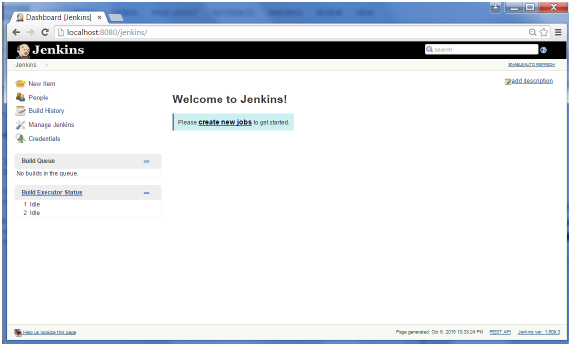
Jenkins - Git Setup

For this exercise, you have to ensure that Internet connectivity is present from the machine on which Jenkins is installed. In your Jenkins Dashboard (Home screen), click the Manage Jenkins option on the left hand side.



In the next screen, click the ‘Manage Plugins’ option.



In the next screen, click the Available tab. This tab will give a list of plugins which are available for downloading. In the ‘Filter’ tab type ‘Git plugin’



The list will then be filtered. Check the Git Plugin option and click on the button ‘Install without restart’



The installation will then begin and the screen will be refreshed to show the status of the download.



Once all installations are complete, restart Jenkins by issue the following command in the browser. **http://localhost:8080/jenkins/restart**

After Jenkins is restarted, Git will be available as an option whilst configuring jobs. To verify, click on New Item in the menu options for Jenkins. Then enter a name for a job, in the following case, the name entered is ‘Demo’. Select ‘Freestyle project’ as the item type. Click the Ok button.



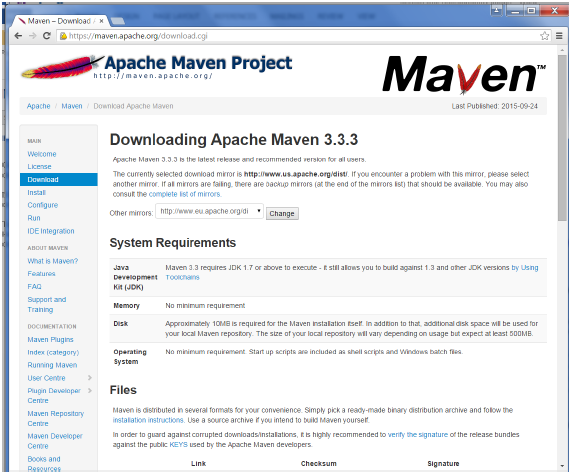
In the next screen, if you browse to the Source code Management section, you will now see ‘Git’ as an option.



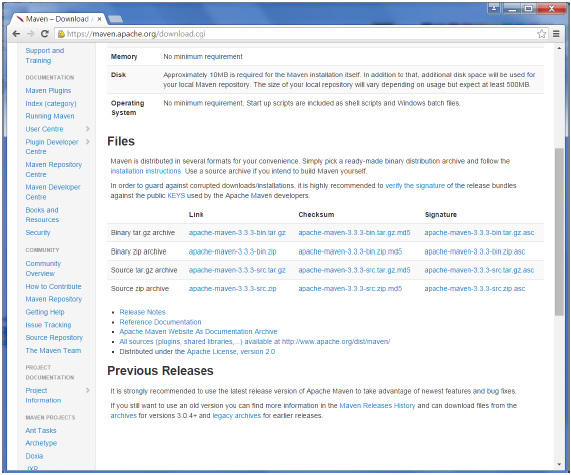
# Jenkins – Maven Setup

## **Step 1: Downloading and Setting Up Maven**

The official website for maven is Apache Maven. If you click the given link, you can get the home page of the maven official website as shown below.



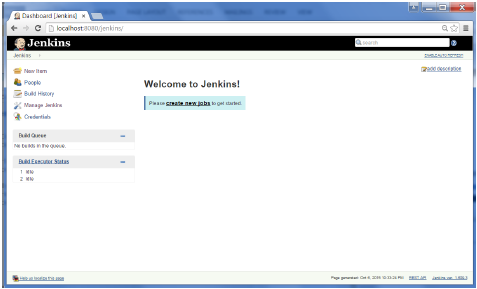
While browsing to the site, go to the Files section and download the link to the Binary.zip file.



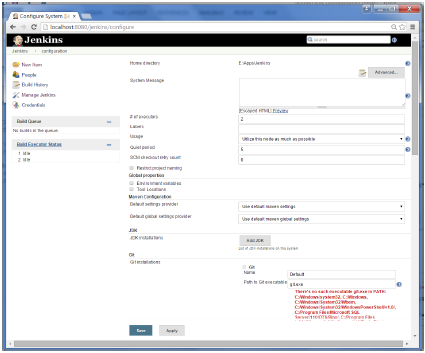
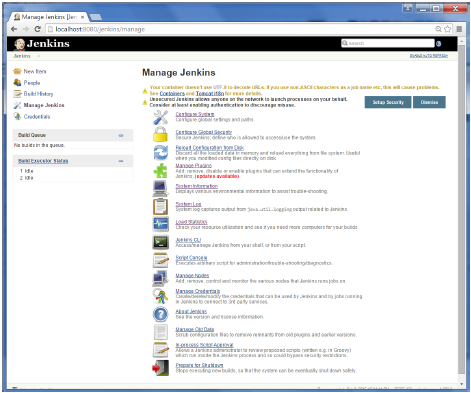
Once the file is downloaded, extract the files to the relevant application folder. For this purpose, the maven files will be placed in E:\Apps\apache-maven-3.3.3.

## **Step 2: Setting up Jenkins and Maven**

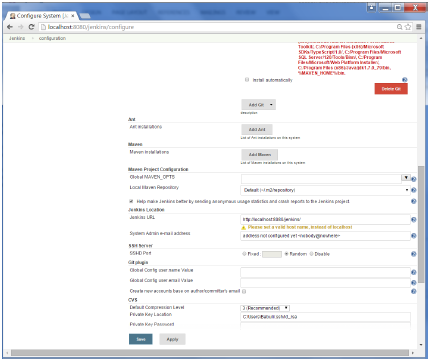
In the Jenkins dashboard (Home screen), click Manage Jenkins from the left-hand side menu.



Then, click on ‘Configure System’ from the right hand side.



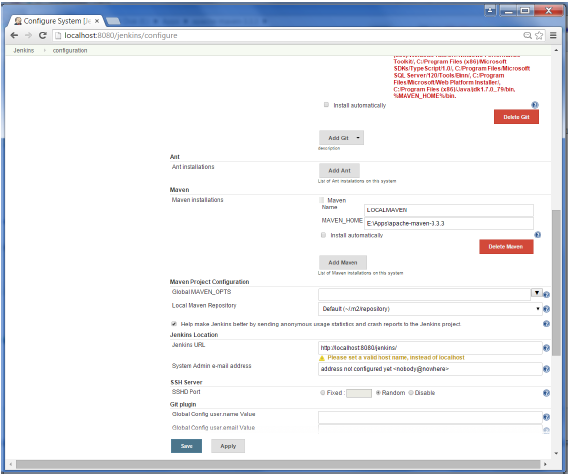
In the Configure system screen, scroll down till you see the Maven section and then click on the ‘Add Maven’ button.



