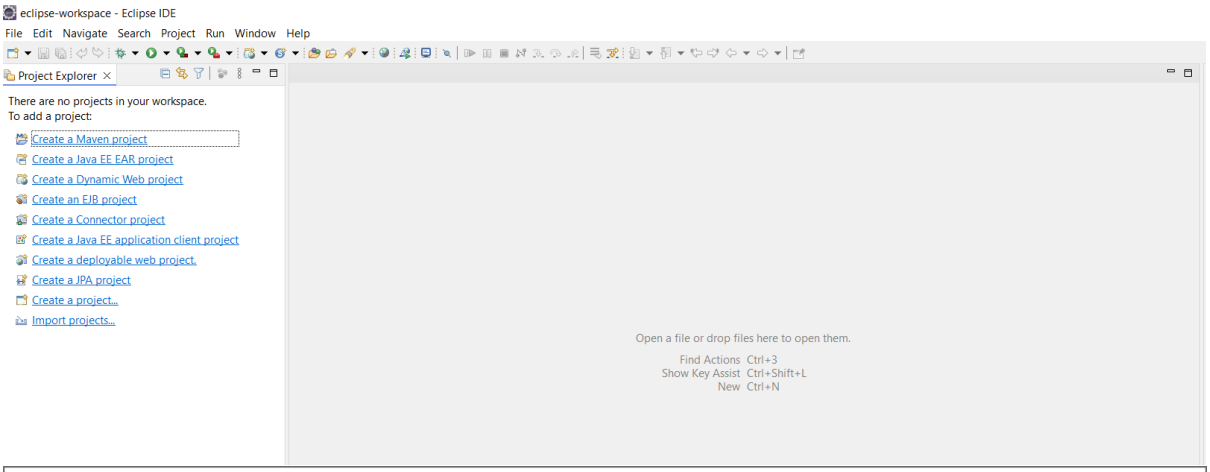
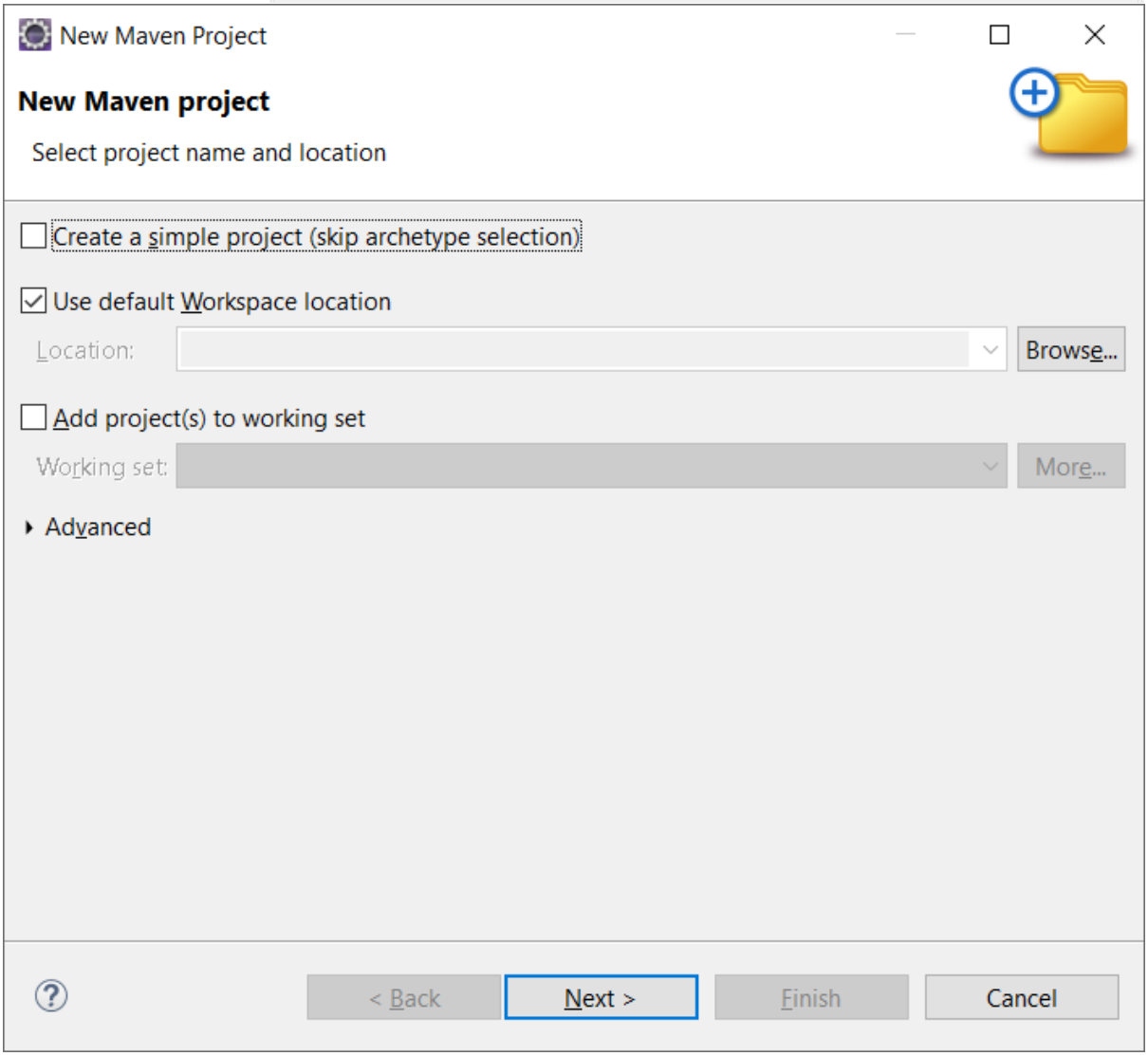
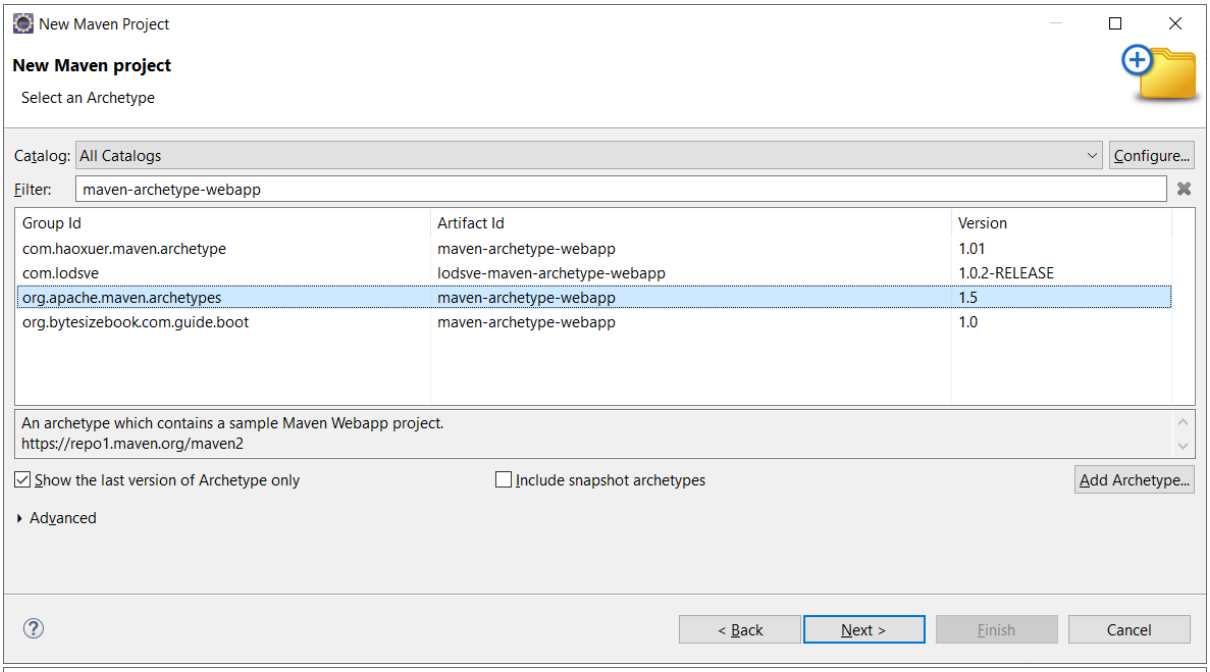
1. Install Jdk21
2. Download eclipse zip and extract the contents to C:\
3. Create a Maven Project from eclipse as:
   a. 

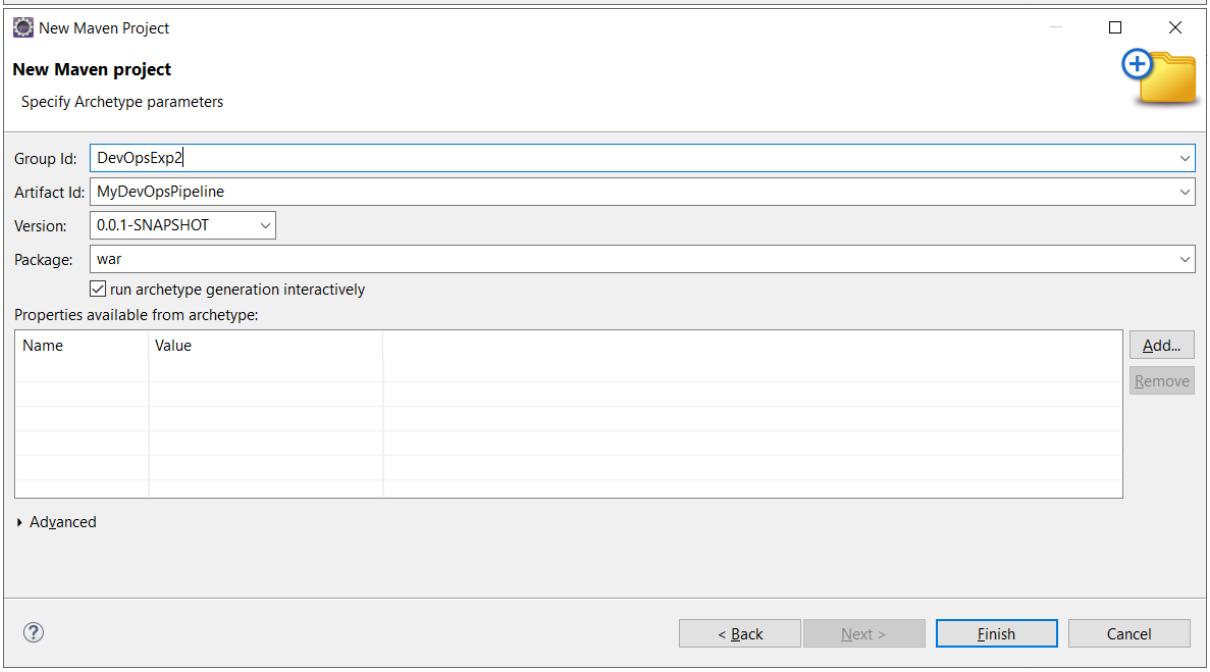
   b. 

   c. Click Next

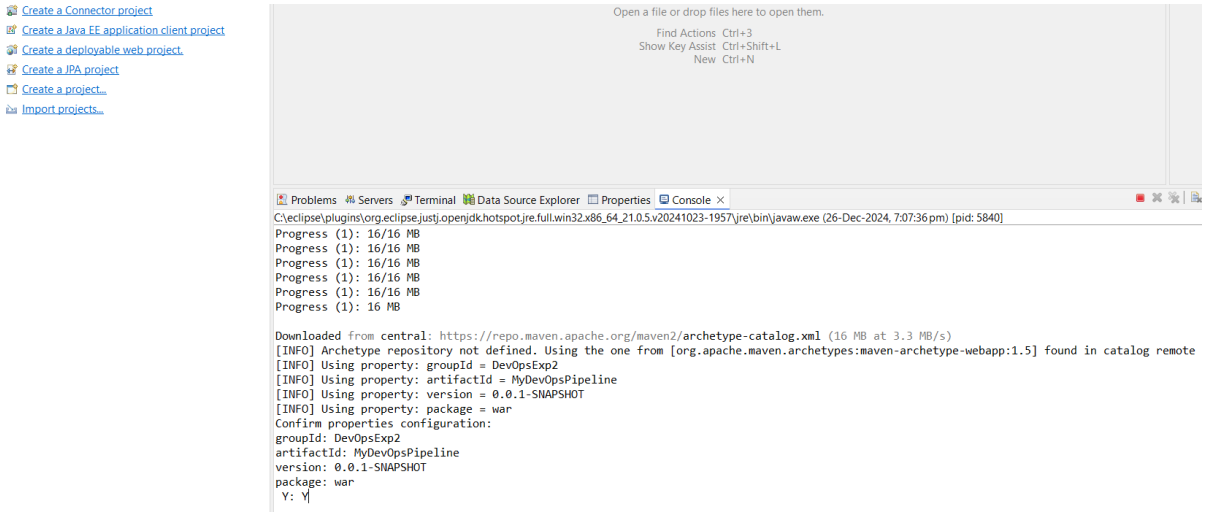
d.



e.

