

Name: Venkata Lahari Goruputi

Role: DevOps Intern

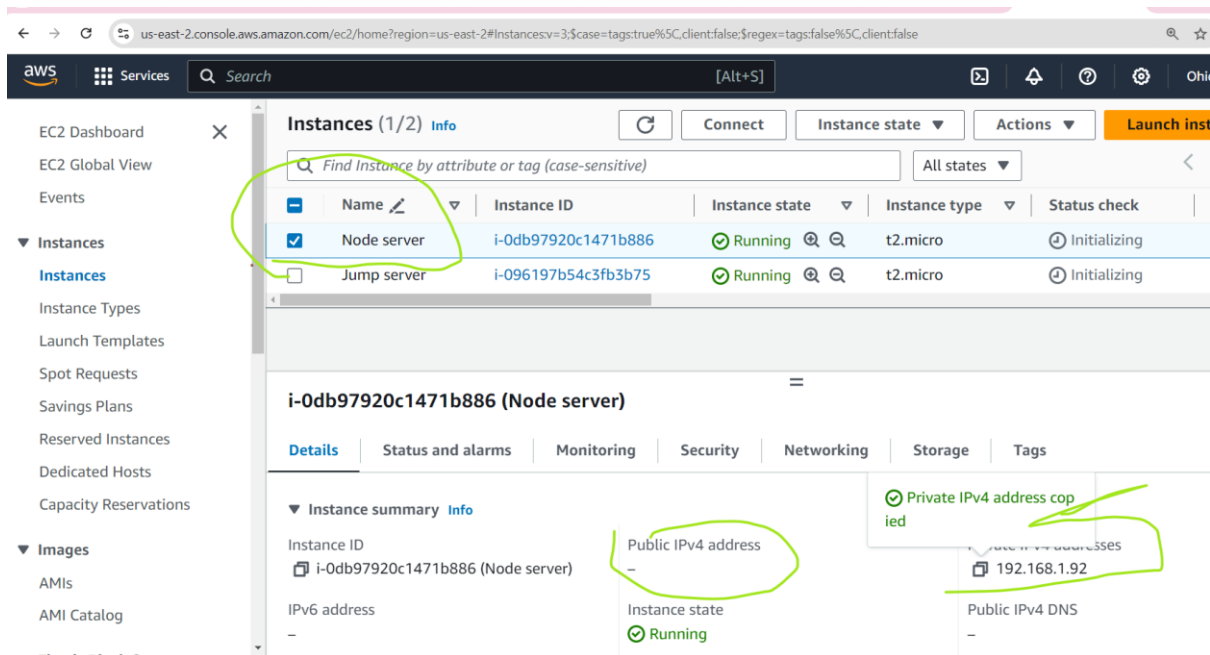
Task: Ansible Master and worker authentication through Jump server

What is Ansible:

- Ansible is an open-source automation tool used for configuration management, application deployment, and task automation. It simplifies the management of IT infrastructure by automating repetitive tasks and ensuring consistency across servers.

What is Jump Server:

- A jump server (also known as a bastion host) is a server used to securely access and manage other servers or systems in a network, typically within a more secure or isolated environment. The jump server acts as an intermediary between a user and the target systems, providing a controlled and monitored point of entry.
- ✓ As part of this task Firstly I have created a ssh key in local using **ssh-keygen** command, then I have imported the public to AWS.
- ✓ Then I have created a full network setup in AWS in n.virginia region and Ohio region and also created master EC2 instance in n.virginia region, Jump and node servers in Ohio region
- ✓ Network and EC2 instances setup is as follows:
 - **N.Virginia:**
 - Created a VPC and also created a Subnet under that VPC
 - Created an IGW and attached it to VPC to provide internet to VPC.
 - Created a new Route Table under this VPC and given the route to above created IGW and attached the subnet which is created above.
 - Created a Security Group under this VPC.
 - Created an EC2 instance under the newly created network and using the keypair and then enabled the public-IP and used the same Security group then logged into it.
 - **Ohio:**
 - Created a VPC and IGW and attached the IGW to VPC.
 - Created two subnets under this VPC and one is public and another one is private
 - Created two route tables under this VPC one is public and another is private.
 - Then created a NAT Gateway using Elastic IP.
 - For the public-route table I have attached one subnet (Having Name Public) and given the route to IGW to this Route Table, then this subnet will become public-subnet.
 - For the private-route table I have attached the private subnet and given the route to NAT Gateway which makes this subnet as a private subnet.
 - Created a security group under this VPC.
 - Then I have created the Jump-Server by using the public subnet and enabled the public-IP and logged into it.
 - Then I have created Node-Server by using the private subnet and disabled the public-IP.



- ✓ Then I have copied the private key to jump-server and Master in order to access the node-server using private IP from the Jump-Server and Master-Server, Used the below command to copy the keys from local to the servers.

scp -i localpath-to-master-ssh-pvt-key localpath-to-jump-server-ssh-pvt-key ubuntu@<master-server-public-ip>:/home/ubuntu/ → This command will copy the jump-server's private key to MasterServer

scp -i localpath-to-jump-ssh-pvt-key localpath-to-node-ssh-pvt-key ubuntu@<jump-server-public-ip>:/home/ubuntu/ → This command will copy the node-server's private key to JumpServer

- ✓ The above commands will copy the keys to master and jump serves home paths (here I have used the same keys for all three servers).

```
ssh username@jumpserver-publicip
chmod 600 /home/ubuntu/id_rsa
ssh -i /home/ubuntu/id_rsa ubuntu@<node-server-private-ip>
```

- ✓ Follow the above commands.
 - First command will let us login to JumpServer from local with jump server's public Ip.
 - Second command will give the necessary permissions to the node server's private key which is copied above.
 - Last command will let us login to node server by using node server's private-IP from JumpServer.

```
jump server
ubuntu@ip-192-168-0-242:~$ ls
id_rsa
ubuntu@ip-192-168-0-242:~$ |
```

```
ubuntu@ip-192-168-0-242:~$ ls
id_rsa
ubuntu@ip-192-168-0-242:~$ ssh -i id_rsa ubuntu@192.168.1.92
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Sun Aug  4 12:23:21 UTC 2024

System load:  0.12               Processes:            106
Usage of /:   39.9% of 7.57GB    Users logged in:     0
Memory usage: 26%               IPv4 address for eth0: 192.168.1.92
Swap usage:   0%

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Sat Aug  3 05:50:57 2024 from 10.0.0.249
ubuntu@ip-192-168-1-92:~$
```

✓ Then login to master and then login to jump server using the below.

```
ssh ubuntu@master-public-IP

mv /home/ubuntu/JumpKey ~/.ssh/
chmod 400 ~/.ssh/JumpKey

ssh -i id_rsa ubuntu@jump-server-public-ip
```

```
jump server  master
ubuntu@ip-10-0-0-249:~$ ls -al .ssh/
total 28
drwx----- 2 ubuntu ubuntu 4096 Aug  2 08:39 .
drwxr-x--- 5 ubuntu ubuntu 4096 Aug  2 10:33 ..
-rw----- 1 ubuntu ubuntu 560 Aug  2 07:26 authorized_keys
-rw----- 1 root root 210 Aug  2 08:30 config
-r----- 1 ubuntu ubuntu 2610 Aug  2 08:23 id_rsa
-rw----- 1 ubuntu ubuntu 1404 Aug  2 09:02 known_hosts
-rw-r--r-- 1 ubuntu ubuntu 426 Aug  2 08:39 known_hosts.old
ubuntu@ip-10-0-0-249:~$ ssh -i .ssh/id_rsa ubuntu@3.147.2.104
The authenticity of host '3.147.2.104 (3.147.2.104)' can't be established.
ED25519 key fingerprint is SHA256:f/2vzFXsGb/SVeJS57F1wmEKrkRhT10EJUBkvGtFXQE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:2: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.147.2.104' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1023-aws x86_64)

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

System information as of Sun Aug  4 12:30:12 UTC 2024

System load:  0.0               Processes:    102
Usage of /:   29.2% of 7.57GB   Users logged in: 1
Memory usage: 21%              IPv4 address for eth0: 192.168.0.242
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

14 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Last login: Sun Aug  4 12:21:22 2024 from 223.238.41.54
ubuntu@ip-192-168-0-242:~$
```

✓ We can login to servers using their respective keys like in the below pic.

marker

Jump

Note

```
Windows PowerShell
Master

ubuntu@ip-172-31-38-178:~$ ssh -i ~/.ssh/jumpkey.pem ubuntu@18.227.10.51
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Sun Jul 28 12:02:06 UTC 2024

System load:  0.0          Processes:      107
Usage of /:   23.6% of 7.57GB Users logged in: 1
Memory usage: 23%         IPv4 address for eth0: 172.31.12.80
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

23 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Sun Jul 28 11:57:30 2024 from 54.162.10.180
ubuntu@ip-172-31-12-80:~$ ls
jumpkey.pem
ubuntu@ip-172-31-12-80:~$ ssh -i jumpkey.pem ubuntu@172.31.62.115
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Sun Jul 28 12:03:26 UTC 2024

System load:  0.08         Processes:      101
Usage of /:   23.6% of 7.57GB Users logged in: 1
Memory usage: 22%         IPv4 address for eth0: 172.31.62.115
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

23 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

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

Last login: Sun Jul 28 11:58:07 2024 from 172.31.12.80
ubuntu@ip-172-31-62-115:~$
```

- ✓ After successful login now we need to connect to node-server from master-server. For that we need to follow the below steps.
- ✓ Instead of log into respective servers using ssh keys every time we are using the ssh config file to login to servers using one simple ssh instruction for that we need to add the below context to the **.ssh** path create a file named as **config** in **.ssh** folder and add the below context and save the file.

Host jump-server**HostName** <JumpServer_Public_IP>**User** ubuntu**IdentityFile** ~/.ssh/id_rsa**Host node-server****HostName** <NodeServer_Private_IP>**User** ubuntu**IdentityFile** ~/.ssh/ id_rsa**ProxyCommand** ssh -W %h:%p jump-server

- ✓ Then save the file given the sufficient permissions. **chmod 600 .ssh/config**.
- ✓ Then run **ssh jump-server** this will make direct login to jump-server.
- ✓ In order make the node server also like the jump-server we need to make vpc-peering connection between two vpc's because master and node were in two different vpc's in two different regions.
- ✓ After the successful vpc-peering between the two VPC's we can able to login to node-server directly from master using **ssh node-server**.
- ✓ Then install ansible in master and try to ping the node from master for the follow the below steps.
- ✓ Create an inventory file add the below context to ping the node-server from ansible master.

[node]**node-server ansible_host**=<NodeServer_Private_IP>**[all:vars]****ansible_ssh_private_key_file**=~/.ssh/id_rsa**ansible_user**=ubuntu**ansible_ssh_common_args**='-o ProxyCommand="ssh -W %h:%p jump-server"'

