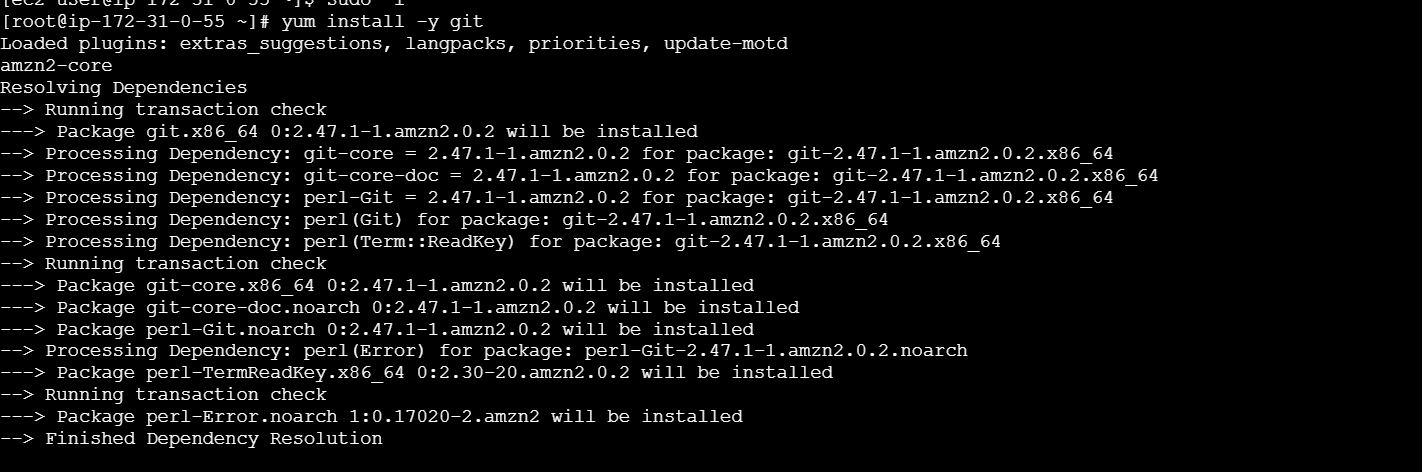
**Jenkins task for Day-02**

**1) Configure 2 slave machines in jenkins master.**

First create an instance named slave-01 and install java and git in it

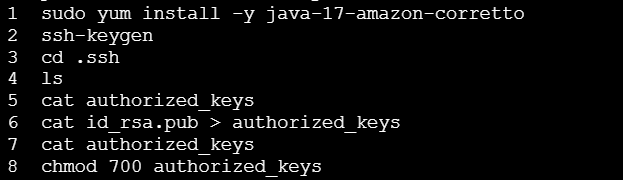


Now do

**ssh-keygen**

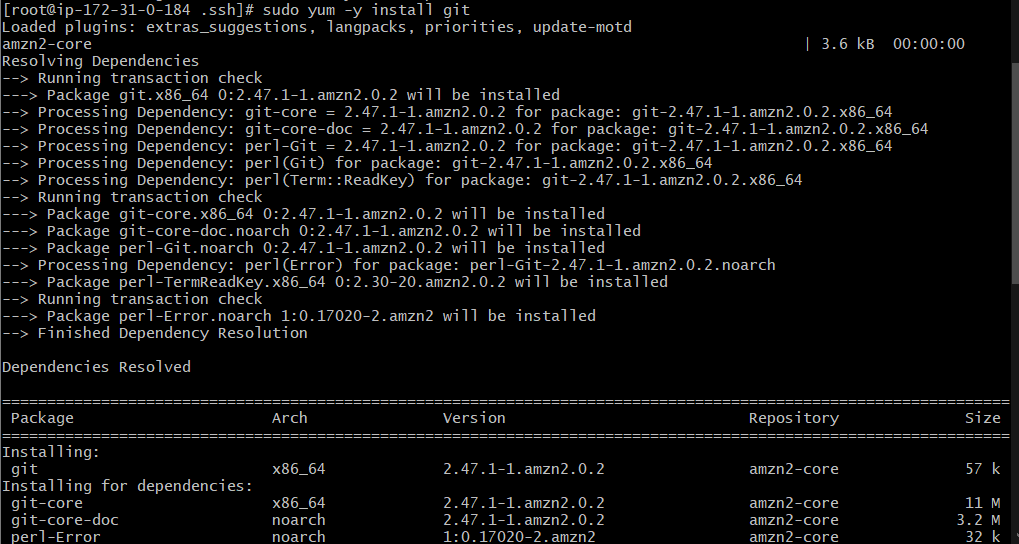
**cd .ssh**

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

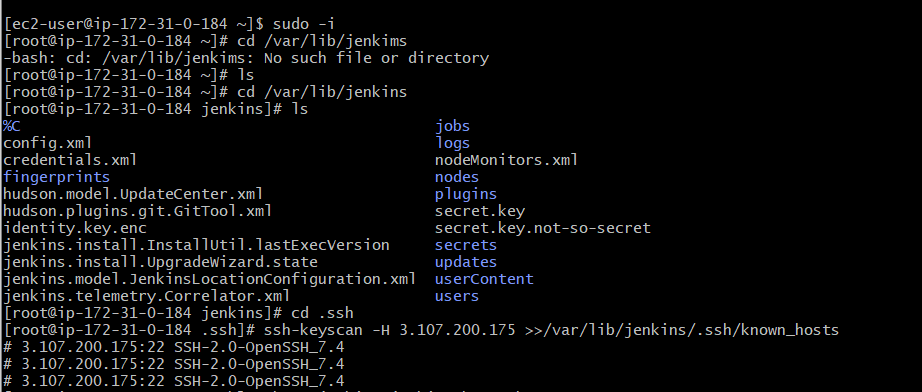
**chmod 700 authorized\_keys**

Now connect with other instance named Jenkins-master

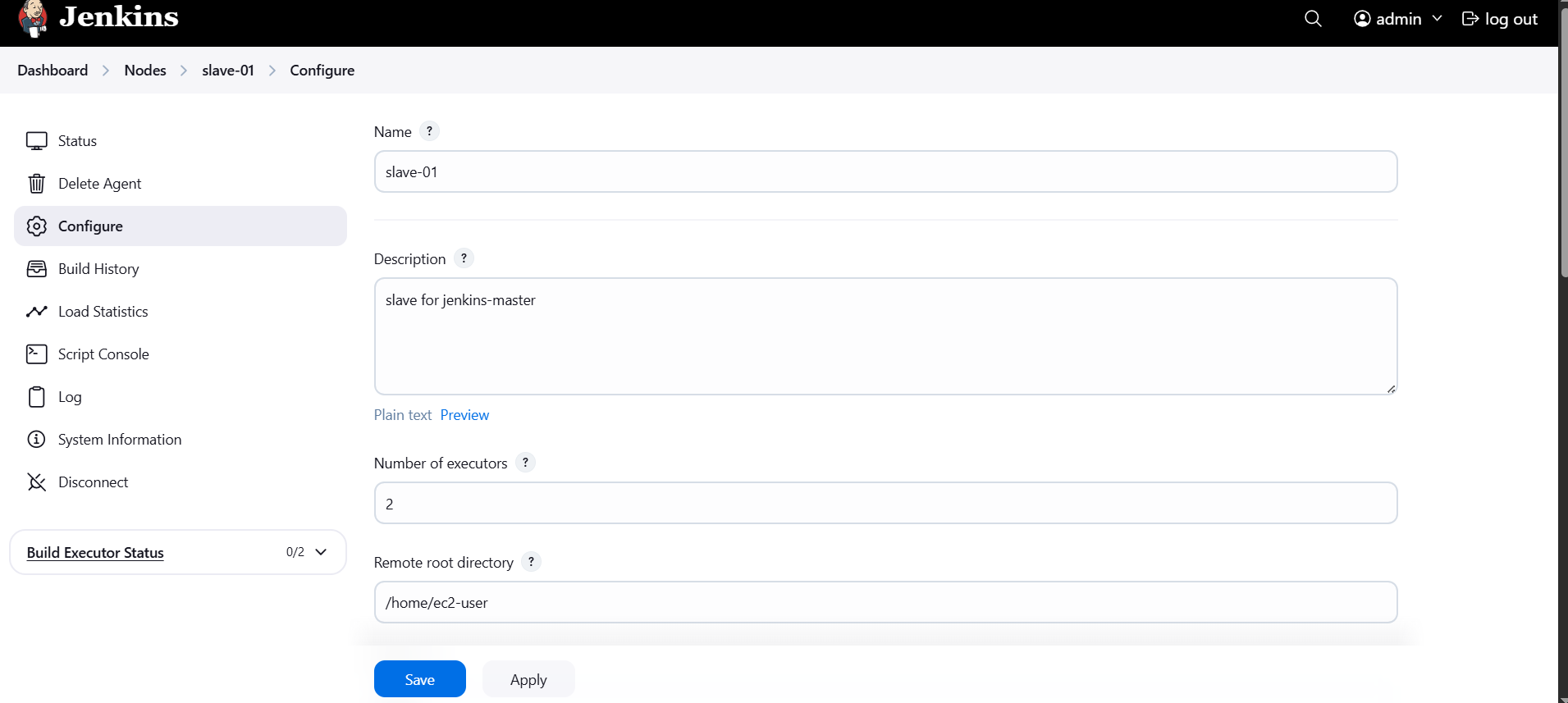
Install java, Jenkins and git in it

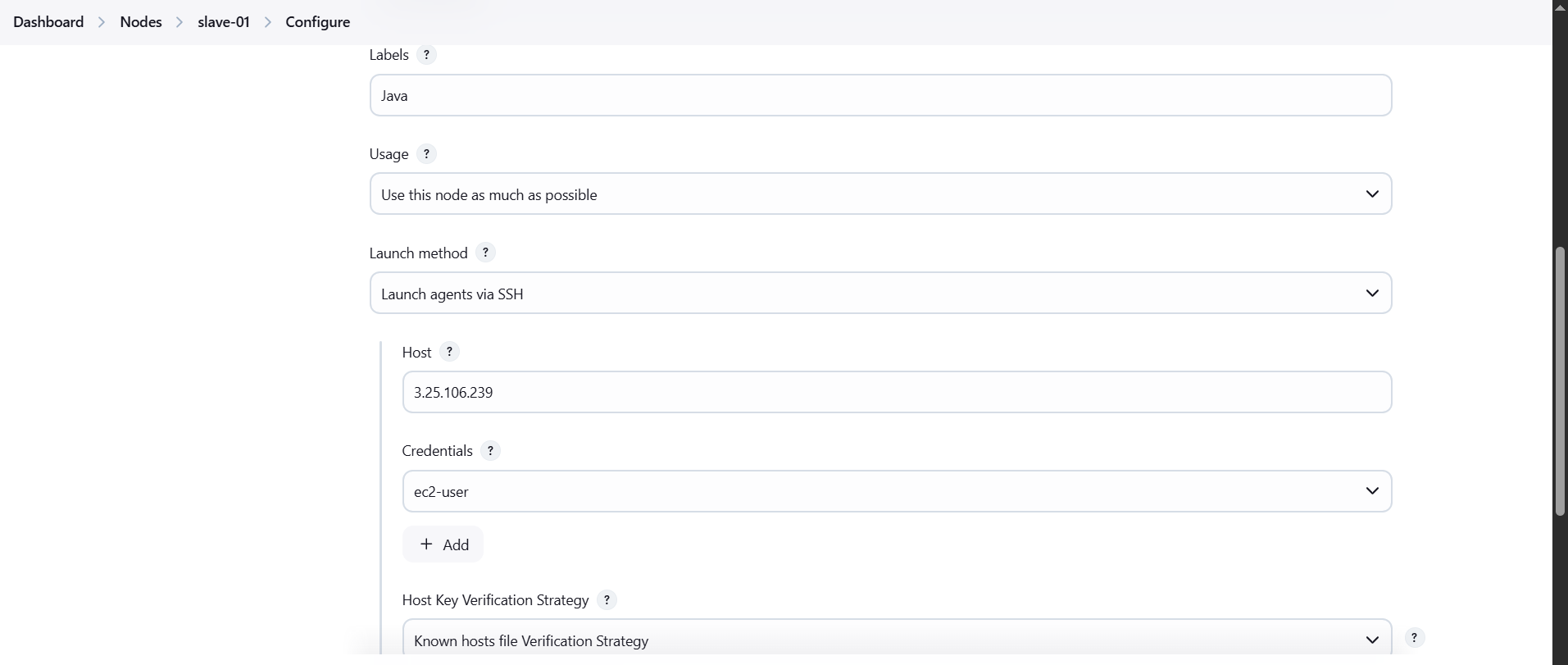


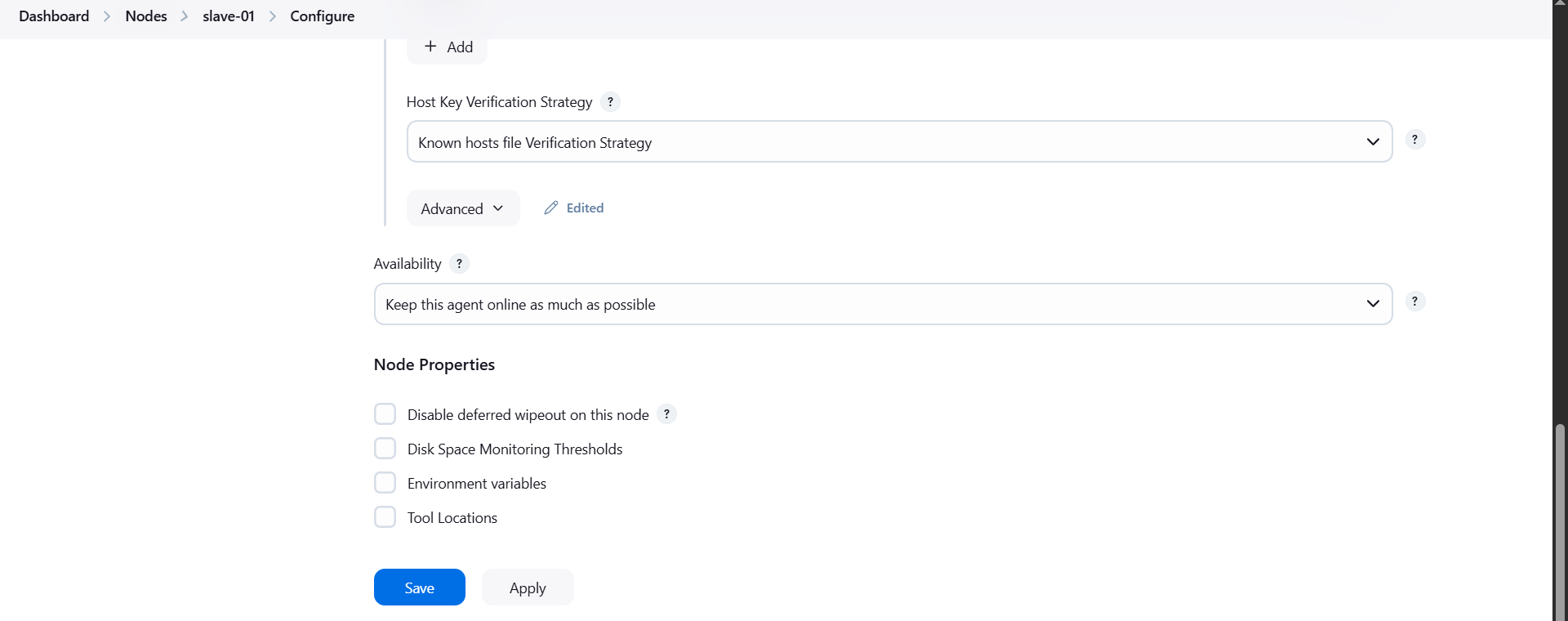
Now scan the slave01 Public IP and copy to the Jenkins-master under the directory known\_hosts

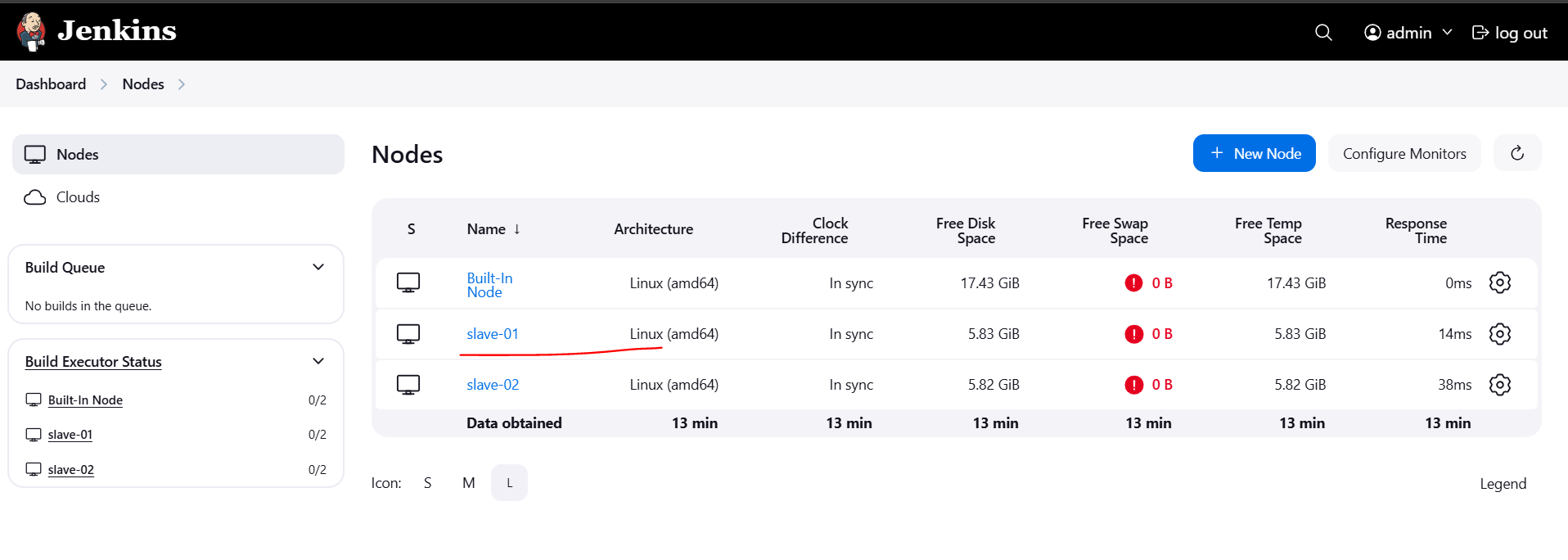


Now go to Jenkins GUI and create the node for slave01

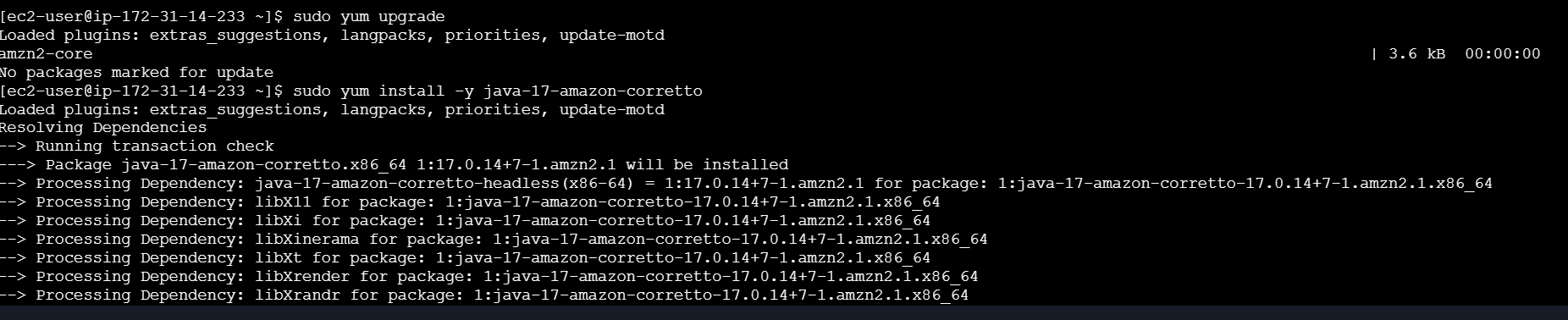








Similarly follow the same steps for the second instance named slave02 and install java and git in it



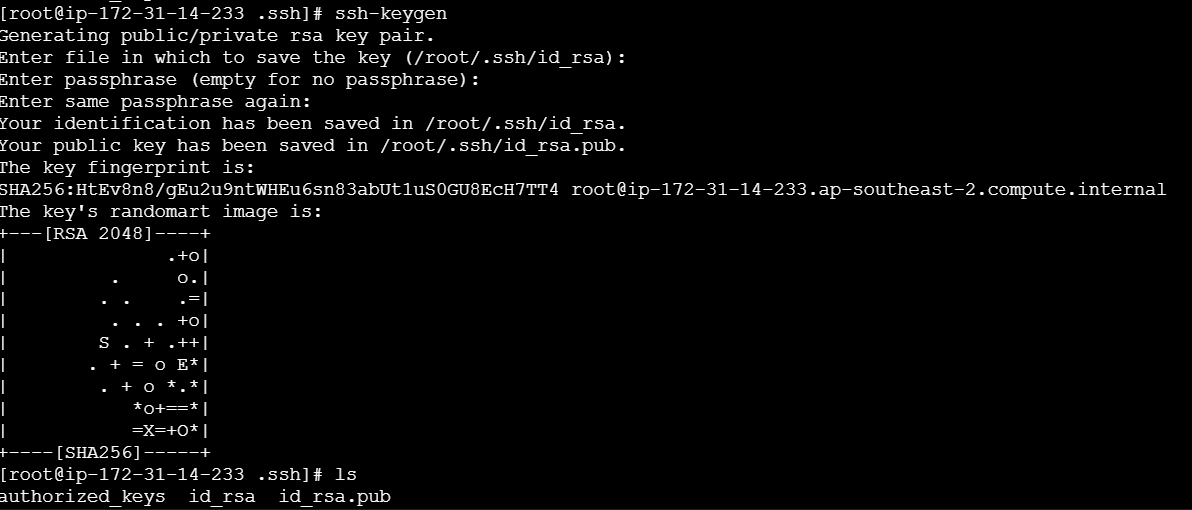
Now do

**ssh-keygen**

**cd .ssh**

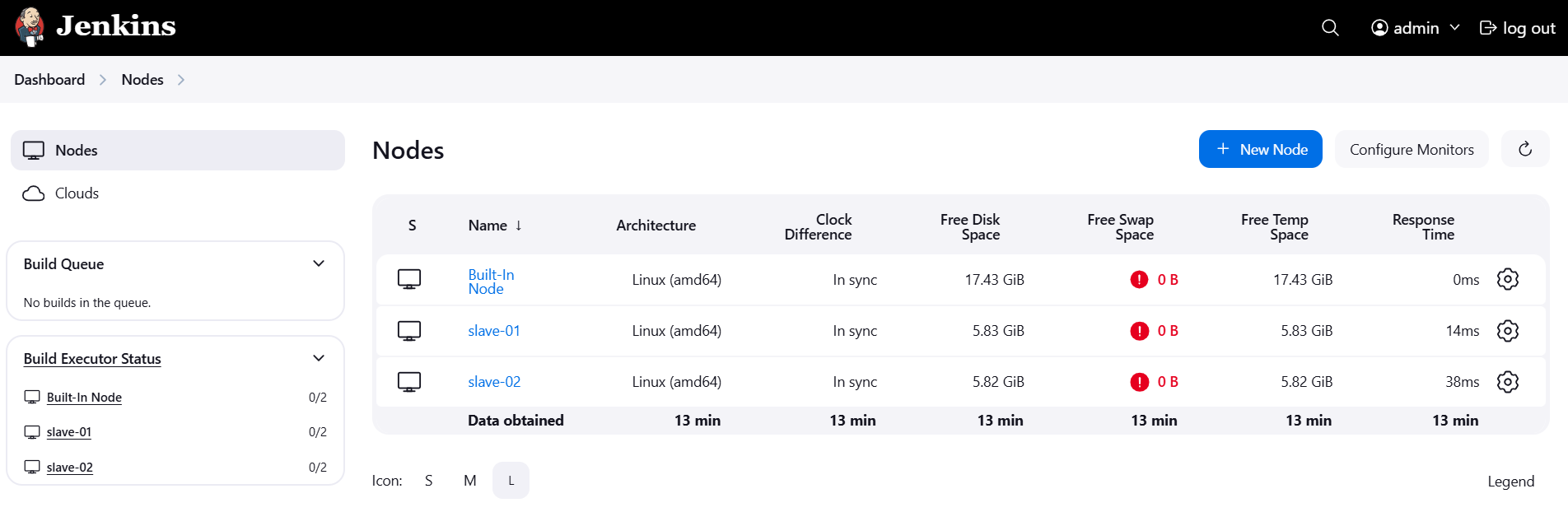
**cat id\_rsa.pub > authorized\_keys**

**chmod 700 authorized\_keys**

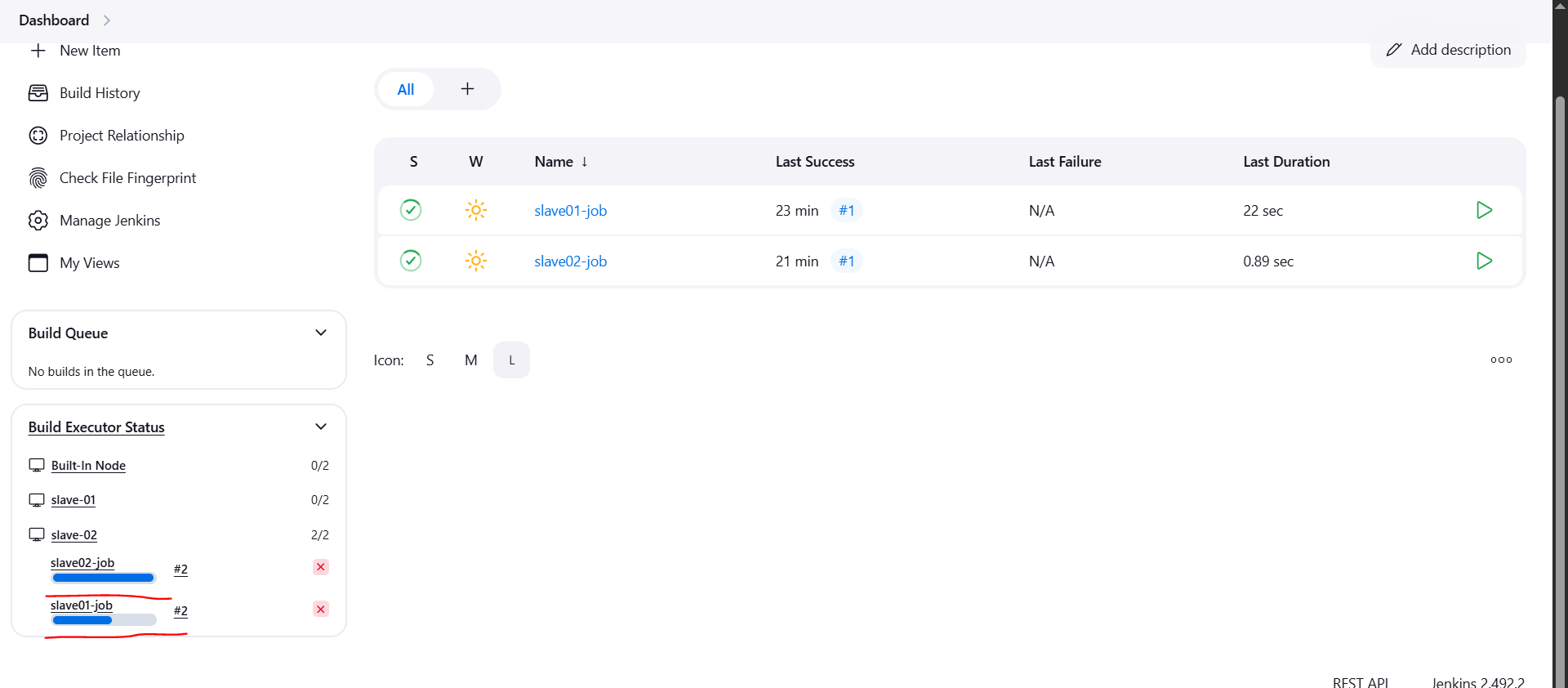




Now add node in Jenkins GUI like how we did for slave-01



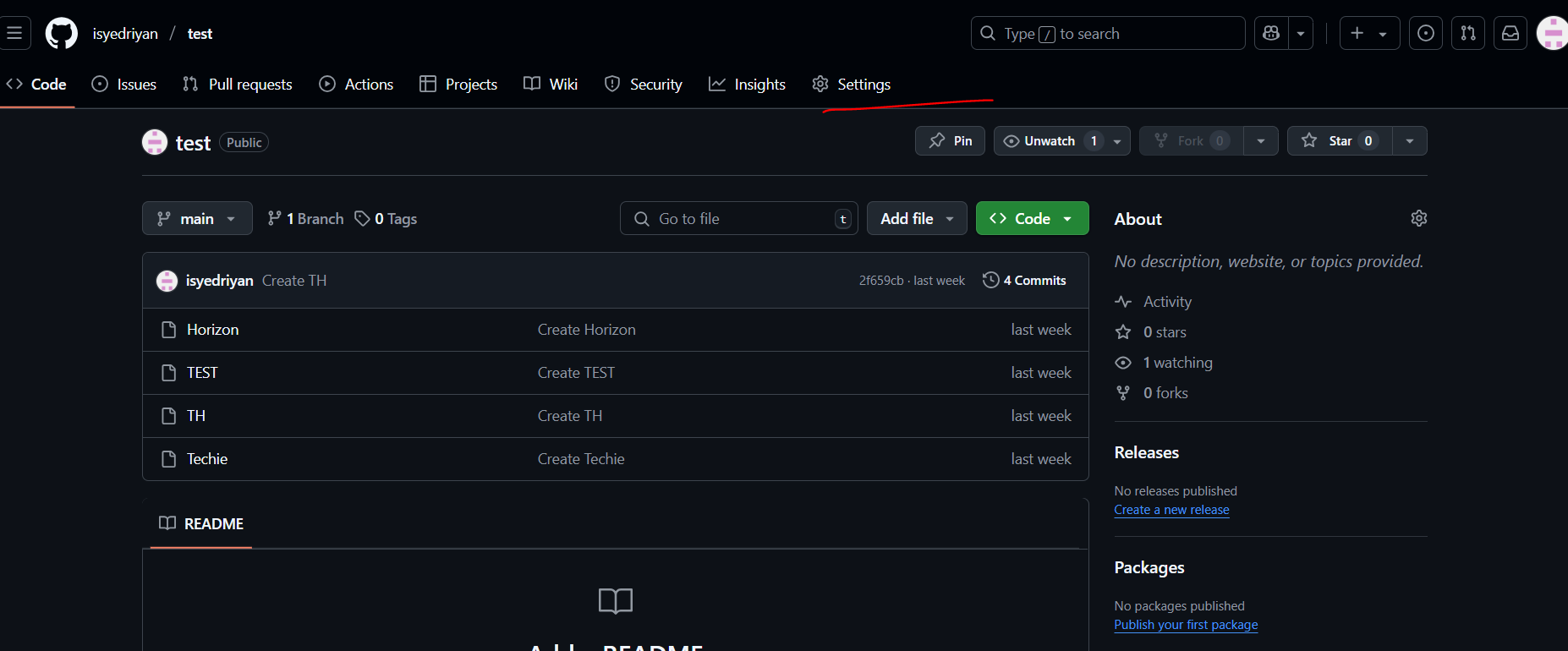
Now create 2 jobs and start executing the jobs

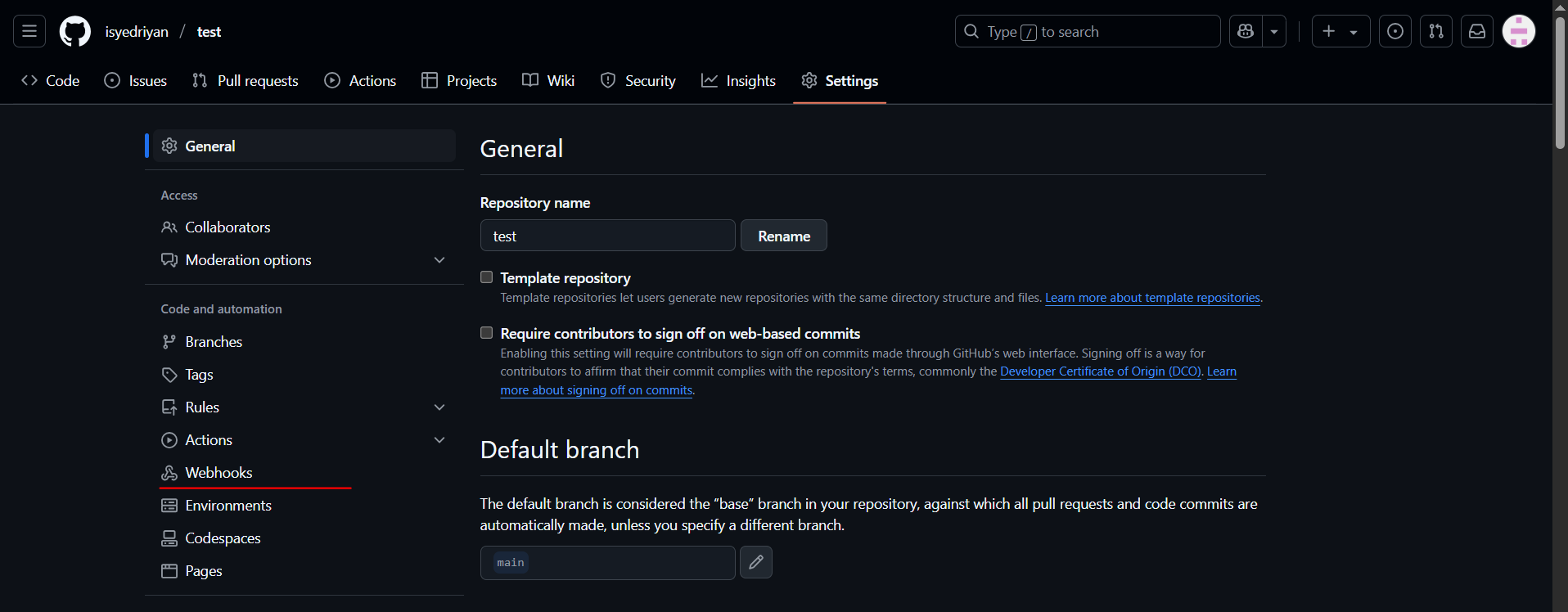


**2) Configure webhooks to jenkins job.**

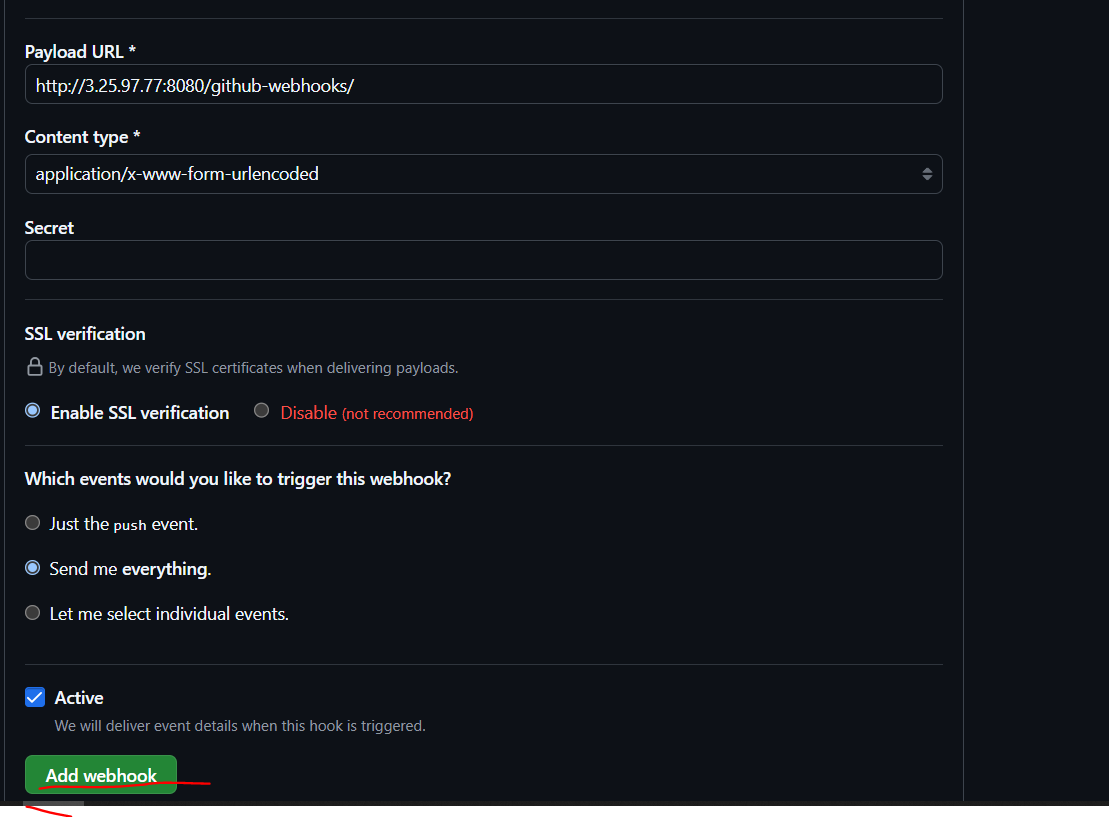
Create a repo in github and add webhooks for that repo

