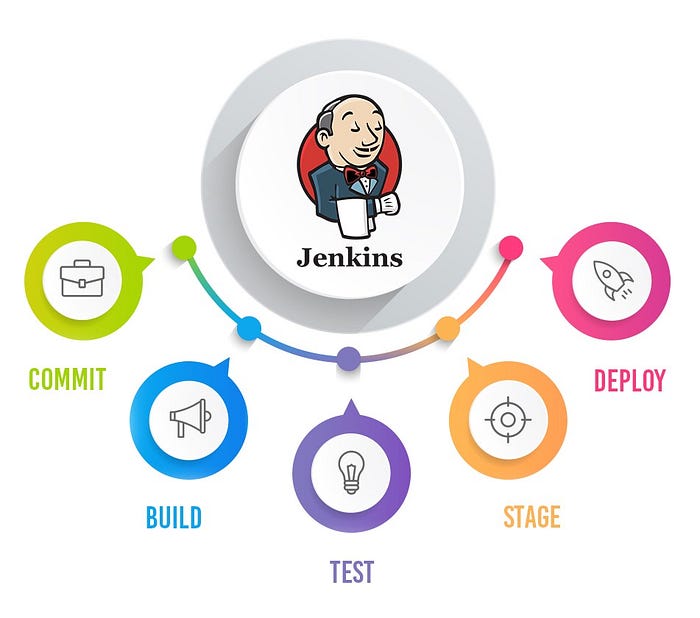
**Jenkins**

Jenkins is an open source [continuous integration](https://www.techtarget.com/searchsoftwarequality/definition/continuous-integration)/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the [Java](https://www.theserverside.com/definition/Java) programming language. It is used to implement CI/CD workflows, called pipelines.



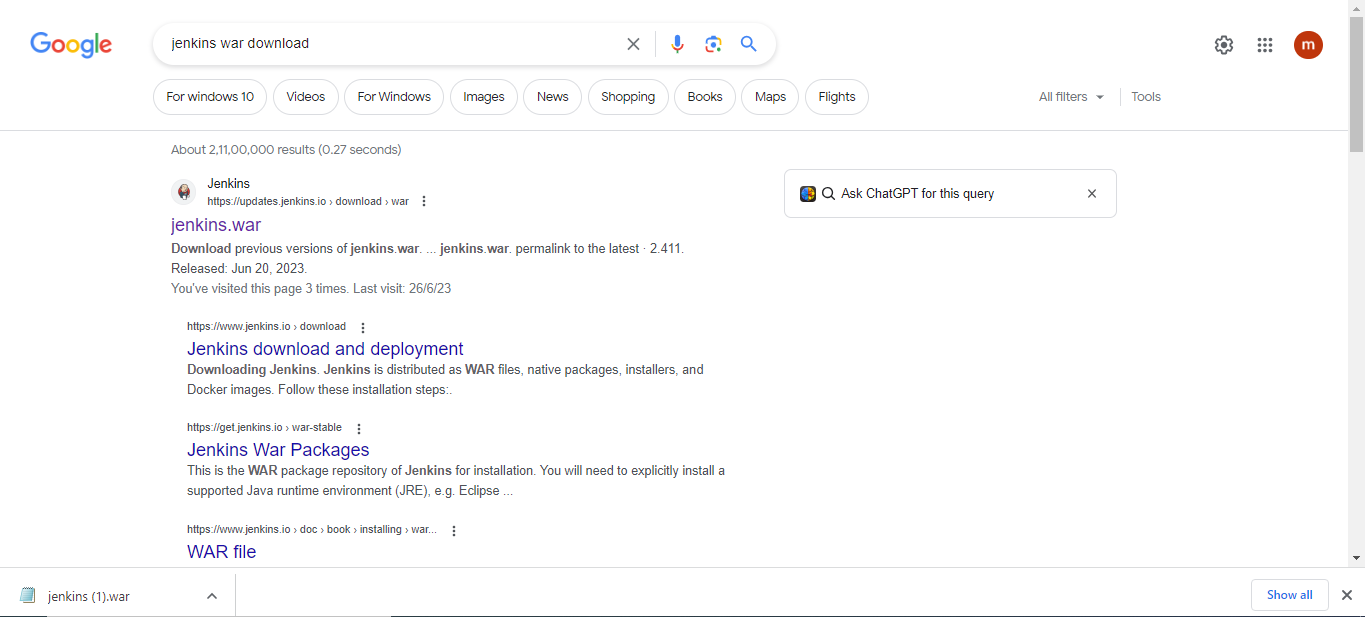
**How Jenkins works**

Jenkins runs as a server on a variety of platforms including Windows, MacOS, Unix variants and especially, [Linux](https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system). It requires a Java 8 VM and above and can be run on the Oracle JRE or [OpenJDK](https://www.theserverside.com/definition/OpenJDK). Usually, Jenkins runs as a Java servlet within a Jetty application server. It can be run on other Java application servers such as Apache Tomcat. More recently, Jenkins has been adapted to run in a [Docker container](https://www.techtarget.com/searchitoperations/definition/Docker). There are read-only Jenkins images available in the Docker Hub online repository.

### Plugins

A [plugin](https://www.techtarget.com/whatis/definition/plug-in) is an enhancement to the Jenkins system. They help extend Jenkins capabilities and integrated Jenkins with other software. Plugins can be downloaded from the online Jenkins Plugin repository and loaded using the Jenkins Web UI or CLI. Currently, the Jenkins community claims over 1500 plugins available for a wide range of uses.

How to build a spring boot application in Jenkins using Jenkinsfile

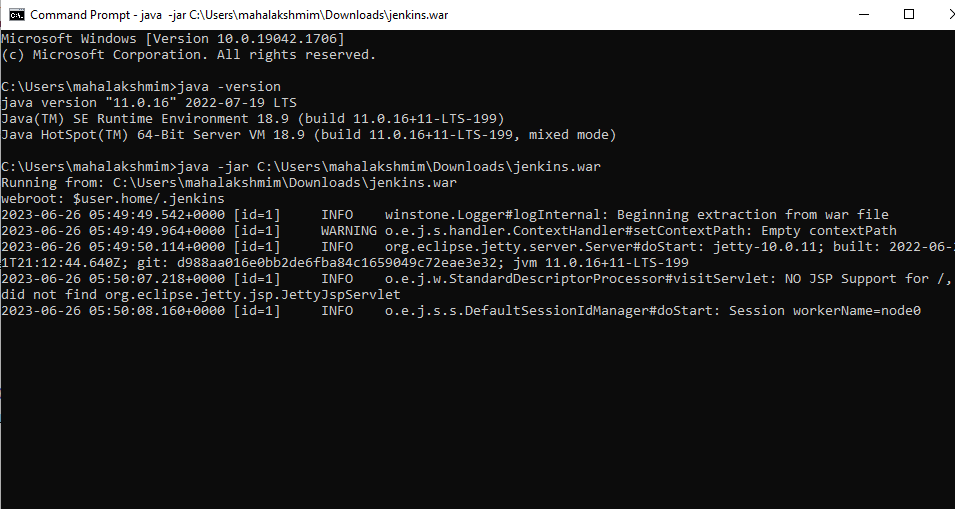
**Step1**:First step is to download the Jenkins war file

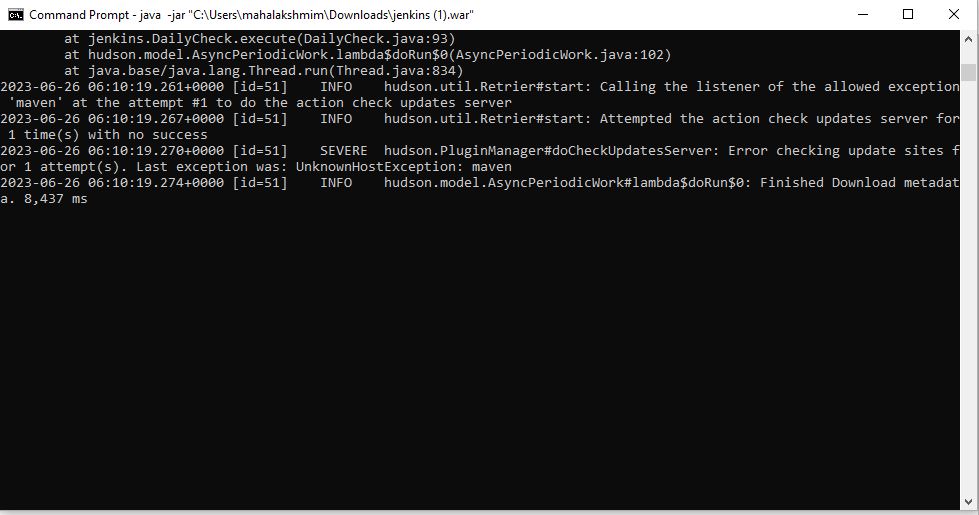
**Step2:** After downloaded it run the command in cmd

1.jenkins support on java version is above 1.8 in local system

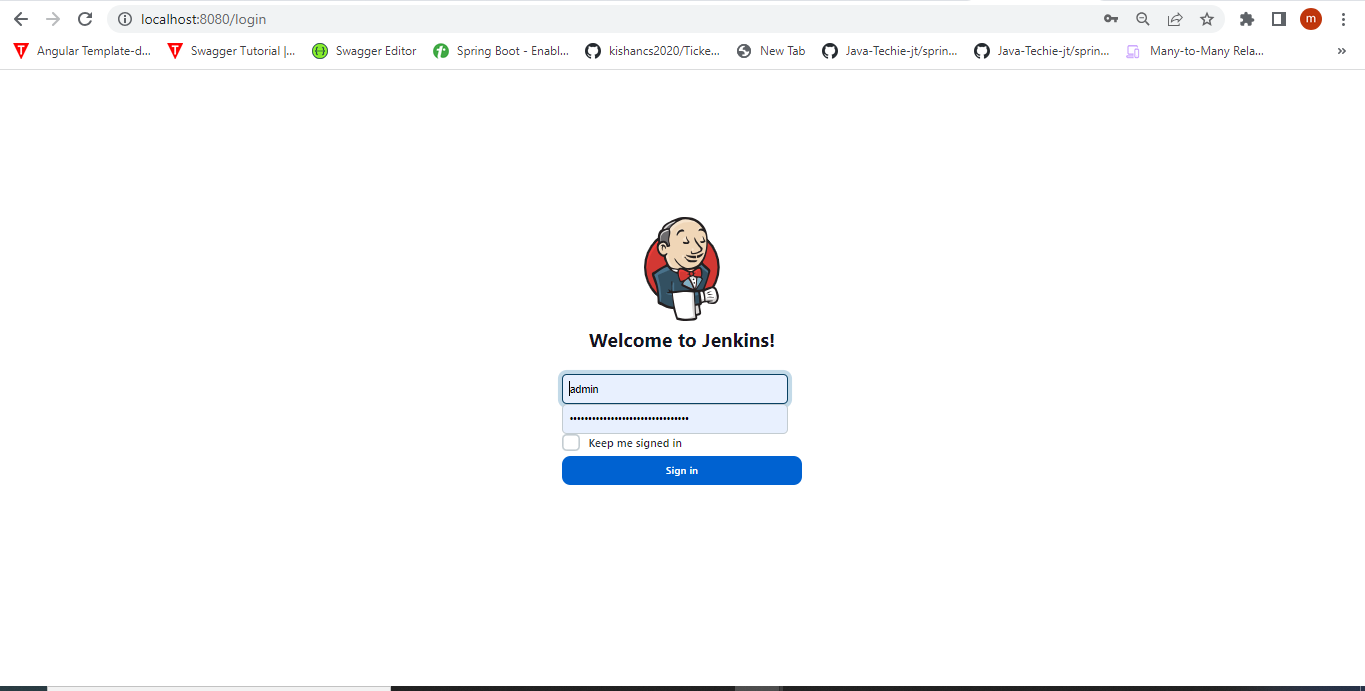
So check by using the command **java-version**

2.Next command is **java –jar <workspace of Jenkins war file>**

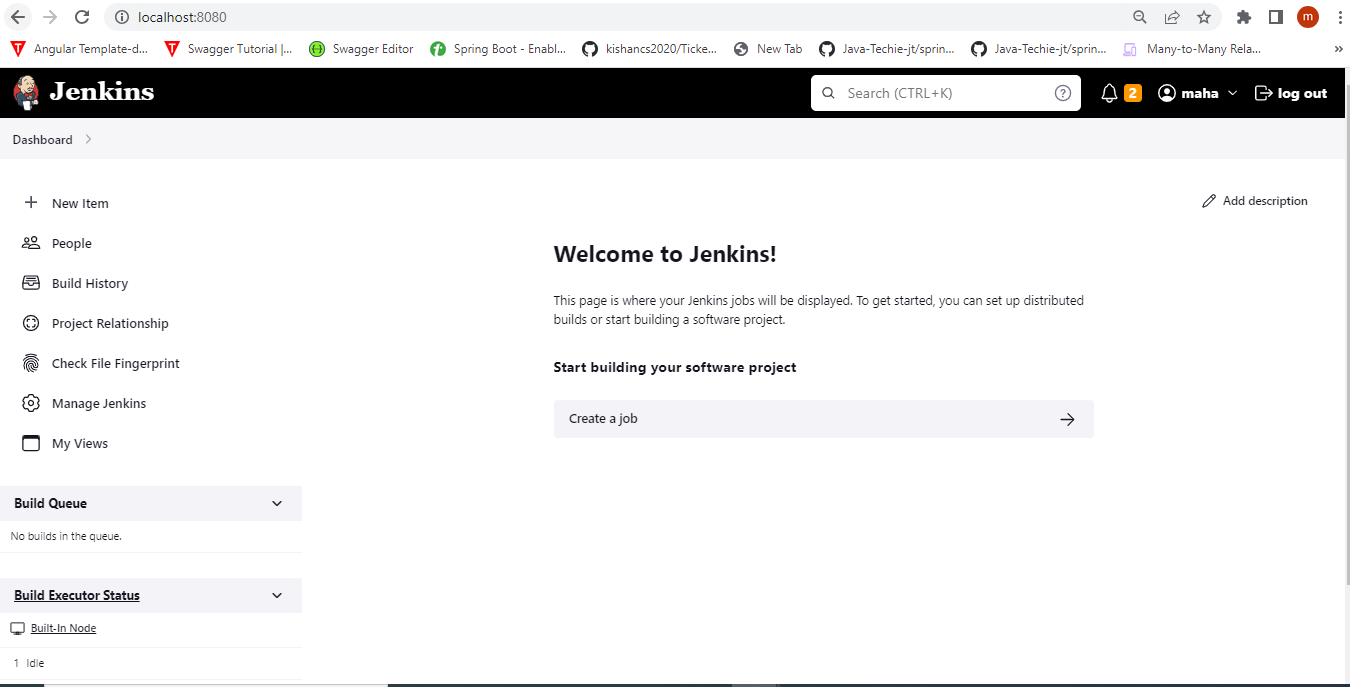




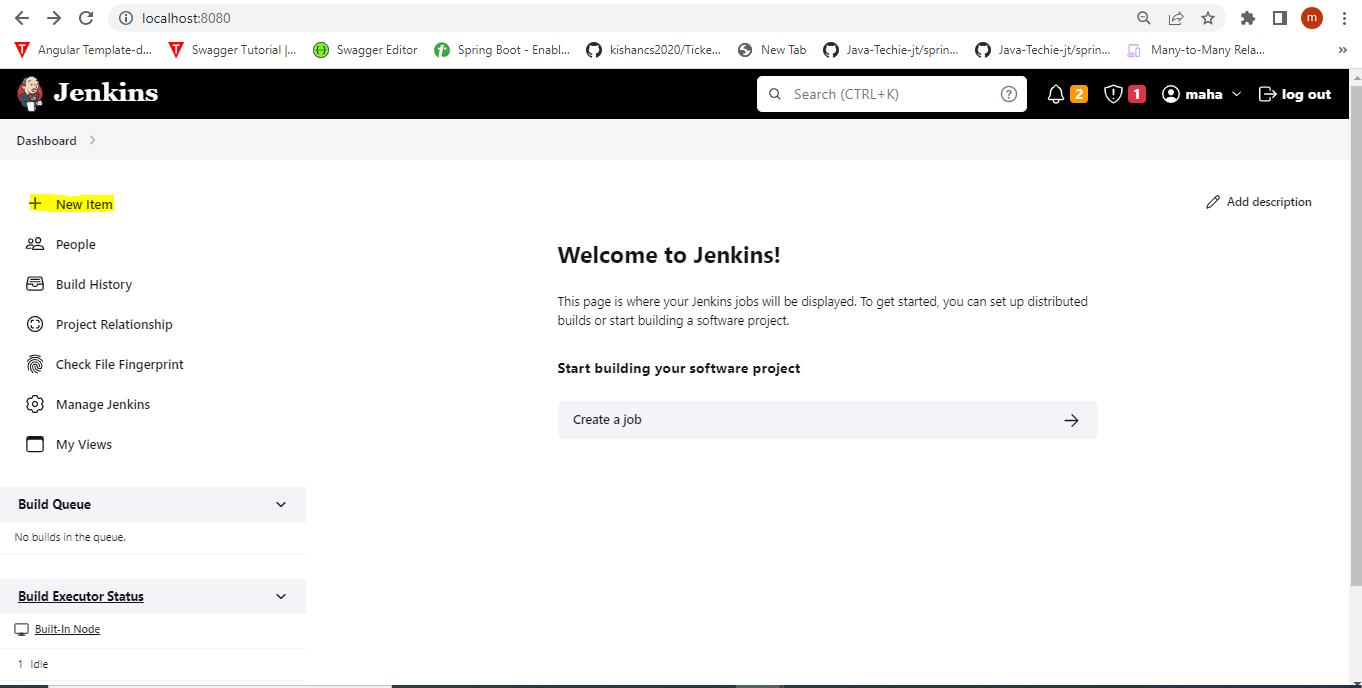
Once running successful means go and check the url http://localhost:8080/login



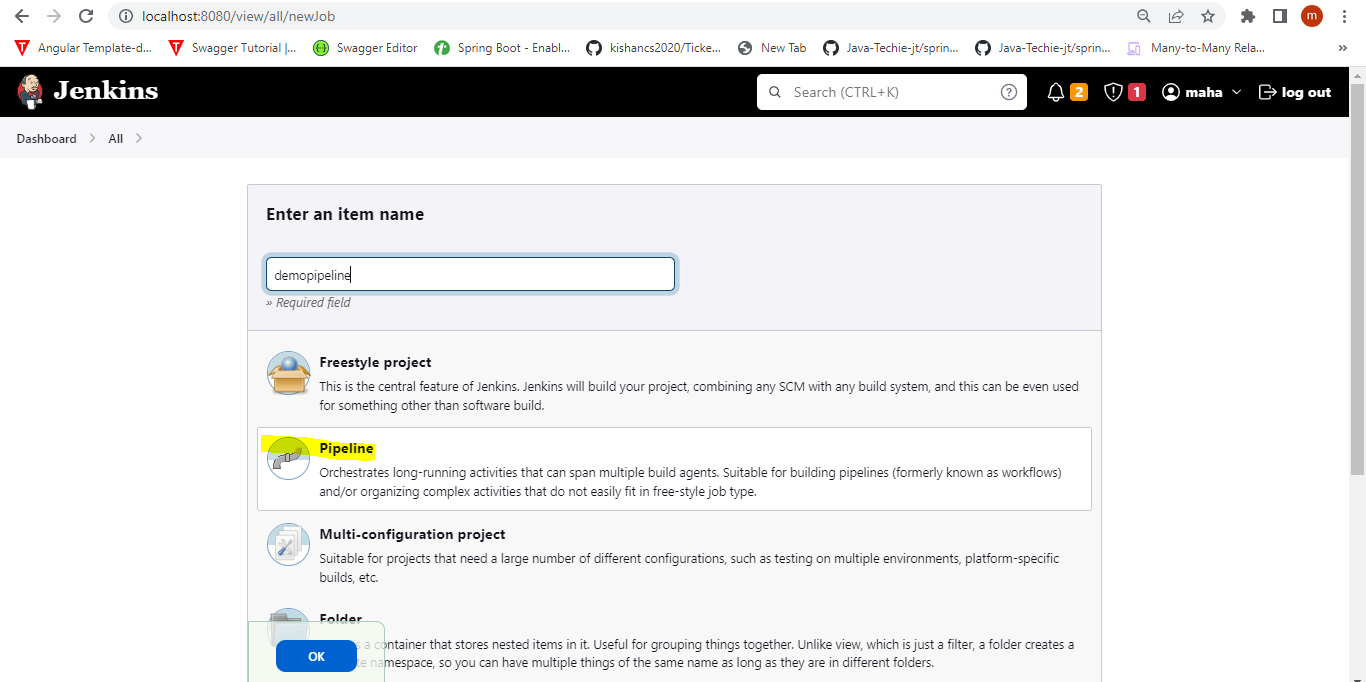
**Step3**:After login successfully it will move to the main page



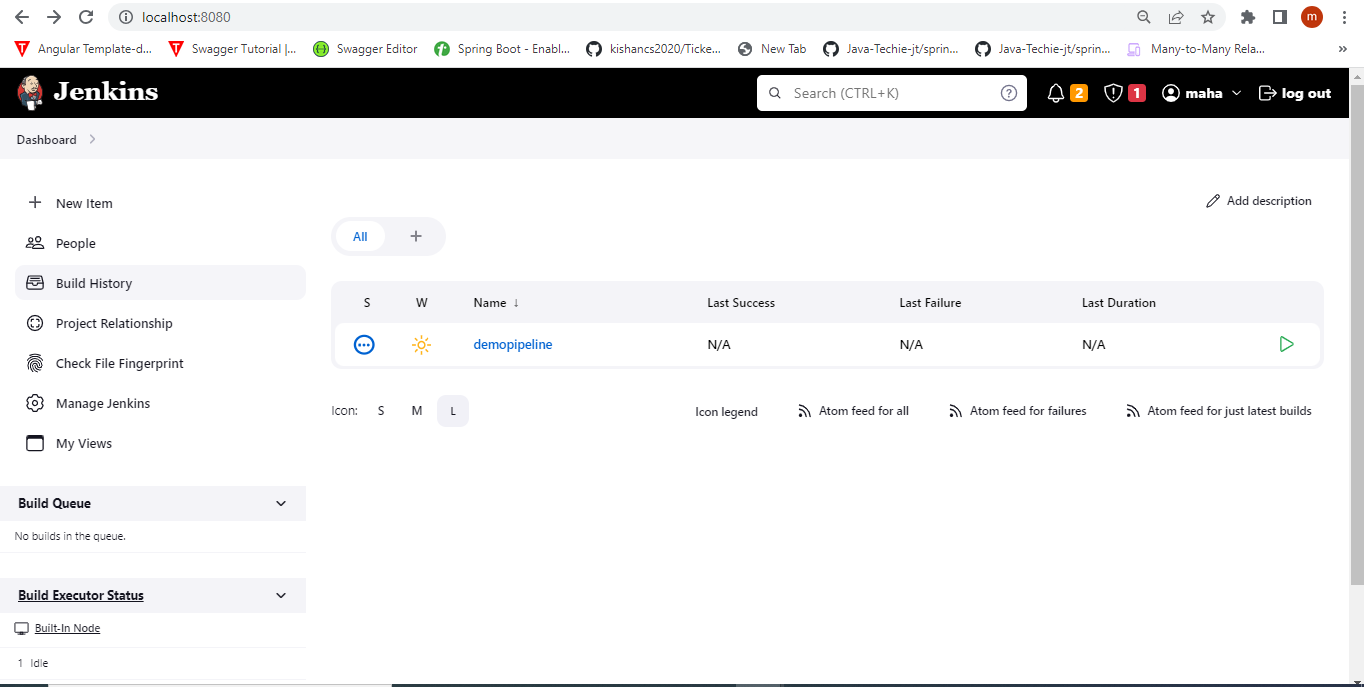
**Step4**: First in Jenkins need to create a Pipeline so,click on **New Item**



Here I am created one demopipeline project in pipeline

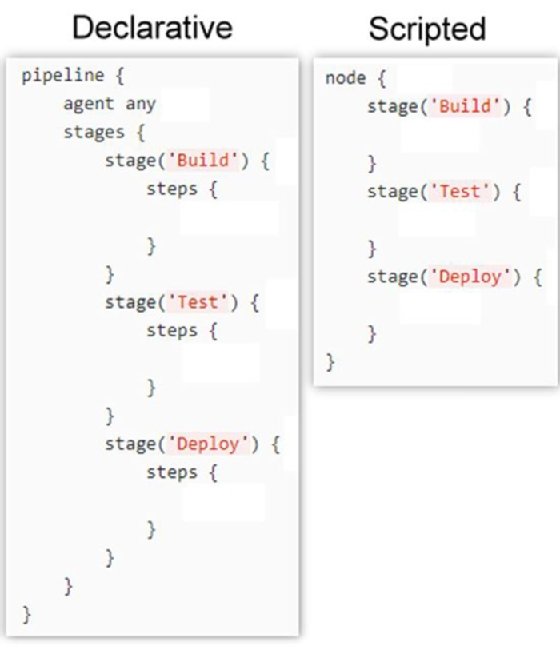


A project has been created based on the name demopipeline



In Jenkins there are two types of pipeline

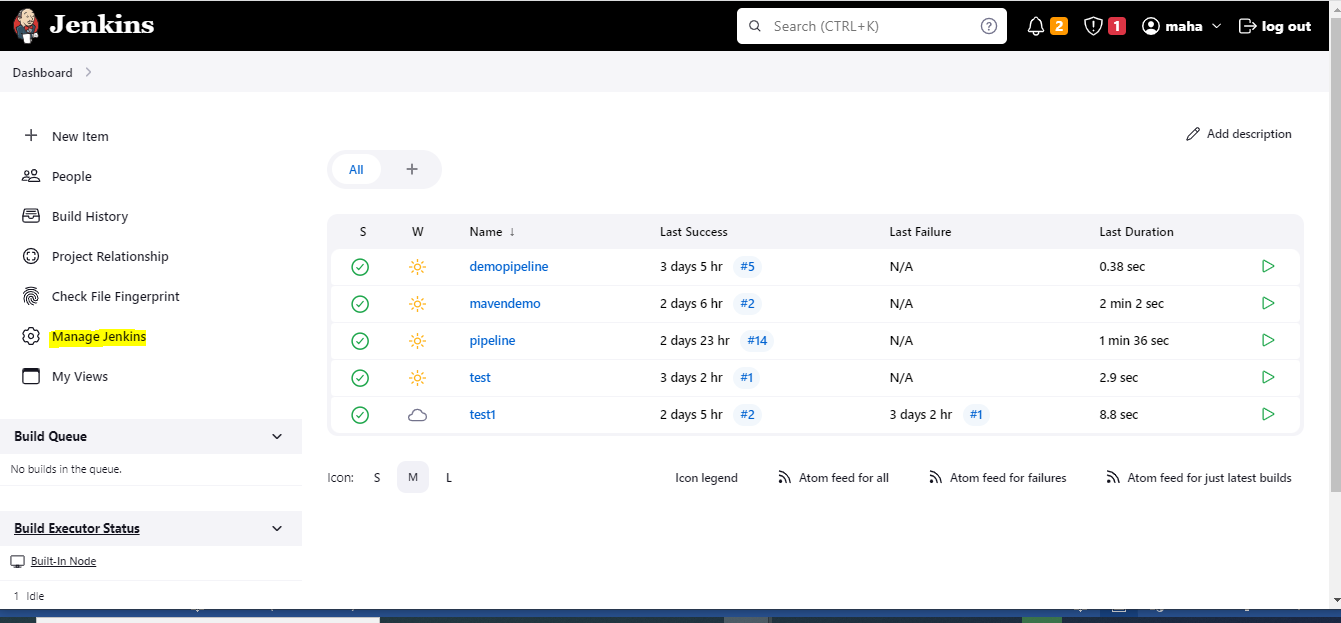
* Declarative Pipeline
* Scripted Pipeline



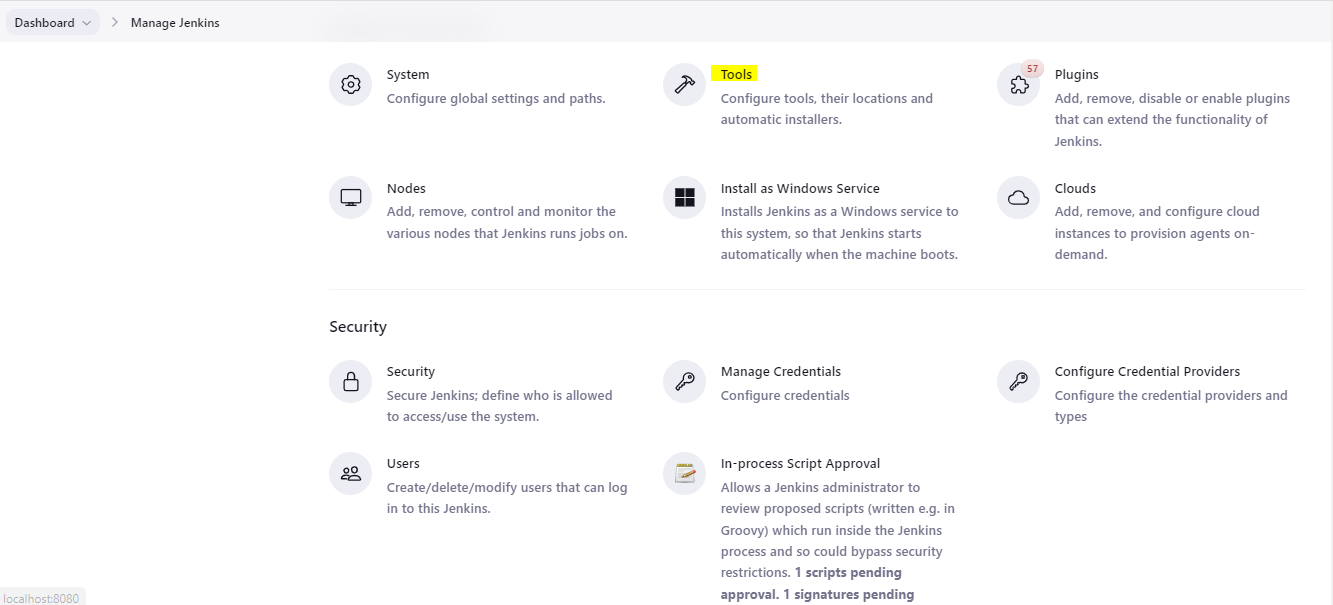
Here I am trying to push my code from Github and Build in Jenkins

**Step5**: Before going to running a pipeline in jenkins need to add the configuration for java,maven and git

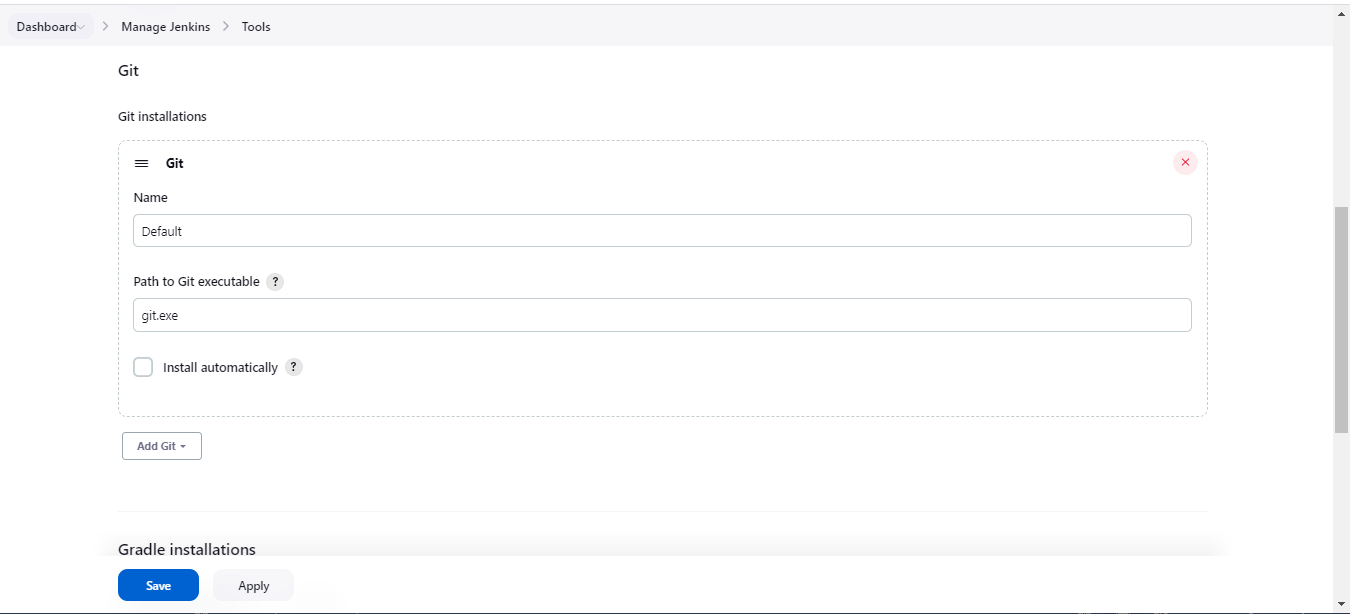
Choose the manage jenkins option in jenkins



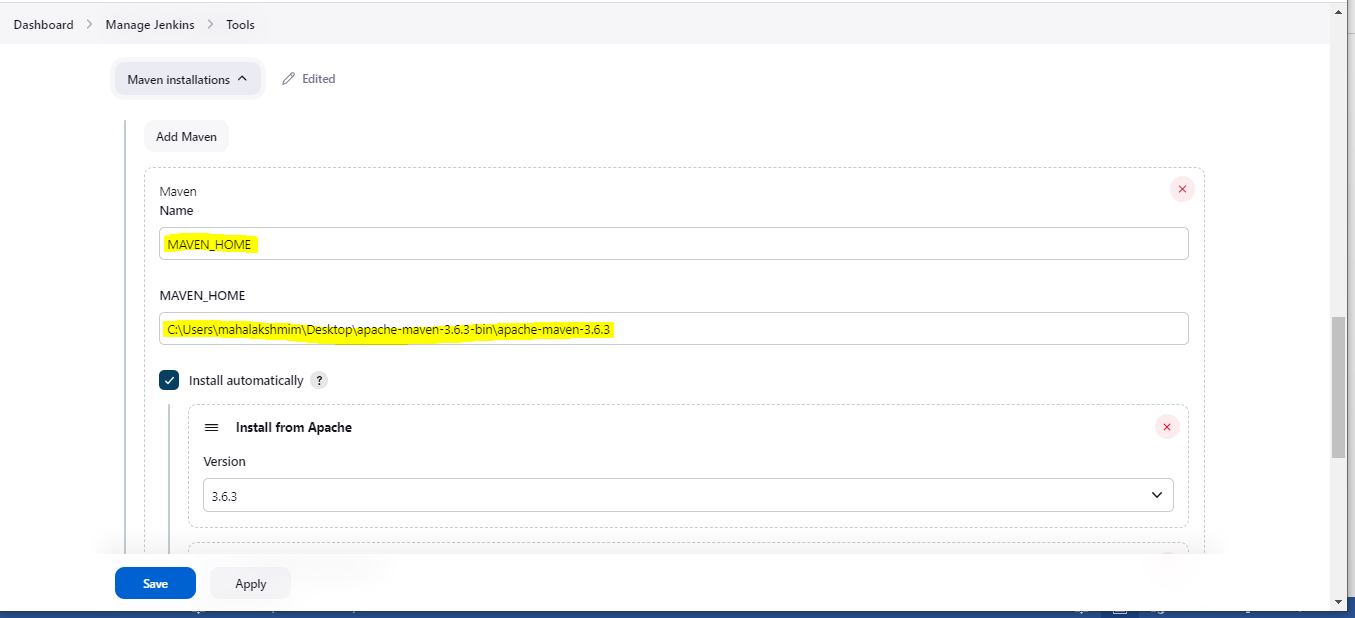
Choose Manage jenkins ->Tools



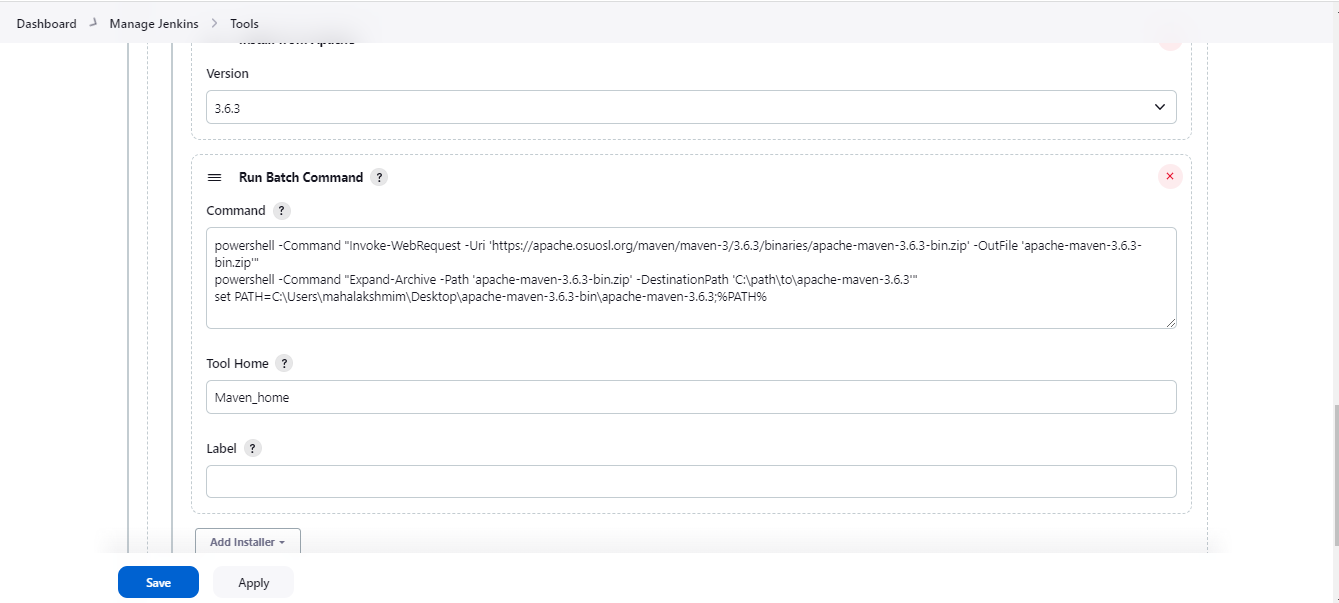
Git



Maven path for your local machine need to give here

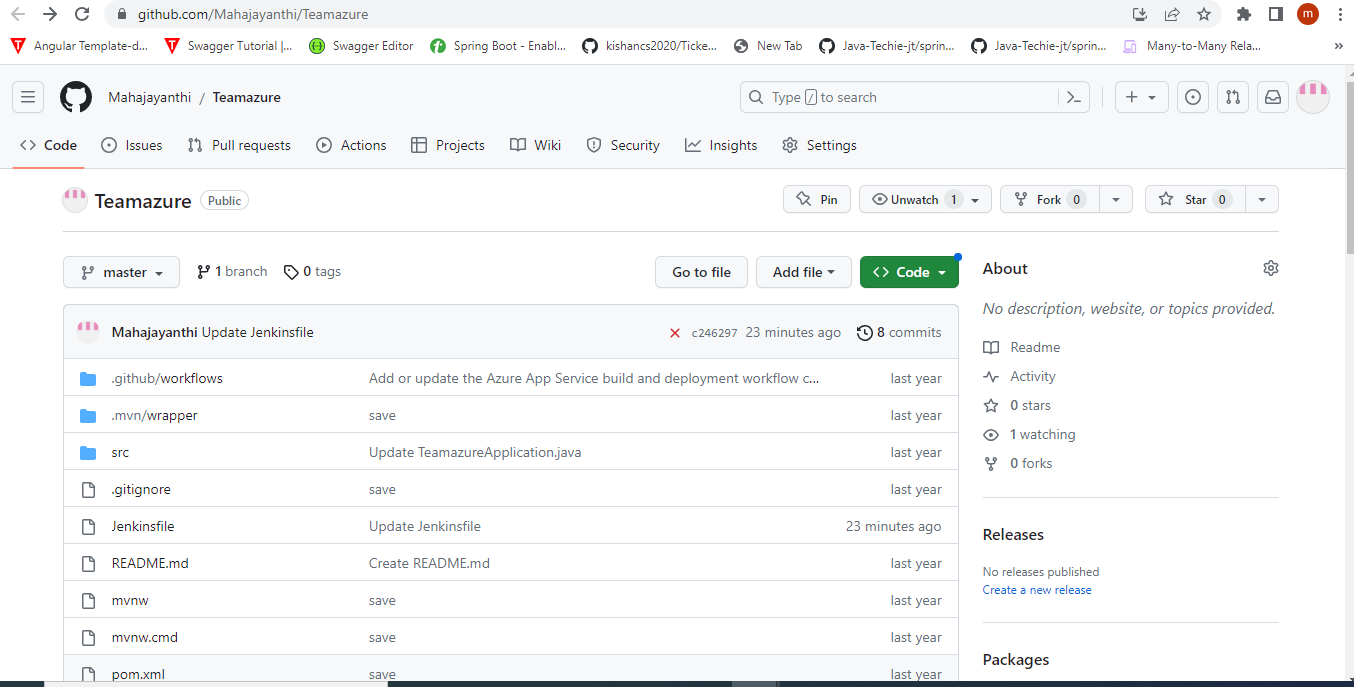


Normally for windows machine using batch commands

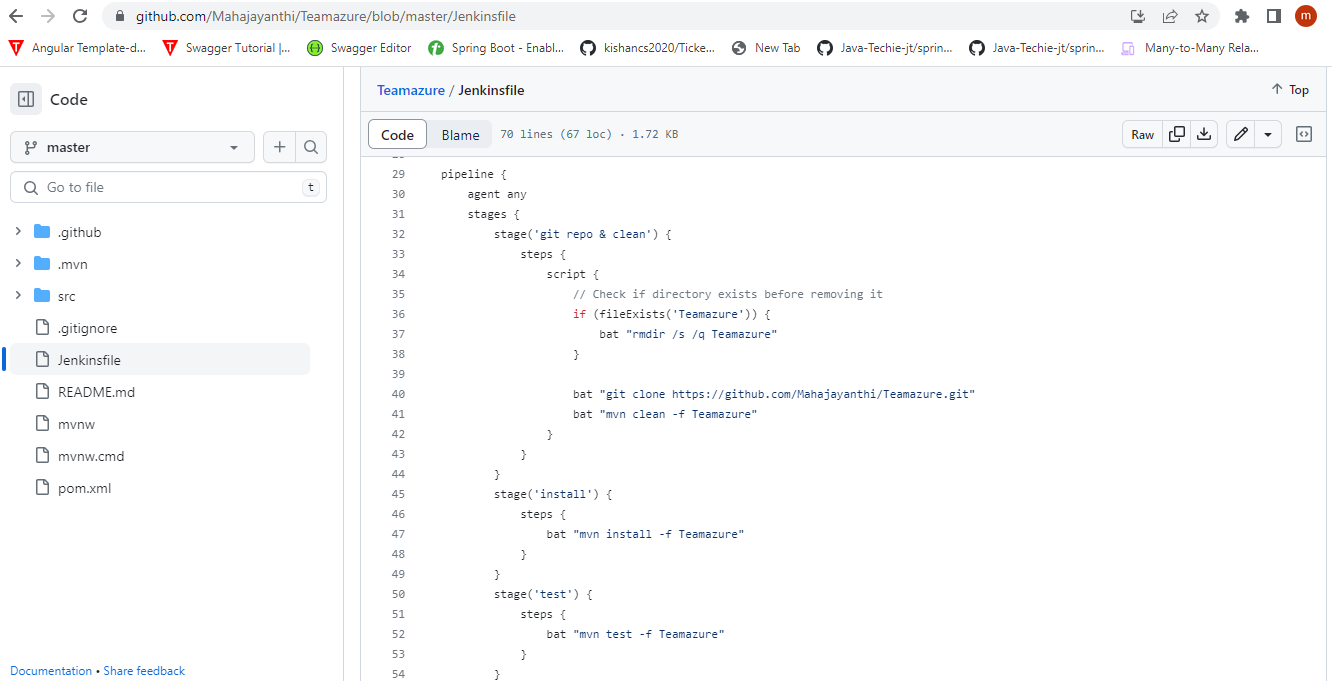


Then click on apply and save

**Step6:** Here I pushed one code from my local to my Github Repository



In my repository I have Jenkinsfile, this Jenkinsfile contain declarative pipeline stages



**Jenkinsfile:**

pipeline {

agent any

stages {

stage('git repo & clean') {

steps {

script {

// Check if directory exists before removing it

if (fileExists('Teamazure')) {

bat "rmdir /s /q Teamazure"

}

bat "git clone https://github.com/Mahajayanthi/Teamazure.git"

bat "mvn clean -f Teamazure"

}

}

}

stage('install') {

steps {

bat "mvn install -f Teamazure"

}

}

stage('test') {

steps {

bat "mvn test -f Teamazure"

}

}

stage('package') {

steps {

bat "mvn package -f Teamazure"

}

}

}

}

def fileExists(String path) {

try {

fileExists = new File(path).isDirectory()

} catch (Exception e) {

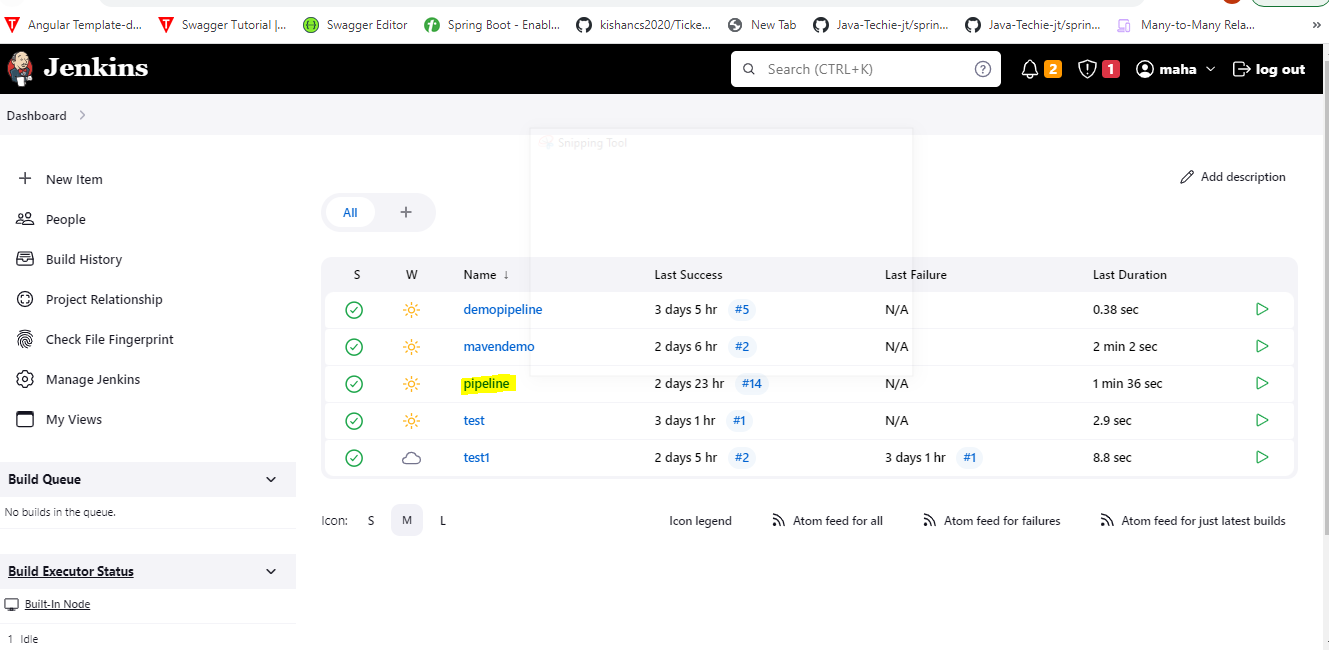
fileExists = false

}

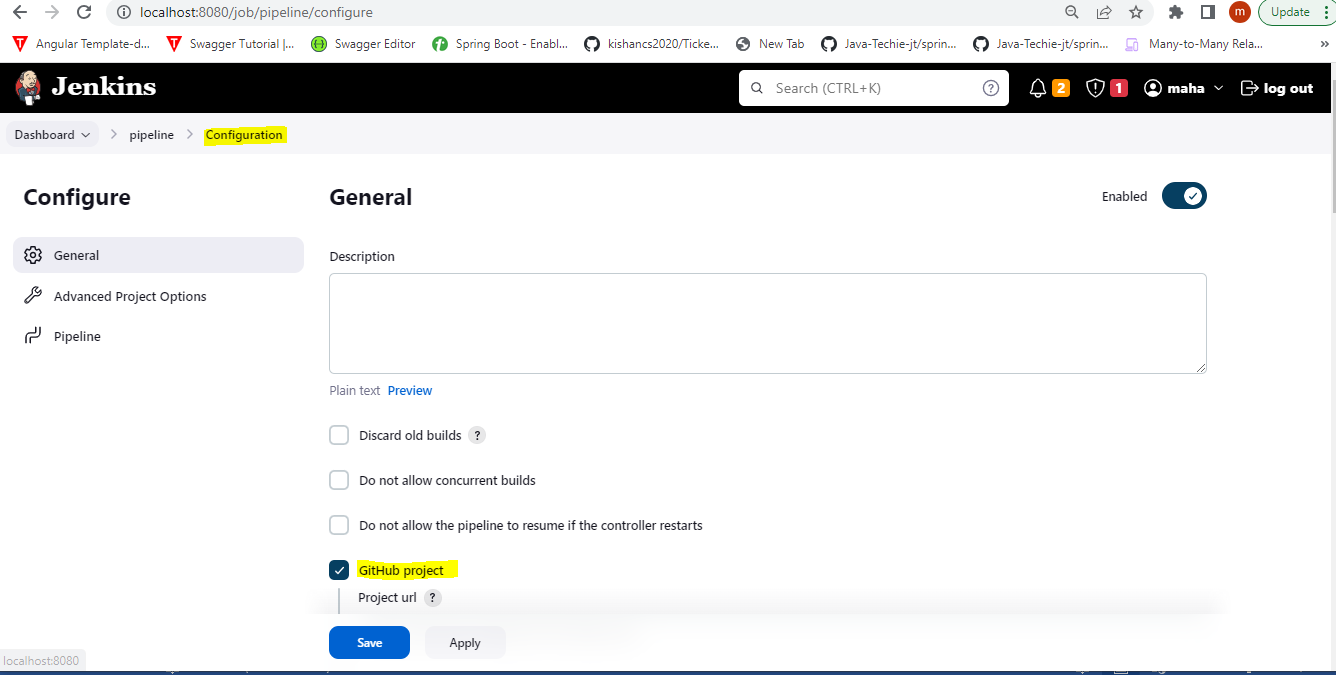
return fileExists

}

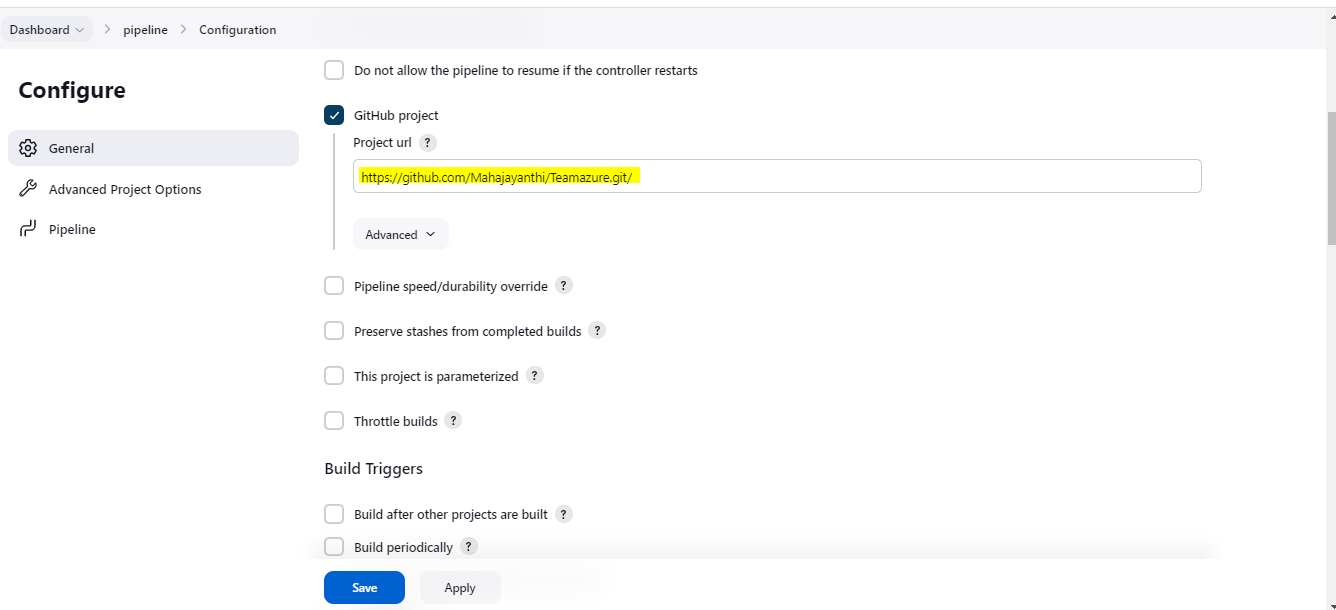
**Step7:** Here I have created one pipeline name as pipeline



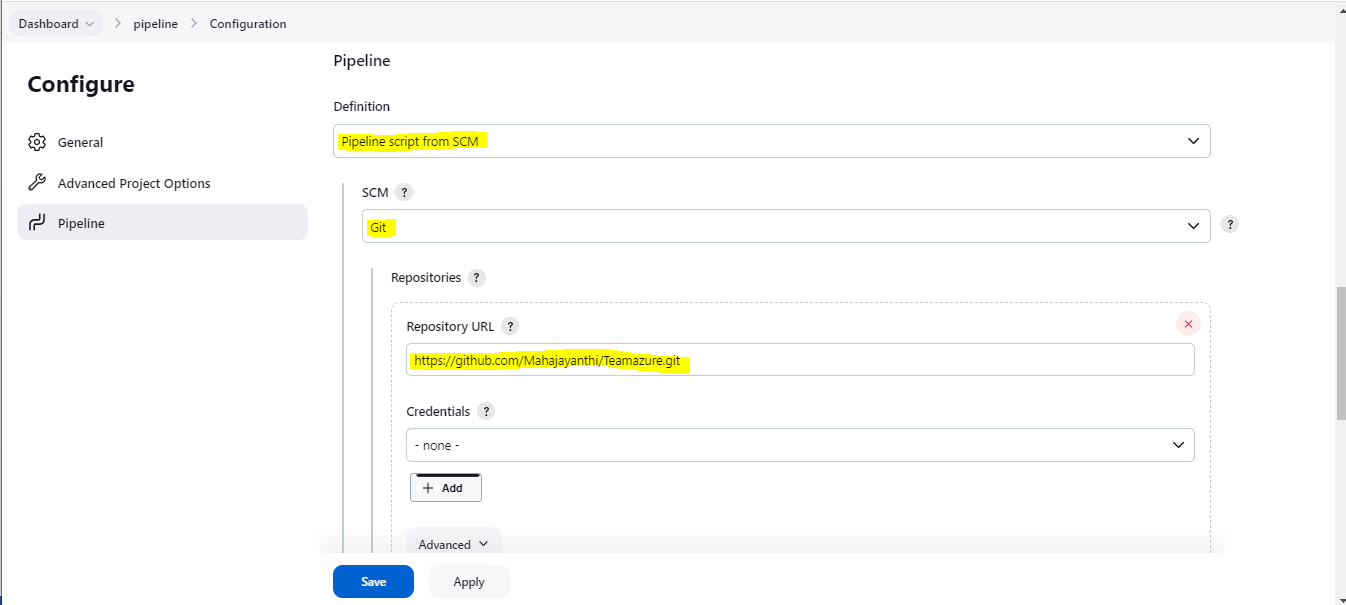
Under the configuration option I need to select the Github and need to give the Repository url



