**Jenkins Master and slave**

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 34.227.74.239 >>/var/lib/jenkins/.ssh/known\_hosts

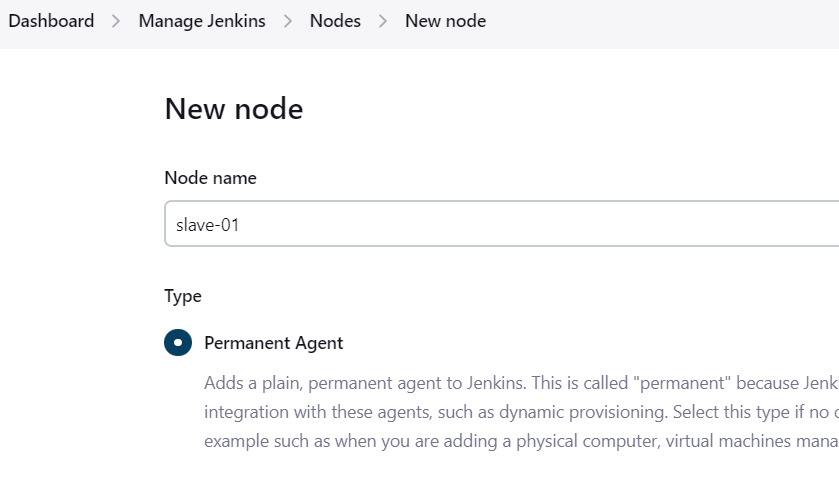
# ssh-keyscan -H 172.31.38.42 >>/var/lib/jenkins/.ssh/known\_hosts

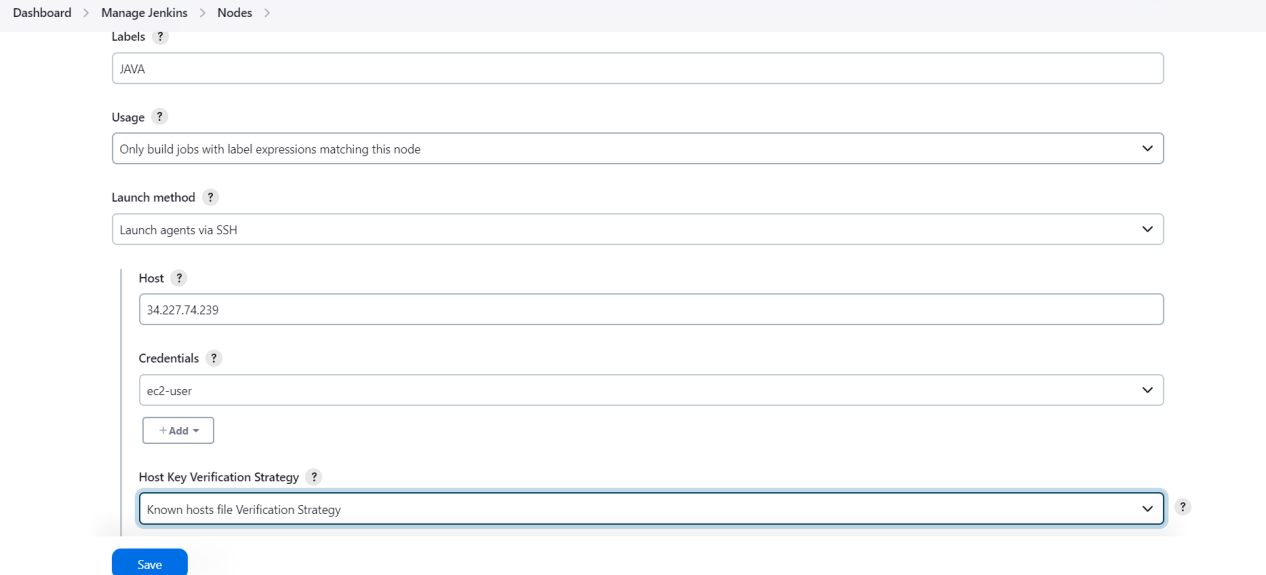
chown jenkins:jenkins known\_hosts

#we need to change the owner as we ran ssh-keyscan command using “root” user.

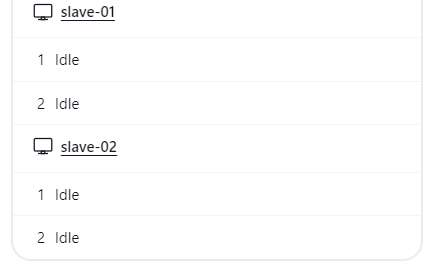
# default user of Jenkins will be “jenkins”

chmod 700 known\_hosts



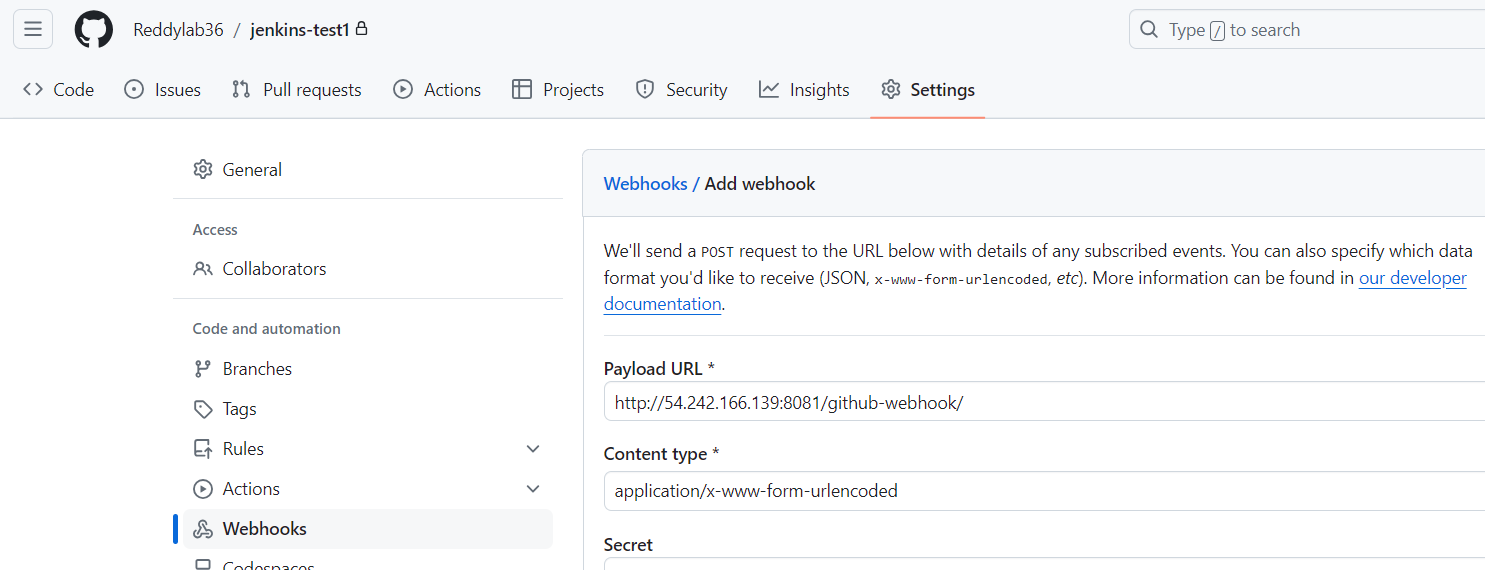


· [Dashboard](http://54.242.166.139:8081/) →[Nodes](http://54.242.166.139:8081/computer/)→[slave-01](http://54.242.166.139:8081/computer/slave-01/) and slave-02→Log

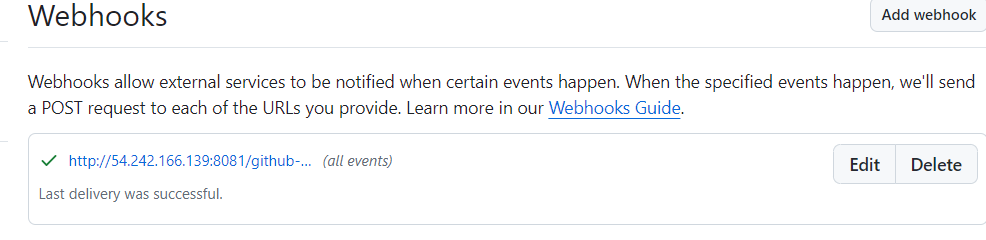


1. **Configure webhooks to jenkins job**.

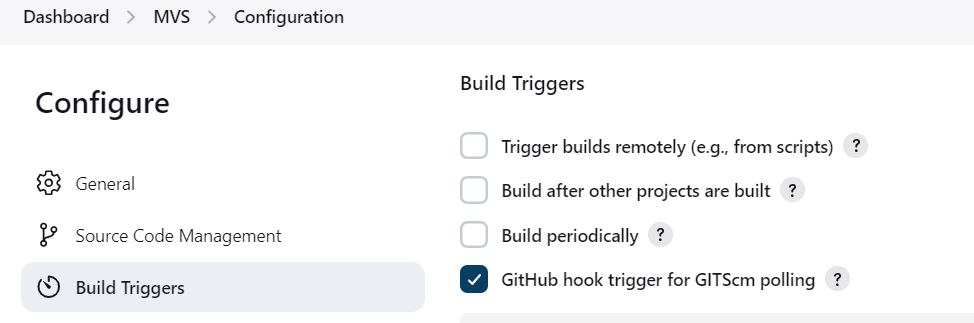
\* Go to github →open the repository→go to settings→select webhook→enter the payload url



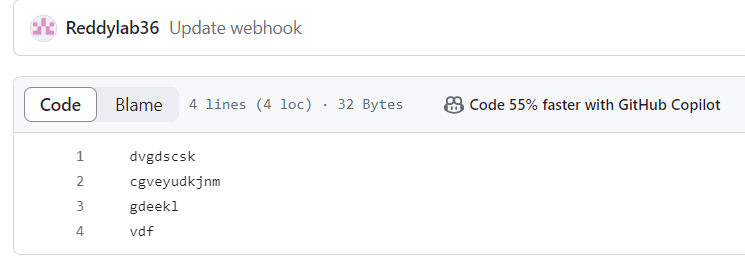
\* Created the webhook request to the jenkins for triggering



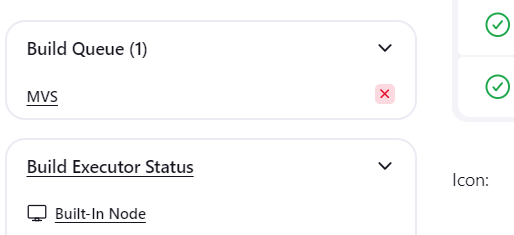
\* configuring the webhook for a job to trigger if anything changes happened in the github



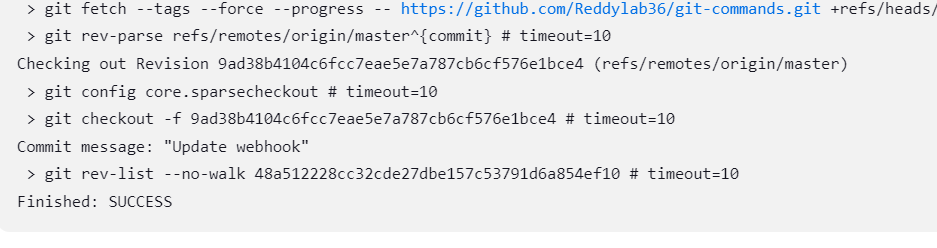
\* Done with some changes and commited the changes in github



\* Then it will triggered automatically and going to build the job

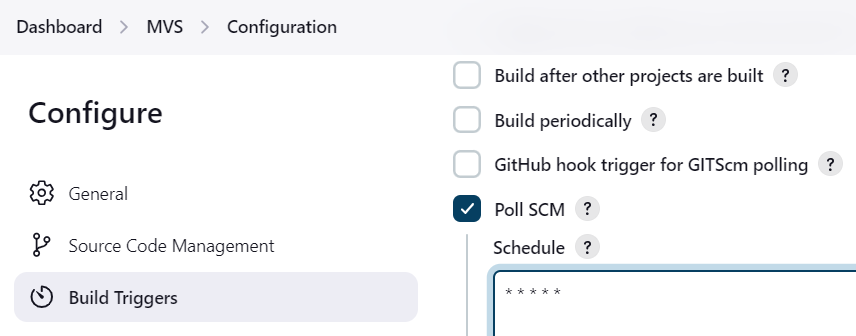


\*Automatically it will triggered and build the job

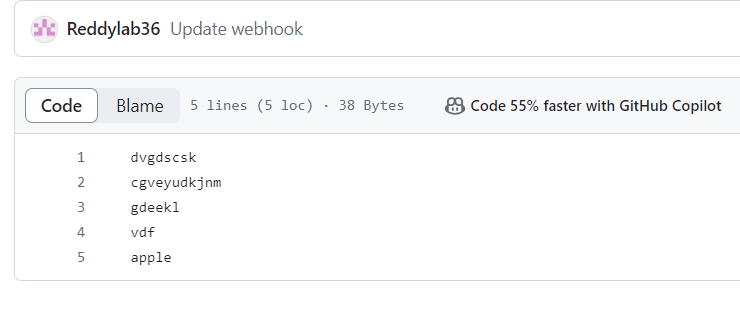


1. **Configure poll scm and build periodical options in jenkins job.**

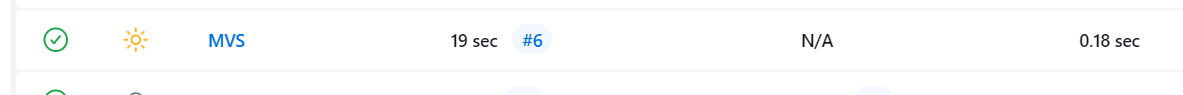
Go to Jenkins Dashboard→selct the MVS job→go to configuration→select Source Code Management (enter the github url)→go to Build Triggers→select **Poll SCM**→give the timing for trigger



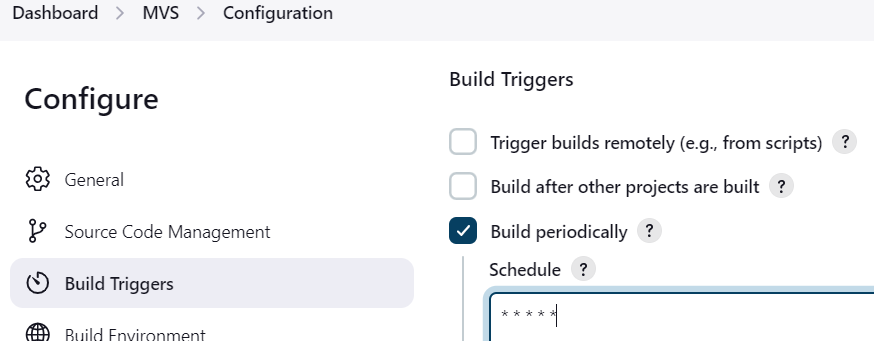
\* Done with some changes in github



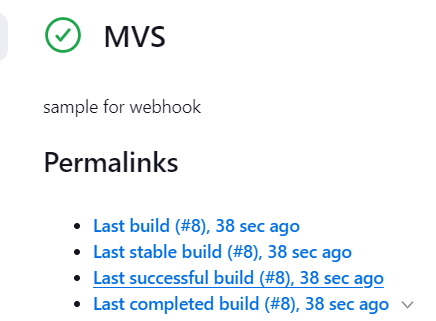
\* Then it will trigger after a minute



\* Go to Jenkins Dashboard→selct the MVS job→go to configuration→select Source Code Management (enter the github url)→go to Build Triggers→select **Build Periodically**→give the timing for trigger



\* Nothing will commit the changes in github but it will trigger in every minute by Build periodically

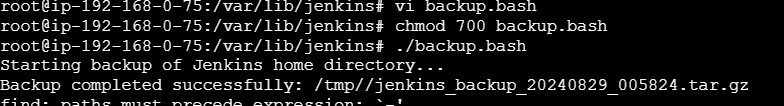


1. **Take backup of Jenkins server by using bash script.**

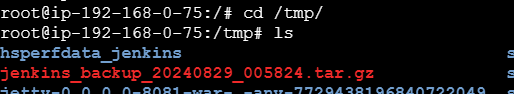
\* First go to jenkins server and make script for creatingthe backup

Script

#!/bin/bash  
# Configuration  
JENKINS\_HOME="/var/lib/jenkins" # Path to Jenkins home directory  
BACKUP\_DIR="/tmp/" # Directory where backups will be stored  
TIMESTAMP=$(date +"%Y%m%d\_%H%M%S") # Timestamp for backup file name  
BACKUP\_FILE="jenkins\_backup\_$TIMESTAMP.tar.gz" # Backup file name  
# Ensure the backup directory exists  
mkdir -p "$BACKUP\_DIR"  
# Create a backup of the Jenkins home directory  
echo "Starting backup of Jenkins home directory..."  
tar -czf "$BACKUP\_DIR/$BACKUP\_FILE" -C "$JENKINS\_HOME" .  
# Check if the backup was successful  
if [ $? -eq 0 ]; then  
echo "Backup completed successfully: $BACKUP\_DIR/$BACKUP\_FILE"  
else  
echo "Backup failed!"  
exit 1  
fi  
# Optionally, you can remove old backups to save space  
# Find and delete backups older than 30 days  
find "$BACKUP\_DIR" -name "jenkins\_backup\_\*.tar.gz" -type f -mtime +30 -  
exec rm {} \;  
echo "Old backups cleaned up."  
exit 0

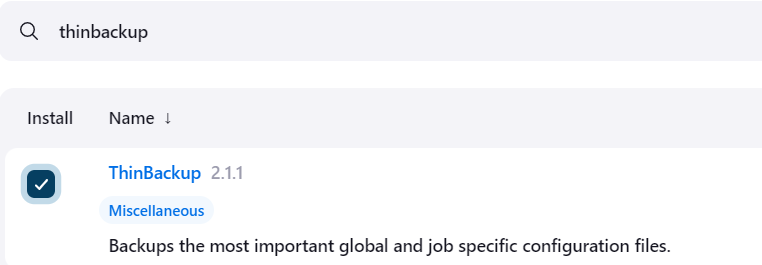


\* Backup has been stored in “**/tmp/”** folder

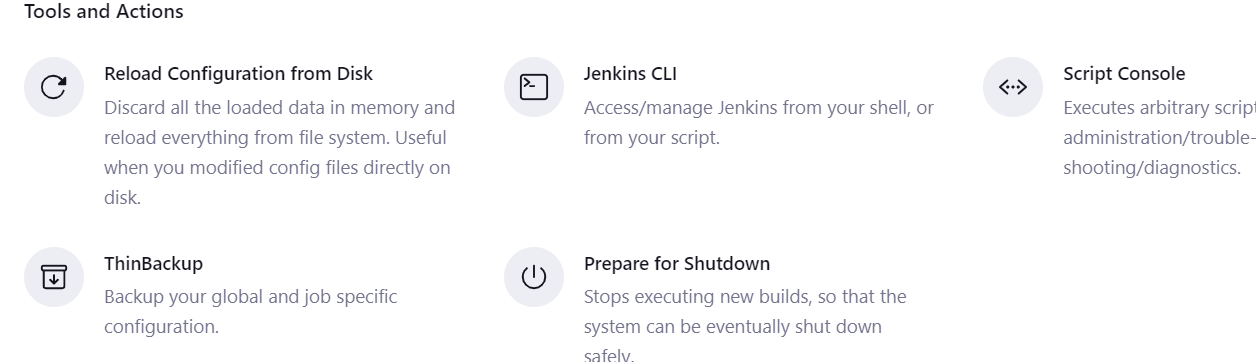


1. **Take backup of Jenkins using rethin backup plugin.**

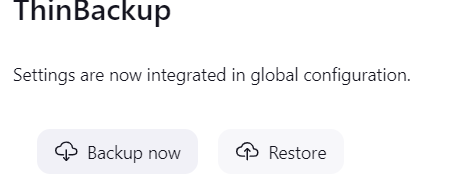
Go to Dashboard→Manage Jenkins→click on plugin→Search thinbackup Plugin→Install it



\* The **Thinbackup plugins** are integrated in gobally



\* Now we can see the backup now option to take up of thinbackup plugin



Backingup is not working

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

\* Firest create a new jenkins instance and install the packages using below

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 amazon-linux-extras install java-openjdk11

sudo yum install jenkins

systemctl start jenkins

systemctl status jenkins

\* Then access into the jenkins server

cat /var/lib/jenkins/secrets/initialAdminPassword

\* we can copy the pemkey from local to jenkins-master server

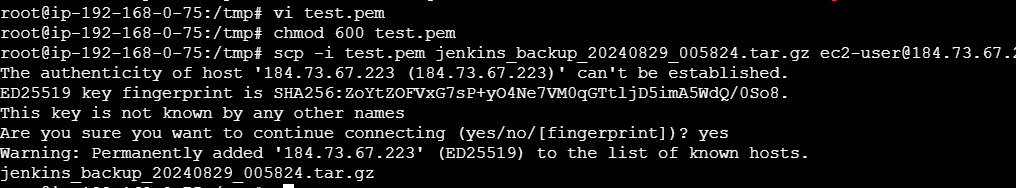
vi test.pem

\* Then given permissions to this pem key

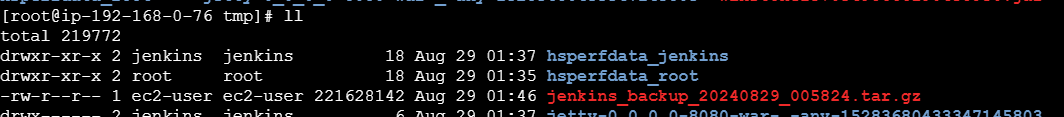
chmod 600 test.pem

\* Securely copy the backup file from jenkins-master server to jenkins-backup server

scp -i test.pem jenkins\_backup\_20240829\_005824.tar.gz ec2-user@184.73.67.223:/tmp



\* Now backup file has been dumped into new jenkins server



\* Go to /tmp/ folder and unzip the backup file

\*Then restart the jenkins using

systemctl restart jenkins

Or

http://<public-ip>:8080/restart



