



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

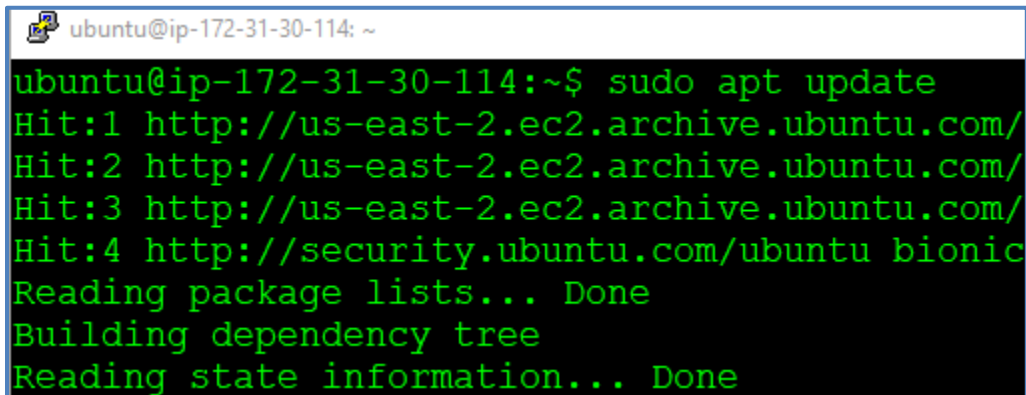
C-Setting Up Ansible Hosts

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.

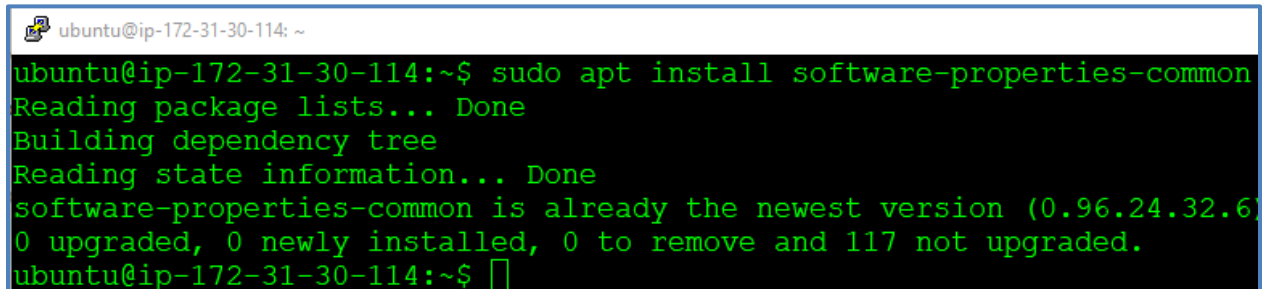
```
$ sudo apt update
```



```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo apt update  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/  
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/  
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/  
Hit:4 http://security.ubuntu.com/ubuntu bionic  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

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

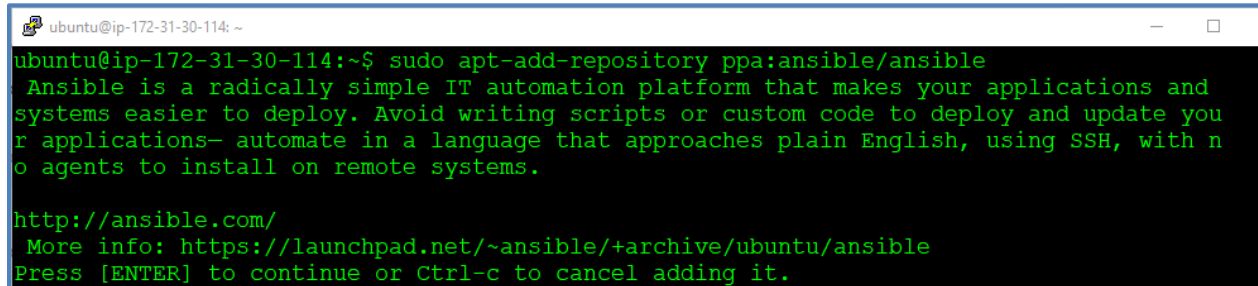
```
$ sudo apt install software-properties-common
```



