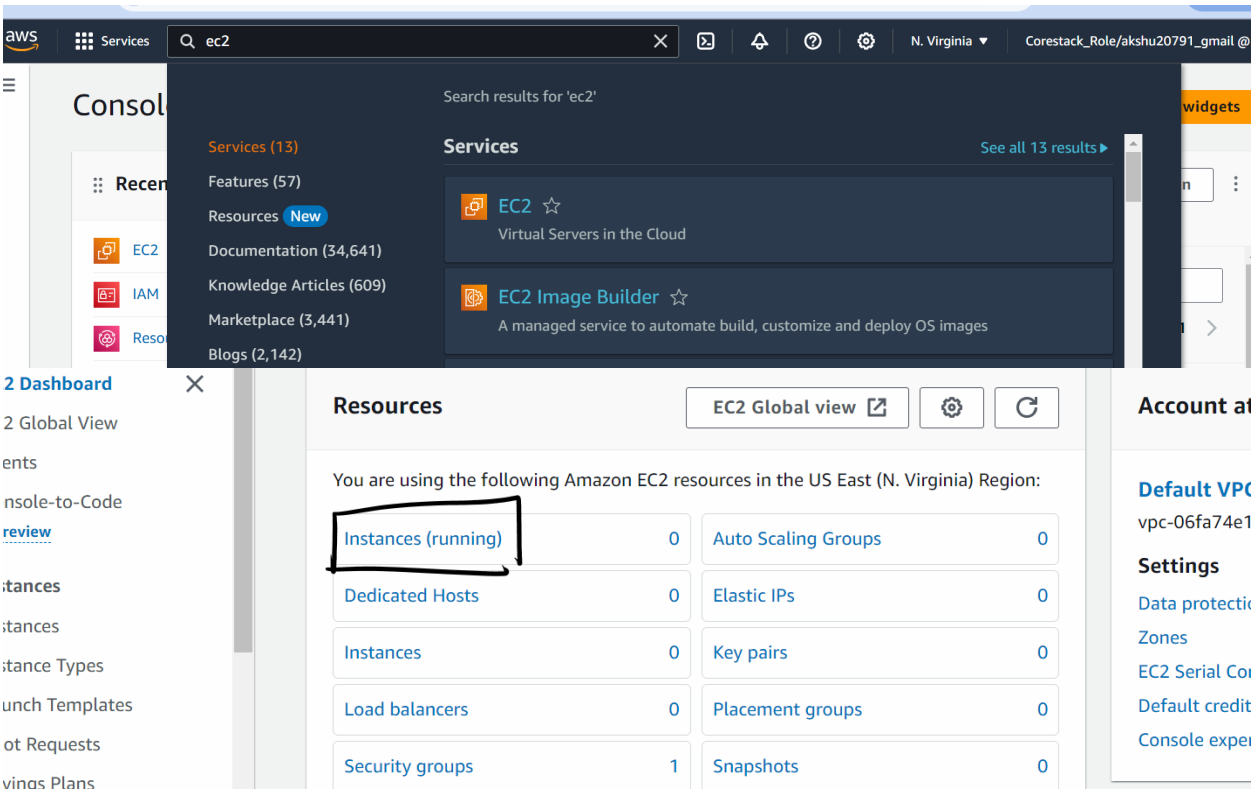
.

## Lesson 01 Demo 01

# INSTALLING AND CONFIGURING ANSIBLE

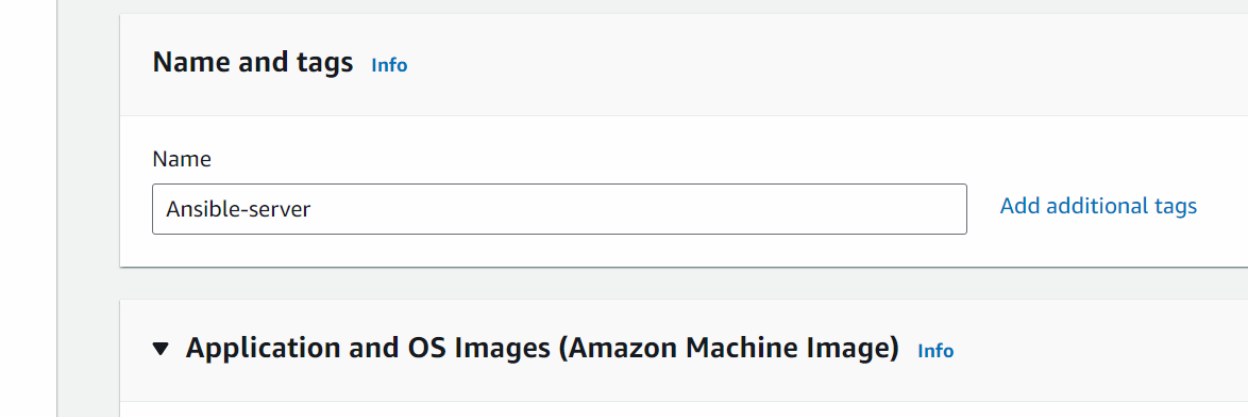
**Objective:** Using Ansible as a master node architecture in aws Ec2 machine
**Tools required:** Python , Ansible, AWS

**Prerequisites:** NA

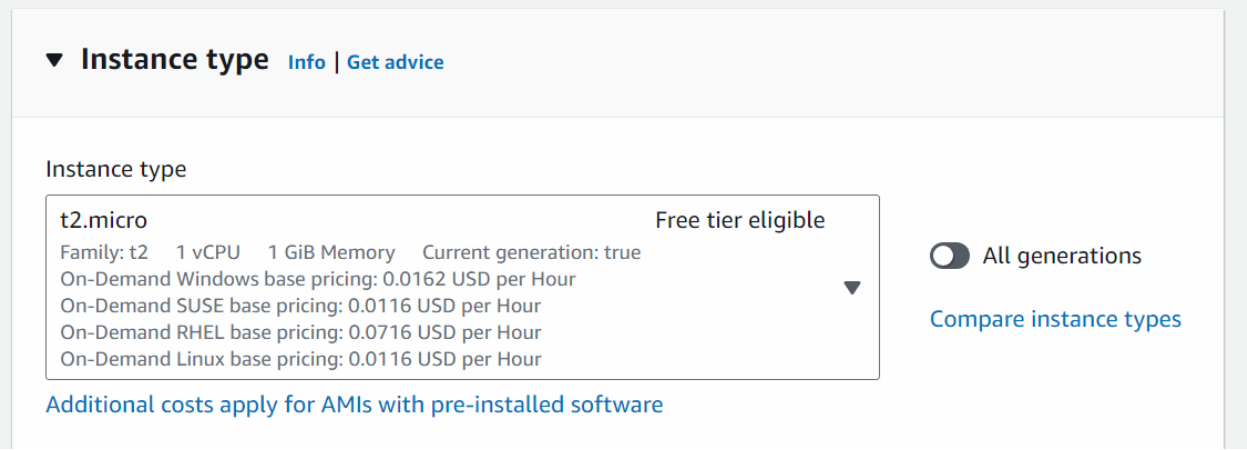



**Click on Launch instances**

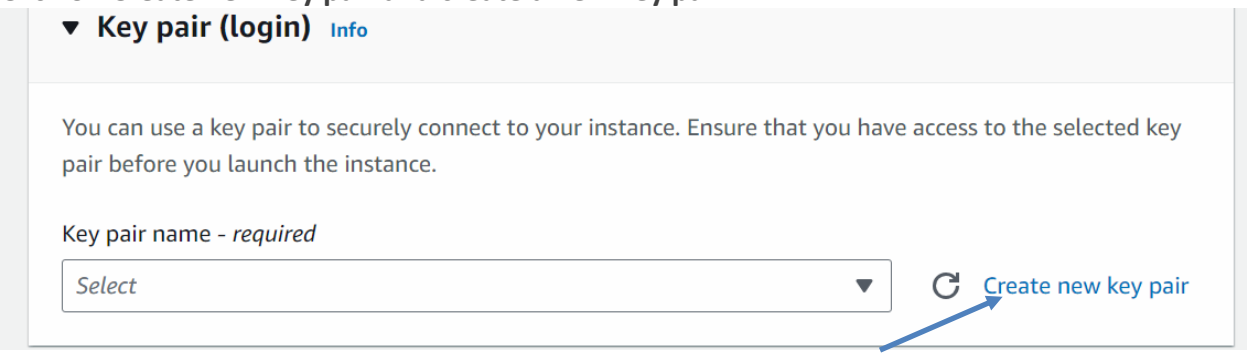
.



