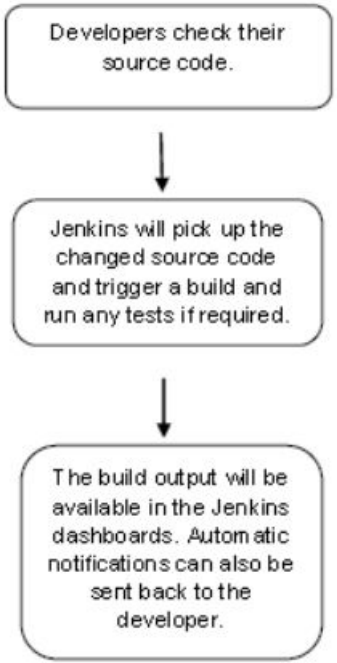
Jenkins

What is Jenkins

- Jenkins is an open source automation tool written in Java programming language that allows continuous integration.
- Jenkins builds and tests our software projects which continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.
- It also allows us to continuously deliver our software by integrating with a large number of testing and deployment technologies.

Steps in Jenkins

- Possible steps executed by Jenkins are for example:
- Perform a software build using a build system like Gradle or Maven Apache
- Execute a shell script
- Archive a build result
- Running software tests.



History of Jenkins

- Kohsuke Kawaguchi, who is a Java developer, working at SUN Microsystems, was tired of building the code and fixing errors repetitively. In 2004, he created an automation server called Hudson that automates build and test task.

- In 2011, Oracle who owned Sun Microsystems had a dispute with Hudson open source community, so they forked Hudson and renamed it as Jenkins.

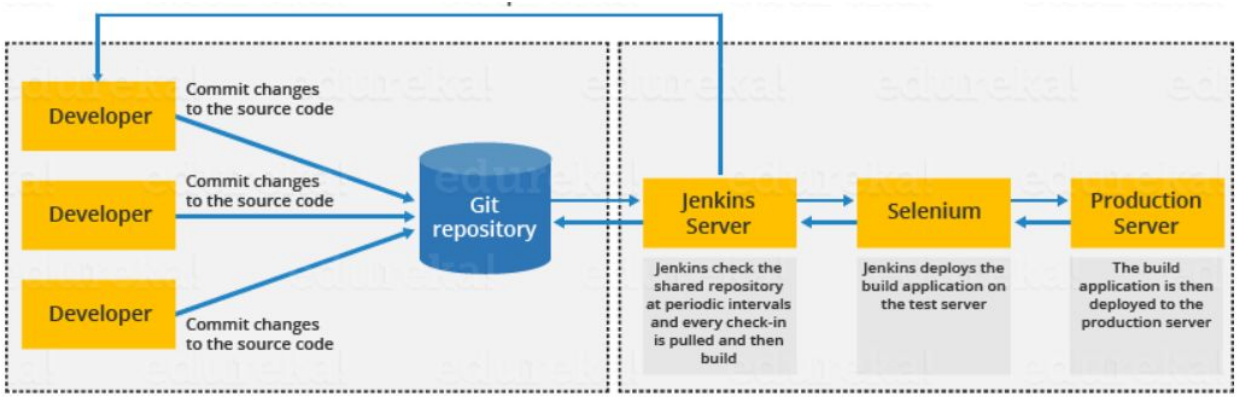
- Both Hudson and Jenkins continued to operate independently. But in short span of time, Jenkins acquired a lot of contributors and projects while Hudson remained with only 32 projects. Then with time, Jenkins became more popular, and Hudson is not maintained anymore.

What is Contineous Integration (CI)

- Continuous Integration (CI) is a development practice in which the developers are needs to commit changes to the source code in a shared repository at regular intervals. Every commit made in the repository is then built. This allows the development teams to detect the problems early.

- Continuous integration requires the developers to have regular builds. The general practice is that whenever a code commit occurs, a build should be triggered.

How Jenkins builds the code.

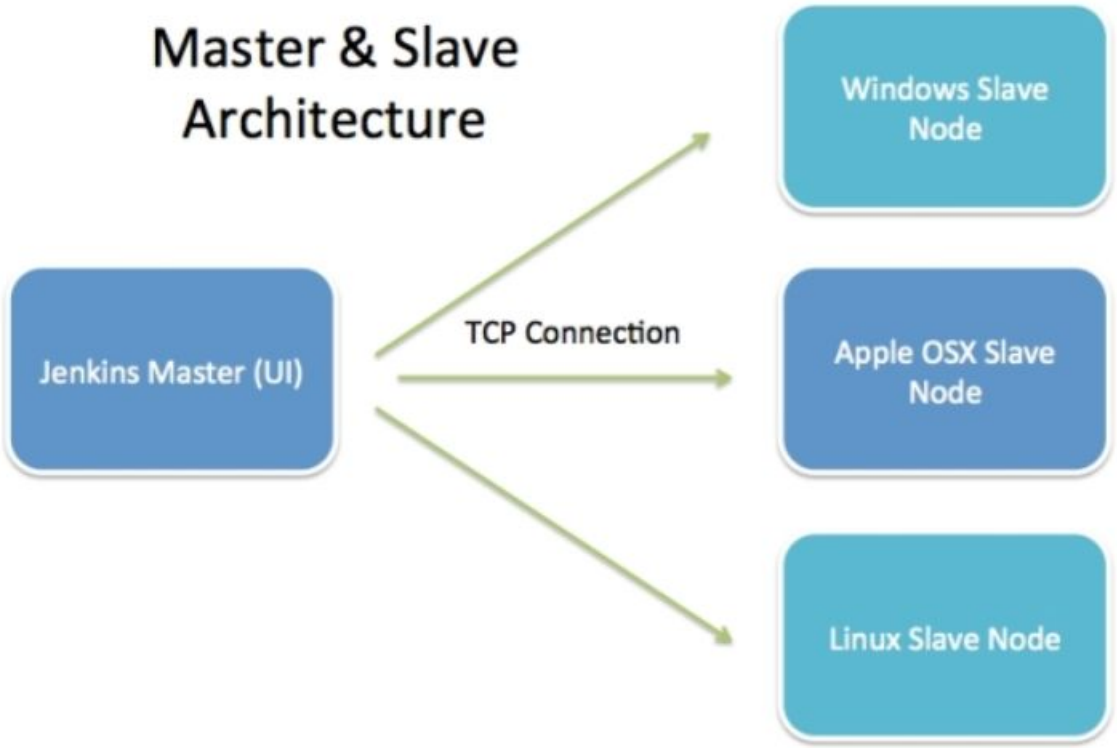


- First of all, a developer commits the code to the source code repository. Meanwhile, the Jenkins checks the repository at regular intervals for changes.
- Soon after a commit occurs, the Jenkins server finds the changes that have occurred in the source code repository. Jenkins will draw those changes and will start preparing a new build.
- If the build fails, then the concerned team will be notified.
- If built is successful, then Jenkins server deploys the built in the test server.
- After testing, Jenkins server generates a feedback and then notifies the developers about the build and test results.
- It will continue to verify the source code repository for changes made in the source code and the whole process keeps on repeating.

Benefits of Jenkins

- It is an open source tool.
- It is free of cost.
- It does not require additional installations or components. Means it is easy to install.
- Easily configurable.
- It supports 1000 or more plugins to ease your work. If a plugin does not exist, you can write the script for it and share with community.
- It is built in java and hence it is portable.
- It is platform independent. It is available for all platforms and different operating systems. Like OS X, Windows or Linux.
- Easy support, since it open source and widely used.
- Jenkins also supports cloud based architecture so that we can deploy Jenkins in cloud based platforms.

Jenkins Architecture



Jenkins Master

- The main server of Jenkins is the Jenkins Master. It is a web dashboard which is nothing but powered from a war file. By default it runs on 8080 port. With the help of Dashboard, we can configure the jobs/projects but the build takes place in Nodes/Slave. By default one node (slave) is configured and running in Jenkins server.

- **The server's job or master's job is to handle:**
- Scheduling build jobs.
- Dispatching builds to the nodes/slaves for the actual execution.
- Monitor the nodes/slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master/Server instance of Jenkins can also execute build jobs directly.

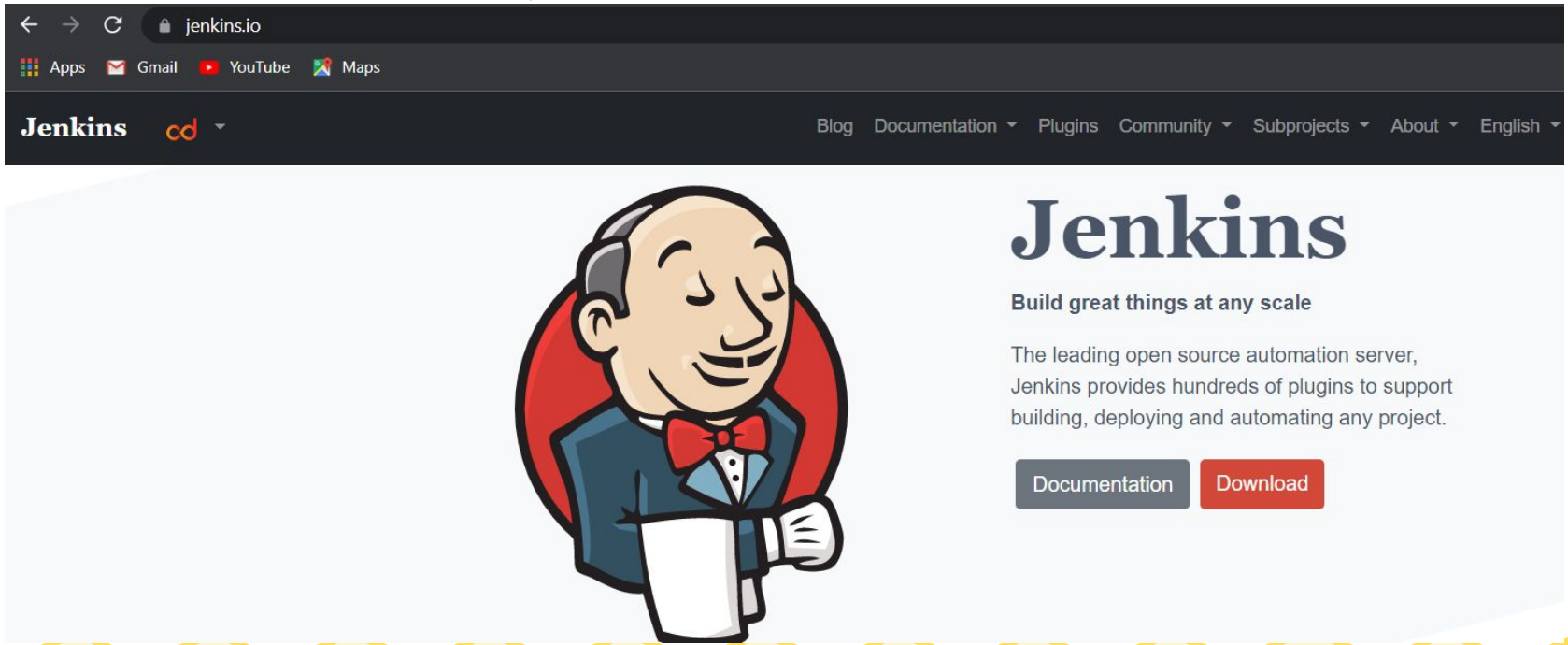
Jenkins Slave

- Jenkins slave is used to execute the build jobs dispatched by the master. We can configure a project to always run on a particular slave machine, or particular type of slave machine, or simple let the Jenkins to pick the next available slave/node.

- As we know Jenkins is developed using Java is platform independent thus Jenkins Master/Servers and Slave/nodes can be configured in any servers including Linux, Windows, and Mac.

Jenkins Installation

- Please ensure Java is already installed on your system.
- Download jenkins from this site: http://www.jenkins.io and click on download button,.



