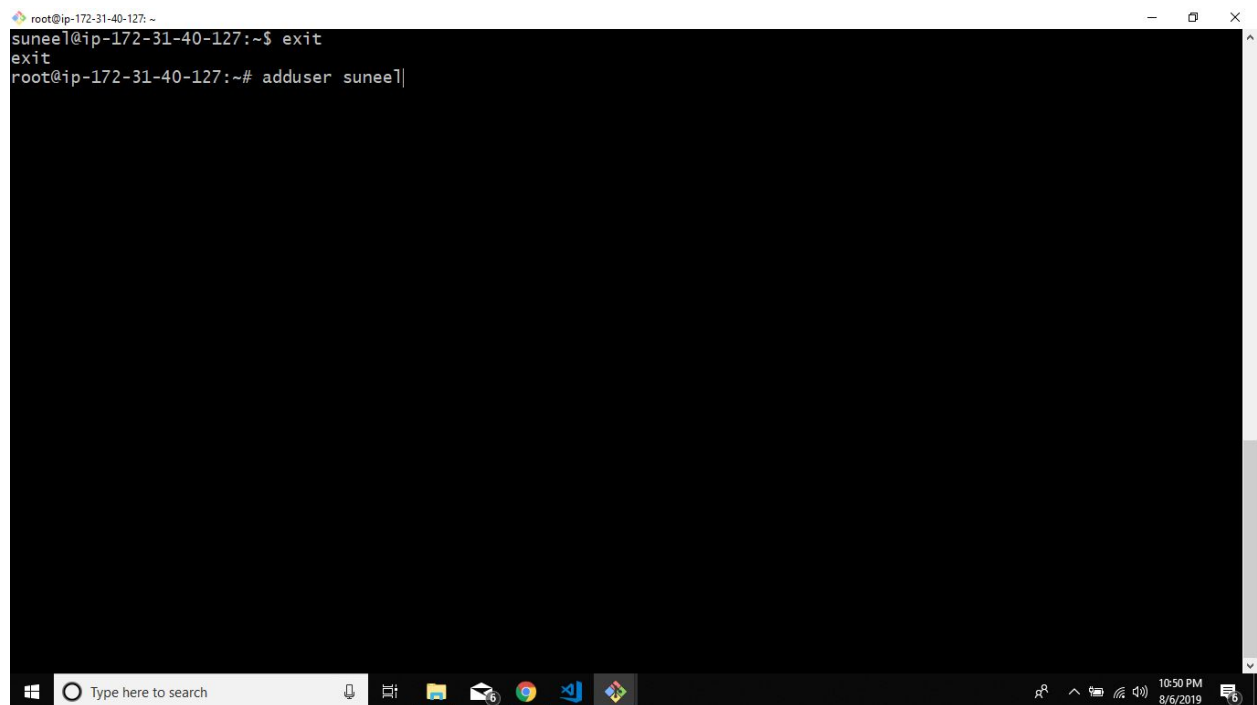


Ansible Documentry

Ssh password less authentication :

