

Integration of Jenkins with Maven & GitHub

~By Neha Singh

GitHub is a multinational corporation that provides hosting for software development and version control using Git.

Prerequisites

- Maven Installed, running
- Jenkins running

In this module, we will see how to integrate Jenkins with Maven and GitHub.

Make sure your Jenkins is up and running.

```
C:\Windows\System32\cmd.exe - java -jar jenkins.war
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Neha\Downloads>java -jar jenkins.war
Running from: C:\Users\Neha\Downloads\jenkins.war
webroot: $user.home/.jenkins
2020-10-08 14:11:54.804+0000 [id=1] INFO org.eclipse.
2020-10-08 14:11:55.382+0000 [id=1] INFO winstone.Log
2020-10-08 14:11:55.580+0000 [id=1] WARNING o.e.j.s.hand
2020-10-08 14:11:55.847+0000 [id=1] INFO org.eclipse
```

Open jenkins in web browser, I will be using Chrome
“Localhost:8080”

On left-hand side, move to manage Jenkins.

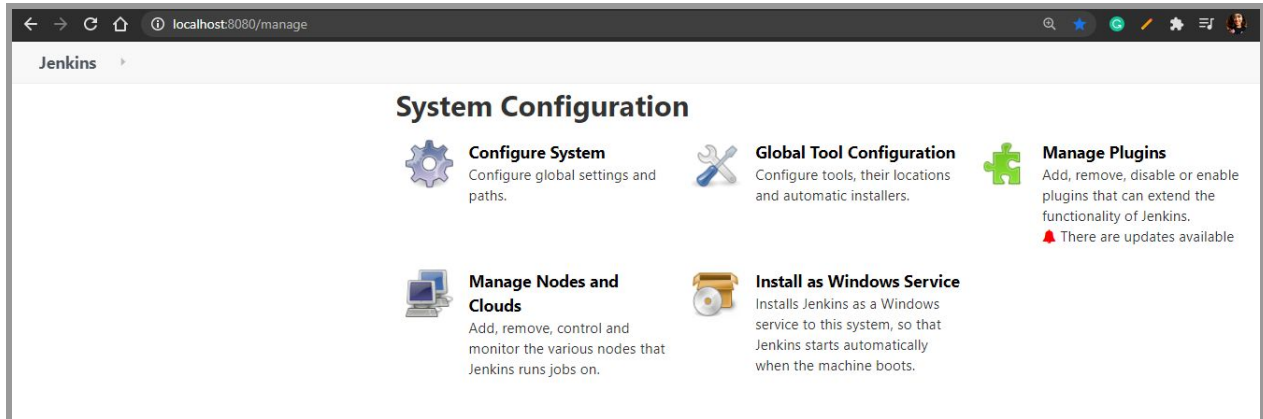
The screenshot shows the Jenkins web interface in a Chrome browser at localhost:8080. The left sidebar contains navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (highlighted), My Views, Open Blue Ocean, Lockable Resources, and New View. The main content area displays a table of build jobs. The table has columns for status (S), weather icon (W), name, last success, last failure, last duration, and favorite (Fav). The jobs listed are BlueOcean, Blueocean hands on, CI_CD_LAB1, CI_CD_SonarQube, Ocean, test2_Jenkinsfile, and test_jenkinsfile. At the bottom, there is a 'Build Queue' section showing 'No builds in the queue.' and a legend for Atom feeds.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		BlueOcean	15 days - log	N/A	3.6 sec	
		Blueocean hands on	13 days - log	N/A	5.4 sec	
		CI_CD_LAB1	1 mo 13 days - #2	1 mo 13 days - #1	35 sec	
		CI_CD_SonarQube	1 mo 7 days - #3	1 mo 7 days - #2	1 min 43 sec	
		Ocean	15 days - log	N/A	3.5 sec	
		test2_Jenkinsfile	N/A	1 mo 6 days - #2	33 sec	
		test_jenkinsfile	1 mo 6 days - #2	N/A	24 sec	

Icon: [S](#) [M](#) [L](#)

Legend [Atom feed for all](#) [Atom feed for failures](#) [Atom feed for just latest builds](#)

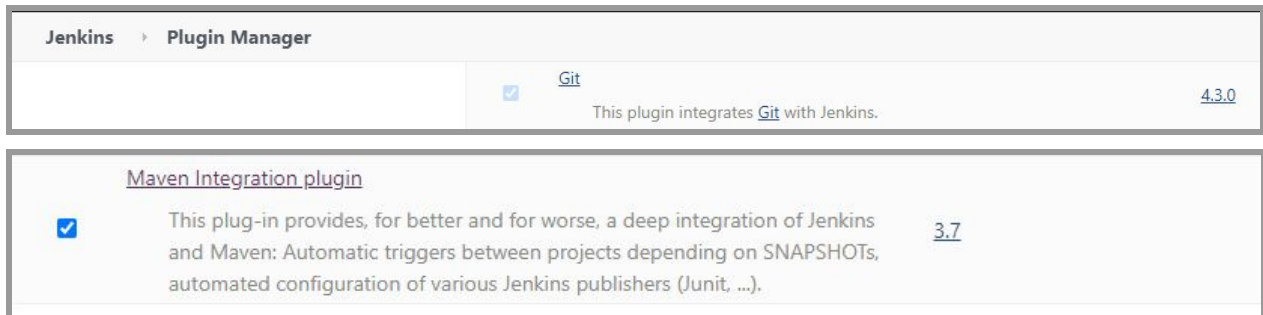
Manage Plugins



The screenshot shows the Jenkins 'System Configuration' page in a web browser. The browser's address bar displays 'localhost:8080/manage'. The page title is 'Jenkins'. The main heading is 'System Configuration'. Below this heading, there are five configuration options, each with an icon and a description:

- Configure System** (gear icon): Configure global settings and paths.
- Global Tool Configuration** (wrench icon): Configure tools, their locations and automatic installers.
- Manage Plugins** (puzzle piece icon): Add, remove, disable or enable plugins that can extend the functionality of Jenkins. A red bell icon indicates 'There are updates available'.
- Manage Nodes and Clouds** (laptop icon): Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Install as Windows Service** (CD icon): Installs Jenkins as a Windows service to this system, so that Jenkins starts automatically when the machine boots.

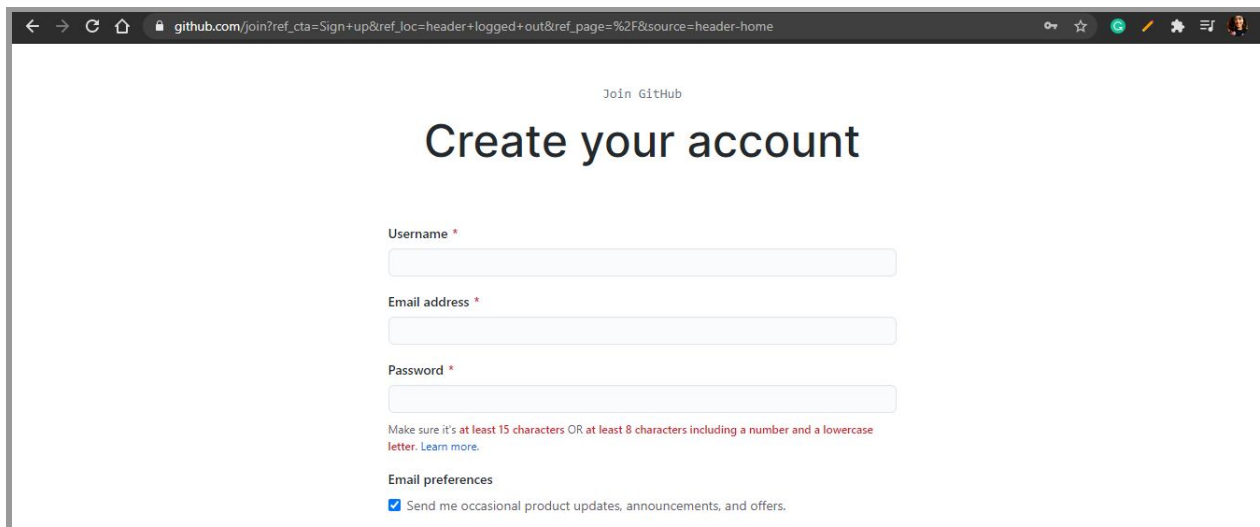
Now, you can install **Maven, Github Plugins**



The screenshot shows the Jenkins 'Plugin Manager' page. The breadcrumb navigation shows 'Jenkins' > 'Plugin Manager'. The page lists installed and available plugins:

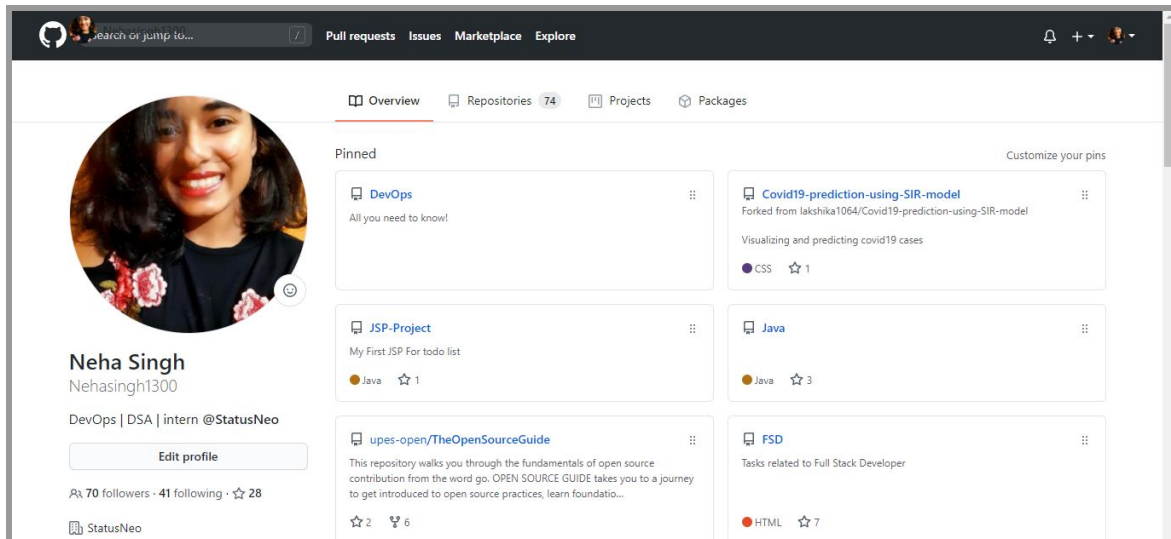
Plugin Name	Version	Description
<input checked="" type="checkbox"/> Git	4.3.0	This plugin integrates Git with Jenkins.
<input checked="" type="checkbox"/> Maven Integration plugin	3.7	This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...).

If you do not have GitHub Account, move to <https://github.com/>
Enter credentials



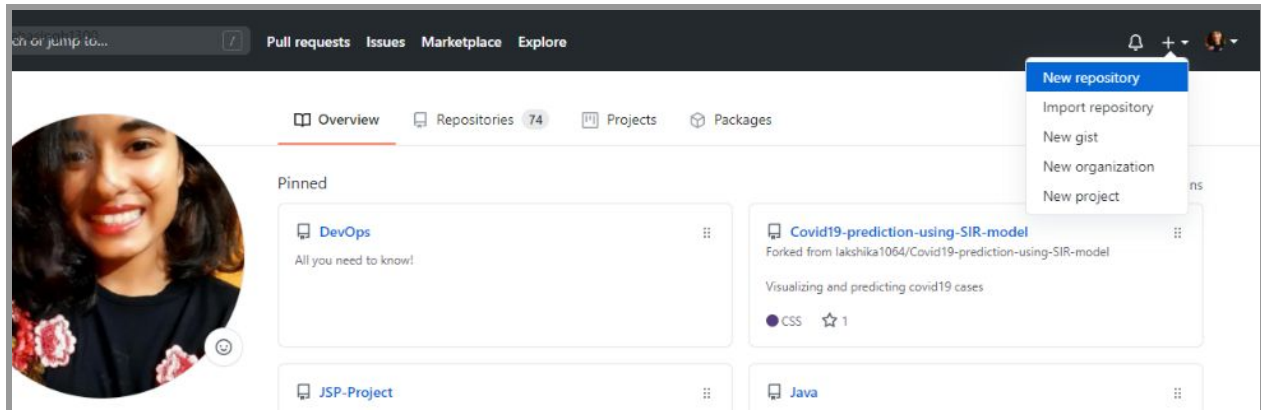
The screenshot shows the GitHub 'Create your account' page. At the top, it says 'Join GitHub' and 'Create your account'. There are three input fields for 'Username', 'Email address', and 'Password'. Below the password field, there is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. There is also an 'Email preferences' section with a checked box for 'Send me occasional product updates, announcements, and offers.'