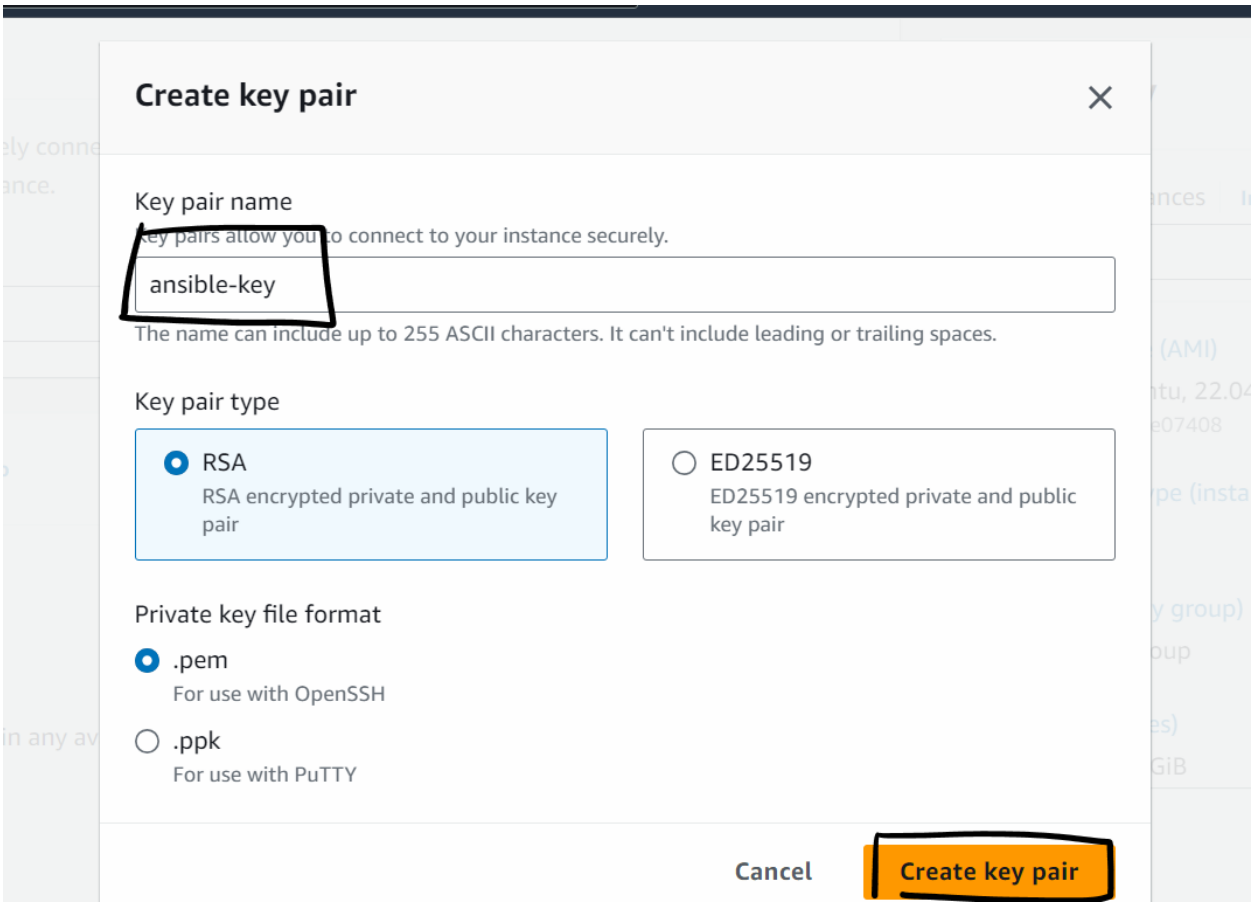

**Select AMI as Ubuntu 24.4**




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



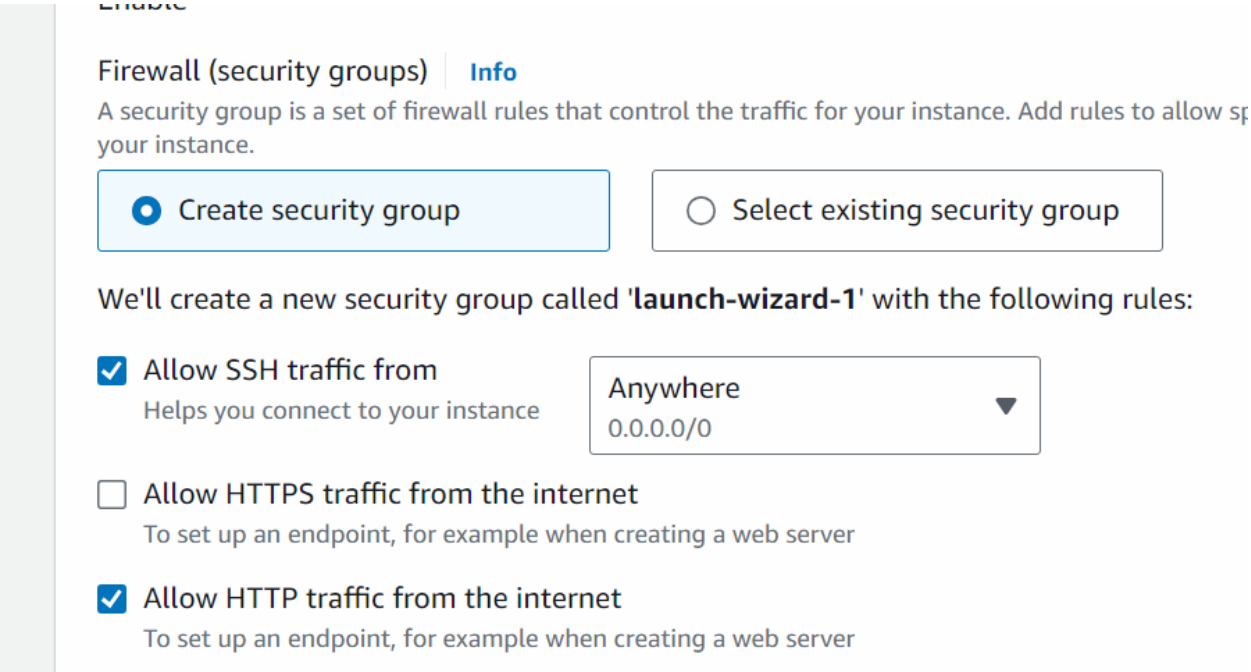
.



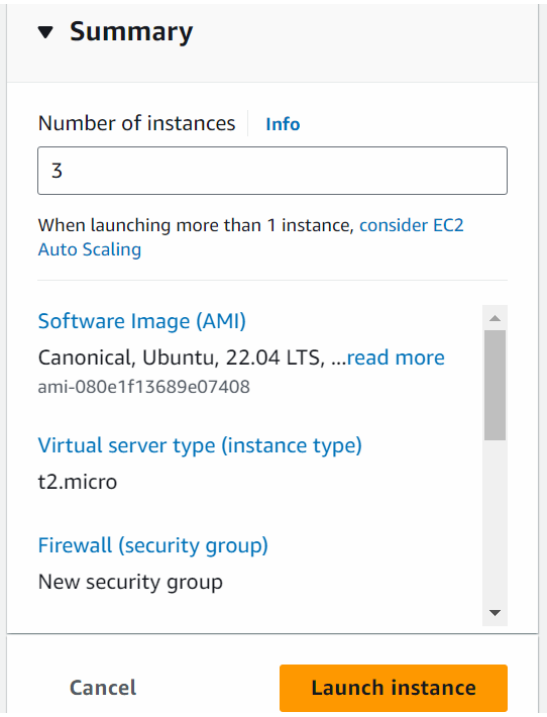

**The key would be downloaded to the machine**

**In Firewall ,**

.

