Lab Exercise 01
Creating Static Host Inventory

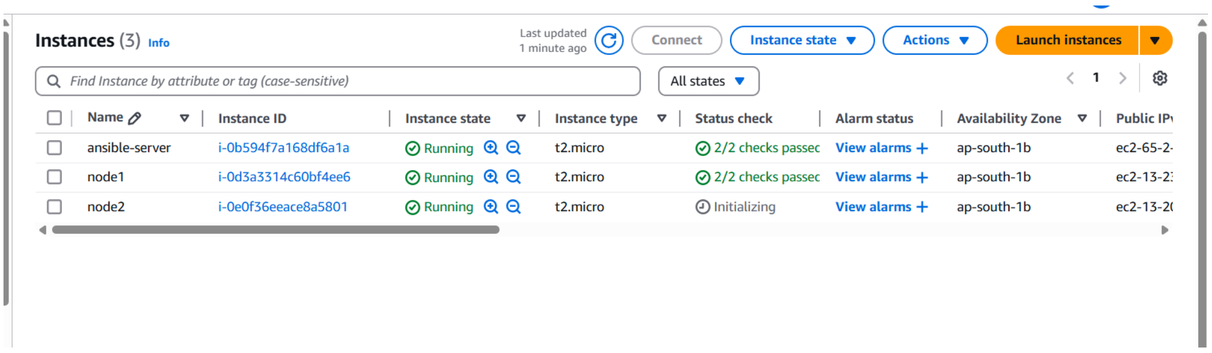
Objective: To create a static host inventory for managing and automating infrastructure tasks efficiently across multiple servers using Ansible

Tools required: Ubuntu OS

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

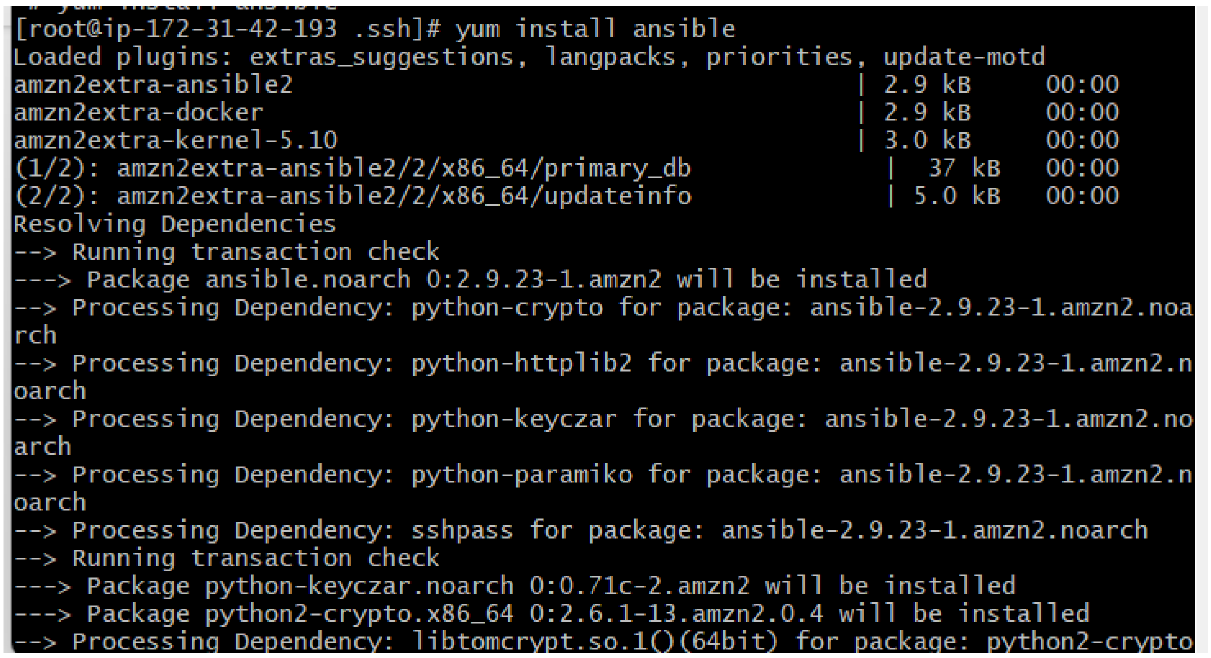
Steps to be followed:
1. Generate SSH key pair on the main node
2. Copy the SSH key to the two other nodes
3. Update the inventory or host file with the host IP address
4. Establish connectivity between the hosts specified in the host file and the Ansible server