You will see account like this, but without repositories.





The screenshot shows a GitHub user profile for 'Neha Singh' (Nehasingh1300). The profile includes a circular profile picture, the name 'Neha Singh', the username 'Nehasingh1300', and the bio 'DevOps | DSA | intern @StatusNeo'. There is an 'Edit profile' button. Below the bio, it shows '70 followers · 41 following · 28 stars'. The main content area is titled 'Pinned' and shows a grid of pinned repositories: 'DevOps', 'JSP-Project', 'Covid19-prediction-using-SIR-model', 'Java', 'TheOpenSourceGuide', and 'FSD'. Each repository card shows the repository name, a brief description, the language, and the number of stars.

Now, you can move to the upper right-hand side and click on + sign to make new repo.




You can enter any name for your repository, and description if you want. Then, go and click Create Repository


Owner * Repository name *

 Nehasingh1300 / CI_CD 

Great repository names are short and memorable. Need inspiration? How about [stunning-octo-enigma?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

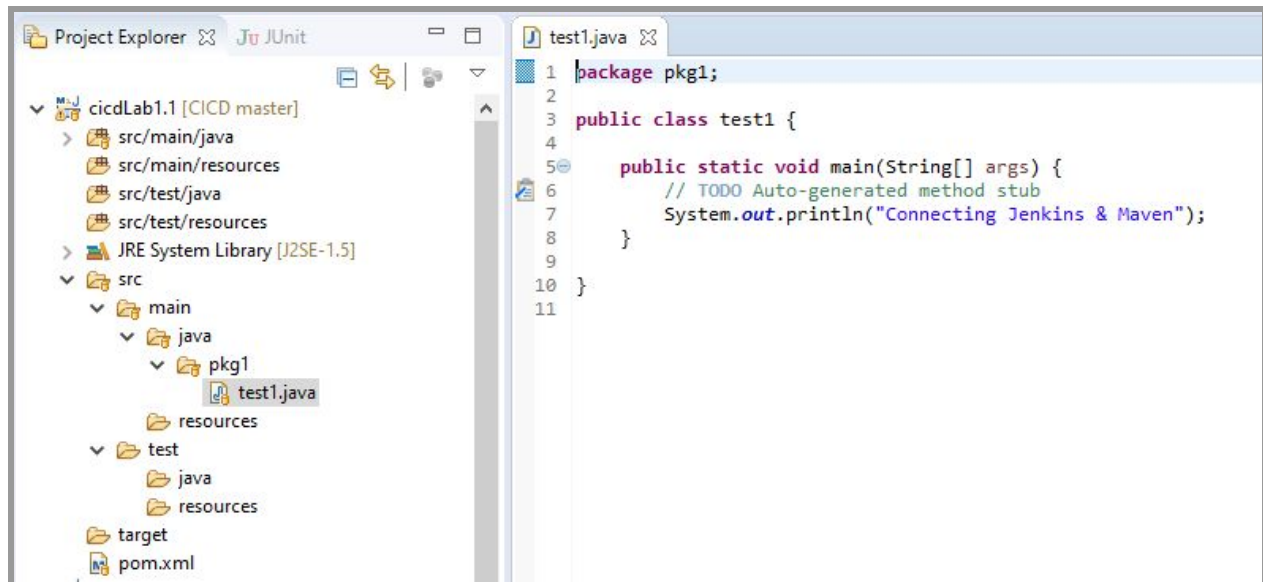
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

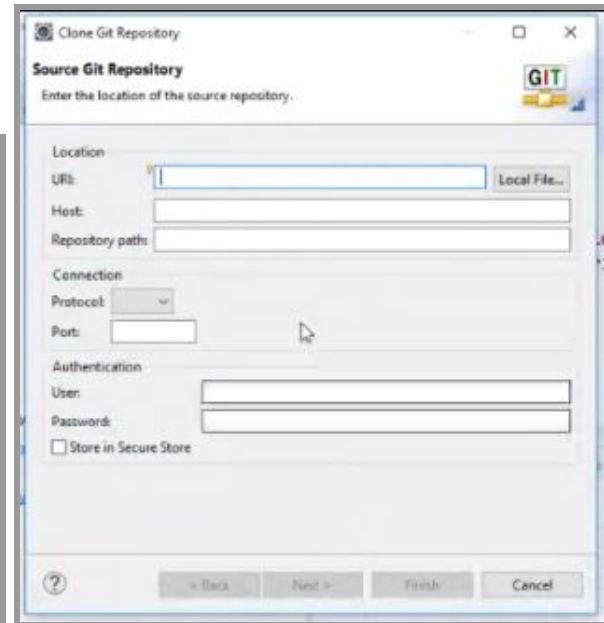
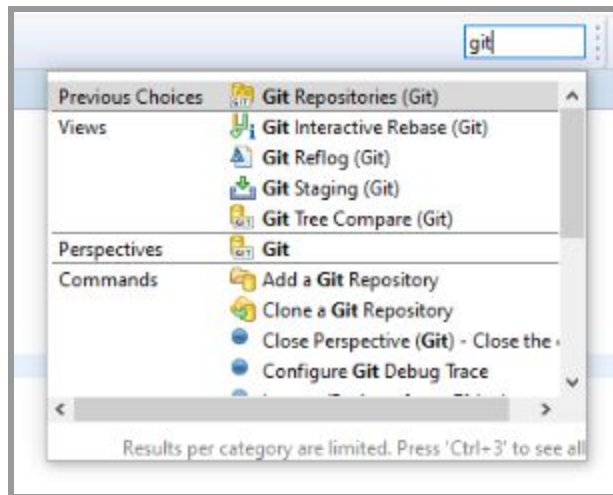
This will set  **master** as the default branch. Change the default name in your [settings](#).

