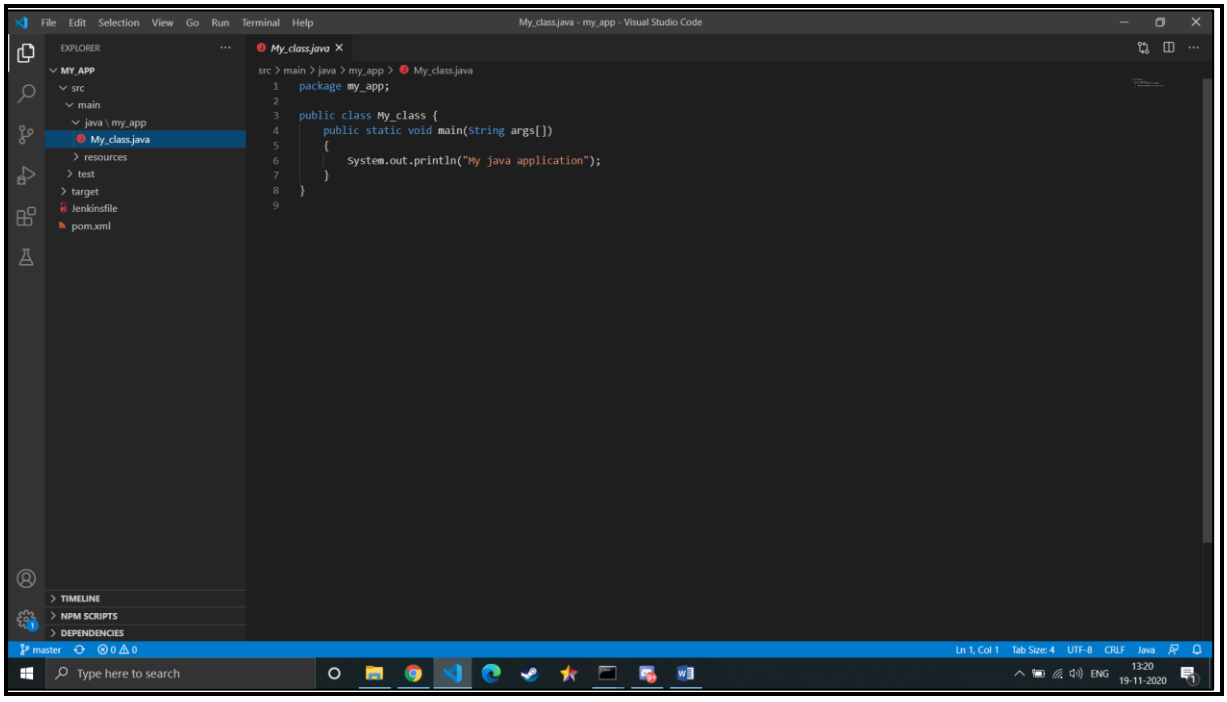
CICD ASSIGNMENT-2

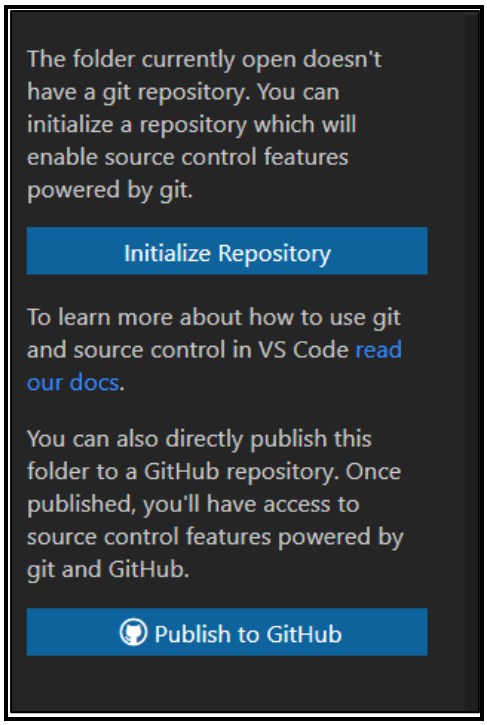
Submitted by:
Ajay Kumar Tyagi
500067403
R171218008

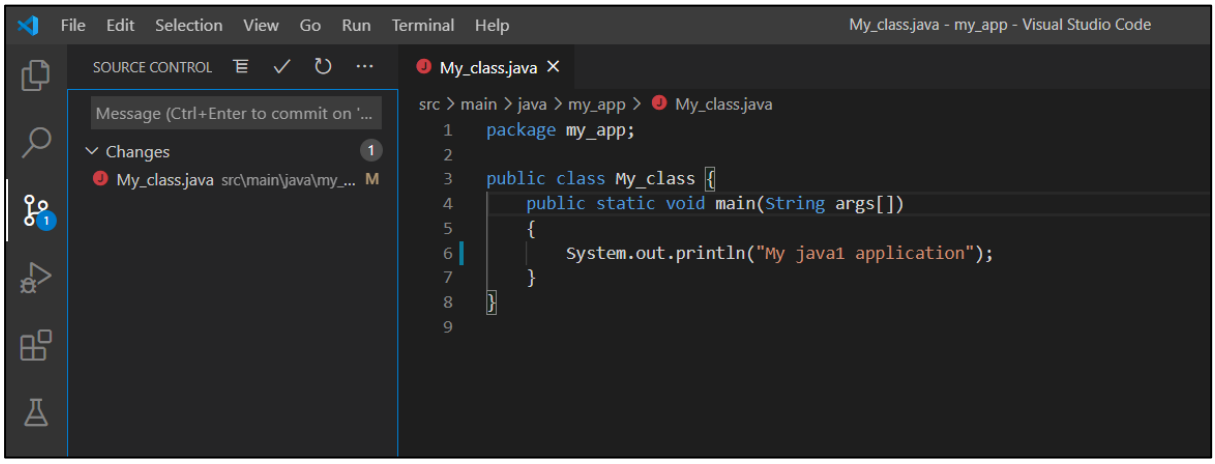
Aim: To create a Jenkinsfile Pipeline to generate build and push the generated build on Nexus server using Jenkinsfile on GitHub.

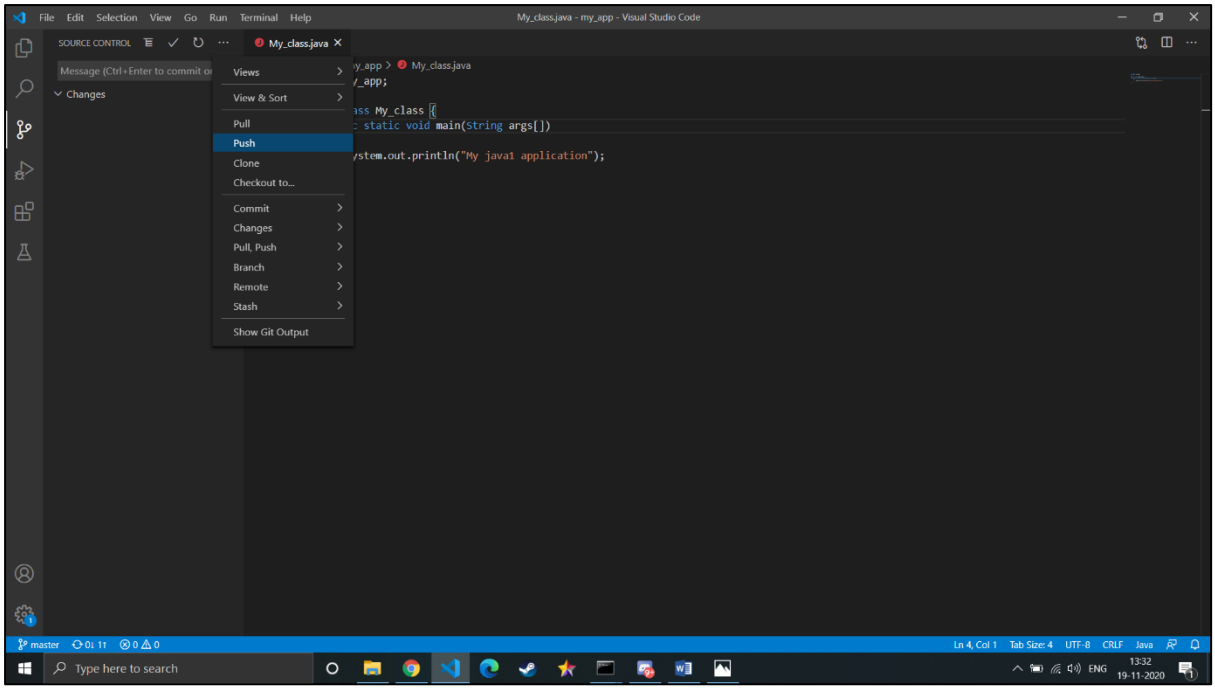
1) Create a maven project using Eclipse or any IDE. Here I have use visual studio code for creating maven project.

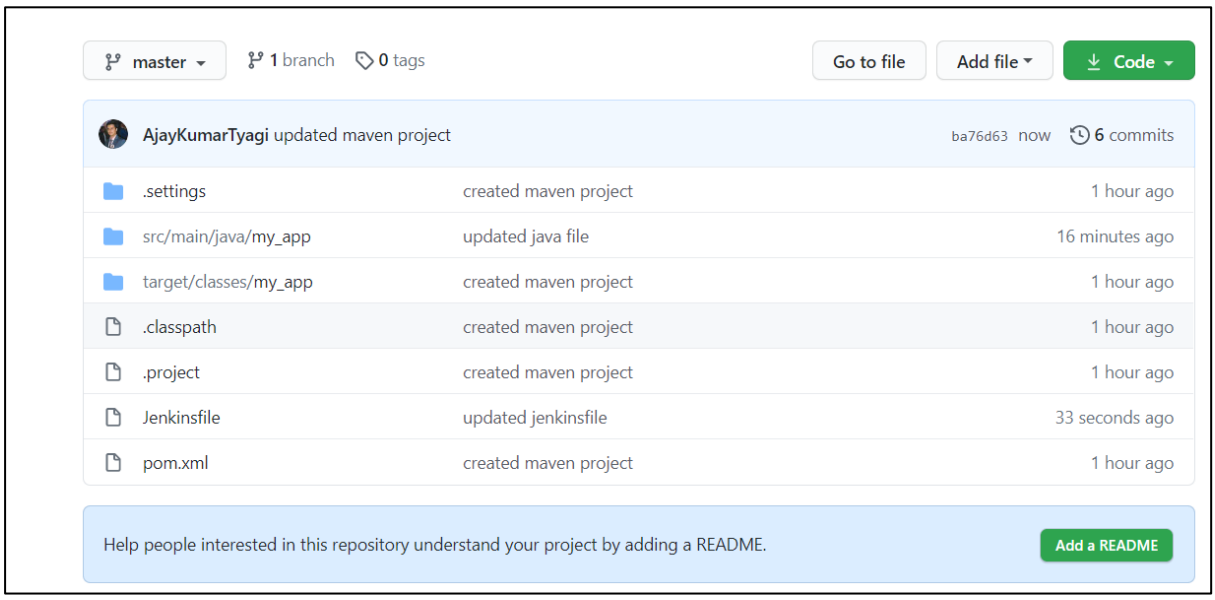


2) Now stage, commit and push the project on GitHub from the IDE.


The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.
Initialize Repository
To learn more about how to use git and source control in VS Code read our docs.
You can also directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.
Publish to GitHub


File Edit Selection View Go Run Terminal Help
My_class.java - my_app - Visual Studio Code
SOURCE CONTROL
Message (Ctrl+Enter to commit on '...
Changes 1
My_class.java src\main\java\my_... M
My_class.java
src > main > java > my_app > My_class.java
1 package my_app;
2
3 public class My_class {
4 public static void main(String args[])
5 {
6 System.out.println("My java1 application");
7 }
8 }
9





3) Also create a Jenkinsfile to build a pipeline and deploy jar file on nexus. This Jenkinsfile will create three stages in pipeline: Build, Test and deploy. Running this file will first build the project and at last deploy project on Nexus server.

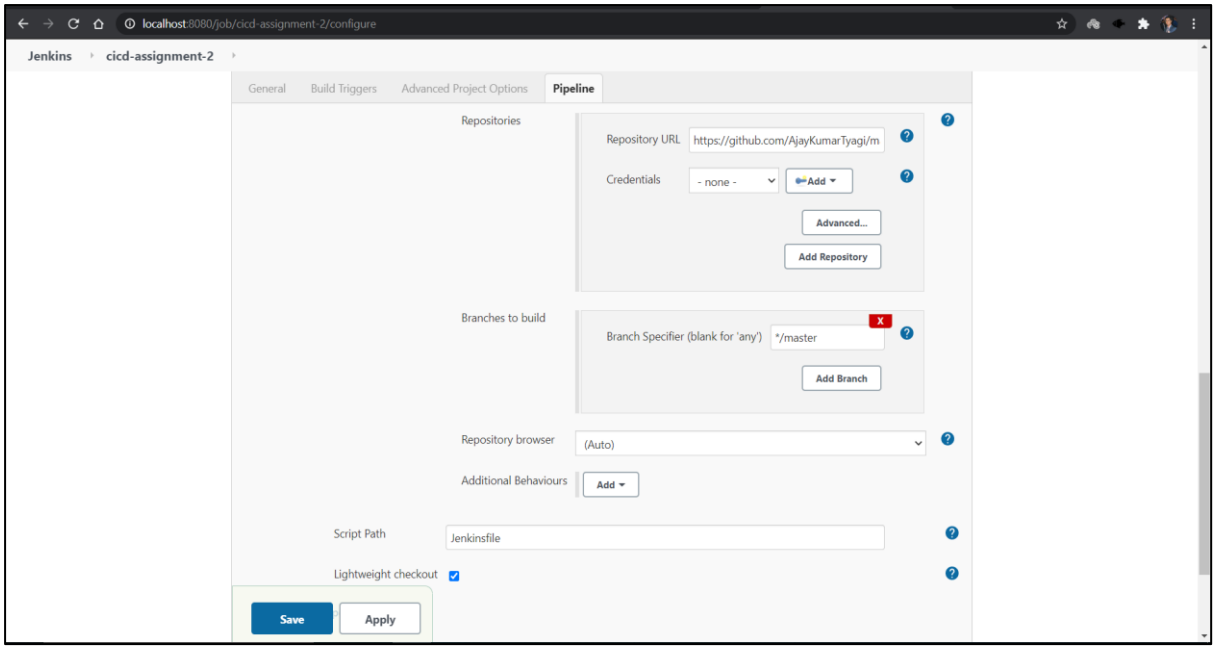
21 lines (20 sloc) | 324 Bytes

```
pipeline{
    agent any
    stages{
      stage('Build'){
          steps{
          echo 'Build Stage'
          }
      }
      stage('Test'){
          steps{
          shell('mvn clean')
          }
      }
          stage('deploy'){
          steps{
          shell('mvn deploy')
          }
          }
      }
    }

```

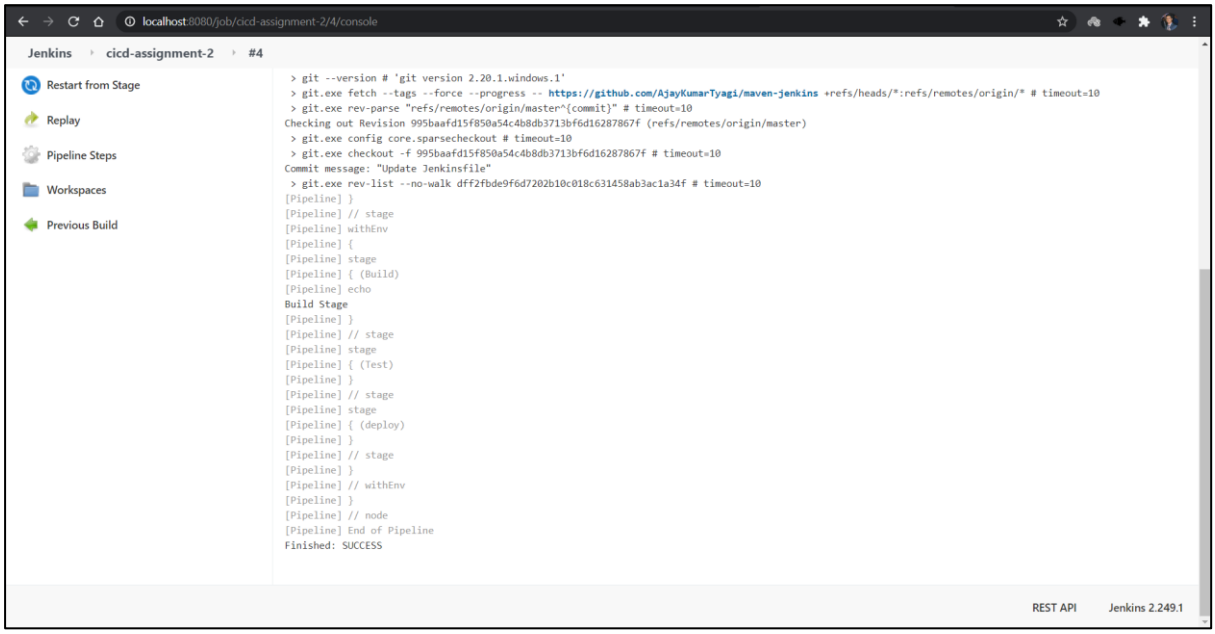
4) Also in pom.xml file we have to add details of our Nexus server repository where we want to deploy our project.

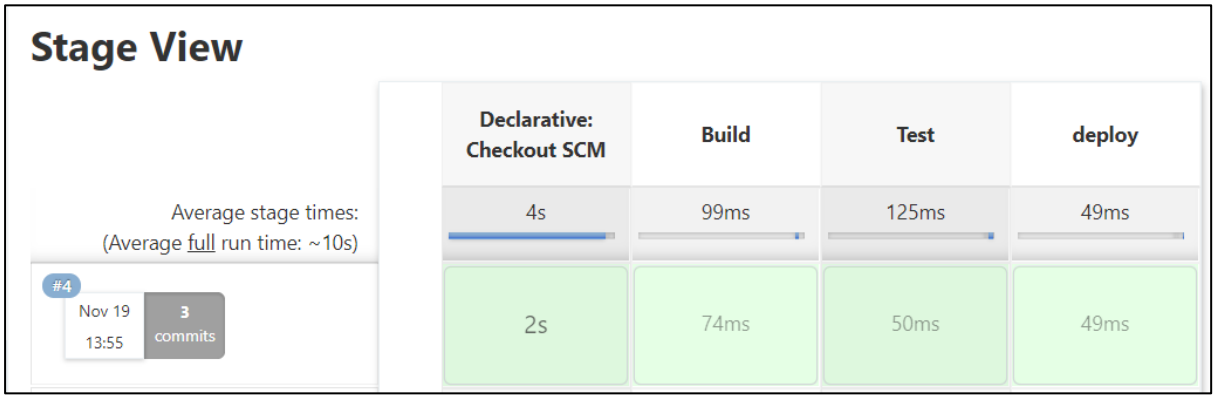
```
  <distributionManagement>
<snapshotRepository>
<id>cicd-lab</id>
<name>cicd-lab</name>
<url>http://localhost:8081/repository/cicd-lab/</url>
</snapshotRepository>
</distributionManagement>

```

5) Now in Jenkins create a Pipeline job. Here give GitHub repository url and Jenkinsfile location.



6) On successful run of this job we can see the following output.





7) At last, on Nexus server we can verify that our project is successfully deployed.