Step 1:Launch 3 instance of type t2.micro


Instances (3) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
ansible-server	i-0b594f7a168df6a1a	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-65-2-
node1	i-0d3a3314c60bf4ee6	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-2
node2	i-0e0f36eeace8a5801	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-2

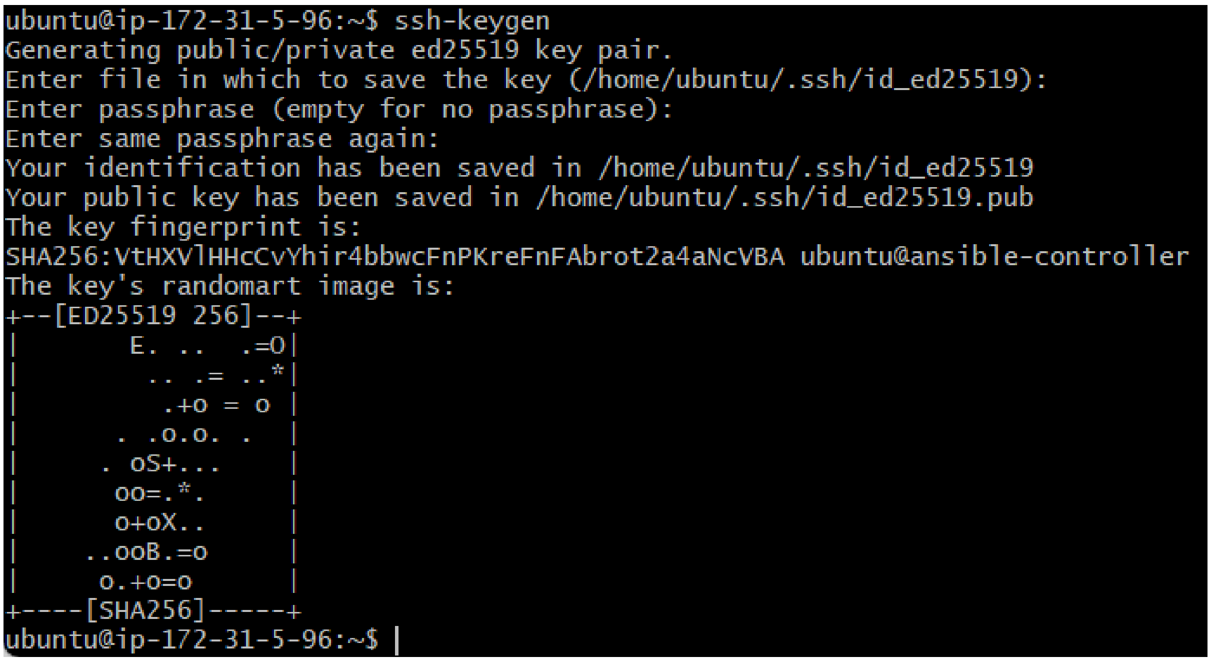
Install ansible on ansible server



```
[root@ip-172-31-42-193 .ssh]# yum install ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2extra-ansible2                                      | 2.9 kB     00:00
amzn2extra-docker                                        | 2.9 kB     00:00
amzn2extra-kernel-5.10                                   | 3.0 kB     00:00
(1/2): amzn2extra-ansible2/2/x86_64/primary_db             |  37 kB   00:00
(2/2): amzn2extra-ansible2/2/x86_64/updateinfo             | 5.0 kB   00:00
Resolving Dependencies
--> Running transaction check
---> Package ansible.noarch 0:2.9.23-1.amzn2 will be installed
--> Processing Dependency: python-crypto for package: ansible-2.9.23-1.amzn2.noa
rch
--> Processing Dependency: python-httplib2 for package: ansible-2.9.23-1.amzn2.n
oarch
--> Processing Dependency: python-keyczar for package: ansible-2.9.23-1.amzn2.no
arch
--> Processing Dependency: python-paramiko for package: ansible-2.9.23-1.amzn2.n
oarch
--> Processing Dependency: sshpass for package: ansible-2.9.23-1.amzn2.noarch
--> Running transaction check
---> Package python-keyczar.noarch 0:0.71c-2.amzn2 will be installed
---> Package python2-crypto.x86_64 0:2.6.1-13.amzn2.0.4 will be installed
--> Processing Dependency: libtomcrypt.so.1()(64bit) for package: python2-crypto
```

