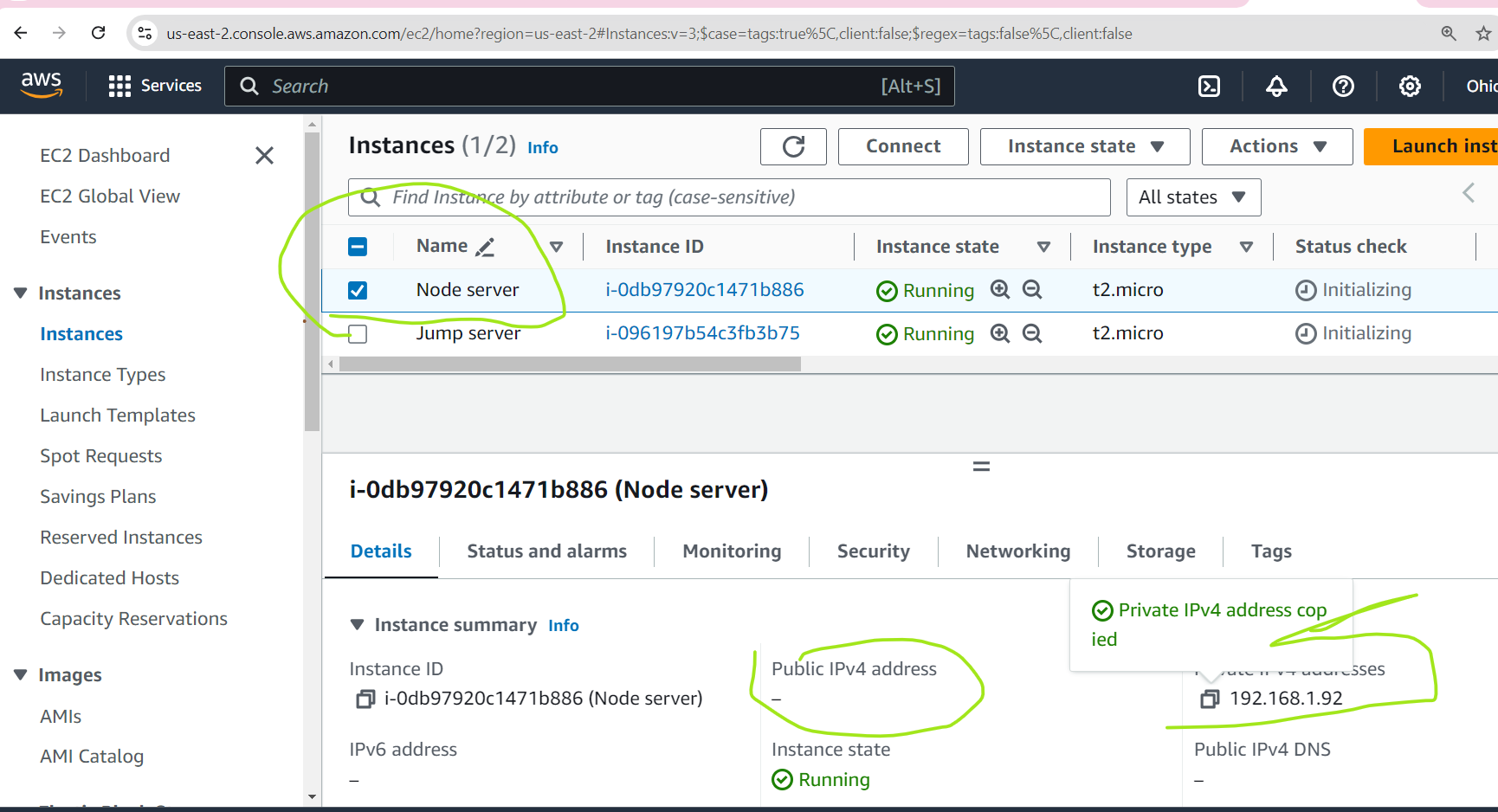
**What is Ansible:**

* 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.

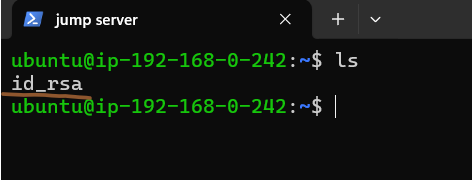
**scp -i localpath-to-master-ssh-pvt-key localpath-to-jump-server-ssh-pvt-key ubuntu@<master-server-public-ip>:/home/ubuntu/ 🡪 This command will copy the jump-server’s private key to MasterServer**