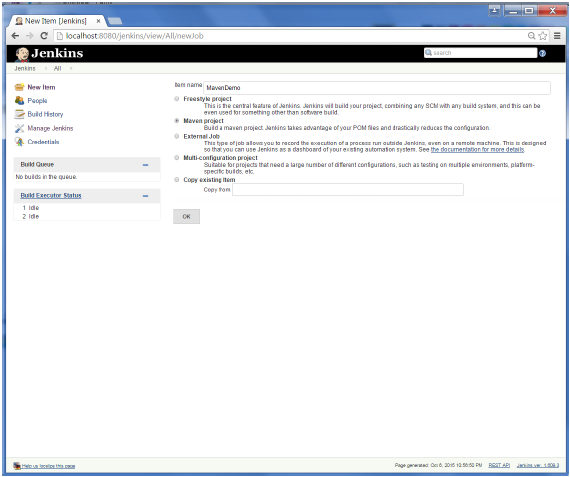
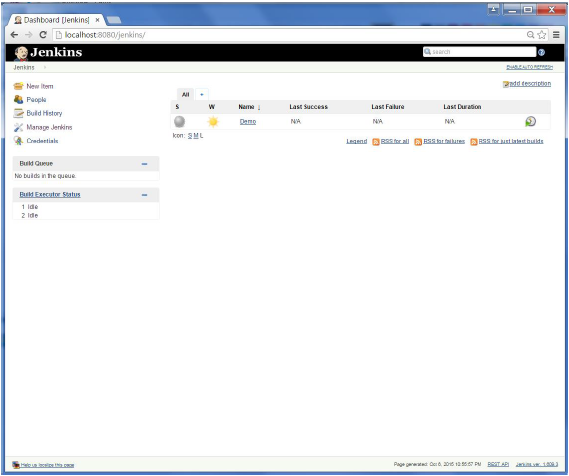
Uncheck the ‘Install automatically’ option.

Add any name for the setting and the location of the MAVEN\_HOME.

Then, click on the ‘Save’ button at the end of the screen.



You can now create a job with the ‘Maven project’ option. In the Jenkins dashboard, click the New Item option.



## Installation of Plugins in Jenkins

Jenkins comes with a pretty basic setup, so you will need to install the required plugins to enable respective third-party application support.

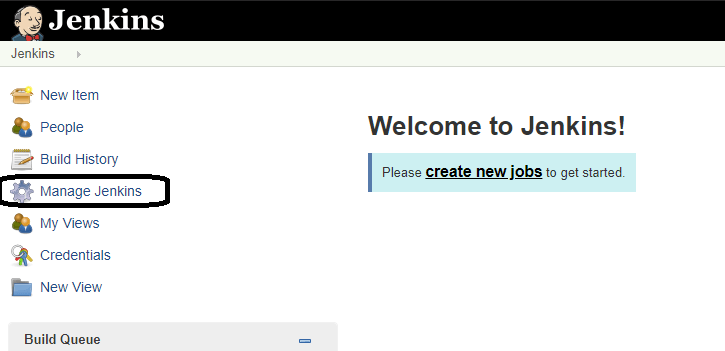
GitHub is a web-based repository of code which plays a major role in DevOps. It provides a common platform for multiple developers working on the same code/project to upload and retrieve updated code, thereby facilitating continuous integration.

Jenkins needs to have GitHub plugin installed to be able to pull code from the GitHub repository.

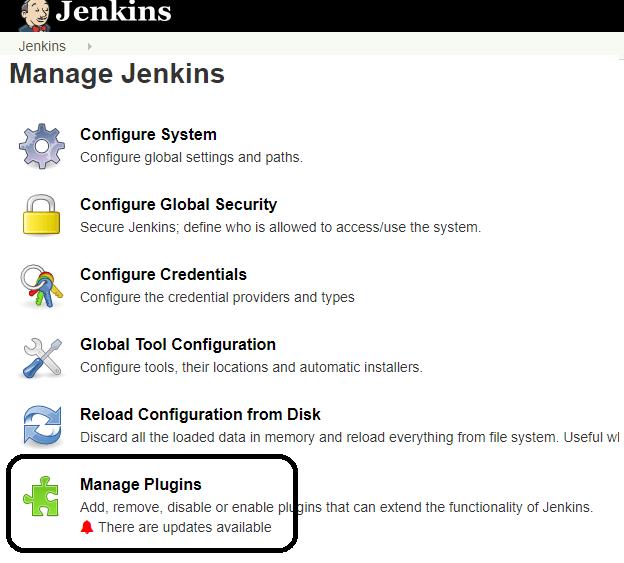
You need not install a GitHub plugin if you have already installed the Git plugin in response to the prompt during the Jenkins' installation setup. But if not, here is how you install GitHub plugins in Jenkins and pull code from a GitHub repository.

## Install GIT Plugin

**Step 1:**Click on the **Manage Jenkins** button on your Jenkins dashboard:

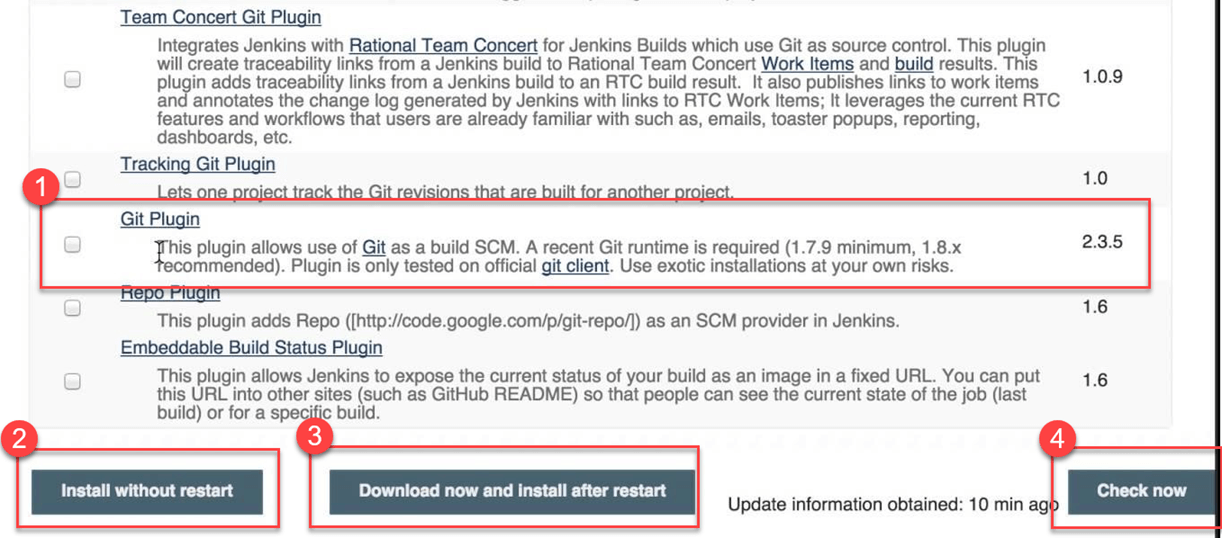


**Step 2:**Click on **Manage Plugins**:

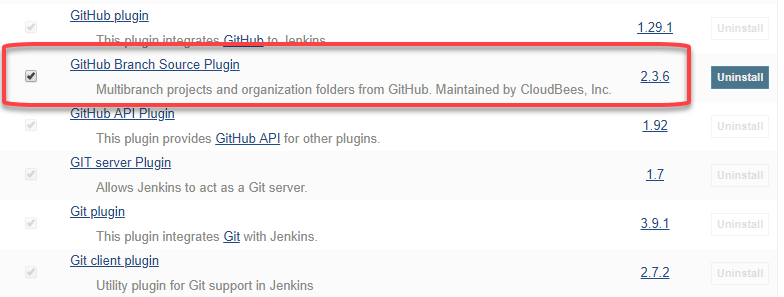


**Step 3:**In the Plugins Page

1. Select the GIT Plugin
2. Click on **Install without restart.**The plugin will take a few moments to finish downloading depending on your internet connection, and will be installed automatically.
3. You can also select the option **Download now and Install after restart** button. In which plugin is installed after restart
4. You will be shown a "No updates available" message if you already have the Git plugin installed.



**Step 4:** Once the plugins have been installed, go to **Manage Jenkins** on your Jenkins dashboard. You will see your plugins listed among the rest.

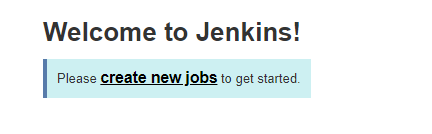


## Integrating Jenkins with GitHub

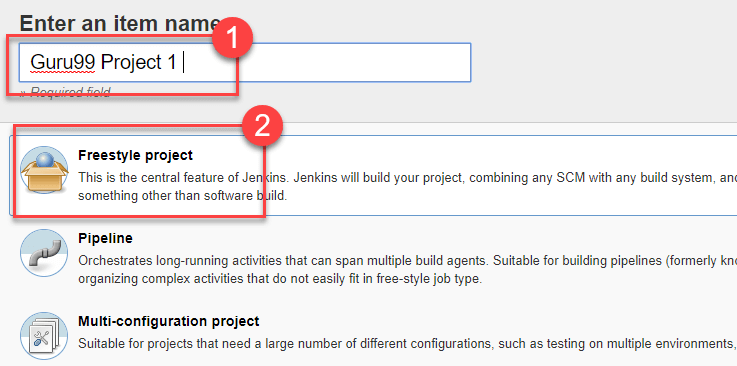
We shall now discuss the process of integrating GitHub into Jenkins in a Windows system.

**Step 1)**Create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL. For example, http://localhost:8080/

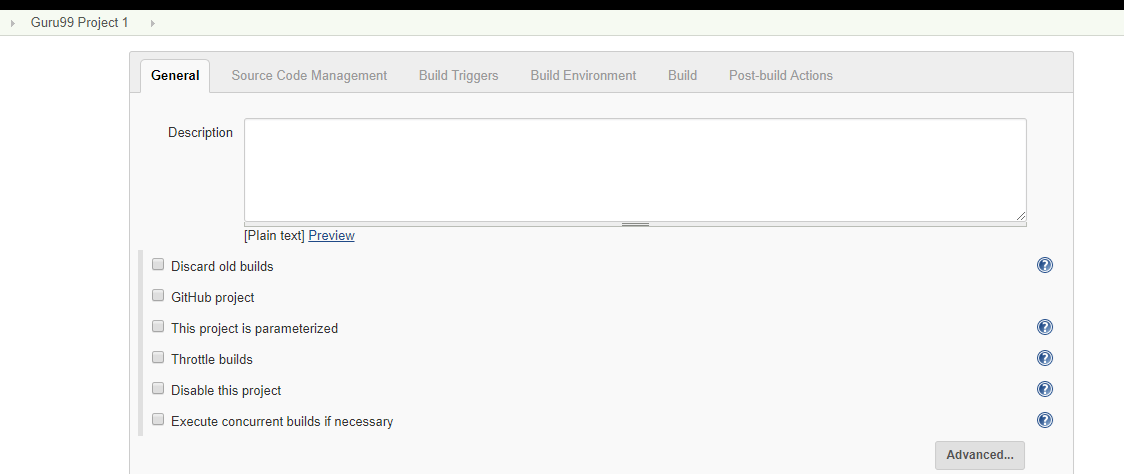
Click on **create new jobs**:



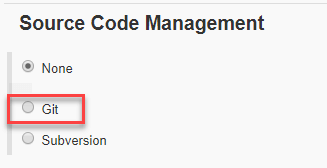
**Step 2)**Enter the item name, select job type and click **OK**. We shall create a Freestyle project as an example.



**Step 3)**Once you click **OK,**the page will be redirected to its project form. Here you will need to enter the project information:

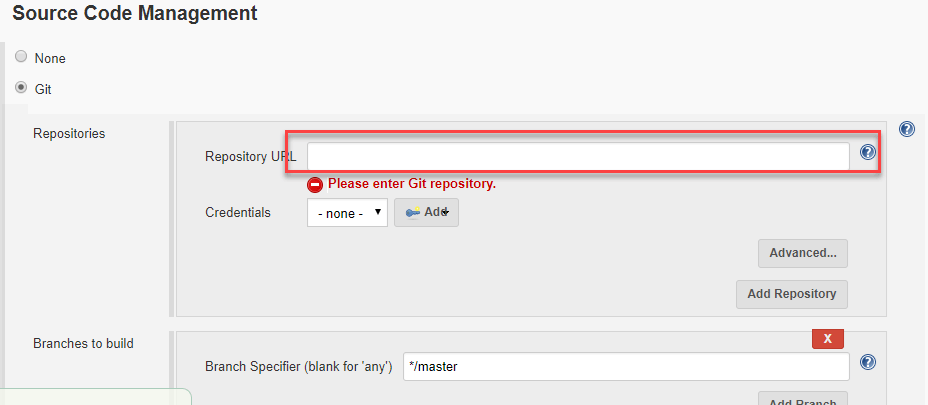


**Step 4)**You will see a **Git**option under **Source Code Management**if your Git plugin has been installed in Jenkins:

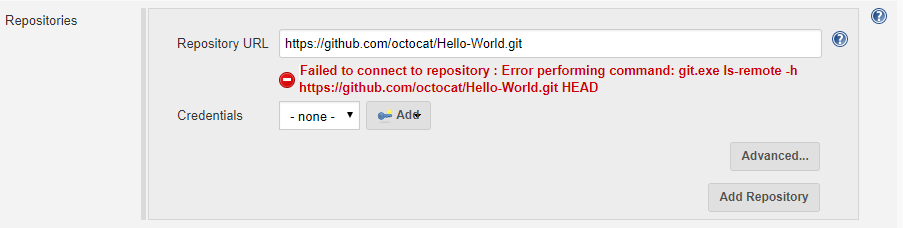


NOTE: If the **Git**option does not appear, try re-installing the plugins, followed by a restart and a re-login into your Jenkins dashboard.You will now be able to see the **Git**option as mentioned above.

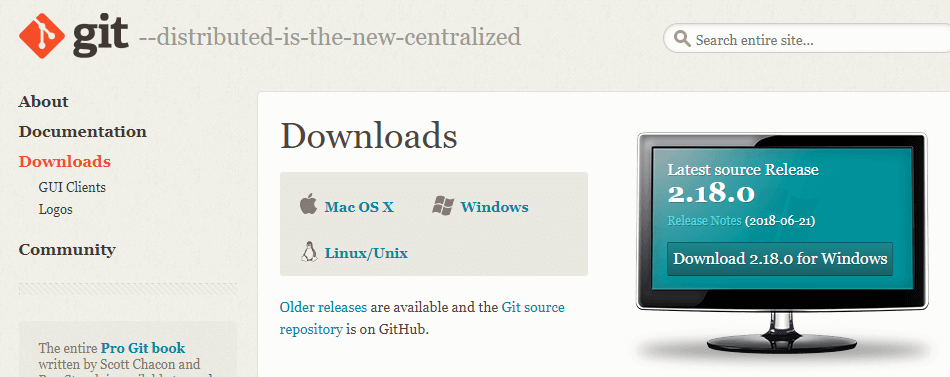
**Step 5)**Enter the Git repository URL to pull the code from GitHub.



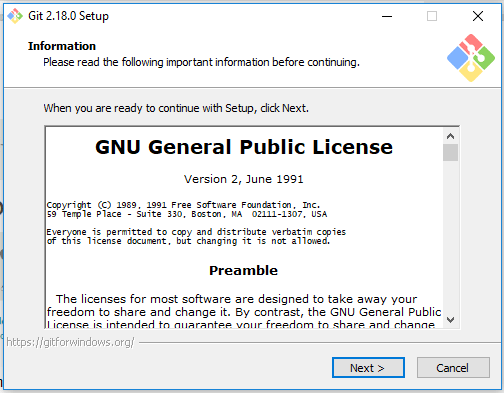
**Step 6)**You might get an error message the first time you enter the repository URL. For example:



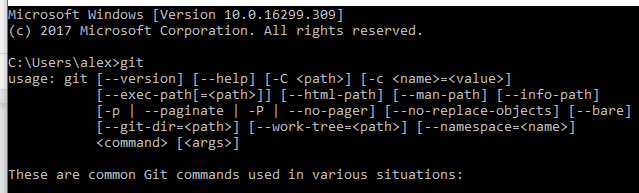
This happens if you do not have Gitinstalled in your local machine. To install Git in your local machine, go to https://git-scm.com/downloads



Download the appropriate Git file for your Operating System, in this case, Windows, and install it onto your local machine running Jenkins. Complete the onscreen instructions to install GIT.



**Step 6)**You can execute Git repositories in your Jenkins once Git has been installed on your machine. To check ifithas been successfully installed onto your system, open your **command prompt,** type "Git"and press enter. You should see different options come up for Git:

[](https://www.guru99.com/images/1/091318_0440_JenkinsGitH14.png)

This means that Git has been installed in your system.

**Step 7)**Once you have everything in place, try adding the Git URL into Jenkins. You will not see any error messages:

