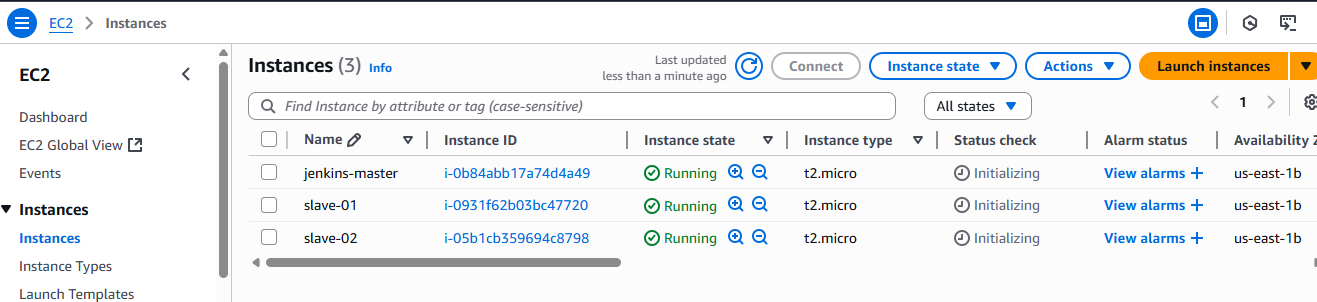
**Jenkins task-2**

1. **Configure 2 slave machines in Jenkins master.**

**First to create instances Jenkins-master,slave-01 and slave -02**



**Step-1 :** Create an ec2 –slave-01

Login to slave machine..

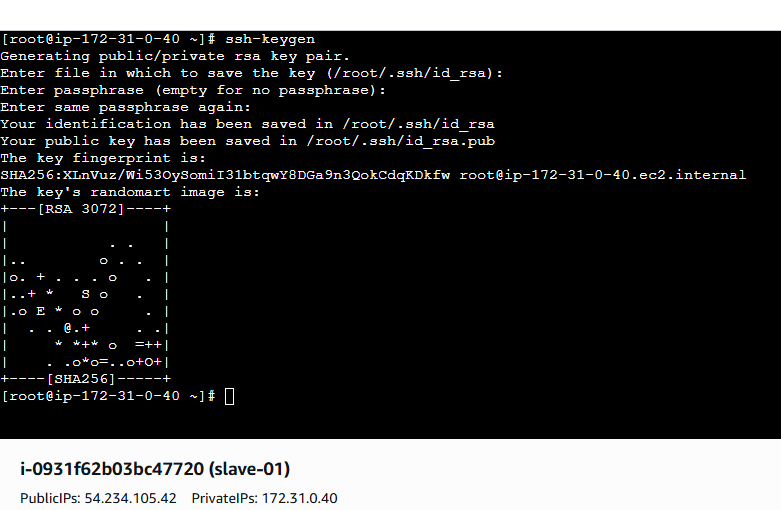
switch to **root user**

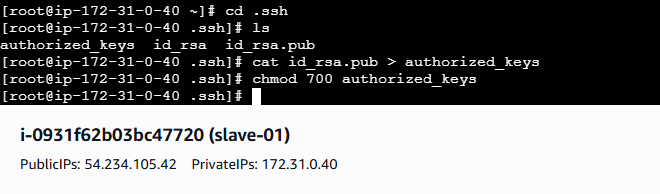
**dnf install -y java-17-amazon-corretto**

Create **ssh-keygen**

**cat id\_rsa.pub > authorized\_keys**

**chmod 700 authorized\_keys**





**Login to master machine**

**switch to root user**.

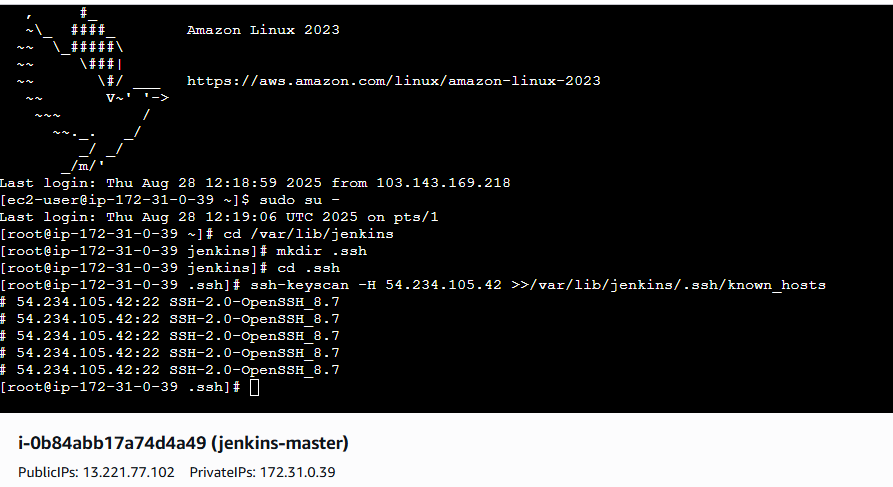
**mkdir -p /var/lib/jenkins/.ssh**

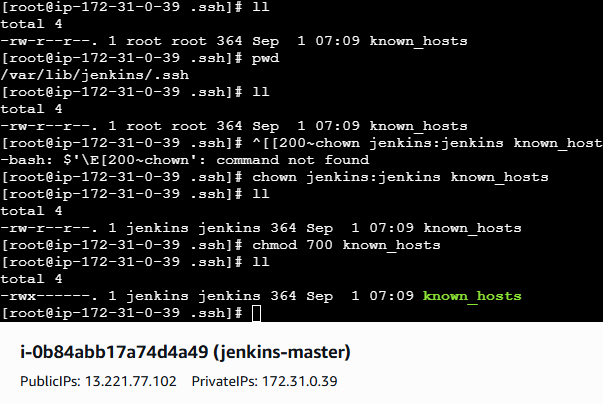
**cd /var/lib/jenkins/.ssh**

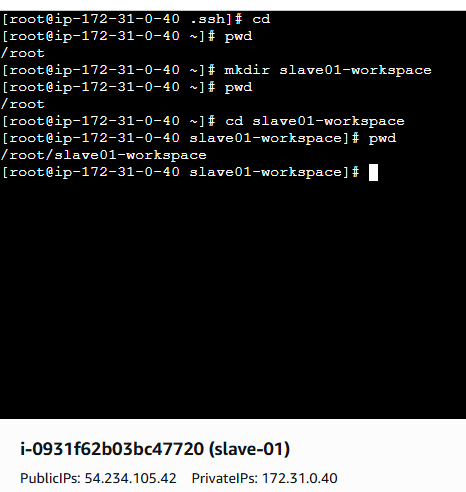
**ssh-keyscan -H SLAVE-NODE-IP-OR-HOSTNAME >>/var/lib/jenkins/.ssh/known\_hosts**

# **ssh-keyscan -H 54.234.105.42 >> /var/lib/jenkins/.ssh/known\_hosts**

**chown jenkins:jenkins known\_hosts**





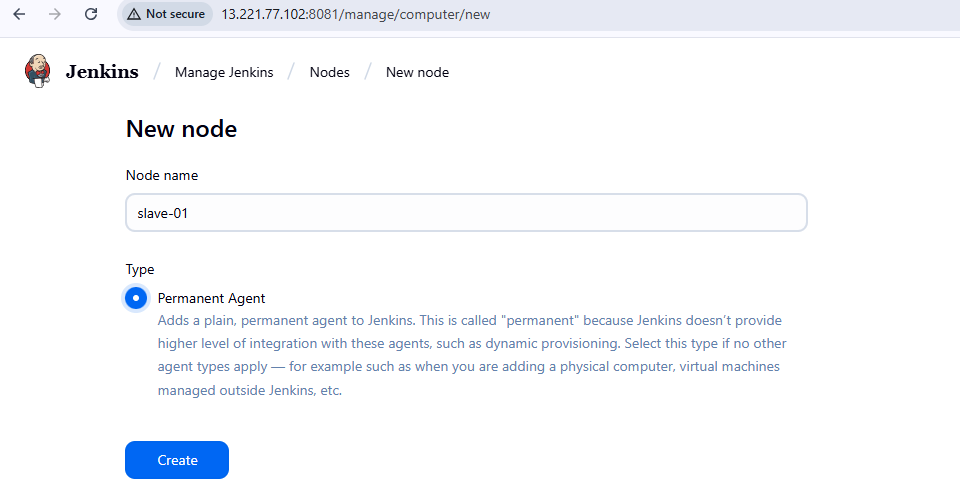


**Steps on Jenkins GUI**

Go to **Manage Jenkins → Manage Nodes and Clouds → New Node**

Give it a name (e.g., **slave-1)**

Select **Permanent Agent**



**To Configure Node**

* + **Remote root directory**: /home/ec2-user (or another working dir on slave)
  + **Labels**: (optional, e.g., Java)
  + **Launch method**: *Launch agent via SSH*
  + **Host**: *Public IP of Slave*

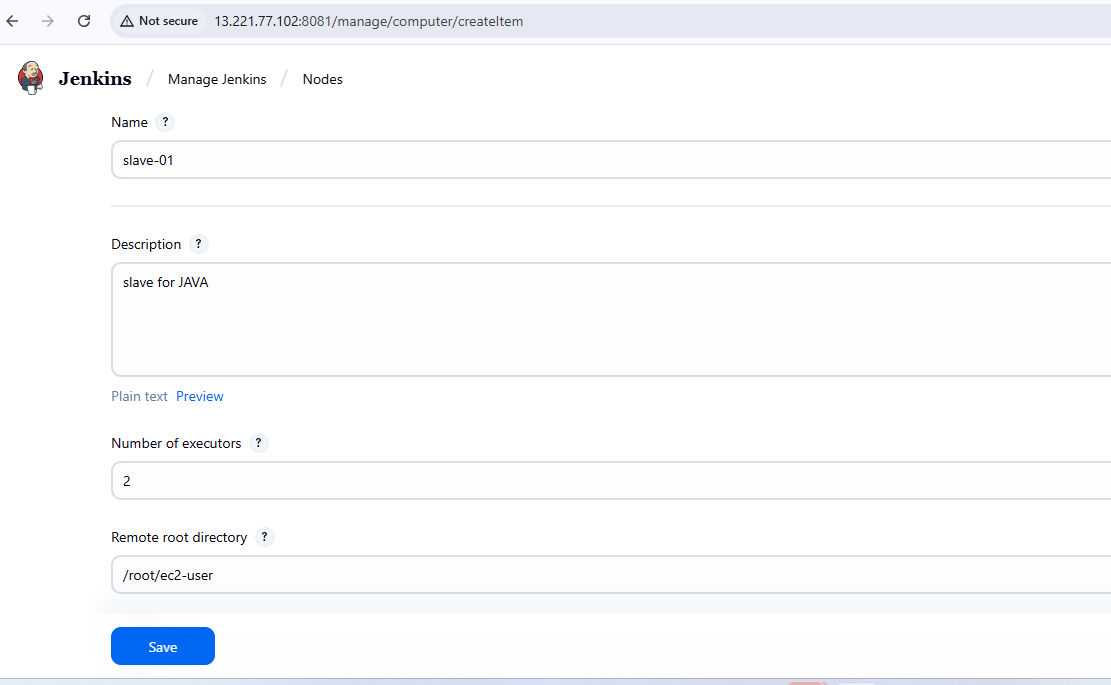
**Add Credentials**

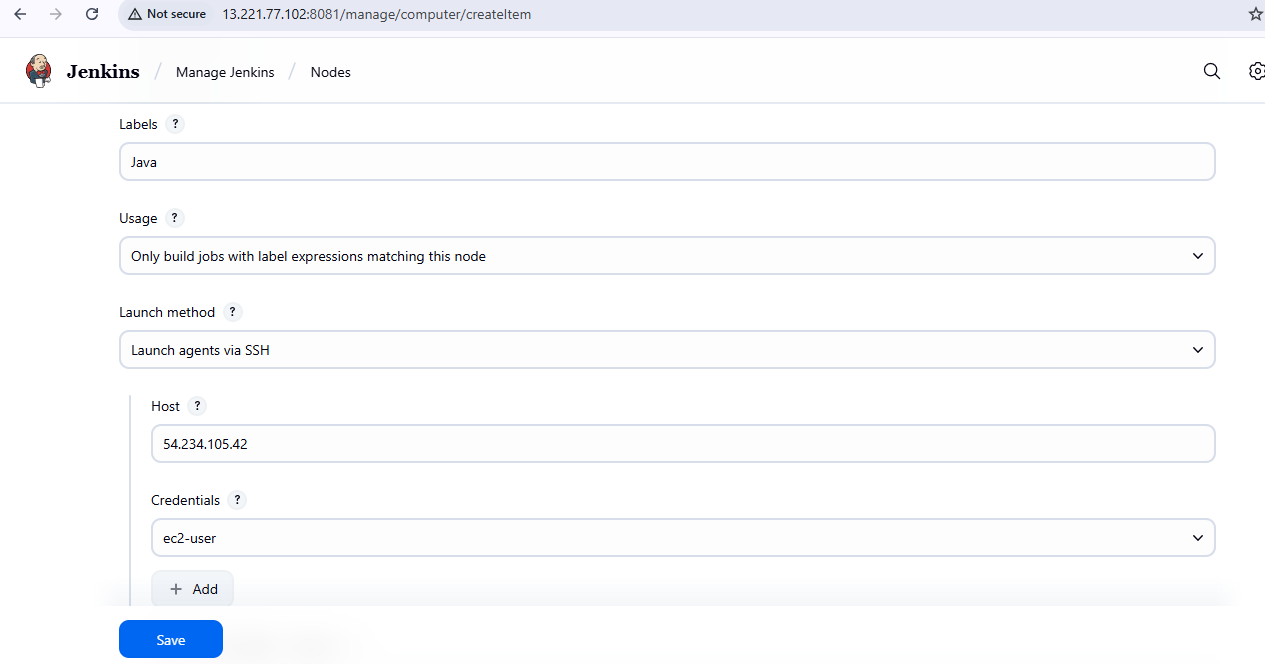
* + Kind: *SSH Username with private key*
  + Username: ec2-user
  + Private key: Paste your .pem key (from AWS)
  + Save

**Save & Test Connection**

* + Jenkins master should now connect to the slave machine via SSH.

If successful, the node status will show as **online**

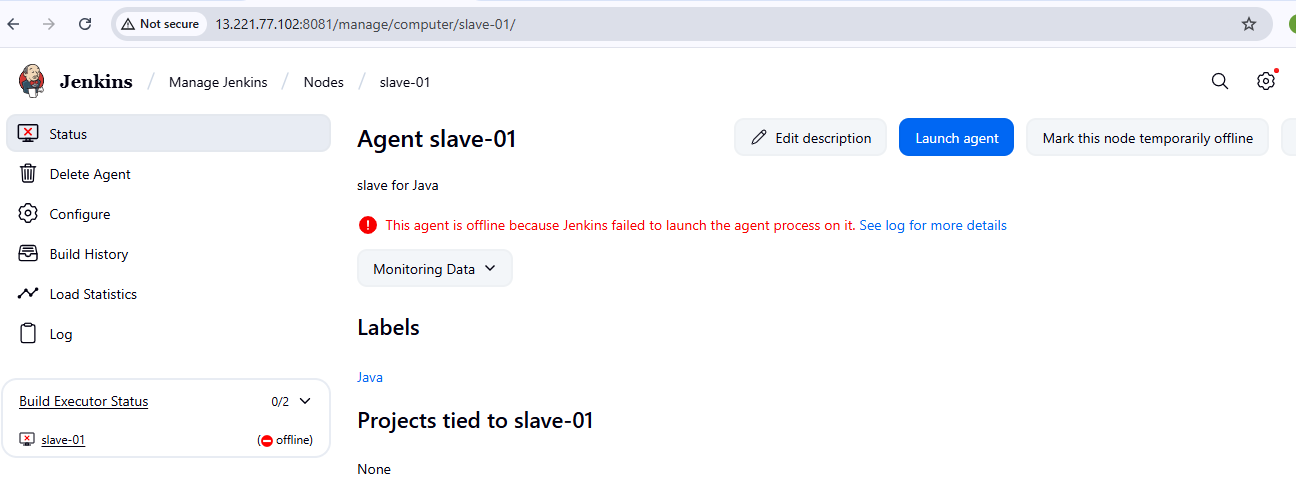




**Save & Test Connection**

* + Jenkins master should now connect to the slave machine via SSH.

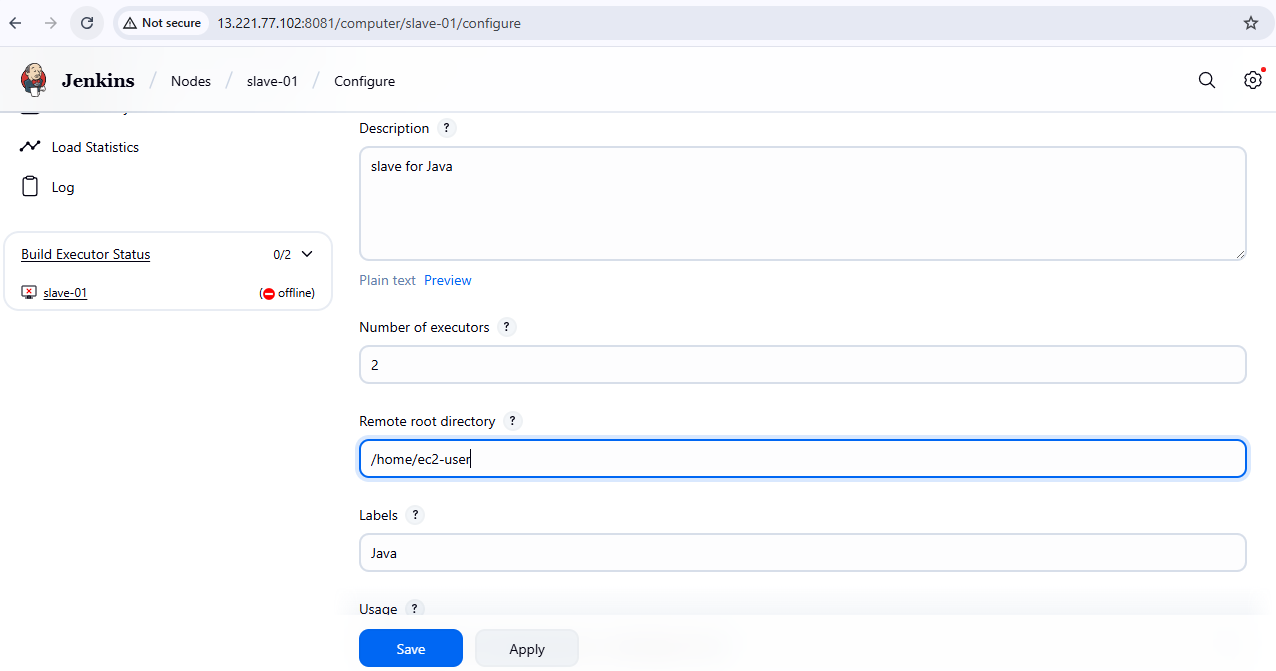
If successful, the node status will show as **online**

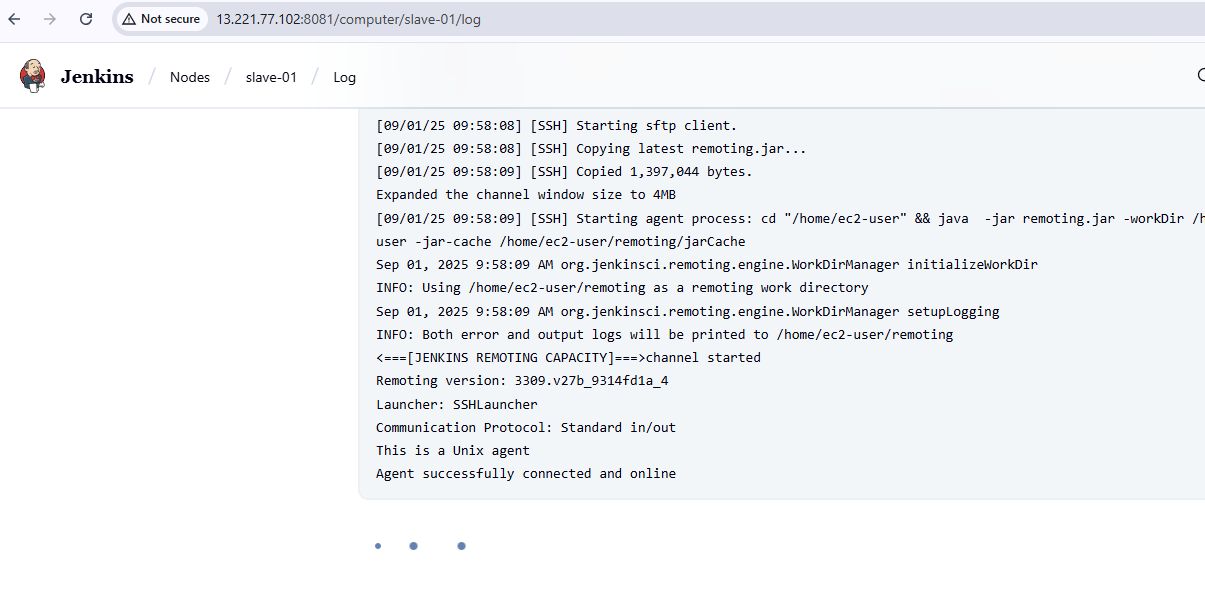


It shows like this to exit from root user

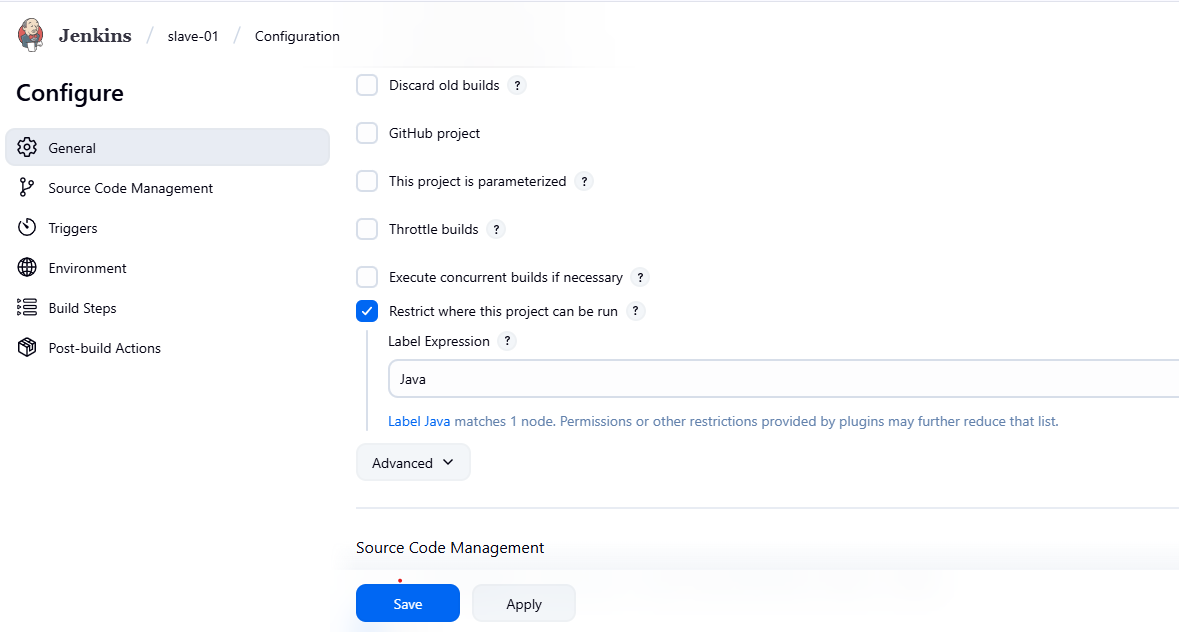
In **pwd** the path is **/home/ec2-use**r add this **path Remote root directory** of **created node slave-01**



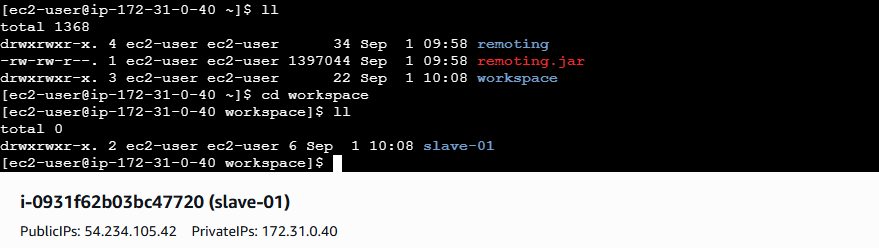




To create a **job(slave-01)** and **configure** it in **Java Label**

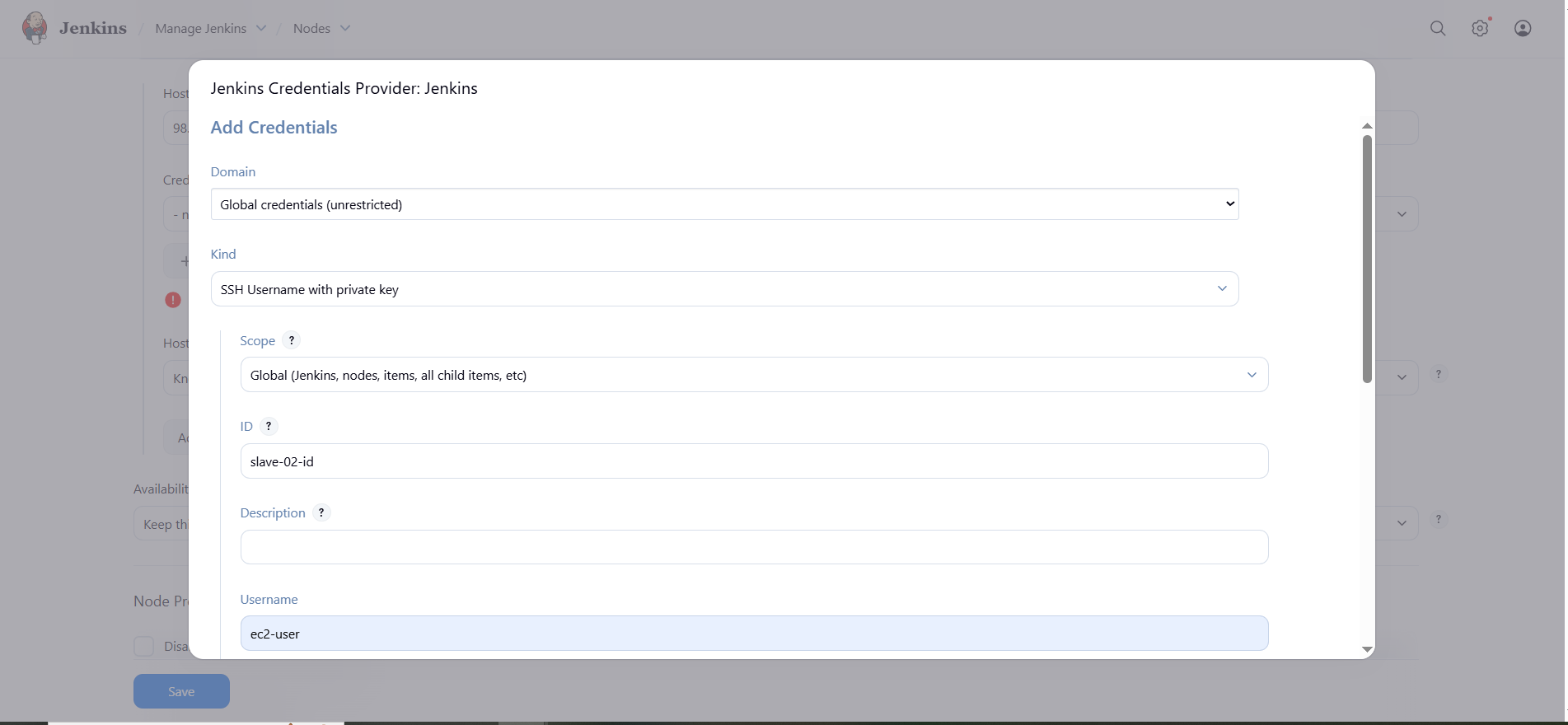






### **Create the node in Jenkins GUI**

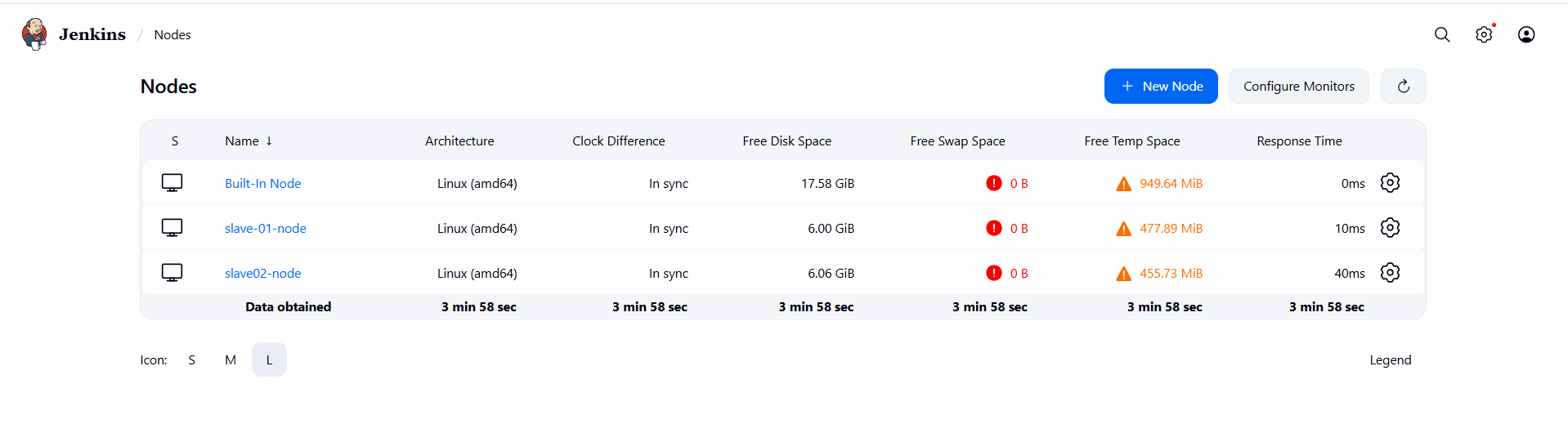
1. **Manage Jenkins → Nodes → New Node**
2. **Name: slave2 → Type: Permanent Agent**
3. **Remote root directory: /home/ec2-user**
4. **This path is taken from slave machine**
5. **Slave ec2 → cat /etc/passwd**
6. **chown ec2-user:ec2-user slave02-workspace**
7. **chmod 777 slave02-workspace**
8. **Labels: slave2 java (as you like)**
9. **Usage: “Only build jobs with label expressions” (optional)**
10. **Launch method: Launch agents via SSH**
    * **Host: <NEW\_AGENT\_IP\_OR\_DNS>**
    * **Credentials: create/select “SSH Username with private key”**
      + **Username: ec2-user**
      + **Private key: paste content of /var/lib/jenkins/.ssh/id\_rsa from the master**
    * **Host Key Verification Strategy: Known hosts file (recommended)**

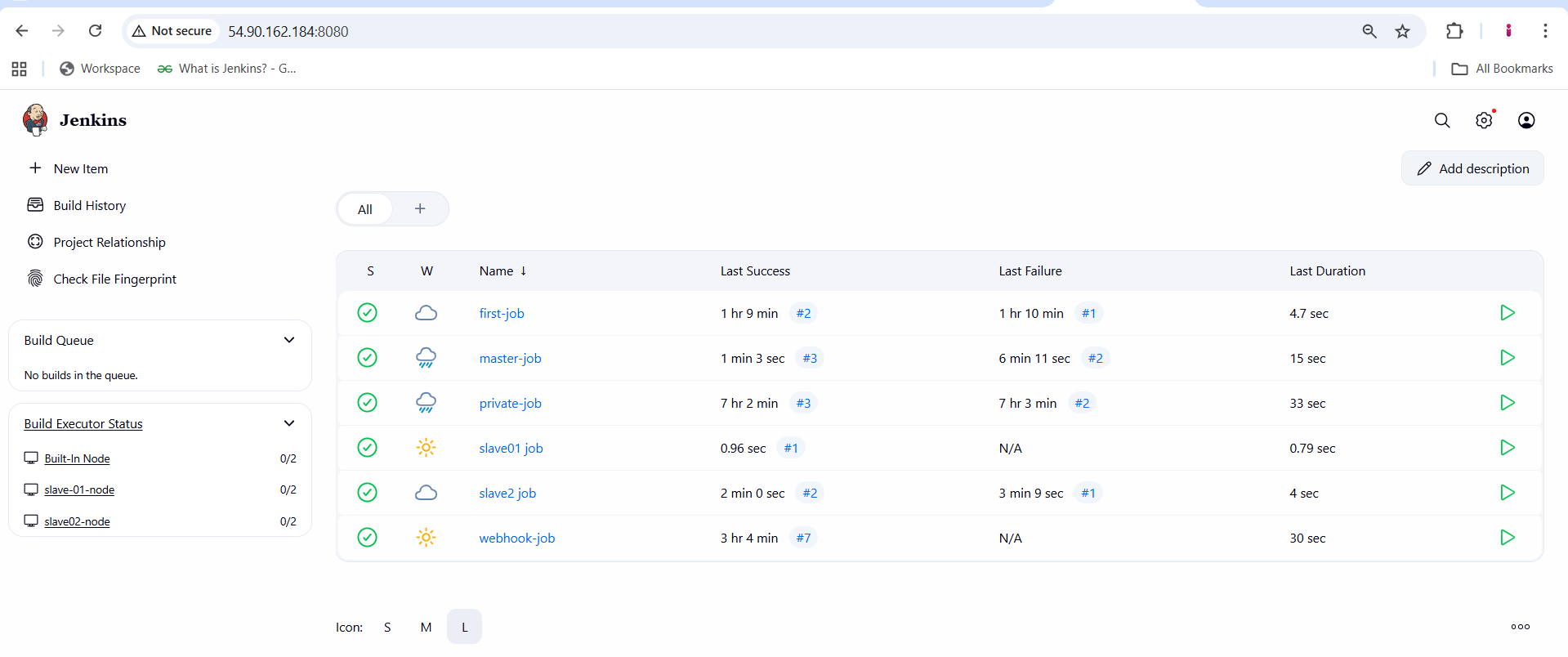
****

1. **Save → Jenkins should connect and show Online.**

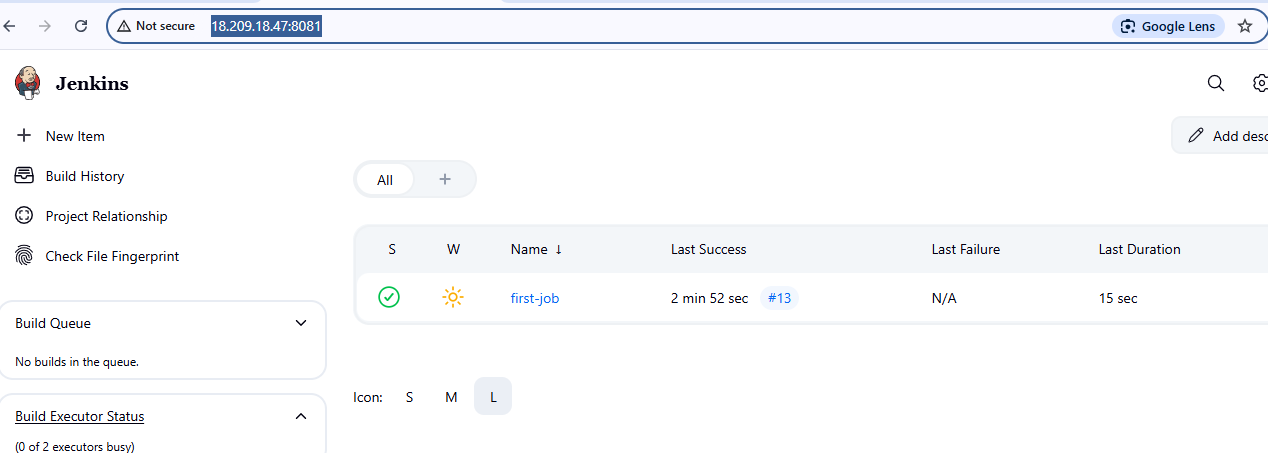
****

**Now you can see one master node and two slave nodes available.**

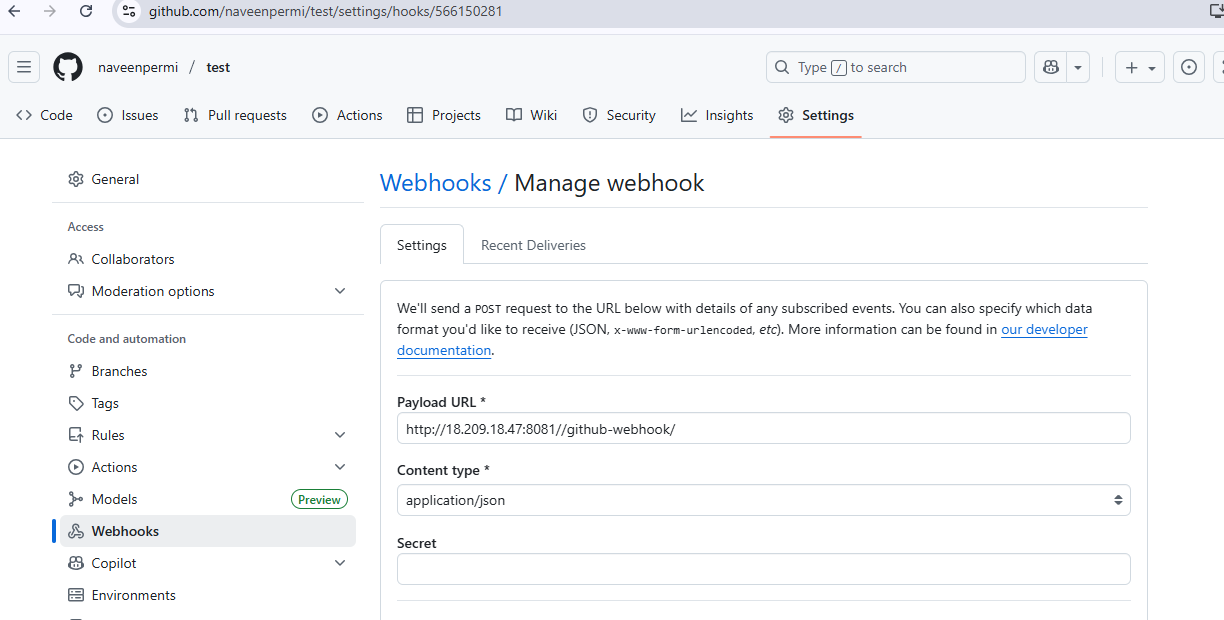
****

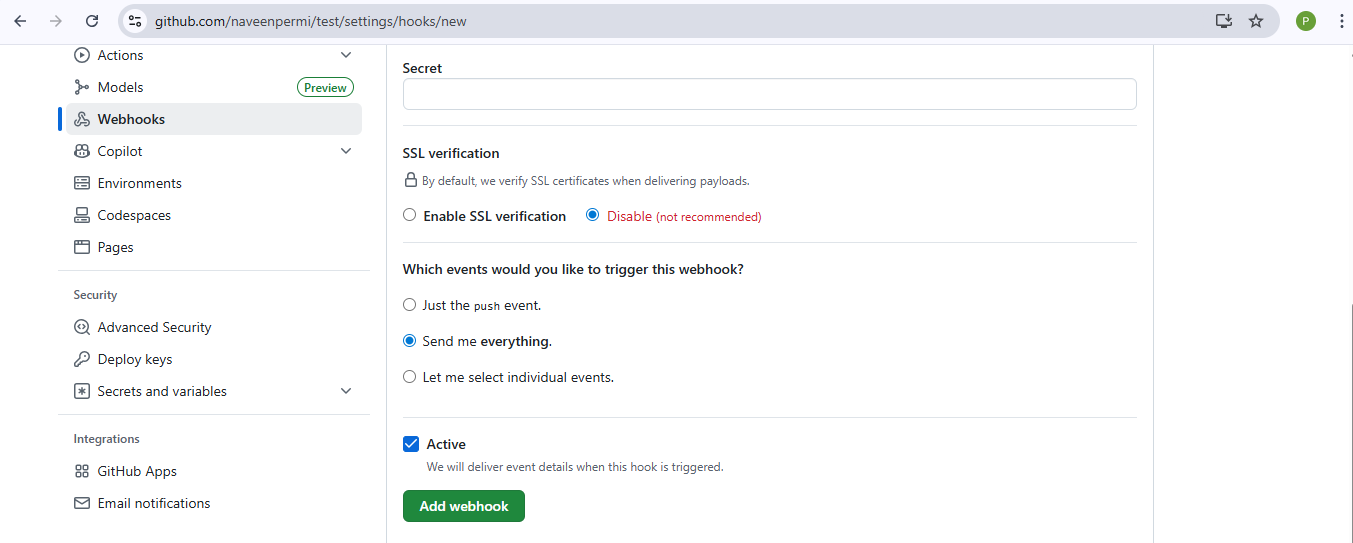
****

**2) Configure webhooks to Jenkins job**

First we need to copy **Jenkins url** 

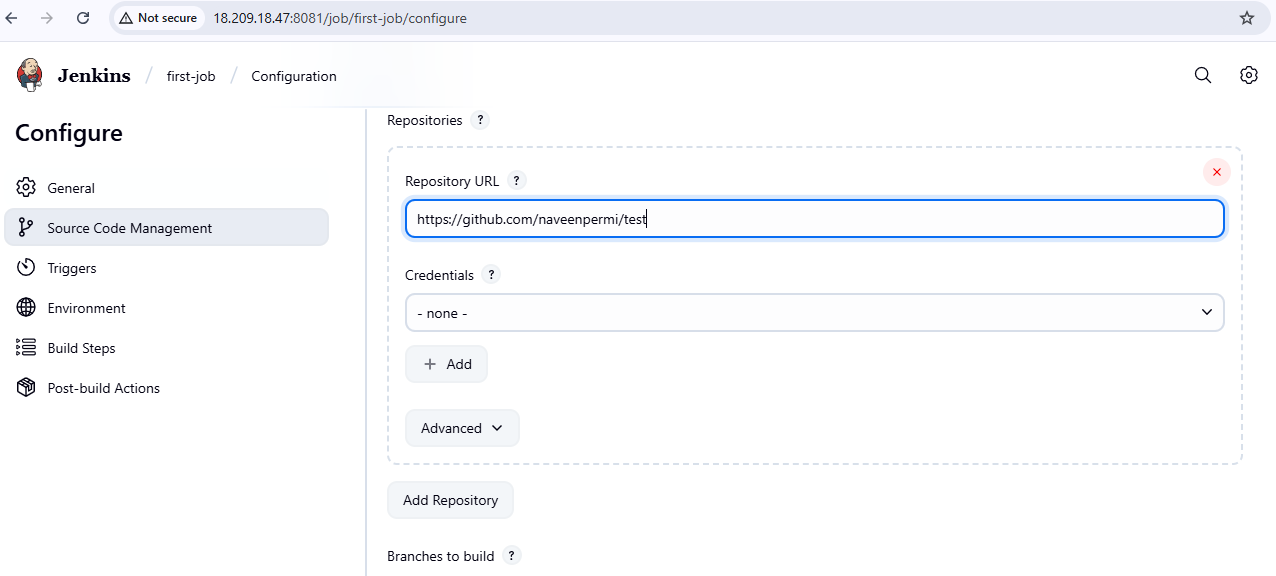
Go to **github repo → settings → webhooks → payload URL ([http://18.209.18.47:8081/github-webhook/](http://54.90.162.184:8080/github-webhook/)) → content type (application/json) →** select **send me everything →**click **Add webhook**

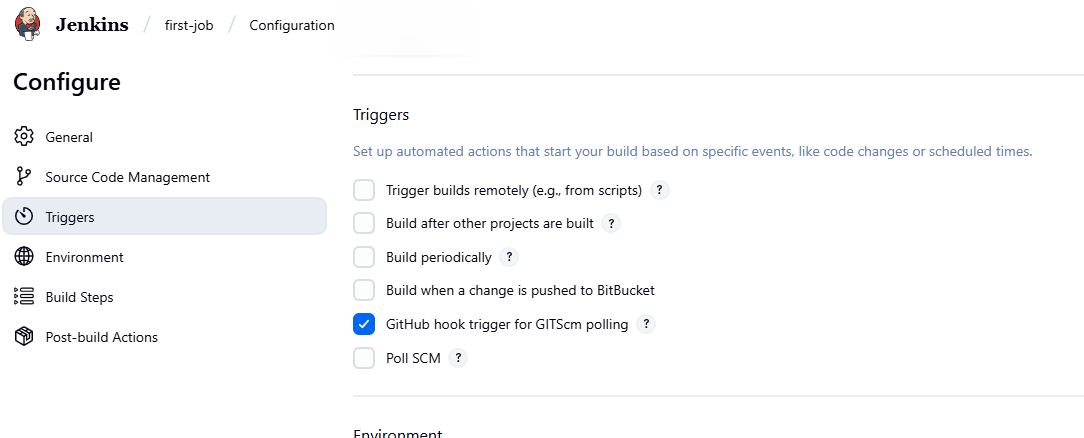


****

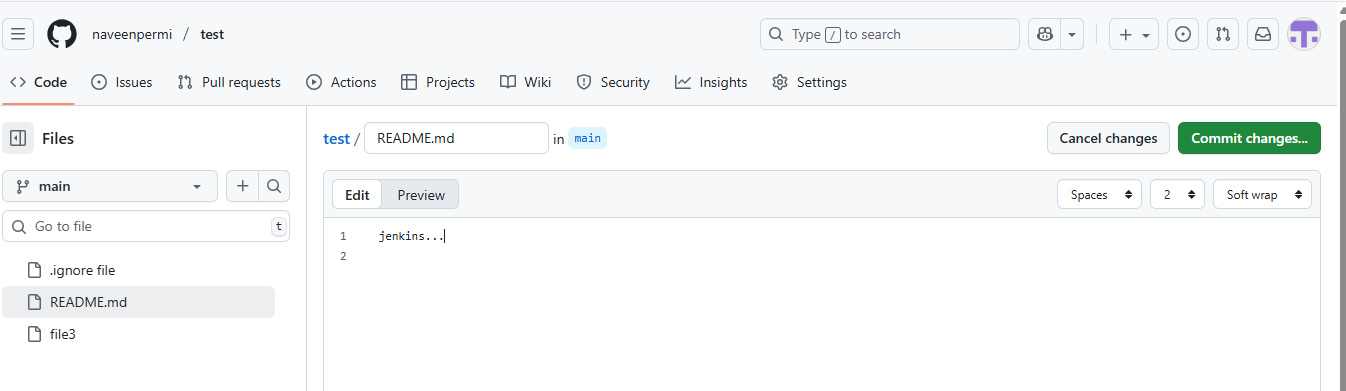
**Go to Jenkins →** select your job(**first-job) →** add **github u**r**l** in **git repository**

**→** select **Poll SCM** in **Triggers →** To keep **sleep 60** in **Execute shell →** click **save**

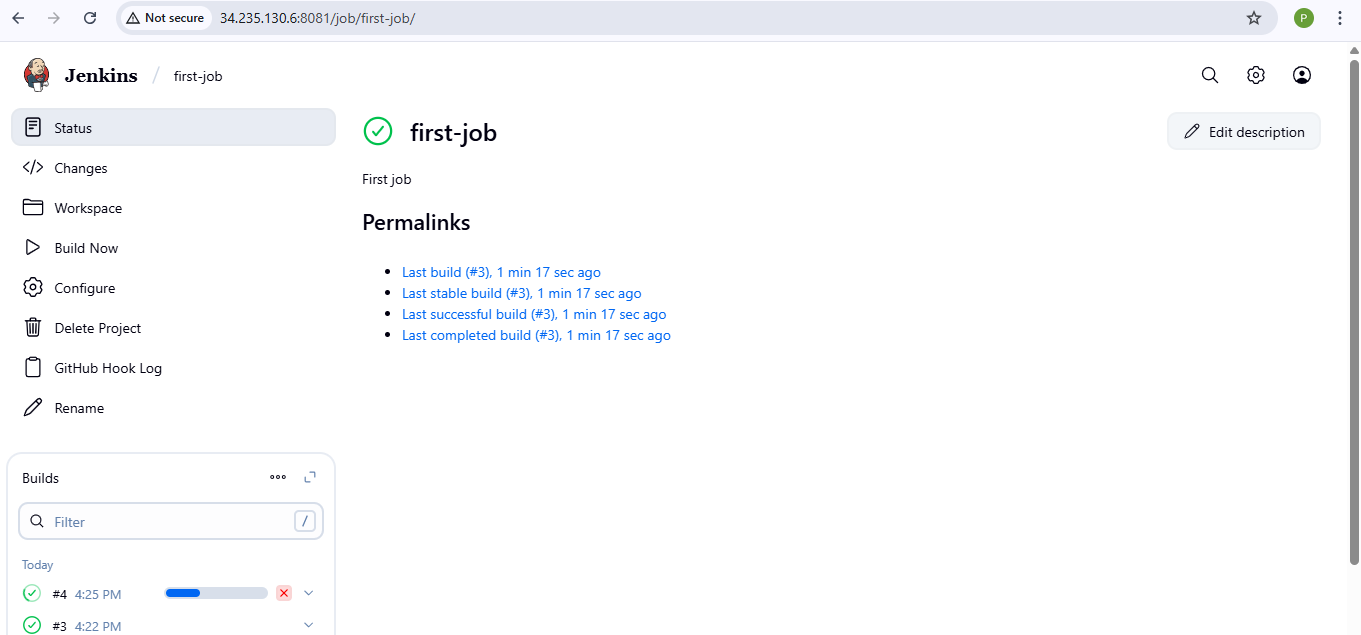




Now go to your **github repo** and **change or edit or update** the any file of the **source code and commit**



Then we see the **automatically job triggers** in **Jenkins**

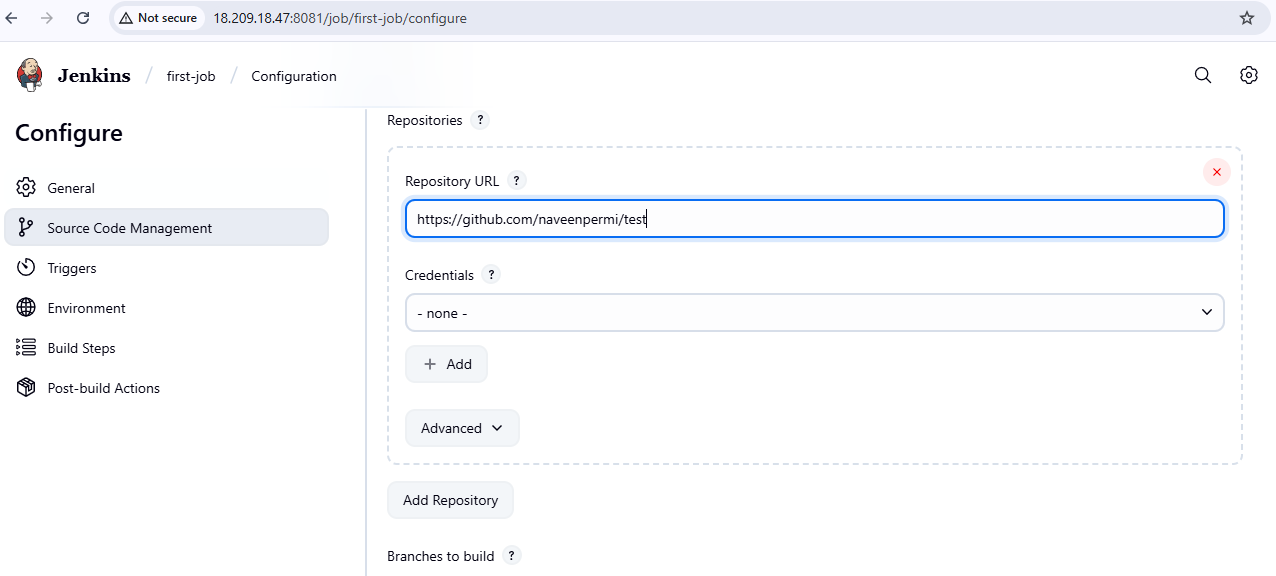


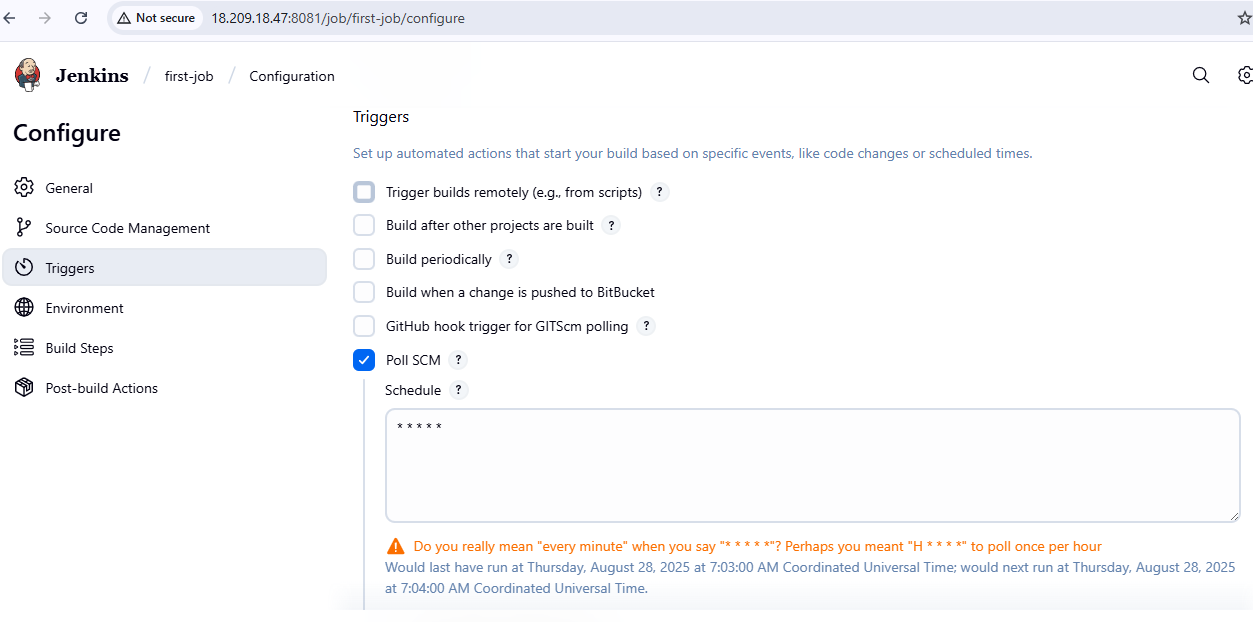
**3) Configure poll scm and build periodical options in Jenkins job.**

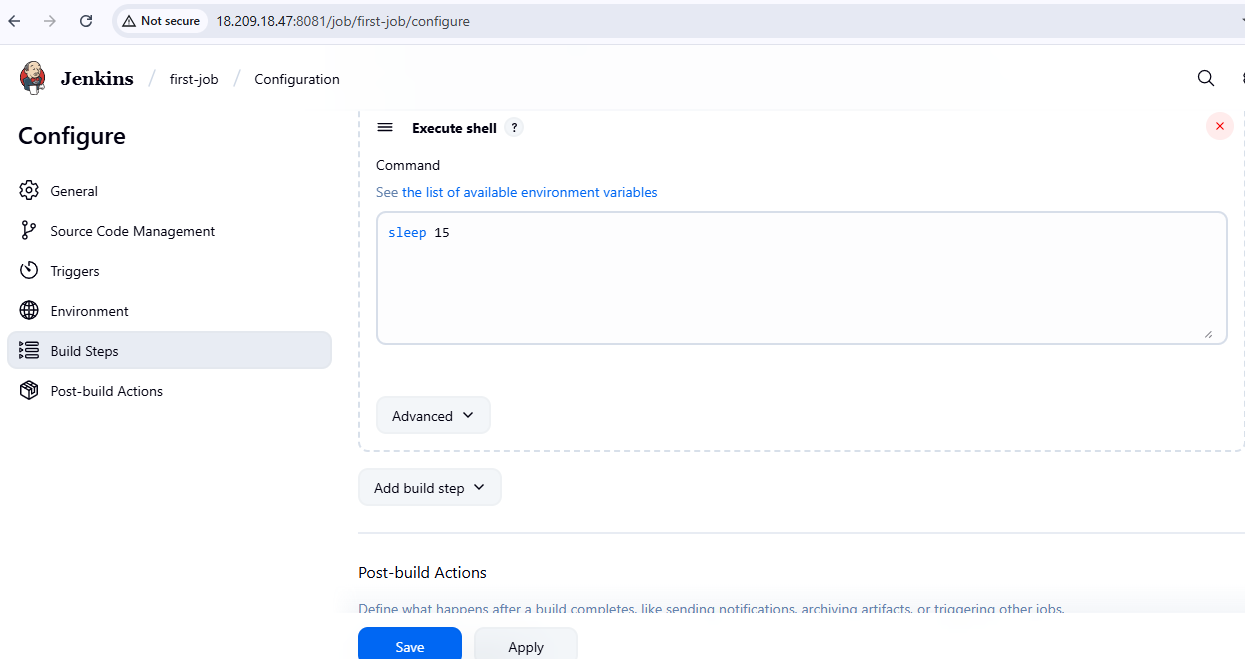
**i)To configure poll scm in Jenkins job.**

**Go to Jenkins →** select your job(**first-job) →** add **github u**r**l** in **git repository**

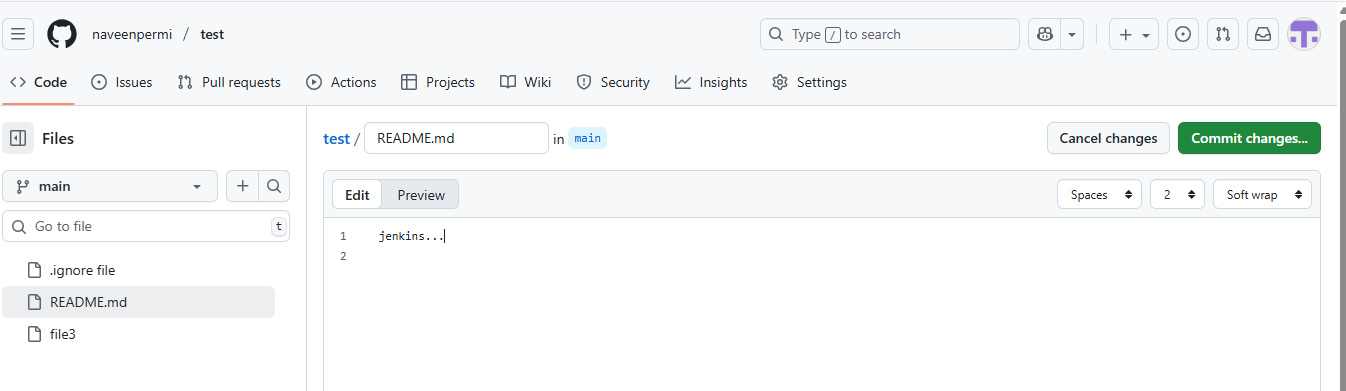
**→** select in **Triggers →** To keep **sleep 15** in **Execute shell →** click **save**



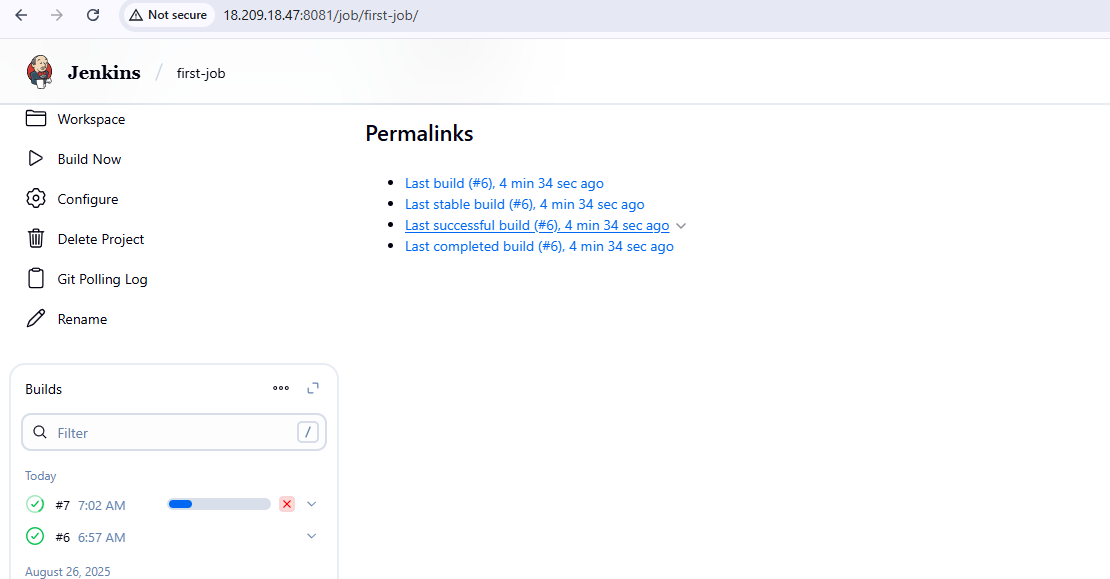




Go to **Github** to **commit changes**

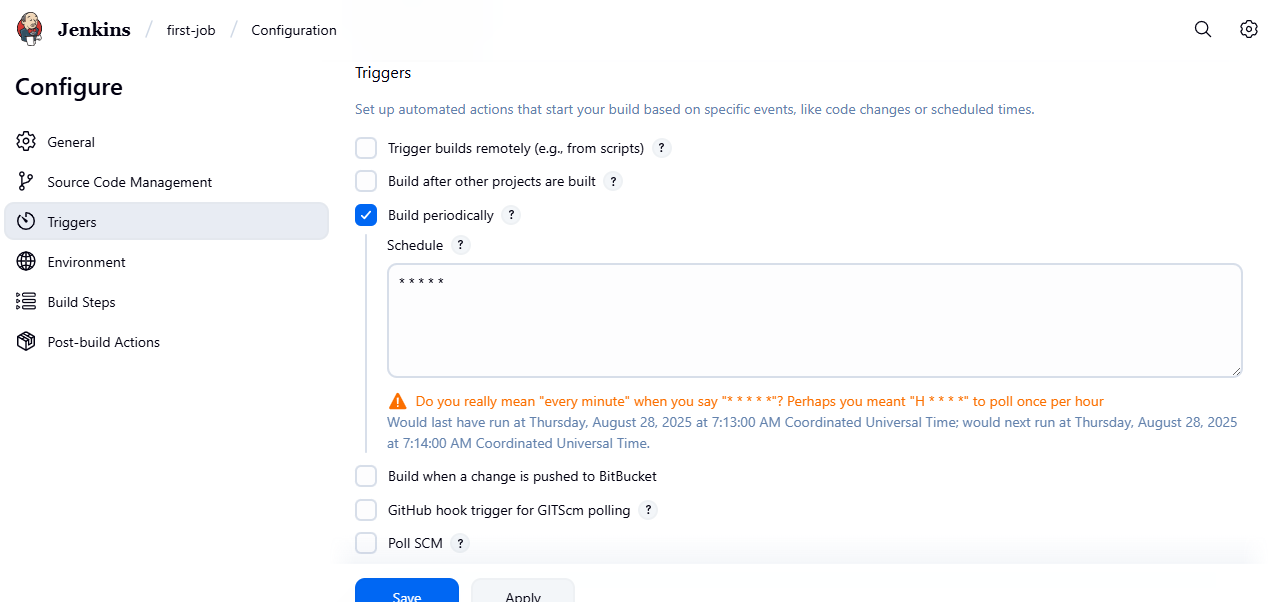


Then we see the **job trigger** in **Jenkins**

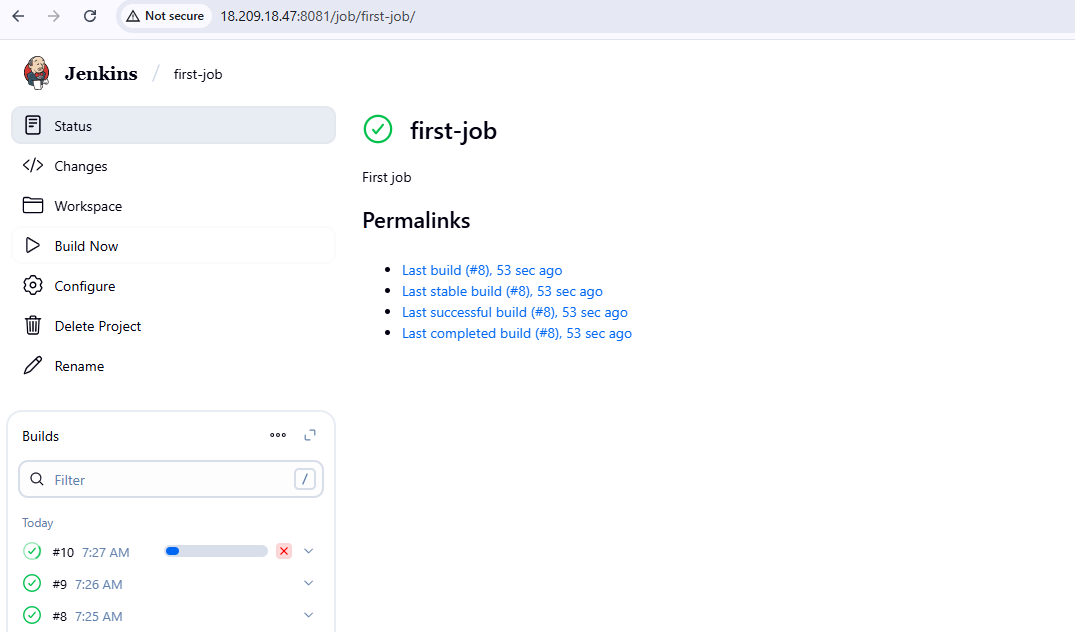


**ii)To configure build periodical options in Jenkins job.**

**Go to Jenkins →** select your job(**first-job) →** add **github u**r**l** in **git repository →** select **Build periodically** in **Triggers →** To keep **sleep 15** in **Execute shell →** click **save**

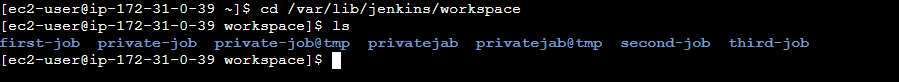


Then we can see **it triggers continously without commit changes**



**You can check files in our EC2 jenkins file.**

**cd /var/lib/jenkins/workspace**



**4) Take backup of Jenkins server by using bash script.**

Create a script

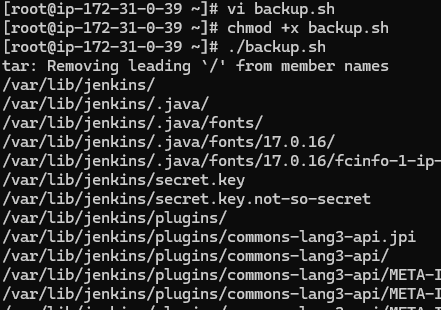
**vi jenkins-backup.bash**

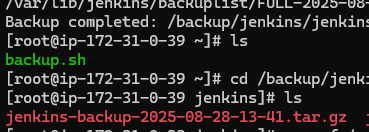
|  |
| --- |
| **#!/bin/bash**  **BACKUP\_DIR="/backup/jenkins"**  **JENKINS\_HOME="/var/lib/jenkins"**  **DATE=$(date +%F-%H-%M)**  **mkdir -p $BACKUP\_DIR**  **tar -czvf $BACKUP\_DIR/jenkins-backup-$DATE.tar.gz $JENKINS\_HOME**  **echo "Backup completed: $BACKUP\_DIR/jenkins-backup-$DATE.tar.gz"** |

give permission

**chmod 700 backup.bash**

**./backup.bash**

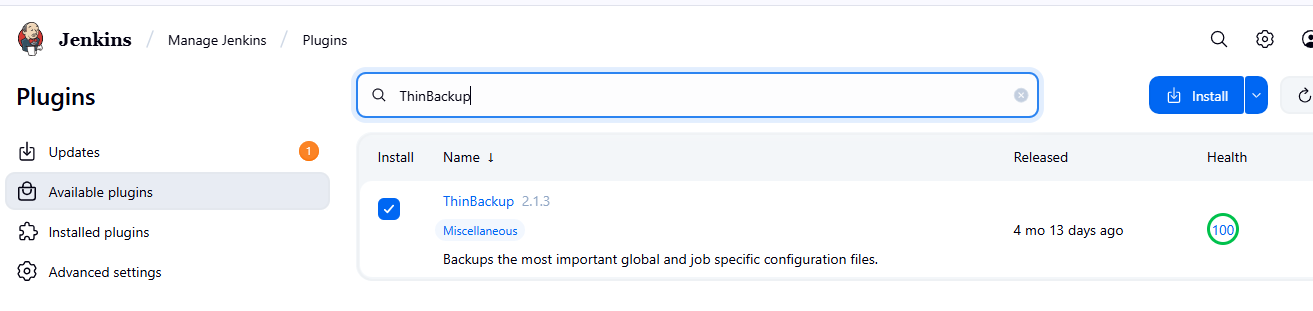




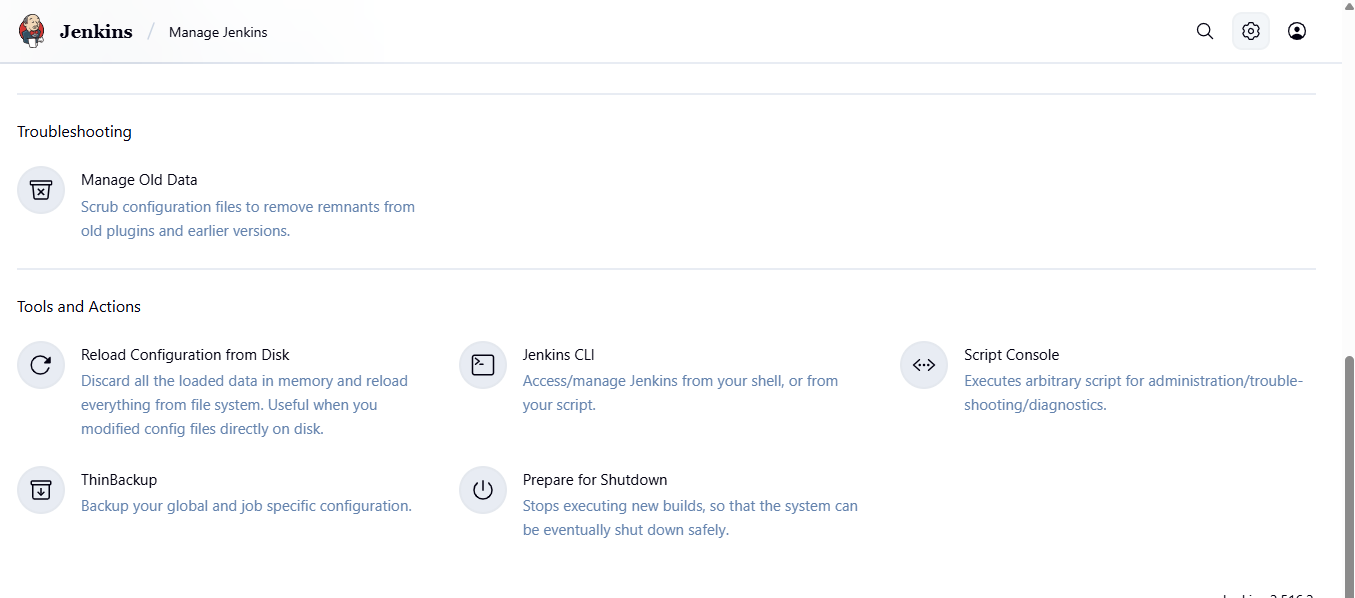
**5) Take backup of Jenkins using rethin backup plugin.**

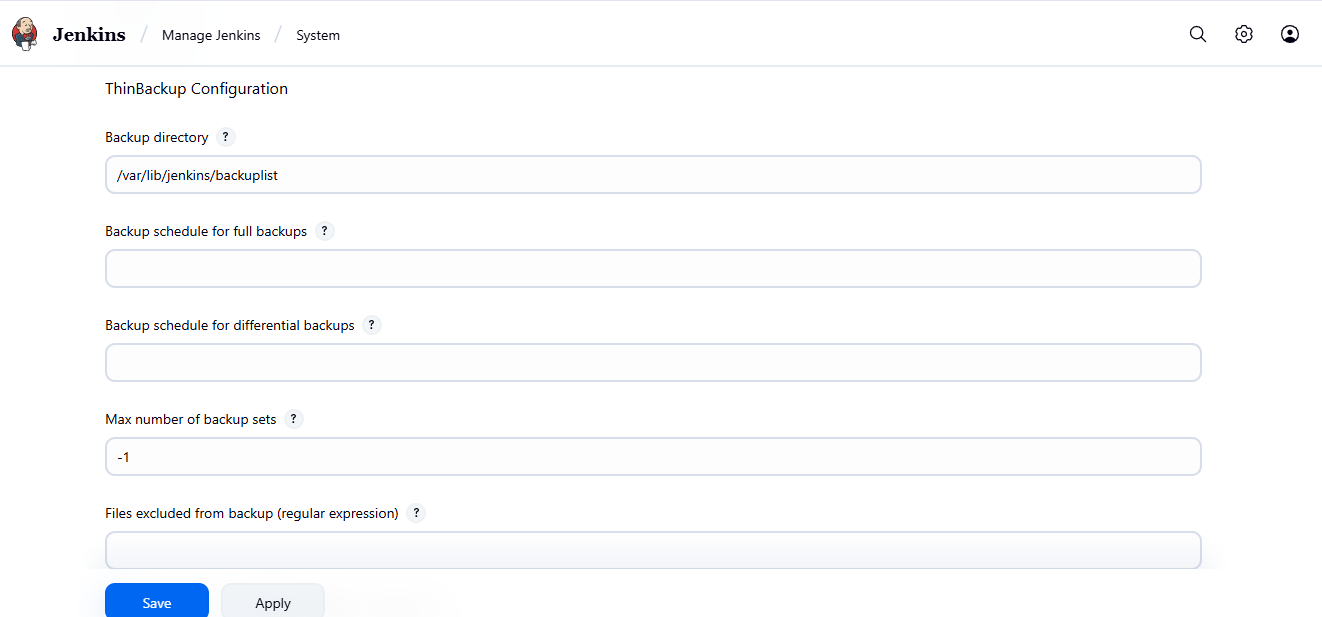
Install **thinbackup** from **plugin** in **jenkins**

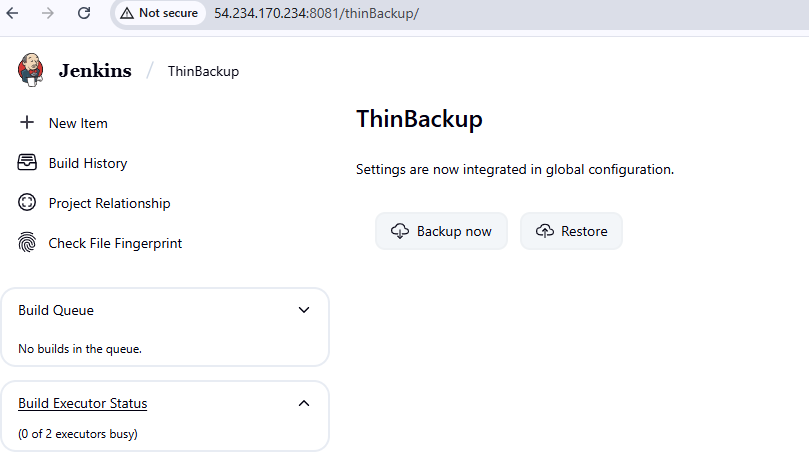
Go to **Jenkins --> Manage Jenkins --> Plugin**

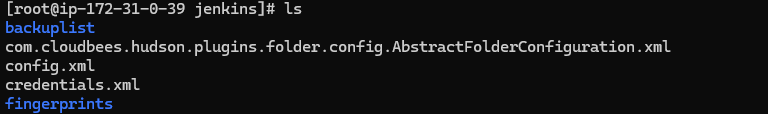


In **tools and actions** you get **Thinbackup**



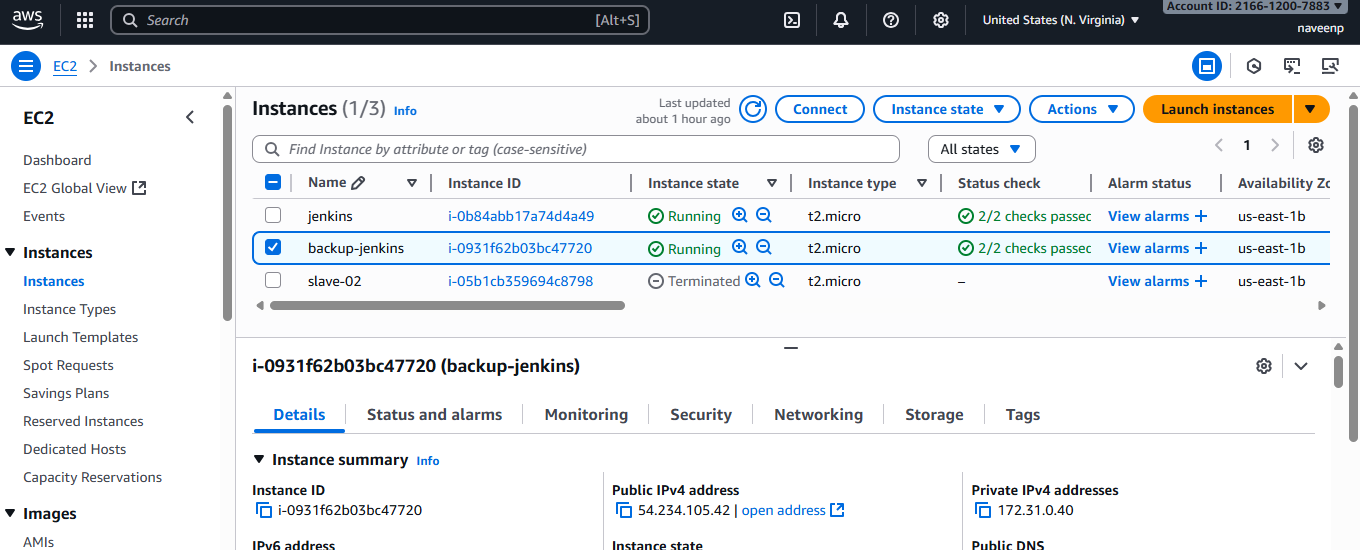






**6) Setup a new Jenkins server and dump the backup taken in task4.**

**Launch a new EC2 instance(backup-jenkins)**



#Install Java (required for Jenkins)

