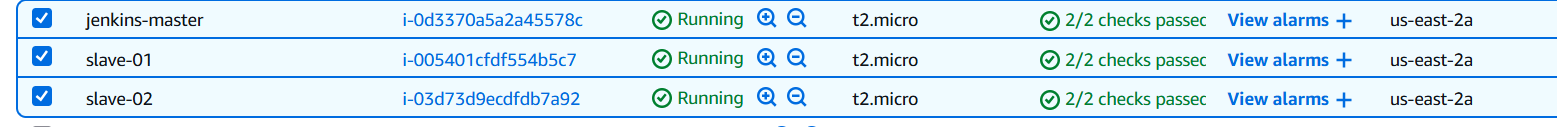
Jenkins task for Day-02.

===========================

1. Configure 2 slave machines in jenkins master.

First creat 3 ec2 instance



* Master in master install >> java prequesite and Jenkins and git

**Install sudo yum update –y**

**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**

**sudo yum upgrade**

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

**sudo yum install jenkins -y**

**sudo systemctl start jenkins**

**sudo systemctl enable jenkins**

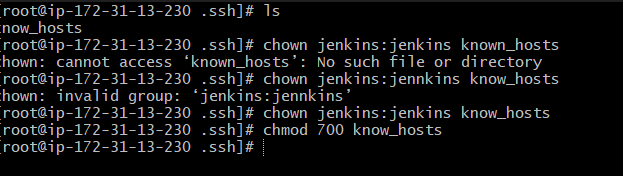
**sudo systemctl status jenkinsJava (Amazon Linux 2023):**

after install in Jenkins in master

install git command >> **sudo yum install git –y**

**step: 2**

**after instllaing java in slave 01 and git ,I created ssh-keygen and**

****

**Slave 01**

Now to slave 01 ec2

Install java **Install Java (Amazon Linux 2023):  
[ec2-user ~]$** yum install -y java-17-amazon-corretto

Now do

**NOTE:DO SSH-KEYGEN , in the root command**

[root@ip-172-31-9-222 ~]# cd .ssh

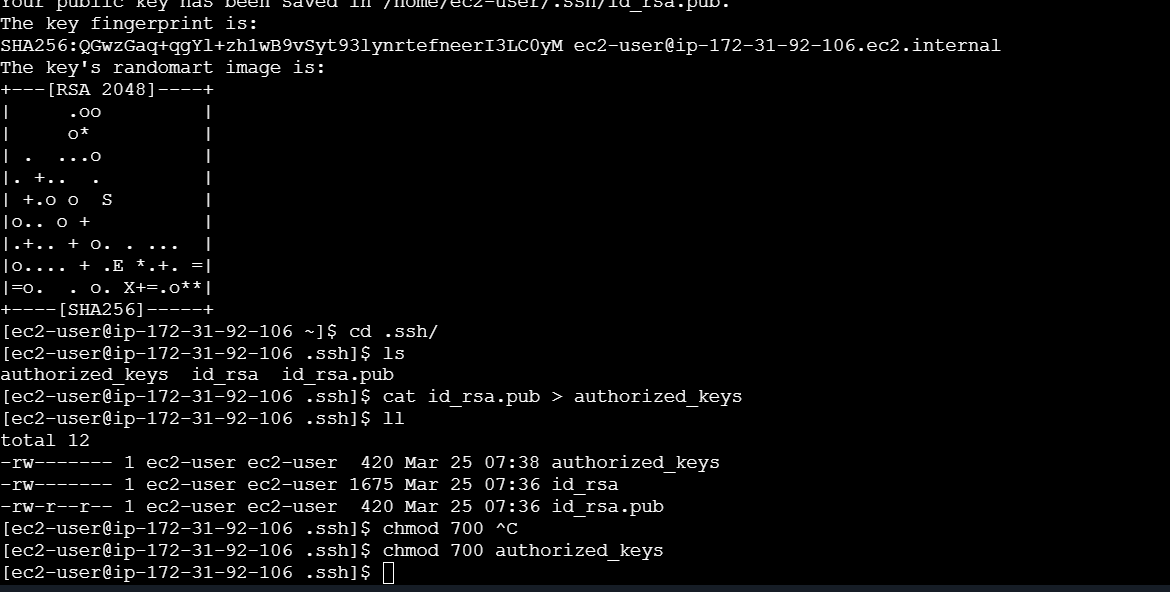
[root@ip-172-31-9-222 .ssh]# ls

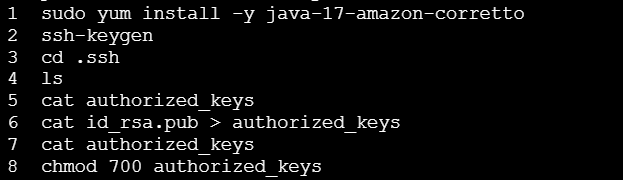
authorized\_keys id\_rsa id\_rsa.pub

[root@ip-172-31-9-222 .ssh]# cat id\_rsa.pub > authorized\_keys

[root@ip-172-31-9-222 .ssh]# chmod 700 authorized\_keys

[root@ip-172-31-9-222 .ssh]# ll



**chmod 700 authorized\_keys**

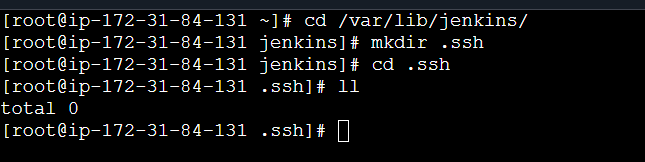
now succefully created a keygen and added permission in slave-01

now go to jenkins master,

create one directiory mkdir .ssh in /var/lib/jenkins/

now go to in to directoy .ssh

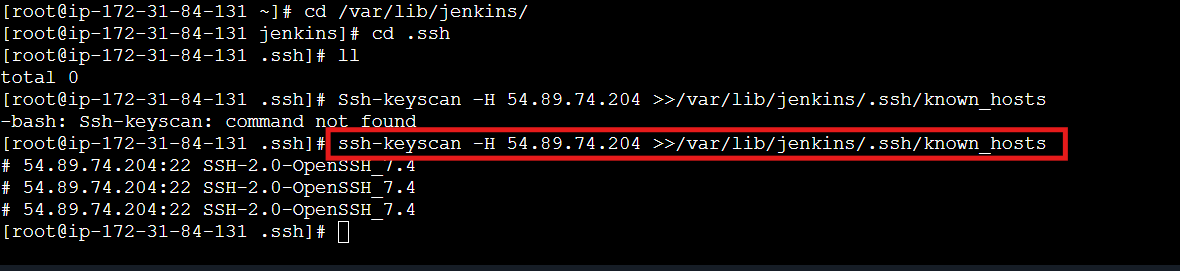
cd .ssh



Use this command to in Jenkins master now .

