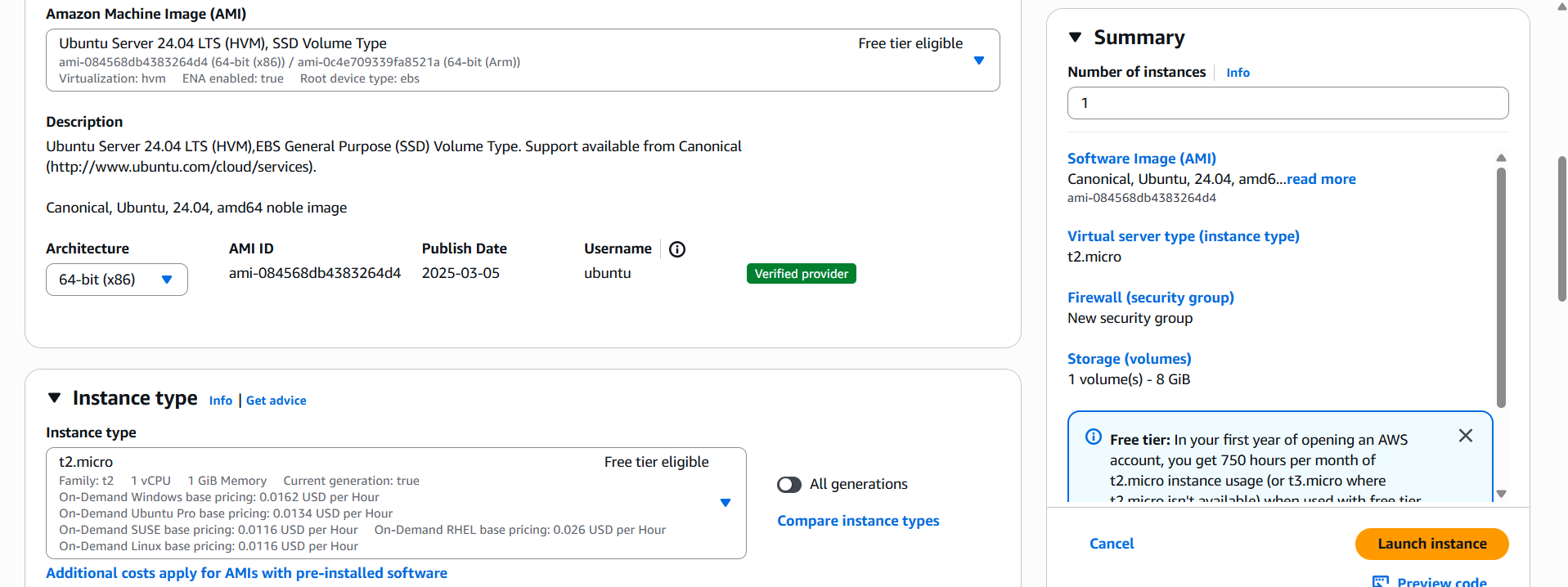
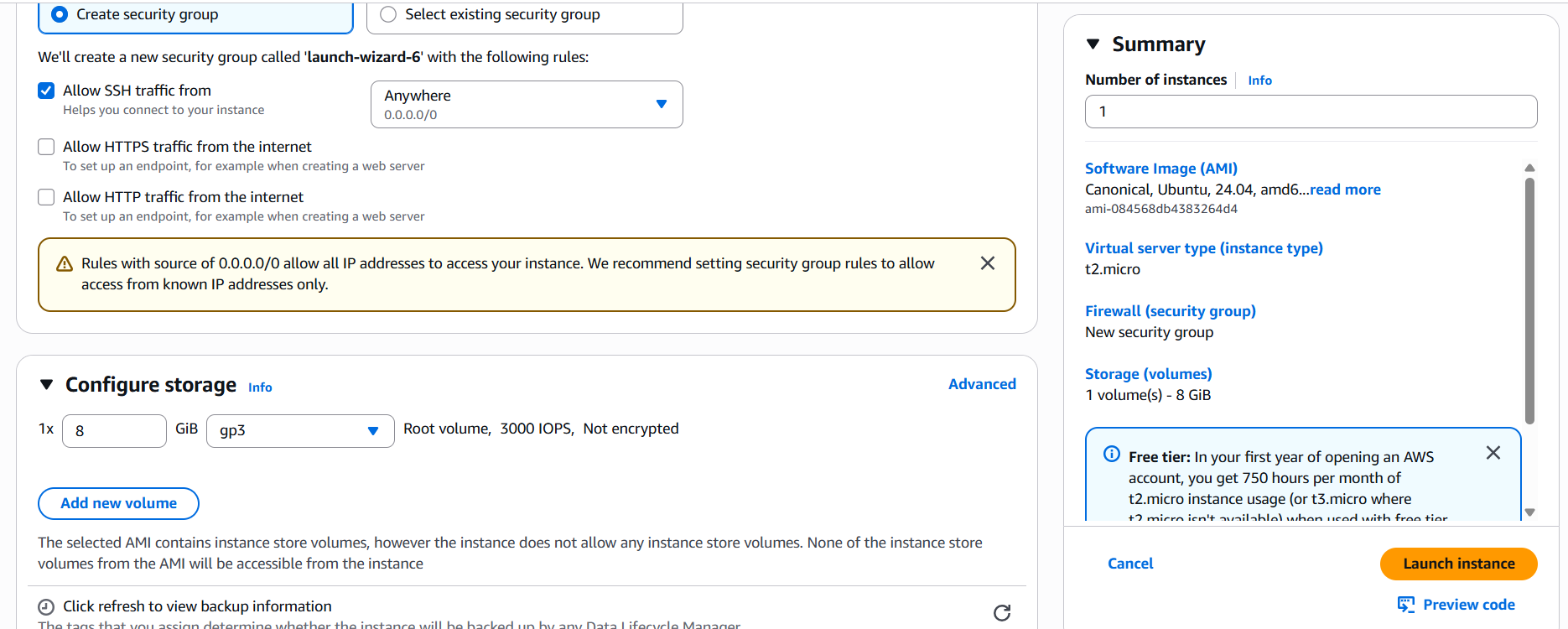
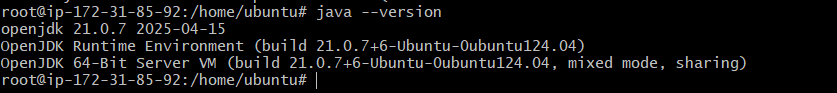
1. Configure 2 slave machines in Jenkins master.

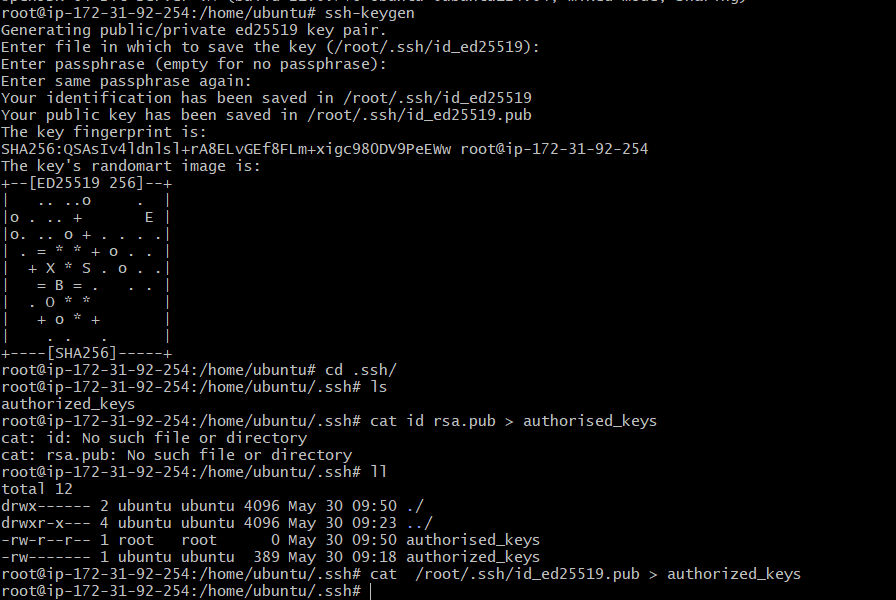
🡺steps to be done on slave  
🡺Login to slave machine..  
🡺switch to root user  
🡺install jdk 11 (amazon-linux-extras install java-openjdk11)  
🡺Create ssh-keygen  
🡺cat id\_rsa.pub > authorized\_keys  
🡺chmod 700 authorized\_keys  
🡺Steps to be done on master machine:  
🡺Login to master machine  
🡺switch 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 172.31.85.92>>/var/lib/jenkins/.ssh/known\_hosts  
🡺chown jenkins:jenkins known\_hosts  
🡺#we need to change the owner as we ran ssh-keyscan command   
🡺using “root” user.  
🡺# default user of Jenkins will be “jenkins”  
🡺chmod 700 known\_hosts  
# DOne !!  
🡺Steps to be done on jenkins GUI:  
🡺Go to manage jenkins  
🡺open manage node  
🡺under launch method select "Launch via SSh"  
Host "public ip of slave"  
🡺creditals: select user with private key  
🡺enter ubuntu as name and paste the pem key.  
host key verification startegy to be selected  
save..  
we are good with the master and slave configuration

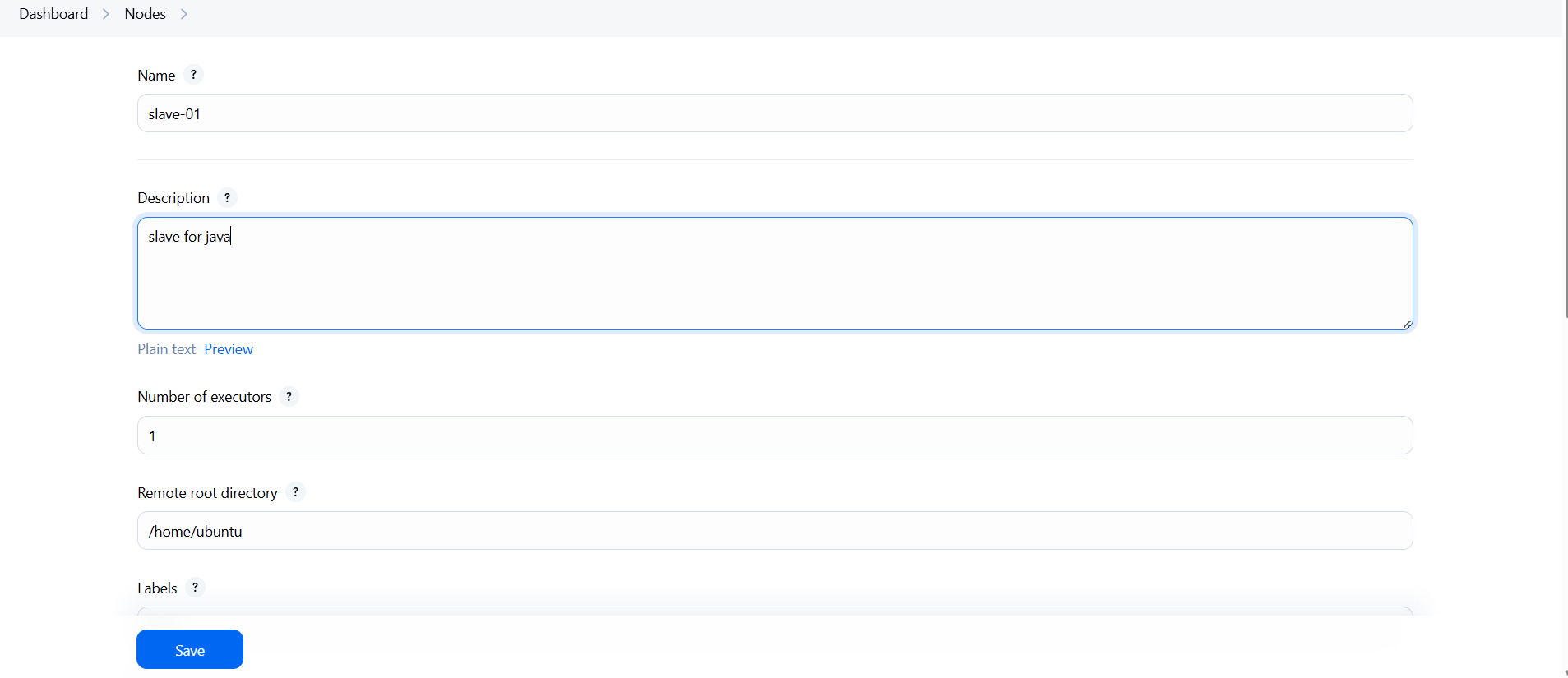


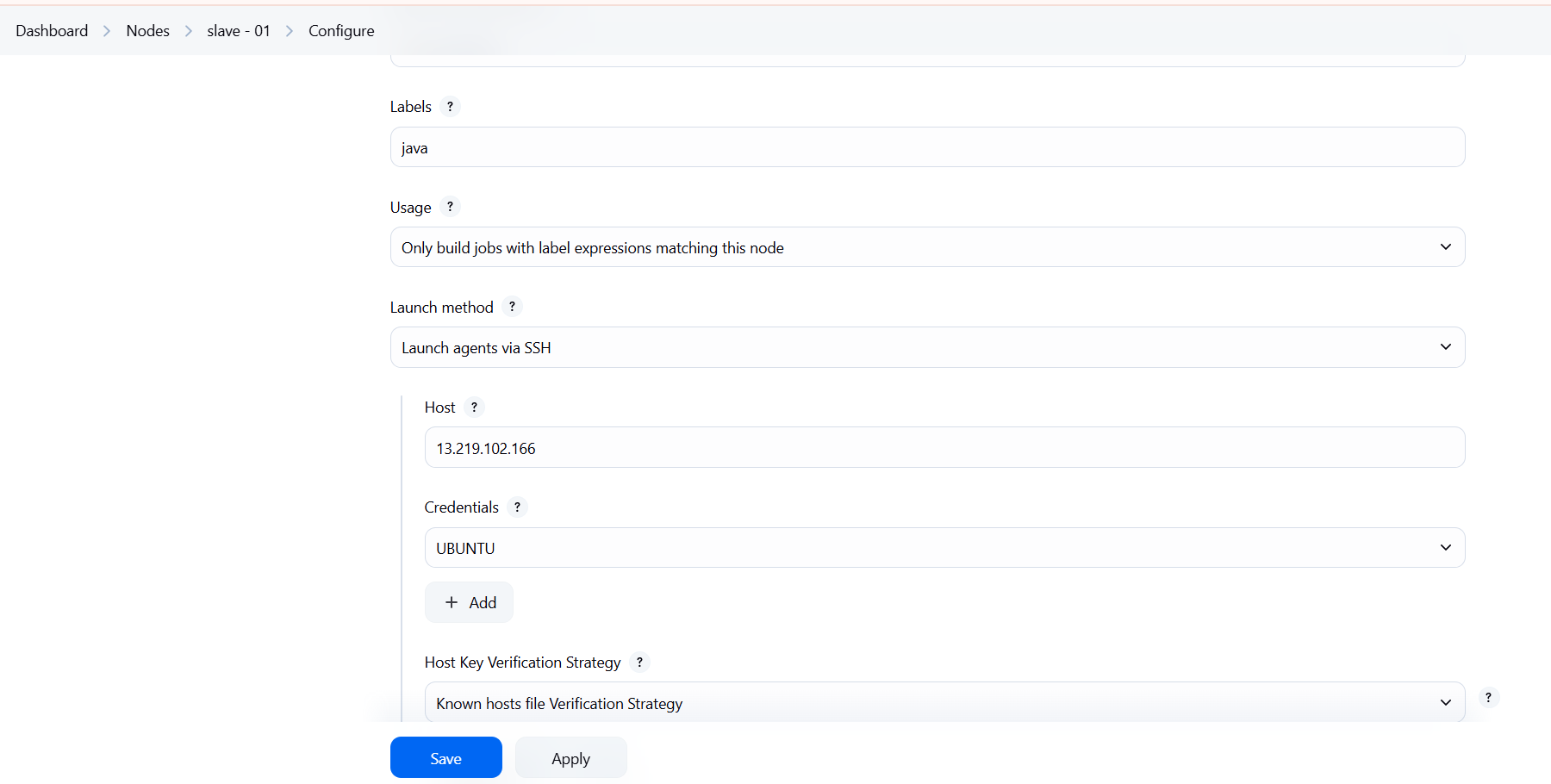




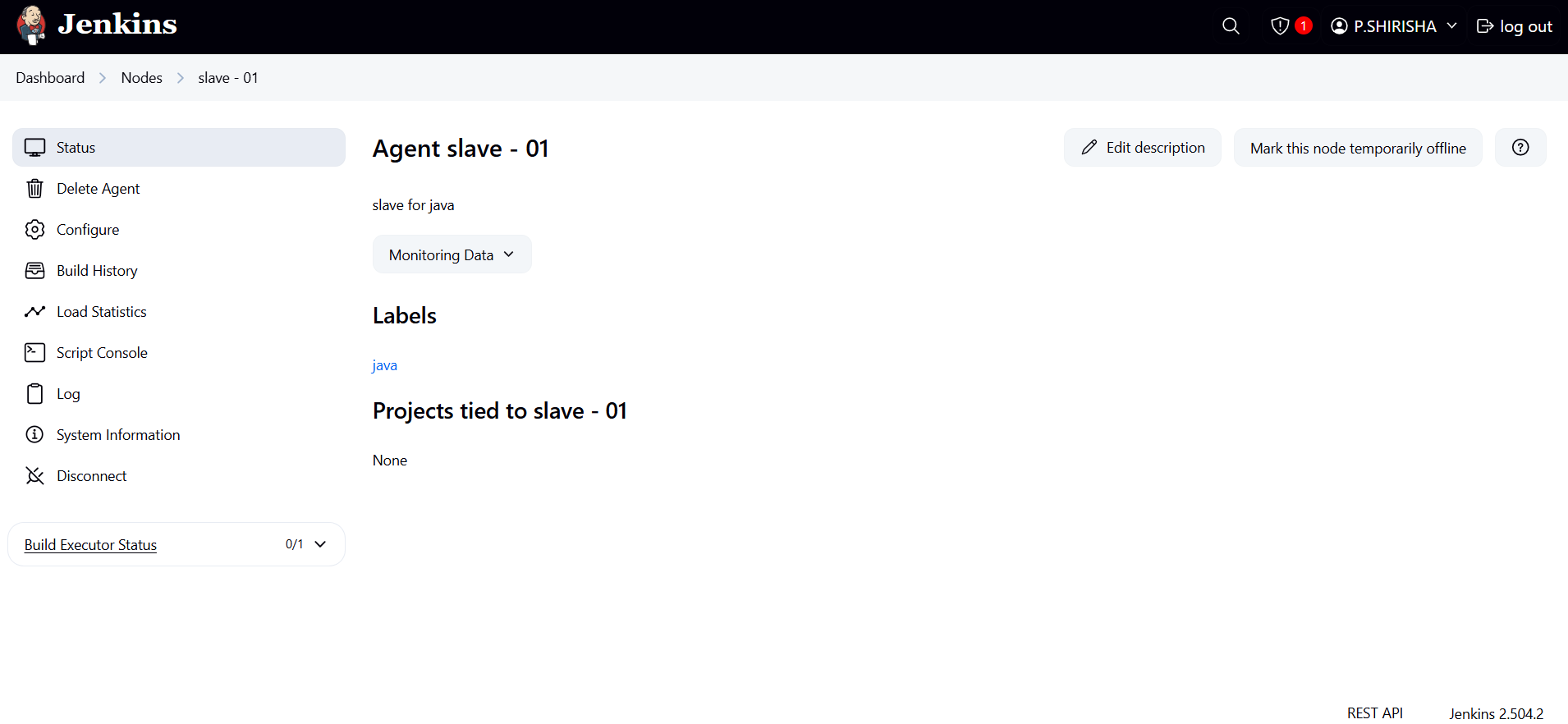


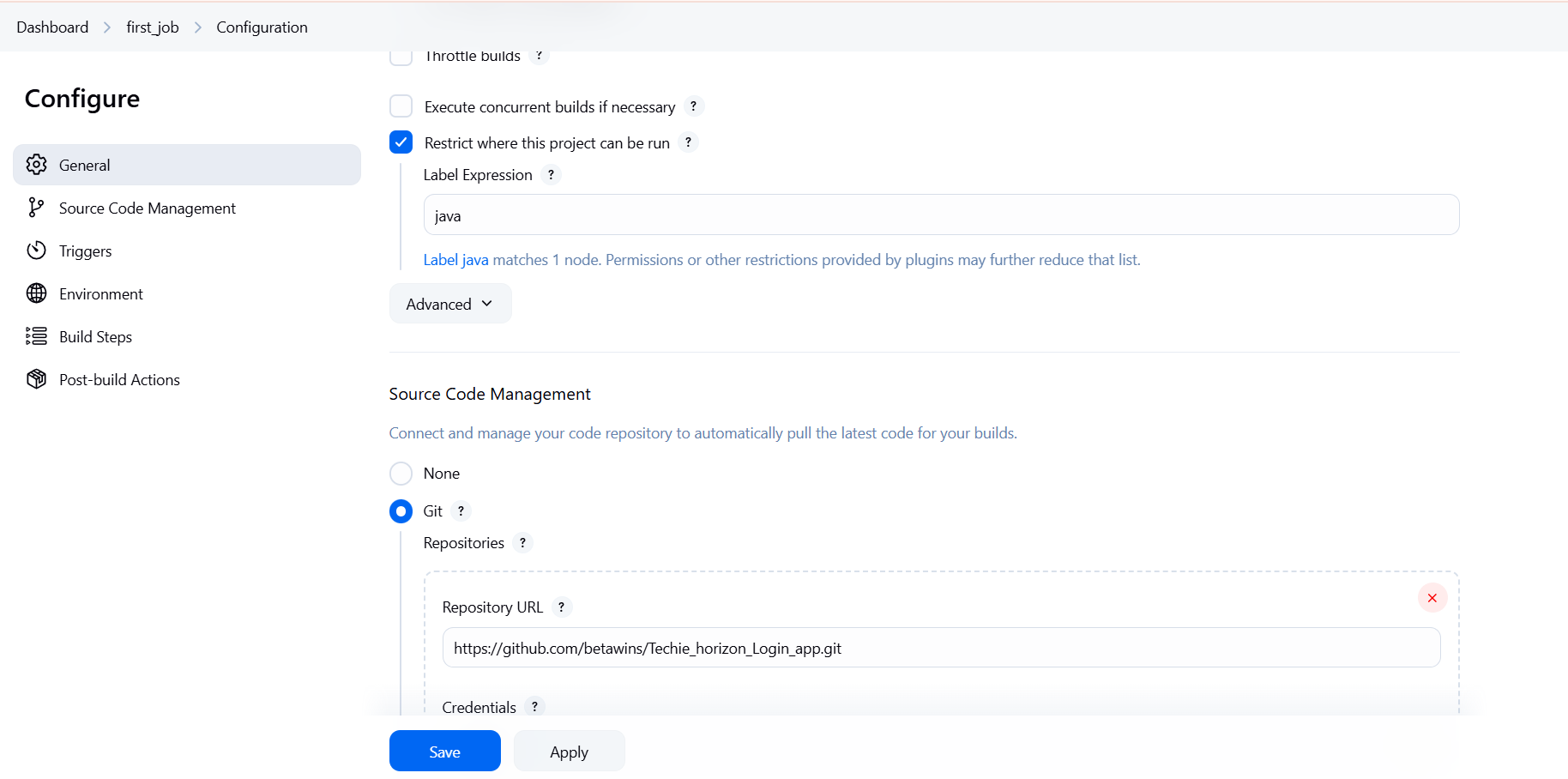


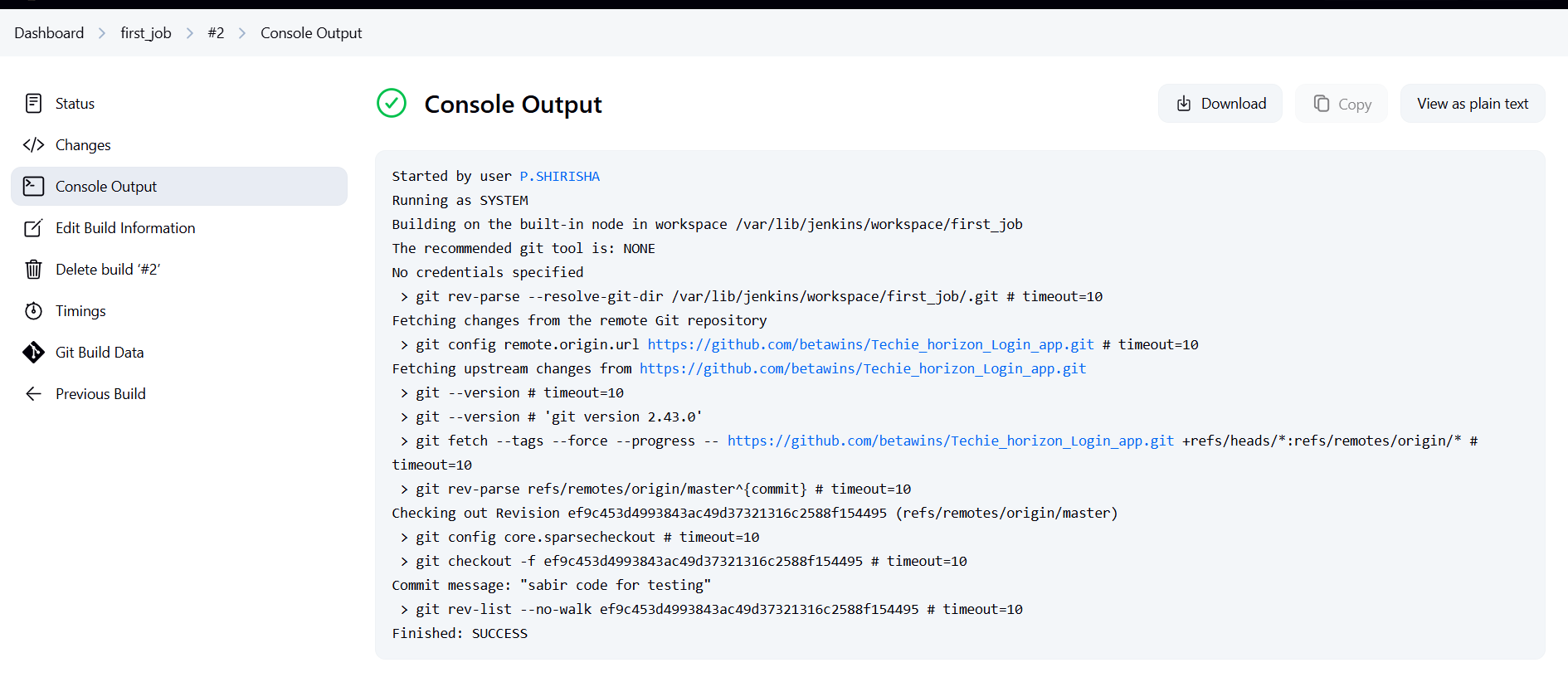


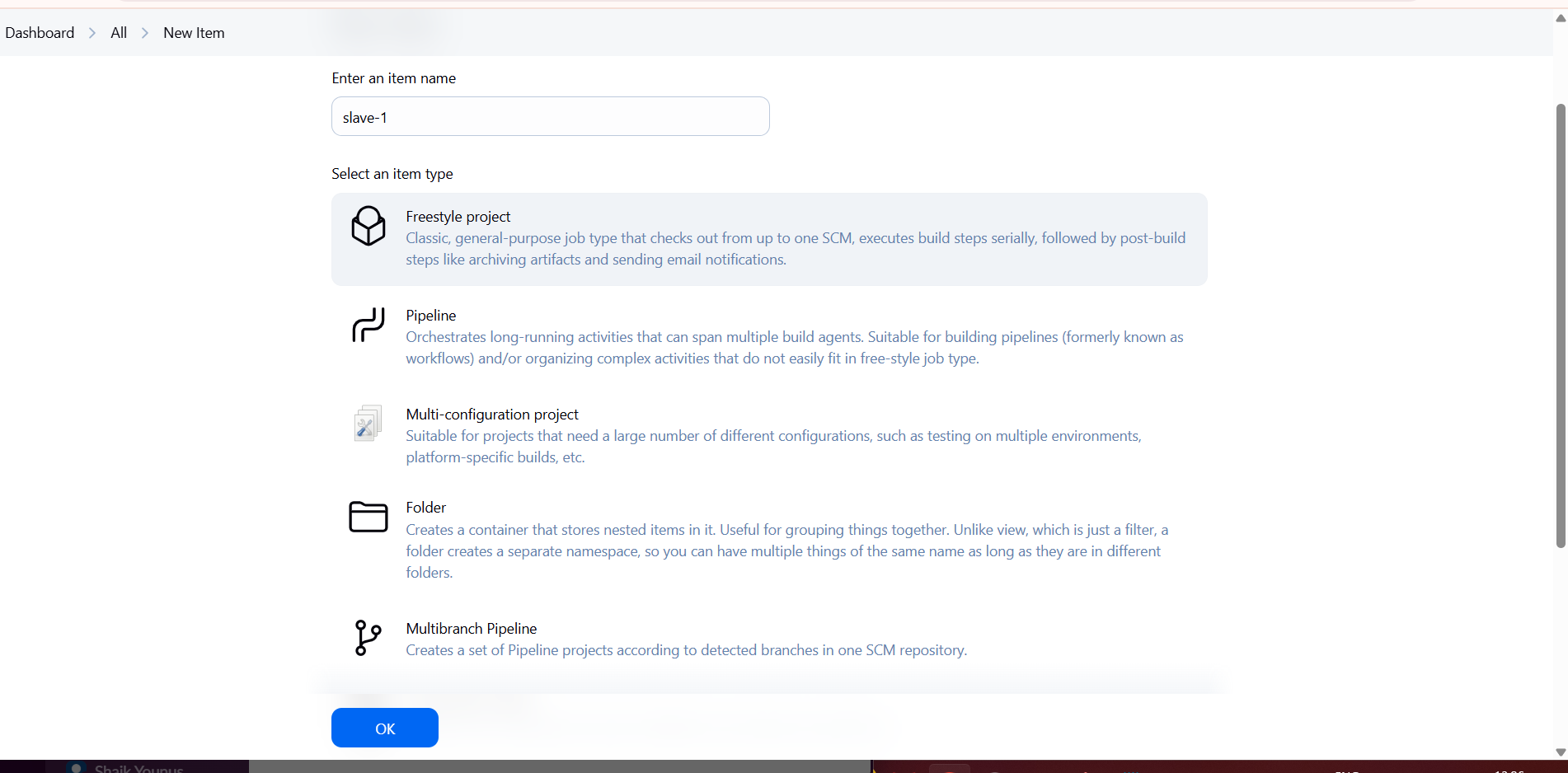


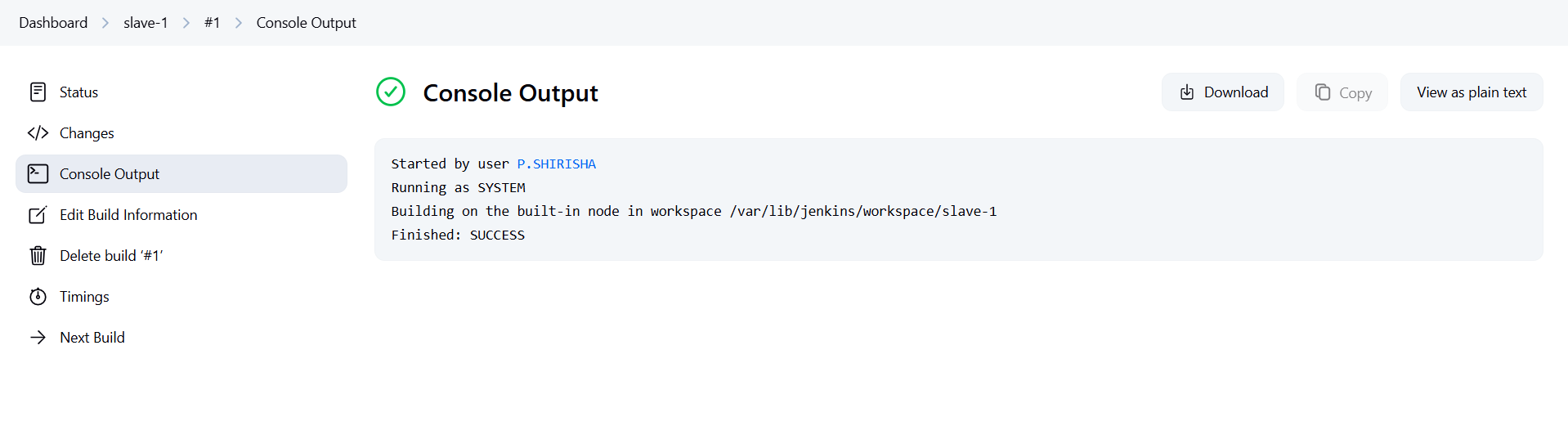


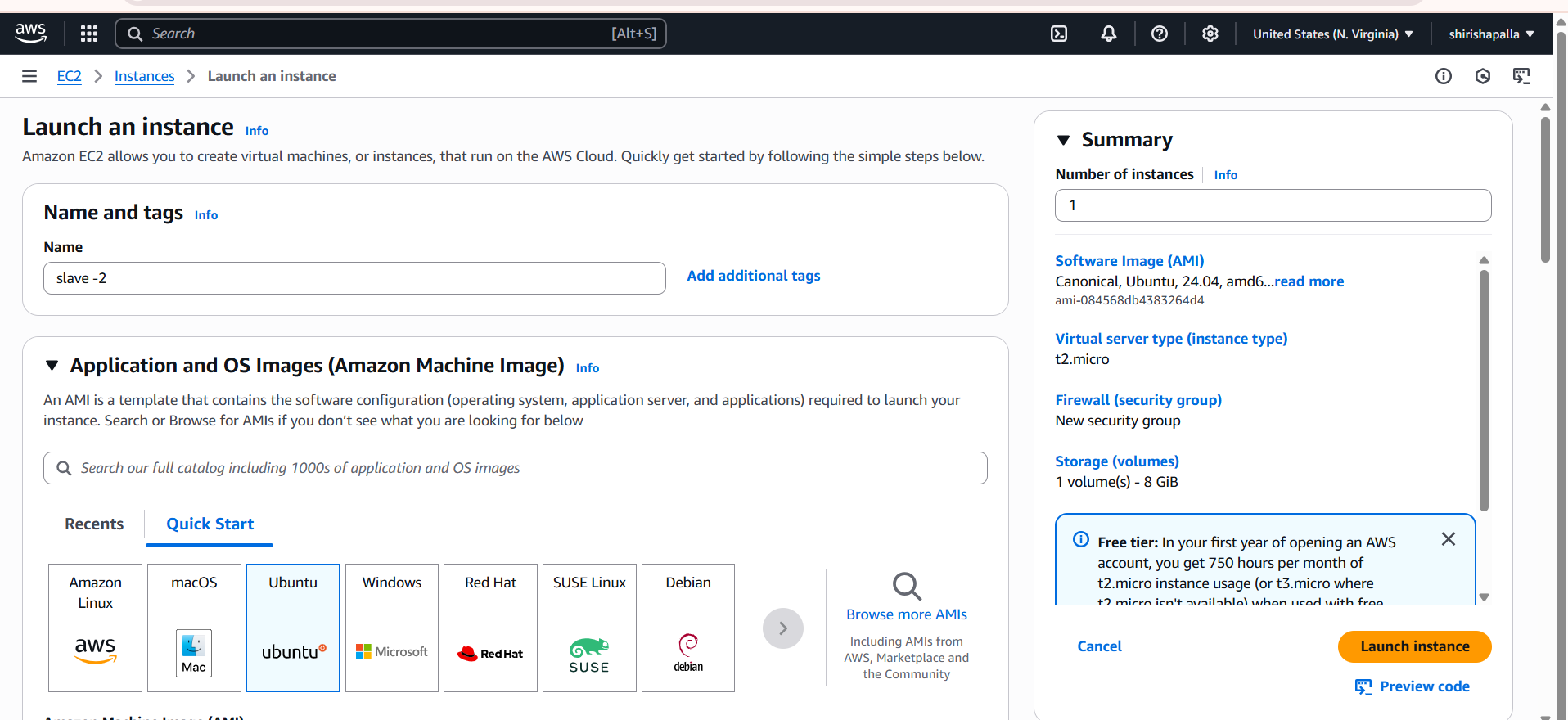


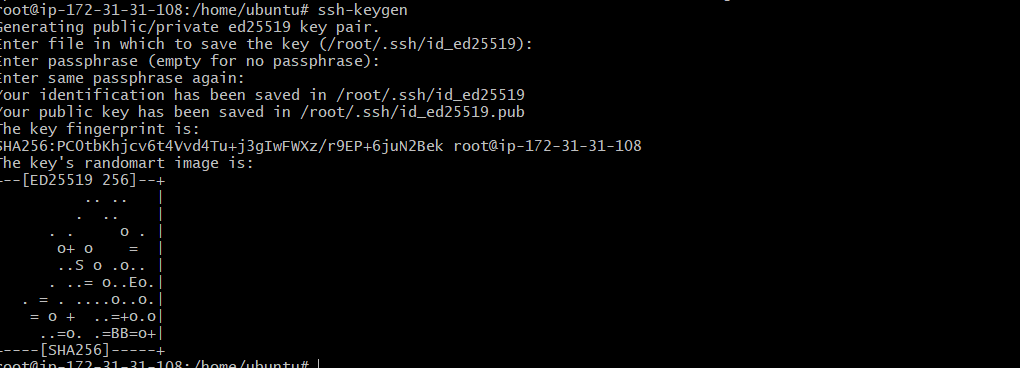


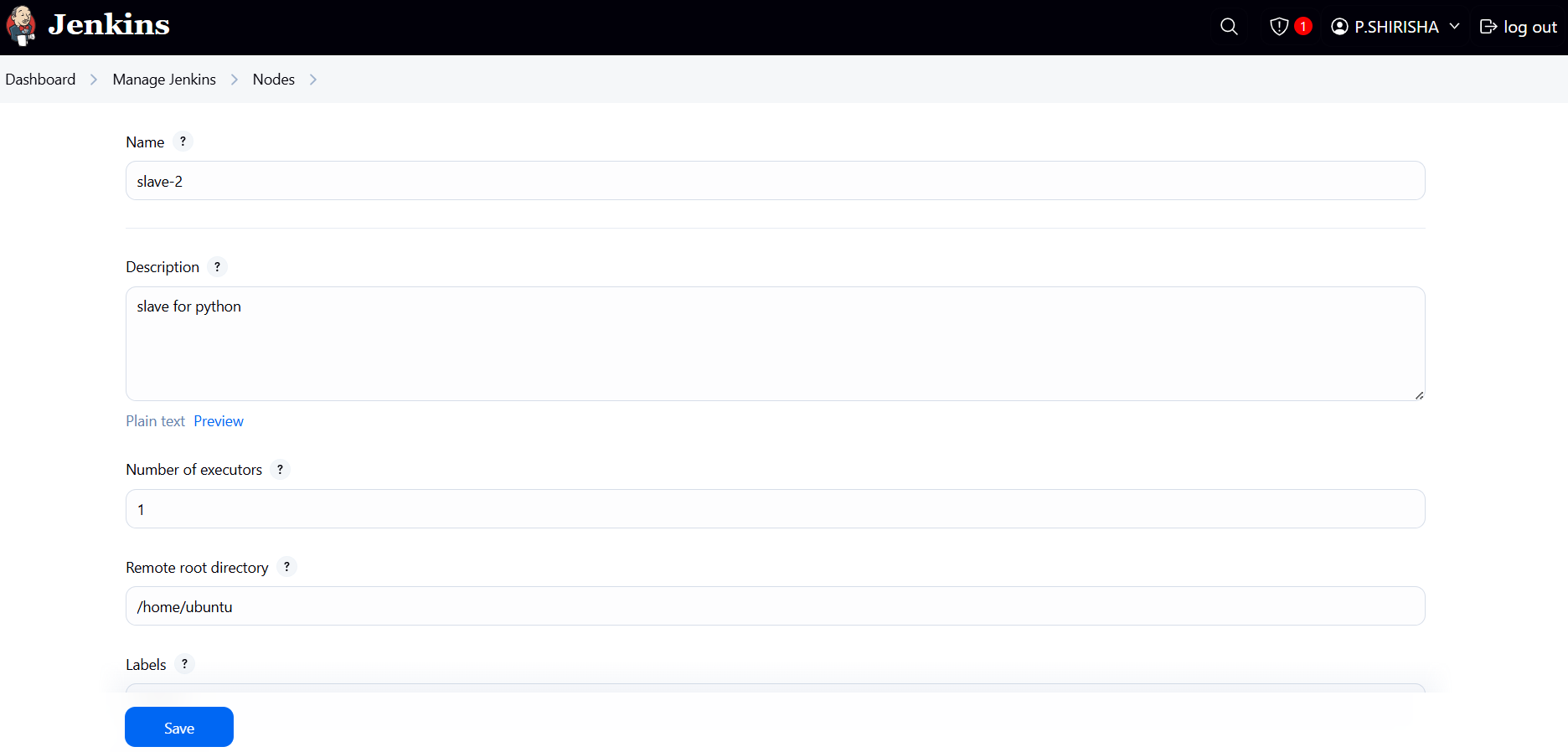


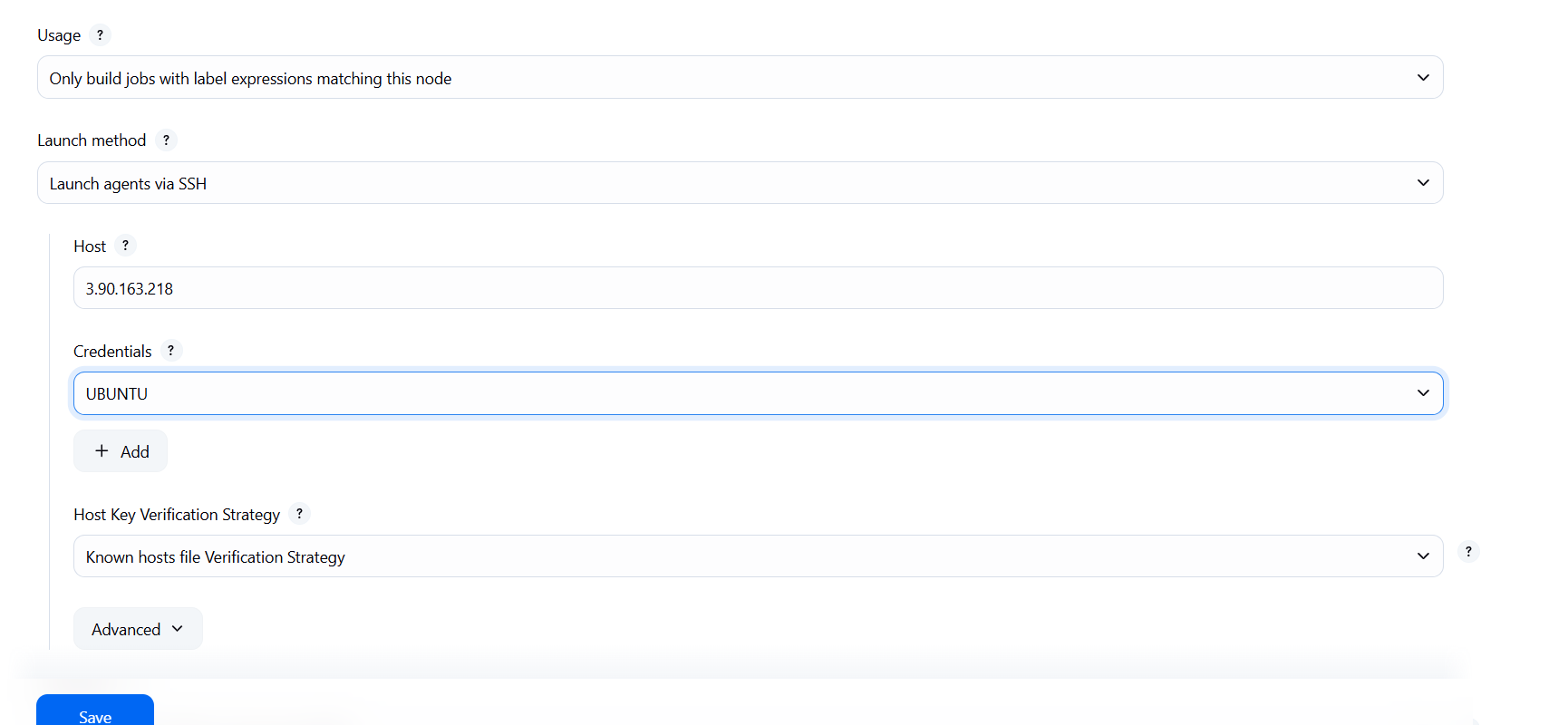








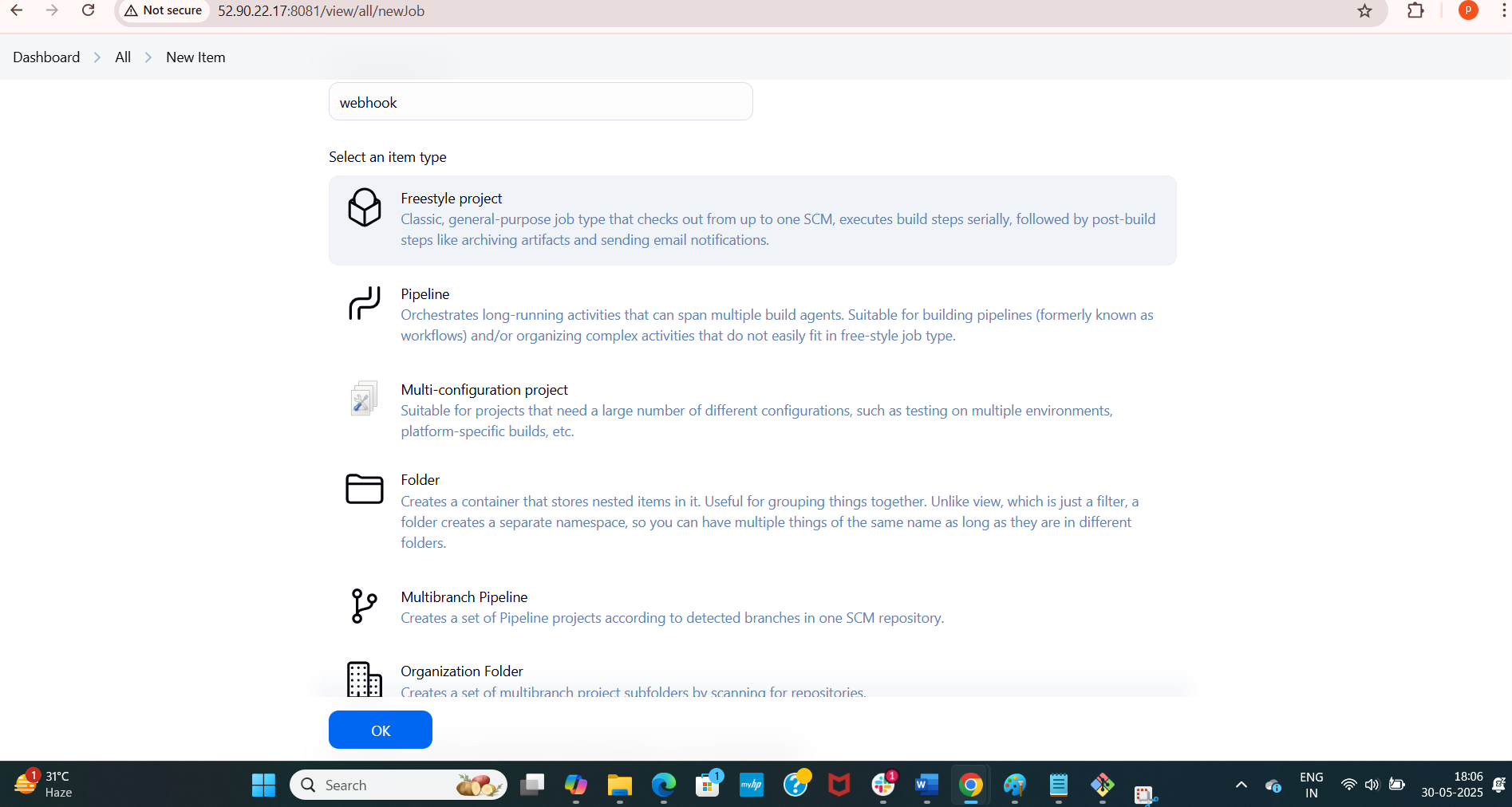




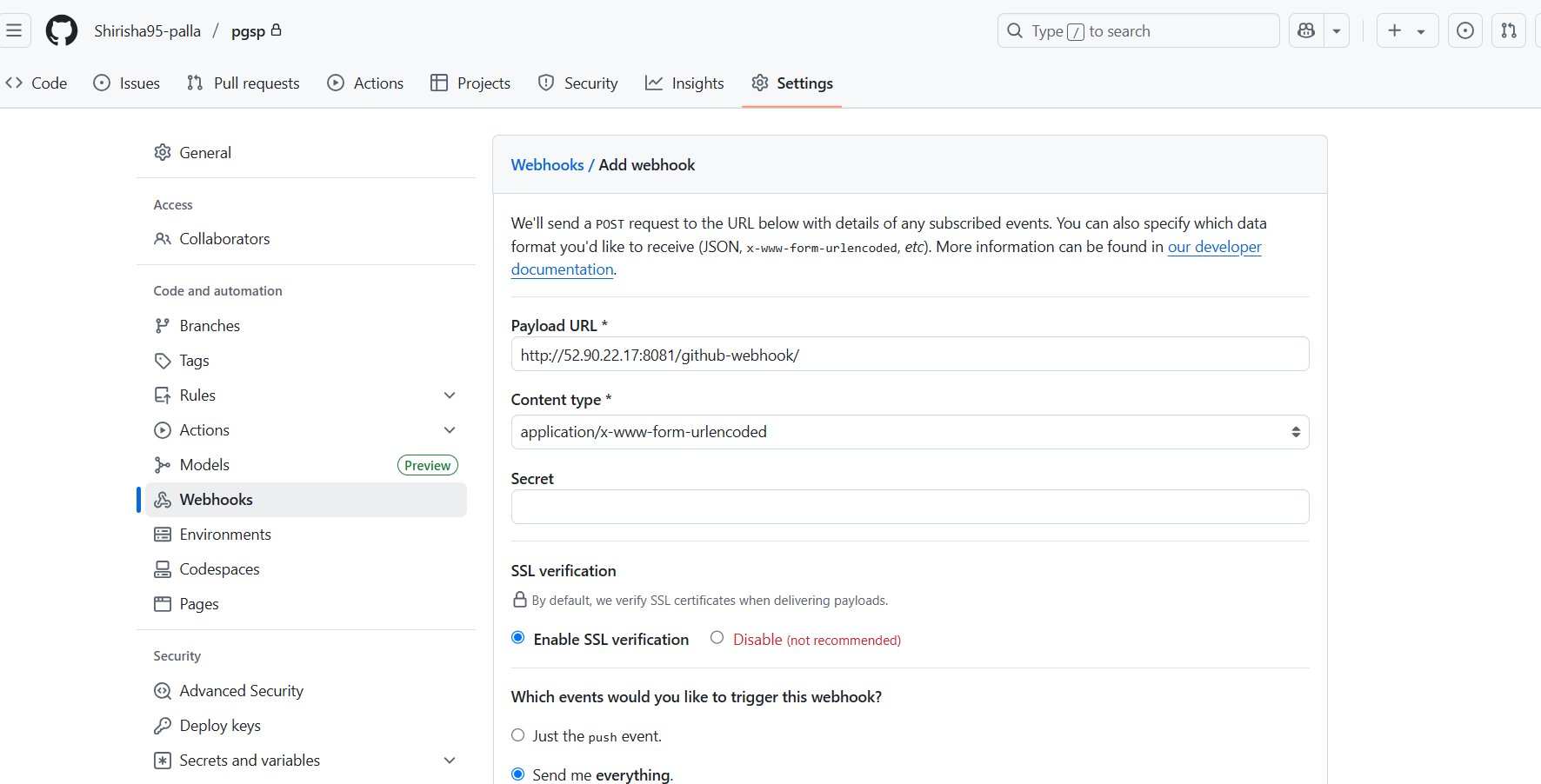


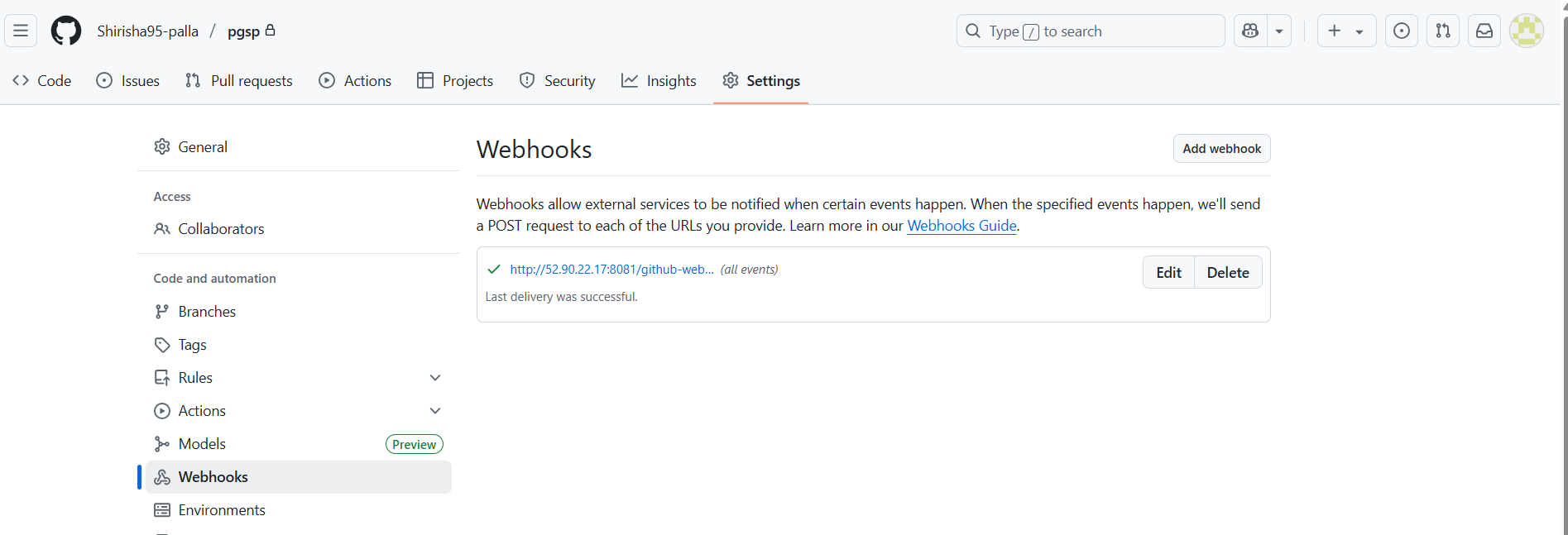
1. Configure webhooks to Jenkins job.

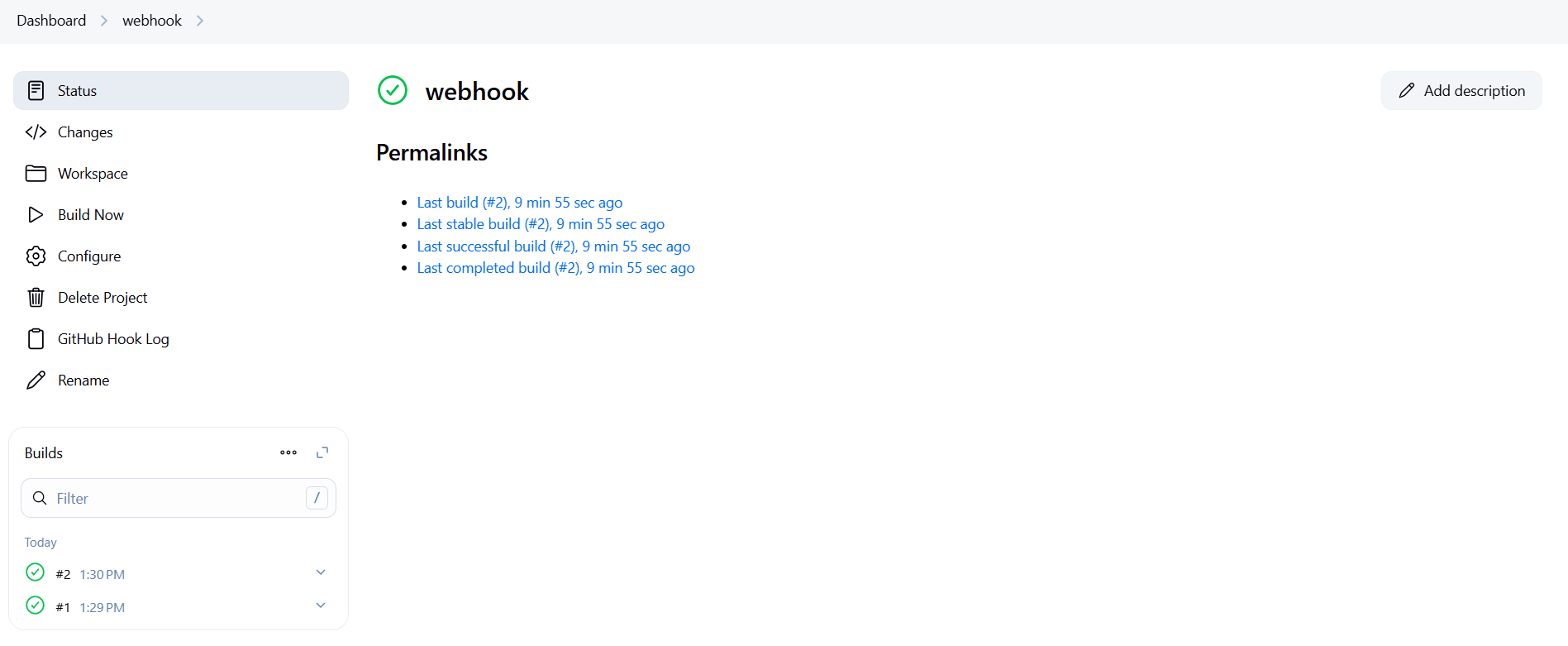
🡺Go to github and in the settings >>select webhoots>>give the   
url with ipaddres/github-webhook>>save it  
🡺In the jenkins job create a job >>give the git url>>select git   
🡺hooks trigger>>click on save>>for very change anything in the  
github it will automatically run the jobs