Create repository

Now, move to Maven, where you have your maven project.

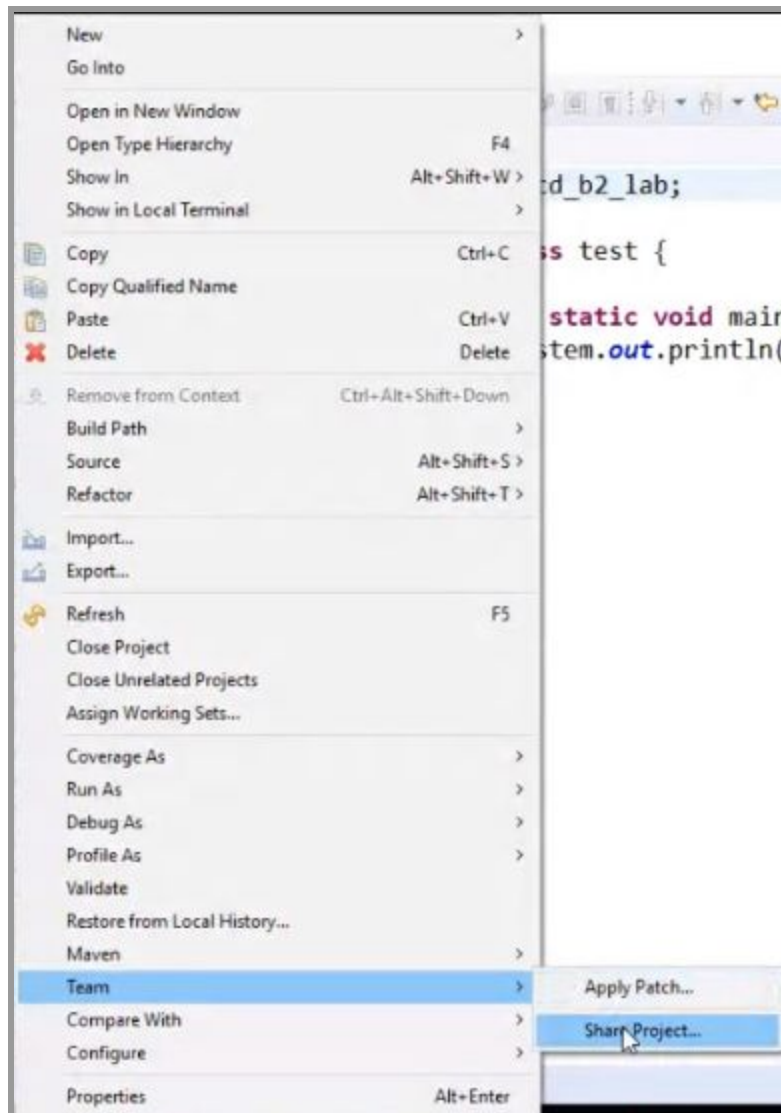


On the upper right-hand side search for git. Click on it.
Then click on the clone repository icon.
Enter the details and URL of your repository.

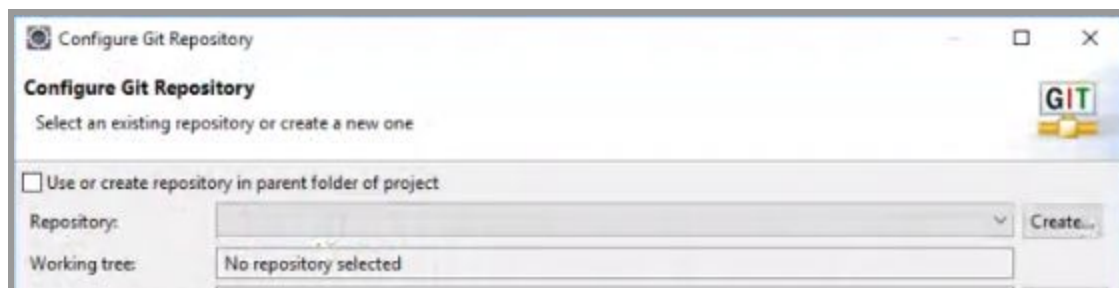


Go to your project and right-click on it.

