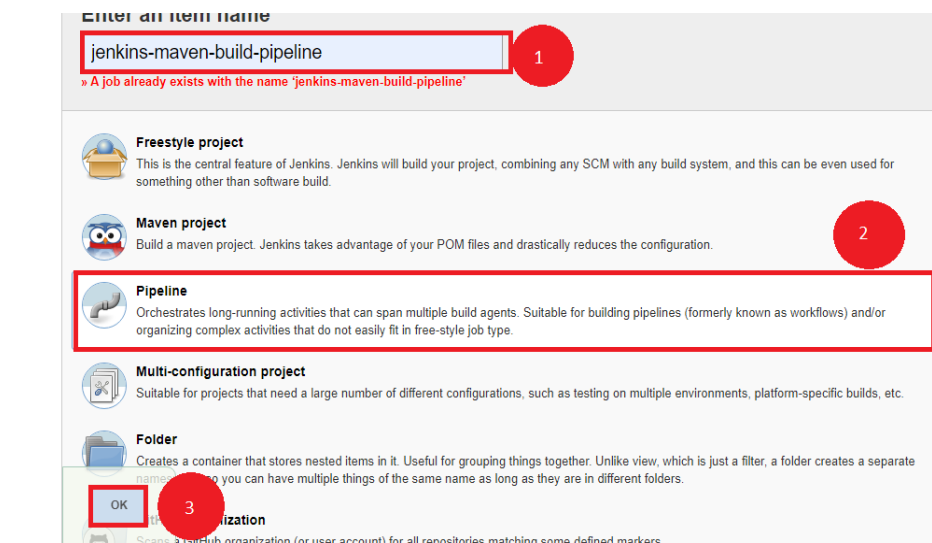## 19. Demonstrate Selenium integration with Jenkins
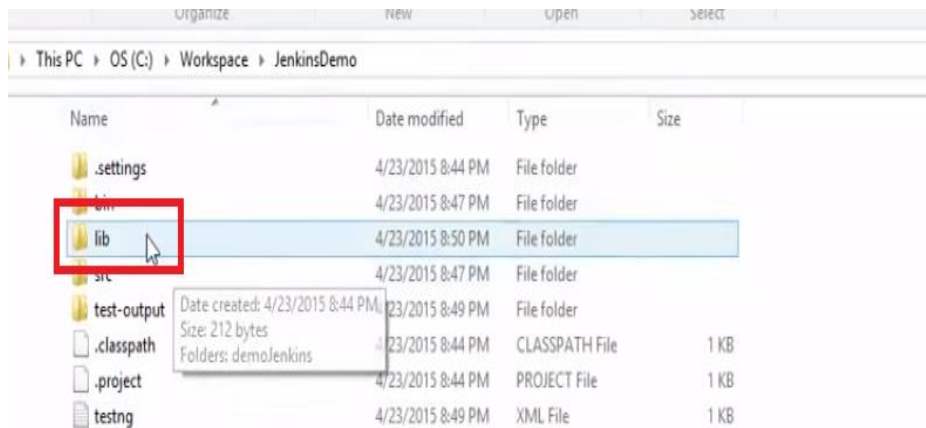
Step 1: Creating a Jenkins Pipeline job

- Java 1.8 is already installed in your practice lab.

- Jenkins.war file is already present in your practice lab in directory /usr/share/jenkins.

- Go to the jenkins.war location. Now, start Jenkins by using the following command on command prompt: **java -jar jenkins.war.**

- Open your browser and type- **localhost:8080.**

- Enter the password.

- Create a job.

- Enter the name.
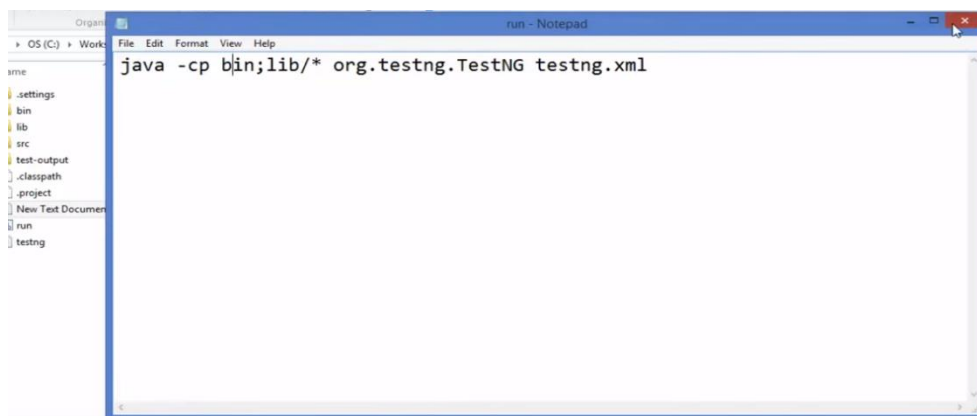
- Select the **Pipeline.**

- Now, click on OK.

Step 2: Integrating Selenium WebDriver with Jenkins

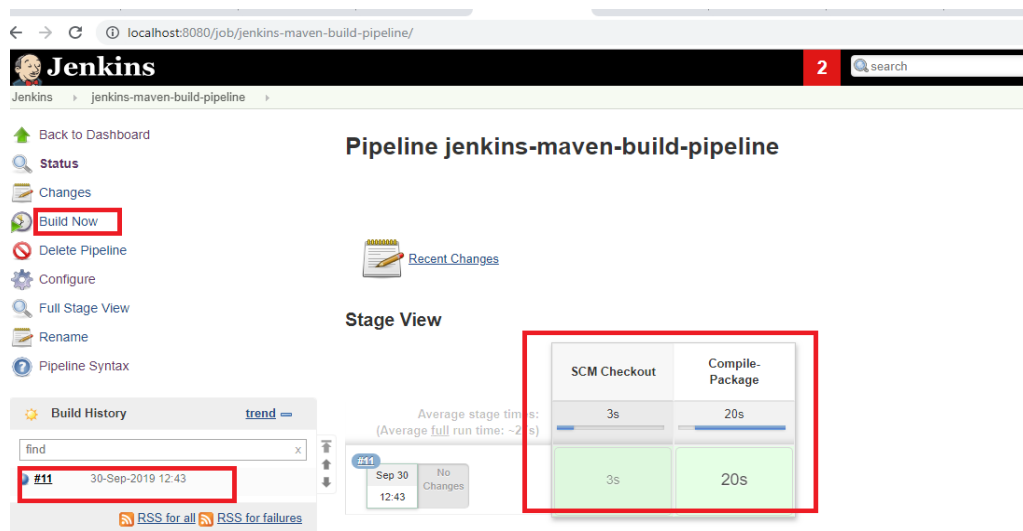- Go to the Selenium script location.

- Add all the Selenium libraries.

- Create a text file **run.bat** and keep it within double quotation marks.


```
java -cp bin;lib/* org.testng.TestNG testng.xml
```

- Go to the Jenkins Pipeline job.

- Pass the Selenium script location.

- Write a groovy script in Pipeline.

- Click on Apply and then Save.


```
1  node {
2      def mvnHome
3      stage('Preparation') { // for display purposes
4          // Get some code from a GitHub repository
5          git 'https://github.com/jglick/simple-maven-project-with-tests.git'
6          // Get the Maven tool.
7          // ** NOTE: This 'M3' Maven tool must be configured
8          // **       in the global configuration.
9          mvnHome = tool 'M3'
10     }
11     stage('Build') {
12         // Run the maven build
13         withEnv(["MVN_HOME=$mvnHome"]) {
14             if (isUnix()) {
15                 sh '"$MVN_HOME/bin/mvn" -Dmaven.test.failure.ignore clean package'
16             } else {
17                 bat(/"%MVN_HOME%\bin\mvn" -Dmaven.test.failure.ignore clean package)/
```

☑ Use Groovy Sandbox

Pipeline Syntax

Save   **Apply**

● Now, click on **Build Now.**

Step 3: Pushing the code to 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