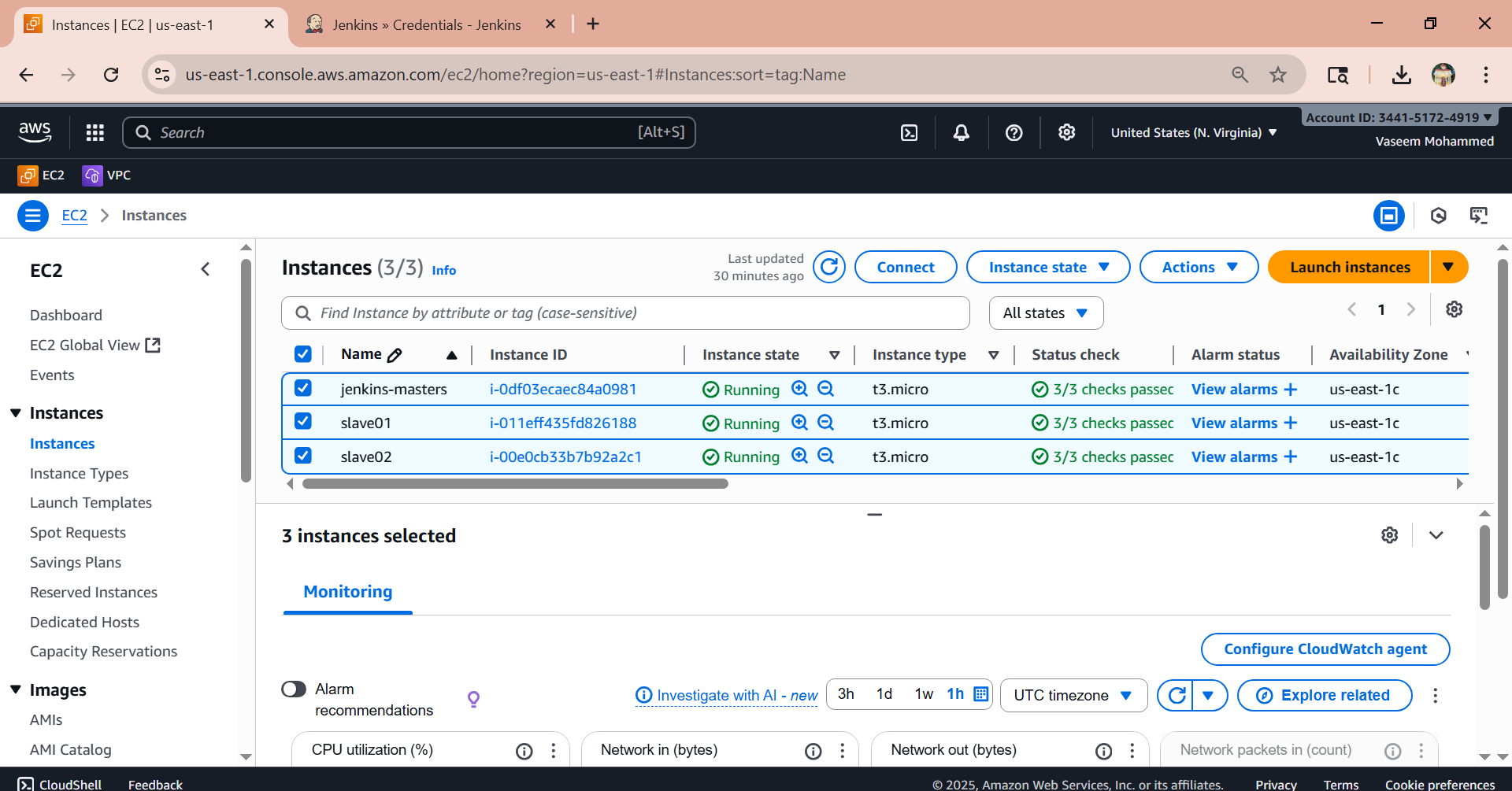
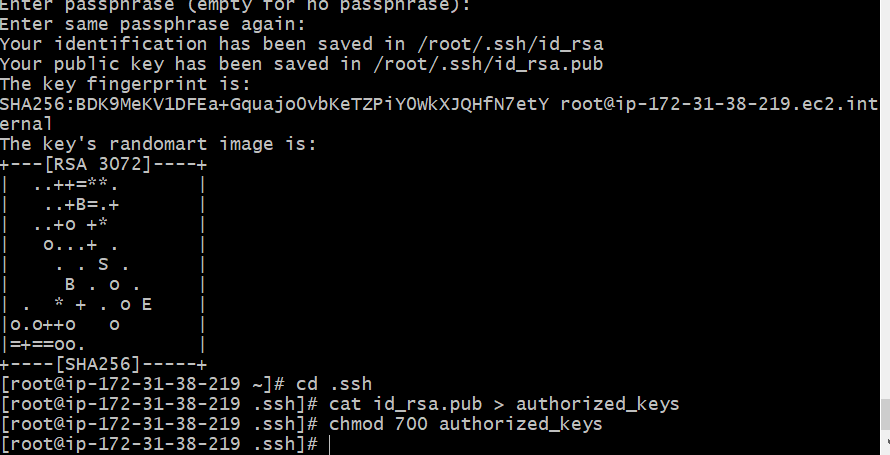
1. **Configure 2 slave machines in Jenkins master.**



Slave01

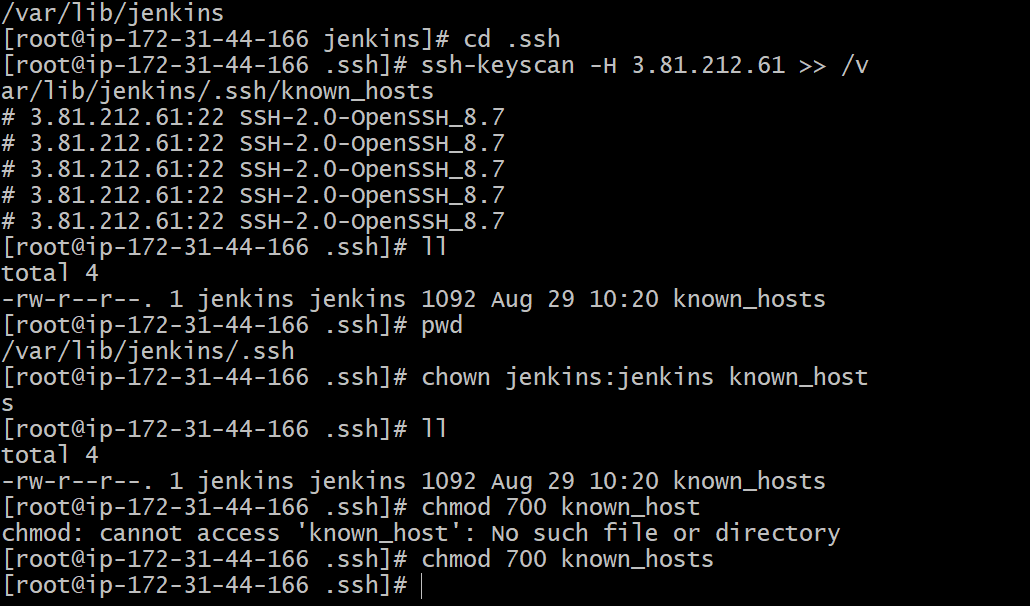
Install java and git on both ec2  
sudo yum install -y git  
sudo dnf install -y java-17-amazon-corretto  
Copy the key gen → ssh-keygen  
cd .ssh  
cat id\_rsa.pub > authorized\_keys  
chmod 700 authorized\_keys



Slave02  
Copy the key gen → ssh-keygen  
cd .ssh  
cat id\_rsa.pub > authorized\_keys  
chmod 700 authorized\_keys



Step to do in Jenkins-master instance

Create Jenkins SSH directory:  
mkdir -p /var/lib/jenkins/.ssh  
cd /var/lib/jenkins/.ssh  
ssh-keyscan -H SLAVE-NODE-PUBLIC-IP >> /var/lib/jenkins/.ssh/known\_hosts  
chown jenkins:jenkins /var/lib/jenkins/.ssh/known\_hosts  
chmod 700 /var/lib/jenkins/.ssh/known\_hosts  


Now need to create node for slave

Go to >jenkuns>manage jenkins>Nodes>create node as slave01

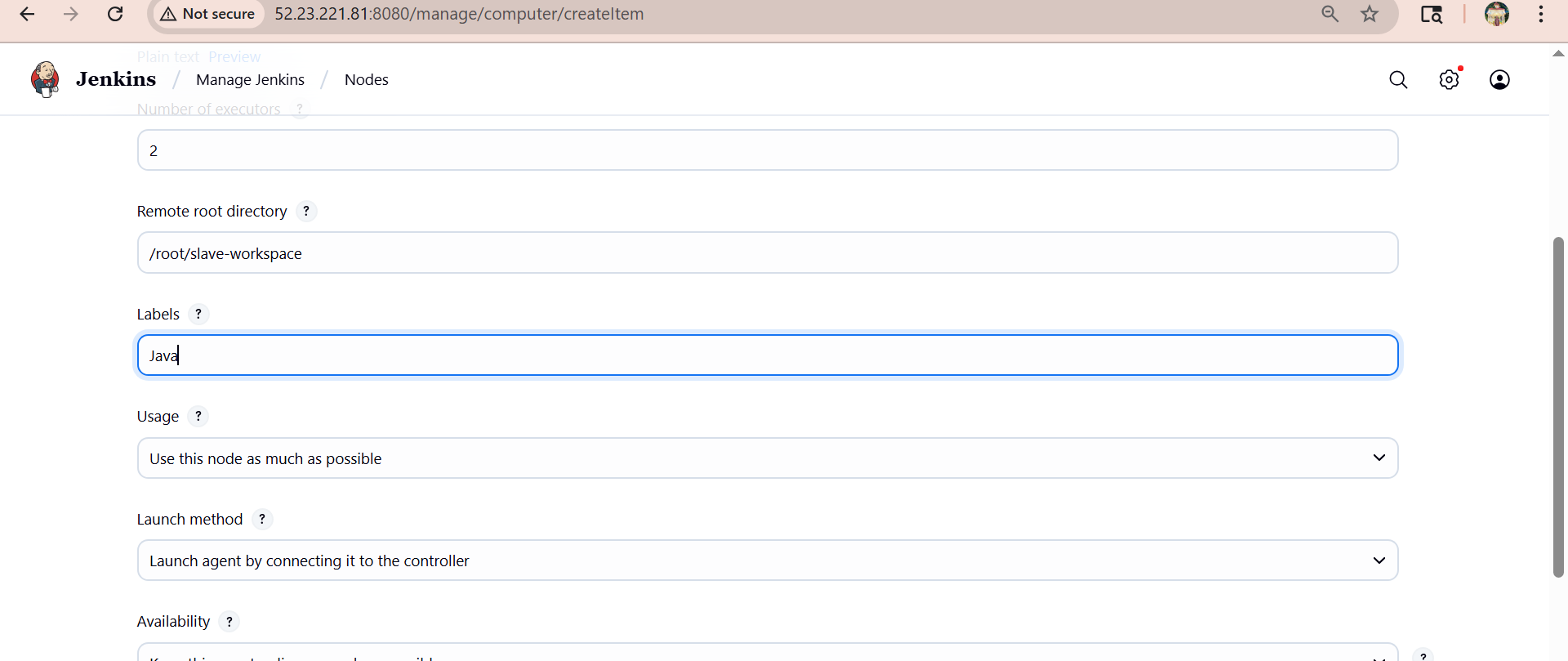
In remote root directory need to mention slave directory path