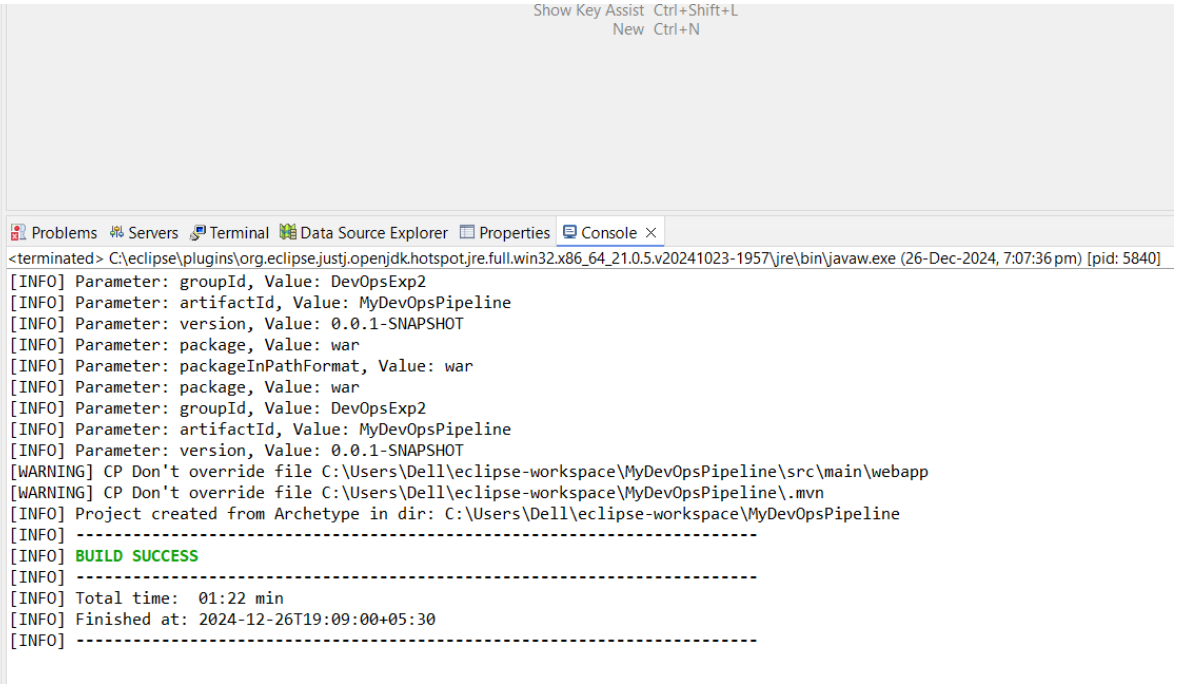


f. Click Finish



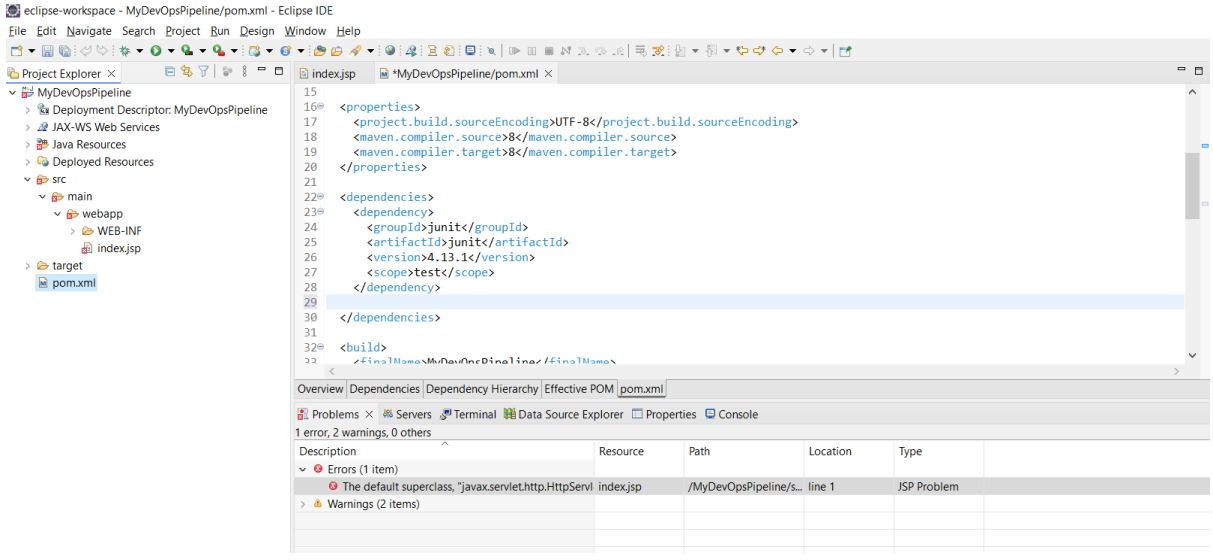
g.

h. Type Y and Press enter, You should see a Build Success message as below:



i.

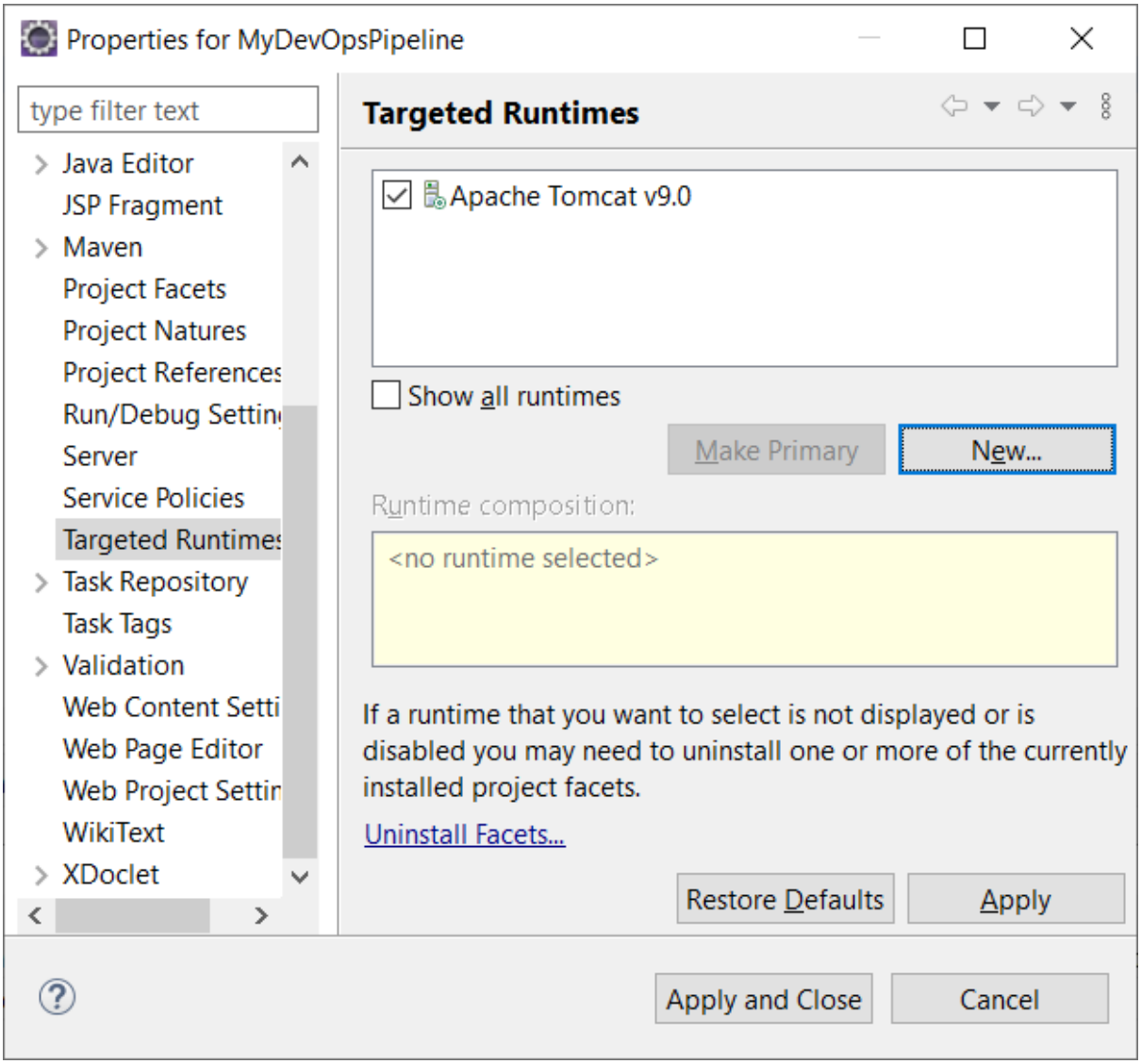
4. You should see the below structure in eclipse:



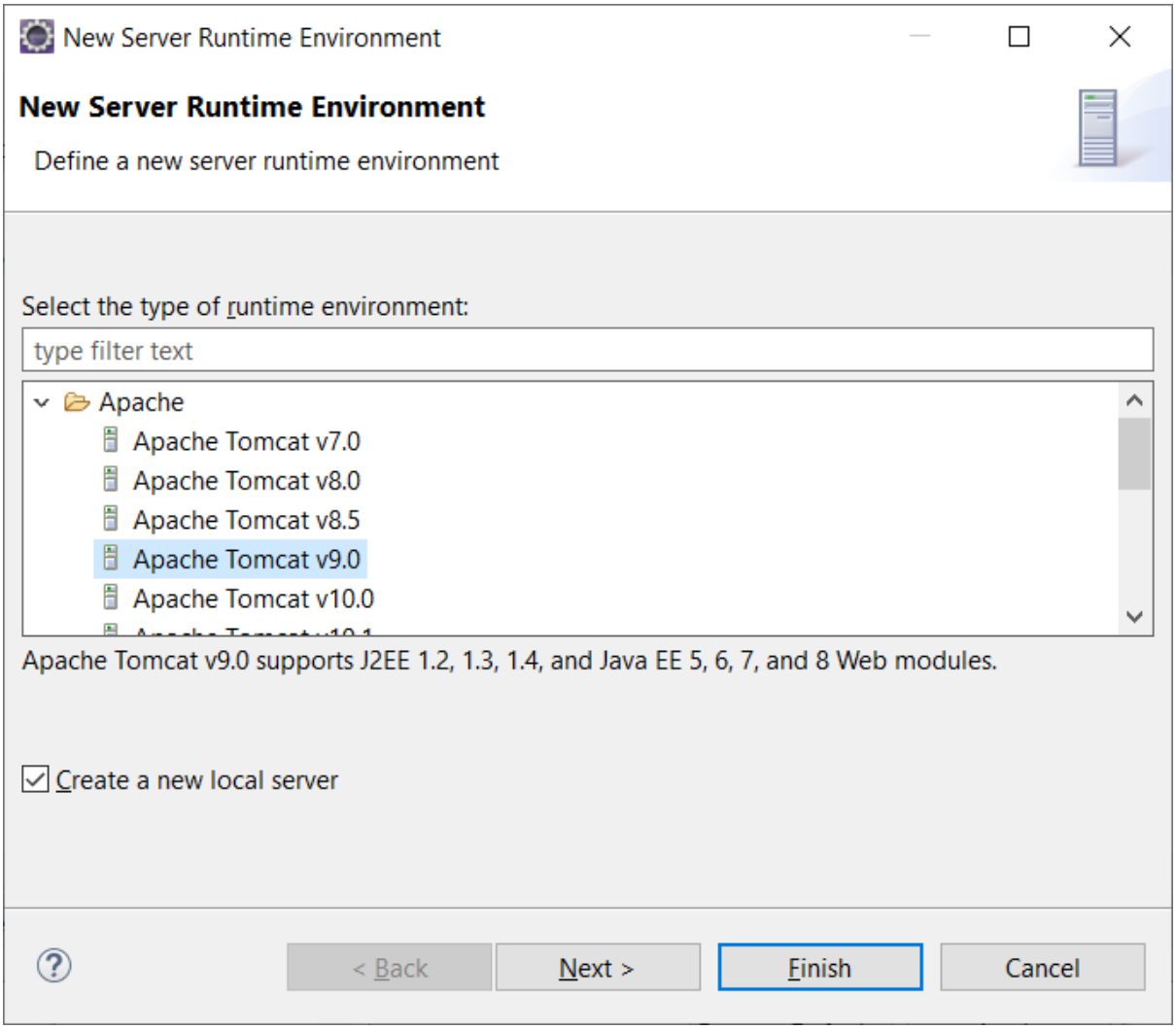
Click on pom.xml to open the file.
Add the below dependencies to the pom.xml:

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>

<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.10.2</version>
  <scope>test</scope>
</dependency>