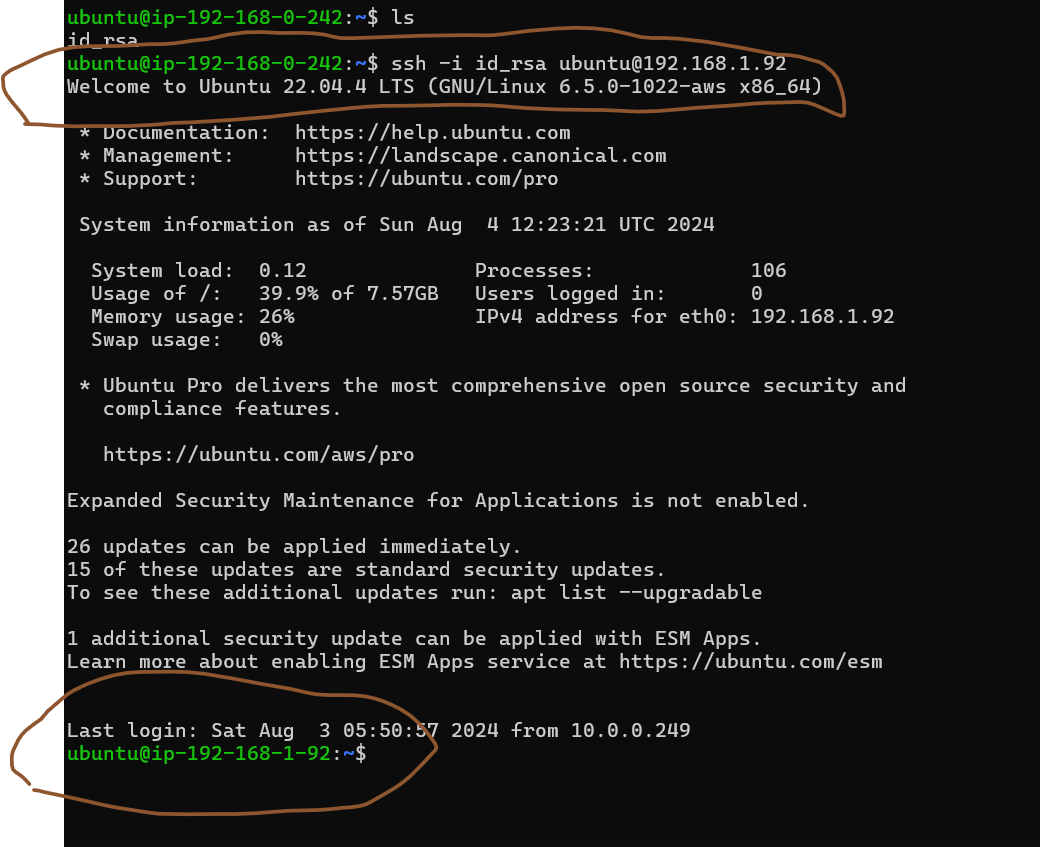
**scp -i localpath-to-jump-ssh-pvt-key localpath-to-node-ssh-pvt-key ubuntu@<jump-server-public-ip>:/home/ubuntu/ 🡪 This command will copy the node-server’s private key to JumpServer**

* 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.

