ANSIBLE INSTALLATION

DevOps Certification Training

support@intellipaat.com
+91-7022374614
US: 1-800-216-8930(Toll Free)

ANSIBLE INSTALLATION ON UBUNTU

Ansible installation documentation has been divided into 3 segments.

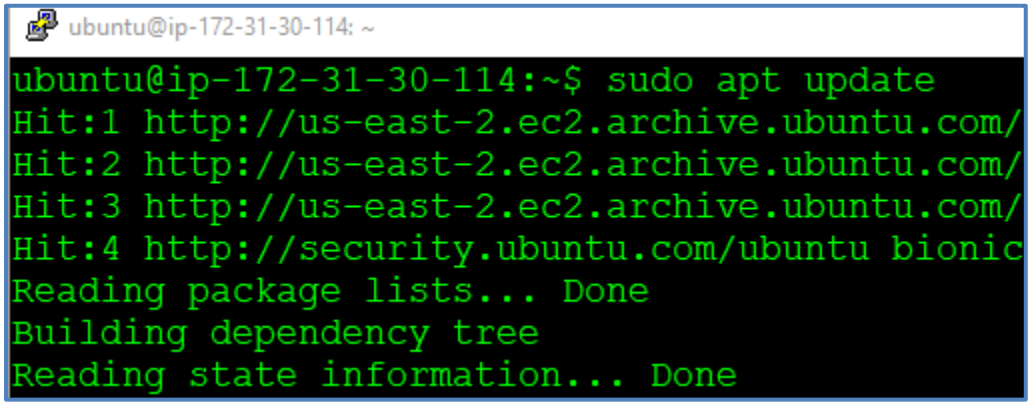
A-Install Ansible
B-Configure the SSH Access to the Ansible Hosts
C-Setting Up Ansible Hosts

The terminal with green color commands represents terminal and the one with white color commands represents slave terminal.

A-Install Ansible

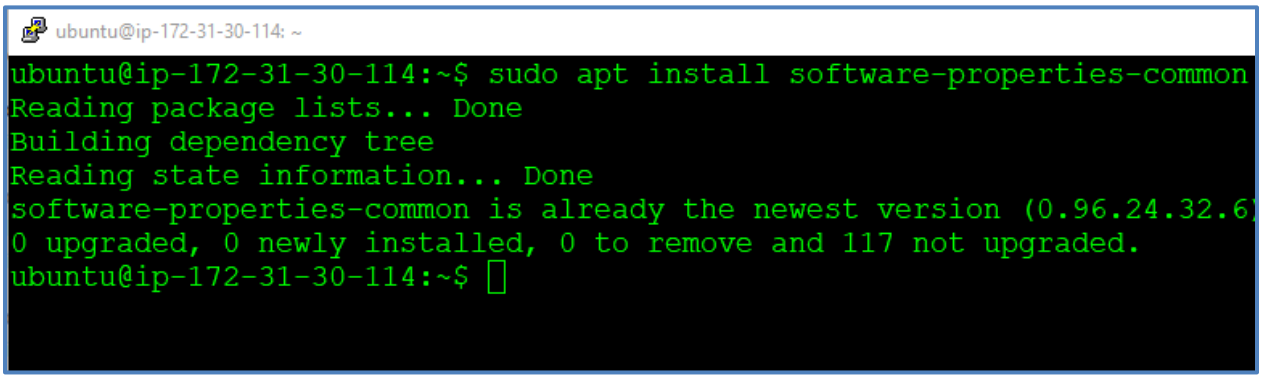
Step 1: Update package index.

```
$ sudo apt update
```



