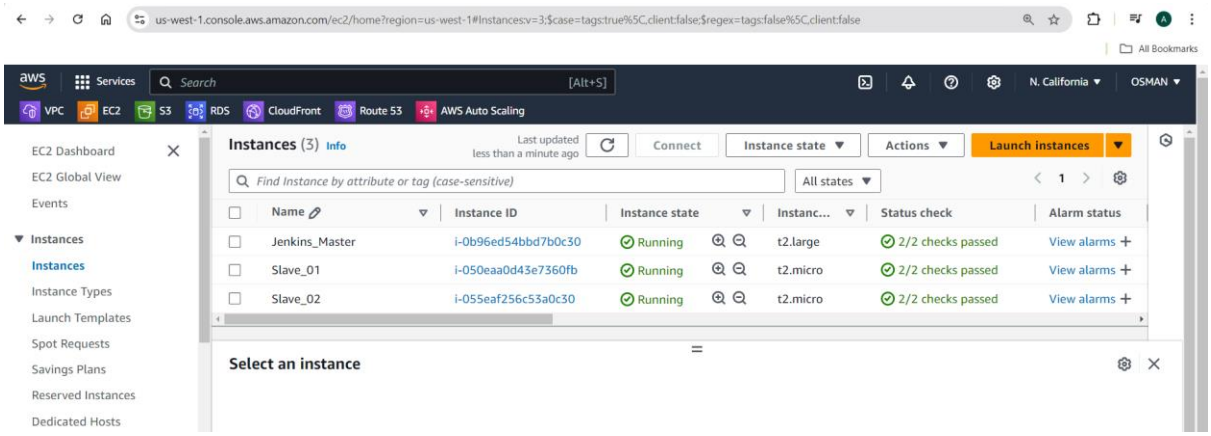
1) Configure 2 slave machines in jenkins master.

Here I have one master Server and Two Slave Nodes.



- Connect two slave servers and install java only because these is prerequisite.
- No need to install the Jenkins in the salve servers.
- I am connect slave_01 server and install java-17 version.

```
[root@Slave_01 ~]# yum -y install java-17*
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.6 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package java-17-amazon-corretto.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXinerama for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrender for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
---> Package java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
---> Package java-17-amazon-corretto-headless.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: alsa-lib for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
---> Package java-17-amazon-corretto-javadoc.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
---> Package java-17-amazon-corretto-jmods.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Running transaction check
---> Package alsa-lib.x86_64 0:1.1.4.1-2.amzn2 will be installed
---> Package dejavu-sans-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Processing Dependency: dejavu-fonts-common = 2.33-6.amzn2 for package: dejavu-sans-fonts-2.33-6.amzn2.noarch
---> Package dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2 will be installed
---> Package dejavu-serif-fonts.noarch 0:2.33-6.amzn2 will be installed
---> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: fontpackages-filesystem for package: fontconfig-2.13.0-4.3.amzn2.x86_64
---> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libICE.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libSM.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
---> Package javapackages-tools.noarch 0:3.4.1-11.amzn2 will be installed
--> Processing Dependency: python-javapackages = 3.4.1-11.amzn2 for package: javapackages-tools-3.4.1-11.amzn2.noarch
--> Processing Dependency: libxslt for package: javapackages-tools-3.4.1-11.amzn2.noarch
---> Package libX11.x86_64 0:1.6.7-3.amzn2.0.5 will be installed
--> Processing Dependency: libX11-common >= 1.6.7-3.amzn2.0.5 for package: libX11-1.6.7-3.amzn2.0.5.x86_64
```

- I am connect slave_02 server and install java-17 version.

```
[root@Slave_02 ~]#
[root@Slave_02 ~]# yum -y install java-17*
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.6 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package java-17-amazon-corretto.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: libX11 for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXi for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXinerama for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXt for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrender for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: giflib for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
---> Package java-17-amazon-corretto-devel.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
---> Package java-17-amazon-corretto-headless.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: alsa-lib for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
--> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
---> Package java-17-amazon-corretto-javadoc.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
---> Package java-17-amazon-corretto-jmods.x86_64 1:17.0.12+7-1.amzn2.1 will be installed
--> Running transaction check
---> Package alsa-lib.x86_64 0:1.1.4.1-2.amzn2 will be installed
---> Package dejavu-sans-fonts.noarch 0:2.33-6.amzn2 will be installed
--> Processing Dependency: dejavu-fonts-common = 2.33-6.amzn2 for package: dejavu-sans-fonts-2.33-6.amzn2.noarch
---> Package dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2 will be installed
---> Package dejavu-serif-fonts.noarch 0:2.33-6.amzn2 will be installed
---> Package fontconfig.x86_64 0:2.13.0-4.3.amzn2 will be installed
--> Processing Dependency: fontpackages-filesystem for package: fontconfig-2.13.0-4.3.amzn2.x86_64
---> Package giflib.x86_64 0:4.1.6-9.amzn2.0.2 will be installed
--> Processing Dependency: libICE.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
--> Processing Dependency: libSM.so.6()(64bit) for package: giflib-4.1.6-9.amzn2.0.2.x86_64
```

Steps to be done on slave

==========

- Create ssh-keygen
- cat id_rsa.pub > authorized_keys

chmod 700 authorized_keys

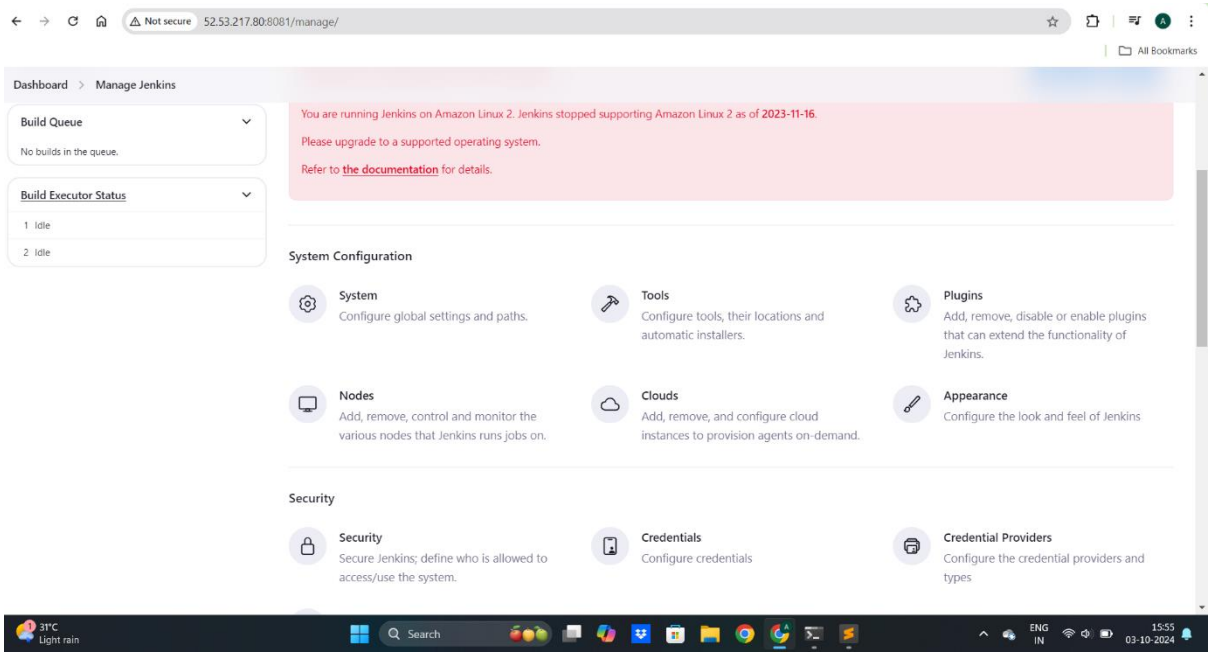
Next Steps to be done on master machine:

==========

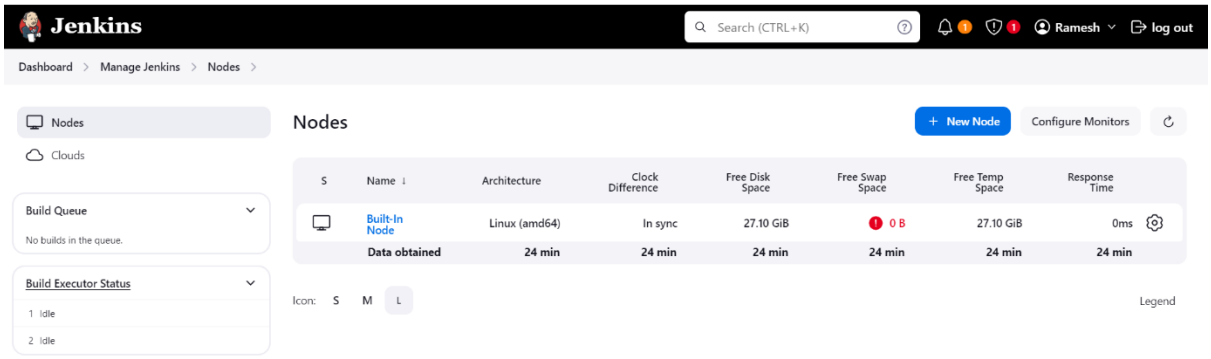
- Login to master machine
- switch to root user.
- mkdir -p /var/lib/jenkins/.ssh
- cd /var/lib/jenkins/.ssh
- ssh-keyscan -H SLAVE-NODE-IP-OR-HOSTNAME >>/var/lib/jenkins/.ssh/known_hosts
- # ssh-keyscan -H 172.31.38.42 >>/var/lib/jenkins/.ssh/known_hosts
- chown jenkins:jenkins known_hosts
- #we need to change the owner as we ran ssh-keyscan command using “root” user.
- # default user of Jenkins will be “jenkins”
- chmod 700 known_hosts

these all configurations Done Then go to the jenkins GUI

Go to the manage Jenkins , and click on Nodes.

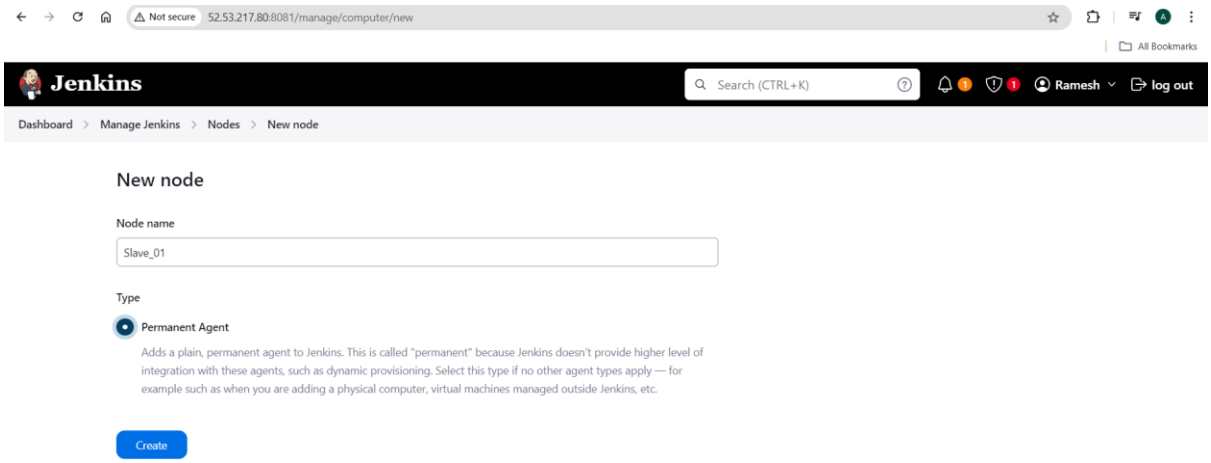


Here click on the New Node.

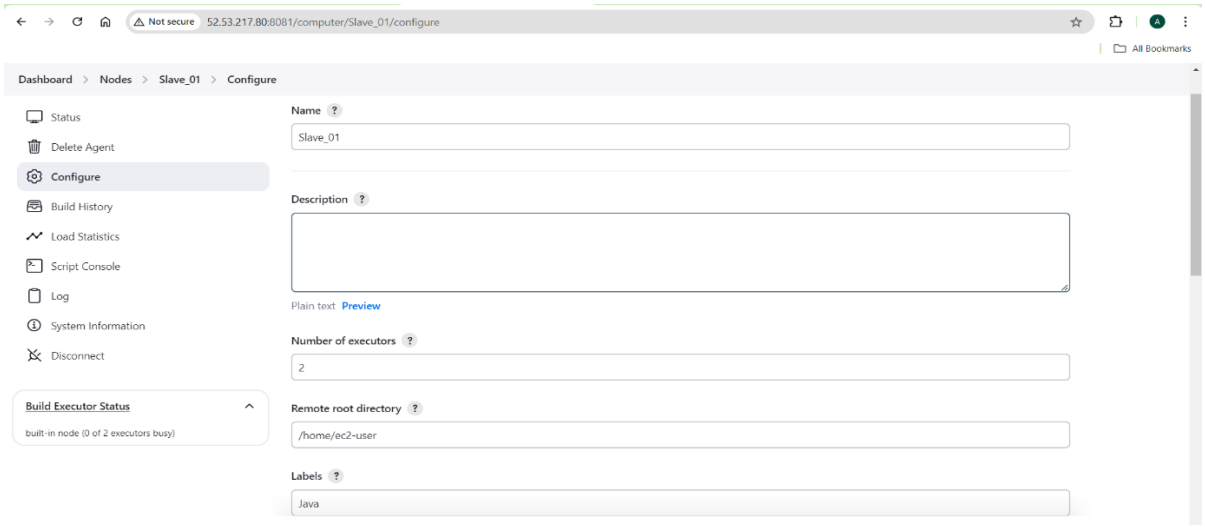


Give Node name and select permanent Agent.

Click on create.



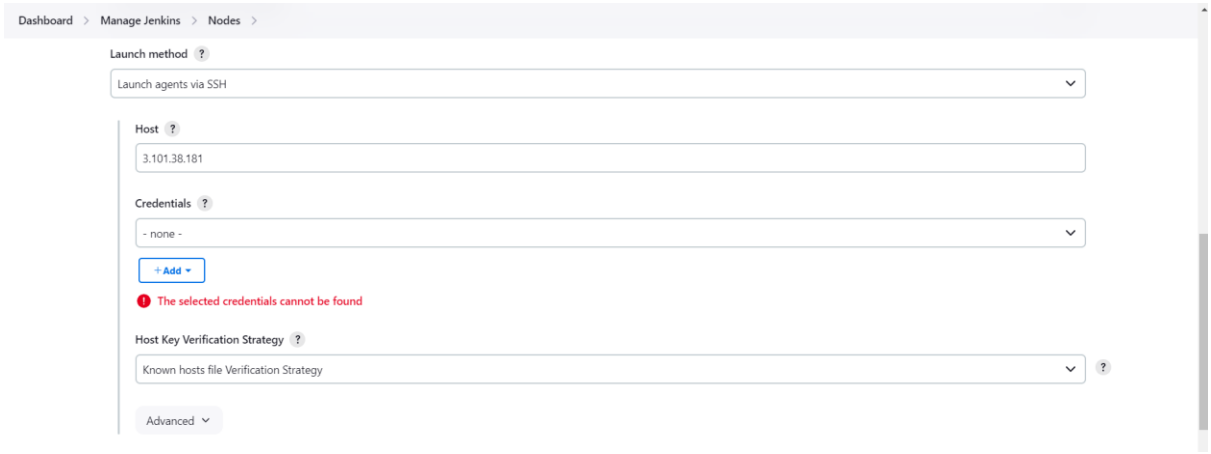
- If you want Build Executors extra increase.
- Remote root directory ---- give path --/home/ec2-user
- Label as Java



Launch method – launch agents via SSH

Host --- give here you slave IP

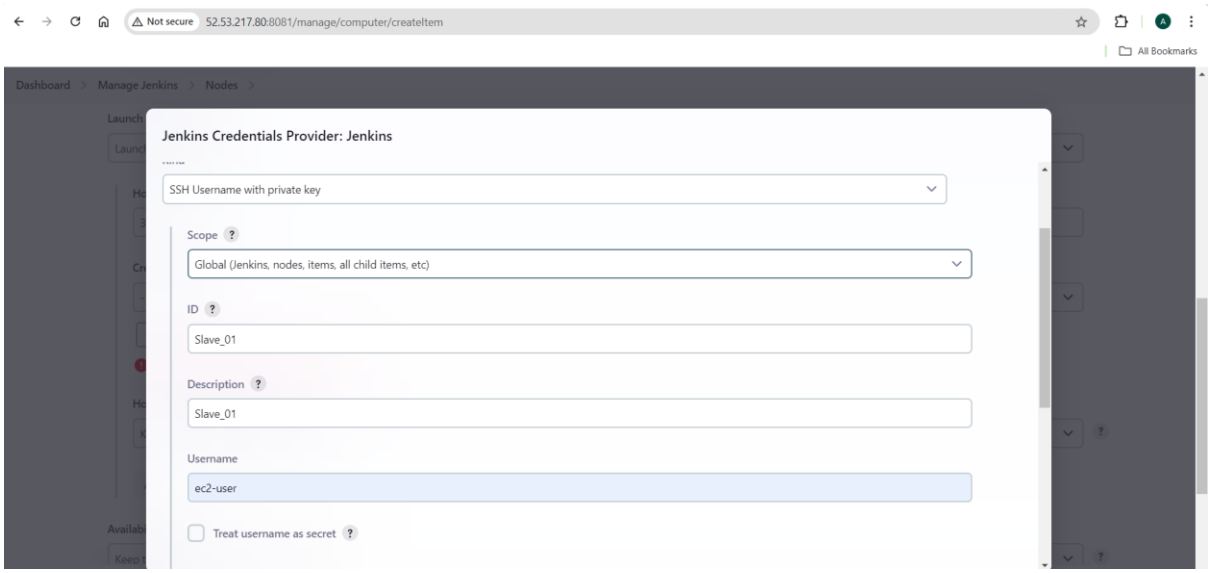
Under the credentials click on add.



Select ---- SSH username with Private key.

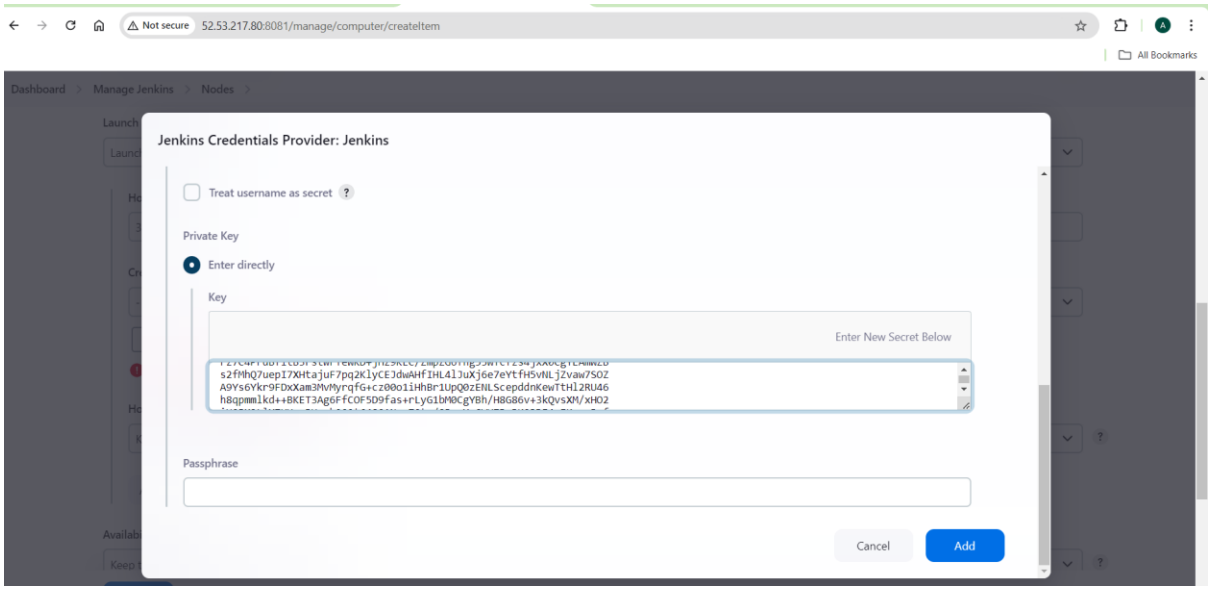
Scope ---- Global and Give description.

User name --- ec2-user



Here click on enter directly and paste the pem key.

And click on add.

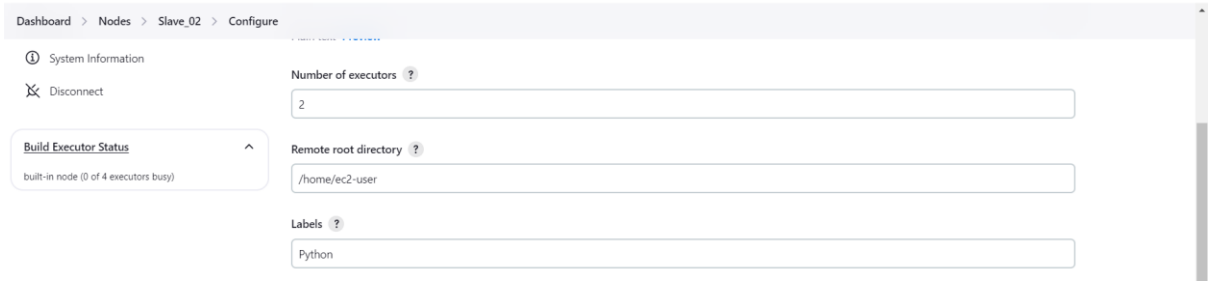


After that select the select credential and save.

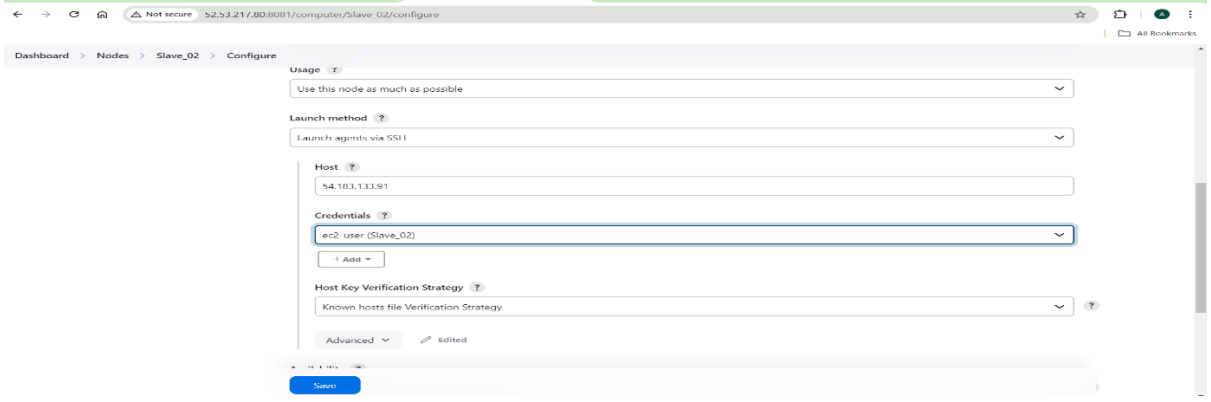
As it’s create the second slave node. but need give some changes.