Note: do this below command as root user use (sudo su -)

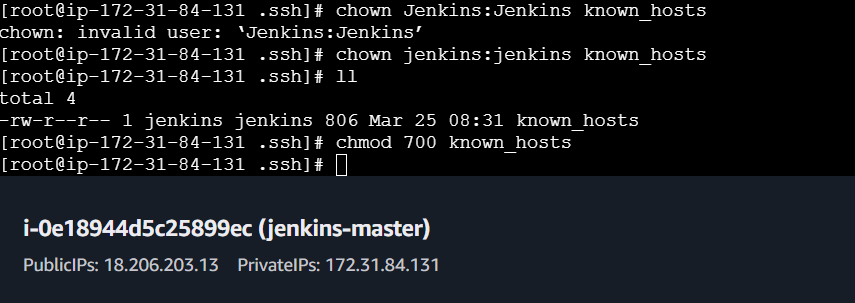
**ssh-keyscan –H 54.89.74.204 >>/var/lib/jenkins/.ssh/known\_hosts** (N0te : in this command use ip address of slave macheine



Then change permissions by using command

[root@ip-172-31-84-131 .ssh]# chown jenkins:jenkins known\_hosts

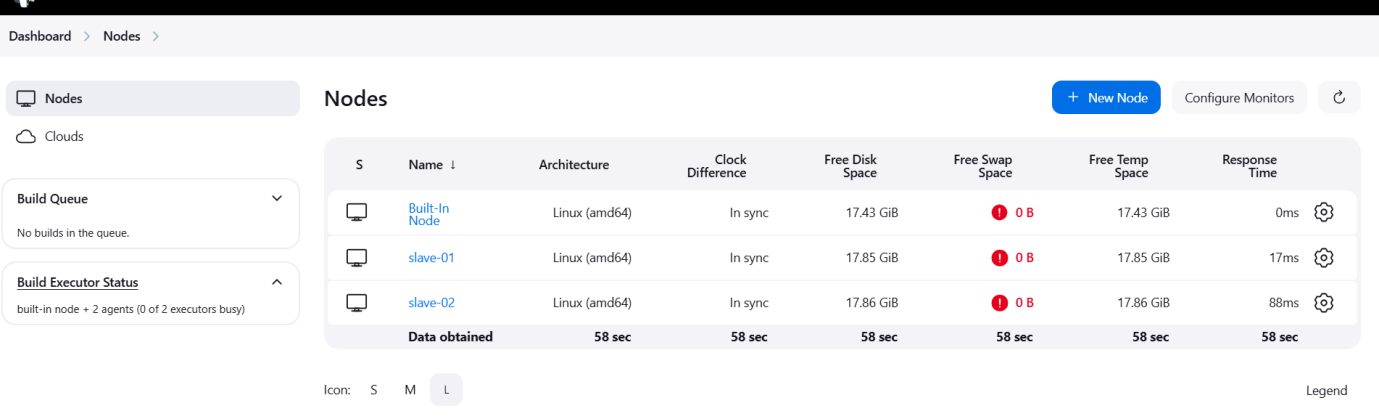
[root@ip-172-31-84-131 .ssh]# chmod 700 known\_hosts

****

Slave 02

Import step for slave 02

[root@ip-172-31-44-116 .ssh]# ssh-keygen -t rsa -b 4096 -f /root/.ssh/id\_rsa\_slave02

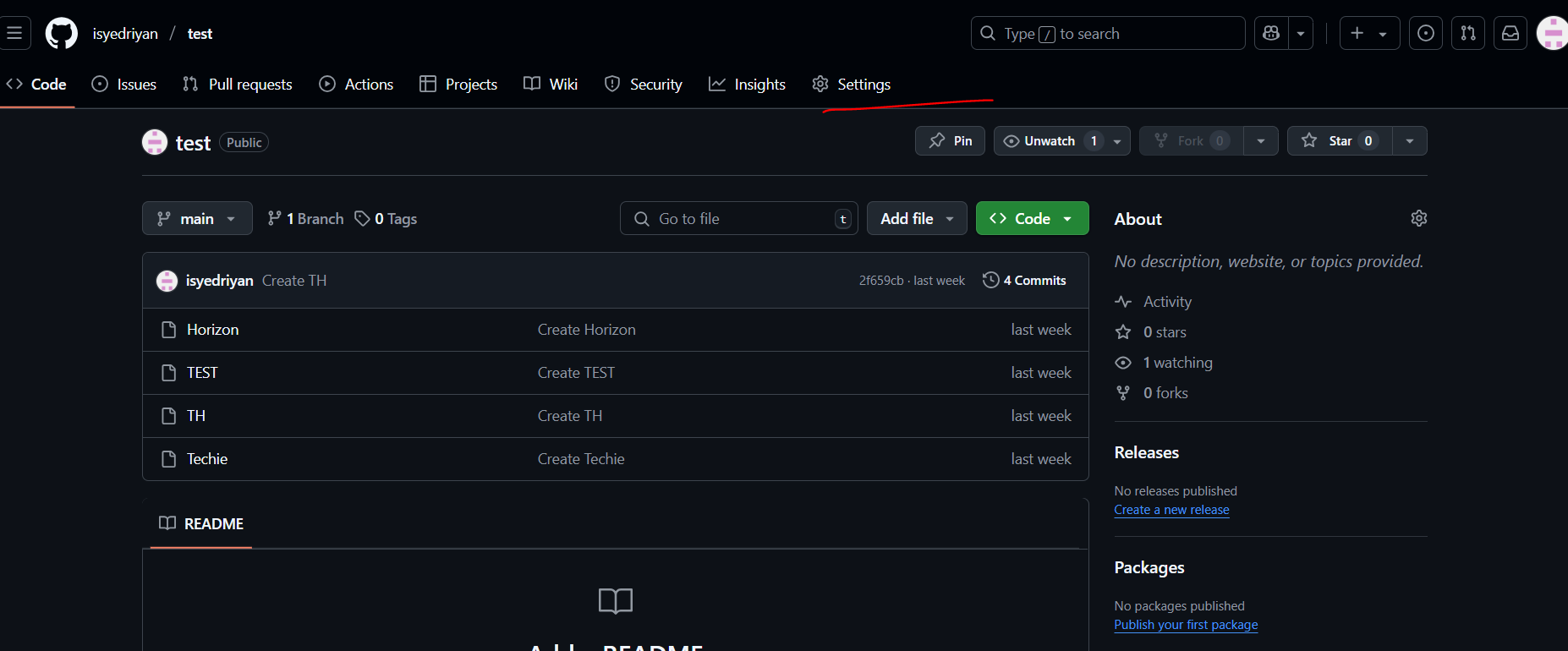


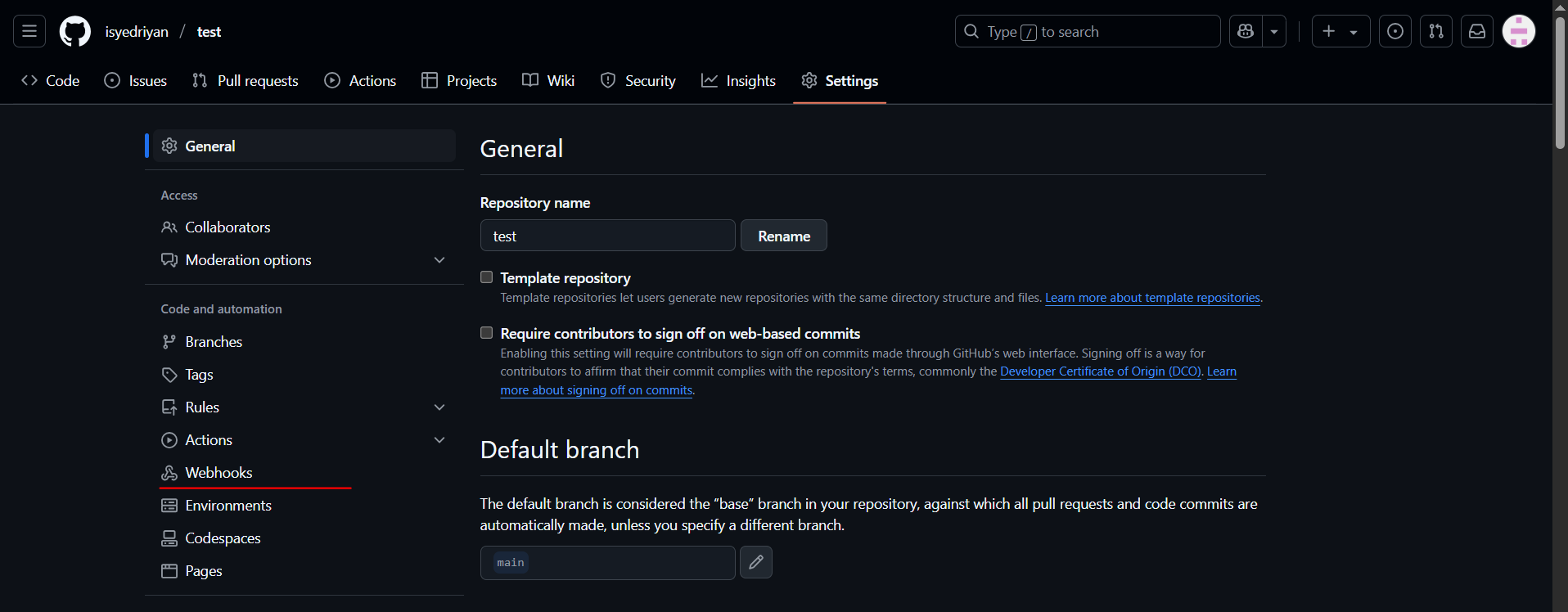
1. Configure webhooks to jenkins job.

In this whenever changes in sources , ones we configured , webhook ,it will automatically trigred

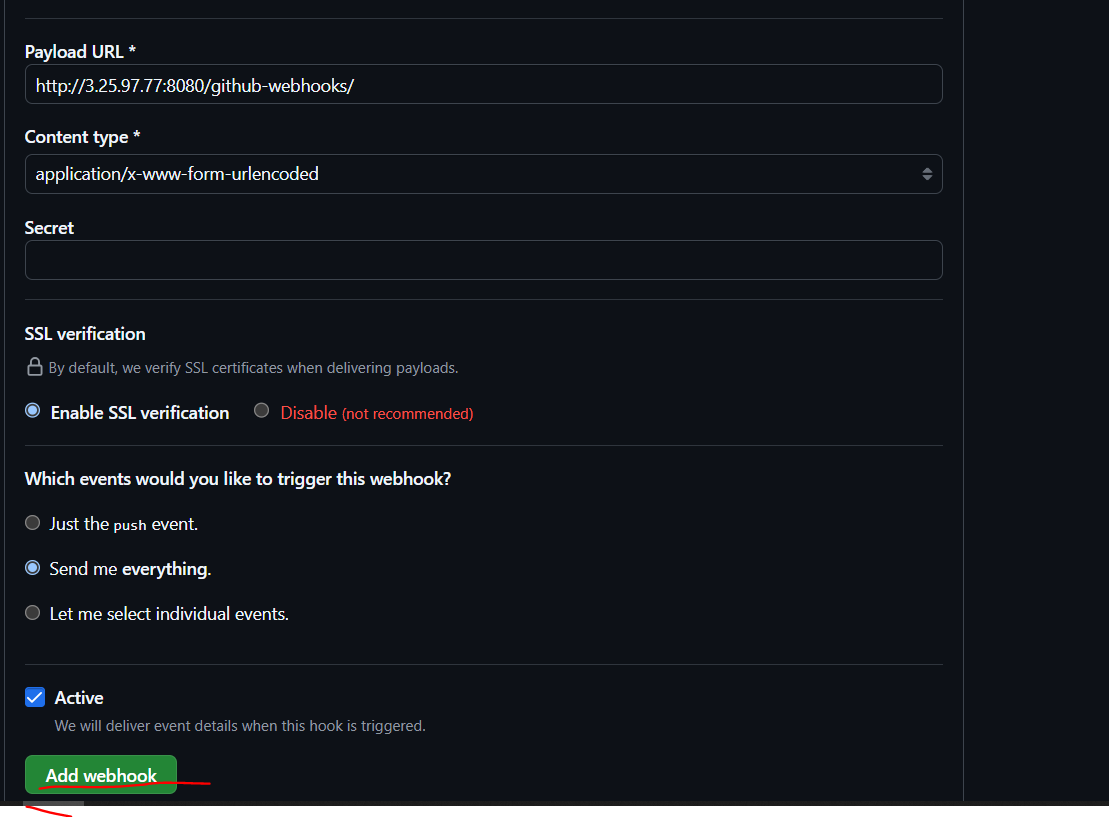
Create a repo in github and add webhooks for that repo

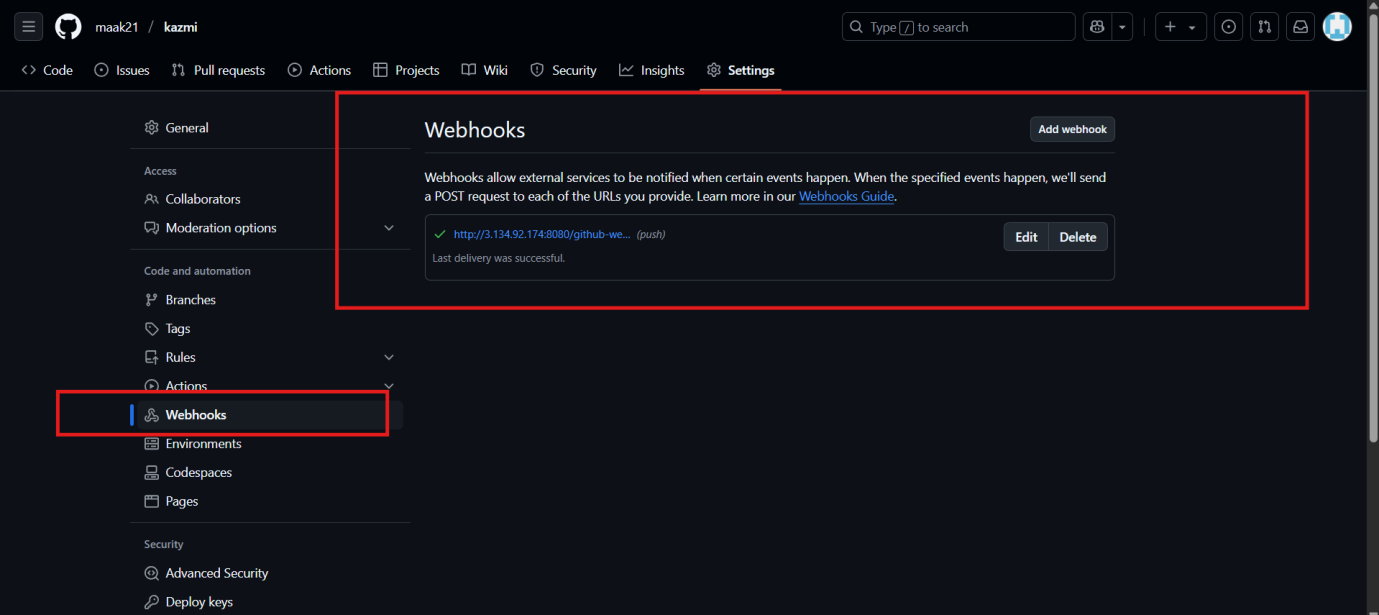
Go to settings

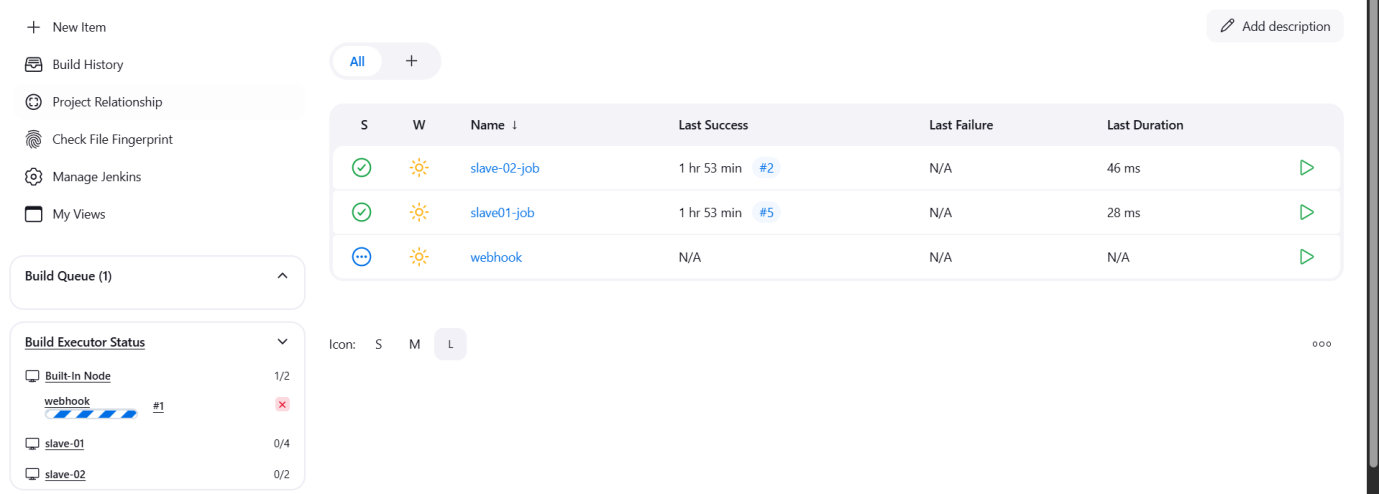


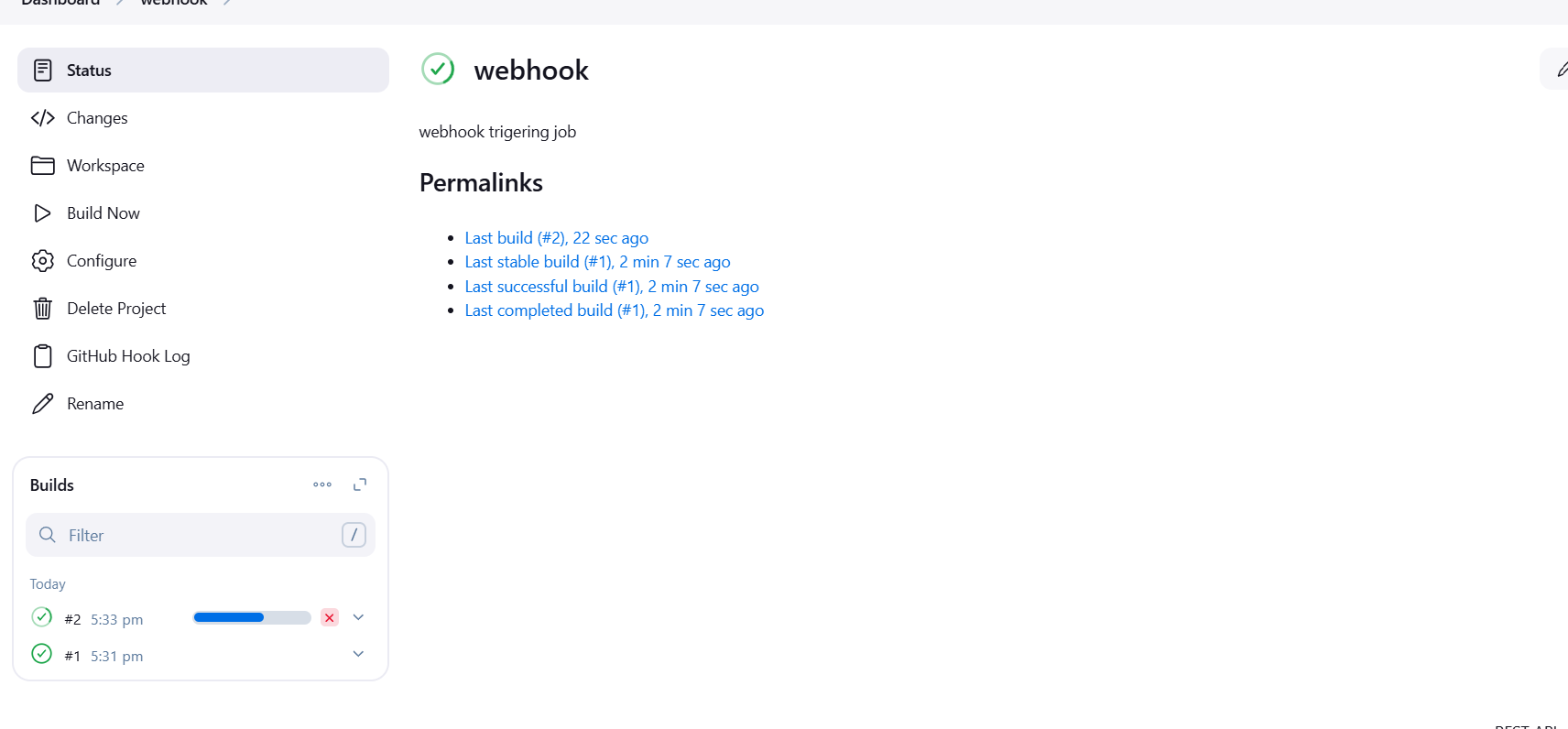


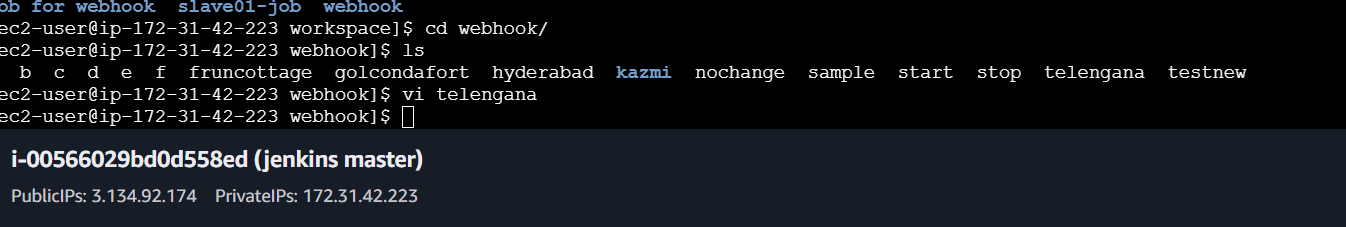
Now go to Jenkins GUI copy the url till 8080 and add in payload url













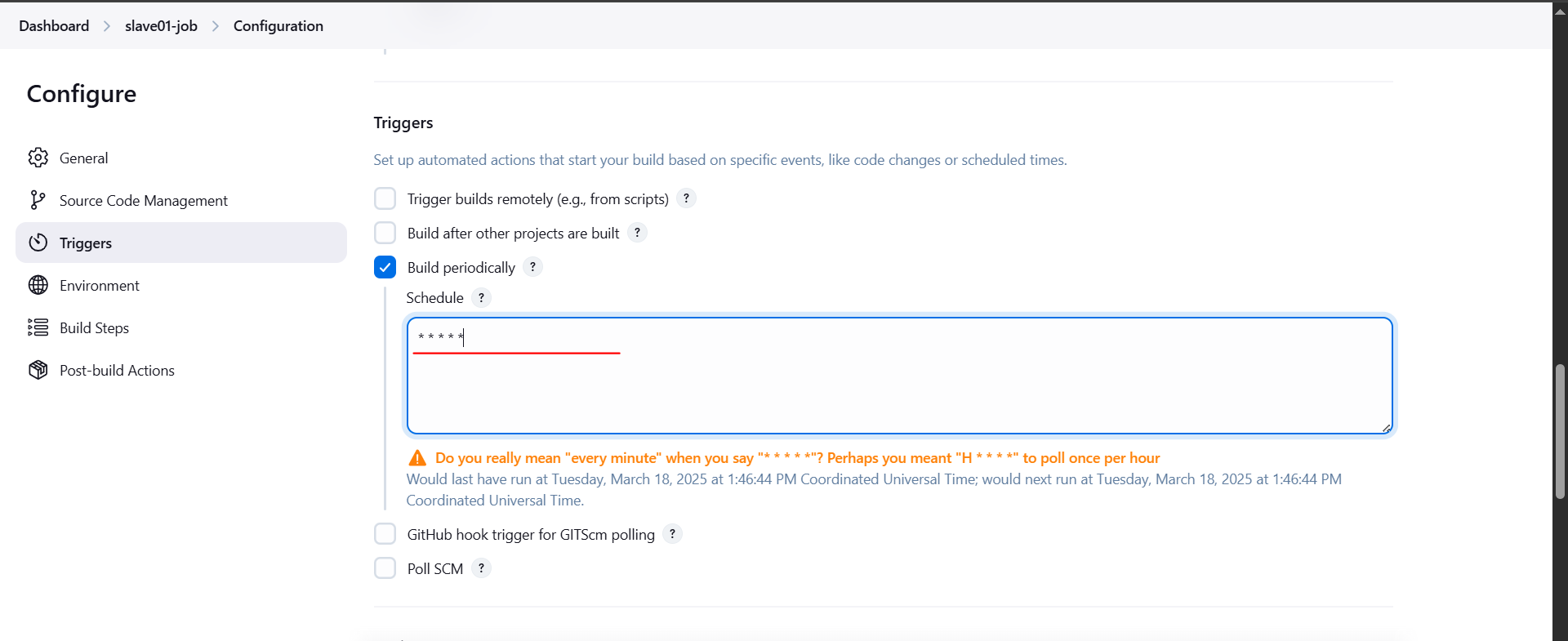
1. Configure poll scm and build periodical options in jenkins job.

To build periodically

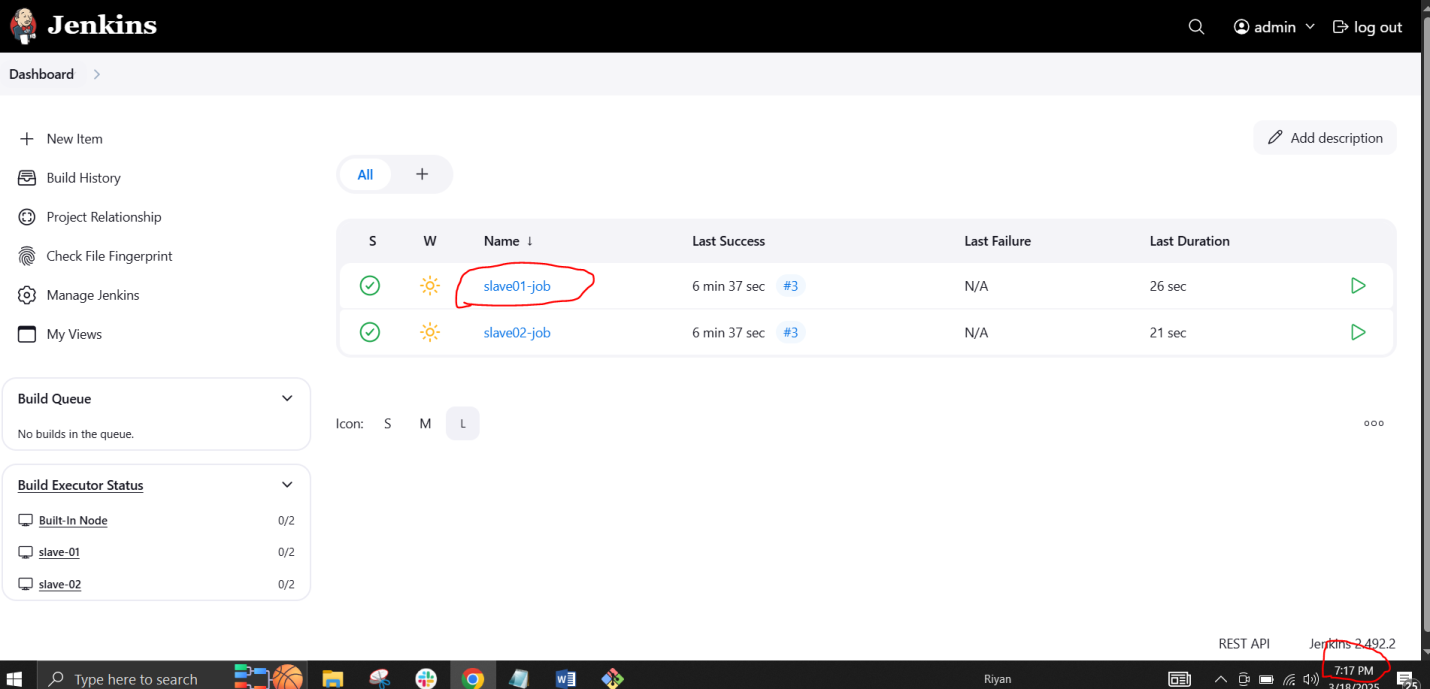
Click on the job, scroll below and check for Triggers

Under build periodically

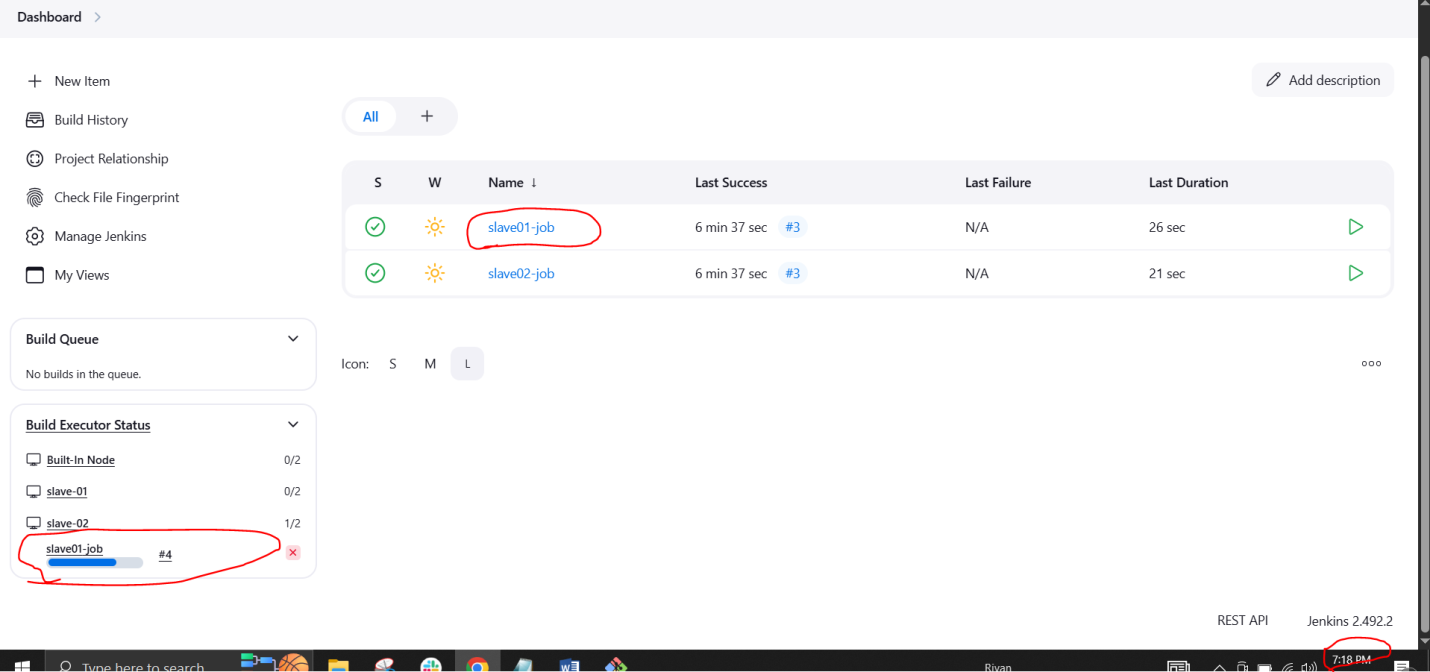
Add \* \* \* \* \*



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



After 1 minute it will trigger automatically

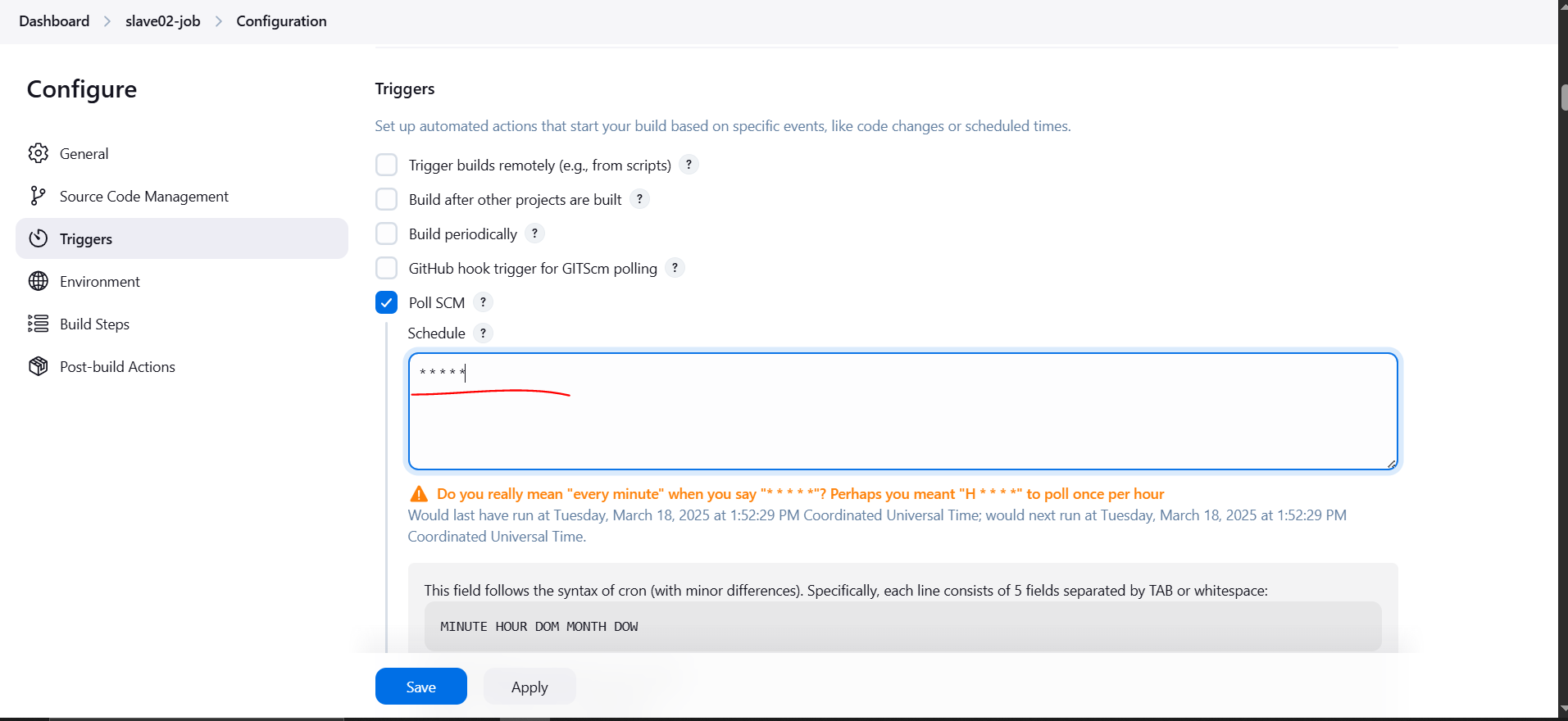


To configure poll SCM

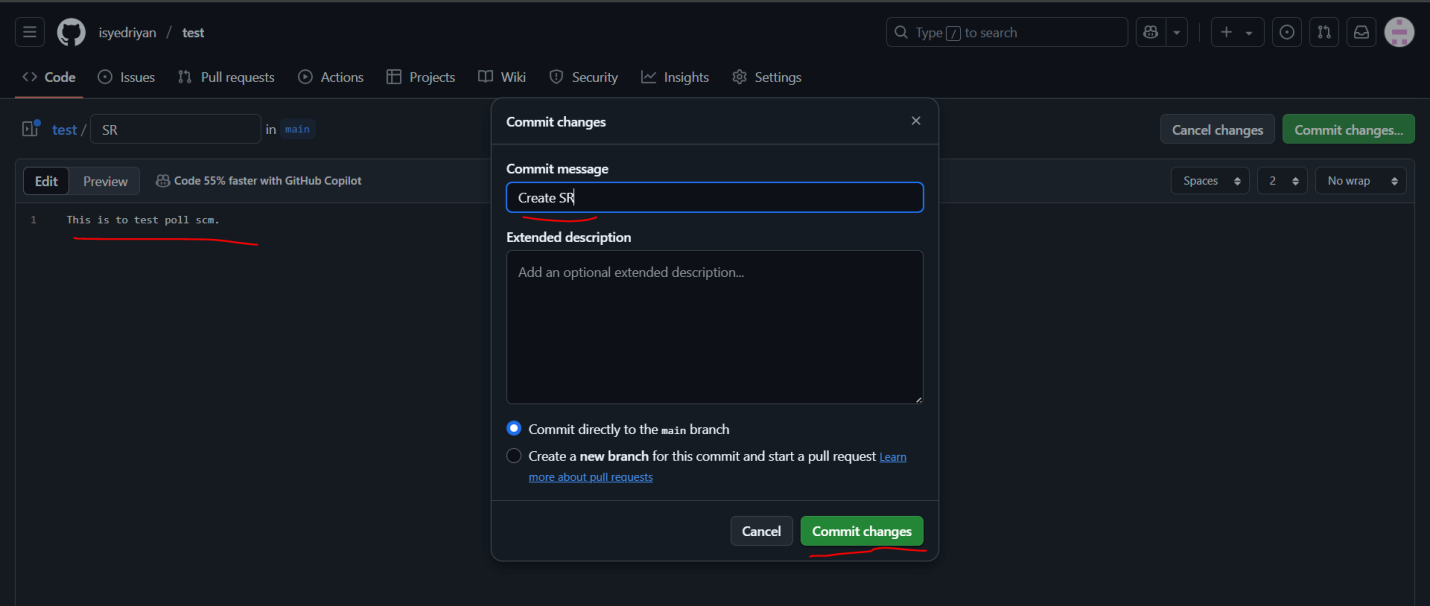
Click on the job, scroll below and check for Triggers

