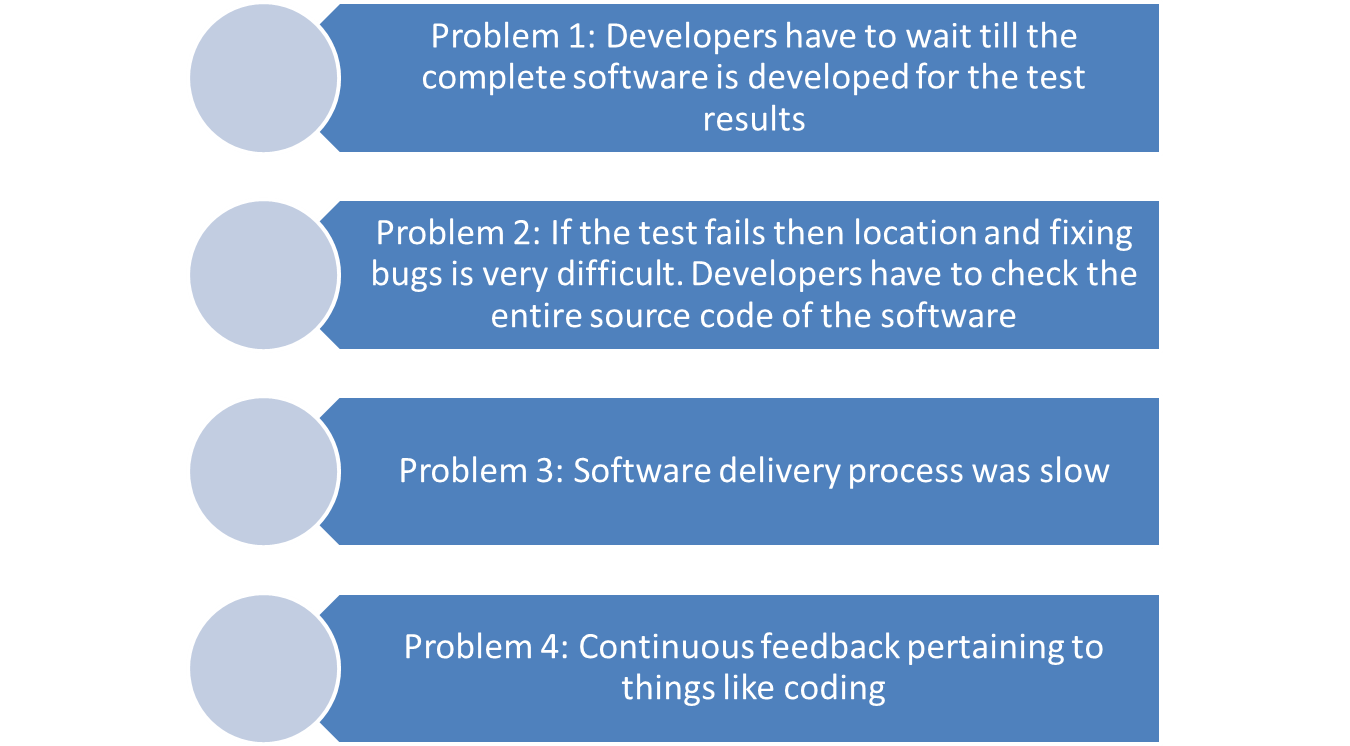
**Jenkins**

* Jenkins is an open source automation server written in Java.
* Jenkins helps to automate the non-human part of software development process, with continuous integration and facilitating technical aspects of continuous delivery
* Jenkins focus on Build and test software project continuously and monitoring externally run jobs
* Software builds can be triggered by various means, for example by commit in a version control system, by scheduling via a cron-like mechanism and by requesting a specific build URL. It can also be triggered after the other builds in the queue have completed. Jenkins functionality can be extended with plugins.

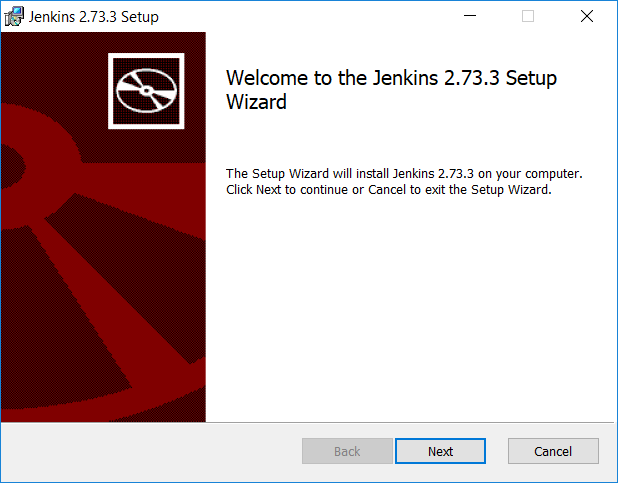
**Why Jenkins?**

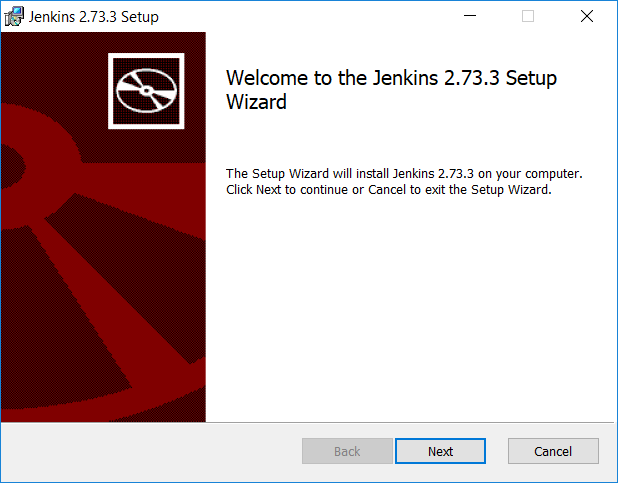


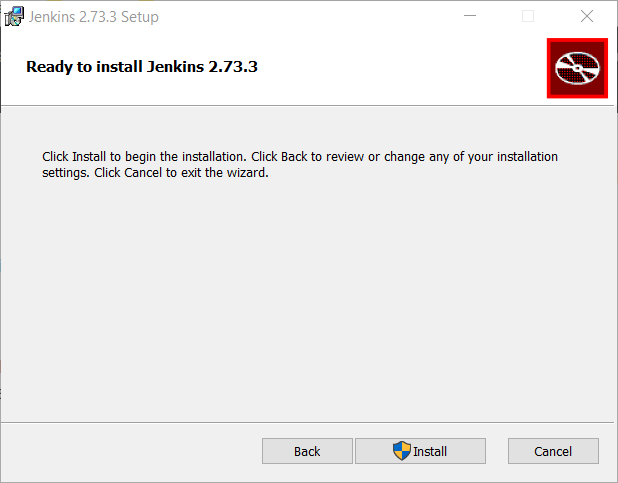
**Benefits of Jenkins**

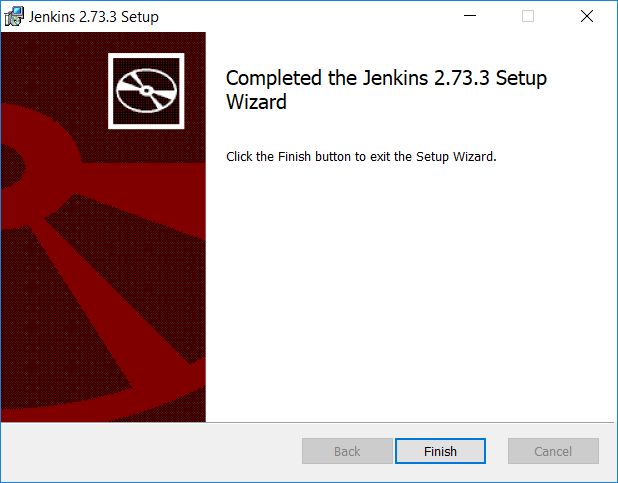
**Installing Jenkins**

1. Go to URL : <https://jenkins.io/download/>
2. Downlaod Jenkins for Windows. We will use Jenkins **Long Term Support(LTS)** release.
3. Unzip the files. Double click on Jenkins installer
4. Follow the instruction as below screens

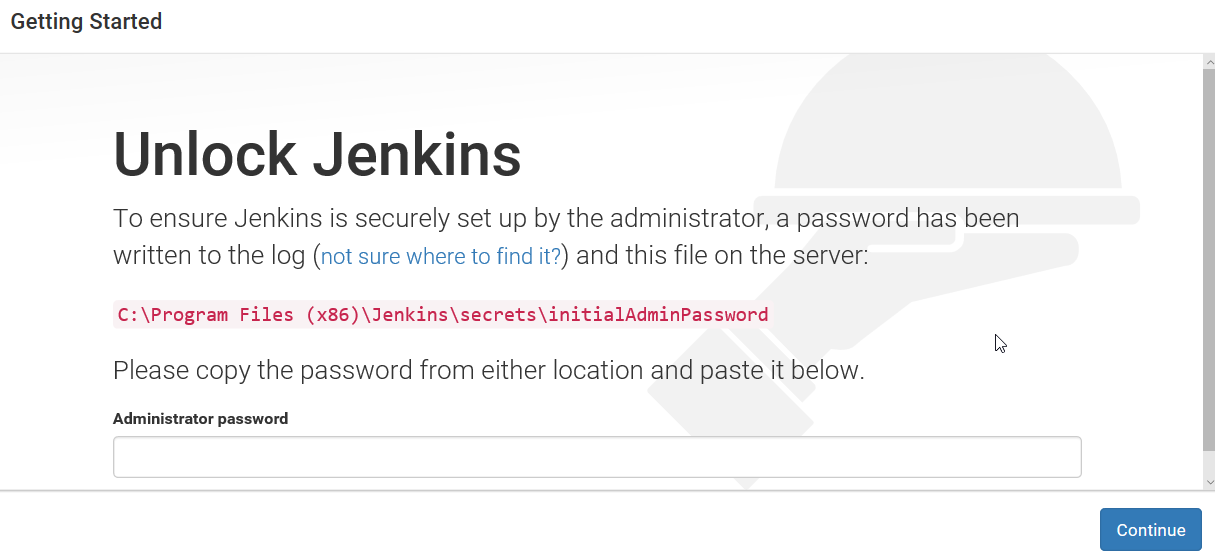








1. After click on ‘Finish’ button, your default browser will auto open with url – <http://localhost:8080/login?from=%2F> displaying information of your Jenkins default administrator password



1. Copy paste the password and click Continue button
2. Click on **install suggested plugins**
3. This will start downloading and installing default suggested plugins
4. Create first Admin user. This will be your first user created administrator account. So make sure you keep your administrator user name and password some where safe so when you login next time you know your administrator username and password.
5. Click on Save and Finish button
6. You have successful completed Jenkins installation!!!
7. Click on start using Jenkins
8. Open chrome browser and type following and hit enter

Localhost:8080

It will open Jenkins window and ask you to enter your username/password!!!

To open Jenkins again, you should open command prompt. Move to folder where Jenkins.war is present then type command java -jar Jenkins.war

Once you get a message Jenkins is up and running, you can go to browser and type Localhost:8080

**First Jenkins configuration**

Jenkins->ManageJenkins->Global Tool Configuration

Scroll to Java -> Add java -> Name as JavaTest-> Uncheck Install Automatically -> Enter java\_home as in your pc.

To install a maven project -

Jenkins->ManageJenkins-> ManagePlugins->Avaialable

Search for maven

Select following maven options –

[Maven Artifact ChoiceListProvider (Nexus)](https://plugins.jenkins.io/maven-artifact-choicelistprovider)

[Maven Metadata Plugin for Jenkins CI server](https://plugins.jenkins.io/maven-metadata-plugin)

[Maven Integration](https://plugins.jenkins.io/maven-plugin)

[Maven Dependency Update Trigger](https://plugins.jenkins.io/maven-dependency-update-trigger)

[Unleash Maven](https://plugins.jenkins.io/unleash)

[Maven Release Plug-in](https://plugins.jenkins.io/m2release)

[Maven Invoker](https://plugins.jenkins.io/maven-invoker-plugin)

[Maven Repository Scheduled Cleanup](https://plugins.jenkins.io/maven-repo-cleaner)

[Maven Info](https://plugins.jenkins.io/maven-info)

Search for testNG ->[TestNG Results](https://plugins.jenkins.io/testng-plugin)

Click install without restart

Click Go back to top page

Now we can start creating a maven project.

1. Jenkins -> New item -> Maven project
2. Give name as ExecutionsJob – ok
3. Enter description
4. Source code management – None
5. Under Build -> Click the tool configuration
6. Scroll to Maven -> Add maven -> Name as MavenTest-> Uncheck Install Automatically -> Enter maven\_home as in your pc.
7. Under build -> Root POM -> Enter the address of your pom file
8. Goals and options -> Clean install
9. Add post-build actions ->Publish testNG Results -> save