Jenkins Installation

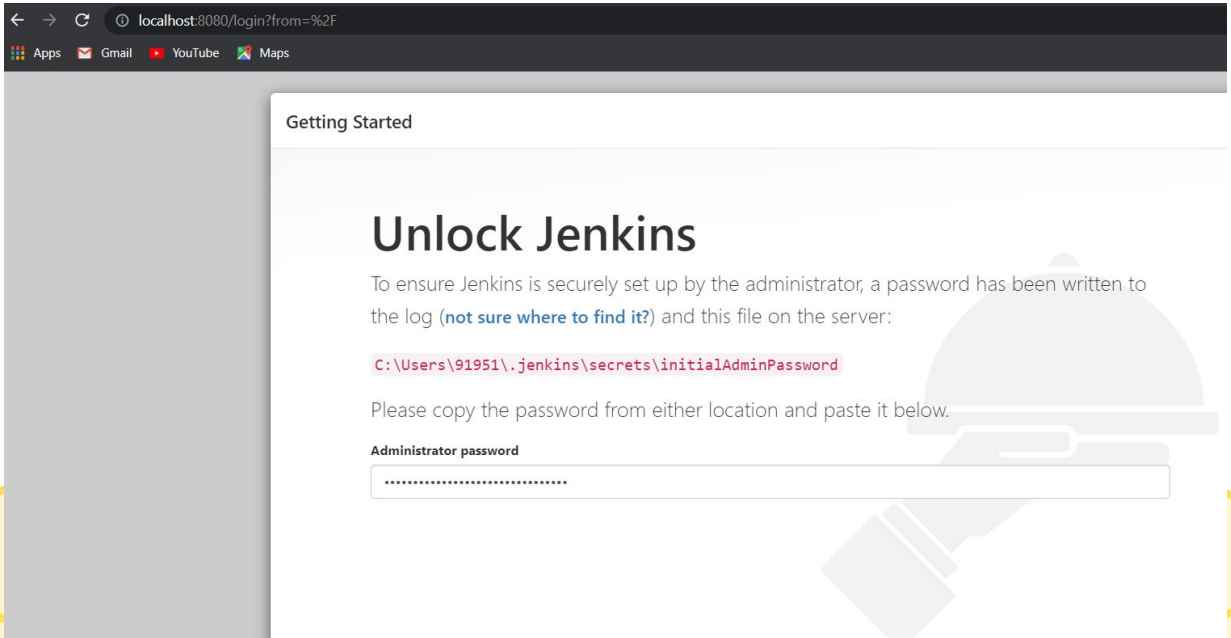
- Select

- And it starts downloading automatically.
- Copy this war file into a newly created folder named jenkins and run the following command : java -jar jenkins.war

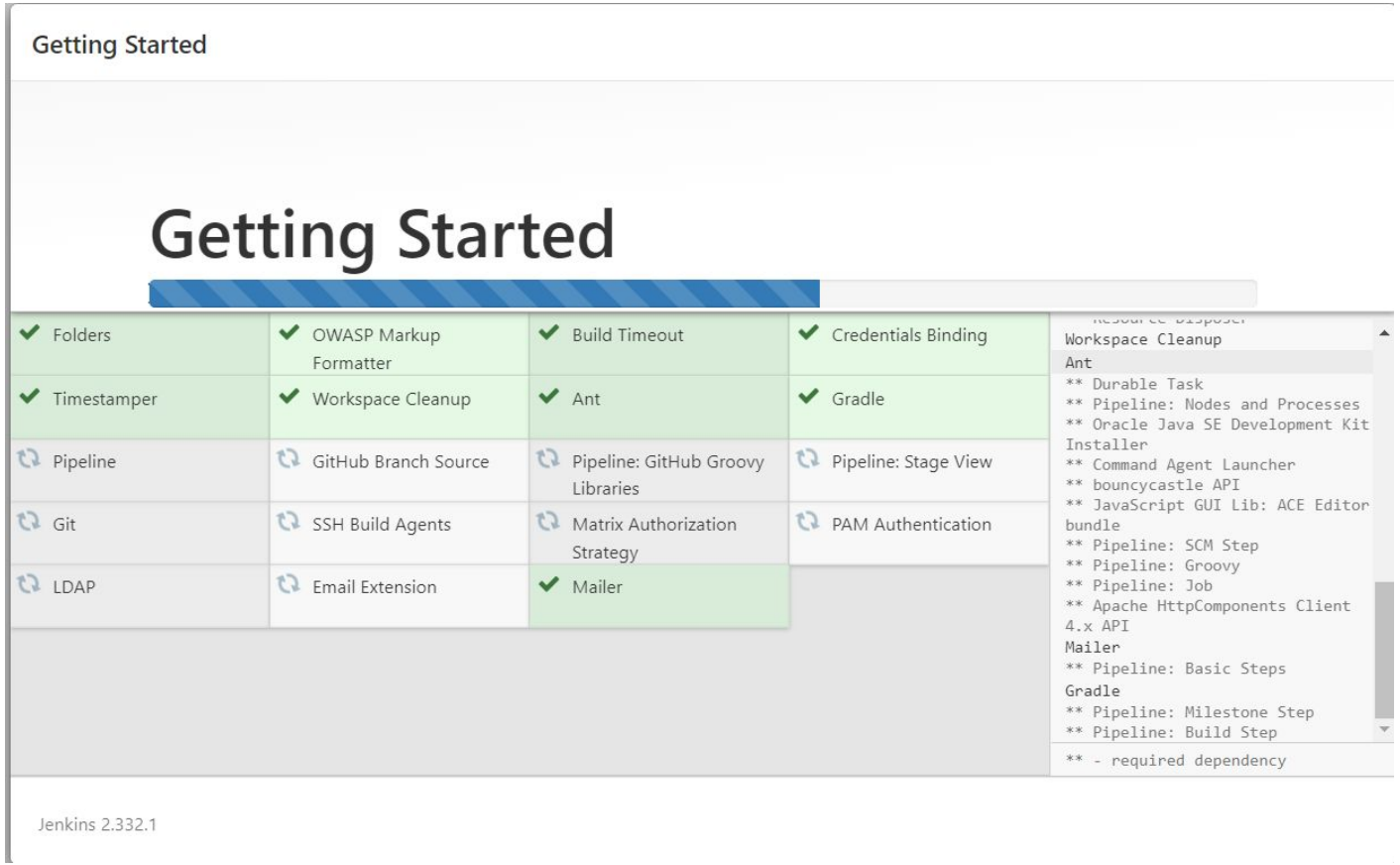
```
C:\Java\Jenkins>dir
 Volume in drive C is Windows
 Volume Serial Number is D87D-3DAA

 Directory of C:\Java\Jenkins

14-03-2022  23:29    <DIR>          .
14-03-2022  23:29    <DIR>          ..
14-03-2022  23:28        94,952,903 jenkins.war
               1 File(s)     94,952,903 bytes
               2 Dir(s)  404,899,405,824 bytes free

C:\Java\Jenkins>java -jar jenkins.war
Running from: C:\Java\Jenkins\jenkins.war
webroot: $user.home/.jenkins
2022-03-14 18:00:20.435+0000 [id=1]     INFO    org.eclipse.jetty.util.log.Log#initialized: Logging initialized @472ms t
o org.eclipse.jetty.util.log.JavaUtilLog
2022-03-14 18:00:20.497+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
2022-03-14 18:00:22.144+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2022-03-14 18:00:22.191+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-9.4.43.v20210629; built:
2021-06-30T11:07:22.254Z; git: 526006ecfa3af7f1a27ef3a288e2bef7ea9dd7e8; jvm 1.8.0_321-b07
2022-03-14 18:00:22.583+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /,
did not find org.eclipse.jetty.jsp.JettyJspServlet
2022-03-14 18:00:22.614+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSessionIdManager worke
rName=node0
2022-03-14 18:00:22.614+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: No SessionScavenger set, usin
```

Jenkins Installation

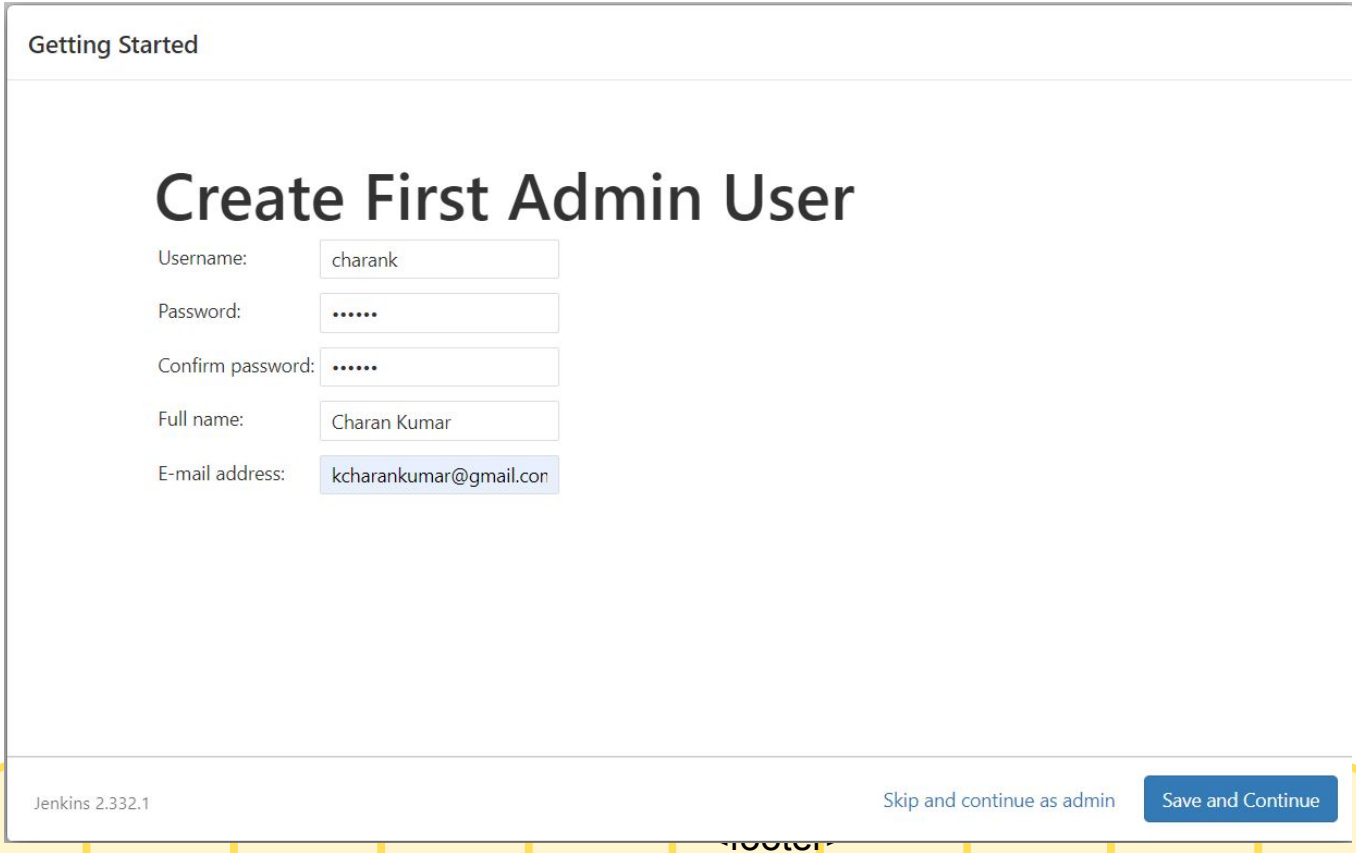
- When the jenkins is opened in the browser, it shows the following. Enter the password generated by the intstallation and select the first option of installation that is “most suited for industry”
- j



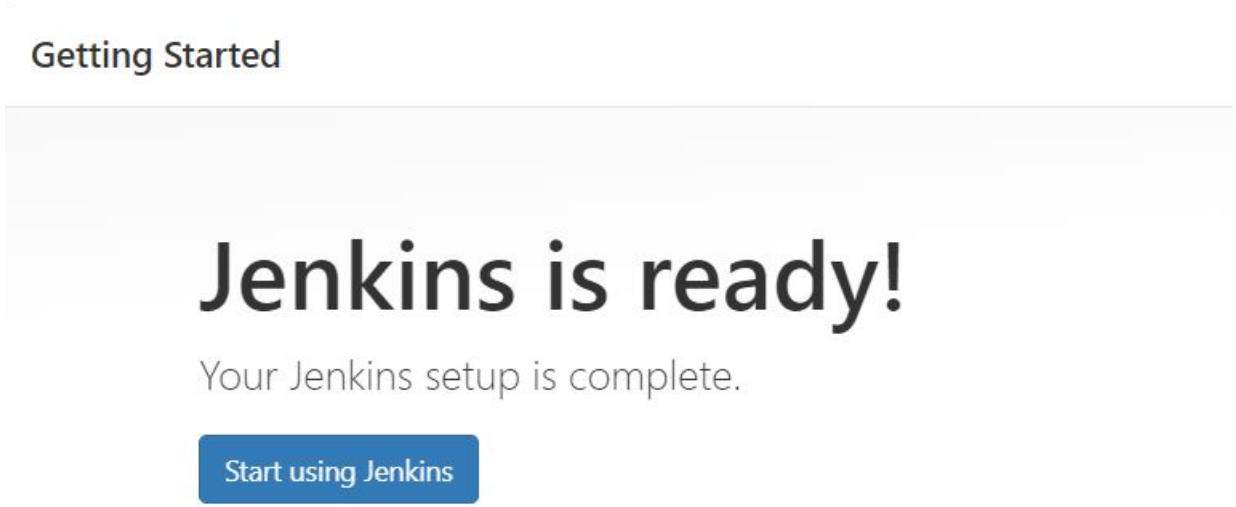
- Jenkins will install these modules...
-



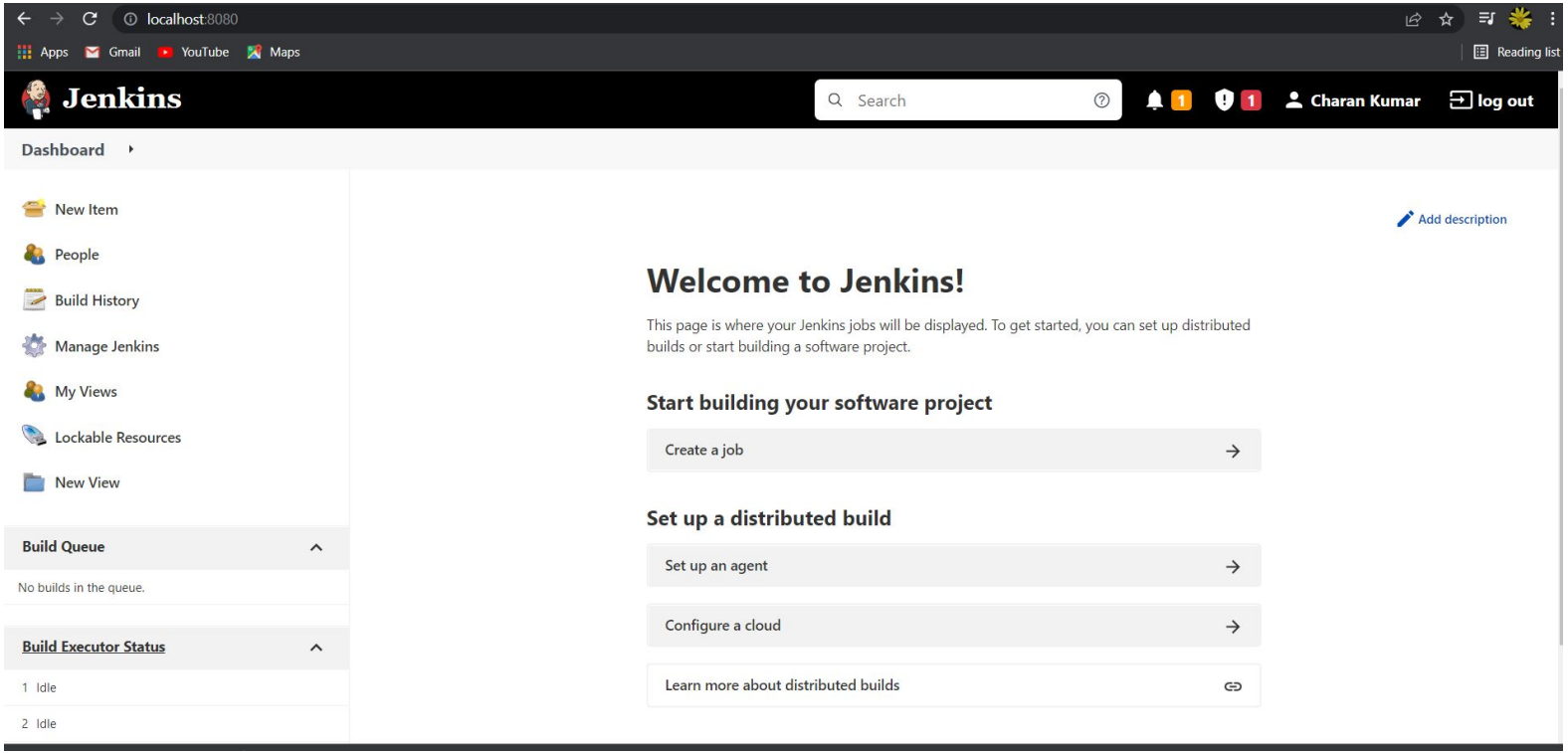
- Registering a new user in jenkins.
- Click on Save and Finish on configuration screen.
-



- Jenkins installation is done and it is ready to use.

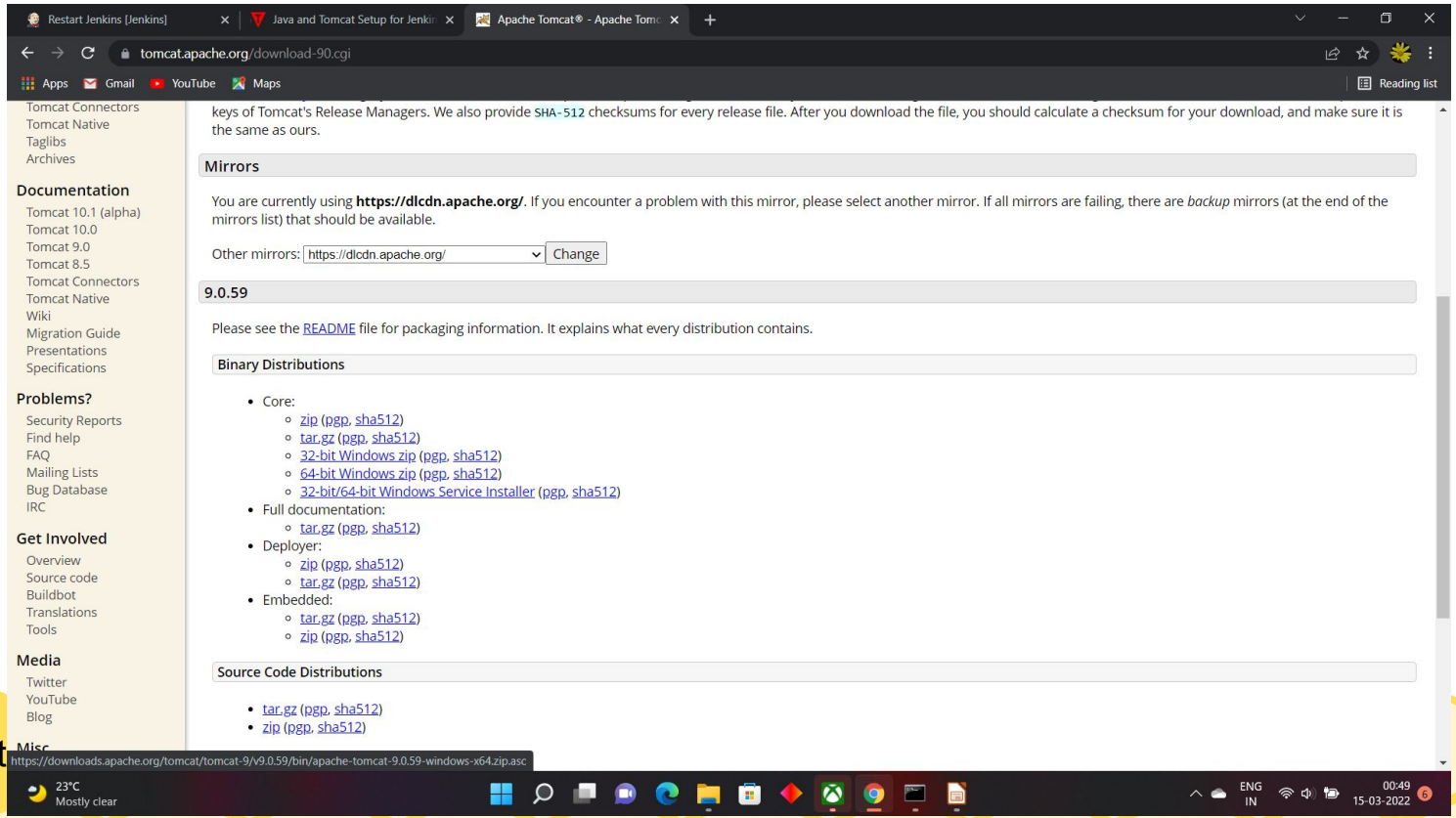


- Jenkins dashboard looks like this.
-



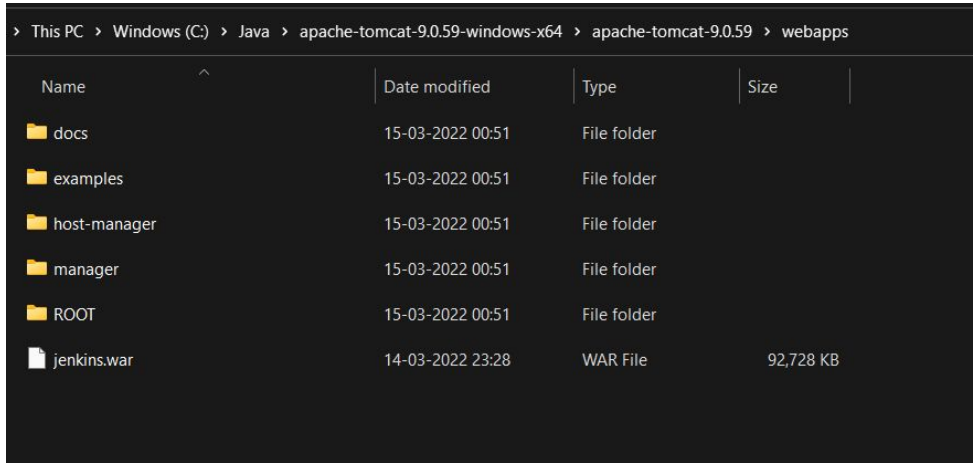
Install Tomcat for Jenkins

- Go to tomcat site *https://tomcat.apache.org/*
- and download tomcat binaries 64 bit zip.



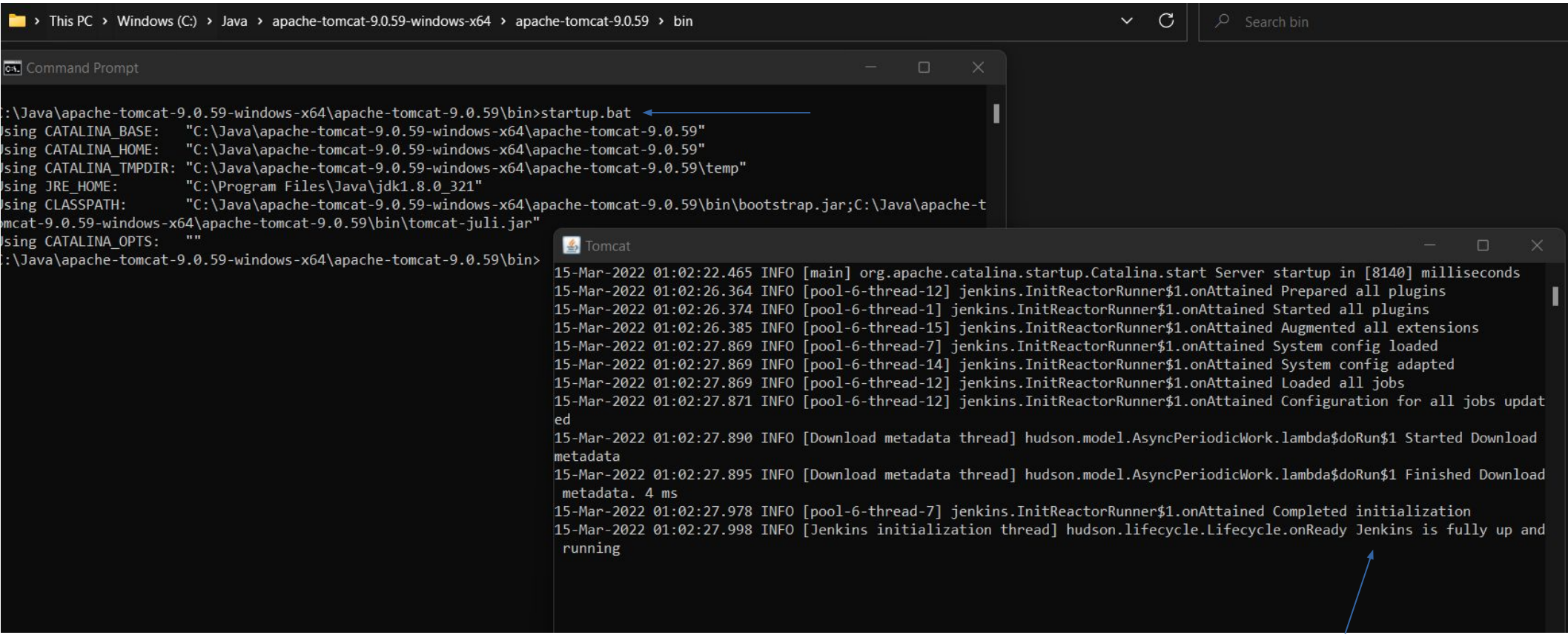
Install Tomcat for Jenkins

- Unzip apache tomcat into a directory.
- Go to WebApps directory of Apache Tomcat and copy jenkins.war over there.



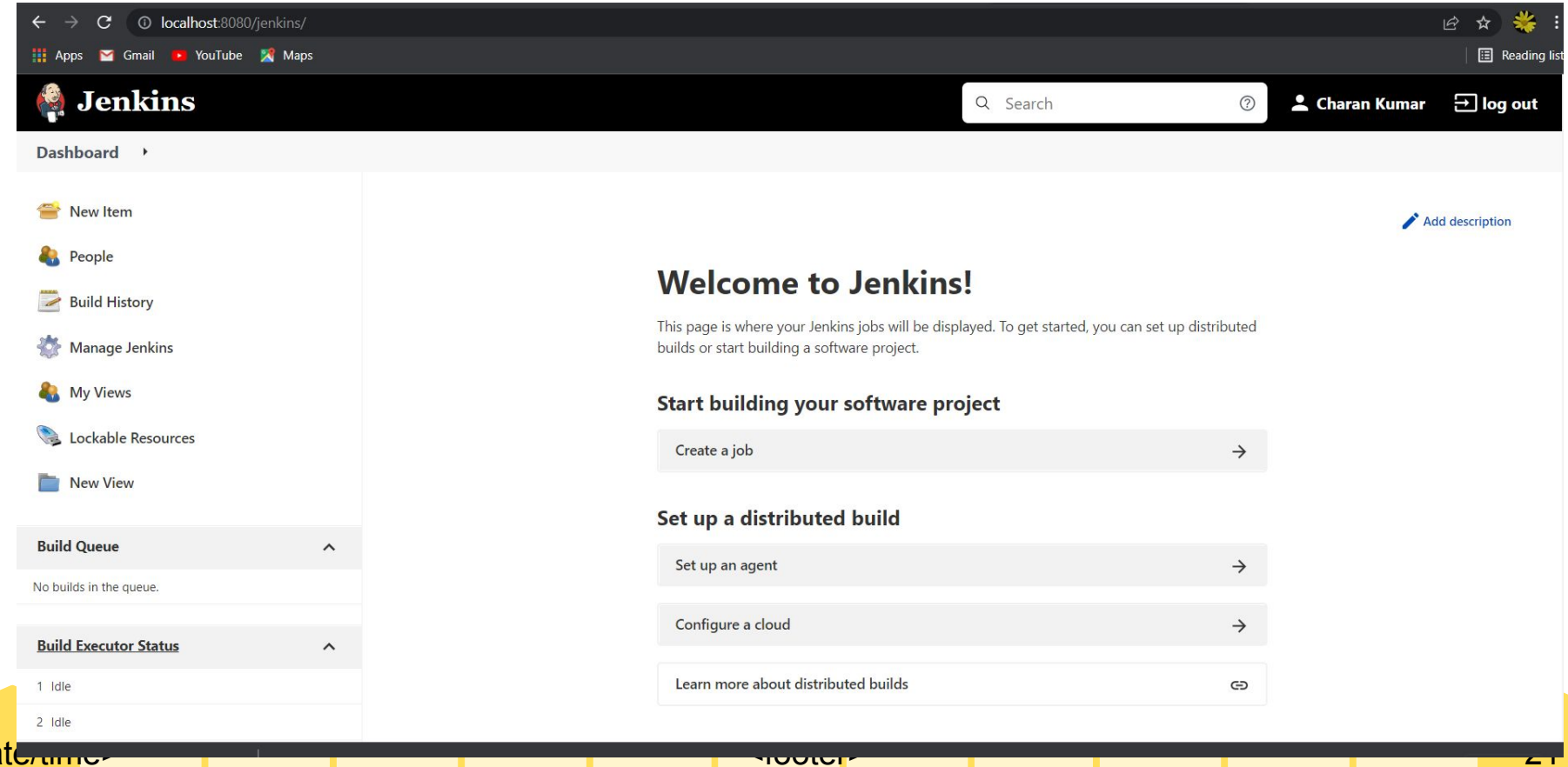
- Ensure the environmental variable JAVA_HOME is set. If not, set it.
- From the bin directory, give the command **startup.bat.** With that apache tomcat is started and along with that jenkins also is started.

Install Tomcat for Jenkins



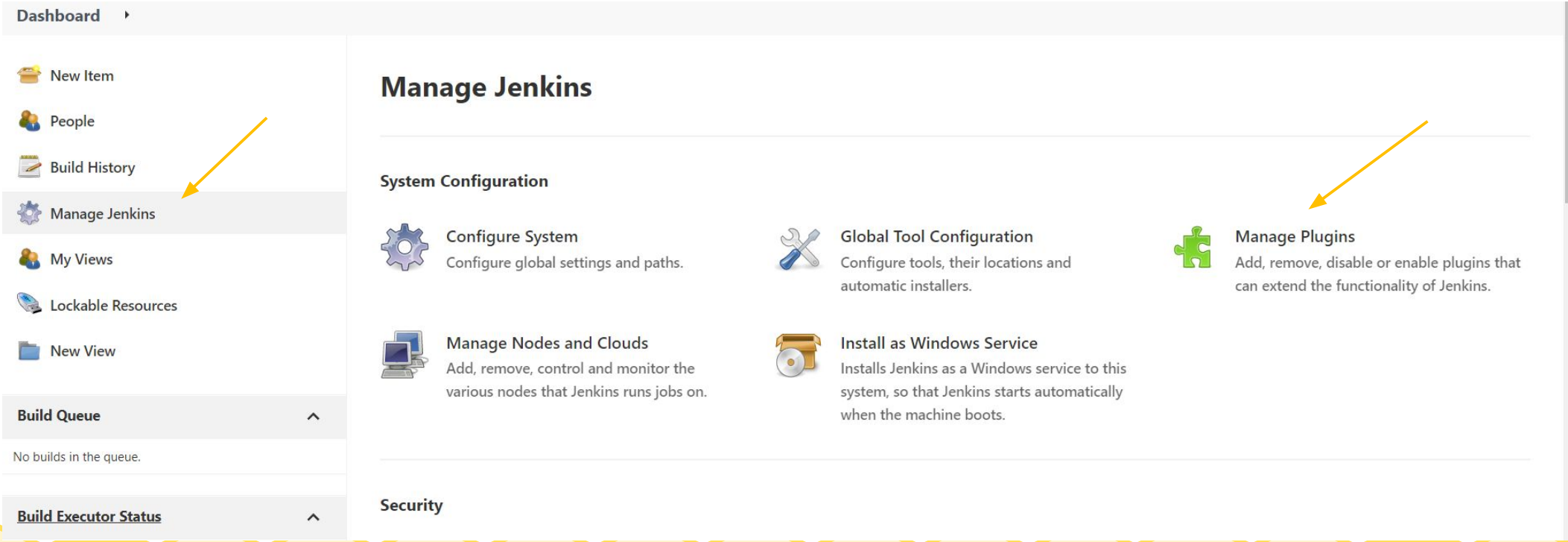
Install Tomcat for Jenkins

- Jenkins dashboard can be started with http://localhost:8080/jenkins



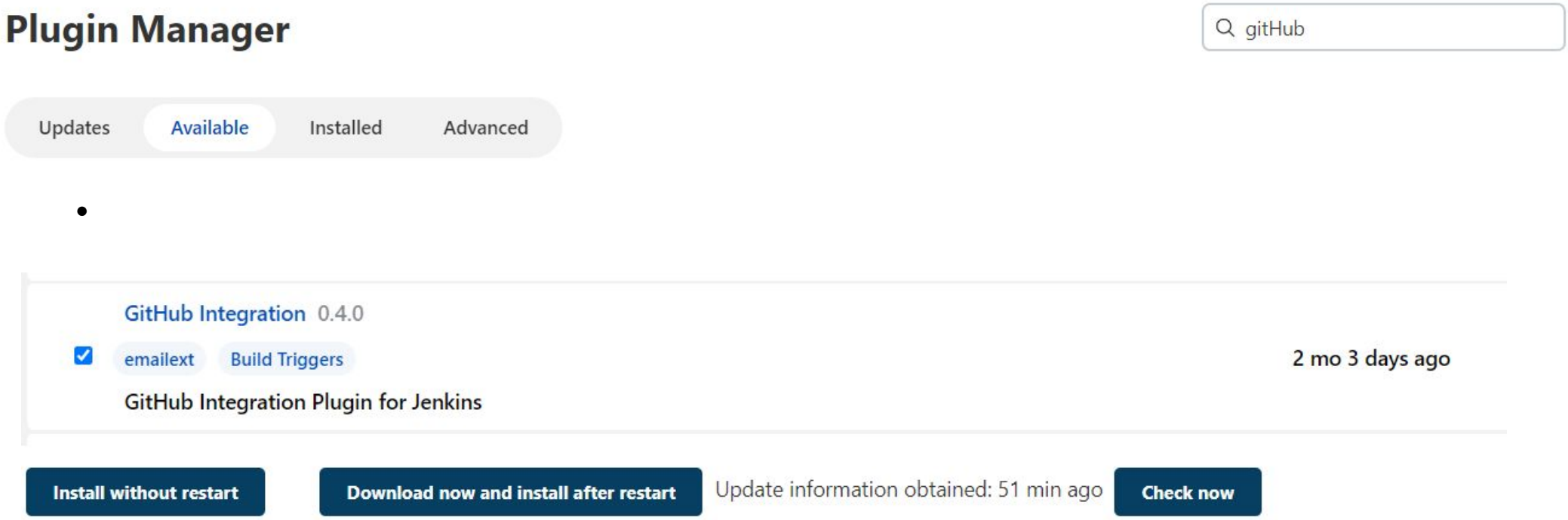
Git hub setup in Jenkins.

- Click on Manage Jenkins in the main dashboard and click on Manage Plugins



Git hub setup in Jenkins.

- Click on Available tab and enter github and



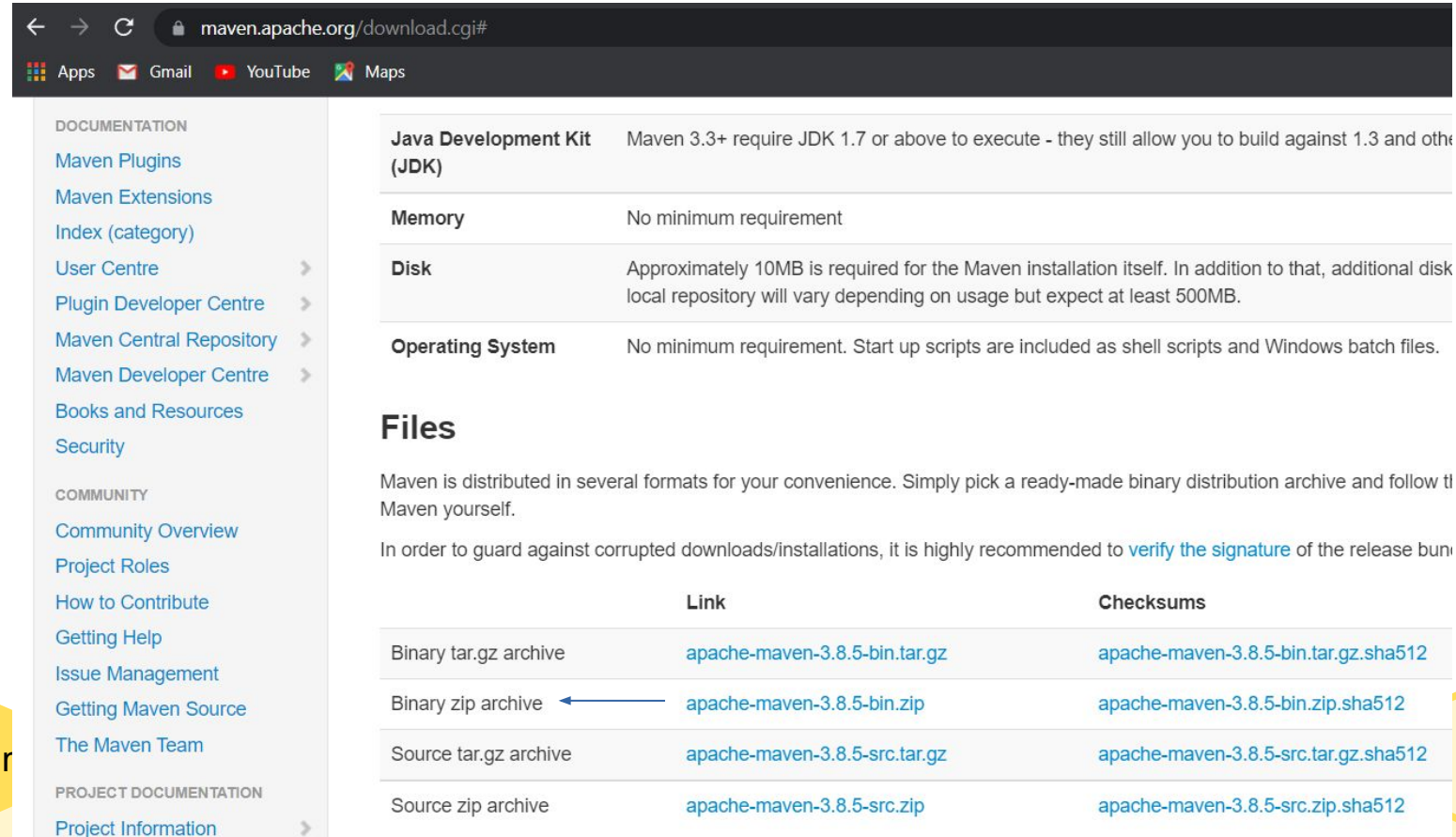
Git hub setup in Jenkins.

- From the jenkins dashboard, select create a job.

- Enter a name and select Free Style Project.

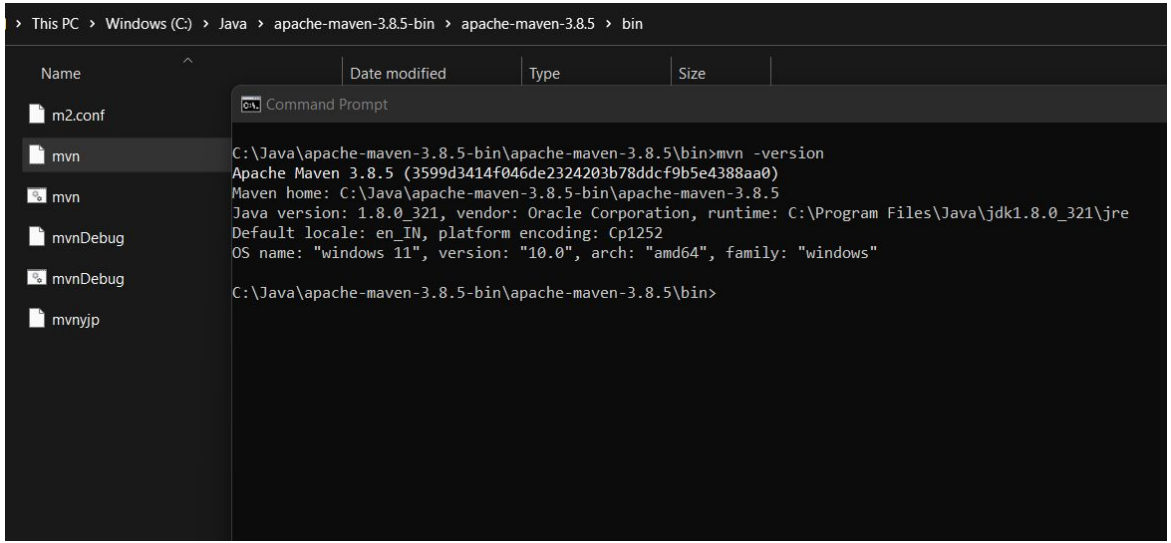
Maven setup in Jenkins.

- Goto the url maven.apache.org and click on donwload and download the binaries in zip format. After downloading, extract to maven directory.



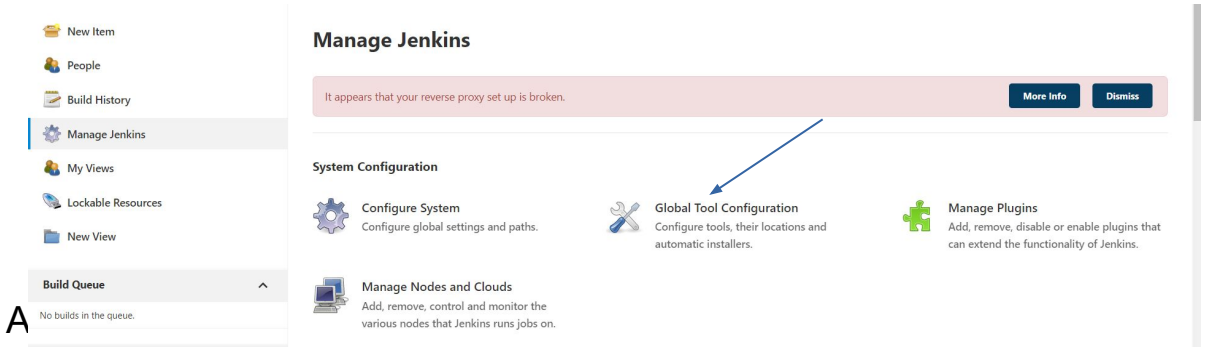
Maven setup in Jenkins.

- Given mvn -version in the command prompt by going to bin directory of maven.

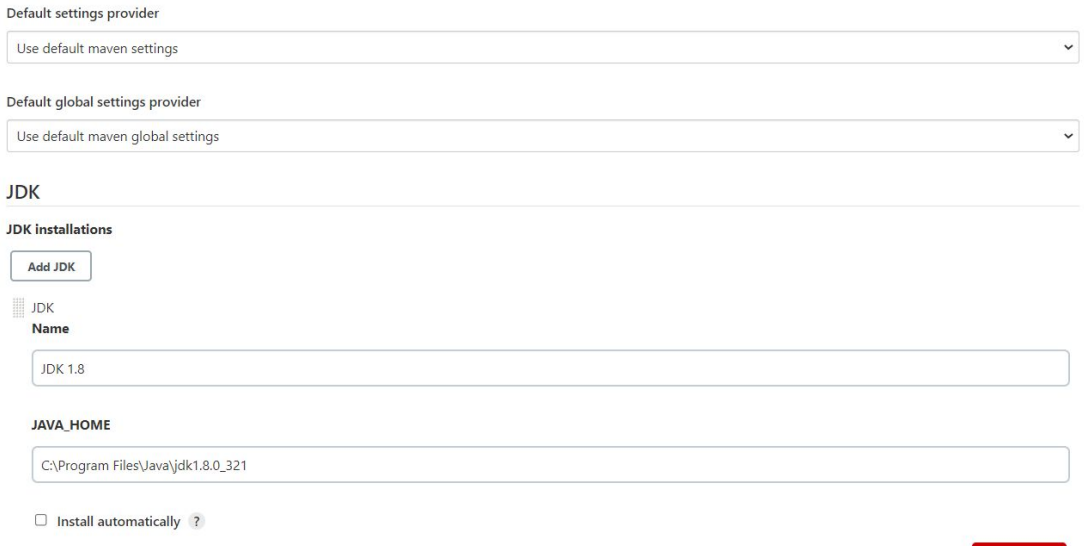


Maven setup in Jenkins.

- Select Global Tool Configuration in Main Dashboard.

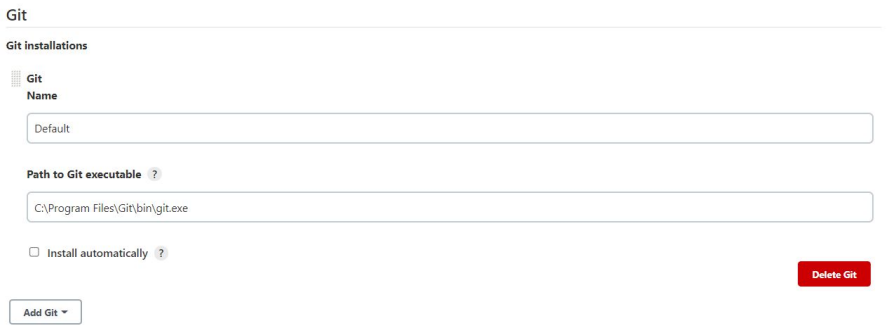


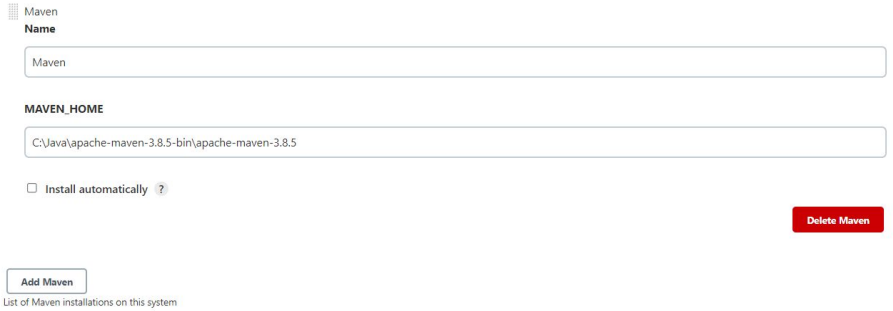
A



Maven setup in Jenkins.

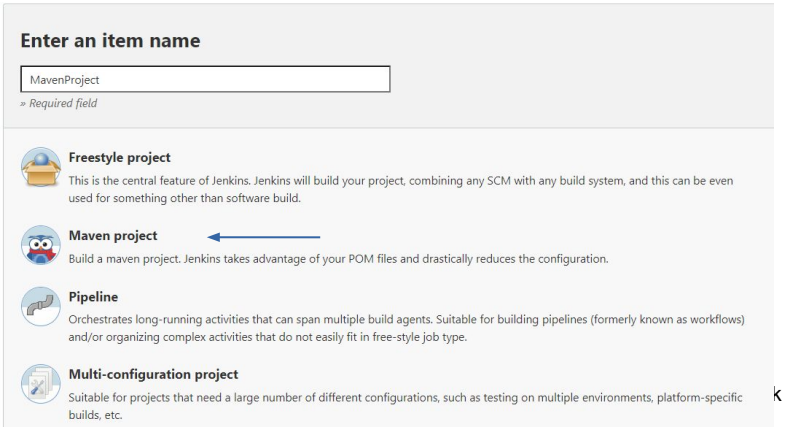
- Add git information:





Maven setup in Jenkins.

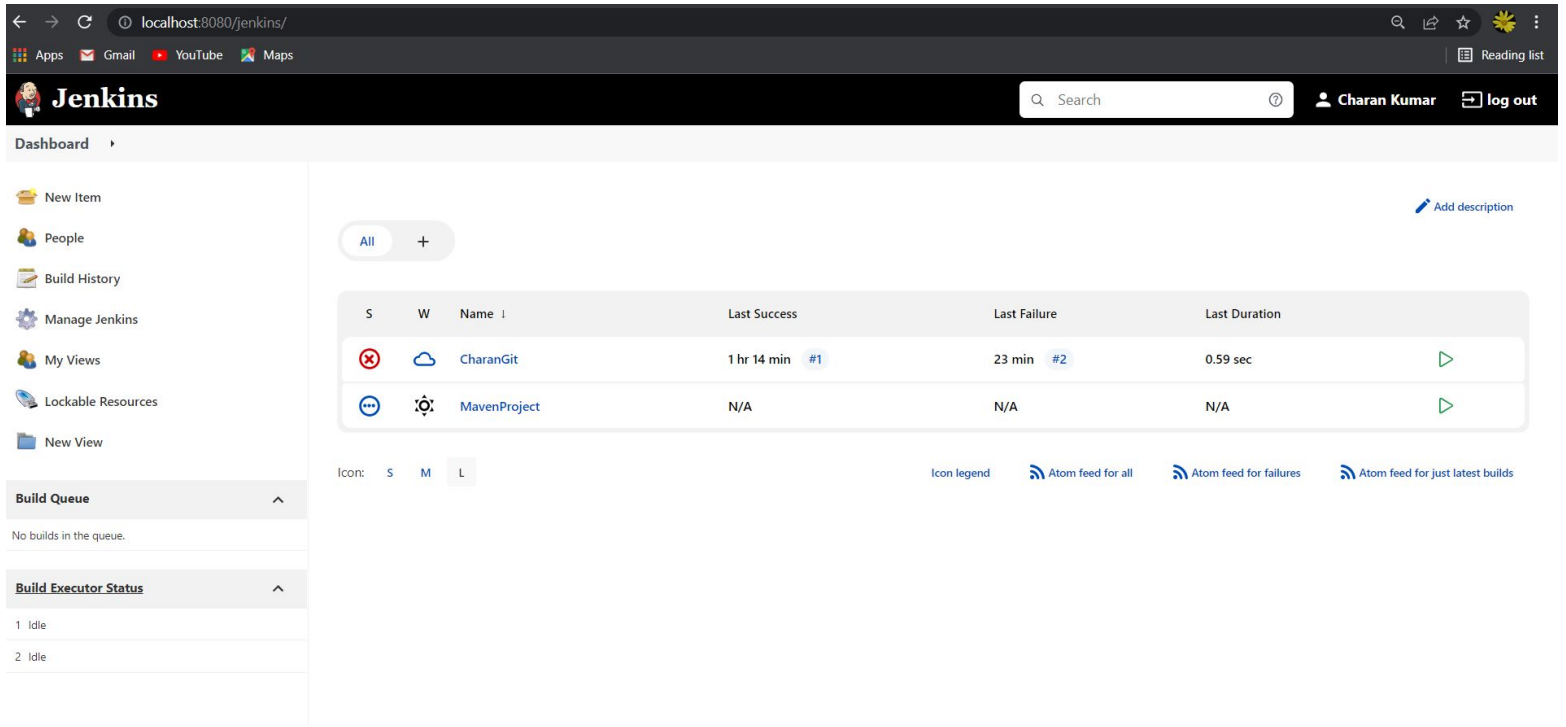
- If Maven is not seen as part of build then install maven plug-in by going to Manage Plugins –> Aavailable --> Entet Maven in find --> Select the Maven plugin and install it.
- Select New Item from Dashboard and enter Name as MavenProject and Select Maven Project is the option.



- E k on OK.

Jenkins Dashboard.

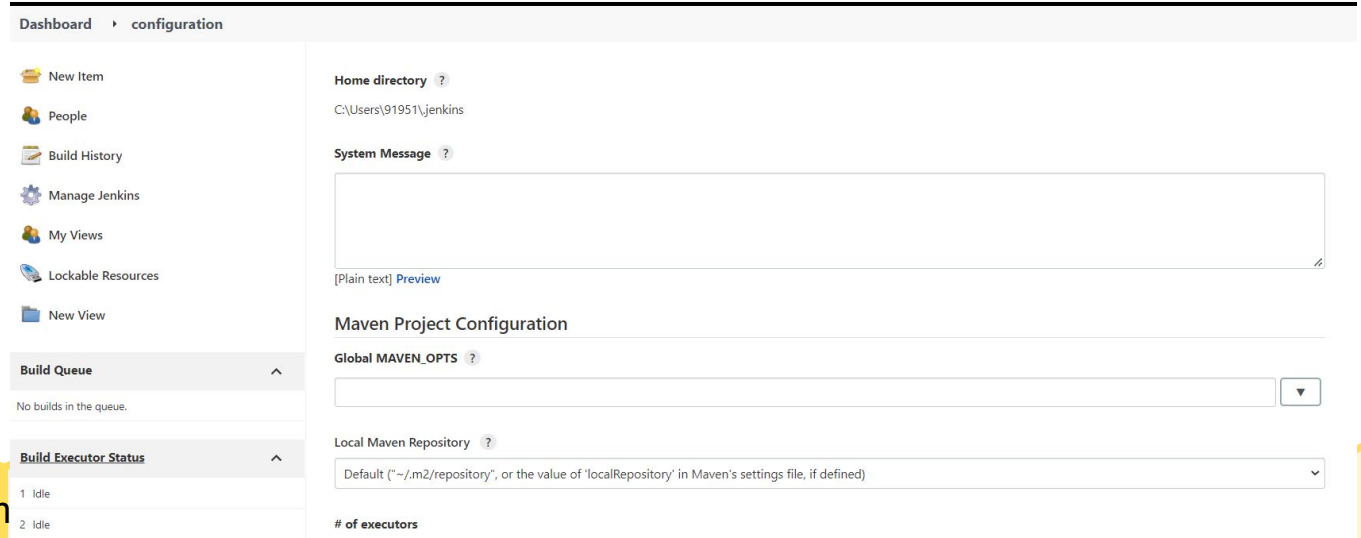
- After adding the projects for Git and Maven, the dashboard looks like this:



Jenkins Configuration.

- From the dashboard, click on **Manage Jenkins** and select **Configure System**.

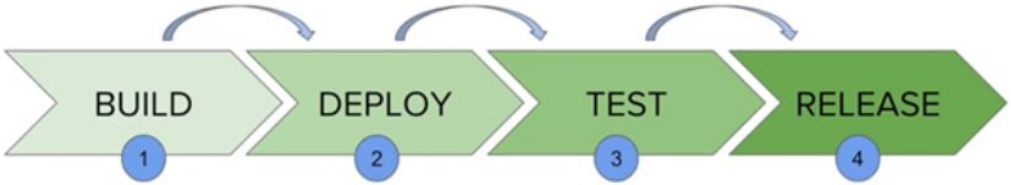
- This has all the configuration information about Jenkins.



Jenkins Management.

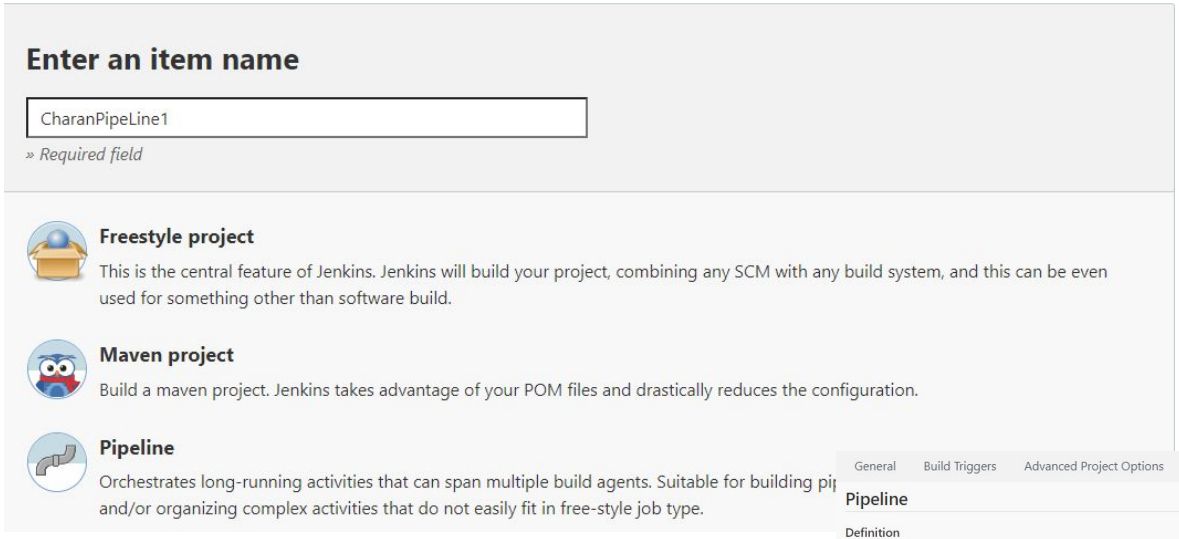
- When Manage Jenkins is clicked on the left hand side of the dashboard, following options are displayed.
- (1)Configuration System – It is used to configure Jenkins Locaiton, Setting up variables, Setting up Email Servers and many more.
- (2) Global Tool Configuration – It sets up the installed location for JDK, GIT and Maven.
- (3) Manage Plugin – New Plugins can be added/deleted and updated.
- (4) Manage Node and clouds – For the management of new Nodes.
- (5) Security – Configure Global Security -
- (6) Security – Manage Credentials -
- (7) Security – Configure credential Providers.
- (8) Security – Manage Users
- (9) Status – System Information – Provides the environmental information like env variables etc., Helps for troubleshooting.
- (10) Status – System Log – Logs the information from java.util.Logging.
- (11) Status – Load Statiscs – Gives the information about how much the system is loaded.
- (12) Status – About Jenkins.
- (13) Manage Old Data.

Jenkins - Pipeline

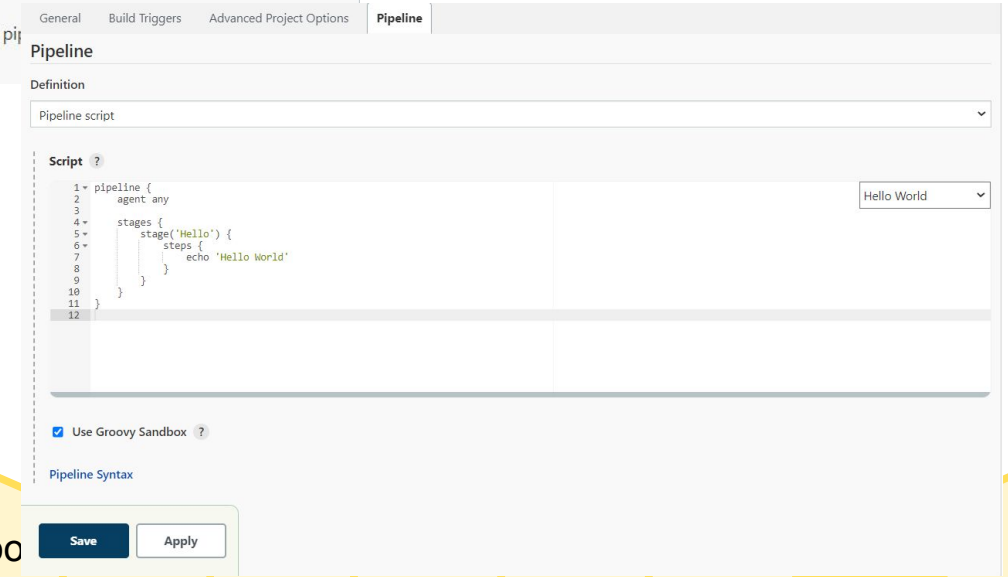
- In Jenkins, a pipeline is a collection of events or jobs which are interlinked with one another in a sequence.
- It is a combination of plugins that support the integration and implementation of continuous delivery pipelines using Jenkins.
- 

- These are phases of the build. i.e., the build the project, deploy on server, Test it with selenium and Release it to the customer. In each phase some action to be taken. A build script or Jenkins script file is created so that each phase action is taken and goes to the next action / next phase of action.
- A pipeline can be created in the following manner.
- Select the create a new job, give a name and select a pipeline.

Jenkins - Pipeline



- Jenkins pipeline script for hello world is given :



Jenkins - Pipeline

- Format of the Jenkins pipeline file is

```
pipeline {
    agent any
    stages {
        stage ('Build') {
            ...
        }
        stage ('Test') {
            ...
        }
        stage ('QA') {
            ...
        }
        stage ('Deploy') {
```

Jenkins - Pipeline

- Sample build script for github and maven is

Script ?

try sample Pipeline...