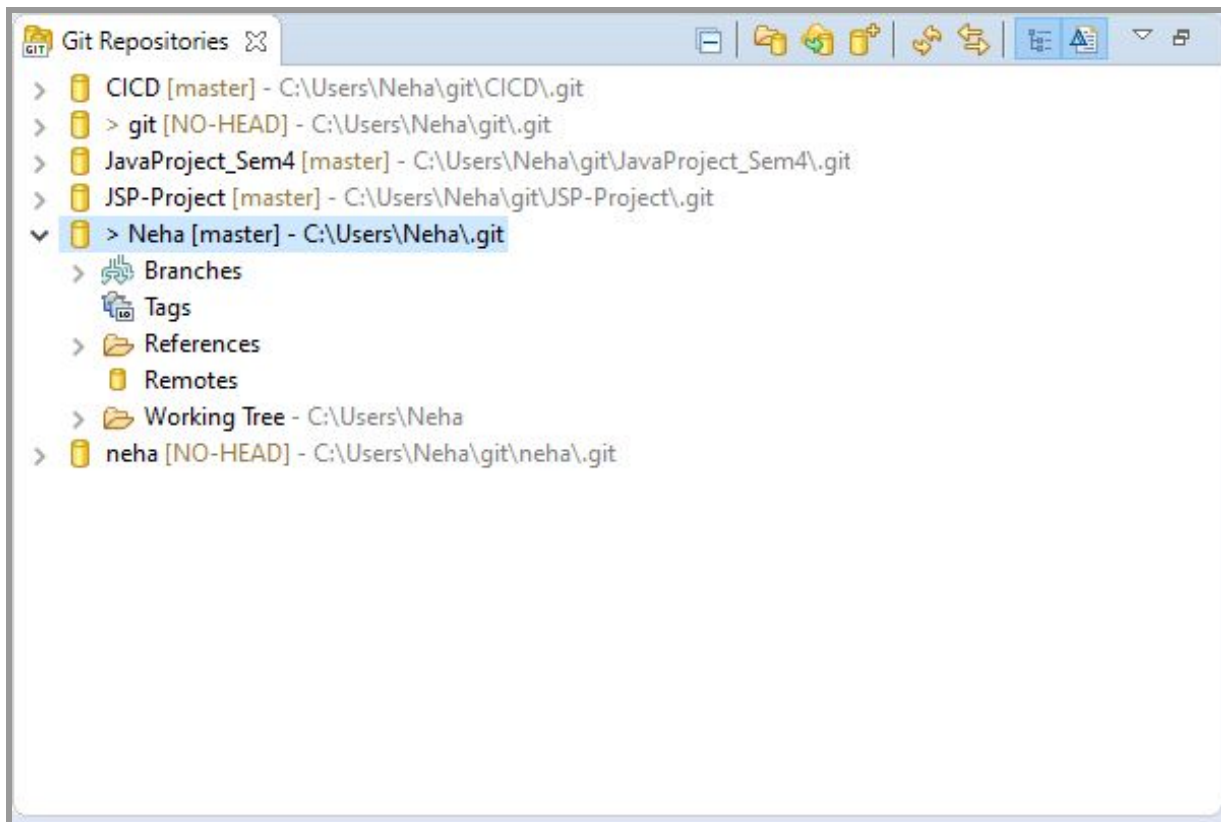
Move to Team >> share project >> Configure GitHub Repository >> enter credentials



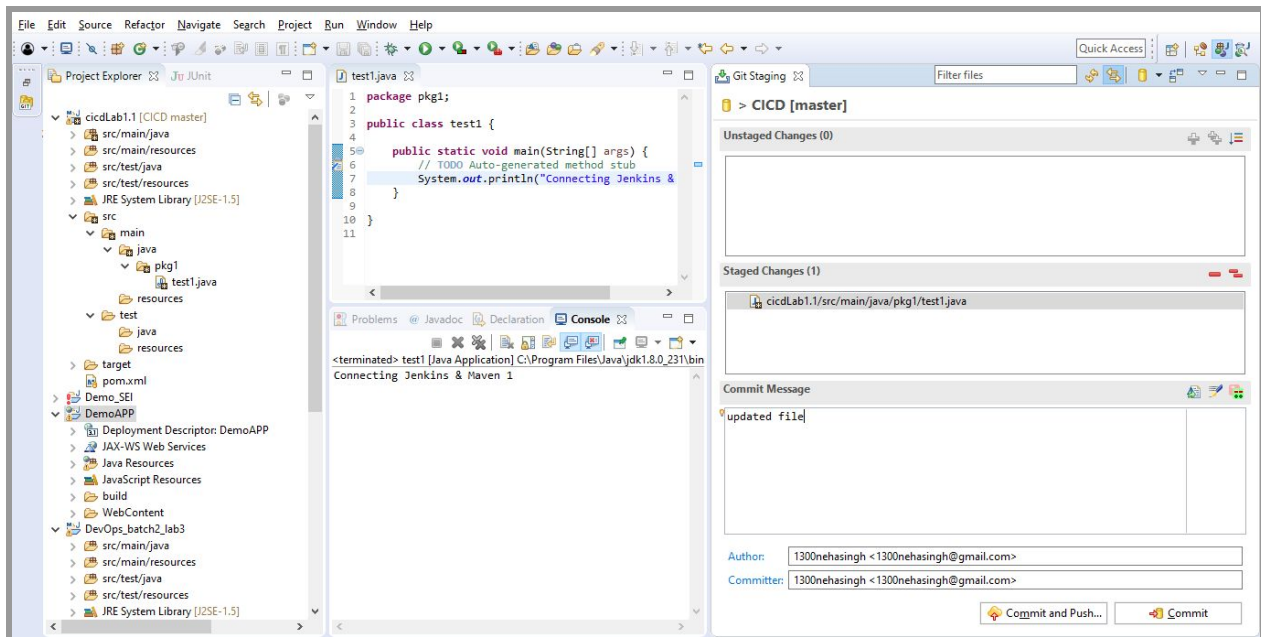
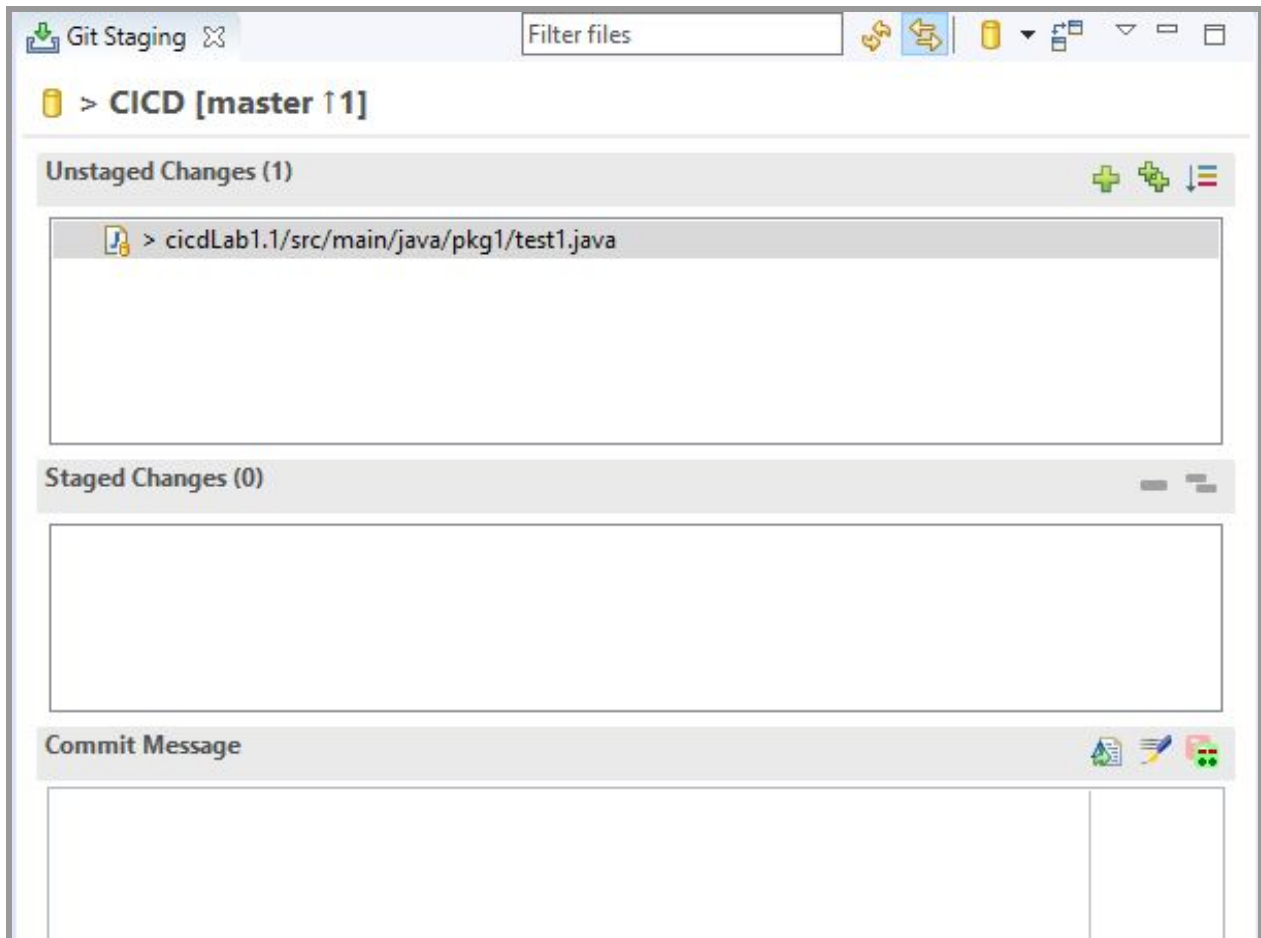
Configure Git Repository



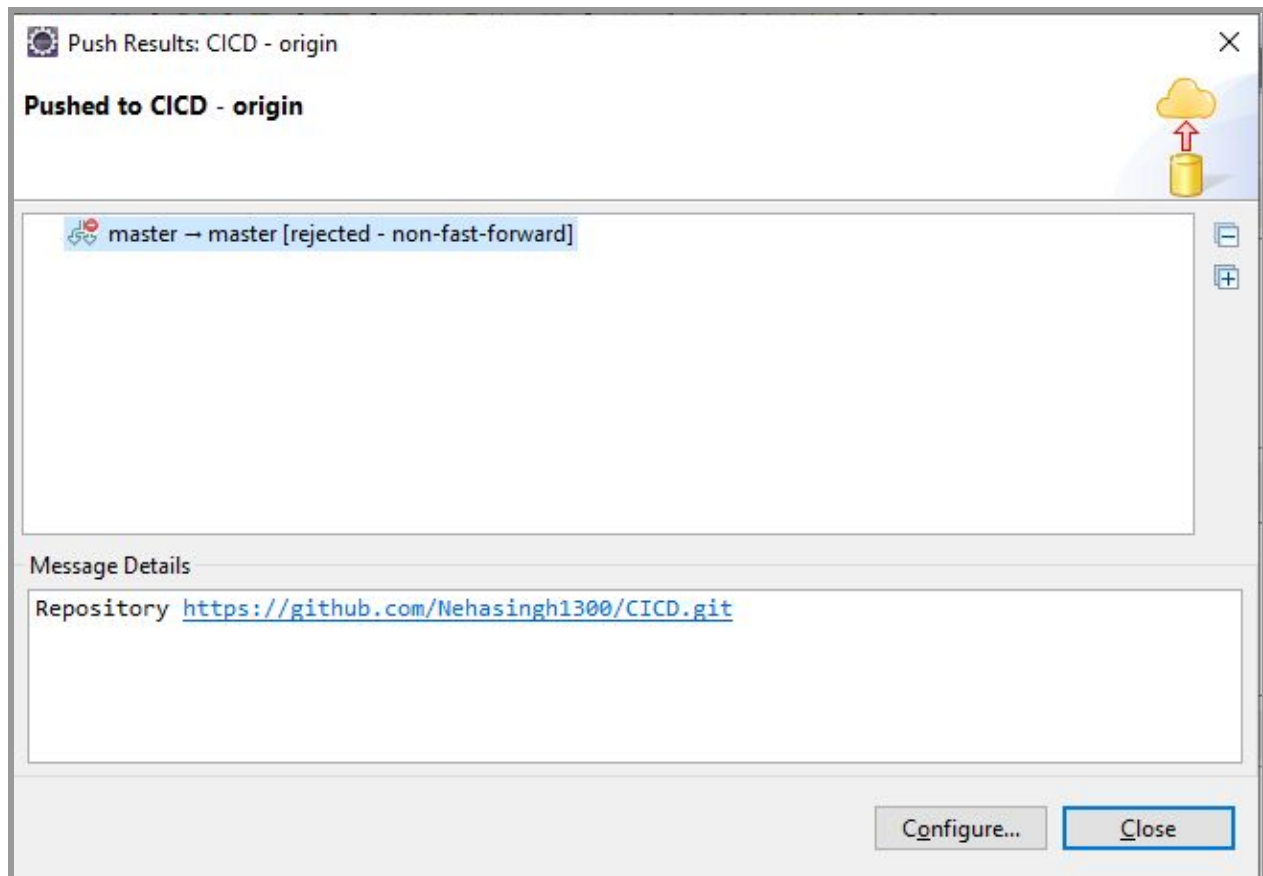
You will be able to see that your repository is connected with GitHub.



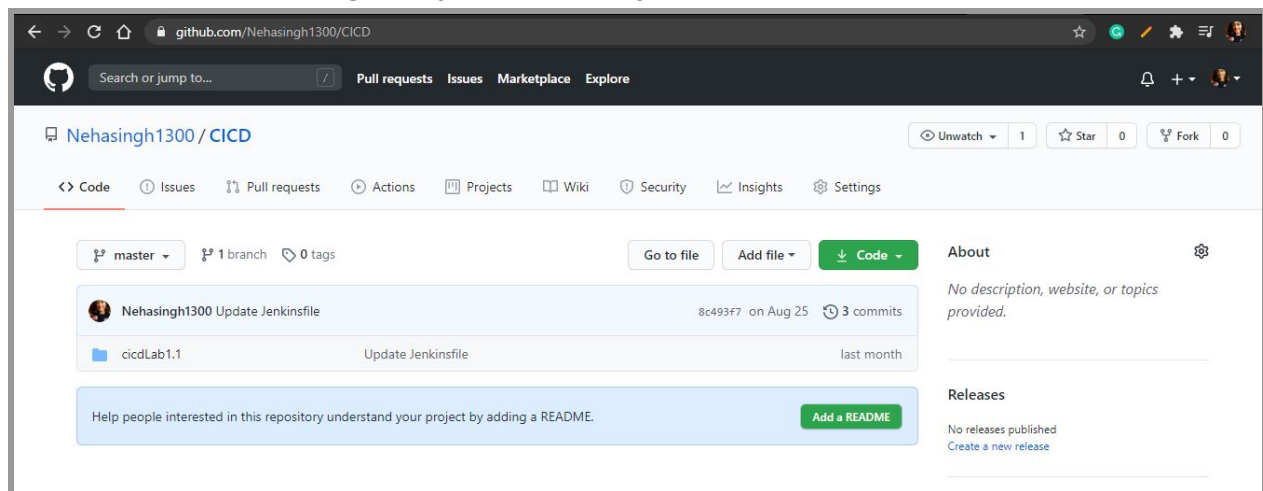
Now, drag all the changes from unstaged to staged. Enter the Commit message then click commit and push.



Enter



You can see all the changes in your repository.



Move to new items

The Jenkins dashboard shows a list of jobs under the 'All' filter. The jobs are listed in a table with columns: S, W, Name, Last Success, Last Failure, Last Duration, and Fav. The jobs are: BlueOcean, Blueocean.hands.on, CI_CD_LAB1, CI_CD_SonarQube, Ocean, test2_Jenkinsfile, and test_jenkinsfile. The sidebar on the left contains navigation options: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Open Blue Ocean, Lockable Resources, and New View. The bottom of the sidebar shows the Build Queue and Build Executor Status sections.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		BlueOcean	16 days - log	N/A	3.6 sec	
		Blueocean.hands.on	13 days - log	N/A	5.4 sec	
		CI_CD_LAB1	1 mo 13 days - #2	1 mo 13 days - #1	35 sec	
		CI_CD_SonarQube	1 mo 7 days - #3	1 mo 7 days - #2	1 min 43 sec	
		Ocean	16 days - log	N/A	3.5 sec	
		test2_Jenkinsfile	N/A	1 mo 7 days - #2	33 sec	
		test_jenkinsfile	1 mo 7 days - #2	N/A	24 sec	

Icon: S M L

Legend Atom feed for all Atom feed for failures Atom feed for just latest builds

add description

localhost:8080/view/all/newJob

Select Maven Project and enter the name of the project.

The 'Enter an item name' dialog box in Jenkins shows a text input field with the value 'CI_CD_LAB1.1'. Below the input field, there are four project type options: Freestyle project, Maven project, Pipeline, and Multi-configuration project. The 'Maven project' option is selected. The 'OK' button is visible at the bottom.

Enter an item name

CI_CD_LAB1.1

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Build projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

OK



Jenkins

CI_CD_LAB1

General

Source Code Management

Build Triggers

Build Environment

Pre Steps

Build

Post Steps

Build Settings

Post-build Actions

Description

[Plain text] [Preview](#)

Jira site

☐ Discard old builds☐ GitHub project☐ This build requires lockable resources☐ This project is parameterized☐ Throttle builds☐ Disable this project☐ Execute concurrent builds if necessary

Advanced...

Source Code Management

☐ None☒ Git

Repositories

Repository URL [https://github.com/Nehasingh1300/](#)

Credentials

- none -

Add

Advanced...

Save

Apply

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/master



Jenkins

CI_CD_LAB1

Repository browser

(Auto)



Additional Behaviours

Add ▾

☐ Mercurial☐ Subversion

Build Triggers

☒ Build whenever a SNAPSHOT dependency is built☐ Schedule build when some upstream has no successful builds☐ Trigger builds remotely (e.g., from scripts)☐ Build after other projects are built☐ Build periodically☐ GitHub hook trigger for GITScm polling☐ Poll SCM

Build Environment

☐ Delete workspace before build starts☐ Use secret text(s) or file(s)☐ Abort the build if it's stuck☐ Add timestamps to the Console Output☐ Generate Release Notes☐ Inspect build log for published Gradle build scans☐ With Ant

Pre Steps

Add pre-bui

Save

Apply

Build

Jenkins

CI_CD_LAB1

Goals and options

[Advanced...](#)

Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

[Add post-build step ▾](#)

