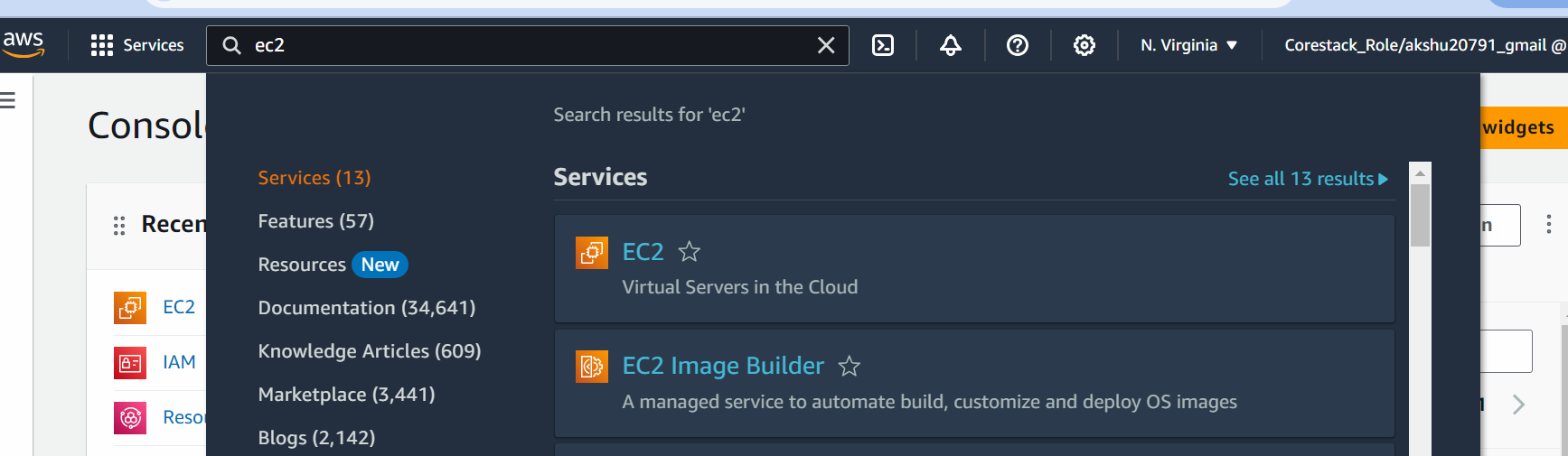
**Lesson 01 Demo 01**

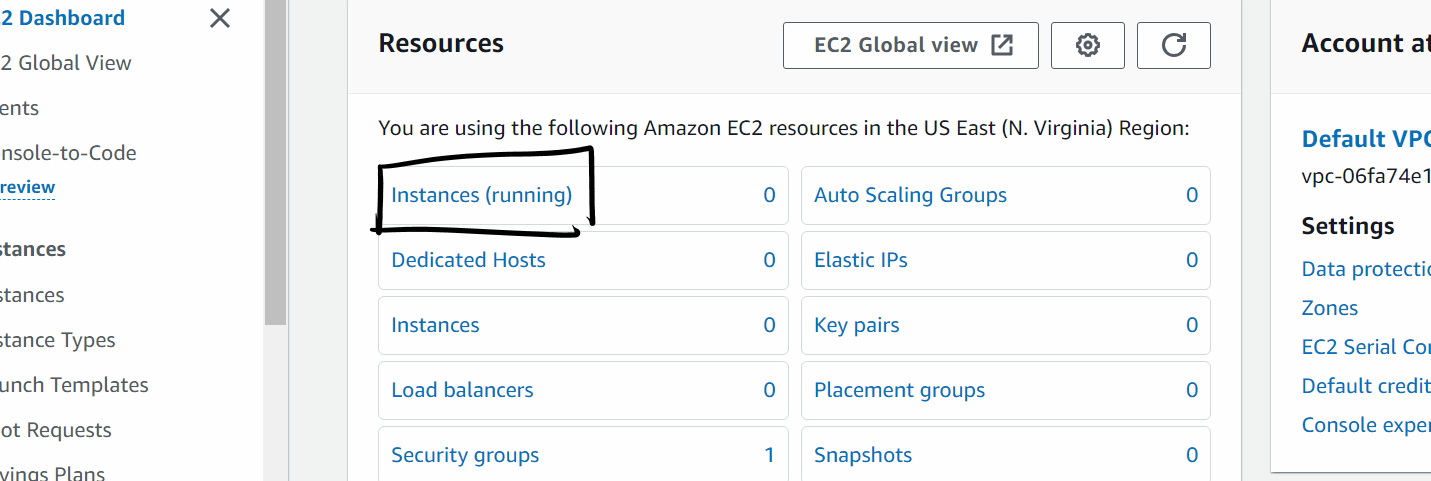
**INSTALLING AND CONFIGURING ANSIBLE**

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

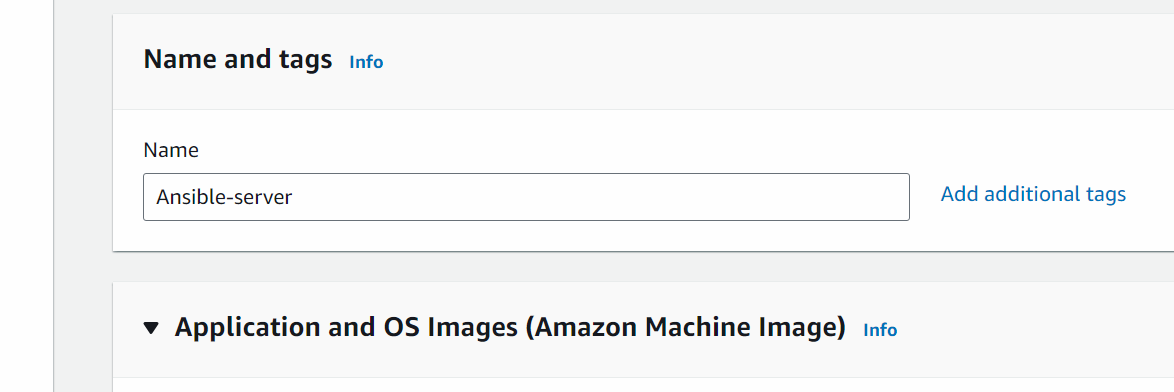
**Tools required:** Python , Ansible, AWS

**Prerequisites:** NA

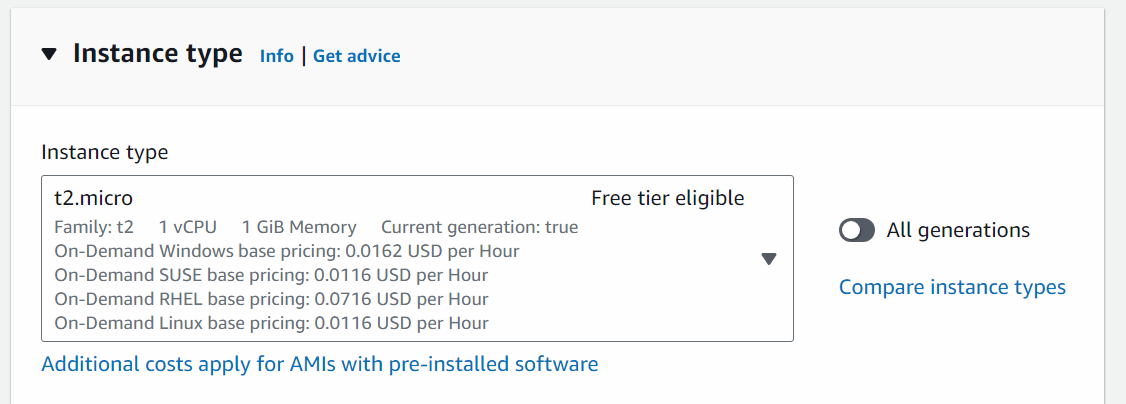




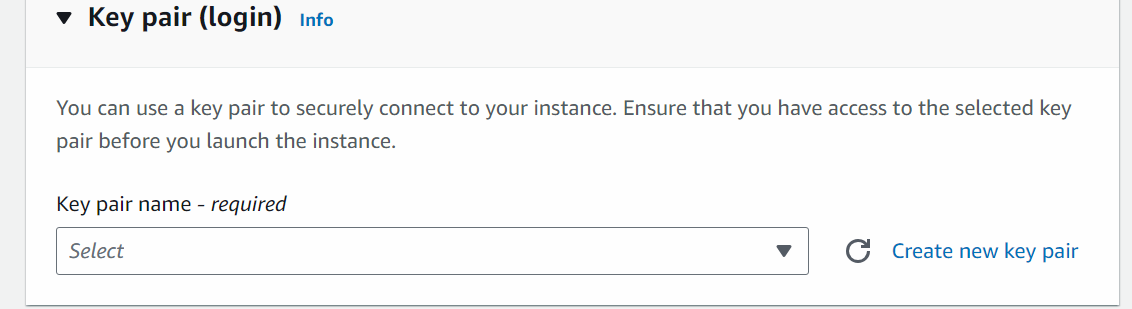
**Click on Launch instances**

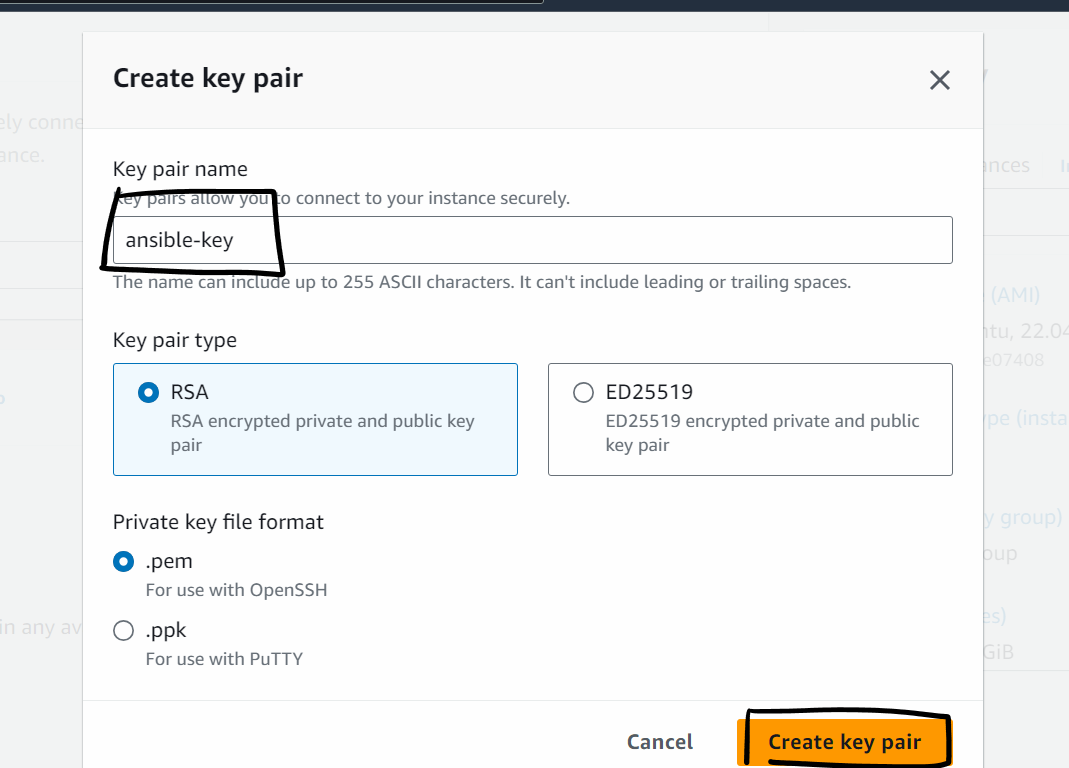


**Select AMI as Ubuntu 24.4**



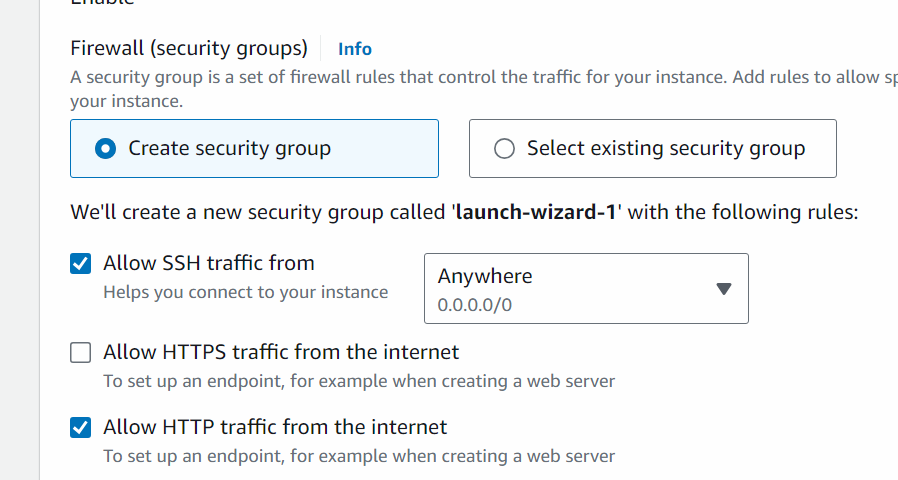
**Click on Create new key pair and create a new key pair**



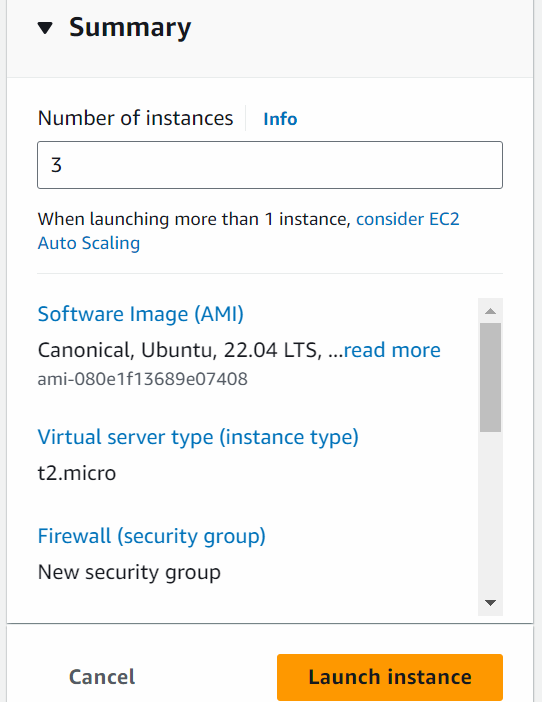


**The key would be downloaded to the machine**

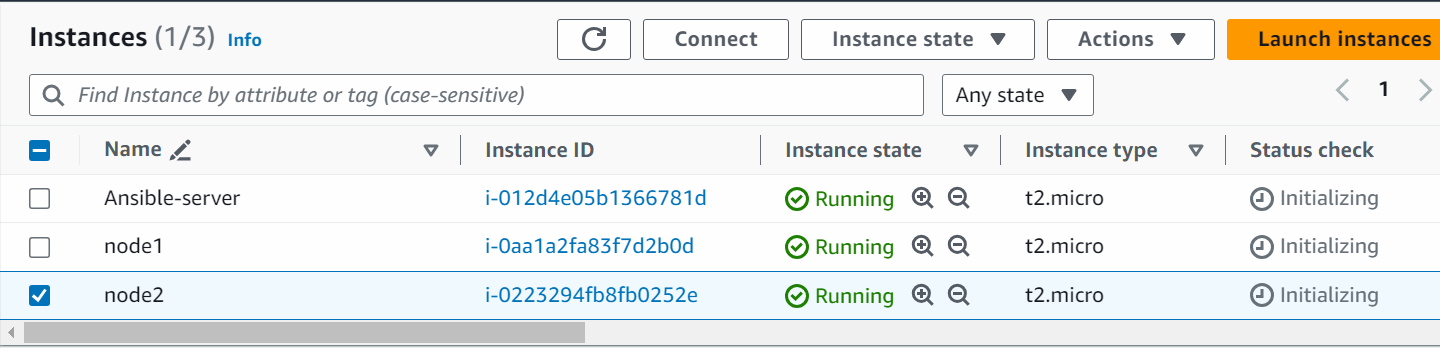
**In Firewall ,**



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



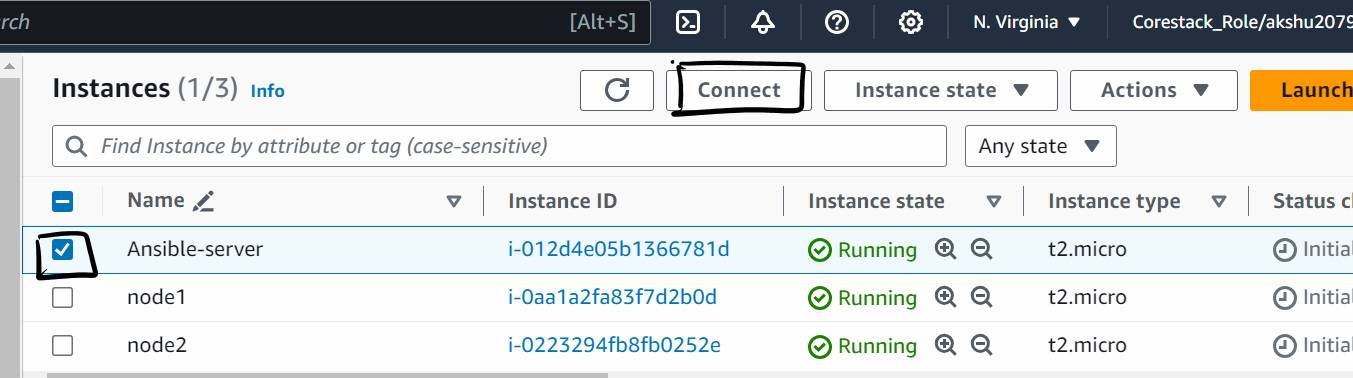
**After Machines are launched We can rename them as :**



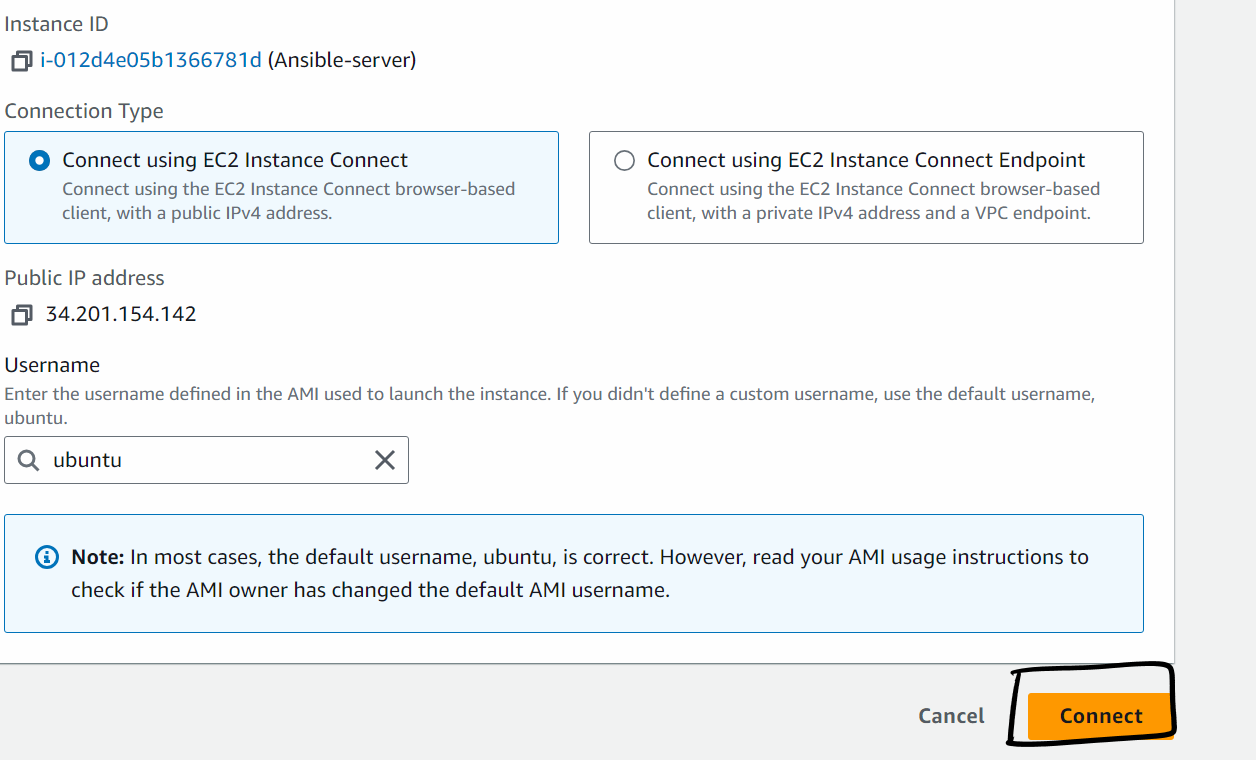
**Step 2: Now we will connect to these machines**

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 su

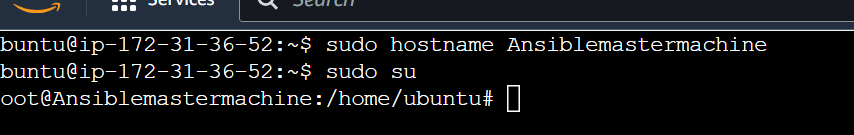
apt update -y

apt-get install -y software-properties-common

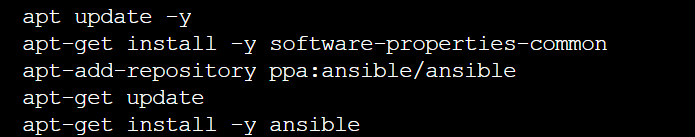
apt-add-repository ppa:ansible/ansible

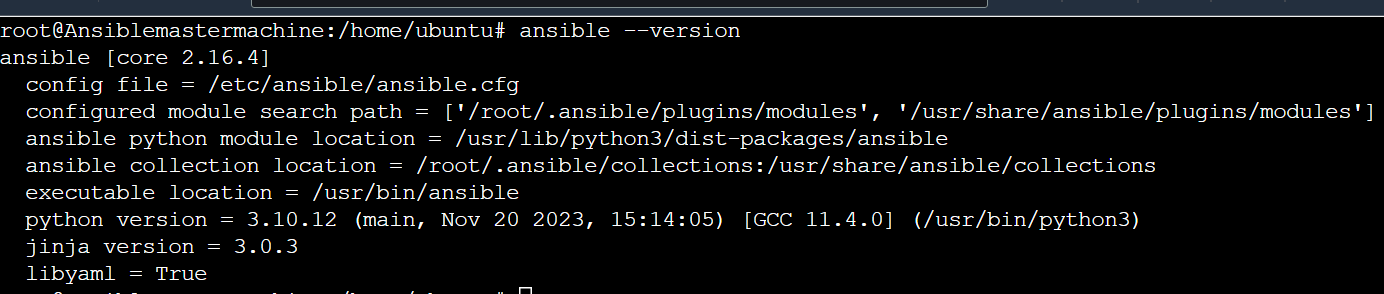
apt-get update

apt-get install -y ansible



# ansible --version





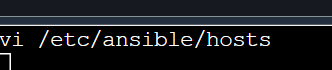
Step 4: we will now define the hosts from the master machine (ansible server)

vi /etc/ansible/hosts

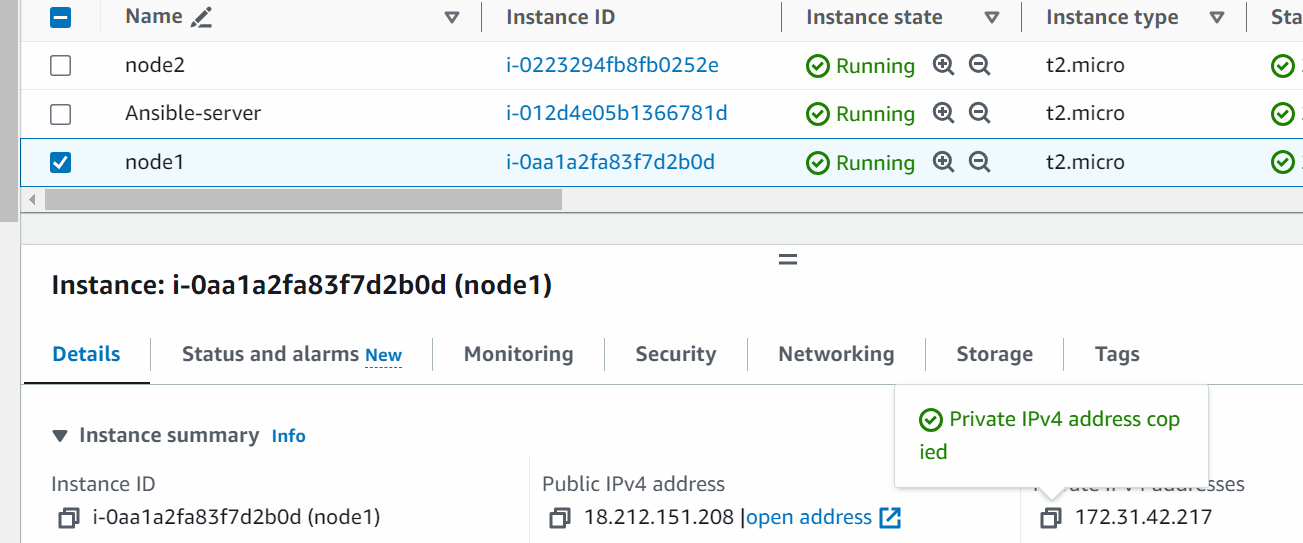
[ansiblegroup]

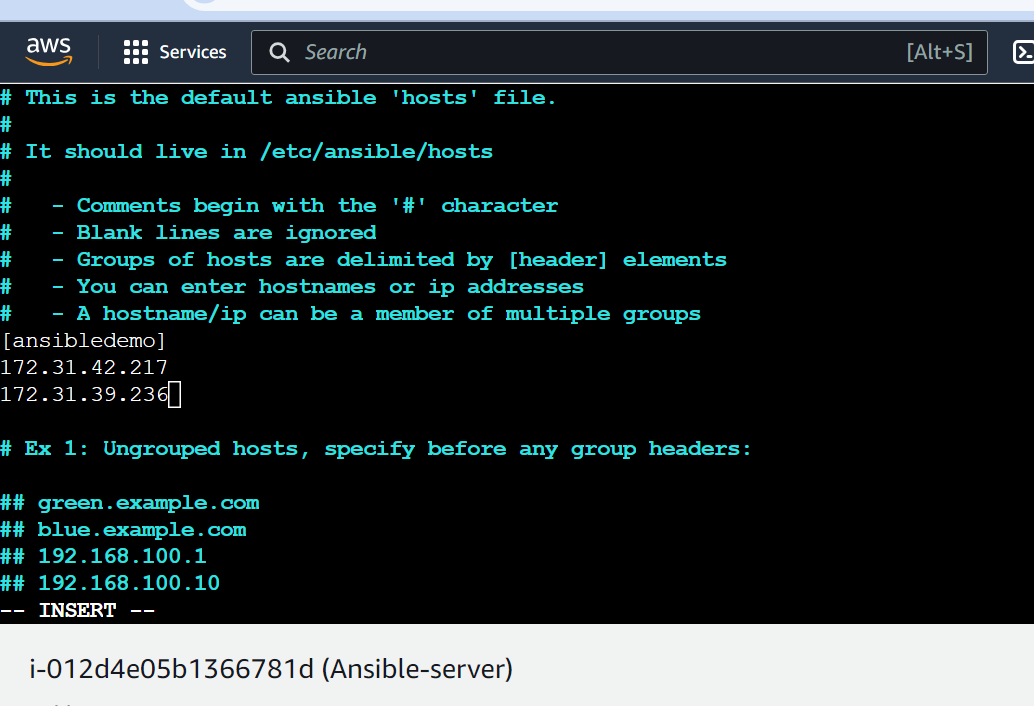
Privateip of node1

Privateip of node2



Copy the private ip of the node 1





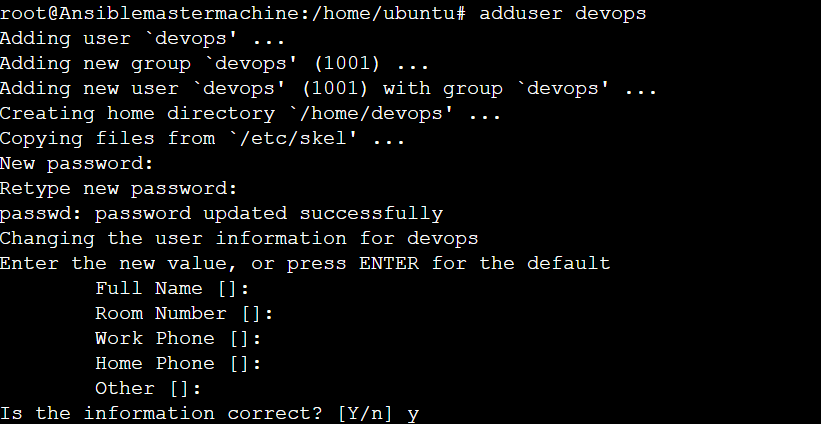
(we have to also copy the paste the private ip of the node2 as well)

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)

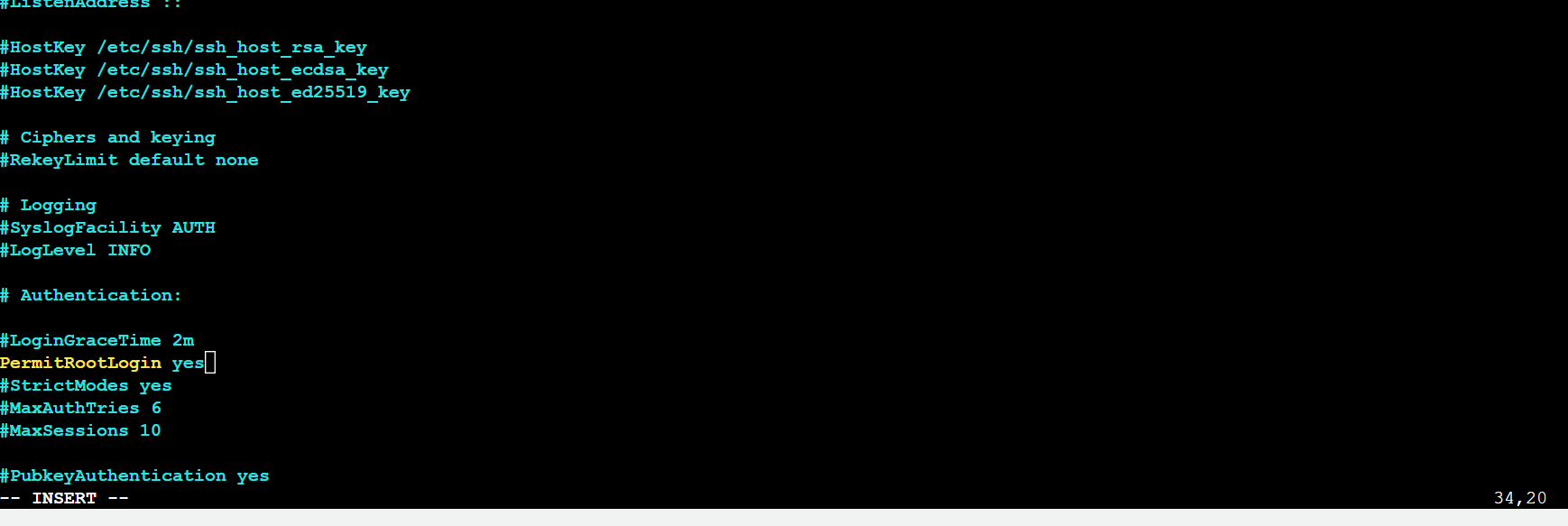
Step 6: We will now configure sshd configuration in master and node machines

# vi /etc/ssh/sshd\_config

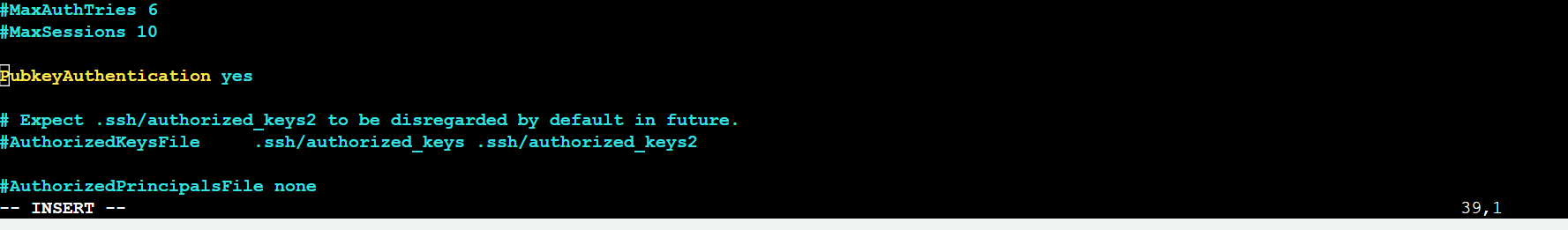
press i



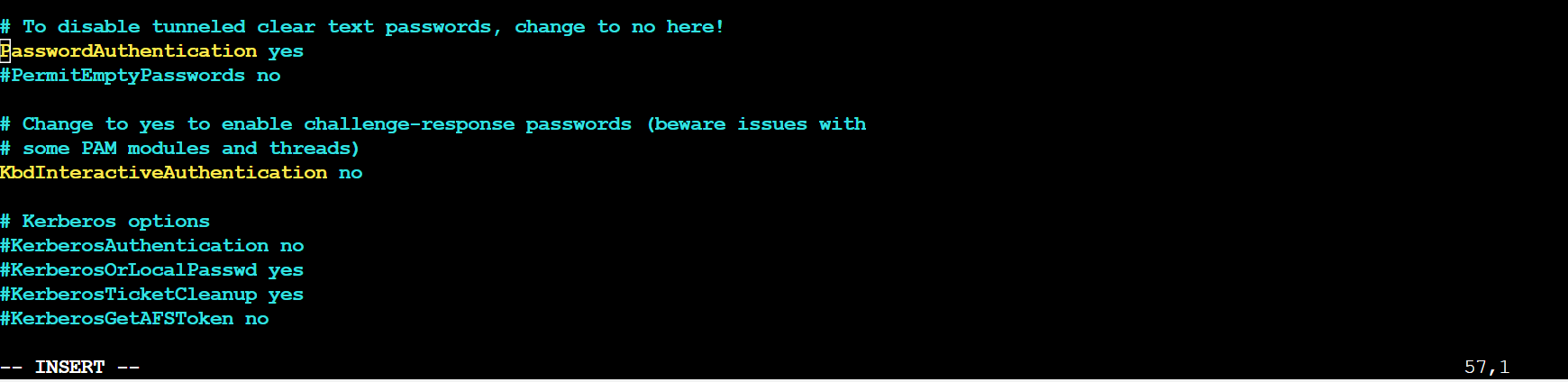
On line 34 change to PermitRootlogin yes and remove #



Remove # from line 38



On line 57 enable PasswordAuthentication as yes by removing #

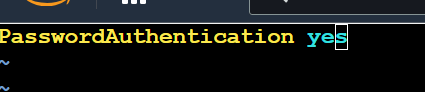


(perform same activity in node1 and node2 as well)

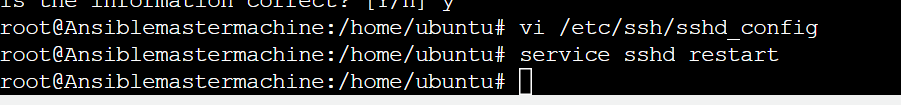
(with latest ubuntu these steps need to be added up in master and nodes)

#vi /etc/ssh/sshd\_config.d/60-cloudimg-settings.conf

Change password authentication to yes



# service ssh restart



Step 6: 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

jenkins 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(we will generate a key in the master and paste it in the ansible nodes to establish the ssh connection w/o putting password) with the nodes

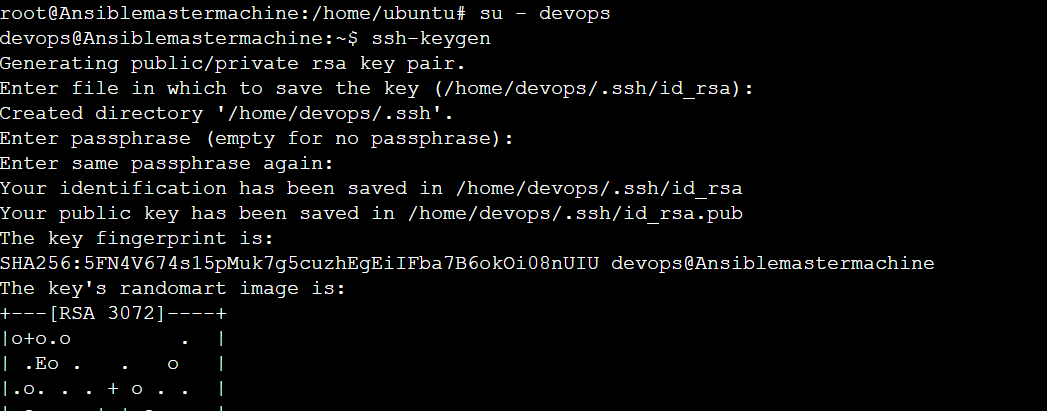
In Ansible-server (master machine):

# su – devops

# ssh-keygen

(above command generate the key in master machine)

(press enter three times)

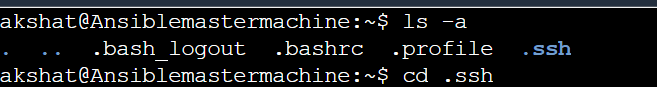


Now we need to copy the keypair in the node 1 and node2

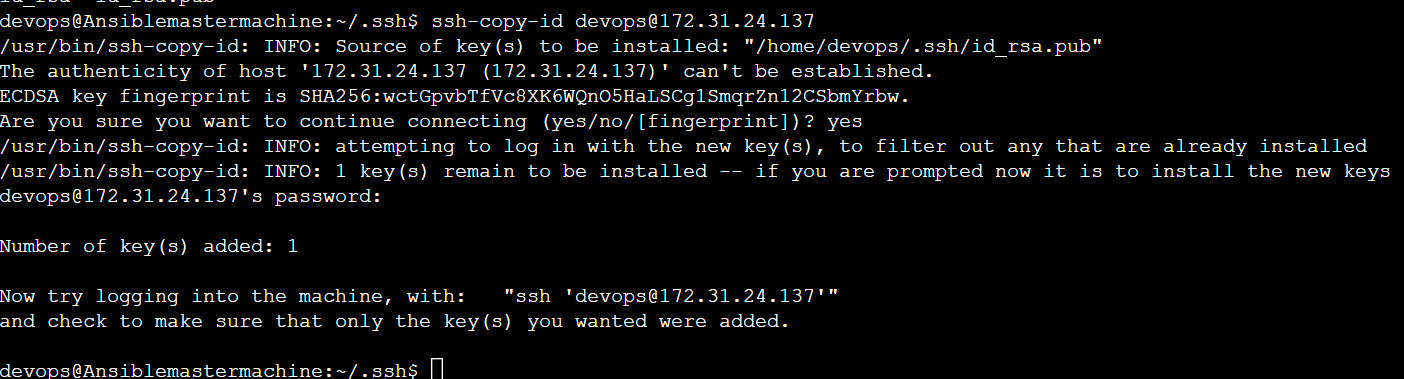
Go to Ansible-server(master)