Create directory in slave intance git bash

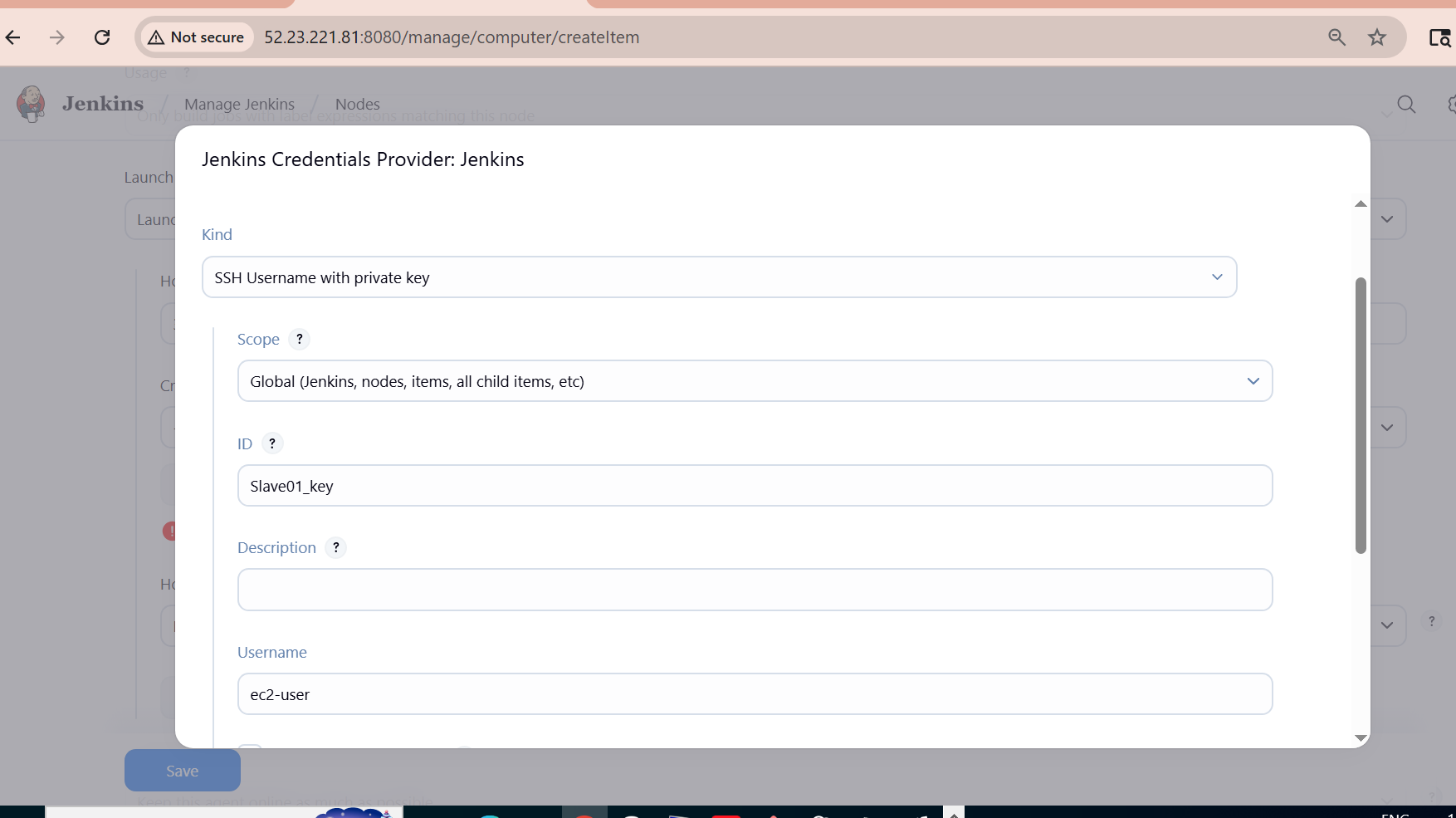
Mkdir slave-workspace

Pwd>/root/slave-workspace

And mention label as java

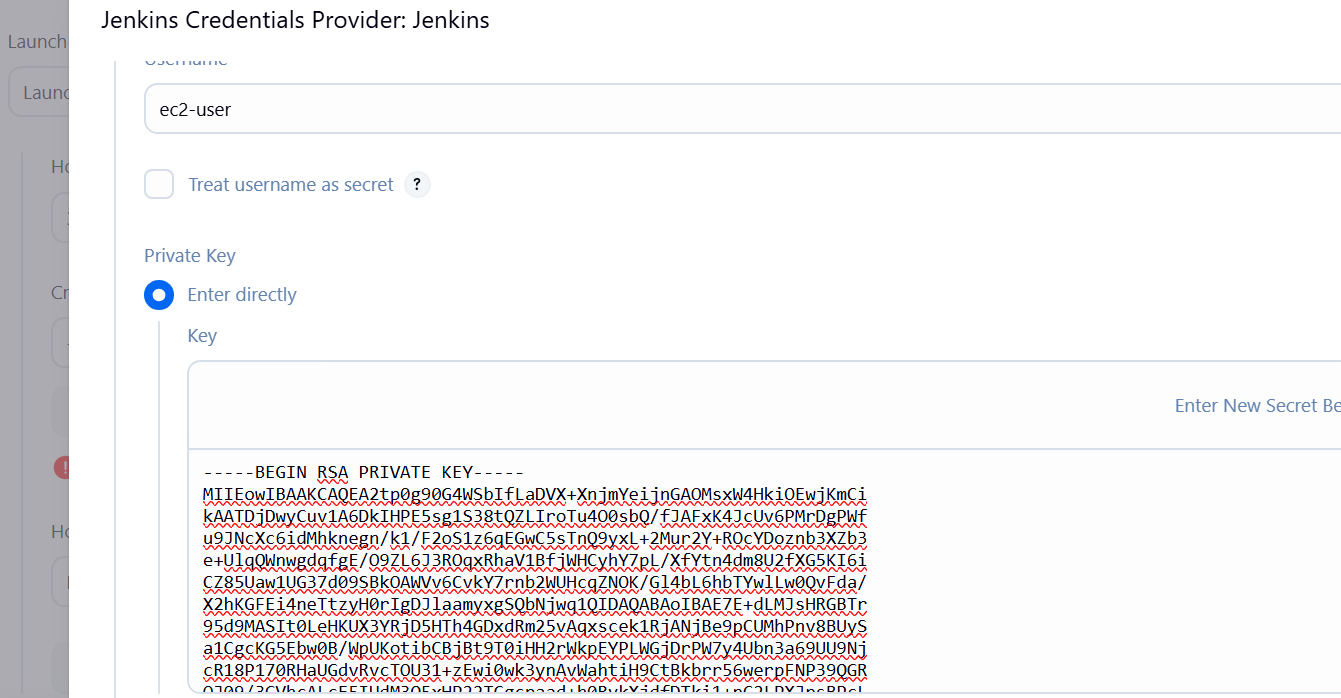


Select this option in dropdown



And then copy the pem key

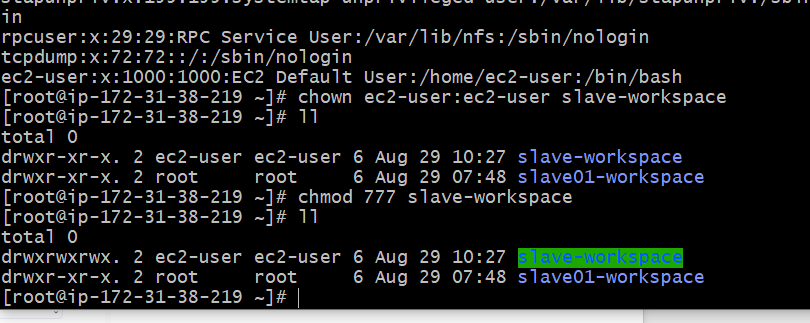
Cat waseem.key



Than we need to change the ownership in slave01 gitbash

chown ec2-user:ec2-user slave-workspace

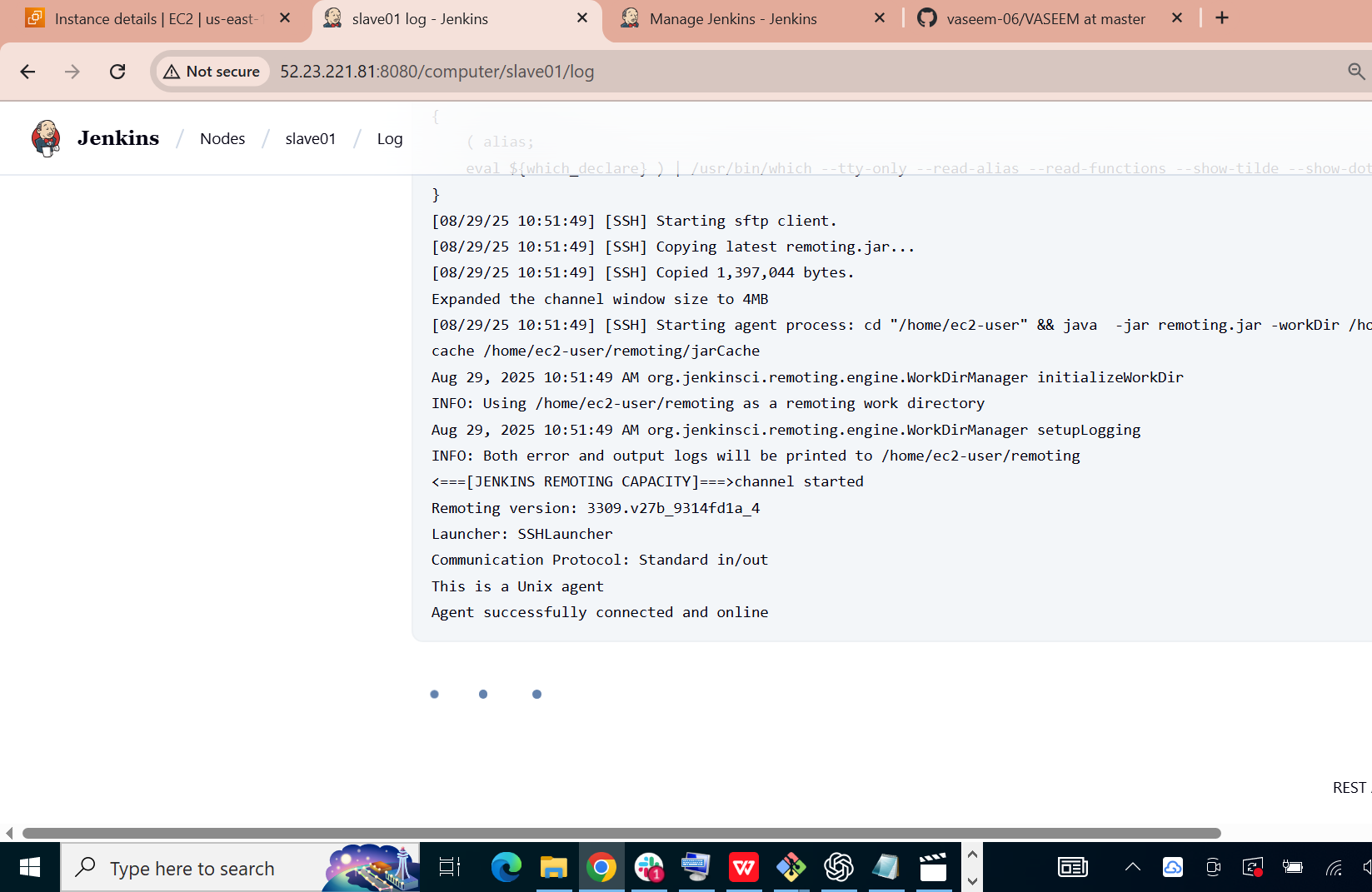
Chmod 777 slave-workspace



If there is any error came like its not getting failed

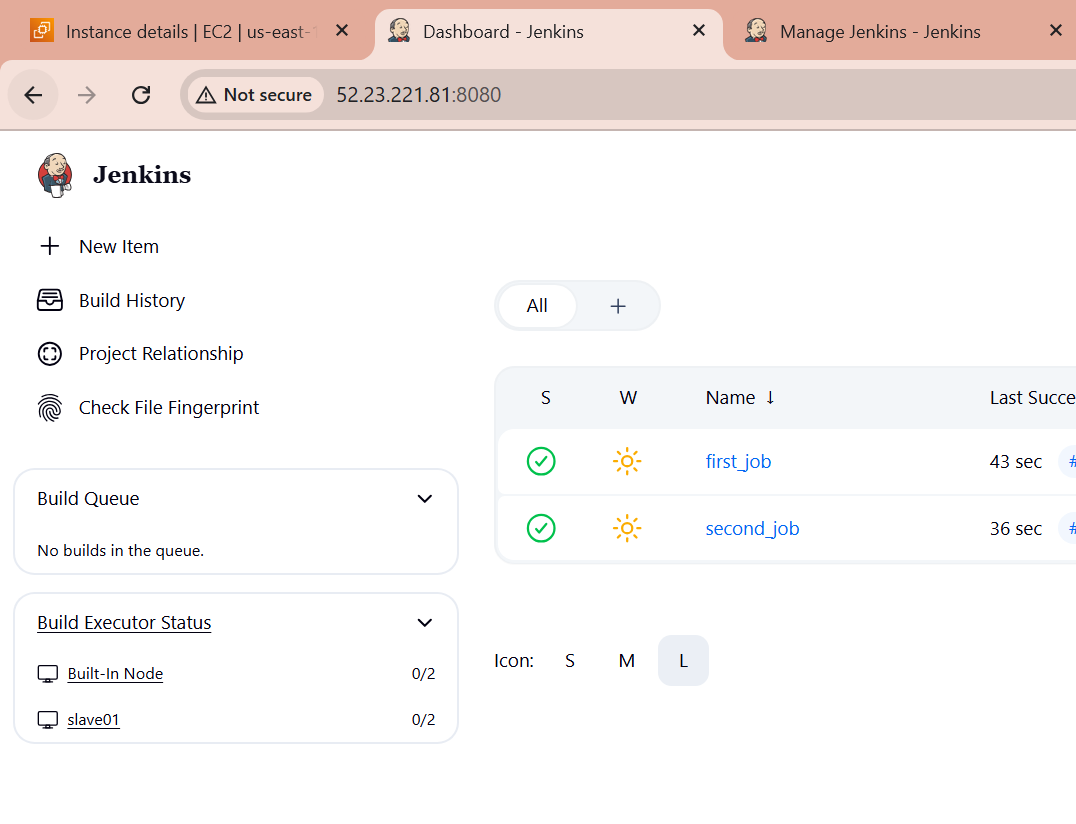
Than we need to change the location in slave01 gitbash as

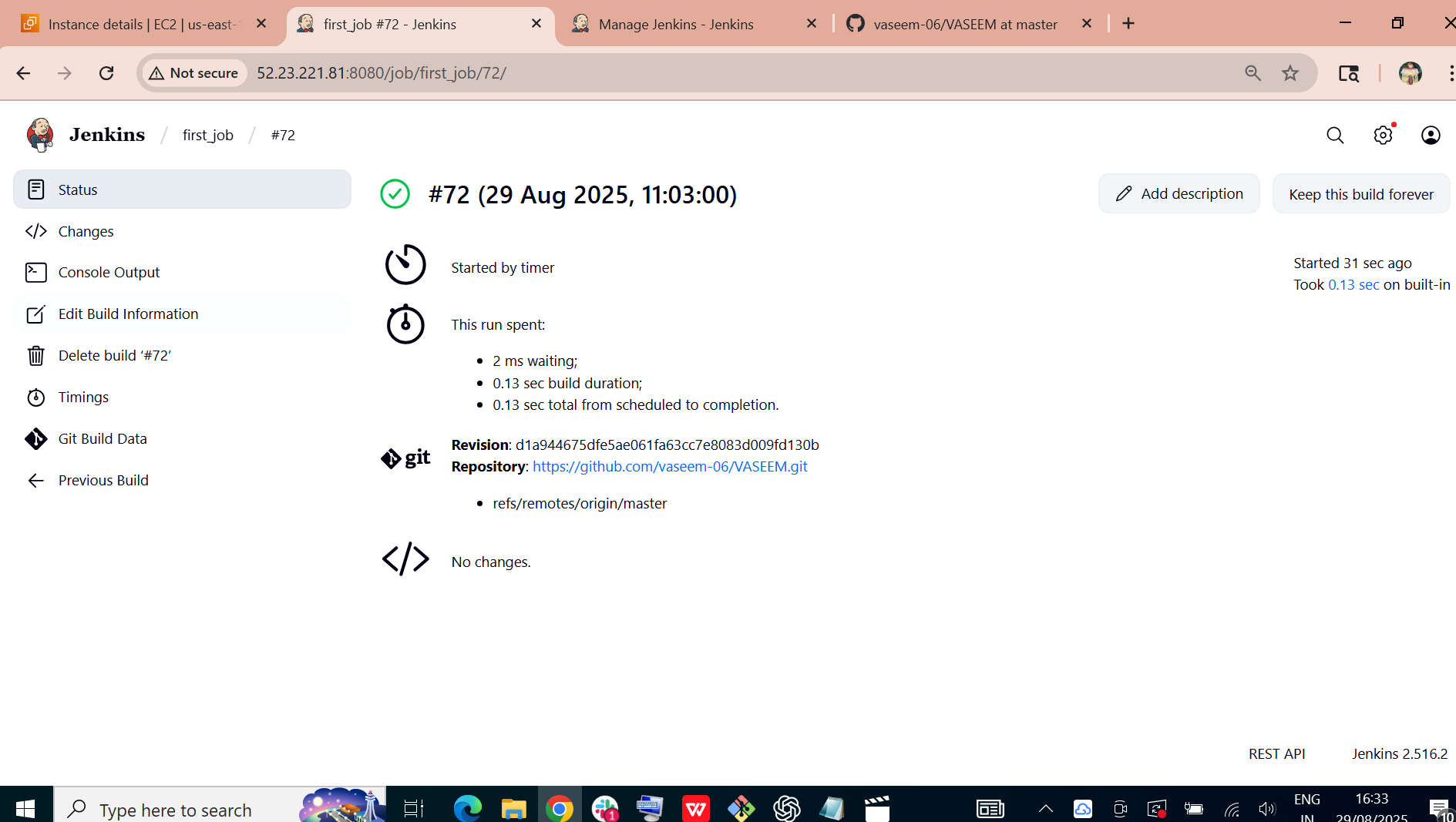
Exit>pwd>/home/ec2-user



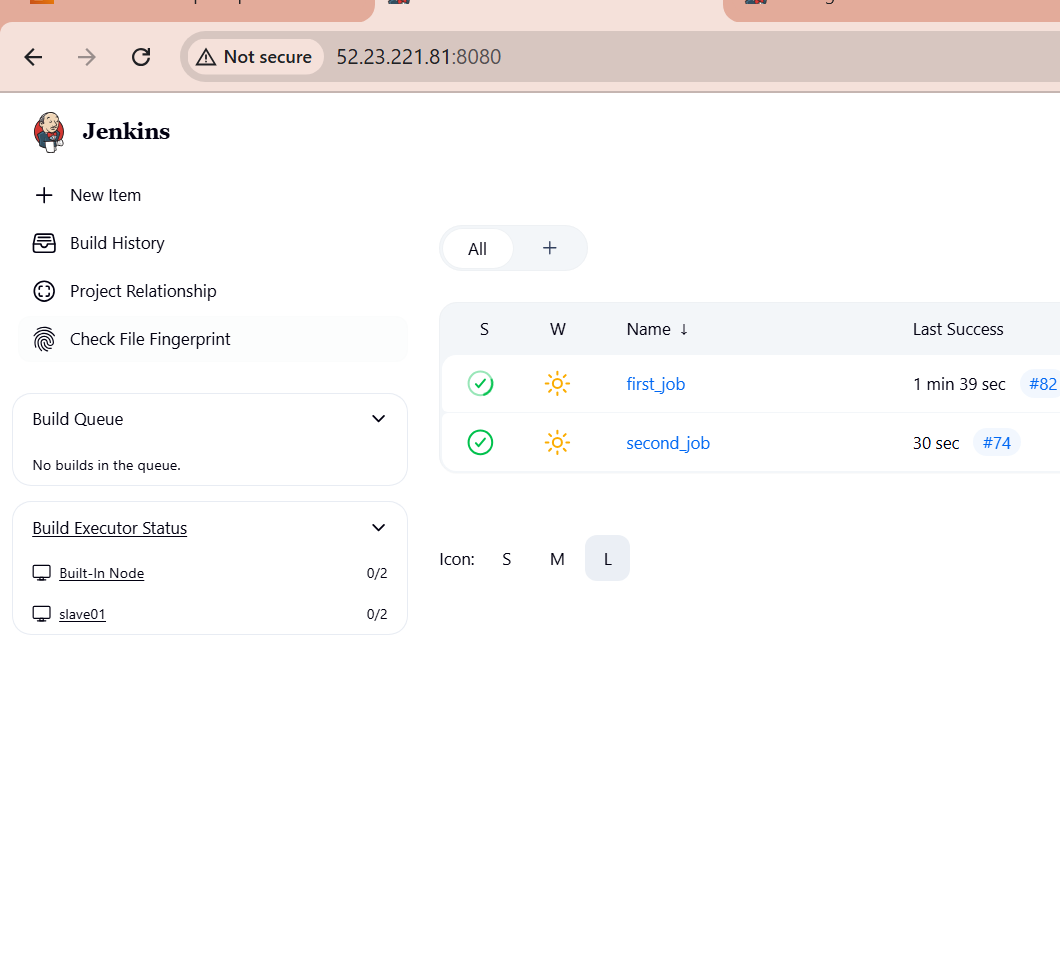
Than we can se ethat slave01 is connected

Go to jenkins dashboard





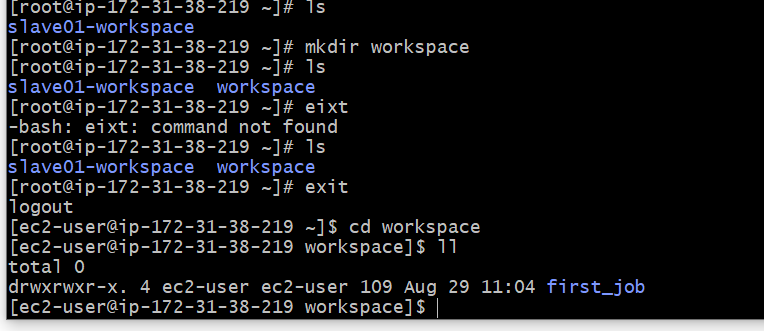


Than it will get job in slave01 in below 

And create mkdir workspace, check in ec2-user

Cd workspace>ll

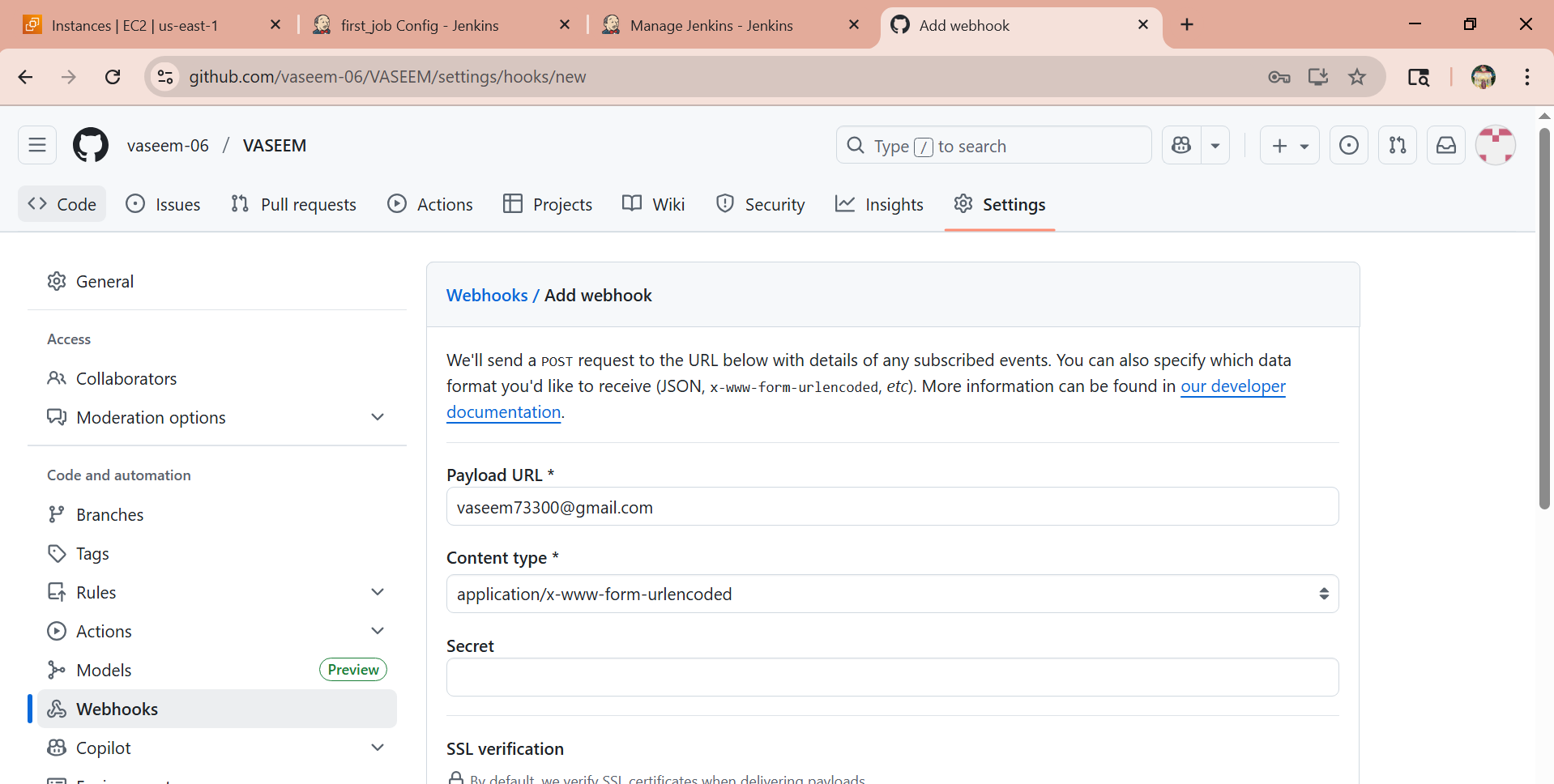
It will appear in that as first\_job

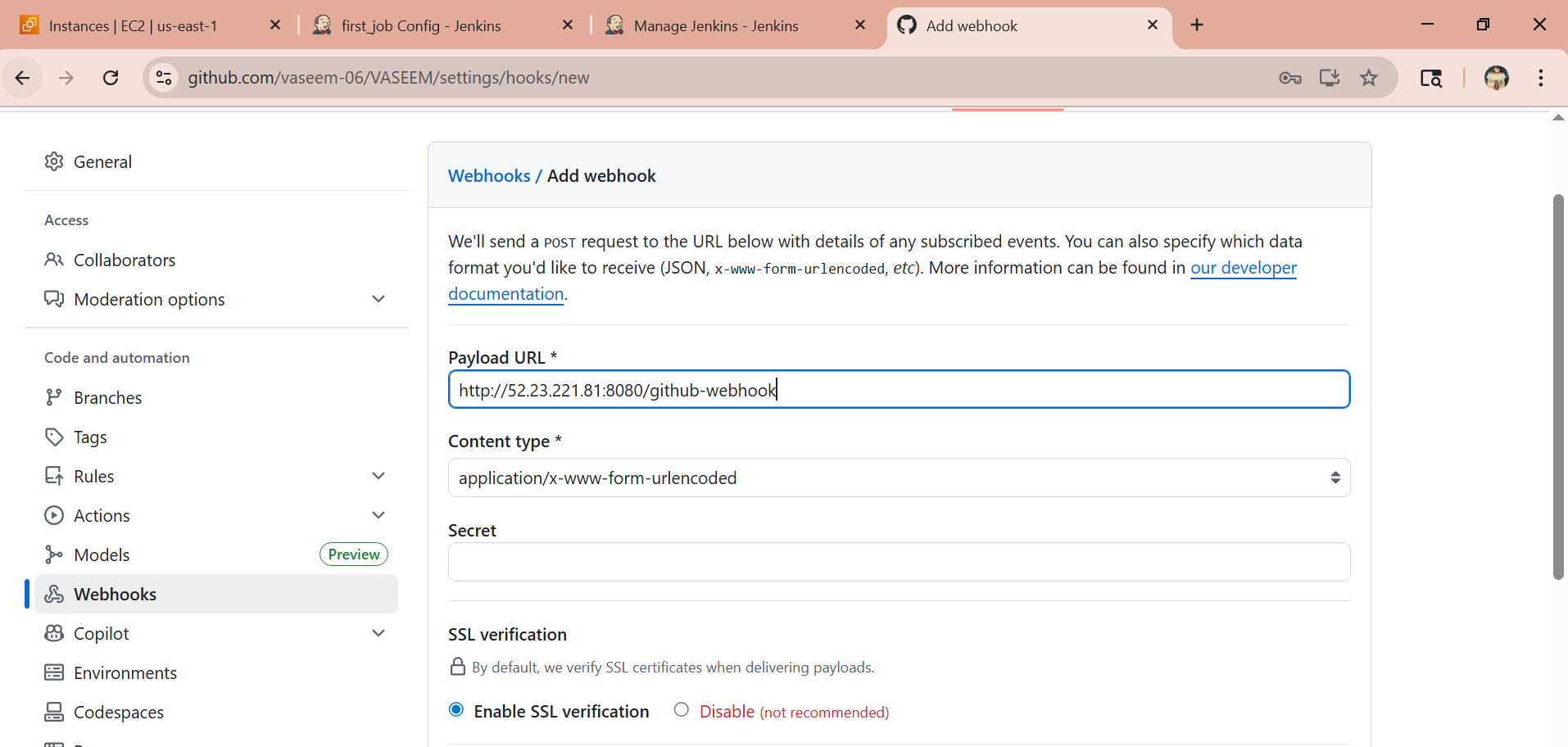


**2) Configure webhooks to Jenkins job.**

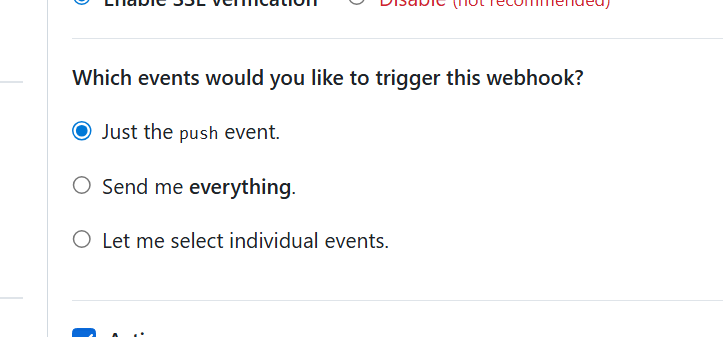
Go to github>our repo>setting>left side >webhooks

Then in Payload URl need to add Jenkins URL as (http://52.23.221.81:8080/github-webhooks)

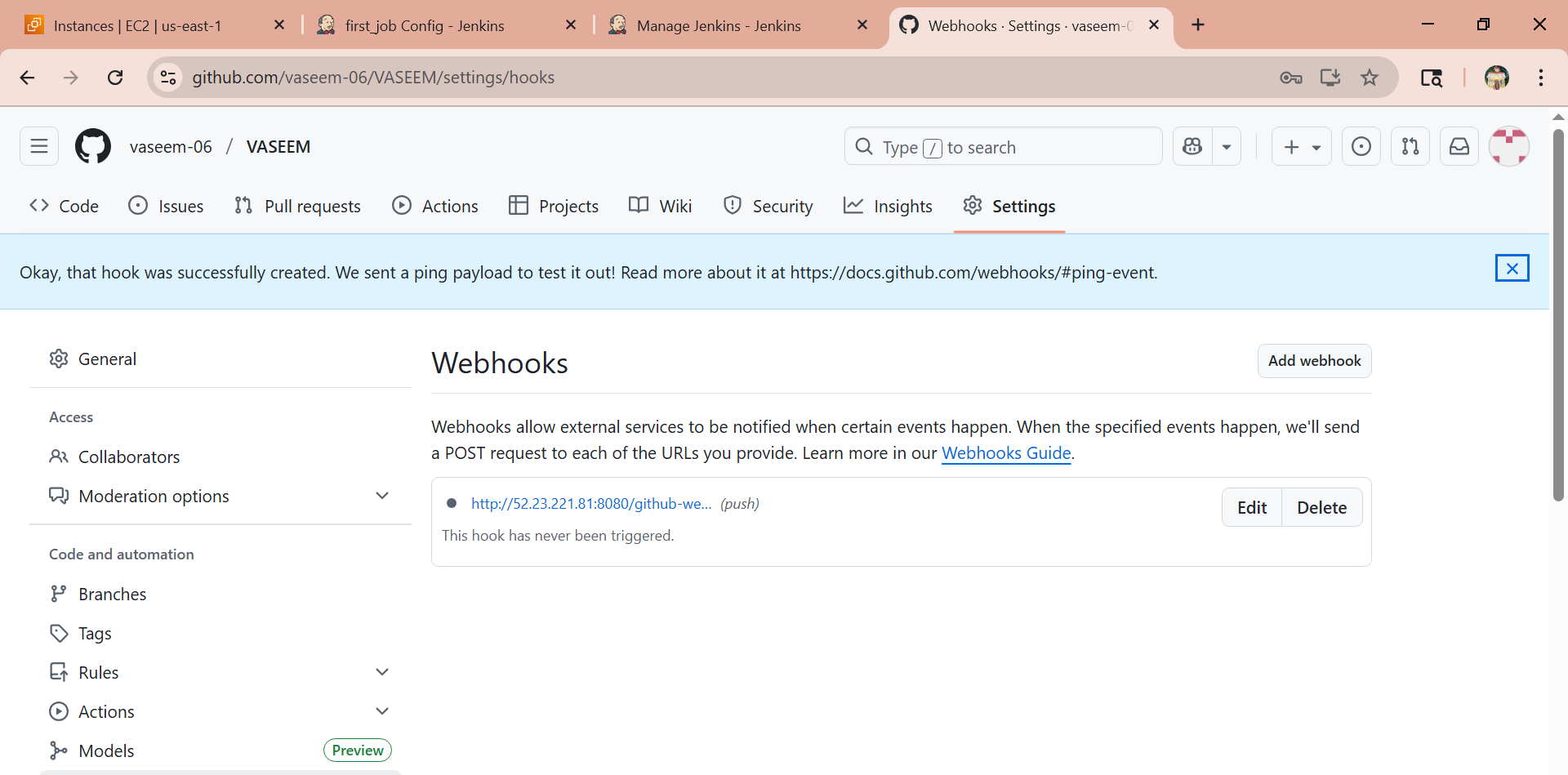




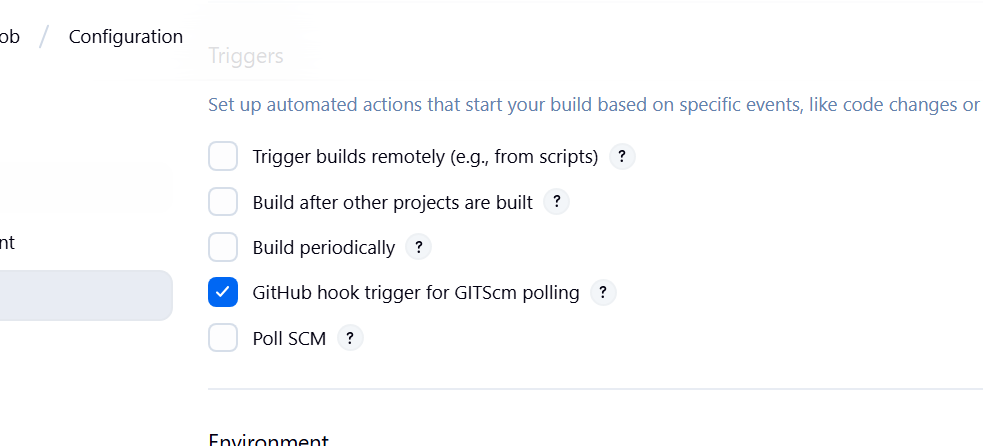
Here we need to select >(Select Everthing)



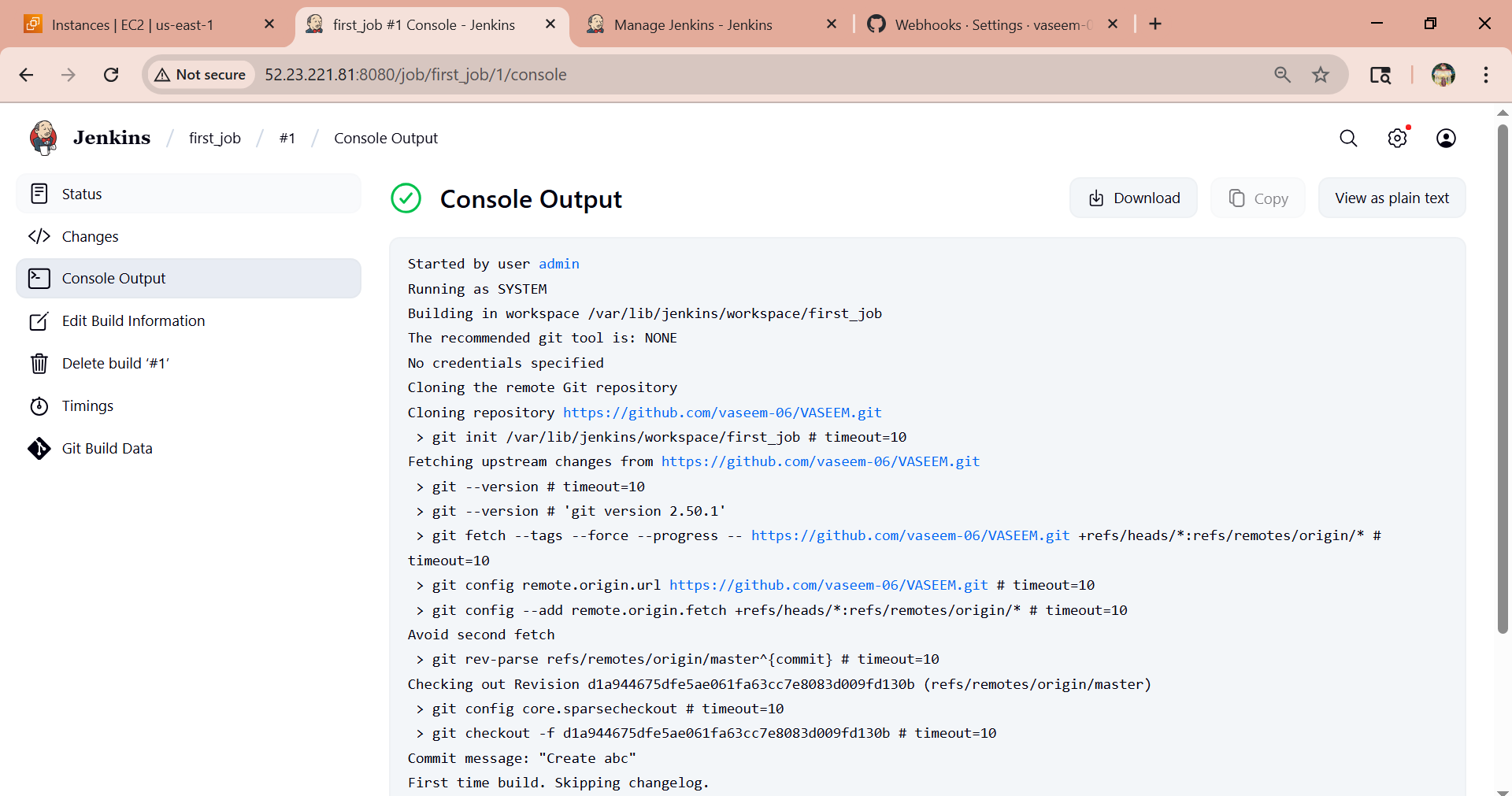
Here we have successsfully created webhooks