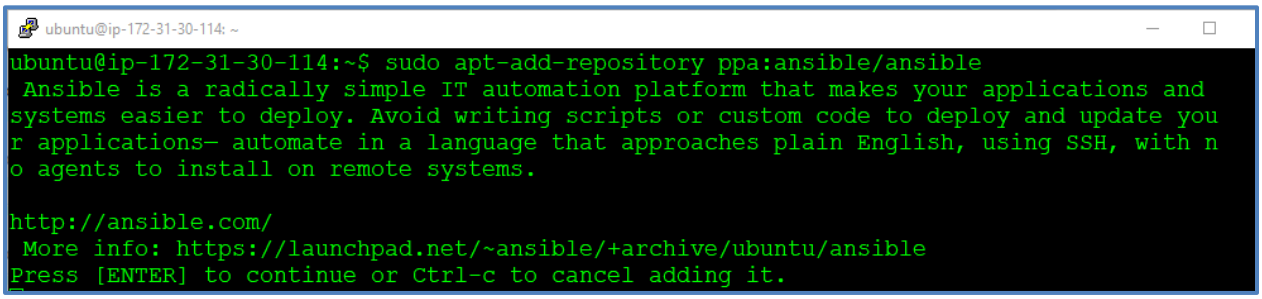
Step 2: Install the *software-properties-common* package.

```
$ sudo apt install software-properties-common
```



Step 3: Add the Ansible PPA (personal package archive)

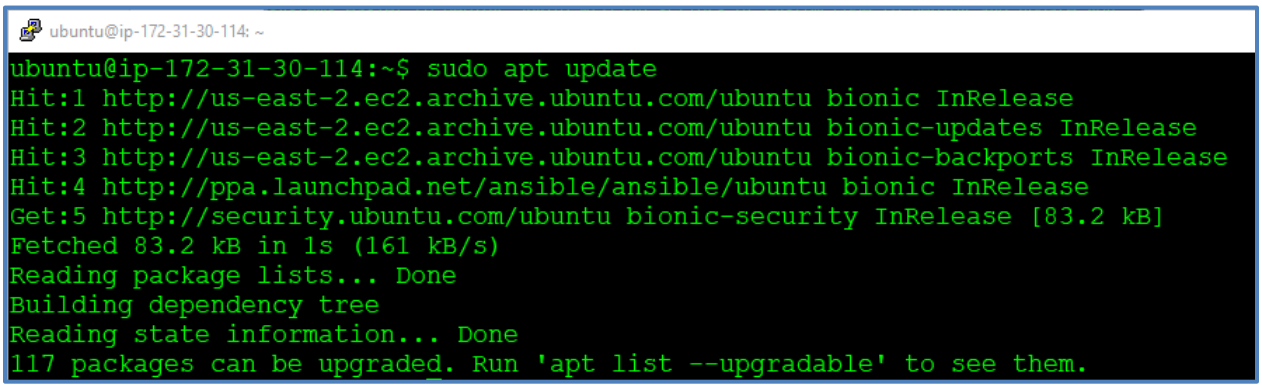
```
$ sudo apt-add-repository ppa:ansible/ansible
```



Press Enter to accept the PPA addition.

Step 4: Now, refresh the system's package index again to make it aware of the packages available in the PPA by using the following command.

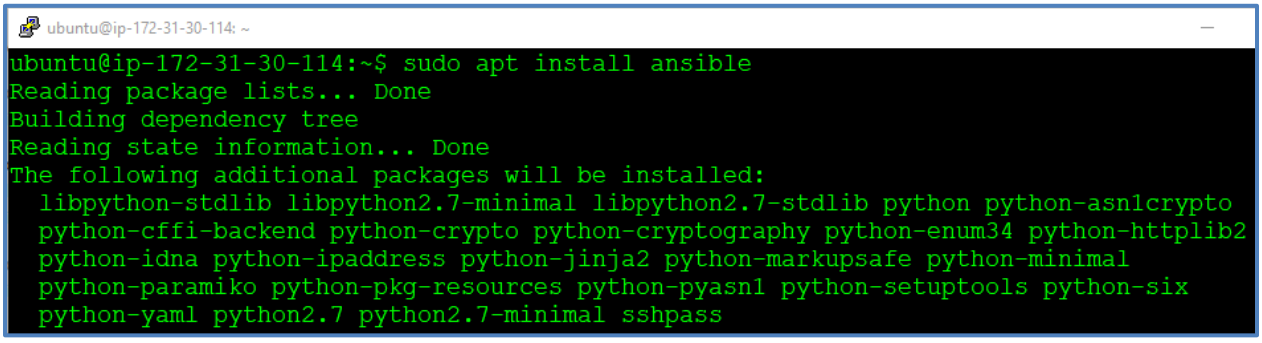
```
$ sudo apt update
```



Now, we are set to install Ansible software.