<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>4.27.0</version>
</dependency>
```

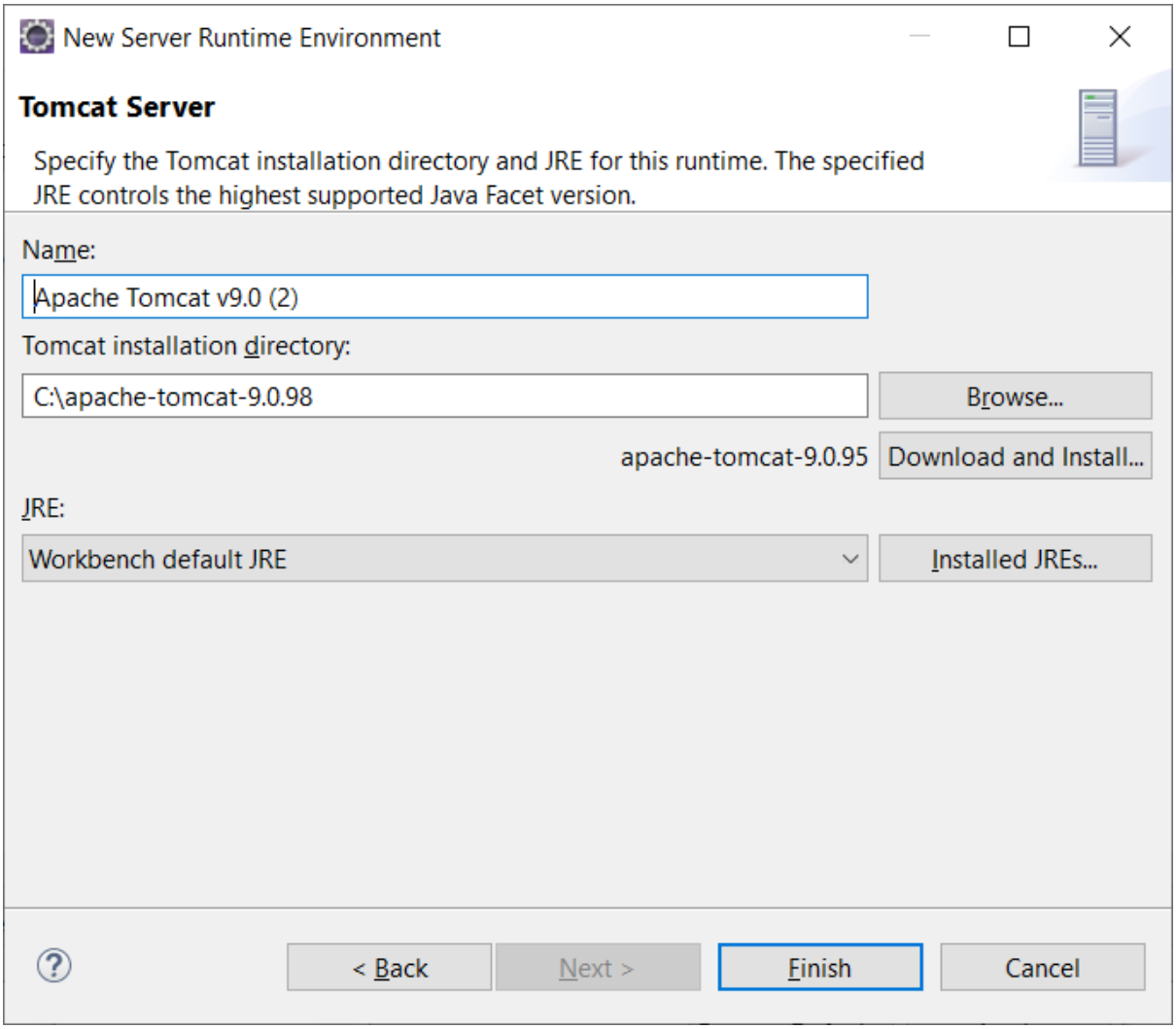
5. Install Tomcat 9.0-
    a. Download apache-tomcat-9.0.98.zip
    b. Extract to C:\
    c. Add Tomcat to Runtime by:
    d. Open Project Properties, and Click on TargettedRuntime



Click on New and Select Apache Tomcatv9.0 as below, and click Next



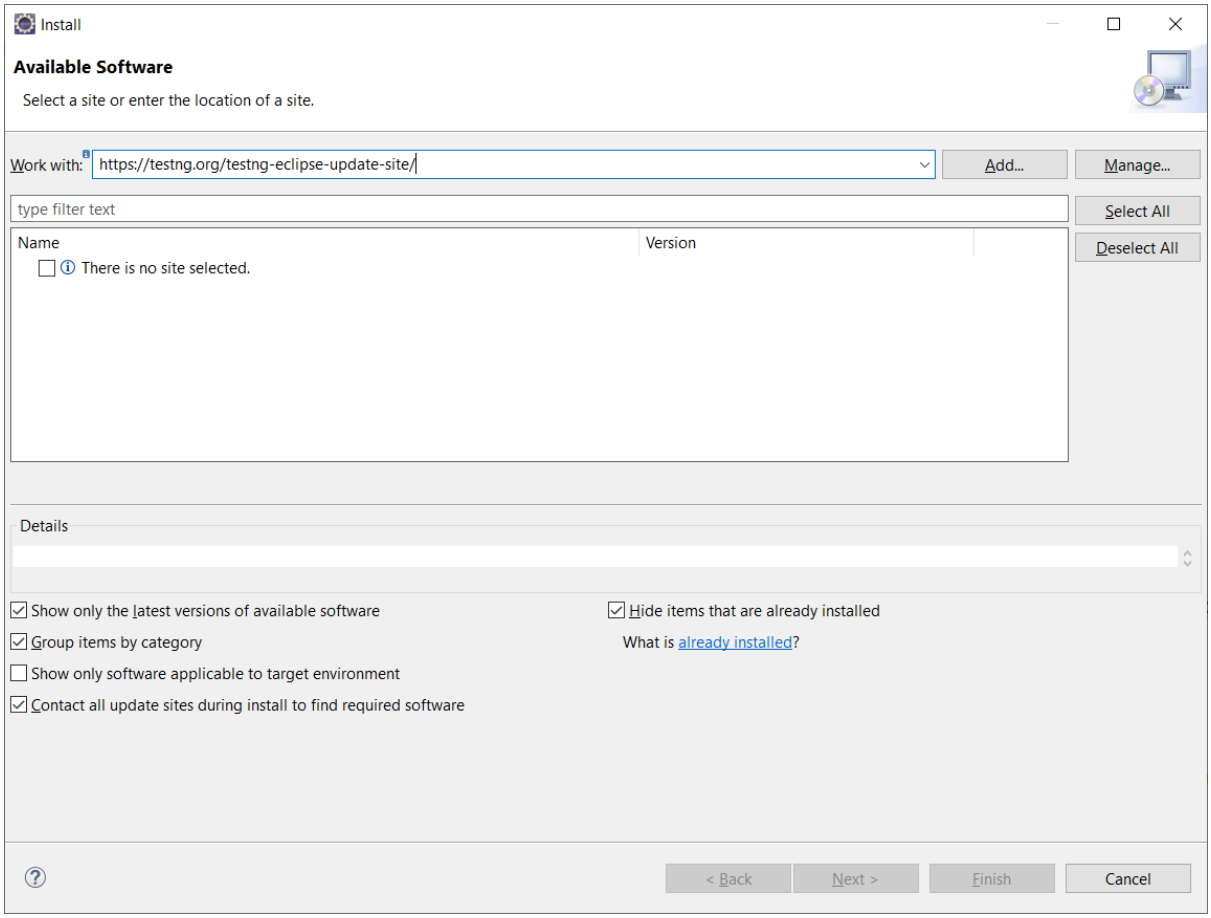
Give the extracted folder of Tomcat 9.0 as below, and Click Finish.

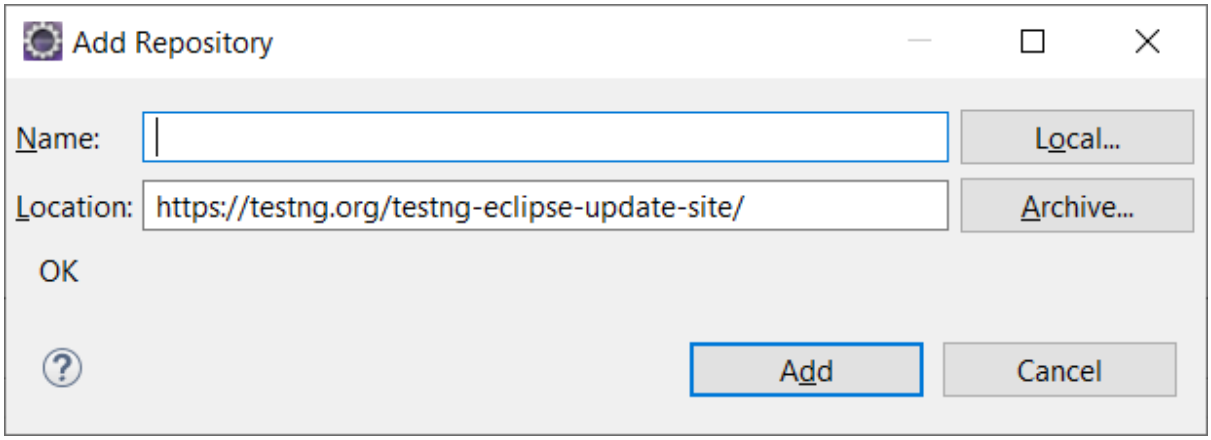


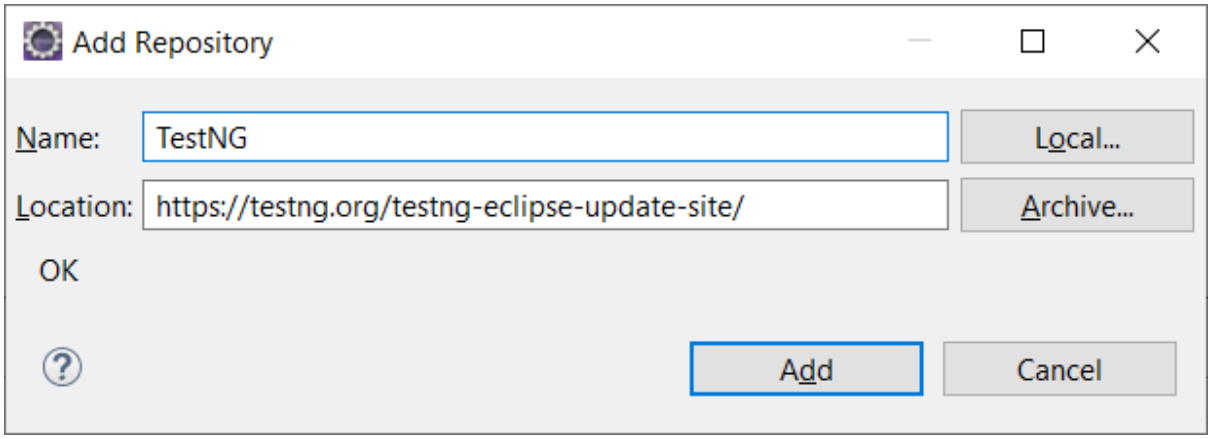
,

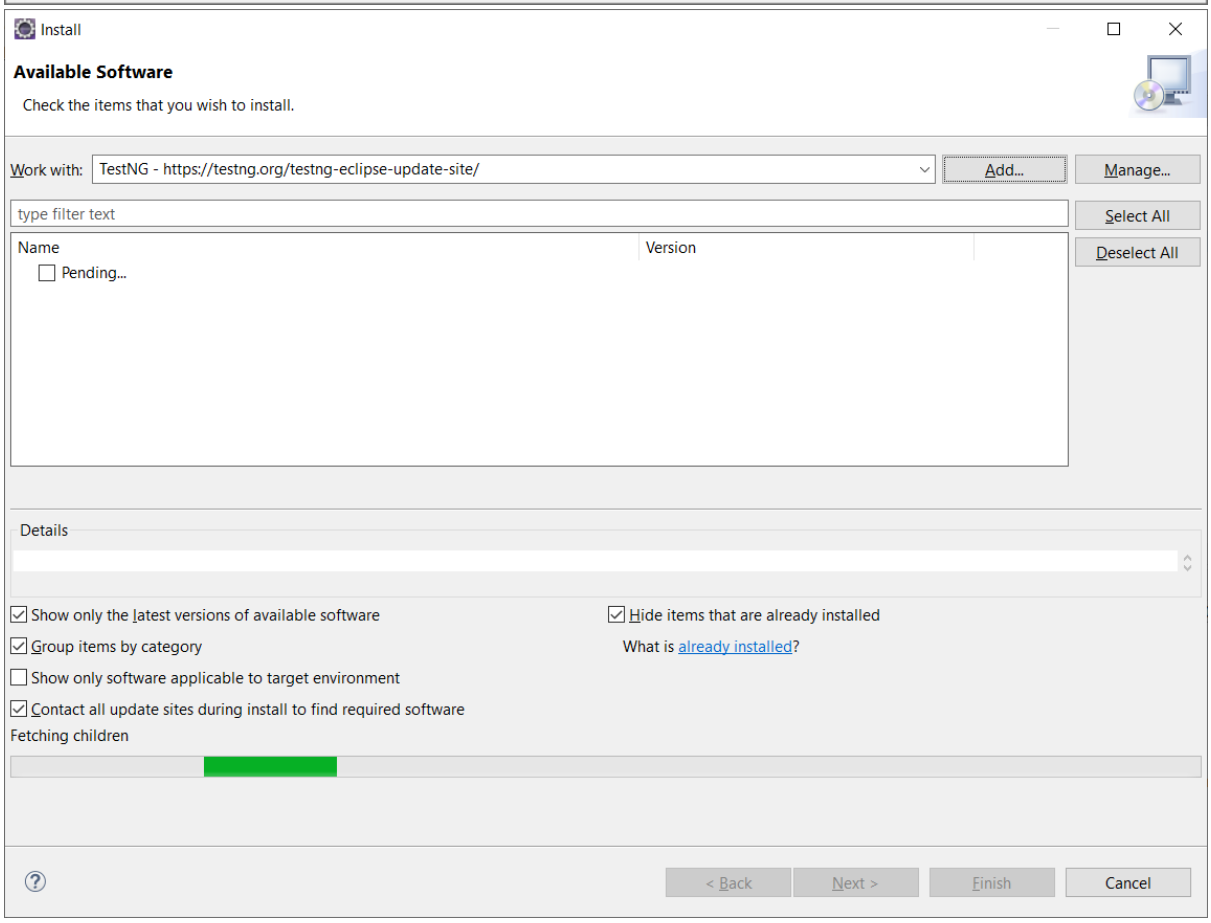
6. Open index.jsp and Press Ctrl+Shift+f , this will set the alignment of the jsp.

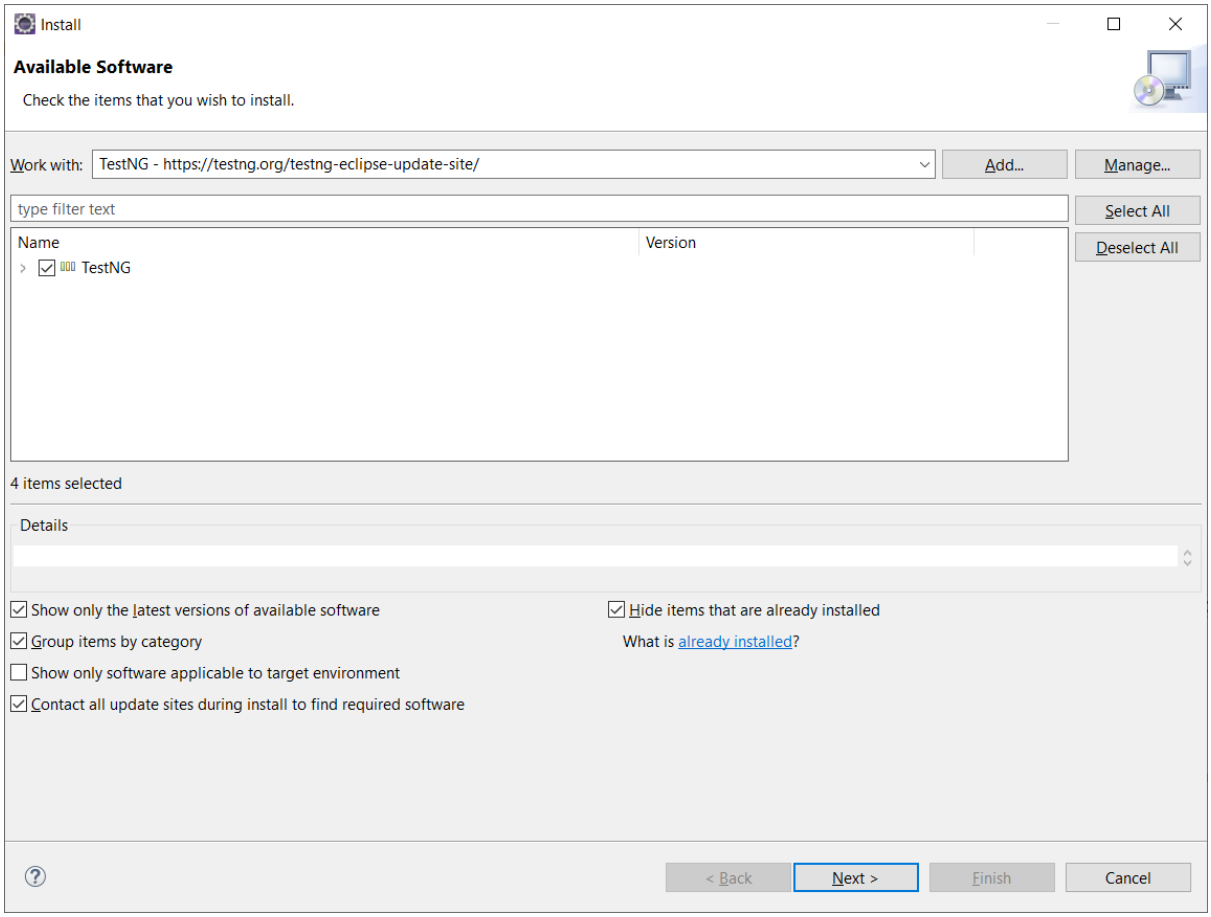
7. Click Help-> Install New Software, and give https://testng.org/testng-eclipse-update-site/ and Click Add



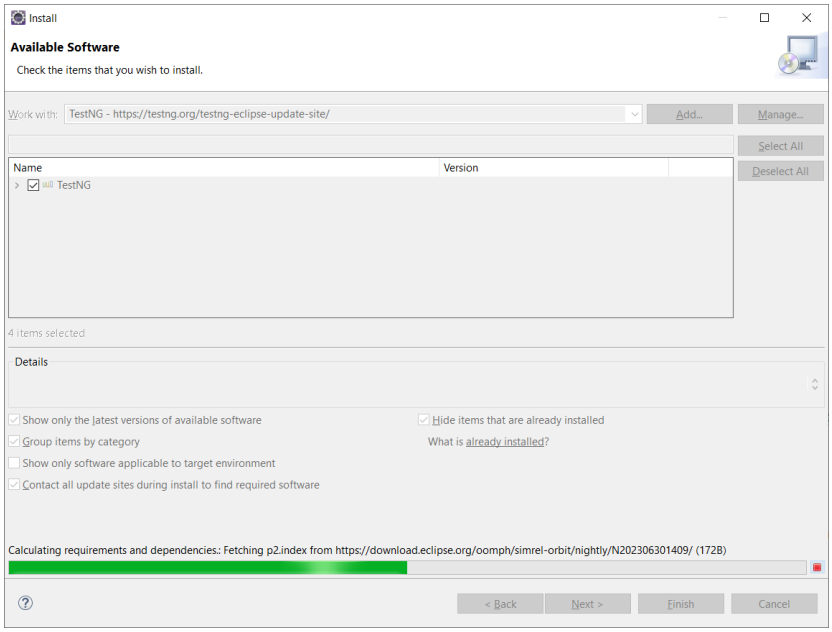



Add Repository
Name:
TestNG
Local...
Location:
https://testng.org/testng-eclipse-update-site/
Archive...
OK
Add
Cancel

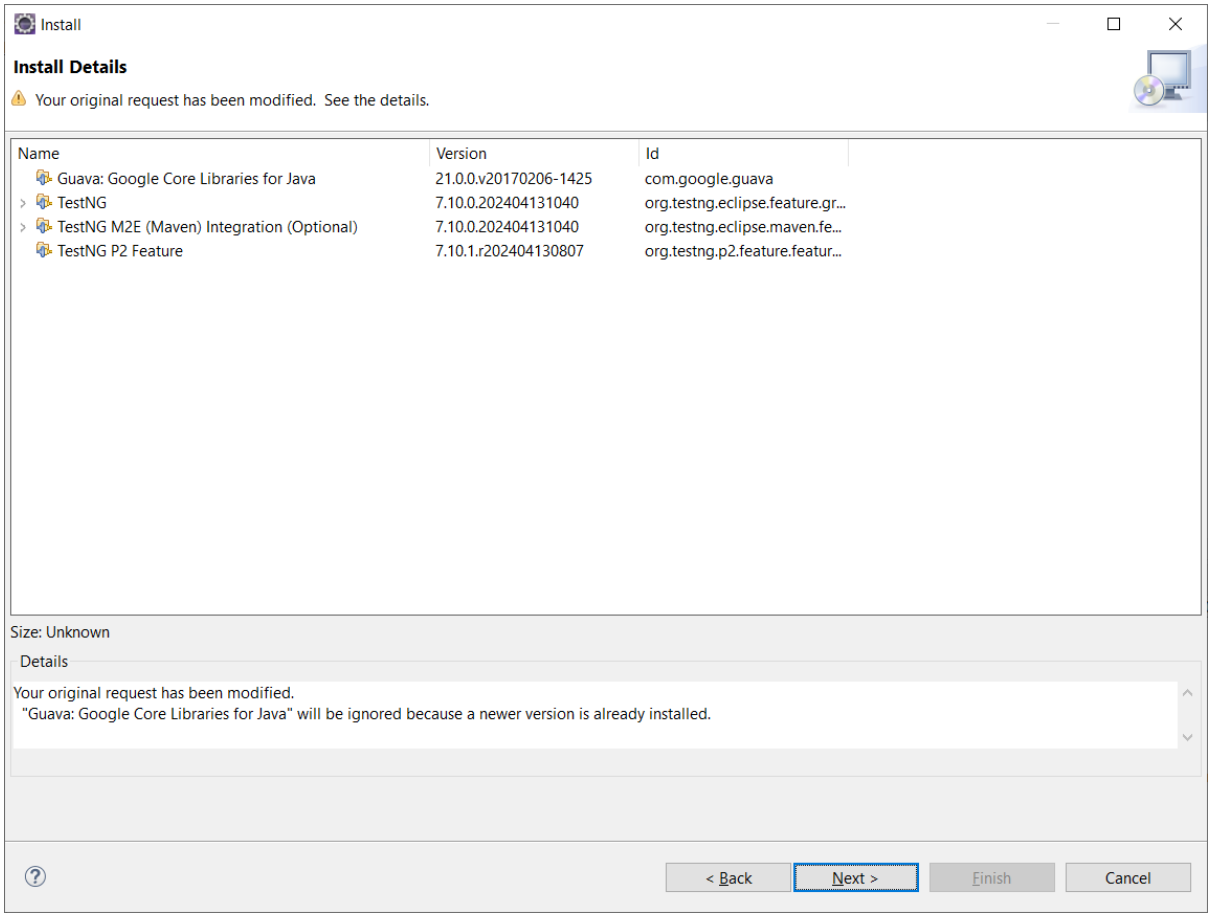

Install
Available Software
Check the items that you wish to install.
Work with:
TestNG - https://testng.org/testng-eclipse-update-site/
Add...
Manage...
type filter text
Select All
Deselect All
Name
Version
Pending...
Details
Show only the latest versions of available software
Group items by category
Show only software applicable to target environment
Contact all update sites during install to find required software
Hide items that are already installed
What is already installed?
Fetching children
< Back
Next >
Finish
Cancel



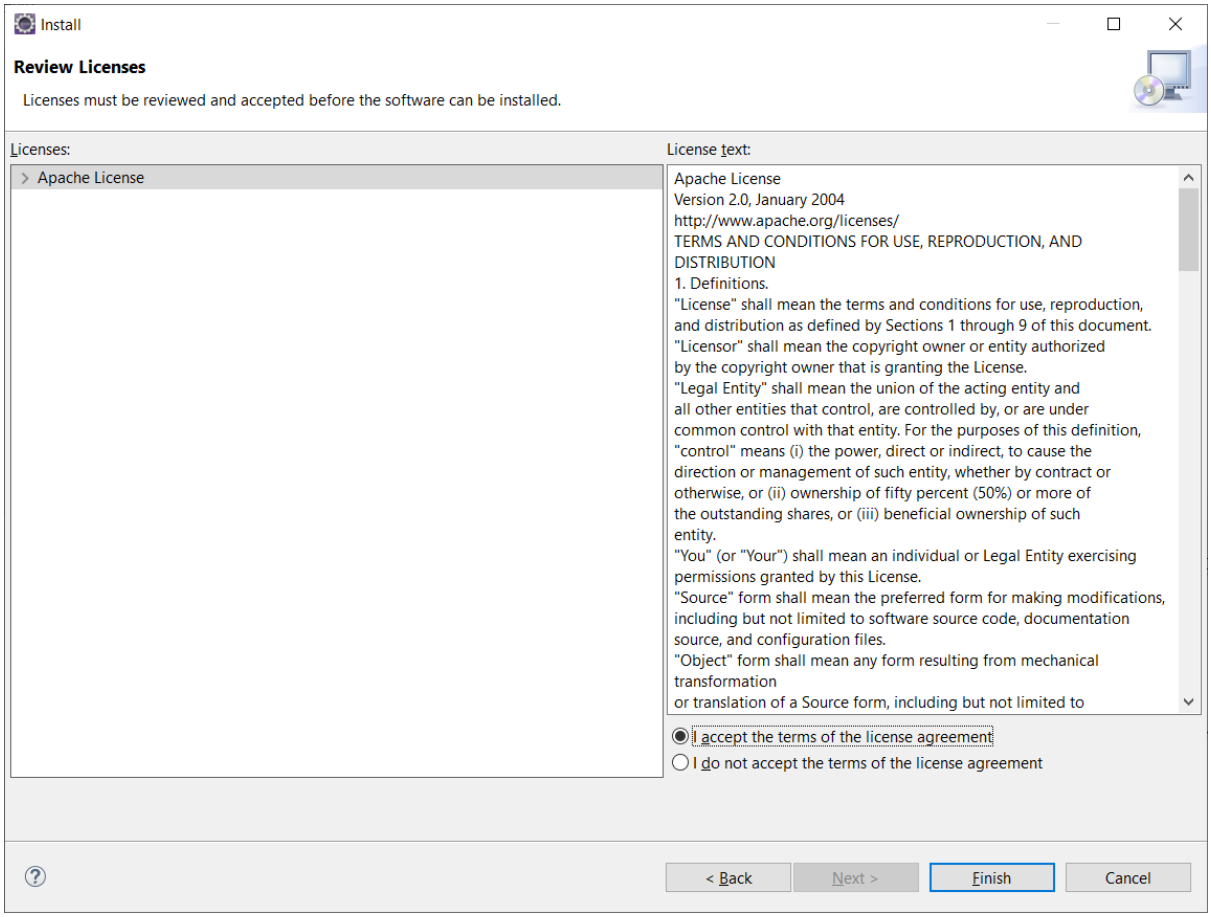
Click Next,

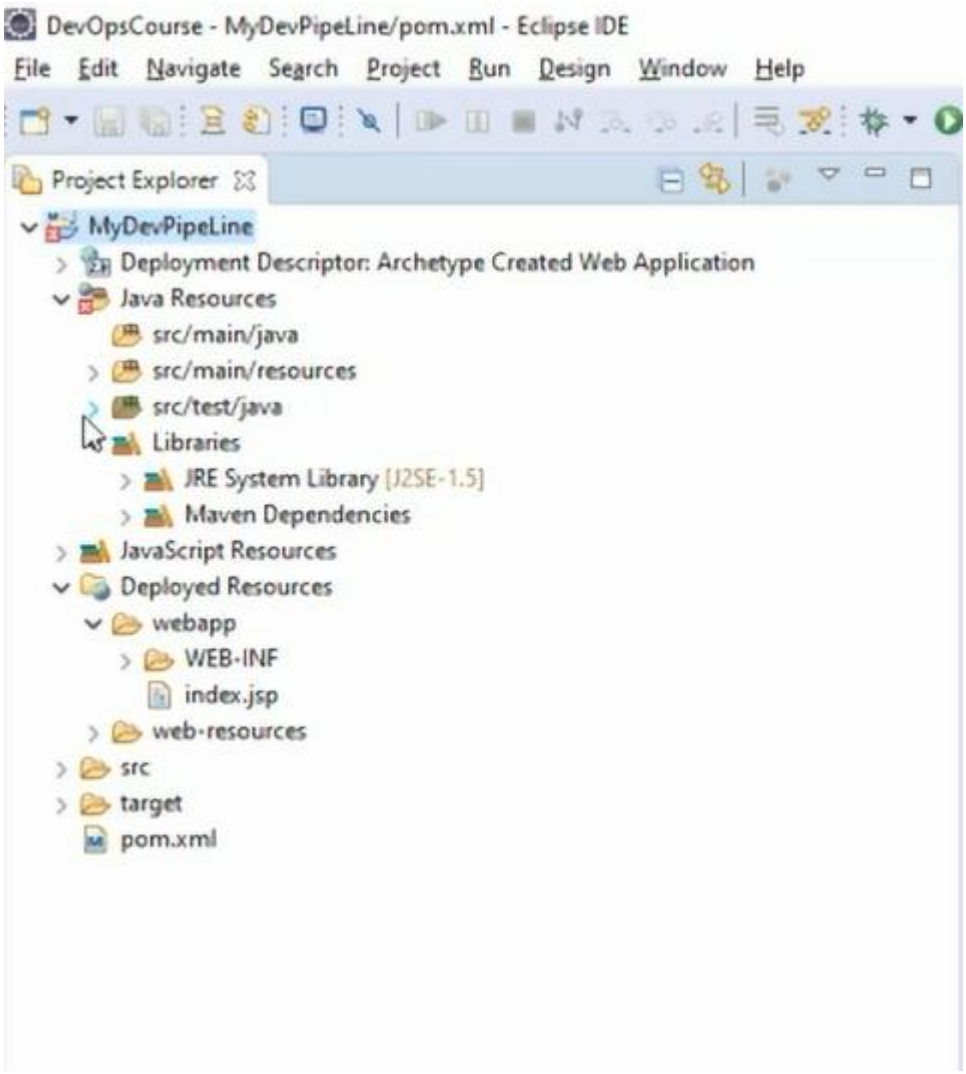


Wait until you get the below;



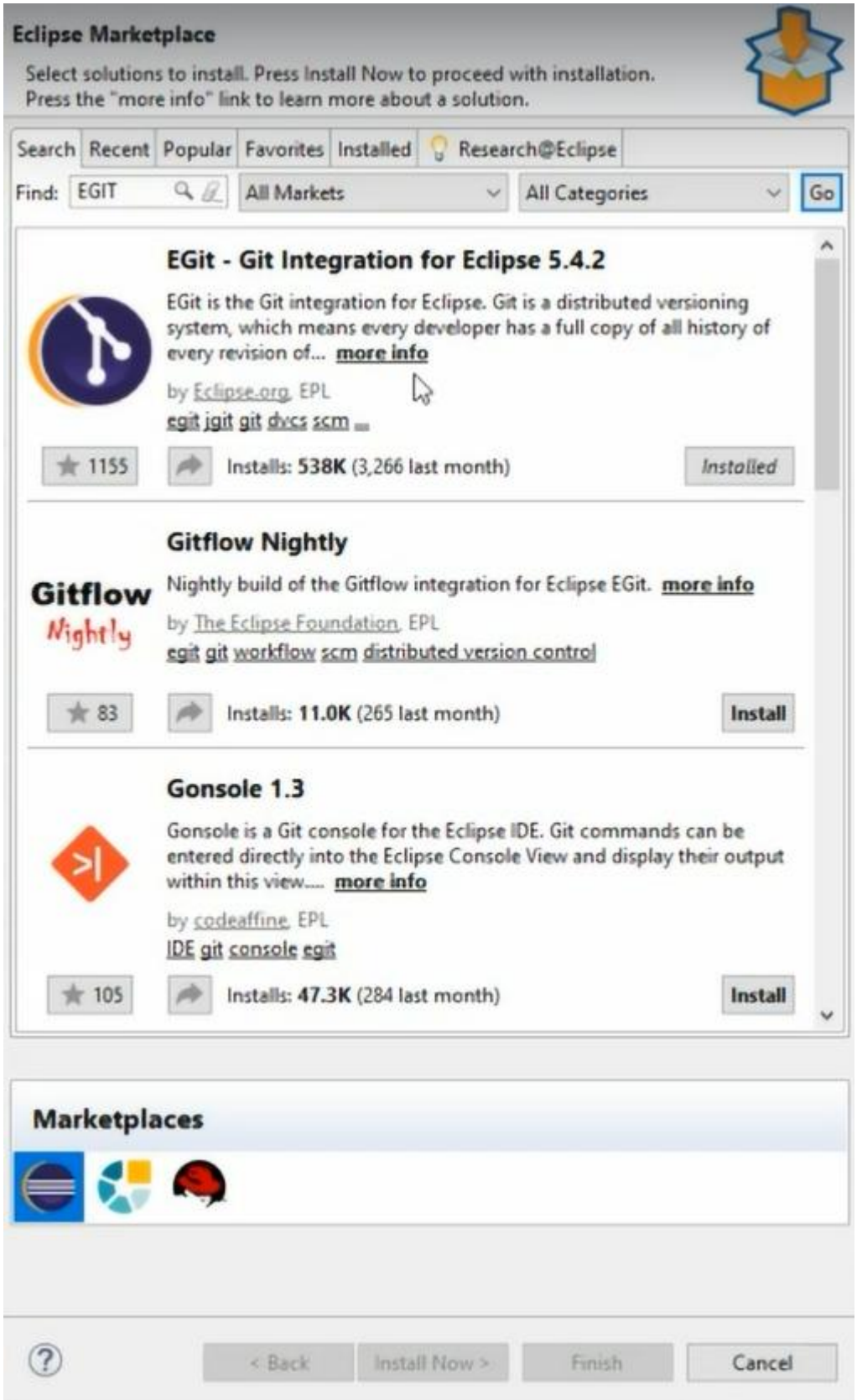
Click Next, and accept the license through the radio button and then click Finish as below:


Install
Review Licenses
Licenses must be reviewed and accepted before the software can be installed.
Licenses:
Apache License
License text:
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/
TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
1. Definitions.
"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.
"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.
"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.
"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.
"Object" form shall mean any form resulting from mechanical transformation
or translation of a Source form, including but not limited to
I accept the terms of the license agreement