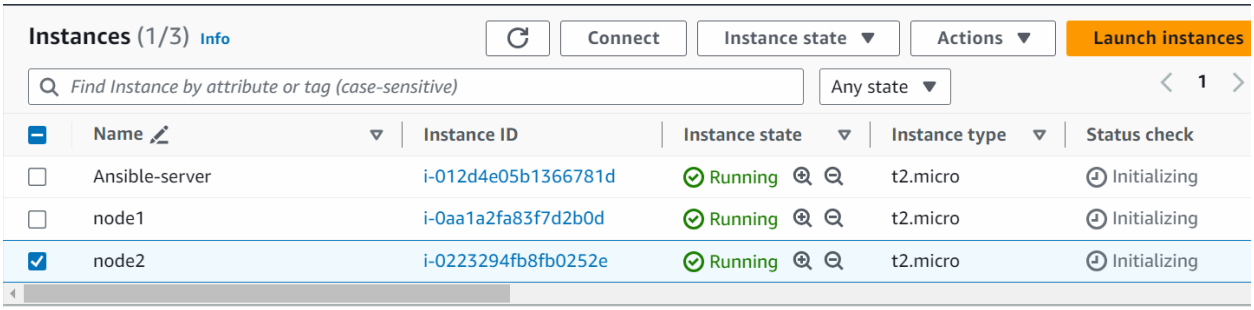



**Change the number of instances to 3 and launch the instances. We will consider one machine as master machine and other two as nodes**




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

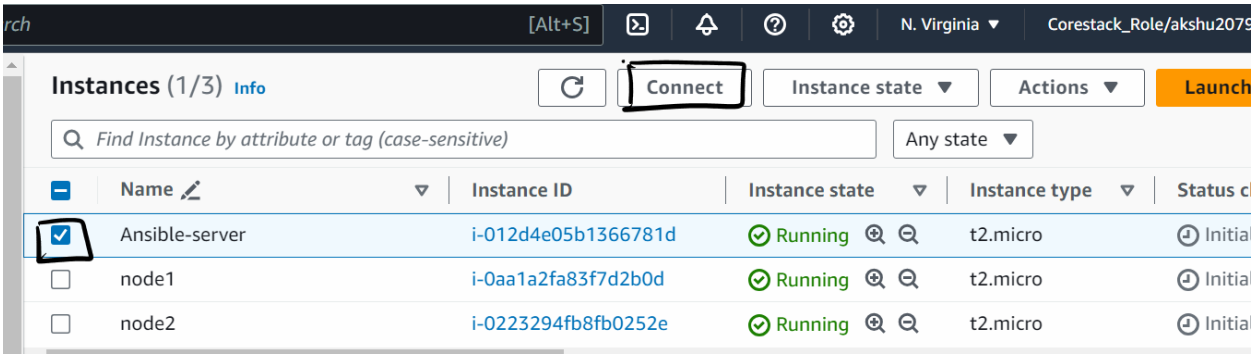
.




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

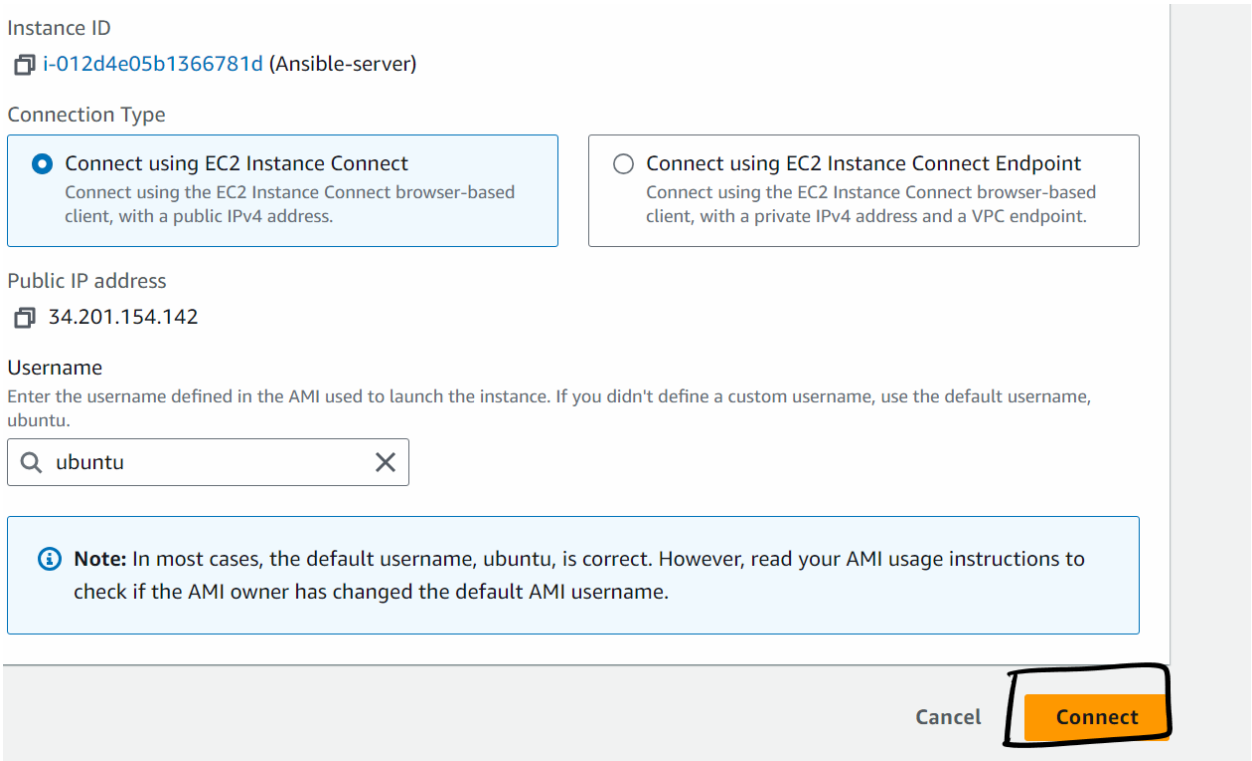
We can connect to the machine directly via browser