Step 5: Install Ansible software.

```
$ sudo apt install ansible
```

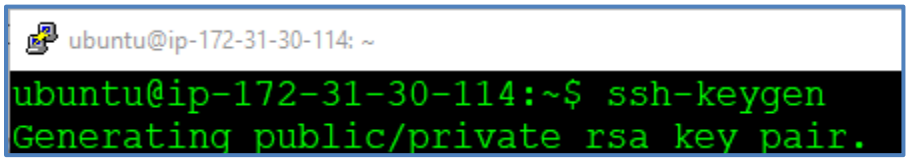


Now, your Ansible server now has all the software required to administer your hosts.

B-Configure the SSH Access to the Ansible Hosts

Step 6: Create a key pair on the master terminal.

```
$ ssh-keygen
```



Press Enter, you will get the following output. You can optionally enter passphrase to prevent unauthorized users from logging in.

```
ubuntu@ip-172-31-30-114: ~
ubuntu@ip-172-31-30-114:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:GqNgcTwUqOAbsWdTsWqUKbq1pFhfrNJ8+g4EDjkjPFg ubuntu@ip-172-31-30-114
The key's randomart image is:
+---[RSA 2048]----+
|  E.oo.          |
|++.oo..          |
|X==+=.           |
|+O+B.o           |
|. %oo = S        |
|.O.B + +         |
|+ o B o          |
|   . +           |
|    .oo          |
+----[SHA256]-----+
```

Now you have a public and private key. You can use them to authenticate from the slave terminal.

Step 7: On the master's terminal use the **cat command.**

```
$ cat ~/.ssh/id_rsa.pub
```

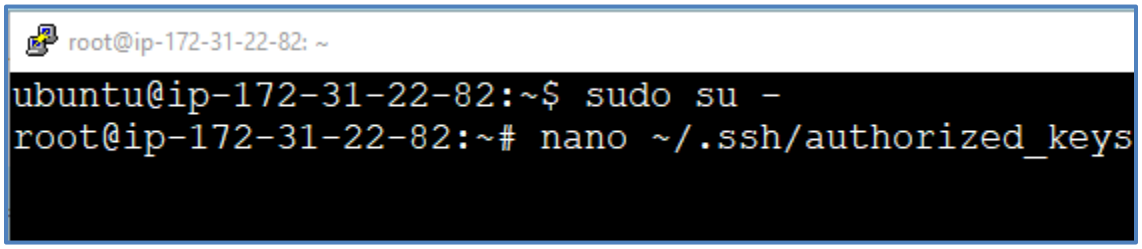
```
ubuntu@ip-172-31-30-114: ~
ubuntu@ip-172-31-30-114:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCzHIg6vOmzPjzLhtenm4NjGV85VGoCK45tCSJGI
+zs91epwcxMVpP54fl1uMSyN8zi3+fUj9IoT4gnhZcD2AWqWbqwIk1JFYvhA9LBcE0UGoFHdptj+w
10AJqN72KrhvQGDPnFuhUGLdw/Aq/FVu5akFGt+kmcu1Z2ImJn7W+zqg4LRkG3FaDarc7JDVwccX7
PPADIUYLZiNViHTxovIshv5IzqSkKSqYSX2gySTp1pqB3FpJgFlXplp/EG9rJYbopwpBnMJRO7Ptx
8z5faUTLh4WGGR1t ubuntu@ip-172-31-30-114
ubuntu@ip-172-31-30-114:~$ ^C
ubuntu@ip-172-31-30-114:~$ ^C
ubuntu@ip-172-31-30-114:~$ ssh root@3.16.255.89
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1021-aws x86_64)
```

Copy the output in clipboard. Now we will open the ***authorized_keys*** within the **~/.ssh directory** and paste the copied output there.

