Lesson 6 Demo 4: Code Coverage with Clover

This section will guide you to:

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

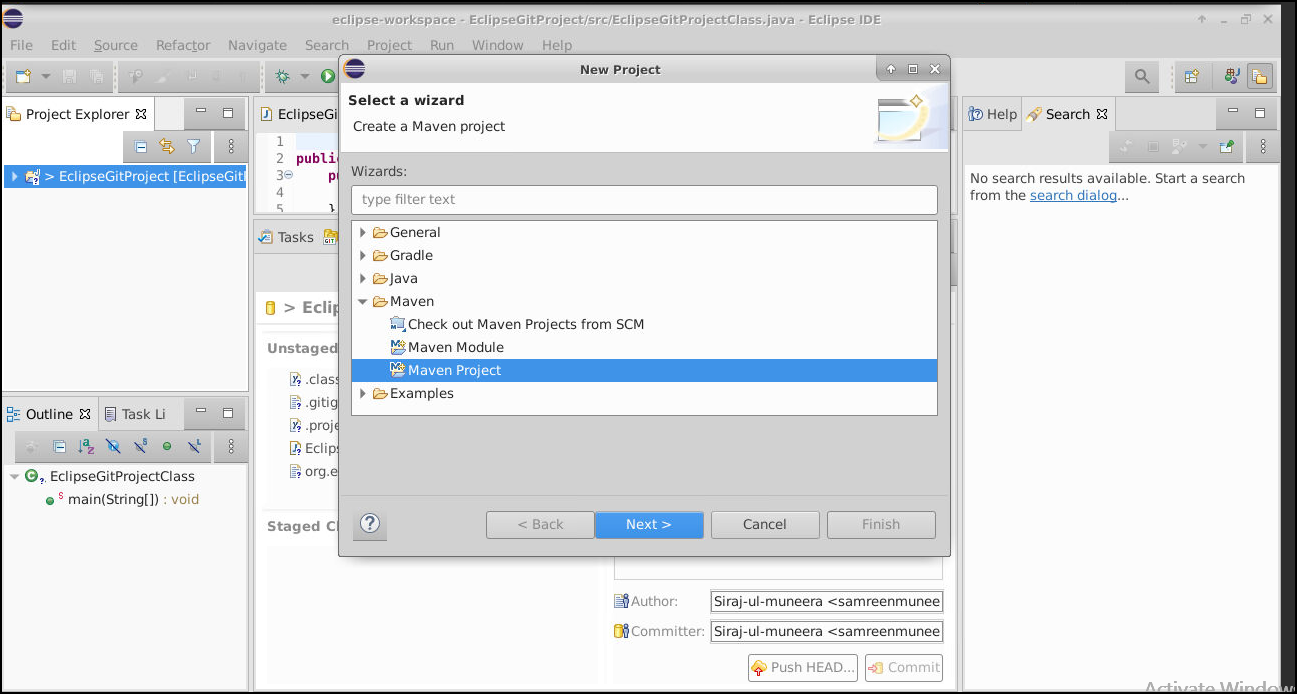
This guide has three subsections, namely:

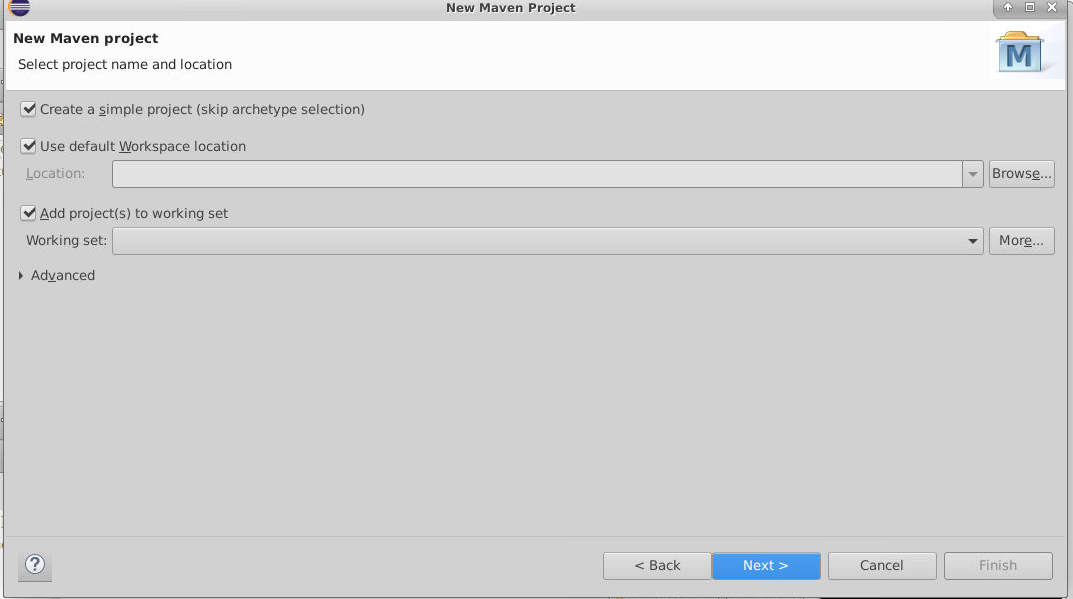
1. Logging in to Jenkins
2. Adding Clover in Jenkins and Maven project
3. Creating Jenkins job for Maven

**Step 1:** Logging in to Jenkins

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

**Step 2:** Adding 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 after the <version> section in the **pom.xml** file:

**<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>**

**<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>**

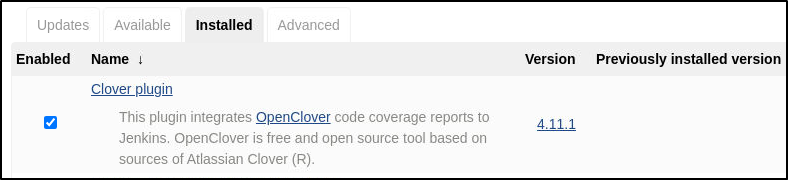
**</plugin>**

**</plugins>**

**</pluginManagement>**

**</build>**

* After adding the code in pom.xml, save the file and then **right click** on your project folder in eclipse, and select **maven**-> **update project**
* In your Jenkins, install the Clover plugin via **Manage Jenkins -> Manage Plugins ->** Under **Available** tabsearch for **Clover Plugin**
* Click on **Install without restart**



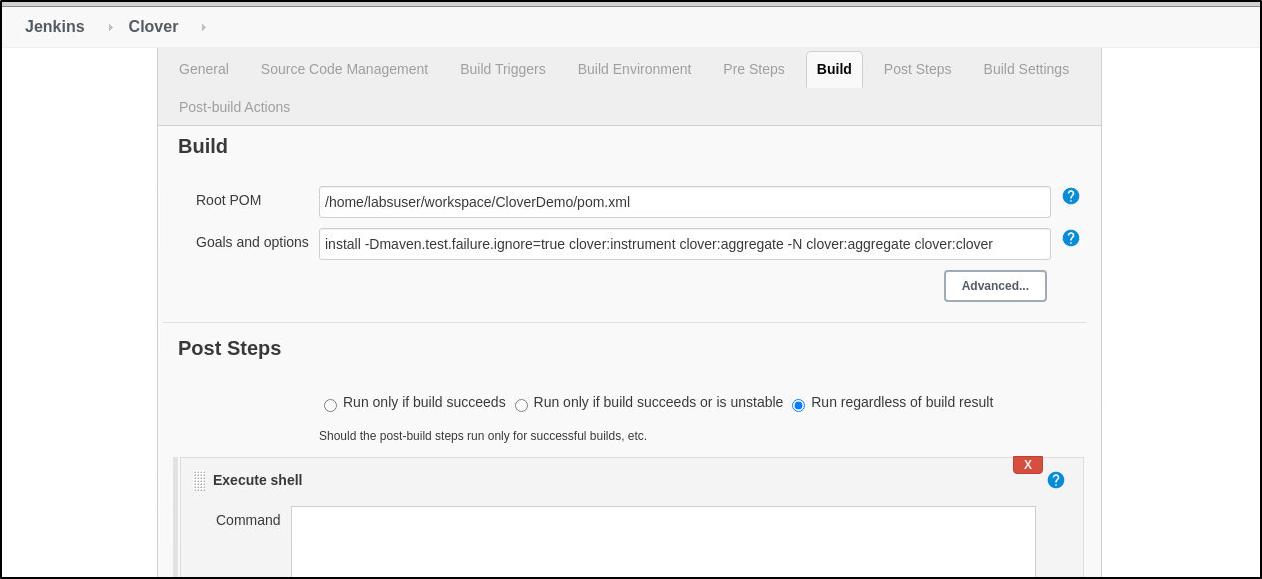
**Step 3:** Creating Jenkins job for Maven

* To create a new job in Jenkins, open the Jenkins dashboard with your Jenkins URL. For example, http://localhost:8081/
* Click on **New Item**. Enter the item name, select **Maven Project**, and click **OK**
* Once you click **OK,**the page will be redirected to its project form.
* In the **Build** section of your job, for Root POM give the path of the pom.xml in your local system as shown:

**/home/labsuser/workspace/CloverDemo/pom.xml**

* Specify the following goals and options for maven as:

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



* Build the job and see the logs in the console output