Select the ansible-server and click on connect

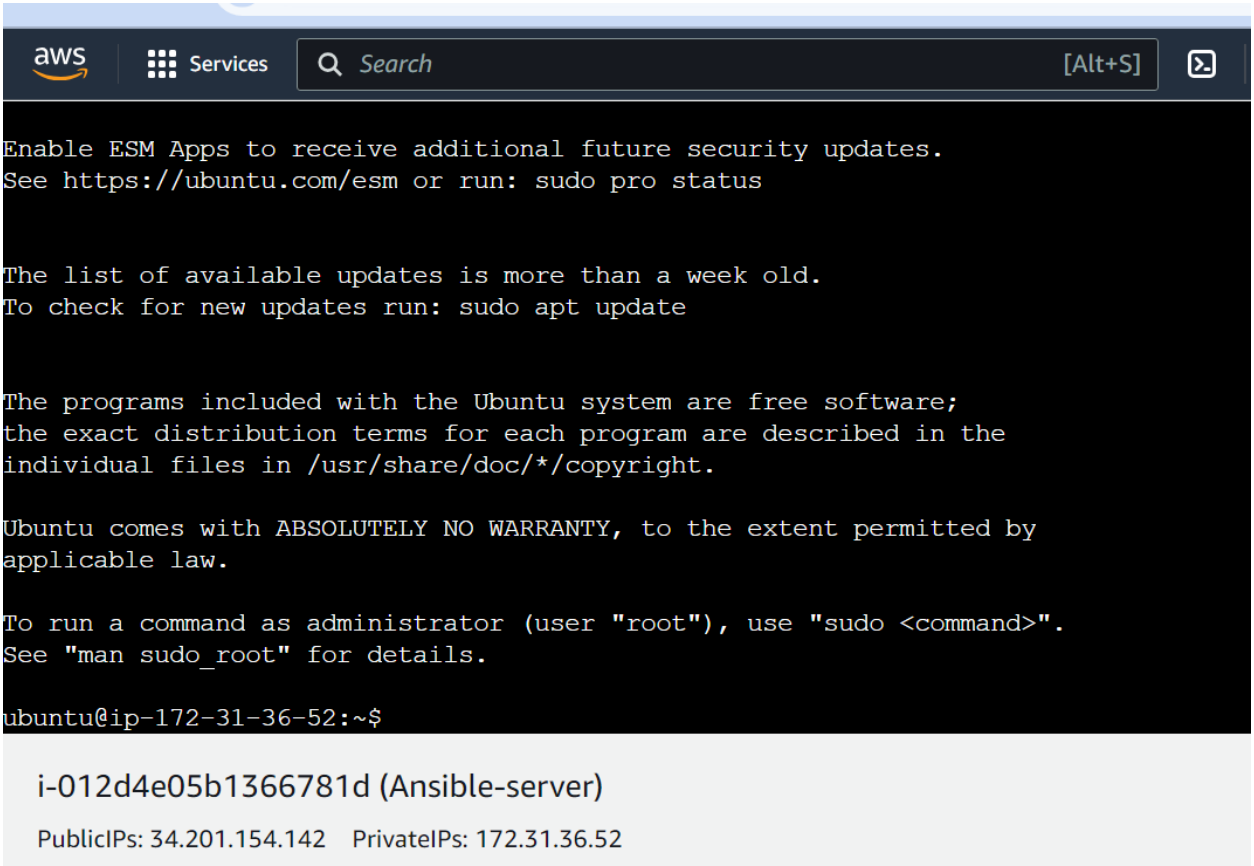



Click on Ec2 instance connect tab

.




Note: Do not change the username

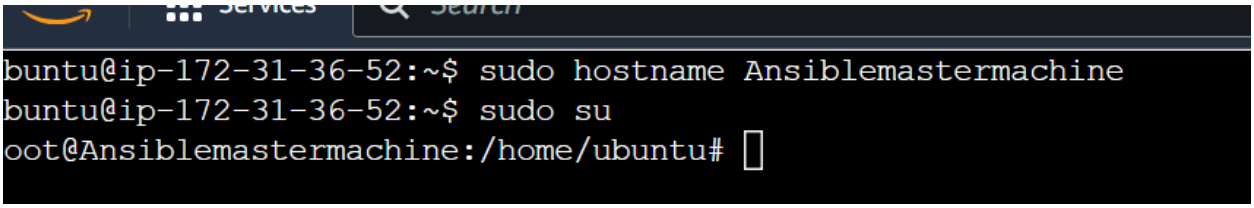


.

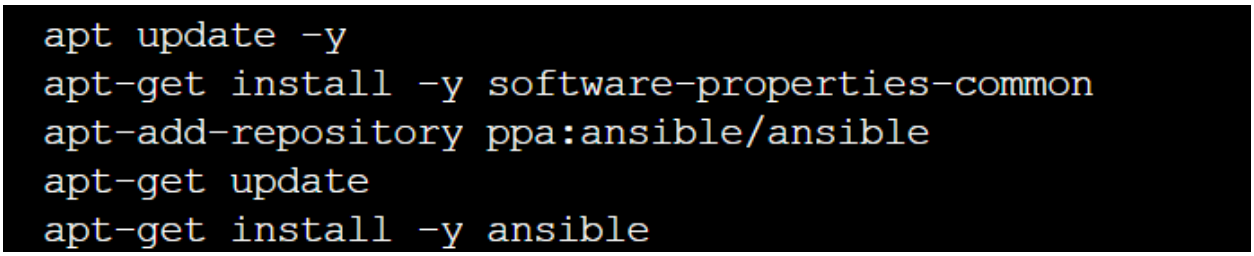
Similarly, We can connect with other machines as well