Step 8: Go to the client machines's root user and open the authorized_keys within the **~/.ssh** directory.

```
$ sudo su -
```

```
#nano ~/.ssh/authorized_keys
```

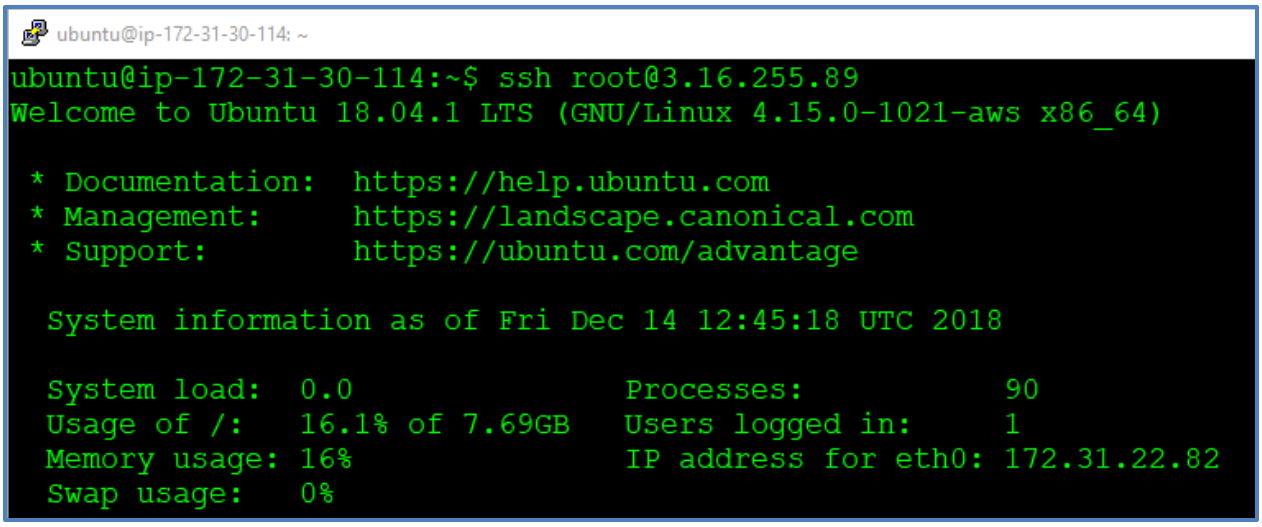


Paste Enter and once you are inside *Authorized_keys*, paste the copied Ansible server user's SSH key in the directory. Then Save and Exit by Pressing press CTRL + X, CTRL+ Y, then again press ENTER.

C-Setting Up Ansible Hosts

Step 9: Now, check the connection, by entering the following command.

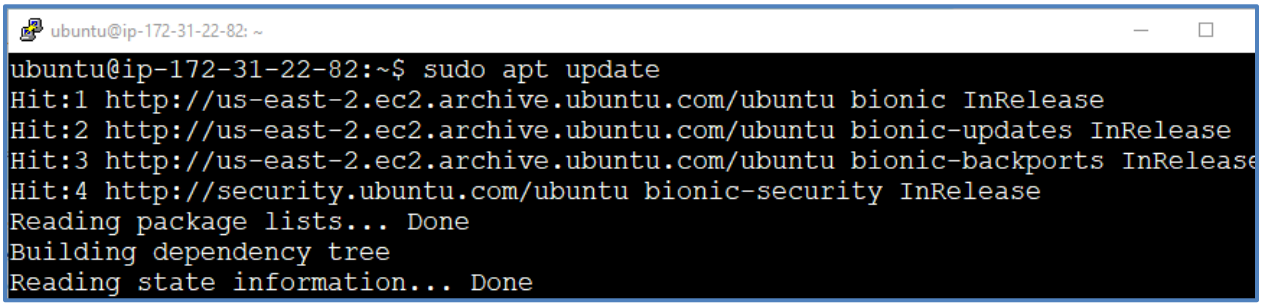
```
$ ssh root@ansible_host_ip
```



Now, before setting up the Ansible host we will install python on the host so that Ansible can communicate with it.

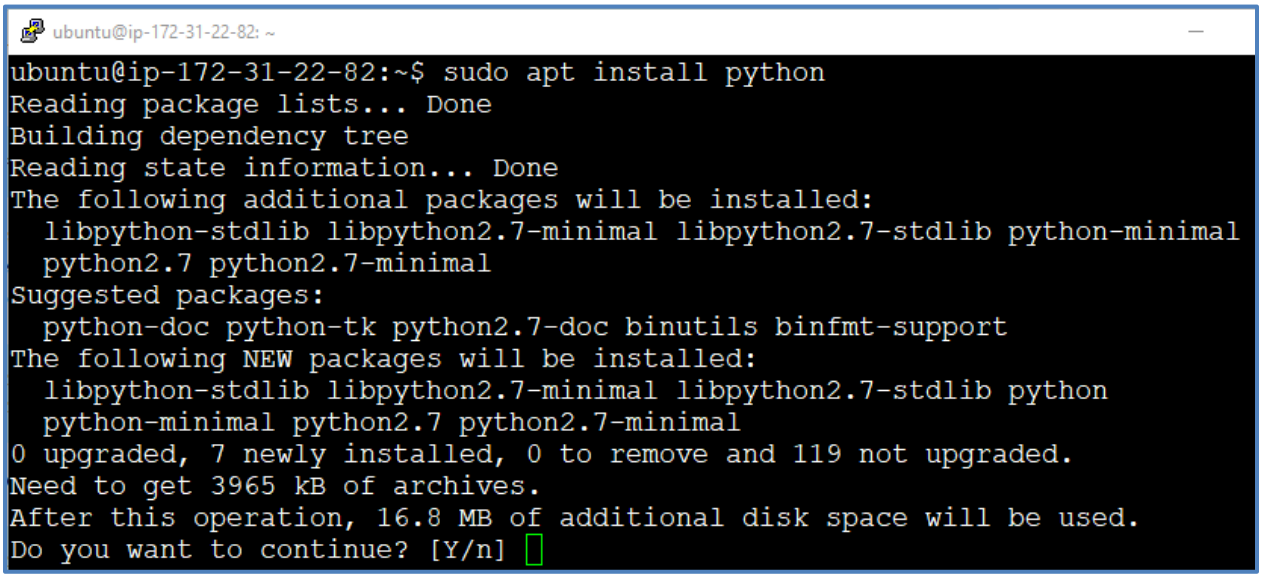
Step 10: First, update the host's package.

```
$ sudo apt update
```



Step 11: Install Python in host's machine.

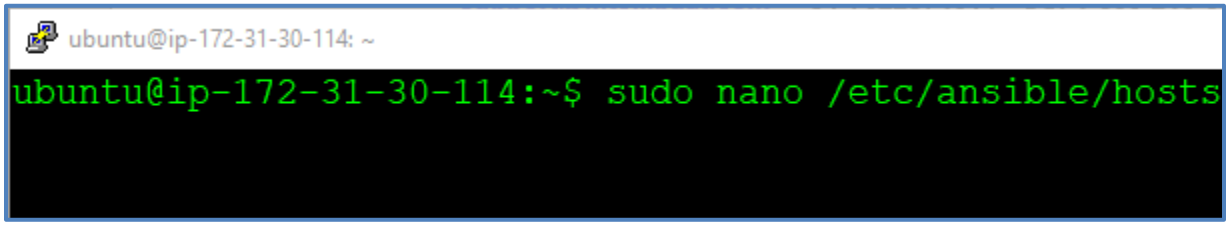
```
$ sudo apt install python
```



Press Y and finish the installation. Then Press Exit.