In Jenkins we need to select this option to get trigger



After the it will get trigger in below image



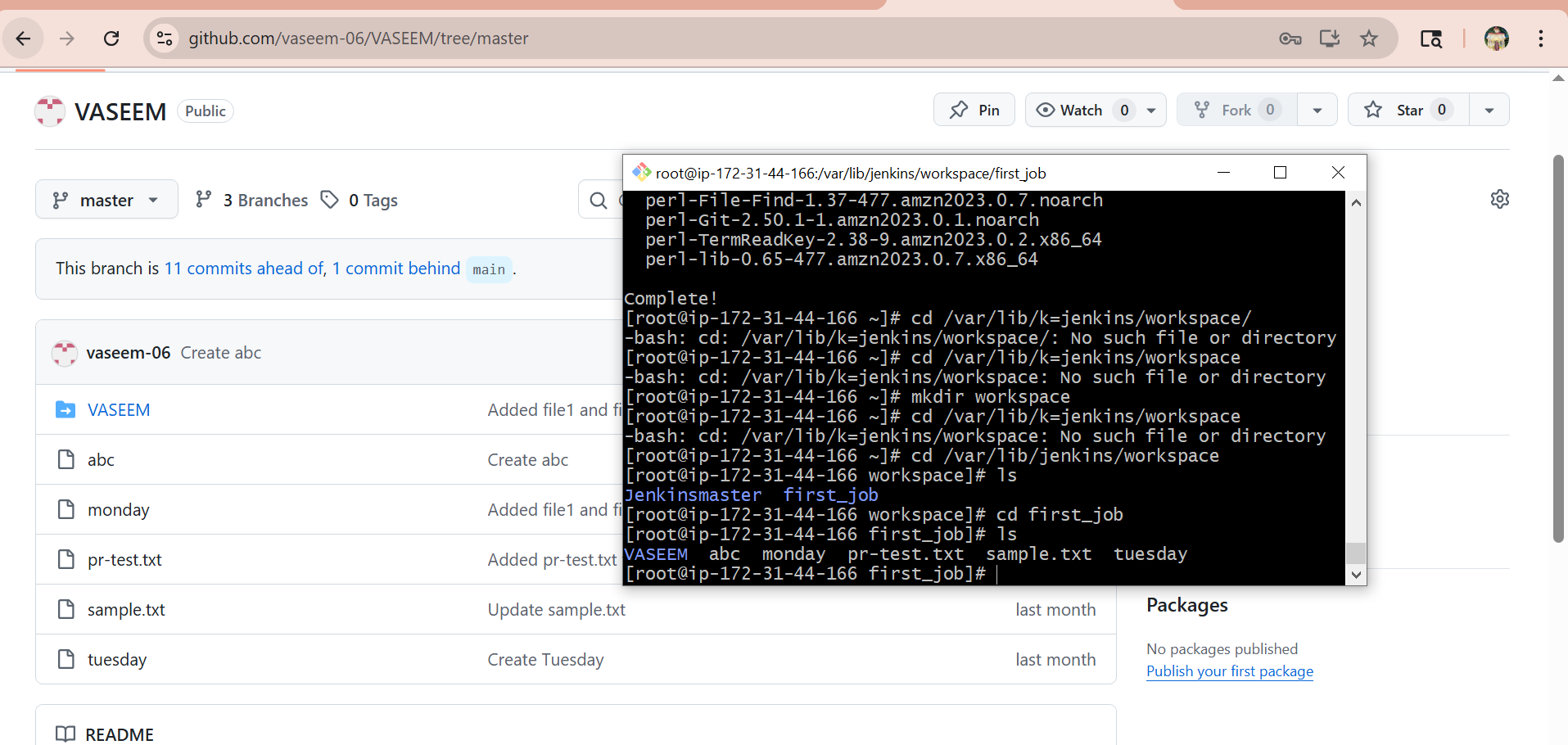
Than we can check bu=y using command

Need to create directory as mkdir workspace

cd /var/lib/jenkins/workspace

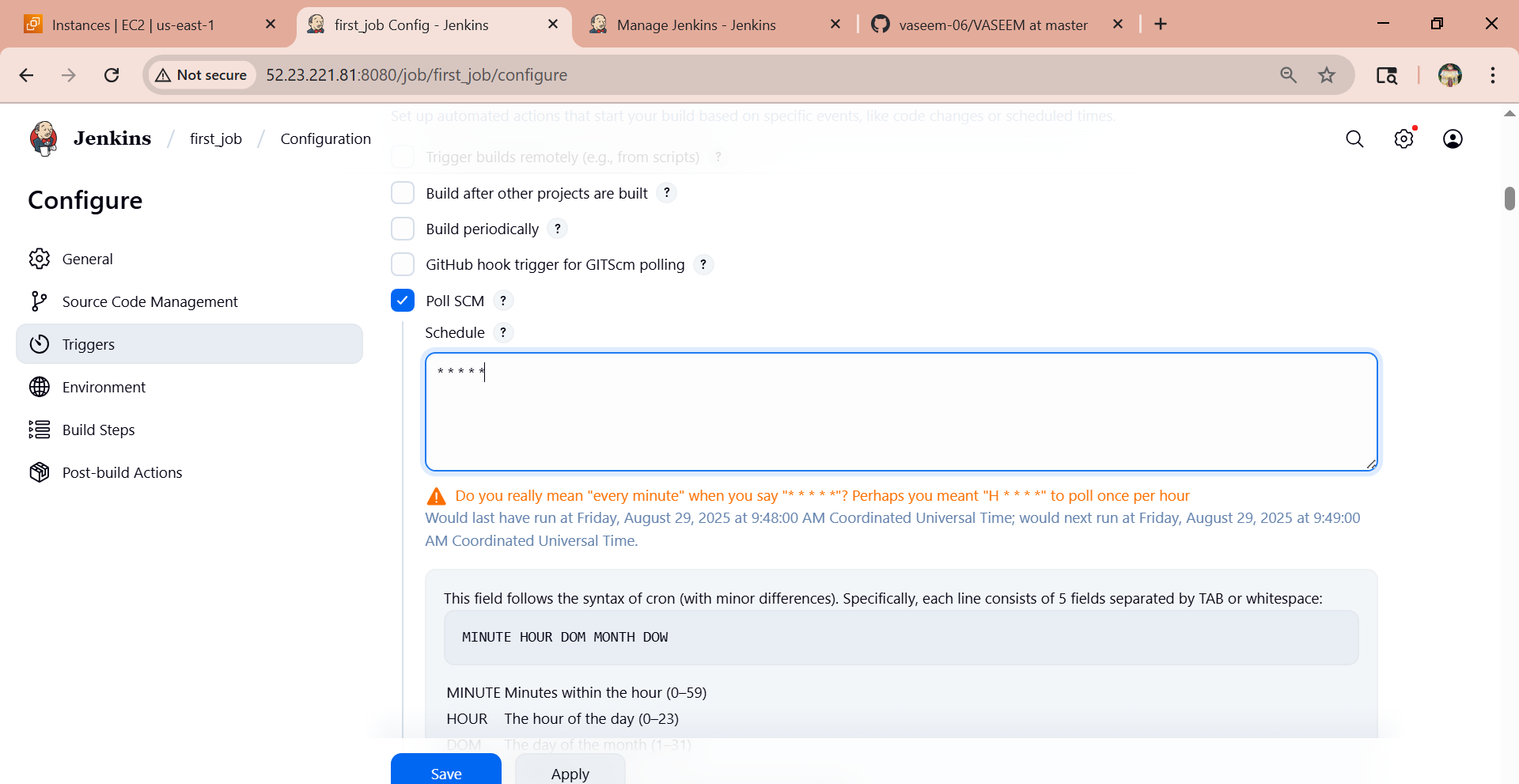
cd first\_job>ls

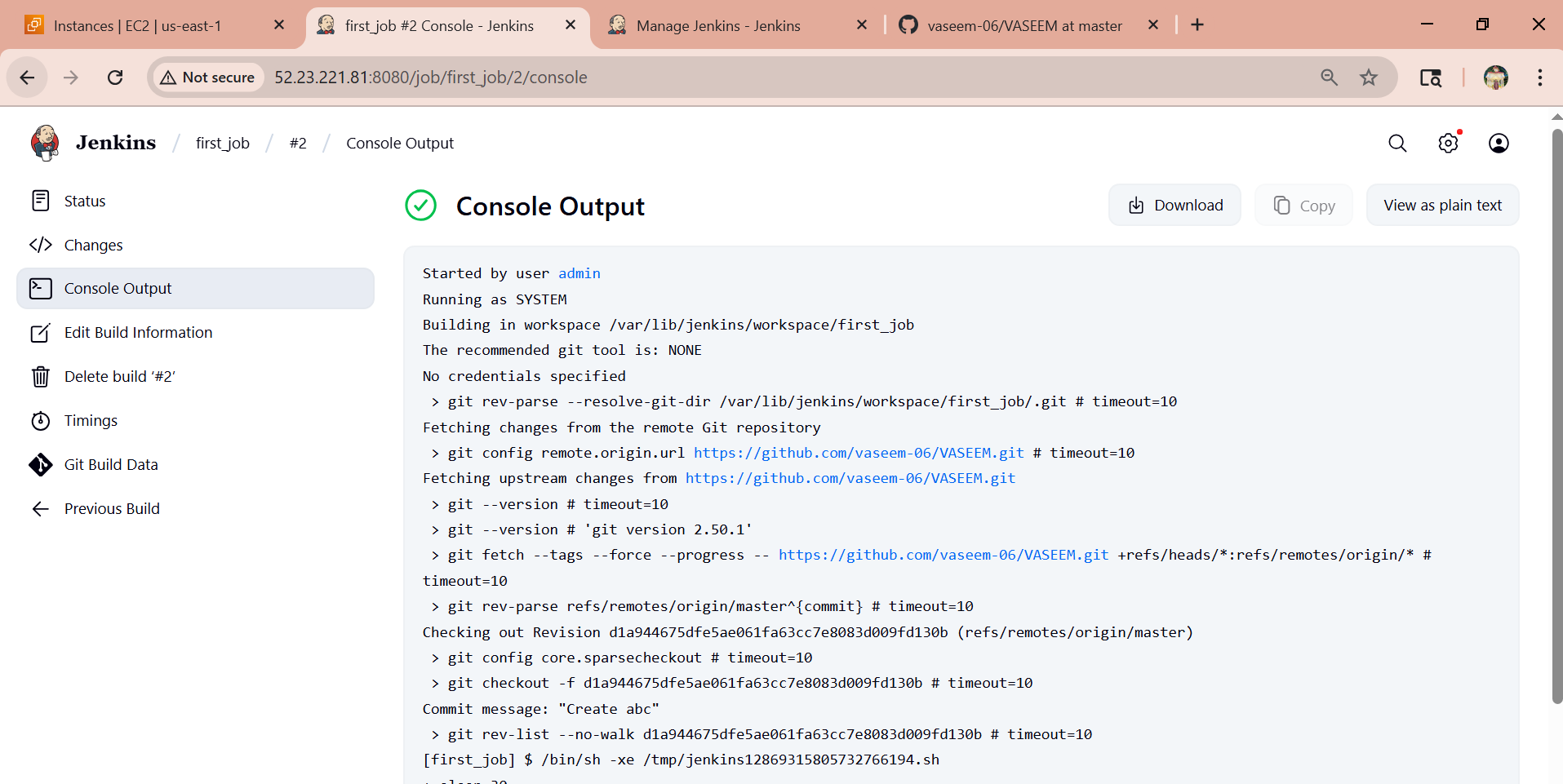
It will show the files which is in repo



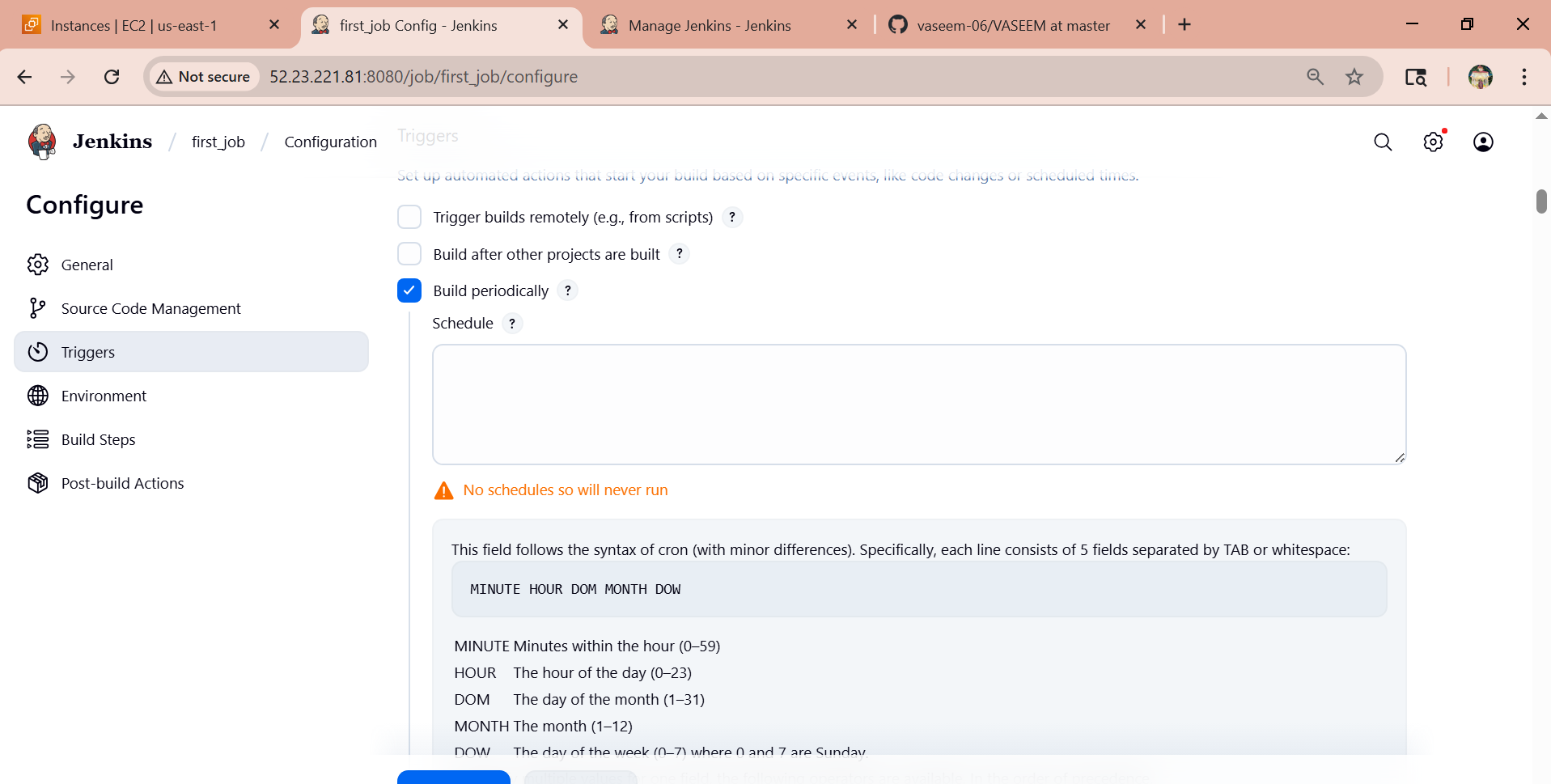
**3) Configure poll scm and build periodical options in Jenkins job.**