****

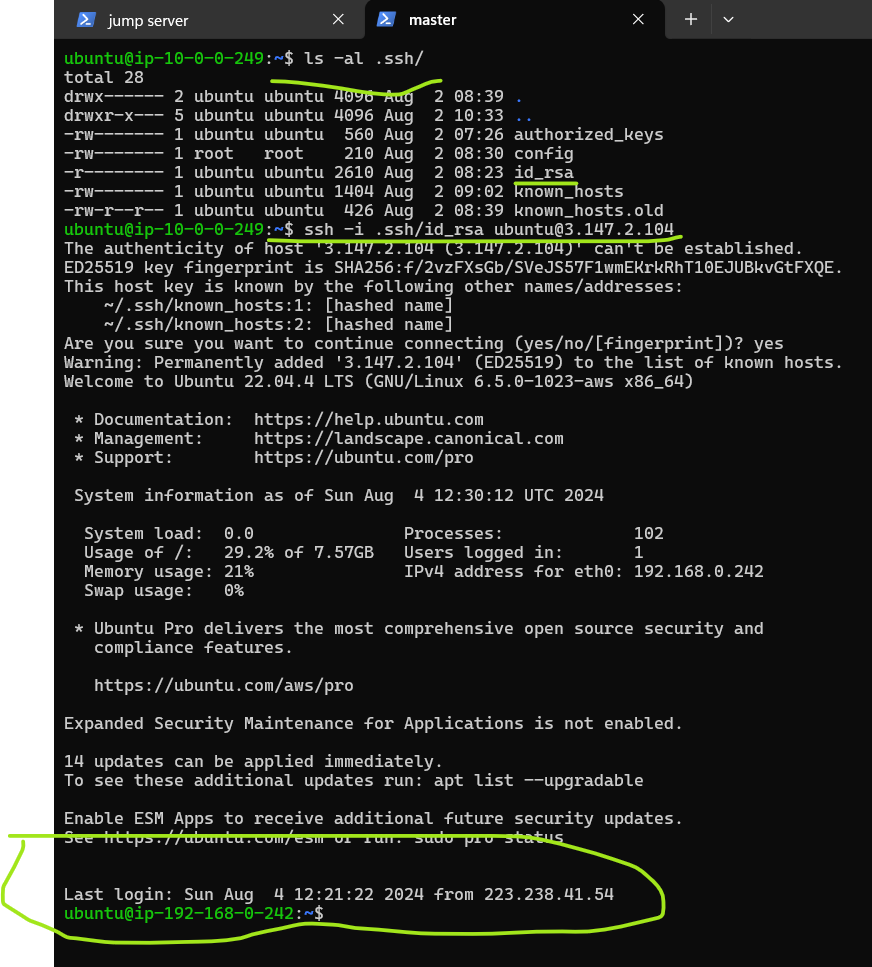
****

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

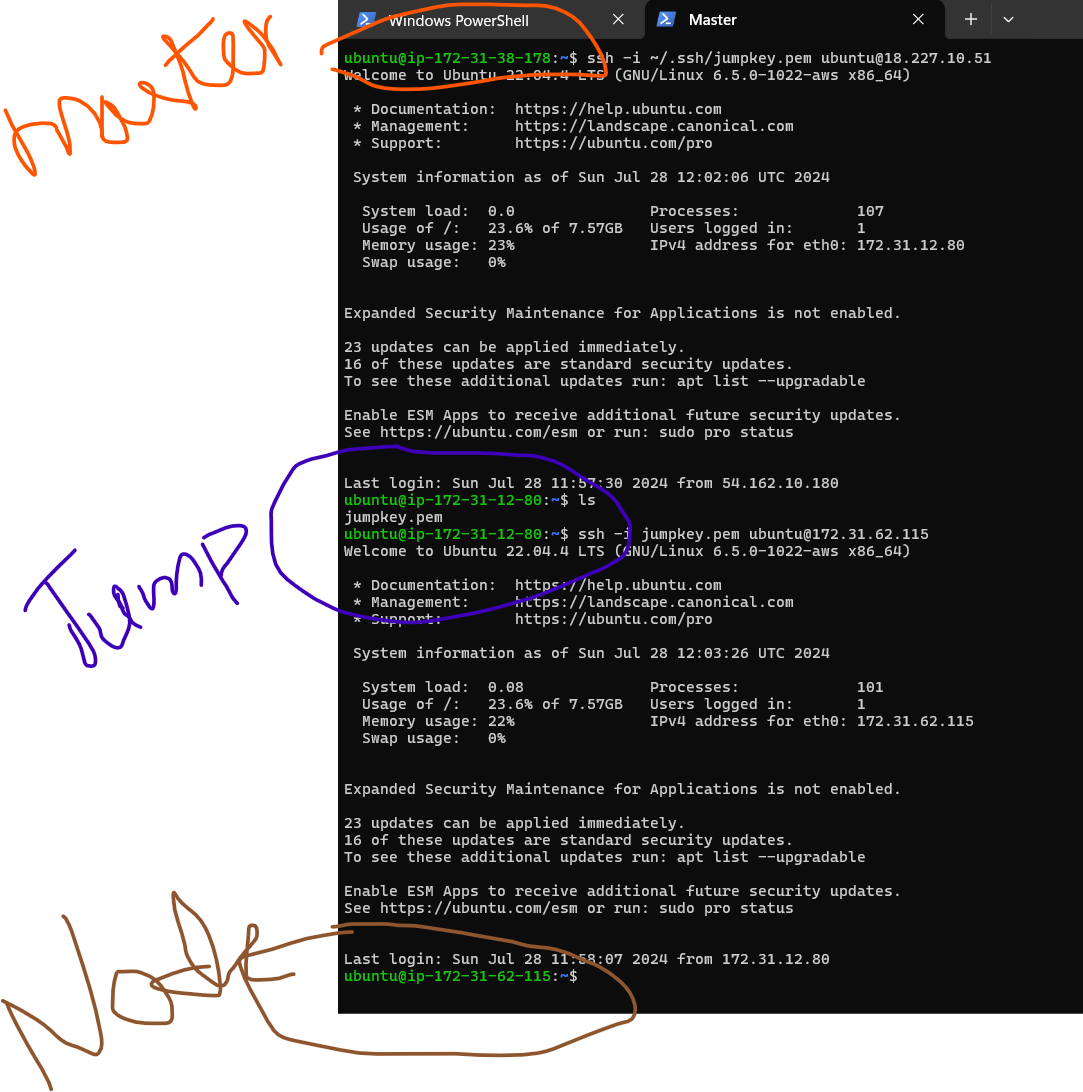
**ssh ubuntu@master-public-IP**

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

**ssh -i id\_rsa ubuntu@jump-server-public-ip**

****

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

****

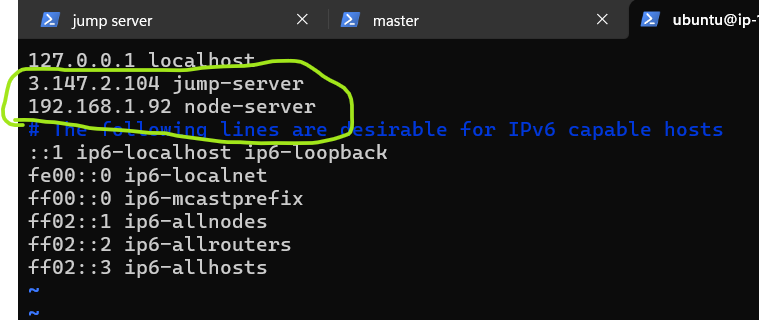
* After successful login now we need to connect to node-server from master-server. For that we need to follow the below steps.
* Instead of log into respective servers using ssh keys every time we are using the ssh config file to login to servers using one simple ssh instruction for that we need to add 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.

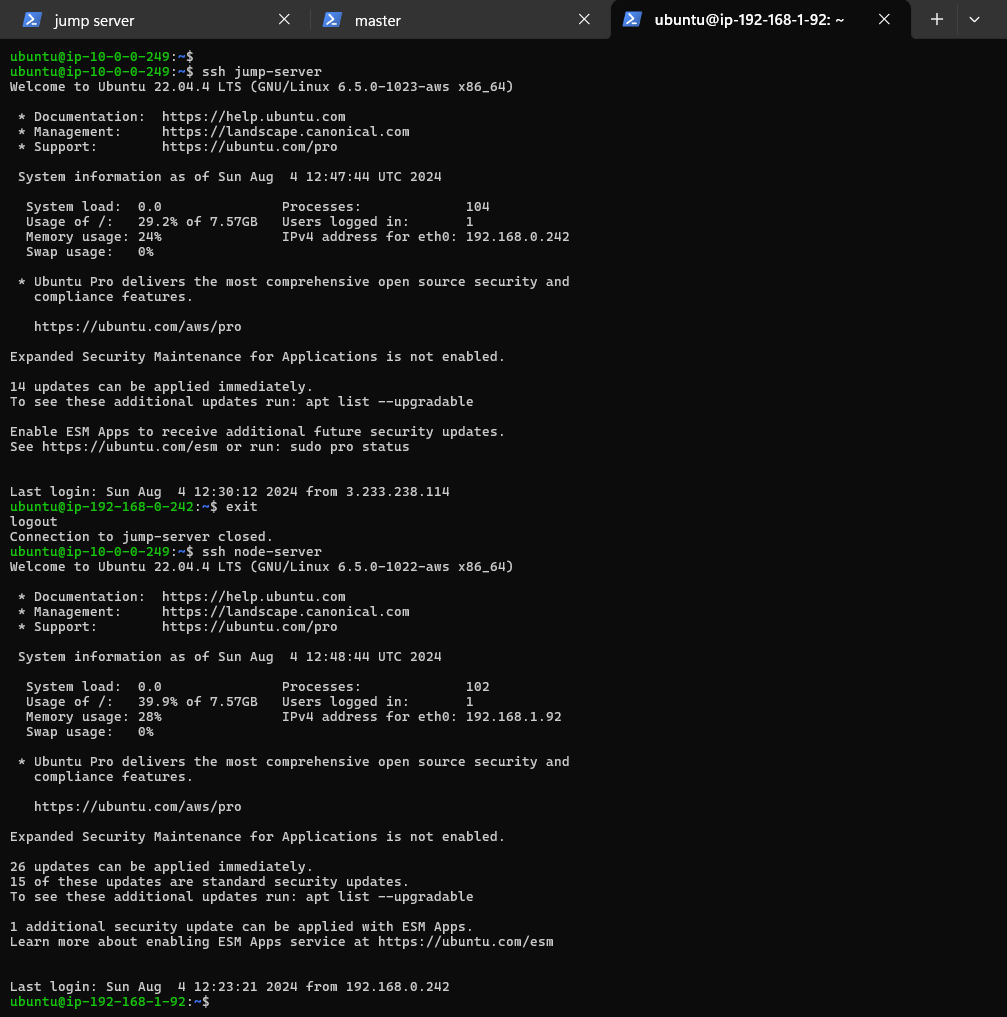
**Host jump-server  
  HostName <JumpServer\_Public\_IP>  
  User ubuntu  
  IdentityFile ~/.ssh/id\_rsa  
  
Host node-server  
  HostName <NodeServer\_Private\_IP>  
  User ubuntu  
  IdentityFile ~/.ssh/ id\_rsa  
  ProxyCommand ssh -W %h:%p jump-server**

