**Jenkins GitHub Integration:**

**Basic overview of Jenkins GitHub Integration using diagram:**

**Git hub Repository**

**Eclipse**

**Git Commands**

Jenkins will take the code from GitHub Repo and Run the code.

Jenkins can trigger the code using **Testng.xml** or **POM.xml**

**Jenkins**

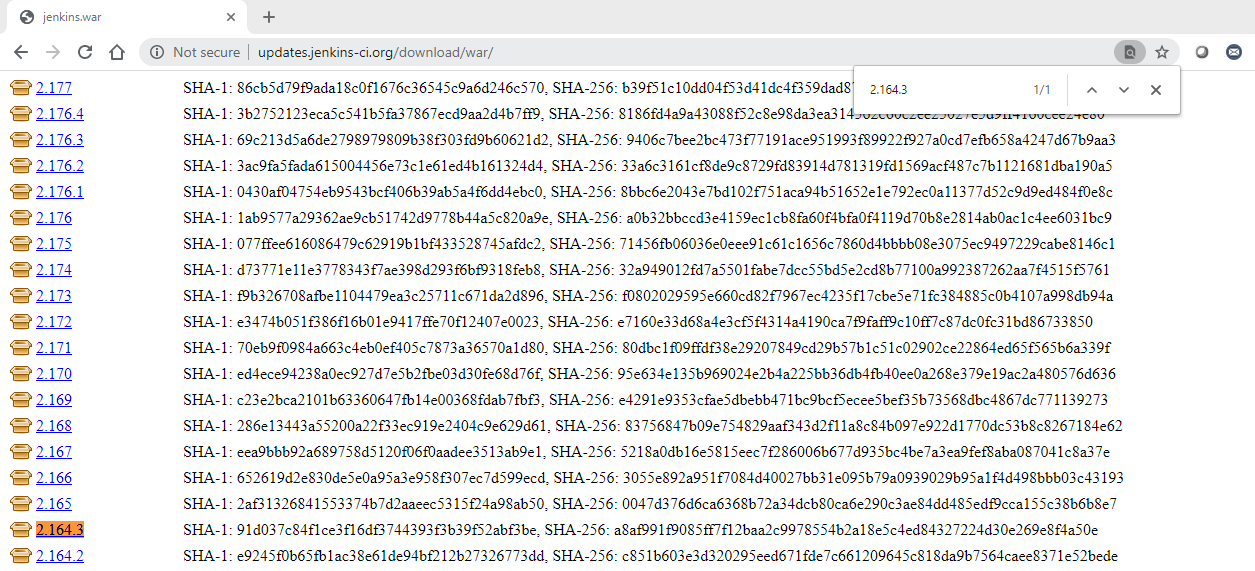
**Follow below steps to trigger the code using Jenkins:**

**Step-1:** Firstly, we need Jenkins WAR file with version **(2.164.3)**

**Download it from the below link:**

<https://updates.jenkins-ci.org/download/war/>

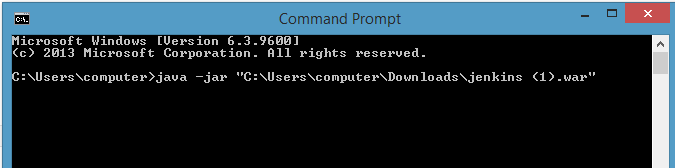
And [**Search**] the version- **2.164.3**



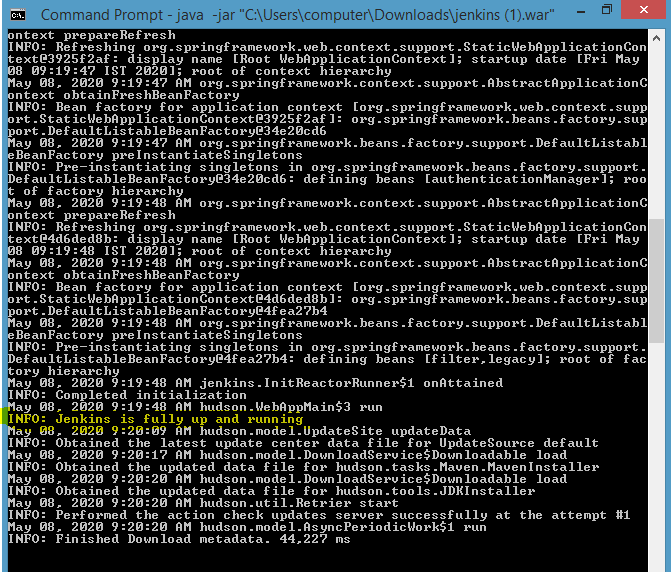
And now click on the above highlighted hyperlink (2.164.3): Jenkins WAR file will be downloaded

**Step-2:** After downloading the **Jenkins WAR File**, we need to start the Jenkins server from **cmd** using below command:

**Java –jar Jenkins War File**

****

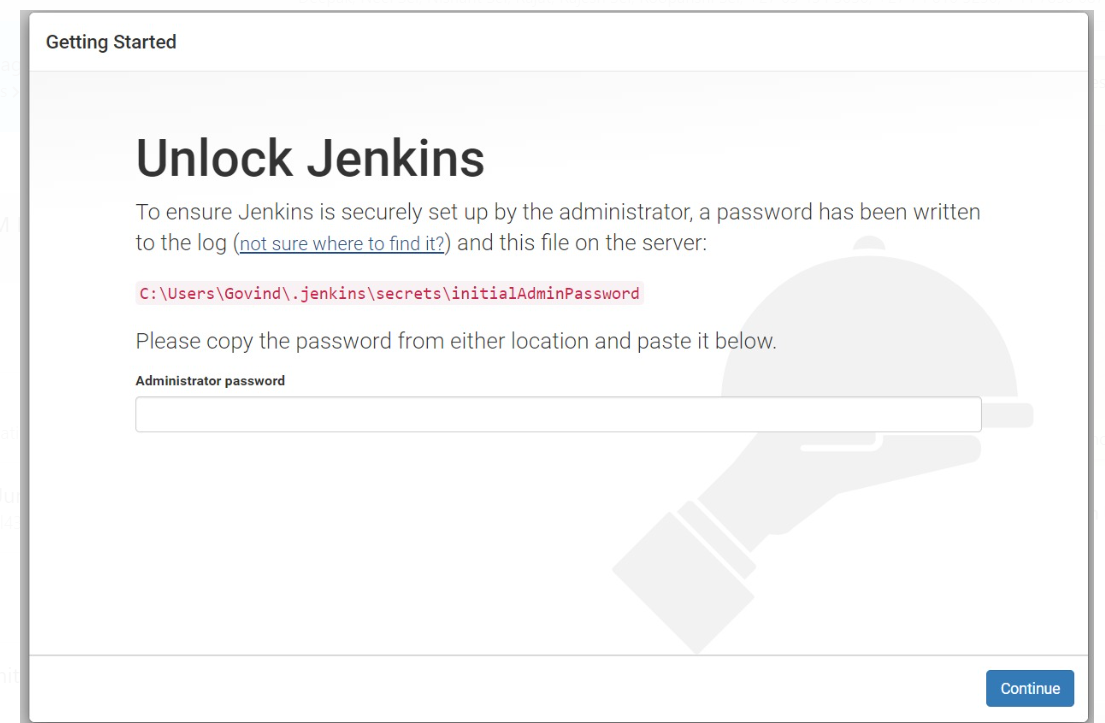
Press **[Enter]**

****

**Now Jenkins is fully started.**

**Step-3:** Now open Jenkins using the below URL:

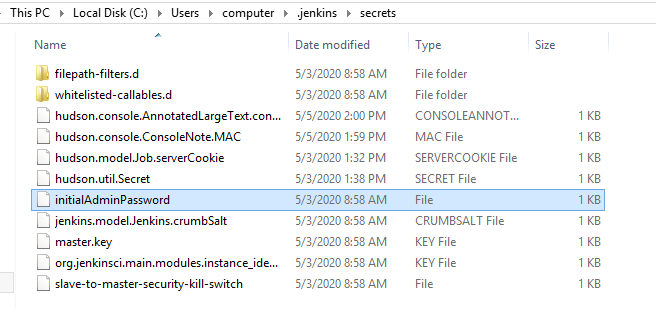
<http://localhost:8080/>



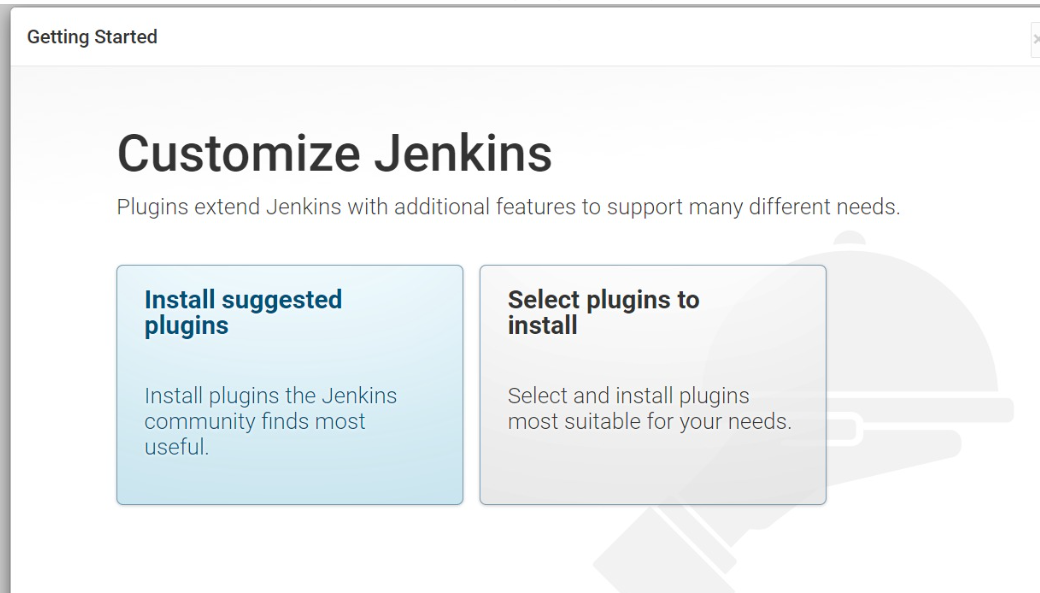
If asked Enter **[User Name]** as “Admin”

For **Password**, we need to go the below path:

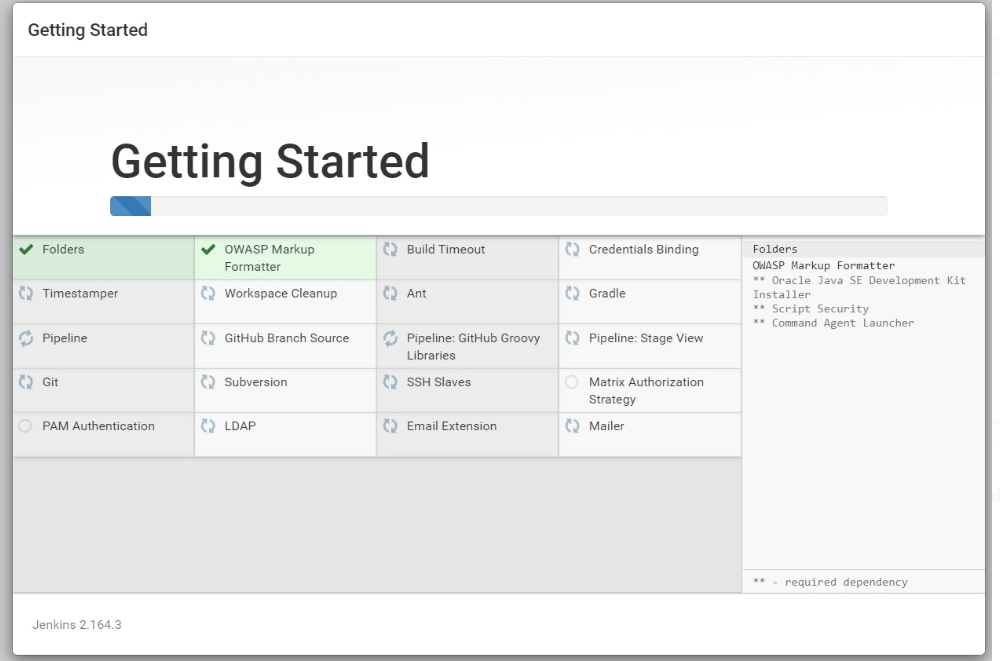
**C:\Users\computer\.jenkins\secrets** and open the file **“initialAdminPassword”**

****

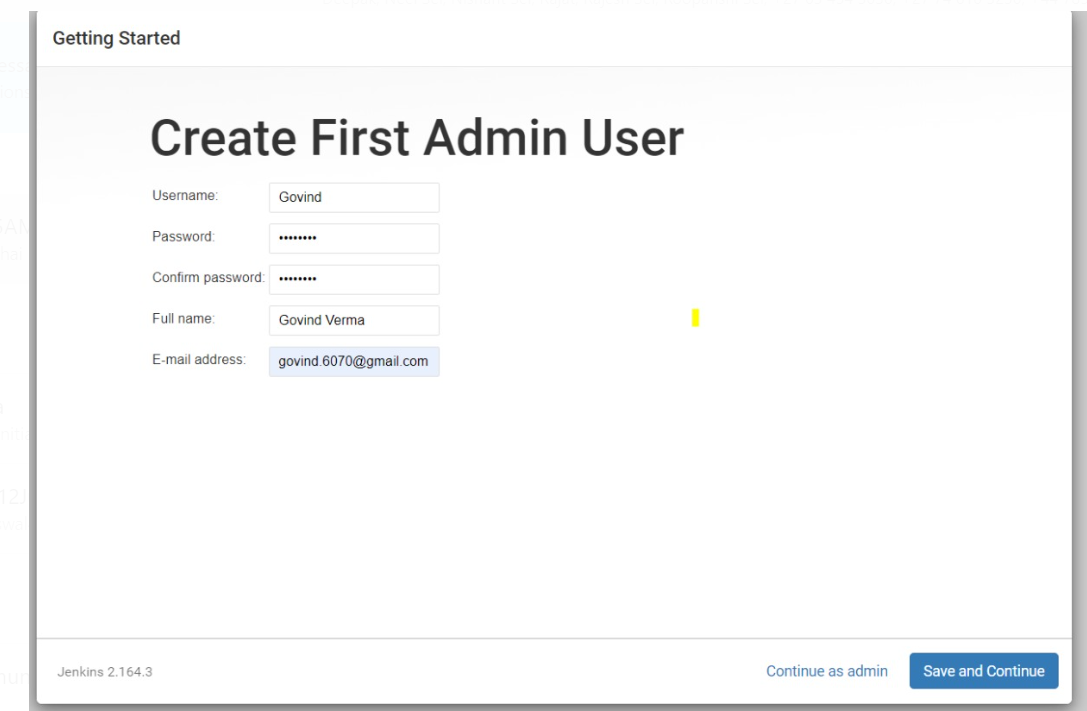
After providing the Admin password, it would ask to install plugins -



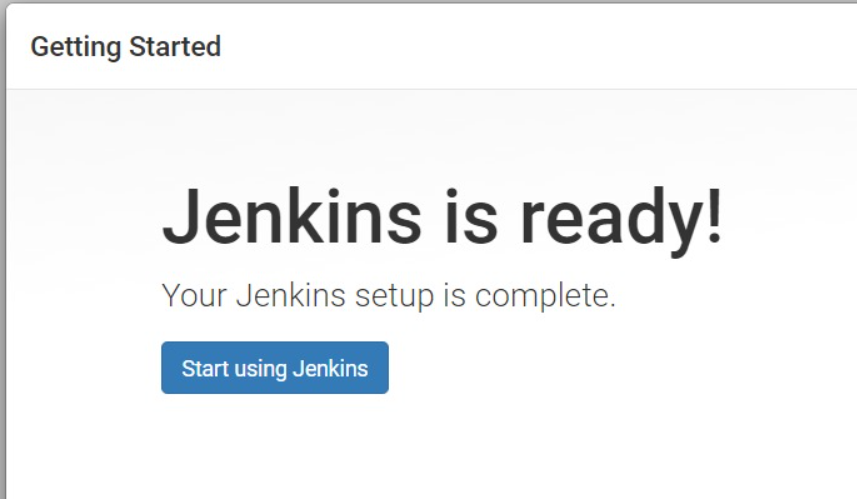
Once you choose to install suggested plugins, it would install following plugins one by one –