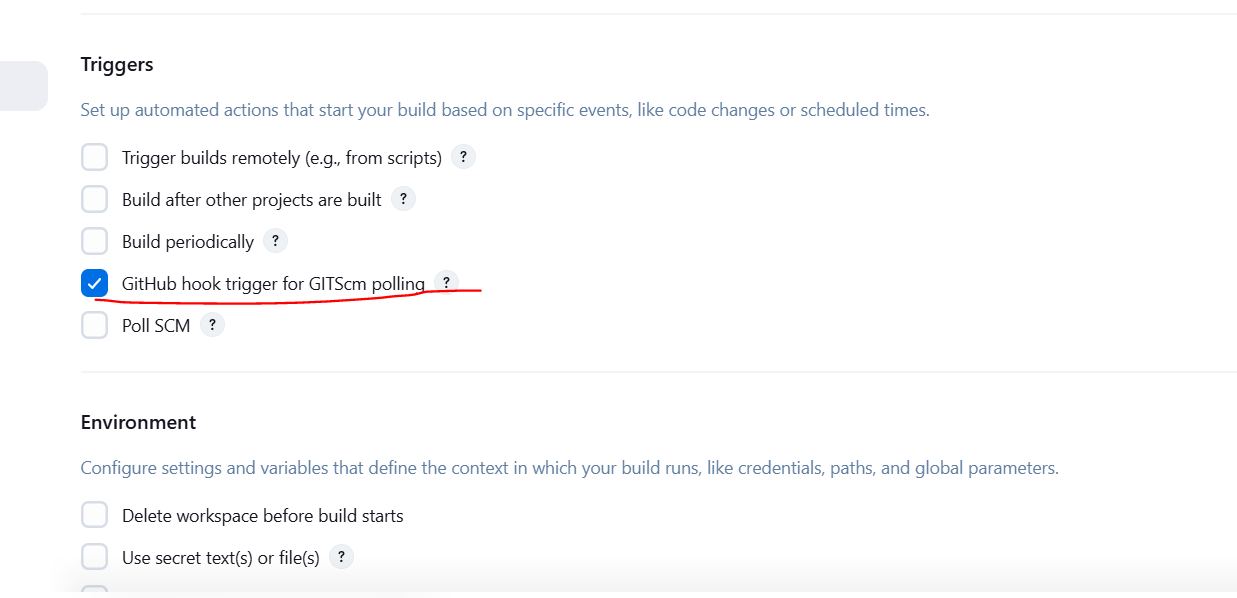
Go to settings



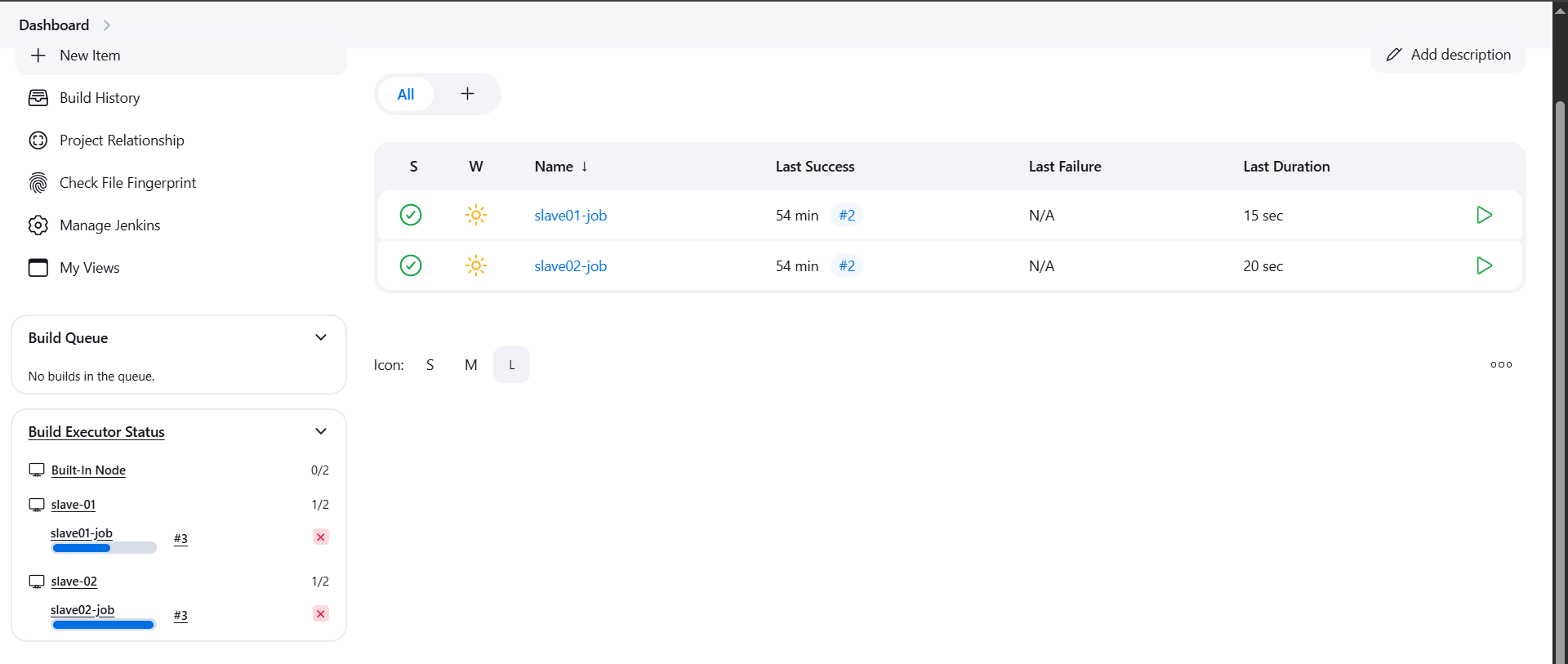


Now go to Jenkins GUI copy the url till 8080 and add in payload url





Now add some files in the Github repository then the jobs in the Jenkins will trigger automatically



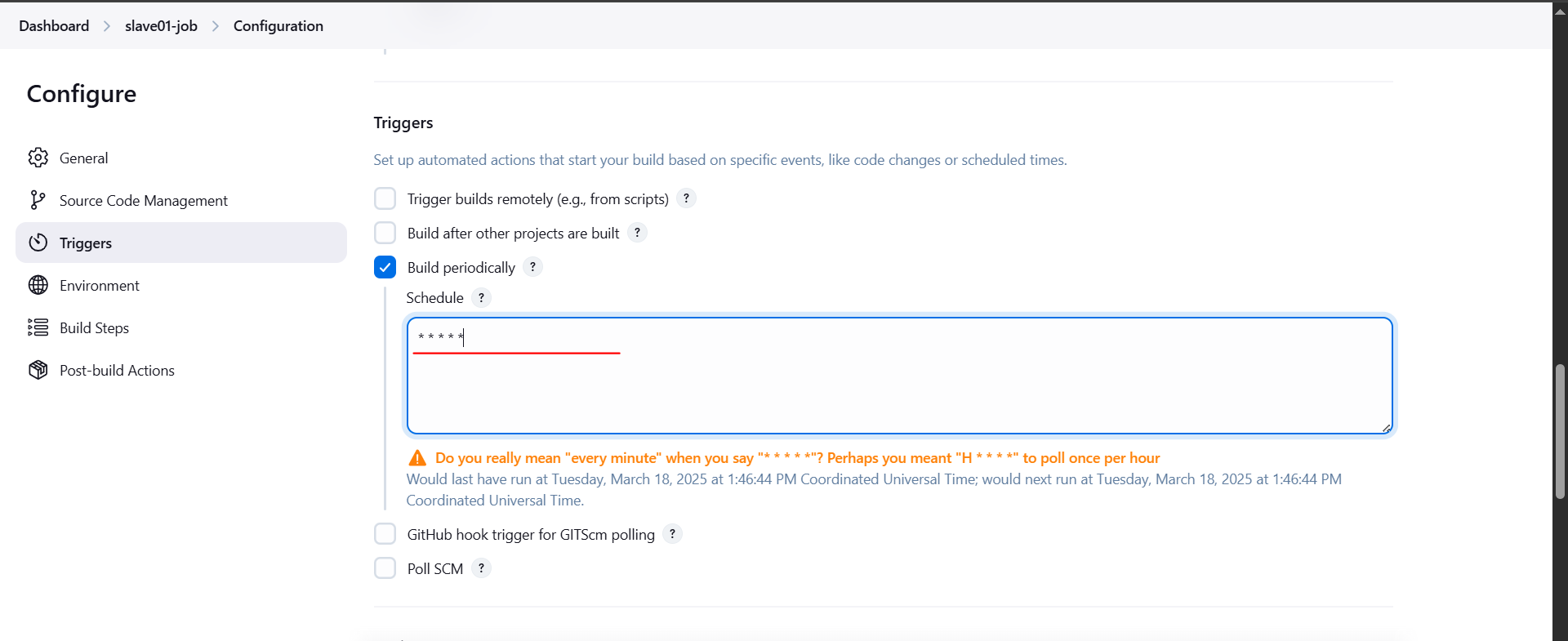
**3) Configure poll scm and build periodical options in jenkins job.**

To build periodically

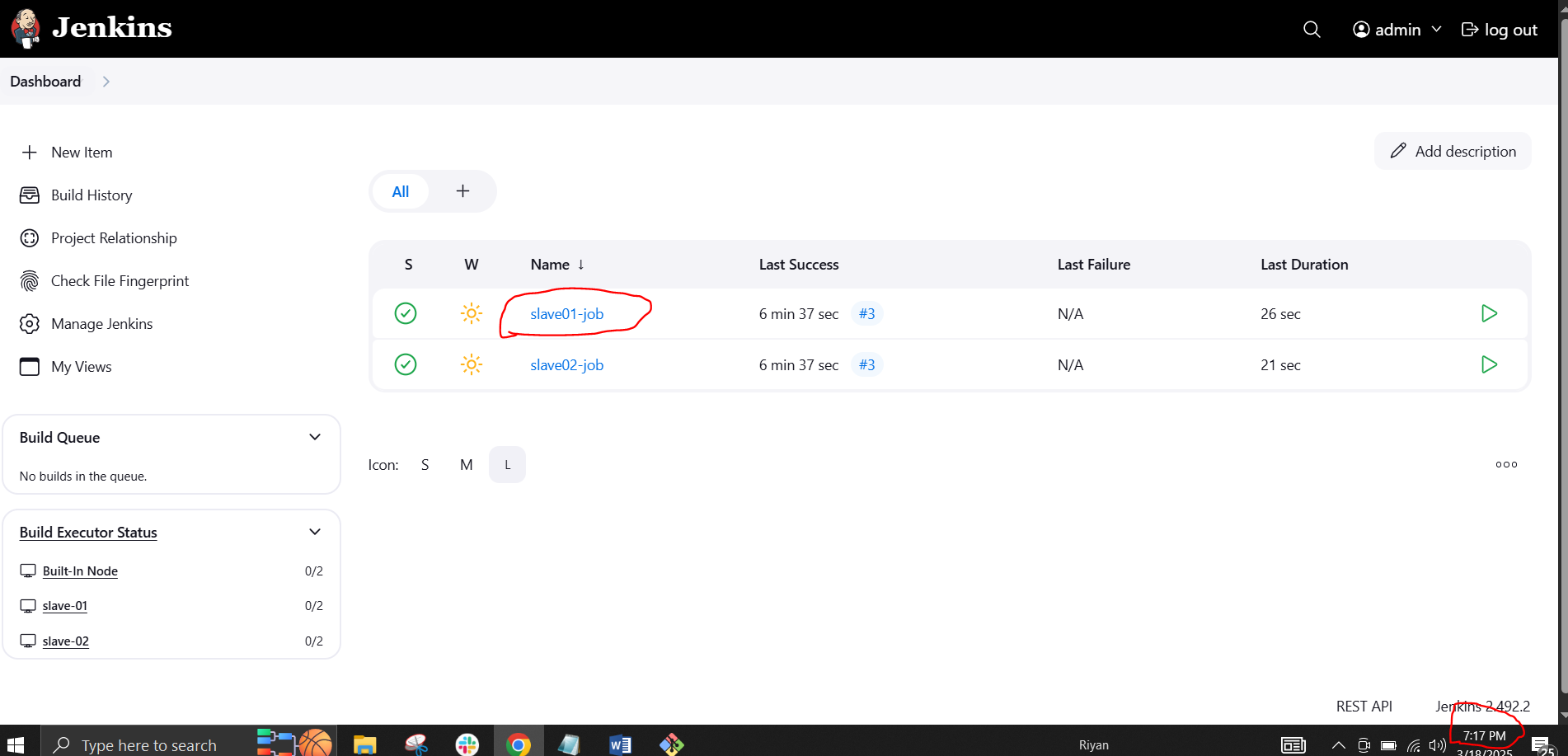
Click on the job, scroll below and check for Triggers

Under build periodically

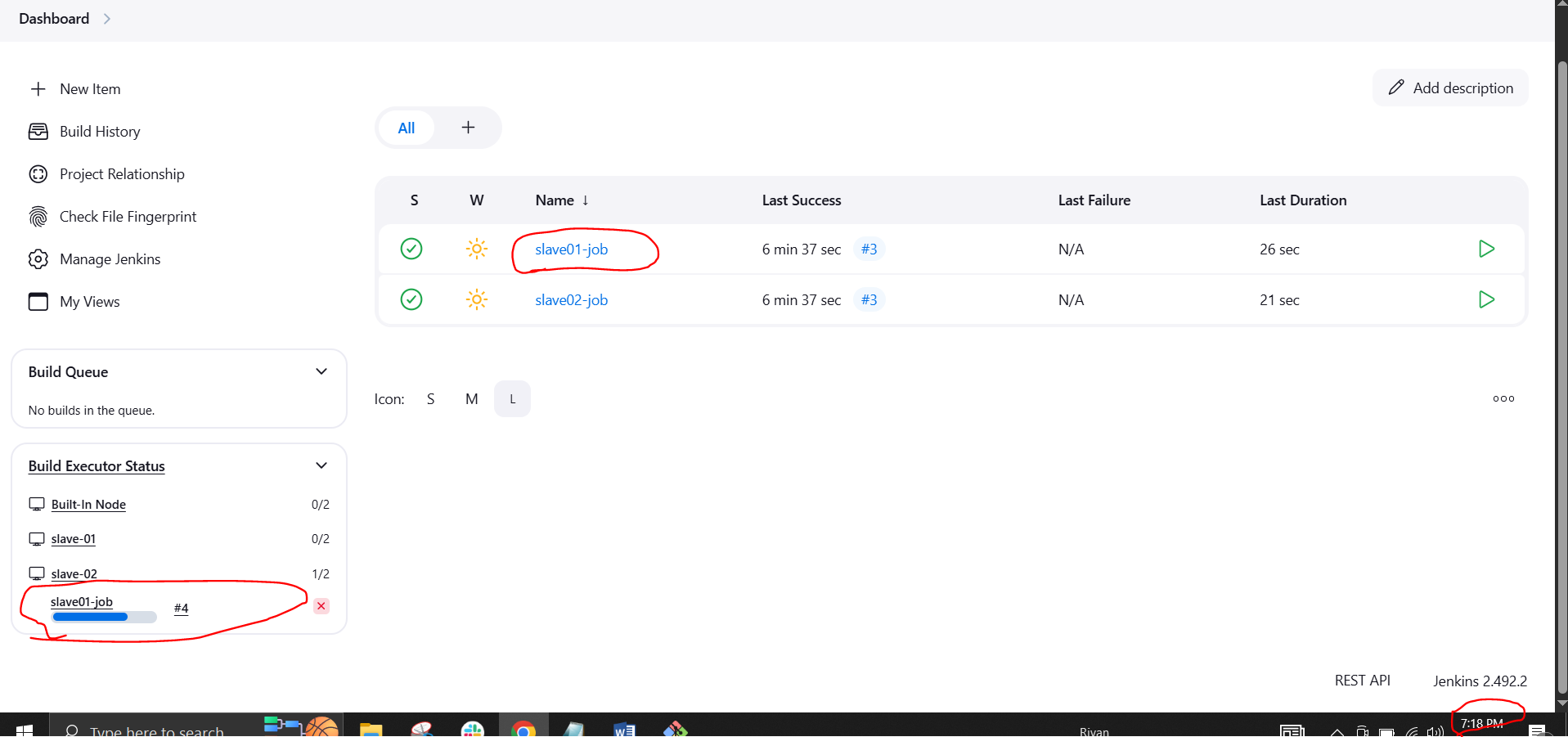
Add \* \* \* \* \*



Now wait for 1 minute, it will automatically trigger the job



After 1 minute it will trigger automatically

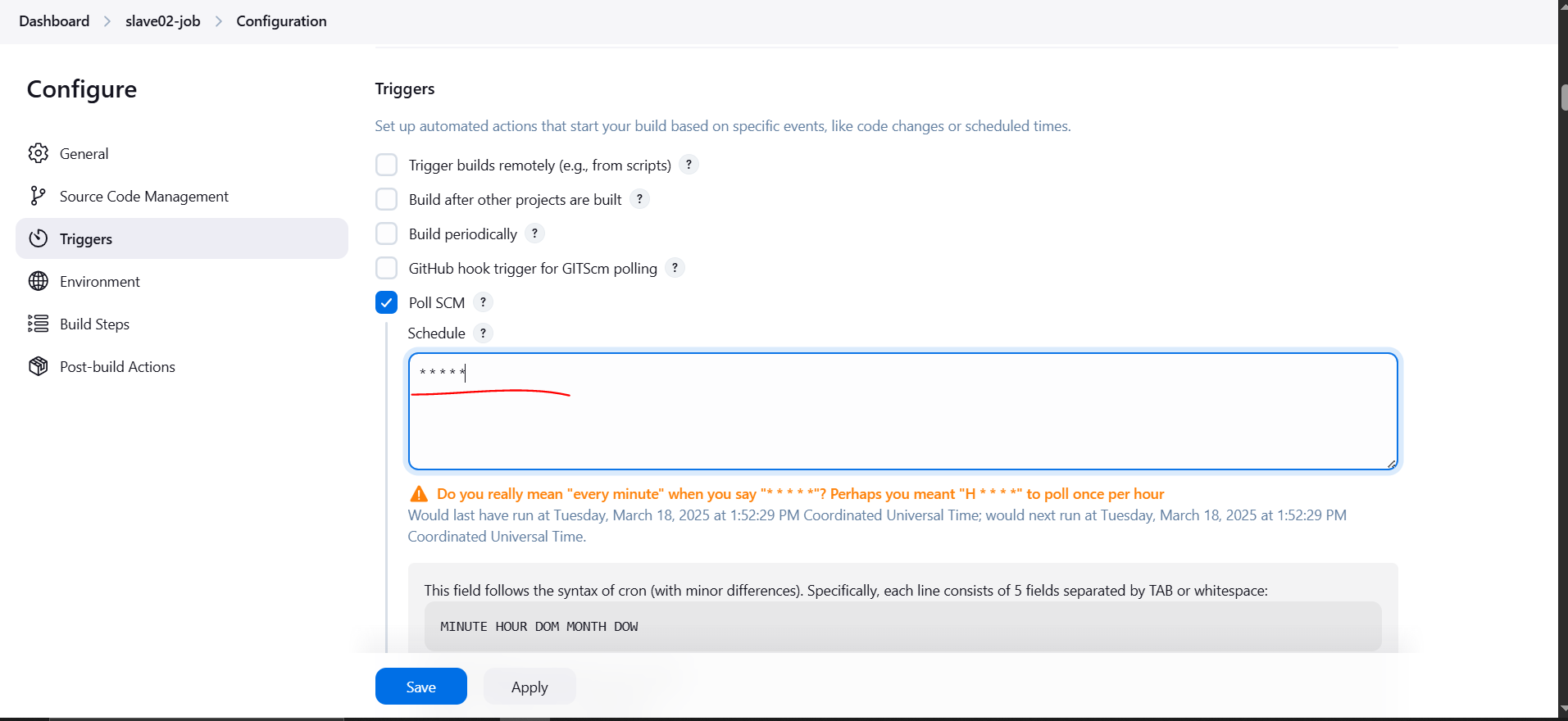


To configure poll SCM

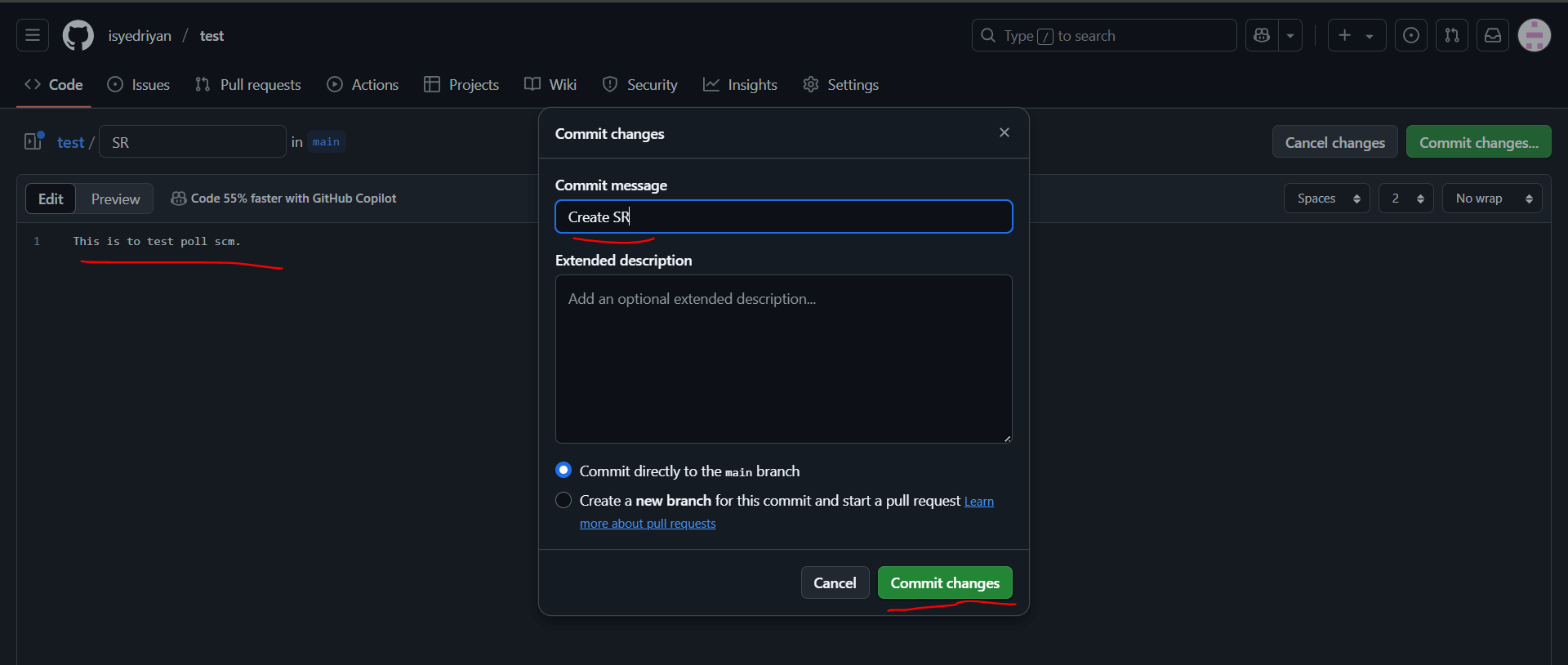
Click on the job, scroll below and check for Triggers

Under poll SCM

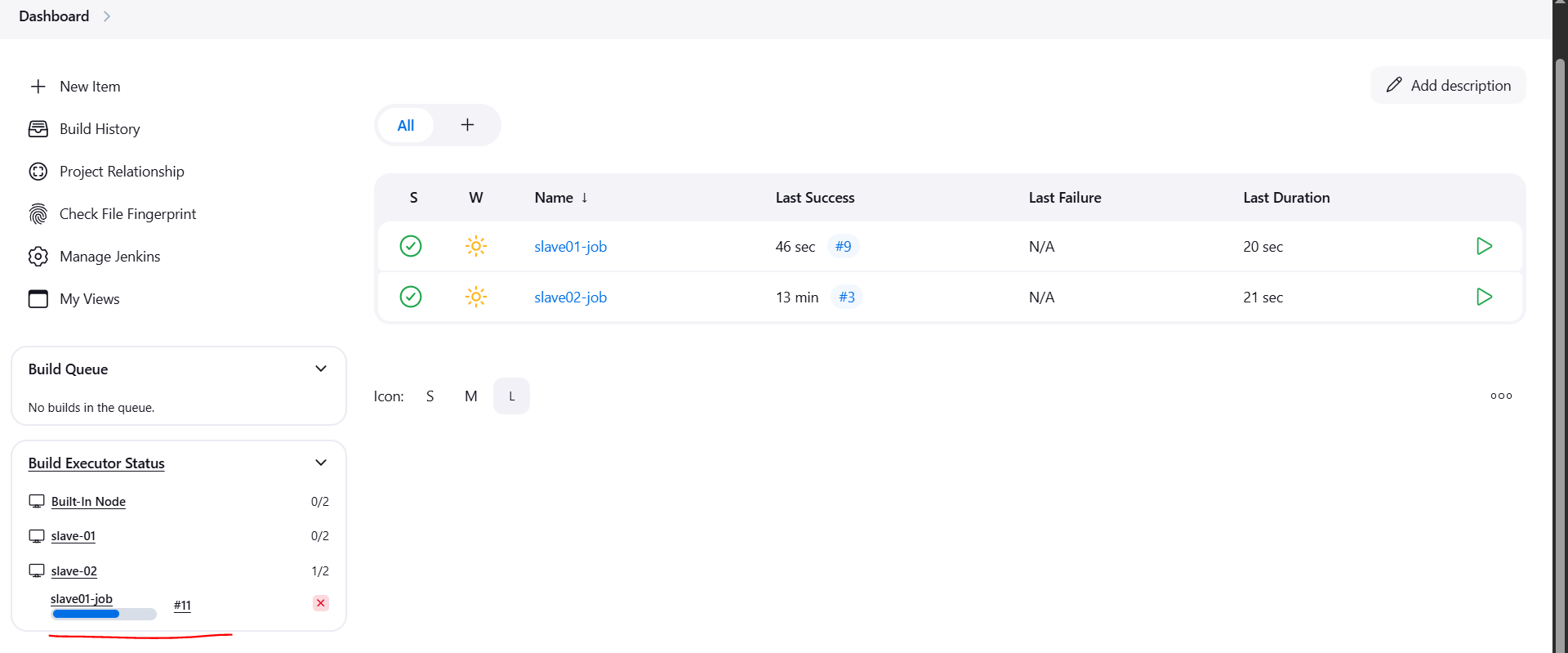
Add \* \* \* \* \*



Now go to github and do some changes it will reflect in the Jenkins server



It has triggered in Jenkins server



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

First connect to Jenkins master ec2 instance

**sudo –i**

Create the script file using the command

**sudo vi /opt/jenkins\_backup.sh**

Paste the script in the file (Jenkins\_backup.sh)

#!/bin/bash

# Backup directory

BACKUP\_DIR="/backup/jenkins"

TIMESTAMP=$(date +"%Y-%m-%d\_%H-%M-%S")

BACKUP\_FILE="jenkins\_backup\_$TIMESTAMP.tar.gz"

# Jenkins home directory

JENKINS\_HOME="/var/lib/jenkins"

# Days to keep old backups

RETENTION\_DAYS=7

echo "Starting Jenkins backup..."

# Stop Jenkins service (optional but safer)

sudo systemctl stop jenkins

# Ensure backup directory exists

mkdir -p "$BACKUP\_DIR"

# Create a compressed backup

tar -czvf "$BACKUP\_DIR/$BACKUP\_FILE" "$JENKINS\_HOME"

# Restart Jenkins service

sudo systemctl start jenkins

echo "Backup completed: $BACKUP\_DIR/$BACKUP\_FILE"

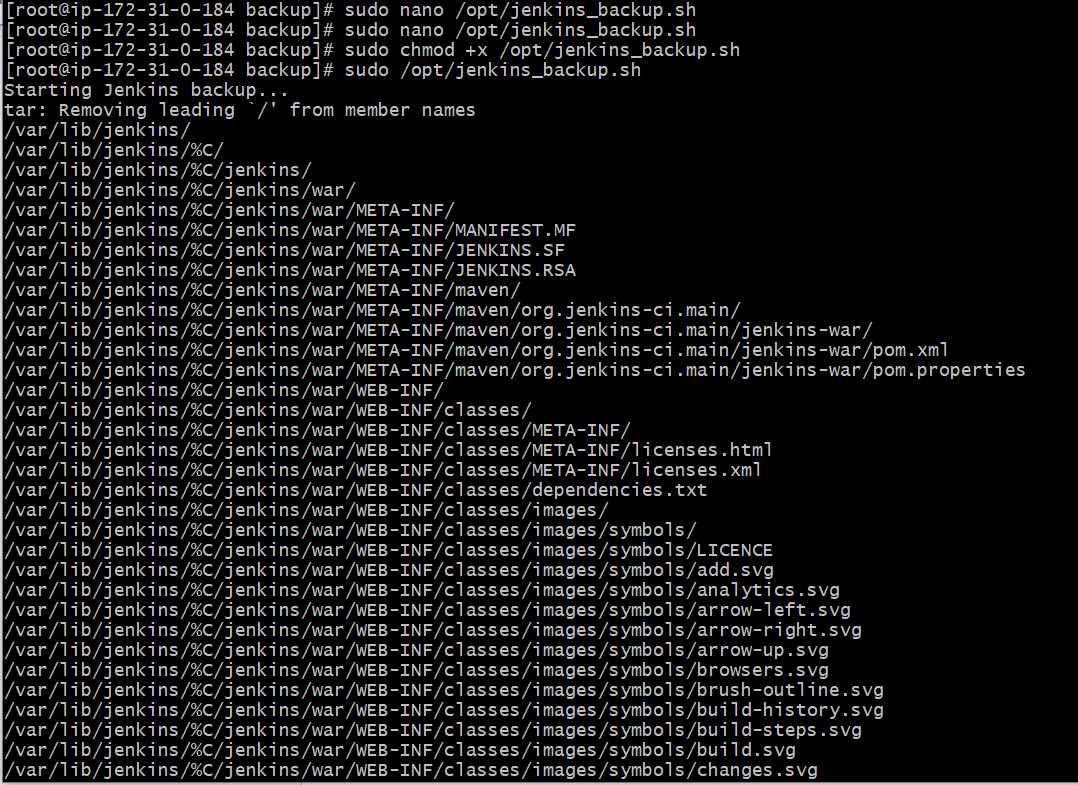
# Delete old backups

find "$BACKUP\_DIR" -name "jenkins\_backup\_\*.tar.gz" -mtime +$RETENTION\_DAYS -exec rm -f {} \;

echo "Old backups older than $RETENTION\_DAYS days deleted."

**sudo chmod +x /opt/jenkins\_backup.sh**

**sudo /opt/jenkins\_backup.sh**



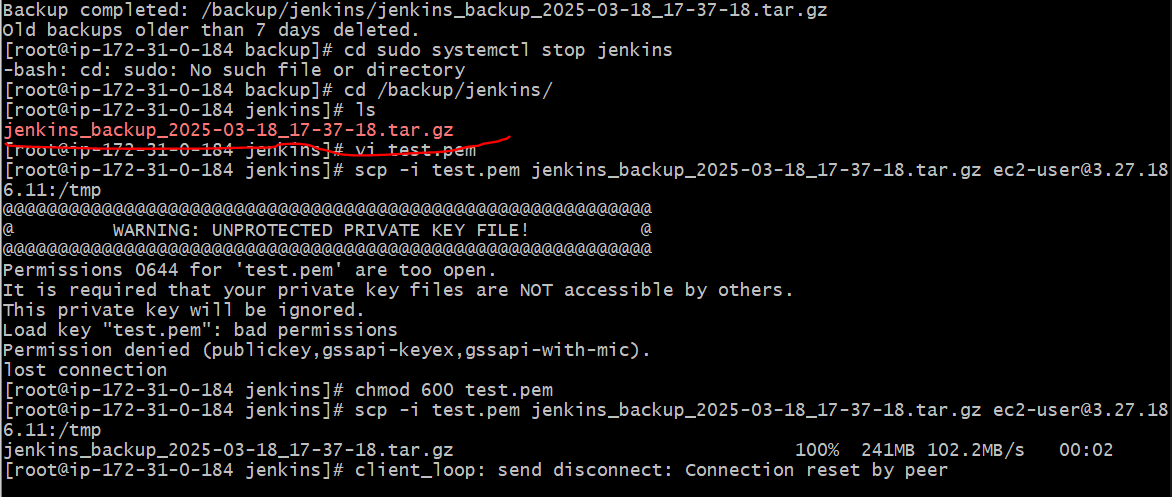
Copy the pem key in our master-jenkins ec2 instance

**Vi test.pem**

**chmod 600 test.pem**

Now copy the .tar.gz to the backup-jenkins ec2 instance using the command:

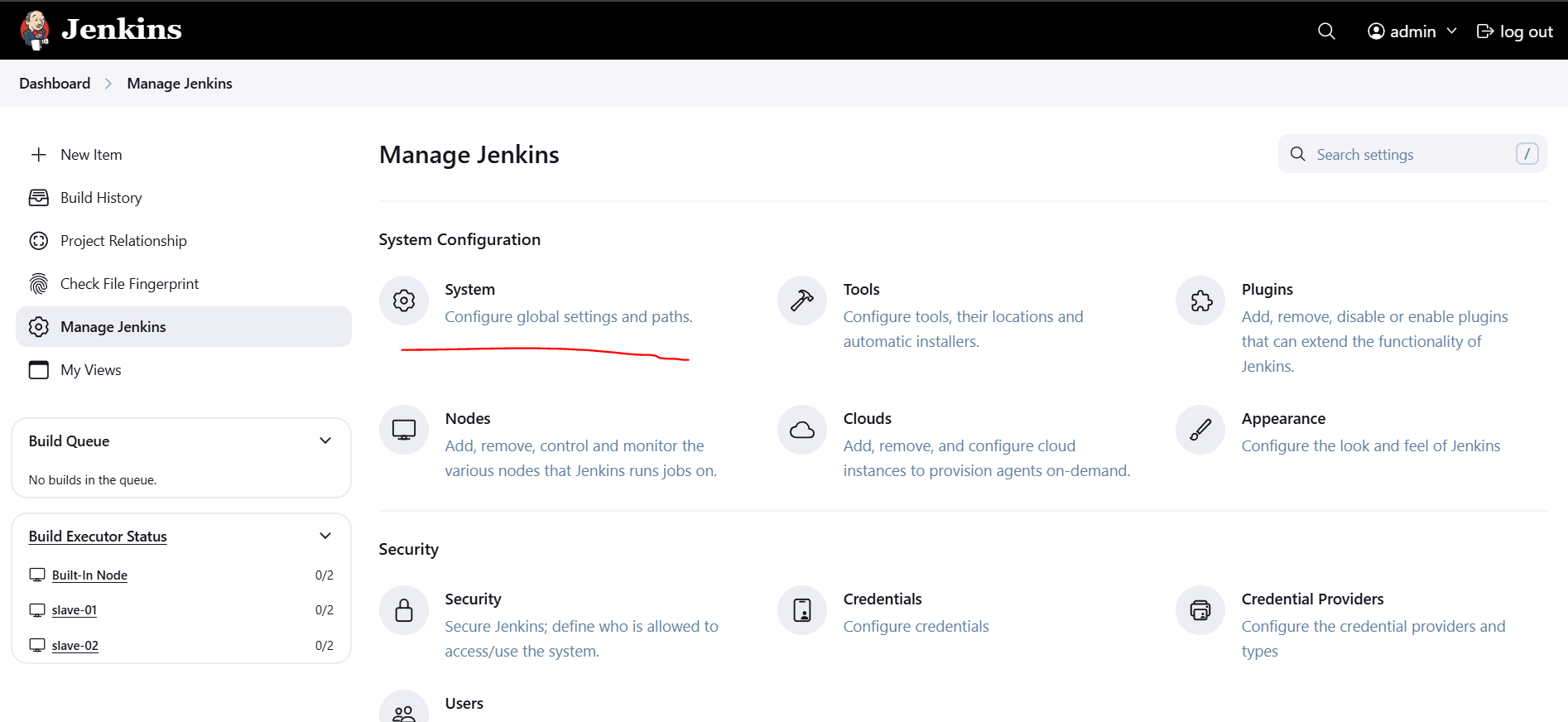
**scp -i test.pem jenkins\_backup\_2025-03-18\_17-37-18.tar.gz** [**ec2-user@3.27.186.11:/tmp**](mailto:ec2-user@3.27.186.11:/tmp)



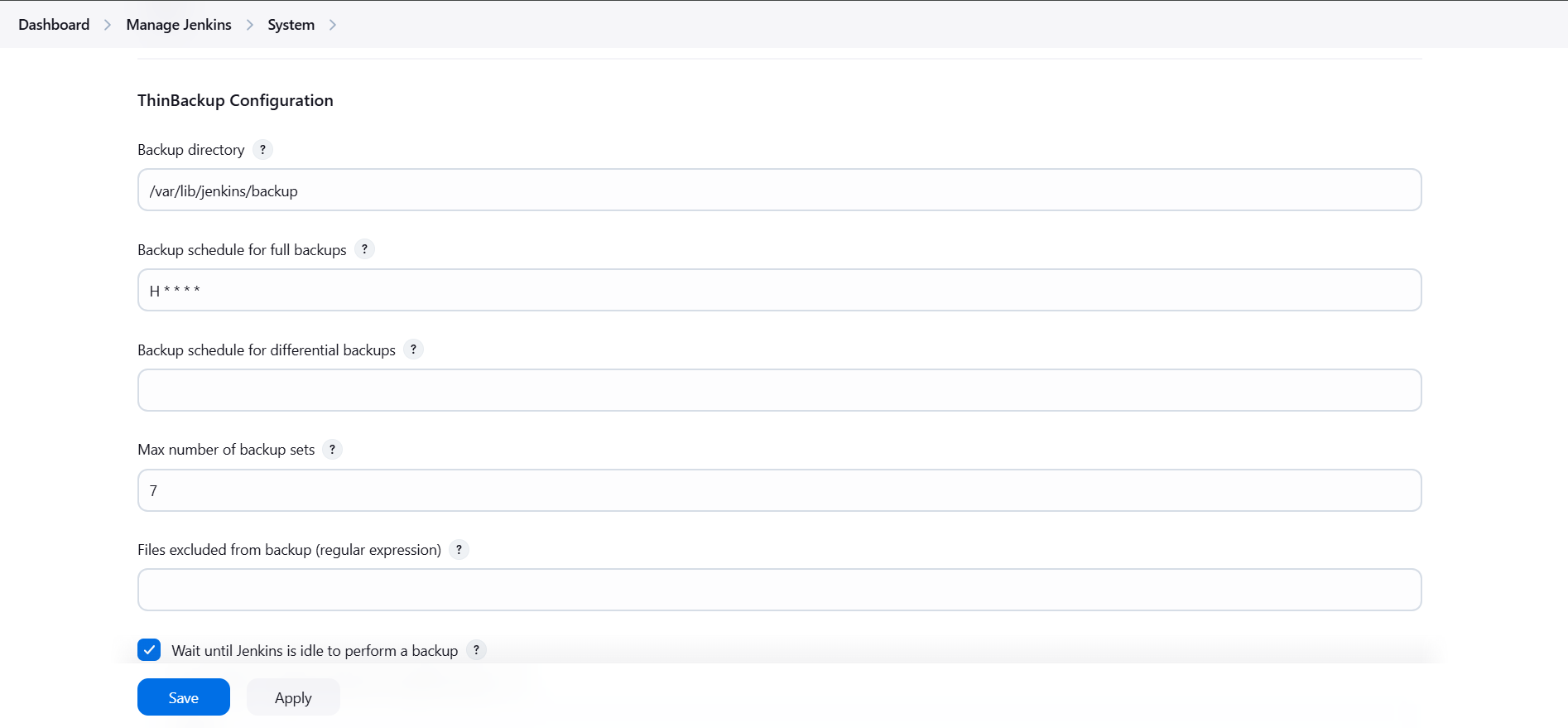
We have our Jenkins server backup in **jenkins\_backup\_2025-03-18\_17-37-18.tar.gz**

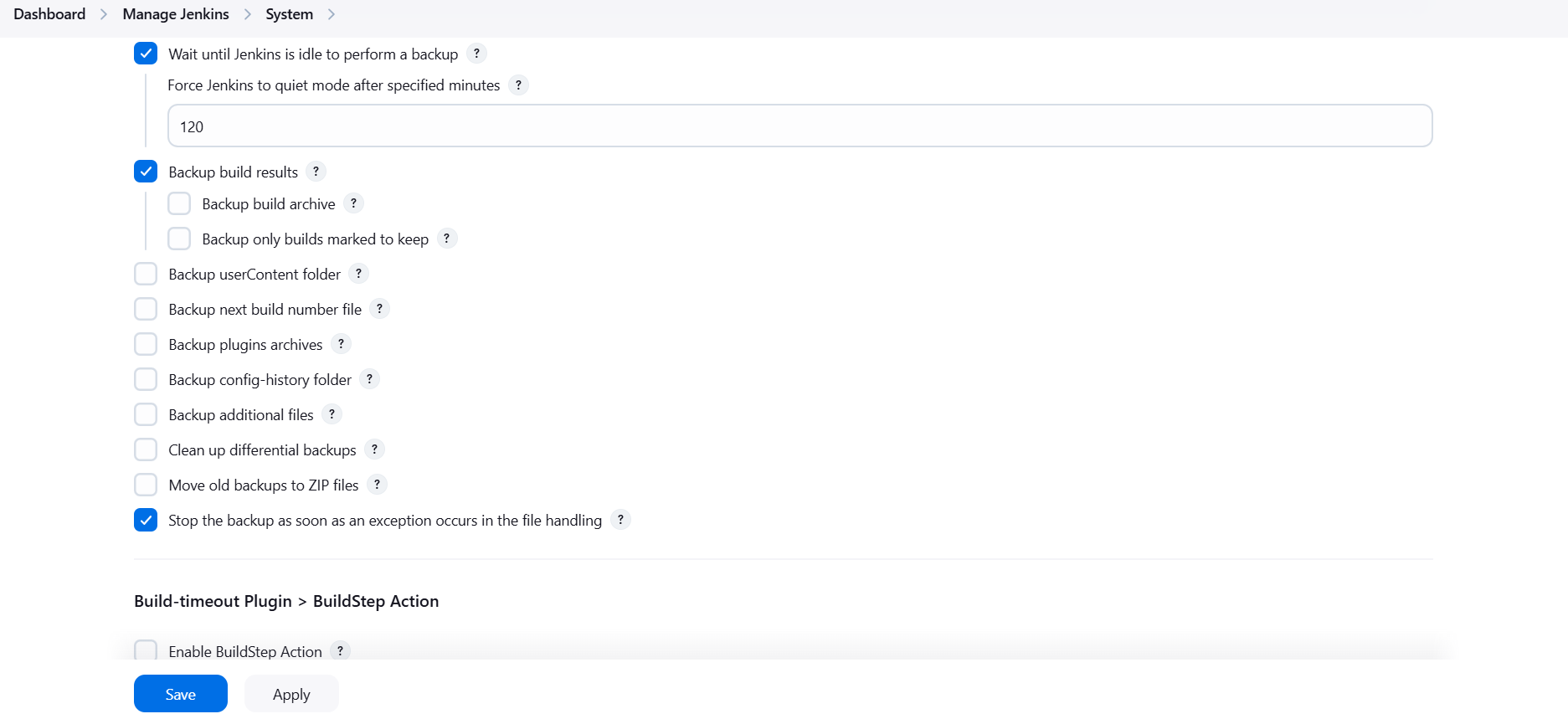
**5) Take backup of jenkins using rethin backup plugin.**

First go to Manage Jenkins🡪Settings🡪



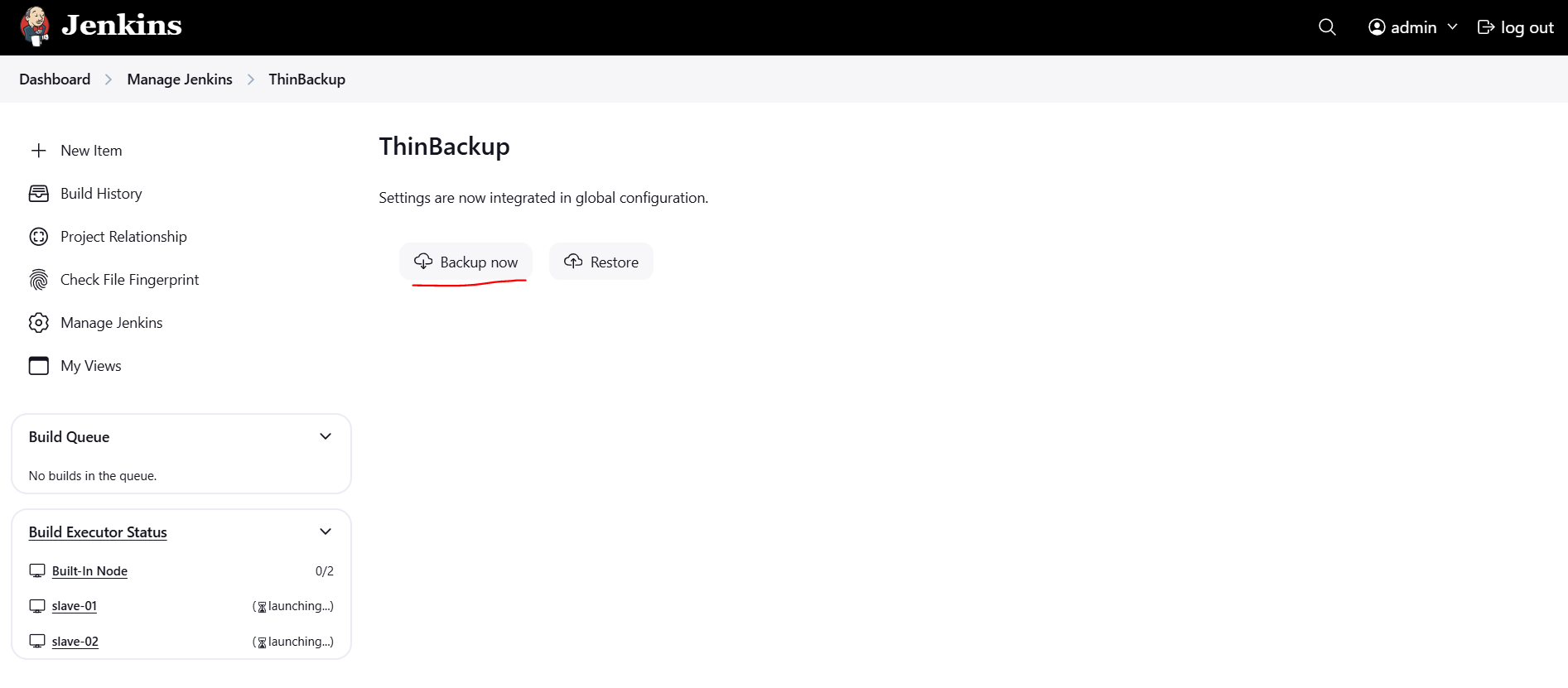
Provide the details based on your requirements





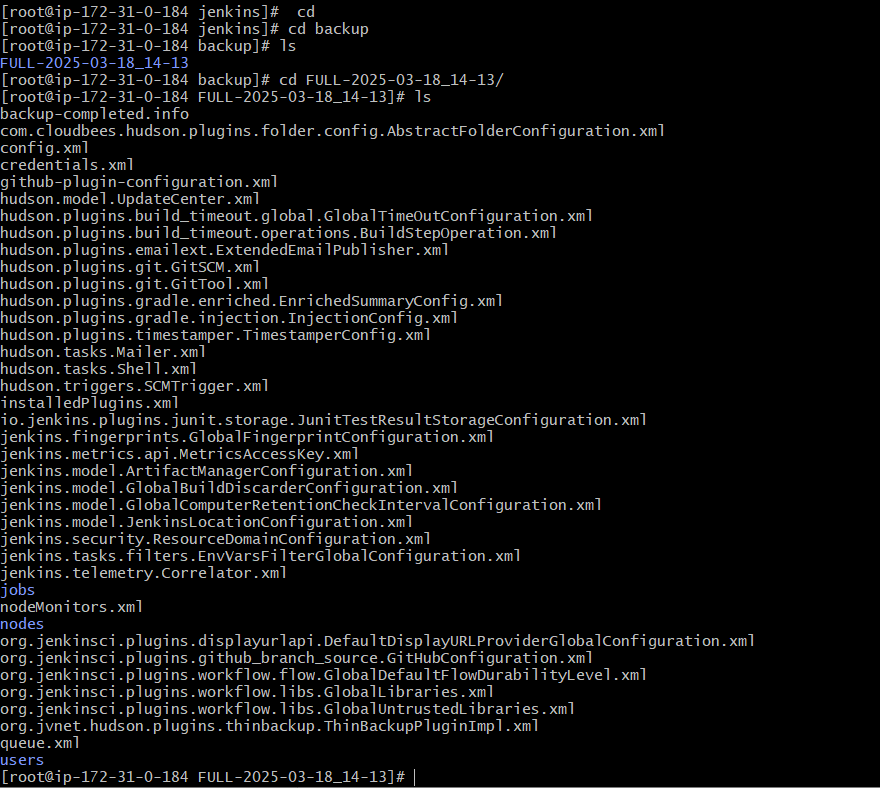
It will take backup for every 1 hour in /var/lib/Jenkins/backup

To take backup manually, go to ThinBackup and click on Backup now



It will take backup now in /var/lib/Jenkins/backup

Go to Jenkins instance and check now

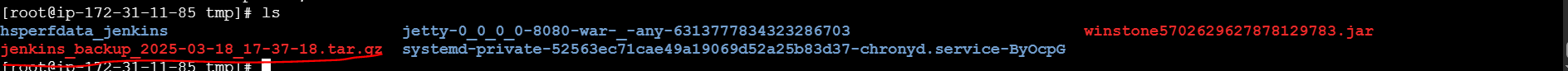


**6) Setup a new jenkins server and dump the backup taken in task4.**

First create a new backup-jenkins ec2 instance

Install java and Jenkins in it

Now move /tmp folder where we have the backup from the master-jenkins server



Copy this .tar.gz file to Jenkins default location /var/lib/Jenkins

**cp jenkins\_backup\_2025-03-18\_17-37-18.tar.gz /var/lib/Jenkins**

Now extract the file using the command:

**tar -xvzf jenkins\_backup\_2025-03-18\_16-33-01.tar.gz**

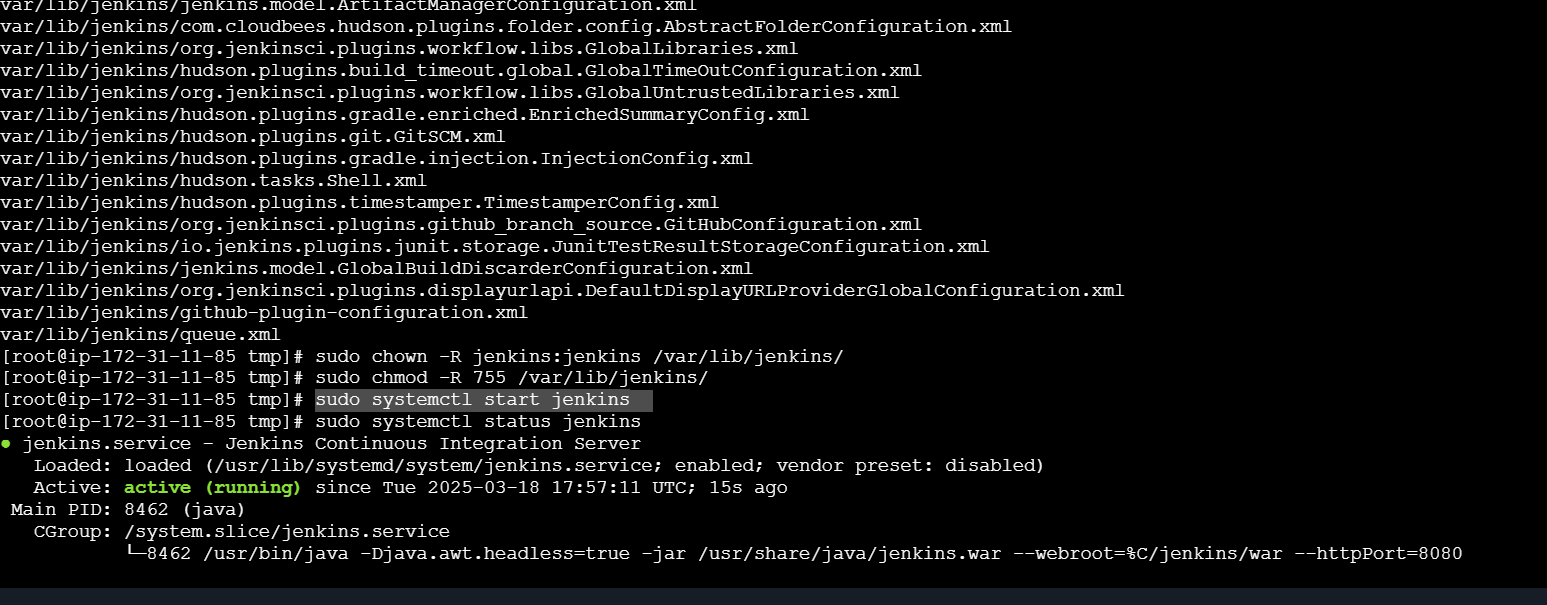
Change permissions and ownership for /var/lib/jenkins/