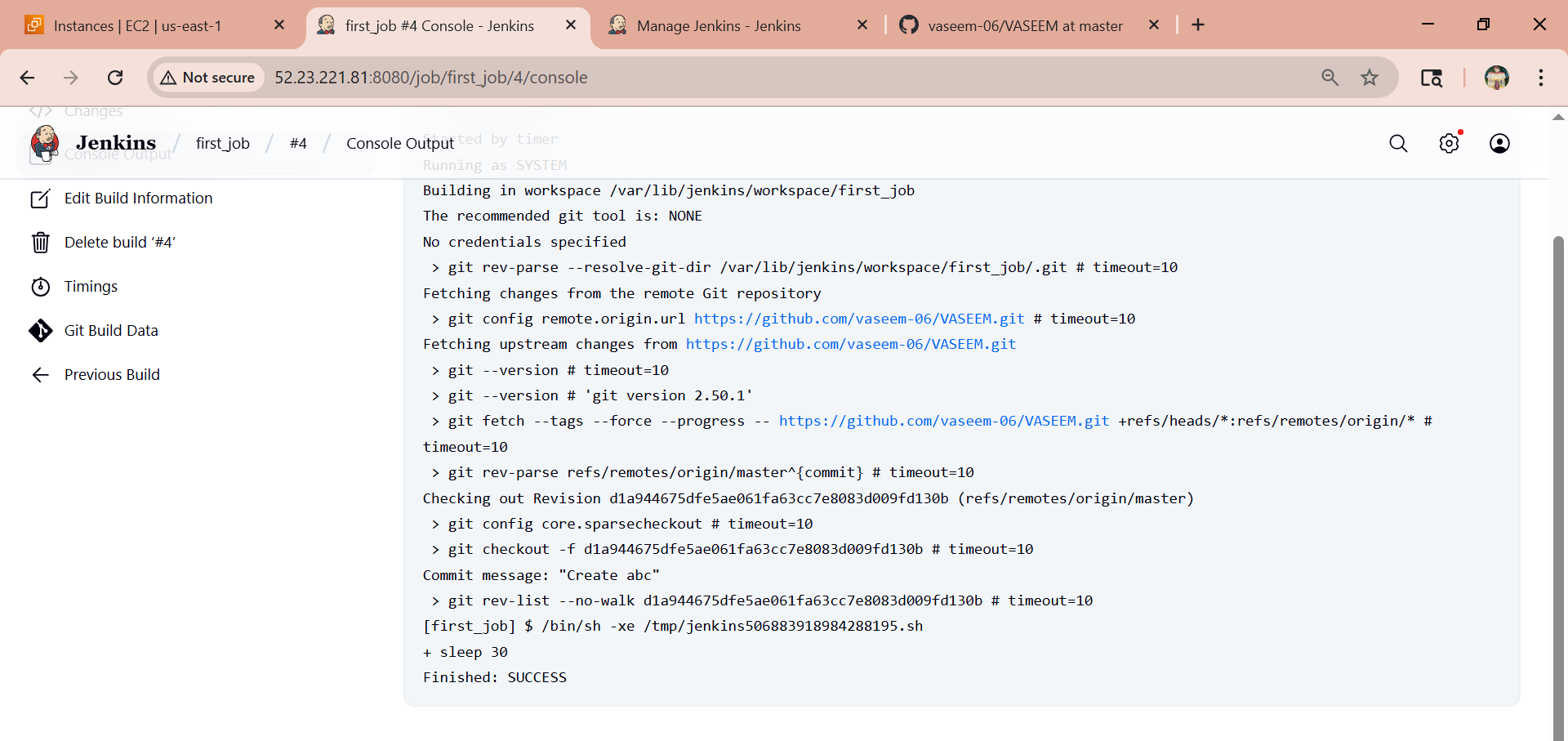
1. Here we have Poll scm
2. Go to Jenkins>first\_job>configurations>scroll down>trigger options it will there
3. I have gave stars means every minute, every hour, every day, every month



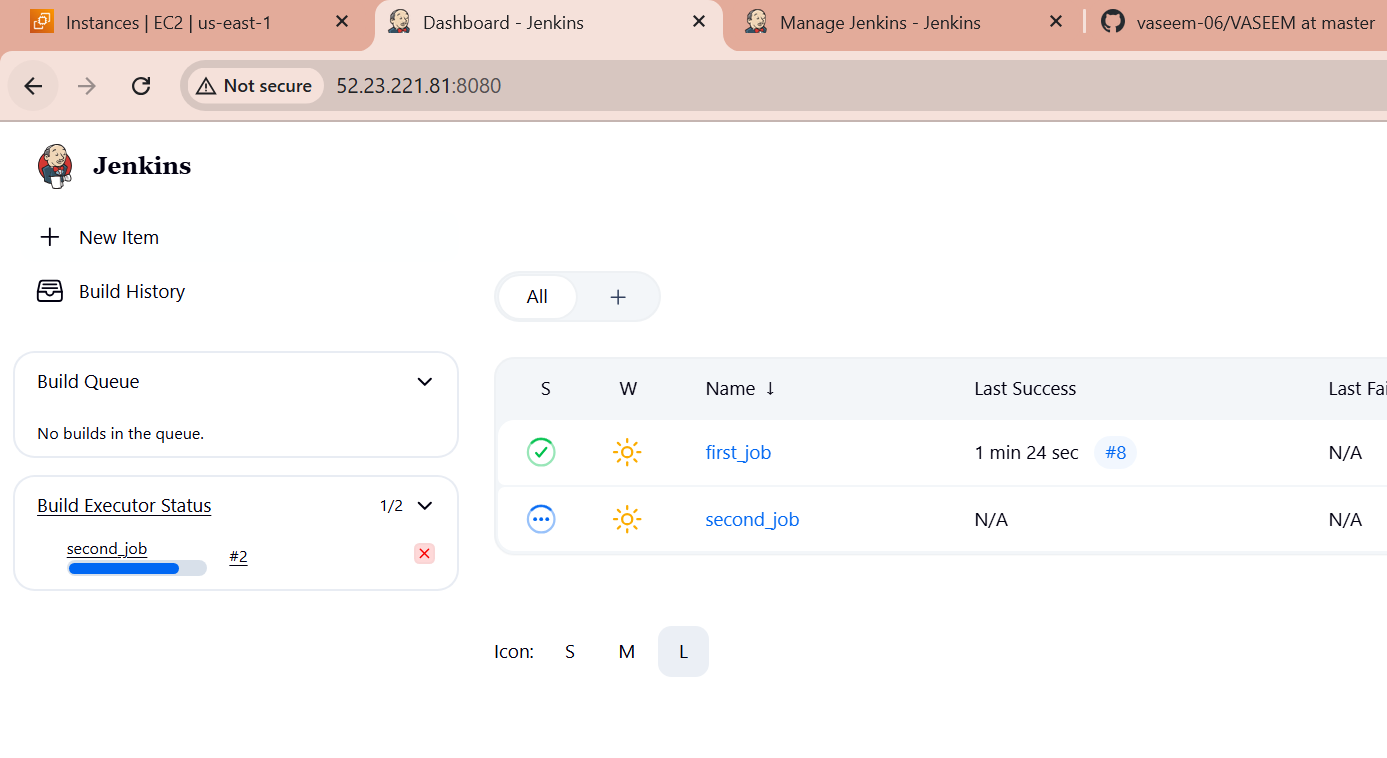


1. And we have another option is Build periodically
2. Go to Jenkins>first\_job>configurations>scroll down>trigger options it will there as Build periodically

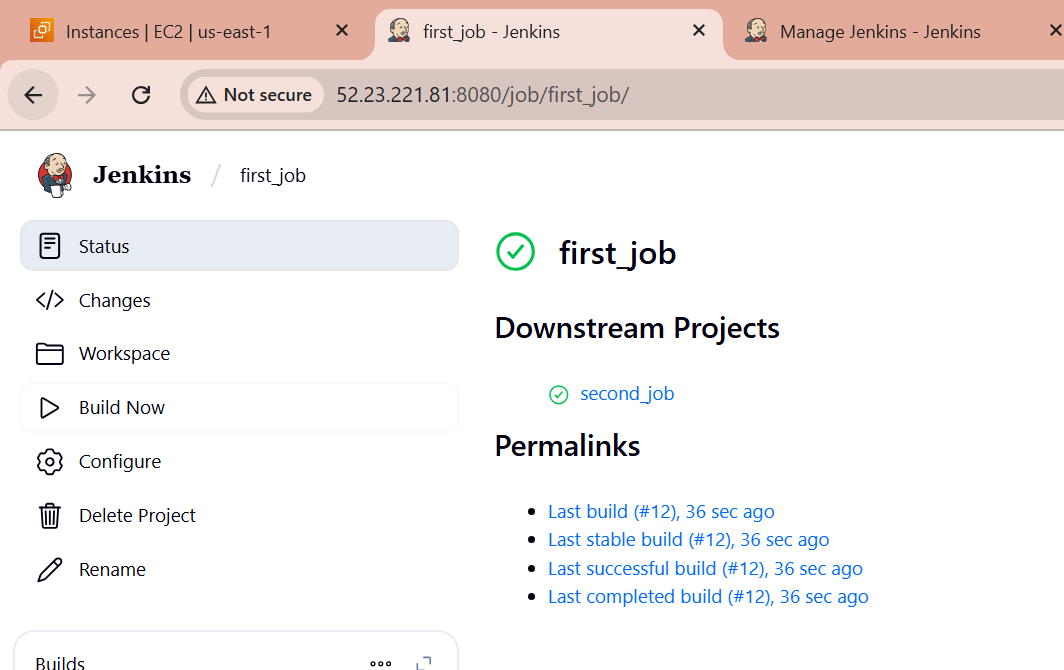




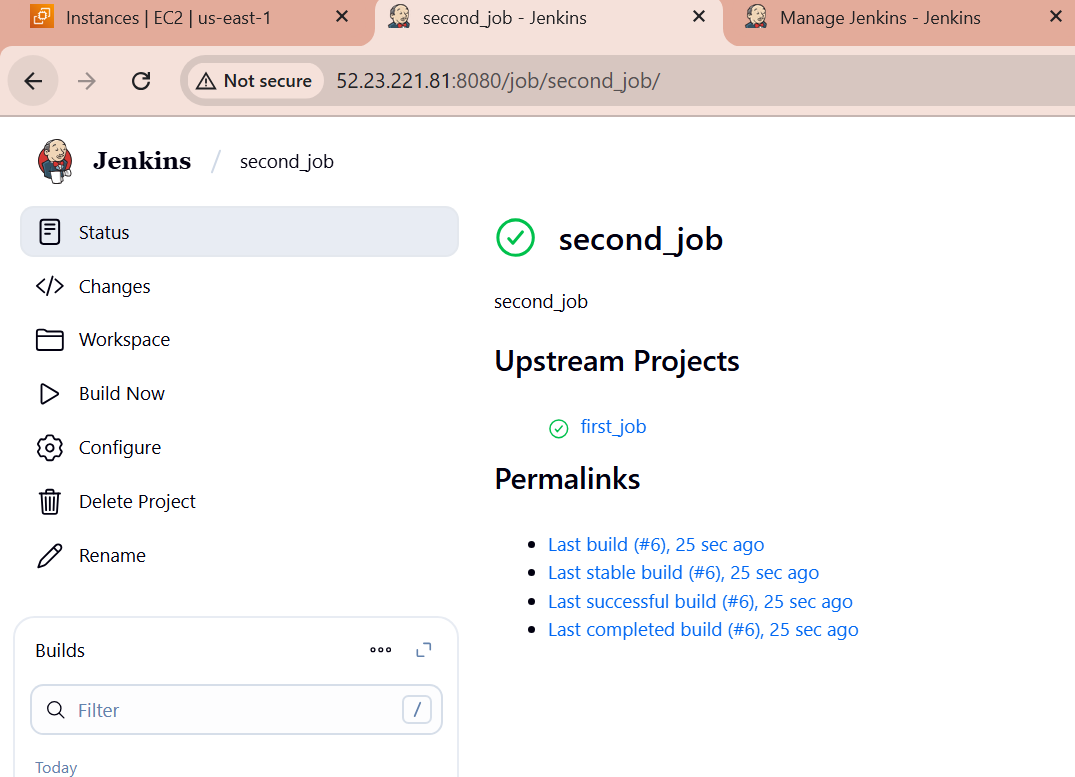
1. And here the second\_job it is get triggered after the first\_job is triggers



Here the downstream job is second and got triggered because I havent gave any much time for that

