

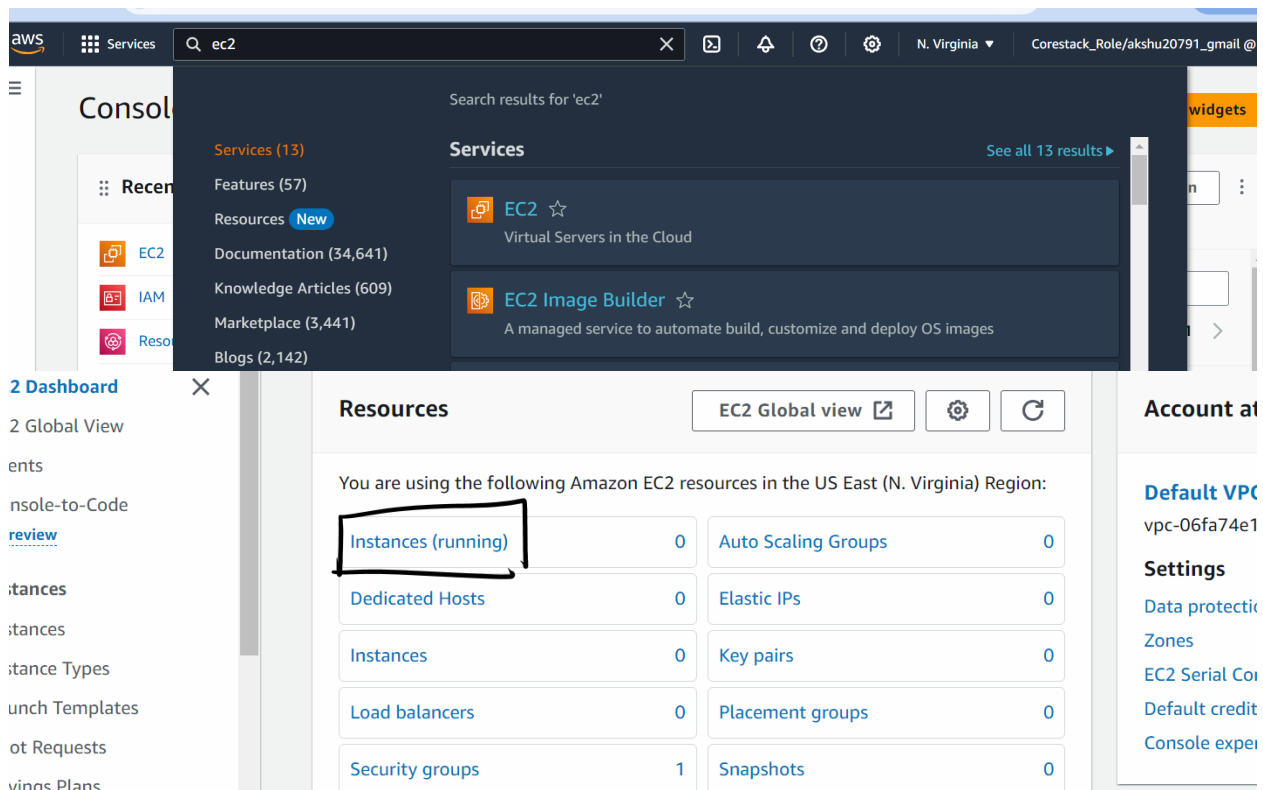
## Lesson 01 Demo 01

# 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

**Name and tags** [Info](#)

Name

Ansible-server

Add additional tags

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

Select AMI as Ubuntu 24.4

**▼ Instance type** [Info](#) | [Get advice](#)

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

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

**▼ Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

↕

Create new key pair

### Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

ansible-key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

Cancel

Create key pair

The key would be downloaded to the machine

In Firewall ,

Launch wizard

### Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☒ Allow SSH traffic from  
Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet  
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet  
To set up an endpoint, for example when creating a web server

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

#### ▼ Summary

Number of instances [Info](#)

3

When launching more than 1 instance, [consider EC2 Auto Scaling](#)

#### Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-080e1f13689e07408

#### Virtual server type (instance type)

t2.micro

#### Firewall (security group)

New security group

Cancel

Launch instance

After Machines are launched We can rename them as :

Instances (1/3) <a href="#">Info</a>						<input type="button" value="Refresh"/> <input type="button" value="Connect"/> <input type="button" value="Instance state ▼"/> <input type="button" value="Actions ▼"/> <input type="button" value="Launch instances"/>	
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>						Any state ▼	< 1 >
<input type="checkbox"/>	Name <a href="#">✎</a>	Instance ID	Instance state	Instance type	Status check		
<input type="checkbox"/>	Ansible-server	i-012d4e05b1366781d	✓ Running <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	🕒 Initializing		
<input type="checkbox"/>	node1	i-0aa1a2fa83f7d2b0d	✓ Running <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	🕒 Initializing		
<input checked="" type="checkbox"/>	node2	i-0223294fb8fb0252e	✓ Running <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	🕒 Initializing		

## 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

rch

[Alt+S]

N. Virginia

Corestack\_Role/akshu2075

Instances (1/3) Info

Connect

Instance state

Actions

Launch

Find Instance by attribute or tag (case-sensitive)

Any state

	Name	Instance ID	Instance state	Instance type	Status c
	Ansible-server	i-012d4e05b1366781d	Running	t2.micro	Initial
	node1	i-0aa1a2fa83f7d2b0d	Running	t2.micro	Initial
	node2	i-0223294fb8fb0252e	Running	t2.micro	Initial

Click on Ec2 instance connect tab

Instance ID

 i-012d4e05b1366781d (Ansible-server)

Connection Type

☒ **Connect using EC2 Instance Connect**  
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.


☐ **Connect using EC2 Instance Connect Endpoint**  
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address


 34.201.154.142

Username


Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

 ubuntu



 **Note:** In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel



Note: Do not change the username

```
aws Services Search [Alt+S]

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

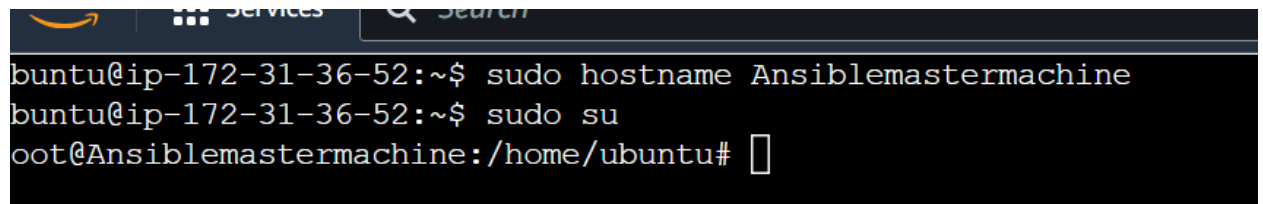
ubuntu@ip-172-31-36-52:~$

i-012d4e05b1366781d (Ansible-server)
PublicIPs: 34.201.154.142 PrivateIPs: 172.31.36.52
```

Similarly, We can connect with other machines as well

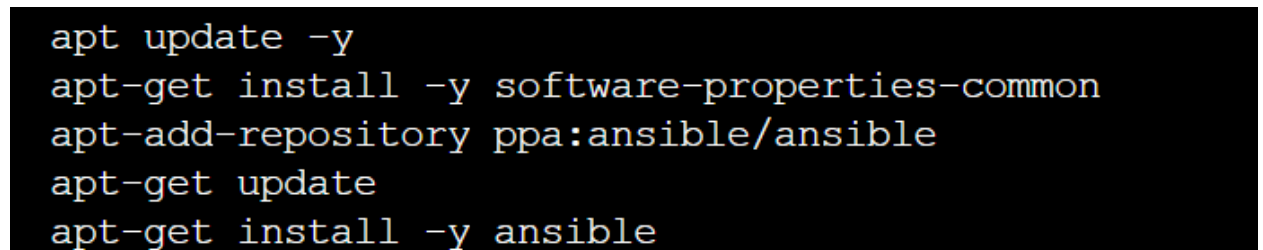
Step 3: lets now install ansible in ansible server (execute the below command only in master machine)

```
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
```

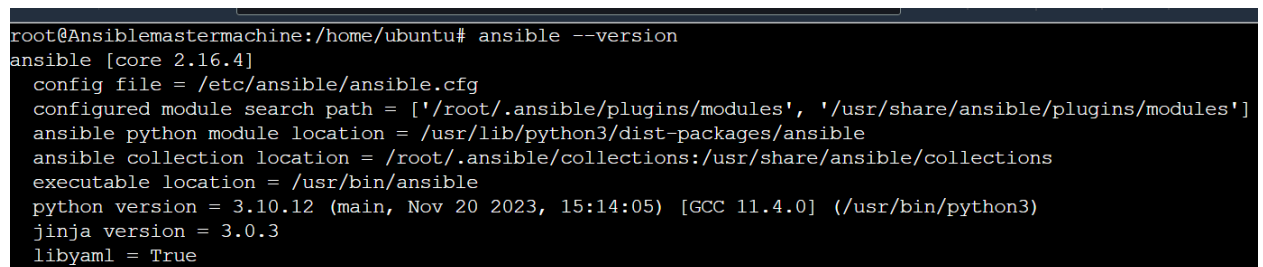


```
buntu@ip-172-31-36-52:~$ sudo hostname Ansiblemastermachine
buntu@ip-172-31-36-52:~$ sudo su
oot@Ansiblemastermachine:/home/ubuntu#
```

# ansible --version



```
apt update -y
apt-get install -y software-properties-common
apt-add-repository ppa:ansible/ansible
apt-get update
apt-get install -y ansible
```



```
root@Ansiblemastermachine:/home/ubuntu# ansible --version
ansible [core 2.16.4]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

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

	Name	Instance ID	Instance state	Instance type	Sta
<input type="checkbox"/>	node2	i-0223294fb8fb0252e	Running	t2.micro	
<input type="checkbox"/>	Ansible-server	i-012d4e05b1366781d	Running	t2.micro	
<input checked="" type="checkbox"/>	node1	i-0aa1a2fa83f7d2b0d	Running	t2.micro	

### Instance: i-0aa1a2fa83f7d2b0d (node1)

DetailsStatus and alarmsNewMonitoringSecurityNetworkingStorageTags

▼ Instance summary Info

Instance ID

i-0aa1a2fa83f7d2b0d (node1)

Public IPv4 address

18.212.151.208 [open address](#)

Private IPv4 address

172.31.42.217

Private IPv4 address copied



```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
[ansibledemo]
172.31.42.217
172.31.39.236
# Ex 1: Ungrouped hosts, specify before any group headers:
## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10
-- INSERT --
```

i-012d4e05b1366781d (Ansible-server)

(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

```

Similarly 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 #

```

#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

```
PasswordAuthentication yes
```

# service ssh restart

```
is the information correct? [Y/n] y
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 privilege 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:wctGpvbTfVc8XK6WQn05HaLSCglSmqr2n12CSbmYrbw.
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+kvRqnwYIk6my8nV6ELL9aLG4sywmWTeRMpw.
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.

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

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
```

```
file1
devops@ip-172-31-35-137:~/.ssh$ vi jenkins.yml
devops@ip-172-31-35-137:~/.ssh$ ansible-playbook jenkins.yml
```

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

PublicIPs: 3.85.174.133 PrivateIPs: 172.31.35.137

Now run the playbook

Ansible-playbook Jenkins.yml

```

devops@ip-172-31-35-137:~/.ssh$ ansible all -a"ls"
[WARNING]: Platform linux on host 172.31.32.171 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more
information.
172.31.32.171 | CHANGED | rc=0 >>
file!
devops@ip-172-31-35-137:~/.ssh$ vi jenkins.yml
devops@ip-172-31-35-137:~/.ssh$ ansible-playbook jenkins.yml

PLAY [ansiblegroup] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host 172.31.32.171 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.32.171]

TASK [Ensure wget is installed] *****
ok: [172.31.32.171]

TASK [Download Jenkins installation script] *****
changed: [172.31.32.171]

TASK [Run Jenkins installation script] *****
changed: [172.31.32.171]

PLAY RECAP *****
172.31.32.171 : ok=4 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

devops@ip-172-31-35-137:~/.ssh$

```

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)

```

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 tomcat from url
      ansible.builtin.get_url:
        url: https://d1cdn.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("running 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