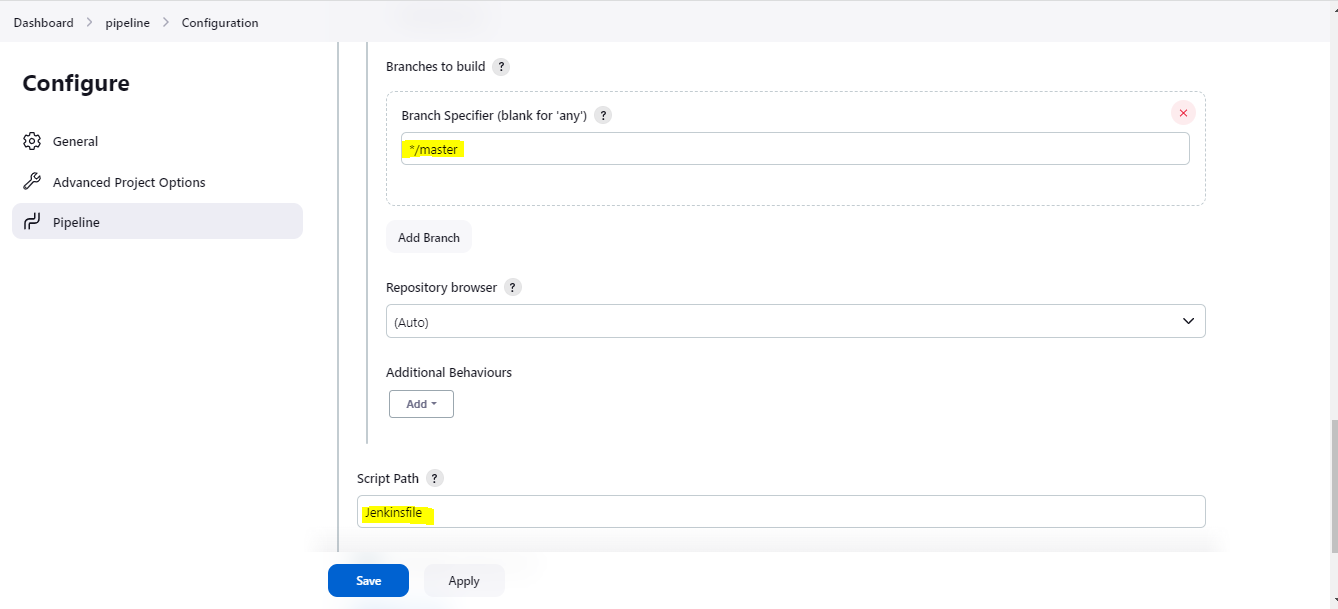
Under the Github url, need to give the Github url



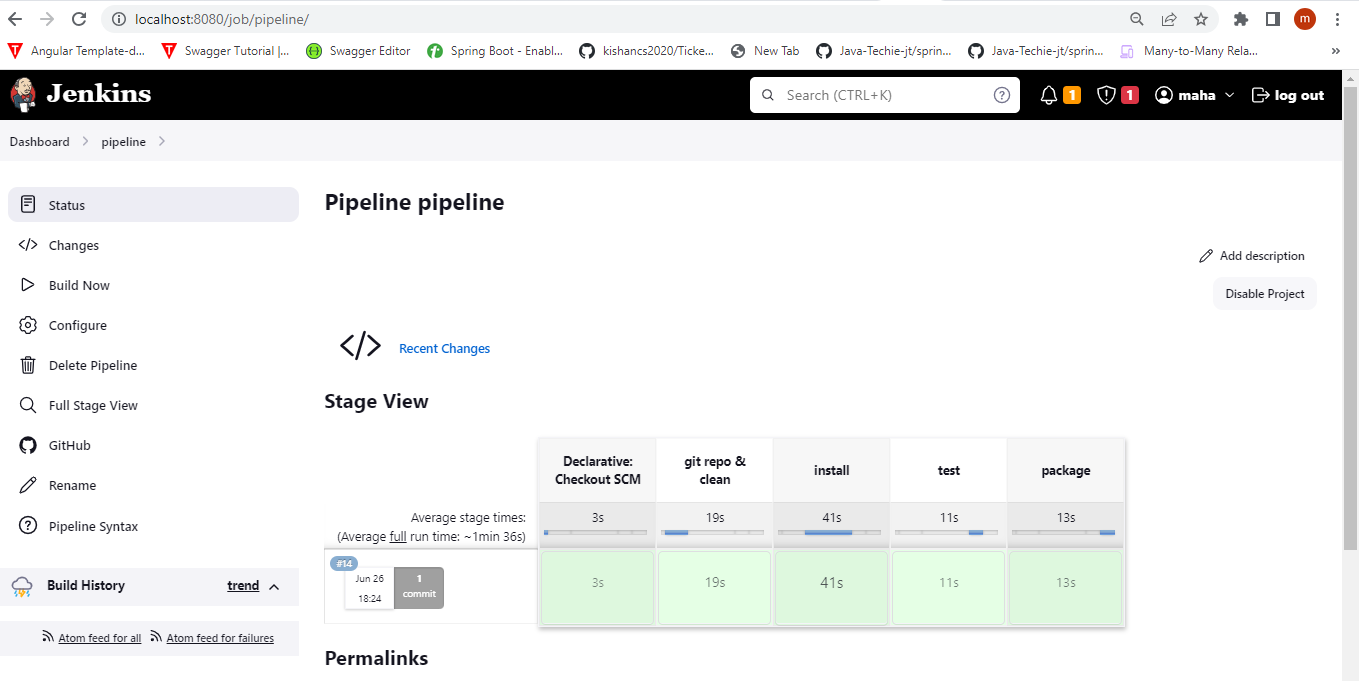
Here chosen the pipeline script from SCM ,SCM as Git, and give the Repository url



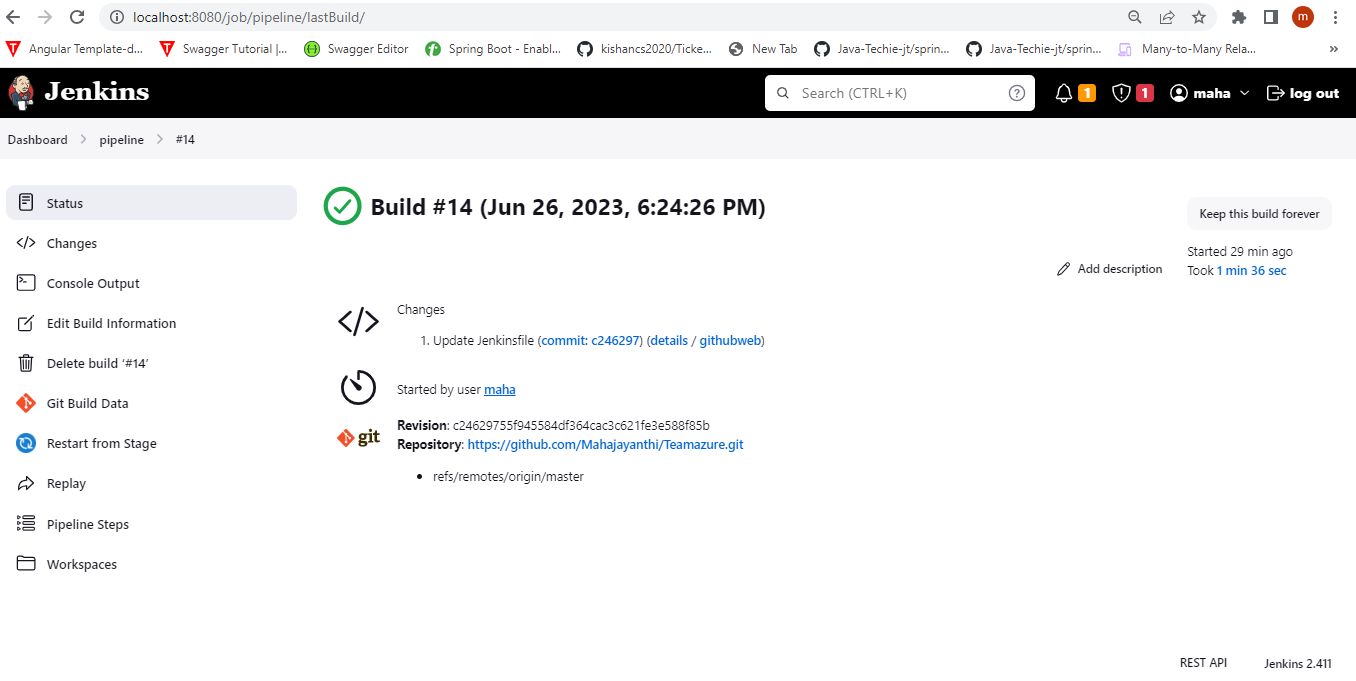
Then Click on Apply and save



The pipeline was running successfully



Under the stage by stage activity can able to see the logs



Through Console output can able to see the output