```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo apt install software-properties-common  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
software-properties-common is already the newest version (0.96.24.32.6)  
0 upgraded, 0 newly installed, 0 to remove and 117 not upgraded.  
ubuntu@ip-172-31-30-114:~$
```

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

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

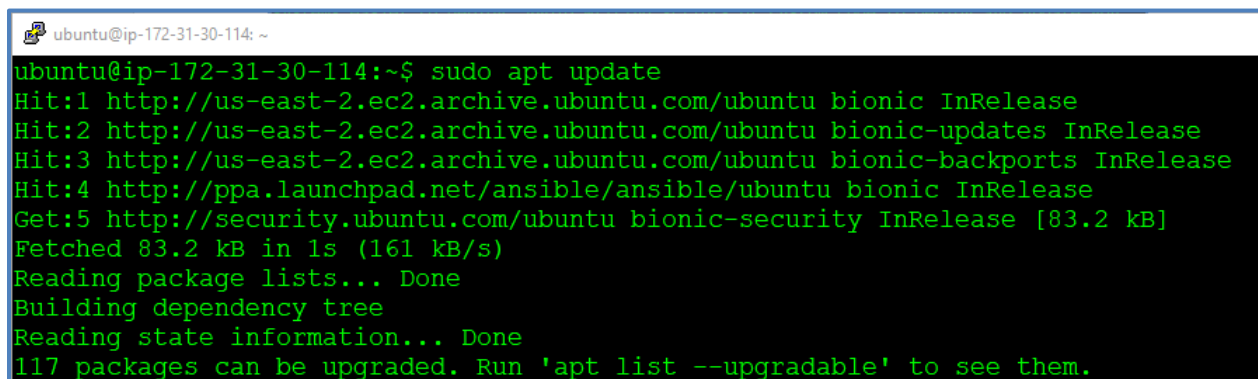


```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo apt-add-repository ppa:ansible/ansible  
Ansible is a radically simple IT automation platform that makes your applications and  
systems easier to deploy. Avoid writing scripts or custom code to deploy and update you  
r applications- automate in a language that approaches plain English, using SSH, with n  
o agents to install on remote systems.  
  
http://ansible.com/  
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible  
Press [ENTER] to continue or Ctrl-c to cancel adding it.
```

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
```

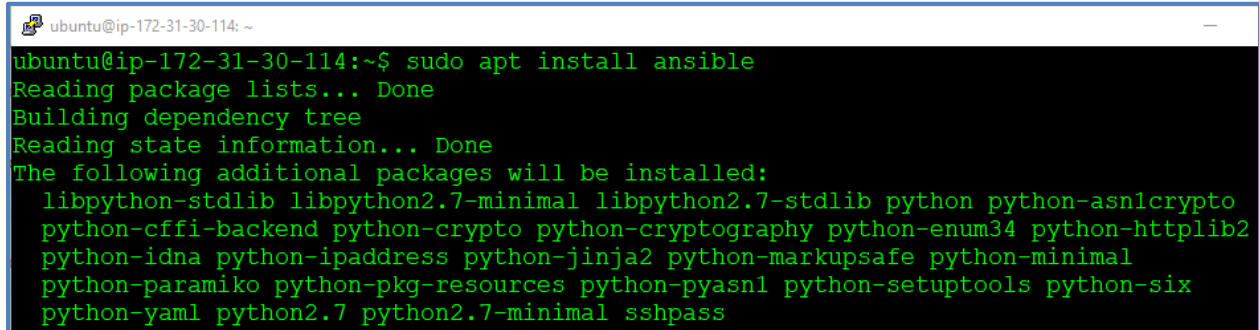


```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo apt update  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:4 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease  
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]  
Fetched 83.2 kB in 1s (161 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
117 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Now, we are set to install Ansible software.

Step 5: Install Ansible software.

```
$ sudo apt install ansible
```



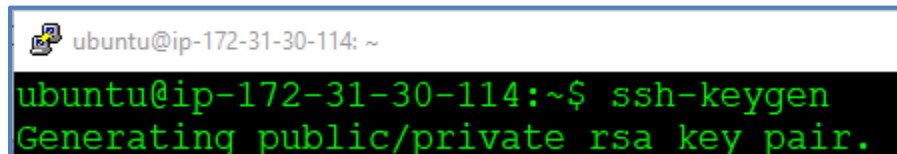
```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo apt install ansible  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python python-asn1crypto  
  python-cffi-backend python-crypto python-cryptography python-enum34 python-httplib2  
  python-idna python-ipaddress python-jinja2 python-markupsafe python-minimal  
  python-paramiko python-pkg-resources python-pyasnl python-setuptools python-six  
  python-yaml python2.7 python2.7-minimal sshpass
```

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
```



```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ ssh-keygen  
Generating public/private rsa key pair.
```

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:GqNgcTwUqOAbsWdTswqUKbqlpFhfrNJ8+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 AAAAB3NzaC1yc2EAAAADAQABAAQACzHIg6vOmzPjzLhtenm4NjGV85VGoCK45tCSJGI
+zs91epwcxMVpP54f1luMSyN8zi3+fUj9IoT4gnhZcD2AWqWbqwIk1JFYvhA9LBCE0UGoFHdptj+w
10AJqN72KrhvQGDPnFuhUGLdw/Aq/FVu5akFGt+kmcu1Z2ImJn7W+zqg4LRkG3FaDarc7JDVwccX7
PPADIUYLZiNviHTxovIshv5IzqSkKSqYSX2gySTplpqB3FpJgFlXplp/EG9rJYbopwpBnMJRO7Ptx
8z5faUTLh4WGGRI t 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)

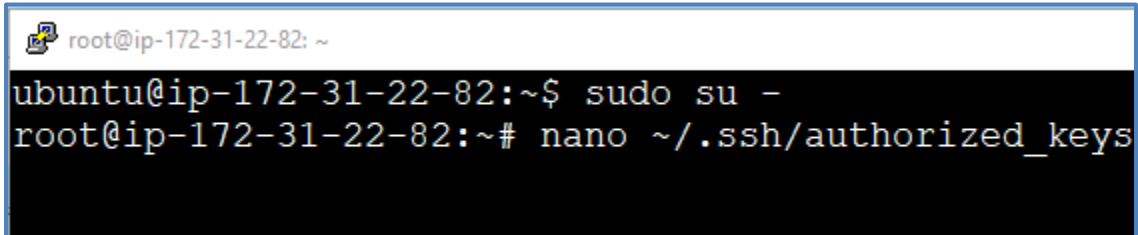
```

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
```



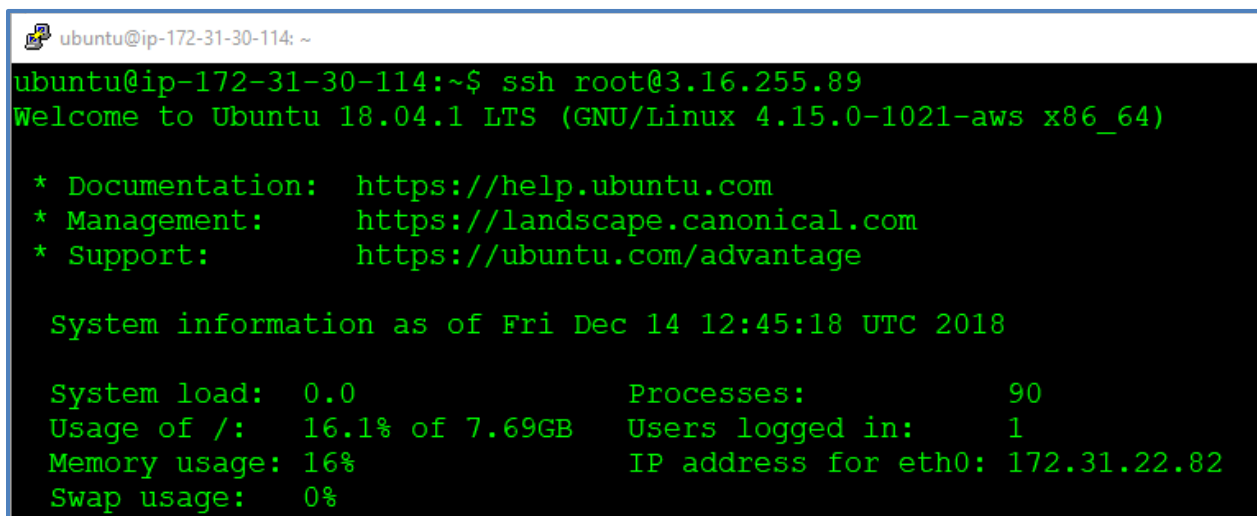
```
root@ip-172-31-22-82: ~  
ubuntu@ip-172-31-22-82:~$ sudo su -  
root@ip-172-31-22-82:~# nano ~/.ssh/authorized_keys
```

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
```



```
ubuntu@ip-172-31-30-114: ~  
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)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Fri Dec 14 12:45:18 UTC 2018  
  
System load:  0.0                Processes:            90  
Usage of /:   16.1% of 7.69GB     Users logged in:     1  
Memory usage: 16%                IP address for eth0: 172.31.22.82  
Swap usage:   0%
```

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
```

```
ubuntu@ip-172-31-22-82: ~  
ubuntu@ip-172-31-22-82:~$ sudo apt update  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

Step 11: Install Python in host's machine.

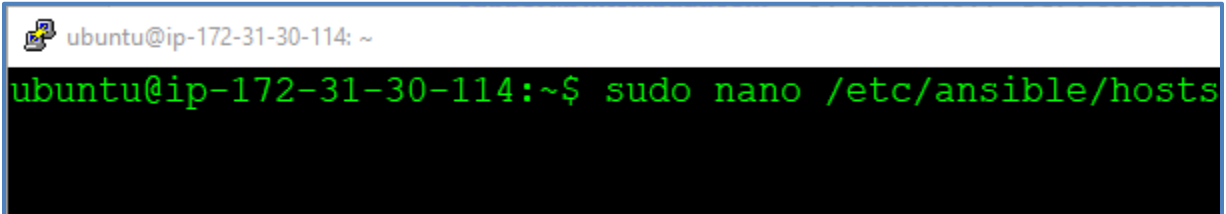
```
$ sudo apt install python
```

```
ubuntu@ip-172-31-22-82: ~  
ubuntu@ip-172-31-22-82:~$ sudo apt install python  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python-minimal  
  python2.7 python2.7-minimal  
Suggested packages:  
  python-doc python-tk python2.7-doc binutils binfmt-support  
The following NEW packages will be installed:  
  libpython-stdlib libpython2.7-minimal libpython2.7-stdlib python  
  python-minimal python2.7 python2.7-minimal  
0 upgraded, 7 newly installed, 0 to remove and 119 not upgraded.  
Need to get 3965 kB of archives.  
After this operation, 16.8 MB of additional disk space will be used.  
Do you want to continue? [Y/n] ☐
```

Press Y and finish the installation. Then Press **Exit**.

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

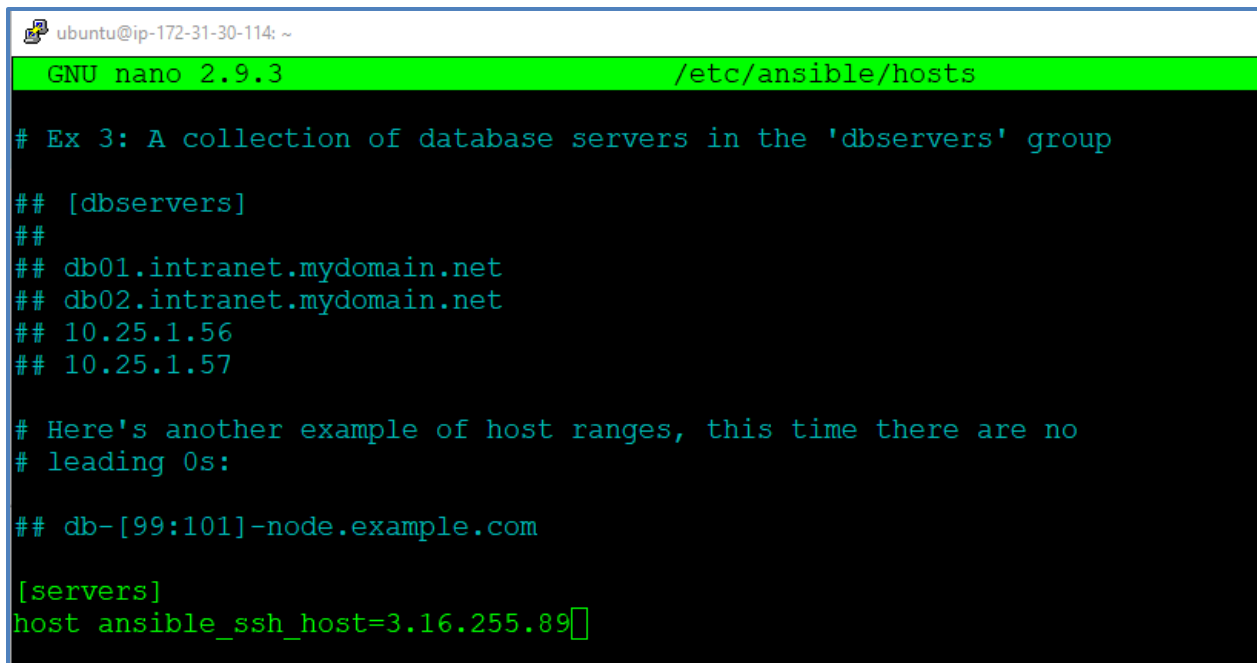
```
$ sudo nano /etc/ansible/hosts
```



```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ 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
```



```
ubuntu@ip-172-31-30-114: ~  
GNU nano 2.9.3 /etc/ansible/hosts  
  
# 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  
  
[servers]  
host ansible_ssh_host=3.16.255.89
```

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.

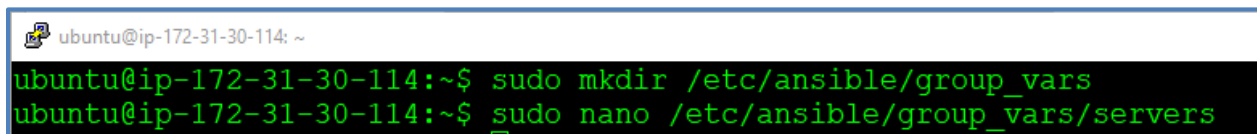
Now we will create a file that tells all the servers in the "servers" group to connect as the root user.

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.

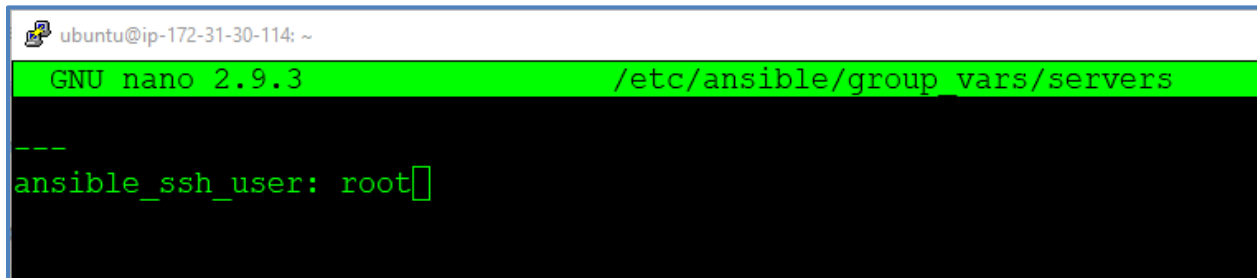
```
$ sudo mkdir /etc/ansible/group_vars
```

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



```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ sudo mkdir /etc/ansible/group_vars  
ubuntu@ip-172-31-30-114:~$ sudo nano /etc/ansible/group_vars/servers
```

put our configuration in there as shown below.



```
ubuntu@ip-172-31-30-114: ~  
GNU nano 2.9.3 /etc/ansible/group_vars/servers  
---  
ansible_ssh_user: root
```

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
```

```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ ansible -m ping all  
host | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

The same can also be achieved by using the following commands

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

```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ ansible -m ping servers  
host | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

We could also specify an individual host.

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

```
ubuntu@ip-172-31-30-114: ~  
ubuntu@ip-172-31-30-114:~$ ansible -m ping host  
host | SUCCESS => {  
    "changed": false,  
    "ping": "pong"  
}
```

Congratulations! You have successfully setup Ansible Master-Slave Cluster