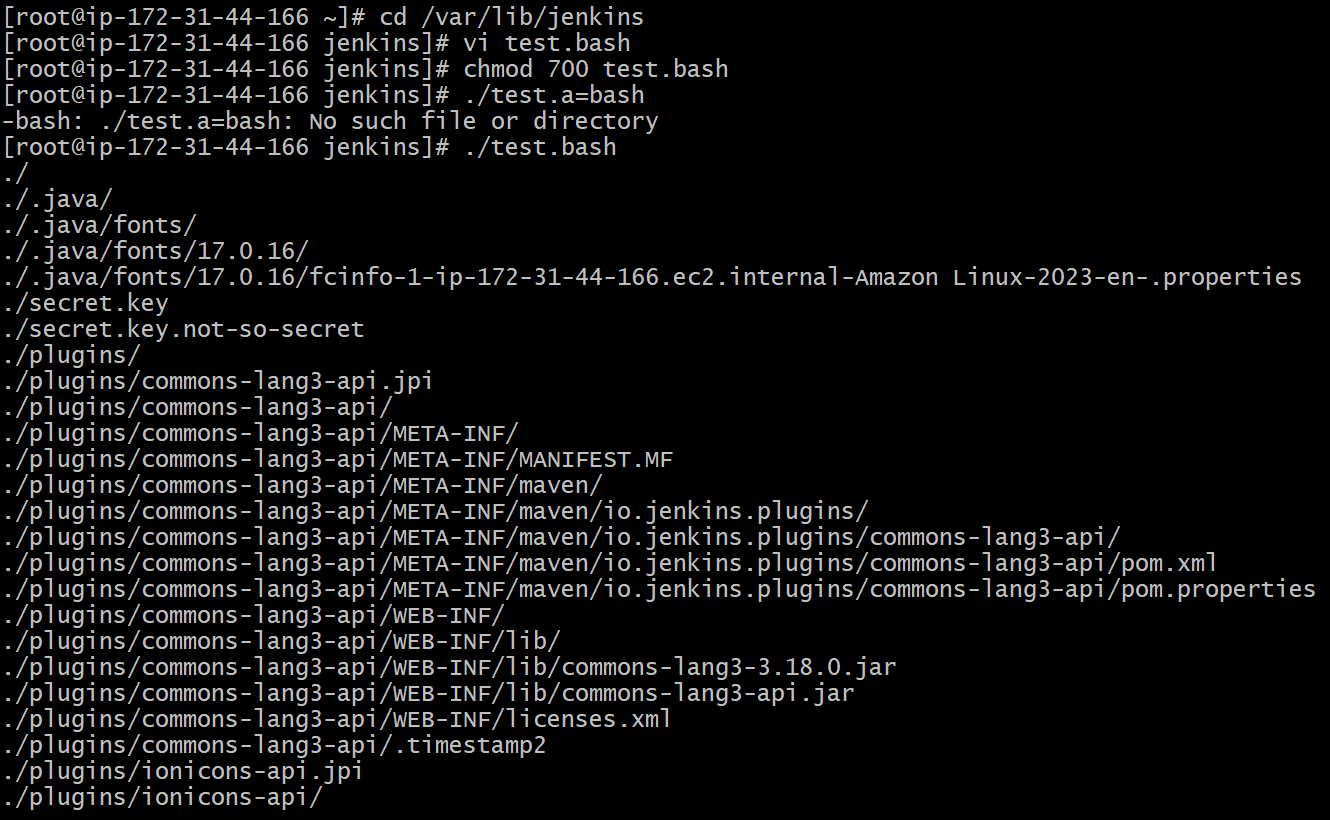


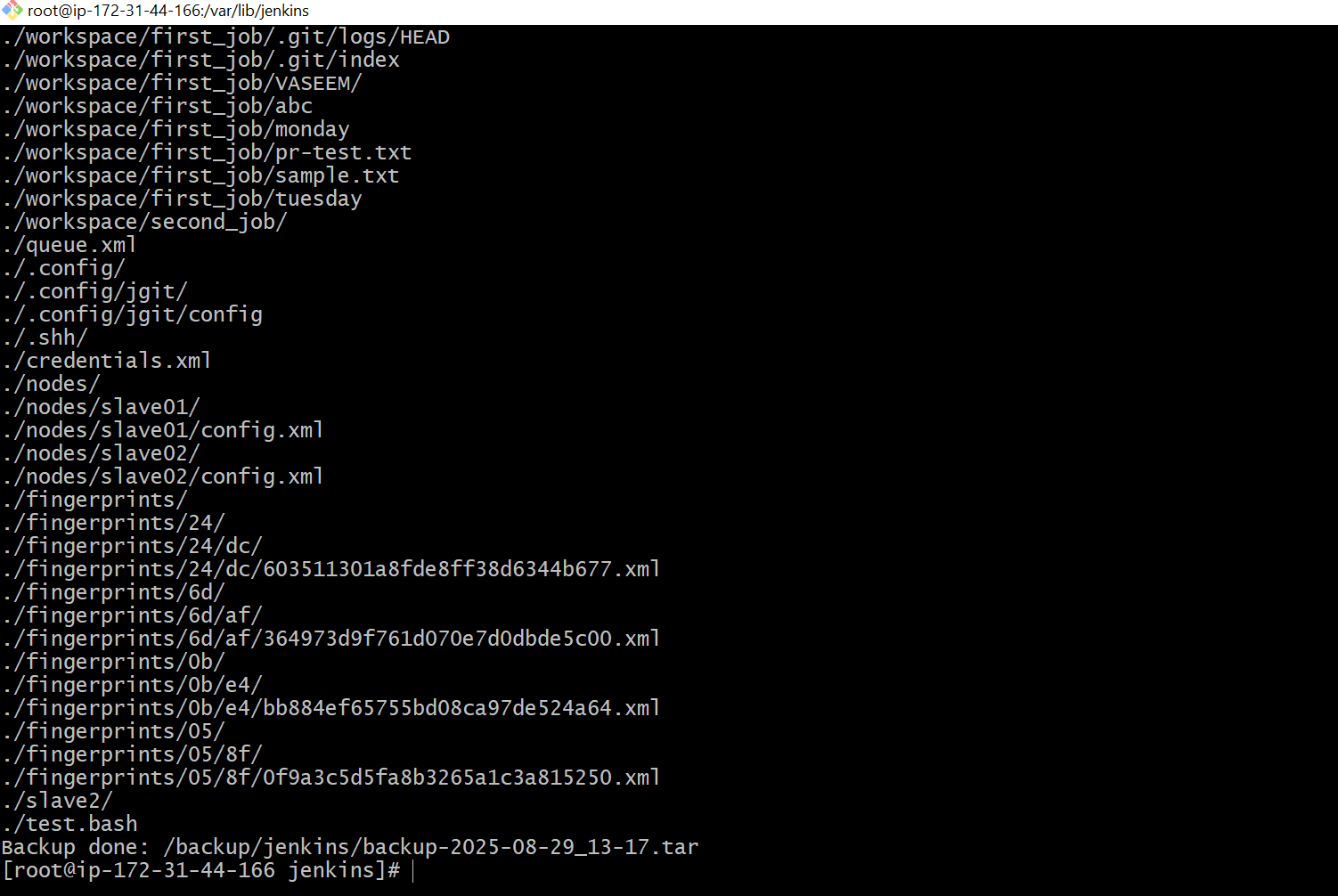
And upstream is first\_job



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

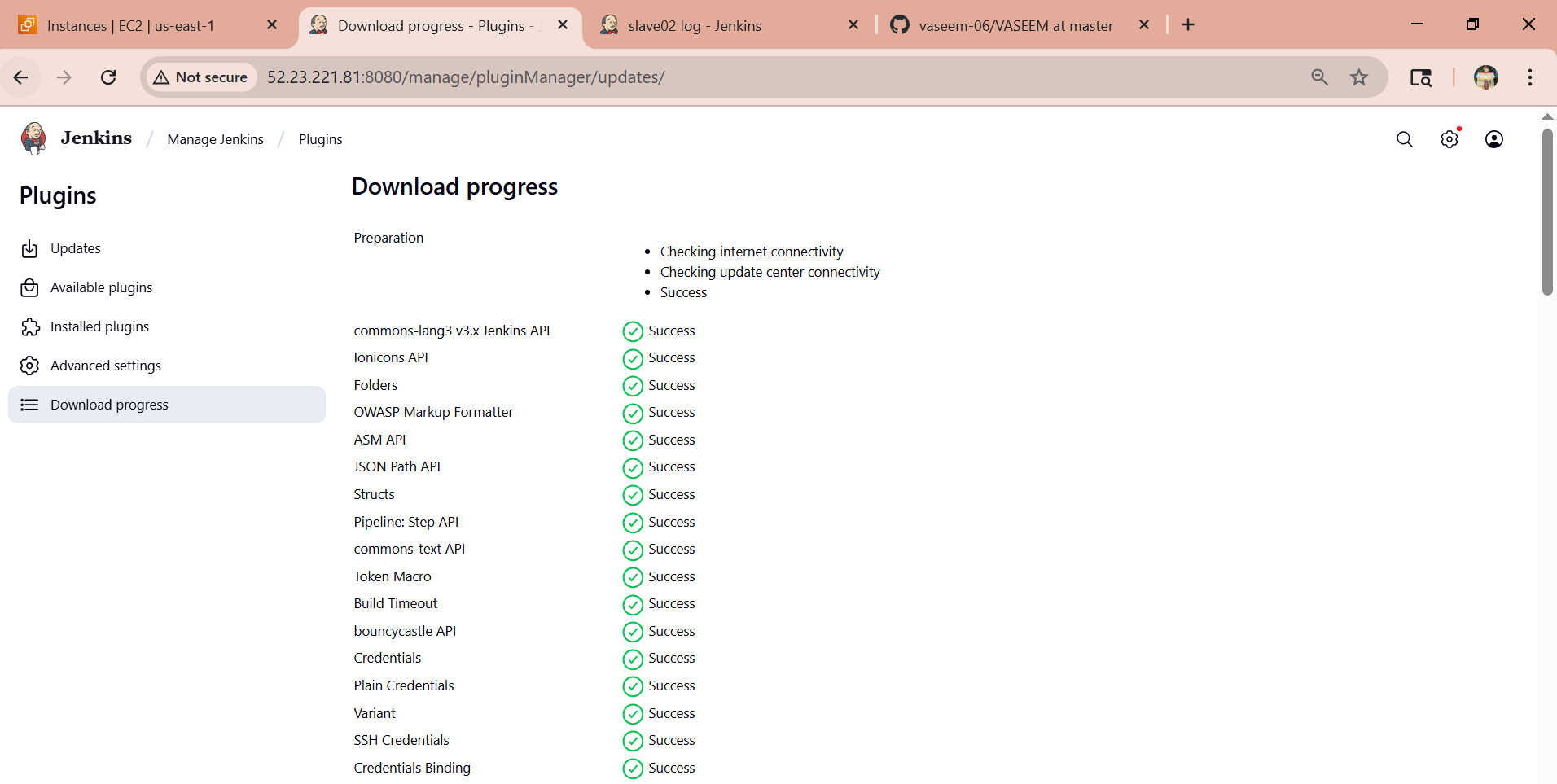
1. Jenkins configuration is stored in /var/lib/jenkins/
2. Create a script and give chmod permissions.  
   Vi test.bash  
   #!/bin/bash  
   set -euo pipefail # Fail on undefined vars, plus safer error handling  
   BACKUP\_DIR=${BACKUP\_DIR:-/backup/jenkins}  
   JENKINS\_HOME=${JENKINS\_HOME:-/var/lib/jenkins}  
   STAMP=$(date +%F\_%H-%M)  
   mkdir -p "$BACKUP\_DIR"  
   tar -cvf "$BACKUP\_DIR/backup-$STAMP.tar" -C "$JENKINS\_HOME" .  
   echo "Backup done: $BACKUP\_DIR/backup-$STAMP.tar"





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

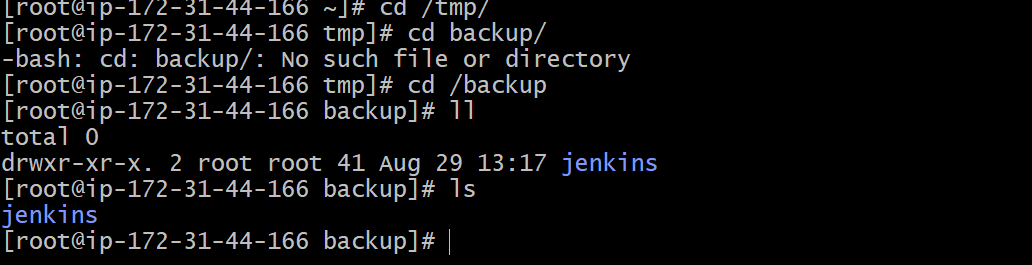
1. Go to jenkins>manage jenkins>Plugins>available plugins>search as ThinBackup>install