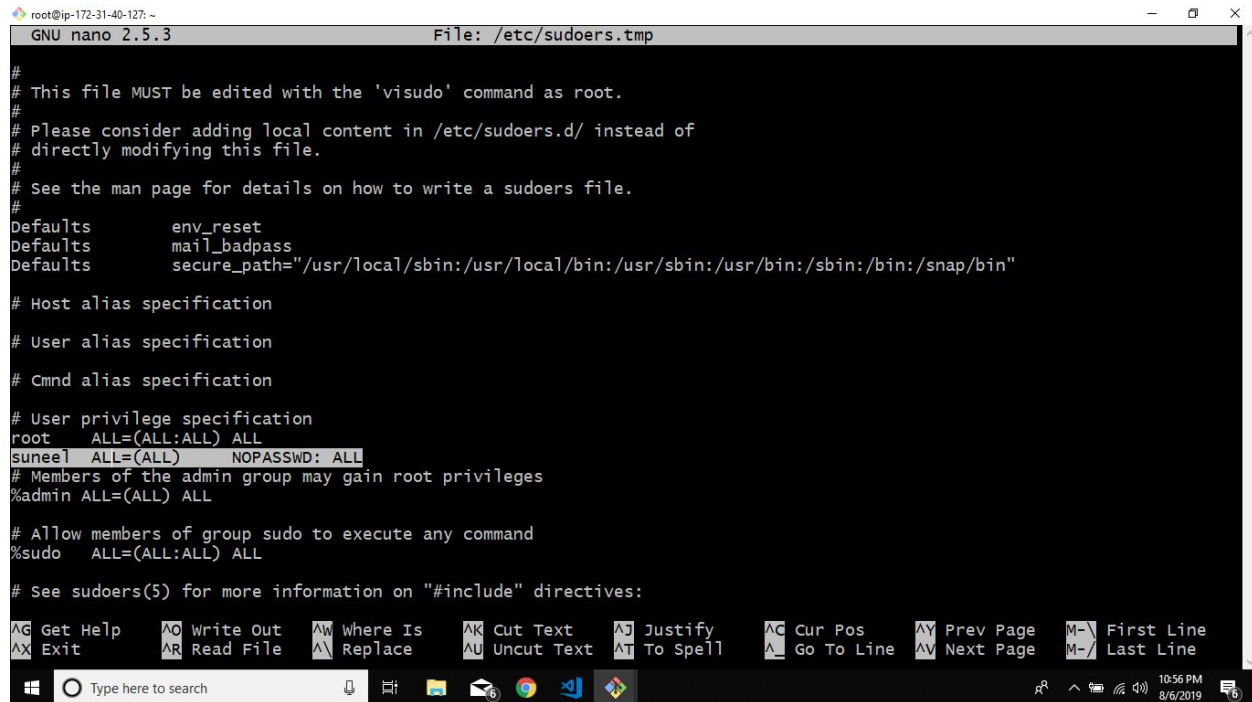
IN MASTER

- 1) Start two instances and name it as master and slave respectively
- 2) Give ssh and all traffic in security groups
- 3) launch the master through git bash

A screenshot of a terminal window titled 'root@ip-172-31-40-127: ~'. The terminal shows a shell session where a user named 'suneel' has logged in. The prompt is 'suneel@ip-172-31-40-127:~\$'. The user enters 'exit', and the prompt changes to 'root@ip-172-31-40-127:~#'. The user then enters 'adduser suneel'. The terminal window is running on a Windows operating system, as evidenced by the taskbar at the bottom showing the Start button, search bar, and various application icons. The system clock in the bottom right corner indicates the time is 10:50 PM on 8/6/2019.

```
root@ip-172-31-40-127: ~
suneel@ip-172-31-40-127:~$ exit
exit
root@ip-172-31-40-127:~# adduser suneel|
```

- 4) add a new user using “adduser” command and give password



```
root@ip-172-31-40-127: ~
GNU nano 2.5.3 File: /etc/sudoers.tmp

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
suneel  ALL=(ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos  ^Y Prev Page M-^ First Line
^X Exit      ^R Read File ^_ Replace   ^U Uncut Text ^T To Spell  ^G Go To Line ^V Next Page M-^ Last Line
```

5) and give user privileges using “visudo” and entering the command “ suneel ALL=(ALL) NOPASSWD: ALL ” all caps..

6) install ansible using “ apt-get install ansible ”.

6.1) it will pick the latest version of ansible and it will be installed.

Note: when you installing ansible the python is also installed in master node but we have to install python specifically on node instance

6.2) Install python along with it by using “ apt-get install python ”.

If you want to check the version try these commands

“ ansible --version ”

“ python --version ”

7) for permitting password less authentication we have to do the following steps

Enter the command

“ vi /ect/ssh/sshd_config ”

```
root@ip-172-31-88-146: ~  
RSAAuthentication yes  
PubkeyAuthentication yes  
#AuthorizedKeysFile      %h/.ssh/authorized_keys  
  
# Don't read the user's ~/.rhosts and ~/.shosts files  
IgnoreRhosts yes  
# For this to work you will also need host keys in /etc/ssh_known_hosts  
RhostsRSAAuthentication no  
# similar for protocol version 2  
HostbasedAuthentication no  
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication  
#IgnoreUserKnownHosts yes  
  
# To enable empty passwords, change to yes (NOT RECOMMENDED)  
PermitEmptyPasswords no  
  
# Change to yes to enable challenge-response passwords (beware issues with  
# some PAM modules and threads)  
ChallengeResponseAuthentication no  
  
# Change to no to disable tunnelled clear text passwords  
PasswordAuthentication yes  
  
# Kerberos options  
#KerberosAuthentication no  
#KerberosGetAFSToken no  
#KerberosOrLocalPasswd yes  
#KerberosTicketCleanup yes  
  
# GSSAPI options  
#GSSAPIAuthentication no  
#GSSAPICleanupCredentials yes
```

Edit it from no to yes and save it

8) Now you have to perform service restart by using the command

“ service ssh restart ”

9) now we have to perform keygen in the user in master node for that we have to log on to sub user by using su suneel (in this case suneel is user name)

```
suneel@ip-172-31-88-146: ~$
Selecting previously unselected package ansible.
Preparing to unpack .../ansible_2.0.0.2-2ubuntu1.3_all.deb .
..
Unpacking ansible (2.0.0.2-2ubuntu1.3) ...
Selecting previously unselected package python-selinux.
Preparing to unpack .../python-selinux_2.4-3build2_amd64.deb
...
Unpacking python-selinux (2.4-3build2) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up libpython2.7-stdlib:amd64 (2.7.12-1ubuntu0~16.04.4) ...
Setting up python2.7 (2.7.12-1ubuntu0~16.04.4) ...
Setting up libpython-stdlib:amd64 (2.7.12-1~16.04) ...
Setting up python (2.7.12-1~16.04) ...
Setting up python-crypto (2.6.1-6ubuntu0.16.04.3) ...
Setting up python-markupsafe (0.23-2build2) ...
Setting up python-jinja2 (2.8-1ubuntu0.1) ...
Setting up python-six (1.10.0-3) ...
Setting up python-ecdsa (0.13-2) ...
Setting up python-paramiko (1.16.0-1ubuntu0.2) ...
Setting up python-pkg-resources (20.7.0-1) ...
Setting up python-yaml (3.11-3build1) ...
Setting up python-httplib2 (0.9.1+dfsg-1) ...
Setting up ieee-data (20150531.1) ...
Setting up python-netaddr (0.7.18-1) ...
Setting up ansible (2.0.0.2-2ubuntu1.3) ...
Setting up python-selinux (2.4-3build2) ...
root@ip-172-31-88-146:~# vi /etc/ssh/sshd_config
root@ip-172-31-88-146:~# service ssh restart
root@ip-172-31-88-146:~# su suneel
suneel@ip-172-31-88-146:/root$ cd
suneel@ip-172-31-88-146:~$
```

10) we have to perform ssh keygen using the command
10.1) ” **ssh-keygen**” enter this command in user

```
suneel@ip-172-31-88-146: ~$
suneel@ip-172-31-88-146:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/suneel/.ssh/id_rsa):
Created directory '/home/suneel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/suneel/.ssh/id_rsa.
Your public key has been saved in /home/suneel/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:qTsgD6mI1B3Y0wXsks+QbyoYibo5TbKE/MmBC7G1wq0 suneel@ip-172-31-88-146
The key's randomart image is:
+----[RSA 2048]-----+
|  o.o                    |
| o + .                   |
|..o = +                  |
|*..+ * . .               |
|+X+O= . S                |
|XB*+o .                  |
|BO==o..                  |
|B.E+. .                  |
|..                        |
+----[SHA256]-----+
suneel@ip-172-31-88-146:~$
```

NOW WE HAVE TO CONFIGURE NODE :

- 1) We have to perform all the tasks which are done in master except installing of ansible.
- 2) Now add a new user and the username and password of the node must match with the user name and password in master.
- 3) And install python in the node using command
“ apt-get install python ”

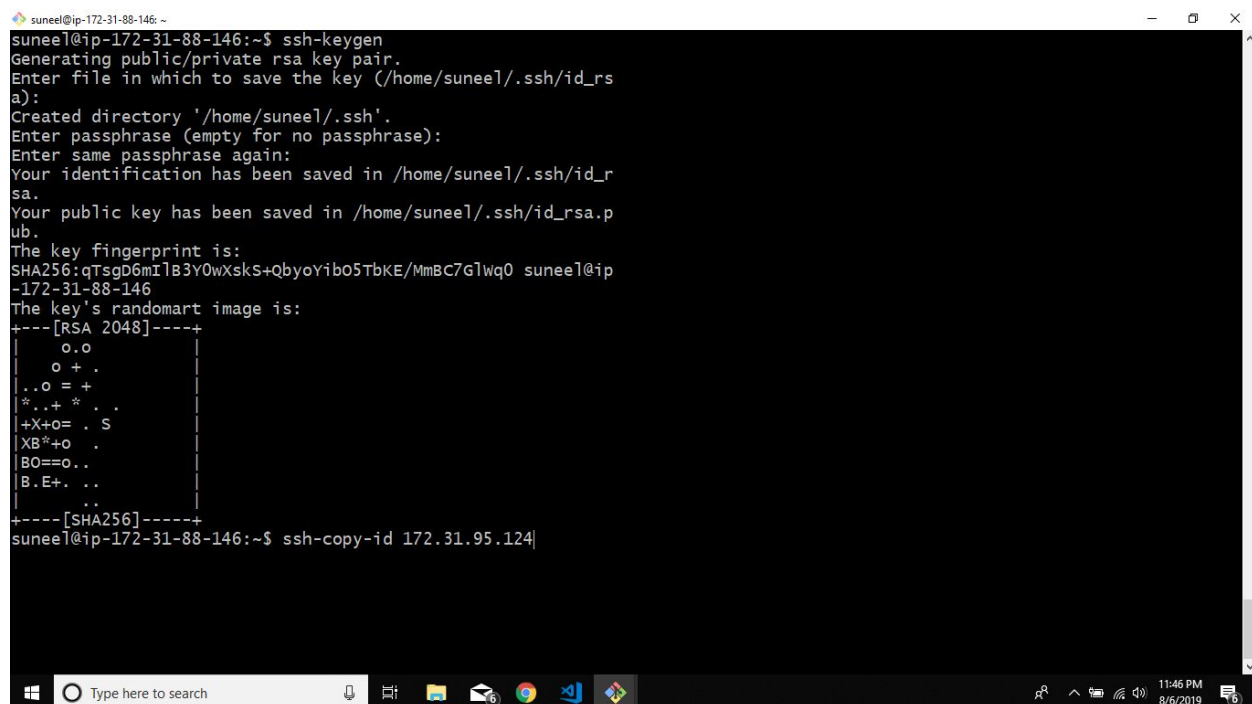
And now go back to master node

Again in master node we have to log on to the user using su sunee1

Cd ..

10.2) we have to add the node to master by using the following command

“ ssh-copy-id (add a space and paste the private ip of the node in this case 172.31.95.124)” .



```
suneel@ip-172-31-88-146: ~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/suneel/.ssh/id_rsa):
Created directory '/home/suneel/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/suneel/.ssh/id_rsa.
Your public key has been saved in /home/suneel/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:qTsgD6mI1B3Y0wXskS+QbyoYibo5TbKE/MmBC7G1Wq0 suneel@ip-172-31-88-146
The key's randomart image is:
+---[RSA 2048]---+
  o.o
  o + .
  ..o = +
  *..+ * .
  |X+O= . S
  |XB*+o .
  |BO==o..
  |B.E+. ..
  |
+---[SHA256]---+
suneel@ip-172-31-88-146:~$ ssh-copy-id 172.31.95.124|
```

10.2) After this it will add the node to master and it will ask a (yes or no) hit yes to continue adding and it will ask for password of the user of the node enter the password and hit enter

```
suneel@ip-172-31-88-146: ~
sa.
Your public key has been saved in /home/suneel/.ssh/id_rsa.p
ub.
The key fingerprint is:
SHA256:qTsgD6mI1B3Y0wXsks+QbyoYibo5TbKE/MmBC7G1wq0 suneel@ip
-172-31-88-146
The key's randomart image is:
+---[RSA 2048]-----+
|  o.o                    |
|  o + .                  |
| ..o = +                |
| *.+ * . .              |
| +X+o= . S              |
| XB*+o .                |
| B0==o..                |
| B.E+. .                |
| ..                      |
+---[SHA256]-----+
suneel@ip-172-31-88-146:~$ ssh-copy-id 172.31.95.124
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/suneel/.ssh/id_rsa.pub"
The authenticity of host '172.31.95.124 (172.31.95.124)' can't be established.
ECDSA key fingerprint is SHA256:NPTpVq7ijsGYC5zFnpVG0KZHAUiYkqpPV10UB4r05us.
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 prompted now it is to install the new keys
suneel@172.31.95.124's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '172.31.95.124'"
and check to make sure that only the key(s) you wanted were added.

suneel@ip-172-31-88-146:~$ |
```

You can now log on to the user on the node using the given command
“ ssh 172.31.95.124 ”

```
suneel@ip-172-31-95-124: ~
suneel@ip-172-31-88-146:~$ ssh 172.31.95.124
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1087-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

26 packages can be updated.
19 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

suneel@ip-172-31-95-124:~$ |
```

Then it will log on to the user in the node through ssh password less authentication method.

Now we can directly ping the node from master if we want to

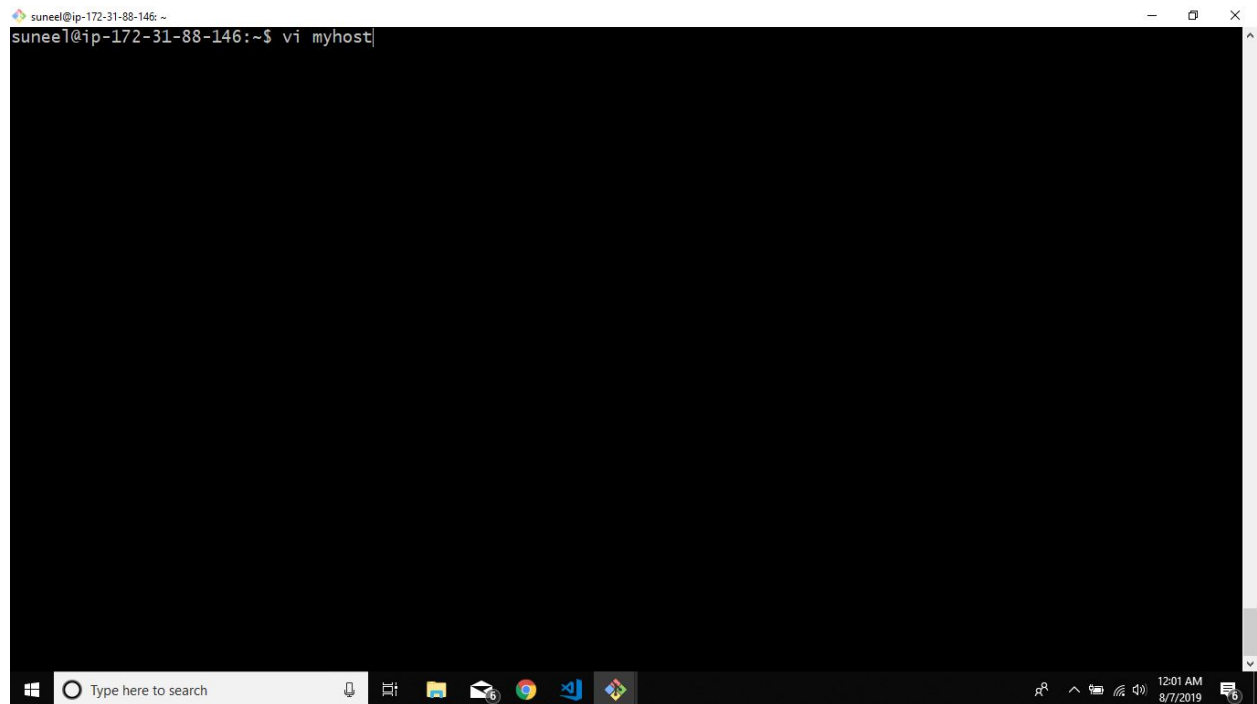
```
suneel@ip-172-31-88-146: ~  
suneel@ip-172-31-88-146:~$ ssh 172.31.95.124  
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1087-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
26 packages can be updated.  
19 updates are security updates.  
  
New release '18.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
suneel@ip-172-31-95-124:~$ exit  
logout  
Connection to 172.31.95.124 closed.  
suneel@ip-172-31-88-146:~$ ping 172.31.95.124  
PING 172.31.95.124 (172.31.95.124) 56(84) bytes of data.  
64 bytes from 172.31.95.124: icmp_seq=1 ttl=64 time=0.452 ms  
64 bytes from 172.31.95.124: icmp_seq=2 ttl=64 time=0.564 ms  
64 bytes from 172.31.95.124: icmp_seq=3 ttl=64 time=0.460 ms  
64 bytes from 172.31.95.124: icmp_seq=4 ttl=64 time=0.422 ms  
64 bytes from 172.31.95.124: icmp_seq=5 ttl=64 time=0.471 ms  
64 bytes from 172.31.95.124: icmp_seq=6 ttl=64 time=0.488 ms  
64 bytes from 172.31.95.124: icmp_seq=7 ttl=64 time=0.429 ms  
64 bytes from 172.31.95.124: icmp_seq=8 ttl=64 time=0.537 ms  
64 bytes from 172.31.95.124: icmp_seq=9 ttl=64 time=0.688 ms  
64 bytes from 172.31.95.124: icmp_seq=10 ttl=64 time=0.497 ms
```

The next step is we have to run adhoc command ping the node using the following adhoc command

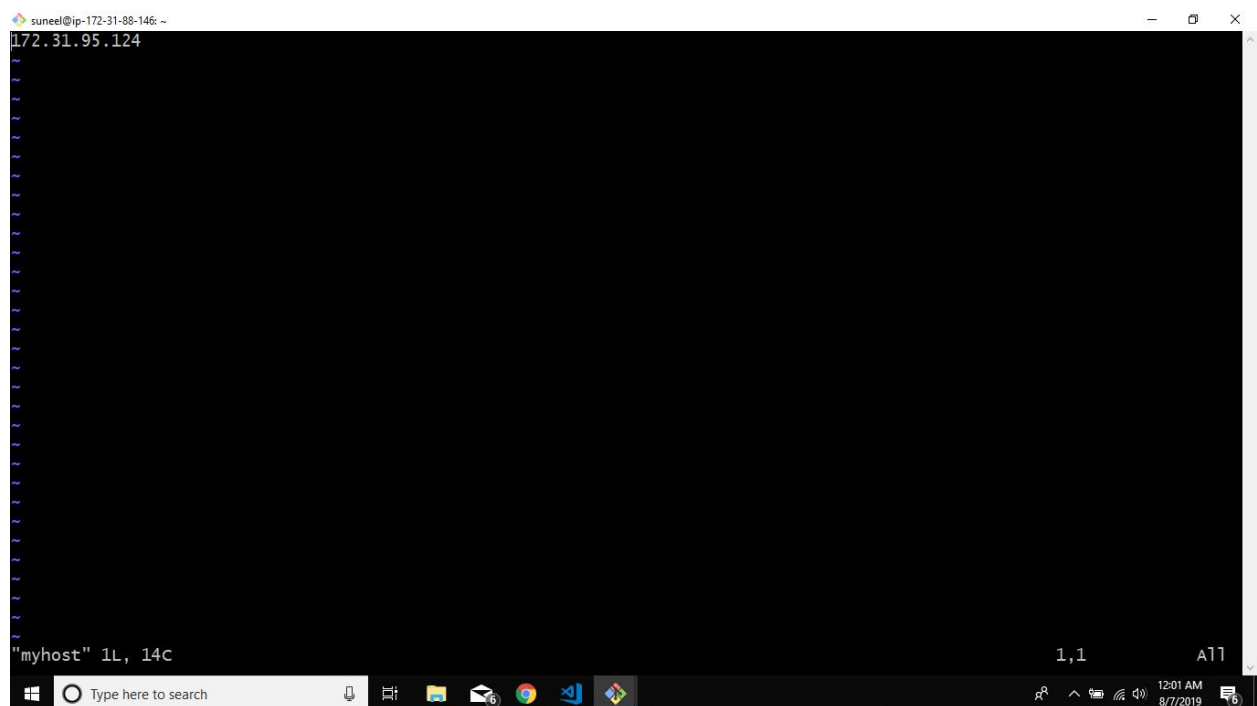
For that we need to create a file using vi editor

Eg: vi myhost

Enter the private ip of th node in the file



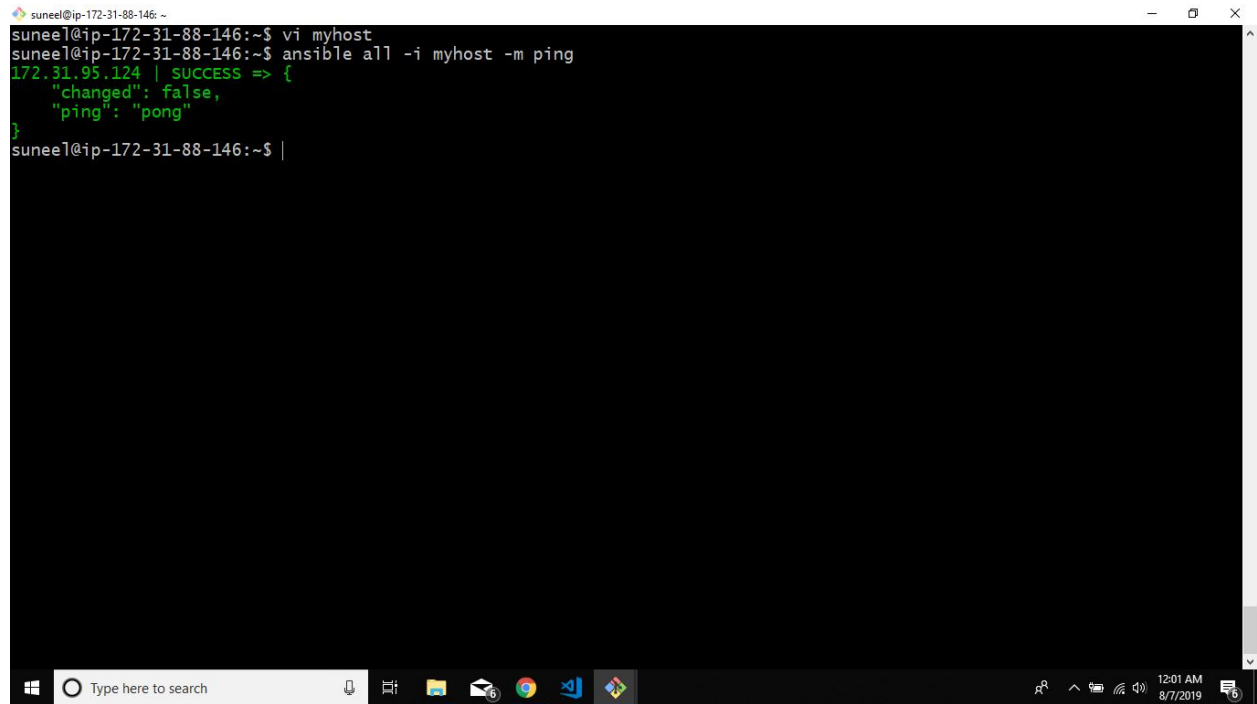
A terminal window titled 'suneel@ip-172-31-88-146: ~' with a black background. The command prompt shows 'suneel@ip-172-31-88-146:~\$ vi myhost'. The terminal is open to a new file named 'myhost'. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right indicates the time is 12:01 AM on 8/7/2019.



The same terminal window now displays the contents of the file 'myhost'. The first line is '172.31.95.124', followed by several lines of blue text representing IP addresses. At the bottom of the terminal, a status bar shows '"myhost" 1L, 14C' on the left, '1,1' in the center, and 'All' on the right. The Windows taskbar and system tray remain the same.

“ ansible all -i myhost -m ping “

The result will be like below

A terminal window titled 'suneel@ip-172-31-88-146: ~' with a black background and green text. The user has executed 'vi myhost' and then 'ansible all -i myhost -m ping'. The output shows a successful ping to 172.31.95.124 with a 'pong' response. The Windows taskbar is visible at the bottom with the search bar and system tray showing the time as 12:01 AM on 8/7/2019.

```
suneel@ip-172-31-88-146:~$ vi myhost
suneel@ip-172-31-88-146:~$ ansible all -i myhost -m ping
172.31.95.124 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
suneel@ip-172-31-88-146:~$ |
```

It means we are pinging the nodes with ad-hoc command.

Now we are pinging a node using an Ansible playbook

For that we have to create a .yml file eg: apache2.yml

Vi apache2.yml we have to write a playbook using apt services

Which will be mentioned in the image below

```
suneel@ip-172-31-88-146: ~  
--  
- hosts: all  
  become: yes  
  tasks:  
    - name: apache2  
      apt:  
        name: apache2  
        state: present  
    - name: apache2  
      service:  
        name: apache2  
        state: started  
--  
-- INSERT --  
1,1 All
```

It is a sample playbook to installing apache2 web services on node

Now we have to run the playbook using the adhoc command

“ ansible-playbook -i myhost apache2.yml “

```
suneel@ip-172-31-88-146: ~  
suneel@ip-172-31-88-146:~$ vi apache2.yml  
suneel@ip-172-31-88-146:~$ ansible-playbook -i myhost apache2.yml  
  
PLAY *****  
  
TASK [setup] *****  
ok: [172.31.95.124]  
  
TASK [apache2] *****  
changed: [172.31.95.124]  
  
TASK [apache2] *****  
ok: [172.31.95.124]  
  
PLAY RECAP *****  
172.31.95.124 : ok=3 changed=1 unreachable=0 failed=0  
  
suneel@ip-172-31-88-146:~$ |
```

Now the apache2 was installed on the node which is connected to the master

And we can observe the changes in yellow color which indicates the task has completed successfully