# ls -a

# cd .ssh

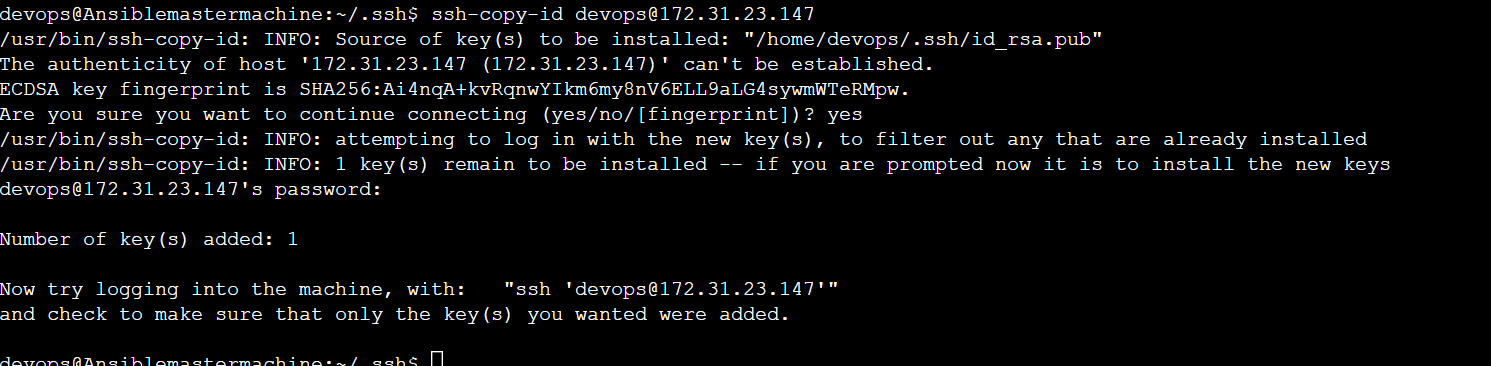


# ssh-copy-id devops@privateipofnode1



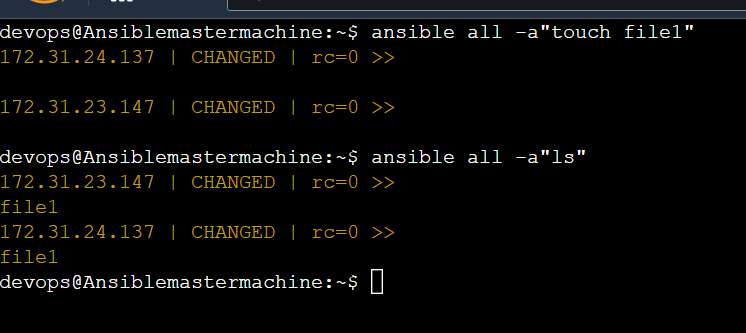
In the password put the password which we set while creating the user devops

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



Then for Jenkins make playbook Jenkins.yml

# vi Jenkins.yml

Now put this script in it

---

- hosts: #host name ex: [ansiblegroup]

become: true

tasks:

- name: Ensure wget is installed

ansible.builtin.package:

name: wget

state: present

- name: Download Jenkins installation script

ansible.builtin.get\_url:

url: "https://raw.githubusercontent.com/akshu20791/Deployment-script/refs/heads/main/jenkins.sh"

dest: "/home/devops/jenkins.sh"

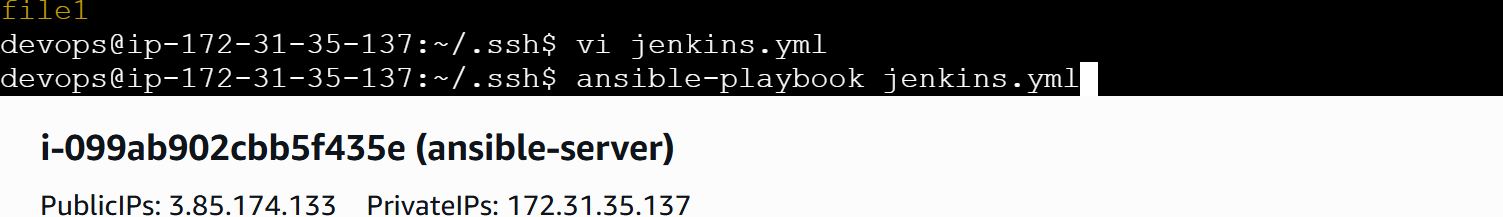
mode: '0755'

- name: Run Jenkins installation script

ansible.builtin.command: "bash /home/devops/jenkins.sh"

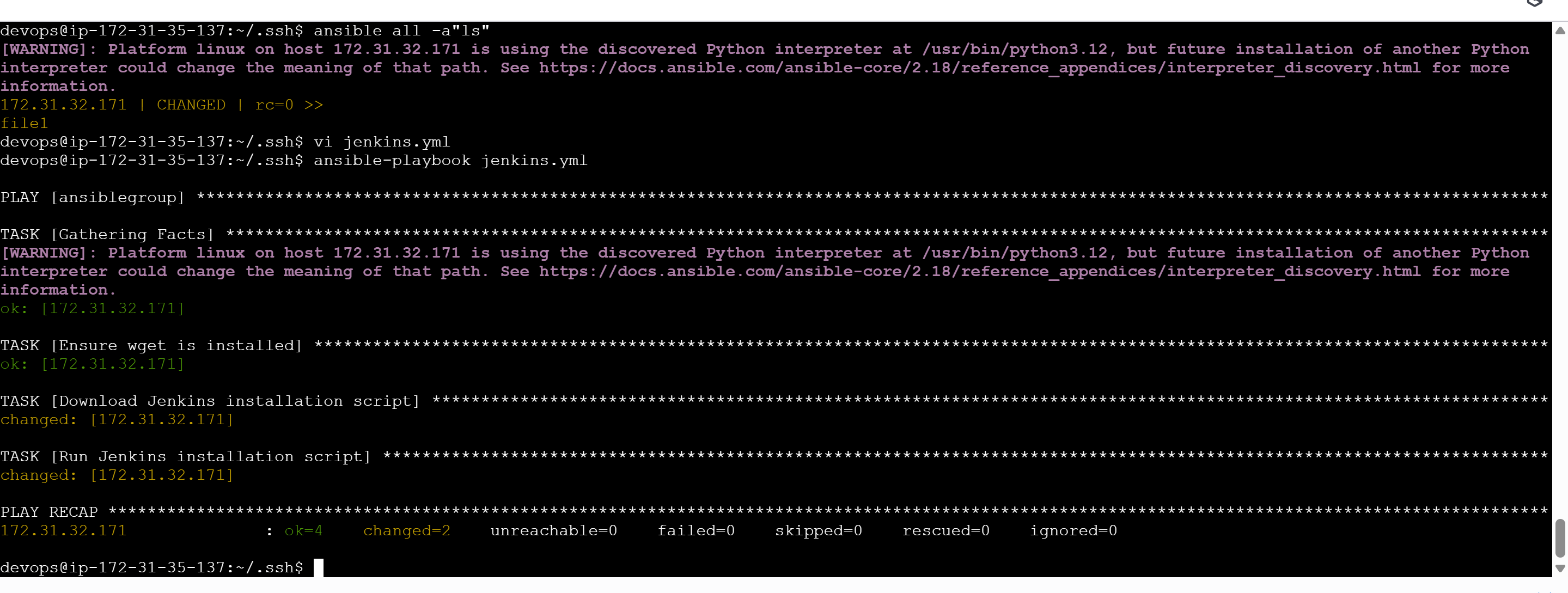
args:

chdir: /home/devops



Now run the playbook

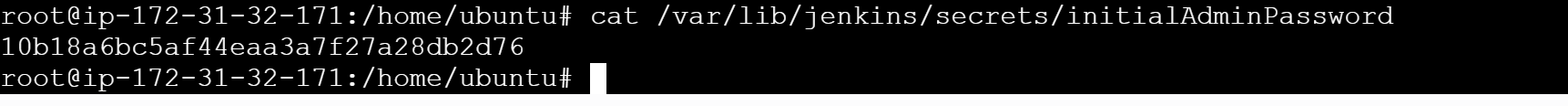
Ansible-playbook Jenkins.yml



Now sign in to Jenkins through node ip and :8080

For eg <http://3.89.205.195:8080/>

It will ask for password copy the link from Jenkins and paste to node (cat link)



