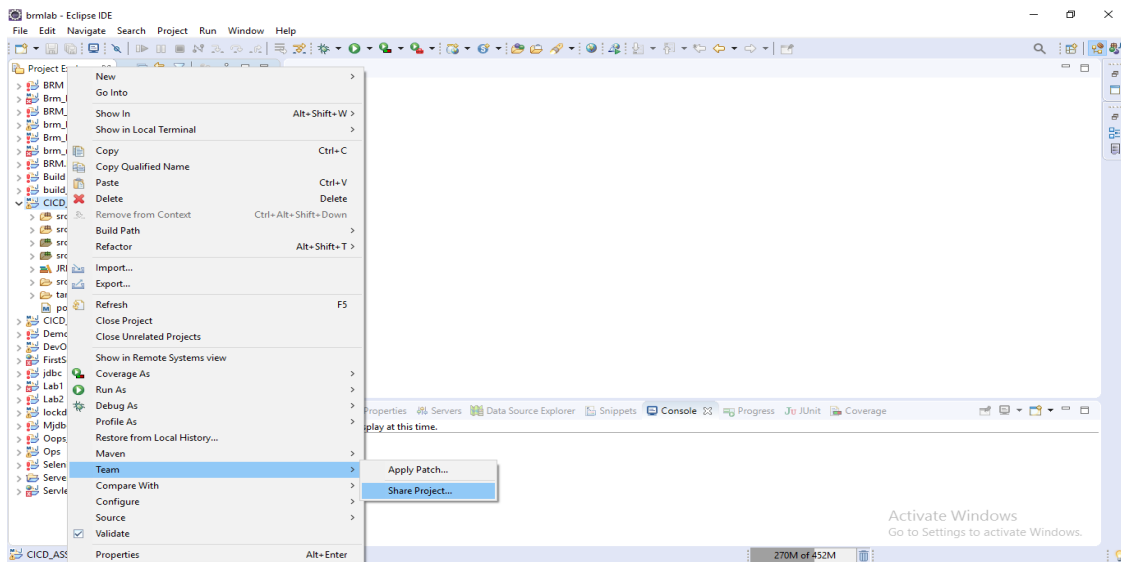


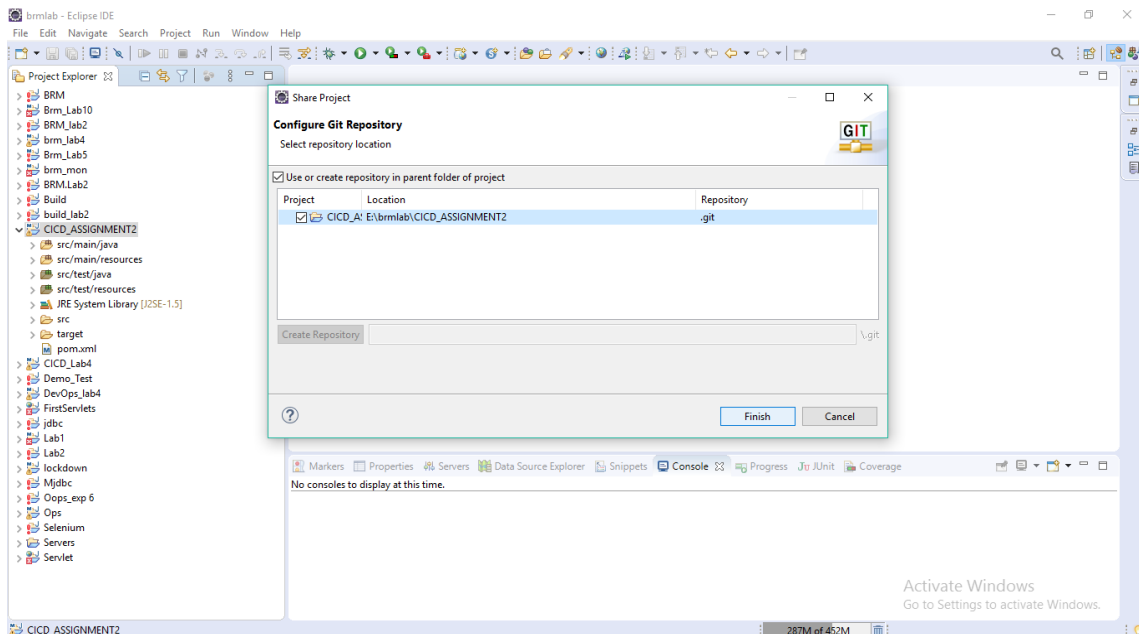
ASSIGNMENT-2 (Integration Of Github, Maven, Jenkins, Nexus)

Step-1 Creating one maven project and then pushing it into github from eclipse only.

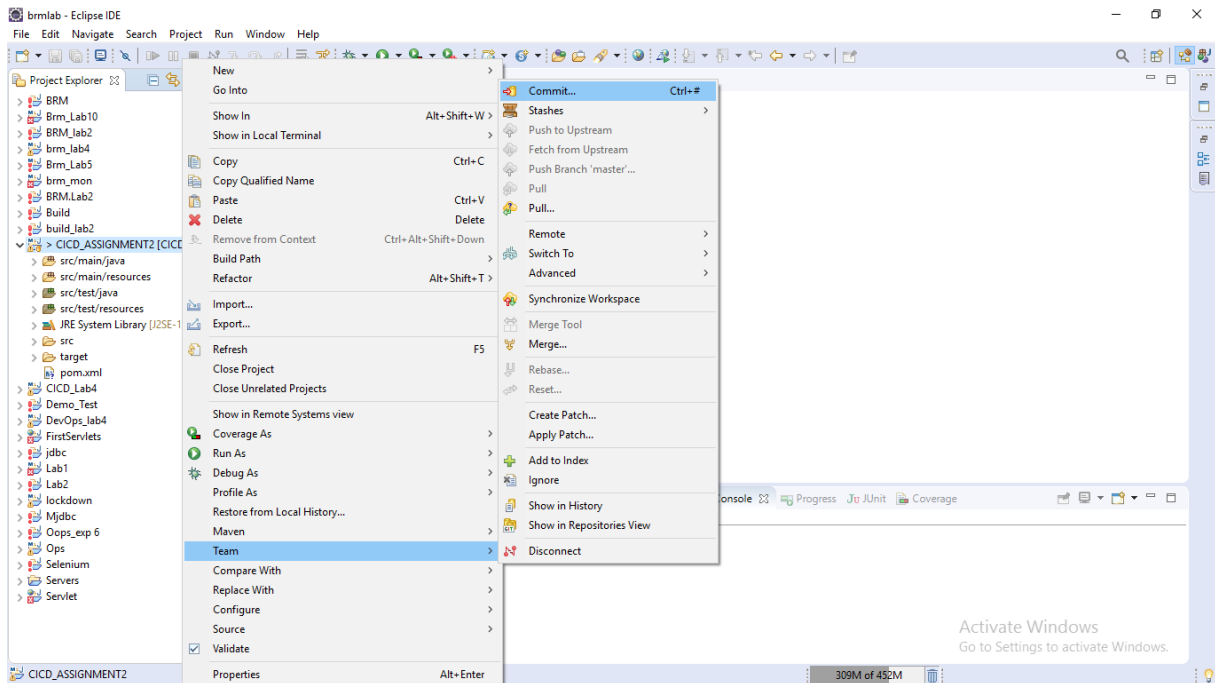
a) Right click on project -> Team -> Share Project



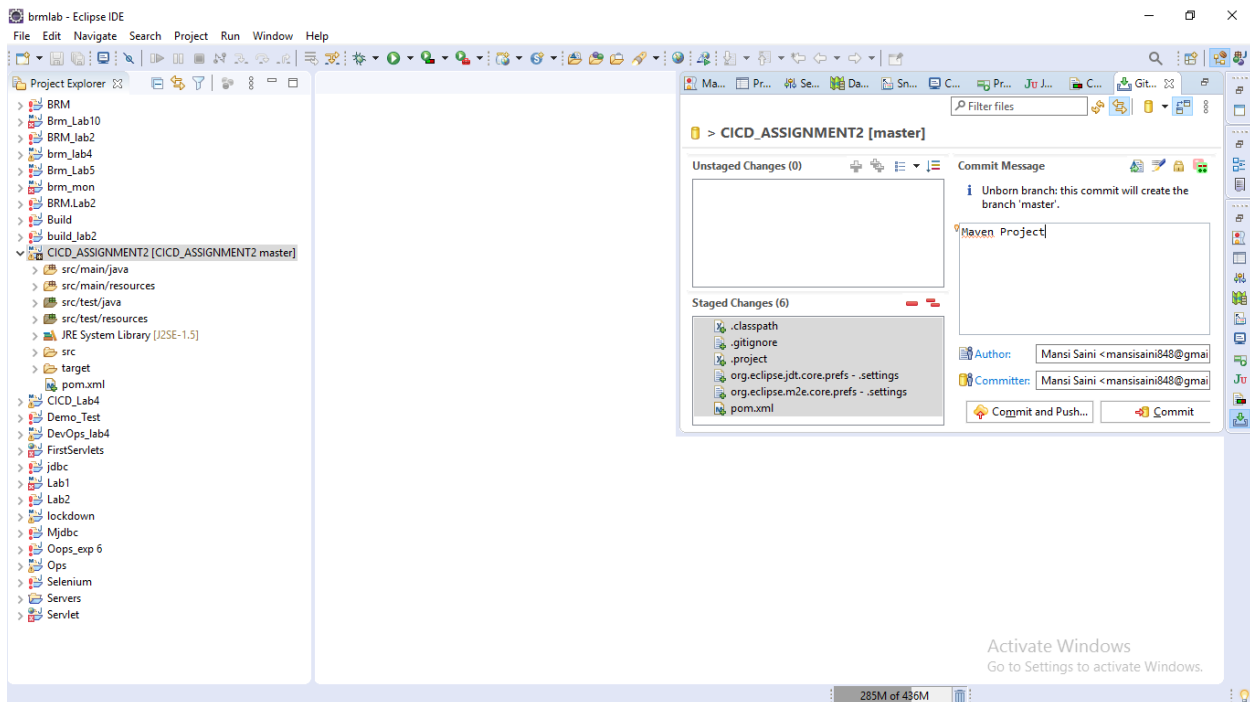
b) Select project and create git repository



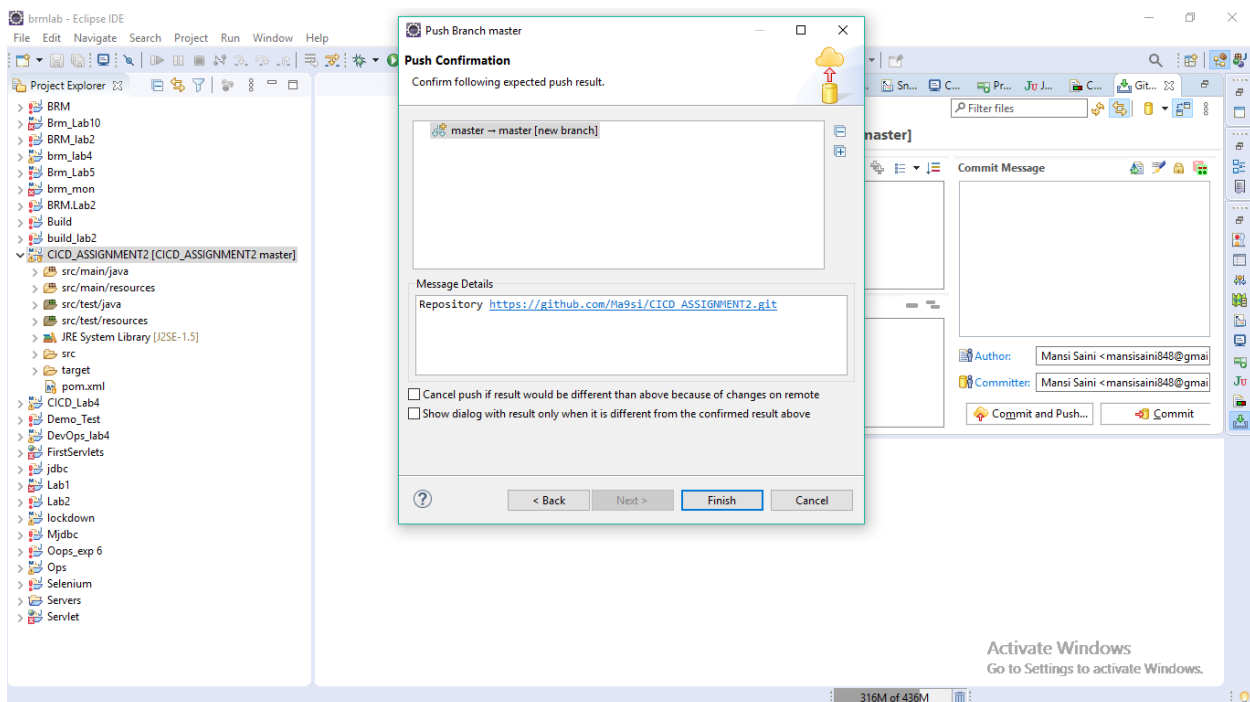
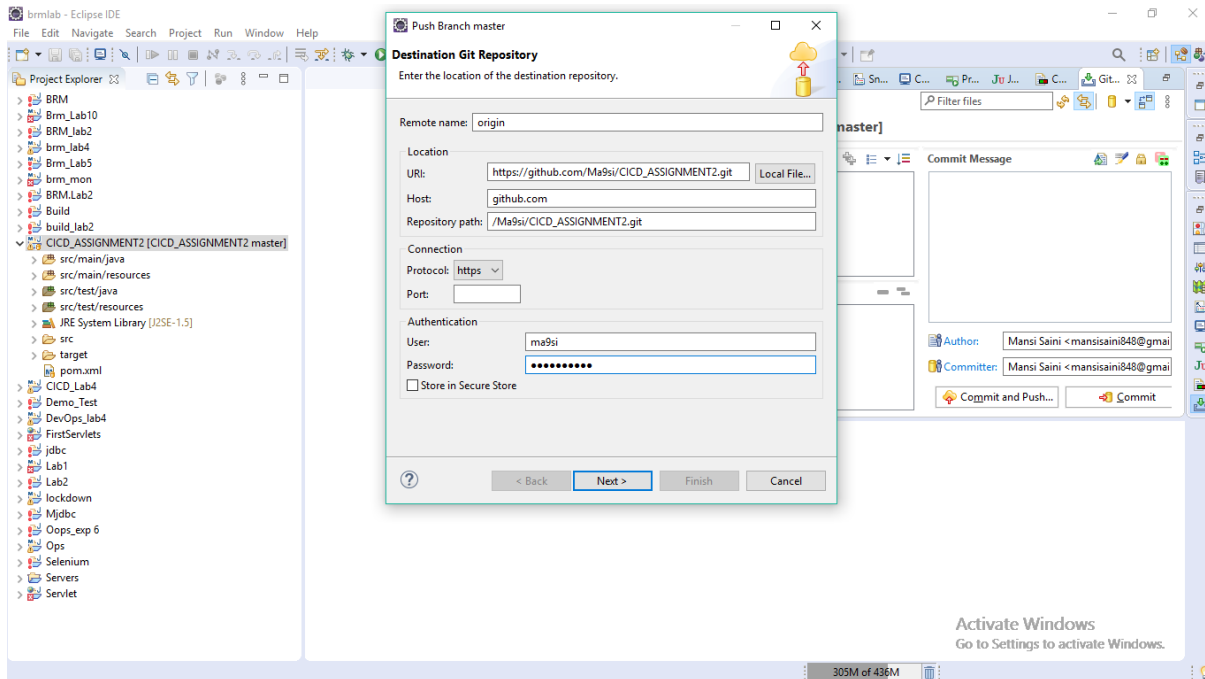
c) Right Click On Project -> Team -> Commit



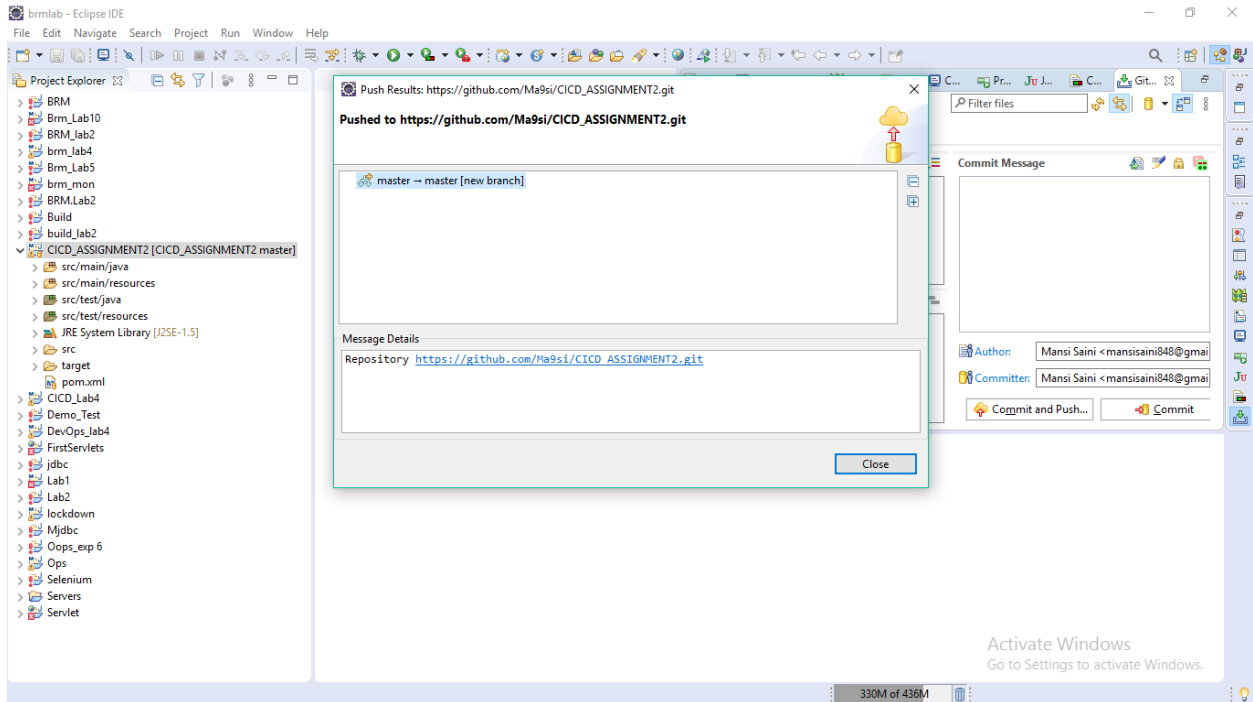
d) Stage all files and write commit message and then commit and push



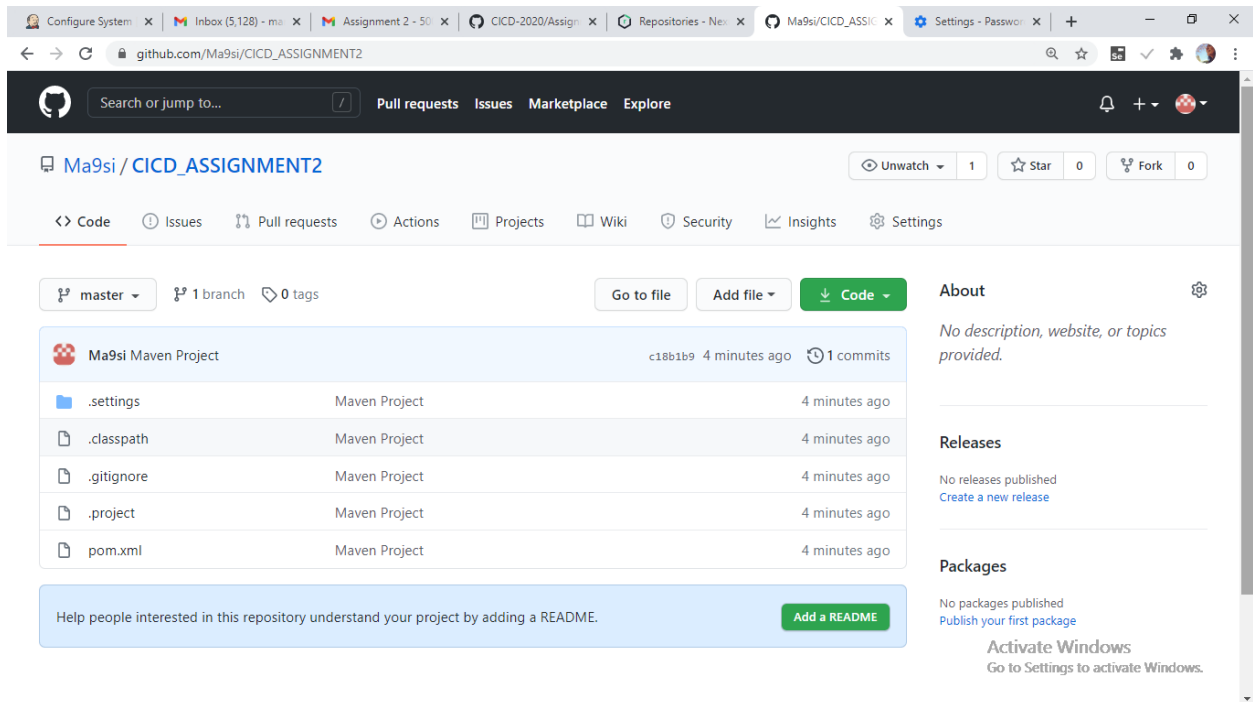
- e) Then, give url of your empty github repository here and github username and password for authentication.