Build Settings

☐ E-mail Notification

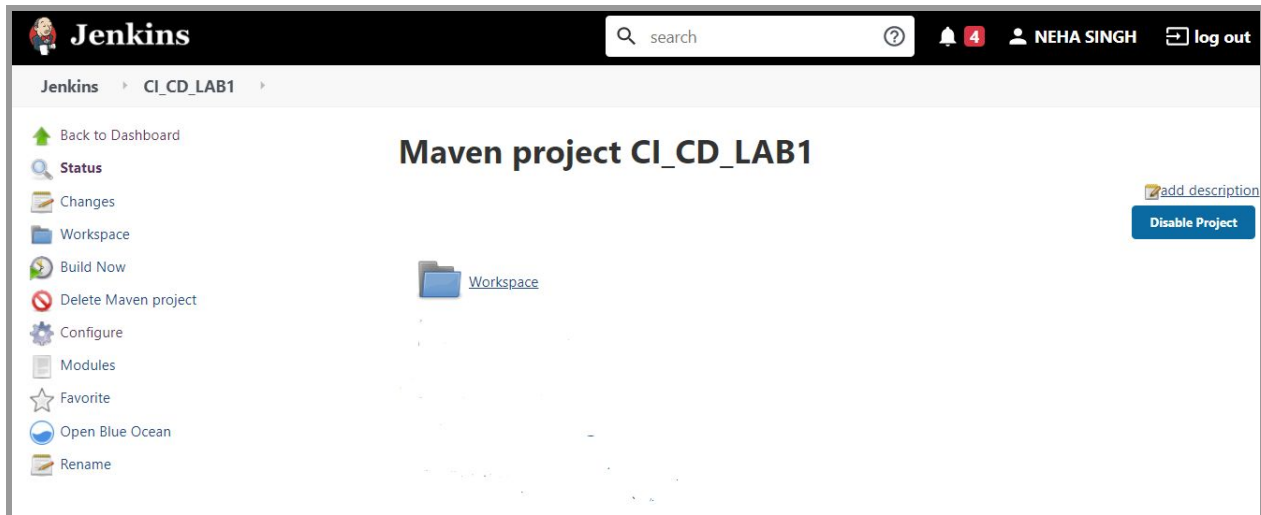
Post-build Actions

[Add post-build action ▾](#)

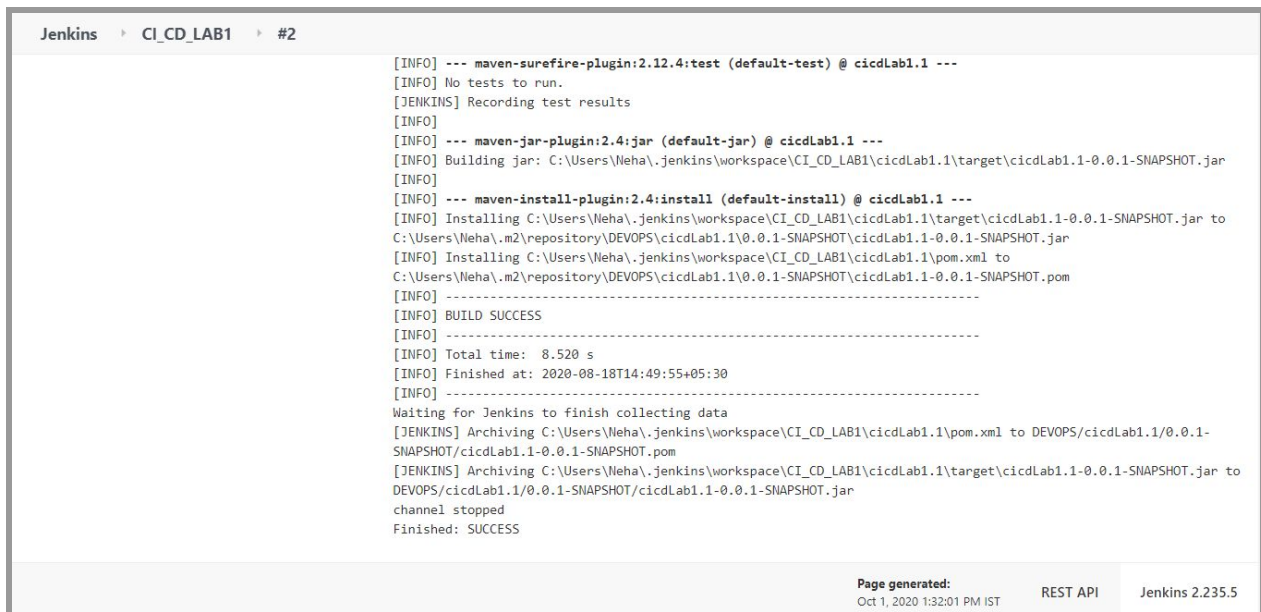
Jenkins 2.235.5

[Save](#)[Apply](#)

Now, you can see your project. Click on Build now



You will be able to see Build Success if your project has no error, else go to error message and make suitable changes.



You can see the build artifact



The image shows a Jenkins build page for the project 'cicdLab1.1'. The breadcrumb navigation at the top reads 'Jenkins > CI_CD_LAB1 > #2 > cicdLab1.1'. On the left sidebar, there are links for 'Back to Project', 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Delete build 'cicdLab1.1'', 'Executed Mojos', 'See Fingerprints', 'Redeploy Artifacts', and 'Open Blue Ocean'. The main content area has a title 'Build cicdLab1.1 (Aug 18, 2020 2:49:2...)' with a blue sphere icon. Below the title is a link to 'add description'. Under the 'Build Artifacts' section, there are two artifacts: 'cicdLab1.1-0.0.1-SNAPSHOT.jar' (1.85 KB) and 'cicdLab1.1-0.0.1-SNAPSHOT.pom' (359 B), each with a 'view' link. Below the artifacts, there is a note 'No changes.' with a notepad icon.

Jenkins > CI_CD_LAB1 > #2 > cicdLab1.1

[Back to Project](#)

[Status](#)

[Changes](#)

[Console Output](#)

[Edit Build Information](#)

[Delete build 'cicdLab1.1'](#)

[Executed Mojos](#)

[See Fingerprints](#)

[Redeploy Artifacts](#)

[Open Blue Ocean](#)

Build cicdLab1.1 (Aug 18, 2020 2:49:2...)

[add description](#)

Build Artifacts

cicdLab1.1-0.0.1-SNAPSHOT.jar	1.85 KB	view
cicdLab1.1-0.0.1-SNAPSHOT.pom	359 B	view

[No changes.](#)

That's all for this module.