Then install plugins

Then go to master node

Create a new playbook vi tomcat.yml

(write a code to install tomcat)

---

- name: install tomcat at node

hosts: ansiblegroup

become: yes

tasks:

- name: update package

apt:

update\_cache: true

- name: get tomact from url

ansible.builtin.get\_url:

url: https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.108/bin/apache-tomcat-9.0.108.zip

dest: /home/ubuntu

- name: get intall unzip packeg

apt:

name: unzip

state: present

- name: unzip package

ansible.builtin.unarchive:

src: /home/ubuntu/apache-tomcat-9.0.108.zip

dest: /home/ubuntu/

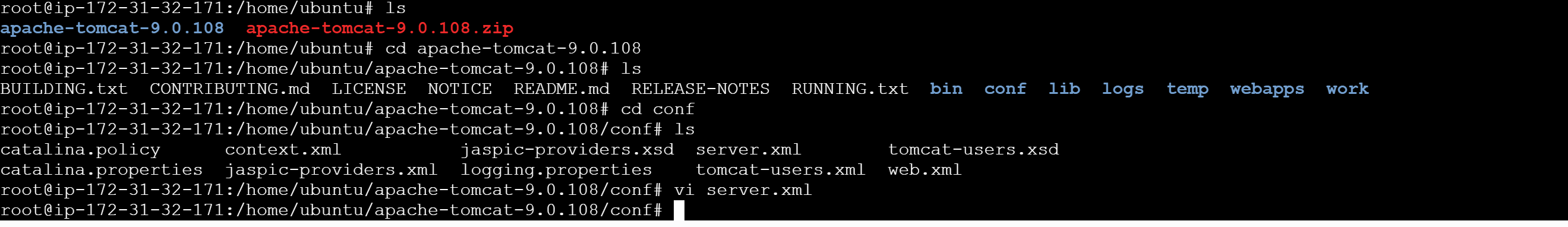
remote\_src: yes



Run the script

# ansible-playbook tomcat.yml

Now tomcat is installed now go to node



Ls

Cd apache tomcat

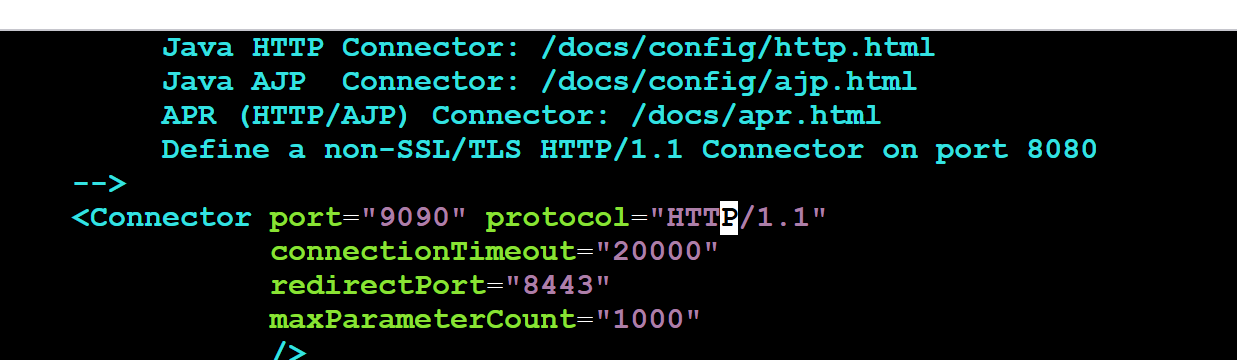
ls

Cd config

Ls

Vi server.xml

Now change the port number 8080 to 9090



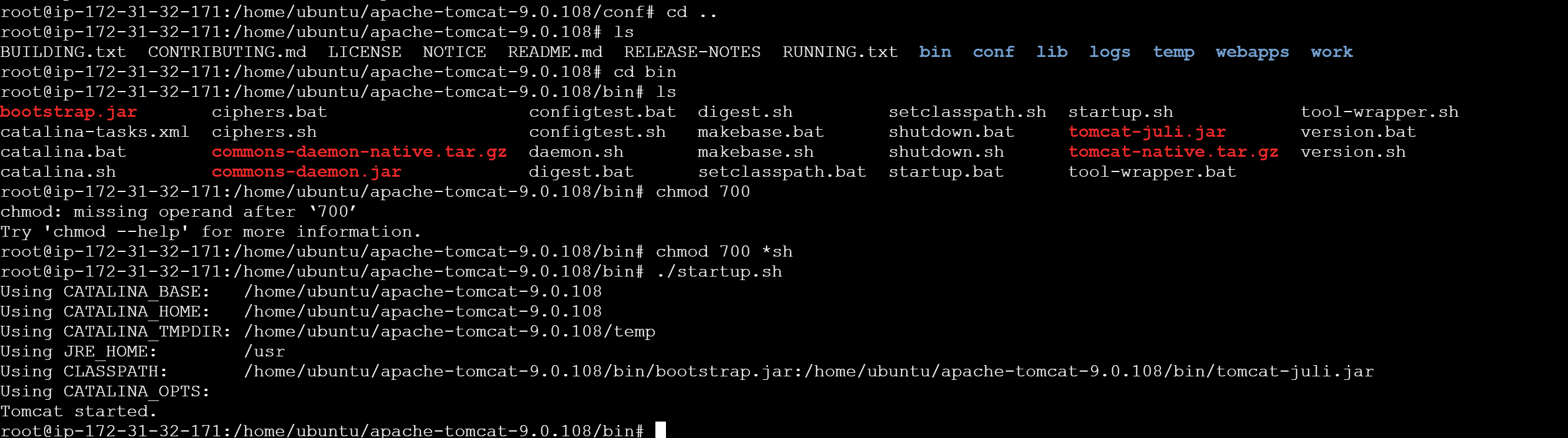
Cd ..

Ls

Cd bin

chmod 700 \*sh

./startup.sh



Now restart the Jenkins

Service Jenkins restart

Now go to Jenkins dashboard create new pipeline akash

pipeline{

agent any

stages{

stage("first stage: here we checkout the code from github"){

steps{

git 'https://github.com/akshu20791/addressbook-cicd-project'

echo "cloning the code in jenkins workspace"

}

}

stage("compile the code by akash"){

steps{

sh 'mvn compile'

echo "compiling the project"

}

}

stage("runnng the test case"){

steps{

sh 'mvn test'

}

}

stage("packaging the project"){

steps{

sh 'mvn package'

}

}

stage("depoy the project"){

steps{

sh 'sudo cp /var/lib/jenkins/workspace/akash/target/addressbook.war /home/ubuntu/apache-tomcat-9.0.108/webapps'

}

}

}

}

Now go to tomcat and /addressbook