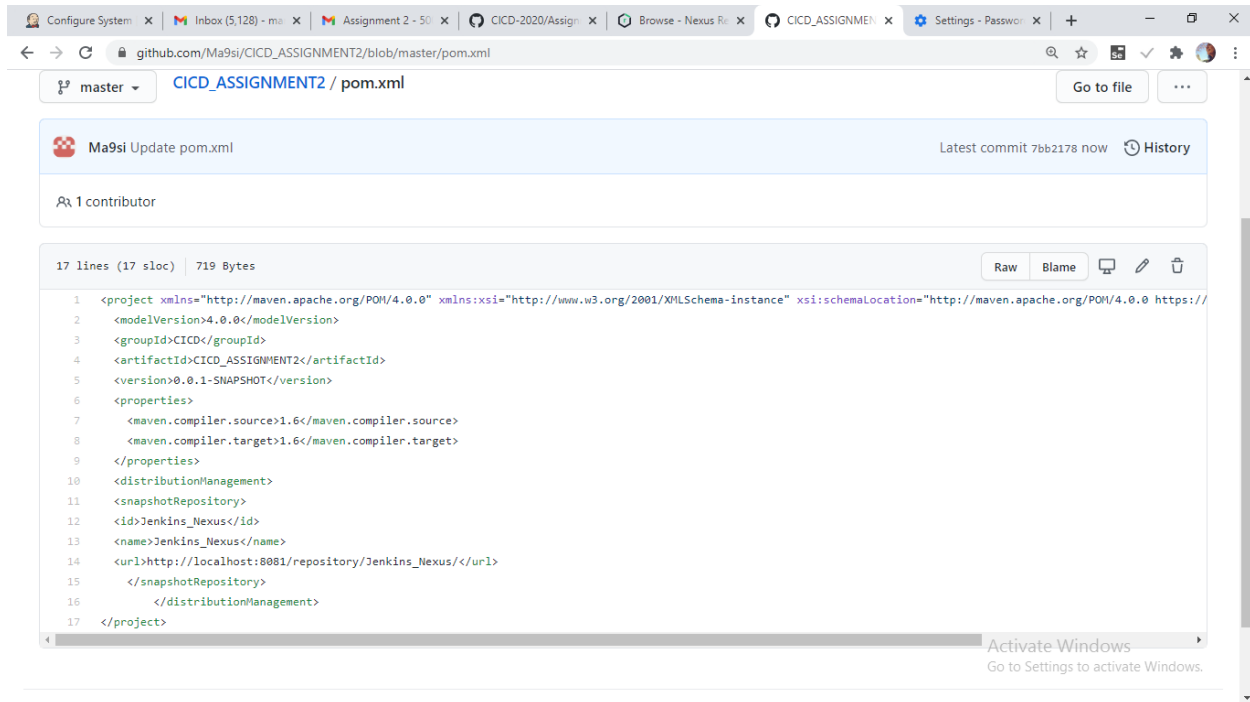
f) Now, our maven project is pushed



g) We can see from our project from github also .



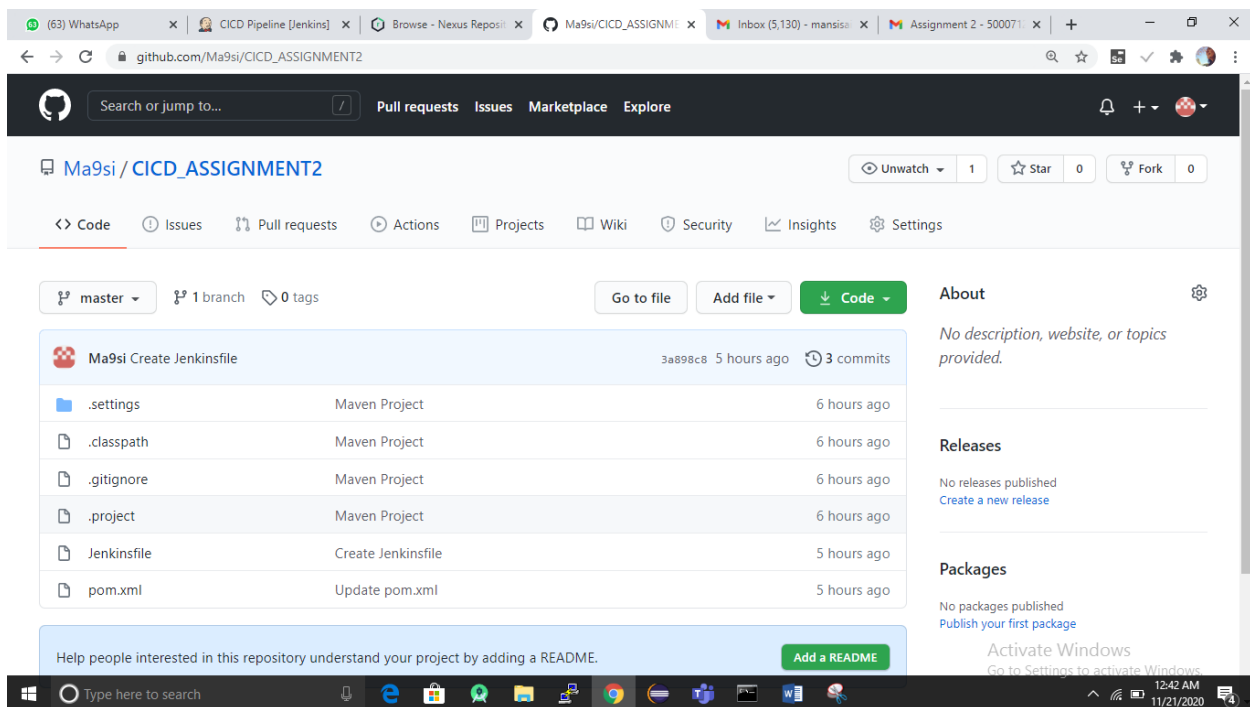
Step-2 Now, we will update pom.xml file and we will give the repo url of nexus where we want our project jar to be uploaded.



The screenshot shows a web browser displaying the GitHub repository page for 'Ma9si/CICD_ASSIGNMENT2'. The file 'pom.xml' is selected, showing its latest commit. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>CICD</groupId>
  <artifactId>CICD_ASSIGNMENT2</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <properties>
    <maven.compiler.source>1.6</maven.compiler.source>
    <maven.compiler.target>1.6</maven.compiler.target>
  </properties>
  <distributionManagement>
    <snapshotRepository>
      <id>Jenkins_Nexus</id>
      <name>Jenkins_Nexus</name>
      <url>http://localhost:8081/repository/Jenkins_Nexus/</url>
    </snapshotRepository>
  </distributionManagement>
</project>
```

Step-3 Now we will create jenkinsfile in our project repo in github so that our jar will automatically deployed to nexus server.



The screenshot shows the GitHub repository page for 'Ma9si/CICD_ASSIGNMENT2'. The file list shows the following files and their commit history:

File	Commit Message	Commit Time
.settings	Maven Project	6 hours ago
.classpath	Maven Project	6 hours ago
.gitignore	Maven Project	6 hours ago
.project	Maven Project	6 hours ago
Jenkinsfile	Create Jenkinsfile	5 hours ago
pom.xml	Update pom.xml	5 hours ago

The commit history for the 'Jenkinsfile' shows 3 commits, with the latest commit being '3a898c8' 5 hours ago.

```
Dashboard [Jenkins] x Welcome - Nexus Repository Ma x CICD_ASSIGNMENT2/Jenkinsfile x +
github.com/Ma9si/CICD_ASSIGNMENT2/blob/master/Jenkinsfile
29 lines (29 sloc) | 611 Bytes
Raw Blame
1 pipeline{
2   agent any
3   stages{
4     stage('Clean'){
5       steps{
6         echo "Clean Stage"
7         sh "mvn clean"
8       }
9     }
10    stage('Test'){
11      steps{
12        echo "Test Stage"
13        sh "mvn test"
14      }
15    }
16    stage('Package'){
17      steps{
18        echo "Package(making jar) Stage"
19        sh "mvn package"
20      }
21    }
22    stage('Deploy'){
23      steps{
24        echo "Deploy(to nexus server) Stage"
25        sh "mvn deploy"
26      }
27    }
28  }
29 }
```

Step-4 a) We will create one pipeline project in jenkins

New Item [Jenkins] x Welcome - Nexus Repository Ma x Ma9si/CICD_ASSIGNMENT2 x +


localhost:8080/view/all/new/job

Jenkins search ? 2 admin log out


Jenkins > All >

Enter an item name


» Required field

**Freestyle project**


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**


Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

**OK** at Team/Project

Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

b) We will give link of our project github repo here in new job of jenkins.

