Build and Release Management Lab Experiment 4

Name: Aryan Mukerji

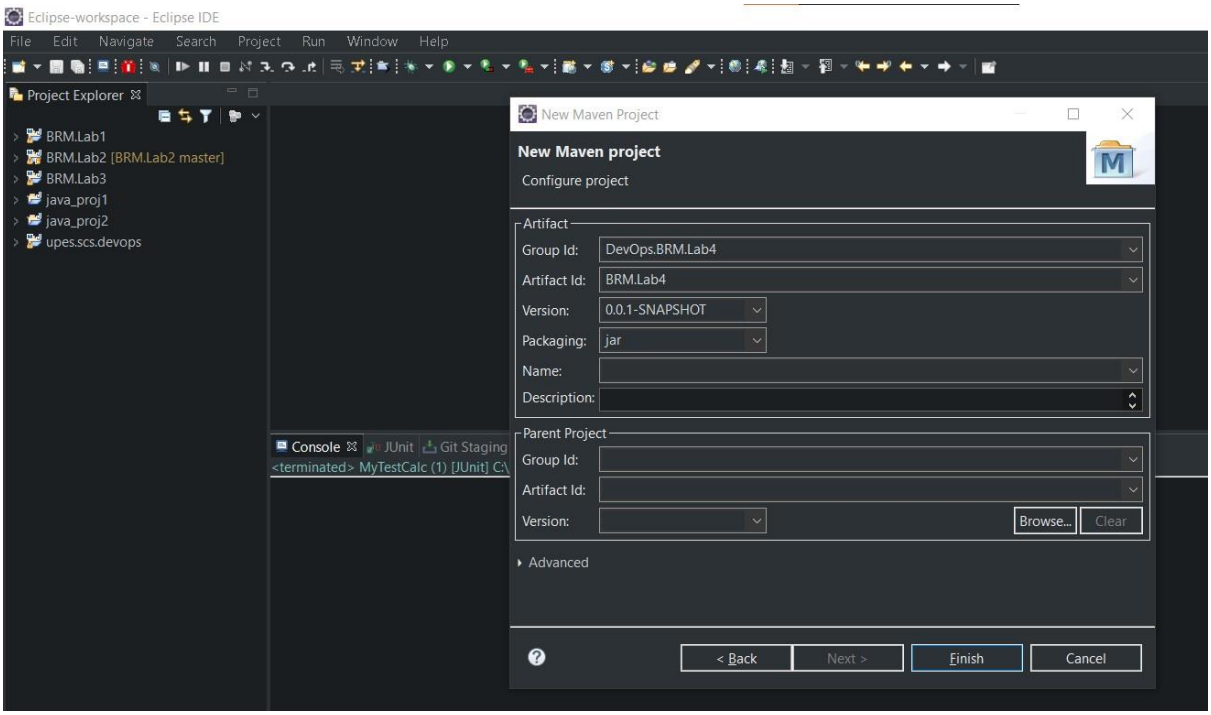
Roll No.: R171219007

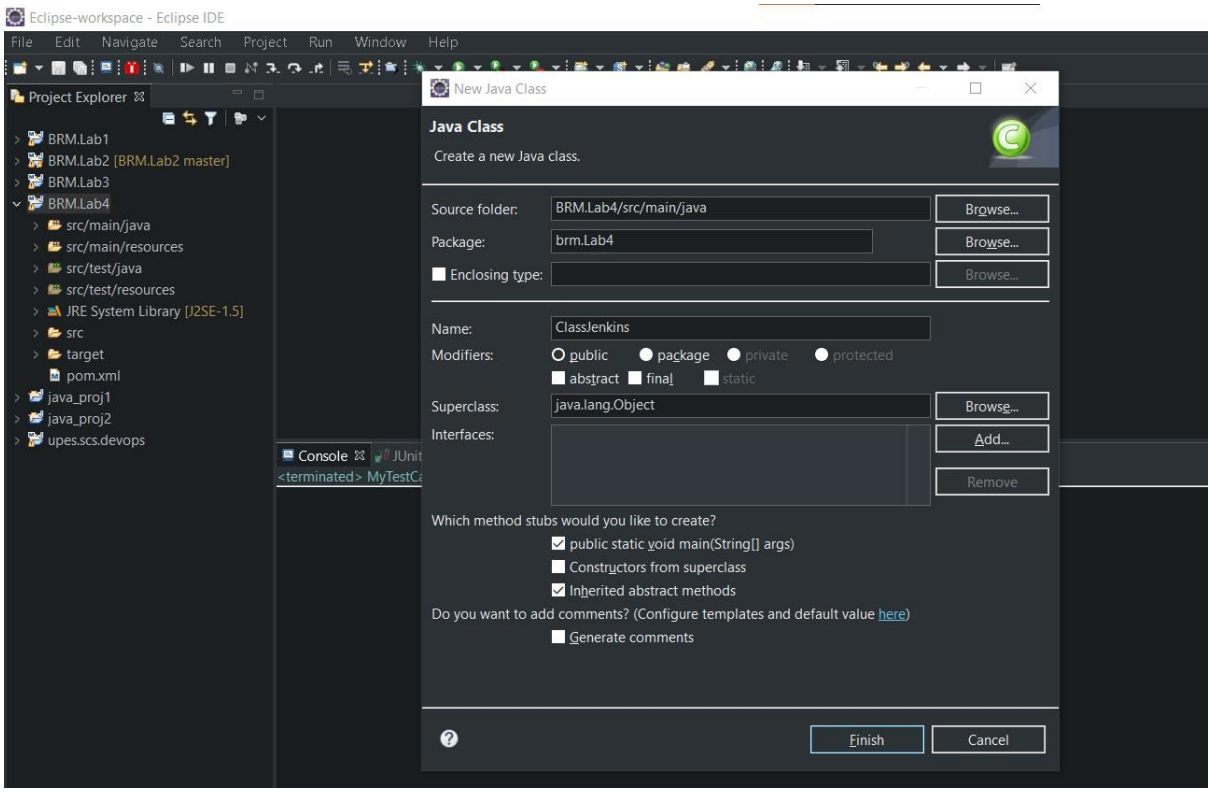
SAP ID.: 500075964

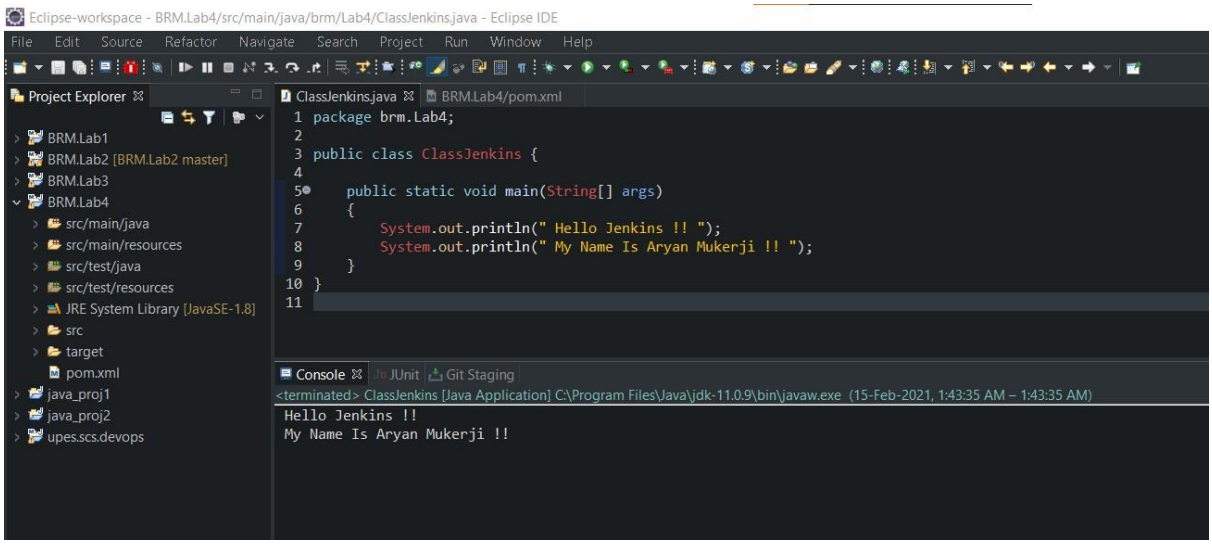
Branch: B.Tech-CSE-DevOps

Aim: Create a maven project with a java class, initialize it with git, push the project into the remote repository and then use jenkins to automate the process of creating a maven build.

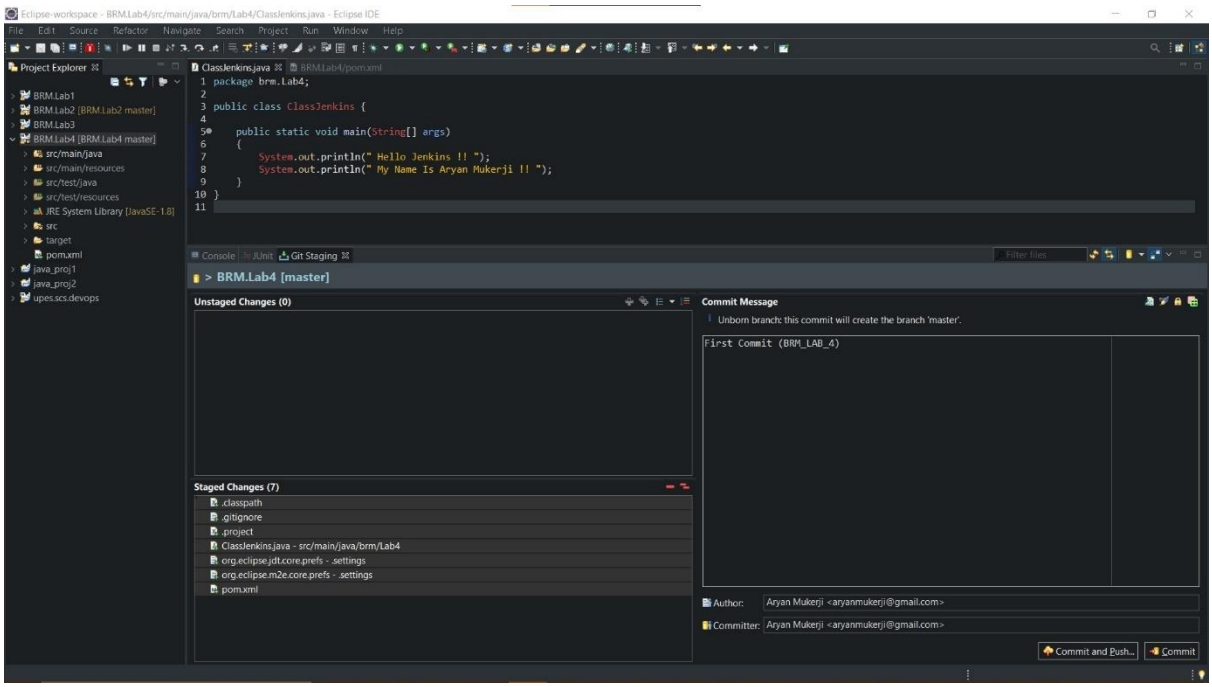
Step 1: Create a Maven project and a Java Class in it and add some java code in the class file.

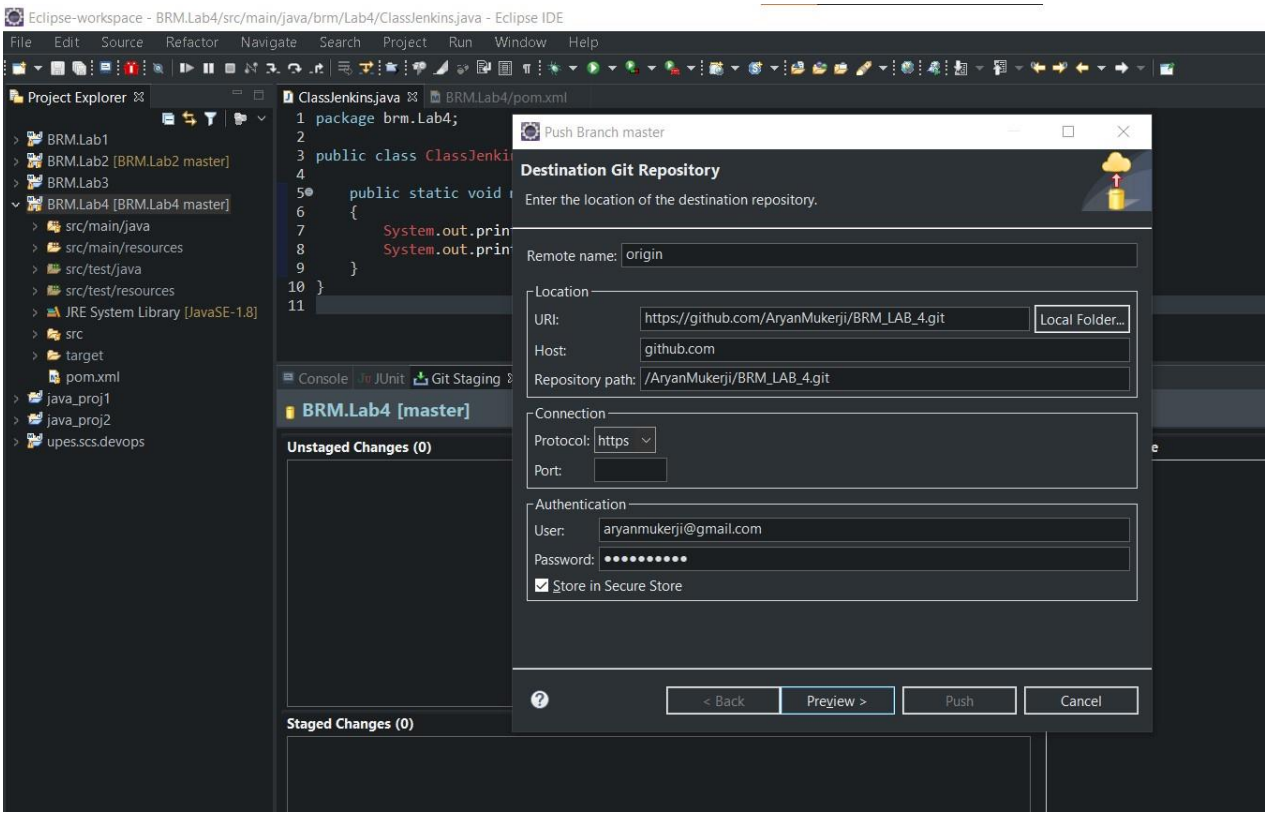


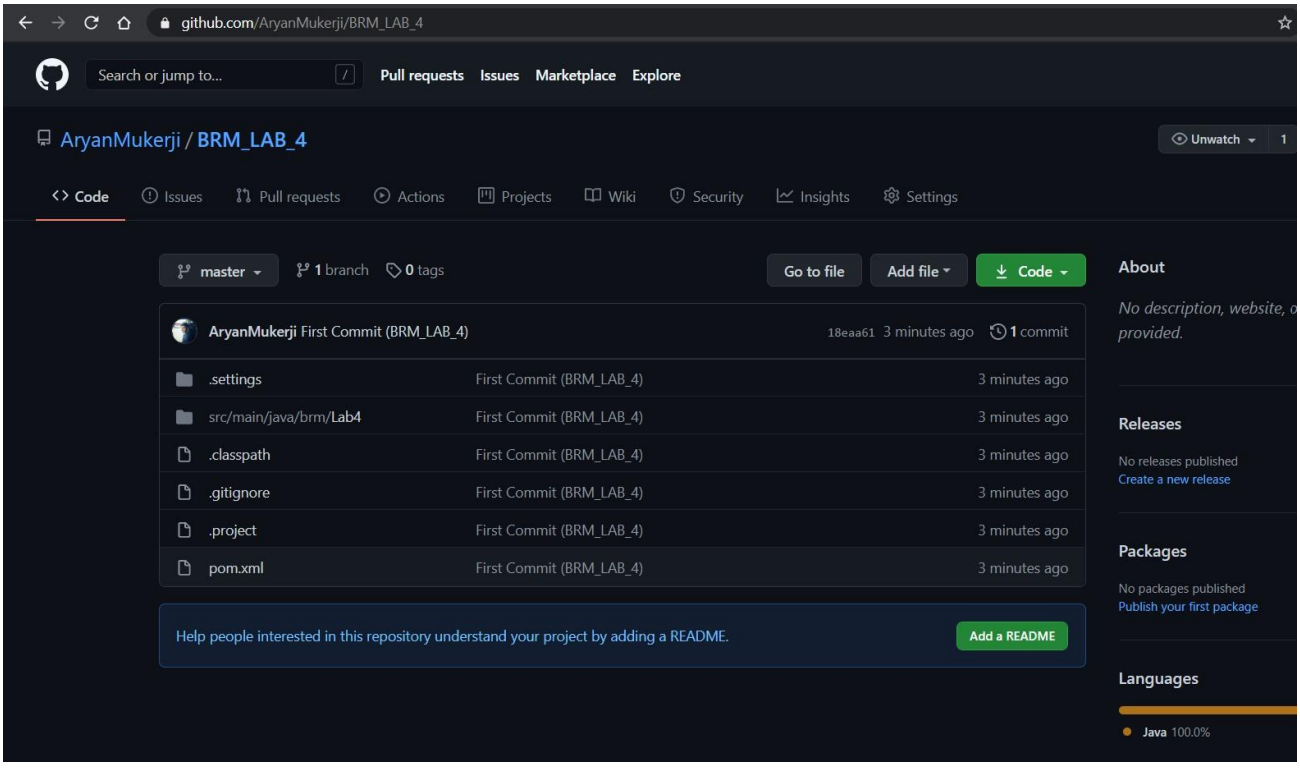




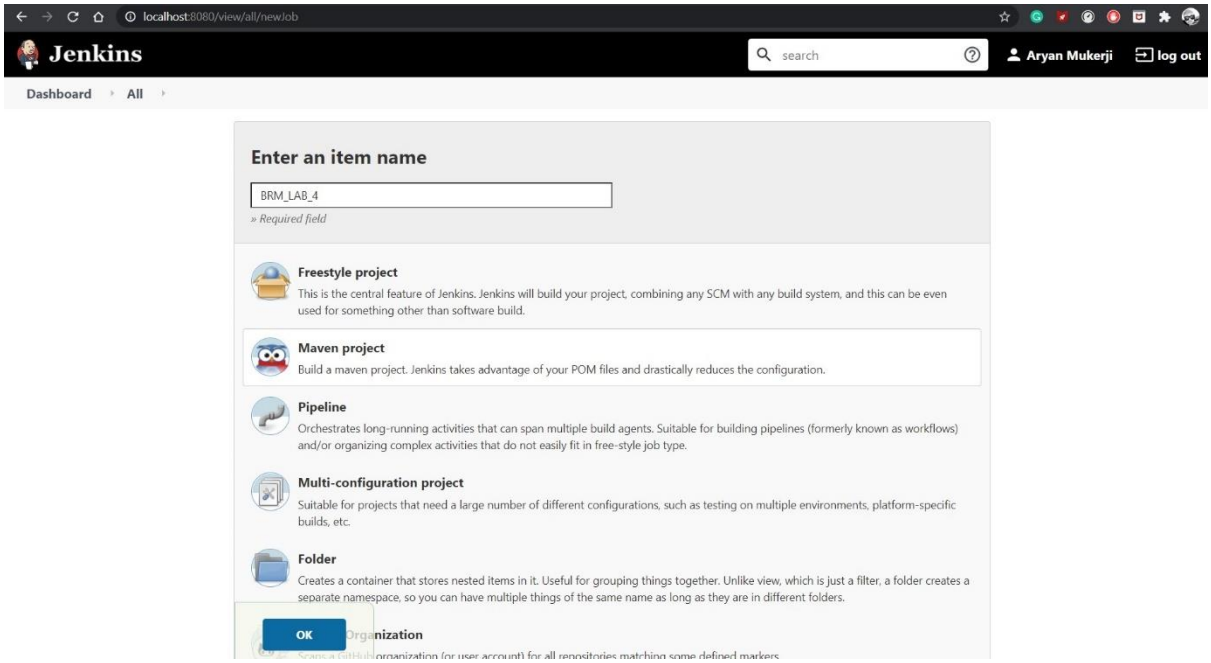
Step 2: Now initialize the project as a local git repository and then push the project into a remote git repository.



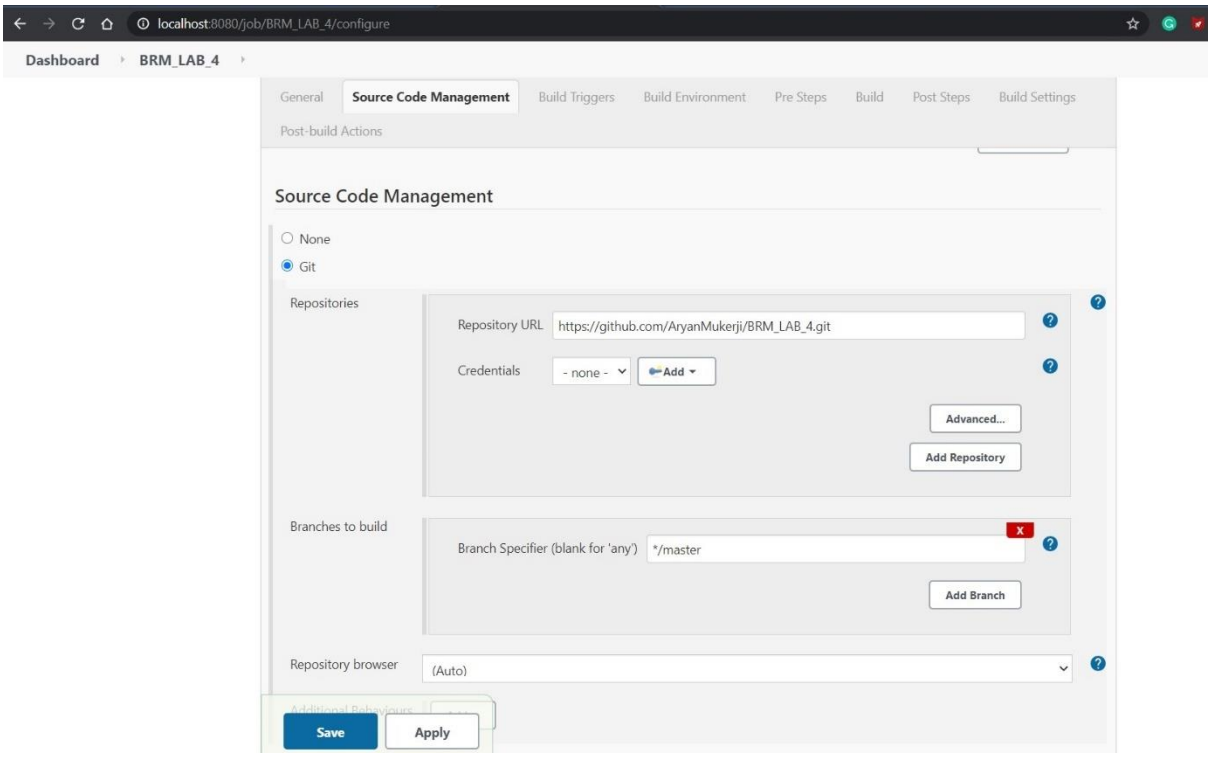

Eclipse-workspace - BRM.Lab4/src/main/java/brm/Lab4/ClassJenkins.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
BRM.Lab1
BRM.Lab2 [BRM.Lab2 master]
BRM.Lab3
BRM.Lab4 [BRM.Lab4 master]
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
src
target
pom.xml
java_proj1
java_proj2
upes.scs.devops
ClassJenkins.java
BRM.Lab4/pom.xml
1 package brm.Lab4;
3 public class ClassJenki
5 public static void
6 {
7 System.out.prin
8 System.out.prin
9 }
10 }
Console JUnit Git Staging
BRM.Lab4 [master]
Unstaged Changes (0)
Staged Changes (0)
Push Branch master
Destination Git Repository
Enter the location of the destination repository.
Remote name: origin
Location
URI: https://github.com/AryanMukerji/BRM_LAB_4.git
Local Folder...
Host: github.com
Repository path: /AryanMukerji/BRM_LAB_4.git
Connection
Protocol: https
Port:
Authentication
User: aryanmukerji@gmail.com
Password: ••••••••••
Store in Secure Store
< Back
Preview >
Push
Cancel


github.com/AryanMukerji/BRM_LAB_4
Search or jump to...
Pull requests Issues Marketplace Explore
AryanMukerji / BRM_LAB_4
Unwatch 1
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
master 1 branch 0 tags
Go to file Add file Code
About
No description, website, o
provided.
AryanMukerji First Commit (BRM_LAB_4)
18eaa61 3 minutes ago 1 commit
.settings First Commit (BRM_LAB_4) 3 minutes ago
src/main/java/brm/Lab4 First Commit (BRM_LAB_4) 3 minutes ago
.classpath First Commit (BRM_LAB_4) 3 minutes ago
.gitignore First Commit (BRM_LAB_4) 3 minutes ago
.project First Commit (BRM_LAB_4) 3 minutes ago
pom.xml First Commit (BRM_LAB_4) 3 minutes ago
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Languages
Java 100.0%
Help people interested in this repository understand your project by adding a README.
Add a README

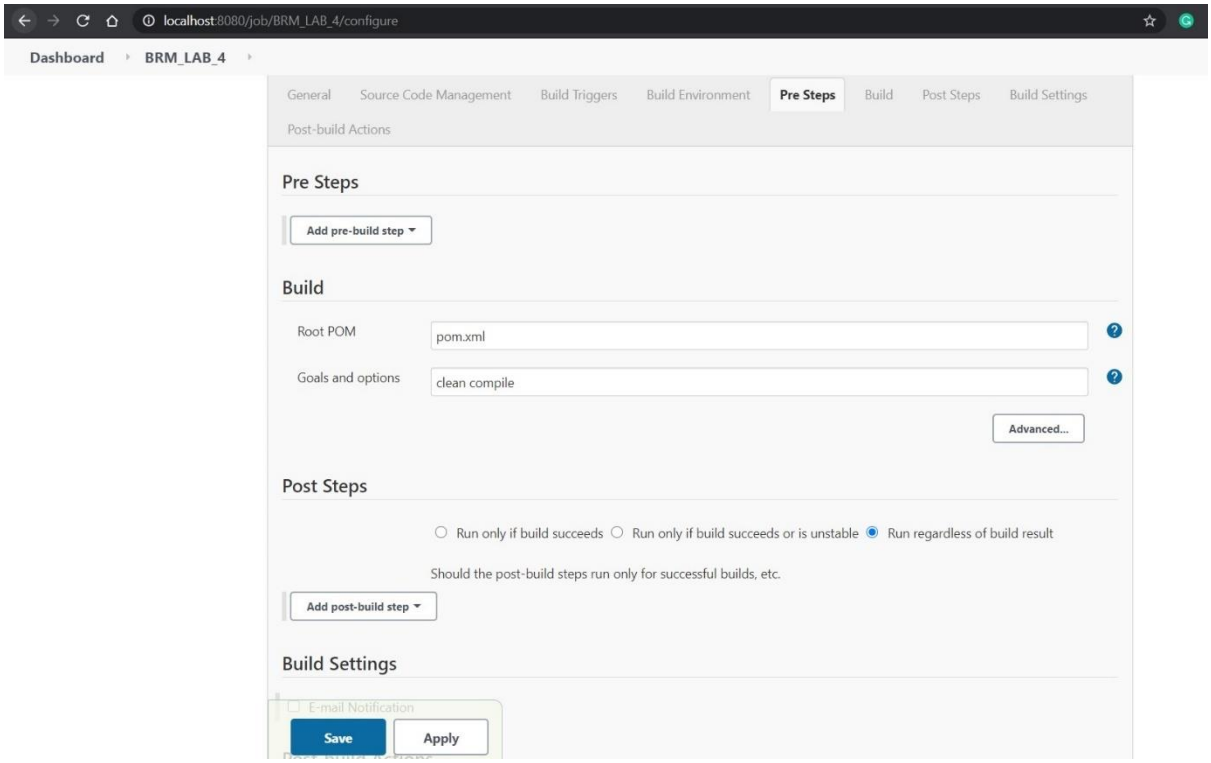
Step 3: Now open Jenkins, in Dashboard, click on "New item", a page will open, now enter a name for your maven project and select "Maven project" and then click "OK".

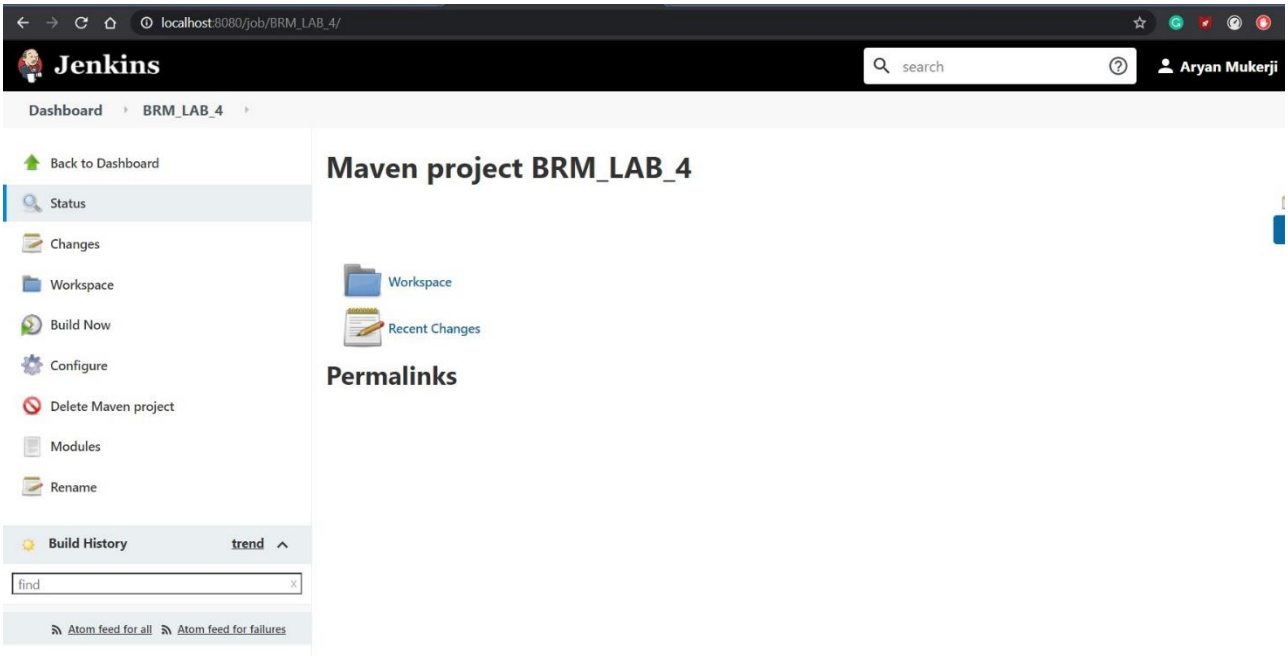


Step 4: A new page will open with many configurations to select from for your maven project. Under "Source Code Management", select "Git" and paste remote repository URL in which the project is pushed.

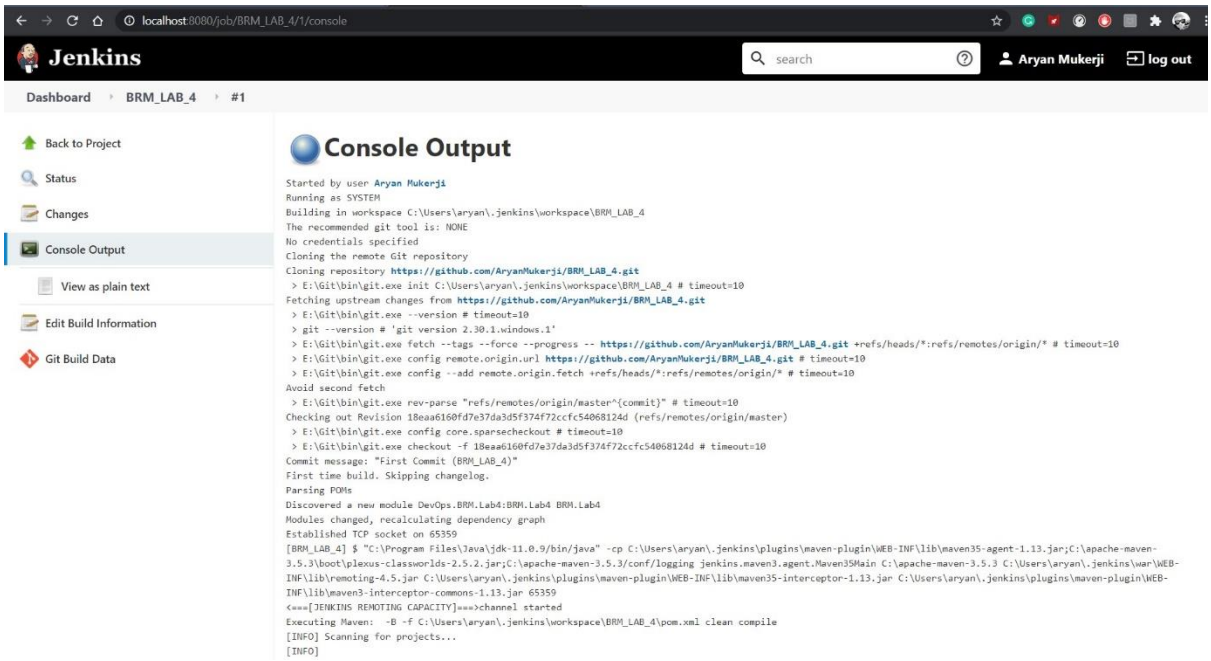


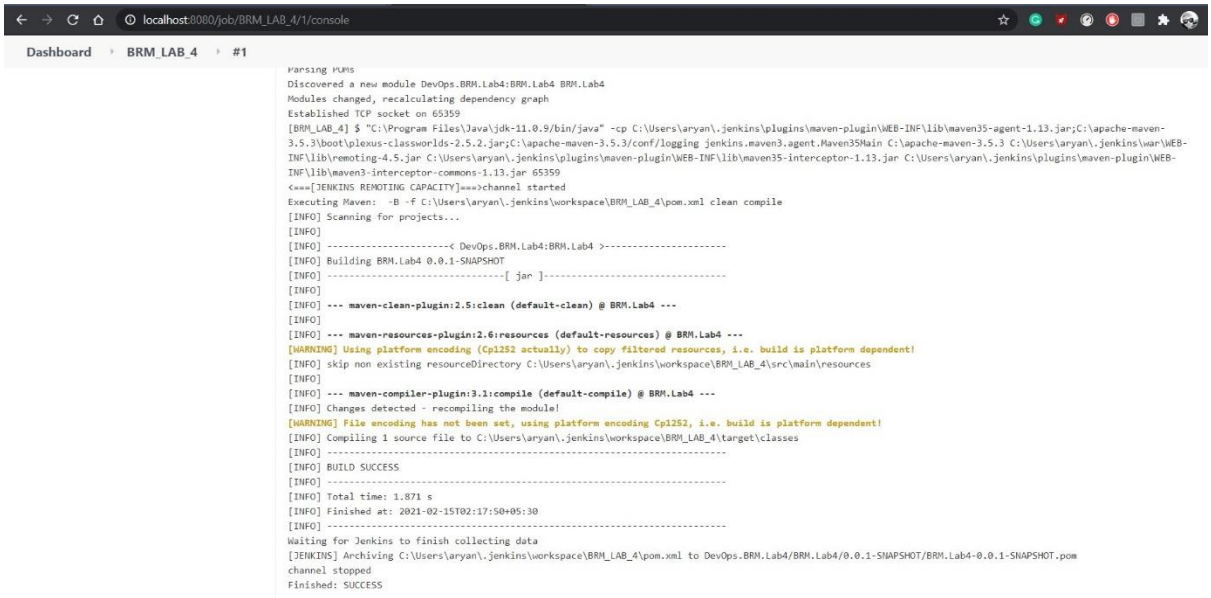
Step 5: Now under "Build", add goals like clean, compile, install and after that click on "Apply" and then click on "Save". Now a maven build job has been created.



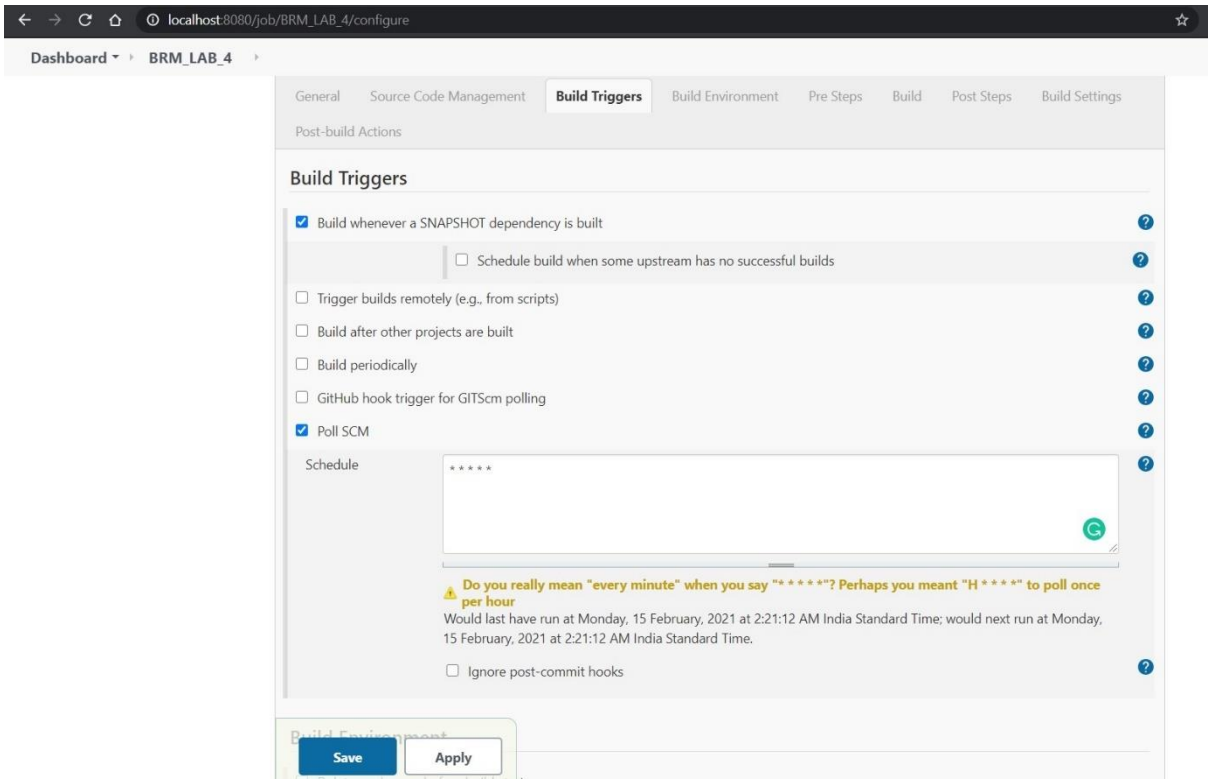


Step 6: Now click on "Build Now" to build the run the job. You can see the job running by opening "Console Output".

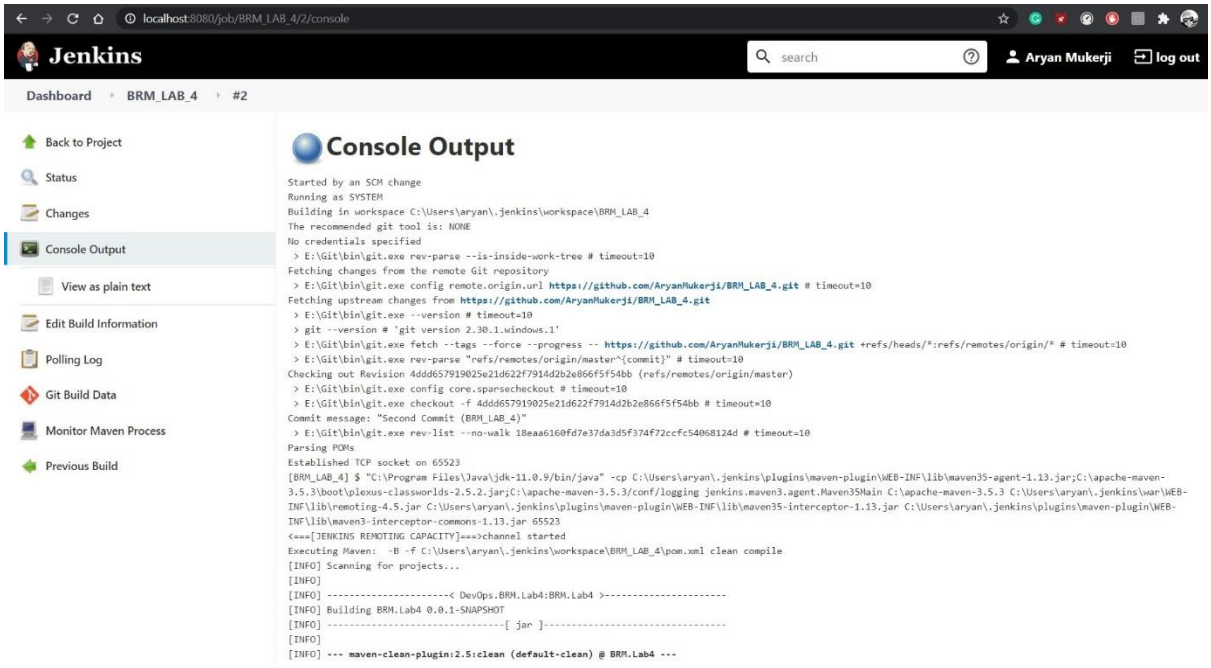


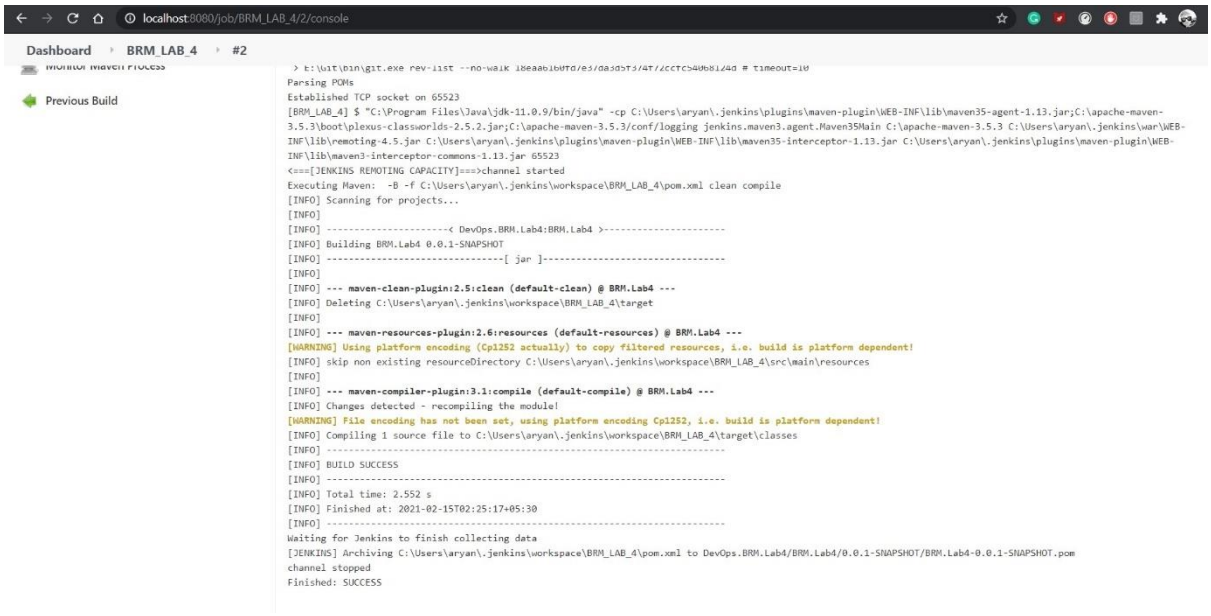


Step 7: We can now automate this process by changing some configurations in the job. Click on "Configure", a page will open with the previous configurations, now under "Build Triggers", select "Poll SCM" and in "Schedule" box type (* * * * *), then click on "Apply" & click on "Save".



Step 8: Now whenever there are some changes the remote repository, jenkins will automatically start the build job to create maven builds.





Result: Hence we successfully created a maven project, initialized it with git, pushed the project into the remote repository and then used jenkins to automate the process of creating a maven build.