* 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"'**

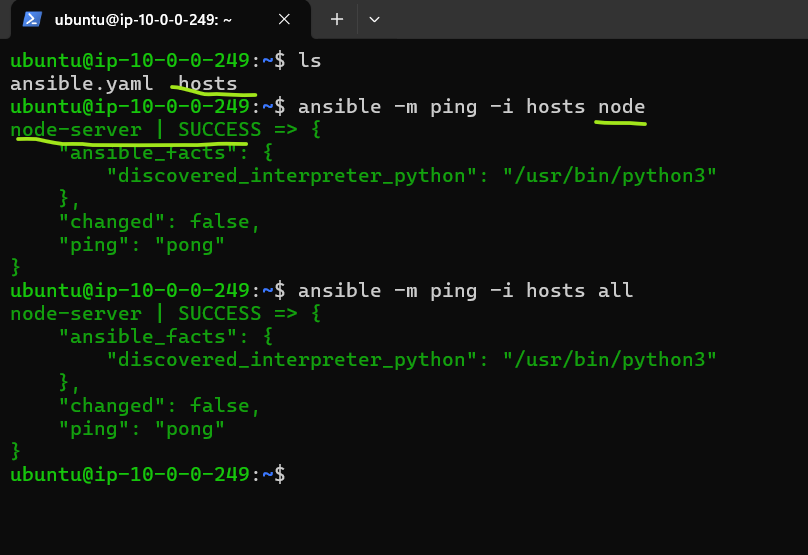
* Then update the **ssh\_config** and **ssh\_config.d/\*.conf** files.
* Then restart the **sshd**.
* Additionally add the IPs of each server in hosts file which is presented in /etc/ in master as below.

****

****

* Then ping the node using ansible with the below command.

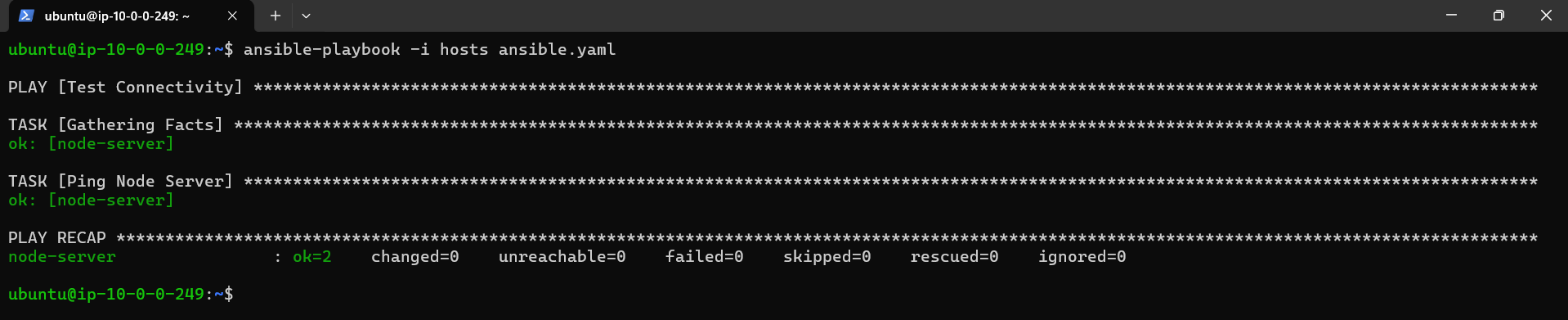
**[ansible -m ping -i inventory all (or) node]**

****

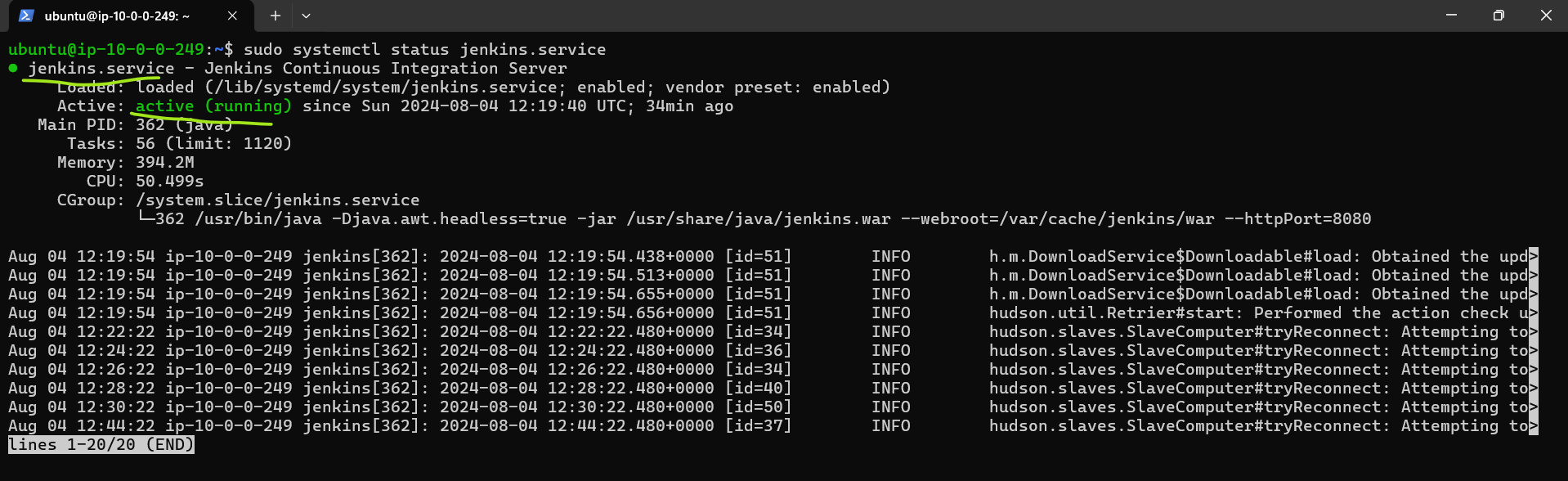
* If everything works fine ansible ping will work.
* Then try to ping the node from ansible-playbook, below is the playbook which I have used.

**---  
- 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.

****

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

****

---

- 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

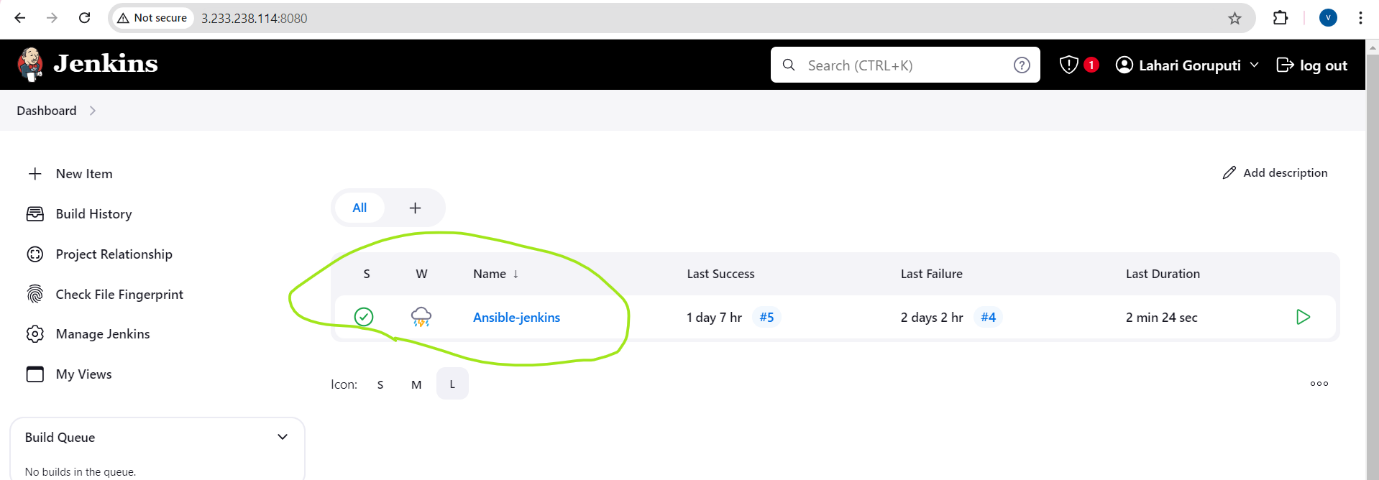
ansible.builtin.shell: |

git clone https://github.com/lahari104/spring-petclinic.git

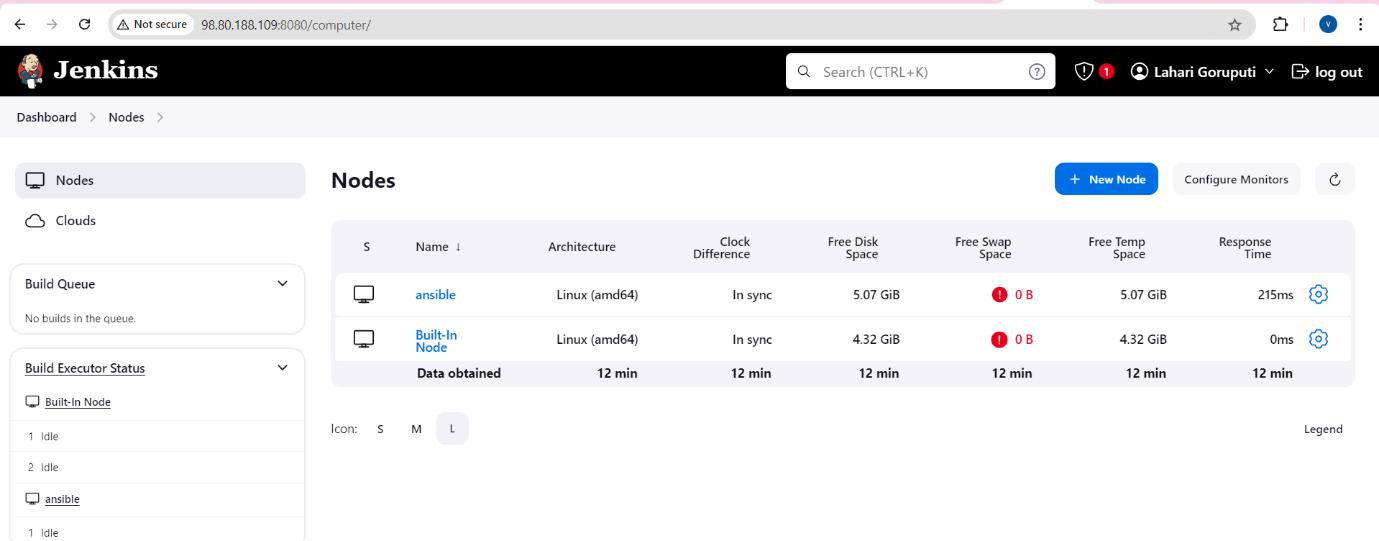
cd spring-petclinic

./mvnw package