I do not accept the terms of the license agreement
< Back
Next >
Finish
Cancel

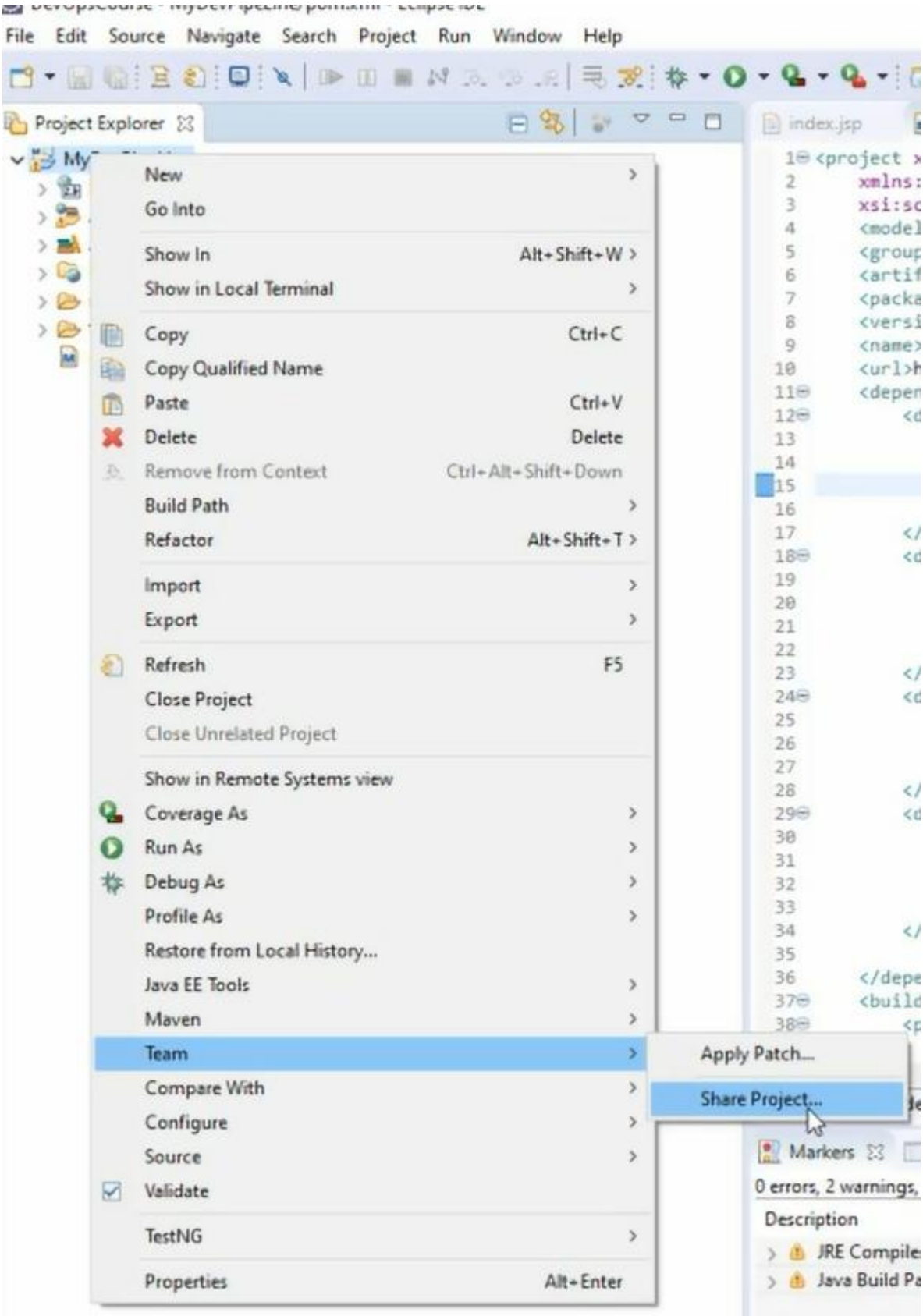


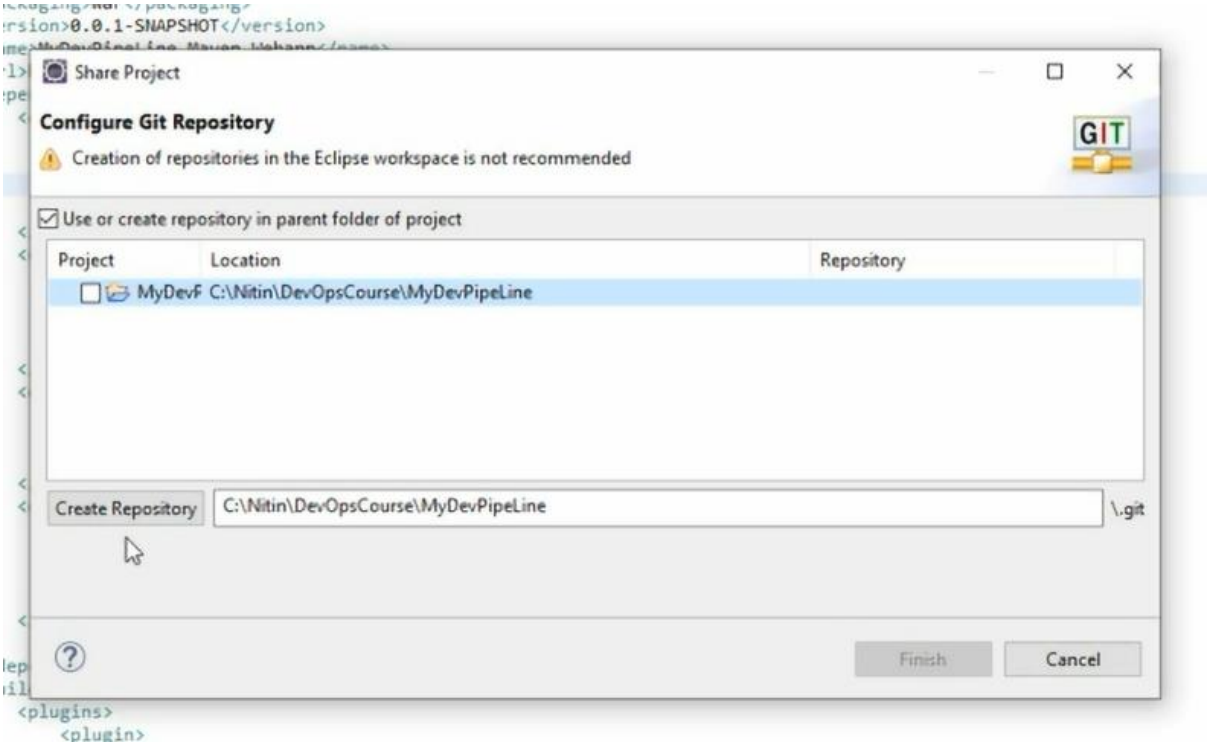
Source code in src/main/java and test cases in src/test/java

Goto Eclipse Marketplace and install EGIT

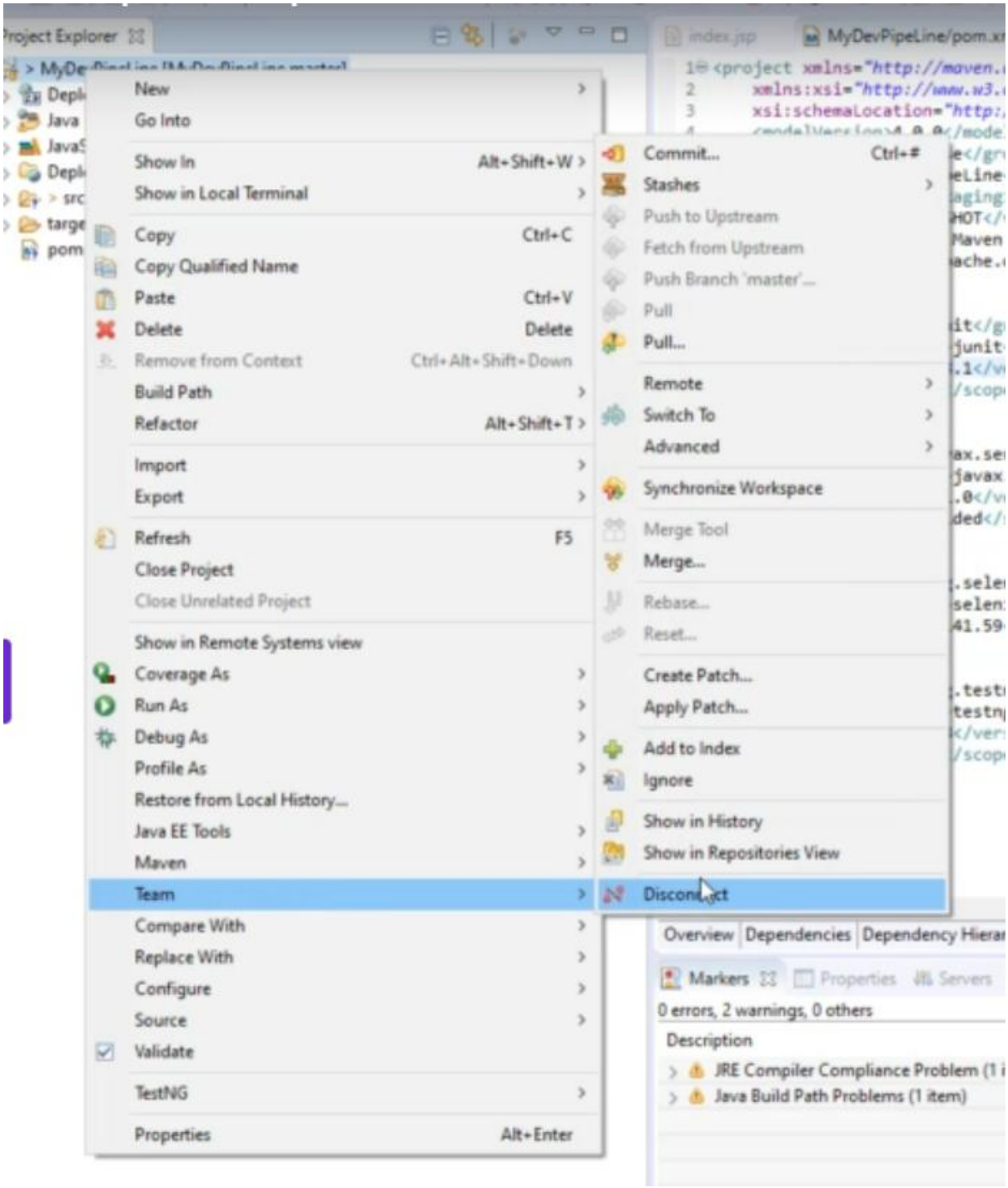


Project-> Team→ share Project

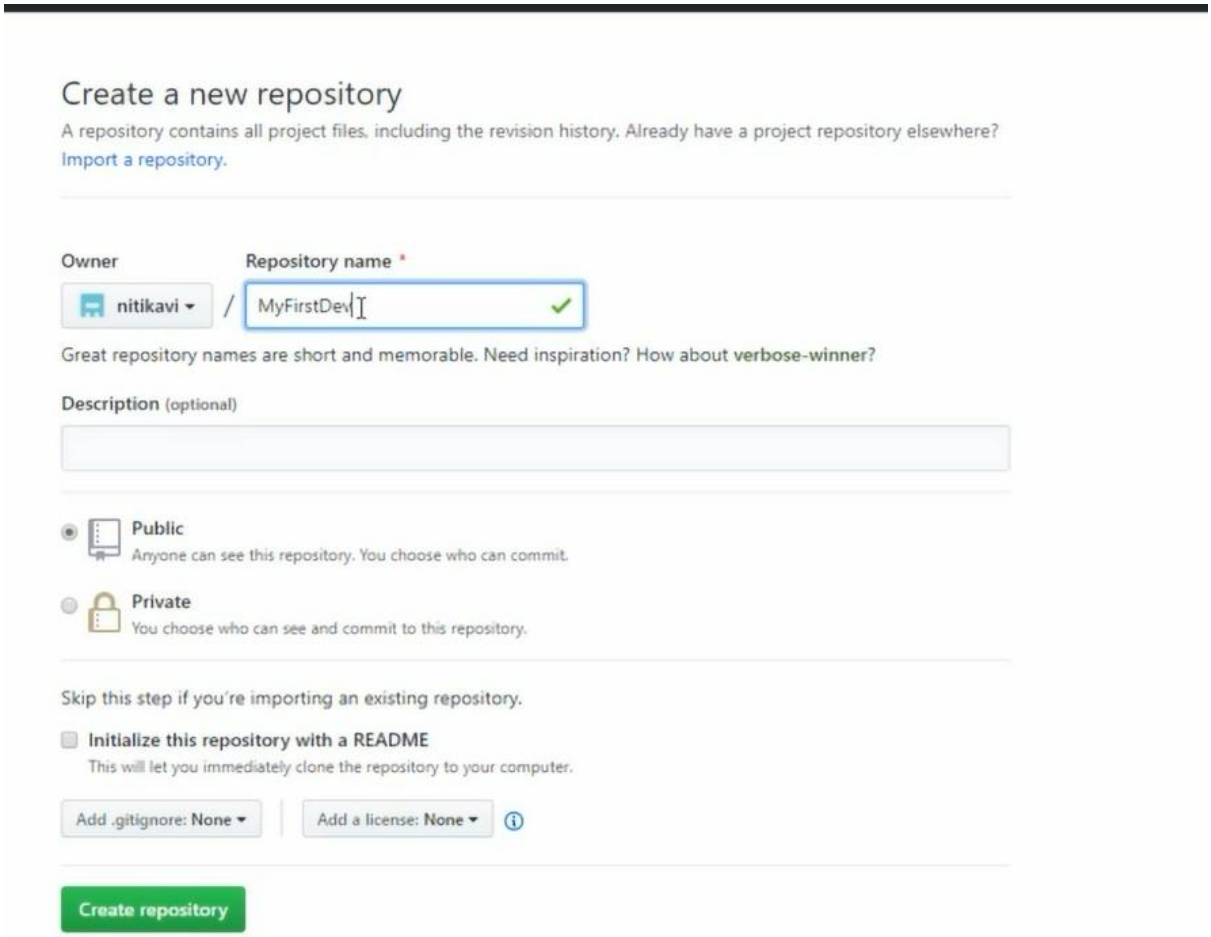

File Edit Source Navigate Search Project Run Window Help
Project Explorer
index.jsp
New
Go Into
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Refactor Alt+Shift+T
Import
Export
Refresh F5
Close Project
Close Unrelated Project
Show in Remote Systems view
Coverage As
Run As
Debug As
Profile As
Restore from Local History...
Java EE Tools
Maven
Team
Compare With
Configure
Source
Validate
TestNG
Properties Alt+Enter
Apply Patch...
Share Project...
Markers
0 errors, 2 warnings,
Description

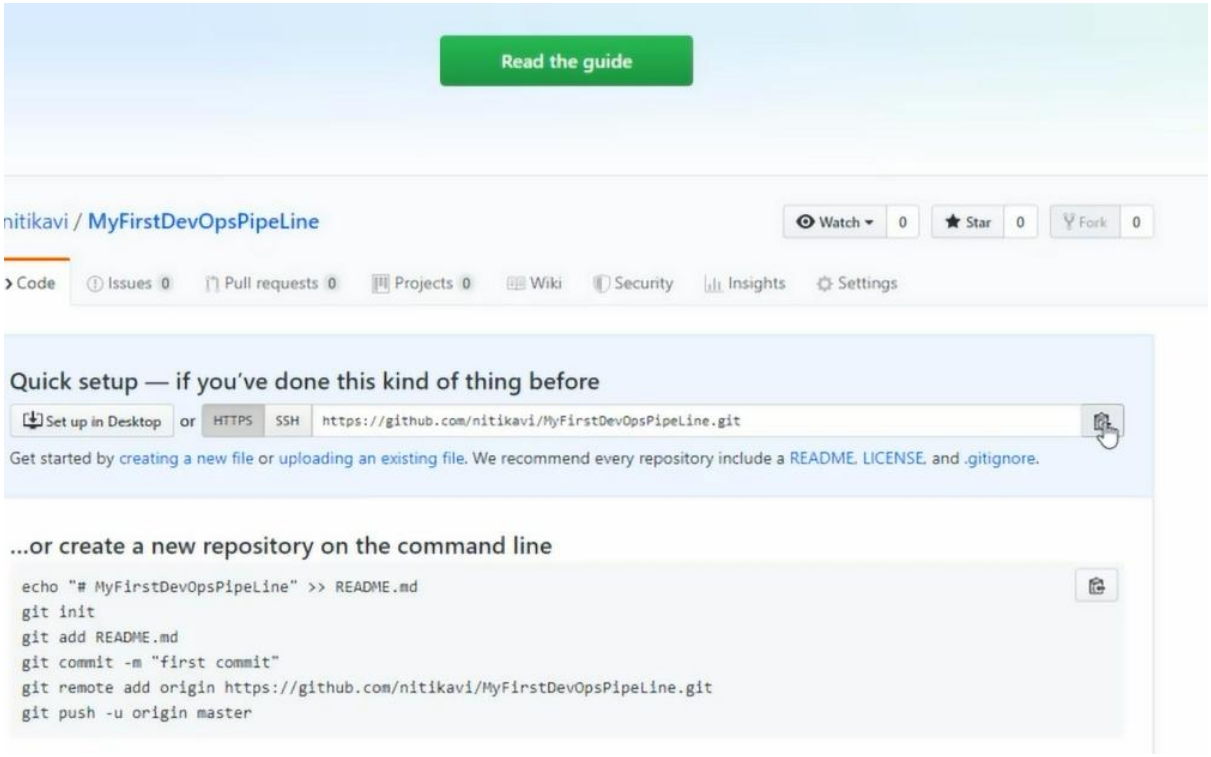


Click Create Repository,

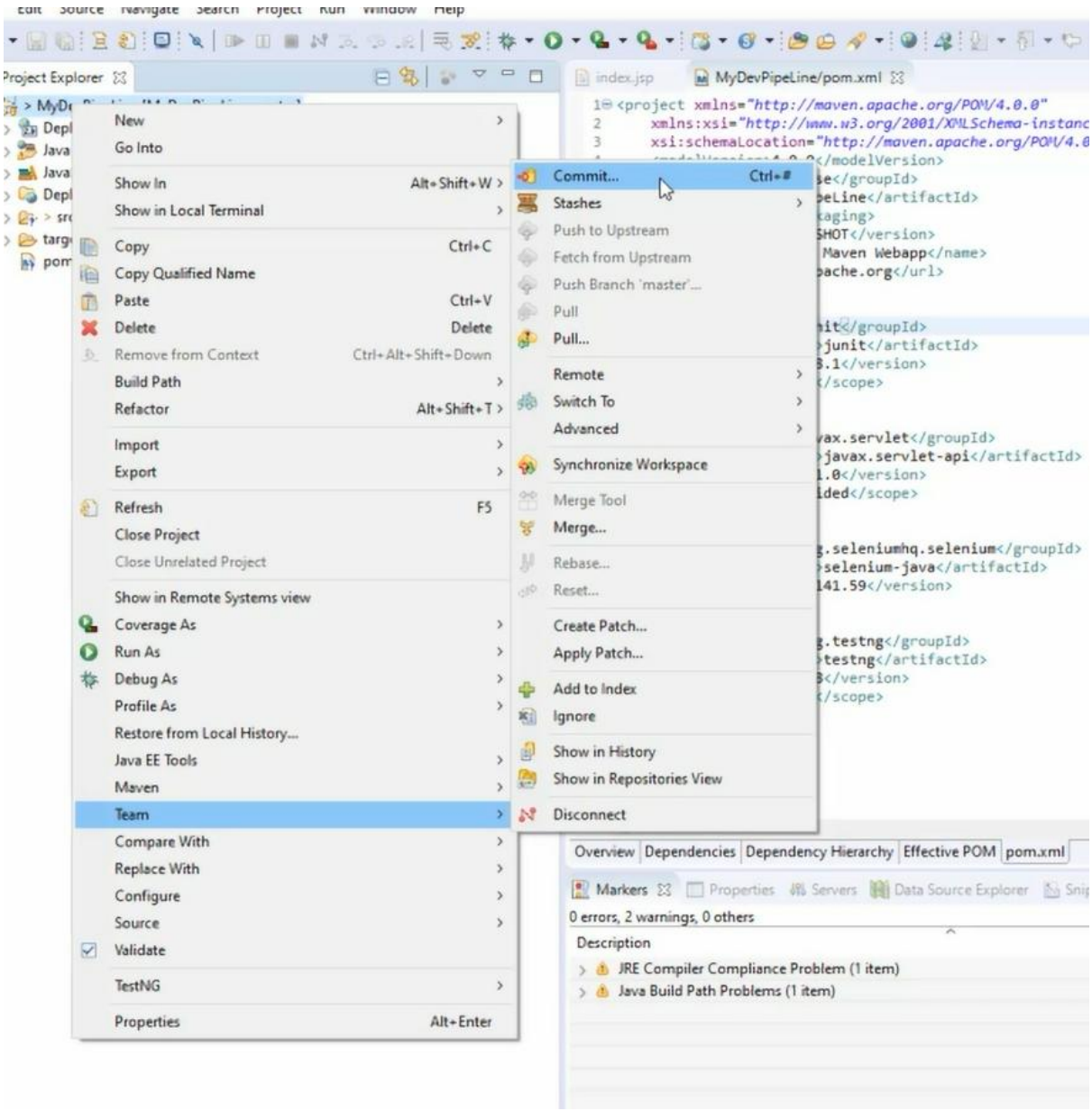


Goto Github and create a repository

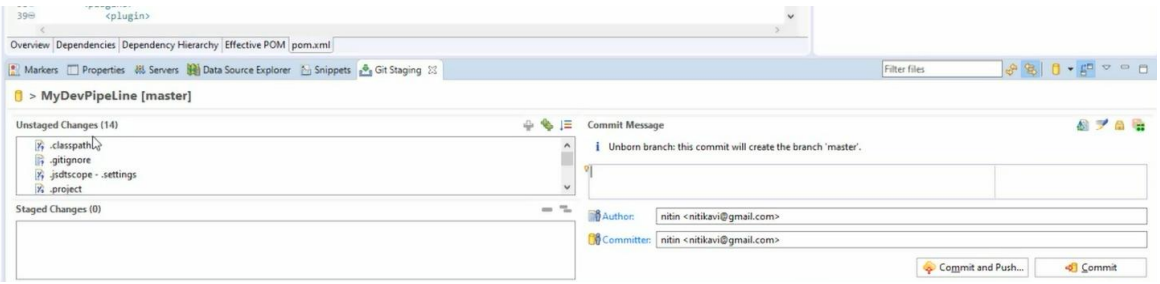

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Owner
Repository name *
nitikavi
MyFirstDev
Great repository names are short and memorable. Need inspiration? How about verbose-winner?
Description (optional)
Public
Anyone can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Skip this step if you're importing an existing repository.
Initialize this repository with a README
This will let you immediately clone the repository to your computer.
Add .gitignore: None
Add a license: None
Create repository


Read the guide
nitikavi / MyFirstDevOpsPipeLine
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Security
Insights
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# MyFirstDevOpsPipeLine" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
git push -u origin master

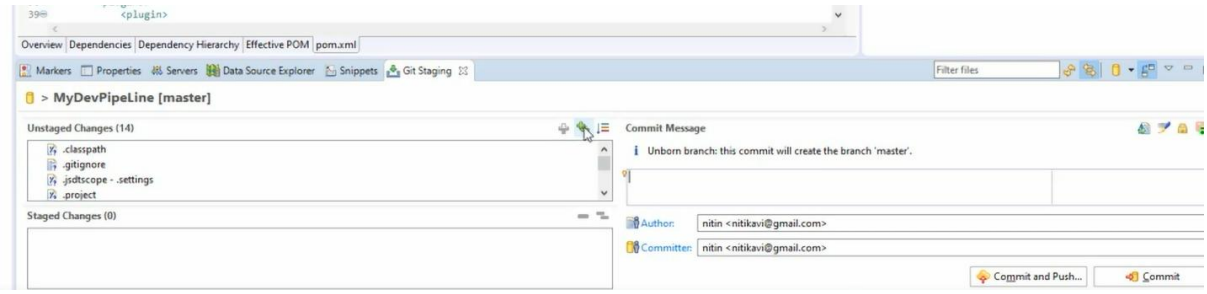
Now, go back to Eclipse and commit,



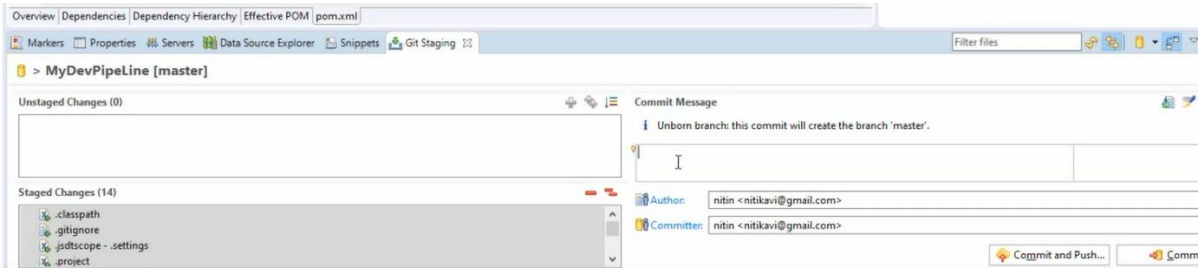
It shows the project is unstaged:



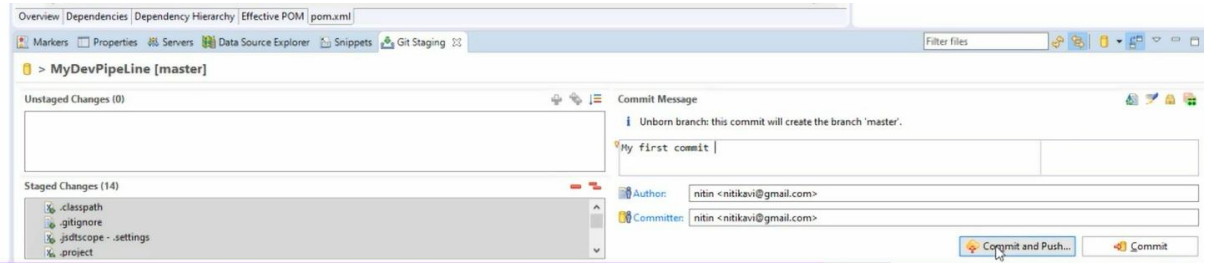
Click on green icon to add files to staging



It looks as below

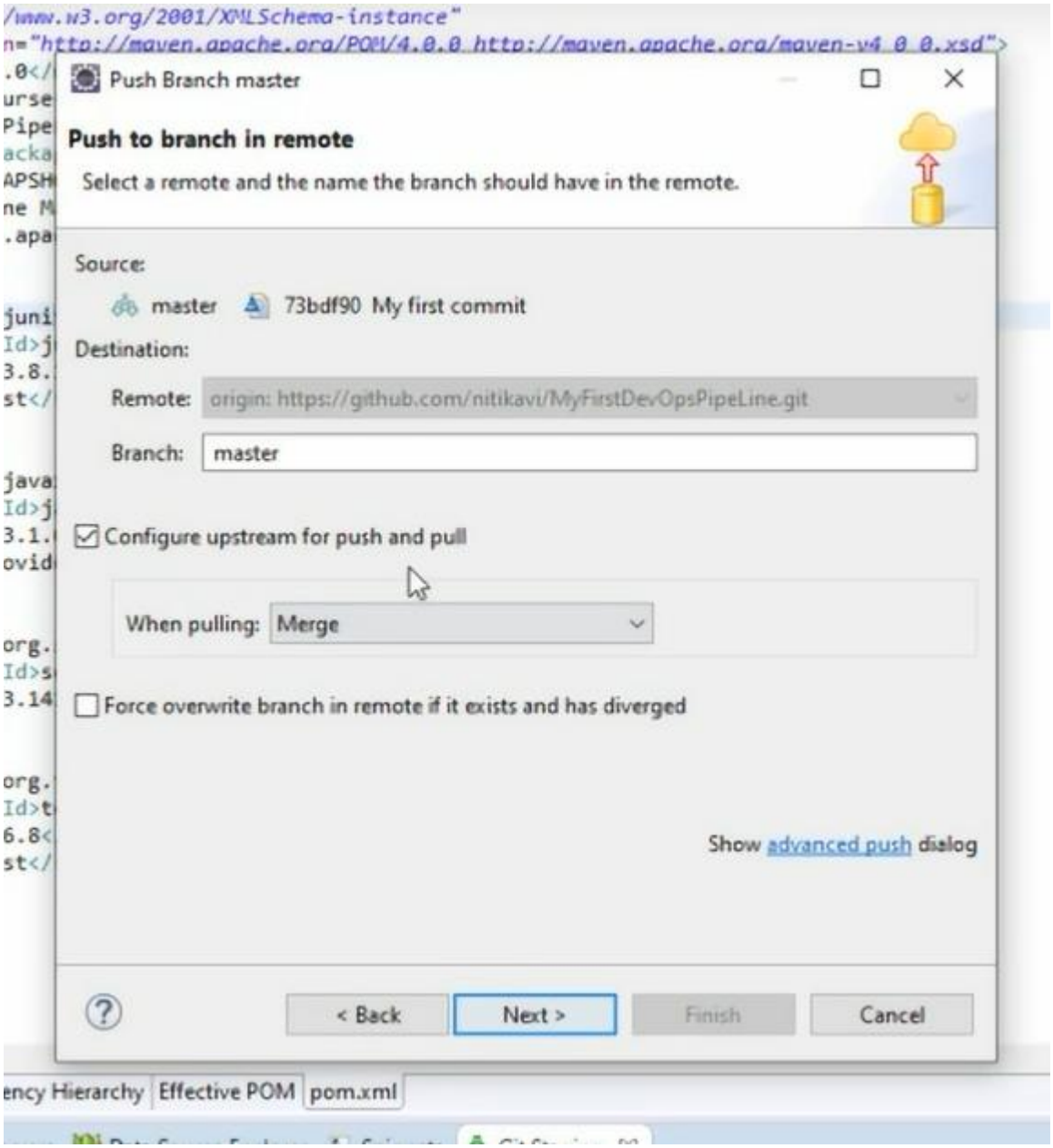


Write a commit message and select Commit and Push.

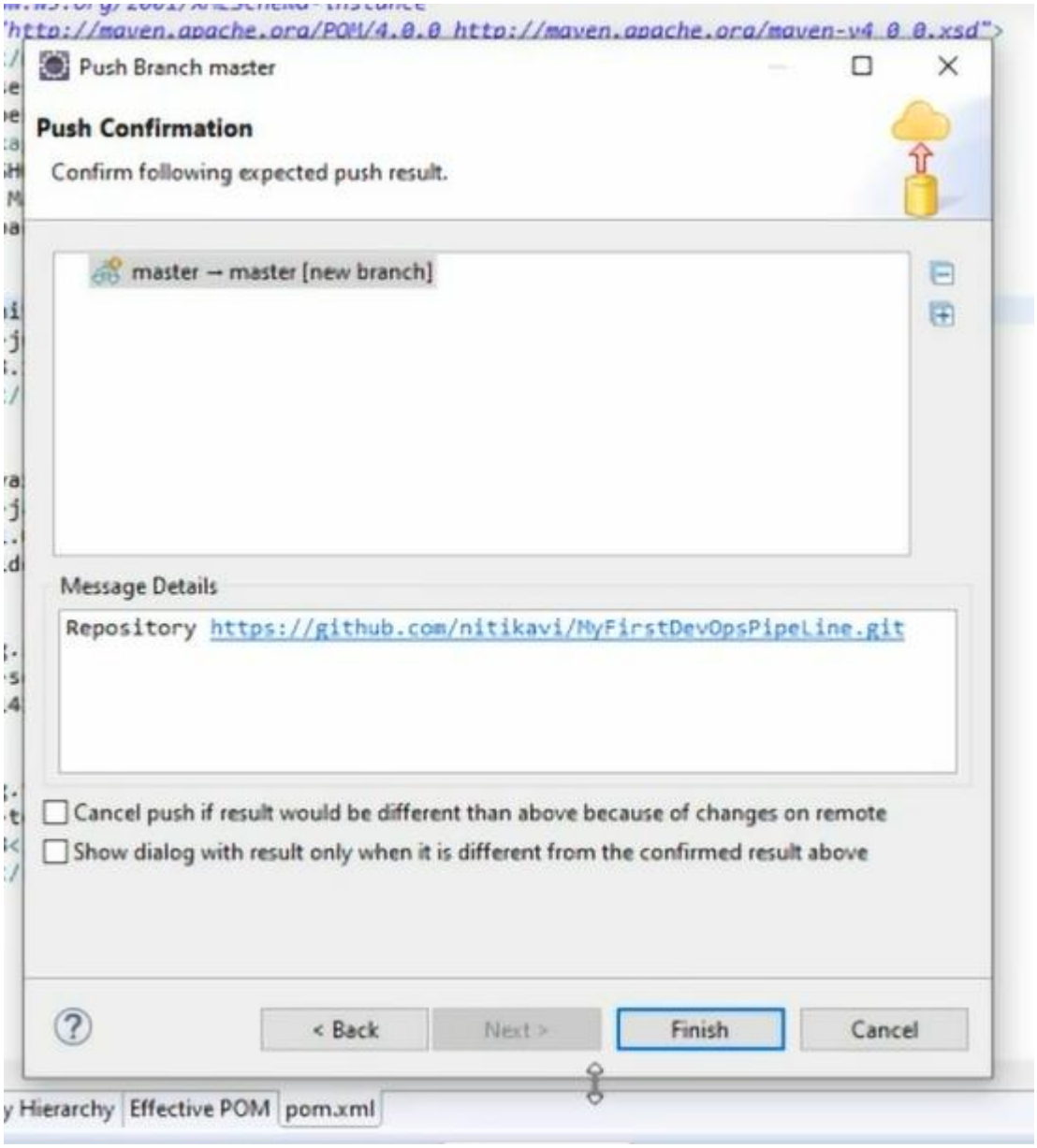


It will ask for repository, automatically pop up otherwise give the URL, username and password for github

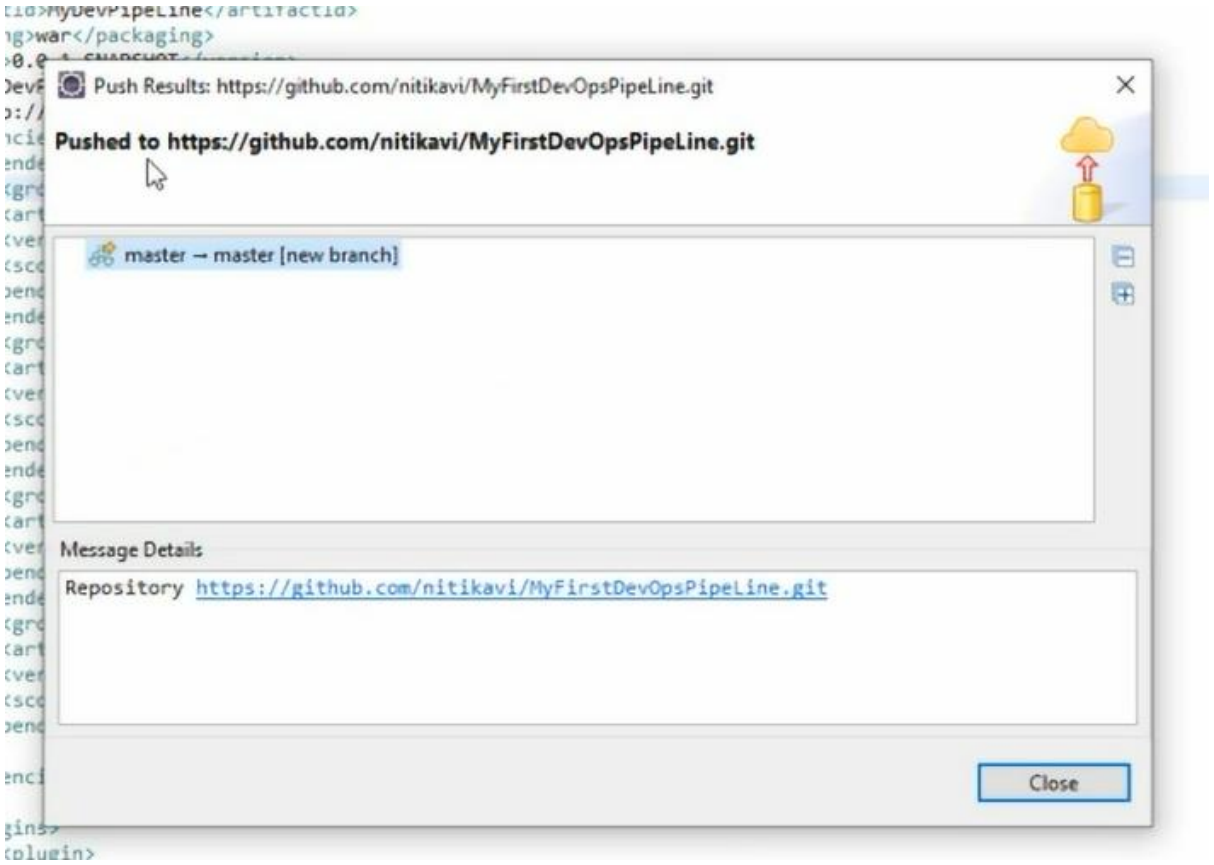
Click Next

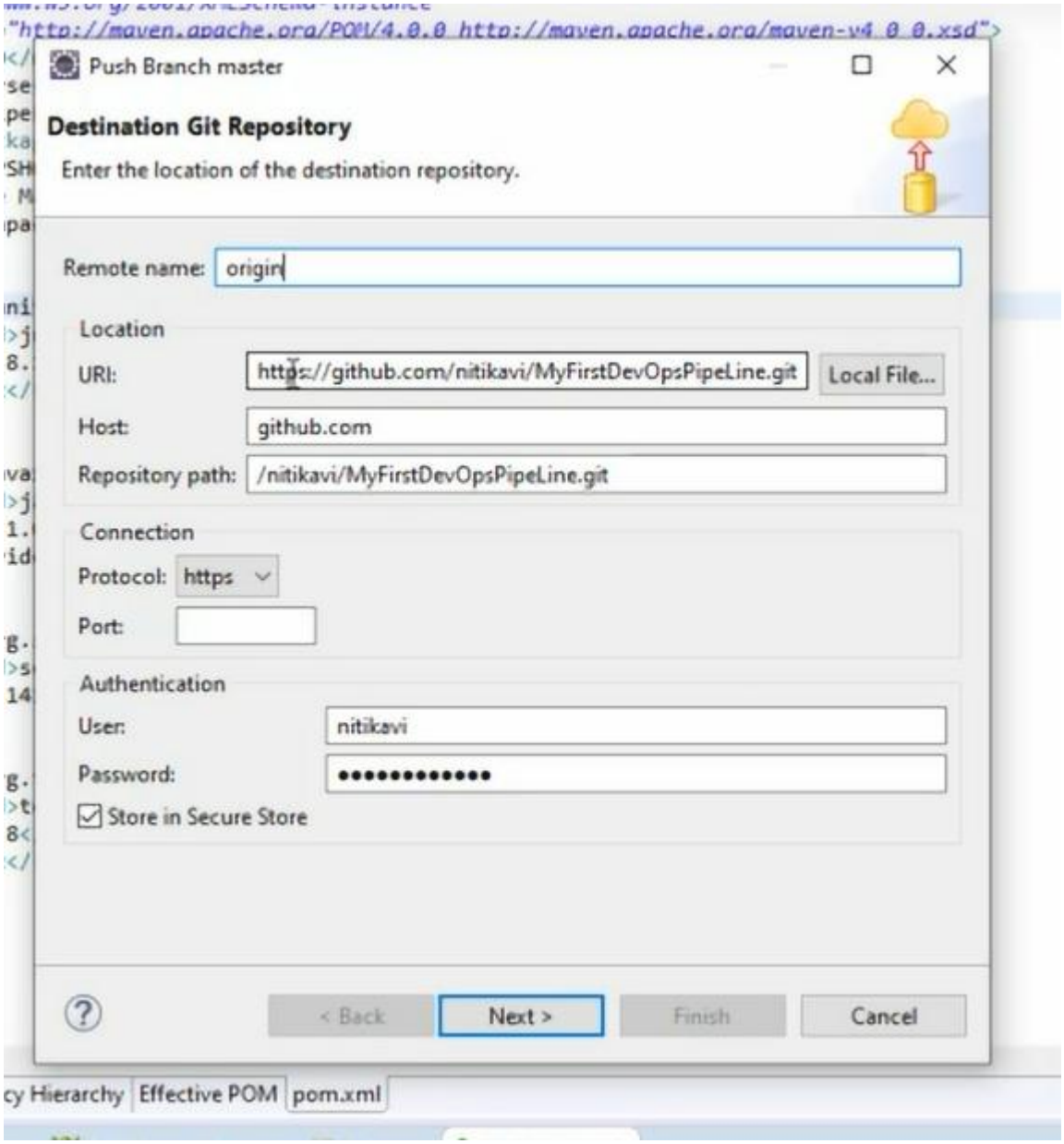

Push Branch master
Push to branch in remote
Select a remote and the name the branch should have in the remote.
Source:
master
73bdf90 My first commit
Destination:
Remote:
origin: https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
Branch:
master
Configure upstream for push and pull
When pulling:
Merge
Force overwrite branch in remote if it exists and has diverged
Show advanced push dialog
< Back
Next >
Finish
Cancel
Effective POM
pom.xml

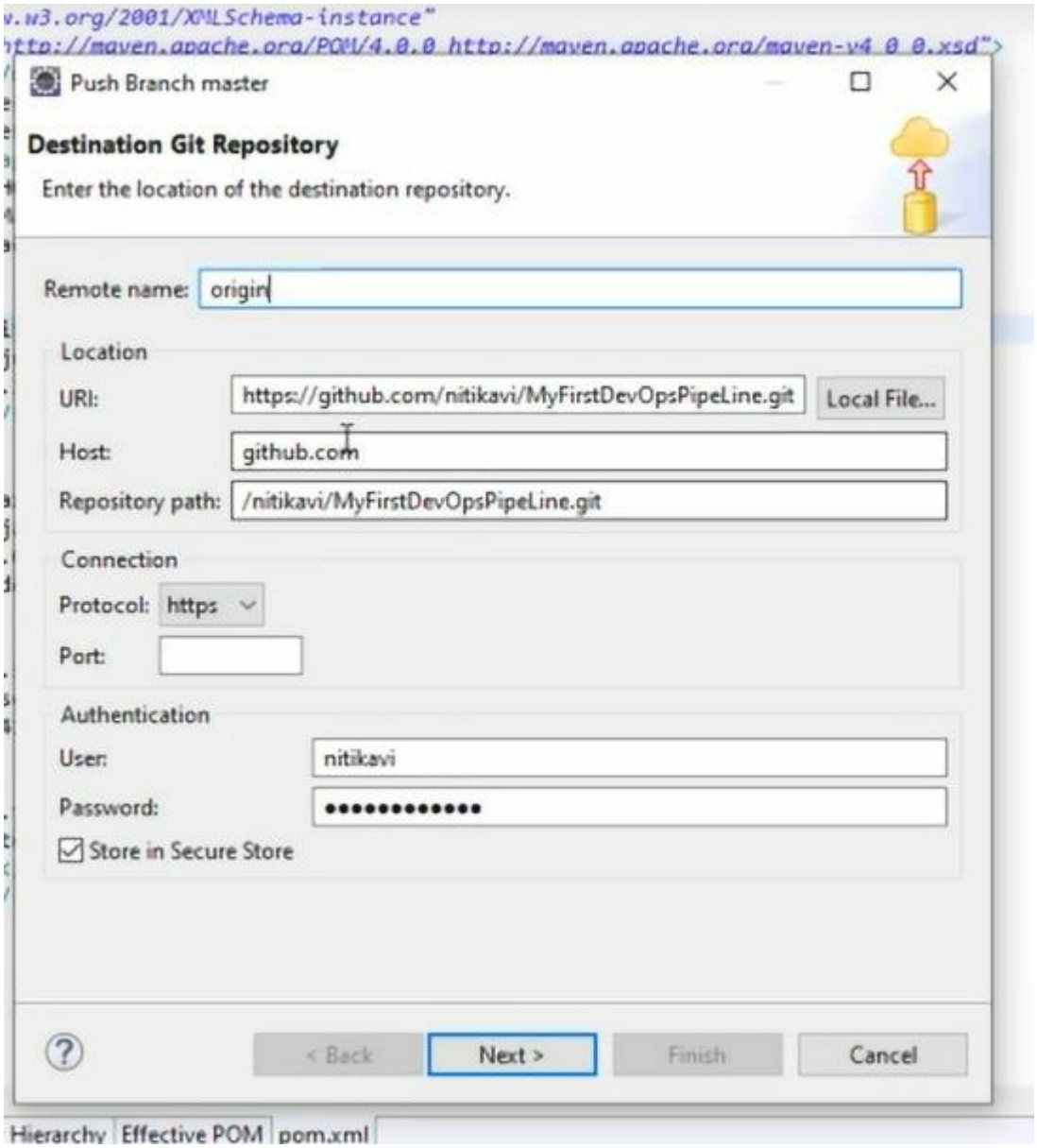
Click Next



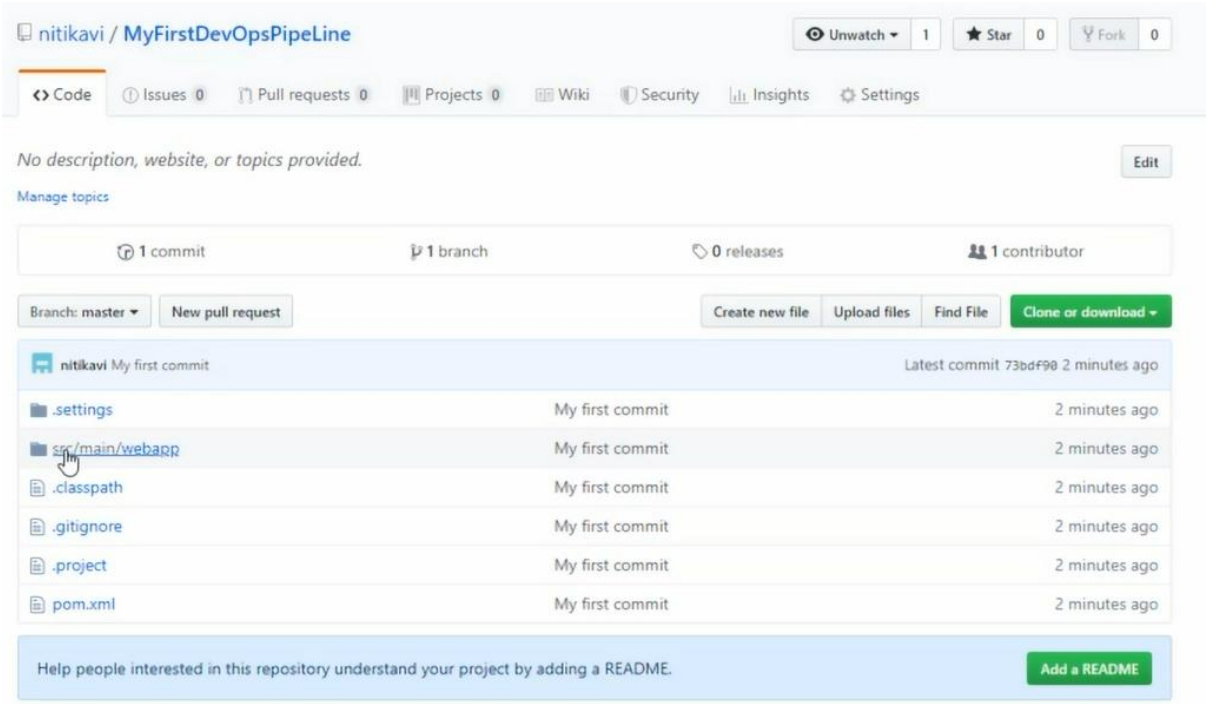
Click Finish


Push Results: https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
Pushed to https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
master → master [new branch]
Message Details
Repository https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
Close

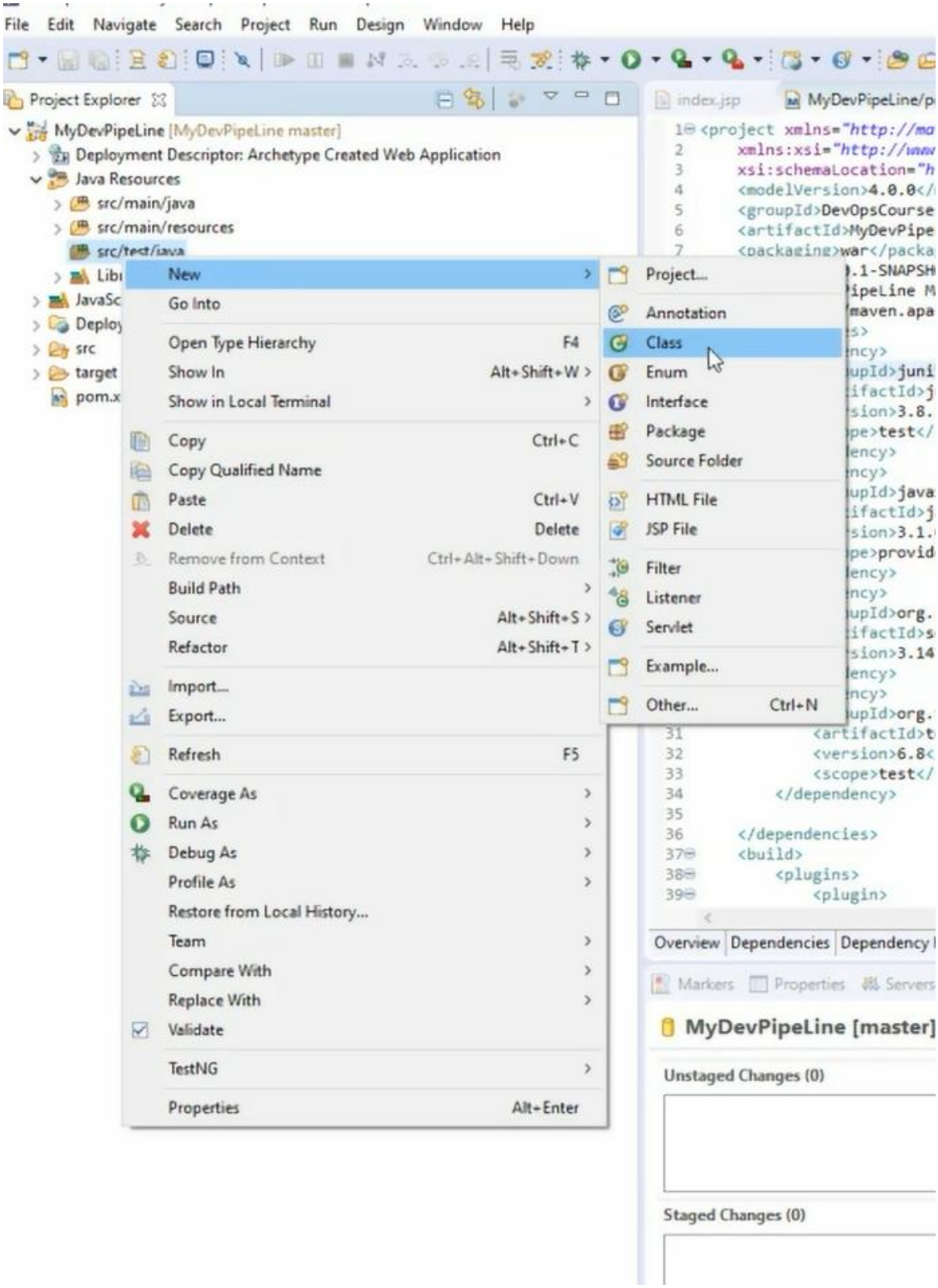

Push Branch master
Destination Git Repository
Enter the location of the destination repository.
Remote name: origin
Location
URI: https://github.com/nitikavi/MyFirstDevOpsPipeLine.git
Local File...
Host: github.com
Repository path: /nitikavi/MyFirstDevOpsPipeLine.git
Connection
Protocol: https
Port:
Authentication
User: nitikavi
Password:
Store in Secure Store
< Back
Next >
Finish
Cancel
cy Hierarchy
Effective POM
pom.xml



