

Jenkins Tutorial



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.

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.

Jenkins offers a straightforward way to set up a continuous integration or continuous delivery environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks.

With the help of Jenkins, organizations can speed up the software development process through automation. Jenkins adds development life-

cycle processes of all kinds, including build, document, test, package, stage, deploy static analysis and much more.

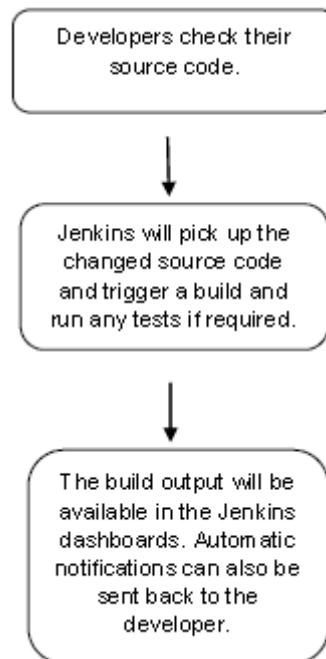
Jenkins achieves CI (Continuous Integration) with the help of plugins. Plugins is used to allow the integration of various DevOps stages. If you want to integrate a particular tool, you have to install the plugins for that tool. For example: Maven 2 Project, Git, HTML Publisher, Amazon EC2, etc.

For example: If any organization is developing a project, then **Jenkins** will continuously test your project builds and show you the errors in early stages of your development.

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

Work Flow:



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 Continuous Integration?

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.

Continuous Integration with Jenkins

Let's consider a scenario where the complete source code of the application was built and then deployed on test server for testing. It sounds like a perfect way to *develop software*, but this process has many problems.

- Developer teams have to wait till the complete software is developed for the test results.
- There is a high prospect that the test results might show multiple bugs. It was tough for developers to locate those bugs because they have to check the entire source code of the application.
- It slows the software delivery process.

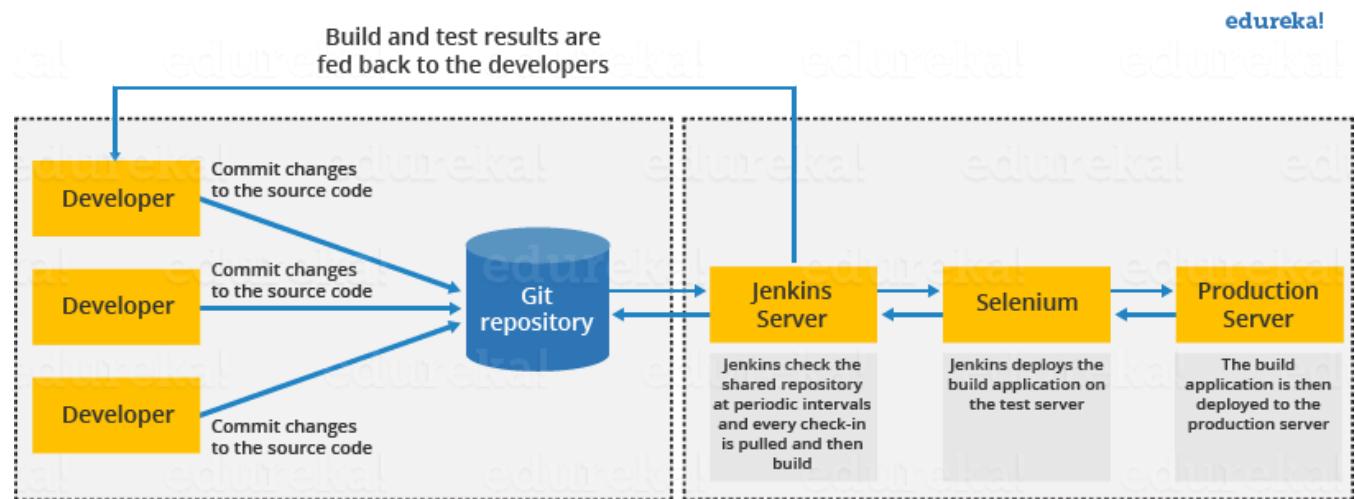
- Continuous feedback pertaining to things like architectural or coding issues, build failures, test status and file release uploads was missing due to which the quality of software can go down.
- The whole process was manual which increases the threat of frequent failure.

It is obvious from the above stated problems that not only the software delivery process became slow but the quality of software also went down. This leads to customer dissatisfaction.

So to overcome such problem there was a need for a system to exist where developers can continuously trigger a build and test for every change made in the source code.

This is what Continuous Integration (CI) is all about. Jenkins is the most mature Continuous Integration tool available so let us see how Continuous Integration with Jenkins overcame the above shortcomings.

Let's see a generic flow diagram of Continuous Integration with Jenkins:



Let's see how Jenkins works. The above diagram is representing the following functions:

- 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.

Advantages and Disadvantages of using Jenkins

Advantages 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.

Disadvantages of Jenkins

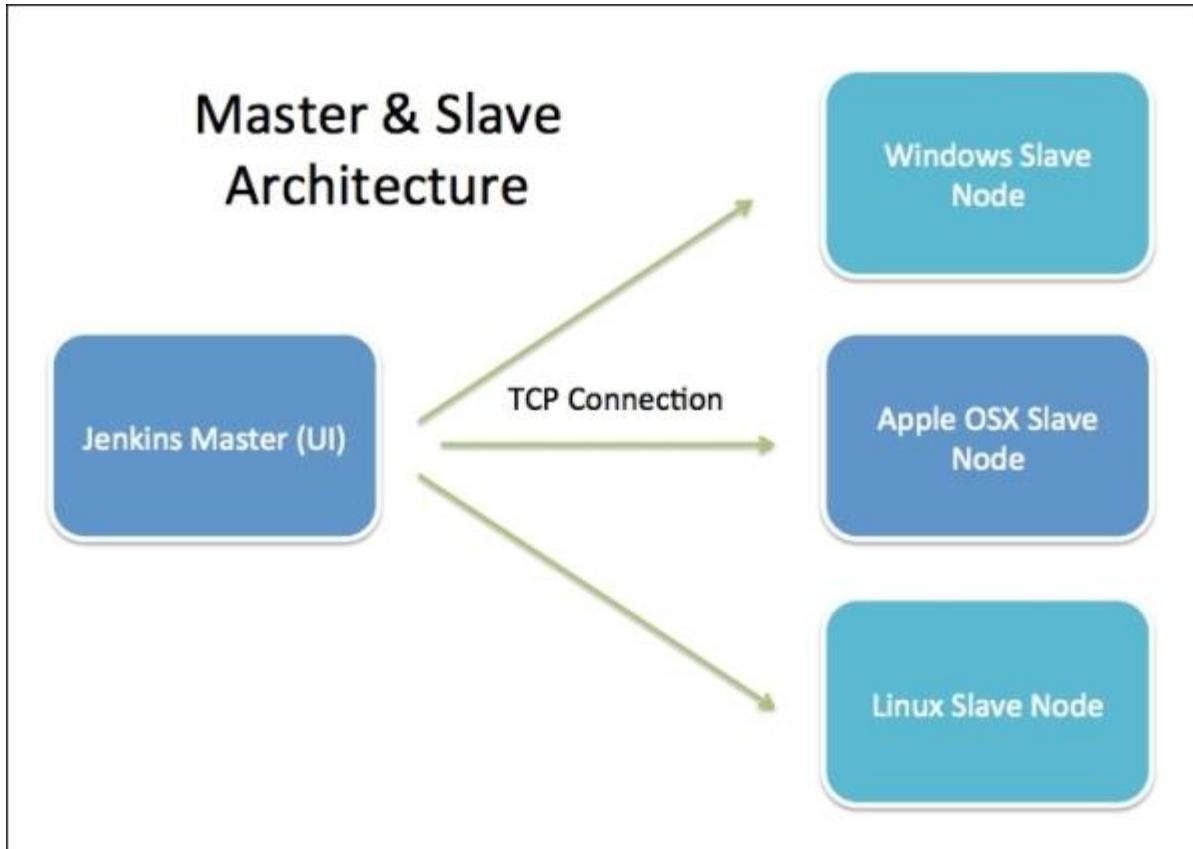
- Its interface is out dated and not user friendly compared to current user interface trends.
- Not easy to maintain it because it runs on a server and requires some skills as server administrator to monitor its activity.
- CI regularly breaks due to some small setting changes. CI will be paused and therefore requires some developer's team attention.

Jenkins Architecture

Jenkins follows Master-Slave architecture to manage distributed builds. In this architecture, slave and master communicate through TCP/IP protocol.

Jenkins architecture has two components:

- Jenkins Master/Server
- Jenkins Slave/Node/Build Server



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. We can add more nodes using IP address, user name and password using the ssh, jnlp or webstart methods.

The server's job or master's job is to handle:

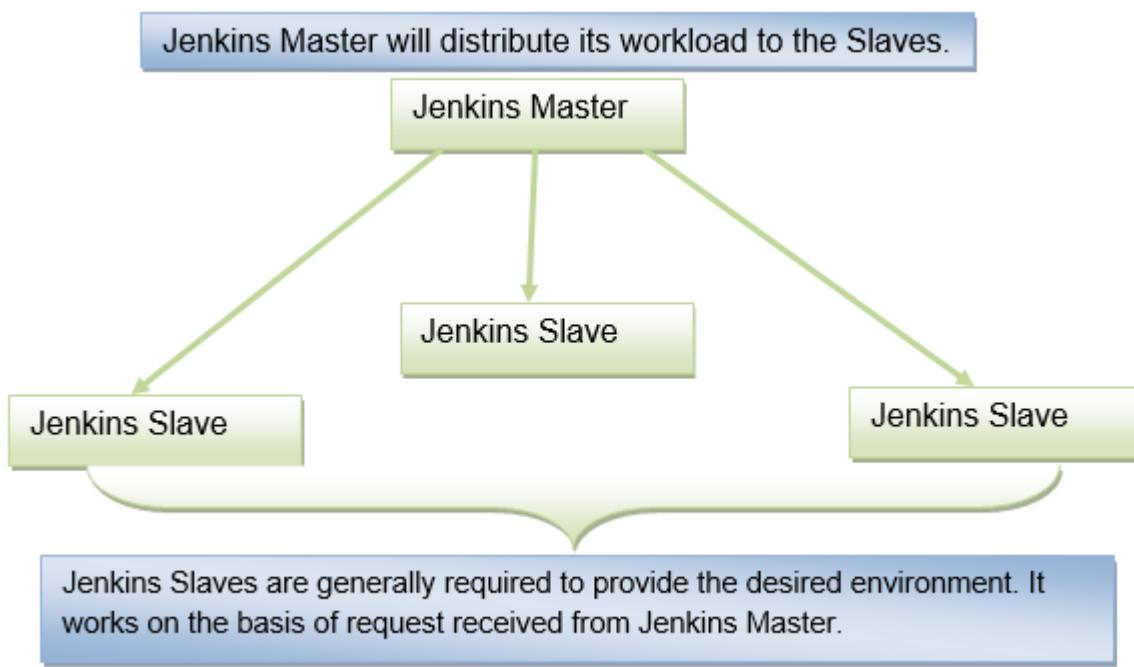
- Scheduling build jobs.
- Dispatching builds to the nodes/slaves for the actual execution.
- Monitor the nodes/slaves (possibly taking them online and offline as required).

- 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.



The above diagram is self explanatory. It consists of a Jenkins Master which is managing three Jenkins Slaves.