Under poll SCM

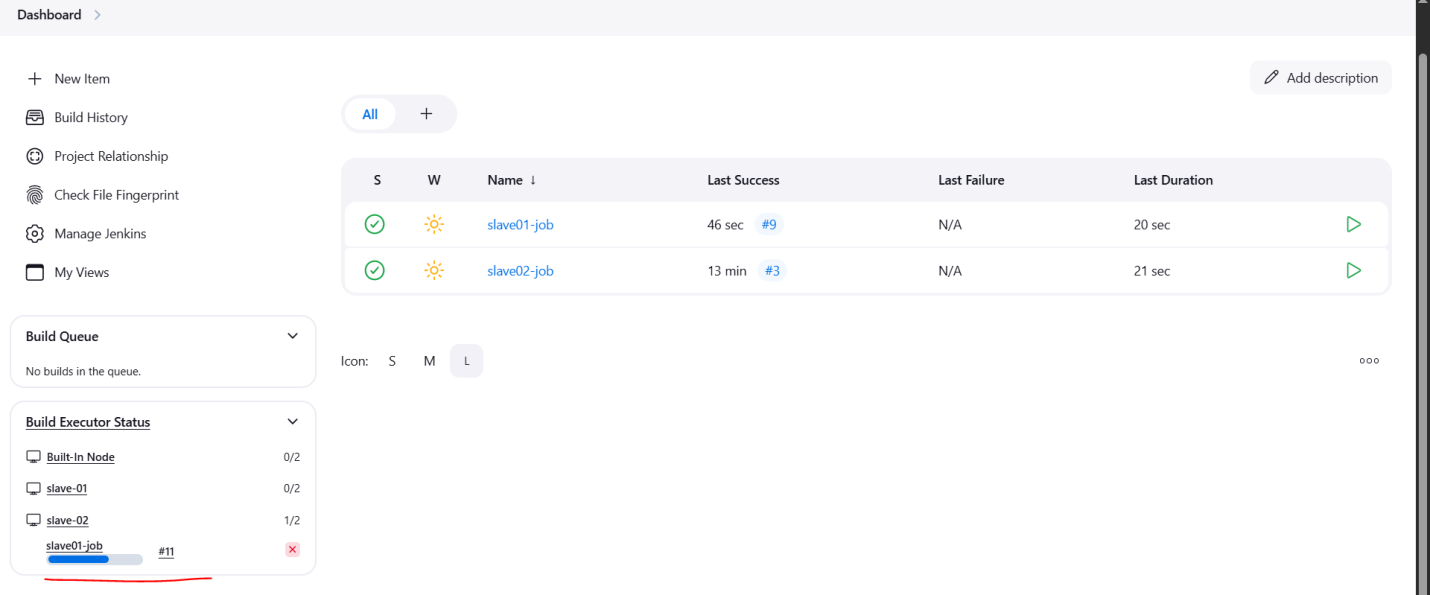
Add \* \* \* \* \*

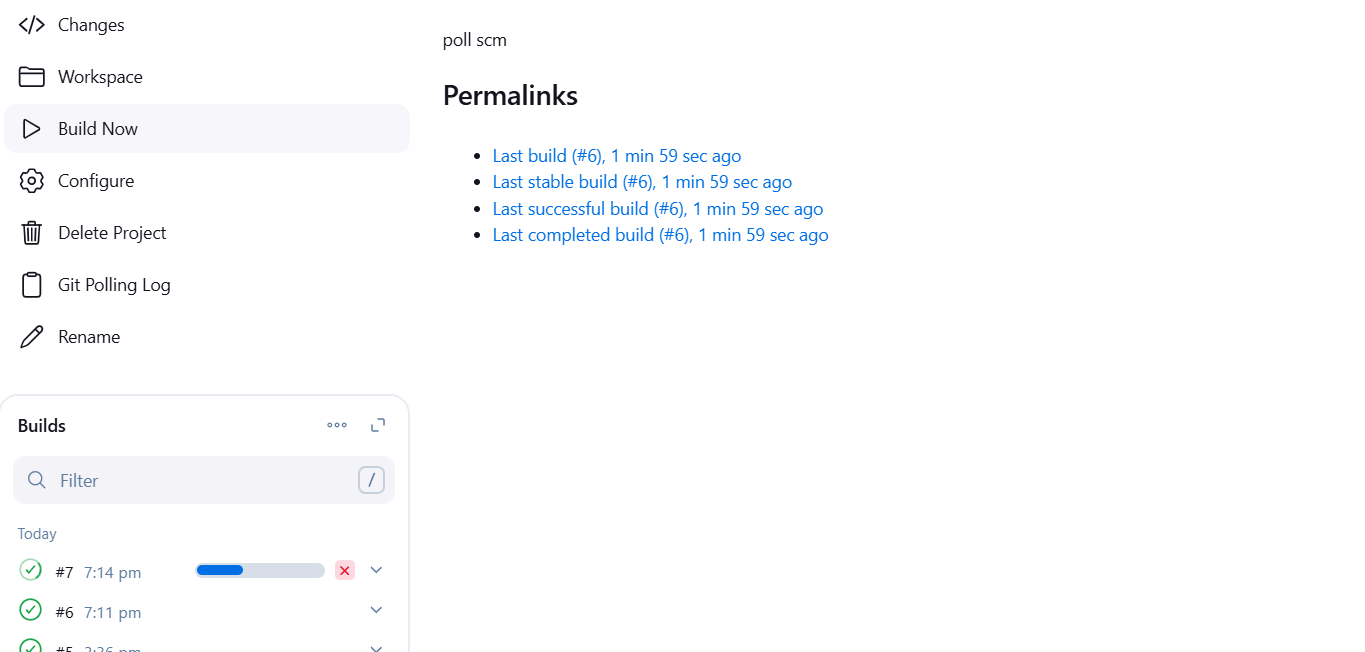


Now go to github and do some changes it will reflect in the Jenkins server



It has triggered in Jenkins server





1. 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

# Define backup directory and timestamp

BACKUP\_DIR="/var/backups/jenkins"

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

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

# Create the backup directory if it doesn't exist

mkdir -p $BACKUP\_DIR

# Stop Jenkins service to ensure consistent backup (optional)

echo "Stopping Jenkins service..."

sudo systemctl stop jenkins

# Compress Jenkins directory and save it as a backup file

echo "Creating backup..."

sudo tar -czvf $BACKUP\_DIR/$BACKUP\_FILE /var/lib/jenkins

# Start Jenkins service after backup

echo "Starting Jenkins service..."

sudo systemctl start jenkins

# Verify the backup

if [ -f $BACKUP\_DIR/$BACKUP\_FILE ]; then

echo "Backup created successfully: $BACKUP\_DIR/$BACKUP\_FILE"

else

echo "Backup failed!"

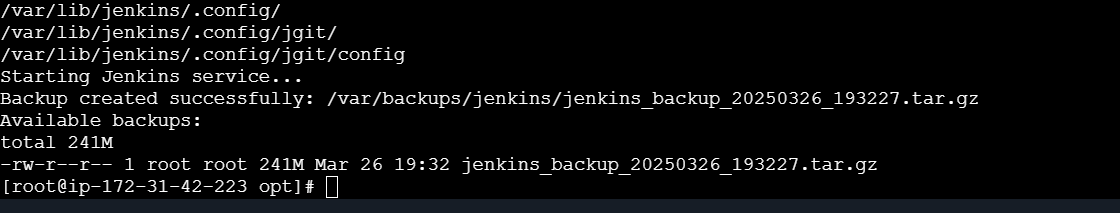
fi

# List backups (optional)

echo "Available backups:"

ls -lh $BACKUP\_DIR

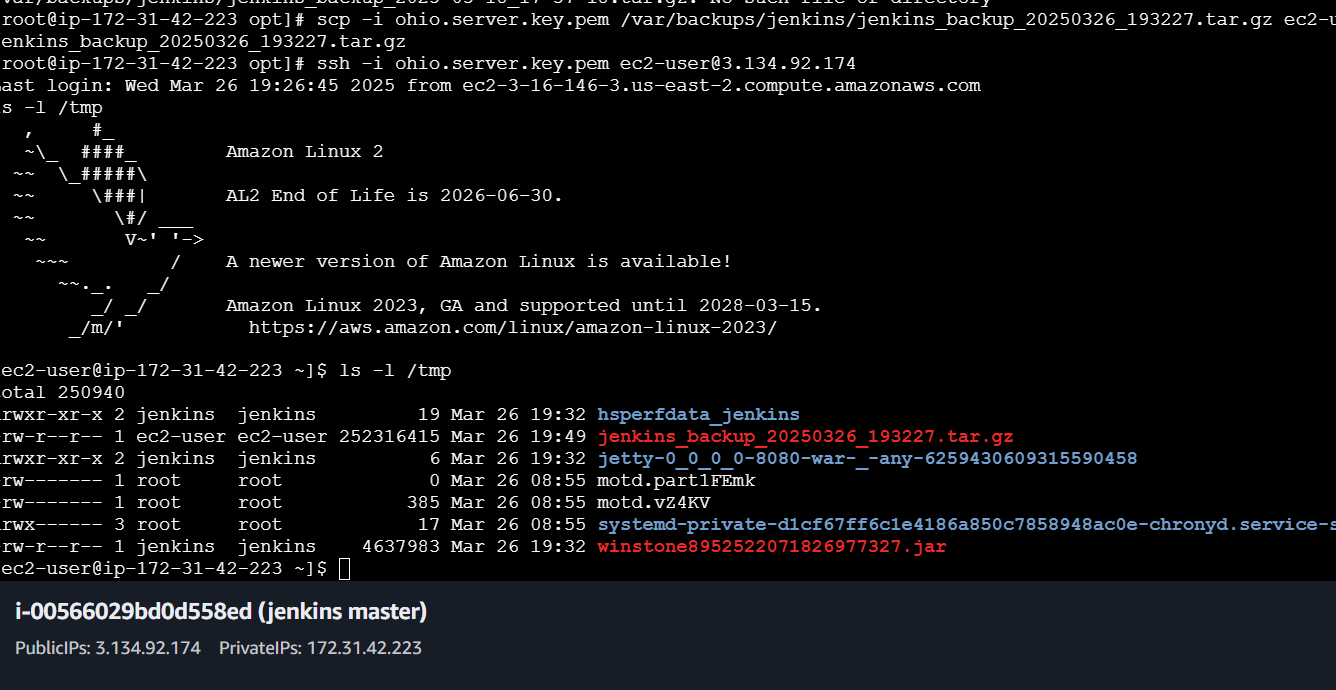
exit 0



scp -i ohio.server.key.pem /var/backups/jenkins/jenkins\_backup\_20250326\_193227.tar.gz [ec2-user@: 13.58.174.58/tmp](mailto:ec2-user@3.134.92.174:/tmp)

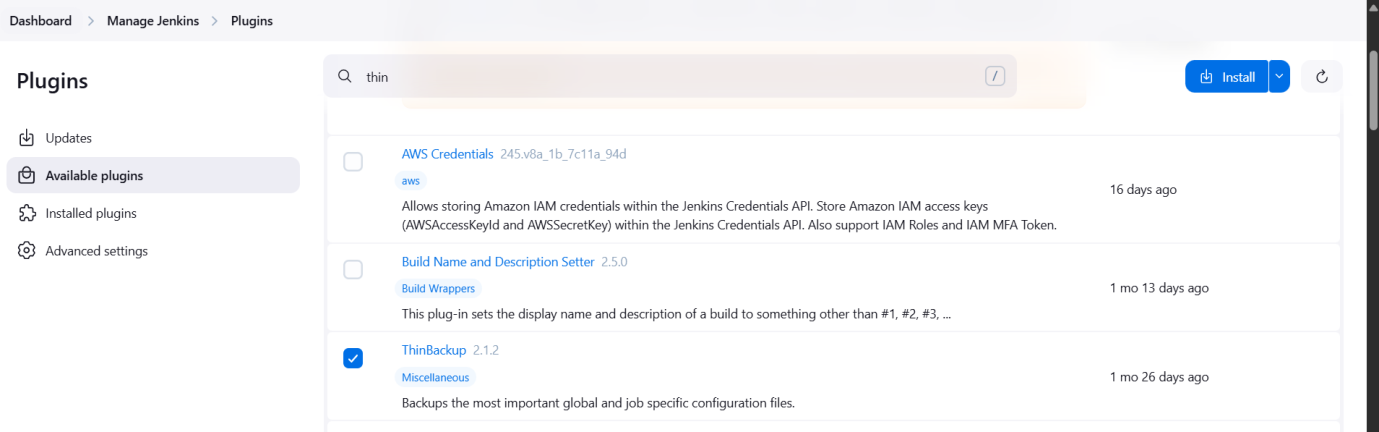
ssh -i ohio.server.key.pem ec2-user@3.134.92.174

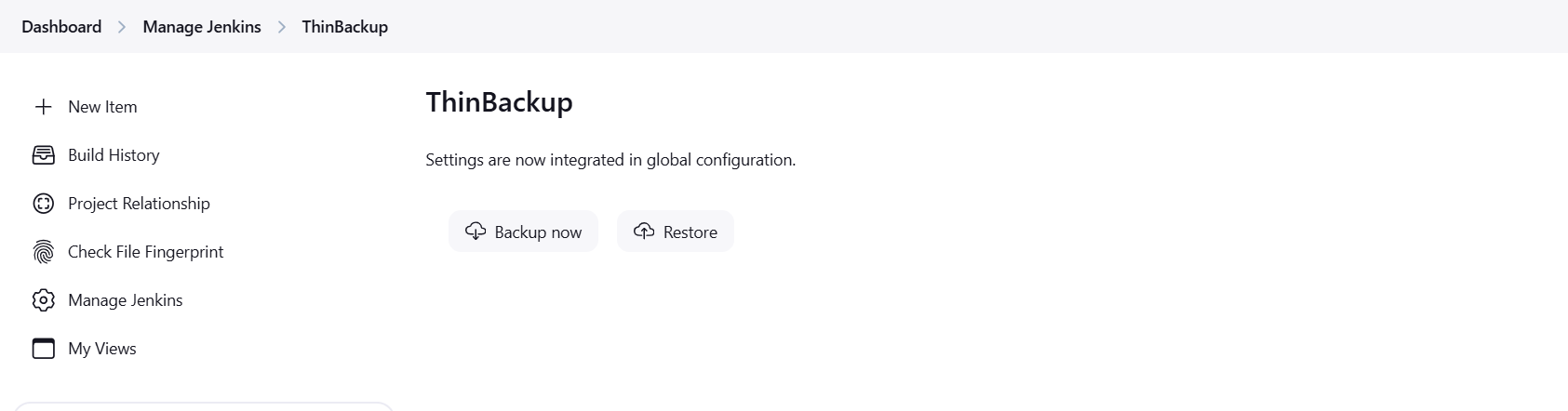
ls -l /tmp



jenkins\_backup\_20250326\_193227.tar.gz

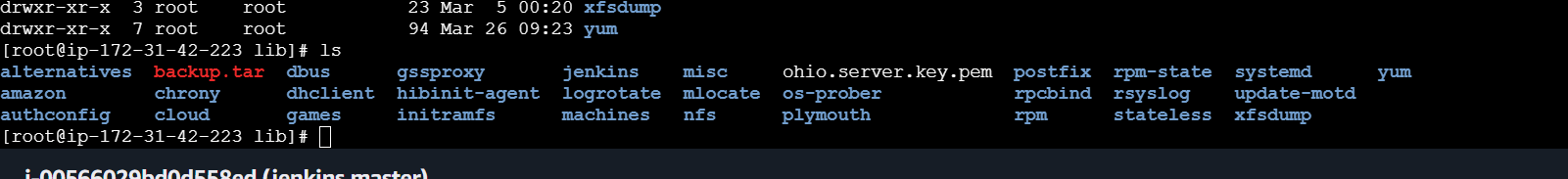
1. Take backup of jenkins using rethin backup plugin.





1. Setup a new jenkins server and dump the backup taken in task4.

[root@ip-172-31-42-223 lib]# tar -cvf backup.tar jenkins/



Here I mention the backup server ip in the below command and pemkey given permsion with chmod 7000 ohio.server.key.pem

scp -i ohio.server.key.pem backup.tar [ec2-user@13.58.174.58:/tmp](mailto:ec2-user@13.58.174.58:/tmp)

New Jenkins server installed and java jdk and Jenkins is installed now

