5eedb (Ansible-server)

4.2 Use the following command to check the number of hosts in the host file:

ansible all --list-hosts



```
[do@ip-172-31-25-249 ~]$ ansible all --list-hosts
  hosts (2):
    172.31.23.82
    172.31.26.57
[do@ip-172-31-25-249 ~]$
```

i-0a0f5514cced5eedb (Ansible-server)

By following these steps, you have successfully created a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible.