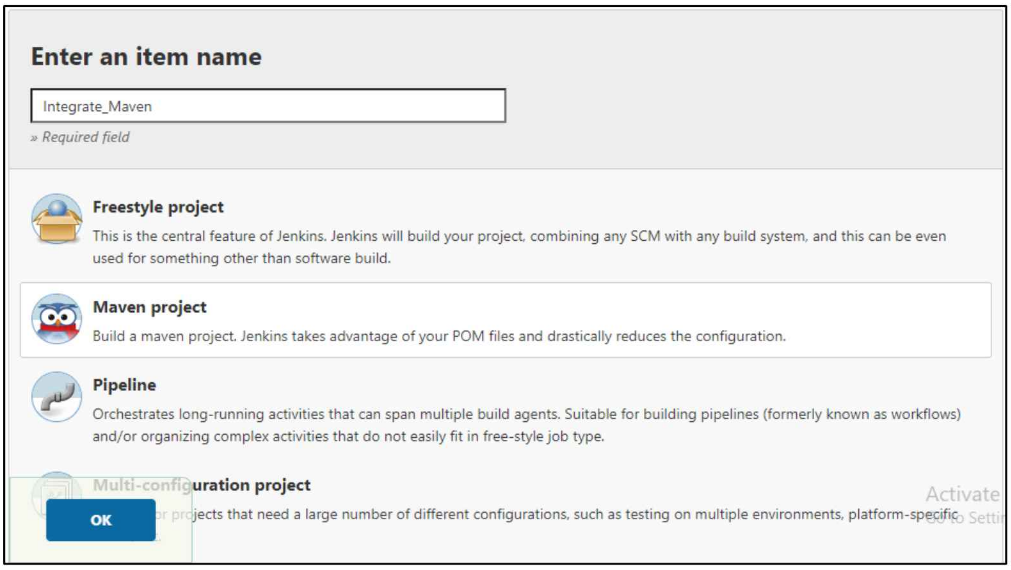
CICD EXPERIMENT – 3

Jenkins integration with GitHub and Maven

1. Start the Jenkins service using “java –jar jenkins.war” in command prompt
2. Go to “localhost:8080” in a web browser
3. Click “New Item” option in Jenkins dashboard. Enter the item name and select maven project



Enter an item name

Integrate_Maven

» Required field

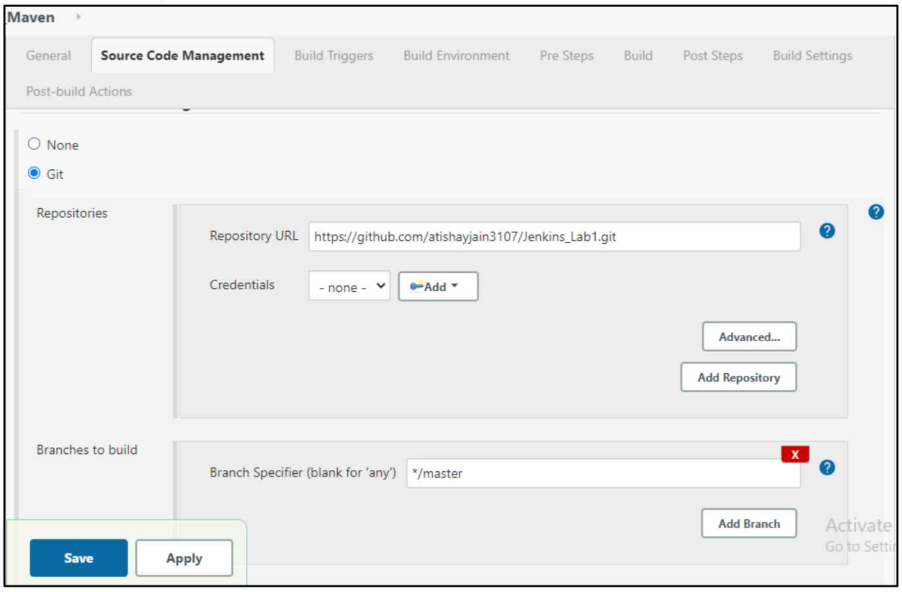
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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
...r projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

4. Now go to source code management tab and enter the URL of the github repository you want to integrate with



Maven

General | Source Code Management | Build Triggers | Build Environment | Pre Steps | Build | Post Steps | Build Settings | Post-build Actions

○ None
● Git

Repositories

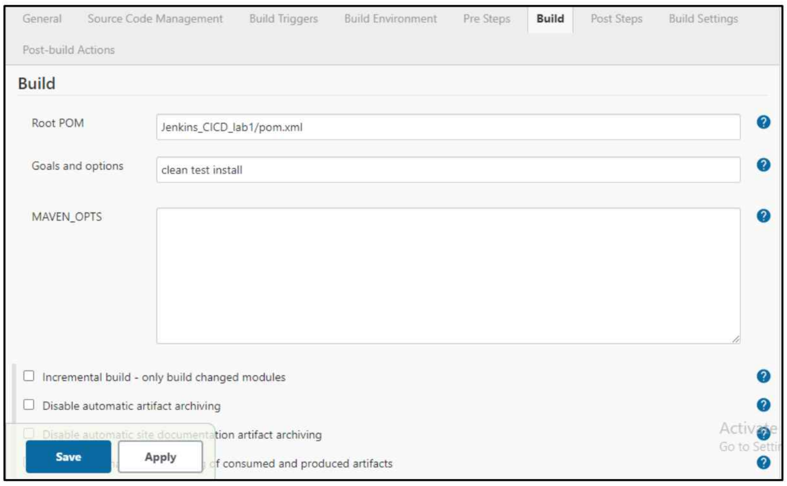
Repository URL: https://github.com/atishayjain3107/Jenkins_Lab1.git

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

5. Now go to build tab and add the exact location of pom.xml file. If it’s inside a folder then, specify it as “{folder name}/pom.xml”
6. In goals and option section, specify “clean test install” as maven goals and hit the save button



Build

Root POM: Jenkins_CICD_lab1/pom.xml

Goals and options: clean test install

MAVEN_OPTS

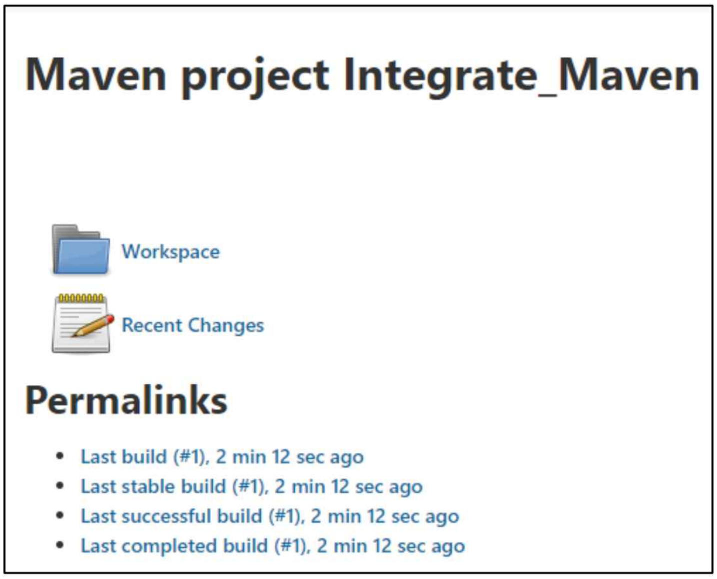
□ Incremental build - only build changed modules

□ Disable automatic artifact archiving

□ Disable automatic site documentation artifact archiving

f consumed and produced artifacts

7. You will be redirected to the dashboard page. Here, select the workspace option

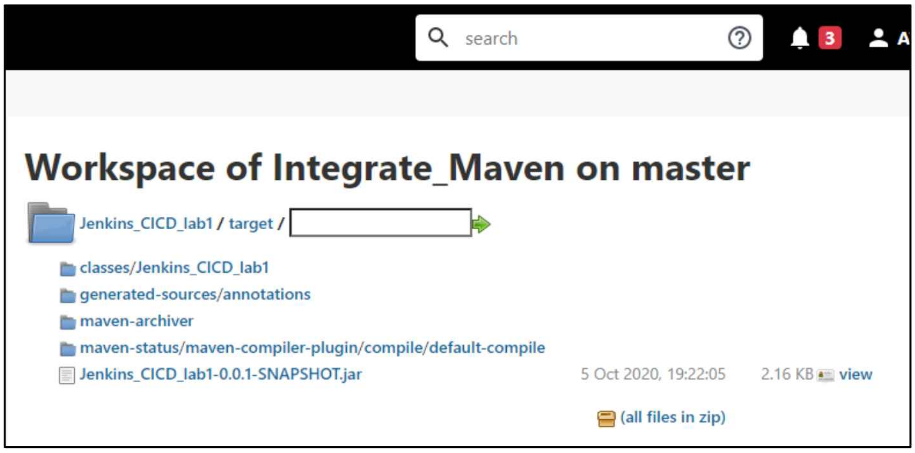


Maven project Integrate_Maven

Permalinks

- Last build (#1), 2 min 12 sec ago
- Last stable build (#1), 2 min 12 sec ago
- Last successful build (#1), 2 min 12 sec ago
- Last completed build (#1), 2 min 12 sec ago

8. In that, you'll find the folder named "target". Open that folder and you'll find a jar file which is created after maven goals run successfully



Workspace of Integrate_Maven on master

Jenkins_CICD_lab1 / target /

- classes/Jenkins_CICD_lab1
- generated-sources/annotations
- maven-archiver
- maven-status/maven-compiler-plugin/compile/default-compile
- Jenkins_CICD_lab1-0.0.1-SNAPSHOT.jar 5 Oct 2020, 19:22:05 2.16 KB view

(all files in zip)