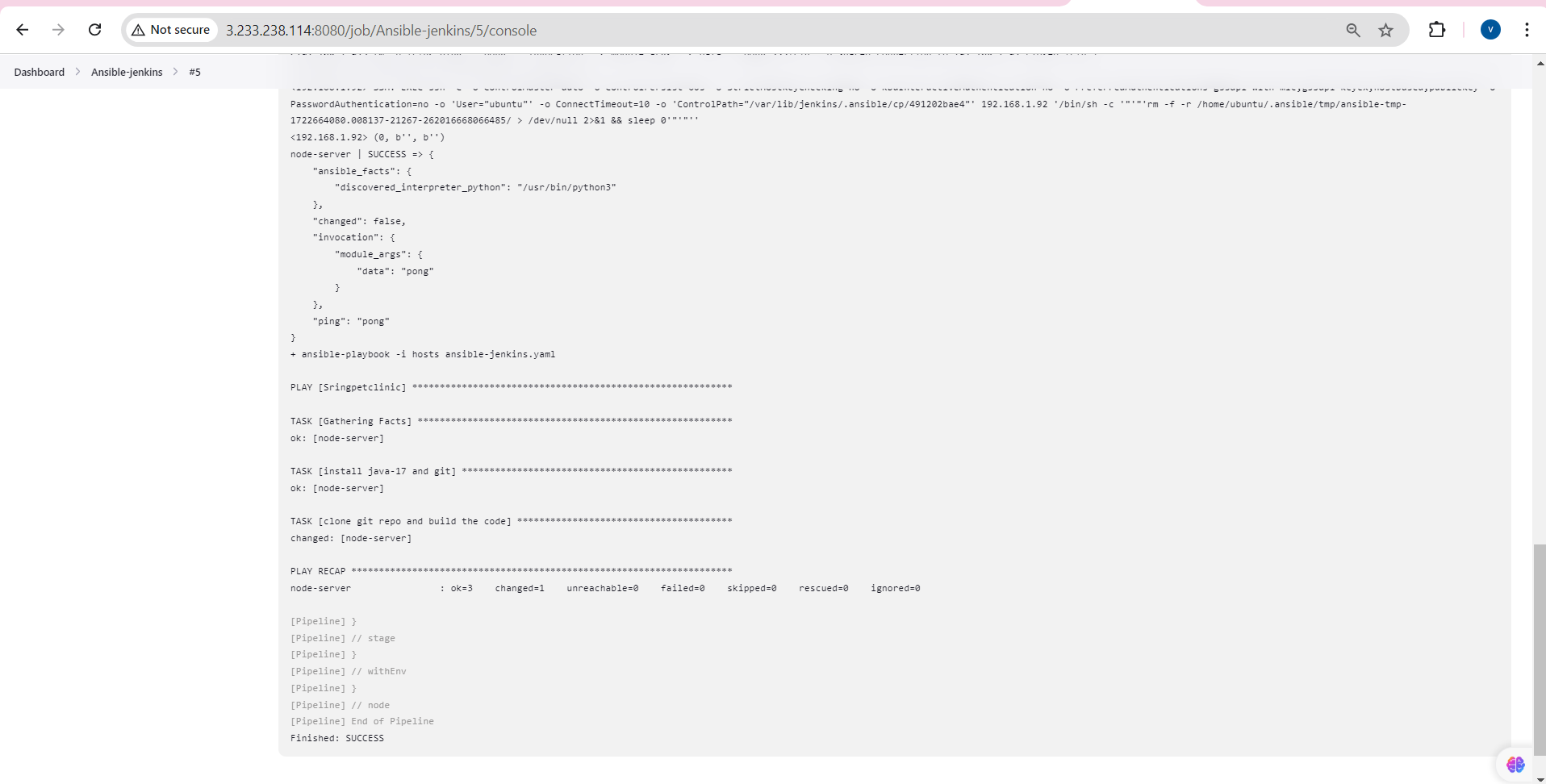
* After that I have taken a open-source java project which is SPC and then written the pipeline to build the code of it in node-server from playbook.
* Then create a new branch in the repo and add the playbook and hosts file and Jenkinsfile and push the code to repo. Use the above content for playbook.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

****

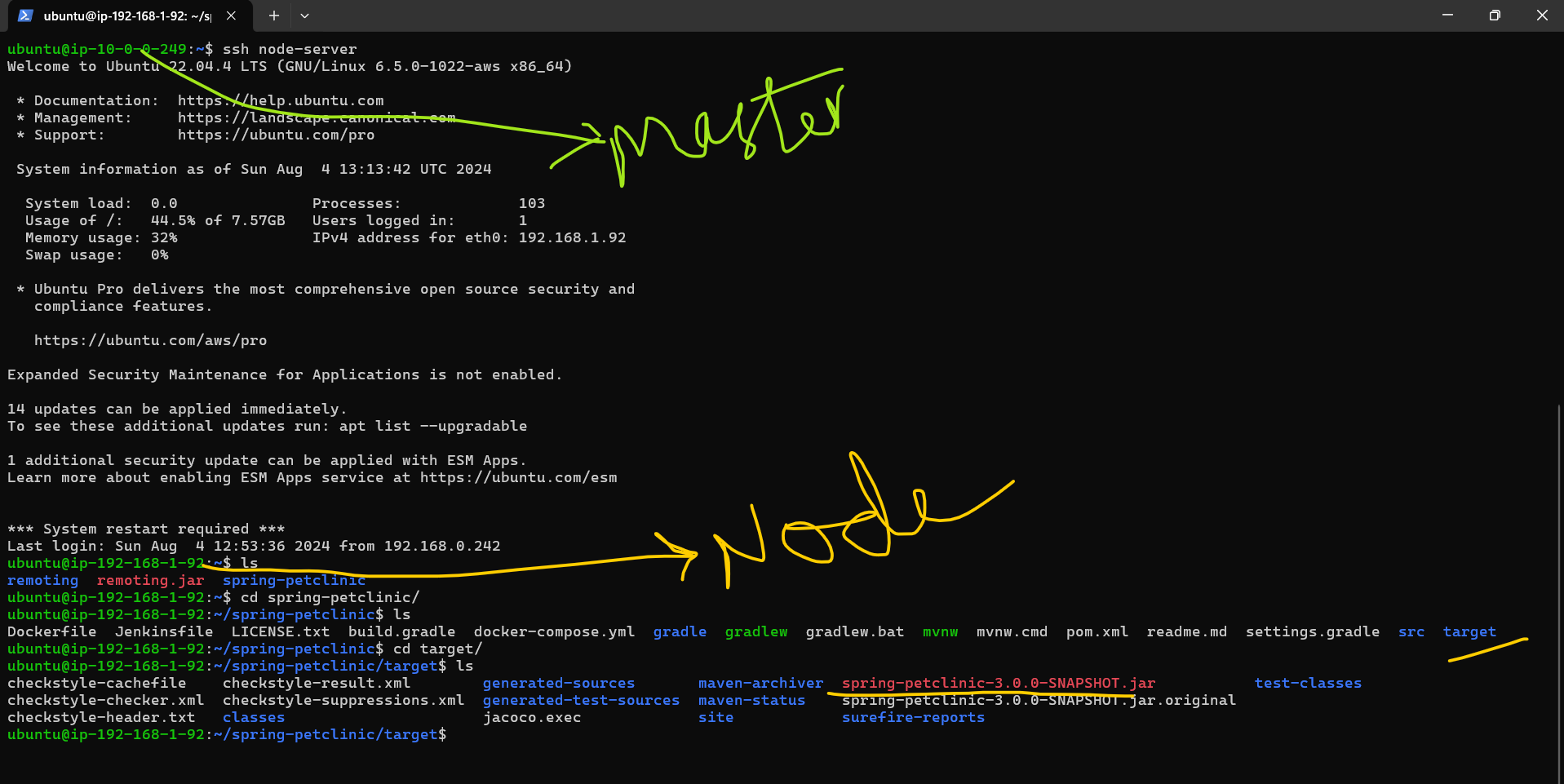
* Then try to connect the node to Jenkins master.

****

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

****

* 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.

****

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

**Thanks,**

**Lahari G**