**sudo chown -R jenkins:jenkins /var/lib/jenkins/**

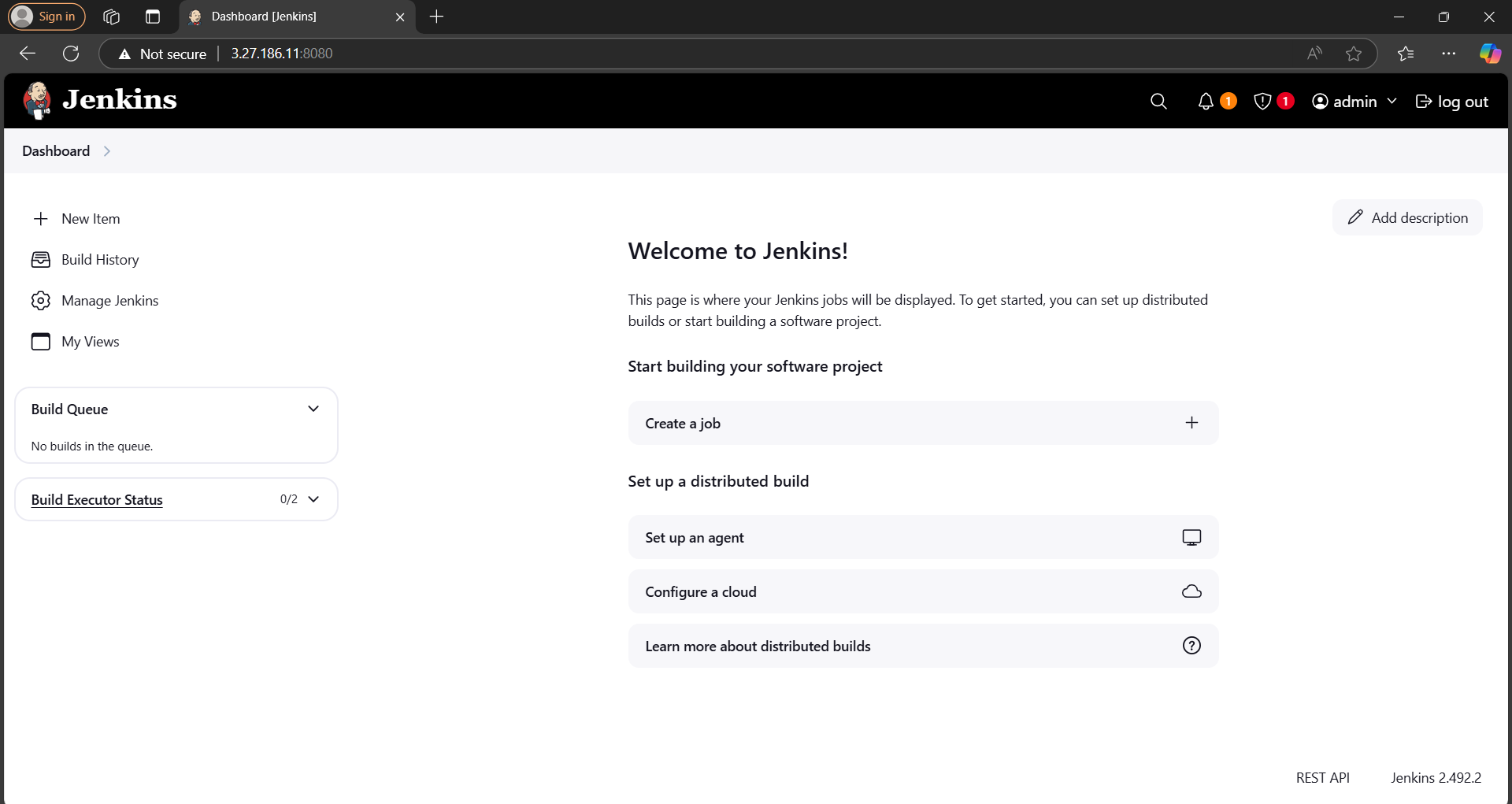
**sudo chmod -R 755 /var/lib/jenkins/**

Now start the Jenkins using the command:

**sudo systemctl start jenkins**

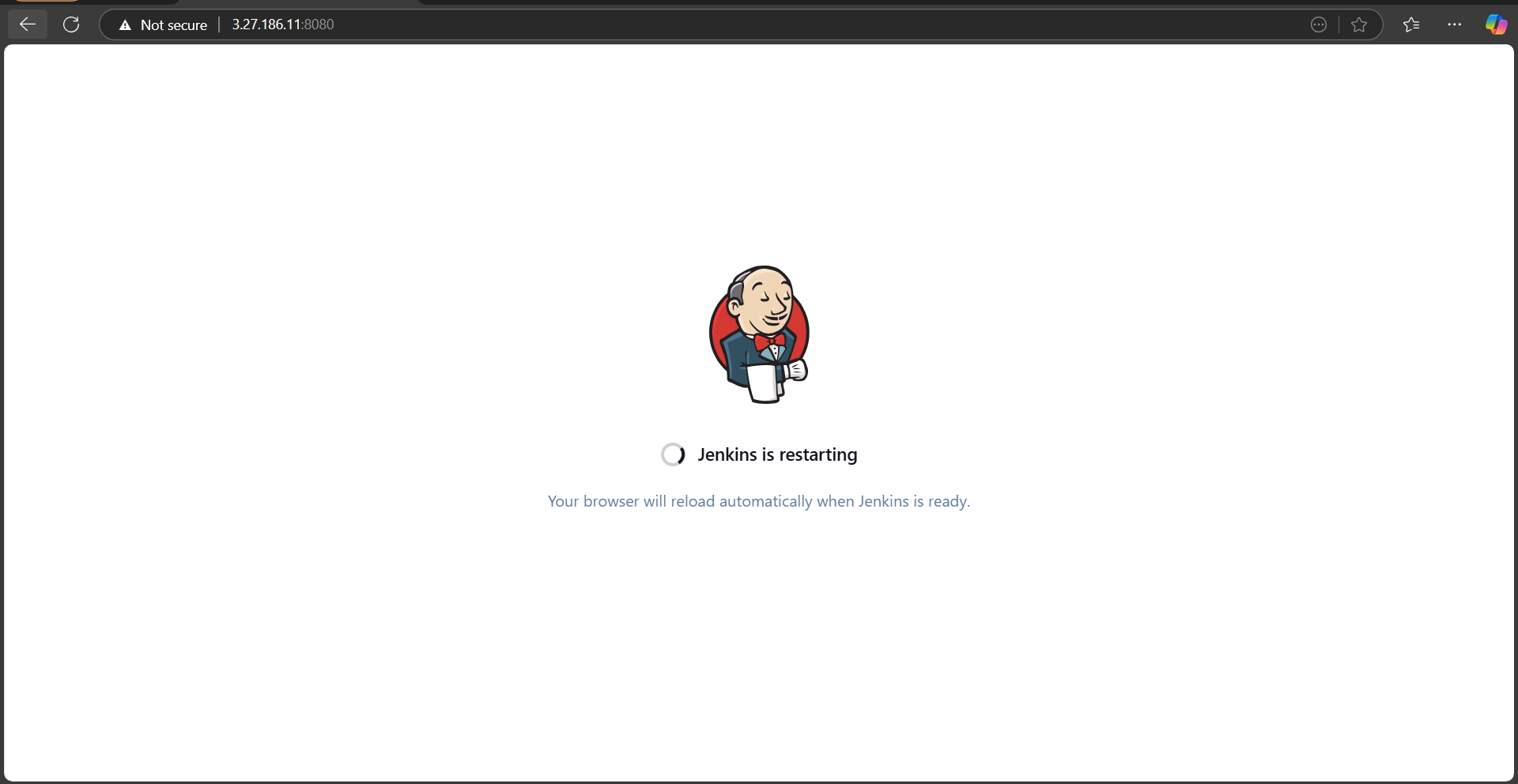


Now open the backup-jenkins server in the browser

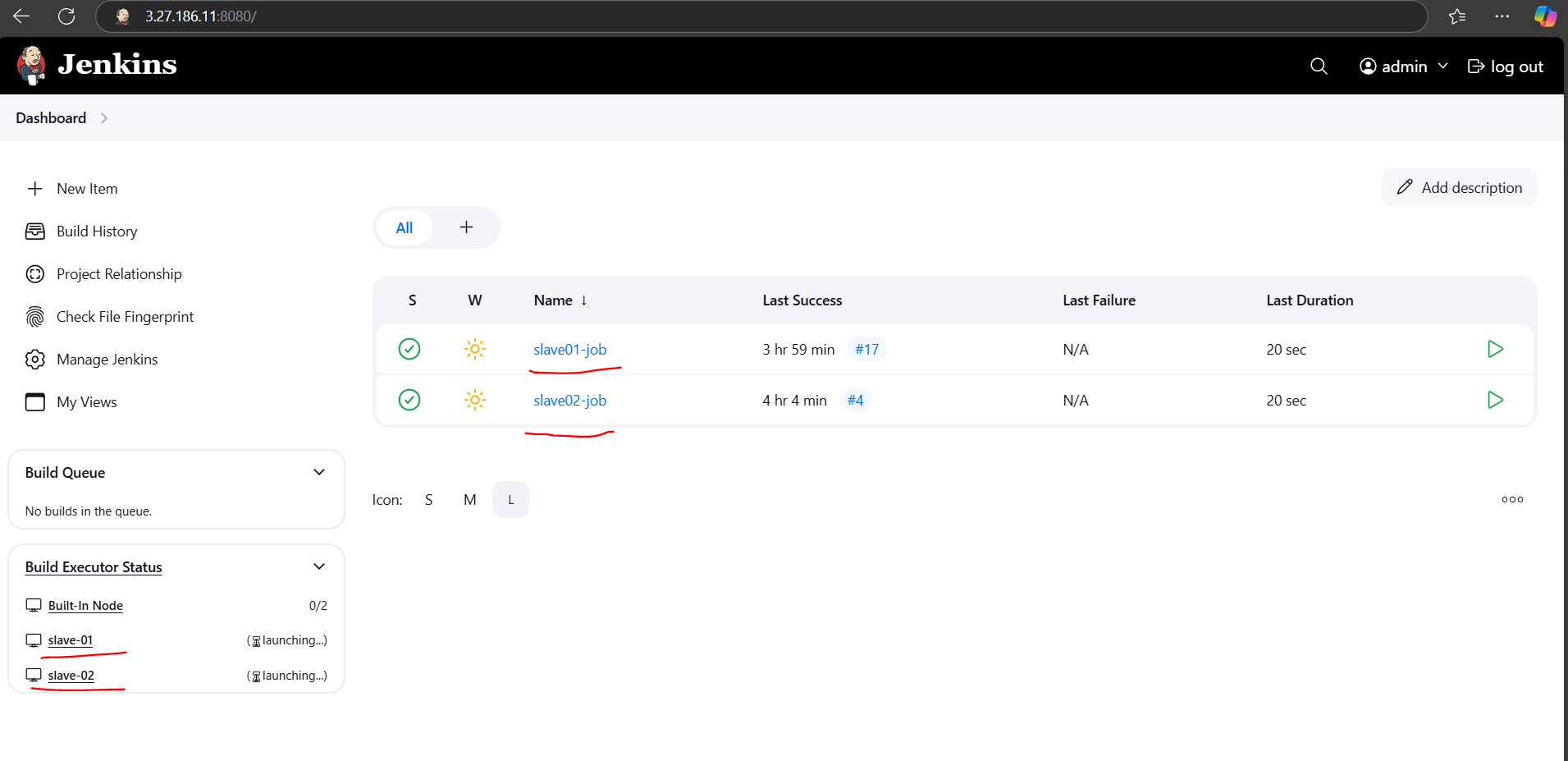


Restart the server with /restart

**http://3.27.186.11:8080/restart**



After restarting our master-jenkins jobs will be seen in the backup-jenkins server



This is how we have successfully dumped the backup to a new Jenkins server