6.4 Code Coverage with Clover

This section will guide you to:

* Perform code coverage using Clover in Jenkins for a Maven application.

This guide has four subsections, namely:

6.4.1 Login to Jenkins

6.4.2 Add Clover in Jenkins and Maven project

6.4.3 Create Jenkins job for Maven

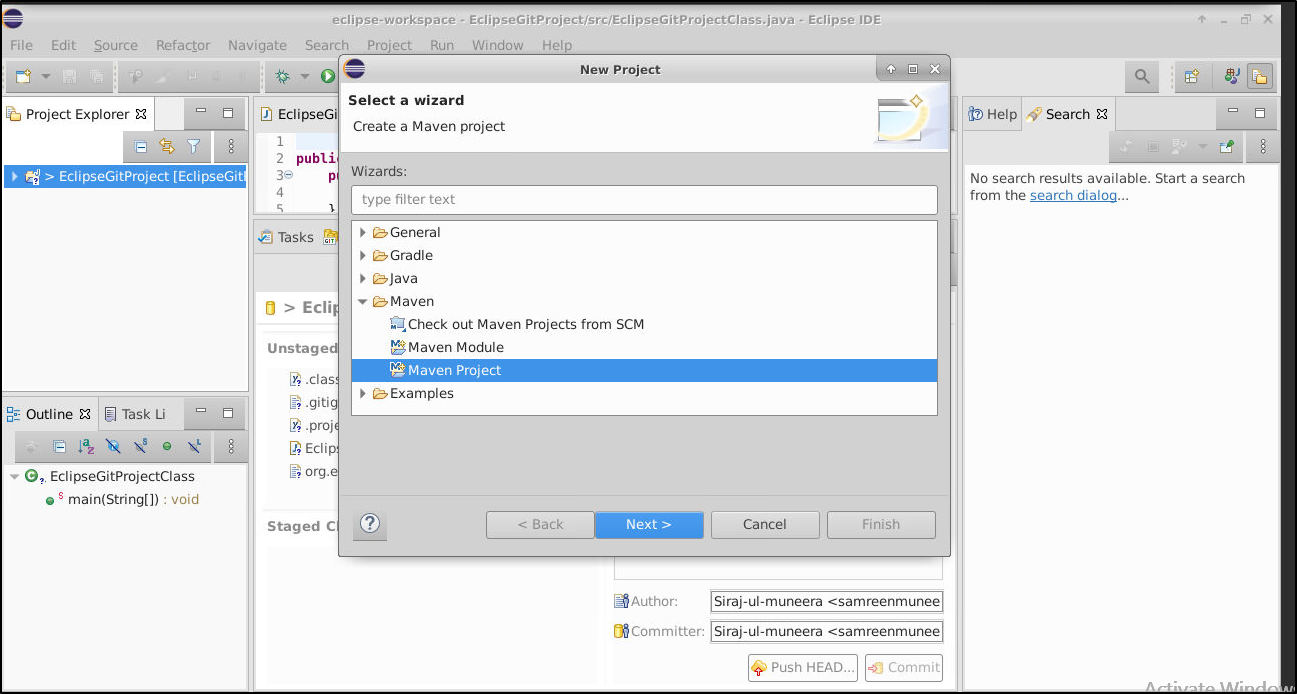
6.4.4 Push code to GitHub repositories

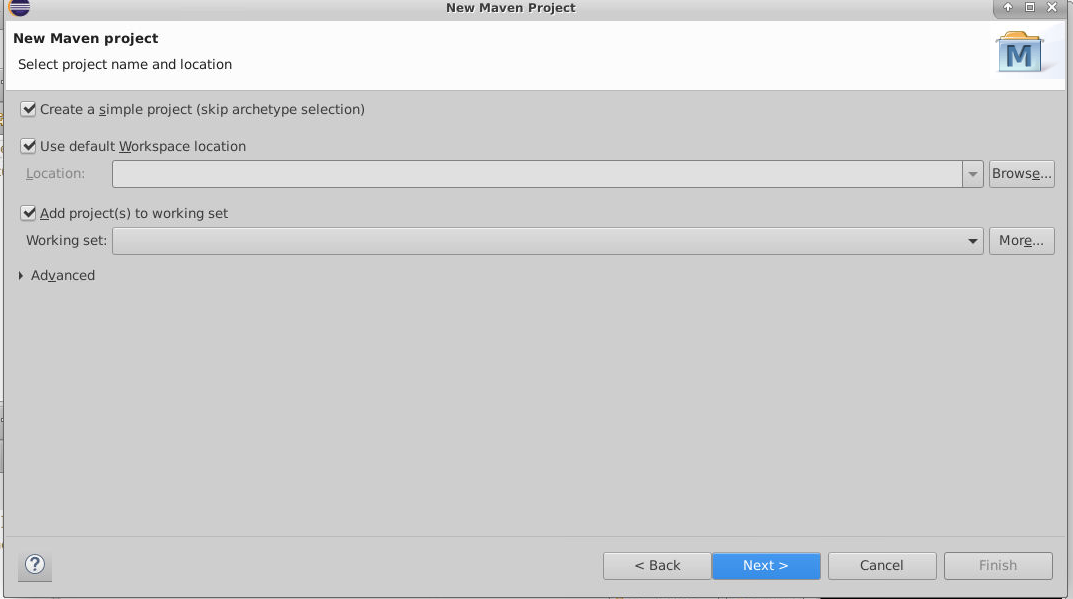
**Step 6.4.1:** Login to Jenkins

* Open your browser and navigate to **localhost:8081**
* Provide your username and password and click on **Login.**

**Step 6.4.2:** Add Clover in Jenkins and Maven project

* Click on the eclipse executable in **/opt/eclipse** or run the following command in the terminal to open eclipse in your lab.  
  ***opt/eclipse/eclipse***
* Create a basic Maven project in Eclipse.  
  Click on **File**->**New** ->**Project**-> **maven**->**maven project**.





* In the New Maven Project Dialog box, enter the following details:  
    
  **group id:** simplilearn

**artifact id:** CloverDemo

* Add the below code in the **pom.xml** file .

**<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">**

**<modelVersion>4.0.0</modelVersion>**

**<groupId>simplilearn</groupId>**

**<artifactId>CloverDemo</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<build>**

**<plugins>**

**<plugin>**

**<groupId>com.atlassian.maven.plugins</groupId>**

**<artifactId>clover-maven-plugin</artifactId>**

**<configuration>**

**<generatePdf>true</generatePdf>**

**<generateXml>true</generateXml>**

**<generateHtml>false</generateHtml>**

**<generateJson>false</generateJson>**

**</configuration>**

**<executions>**

**<execution>**

**<phase>generate-sources</phase>**

**<goals>**

**<goal>instrument</goal>**

**</goals>**

**</execution>**

**</executions>**

**</plugin>**

**</plugins>**

**<pluginManagement>**

**<plugins>**

**<!-- This plugin's configuration is used to store Eclipse m2e settings only. It has no influence on the Maven build itself. -->**

**<plugin>**

**<groupId>org.eclipse.m2e</groupId>**

**<artifactId>lifecycle-mapping</artifactId>**

**<version>1.0.0</version>**

**<configuration>**

**<lifecycleMappingMetadata>**

**<pluginExecutions>**

**<pluginExecution>**

**<pluginExecutionFilter>**

**<groupId> com.atlassian.maven.plugins </groupId>**

**<artifactId> clover-maven-plugin </artifactId>**

**<versionRange>[4.1.2,)</versionRange>**

**<goals>**

**<goal>instrument</goal>**

**</goals>**

**</pluginExecutionFilter>**

**<action>**

**<ignore/>**

**</action>**

**</pluginExecution>**

**</pluginExecutions>**

**</lifecycleMappingMetadata>**

**</configuration>**

**</plugin>**

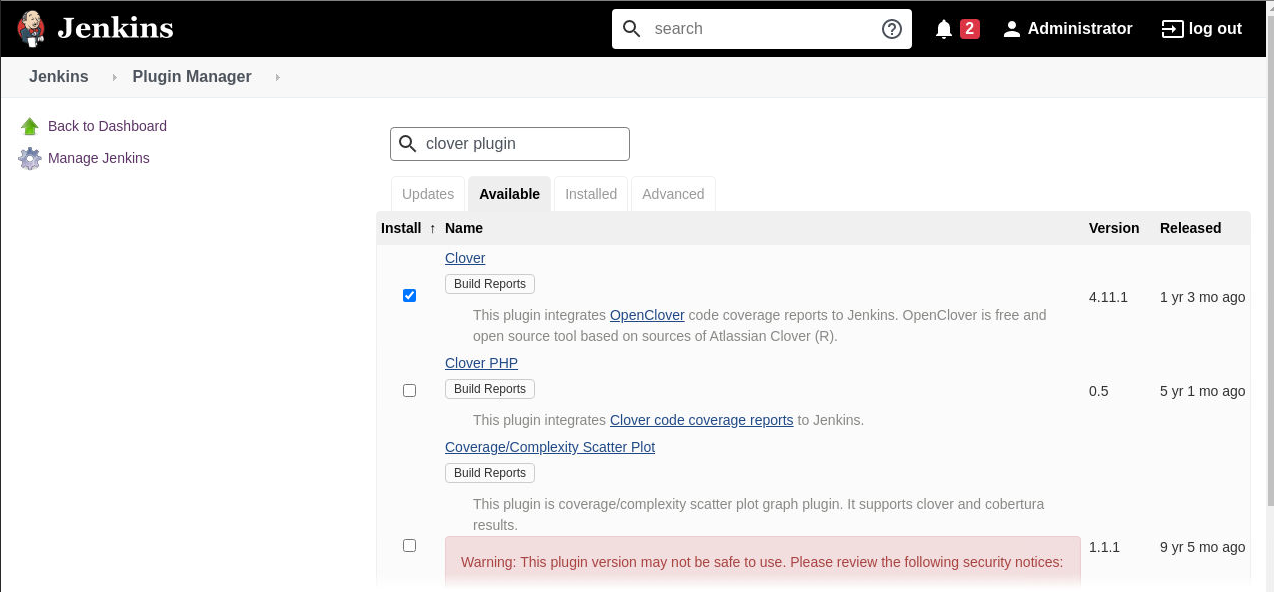
**</plugins>**

**</pluginManagement>**

**</build>**

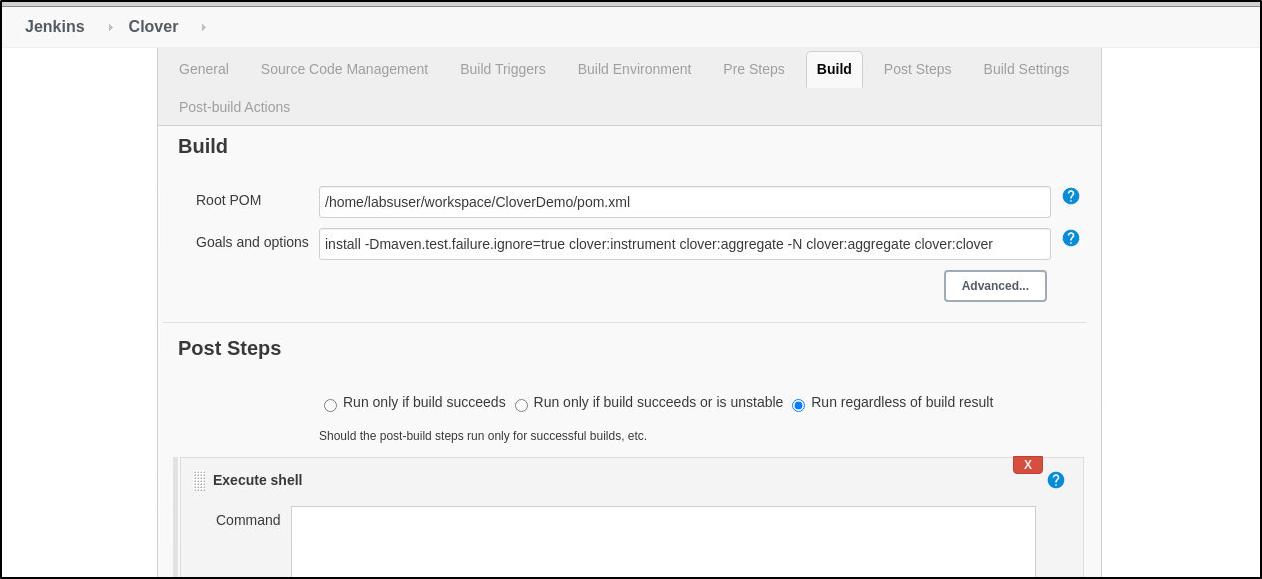
**</project>**

* After adding the code in pom.xml, **right click** on your project folder in eclipse and select **maven**-> **update project**.
* Install the **Clover plugin** via **Manage Jenkins -> Manage Plugins**.



**Step 6.4.3:** Creating Jenkins job for Maven

* To create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL. For example, http://localhost:8081/
* Click on **Create New Job**. Enter the item name, select **Maven Project** and click **OK**.
* Once you click **OK,**the page will be redirected to its project form. Here, you will need to enter the project information.
* In the **build** section provide the location of the pom.xml file in your local system.



* In the **build** section specify the following goals and options for maven:

**install -Dmaven.test.failure.ignore=true clover:instrument clover:aggregate -N clover:aggregate clover:clover**

* Build the job.

**Step 6.3.4:** Push the code to your GitHub repositories

* Open your command prompt and navigate to the folder where you have created your files.

**cd <folder path>**

* Initialize your repository using the following command:

**git init**

* Add all the files to your Git repository using the following command:

**git add .**

* Commit the changes using the following command:

**git commit . -m “Changes have been committed.”**

* Push the files to the folder you initially created using the following command:

**git push -u origin master**