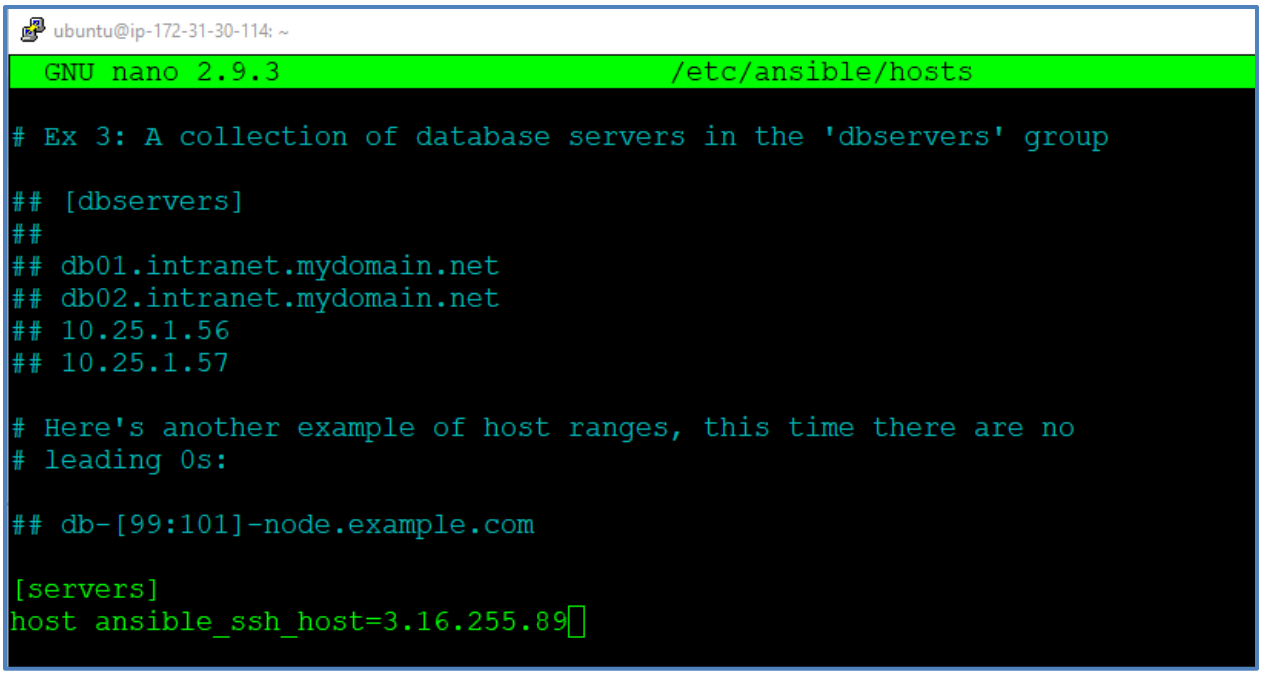
Step 12: In Ansible Server, run the following command.

```
$ sudo nano /etc/ansible/hosts
```



Step 13: In Ansible Server, run the following command. Add the configure the file by adding the block of commands, that consists of the host IP information, as shown below.

```
$ sudo nano /etc/ansible/hosts
```



For more than one hosts following commands can be used.

```
[servers]
host1 ansible_ssh_host= host1 IP
host2 ansible_ssh_host= host2 IP
host3 ansible_ssh_host= host2 IP
```

After that to exit from ansible/hosts Press CTRL+X, CTRL+Y and then press Enter.

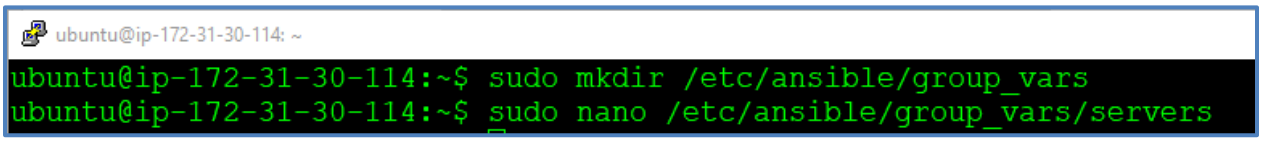
Now we will create a file that tells all the servers in the "servers" group to connect as the root user.