**sudo dnf install -y java-17-amazon-corretto**

#Add Jenkins Repository

****sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo****

****sudo rpm --import** <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>**

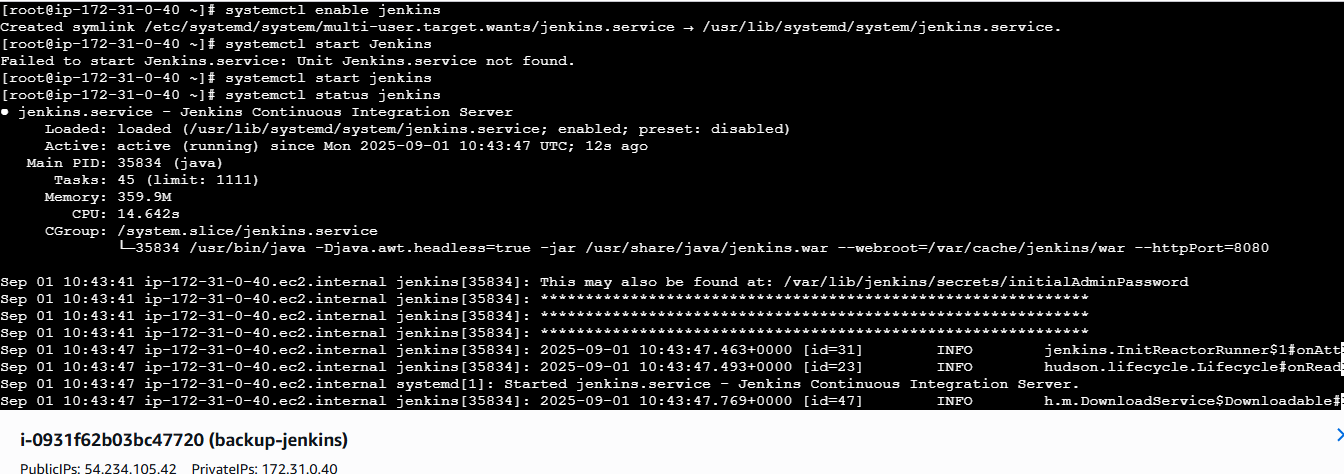
#Install Jenkins

****sudo yum install jenkins –y****

****systemctl enable jenkins****

****systemctl start Jenkins****

****systemctl status jenkins****



**Step 2:** Copy Your Backup to New Server

Login into your old main server you took backup

Create an test.pem file where you copy your pem key of new-server and and paste in the oldserver

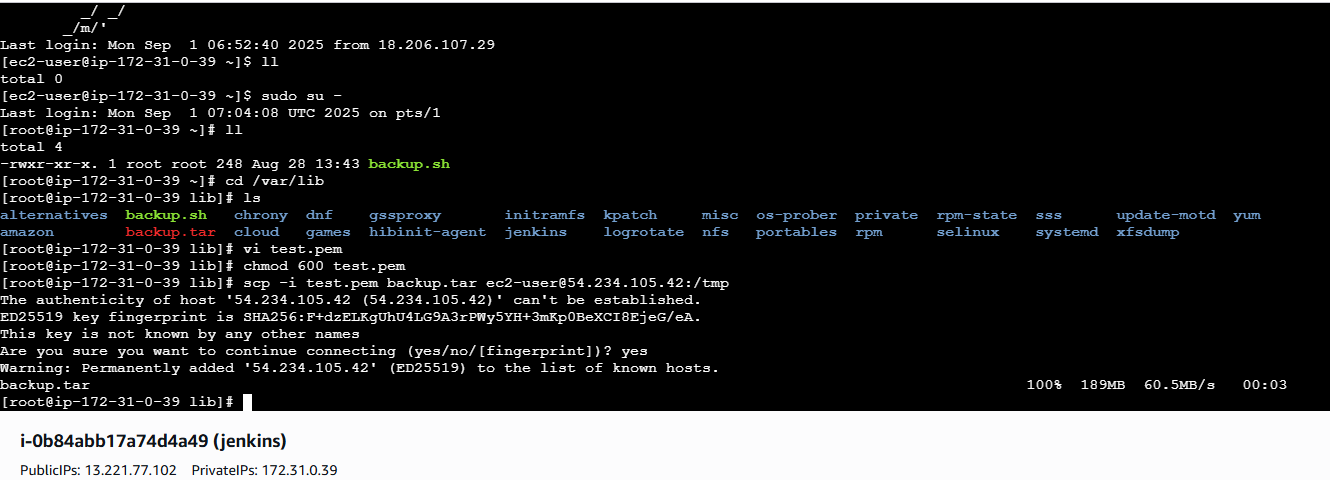
**scp -i test.pem backup.tar [ec2-user@44.203.148.202:/tmp](mailto:ec2-user@44.203.148.202:/tmp/)**

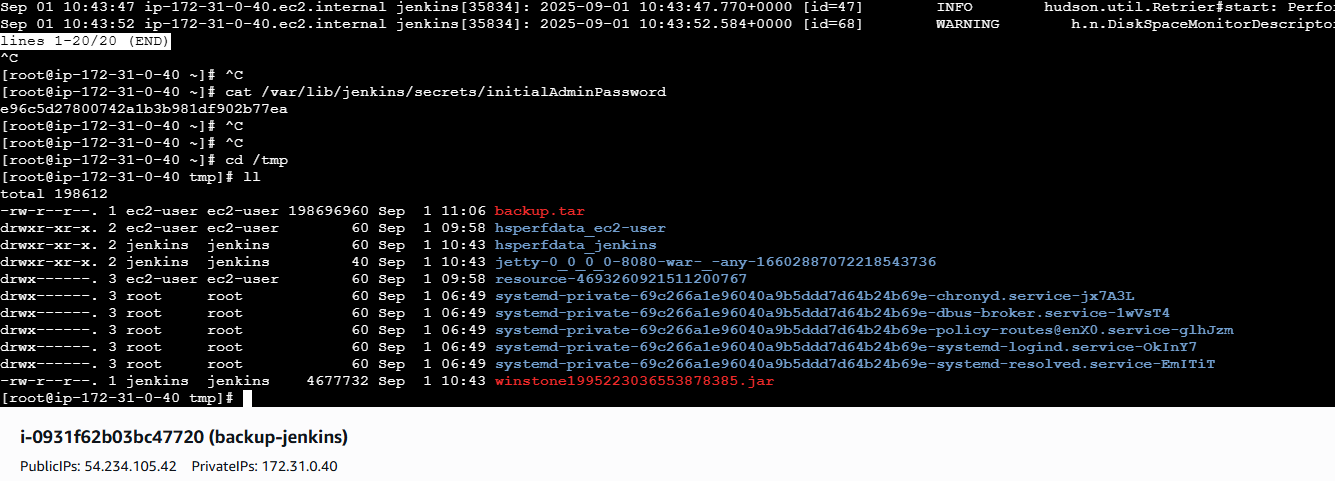
**test.pem**  - is your new server pem key

**backup.tar** - backup file

**[ec2-user@54.234.105.42](mailto:ec2-user@44.203.148.202)** - public ip of new server

**/tmp**  – is the location in new server where you send your backup tar file



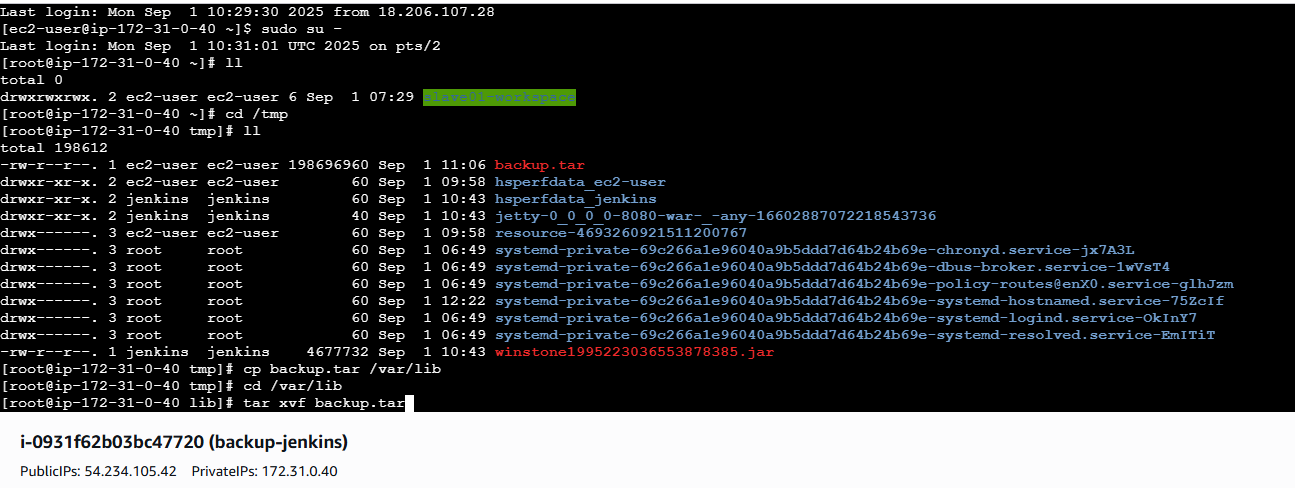


**Step 3:** Login in to new server and check the file in /tmp

Copy the file from tmp to /var/lib

**cp backup.tar /var/lib**

**tar xvf backup.tar**



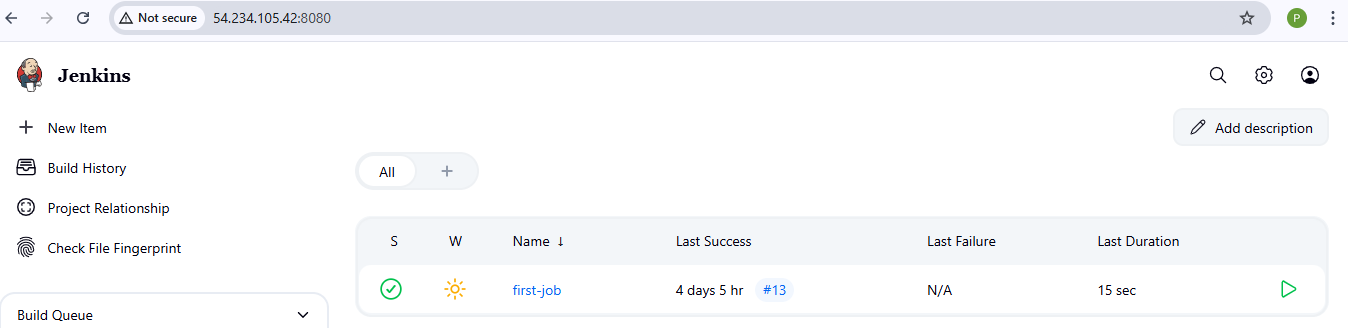
**sudo systemctl start jenkins**

**sudo systemctl status Jenkins**

Verify in browser

http://<EC2-Public-IP>:8080

new server by default running on port no-: 8080



**Old server running on port –no. :- 8081**