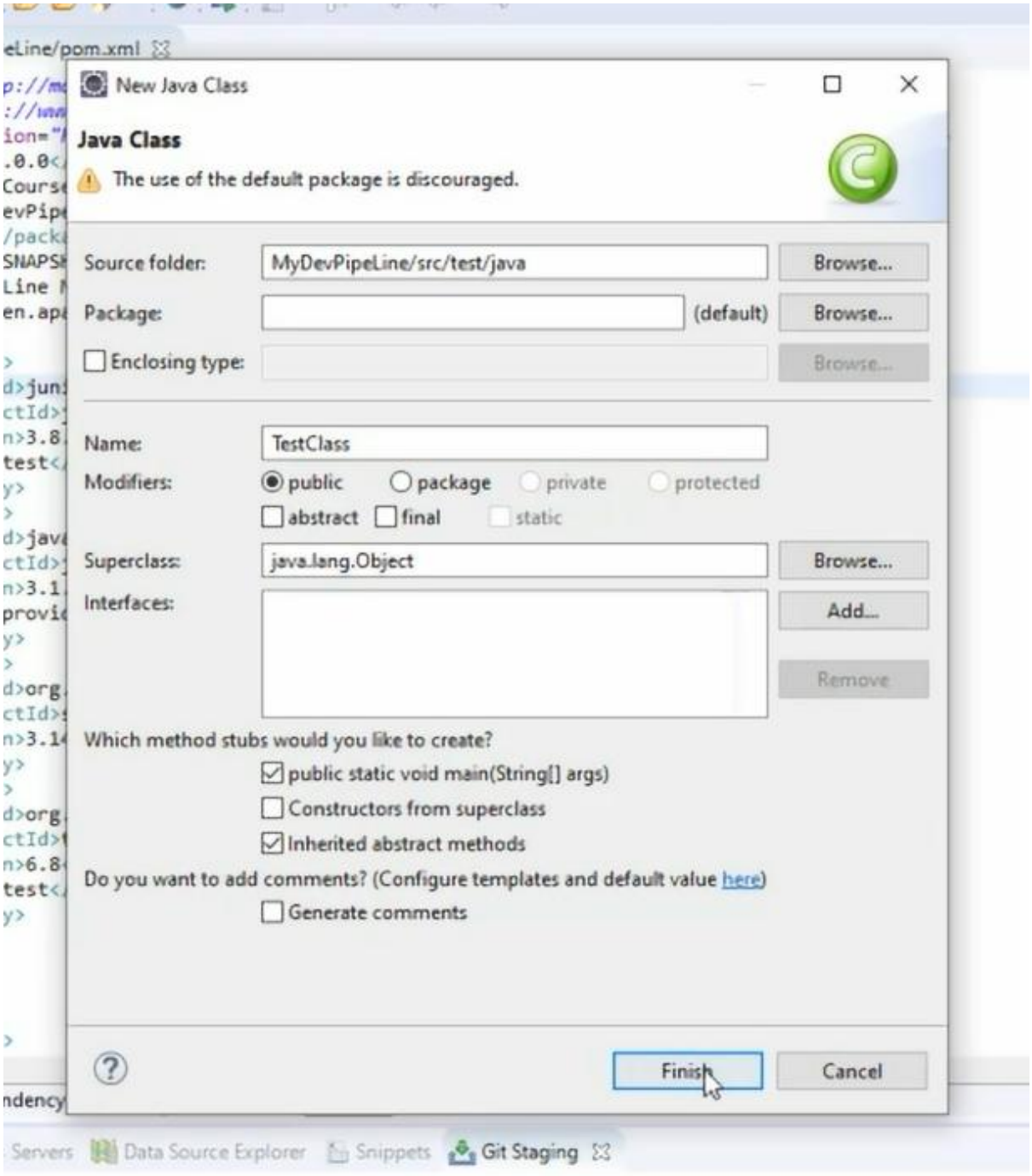
Goto Github and check the repository:



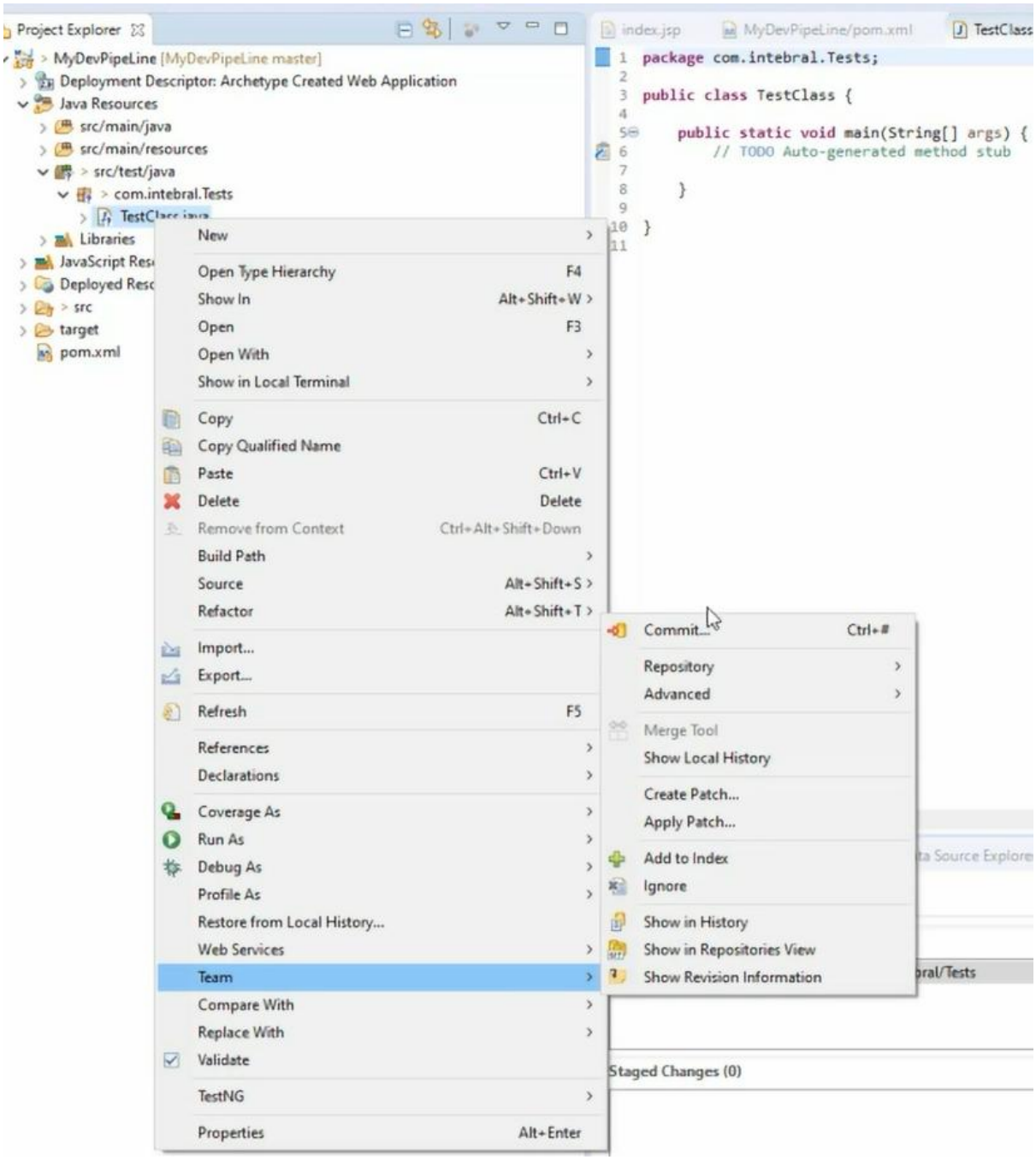
Src/test is not checked in as there are no files, so you can create some testing class as below:



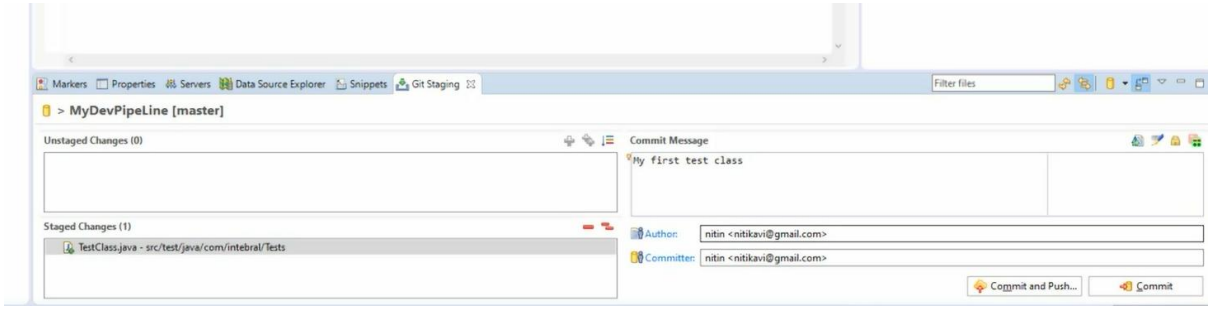
Give a Name and check the checkboc to create publc static void main and click Finish.



Add a package name **com.internal.Tests;**



Add to staging, then give commit message and commit



Note For setting surefire for selenium and Java 1.8 compiler for

Add the below to pom.xml under plugins:

index.jsp *MyDevPipeLine/pom.xml

```
            <version>3.1.0</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>3.141.59</version>
        </dependency>
        <dependency>
            <groupId>org.testng</groupId>
            <artifactId>testng</artifactId>
            <version>6.8</version>
            <scope>test</scope>
        </dependency>

    </dependencies>
    <build>
    <plugins>
       <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.19.1</version>
      </plugin>
       <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>2.3.2</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
</plugins>



        <finalName>MyDevPipeLine</finalName>
    </build>
</project>
```