Give name --- Slave-02

Label is --- python

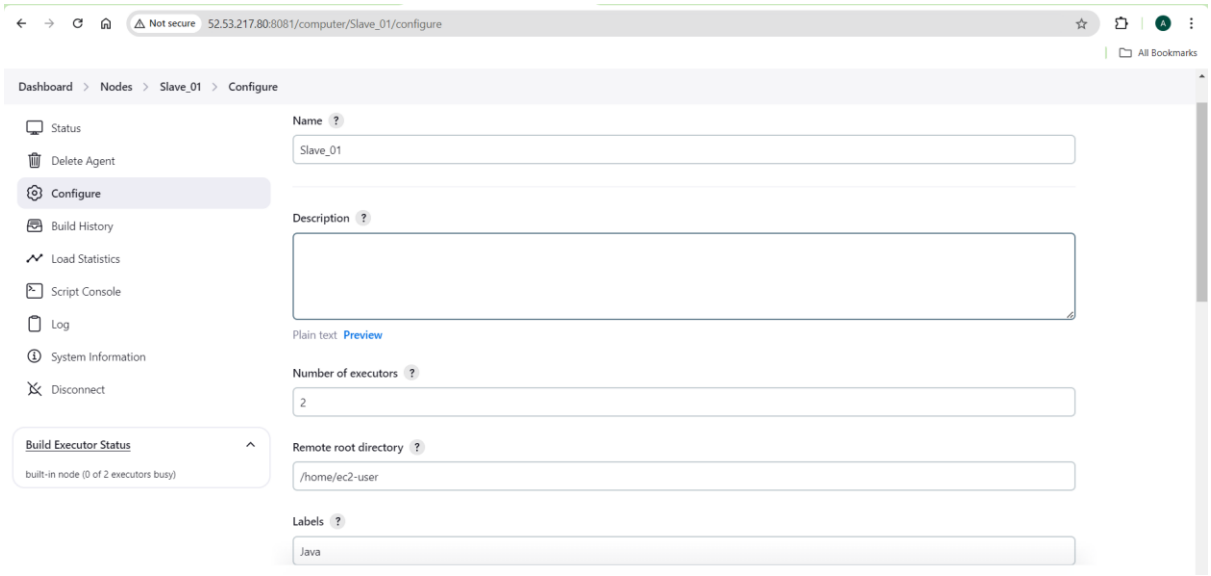


Give credential of slave Node and click on save.



Now go to jobs and attach labels.

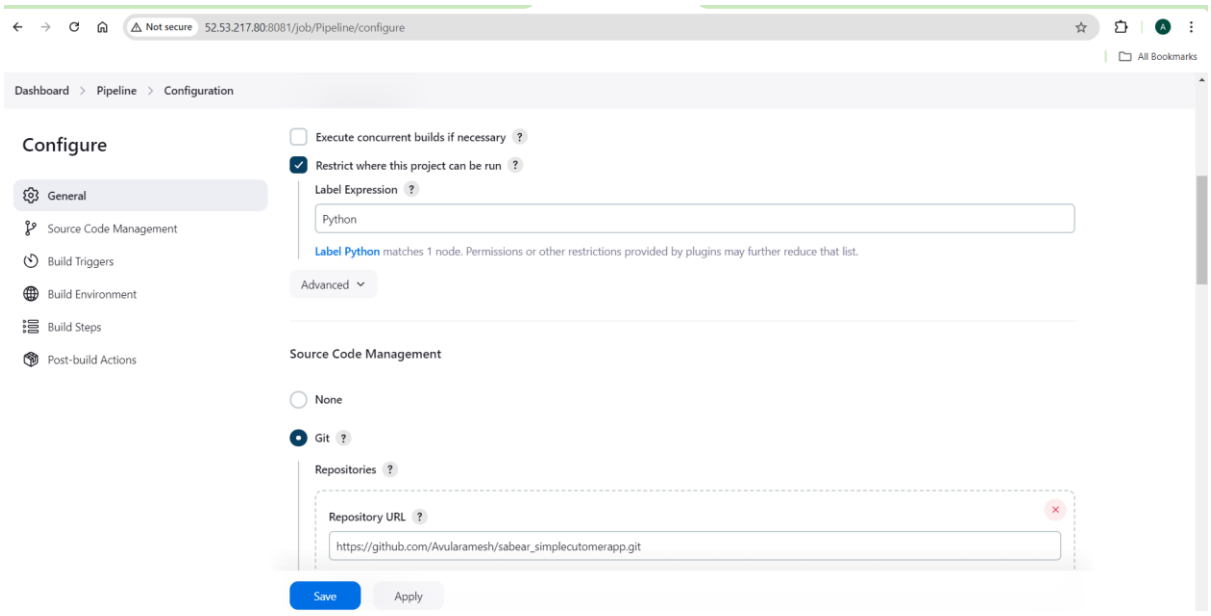
The below one is freestyle job attached the Java label.



Just click on save.

Next go to the another Job.

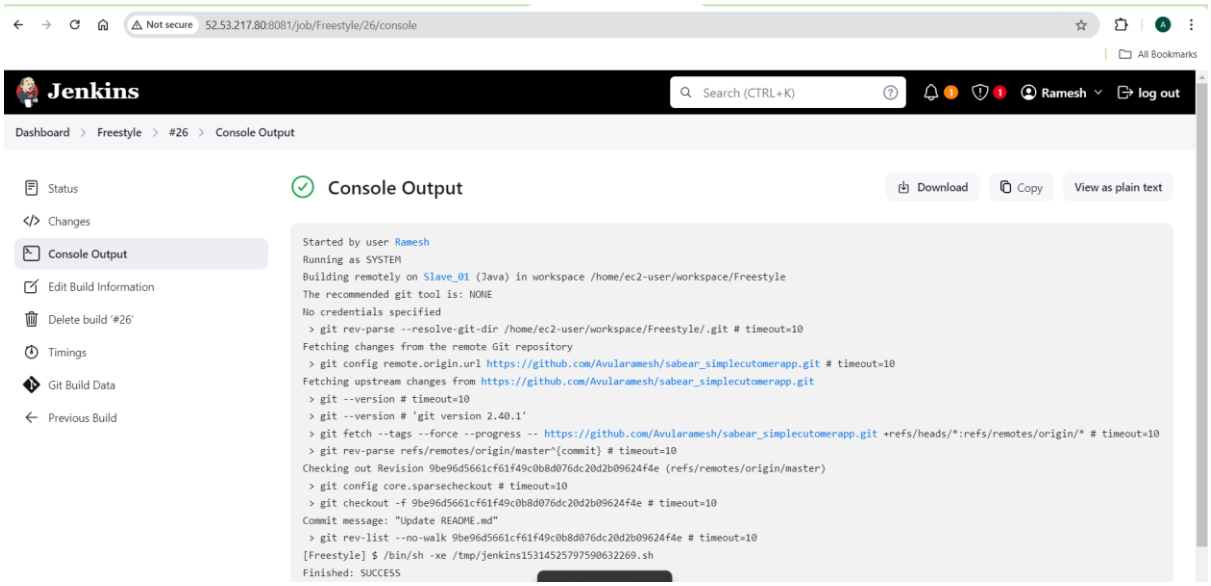
The below one is Pipeline job attached the python label.



Make the branches are main only.

Now try to build jobs.

The below one is Freestyle Job build in slave_01

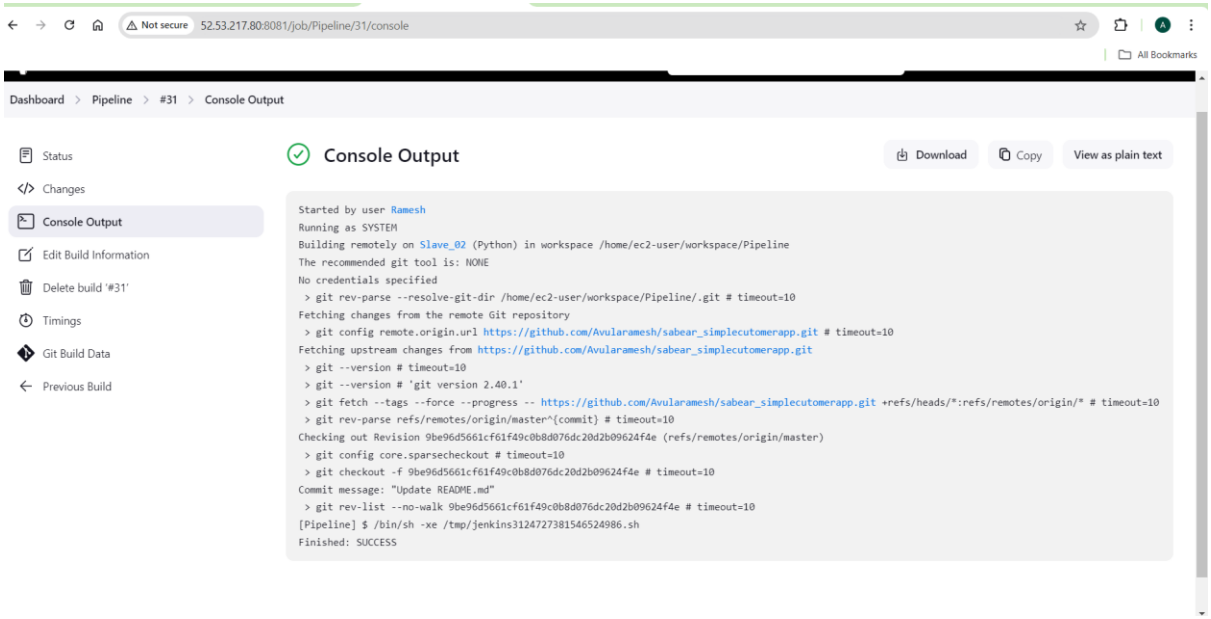


Here I went slave node.

Here my code is there.

```
[ec2-user@Slave_01 Freestyle]$ pwd
/home/ec2-user/workspace/Freestyle
[ec2-user@Slave_01 Freestyle]$ ls
build  build.xml  Jenkinsfile  pom.xml  README.md  Service-API.dockerfile  sonar-project.properties  src  WebContent  XMLSave.txt
[ec2-user@Slave_01 Freestyle]$
```

The below one is Pipeline Job build in slave_02



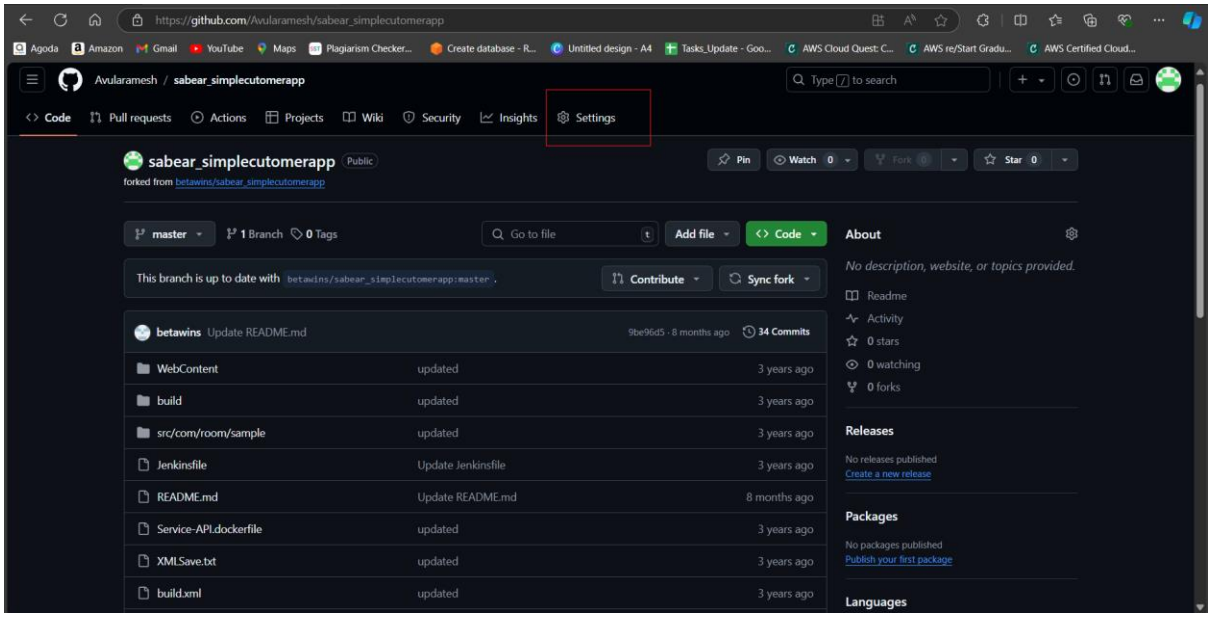
Here my code is there.

```
[ec2-user@Slave_02 Pipeline]$ pwd
/home/ec2-user/workspace/Pipeline
[ec2-user@Slave_02 Pipeline]$ ls
build  build.xml  Jenkinsfile  pom.xml  README.md  Service-API.dockerfile  sonar-project.properties  src  WebContent  XMLSave.txt
[ec2-user@Slave_02 Pipeline]$
```

##########Task is Done#######

2) Configure webhooks to jenkins job.

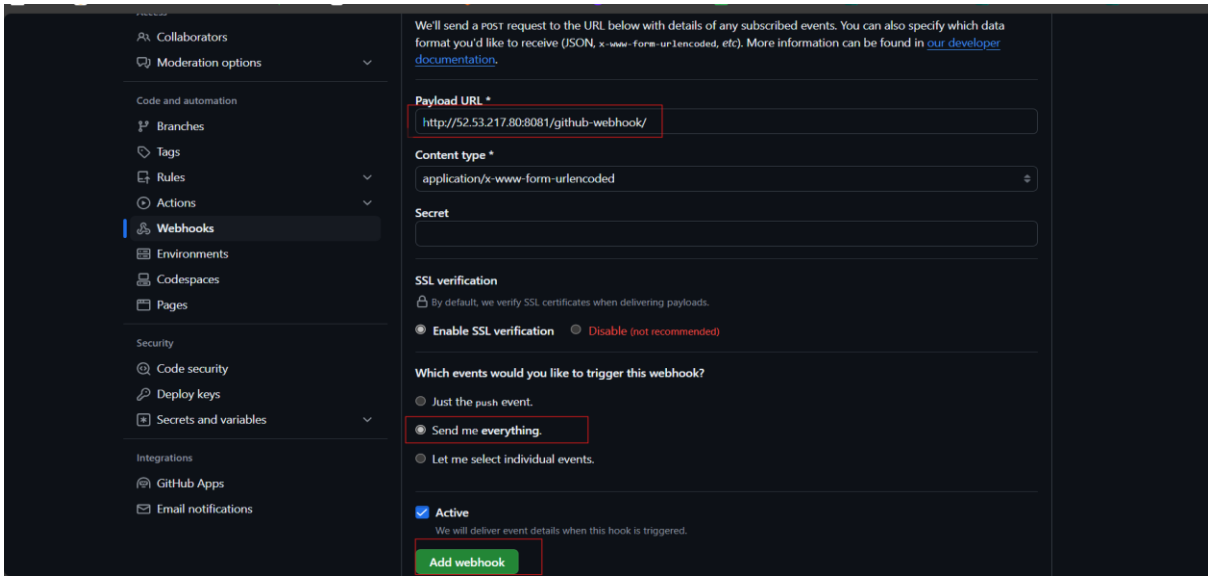
To create webhooks



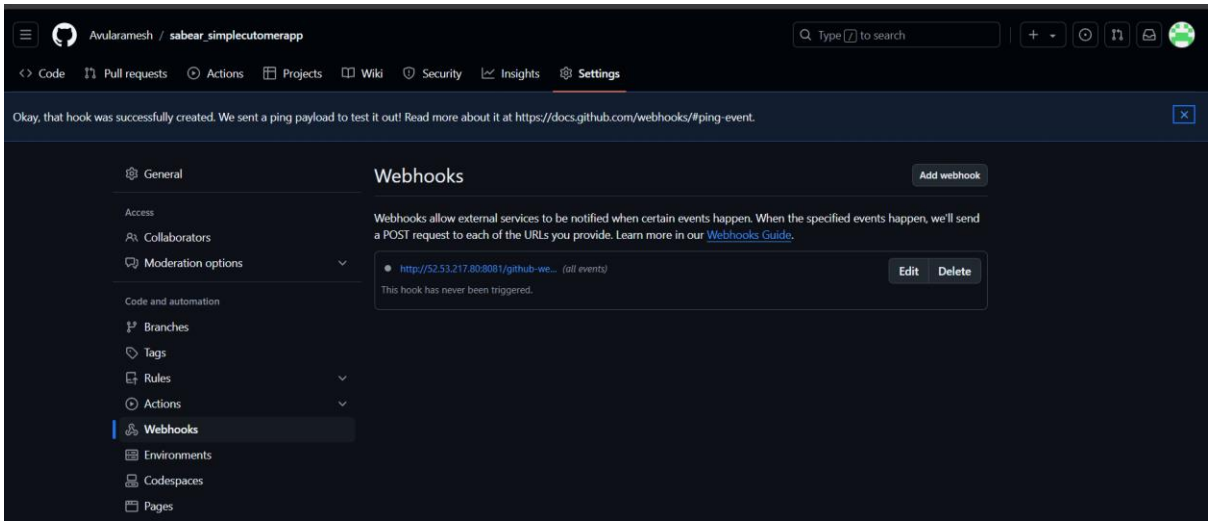
Here Payload URL ---- /Jenkins-url/github-webhook/

I am selecting the – send me every thing.

And click on Add webhook.

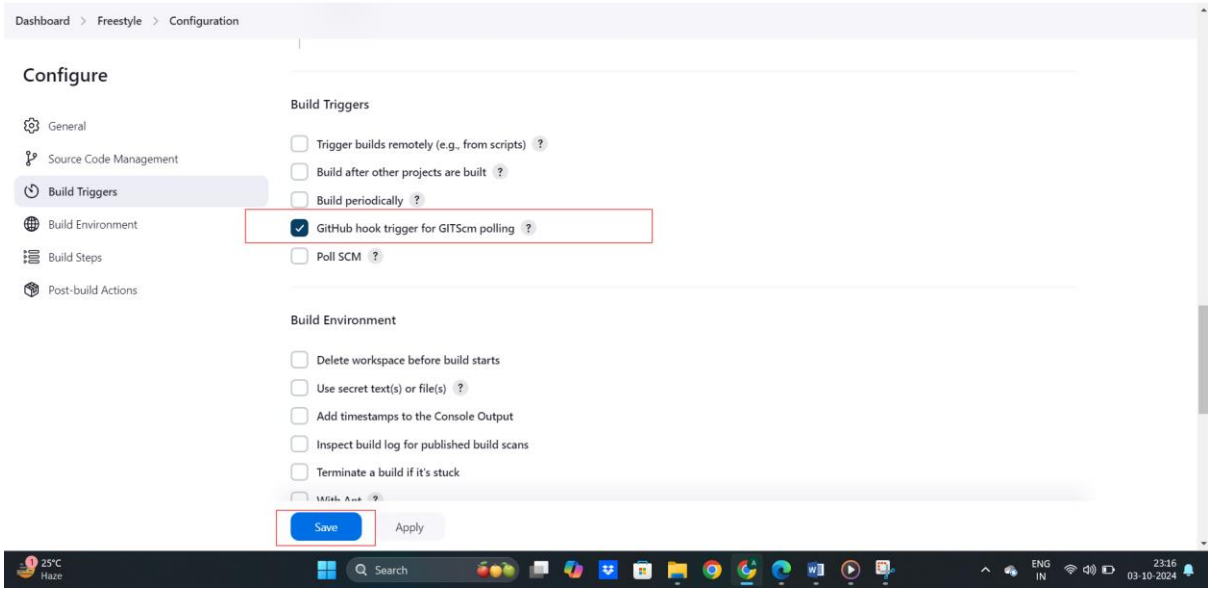


Here webhooks is created.

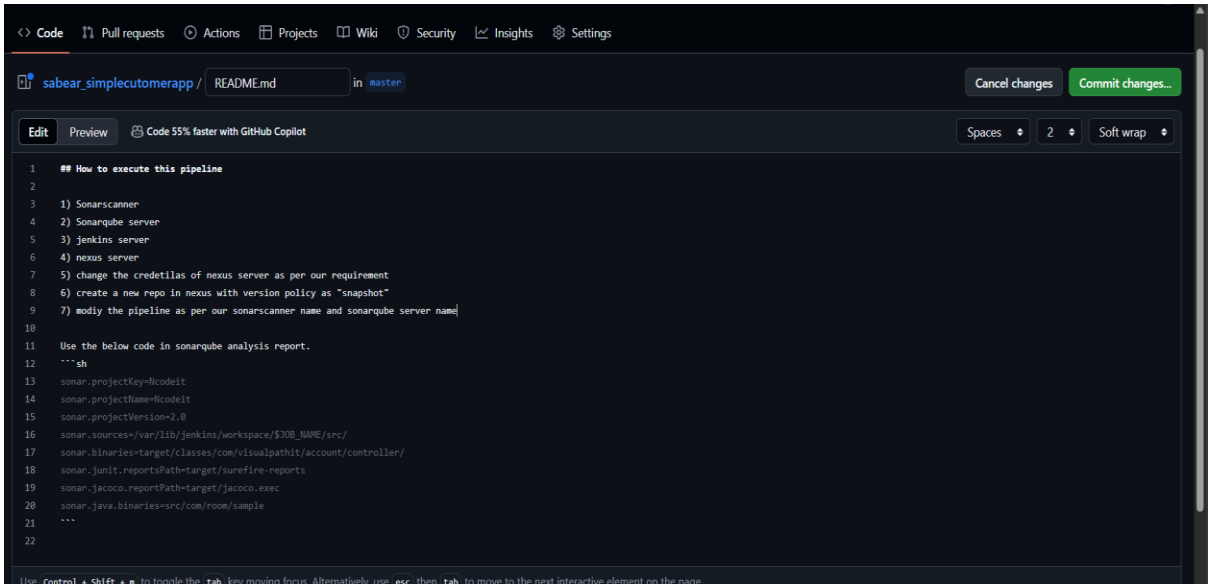


Now go to the Jenkins Job.

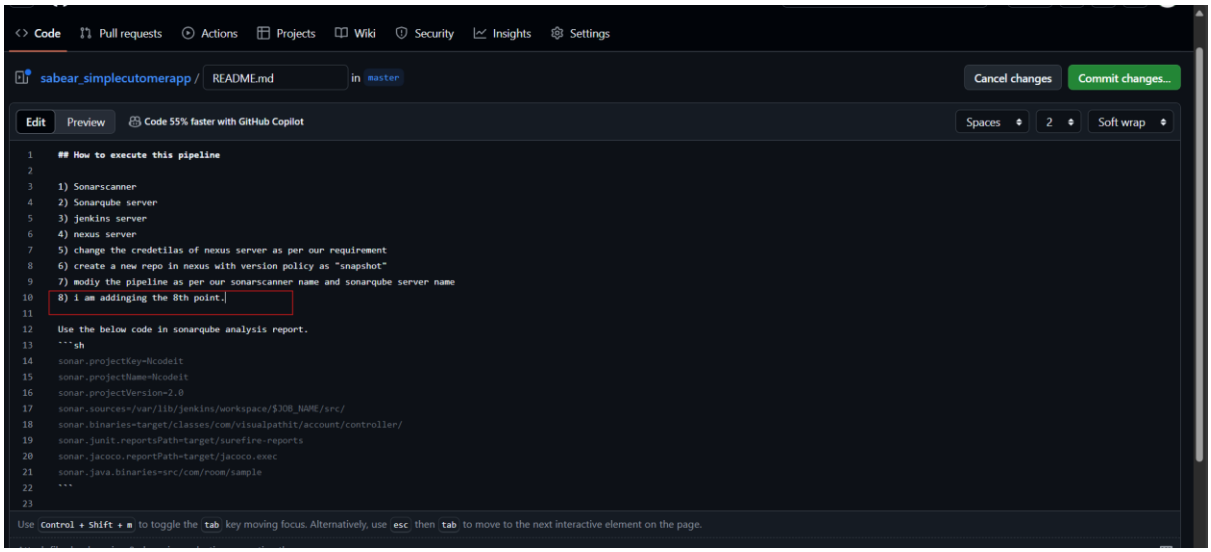
Click on these check box and save.



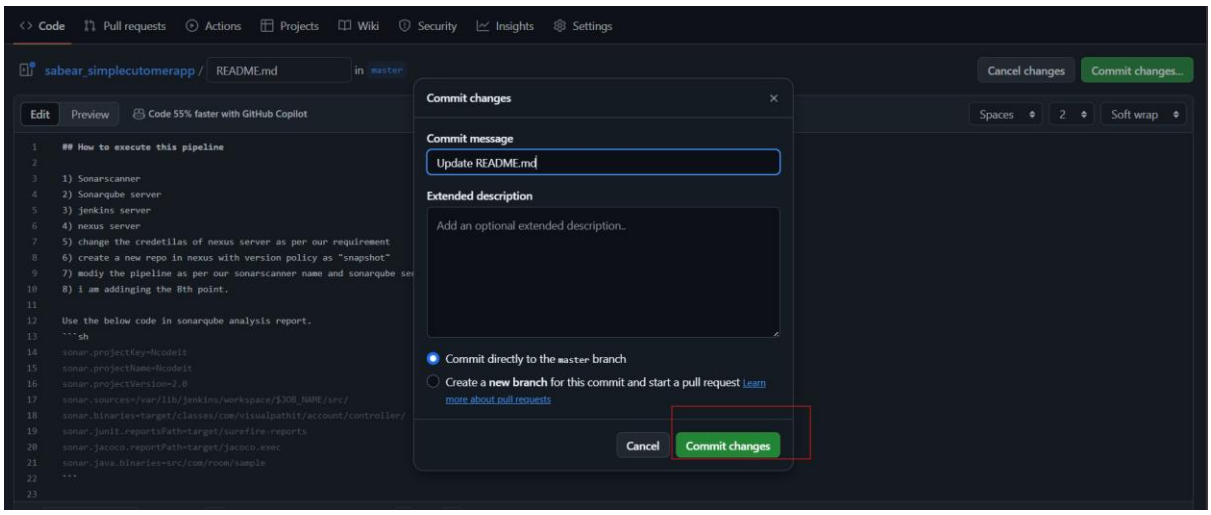
Actually in these readme.md file 7 points is there I am adding the 8 point.



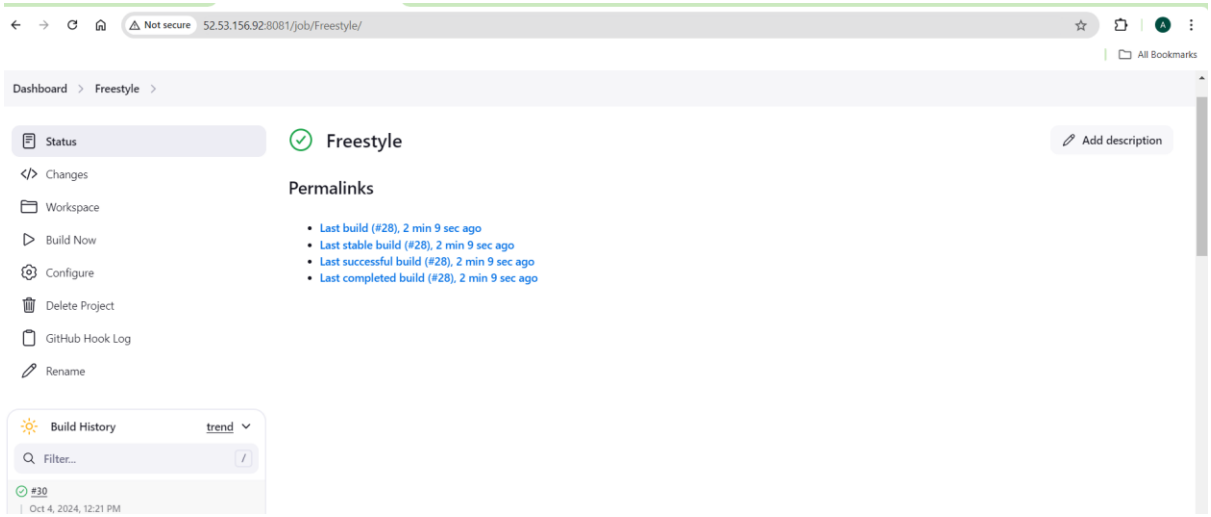
Here I am adding the new point.



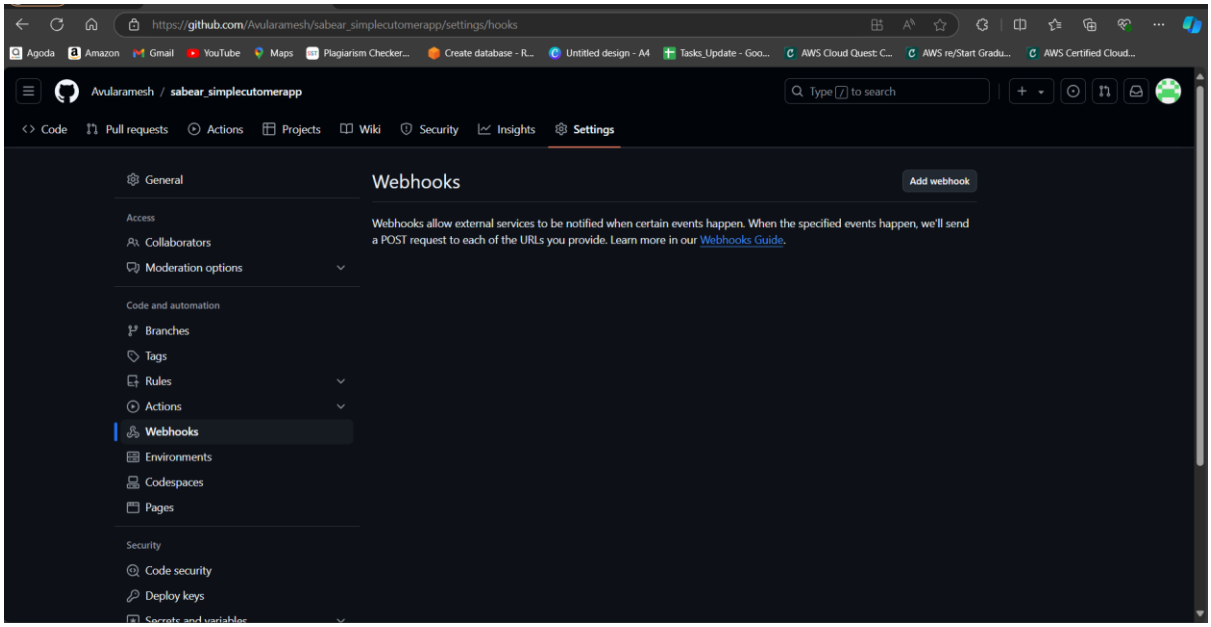
Click on commit changes.



Here automatically build job.



After that I am deleted the webhook.

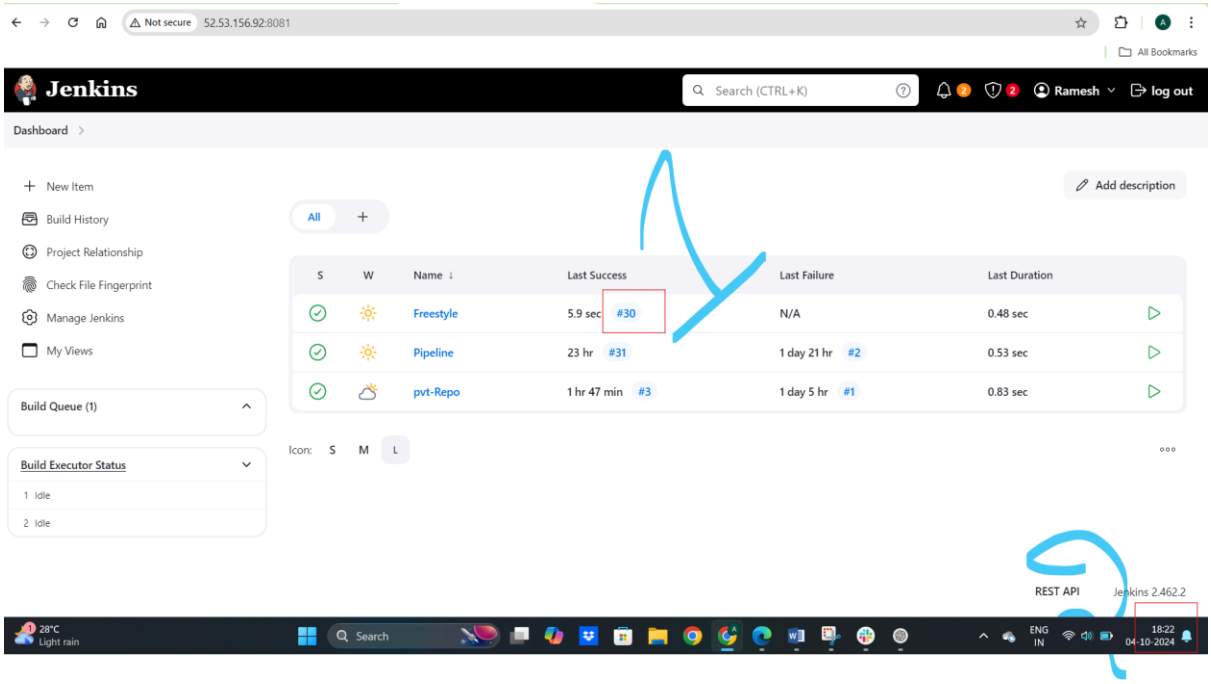


##############Task is Done############

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

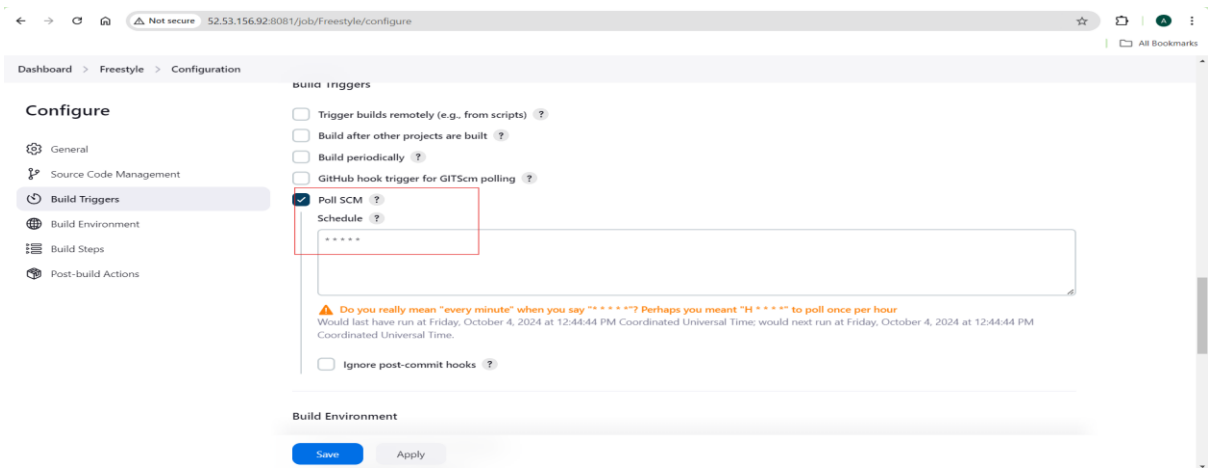
- **Poll SCM**: This option makes Jenkins regularly check the source code repository for any changes (like new commits). If Jenkins finds changes, it will trigger a build.
- It's useful when you only want to build the project when the code is updated.
- **Build Periodically**: This option tells Jenkins to run the job at regular intervals, no matter if there are code changes or not. It doesn't check for commits;
- it just runs the job on a schedule (like daily or weekly builds).

  See the below how many builds happen and what time.



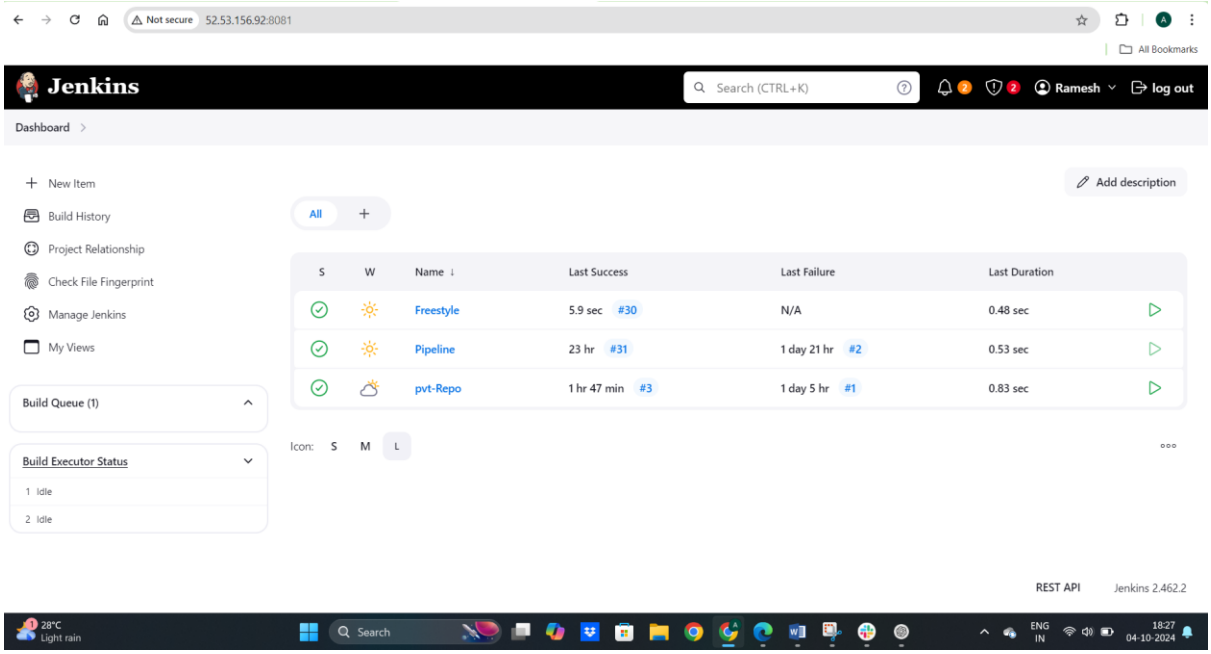
Now we Go to the Job and select **Poll SCM**

* * * * * --- This expression will execute the job every minute of every day.

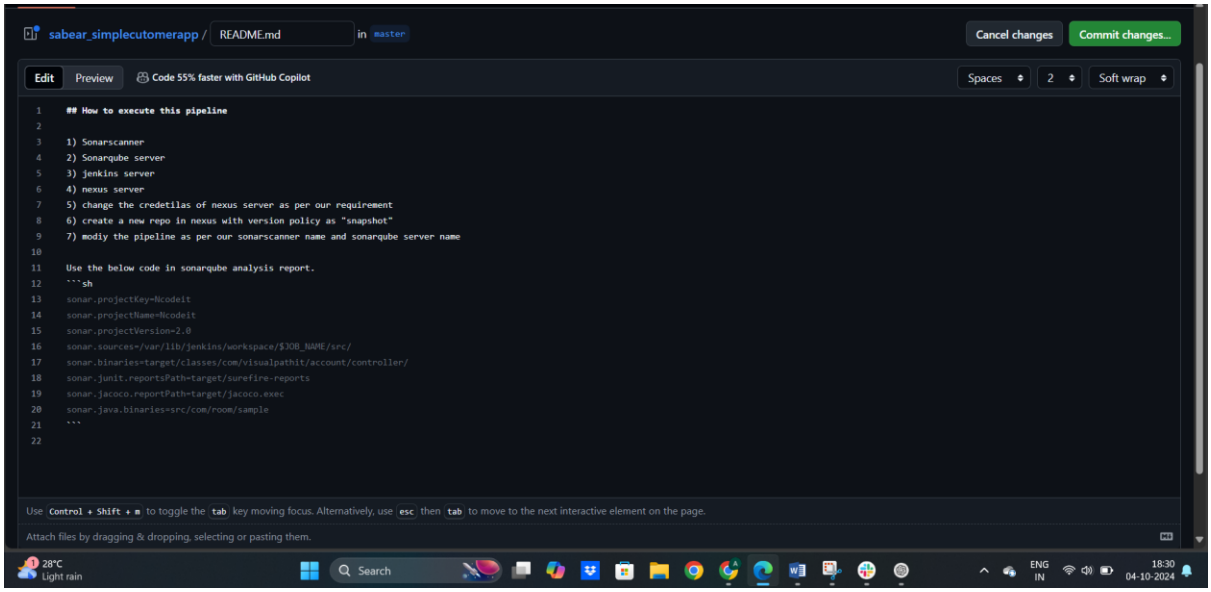


After that I am click on save.

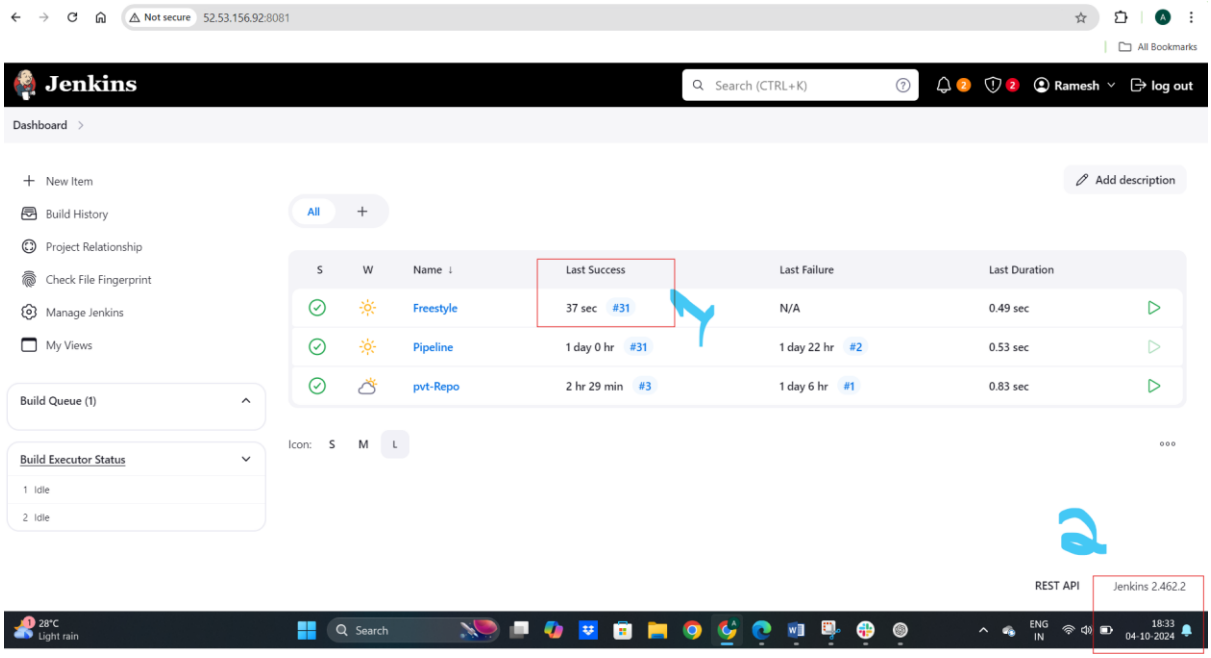
Here we haven’t seen the any build’s because in the repo no actions that’s why.



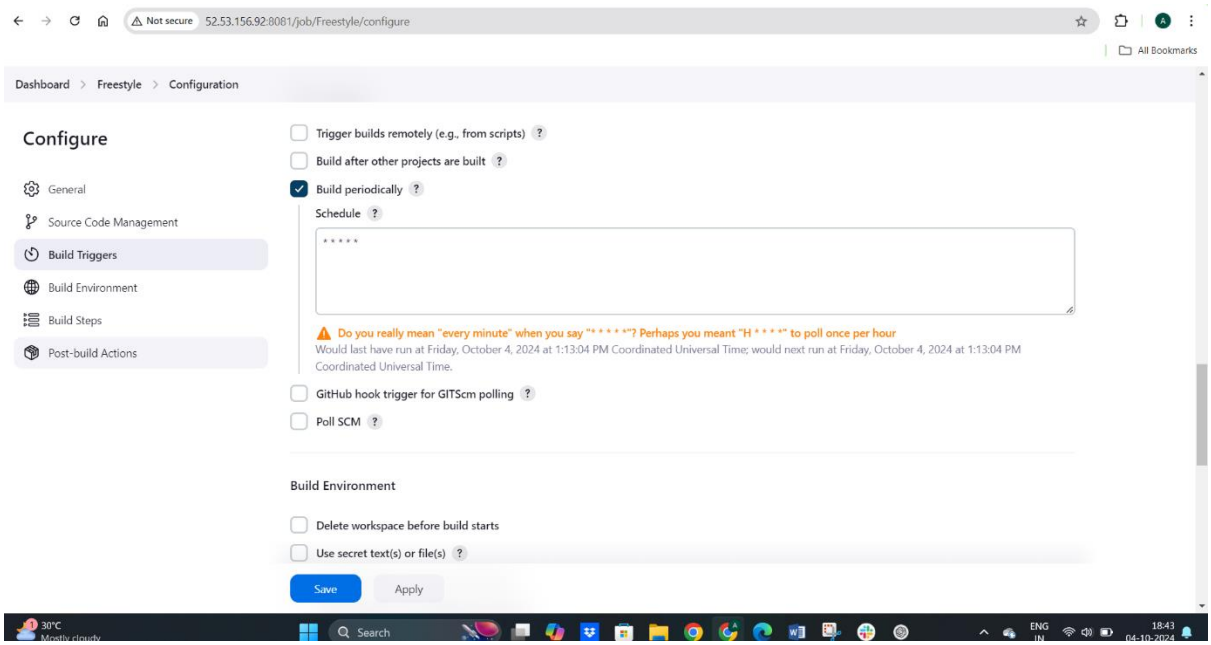
So now I went to the github repo and change the code then automatically build.



As we discuss the job build.

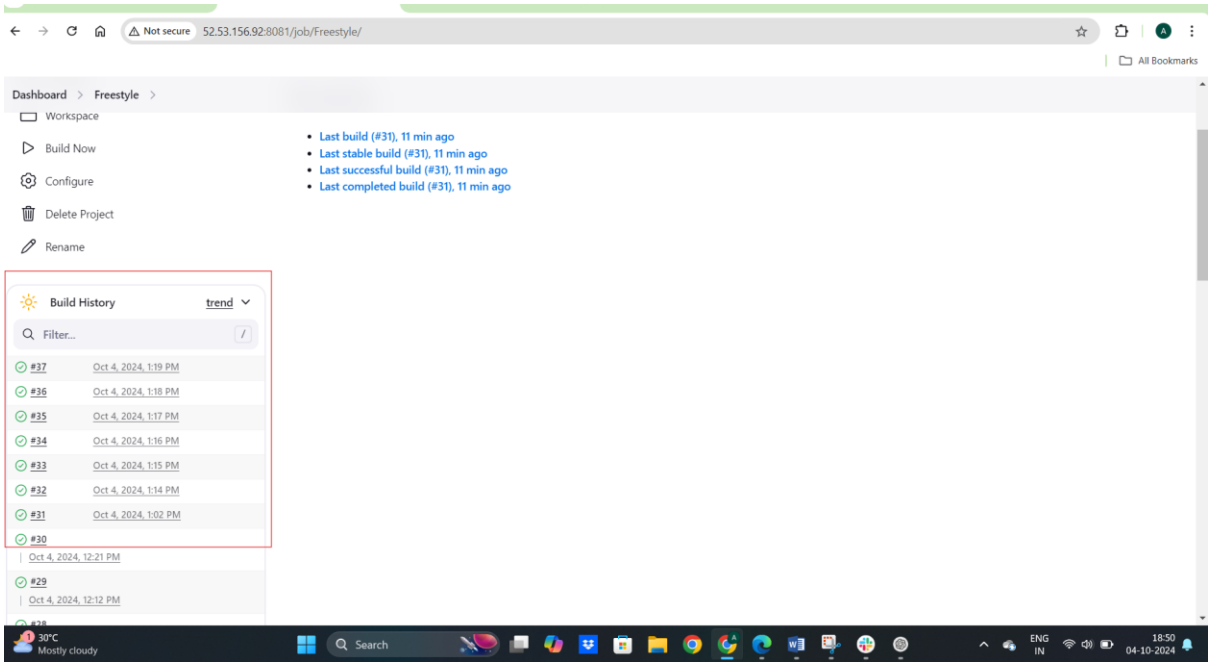


Now went to Job and change – Build Periodically



Now here you see the After selecting these build Periodically.

It will build the job every minute.

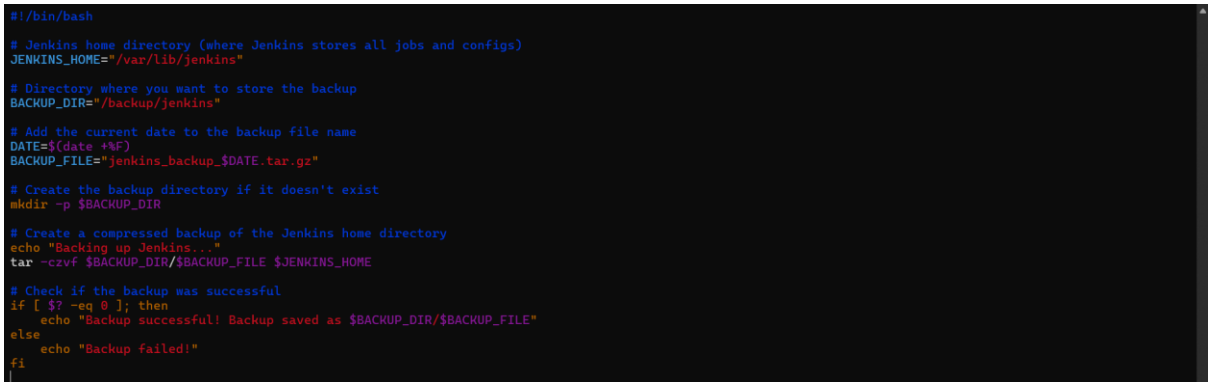


##########Task is Done#########

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

I went to the Jenkins server and there I am creating the on file --- Script_for_Jenkins_Backup

I went to these file and Add the script.



```
#!/bin/bash

# Jenkins home directory (where Jenkins stores all jobs and configs)
JENKINS_HOME="/var/lib/jenkins"

# Directory where you want to store the backup
BACKUP_DIR="/backup/jenkins"

# Add the current date to the backup file name
DATE=$(date +%F)
BACKUP_FILE="jenkins_backup_$DATE.tar.gz"

# Create the backup directory if it doesn't exist
mkdir -p $BACKUP_DIR

# Create a compressed backup of the Jenkins home directory
echo "Backing up Jenkins..."
tar -czvf $BACKUP_DIR/$BACKUP_FILE $JENKINS_HOME

# Check if the backup was successful
if [ $? -eq 0 ]; then
    echo "Backup successful! Backup saved as $BACKUP_DIR/$BACKUP_FILE"
else
    echo "Backup failed!"
fi
```

After change the permissions of the file and Run the file.

```
[root@Jenkins_Master ~]# ls
Script_for_Jenkins_Backup
[root@Jenkins_Master ~]# chmod 777 Script_for_Jenkins_Backup
[root@Jenkins_Master ~]# ls
Script_for_Jenkins_Backup
[root@Jenkins_Master ~]# ./Script_for_Jenkins_Backup
Backing up Jenkins...
tar: Removing leading `/' from member names
/var/lib/jenkins/
/var/lib/jenkins/%C/
/var/lib/jenkins/%C/jenkins/
/var/lib/jenkins/%C/jenkins/war/
/var/lib/jenkins/%C/jenkins/war/META-INF/
/var/lib/jenkins/%C/jenkins/war/META-INF/MANIFEST.MF
/var/lib/jenkins/%C/jenkins/war/META-INF/JENKINS.SF
/var/lib/jenkins/%C/jenkins/war/META-INF/JENKINS.RSA
/var/lib/jenkins/%C/jenkins/war/META-INF/maven/
/var/lib/jenkins/%C/jenkins/war/META-INF/maven/org.jenkins-ci.main/
/var/lib/jenkins/%C/jenkins/war/META-INF/maven/org.jenkins-ci.main/jenkins-war/
/var/lib/jenkins/%C/jenkins/war/META-INF/maven/org.jenkins-ci.main/jenkins-war/pom.xml
/var/lib/jenkins/%C/jenkins/war/META-INF/maven/org.jenkins-ci.main/jenkins-war/pom.properties
/var/lib/jenkins/%C/jenkins/war/WEB-INF/
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/META-INF/
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/META-INF/licenses.html
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/META-INF/licenses.xml
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/dependencies.txt
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/LICENCE
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/add.svg
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/analytics.svg
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/arrow-left.svg
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/arrow-right.svg
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/arrow-up.svg
/var/lib/jenkins/%C/jenkins/war/WEB-INF/classes/images/symbols/browsers.svg
```

The below we can see image Backup successful.

```
/var/lib/jenkins/workspace/pvt-Repo/.git/logs/refs/remotes/
/var/lib/jenkins/workspace/pvt-Repo/.git/logs/refs/remotes/origin/
/var/lib/jenkins/workspace/pvt-Repo/.git/logs/refs/remotes/origin/main
/var/lib/jenkins/workspace/pvt-Repo/.git/logs/HEAD
/var/lib/jenkins/workspace/pvt-Repo/.git/index
/var/lib/jenkins/workspace/pvt-Repo/Java_app_3.0-main.zip
/var/lib/jenkins/workspace/pvt-Repo@tmp/
/var/lib/jenkins/queue.xml
/var/lib/jenkins/hudson.plugins.emailext.ExtendedEmailPublisher.xml
/var/lib/jenkins/jenkins.security.ResourceDomainConfiguration.xml
/var/lib/jenkins/jenkins.metrics.api.MetricsAccessKey.xml
/var/lib/jenkins/org.jenkinsci.plugins.workflow.flow.GlobalDefaultFlowDurabilityLevel.xml
/var/lib/jenkins/com.cloudbees.hudson.plugins.folder.config.AbstractFolderConfiguration.xml
/var/lib/jenkins/hudson.plugins.build_timeout.global.GlobalTimeOutConfiguration.xml
/var/lib/jenkins/hudson.plugins.gradle.enriched.EnrichedSummaryConfig.xml
/var/lib/jenkins/hudson.plugins.gradle.injection.InjectionConfig.xml
/var/lib/jenkins/hudson.plugins.timestamper.TimestamperConfig.xml
/var/lib/jenkins/io.jenkins.plugins.junit.storage.JunitTestResultStorageConfiguration.xml
/var/lib/jenkins/jenkins.model.ArtifactManagerConfiguration.xml
/var/lib/jenkins/org.jenkinsci.plugins.displayurlapi.DefaultDisplayURLProviderGlobalConfiguration.xml
/var/lib/jenkins/github-plugin-configuration.xml
/var/lib/jenkins/org.jenkinsci.plugins.github_branch_source.GitHubConfiguration.xml
/var/lib/jenkins/org.jenkinsci.plugins.workflow.libs.GlobalLibraries.xml
/var/lib/jenkins/org.jenkinsci.plugins.workflow.libs.GlobalUntrustedLibraries.xml
/var/lib/jenkins/hudson.plugins.build_timeout.operations.BuildStepOperation.xml
Backup successful! Backup saved as /backup/jenkins/jenkins_backup_2024-10-04.tar.gz
```

Go to the dir and check file there or not.

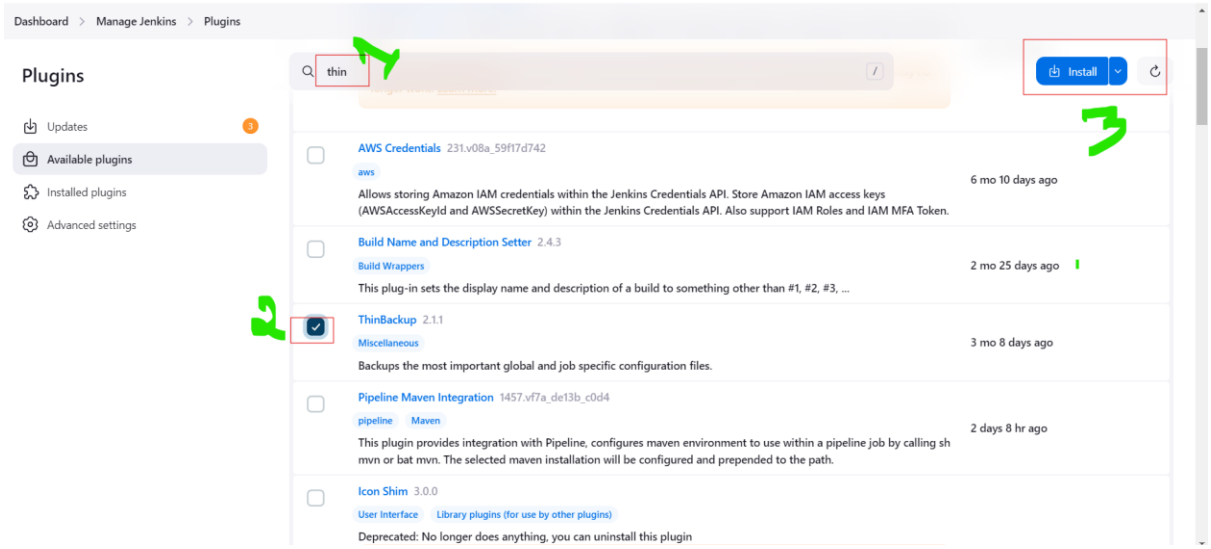
```
[root@Jenkins_Master ~]# find / -name "jenkins_backup_2024-10-04.tar.gz"
/backup/jenkins/jenkins_backup_2024-10-04.tar.gz
[root@Jenkins_Master ~]# cd /backup/jenkins/jenkins_backup_2024-10-04.tar.gz
-bash: cd: /backup/jenkins/jenkins_backup_2024-10-04.tar.gz: Not a directory
[root@Jenkins_Master ~]# cd /backup/jenkins
[root@Jenkins_Master jenkins]# ls
jenkins_backup_2024-10-04.tar.gz
[root@Jenkins_Master jenkins]# pwd
/backup/jenkins
```

The Backup file is Available.

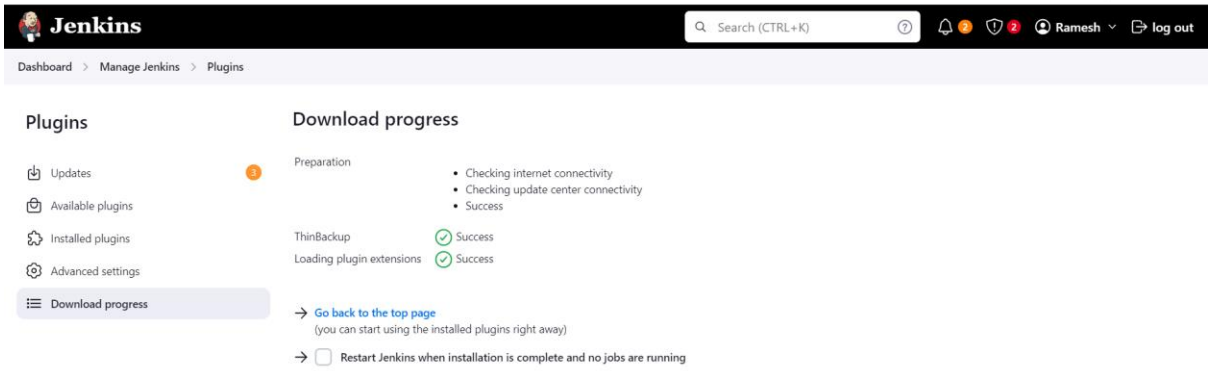
#######Task is Done############

5) Take backup of jenkins using thin backup plugin.

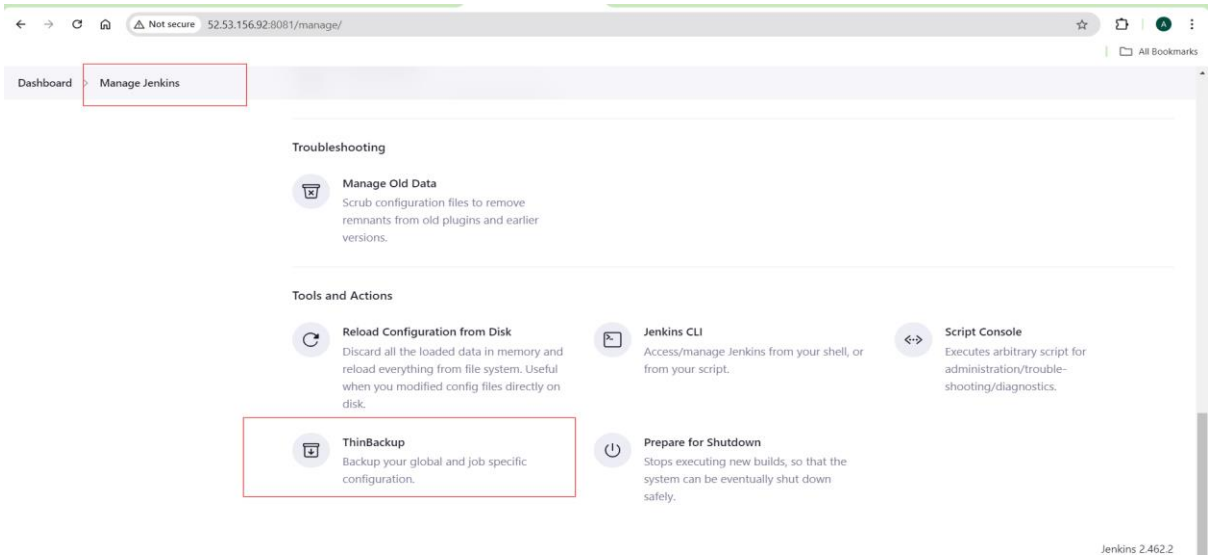
- To Take backup of jenkins using thin backup plugin first we need to install the thin Plugin.
- Go to manage Jenkins, Plugins, Available Plugins.
- Search for the thinBackup.



Thin backup is installed.



Now Go to the manage Jenkins and scroll down you see the thinbackup.



And click on ThinkBackup.

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

Here I am in Jenkins master Server our Backup file is Available.

```
[root@Jenkins_Master ~]# cd /backup/jenkins
[root@Jenkins_Master jenkins]# ls
jenkins_backup_2024-10-04.tar.gz
[root@Jenkins_Master jenkins]#
```

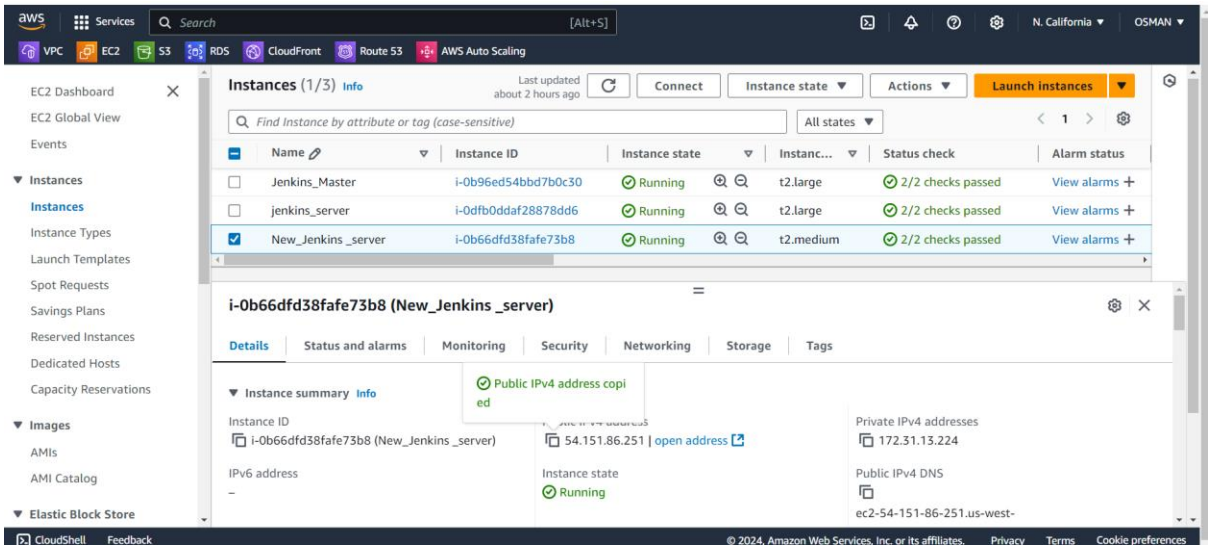
Extract these file—using tar –xvf <tarfile name>

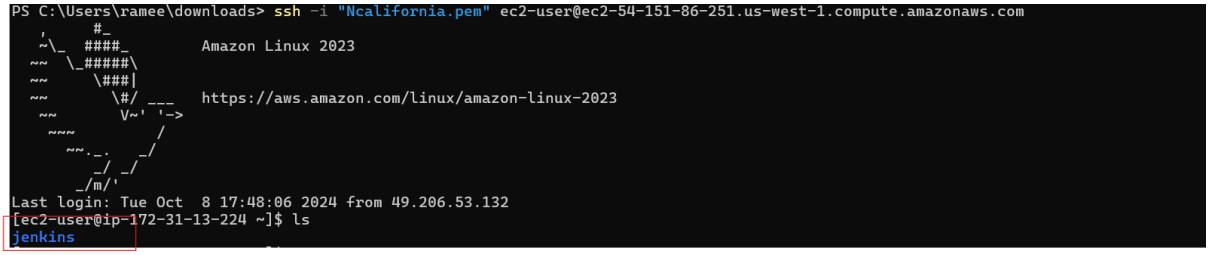
After that Jenkins file move to /home/ec2-user/ loation

Here to transfer the Jenkins folder to new sever using the below command.

```
[ec2-user@Jenkins_Master ~]$ sudo su -
Last login: Tue Oct  8 16:22:37 UTC 2024 on pts/3
[root@Jenkins_Master ~]# cd /backup/jenkins/var/lib
[root@Jenkins_Master lib]# ls
jenkins
[root@Jenkins_Master lib]# cp jenkins /home/ec2-user
cp: omitting directory 'jenkins'
[root@Jenkins_Master lib]# cp -r jenkins /home/ec2-user/
[root@Jenkins_Master lib]# cd
[root@Jenkins_Master ~]# exit
logout
[ec2-user@Jenkins_Master ~]$ ls
jenkins  Ncalifornia.pem  Thin_Backup_file
[ec2-user@Jenkins_Master ~]$ sudo scp -i /home/ec2-user/Ncalifornia.pem -r jenkins ec2-user@54.151.86.251:/home/ec2-user/
MANIFEST.MF                                                                                   100%  127KB  23.8MB/s   00:00
JENKINS.SF                                                                                    100%  126KB  29.6MB/s   00:00
JENKINS.RSA                                                                                   100%   12KB   7.9MB/s   00:00
pom.xml                                                                                       100% 5049     2.4MB/s   00:00
pom.properties                                                                                100%   67    23.4KB/s   00:00
licenses.html                                                                                 100%   28KB   9.4MB/s   00:00
licenses.xml                                                                                  100%   33KB   7.1MB/s   00:00
dependencies.txt                                                                              100% 8335     2.5MB/s   00:00
LICENCE                                                                                       100% 1099   460.4KB/s   00:00
add.svg                                                                                       100%  237    97.7KB/s   00:00
analytics.svg                                                                                 100%  793   240.4KB/s   00:00
arrow-left.svg                                                                                100%  256    89.5KB/s   00:00
arrow-right.svg                                                                               100%  259    88.8KB/s   00:00
arrow-up.svg                                                                                  100%  254    69.5KB/s   00:00
browsers.svg                                                                                  100%  407   104.6KB/s   00:00
brush-outline.svg                                                                             100%  490   148.2KB/s   00:00
build-history.svg                                                                             100%  502   138.6KB/s   00:00
build-steps.svg                                                                               100% 4227     1.1MB/s   00:00
```

Go to the new server check there Jenkins is available.





Here you need to install the java.

CMD: sudo yum –y install java-17*

Next install Jenkins.

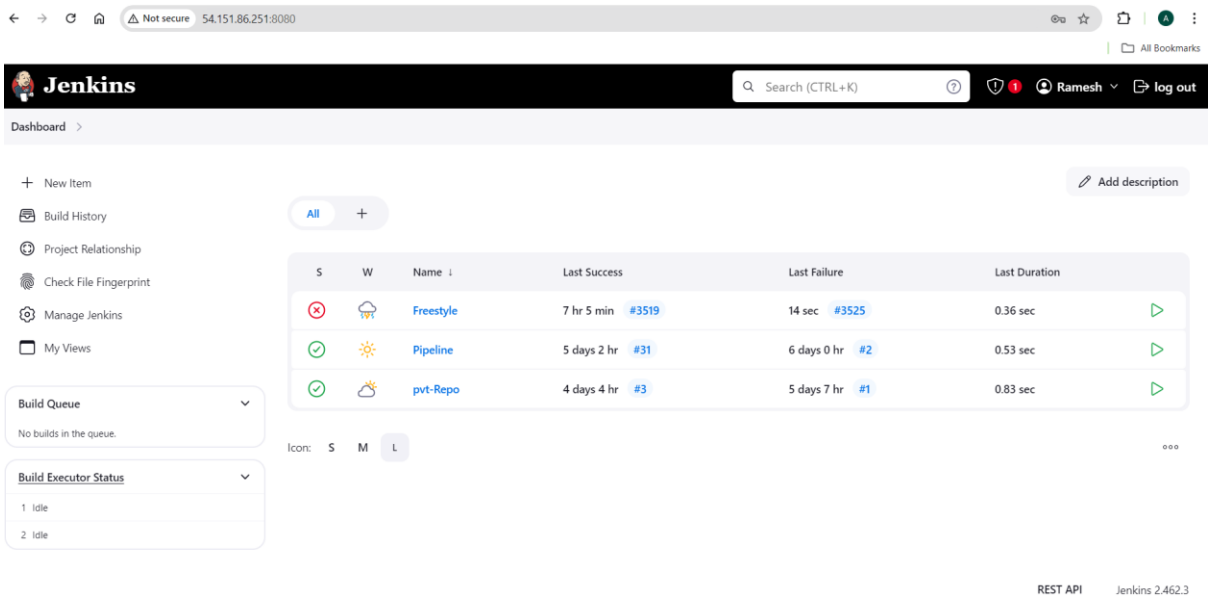
Go the cd /var/lib --- there you need to delete the Jenkins folder

Next remove the Jenkins folder ----- sudo rm –rf Jenkins

CMD : sudo cp –r Jenkins /var/lib ---- move to the Jenkins folder to /var/lib Location.

After that check the files.

Go to the browser and browse.



Here my data is Available Task is Done