The screenshot shows the Jenkins configuration page for a job named 'CICD_ASSIGNMENT2'. The 'Advanced Project Options' tab is selected. The 'Pipeline' section is expanded, showing the following configuration:

- Definition:** Pipeline script from SCM
- SCM:** Git
- Repositories:**
 - Repository URL:** `https://github.com/Ma9si/CICD_ASSIGNMENT2`
 - Credentials:** - none - (with an 'Add' button)
 - Buttons:** Advanced..., Add Repository
- Branches to build:**
 - Branch Specifier (blank for 'any'):** `*/master` (with a red 'X' icon and a help icon)
 - Button:** Add Branch

At the bottom left, there are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins configuration page for the same job, but with the 'Pipeline' tab selected. The configuration is as follows:

- Buttons:** Add Repository
- Branches to build:**
 - Branch Specifier (blank for 'any'):** `*/master` (with a red 'X' icon and a help icon)
 - Button:** Add Branch
- Repository browser:** (Auto)
- Additional Behaviours:** Add
- Script Path:** Jenkinsfile
- Lightweight checkout:** ☒
- Link:** Pipeline Syntax

At the bottom left, there are 'Save' and 'Apply' buttons.

c) Our job has successfully completed

The screenshot shows the Jenkins web interface for job **CICD_ASSIGNMENT2**, build **#1**. The left sidebar contains navigation links: Back to Project, Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#1', Git Build Data, No Tags, Open Blue Ocean, Restart from Stage, Replay, and Pipeline Steps. The main area displays the **Console Output** for build #1, started by user **admin**. The output shows the pipeline starting, cloning the repository from https://github.com/Ma9si/CICD_ASSIGNMENT2.git, and executing a series of git commands to initialize, fetch, and checkout the code. The output ends with the message "git.exe rev-parse 'refs/remotes/origin/master^{commit}' # timeout=10".

```
Started by user admin
Obtained Jenkinsfile from git https://github.com/Ma9si/CICD_ASSIGNMENT2.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\Users\HP\.jenkins\workspace\CICD_ASSIGNMENT2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Ma9si/CICD_ASSIGNMENT2.git
> git.exe init C:\Users\HP\.jenkins\workspace\CICD_ASSIGNMENT2 # timeout=10
Fetching upstream changes from https://github.com/Ma9si/CICD_ASSIGNMENT2.git
> git.exe --version # timeout=10
> git --version # 'git version 2.27.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/Ma9si/CICD_ASSIGNMENT2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/Ma9si/CICD_ASSIGNMENT2.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/Ma9si/CICD_ASSIGNMENT2.git # timeout=10
Fetching upstream changes from https://github.com/Ma9si/CICD_ASSIGNMENT2.git
> git.exe fetch --tags --force --progress -- https://github.com/Ma9si/CICD_ASSIGNMENT2.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse 'refs/remotes/origin/master^{commit}' # timeout=10
> git.exe rev-parse 'refs/remotes/origin/master^{commit}' # timeout=10
```

The screenshot shows the Jenkins web interface for job **CICD_ASSIGNMENT2**. The left sidebar contains navigation links: Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Open Blue Ocean, Rename, Pipeline Syntax, Build History (selected), and trend. The main area displays the **Stage View** for build #1. It shows a table of stage times and a list of recent changes. The stage times are: Declarative: Checkout SCM (6s), Clean (5s), Test (6s), Package (4s), and Deploy (7s). The recent changes list shows a change on Nov 20 at 21:21 with no changes. Below the stage view, there are **Permalinks** for the last build, last stable build, last successful build, and last completed build, all of which are 1 min 44 sec ago.

Stage	Time
Declarative: Checkout SCM	6s
Clean	5s
Test	6s
Package	4s
Deploy	7s

Average stage times:
(Average full run time: ~45s)

Recent Changes

#	Time	Changes
#1	Nov 20 21:21	No Changes

Permalinks

- Last build (#1), 1 min 44 sec ago
- Last stable build (#1), 1 min 44 sec ago
- Last successful build (#1), 1 min 44 sec ago
- Last completed build (#1), 1 min 44 sec ago

Step-5 Our jar has successfully deployed to nexus server.

