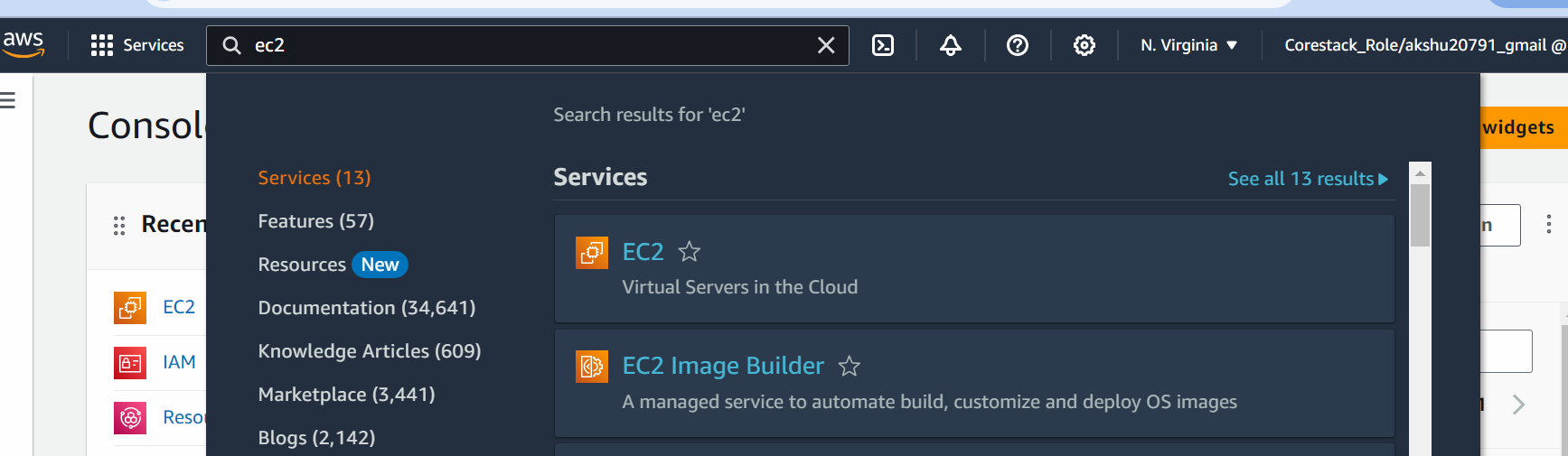
**Lesson 01 Demo 01**

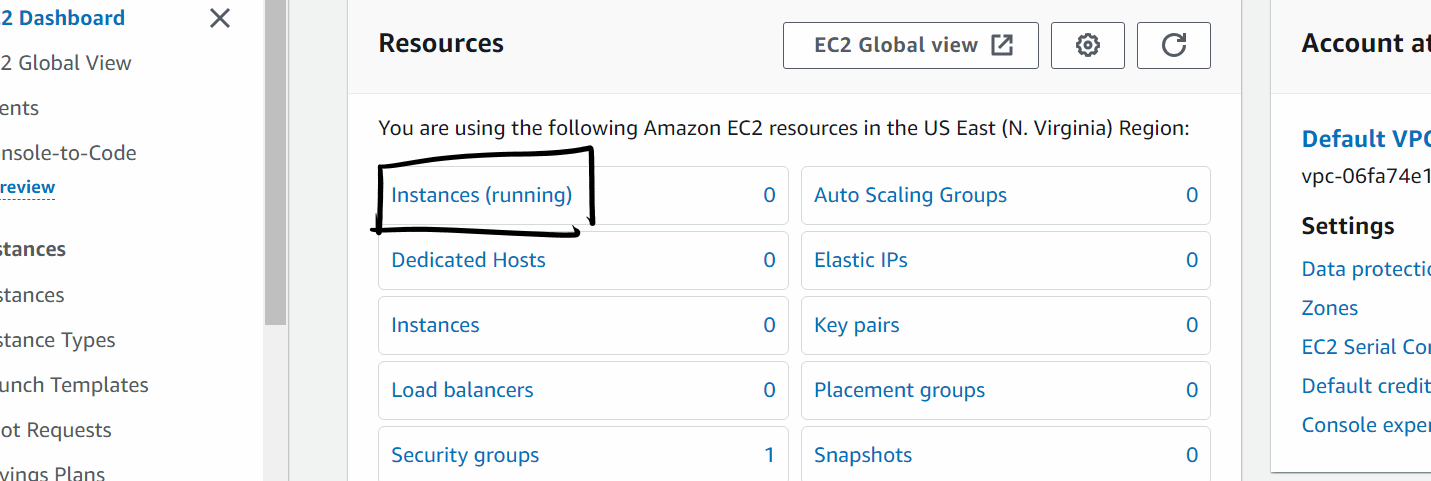
**INSTALLING AND CONFIGURING ANSIBLE**

**Objective:** Using Ansible as a master node architecture in aws Ec2 machine

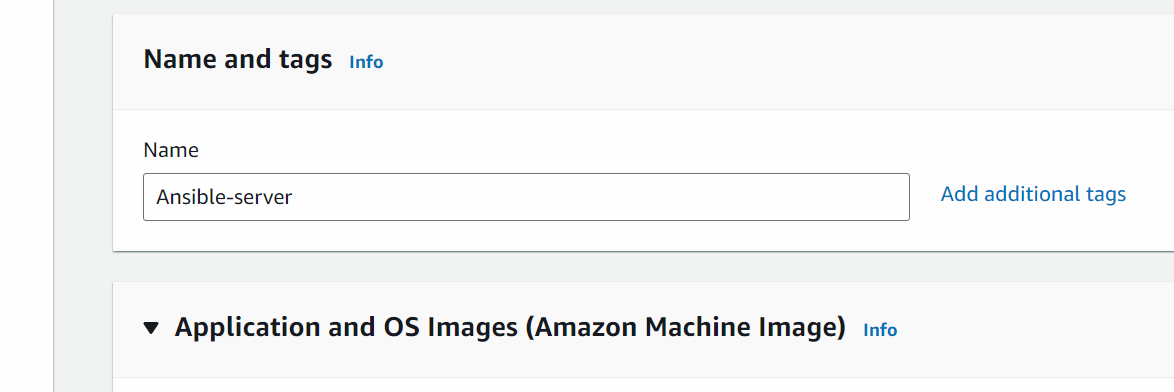
**Tools required:** Python , Ansible, AWS

**Prerequisites:** NA

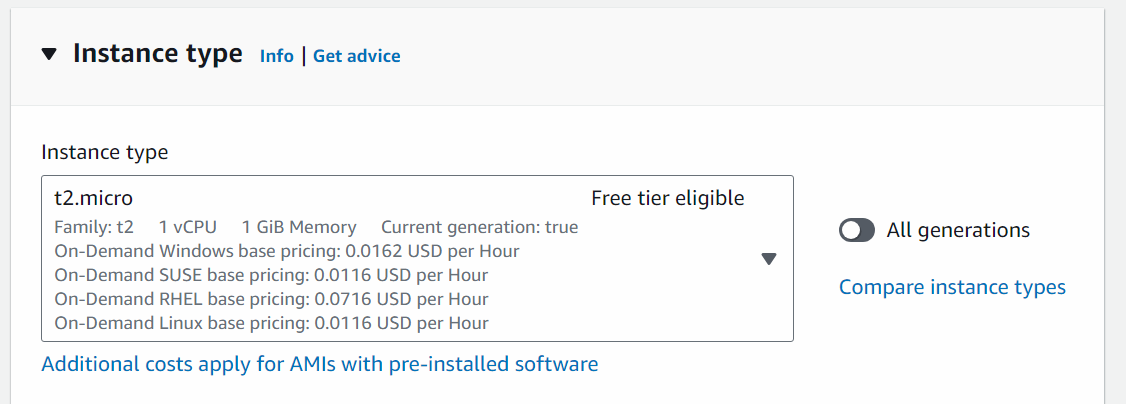




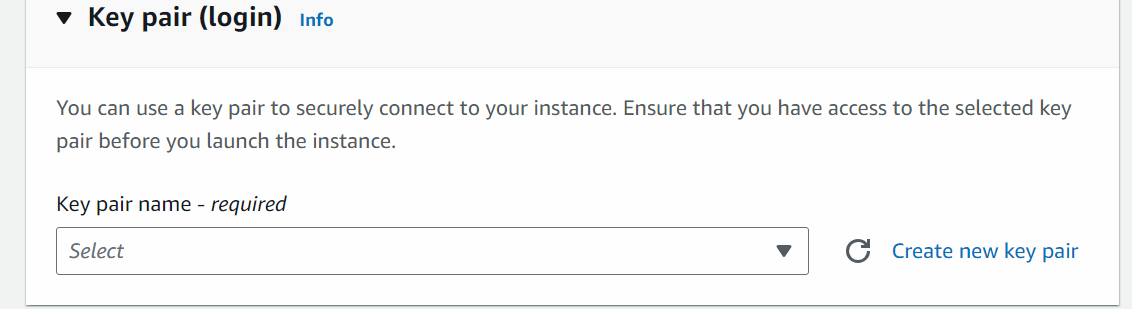
**Click on Launch instances**

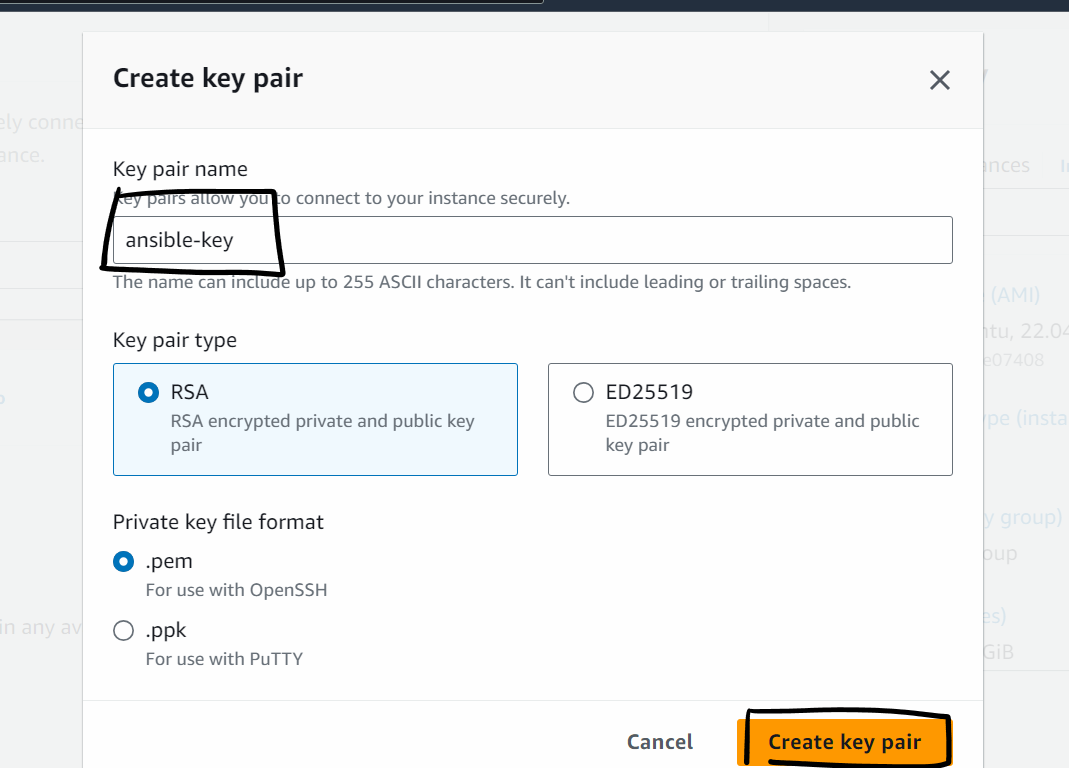


**Select AMI as Ubuntu 24.4**



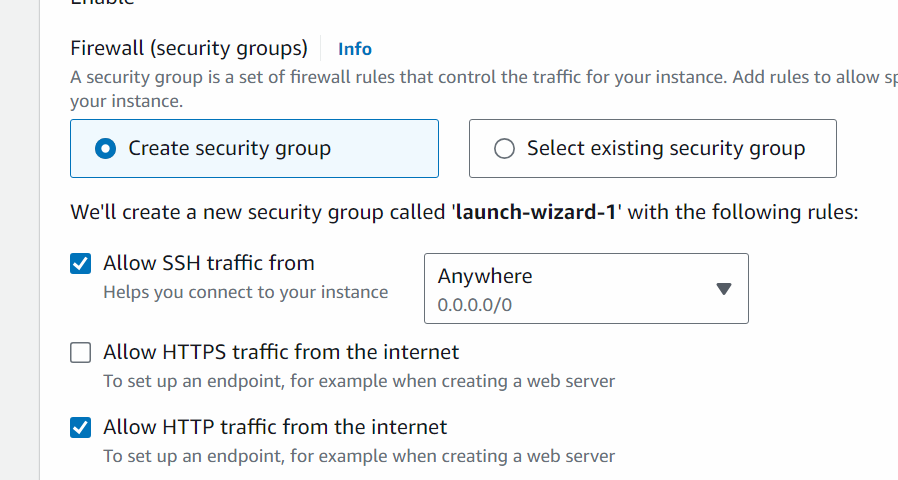
**Click on Create new key pair and create a new key pair**



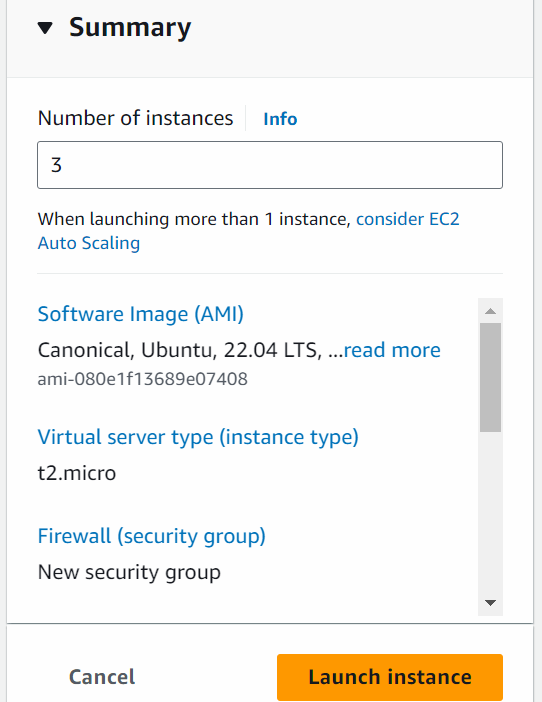


**The key would be downloaded to the machine**

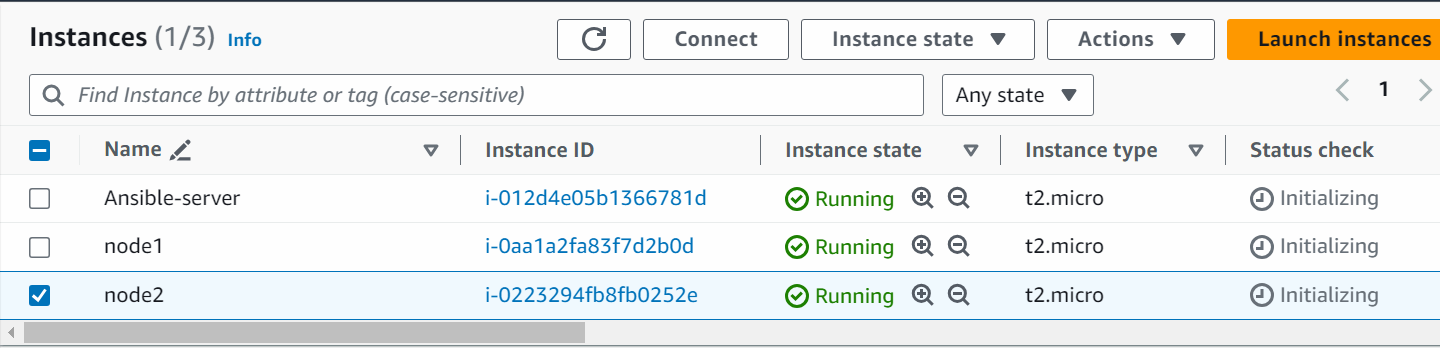
**In Firewall ,**



**Change the number of instances to 3 and launch the instances. We will consider one machine as master machine and other two as nodes**



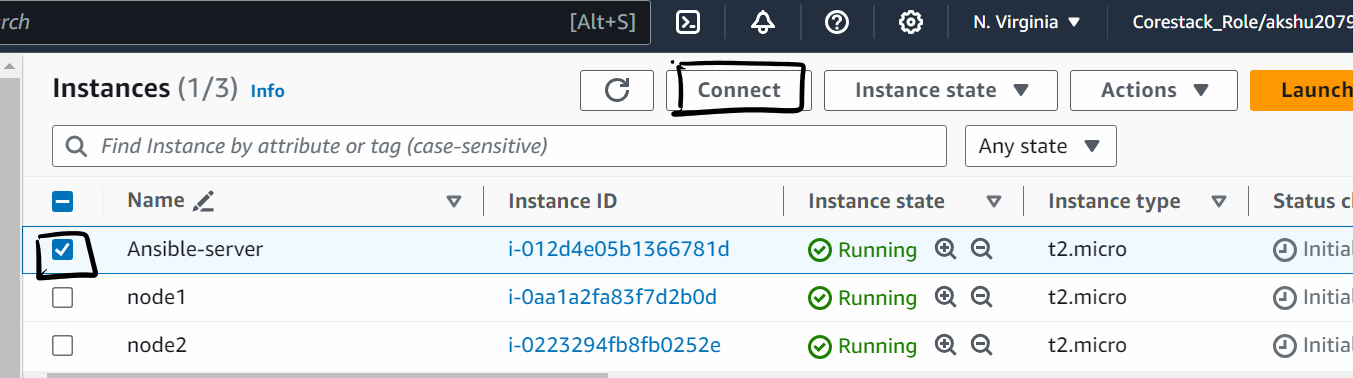
**After Machines are launched We can rename them as :**



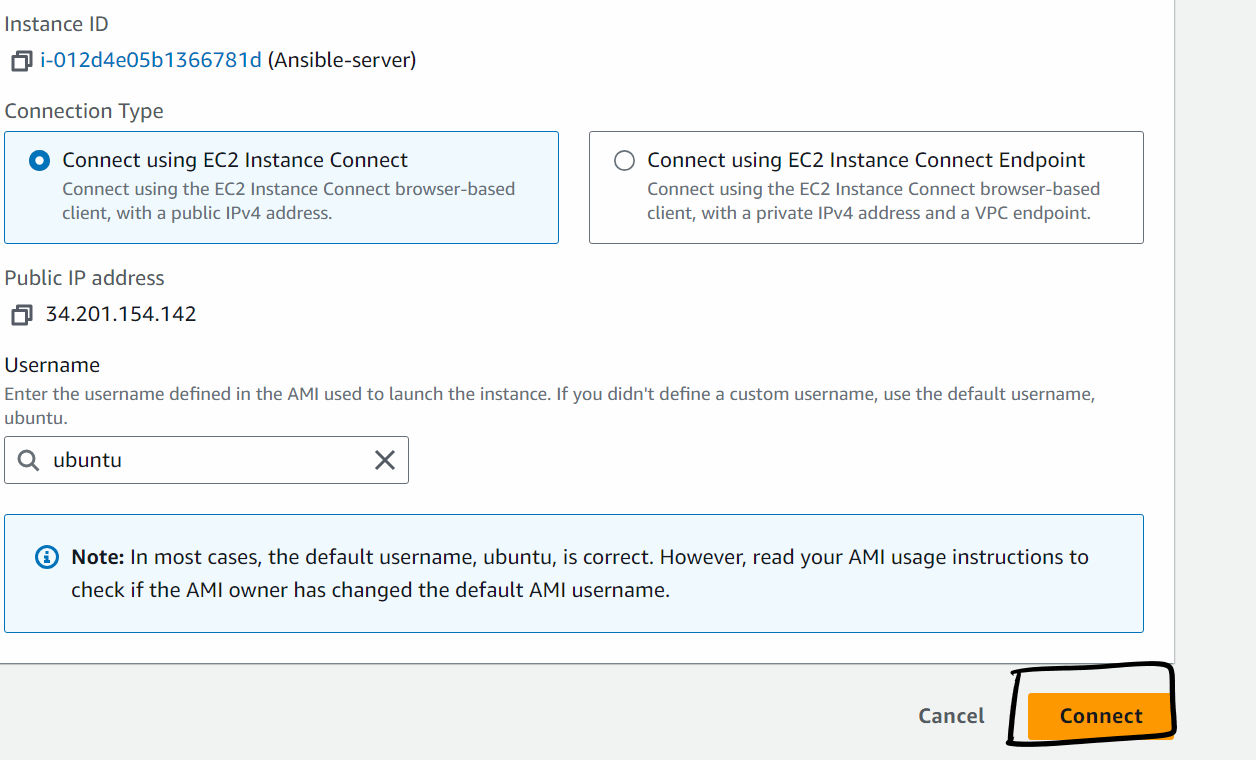
**Step 2: Now we will connect to these machines**

We can connect to the machine directly via browser

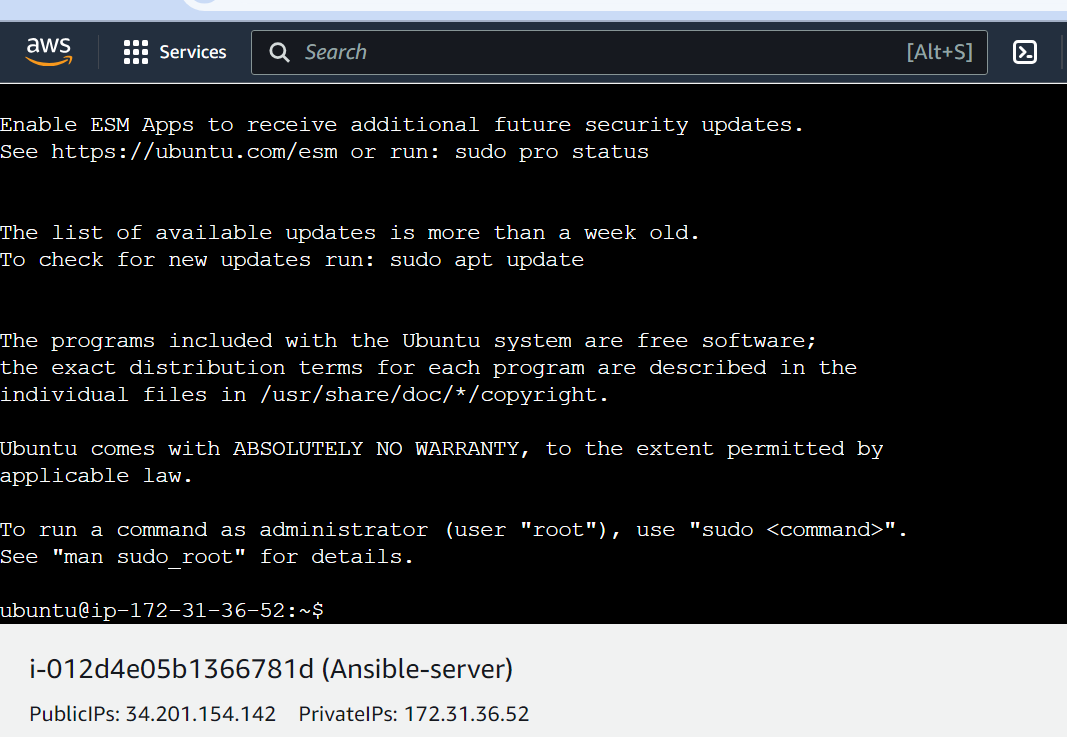
Select the ansible-server and click on connect



Click on Ec2 instance connect tab



Note: Do not change the username



Similarly, We can connect with other machines as well

Step 3: lets now install ansible in ansible server (execute the below command only in master machine)

# sudo hostname Ansiblemastermachine

# sudo su

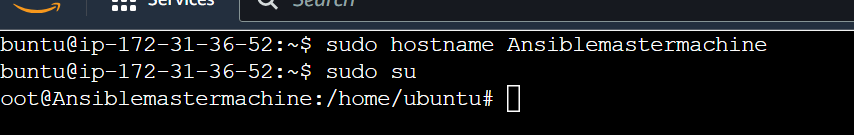
# apt update -y

#  apt-get install -y software-properties-common

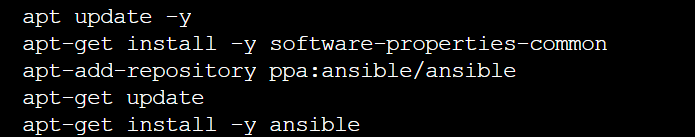
#  apt-add-repository ppa:ansible/ansible

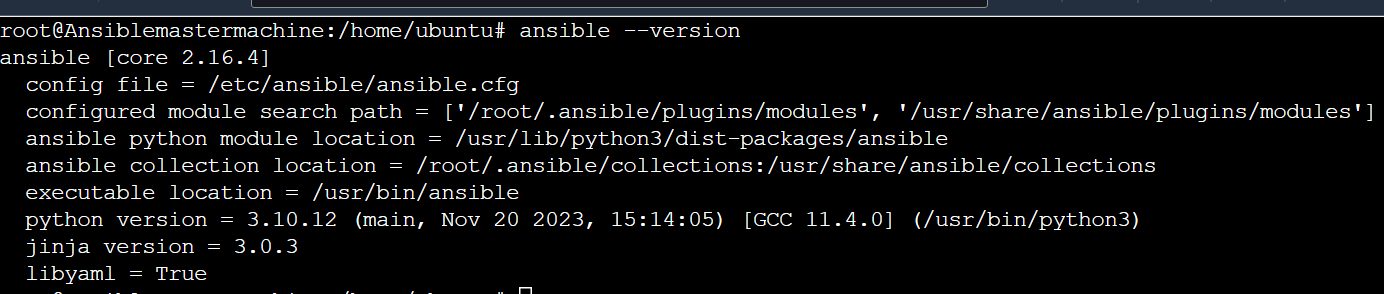
#  apt-get update

#  apt-get install -y ansible



# ansible --version





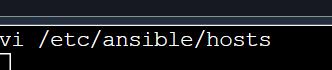
Step 4: we will now define the hosts from the master machine (ansible server)

vi /etc/ansible/hosts

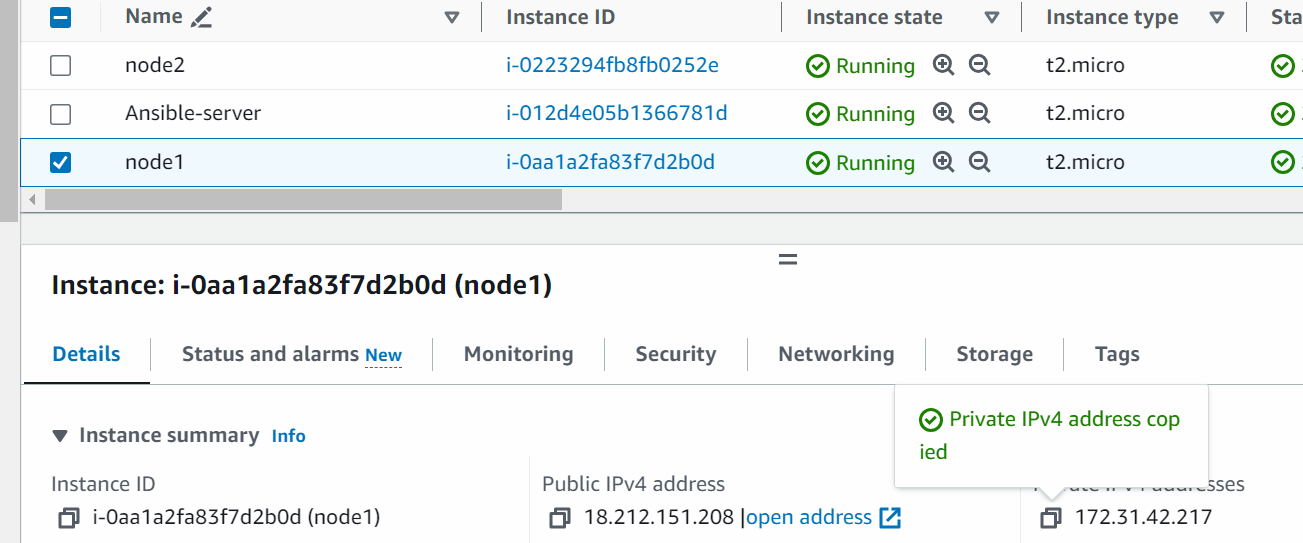
[ansiblegroup]

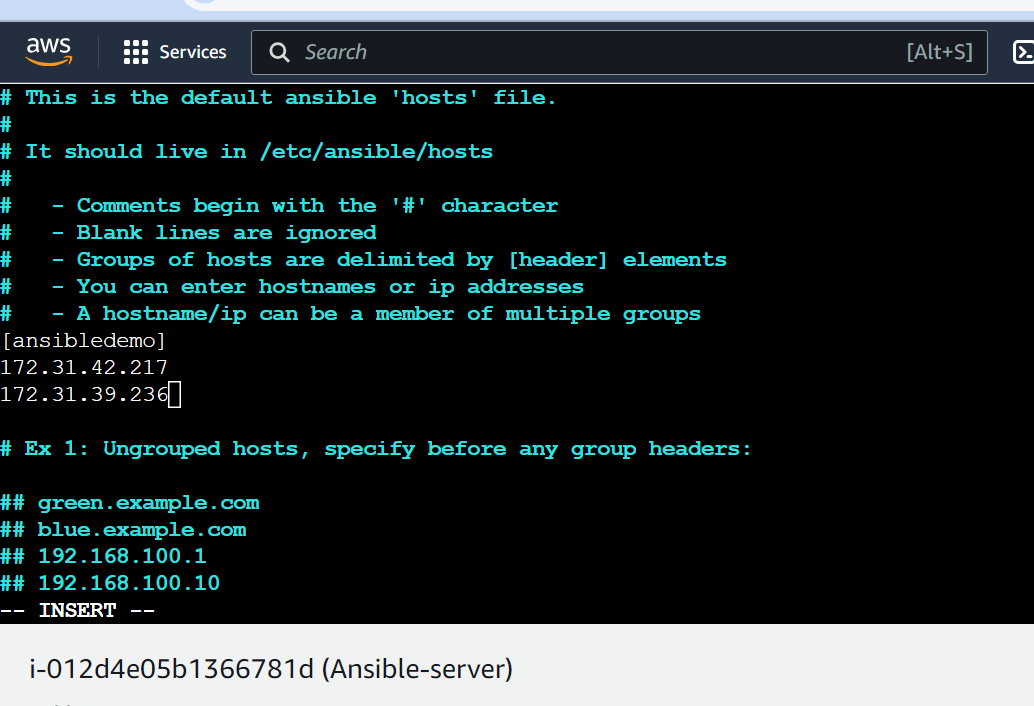
Privateip of node1

Privateip of node2



Copy the private ip of the node 1





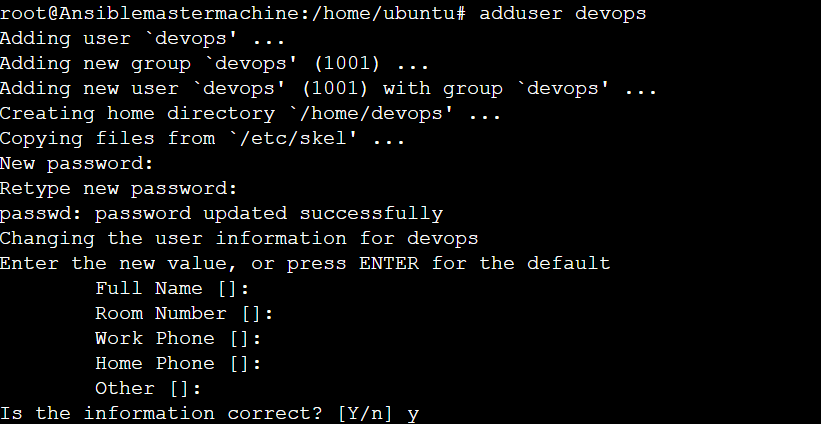
(we have to also copy the paste the private ip of the node2 as well)

Step 5: Create a user in Ansible server (master machine) and the nodes

# adduser devops

(put password as devops)

And press enter three times and press y



Similary create the same username and pass In the nodes as well

Perform same task in node2 as well

(use same username and pass in master and the nodes)

Step 6: We will now configure sshd configuration in master and node machines

# vi /etc/ssh/sshd\_config

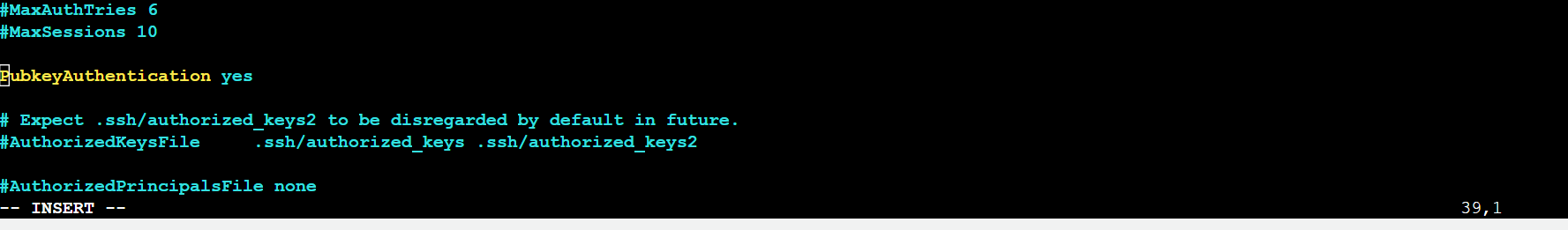
press i



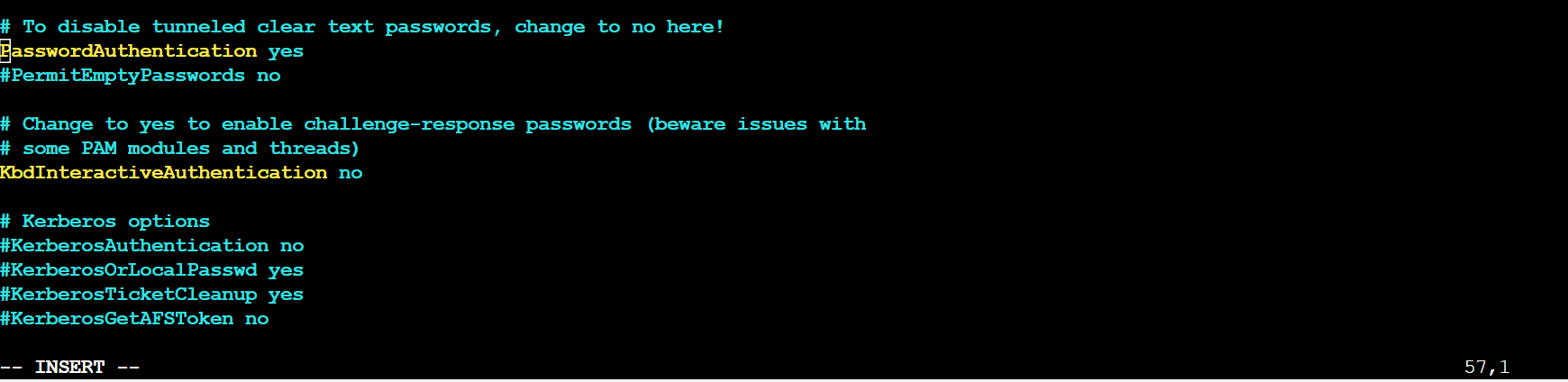
On line 34 change to PermitRootlogin yes and remove #



Remove # from line 39



On line 57 enable PasswordAuthentication as yes by removing #



(perform same activity in node1 and node2 as well)

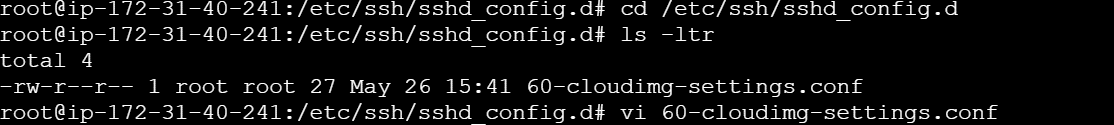
We need to restart ssh in master (ansible-server and nodes)

(with latest ubuntu these steps need to be added up in master and nodes)

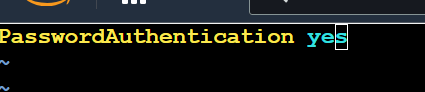
#cd /etc/ssh/sshd\_config.d

# ls -ltr

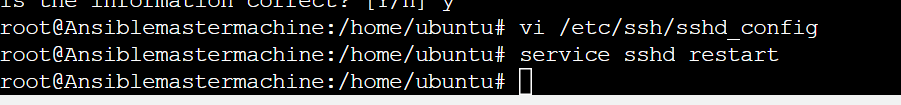
# vi 60-cloudimg-settings.conf



Change password authentication to yes

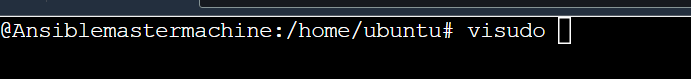


# service ssh restart



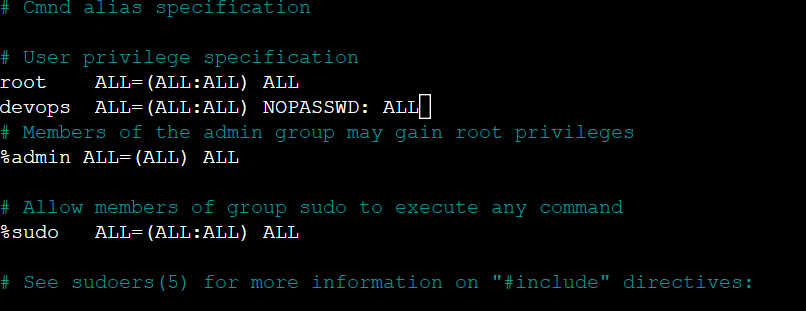
Step 8: We will now give sudo permission to the “devops” user in ansible and the nodes

# visudo



Scroll down to user priviledge specification and add below line

devops ALL=(ALL:ALL) NOPASSWD:ALL



Press ctrl x and then press Y and enter

(We need to perform the same activity for node1 and node2)

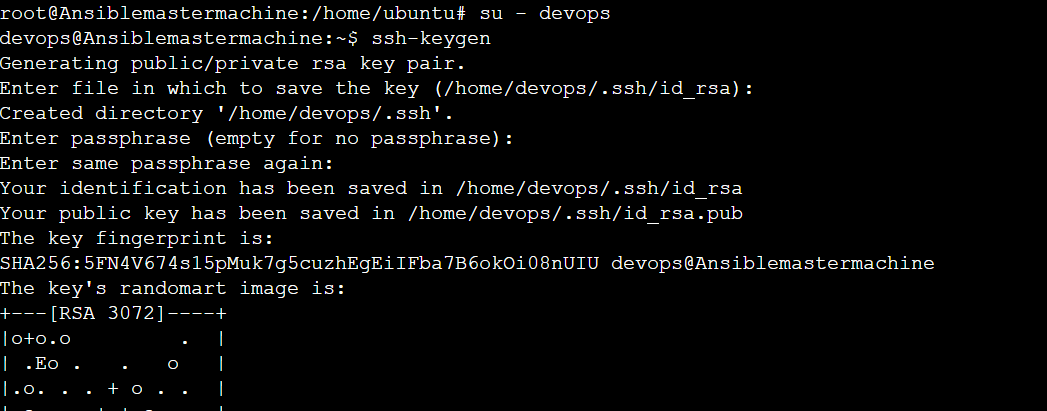
Step 7: From Ansible server we will generate the trust relationship with the nodes

In Ansible-server (master machine):

# su – devops

# ssh-keygen

(press enter three times)

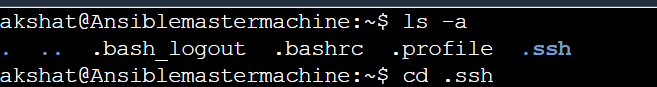


Now we need to copy the keypair in the node 1 and node2

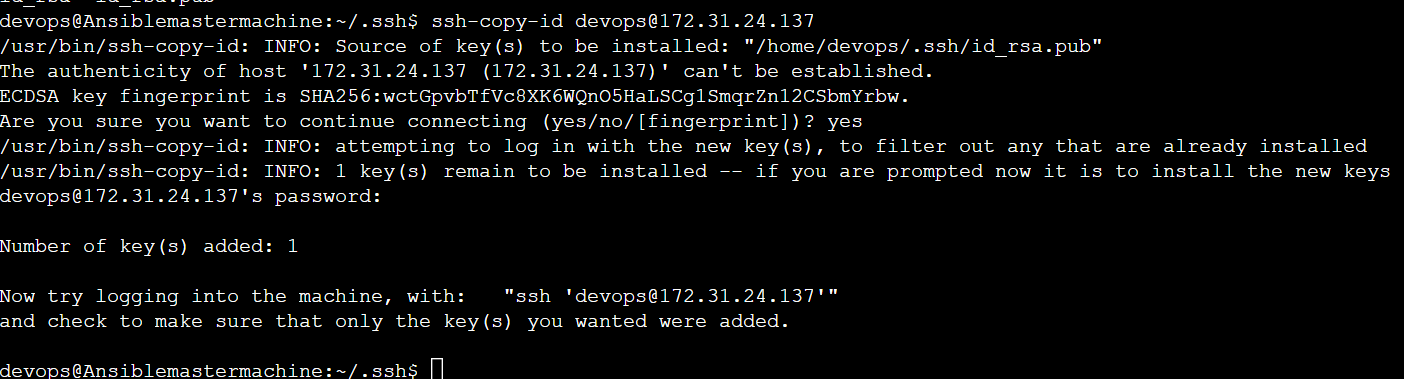
Go to Ansible-server(master)

# ls -a

# cd .ssh

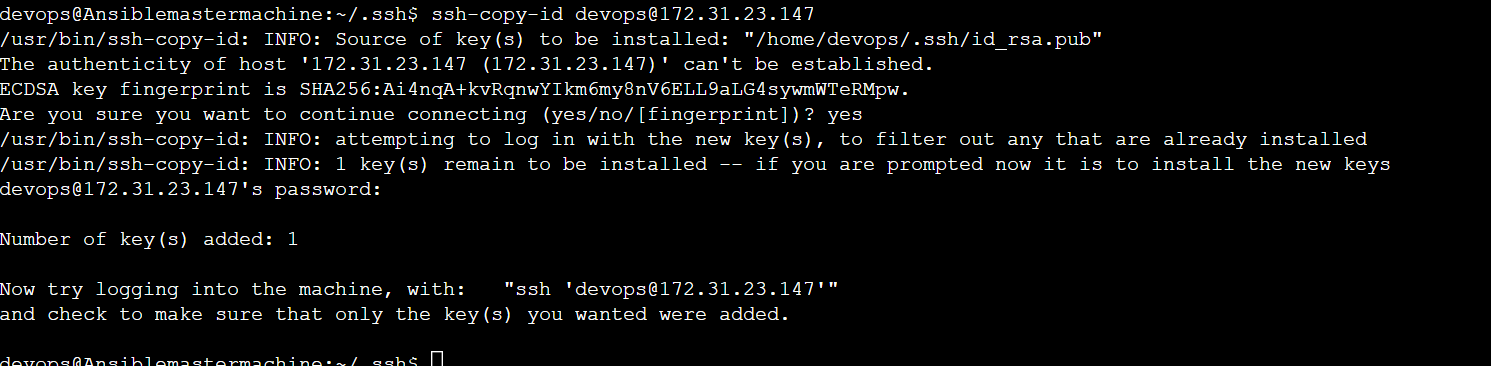


# ssh-copy-id devops@privateipofnode1



In the password put the password which we set while creating the user devops

Similarly copy to the node2 as well



We have now established the trust relationship of the master with the nodes.

Step 8: Lets now check if we are able to see the create files in the nodes via ansible