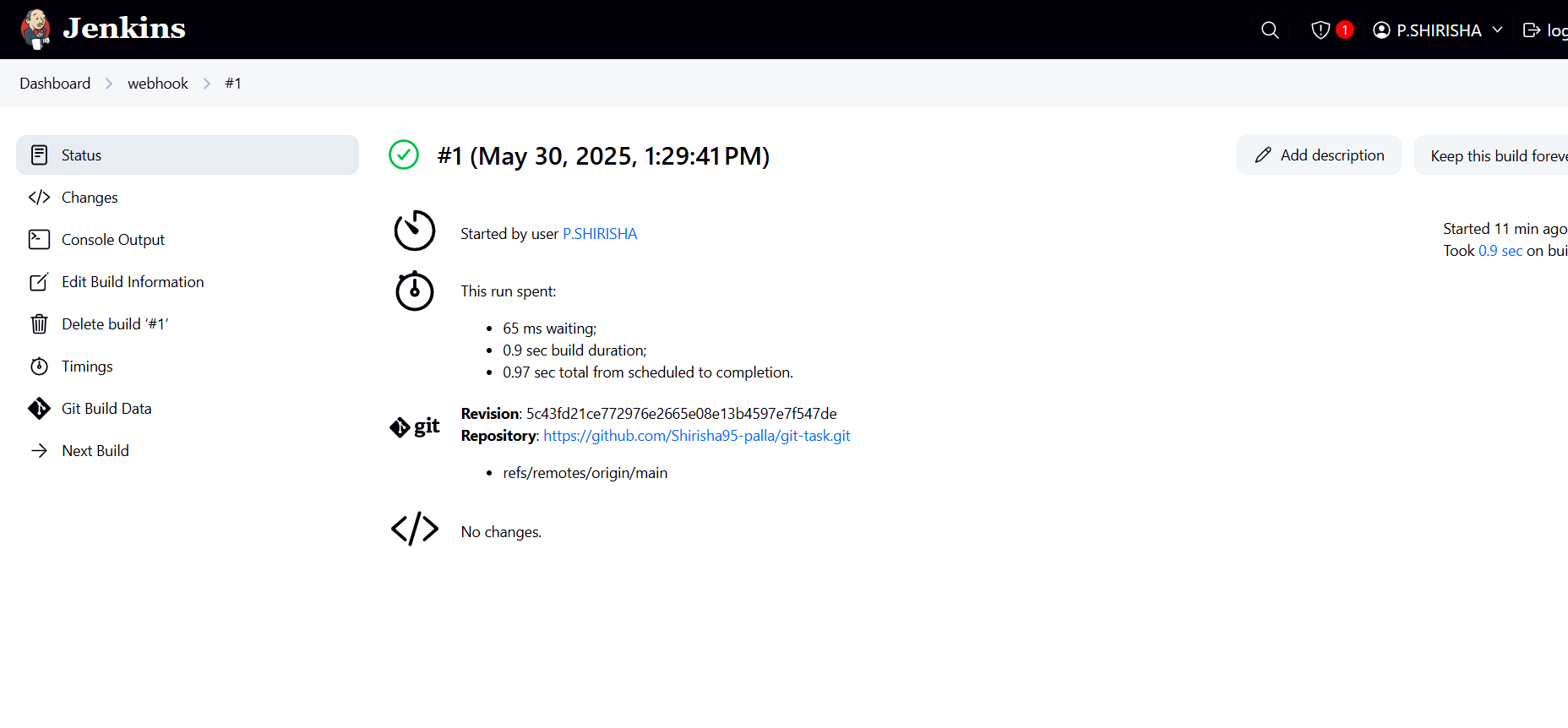


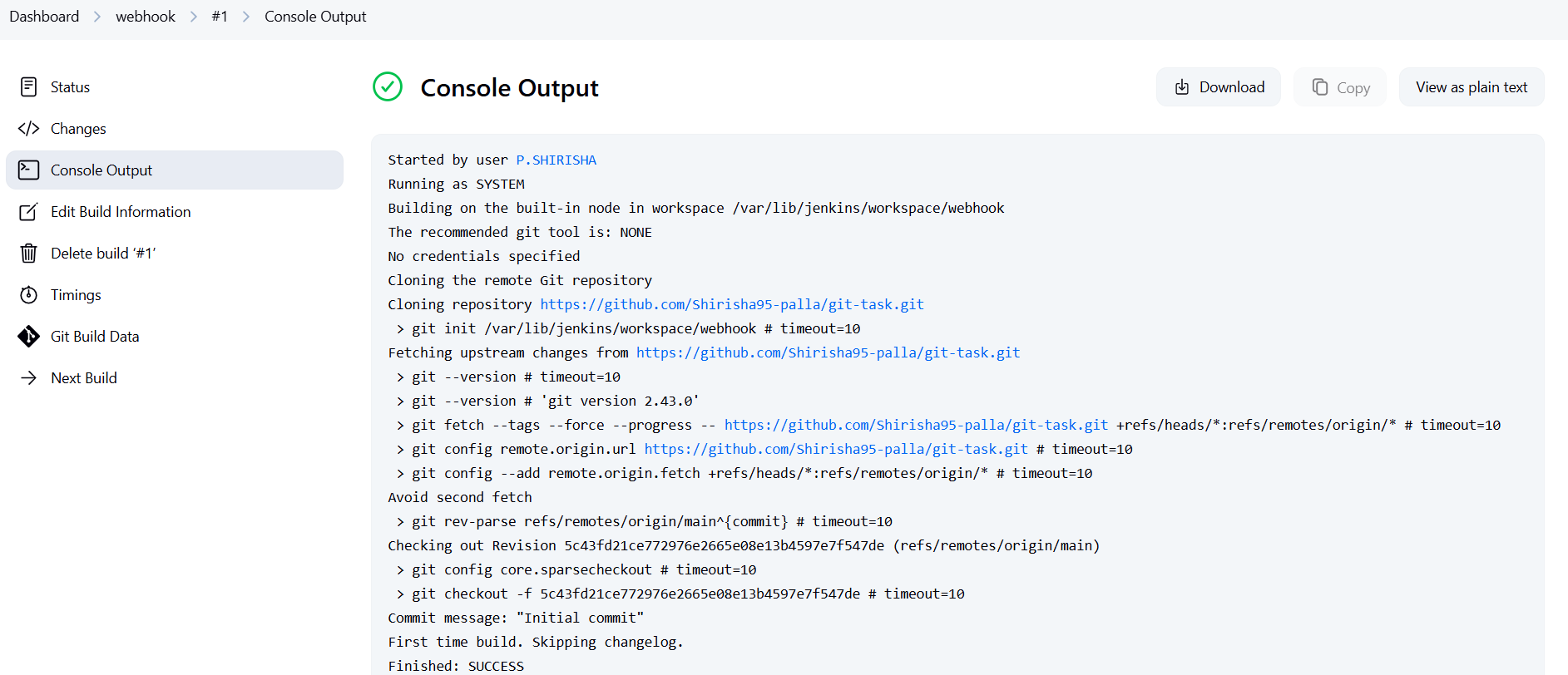


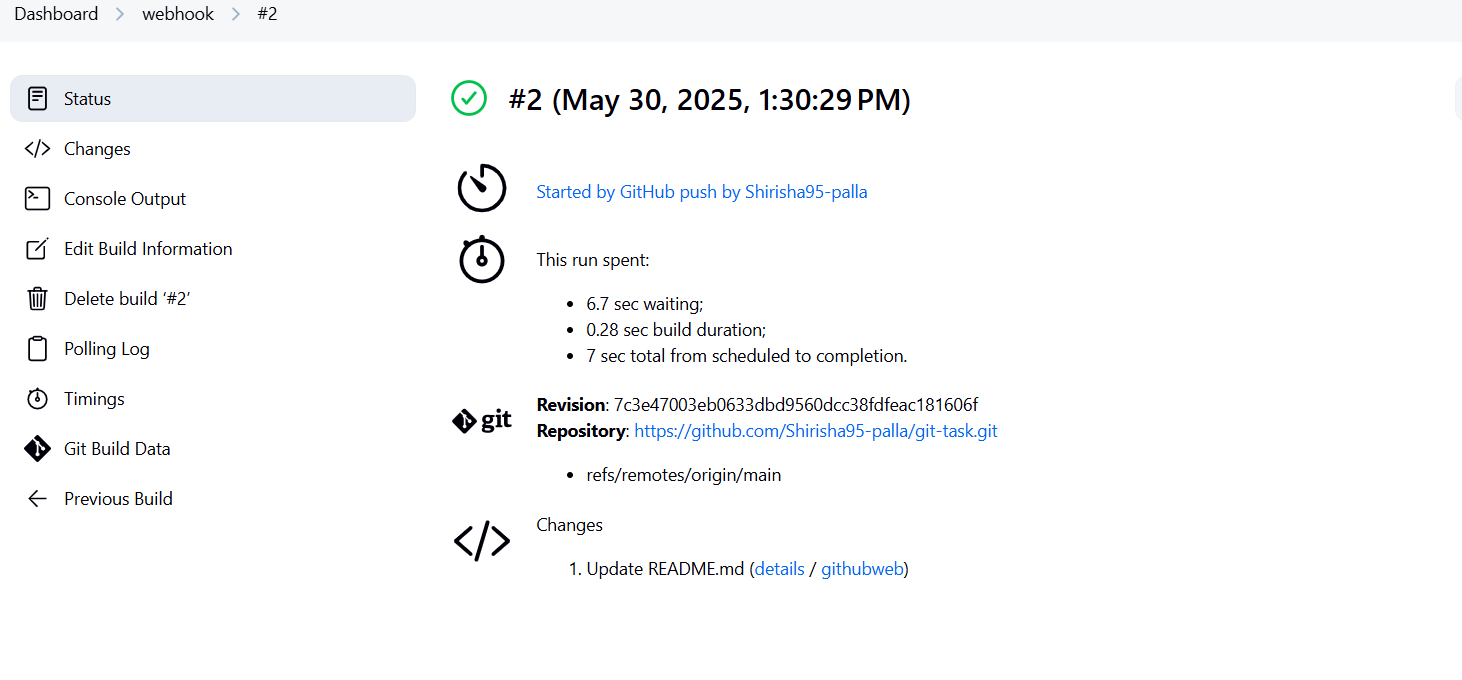








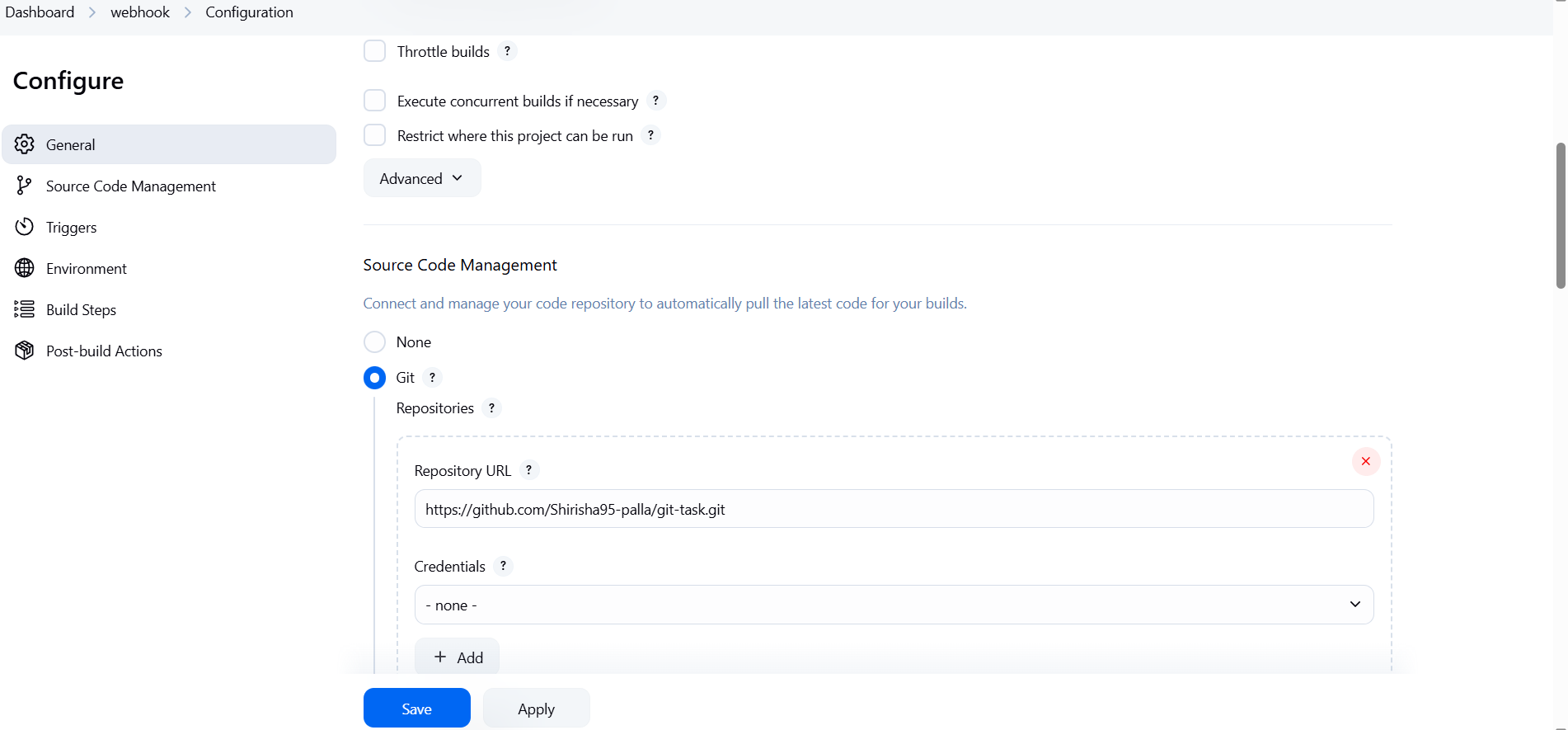


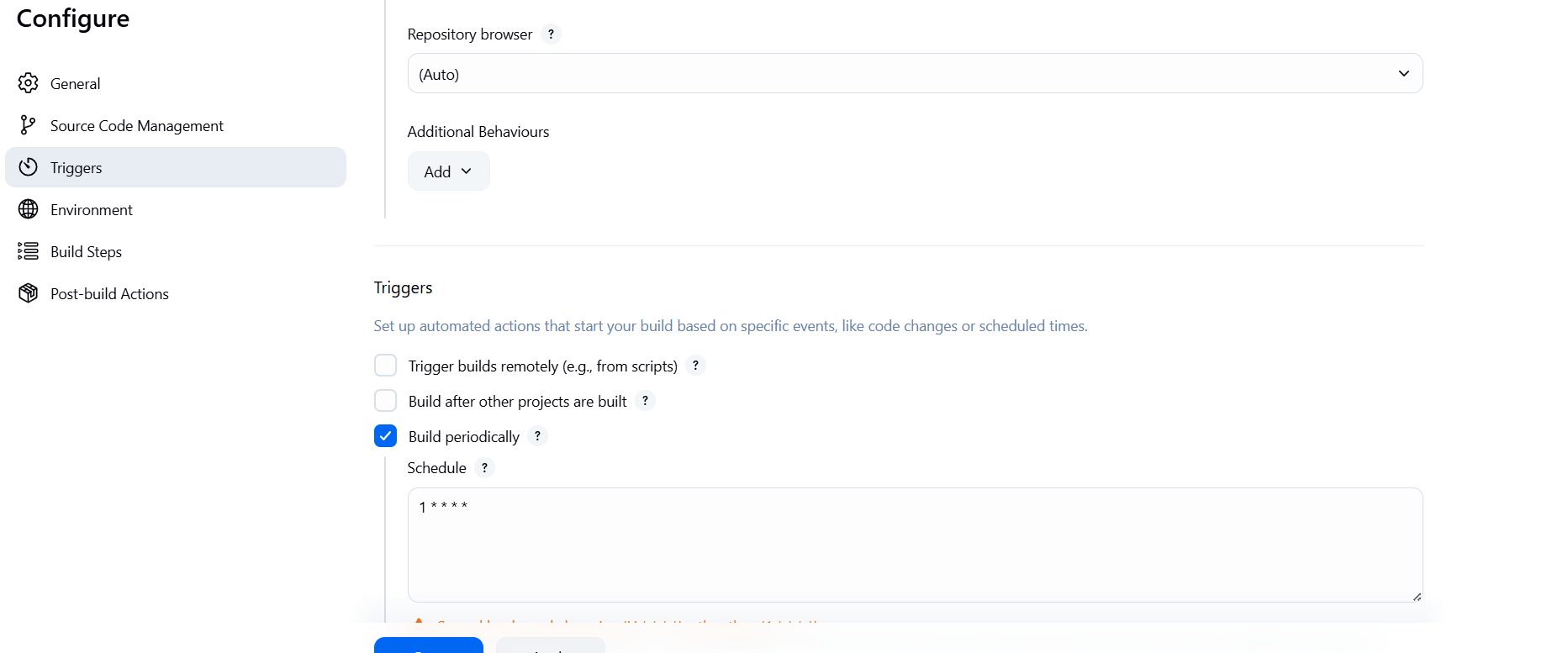


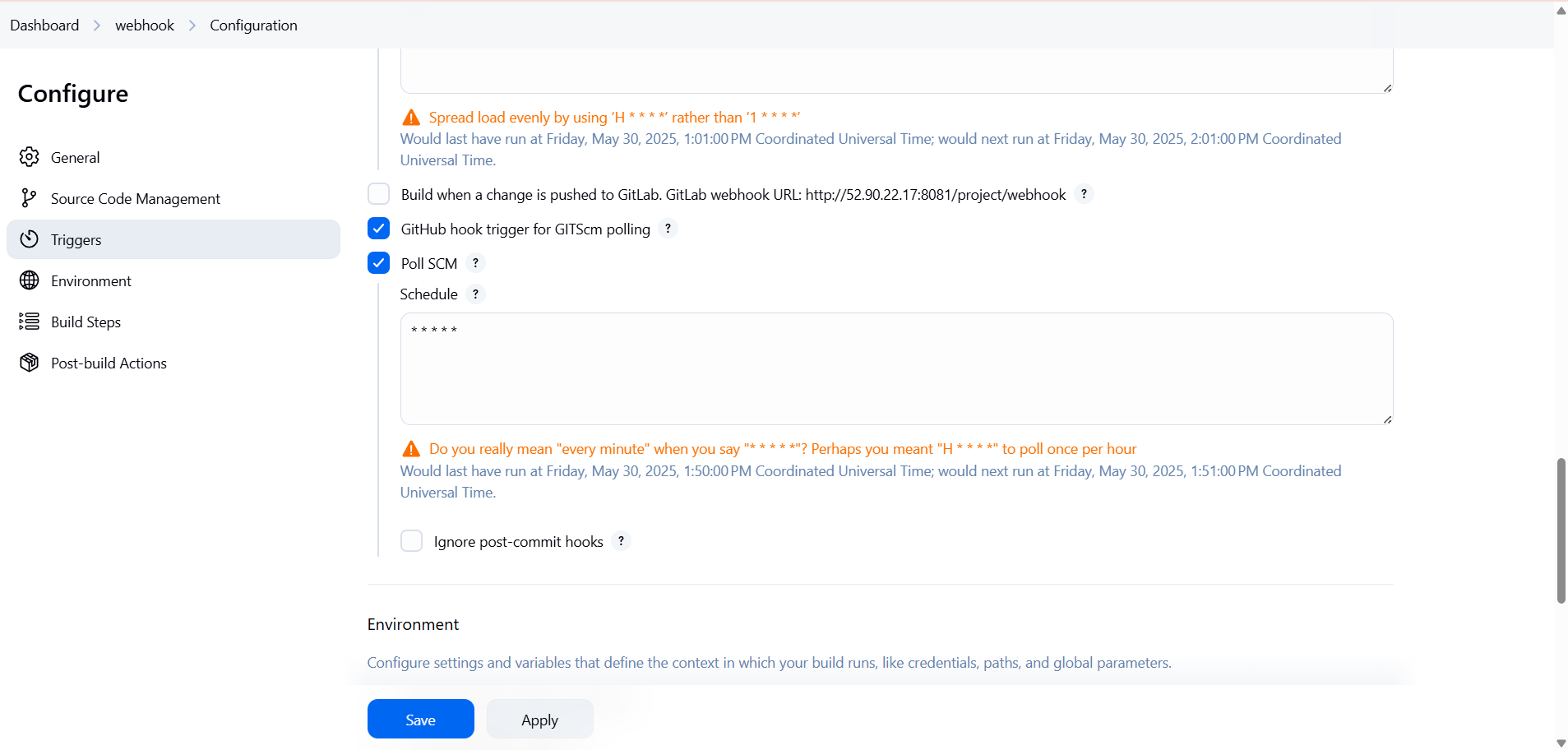


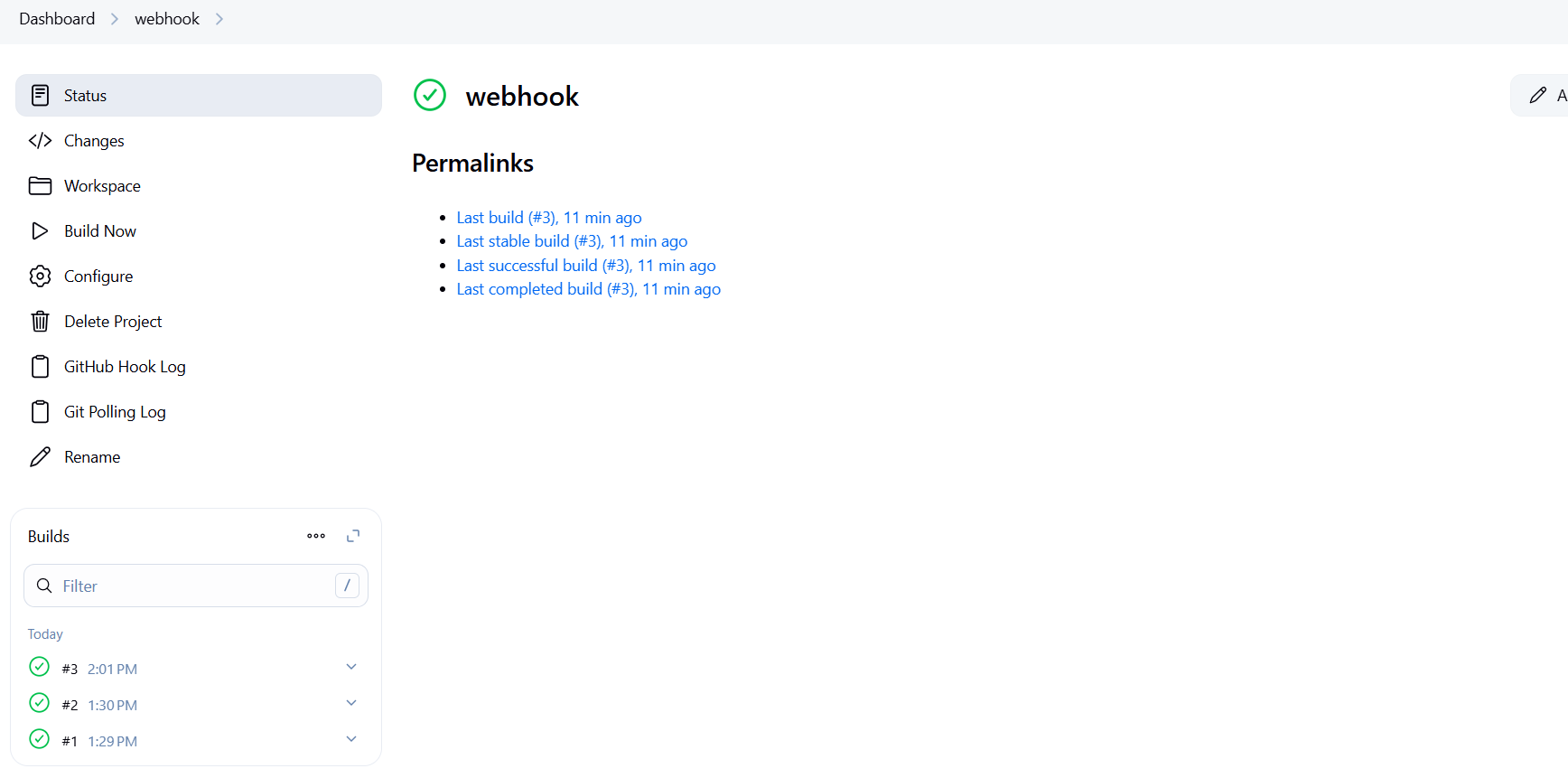
1. Configure poll scm and build periodical options in