- ✓ Then update the **ssh_config** and **ssh_config.d/*.conf** files.
- ✓ Then restart the **sshd**.
- ✓ Additionally add the IPs of each server in hosts file which is presented in /etc/ in master as below.

```
jump server  X  master  X  ubuntu@ip-
127.0.0.1 localhost
3.147.2.104 jump-server
192.168.1.92 node-server
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
~
```

```
jump server  X  master  X  ubuntu@ip-192-168-1-92: ~  +  v
ubuntu@ip-10-0-0-249:~$
ubuntu@ip-10-0-0-249:~$ ssh jump-server
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1023-aws x86_64)

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

System information as of Sun Aug  4 12:47:44 UTC 2024

System load:  0.0               Processes:           104
Usage of /:   29.2% of 7.57GB   Users logged in:    1
Memory usage: 24%              IPv4 address for eth0: 192.168.0.242
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

14 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

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

Last login: Sun Aug  4 12:30:12 2024 from 3.233.238.114
ubuntu@ip-192-168-0-242:~$ exit
logout
Connection to jump-server closed.
ubuntu@ip-10-0-0-249:~$ ssh node-server
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)

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

System information as of Sun Aug  4 12:48:44 UTC 2024

System load:  0.0               Processes:           102
Usage of /:   39.9% of 7.57GB   Users logged in:    1
Memory usage: 28%              IPv4 address for eth0: 192.168.1.92
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Sun Aug  4 12:23:21 2024 from 192.168.0.242
ubuntu@ip-192-168-1-92:~$
```

- ✓ Then ping the node using ansible with the below command.

[ansible -m ping -i inventory all (or) node]

```
ubuntu@ip-10-0-0-249: ~  
ubuntu@ip-10-0-0-249:~$ ls  
ansible.yaml  hosts  
ubuntu@ip-10-0-0-249:~$ ansible -m ping -i hosts node  
node-server | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
ubuntu@ip-10-0-0-249:~$ ansible -m ping -i hosts all  
node-server | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "ping": "pong"  
}  
ubuntu@ip-10-0-0-249:~$
```

- ✓ If everything works fine ansible ping will work.
- ✓ Then try to ping the node from ansible-playbook, below is the playbook which I have used.

```
---  
- name: Test Connectivity  
  hosts: node  
  tasks:  
    - name: Ping Node Server  
      ping:
```

- ✓ Run the playbook using **ansible-playbook -i inventory playbook.yaml** command this should run without fail.


```
ubuntu@ip-10-0-0-249:~$ ansible-playbook -i hosts ansible.yaml

PLAY [Test Connectivity] *****

TASK [Gathering Facts] *****
ok: [node-server]

TASK [Ping Node Server] *****
ok: [node-server]

PLAY RECAP *****
node-server : ok=2  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

ubuntu@ip-10-0-0-249:~$
```

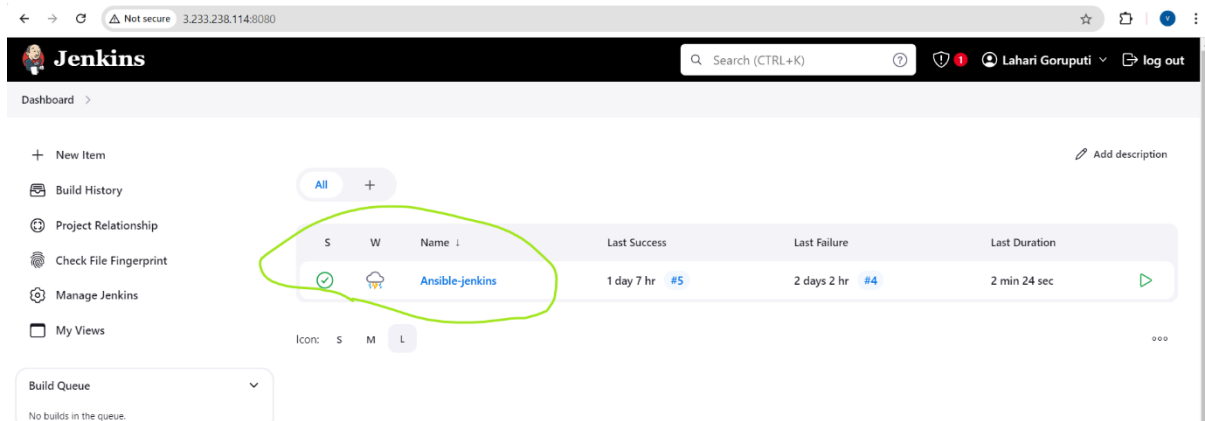
- ✓ Then install Jenkins in master and access Jenkins and try to connect the node in Jenkins master.

```
ubuntu@ip-10-0-0-249:~$ sudo systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2024-08-04 12:19:40 UTC; 34min ago
     Main PID: 362 (java)
       Tasks: 56 (Limit: 1120)
      Memory: 394.2M
         CPU: 50.499s
    CGroup: /system.slice/jenkins.service
            └─362 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

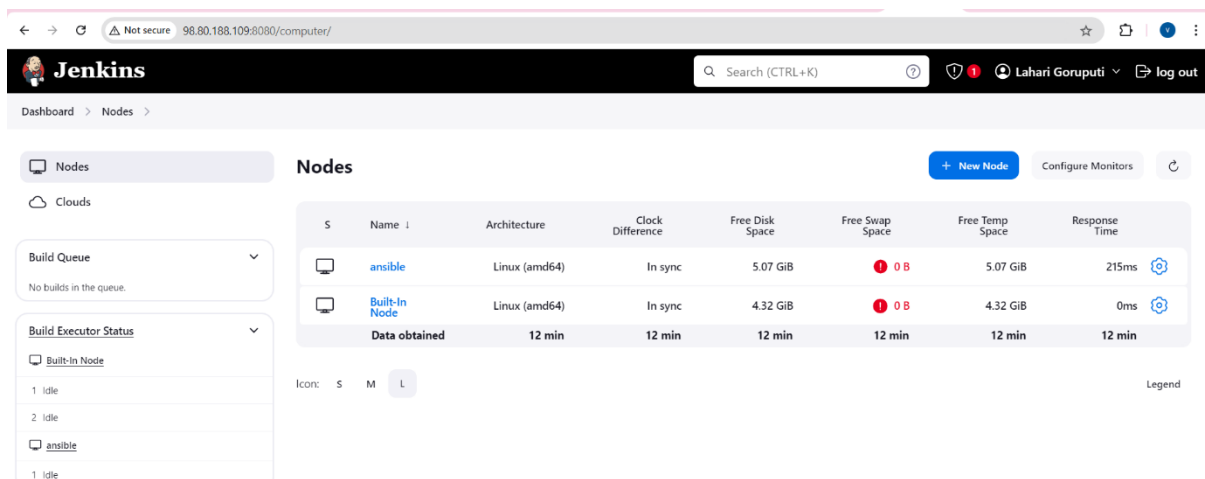
Aug 04 12:19:54 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:19:54.438+0000 [id=51] INFO h.m.DownloadService$Downloadable#load: Obtained the upda
Aug 04 12:19:54 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:19:54.513+0000 [id=51] INFO h.m.DownloadService$Downloadable#load: Obtained the upda
Aug 04 12:19:54 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:19:54.655+0000 [id=51] INFO h.m.DownloadService$Downloadable#load: Obtained the upda
Aug 04 12:19:54 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:19:54.656+0000 [id=51] INFO hudson.util.Retrier#start: Performed the action check u
Aug 04 12:22:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:22:22.480+0000 [id=34] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
Aug 04 12:24:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:24:22.480+0000 [id=36] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
Aug 04 12:26:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:26:22.480+0000 [id=34] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
Aug 04 12:28:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:28:22.480+0000 [id=40] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
Aug 04 12:30:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:30:22.480+0000 [id=50] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
Aug 04 12:44:22 ip-10-0-0-249 jenkins[362]: 2024-08-04 12:44:22.480+0000 [id=37] INFO hudson.slaves.SlaveComputer#tryReconnect: Attempting to
lines 1-20/20 (END)
```