Here the backup was in cd /tmp/

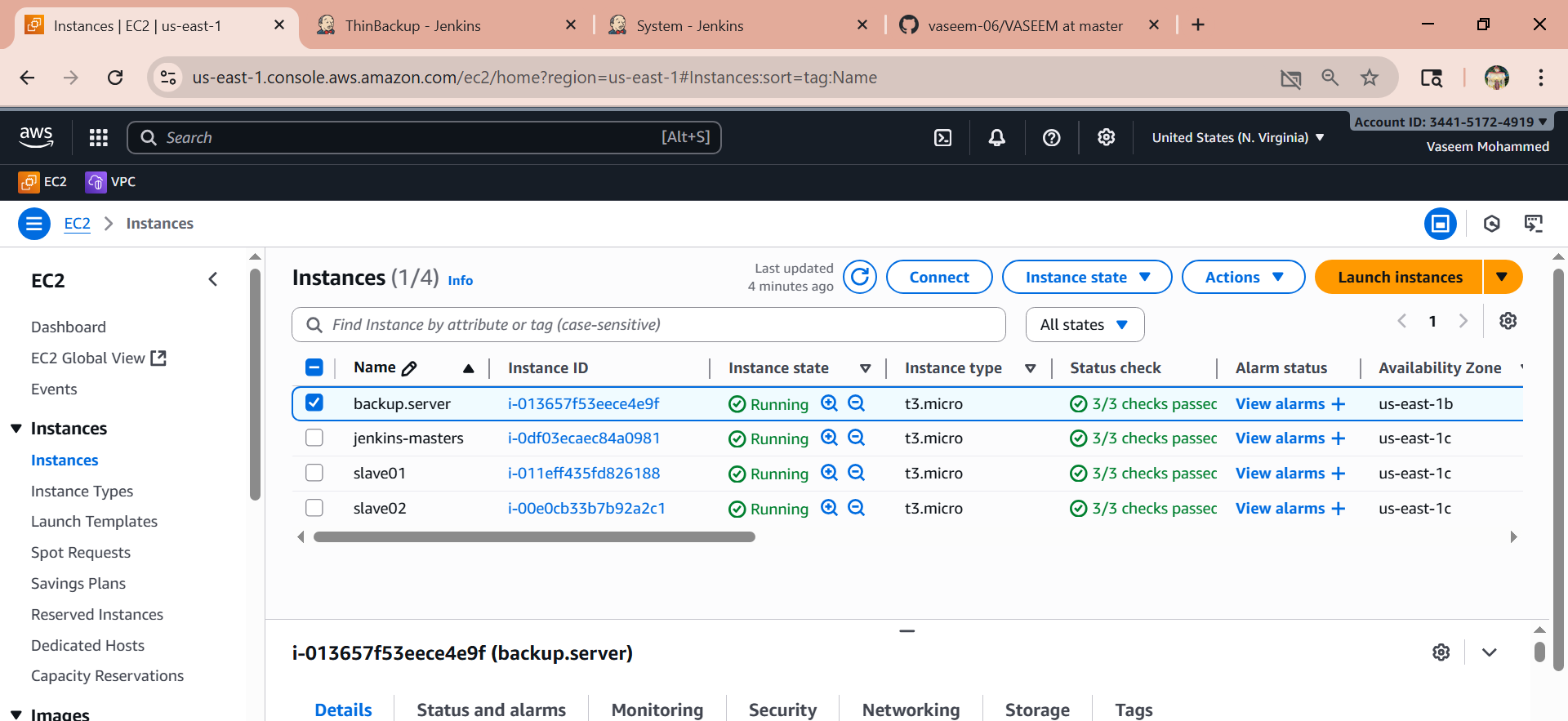
Cd /backup

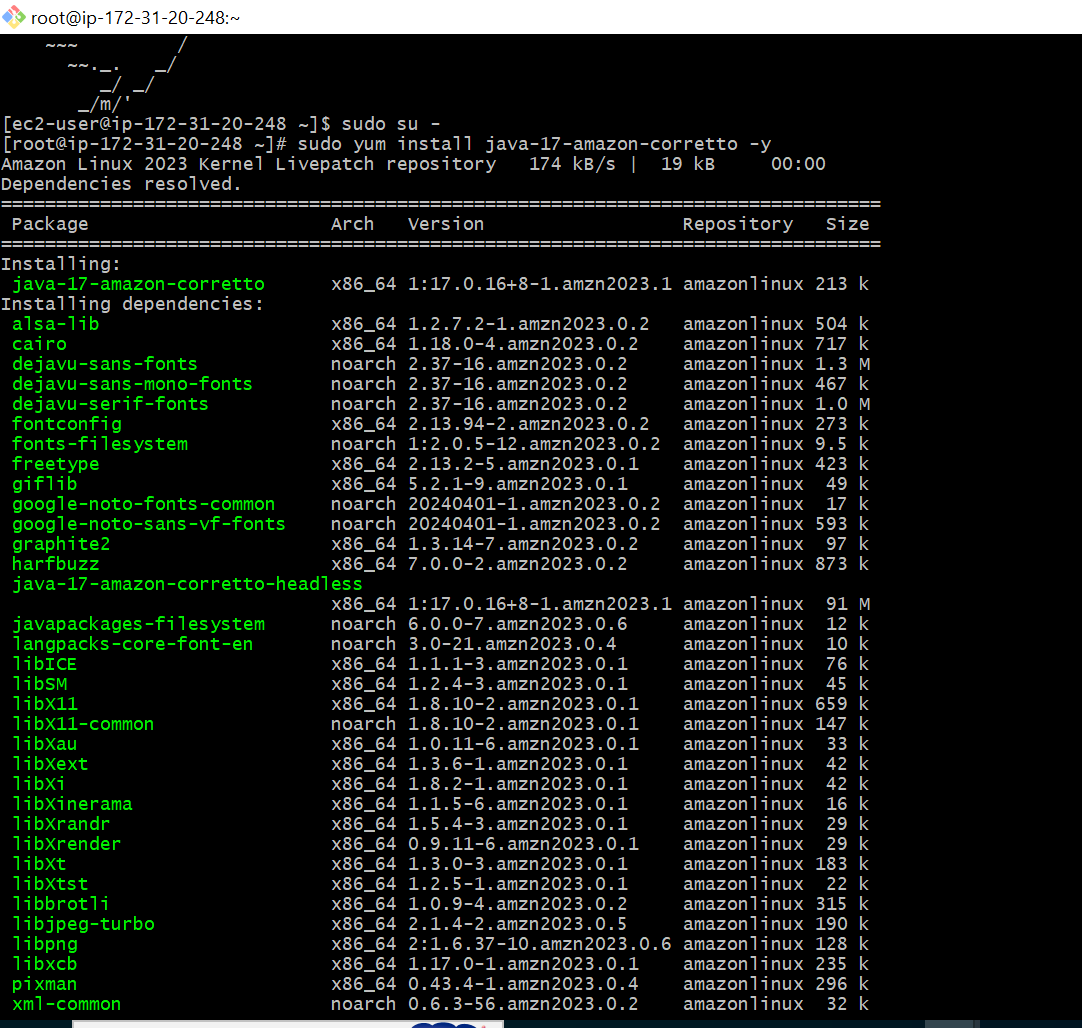
ls

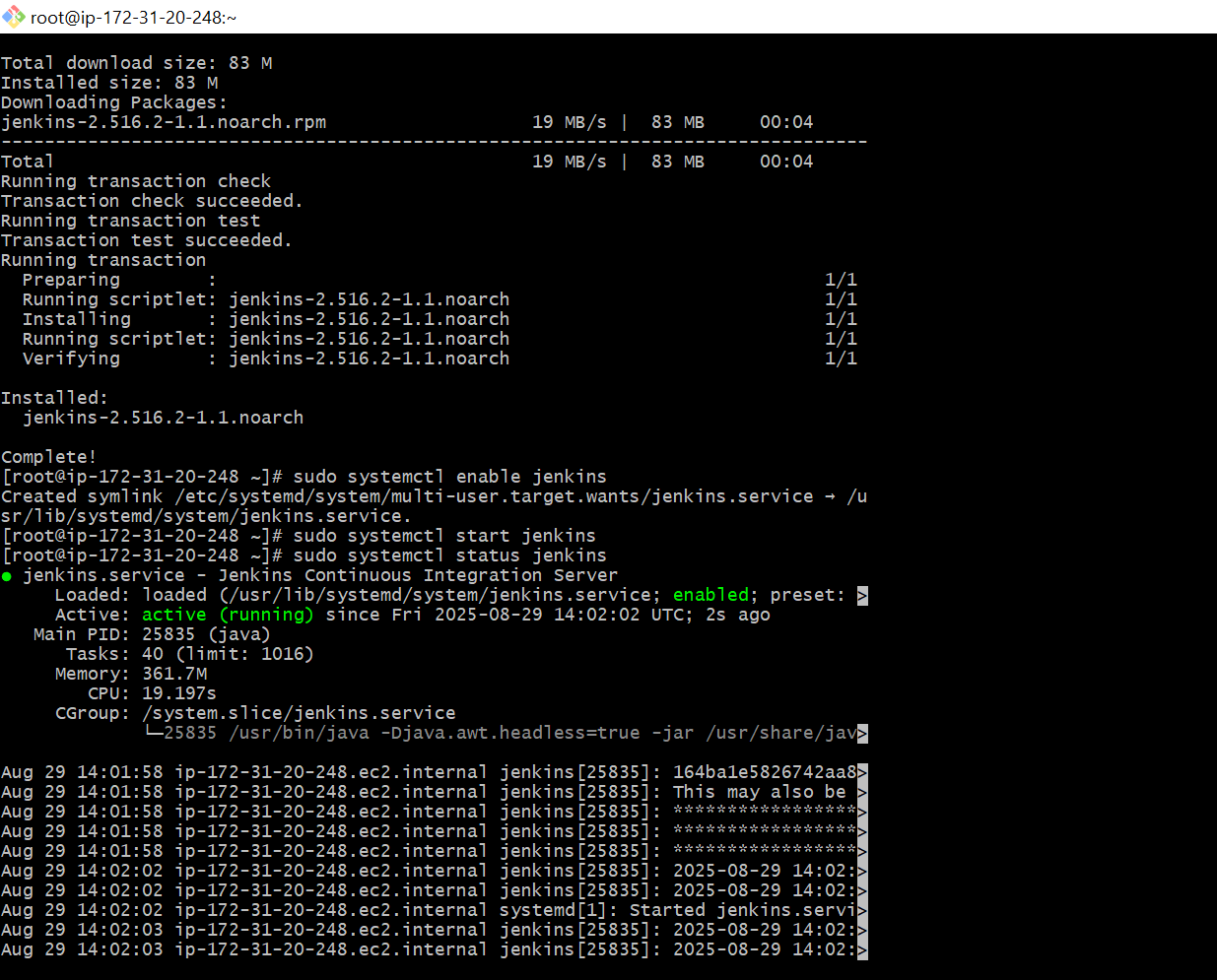


**6) Setup a new Jenkins server and dump the backup taken in task4.**

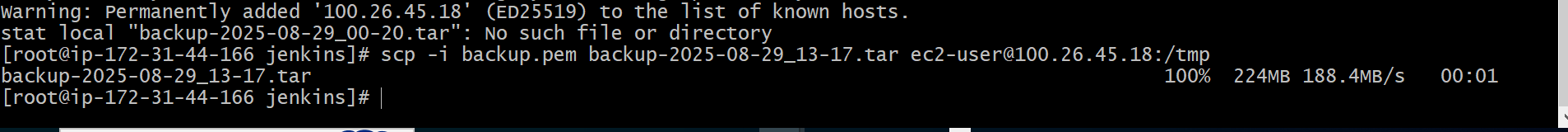
1. Create or launch one instances backup-master ec2  
   sudo yum install java-17-amazon-corretto -y  
   sudo wget -O /etc/yum.repos.d/jenkins.repo \   
   https://pkg.jenkins.io/redhat-stable/jenkins.repo  
   sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key  
   sudo yum install jenkins -y  
   sudo systemctl enable jenkins  
   sudo systemctl start jenkins  
   sudo systemctl status jenkins

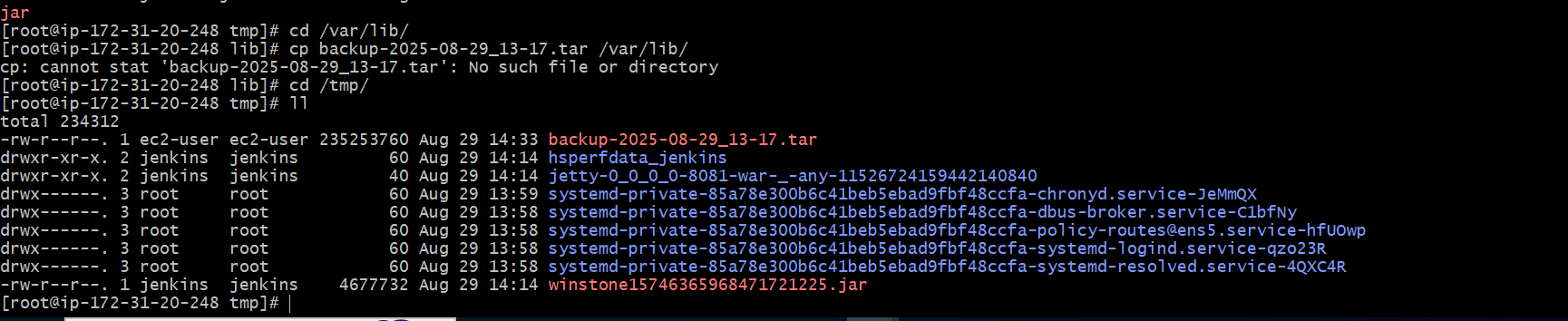






Go to master Jenkins terminal  
[root@ip-172-31-93-42 jenkins]# vi backup.pem  
[root@ip-172-31-93-42 jenkins]# chmod 600 backup.pem  
[root@ip-172-31-93-42 jenkins]# scp -i backup.pem backup-2025-08-29\_00-20.tar ec2-  
user@13.221.184.37:/tmp



Go to backup Jenkins terminal  
[root@ip-172-31-87-43 ~]# cd /tmp/  
[root@ip-172-31-87-43 tmp]# cd /var/lib/  
[root@ip-172-31-87-43 tmp]# cp backup-2025-08-29\_00-20.tar /var/lib/  
[root@ip-172-31-87-43 lib]# tar xvf backup-2025-08-29\_00-20.tar  
[root@ip-172-31-87-43 lib]# cd jenkins  
[root@ip-172-31-87-43 jenkins]# ll  


Than we need to give command

Cp backup.tar /var/lib/  
Cd /var/lib  
ls  
tar xvf backup.tar  
then restart the Jenkins

<http://100.26.45.18:8081/restart>

Im running on 8081, because already 8080 is running