🡺Create a new job and give gitub details >>in the pool scm give   
time run 1 min>>save >>if you done any changes in the github  
🡺It will run again automaticaly



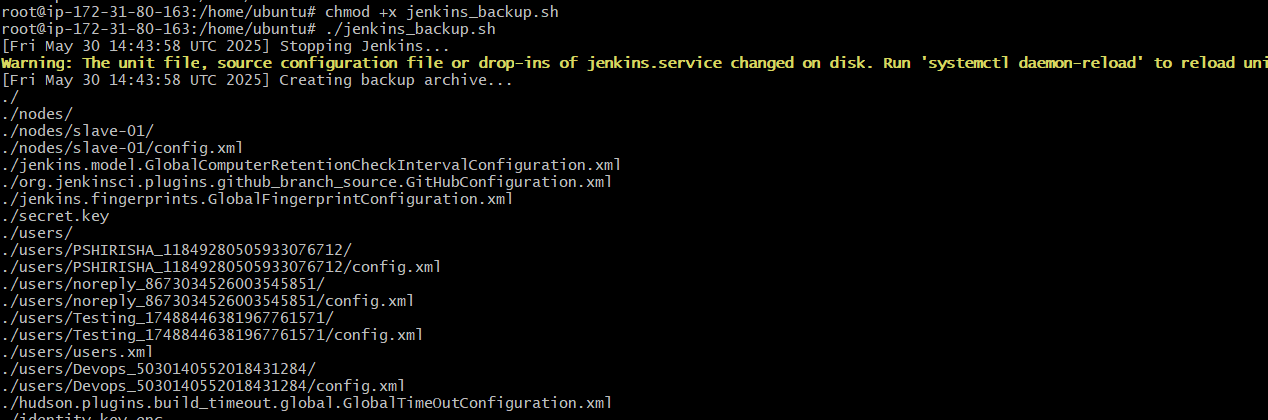


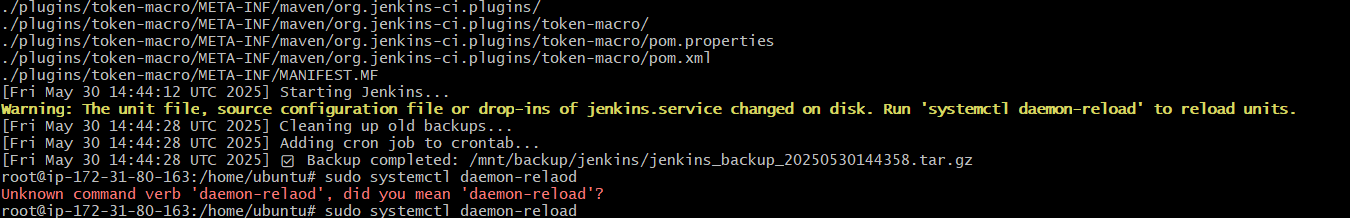


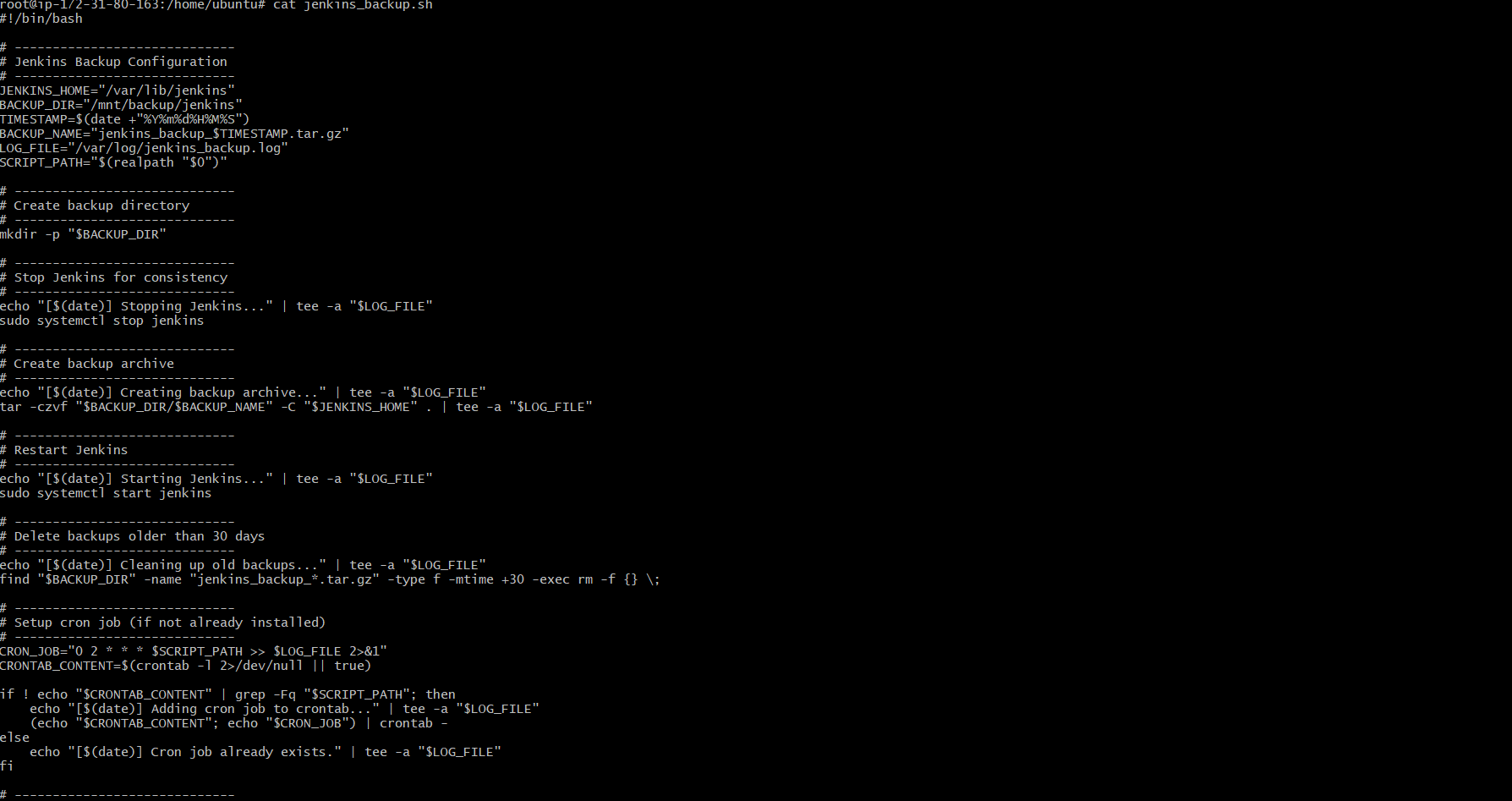


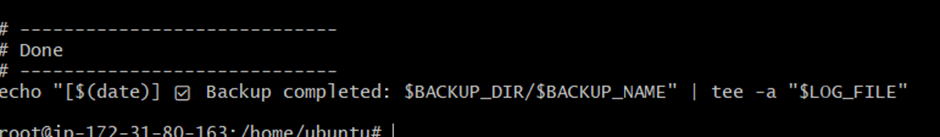
1. Take backup of jenkins server by using bash script.

In the Junkins main CLI  
in that create a file  
mkdir /var/lib/jenkins\_backups  
vi /home/ubuntu/jenkins\_backup.sh  
#!/bin/bash  
# Jenkins home directory  
JENKINS\_HOME="/var/lib/jenkins"  
# Backup destination directory  
BACKUP\_DIR="/var/lib/jenkins\_backups"  
# Create backup directory if it doesn't exist  
mkdir -p "$BACKUP\_DIR"  
# Current timestamp  
TIMESTAMP=$(date +%F\_%H-%M-%S)  
# Backup file name  
BACKUP\_FILE="jenkins\_backup\_$TIMESTAMP.tar.gz"  
# Stop Jenkins before backup (optional but safer)  
echo "Stopping Jenkins service..."  
sudo systemctl stop jenkins  
# Create the backup  
echo "Creating backup..."  
tar -czf "$BACKUP\_DIR/$BACKUP\_FILE" -C "$JENKINS\_HOME" .  
# Start Jenkins after backup  
echo "Starting Jenkins service..."  
sudo systemctl start jenkins  
# List created backup  
echo "Backup created at: $BACKUP\_DIR/$BACKUP\_FILE"  
chmod +x jenkins\_backup.sh  
sudo ./jenkins\_backup.sh