- name: Spring petclinic
- hosts: all
- become: yes
- tasks:
 - name: install java-17 and git
 - ansible.builtin.apt:
 - name:
 - openjdk-17-jdk
 - git
 - update_cache: yes
 - state: present
 - name: clone git repo and build the code
 - ansible.builtin.shell: |
 - git clone https://github.com/lahari104/spring-petclinic.git
 - cd spring-petclinic
 - ./mvnw package

- ✓ After that I have taken a open-source java project which is SPC and then written the pipeline to build the code of it in node-server from playbook.
- ✓ Then create a new branch in the repo and add the playbook and hosts file and Jenkinsfile and push the code to repo. Use the above content for playbook.
- ✓ Then create a pipeline project in Jenkins and add the configuration to run the pipeline in master.



- ✓ Then try to connect the node to Jenkins master.



- ✓ Then run the pipeline. If all the configurations were right the pipeline will succeed.

```
172664088.088137-21287-26201668066485/ > /dev/null 2>&1 && sleep 0"****  
<192.168.1.92> (0, b'', b'')  
node-server | SUCCESS => {  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3"  
  },  
  "changed": false,  
  "invocation": {  
    "module_args": {  
      "data": "pong"  
    }  
  },  
  "ping": "pong"  
}  
+ ansible-playbook -i hosts ansible-jenkins.yml  
  
PLAY [SpringPetclinic] *****  
  
TASK [Gathering Facts] *****  
ok: [node-server]  
  
TASK [Install Java-17 and git] *****  
ok: [node-server]  
  
TASK [Clone git repo and build the code] *****  
changed: [node-server]  
  
PLAY RECAP *****  
node-server      : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0  
  
[Pipeline]   
[Pipeline] // stage  
[Pipeline]   
[Pipeline] // withEnv  
[Pipeline]   
[Pipeline] // node  
[Pipeline] End of Pipeline  
Finished: SUCCESS
```

- ✓ This configuration will build the code in node-server from playbook but this pipeline will run in master.
- ✓ In the below pic we can see that code was built on node-server even though the pipeline was set to ran master. It is because in the playbook hosts the code was mentioned to execute in the node.

```
ubuntu@ip-192-168-1-92:~/$ ssh node-server  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1022-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Sun Aug  4 13:13:42 UTC 2024  
  
System load:  0.0          Processes:      103  
Usage of /:   44.5% of 7.57GB Users logged in:    1  
Memory usage: 32%         IPv4 address for eth0: 192.168.1.92  
Swap usage:   0%  
  
* Ubuntu Pro delivers the most comprehensive open source security and  
compliance features.  
  
https://ubuntu.com/aws/pro  
  
Expanded Security Maintenance for Applications is not enabled.  
  
14 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
1 additional security update can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
  
*** System restart required ***  
Last login: Sun Aug  4 12:53:36 2024 from 192.168.0.242  
ubuntu@ip-192-168-1-92:~$ ls  
remoting  remoting.jar  spring-petclinic  
ubuntu@ip-192-168-1-92:~$ cd spring-petclinic/ && ls  
Dockerfile  Jenkinsfile  LICENSE.txt  build.gradle  docker-compose.yml  gradle  gradlew  gradlew.bat  mvnw  mvnw.cmd  pom.xml  readme.md  settings.gradle  src  target  
ubuntu@ip-192-168-1-92:~/spring-petclinic/target$ ls  
checkstyle-cachefile  checkstyle-result.xml  generated-sources  maven-archiver  spring-petclinic-3.0.0-SNAPSHOT.jar  test-classes  
checkstyle-checker.xml  checkstyle-suppressions.xml  generated-test-sources  maven-status  spring-petclinic-3.0.0-SNAPSHOT.jar.original  
checkstyle-header.txt  classes  jacoco.exec  site  surefire-reports  
ubuntu@ip-192-168-1-92:~/spring-petclinic/target$
```

- ✓ In this way we can connect to Ansible Master and worker node authentication through Jump server.

Thanks,

Lahari G