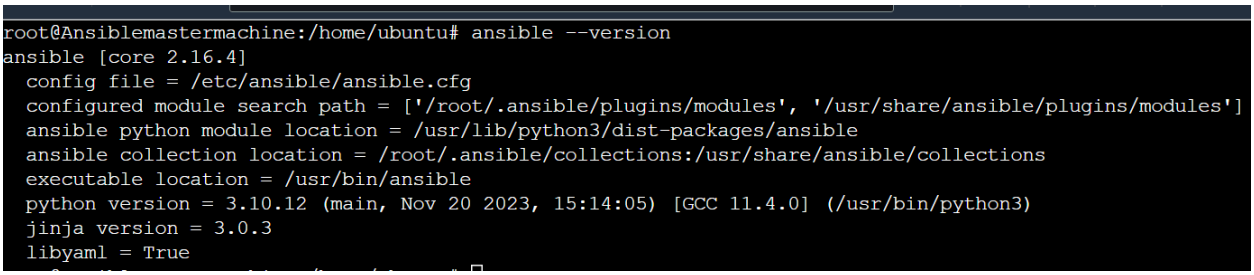
Step 3: lets now install ansible in ansible server (execute the below command only in master machine)

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

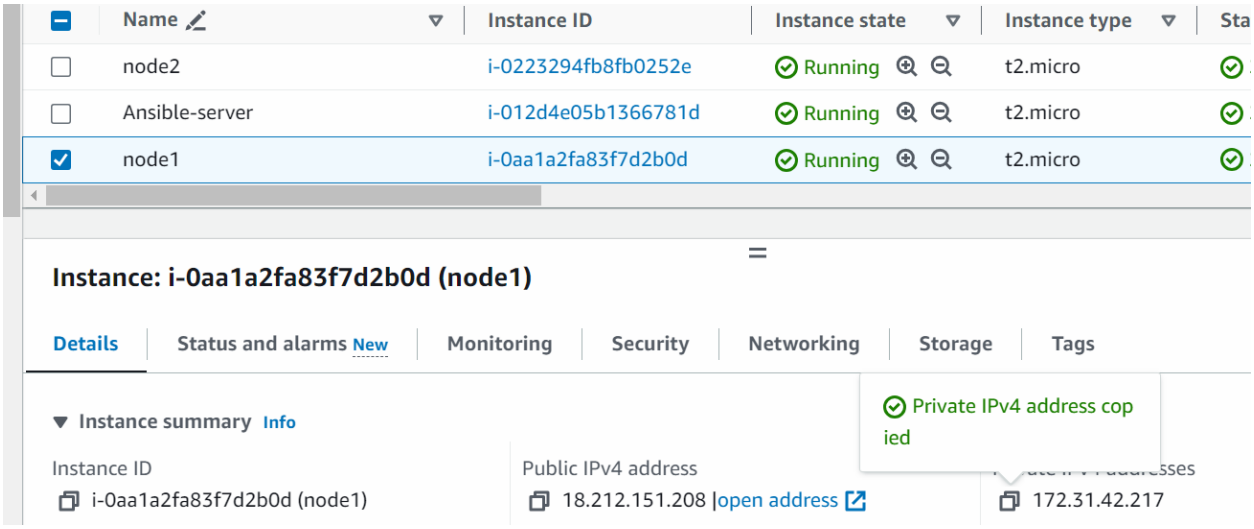
[ansiblegroup]

.

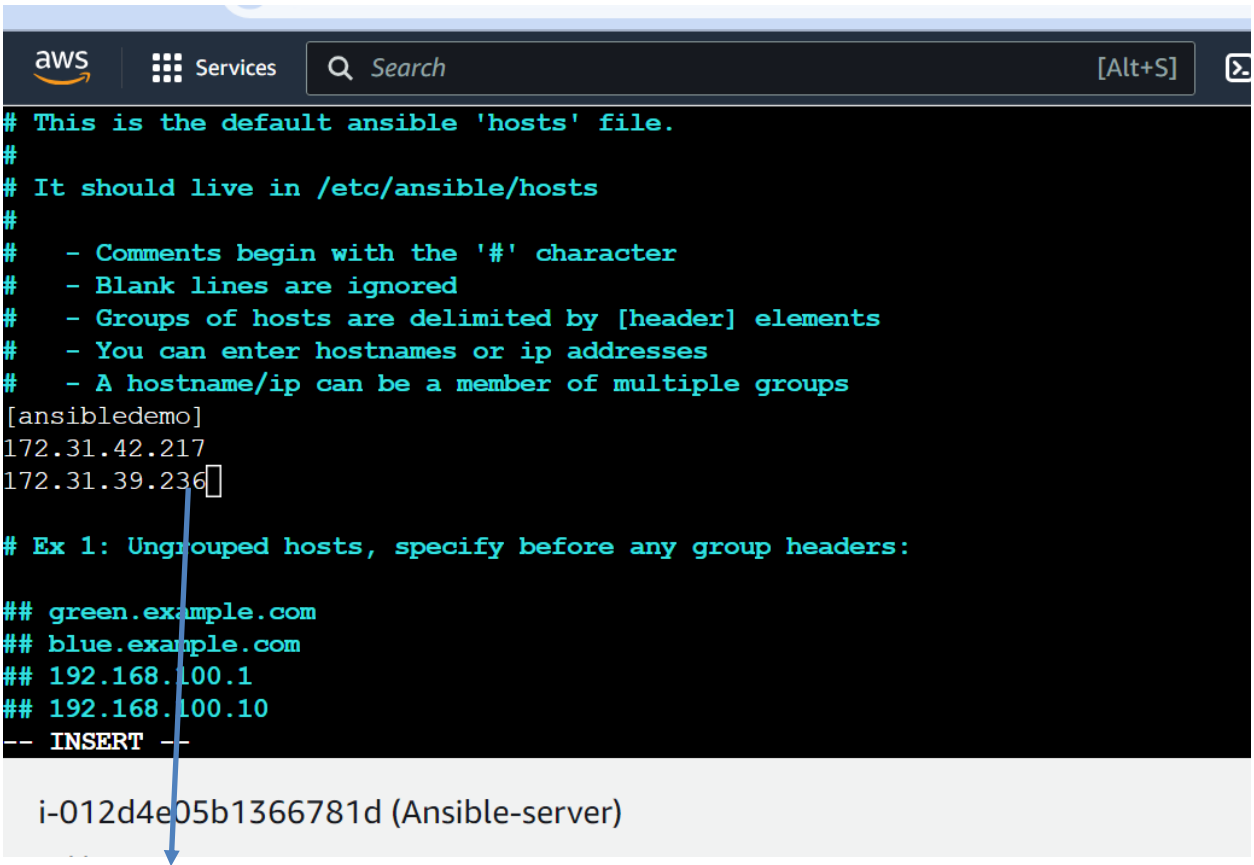
Privateip of node1
Privateip of node2

```
vi /etc/ansible/hosts
```