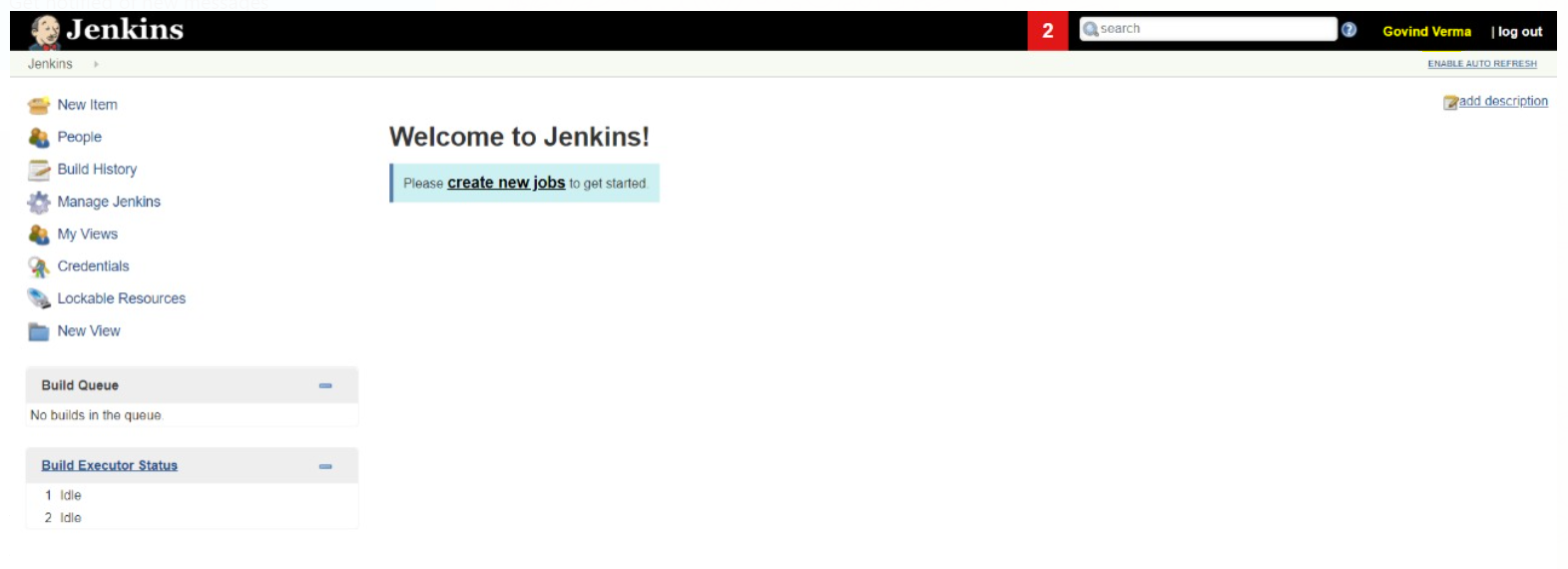
On the Next Screen you can create a new User Id and Password



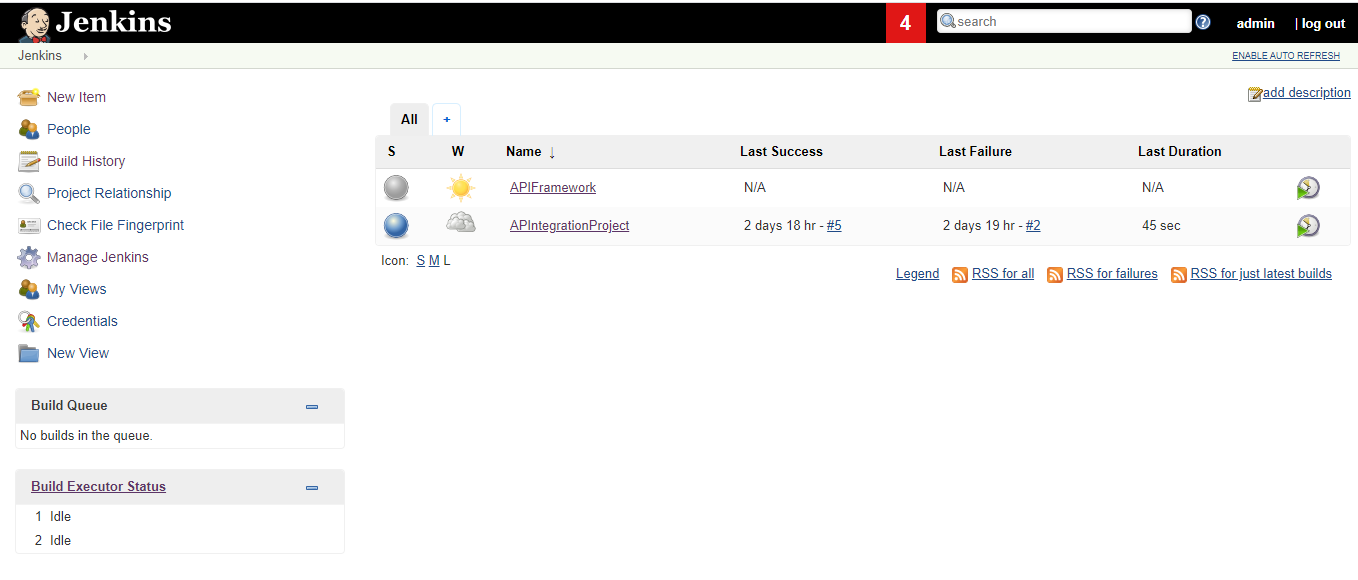
Once New User Id and Password details are provided below is the next screen –



If you have created new user Id and Password then you can login with newly created User Id and Password. Home screen will look like this –

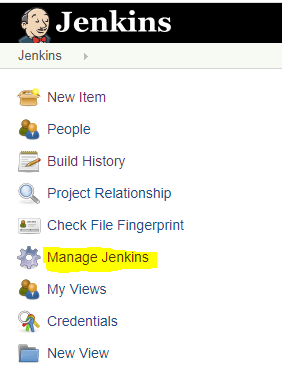


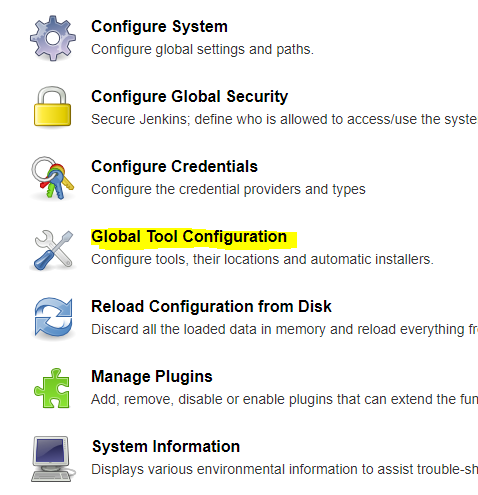
If you don’t want to create new user Id and password then you can login with admin user id and password -