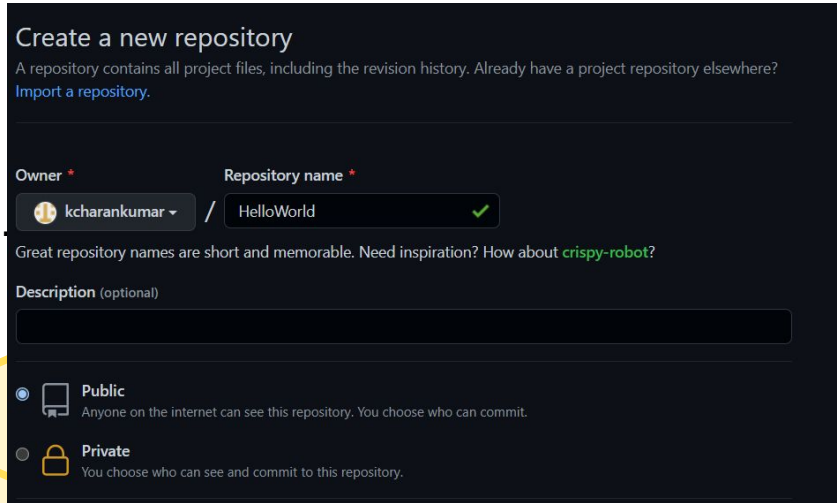
```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"

                // To run Maven on a Windows agent, use
                // bat "mvn -Dmaven.test.failure.ignore=true clean package"
            }

            post {
                // If Maven was able to run the tests, even if some of the test
                // failed, record the test results and archive the jar file.
                success {
                    junit '**/target/surefire-reports/TEST-*.xml'
                    archiveArtifacts 'target/*.jar'
                }
            }
        }
    }
}
```

Jenkins – Build setup

- Create a GitRepo directeory in Git.
- Create HelloWorld.java and compile it.

```
C:\Program Files\Git>cd \java

C:\Java>cd GitRepo

C:\Java\GitRepo>dir
 Volume in drive C is Windows
 Volume Serial Number is D87D-3DAA

 Directory of C:\Java\GitRepo

15-03-2022  17:39    <DIR>          .
15-03-2022  17:38    <DIR>          ..
15-03-2022  17:39               121 HelloWorld.java
               1 File(s)            121 bytes
               2 Dir(s)  404,194,033,664 bytes free

C:\Java\GitRepo>javac HelloWorld.java

C:\Java\GitRepo>java HelloWorld
Hello World...

C:\Java\GitRepo>
```

- Create a New Repository as HelloWorld in GitHub..



Jenkins – Build setup

- Open the git_cmd and execute the following commands. This will create a repository, adds the files in the repository and pushes them to github.
- Git init
- Git status
- Git add . --> Adds both the files in the repository
- Git status
- git config --global user.email "kcharankumar@gmail.com"
- git config --global user.name "K.Charan Kumar"
- Git commit -m “HelloWorld Program is added”
- git remote add origin https://github.com/kcharankumar/HelloWorld.git
- git push -u origin master

Jenkins – Build setup

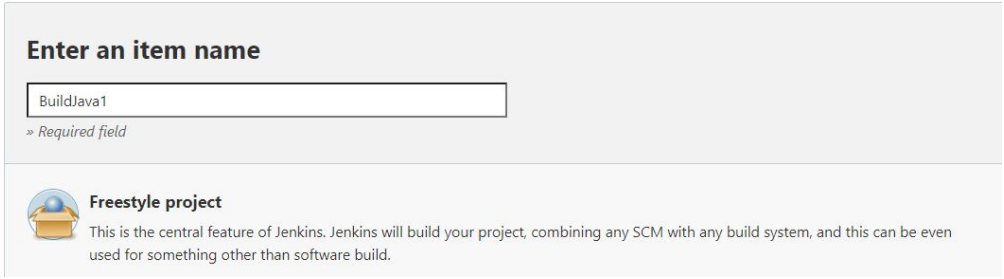
-

```
C:\Program Files\Git\git-cmd.exe

C:\Java\GitRepo>git init
Initialized empty Git repository in C:/Java/GitRepo/.git/

C:\Java\GitRepo>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        HelloWorld.class
        HelloWorld.java

nothing added to commit but untracked files present (use "git add" to track)

C:\Java\GitRepo>git add .

C:\Java\GitRepo>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   HelloWorld.class
        new file:   HelloWorld.java


C:\Java\GitRepo>git config --global user.email "kcharankumar@gmail.com"

C:\Java\GitRepo>git config --global user.name "K.Charan Kumar"

C:\Java\GitRepo>git commit -m
error: switch `m' requires a value

C:\Java\GitRepo>git commit -m "Added HelloWorld Program"
[master (root-commit) 95076a9] Added HelloWorld Program
 2 files changed, 10 insertions(+)
 create mode 100644 HelloWorld.class
 create mode 100644 HelloWorld.java

C:\Java\GitRepo>git remote add origin https://github.com/kcharankumar/HelloWorld.git

C:\Java\GitRepo>git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
```

Jenkins – Build setup

```
C:\Java\GitRepo>git remote add origin https://github.com/kcharankumar/HelloWorld.git

C:\Java\GitRepo>git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 660 bytes | 660.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/kcharankumar/HelloWorld.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Java\GitRepo>
```

Jenkins – Build setup

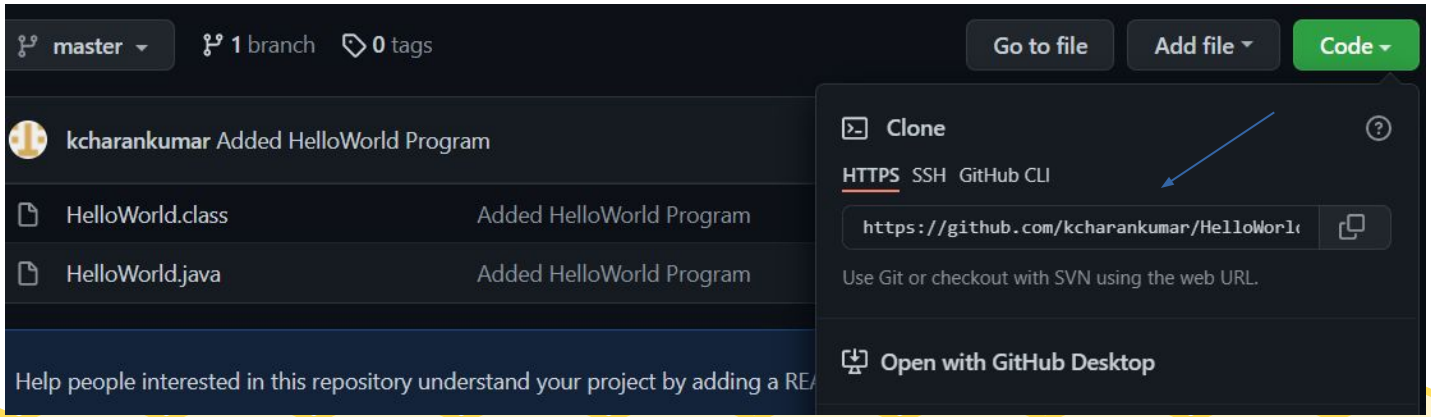
- From the jenkins dashboard, select New Item
- Enter a **name** and select **Free Style Project** and click on **OK.**
- 

- Enter details in General Section :
- 

Jenkins – Build setup

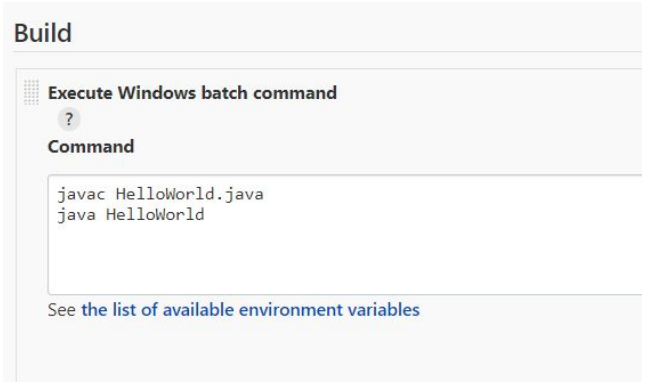
- Enter git details in Source Code Repositories.

- Git location can be picked up from GitHub from here.

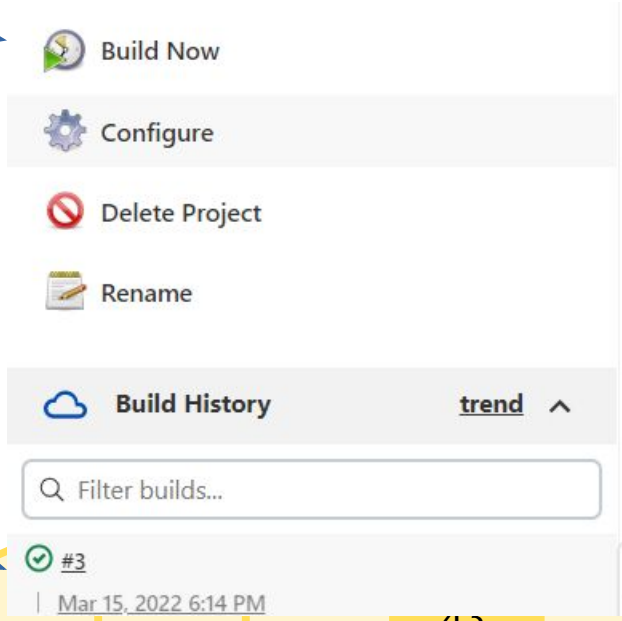


Jenkins – Build setup

- Under Build --> Select **Execute Windows Batch command** and enter the following and click on **Save and Apply**.

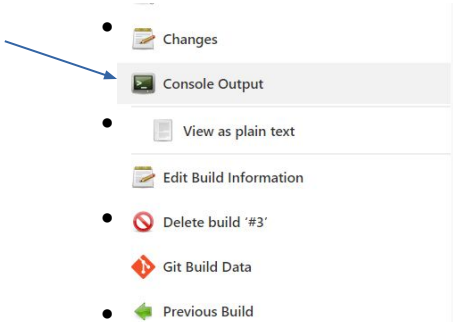


- From Dashboard of BuildJava Project, click on Build Now.
- It will build and gives you the result.



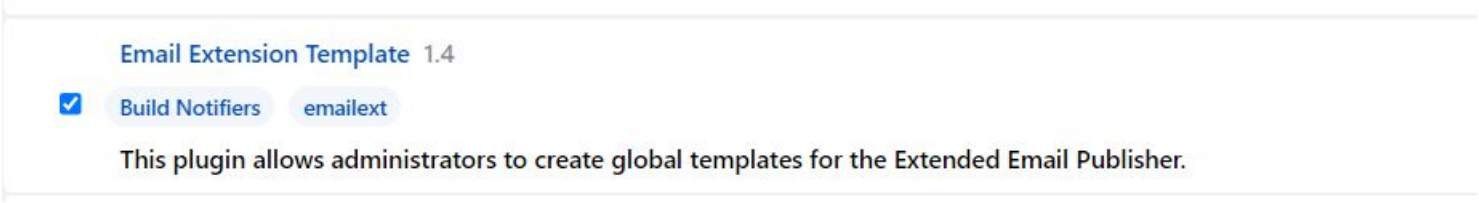
Jenkins – Build setup

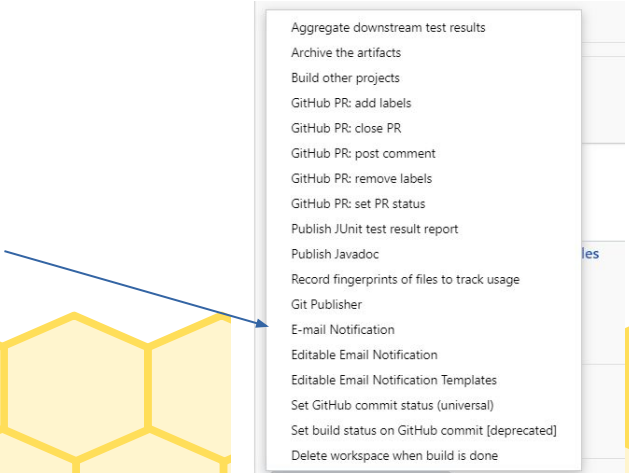
- After clicking on the link #3 or #2 or #1 as per the number of times the build is made,it opens another dashboard. In that select Console output. It shows the following.




```
Started by user Charan Kumar
Running as SYSTEM
Building in workspace C:\Users\91951\.jenkins\workspace\BuildJava
The recommended git tool is: NONE
No credentials specified
 > C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\Users\91951\.jenkins\workspace\BuildJava\.git # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/kcharankumar/HelloWorld.git # timeout=10
Fetching upstream changes from https://github.com/kcharankumar/HelloWorld.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > git --version # 'git version 2.35.1.windows.2'
 > C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/kcharankumar/HelloWorld.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 95076a9391689cc9d3bdddfbfd04bf1739b4208b (refs/remotes/origin/master)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f 95076a9391689cc9d3bdddfbfd04bf1739b4208b # timeout=10
Commit message: "Added HelloWorld Program"
 > C:\Program Files\Git\bin\git.exe rev-list --no-walk 95076a9391689cc9d3bdddfbfd04bf1739b4208b # timeout=10
[BuildJava] $ cmd /c call C:\Java\apache-tomcat-9.0.59-windows-x64\apache-tomcat-9.0.59\temp\jenkins3047393363687644095.bat
The following environment variables were removed by 'Only Keep Specified Environment Variables': , BUILD_DISPLAY_NAME, BUILD_URL, CI, EXECUTOR_NUMBER, GIT_BRANCH, GIT_COMMIT,
GIT_PREVIOUS_COMMIT, GIT_PREVIOUS_SUCCESSFUL_COMMIT, GIT_URL, HUDSON_HOME, HUDSON_URL, JENKINS_HOME, JENKINS_URL, JOB_DISPLAY_URL, JOB_URL, NODE_LABELS, NODE_NAME,
RUN_ARTIFACTS_DISPLAY_URL, RUN_CHANGES_DISPLAY_URL, RUN_DISPLAY_URL, RUN_TESTS_DISPLAY_URL, WORKSPACE, WORKSPACE_TMP
The following environment variables were reset to their system default value by 'Only Keep Specified Environment Variables': Path

C:\Users\91951\.jenkins\workspace\BuildJava>javac HelloWorld.java
'javac' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\91951\.jenkins\workspace\BuildJava>java HelloWorld
Hello World...

C:\Users\91951\.jenkins\workspace\BuildJava>exit 0
Finished: SUCCESS
```

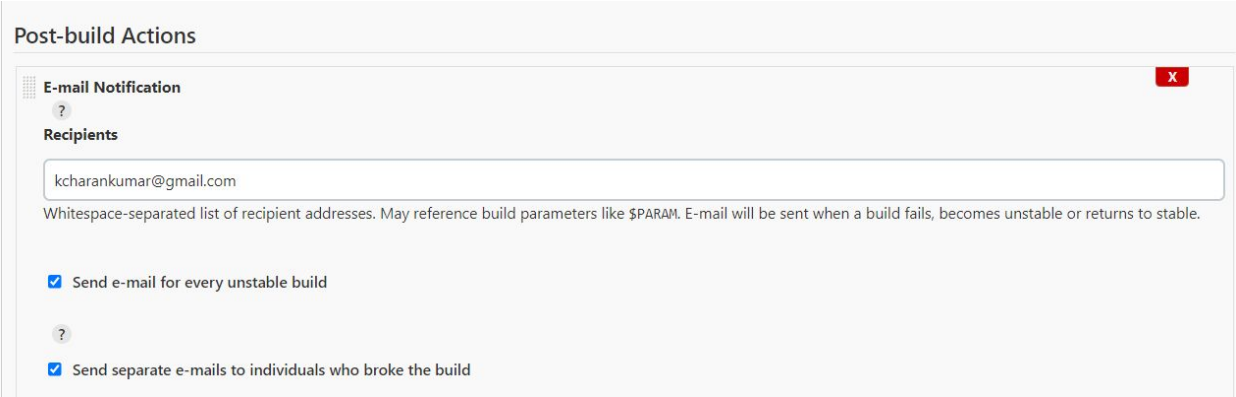
Jenkins – Notification

- Email notification from Jenkins
- 1. Go to **Manage Jenkins** and click on **Manage Plug-ins**
- 2. Click on **Available** and enter **Email** in the search box and install the selected one.
- 

- Select the **BuildJava Project** --> Click on **Configure** --> select post build operations as **Email Notification**.
- 

Jenkins – Notification

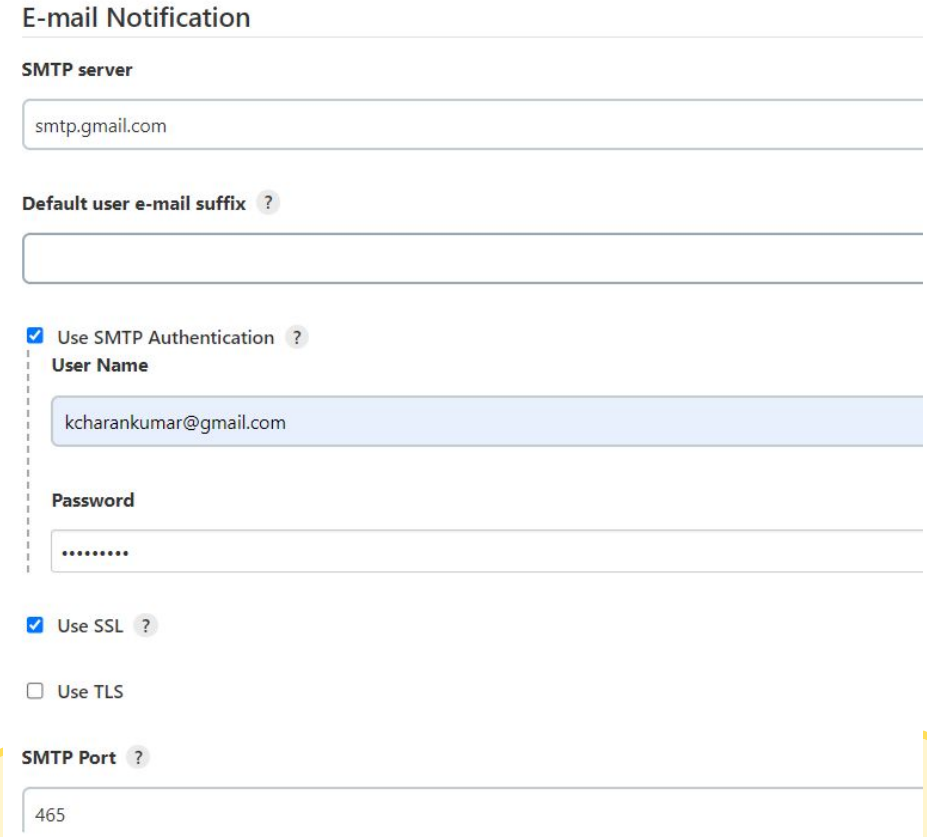
- Enter the email id and click on when the build is broken and click on Apply and Save.



Jenkins – Notification

Click on **Manage Jenkins** from Dashboard --> **Configure System**.

- Go to the section Email Notification --> click on Advanced and enter the following information.
- SMTP Server : smtp.gmail.com
- UserName : kcharankumar@gmail.com
- Password : <password>
- Use SSL : Check this.
- Smtp port : 465
-
-
-



Jenkins – Notification

Click on test email notification. This will send test email to the given mail id.

- Make HelloWorld.java generate errors and make the build fail in local git as
- Removed the quotes in s.o.p.
-

```
class HelloWorld{

  public static void main (String args[]){

    System.out.println (Hello World...);

}

}
```

Jenkins – Notification

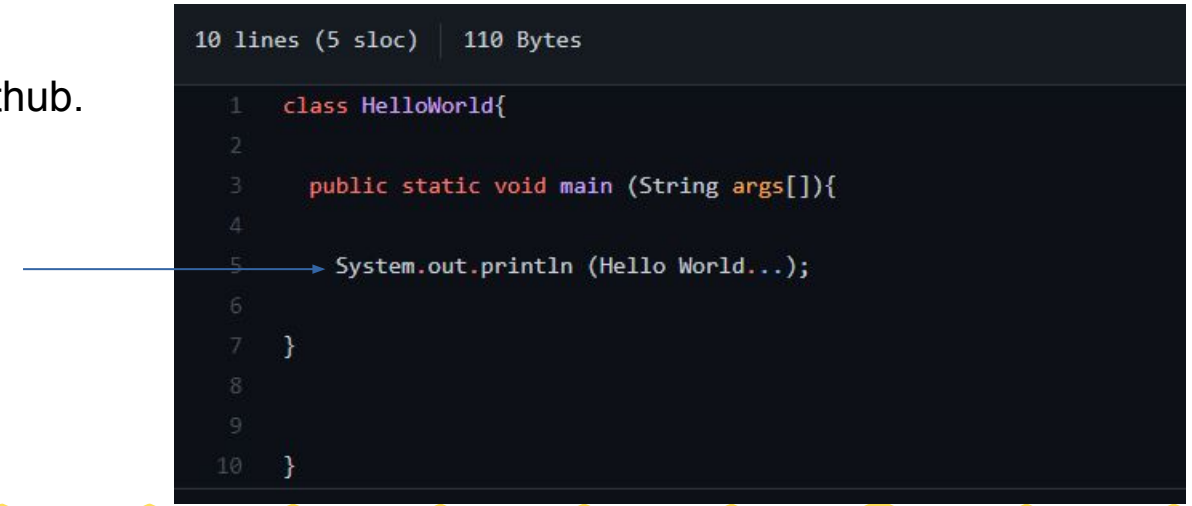
Run the following git commands.

Git add HelloWorld.java

Git commit -m “Removed the quotes”

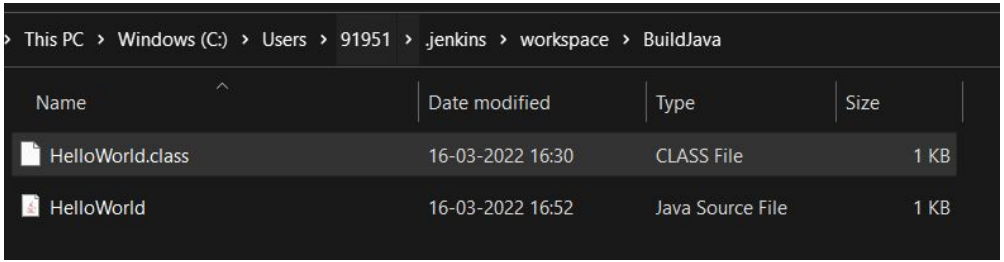
Git push -u origin master

Ensure the modified code is in github.



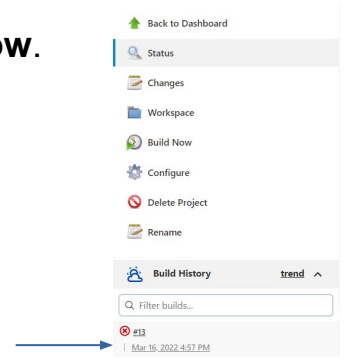
Jenkins – Notification

Remove the previously created HelloWorld.class:



Open Dashboard, click on **BuildJava Project** and click on **Build Now**.

Build Failed as shown in here.



Jenkins – Notification

Can click on the build link and click on Console Output. Following screen will be seen.

```
C:\Users\91951\.jenkins\workspace\BuildJava>javac HelloWorld.java
HelloWorld.java:5: error: ')' expected
    System.out.println (Hello World...);
                        ^
HelloWorld.java:5: error: illegal start of expression
    System.out.println (Hello World...);
                          ^
HelloWorld.java:5: error: ';' expected
    System.out.println (Hello World...);
                                ^
3 errors

C:\Users\91951\.jenkins\workspace\BuildJava>java HelloWorld
Error: Could not find or load main class HelloWorld

C:\Users\91951\.jenkins\workspace\BuildJava>exit 1
Build step 'Execute Windows batch command' marked build as failure
Sending e-mails to: kcharankumar@gmail.com
Finished: FAILURE
```

Email will be sent to the given email id: