**Assignments**

**Jenkins**

**=======**

1) Jenkins Port Change

2) How to change Jenkins Home Directory

3) How to take backup in Jenkins and Restore.

4) How to install Jenkins in Tomcat

5) Remote Job in Jenkins

6) Webhook through Git

7) Jenkins Installation Script - Pending

8) Install old version of Jenkins and then upgrade it to a latest version

9) How to put https in Jenkins URL (Open Cert option)

**1) Jenkins Port Change - Done**

**Solution:**

hostnamectl set-hostname **master** - create hostname permanent

#!/bin/bash

apt update -y && apt upgrade -y && apt install openjdk-17-jdk -y

apt update -y && apt upgrade -y && apt install openjdk-17-jdk -y

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt update -y

sudo apt install jenkins -y

systemctl start jenkins

cat /var/lib/jenkins/secrets/initialAdminPassword

start Jenkins – public ip:8080

goto > /etc/default/jenkins

Environment="JENKINS\_PORT=8080" to 9090

No goto > /usr/lib/systemd/system/jenkins.service

change the port in the line

Environment="JENKINS\_PORT=8080" to 9090

Save.

systemctl daemon-reload

systemctl restart jenkins

check.

Manage Jenkins > System > Jenkins Location > Jenkins URL > change the port no and save

Close Jenkins and Restart Jenkins Service. Port change done!

**2) How to change Jenkins Home Directory – Partial Done**

**Solution:**

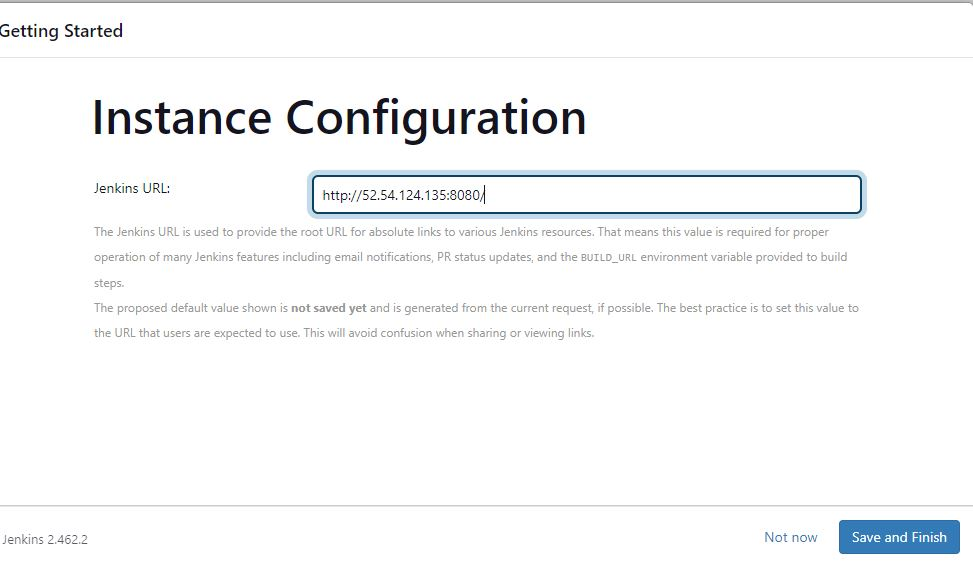
Follow same steps of Assignment 1 to launch jenkins

In CLI: check if jenkins is up and running

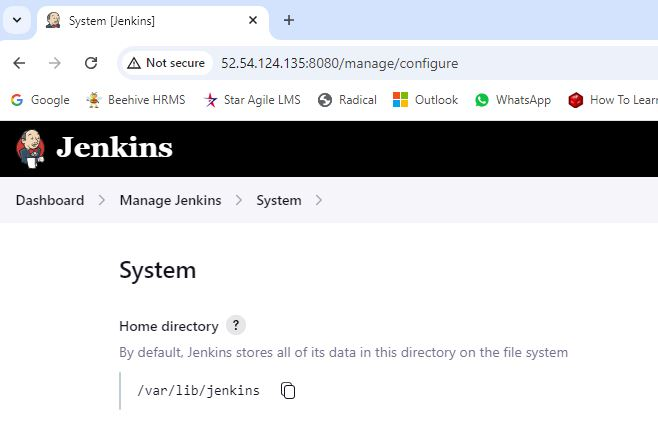
jenkins --version

systemctl status jenkins

Browser: public ip:8080 > Install jenkins and goto dashboard



Manage Jenkins > System > check the home path



Close jenkins browser

Stop jenkins service

* Systemctl stop jenkins

Create a new Jenkins Home directory using the mkdir command

* mkdir /home/jenkins\_home

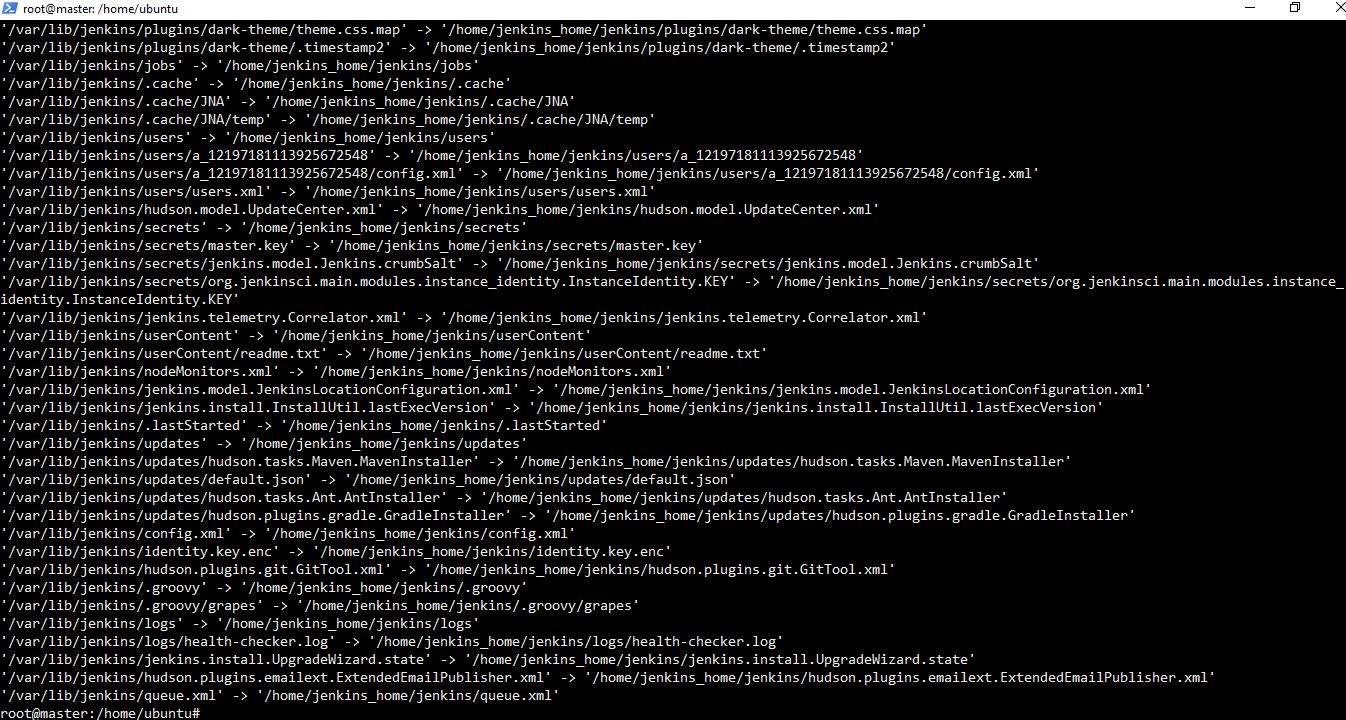
Set ownership to this directory

* chown jenkins:jenkins /home/jenkins\_home

[The chown command in Linux is used to assign ownership. Every file and directory can have only one user and one group as an owner.]

Copy the contents from old jenkins directory to new jenkins directory

* cp -prv /var/lib/jenkins /home/jenkins\_home



[In Linux, the cp command is used to copy files and directories. The options -p, -r, and -v are flags that modify the behavior of the cp command:

* -p (preserve): This option preserves the file attributes such as mode, ownership, and timestamps.
* -r or --recursive (recursive): This option allows cp to copy directories and their contents recursively. Without this option, cp will not copy directories.
* -v (verbose): This option makes cp operate in verbose mode, meaning it will print the names of files and directories as they are being copied.

So, cp -prv is a combination of these options. When you use cp -prv, it will:

1. Copy directories recursively (-r).
2. Preserve file attributes (-p).
3. Print details of the files being copied (-v). ]

Assign jenkins as the user for the new home directory

* usermod -d /home/jenkins\_home jenkins
* ls -la

**[usermod** command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line. After creating a user we have to sometimes change their attributes like password or login directory etc.

**Note:**usermod command needs to be executed only as a root user.**]**



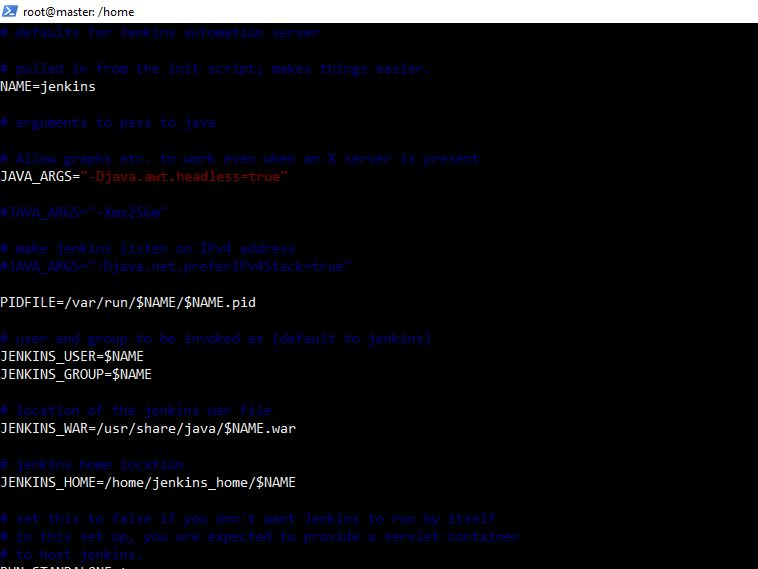
Edit the Jenkins configuration file

* vi /etc/default/jenkins

Scroll down until you reach the JENKINS\_HOME entry and add the new Home directory

* JENKINS\_HOME=/var/lib/$NAME (old home directory path)
* JENKINS\_HOME=/home/jenkins\_home/$NAME (new home directory path)

Esc, then :wq! (Save and Exit the file)



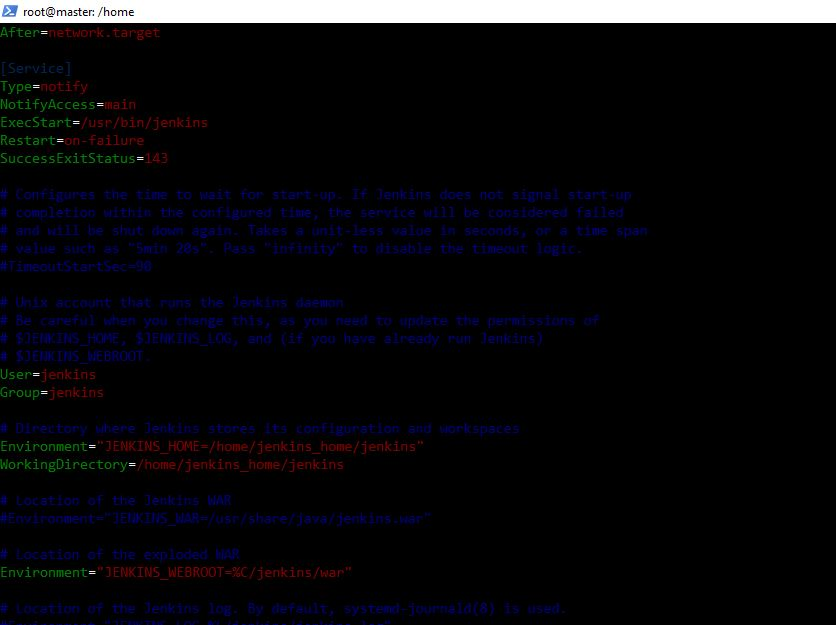
Now we will edit the jenkins service file with new port number

* vi usr/lib/systemd/system/Jenkins.service (Open Jenkins Service)

Scroll down until you reach the Environment and Working Directory, add the same Home directory

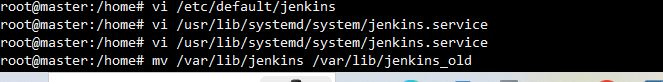
* Environment="JENKINS\_HOME=/var/lib/jenkins" [Old Entry]
* WorkingDirectory=/var/lib/jenkins
* Environment="JENKINS\_HOME=/home/jenkins\_home/jenkins" [New Entry]
* WorkingDirectory=/home/jenkins\_home/jenkins

Esc, then :wq! (Save and Exit the file)



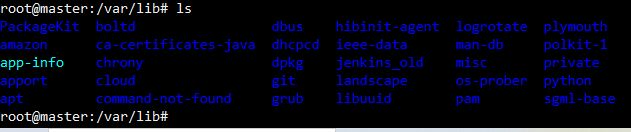
Rename the old Jenkins home directory

mv /var/lib/jenkins /var/lib/jenkins.old



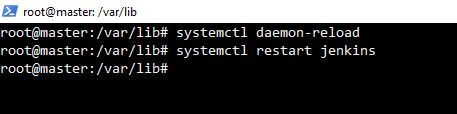
Goto:

* cd /var/lib
* ls

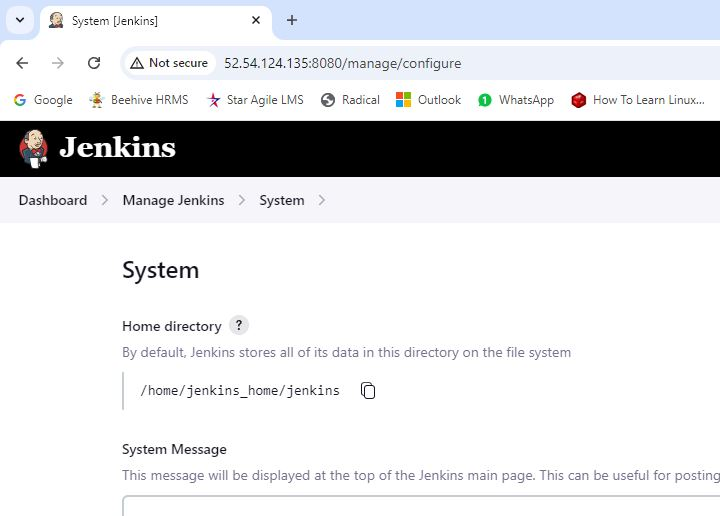


Reload the Daemon

* systemctl daemon-reload
* systemctl restart Jenkins



Verify the New Home directory under > Manage Jenkins > System > Home directory



The Home Directory path has been changed successfully.

**3) How to take backup in Jenkins and Restore.**

**Solution:**

Install jenkins and run.

Create 1-2 jobs and run it.

Create a bucket inside S3

Stop jenkins >

systemctl stop jenkins

systemctl status jenkins

Create a backup folder and

cd /home

mkdir jenkins\_backup

Take backup of jenkins home directory –

tar -zcvf jenkins-backup.tar.gz /var/lib/jenkins

ls

copy the backup to safe location

aws s3 cp jenkins-backup.tar.gz s3://jenkinsbackup02092024/jenkins-backup.tar.gz

it will give error, as we have not configured aws yet and provide role.

Provide role –

Iam > Role > create > aws service > ec2 > next > add policies of s3 > s full access > next > role name (sfullaccess)

EC2 > Instance > Actions > Security > Modify IAM role > attach the created iam role and update

Run the same command –

aws s3 cp jenkins-backup.tar.gz s3://jenkinsbackup02092024/jenkins-backup.tar.gz

if aws if not in ubuntu/linux, follow this -

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

apt install unzip

unzip awscliv2.zip

sudo ./aws/install

it will create a file in s3 bucket

S3 > bucket > jenkins file

Destroy the old jenkins file –

Terminate the instance

Create a new instance and install java, jenkins using shell script

Run jenkins using pub ip and through powershell, check if its running

Do not add password in jenkins

Add IAM role to this instance

Restore backup –

systemctl stop jenkins

systemctl status jenkins

goto /home/jenkins\_backup directory or create a directory if needed else copy in /home/ubuntu

install aws using above syntax

aws s3 cp s3://jenkinsbackup02092024/jenkins-backup.tar.gz jenkins-backup.tar.gz

check using ls

rm -rf /var/lib/jenkins [delete jenkins from current instance server]

tar -zxvf jenkins-backup.tar.gz -C / [restore backup to particular file]

systemctl start jenkins

systemctl status jenkins

jenkins restored successfully.

Check on browser – all jobs are restored back.

tar Command to Compress Files in Linux. The Linux 'tar' stands for tape archive, which is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands that provides archiving functionality in Linux.

**4) How to install Jenkins in Tomcat**

Answer:

Step 1: Install Java

sudo apt update -y && apt upgrade -y && apt install java

Step 2: Install Tomcat

cd /tmp

wget https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.85/bin/apache-tomcat-8.5.85.tar.gz

[Use wget to download the link that you copied from the Tomcat website]

Install Tomcat to the /opt/tomcat directory

mkdir /opt/tomcat  
tar xzvf apache-tomcat-8\*tar.gz -C /opt/tomcat - strip-components=1

Step 3: Update Permissions

cd /opt/tomcat   
chown -R ubuntu:ubuntu /opt/tomcat

Step 4: Create a systemd Service File manually

vi /etc/systemd/system/tomcat.service

Environment=JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre  
Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid  
Environment=CATALINA\_HOME=/opt/tomcat  
Environment=CATALINA\_BASE=/opt/tomcat  
Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'  
Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'ExecStart=/opt/tomcat/bin/startup.sh  
ExecStop=/opt/tomcat/bin/shutdown.shUser=ubuntu  
Group=ubuntu  
UMask=0007  
RestartSec=10  
Restart=always[Install]  
WantedBy=multi-user.target

When you are finished, save and close the file.

Reload the systemd daemon so that it knows about our service file:

$ sudo systemctl daemon-reload

Start the Tomcat service by typing:

sudo systemctl start tomcat

Double check that it started without errors by typing:

$ sudo systemctl status tomcat

Public ip:8080 - Tomcat is running

Step 5 : Download jenkins stable version war file

cd /tmp  
wget https://get.jenkins.io/war-stable/2.346.1/jenkins.war

ls -ltr

Deploy download jenkins war file into tomcat webapps folder

cp -rv jenkins.war /opt/tomcat/webapps/

We will also verify that jenkins.war has been copied successfully and also

jenkins folder is also created.

ls -ltr tomcat/webapps

Restart tomcat service

$ sudo systemctl restart tomcat

Hit the tomcat url in your web browser **http://ip-address:8080/jenkins**

**5) How to run Remote Job in Jenkins**

Answer:

We can trigger a job remotely through a Web URL in Jenkins also.

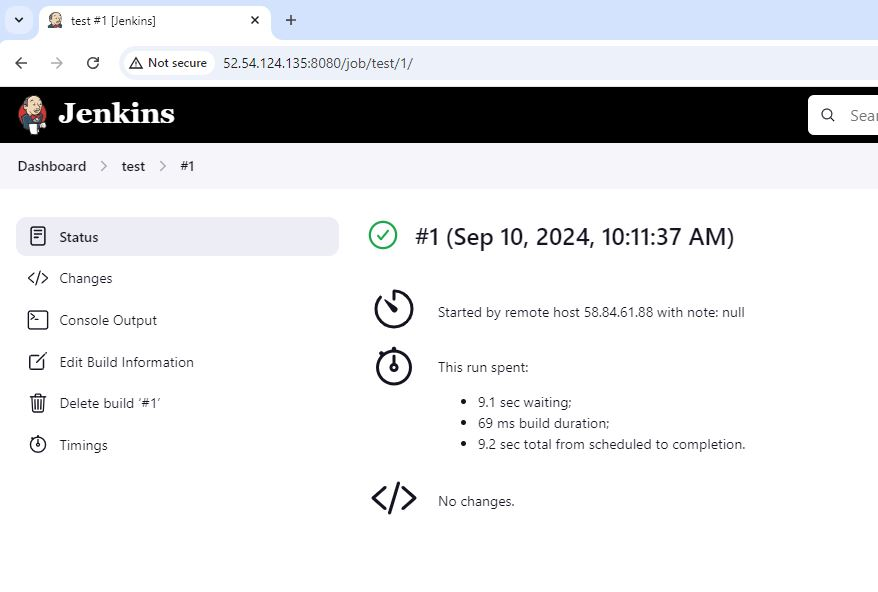
We will follow the below steps to achieve the same:

**Method 1:**

While configuring the job, we need to select Build Triggers > Trigger builds remotely > Authentication token > any name for token, eg. new > Save.

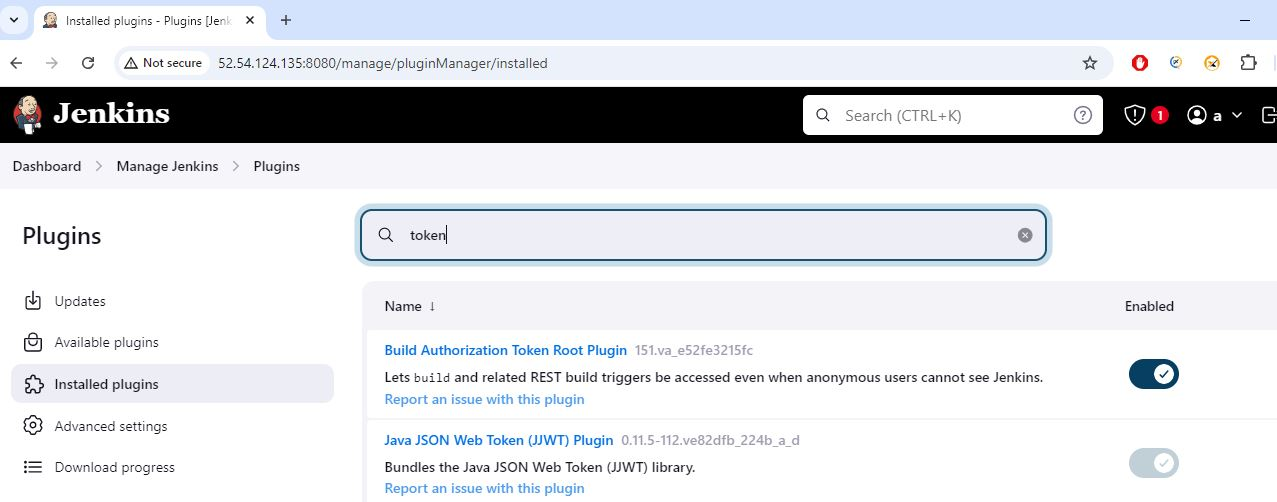
To trigger the build remotely we need use the below URL:

Jenkins\_URL/job/test/build?token=TOKEN\_NAME [or, buildWithParameters?token=TOKEN\_NAME]



**Method 2:**

Alternate Way to run the job through remotely is by using a plugin Build Authorization Token Root –



Now we will run the below URL to see if build runs.

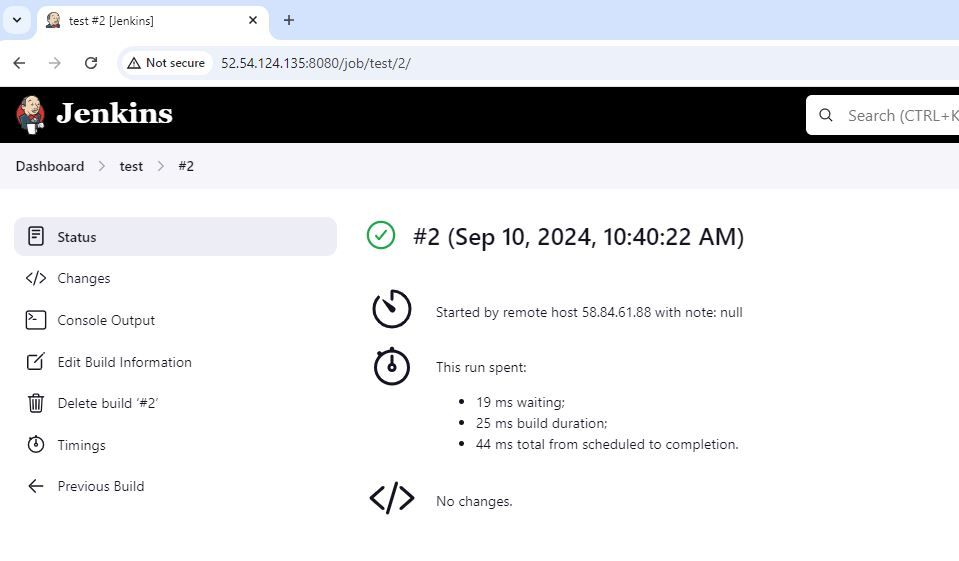
<Jenkins\_URL>/buildByToken/build?job=<Job\_Name>&token=<token Name>

E.G.

http://52.54.124.135:8080/buildByToken/build?job=test&token=new

GoTo: http://52.54.124.135:8080/job/test/

Observe the 2nd job run successfully.

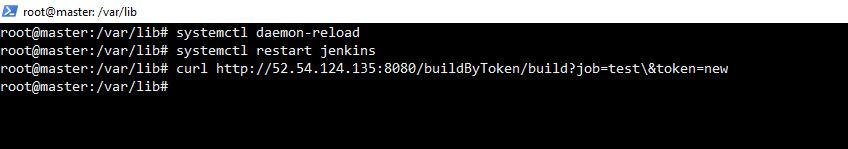


**Method 3:**

We can also run it on the Ubuntu terminal using the below command.

Just need to add “\” before the & and then run.

curl <Jenkins\_URL>/buildByToken/build?job=<Job\_Name>\&token=<token Name>





**6) How to create GitHub hook trigger for GITScm polling through Jenkins**

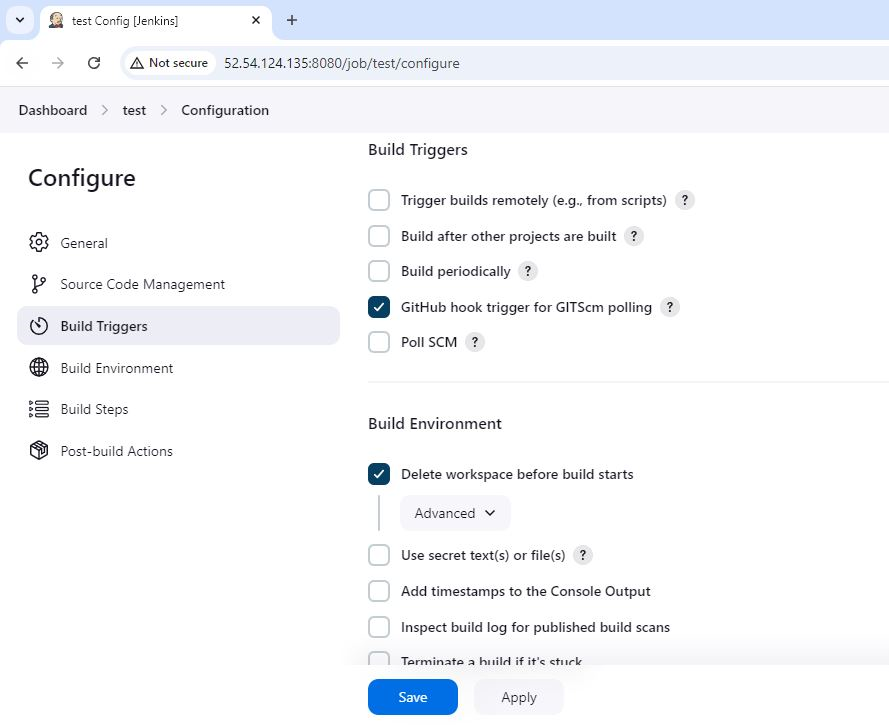
Answer:

This is used when a job needs to be triggered when there is any change

in the GitHub repo.

Steps to followed are below:

* Create a job > configuration > Build Triggers > GitHub hook trigger for GITScm polling (Enable)
* Build Environment > Delete workspace before build starts (optional)
* Save



**7) Install old version of Jenkins and then upgrade it to a latest version**

Answer:

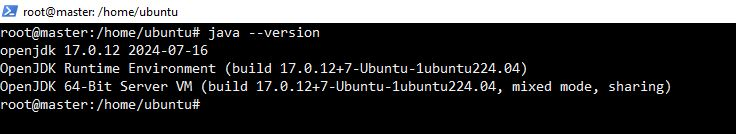
**Important Jenkins files and directory’s in Linux**

1. **/usr/share/jenkins** — Jenkins binaries and the jenkins.war file are kept here.
2. **/var/log/jenkins** — Jenkins logs are kept in this place.
3. **/var/lib/jenkins** — This is an important directory of Jenkins. This directory contains all the details related Jenkins. In this directory it will stored jobs, workspace, users, nodes details, plugins and job configurations.
4. **/run/jenkins**— This directory contains PID of currently running Jenkins.
5. **/usr/lib/systemd/system/jenkins.service –** Jenkins services files, like ports, home dir, env.

[pid file is a process identification file that stores the process ID (PID) of running processes.]

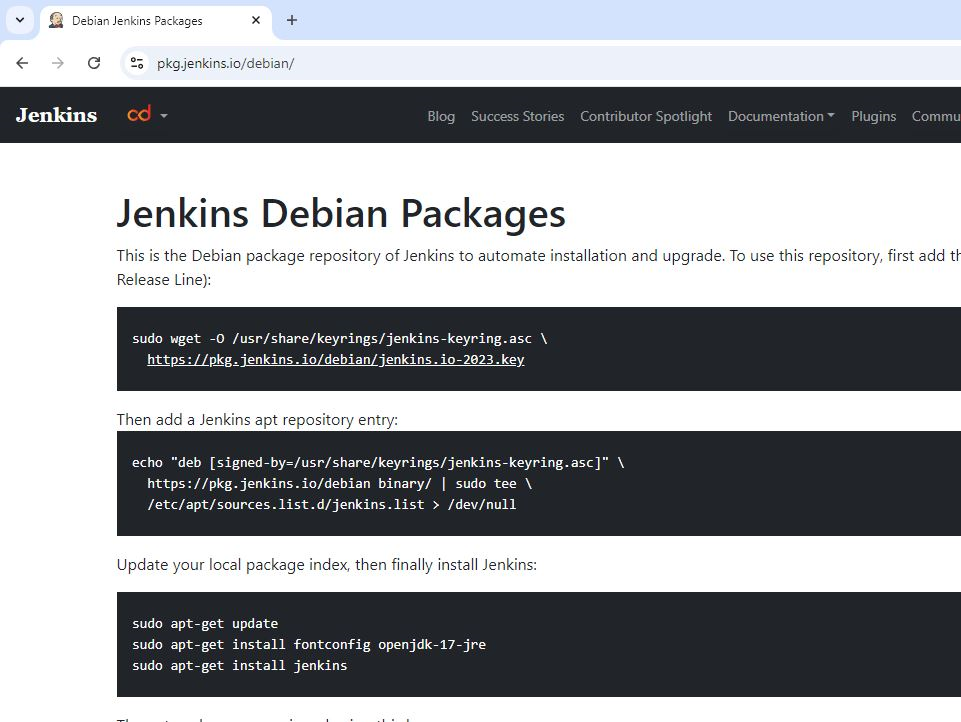
Steps :

1. Run EC2 instance with java installed



2. <https://www.jenkins.io/download/>

Select Ubuntu/Debian

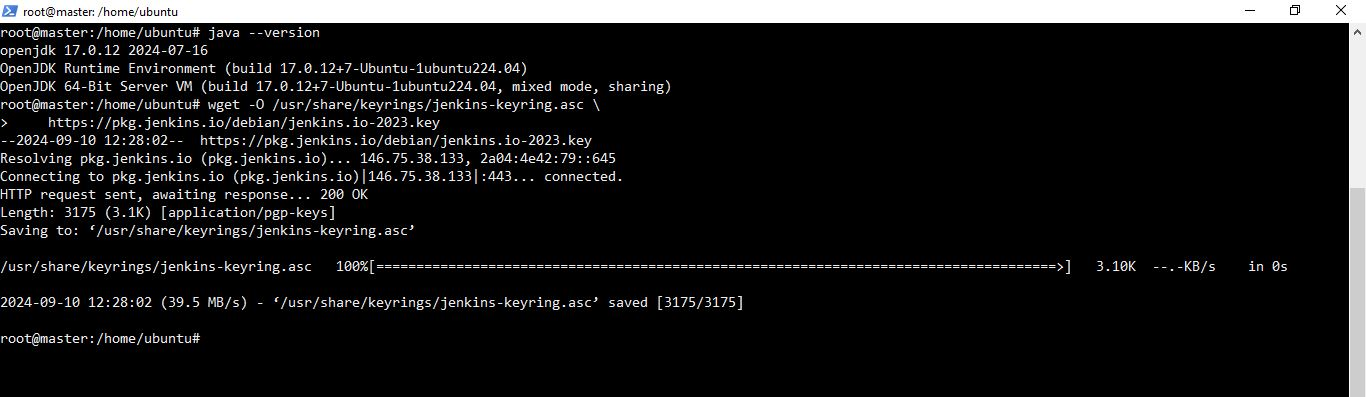


**Jenkins Debian Packages**

This is the Debian package repository of Jenkins to automate installation and upgrade. To use this repository, first add the key to your system:

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

<https://pkg.jenkins.io/debian/jenkins.io-2023.key>

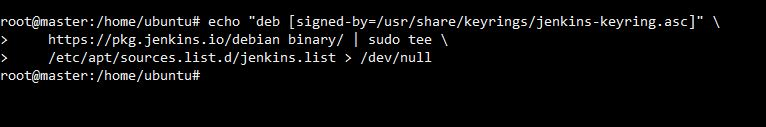


Then add a Jenkins apt repository entry:

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

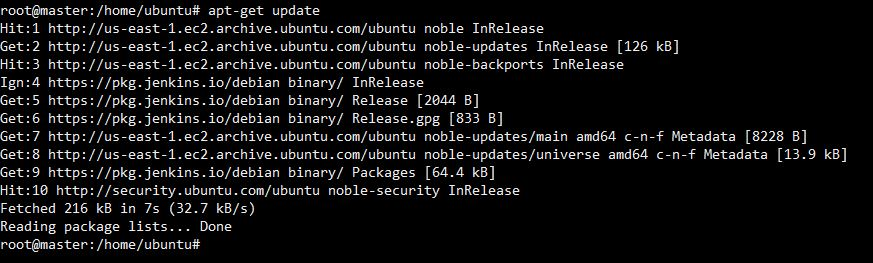
https://pkg.jenkins.io/debian binary/ | sudo tee \

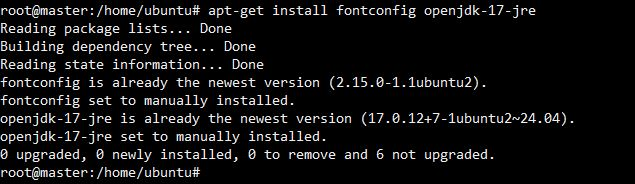
/etc/apt/sources.list.d/jenkins.list > /dev/null



**Update your local package index, then finally install Jenkins:**

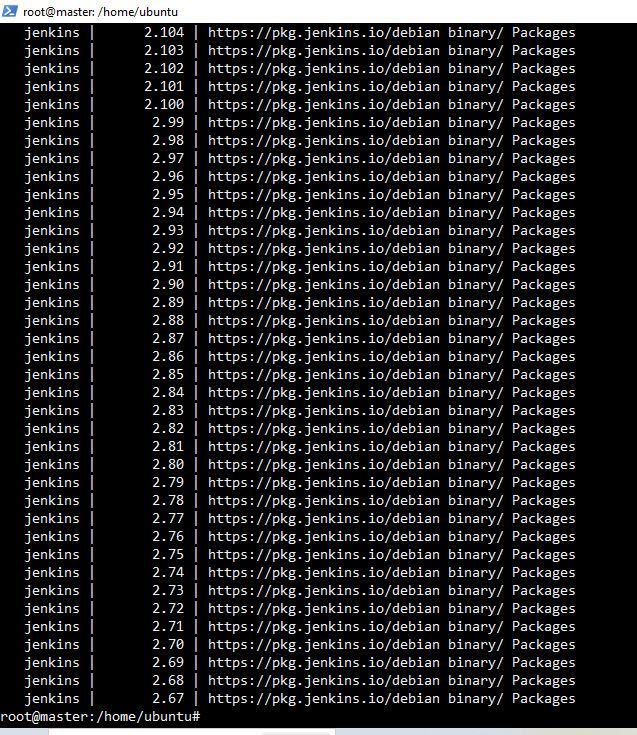
* apt-get update
* apt-get install fontconfig openjdk-17-jre





Now we will check which all versions of Jenkins are available, so that we can install and old version and then perform an upgrade to the latest version.

* apt-cache madison jenkins

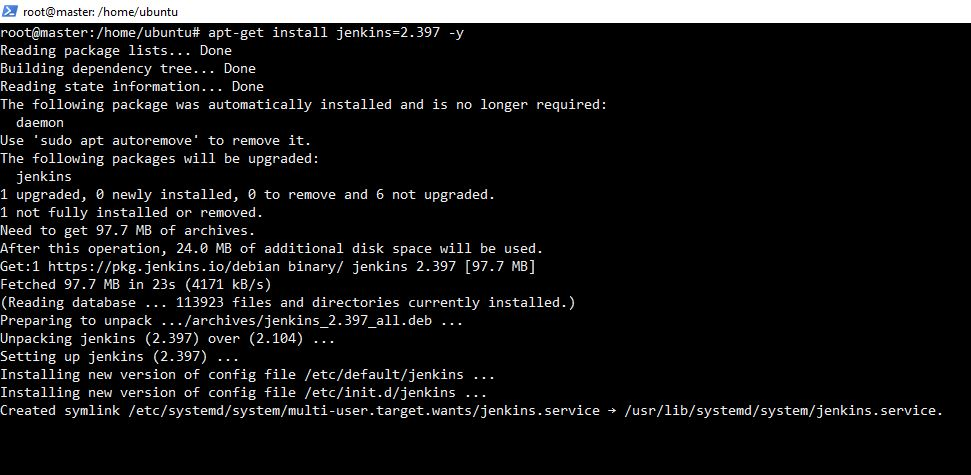


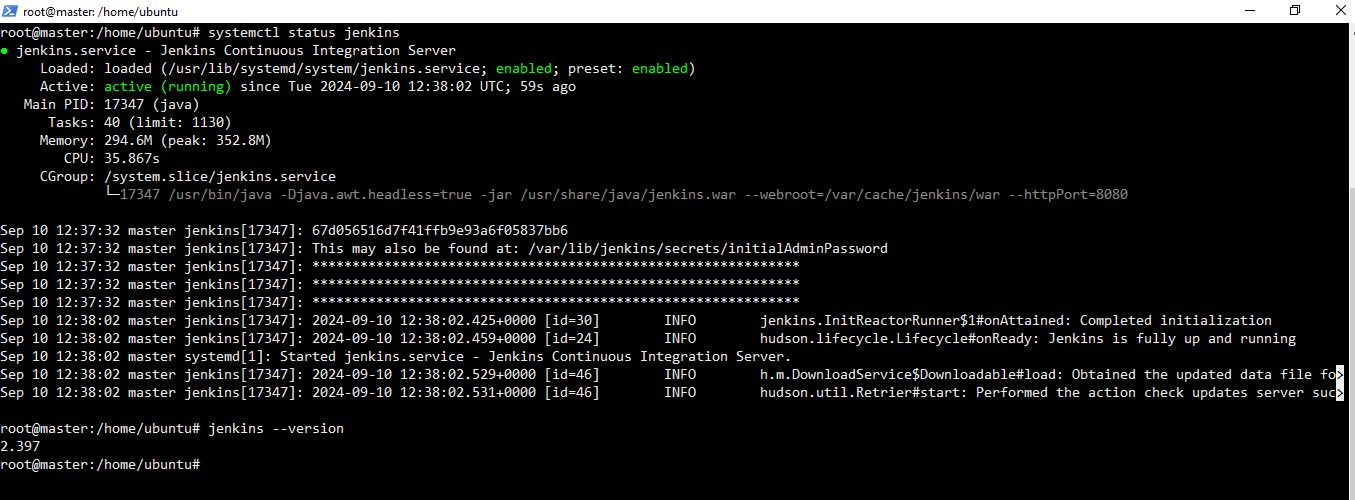
Here we have randomly selected a version to install i.e. 2.397. To install the specific jenkins version, we need to run the below command.

* apt-get install jenkins=2.397 -y - for specific version
* systemctl status jenkins

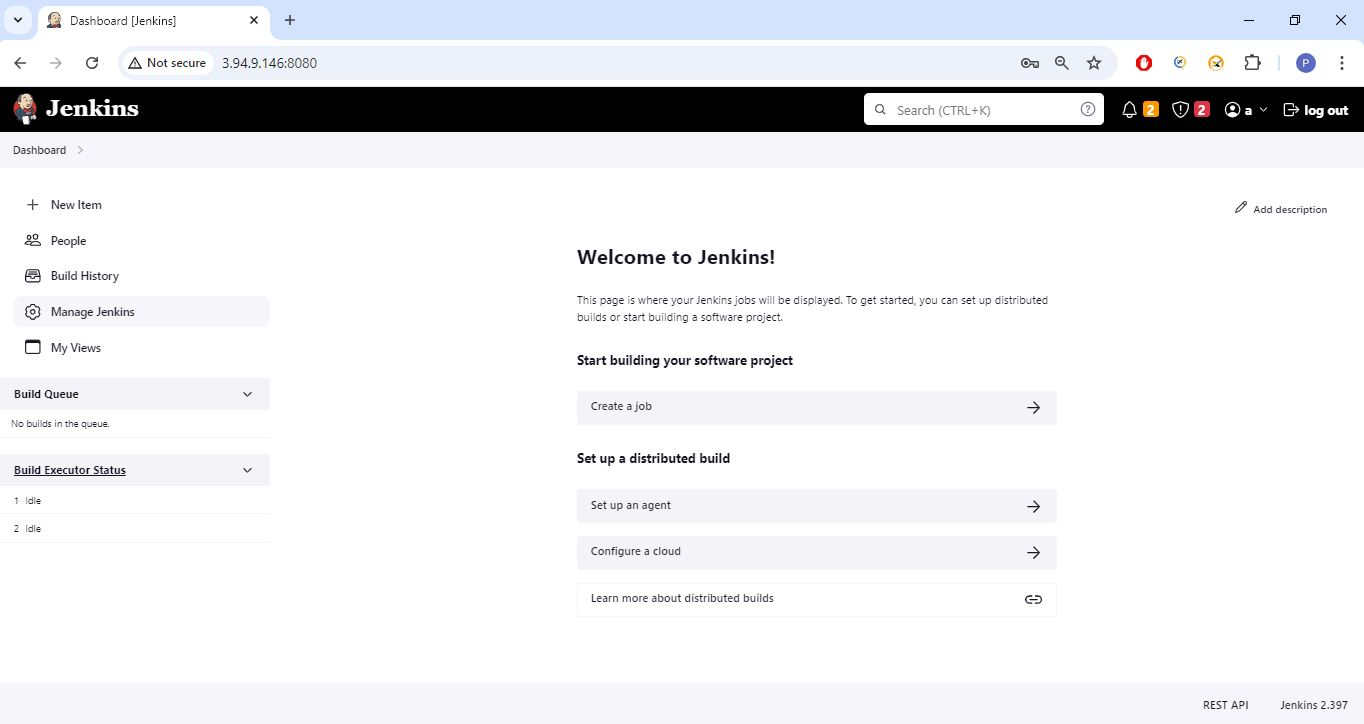
Check the version:

* jenkins --version





Same version can be checked by web, after launching jenkins in browser.

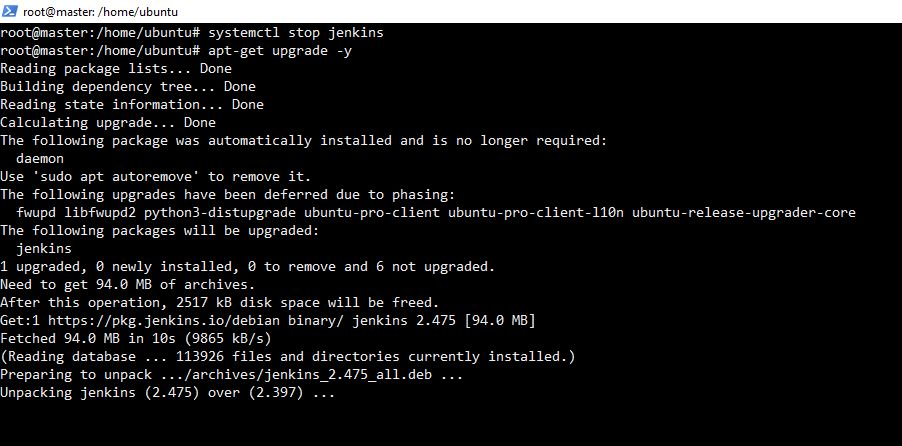


Stop jenkins service:

* systemctl stop jenkins

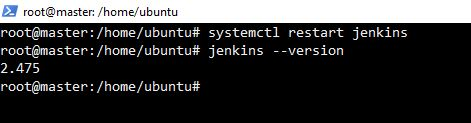
Upgrade this version to latest one:

* apt-get upgrade -y

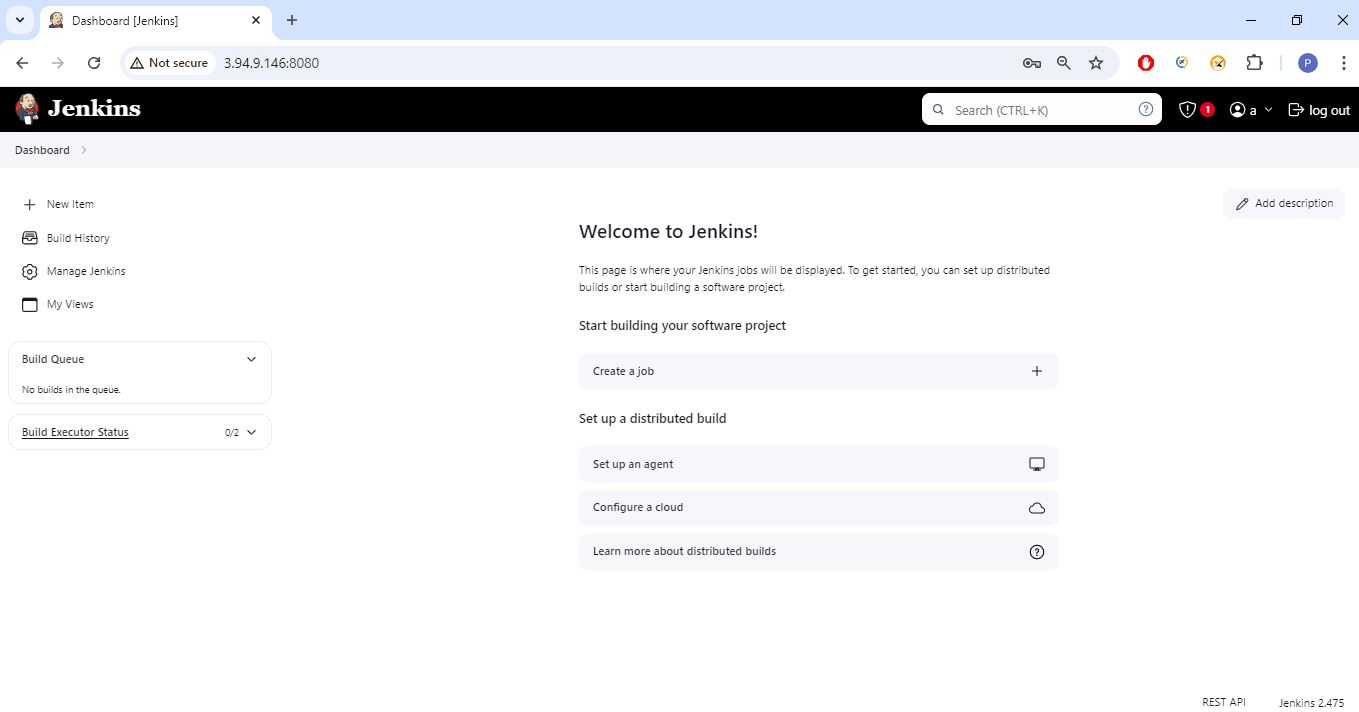


Check the version:

* systemctl restart jenkins
* jenkins --version



Same version can be checked by web, after launching jenkins in browser.



Congratulations! You successfully upgrade the Jenkins. (2.474)

**8) Jenkins Installation** **and Configuration Script**

Answer:

In Jenkins, a crumb is a token that is used to protect against CSRF attacks. A crumb is a token that Jenkins creates and sends to a user. It contains information about the user, such as their username, IP address, and the web session ID. A crumb is required for form submissions and other actions that can modify Jenkins, such as triggering builds or changing configuration.

**9) How to put https in Jenkins URL (Open Cert option)**

Answer:

The default Jenkins installation runs on ports 8080 and 8443. Typically, HTTP/HTTPS servers run on ports 80 and 443, respectively.