To do this, we will create a directory in the Ansible configuration structure named group_vars. In that folder, we will create YAML-formatted files for the group we want to configure.

Step 12: In Ansible Server, run the following command.

```
$ sudo mkdir /etc/ansible/group_vars
```

```
$ sudo nano /etc/ansible/group_vars/servers
```



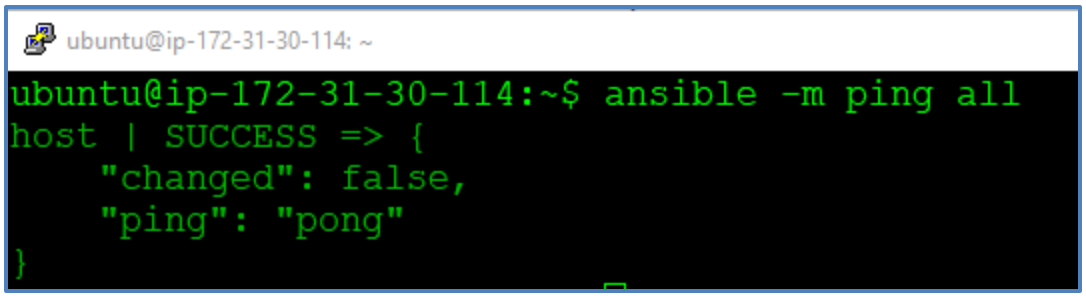
put our configuration in there as shown below.

Press CTRL+X, CTRL+ Y and then press Enter.

C-Testing the connection

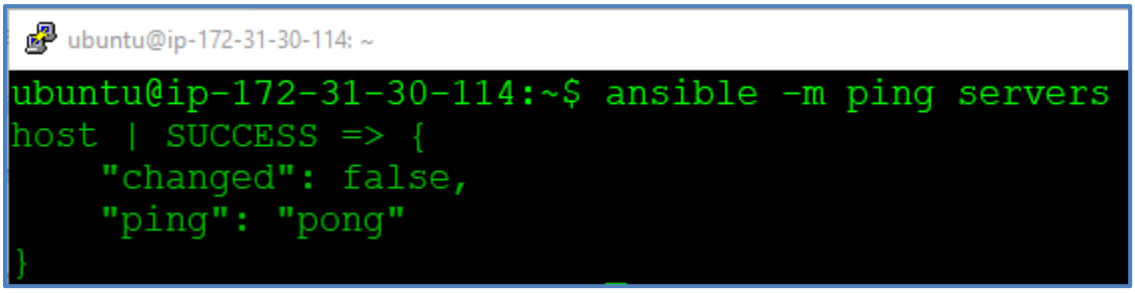
Step 12: In Ansible Server, run the following command.

```
$ ansible -m ping all
```



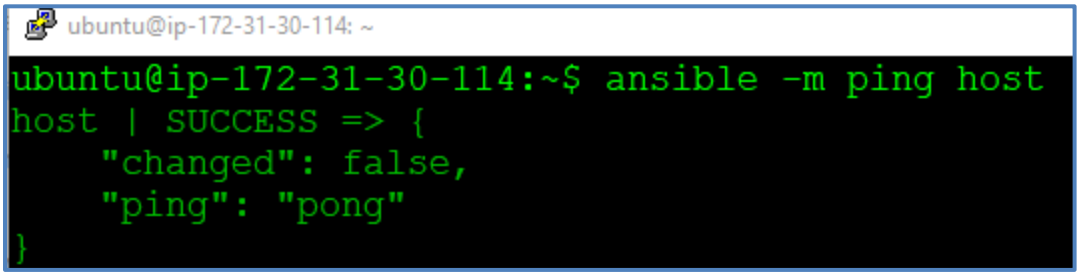
The same can also be achieved by using the following commands

```
$ ansible -m ping servers
```



We could also specify an individual host.

```
$ ansible -m ping host
```



Congratulations! You have successfully setup Ansible Master-Slave Cluster