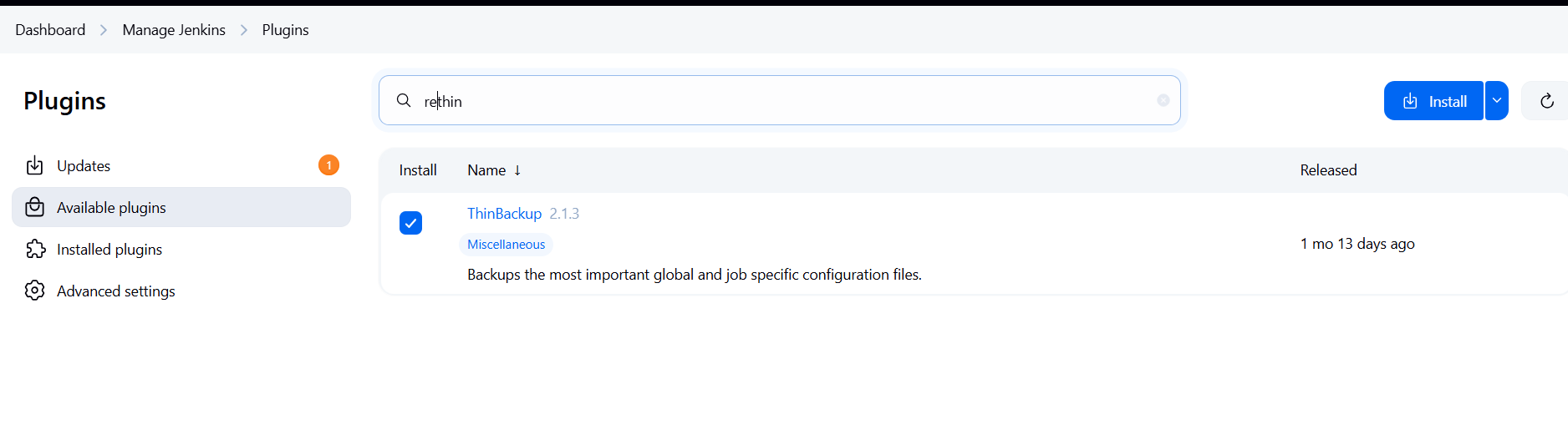


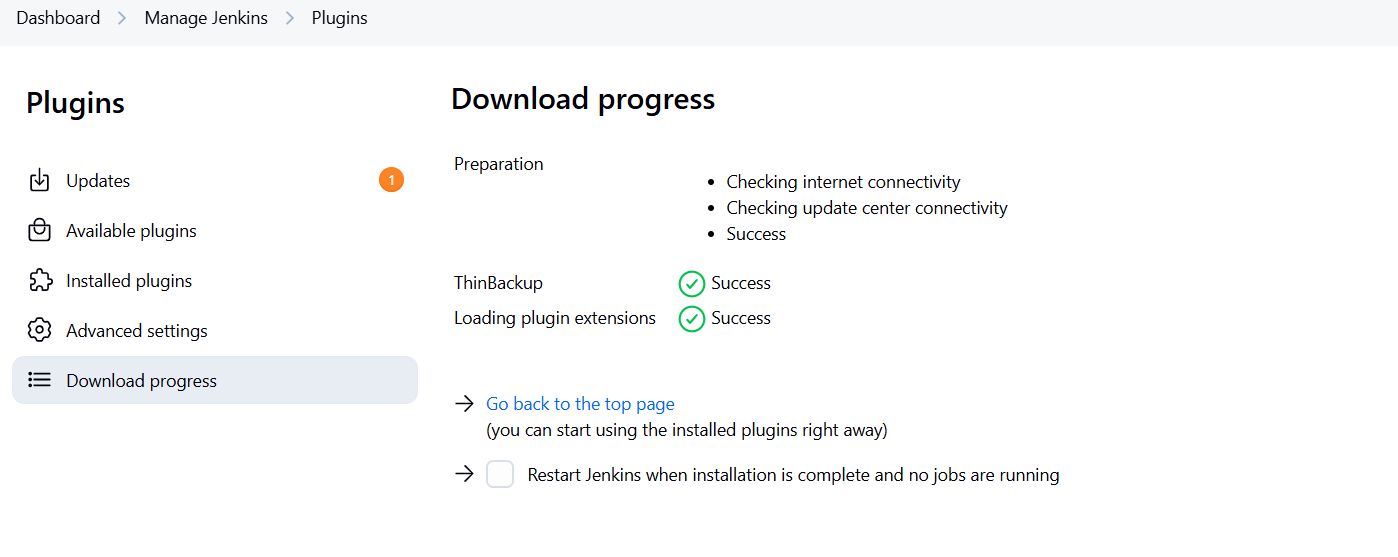




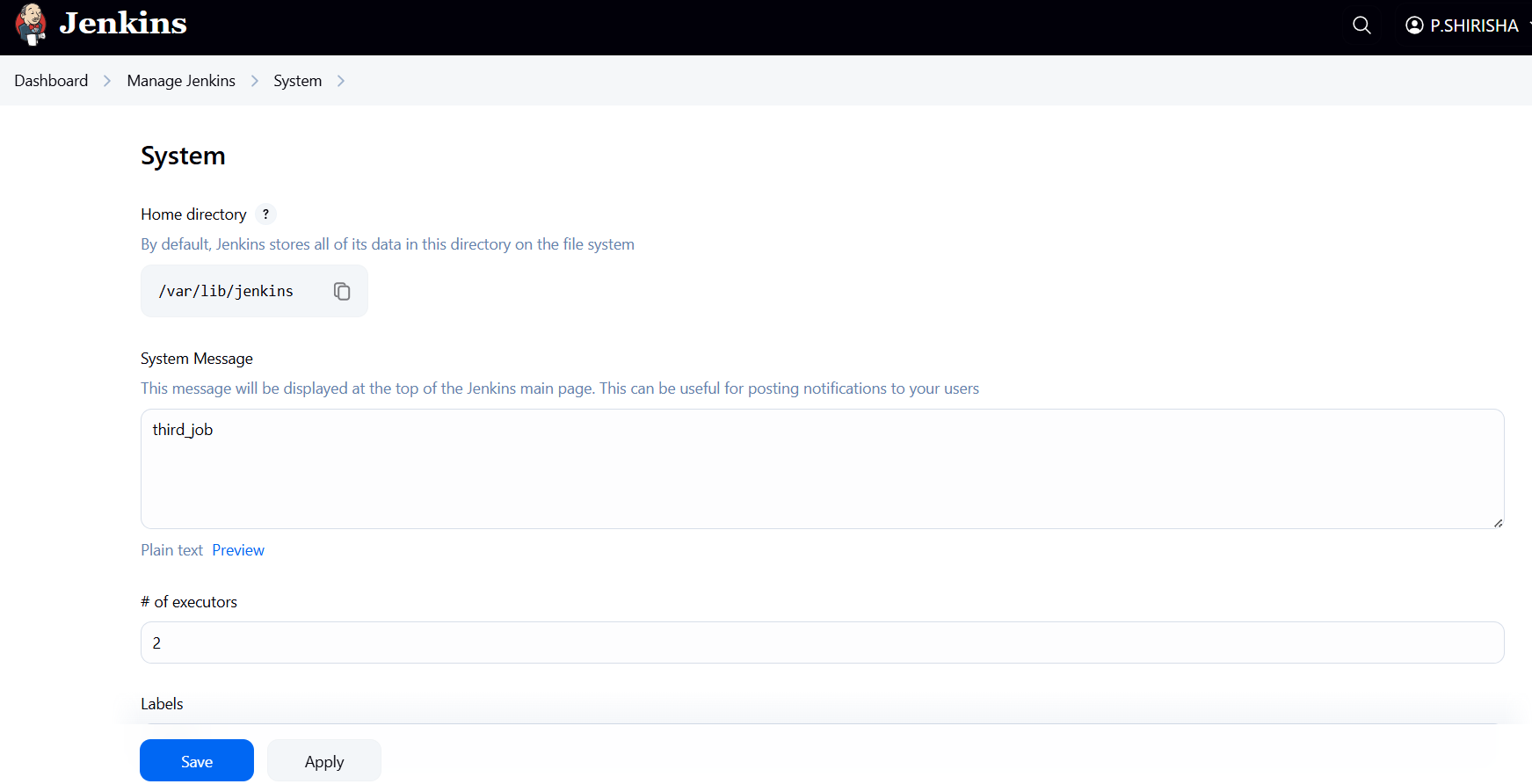
1. Take backup of jenkins using rethin backup plugin.

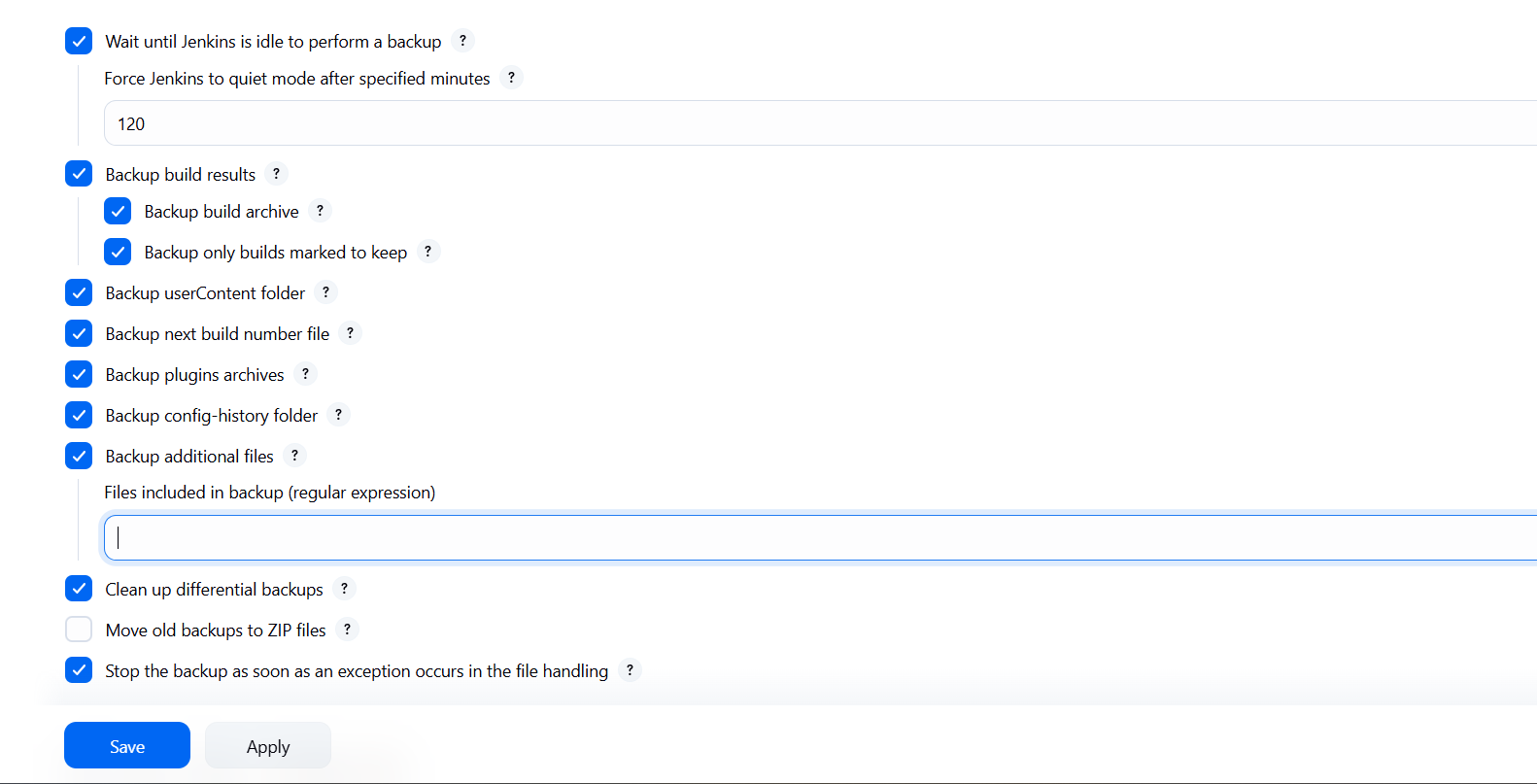
Go to plugin>>click available install>>type   
thinbackup>>install it  
later install go to manage jenkins>>click on system>>go to the   
option backup directory>>give the directory file location   
>>/var/lib/jenkins/thinbackup>>give the corn job time like one   
day or hourly>>select all option for backup>>click on save  
go to manage junckins>>click on thin backup>>click on   
backup>>  
to verify>>go to cli >>go to the back up location>>ls   
/var/lib/jenkins/thinbackup>>you will see the backup   
time>>work done

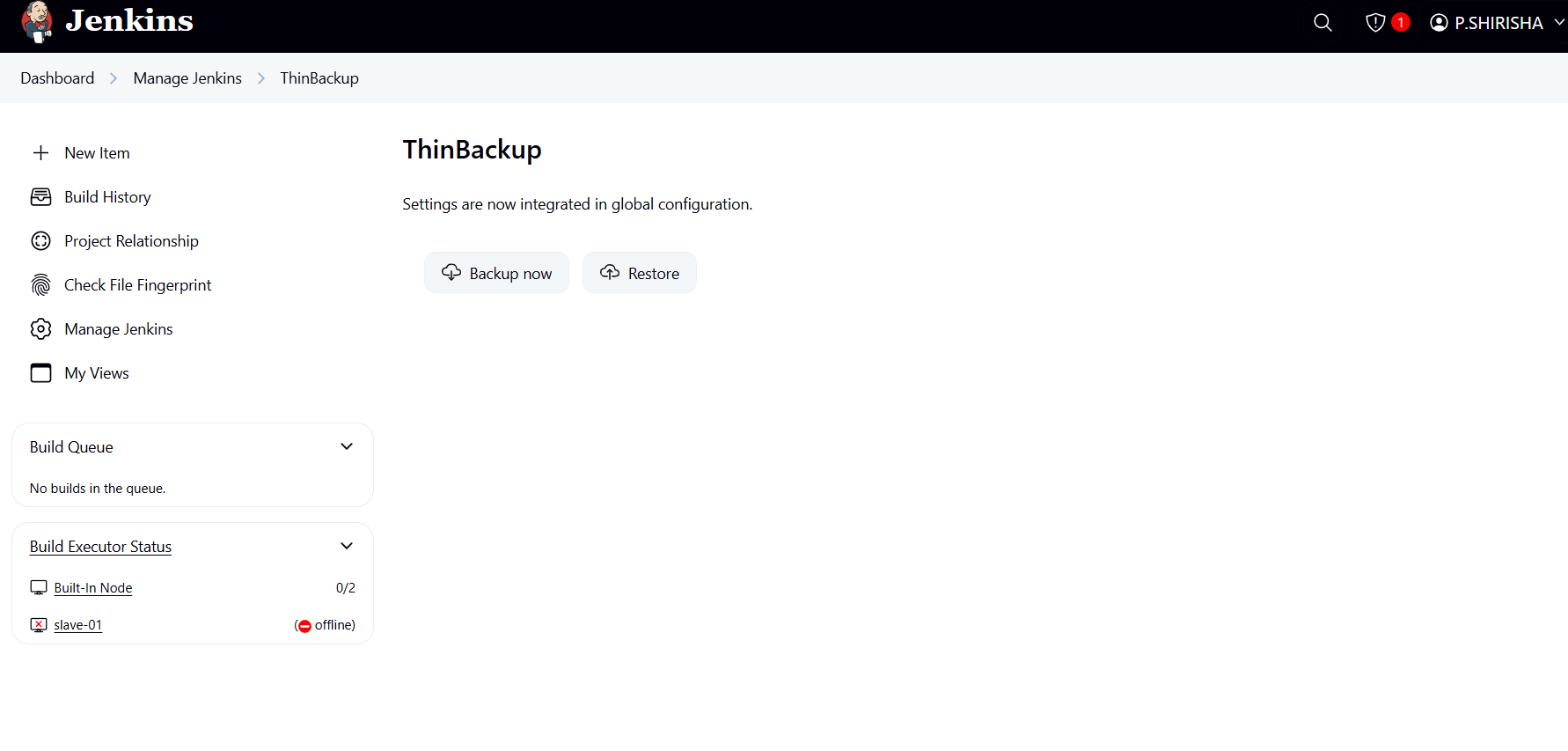














1. Setup a new jenkins server and dump the backup taken in task4.

First create the intance with 20GB storage  
then open the instance in CLI and install the jenkins  
Download jenkins Repo:  
sudo wget -O /etc/yum.repos.d/jenkins.repo   
https://pkg.jenkins.io/redhat-stable/jenkins.repo  
Import the jenkins key:  
sudo rpm --import   
https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key  
Update ec2:  
sudo yum upgrade  
Add required dependencies for the jenkins   
package:  
sudo amazon-linux-extras enable corretto17  
sudo yum clean metadata  
sudo yum install java-17-amazon-corretto-devel -y  
Install Jenkins:  
sudo yum install jenkins  
sudo systemctl start jenkins  
sudo systemctl status jenkins  
I have downloaded it from local repo not in the ec2   
server:  
use old backup ip add:

scp -i ~/Downloads/vpc1-key.pem -r   
ubuntu@35.170.187.102:/var/lib/jenkins/thinbackup   
~/Downloads/  
then download it to the new instance:  
use the new ip address:  
scp -i ~/Downloads/vpc1-key.pem -r   
~/Downloads/thinbackup   
ubuntu@54.164.4.56:/home/ubuntu/  
then login to the new server:  
use new ip and your pem.key  
ssh -i ~/Downloads/vpc1-key.pem   
ubuntu@54.164.4.56  
move backup file to jenkins:  
 sudo mv /home/ubuntu/thinbackup /var/lib/jenkins/  
change the ownership to jenkins :  
sudo chown -R jenkins:jenkins /var/lib/jenkins/thinbackup  
stop the jenkins:  
sudo systemctl stop jenkins  
copy the backup file to library:  
sudo cp -r /var/lib/jenkins/thinbackup/ /var/lib/jenkins/  
change the FULL-2025-05-30\_15-13/\* owner:  
sudo chown -R Jenkins:Jenkins /var/lib/jenkins

then start the jenkins:  
sudo chown -R Jenkins:Jenkins /var/lib/jenkins  
old jenkins