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

.

```
root@Ansiblemastermachine:/home/ubuntu# adduser devops
Adding user `devops' ...
Adding new group `devops' (1001) ...
Adding new user `devops' (1001) with group `devops' ...
Creating home directory `/home/devops' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for devops
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] 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

```
root@Ansiblemastermachine:/home/ubuntu# vi /etc/ssh/sshd_config
```

On line 34 change to PermitRootlogin yes and remove #

```
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
-- INSERT --                                                    34,20
```

Remove # from line 38

.

```
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
-- INSERT --                                                    39,1
```

On line 57 enable PasswordAuthentication as yes by removing #

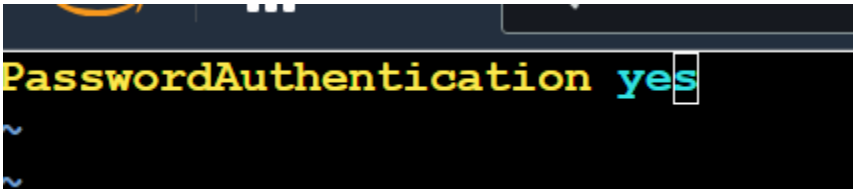
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

-- INSERT --                                                    57,1
```

(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

```
root@Ansiblemastermachine:/home/ubuntu# vi /etc/ssh/sshd_config
root@Ansiblemastermachine:/home/ubuntu# service sshd restart
root@Ansiblemastermachine:/home/ubuntu#
```

Step 6: We will now give sudo permission to the “devops” user in ansible and the nodes

# visudo

.

```
@Ansiblemastermachine:/home/ubuntu# visudo
```

Scroll down to user priviledge specification and add below line

devops ALL=(ALL:ALL) NOPASSWD:ALL
jenkins ALL=(ALL:ALL) NOPASSWD:ALL

```
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
devops  ALL=(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:
```

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)

.

```
root@Ansiblemastermachine:/home/ubuntu# su - devops
devops@Ansiblemastermachine:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/devops/.ssh/id_rsa):
Created directory '/home/devops/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devops/.ssh/id_rsa
Your public key has been saved in /home/devops/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5FN4V674s15pMuk7g5cuzhEgEiIFba7B6okOi08nUIU devops@Ansiblemastermachine
The key's randomart image is:
+---[RSA 3072]----+
|o+o.o         .  |
| .Eo .   .   o   |
|.o. . . + o . .  |
```

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

Go to Ansible-server(master)

# ls -a
# cd .ssh

```
akshat@Ansiblemastermachine:~$ ls -a
.  ..  .bash_logout  .bashrc  .profile  .ssh
akshat@Ansiblemastermachine:~$ cd .ssh
```

# ssh-copy-id devops@privateipofnode1

```
devops@Ansiblemastermachine:~/.ssh$ ssh-copy-id devops@172.31.24.137
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_rsa.pub"
The authenticity of host '172.31.24.137 (172.31.24.137)' can't be established.
ECDSA key fingerprint is SHA256:wctGpvbTfVc8XK6WQnO5HaLSCg1SmqrZn12CSbmYrbw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 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
devops@172.31.24.137's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'devops@172.31.24.137'"
and check to make sure that only the key(s) you wanted were added.

devops@Ansiblemastermachine:~/.ssh$
```

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

Similarly copy to the node2 as well

.

```
devops@Ansiblemastermachine:~/.ssh$ ssh-copy-id devops@172.31.23.147
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/devops/.ssh/id_rsa.pub"
The authenticity of host '172.31.23.147 (172.31.23.147)' can't be established.
ECDSA key fingerprint is SHA256:Ai4nqA+kvRqnwYIkm6my8nV6ELL9aLG4sywmWTeRMpw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 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
devops@172.31.23.147's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'devops@172.31.23.147'"
and check to make sure that only the key(s) you wanted were added.
```

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

```
devops@Ansiblemastermachine:~$ ansible all -a"touch file1"
172.31.24.137 | CHANGED | rc=0 >>

172.31.23.147 | CHANGED | rc=0 >>

devops@Ansiblemastermachine:~$ ansible all -a"ls"
172.31.23.147 | CHANGED | rc=0 >>
file1
172.31.24.137 | CHANGED | rc=0 >>
file1
devops@Ansiblemastermachine:~$
```

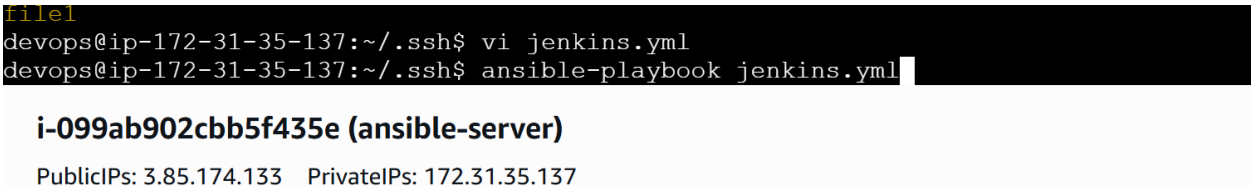
.

Then for Jenkins make playbook Jenkins.yml

# vi Jenkins.yml

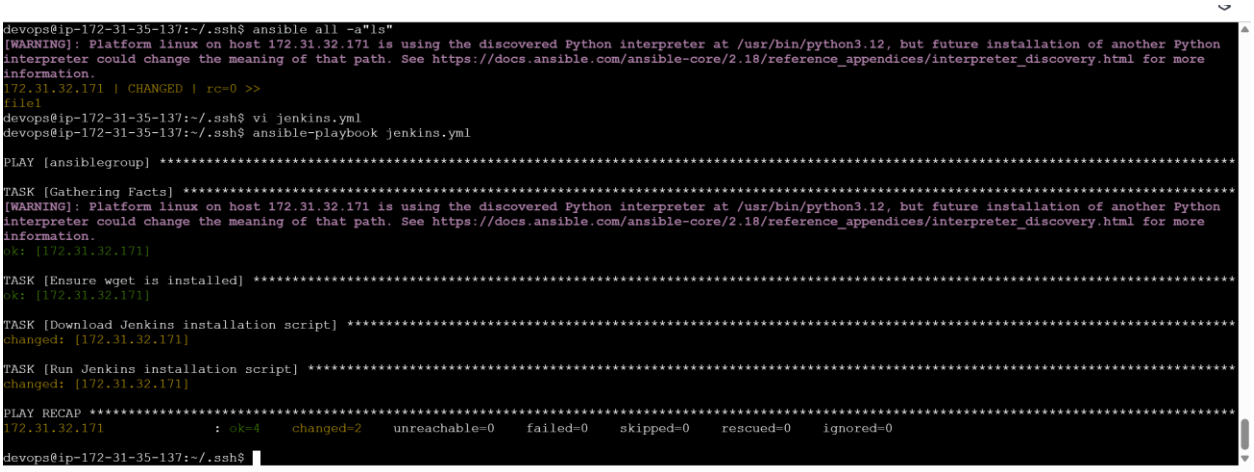
Now put this script in it

```
---
- 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/](http://3.89.205.195:8080/)

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

```
root@ip-172-31-32-171:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
10b18a6bc5af44eaa3a7f27a28db2d76
root@ip-172-31-32-171:/home/ubuntu#
```

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
```

.

```
---
- 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

~
~
~
```

**i-099ab902cbb5f435e (ansible-server)**

PublicIPs: 3.85.174.133 PrivateIPs: 172.31.35.137

Run the script
# ansible-playbook tomcat.yml

Now tomcat is installed now go to node

```
root@ip-172-31-32-171:/home/ubuntu# ls
apache-tomcat-9.0.108  apache-tomcat-9.0.108.zip
root@ip-172-31-32-171:/home/ubuntu# cd apache-tomcat-9.0.108
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108# cd conf
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/conf# ls
catalina.policy      context.xml          jaspic-providers.xsd  server.xml       tomcat-users.xsd
catalina.properties  jaspic-providers.xml  logging.properties    tomcat-users.xml  web.xml
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/conf# vi server.xml
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/conf#
```

Ls
Cd apache tomcat
ls
Cd config
Ls
Vi server.xml

Now change the port number 8080 to 9090

.

```
        Java HTTP Connector: /docs/config/http.html
        Java AJP  Connector: /docs/config/ajp.html
        APR (HTTP/AJP) Connector: /docs/apr.html
        Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
    -->
    <Connector port="9090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443"
               maxParameterCount="1000"
               />
```

Cd ..
Ls
Cd bin
chmod 700 *sh
./startup.sh

```
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/conf# cd ..
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108# ls
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108# cd bin
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/bin# ls
bootstrap.jar       ciphers.bat                 configtest.bat  digest.sh         setclasspath.sh  startup.sh            tool-wrapper.sh
catalina-tasks.xml  ciphers.sh                  configtest.sh   makebase.bat      shutdown.bat     tomcat-juli.jar       version.bat
catalina.bat        commons-daemon-native.tar.gz  daemon.sh       makebase.sh       shutdown.sh      tomcat-native.tar.gz  version.sh
catalina.sh         commons-daemon.jar          digest.bat      setclasspath.bat  startup.bat      tool-wrapper.bat
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/bin# chmod 700
chmod: missing operand after '700'
Try 'chmod --help' for more information.
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/bin# chmod 700 *sh
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/bin# ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.108
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.108
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.108/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.108/bin/bootstrap.jar:/home/ubuntu/apache-tomcat-9.0.108/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@ip-172-31-32-171:/home/ubuntu/apache-tomcat-9.0.108/bin#
```

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