Prerequisite

Before learning Jenkins, you should have a basic understanding of testing and Java.

Audience

Our Jenkins Tutorial is designed to help beginners and professionals.

Problems

We assure that you will not find any problem with this Jenkins Tutorial. But if there is any mistake, please post the problem in the contact form.

Installing Jenkins on Windows

Hardware Requirements

Memory	2 GB RAM (Recommended)
Disk Space	We need at least 1 GB of space in our hard drive for Jenkins

Software Requirements

JDK	We need either Java Development (JDK) or Java Runtime Environment (JRE)
Operating System	Jenkins can be installed on Windows, Mac OS X, Ubuntu/Debian, Red Hat/Fedora/CentOS, openSUSE, FreeBSD, OpenBSD, Gentoo.
Java Container	The WAR (Web Application Resource) file can be run in any container that supports Servlet 2.4/JSP 2.0 or later. (For example Tomcat 5).

Release Types

Jenkins supports two types of releases based on the organization needs:

- Long-term support release
- Weekly release

Long term support release (LTS): Long-term support releases (LTS) are available every 12 weeks. They are stable and are widely used and tested. Basically, this release is intended for end users.

Weekly release: This release is available in every week by fixing bugs in its earlier version. These releases are intended towards plugin developers.

We will use the long term support release (LTS) though the process remains the same for Weekly release.

Install Java Version 8

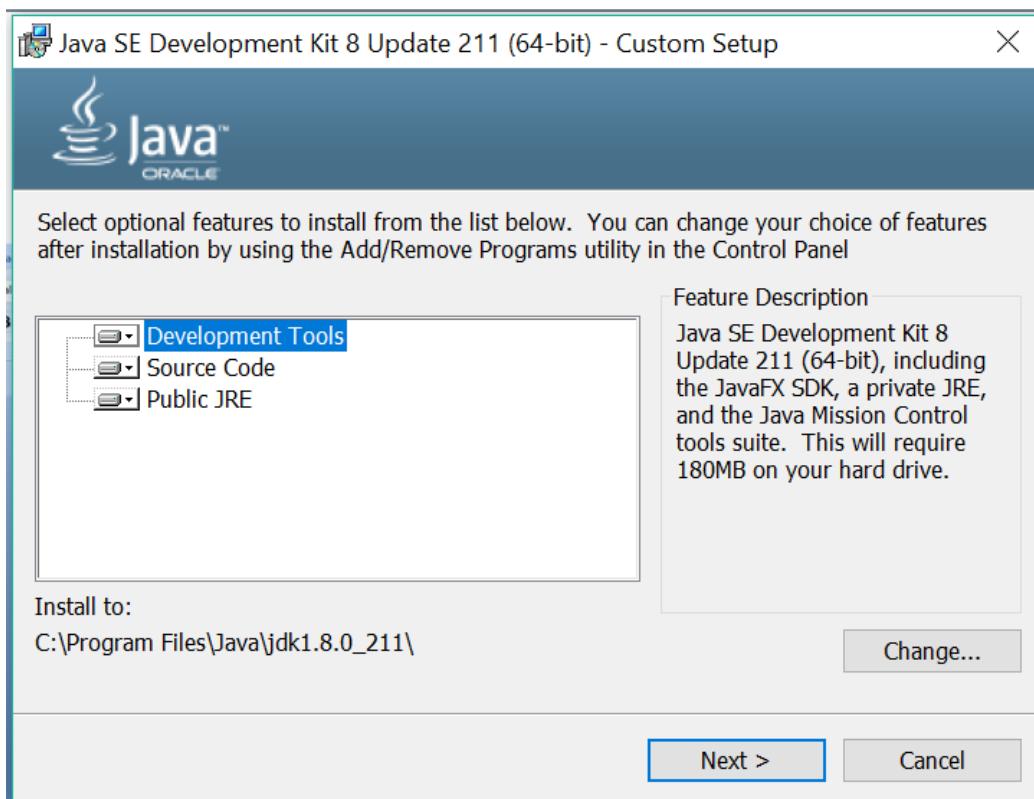
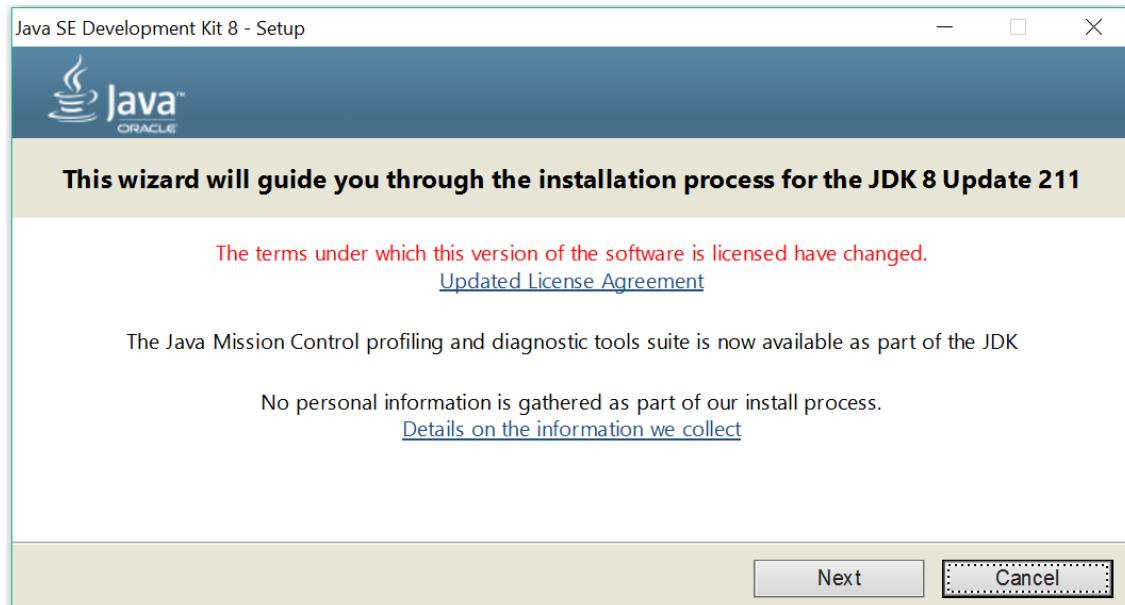
Since Jenkins is a Java based application, therefore Java is a must.

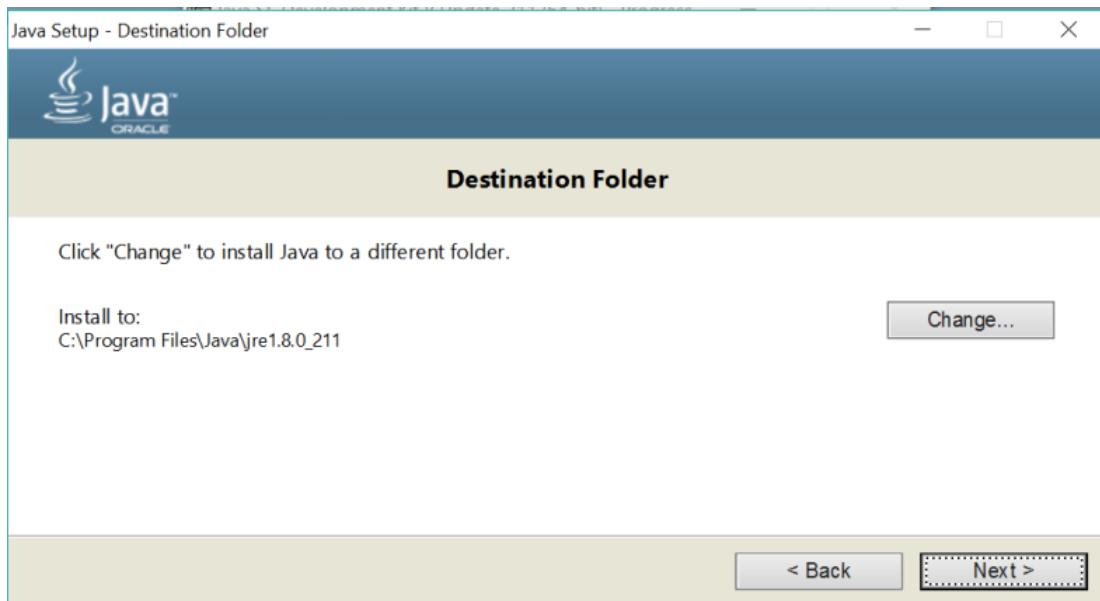
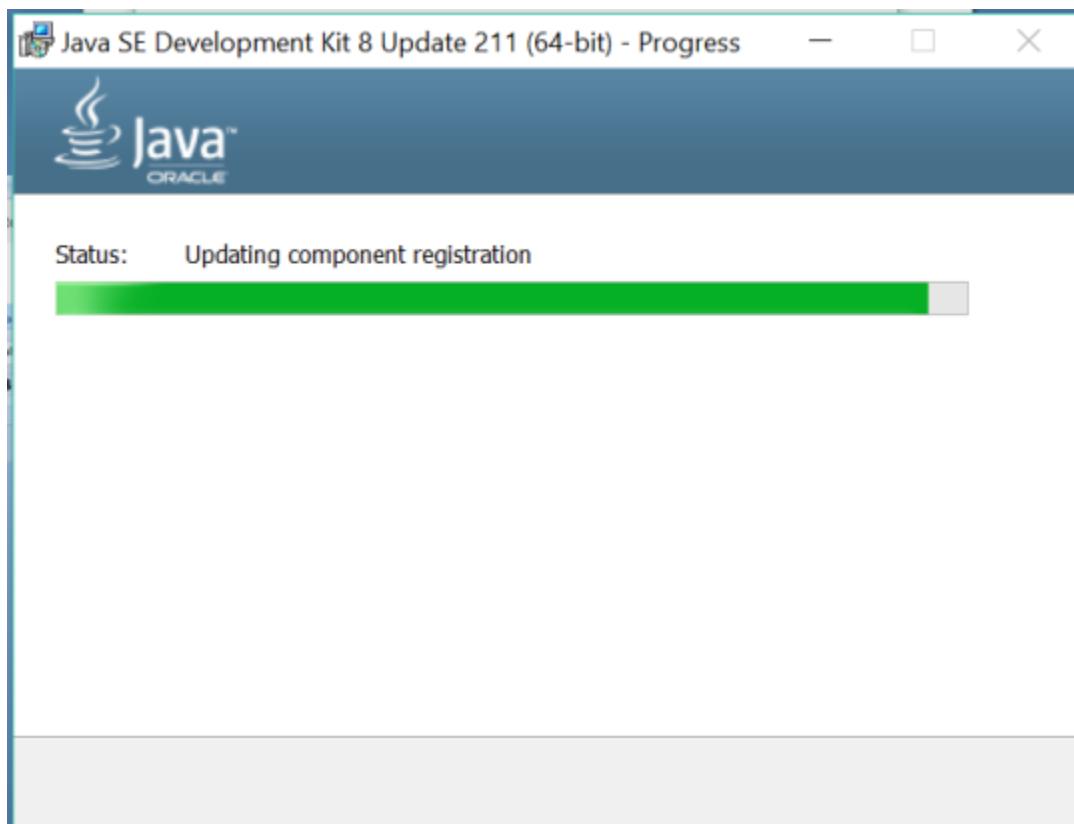
To download the Java [Click here](#). Select file according to your platform.

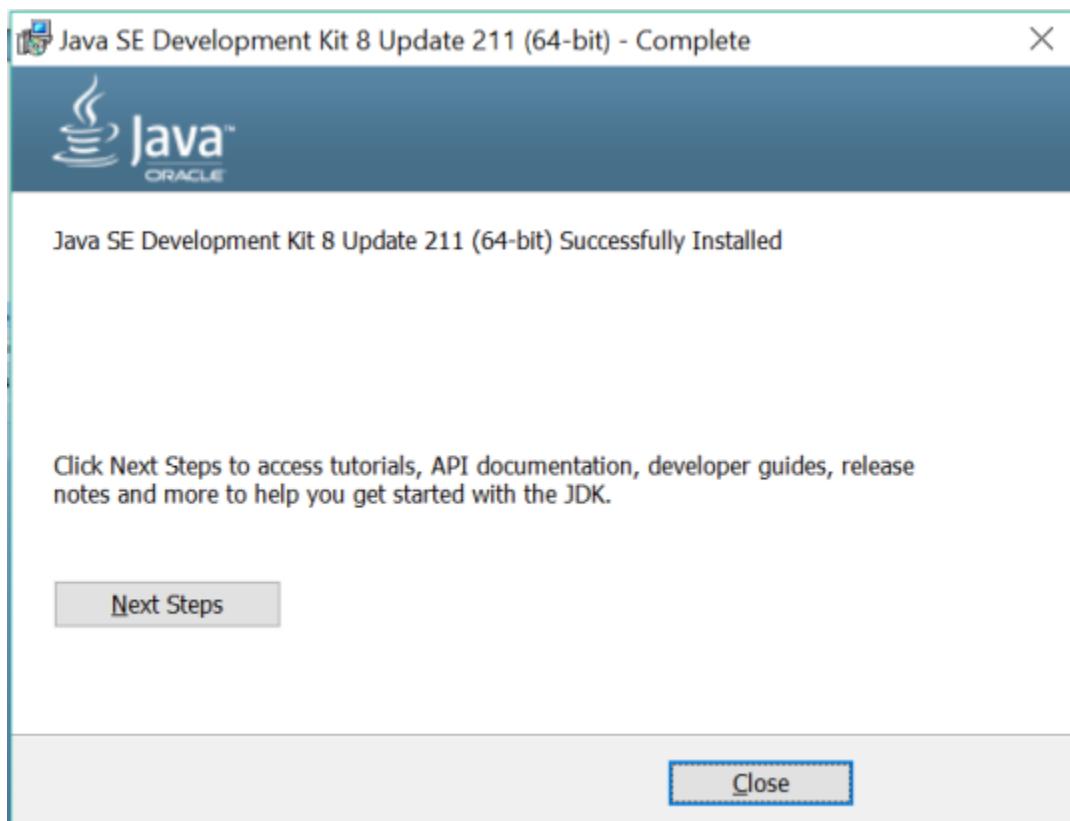
The screenshot shows the Oracle Java SE Development Kit 8u211 download page. At the top, there's a navigation bar with a lock icon and the URL <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Below the navigation bar, there's a "See also:" section with links to the Java Developer Newsletter, Java Developer Day workshops, and Java Magazine. It also includes links to JDK 8u211 and 8u212 checksums. The main content area has a title "Java SE Development Kit 8u211" and a note about accepting the Oracle Technology Network License Agreement. A table below lists Java distributions with their file sizes and download links.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u211-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.76 MB	jdk-8u211-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.11 MB	jdk-8u211-linux-i586.rpm
Linux x86	188.92 MB	jdk-8u211-linux-i586.tar.gz
Linux x64	171.13 MB	jdk-8u211-linux-x64.rpm
Linux x64	185.96 MB	jdk-8u211-linux-x64.tar.gz
Mac OS X x64	252.23 MB	jdk-8u211-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	jdk-8u211-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.18 MB	jdk-8u211-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.57 MB	jdk-8u211-solaris-x64.tar.Z
Solaris x64	91.93 MB	jdk-8u211-solaris-x64.tar.gz
Windows x86	202.62 MB	jdk-8u211-windows-i586.exe
Windows x64	215.29 MB	jdk-8u211-windows-x64.exe

Then install the Java as follows:





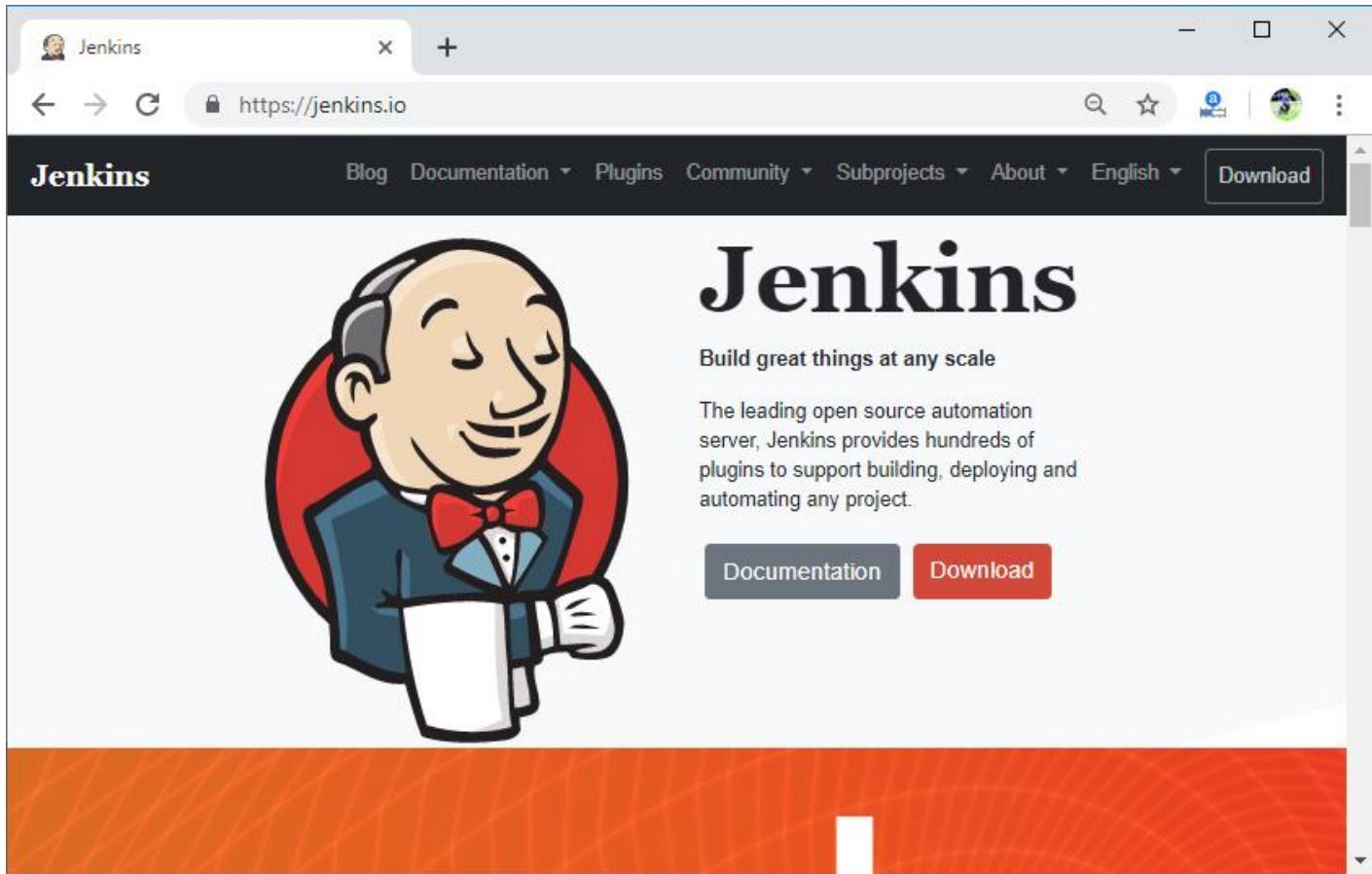


Download Jenkins war File

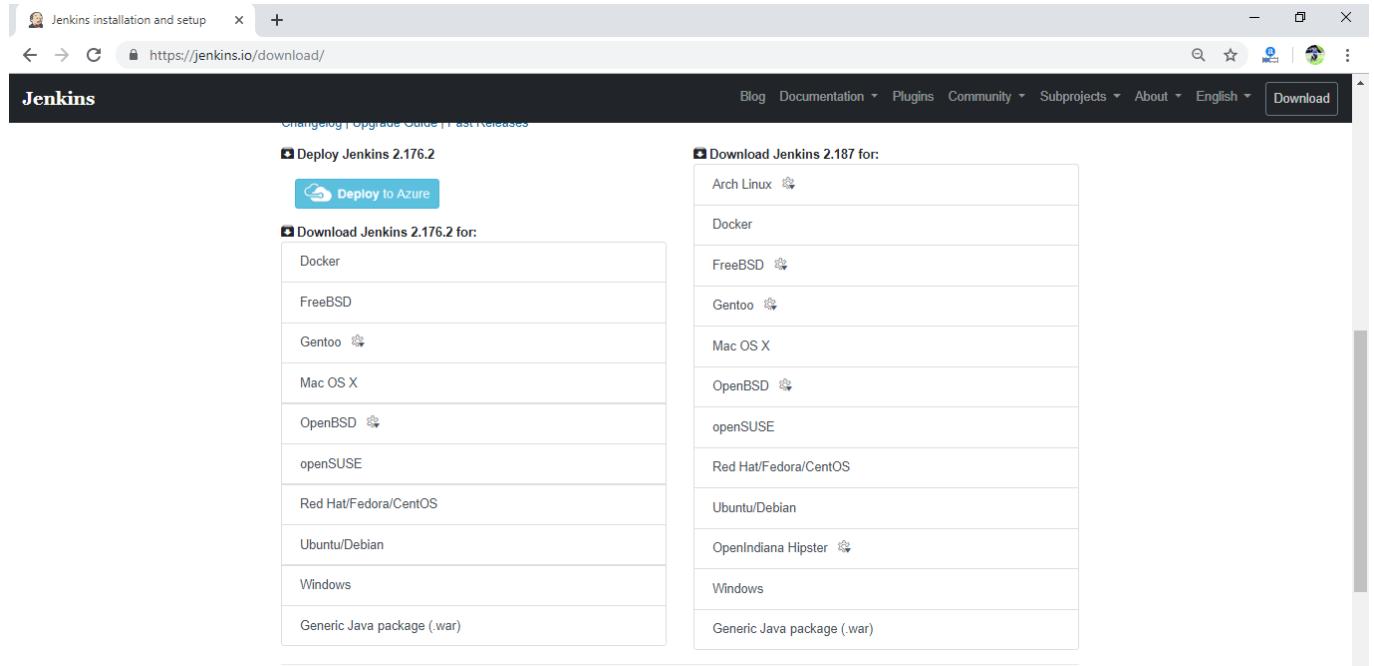
This war is required to install Jenkins.

The official website for Jenkins is <https://jenkins.io/>

When you click the given link, you will get the home page of the Jenkins official website as given below:



Click on the Download button.



Click on Generic Java Package (.war) to download the Jenkins war file.

Starting Jenkins

Open the command prompt and go to the directory where the Jenkins.war file is located. And then run the following command:

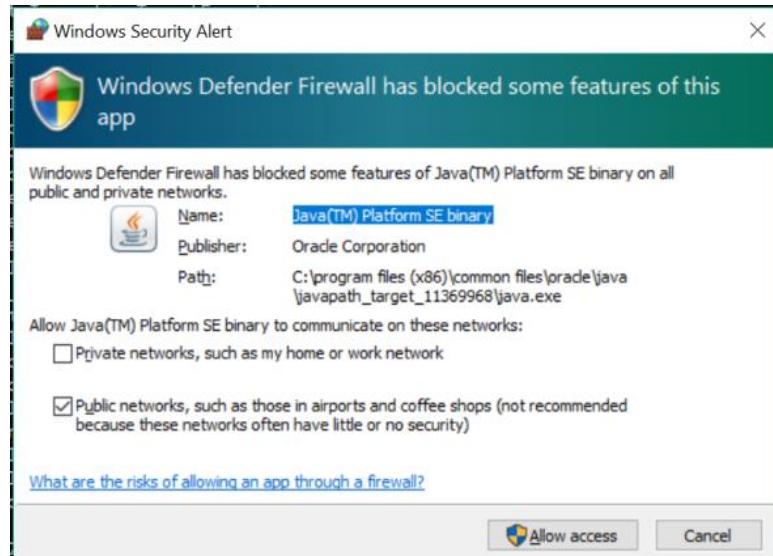
1. C:/Java -jar Jenkins.war

When you run this command, various tasks will run, one of which is the extraction of the war file which is done by an embedded webserver called winstome.

```
Command Prompt - Java - jar Jenkins.war
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nikita>cd c:\jenkins tutorial

c:\Jenkins Tutorial>Java -jar Jenkins.war
Running from: C:\Jenkins Tutorial\jenkins.war
webroot: $user.home/.jenkins
[Winstone 2019/07/06 18:11:26] - Beginning extraction from war file
hudson home directory: C:\Users\Nikita\.jenkins found at: $user.home/.jenkins
[Winstone 2019/07/06 18:11:26] - HTTP Listener started: port=8080
[Winstone 2019/07/06 18:11:26] - AJP13 Listener started: port=8009
Using one-time self-signed certificate
[Winstone 2019/07/06 18:11:26] - Error starting listener instance
java.lang.reflect.InvocationTargetException
    at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
    at sun.reflect.NativeConstructorAccessorImpl.newInstance(Unknown Source)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(Unknown Source)
    at java.lang.reflect.Constructor.newInstance(Unknown Source)
    at winstone.Launcher.spawnListener(Launcher.java:232)
    at winstone.Launcher.<init>(Launcher.java:205)
    at winstone.Launcher.main(Launcher.java:391)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at Main._main(Main.java:214)
    at Main.main(Main.java:61)
Caused by: java.lang.NoClassDefFoundError: sun/security/x509/CertAndKeyGen
    at winstone.ssl.HttpsListener.<init>(HttpsListener.java:111)
    ... 13 more
Caused by: java.lang.ClassNotFoundException: sun.security.x509.CertAndKeyGen
```



Click on **Allow access** button to allow the access.

```
Command Prompt - Java -jar Jenkins.war
... 14 more

[Winstone 2019/07/06 18:11:26] - Winstone Servlet Engine v0.9.10 running: controlPort=disabled
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Started initialization
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Listed all plugins
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Prepared all plugins
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Started all plugins
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Augmented all extensions
Jul 06, 2019 6:11:27 PM hudson.model.Hudson$5 onAttained
INFO: Loaded all jobs
Jul 06, 2019 6:11:28 PM hudson.model.Hudson$5 onAttained
INFO: Completed initialization
Jul 06, 2019 6:11:28 PM hudson.TcpSlaveAgentListener <init>
INFO: JNLP slave agent listener started on TCP port 55609
Jul 06, 2019 6:12:31 PM hudson.model.DownloadService$Downloadable doPostBack
INFO: Obtained the updated data file for hudson.tasks.Ant.AntInstaller
Jul 06, 2019 6:12:31 PM hudson.model.DownloadService$Downloadable doPostBack
INFO: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Jul 06, 2019 6:12:40 PM hudson.model.DownloadService$Downloadable doPostBack
INFO: Obtained the updated data file for hudson.tools.JDKInstaller
Jul 06, 2019 6:12:55 PM hudson.model.UpdateSite doPostBack
INFO: Obtained the latest update center data file for UpdateSource default
```

Accessing Jenkins

Now you can access the Jenkins. Open your browser and type the following url on your browser:

1. <http://localhost:8080>

This url will bring up the Jenkins dashboard.

The screenshot shows the Jenkins dashboard at localhost:8080. The main content area features a cartoon character of a waiter. A message says "Welcome to Jenkins! Please [create new jobs](#) to get started." To the left is a sidebar with links: "New Job", "Manage Jenkins", "People", and "Build History". Below these are sections for "Build Queue" (No builds in the queue) and "Build Executor Status" (two executors listed as Idle). The top right has a search bar, an "ENABLE AUTO REFRESH" link, and a "add description" button.

Page generated: Jul 6, 2019 6:12:28 PM [Jenkins ver. 1.409.1](#)

Java and Tomcat Setup for Jenkins

Java Setup

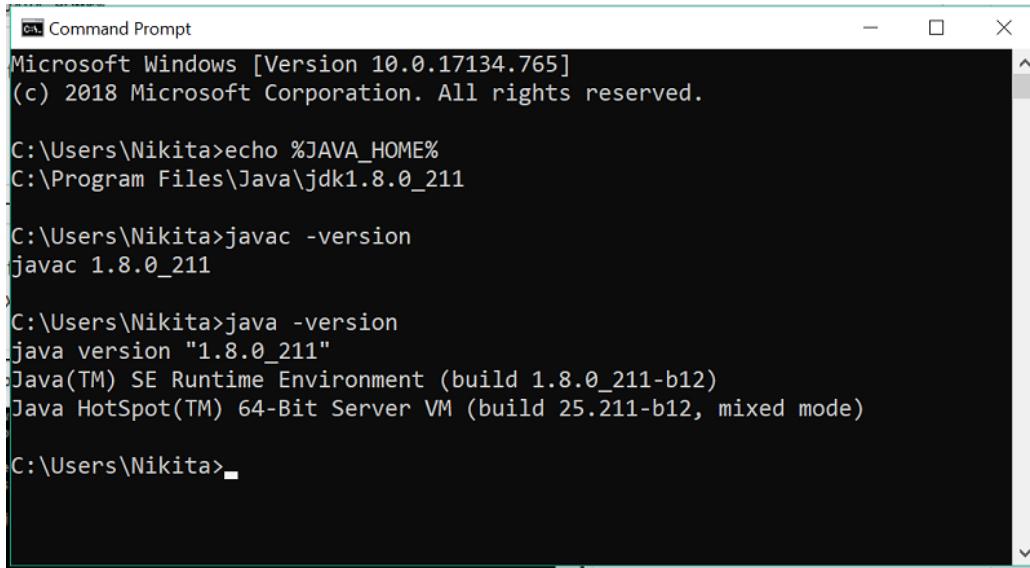
Since Jenkins is written in Java. Therefore, Java must be installed on your system. To download and install the Java, go to our previous chapter of this tutorial.

Now, to set up the Java_Home environment variable on Windows [**click here**](#).

Once the java has been installed properly on your system and Java environment variable has been set, then you can verify it by using the following commands:

1. C:\ echo %JAVA_HOME%
- 2.
3. C:\ javac -version
- 4.
5. C:\ java -version

The following output should come:



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text output:

```
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nikita>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.8.0_211

C:\Users\Nikita>javac -version
javac 1.8.0_211

C:\Users\Nikita>java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)

C:\Users\Nikita>
```

Tomcat Setup

Download Tomcat:

The official website to download the tomcat is <http://tomcat.apache.org/>.

When you click the given link, you will get the home page of the official tomcat website as given below:

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the [Java Community Process](#).

The Apache Tomcat software is developed in an open and participatory environment and released under the [Apache License version 2](#). The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](#).

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [PoweredBy](#) wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 9.0.22 Released 2019-07-09

The Apache Tomcat Project is proud to announce the release of version 9.0.22 of Apache Tomcat. The notable changes compared to 9.0.21 include:

- Add user buildable optional modules for easier CDI 2 and JAX-RS support. Also include a new documentation page describing how to use it.
- Update to Tomcat Native 1.2.23 including Windows binaries built with OpenSSL 1.1.1c.

Go to the link <https://tomcat.apache.org/download-90.cgi> to download the latest version of tomcat.

The screenshot shows a web browser window with the title "Apache Tomcat® - Apache Tomc". The address bar contains the URL <https://tomcat.apache.org/download-90.cgi>. The main content area is titled "9.0.22" and contains a message: "Please see the [README](#) file for packaging information. It explains what every distribution contains." Below this is a section titled "Binary Distributions" which lists various download links:

- Core:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows zip \(pgp, sha512\)](#)
 - [64-bit Windows zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

Below this is another section titled "Source Code Distributions" with the following links:

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

The left sidebar contains the following navigation links:

- Documentation
 - Tomcat 9.0
 - Tomcat 8.5
 - Tomcat 7.0
 - Tomcat Connectors
 - Tomcat Native
 - Wiki
 - Migration Guide
 - Presentations
- Problems?
- Get Involved
 - Overview
 - Source code
 - Buildbot
 - Tools
- Media
 - Twitter
 - YouTube
 - Blog
- Misc
 - Who We Are
 - Swag

Go to the "Binary Distributions" section and download the file according to your platform. Here I am using 64-bit Windows zip for my 64-bit Windows operating system.

Unzip the contents of the downloaded zip file.

Jenkins and Tomcat Setup

Copy the downloaded Jenkins.war file (downloaded from the previous section) and copy it to the **webapps** folder of the tomcat directory.

	Name	Date modified	Type
cess	docs	7/7/2019 9:47 PM	File folder
?	examples	7/7/2019 9:48 PM	File folder
	host-manager	7/7/2019 9:48 PM	File folder
	manager	7/7/2019 9:48 PM	File folder
	ROOT	7/7/2019 9:48 PM	File folder
	jenkins.war	7/1/2019 3:43 PM	WAR File

Now open the command prompt. From the command prompt, browse to the directory where the tomcat is located. Then, go to the bin directory of this tomcat folder and run the **startup.bat** file.

1. C:\apache-tomcat-9.0.21\bin>startup.bat

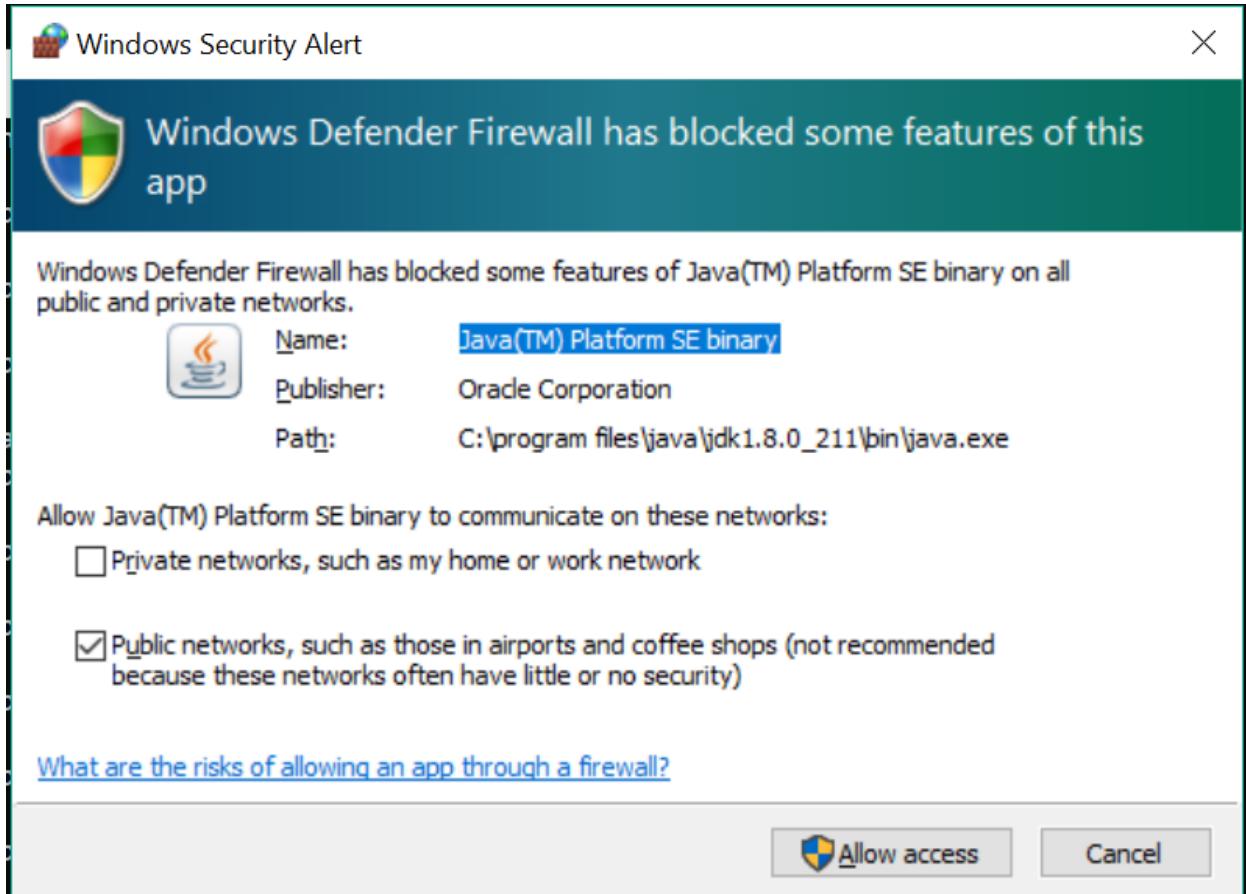
When you run the above command then following output will come:

```
Command Prompt
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nikita>cd C:\apache-tomcat-9.0.21\bin

C:\apache-tomcat-9.0.21\bin>startup.bat
Using CATALINA_BASE:   "C:\apache-tomcat-9.0.21"
Using CATALINA_HOME:  "C:\apache-tomcat-9.0.21"
Using CATALINA_TMPDIR: "C:\apache-tomcat-9.0.21\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk1.8.0_211"
Using CLASSPATH:       "C:\apache-tomcat-9.0.21\bin\bootstrap.jar;C:\apache-tomcat-9.0.21\bin\tomcat-juli.jar"
C:\apache-tomcat-9.0.21\bin>
```

And two more windows will open. i.e., "Tomcat" window and "Windows Security Alert" window. Click on **Allow access** button in Windows Security Alert Window.



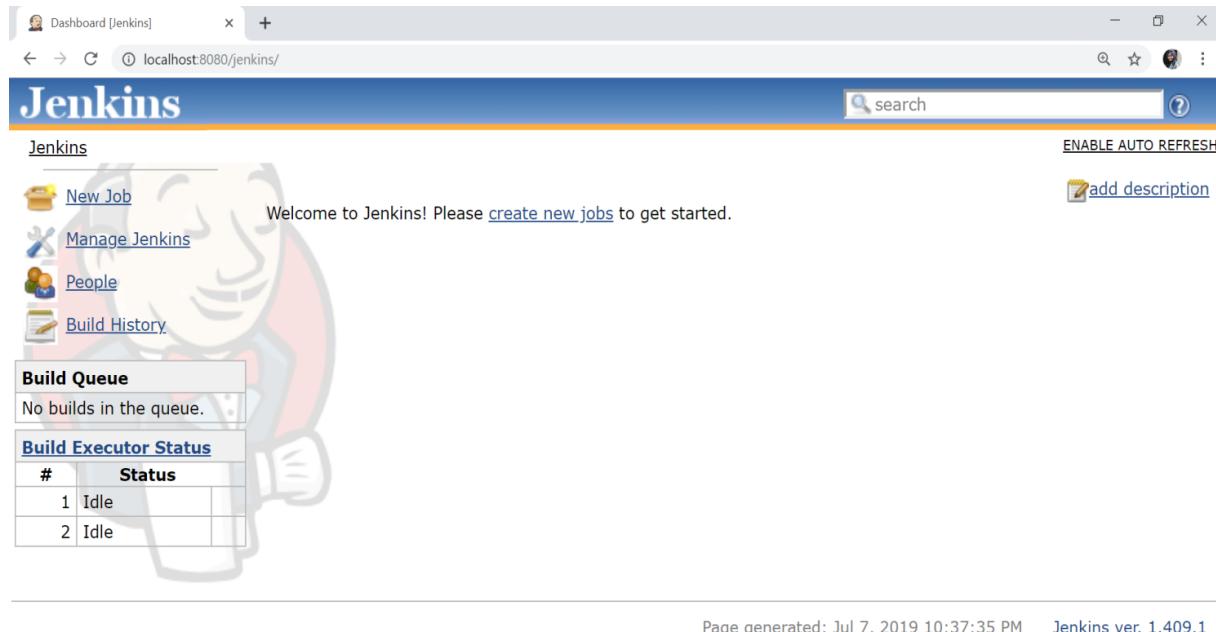
A screenshot of a terminal window titled "Tomcat". The window displays a log of deployment activities for the Jenkins application. The log entries are as follows:

```
07-Jul-2019 22:24:25.636 INFO [main] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [C:\apache-tomcat-9.0.21\webapps\jenkins.war] has finished in [42,402] ms
07-Jul-2019 22:24:25.639 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.21\webapps\docs]
07-Jul-2019 22:24:25.698 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\apache-tomcat-9.0.21\webapps\docs] has finished in [59] ms
07-Jul-2019 22:24:25.702 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.21\webapps\examples]
07-Jul-2019 22:24:26.074 INFO [pool-2-thread-2] hudson.model.Hudson$onAttained Started Initialization
07-Jul-2019 22:24:26.723 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\apache-tomcat-9.0.21\webapps\examples] has finished in [1,021] ms
07-Jul-2019 22:24:26.728 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.21\webapps\host-manager]
07-Jul-2019 22:24:26.803 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\apache-tomcat-9.0.21\webapps\host-manager] has finished in [75] ms
07-Jul-2019 22:24:26.805 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.21\webapps\manager]
07-Jul-2019 22:24:26.868 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\apache-tomcat-9.0.21\webapps\manager] has finished in [61] ms
07-Jul-2019 22:24:26.871 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.21\webapps\ROOT]
07-Jul-2019 22:24:26.915 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\apache-tomcat-9.0.21\webapps\ROOT] has finished in [45] ms
07-Jul-2019 22:24:26.926 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
07-Jul-2019 22:24:26.946 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
07-Jul-2019 22:24:26.951 INFO [main] org.apache.catalina.startup.Catalina$start Server startup in [45,817] milliseconds
07-Jul-2019 22:24:55.214 INFO [pool-2-thread-6] hudson.model.Hudson$onAttained Listed all plugins
07-Jul-2019 22:24:55.345 INFO [pool-2-thread-4] hudson.model.Hudson$onAttained Prepared all plugins
07-Jul-2019 22:24:55.379 INFO [pool-2-thread-3] hudson.model.Hudson$onAttained Started all plugins
07-Jul-2019 22:24:55.381 INFO [pool-2-thread-1] hudson.model.Hudson$onAttained Augmented all extensions
07-Jul-2019 22:24:55.385 INFO [pool-2-thread-2] hudson.model.Hudson$onAttained Loaded all jobs
07-Jul-2019 22:24:57.152 INFO [pool-2-thread-1] hudson.model.Hudson$onAttained Completed Initialization
07-Jul-2019 22:24:57.165 INFO [hudson initialization thread] hudson.TcpSlaveAgentListener.<init> JNLP slave agent listener started on TCP port 61779
```

Once the processing is completed, the following line will come in the output of the Tomcat Window:

1. INFO [hudson initialization thread] hudson.TcpSlaveAgentListener.<init> JNL
P slave agent listener started on TCP port **61779**

Now, open the browser and go the link: <http://localhost:8080/jenkins>. The following page will open:



The setup of Jenkins with Tomcat is successfully done.

GitHub Setup for Jenkins

Jenkins is a CI (Continuous Integration) server and this means that it needs to check out source code from a source code repository and build code. Jenkins has outstanding support for various source code management systems like Subversion, CVS etc.

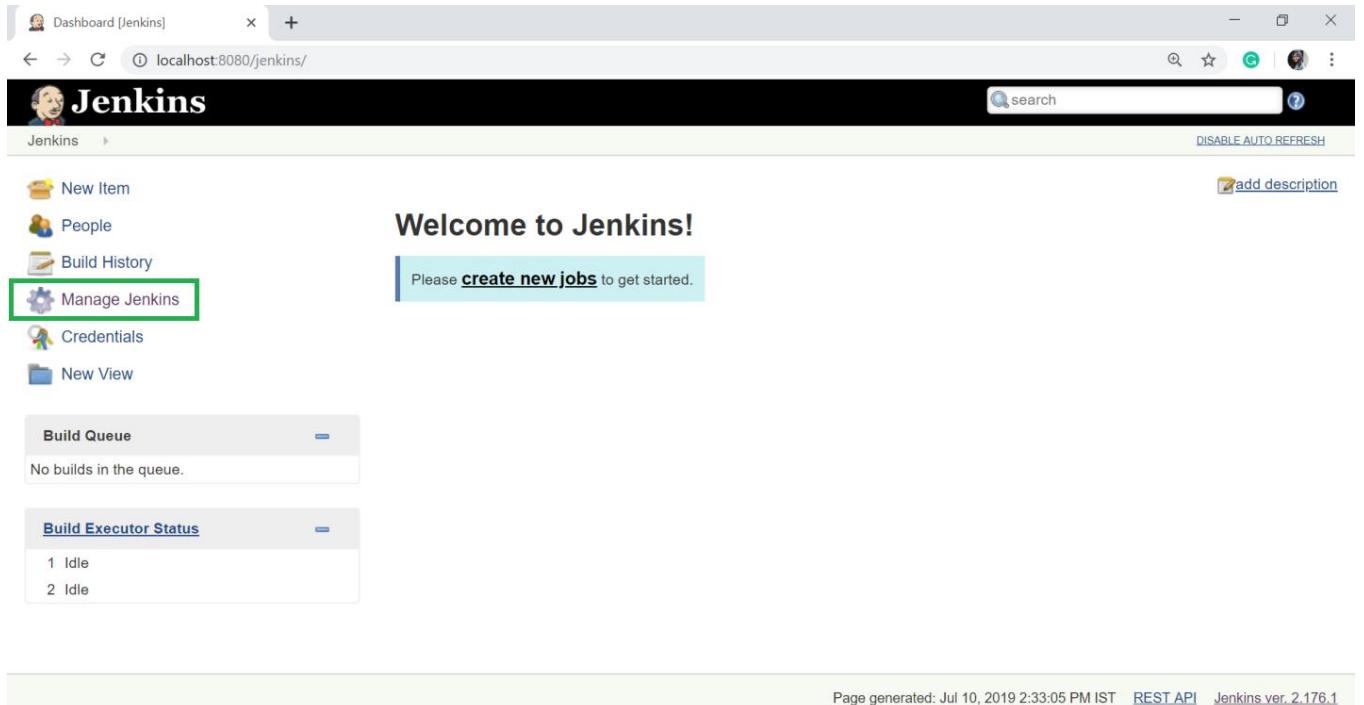
Github is the fast becoming one of the most popular source code management systems. It is a web based repository of code which plays a major role in DevOps. GitHub provides a common platform for many developers working on

the same code or project to upload and retrieve updated code, thereby facilitating continuous integration. Jenkins works with Git through the Git plugin.

Connecting a GitHub private repository to a private instance of Jenkins can be tricky.

To do the GitHub setup, make sure that internet connectivity is present in the machine where Jenkins is installed.

- In the Home screen of the Jenkins (Jenkins Dashboard), click on the **Manage Jenkins** option on the left hand side of the screen.



- Now, click on the **Manage Plugins** option.

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). On the right, several management options are listed: 'Configure Credentials' (icon: key), 'Global Tool Configuration' (icon: wrench), 'Reload Configuration from Disk' (icon: circular arrow), 'Manage Plugins' (icon: puzzle piece, highlighted with a green box), 'System Information' (icon: computer monitor), 'System Log' (icon: clipboard), and 'Load Statistics' (icon: bar chart). At the bottom left, it says 'localhost:8080/jenkins/pluginManager'. A horizontal scrollbar is visible on the right side.

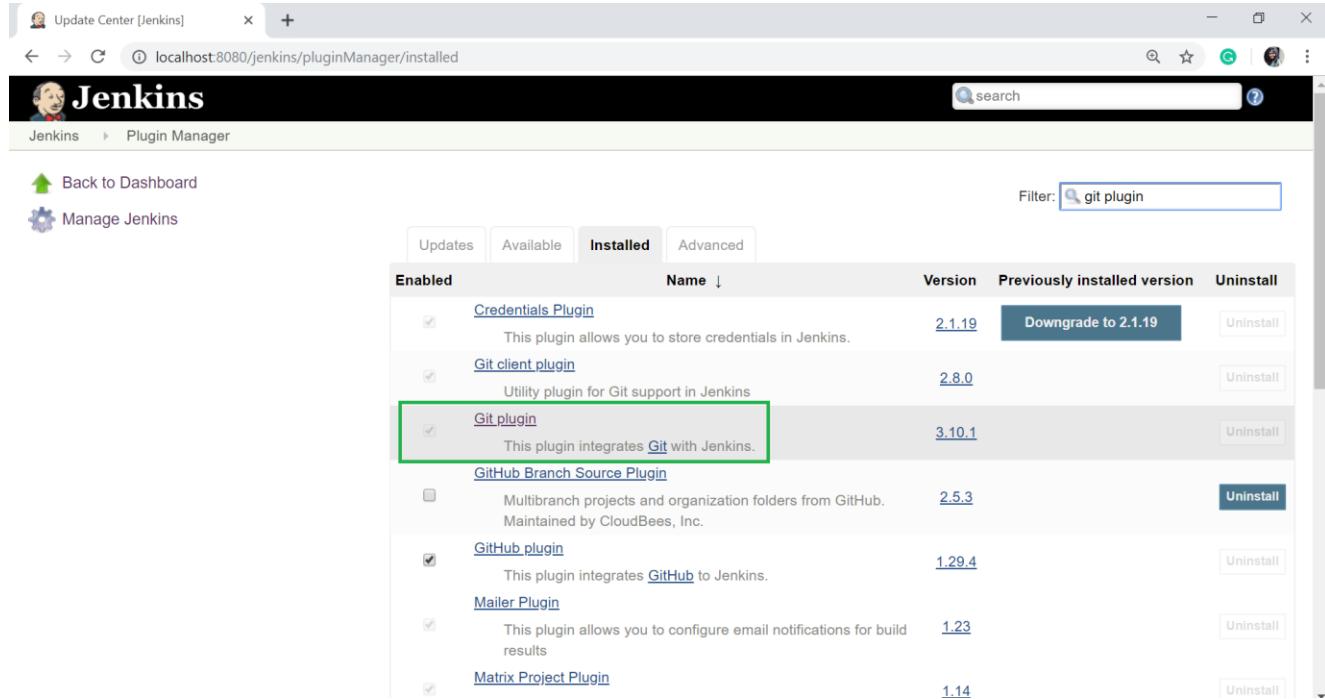
- In the next page, click on the "Available tab".

The screenshot shows the Jenkins Plugin Manager page. At the top, there's a navigation bar with 'Jenkins' and 'Plugin Manager'. Below it is a toolbar with 'Back to Dashboard' and 'Manage Jenkins'. The main area has tabs: 'Updates' (selected), 'Available' (highlighted with a green box), 'Installed', and 'Advanced'. Under 'Available', there's a table with columns: 'Install', 'Name' (sorted by 'Name'), 'Version', and 'Installed'. The table shows 'No updates'. At the bottom, it says 'Update information obtained: 16 hr ago' and has a 'Check now' button. A note below says 'Select: All, Compatible, None' and 'This page lists updates to the plugins you currently use.'

localhost:8080/jenkins/pluginManager/available Page generated: Jul 10, 2019 2:55:11 PM IST REST API Jenkins ver. 2.176.1

- The "Available" tab gives a list of plugins which are available for downloading. In the Filter tab type, type the "Git Plugin".
- Select the Git Plugin.
- Click on the "**install without restart**". The plugin will take some time to finish downloading depending on your internet connection, and will be installed automatically.
- You can also click on "**Download now and install after restart**" button in which the git plugin is installed after restart.

- If you already have the Git plugin installed then go to "Installed" tab and in filter option type Git plugin.



The screenshot shows the Jenkins Plugin Manager interface. The title bar says "Update Center [Jenkins]". Below it, the address bar shows "localhost:8080/jenkins/pluginManager/installed". The main header has tabs for "Updates", "Available", "Installed" (which is selected), and "Advanced". A search bar at the top right contains the text "git plugin". The table below lists several plugins:

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Credentials Plugin This plugin allows you to store credentials in Jenkins.	2.1.19	Downgrade to 2.1.19	Uninstall
<input checked="" type="checkbox"/>	Git client plugin Utility plugin for Git support in Jenkins	2.8.0	Uninstall	Uninstall
<input checked="" type="checkbox"/>	Git plugin This plugin integrates Git with Jenkins.	3.10.1	Uninstall	Uninstall
<input type="checkbox"/>	GitHub Branch Source Plugin Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.5.3	Uninstall	Uninstall
<input checked="" type="checkbox"/>	GitHub plugin This plugin integrates GitHub to Jenkins.	1.29.4	Uninstall	Uninstall
<input checked="" type="checkbox"/>	Mailer Plugin This plugin allows you to configure email notifications for build results	1.23	Uninstall	Uninstall
<input type="checkbox"/>	Matrix Project Plugin	1.14	Uninstall	Uninstall

- Once all the installations are completed, restart Jenkins by giving the command `java -jar jenkins.war` in the browser. <http://localhost:8080/jenkins/restart>

After Jenkins is restarted, Git will available as an option while configuring jobs.

Integrating Jenkins with GitHub

Let's see the process of integrating GitHub into Jenkins in a windows system.

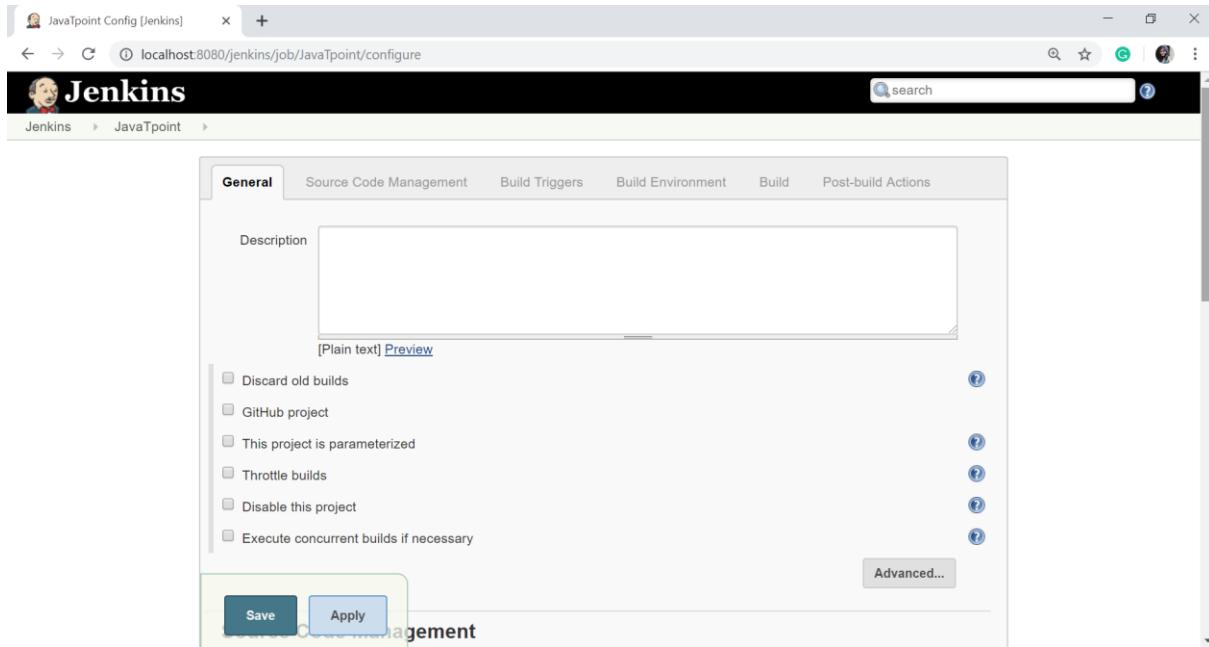
- First create a new job in Jenkins, open the Jenkins Dashboard and click on "create new jobs".

The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. The main header says "Jenkins". On the left, there's a sidebar with links: "New Item", "People", "Build History", "Manage Jenkins", "Credentials", and "New View". Below the sidebar are two collapsed sections: "Build Queue" (No builds in the queue) and "Build Executor Status" (1 Idle, 2 Idle). The central area has a green-bordered box containing the text "Please [create new jobs](#) to get started." At the bottom right, it says "Page generated: Jul 10, 2019 9:48:48 PM IST REST API Jenkins ver. 2.176.1".

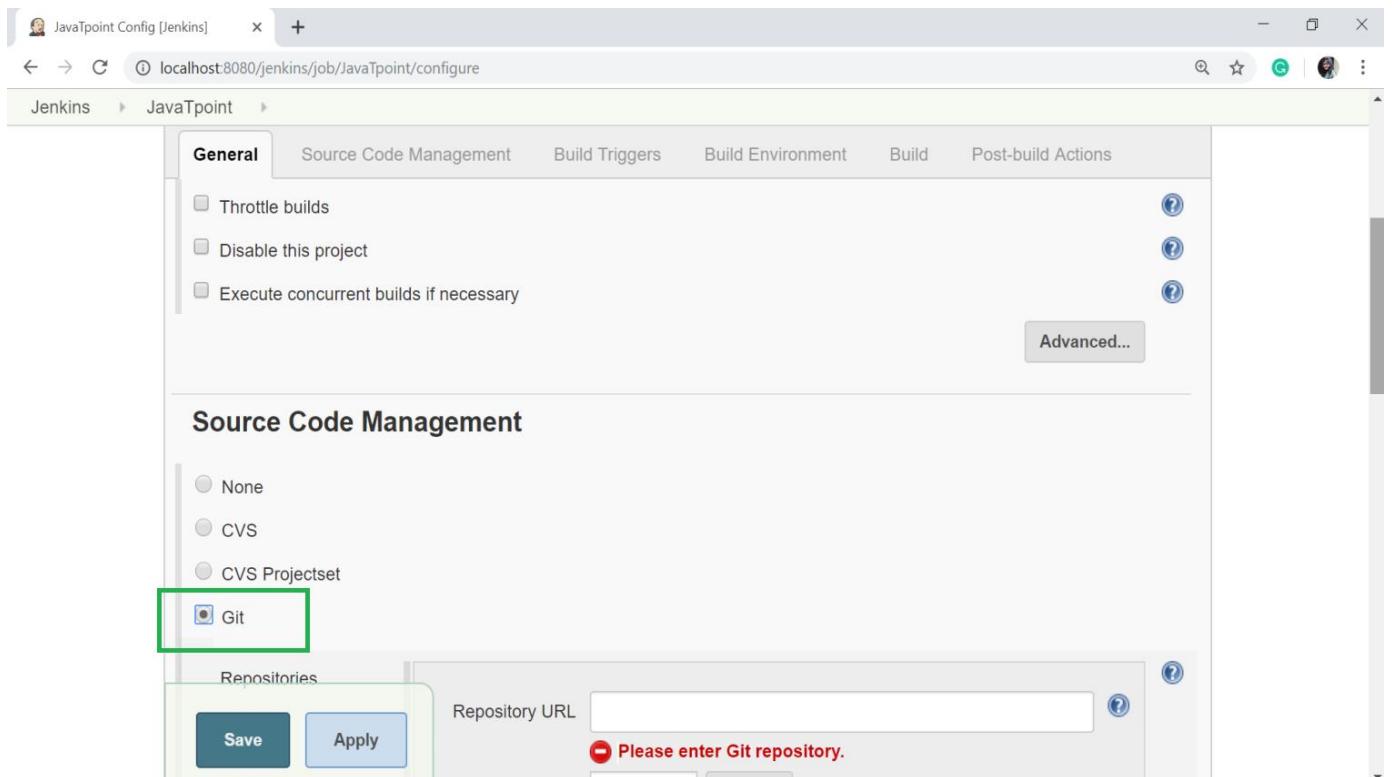
- Now enter the item name and select the job type. For example, item name is **javaTpoint**" and job type is "**Freestyle project**".
- Click on **OK**.

The screenshot shows the "Enter an item name" dialog at localhost:8080/jenkins/newJob. The input field contains "JavaTpoint". A green-bordered box highlights the "Freestyle project" option, which is described as the central feature of Jenkins, combining any SCM with any build system. Other options shown are "Maven project" and "Pipeline". At the bottom, a green-bordered box highlights the "Multi-configuration project" option, which is suitable for projects with many configurations. An "OK" button is visible at the bottom left.

- Once you click OK, the page will be redirected to its project configuration. Enter the project information:

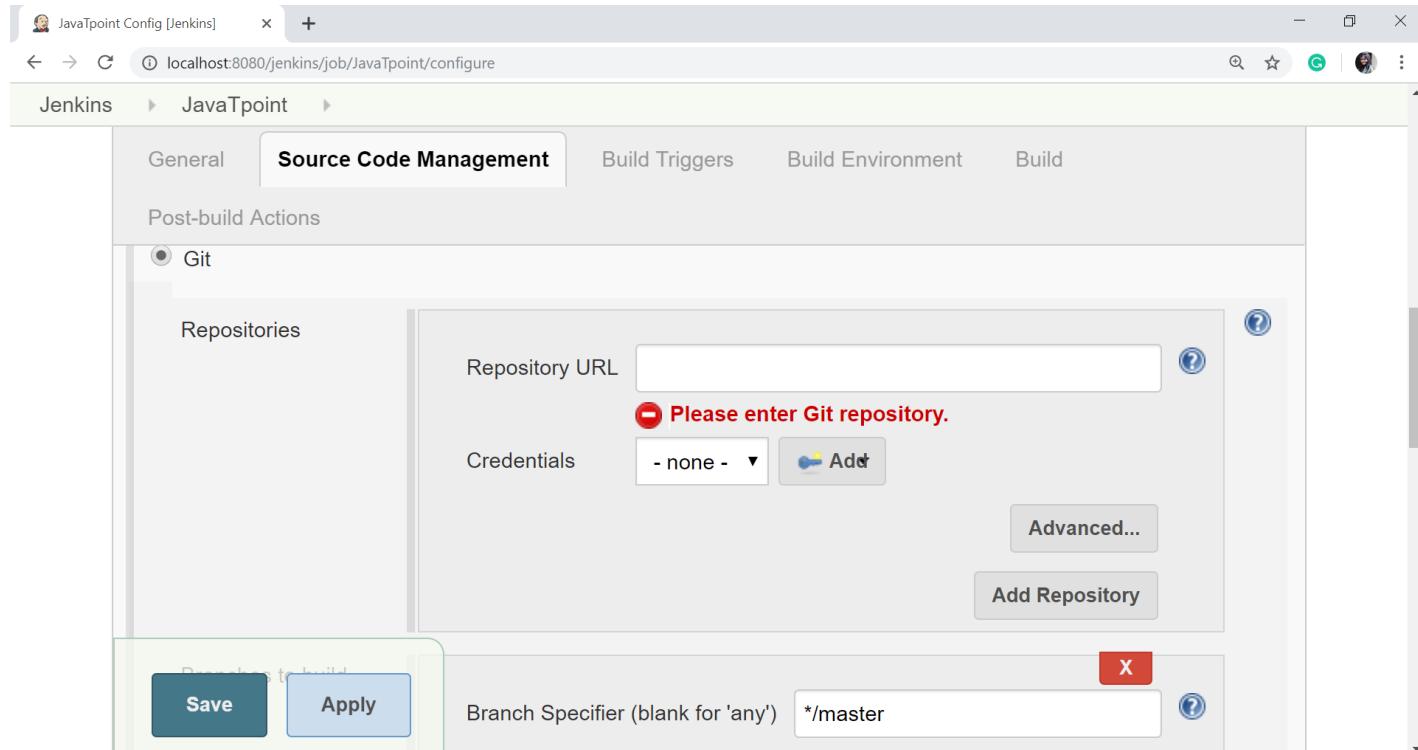


- Now, under the "Source Code Management" you will see the Git option, if your **Git** plugin has been installed in Jenkins:

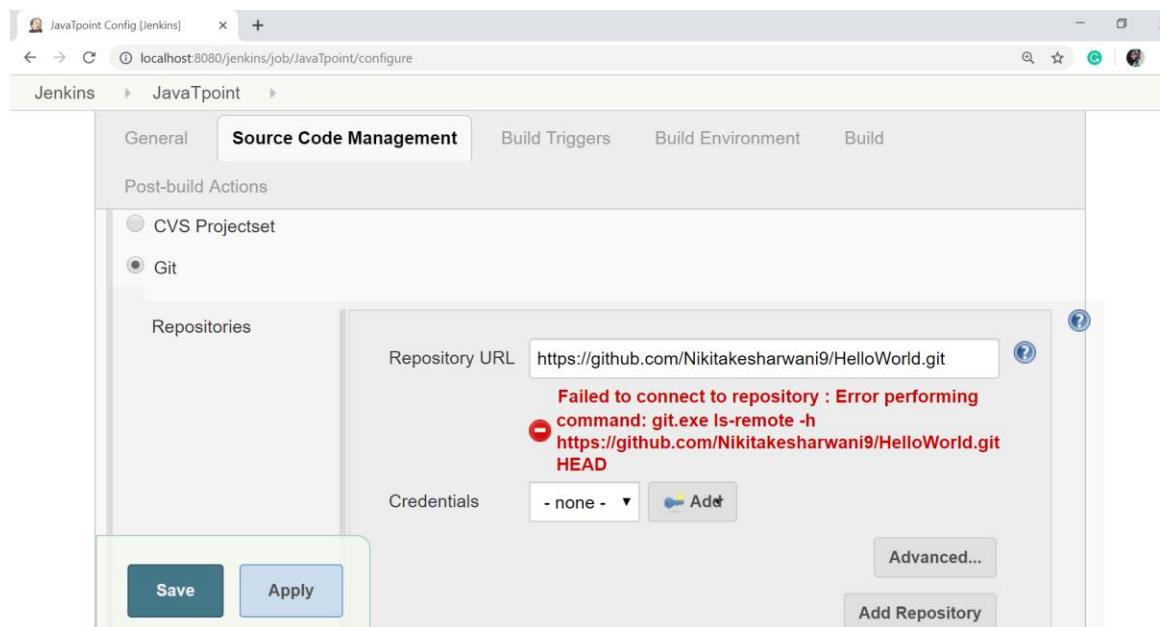


Note: if the **Git** option does not appear, try to reinstall the plugins, followed by a restart into your Jenkins dashboard.

- Enter the Git repository URL on the "Repository URL" option to pull the code from GitHub.



- You might get an error when first time you enter the repository URL. For example:



This happens if you don't have Git installed in your system. To install the Git in your system, download the appropriate Git setup according to your operating system. I am installing for windows. Once the download is completed, install the Git.

The screenshot shows the 'Downloads' section of the official Git website. On the left sidebar, there are links for 'About', 'Documentation', 'Downloads' (which is highlighted in red), and 'Community'. Below the sidebar, a note mentions the availability of the 'Pro Git book' on GitHub and Amazon. The main content area features a large 'Downloads' heading. Underneath it, three download links are shown: 'Mac OS X' (with an Apple logo), 'Windows' (with a Windows logo), and 'Linux/Unix' (with a terminal icon). To the right, a graphic of a computer monitor displays a teal screen with the text 'Latest source Release 2.22.0' and a link 'Download 2.22.0 for Windows'. Below the monitor, there's a 'Logos' section with a link 'View Logos →' and a note about various Git logos in different formats available for download.

Complete the following instructions to install the Git:



Git 2.22.0 Setup



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>

Next >

Cancel

You can execute Git repositories in your Jenkins once Git has been installed on your system. To check if the Git has been installed on your system, open the command prompt, type Git and press Enter.

```
Command Prompt
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nikita>git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset     Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  branch   List, create, or delete branches
  checkout Switch branches or restore working tree files
```

In the above screen, you observe that syntax and different options come up for Git. This means that Git has been installed in your machine.

- Now try to add the Git URL into Jenkins.
 - Git is now successfully configured on your system.
-

Maven Setup

Maven is a powerful project management and comprehension tool that provides complete build life cycle framework to assist developers. It is based on the concept of a POM (Project Object Model) that includes project

information and configuration information for Maven such as construction directory, source directory, test source directory, dependency, Goals, plugins etc.

Maven is build automation tool used basically for Java projects, though it can also be used to build and manage projects written in C#, Scala, Ruby, and other languages. Maven addresses two aspects of building software: 1st it describes how software is build and 2nd it describes its dependencies.

Downloading Maven

The official website for Apache Maven is <https://maven.apache.org/download.cgi>. Click on the given link to download the Maven. When you click on the given link, you will get the home page of the official Maven website as given below:

Go to the files section and download the Maven by the given link for Binary zip archive file.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.1-bin.tar.gz	apache-maven-3.6.1-bin.tar.gz.sha512	apache-maven-3.6.1-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.1-bin.zip	apache-maven-3.6.1-bin.zip.sha512	apache-maven-3.6.1-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.1-src.tar.gz	apache-maven-3.6.1-src.tar.gz.sha512	apache-maven-3.6.1-src.tar.gz.asc
Source zip archive	apache-maven-3.6.1-src.zip	apache-maven-3.6.1-src.zip.sha512	apache-maven-3.6.1-src.zip.asc

Once the file is downloaded, extract the file into your system.

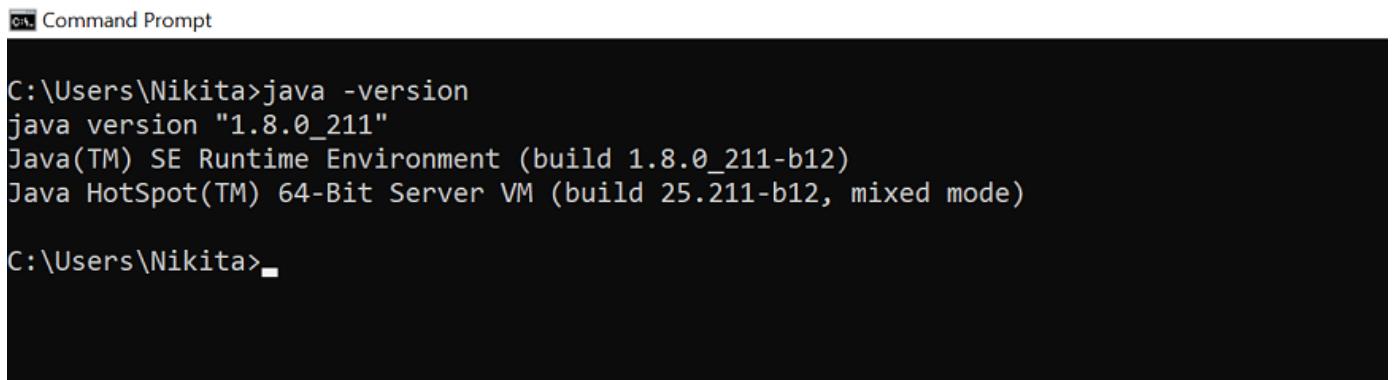
Setting Up Java and Maven in Jenkins

- First of all, you have to set the JAVA_HOME and MAVEN_HOME environment variable in your system.

To set the JAVA_HOME and MAVEN_HOME path, [click here](#).

You can verify that the JAVA_HOME environment variable is properly configured or not by using the following command:

1. C:\java -version

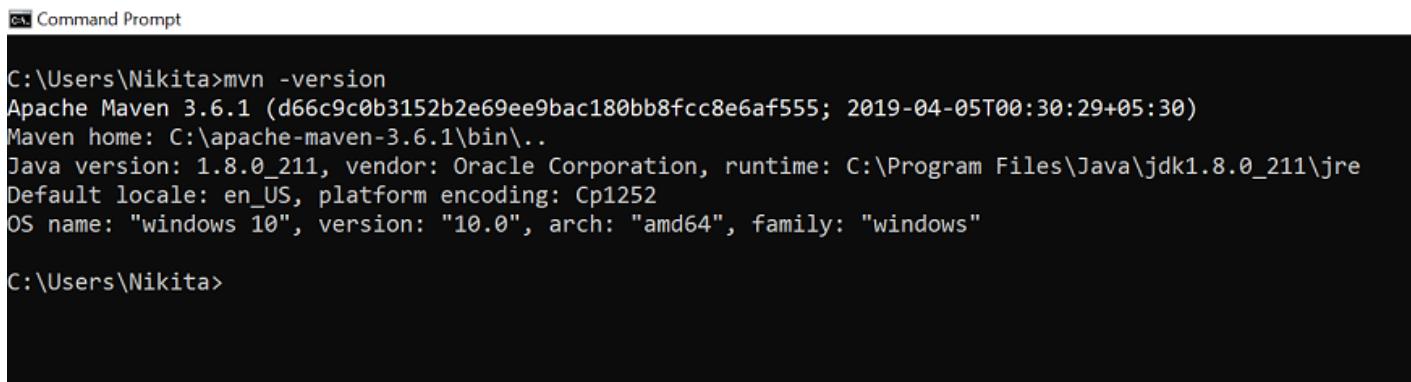


```
cmd Command Prompt
C:\Users\Nikita>java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)

C:\Users\Nikita>
```

Similarly, you can verify that the JAVA_HOME environment variable is properly configured or not by using the following command:

1. C:\ mvn -version



```
cmd Command Prompt
C:\Users\Nikita>mvn -version
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8fcc8e6af555; 2019-04-05T00:30:29+05:30)
Maven home: C:\apache-maven-3.6.1\bin\..
Java version: 1.8.0_211, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_211\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Nikita>
```

- Now, in the Jenkins dashboard (Home screen) click on manage Jenkins from the left-hand side menu.

The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. The main header features the Jenkins logo and a search bar. On the left, there's a sidebar with links: New Item, People, Build History, Manage Jenkins, Credentials, and New View. The central area has a large "Welcome to Jenkins!" message and a call-to-action button: "Please [create new jobs](#) to get started.". Below this are two collapsed sections: "Build Queue" (No builds in the queue) and "Build Executor Status" (1 Idle, 2 Idle). At the bottom right, it says "Page generated: Jul 12, 2019 10:36:30 PM IST REST API Jenkins ver. 2.176.1".

Click on "Global Tool Configuration" option.

To configure Java, click on "**Add JDK**" button in the JDK section.

The screenshot shows the Jenkins Global Tool Configuration page. At the top, there are sections for Maven Configuration, JDK, and Git. Under Maven Configuration, 'Default settings provider' is set to 'Use default maven settings'. Under JDK, 'JDK installations' has an 'Add JDK' button. Under Git, 'Git installations' has an 'Add Git' button. At the bottom are 'Save' and 'Apply' buttons.

Give a **Name** and **JAVA_HOME** path, or check on **install automatically** checkbox.

The screenshot shows the 'JDK' section of the Global Tool Configuration page. A new JDK installation is being added, named 'jdk1.8' with the 'JAVA_HOME' path set to 'C:\Program Files\Java\jdk1.8.0_211'. The 'Install automatically' checkbox is unchecked. There is a 'Delete JDK' button and an 'Add JDK' button below the form. Below the form, a list of existing JDK installations is shown.

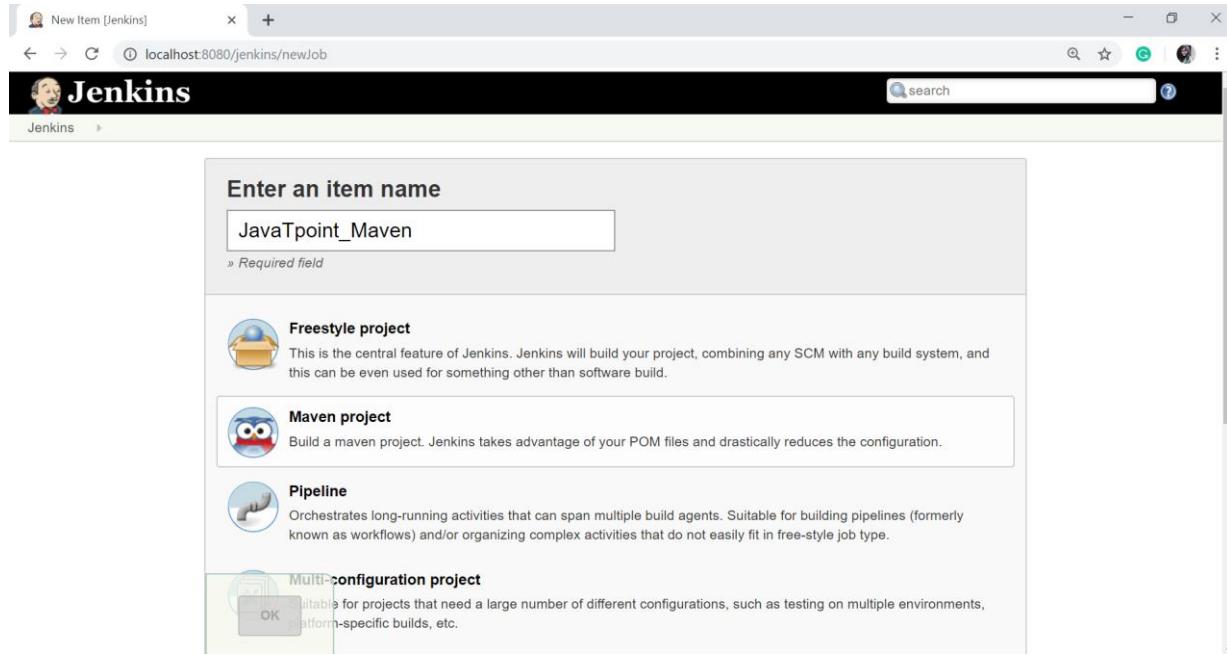
And now, to configure Maven, click on "Add Maven" button in the Maven section, give any **Name** and **MAVEN_HOME** path or check to install automatically checkbox.

The screenshot shows the Jenkins Global Tool Configuration interface. In the top navigation bar, it says 'Global Tool Configuration [Jenkins]'. Below that, the URL is 'localhost:8080/jenkins/configureTools/'. The main content area is titled 'Maven' under 'Maven installations'. A large grey button labeled 'Add Maven' is visible. To its right, there is a section for a new Maven installation. It has a 'Name' field containing 'maven3.6' and a 'MAVEN_HOME' field containing 'C:\apache-maven-3.6.1'. There is also a checked checkbox labeled 'Install automatically' and a red 'Delete Maven' button. Below these fields is another 'Add Maven' button. At the bottom of the page, there are 'Save' and 'Apply' buttons.

Then, click on the "**Save**" button at the end of the screen.

Now, you can create a job with the Maven project. To do that, click on the **New Item** option or **create a new job** option.

Enter the **Item Name** and select the **Maven Project**.



Click OK.

Now configure the job. Give the description and in the **Source Code Management** section, select the required option.

The screenshot shows the Jenkins job configuration page for 'JavaPoint_Maven'. The 'Source Code Management' tab is selected, displaying options like None, CVS, CVS Projectset, Git, and Subversion. The 'Build Triggers' tab is also visible, containing several trigger configurations. At the bottom, there are 'Save' and 'Apply' buttons.

In the Build Triggers section, there are multiple options, select the required one.

Add the pom.xml file's path in the **Root POM** option.

The screenshot shows the Jenkins job configuration page for 'JavaPoint_Maven'. The 'Pre Steps' tab is selected, showing the 'Build' section where the 'Root POM' is set to 'D:\parent\pom.xml' and 'Goals and options' are set to 'clean install'. The 'Post Steps' tab is also visible at the bottom. At the very bottom, there are 'Save' and 'Apply' buttons.

Configure the other fields as per your requirement and then click on the **Save** button.