Step 2: 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

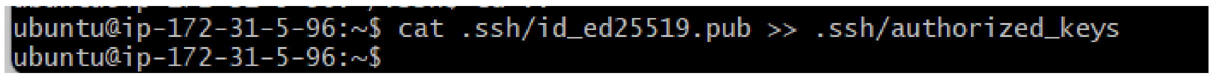


```
ubuntu@ip-172-31-5-96:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:VtHXVlHHcCvYhir4bbwcFnPKreFnFAbrot2a4aNcVBA ubuntu@ansible-controller
The key's randomart image is:
+--[ED25519 256]--+
|       E. ..  .=O|
|        .. .= ..*|
|         .+o = o |
|      . .o.o. .  |
|     . oS+...    |
|      oo=.*.     |
|      o+oX..     |
|    ..ooB.=o     |
|     o.+o=o      |
+----[SHA256]-----+
ubuntu@ip-172-31-5-96:~$ |
```

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:

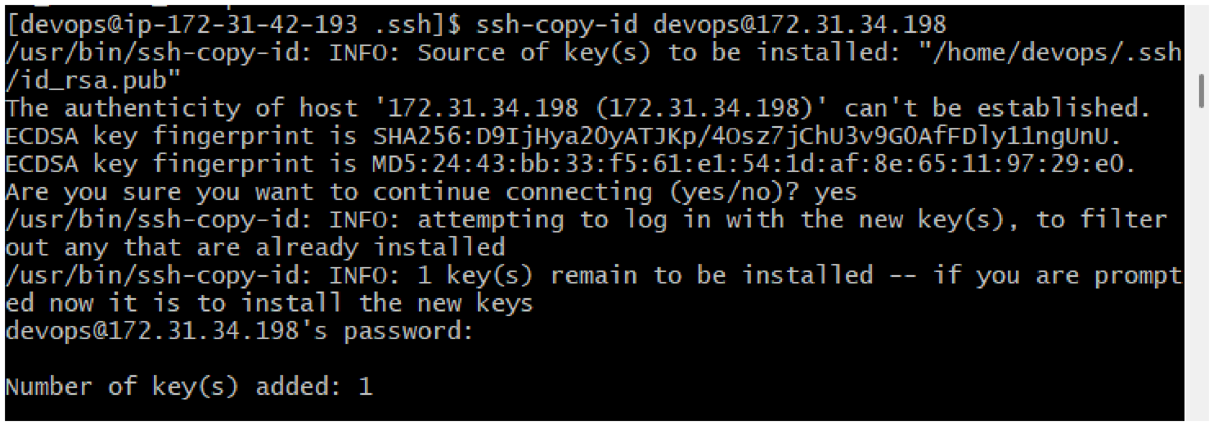
cat .ssh/id_rsa.pub >> .ssh/authorized_keys



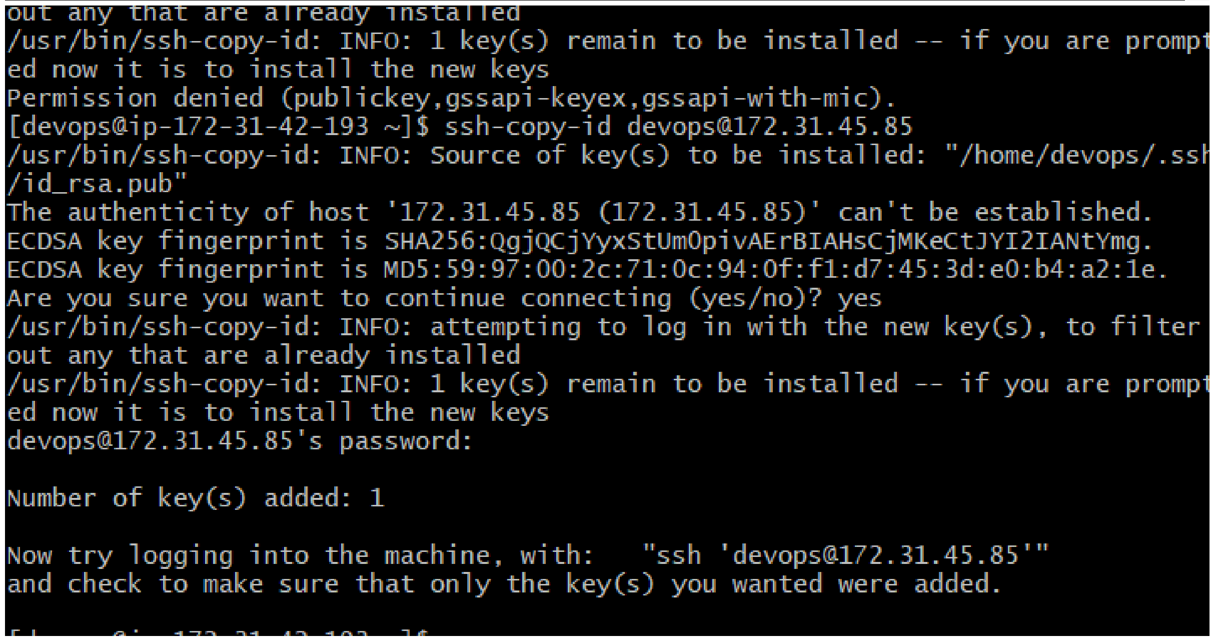
```
ubuntu@ip-172-31-5-96:~$ cat .ssh/id_ed25519.pub >> .ssh/authorized_keys
ubuntu@ip-172-31-5-96:~$
```

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

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



```
[devops@ip-172-31-42-193 .ssh]$ ssh-copy-id devops@172.31.34.198
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh
/id_rsa.pub"
The authenticity of host '172.31.34.198 (172.31.34.198)' can't be established.
ECDSA key fingerprint is SHA256:D9IjHya2OyATJKp/4Osz7jChU3v9G0AfFDly11ngUnU.
ECDSA key fingerprint is MD5:24:43:bb:33:f5:61:e1:54:1d:af:8e:65:11:97:29:e0.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
devops@172.31.34.198's password:

Number of key(s) added: 1
```

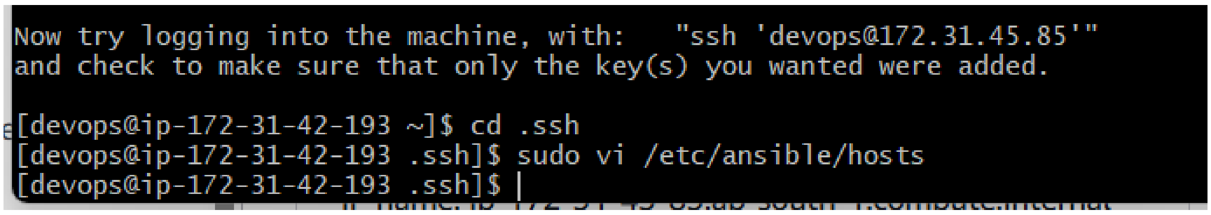


```
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[devops@ip-172-31-42-193 ~]$ ssh-copy-id devops@172.31.45.85
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh
/id_rsa.pub"
The authenticity of host '172.31.45.85 (172.31.45.85)' can't be established.
ECDSA key fingerprint is SHA256:QgjQCjYyxStUmOpivAErBIAHsCjMKeCtJYI2IANtYmg.
ECDSA key fingerprint is MD5:59:97:00:2c:71:0c:94:0f:f1:d7:45:3d:e0:b4:a2:1e.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
devops@172.31.45.85's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'devops@172.31.45.85'"
and check to make sure that only the key(s) you wanted were added.
```

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

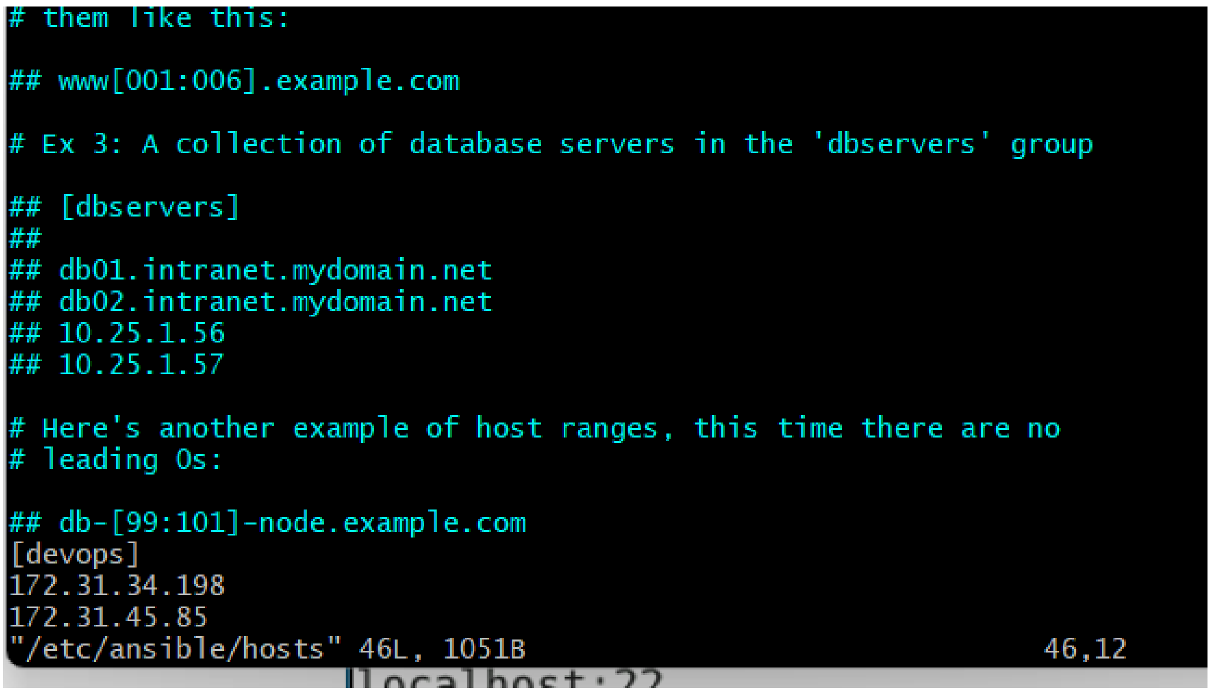
Step 3: Update the inventory or host file with the host IP address

3.1 Use the following command to open the Ansible inventory file and add the host localhost to it:
sudo vi /etc/ansible/hosts



```
Now try logging into the machine, with:   "ssh 'devops@172.31.45.85'"
and check to make sure that only the key(s) you wanted were added.

[devops@ip-172-31-42-193 ~]$ cd .ssh
[devops@ip-172-31-42-193 .ssh]$ sudo vi /etc/ansible/hosts
[devops@ip-172-31-42-193 .ssh]$ |
```

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



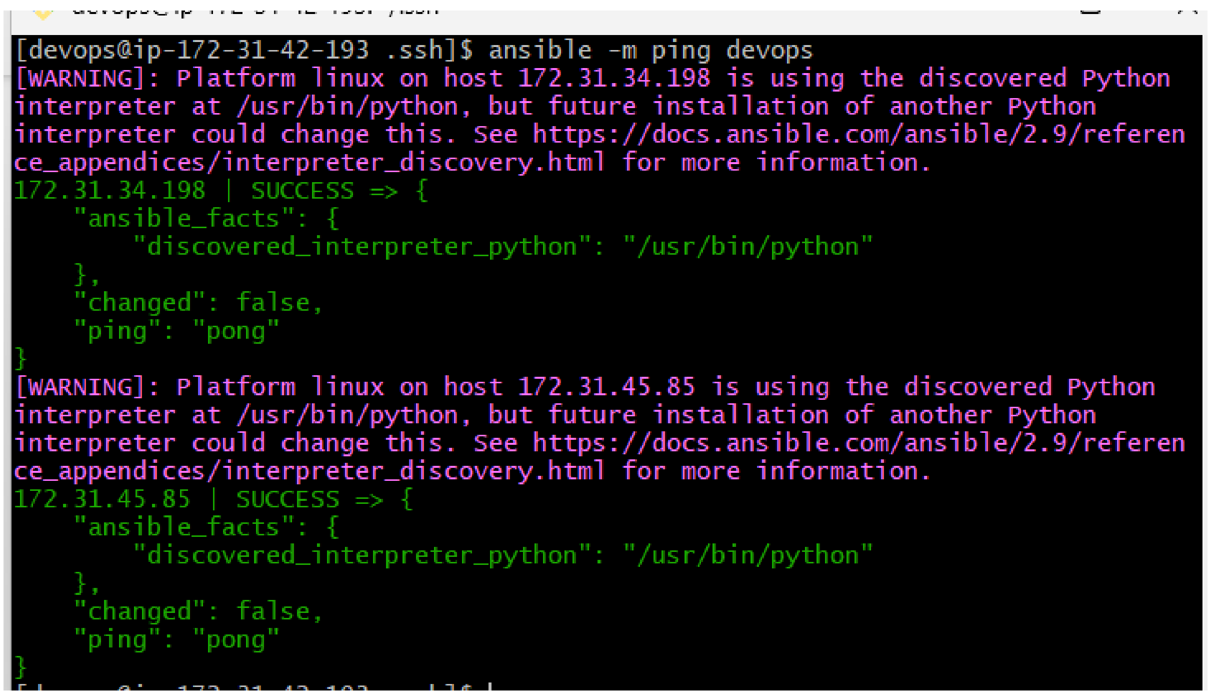
```
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[devops]
172.31.34.198
172.31.45.85
"/etc/ansible/hosts" 46L, 1051B                                  46,12
```

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

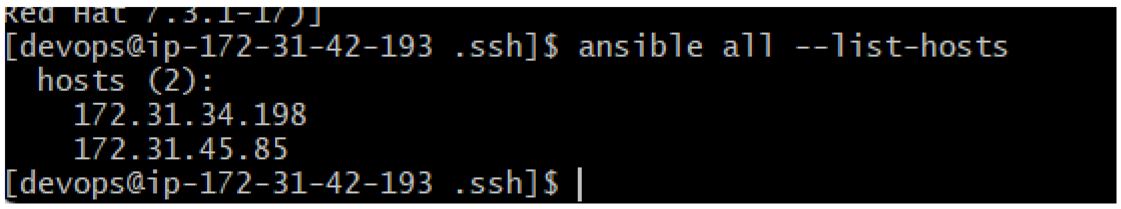
4.1 Run the following command to copy the public key to another node that will connect to the Ansible server:

ansible -m ping dbbservers



```
[devops@ip-172-31-42-193 .ssh]$ ansible -m ping devops
[WARNING]: Platform linux on host 172.31.34.198 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.
172.31.34.198 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.45.85 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.
172.31.45.85 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

4.2 Use the following command to check the number of hosts in the host file:
ansible all --list-hosts



```
[devops@ip-172-31-42-193 .ssh]$ ansible all --list-hosts
  hosts (2):
    172.31.34.198
    172.31.45.85
[devops@ip-172-31-42-193 .ssh]$ |
```

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