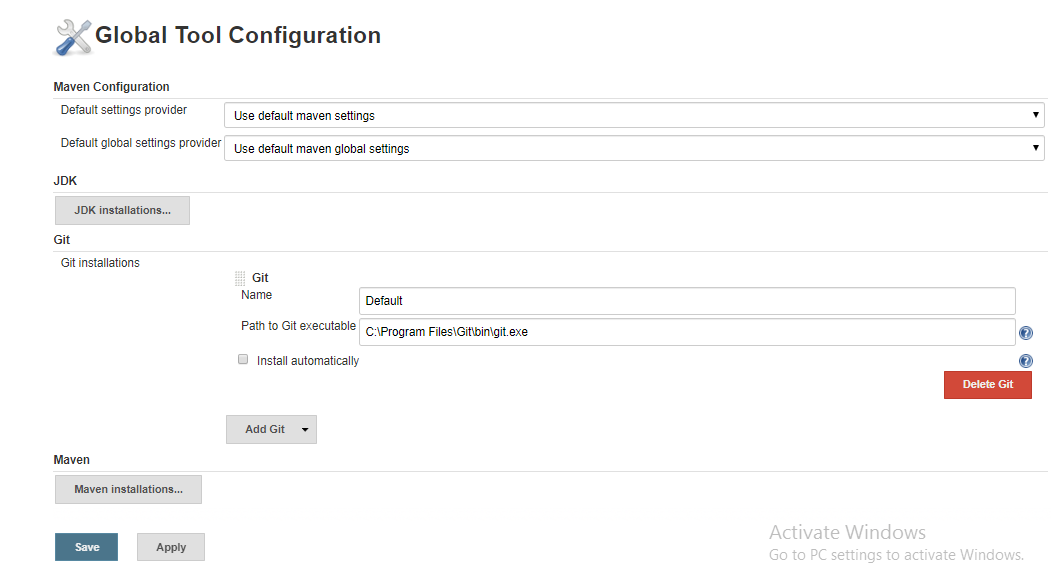
****

**Step-4:** Now, we have to configure following things in Jenkins:

Navigate to **[Manage Jenkins]** -> **[Global Tool Configuration]**

****

****

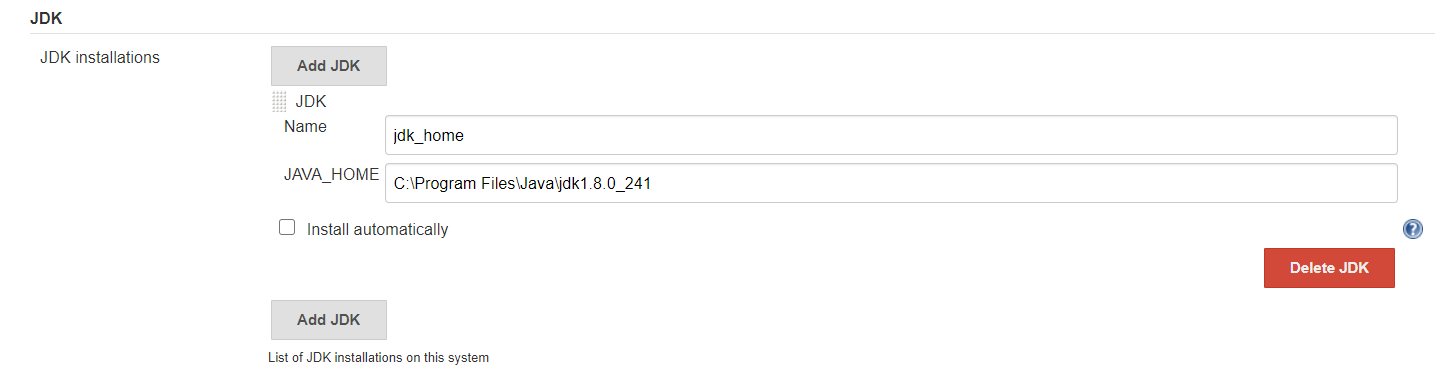
****

**And specify path for below things:**

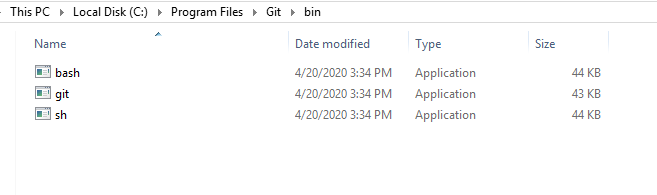
**1. JDK path : C:\Program Files\Java\jdk1.8.0\_221**



Specify JDK path as shown below –



**2. Git path: C:\Program Files\Git\bin\git.exe**

****

**Note:** Pleaseinstall GIT in your machine using the below link:

<https://git-scm.com/download/win>

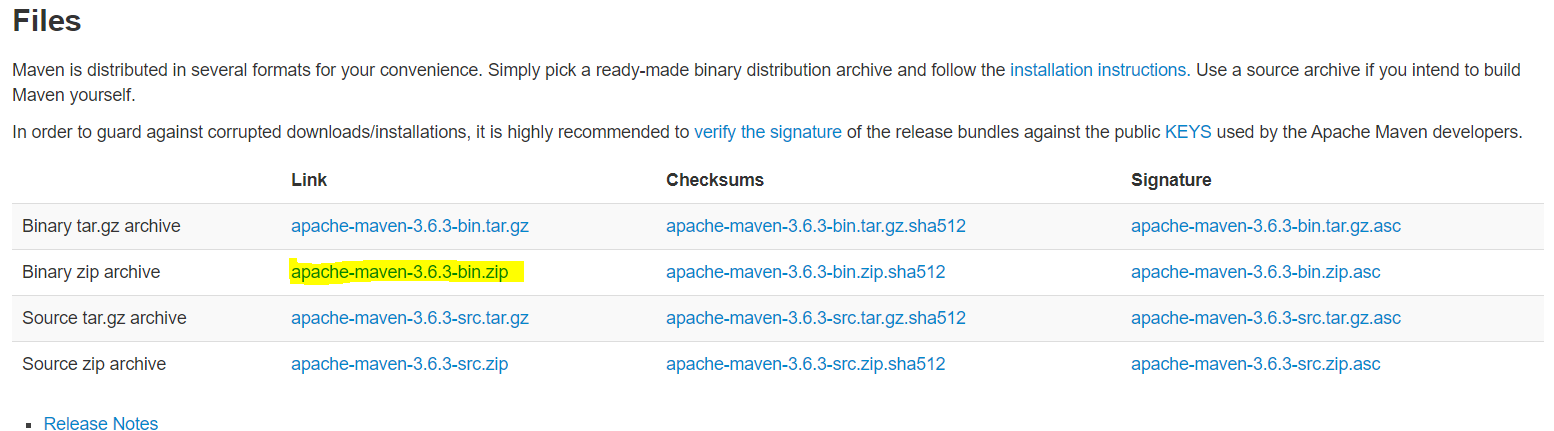
Specify Git path as shown below –



**3. Maven path:** Give the location of Maven (where Maven is installed)

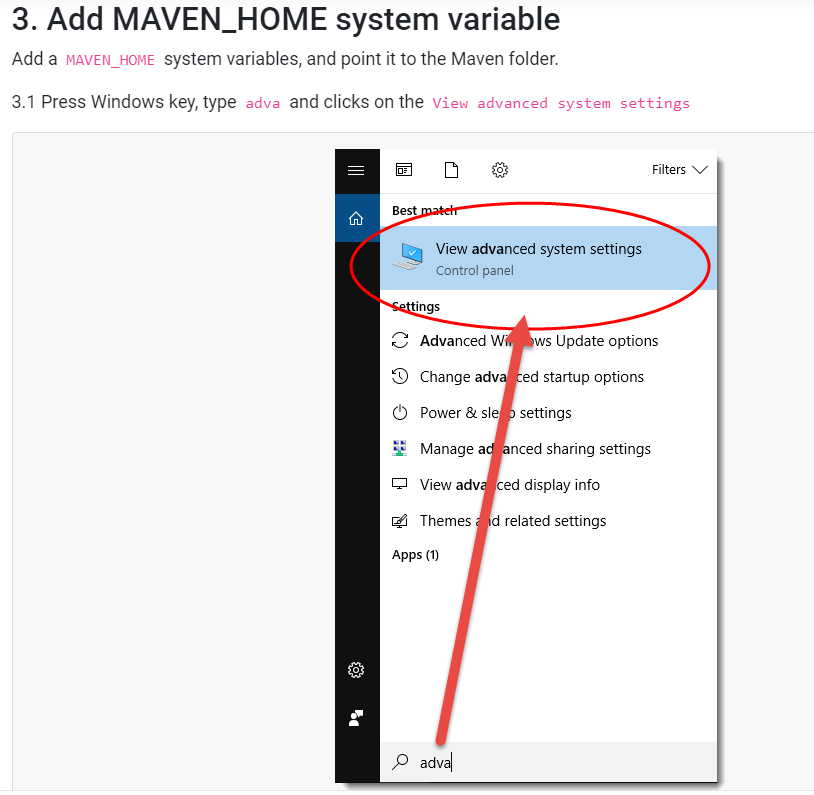
**Download Maven from using below link:**

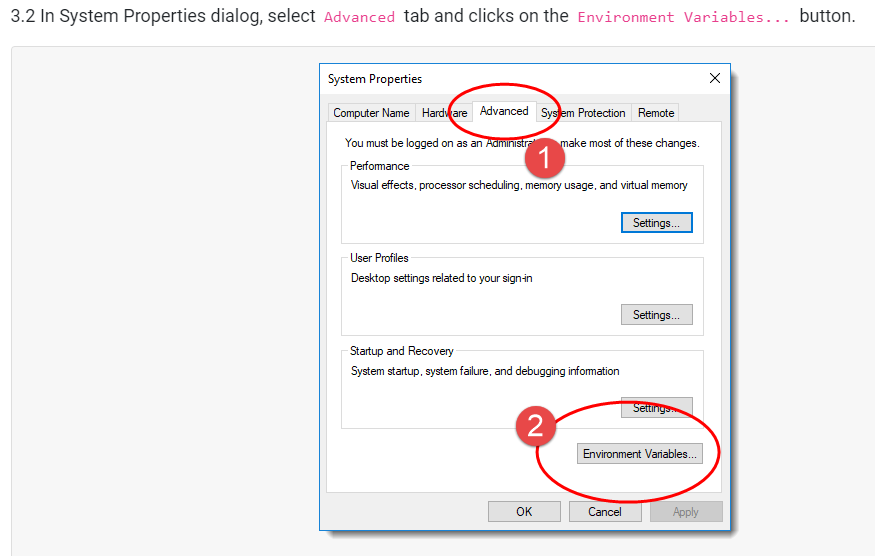
<https://maven.apache.org/download.cgi?Preferred=ftp://ftp.osuosl.org/pub/apache/>

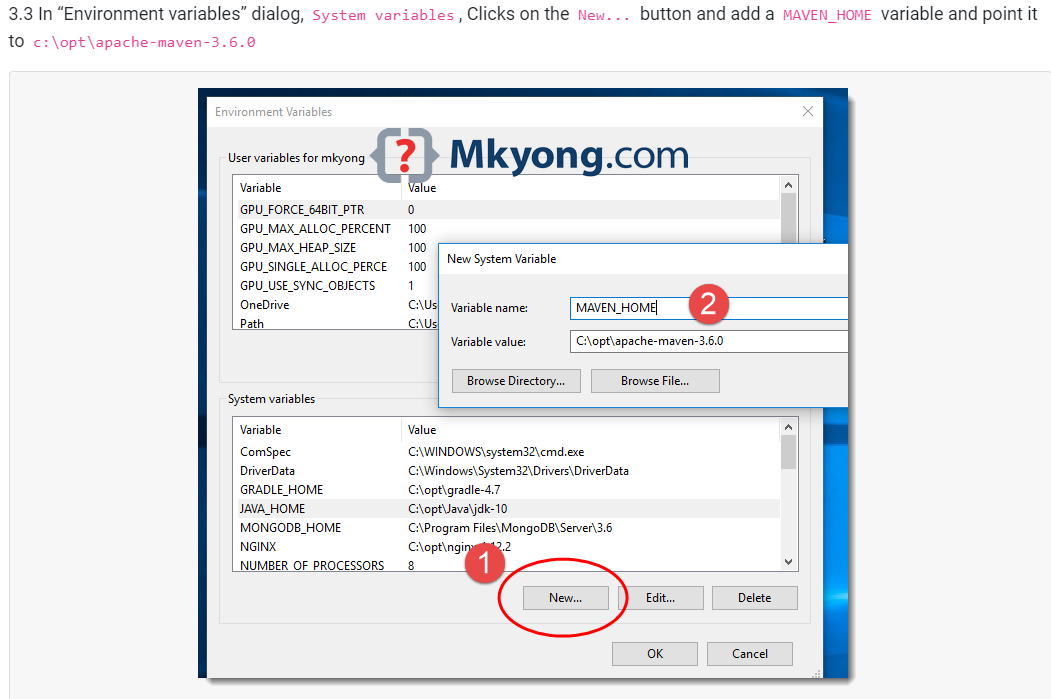


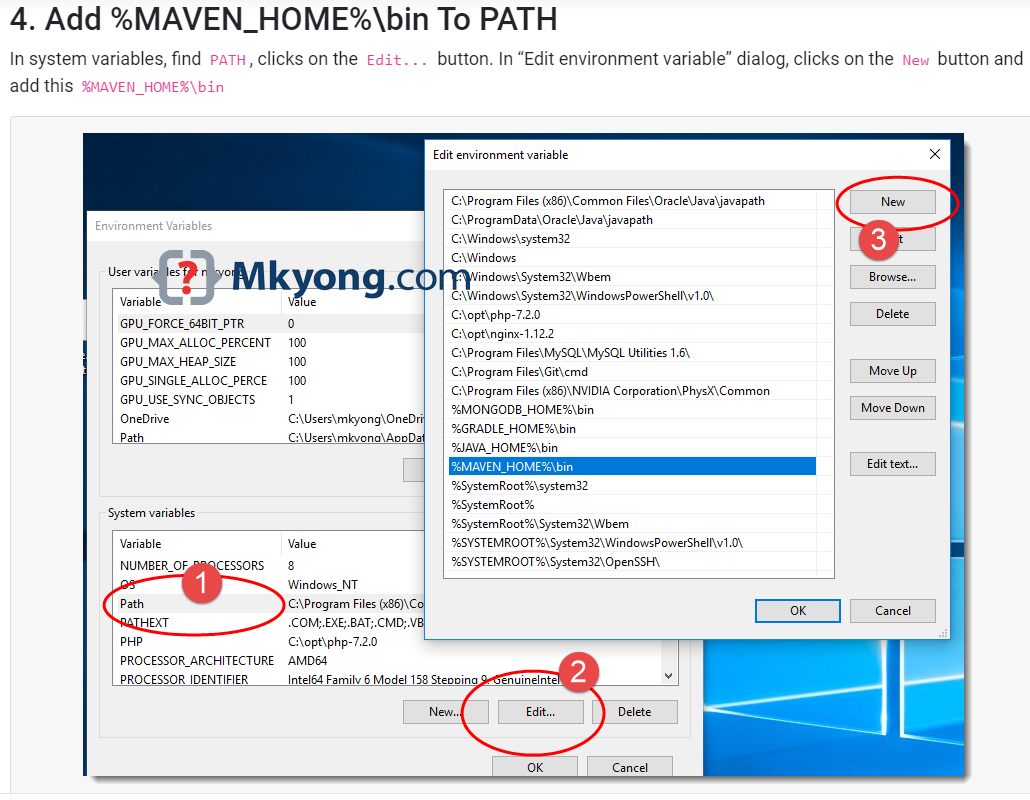
Download the above highlighted source file and Unzip the Folder.

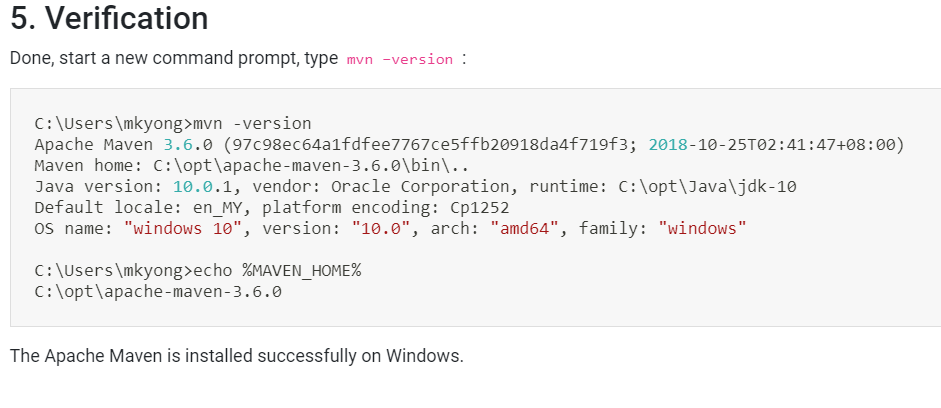
.







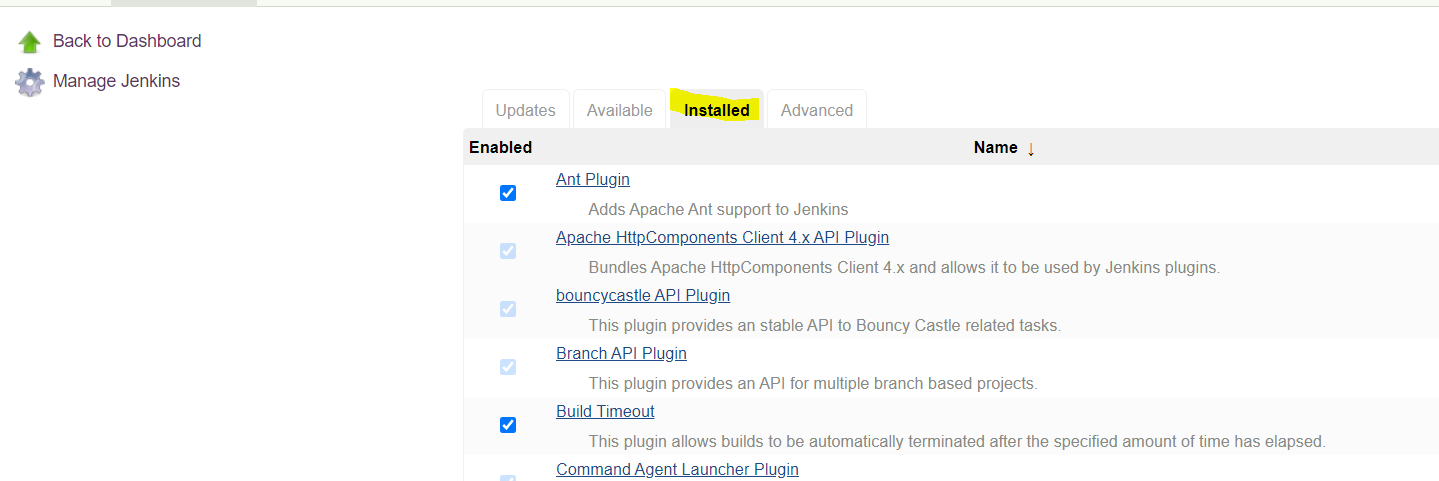




**Step-5:** We have to install below plug-in in Jenkins:

**Note – Some of these would already be installed if you chose to install suggested plugins at Jenkins startup**

**You can check installed plugins in Installed tab-**



1.Git

2.Github

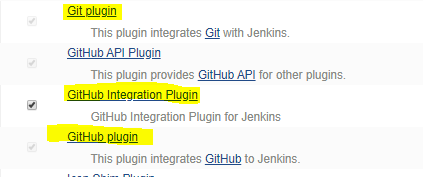
3.Maven Integration

4.GitHub Integration

For installing the above Plug-in, Follow the below steps:

**[Manage Jenkins] 🡪 [Manage Plugins]**

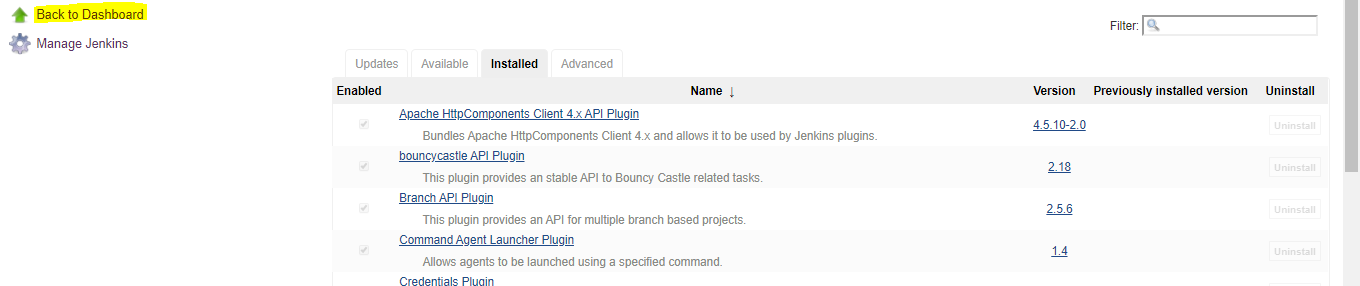
****

****

****

**Step-6:** Now create a **New Project** in Jenkins by following the below steps:

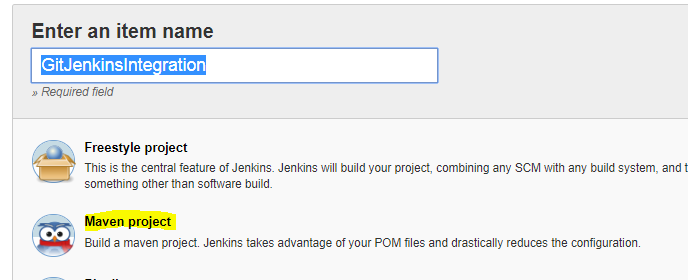
1.Click on **[Back to Dashboard]**



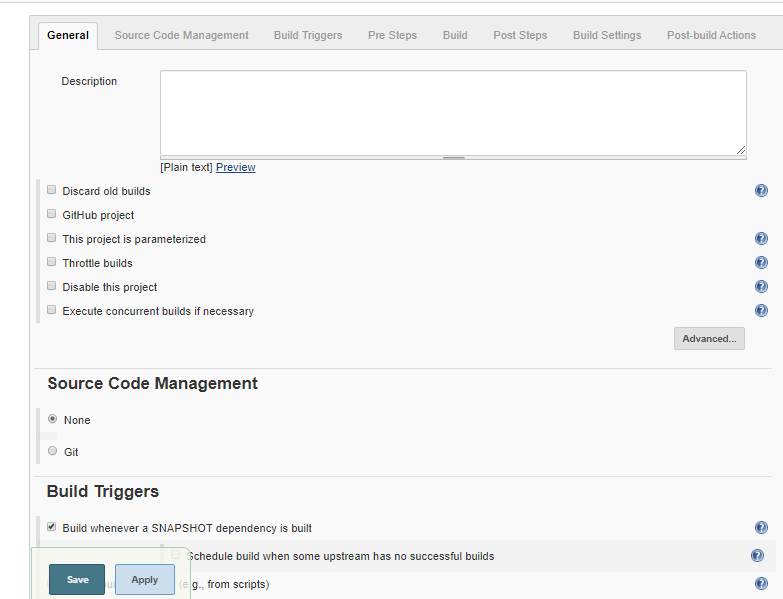
2. Click on **[New Item]**



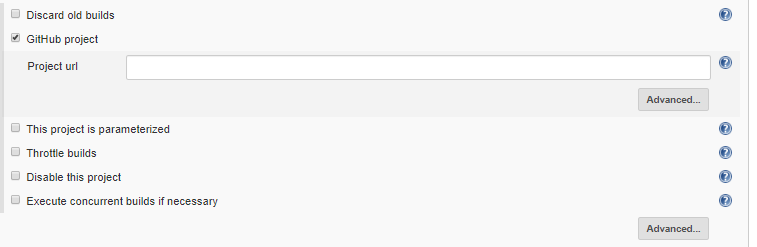
3. Create a **MAVEN** project and click on **[OK]** button at the bottom.



After clicking on [**OK**] button, Below screen will be displayed:

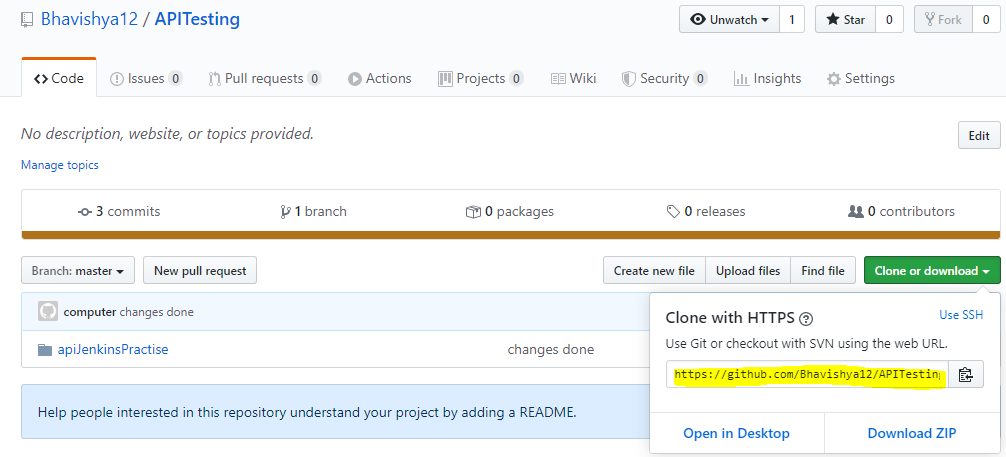


4. Select **GitHub** Project:

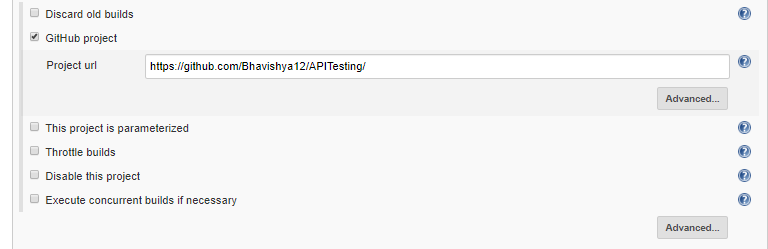


Follow below steps to provide path of GitHub Project in Jenkins

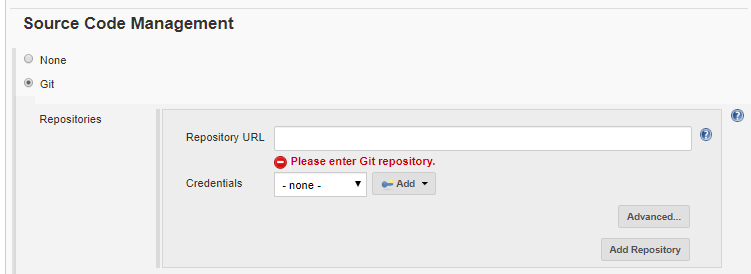
Open **github.com** and copy the path of your **GitHub repository** (Where your GitHub project is)



GitHub path provided inside Jenkins:

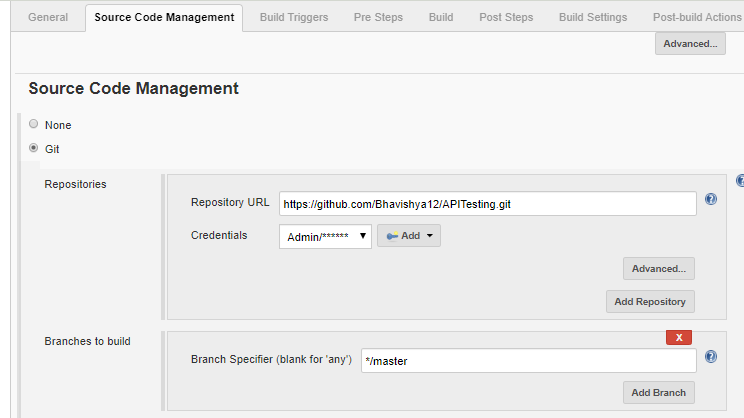


5. Select [**Git**] inside [**Source Code Management**]:



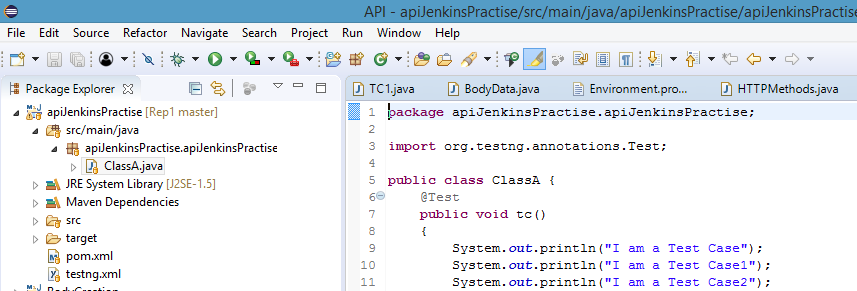
Now, Give the **Git** path and **Jenkins** Credentials:

If you are working with newly created User Id and password then provide those credentials else provide Admin User and password of Jenkins in Credentials

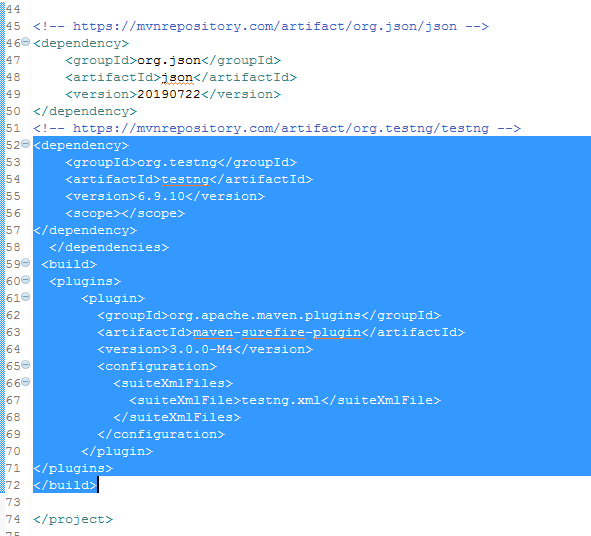


6. Now provide the **POM.XML** path in Jenkins

**Firstly**, Go to your Project in Eclipse:



We should have entry of **TestNG .xml** and **SureFire** plugin in our **POM.XML**



**SureFire Plugin:**

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M4</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

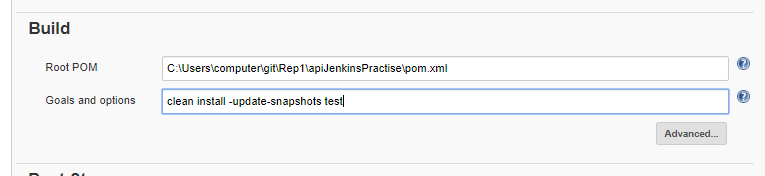
</suiteXmlFiles>

</configuration>

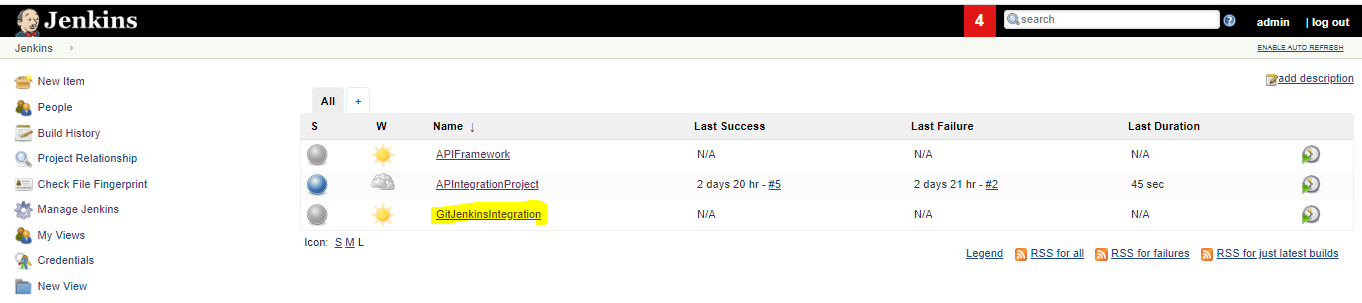
</plugin>

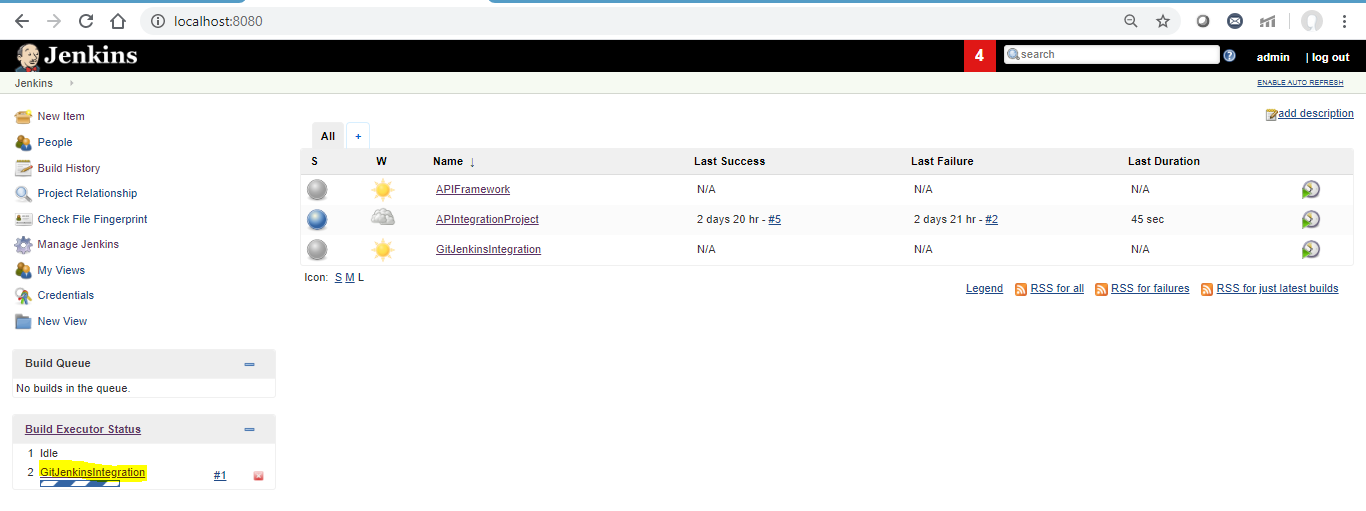
</plugins>

Now, copy the path of **POM.XML** and paste it in Jenkins along with Maven Commands inside “**Goals and Options**”:

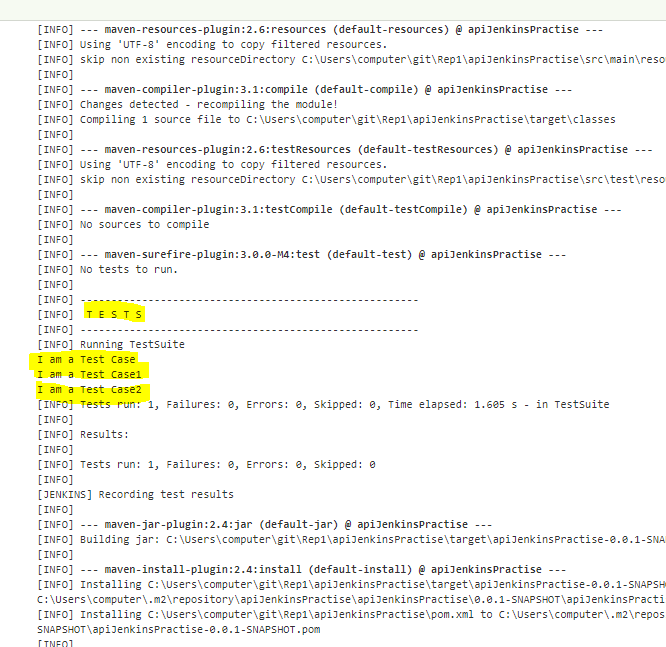


Now go to your Project and trigger the Build:





Click on then **Build** and then click on **[Console Output]:**

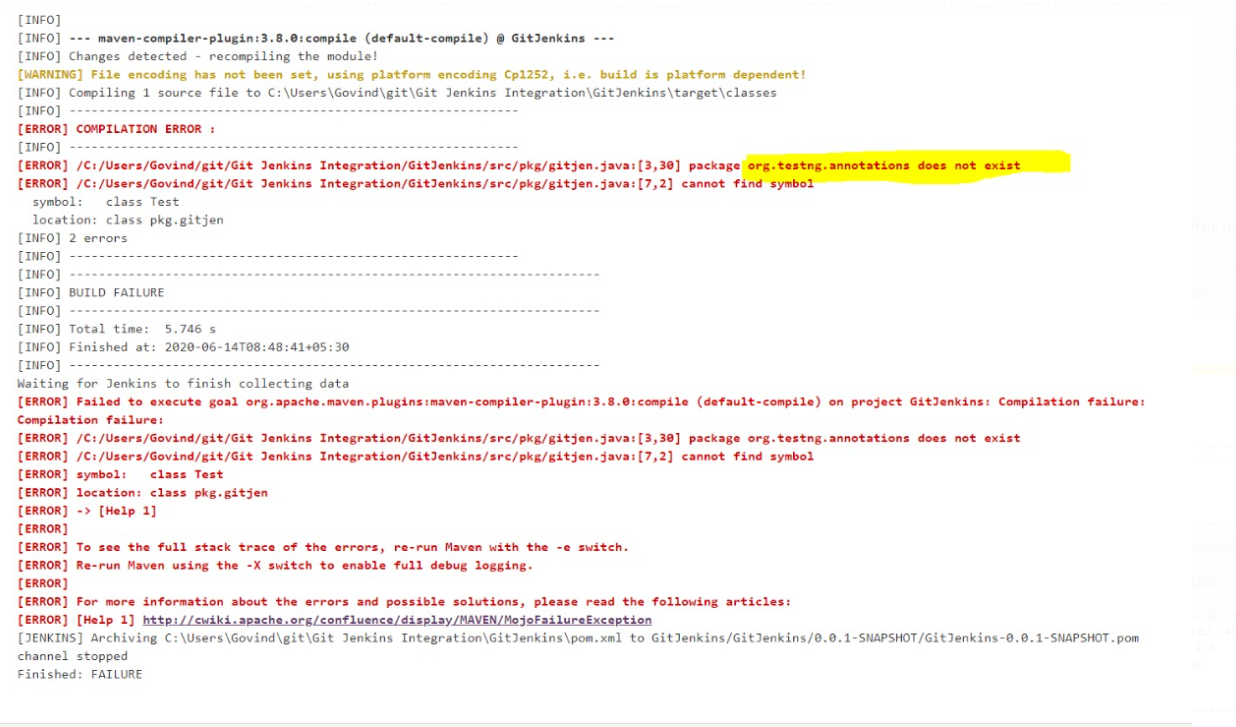




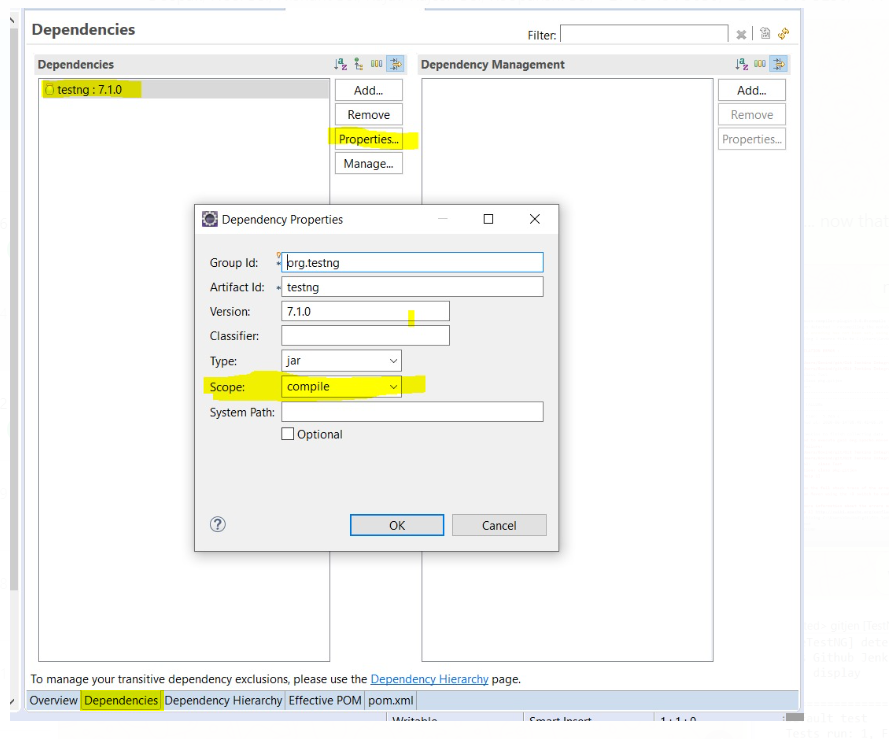
**Troubleshooting Errors if Build Fails in Jenkins**

1)

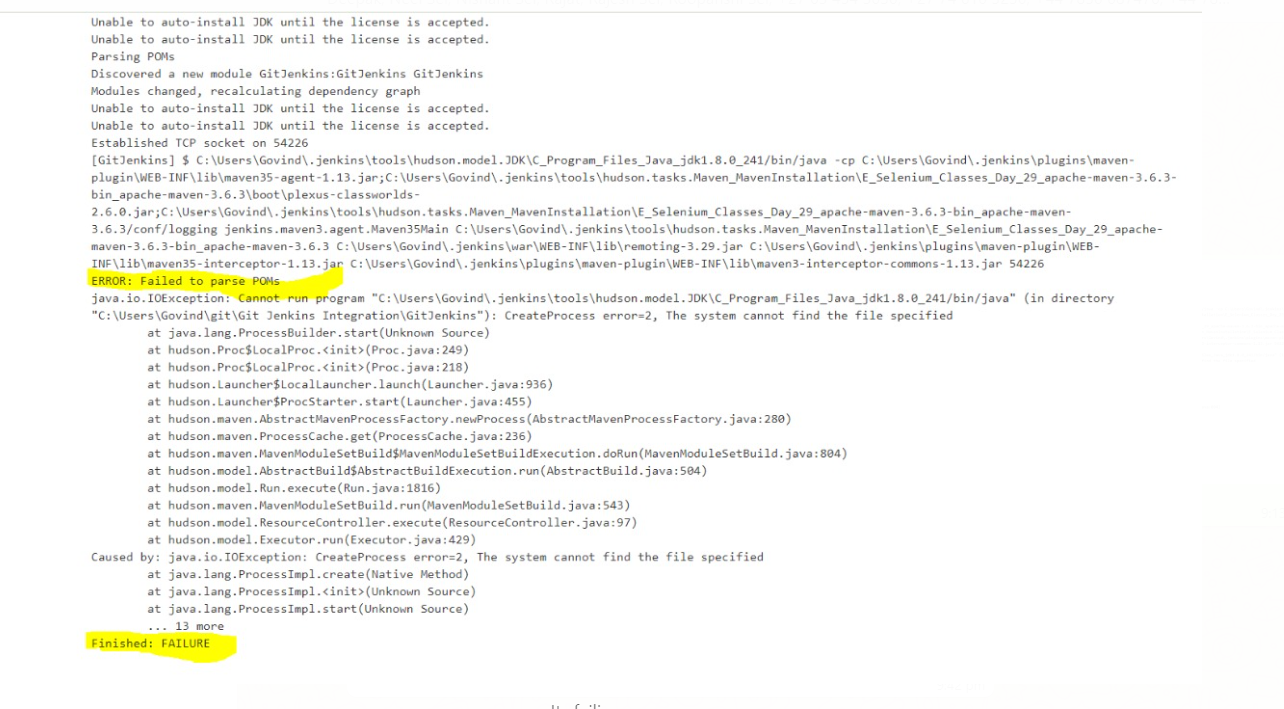
If you are getting below error –



To resolve the above error you need to go to POM.xml in Eclipse and change the scope of the dependencies to “Compile” from “Test”



2) If you are getting below error –



Then double check the JDK path is configured correctly

………………Thanks for reviewing this Document……………..