

**Sumyak Jain, 500068360, R171218106**

## **Lab Final Exp**

**Create a Simple Jenkins pipeline using Jenkinsfile and create a build using it**

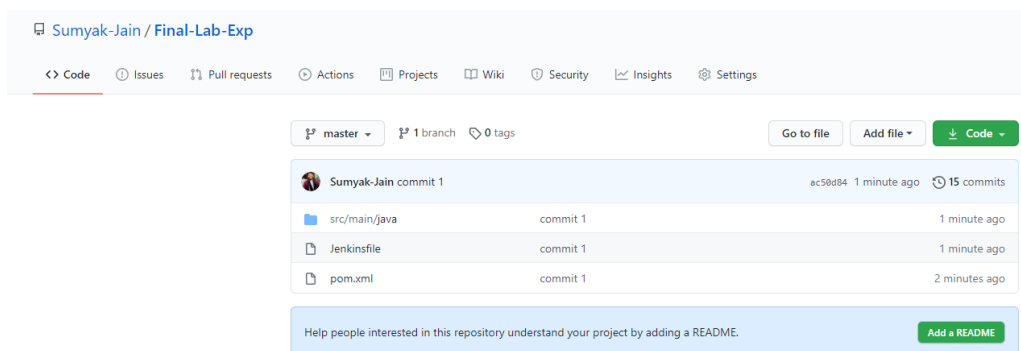
### **STEP 1) Write a Jenkins file**

**For example you can write**

```
pipeline
{
    agent any
    stages
    {
        stage('Clean') {
            steps {
                bat 'mvn clean'
            }
        }
        stage('Install') {
            steps {
                bat 'mvn install'
            }
        }

        stage('Deploy') {
            steps {
                echo 'deploy'
            }
        }
    }
}
```

### **STEP 2) upload your code to Github with a name of JenkinsFile and upload simple maven project with**



Sumyak-Jain / Final-Lab-Exp

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

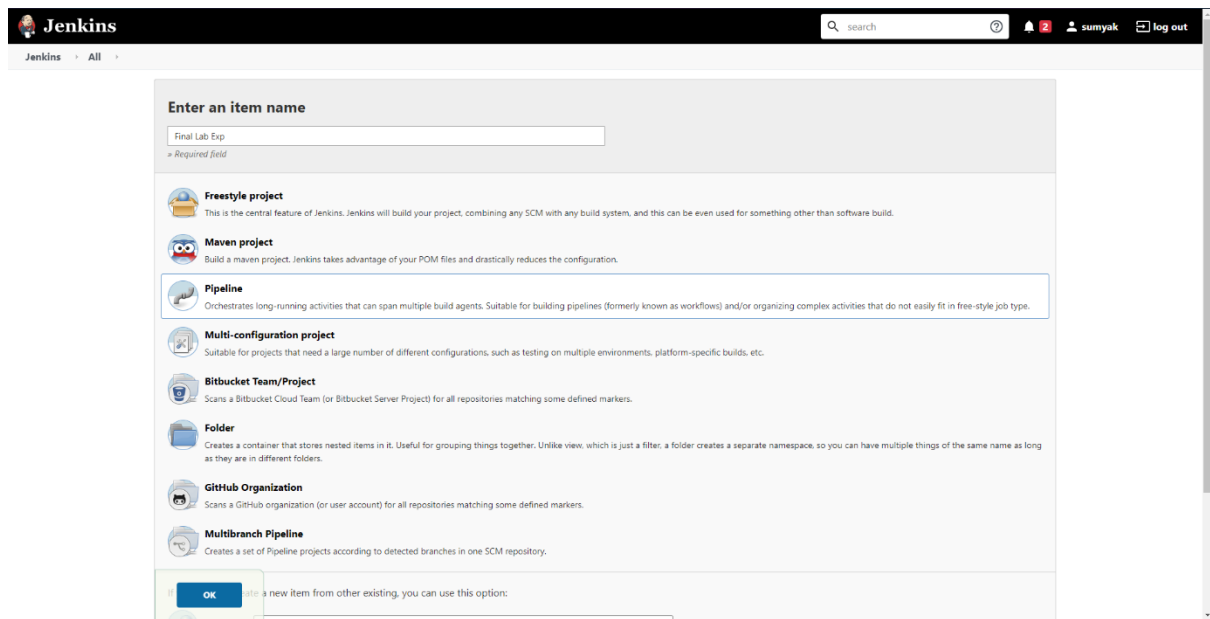
Go to file Add file Code

Sumyak-Jain commit 1 ac58d84 1 minute ago 15 commits

src/main/java	commit 1	1 minute ago
Jenkinsfile	commit 1	1 minute ago
pom.xml	commit 1	2 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

### STEP 3) make a Jenkins pipeline project



The screenshot shows the Jenkins 'Enter an item name' dialog. At the top, there's a search bar and user information. Below, a list of project types is shown: Freestyle project, Maven project, Pipeline (highlighted), Multi-configuration project, Bitbucket Team/Project, Folder, GitHub Organization, and Multibranch Pipeline. Each type has a brief description. At the bottom, there are 'OK' and 'Cancel' buttons, and a note about creating a new item from an existing one.

**Enter an item name**

Final Lab Exp  
→ Required field

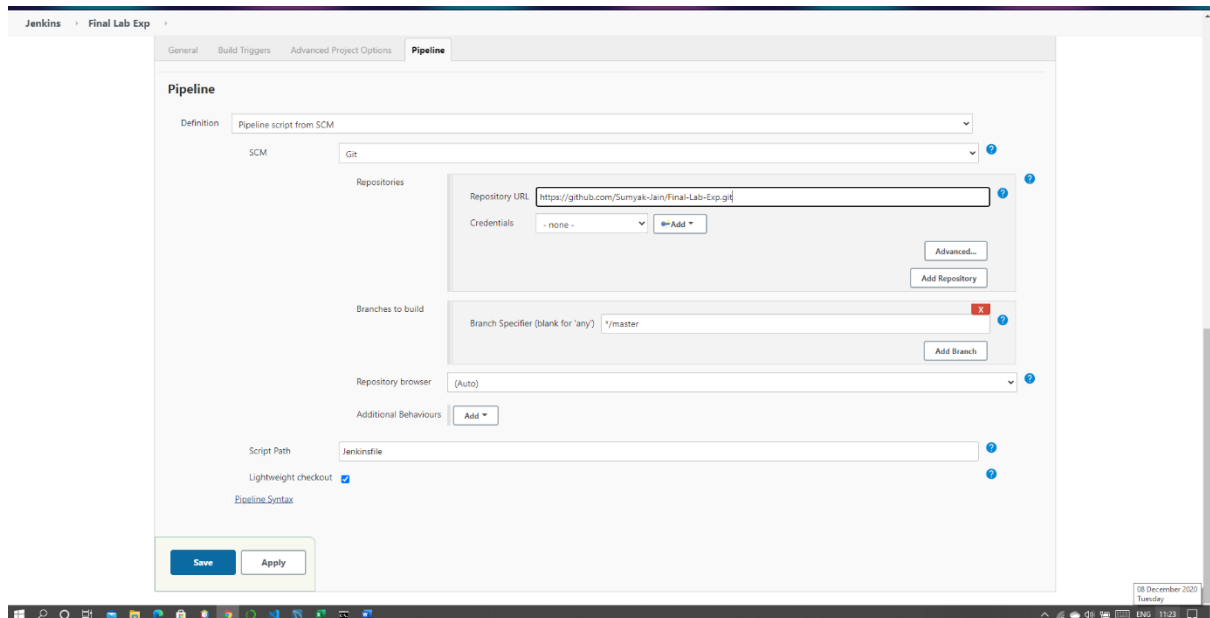
- Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically 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**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Bitbucket Team/Project**  
Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**OK** **Cancel** Use an existing item as a template to create a new item from other existing, you can use this option:

### STEP 4) provide the url of your repo in the project,

And select JenkinsFile as the script path,

Then click save and build.



The screenshot shows the Jenkins Pipeline configuration page for a job named 'Final Lab Exp'. The 'Definition' is set to 'Pipeline script from SCM'. The 'SCM' is set to 'Git'. The 'Repository URL' is 'https://github.com/Sumyak-Jain/Final-Lab-Exp.git'. The 'Branches to build' section shows 'Branch Specifier (blank for 'any')' set to '\*/master'. The 'Script Path' is 'Jenkinsfile'. The 'Lightweight checkout' checkbox is checked. There are 'Save' and 'Apply' buttons at the bottom.

**Pipeline**

Definition: Pipeline script from SCM

SCM: Git

Repositories:

- Repository URL:
- Credentials:
- Buttons: **Advanced...**, **Add Repository**

Branches to build:

- Branch Specifier (blank for 'any'):
- Buttons: **Add Branch**

Repository browser:

Additional Behaviours: **Add**

Script Path:

Lightweight checkout: ☒

[Pipeline Syntax](#)

**Save** **Apply**

## STEP 4) if build successful

The screenshot displays the Jenkins web interface for a job named 'Final Lab Exp' at step #2. The left sidebar contains navigation links: Back to Project, Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Git Build Data, No Tags, Open Blue Ocean, Thread Dump, Pipeline Steps, Workspaces, and Previous Build. The main area is titled 'Console Output' and shows the execution log of the build process. The log begins with 'Started by user sumyak' and 'Obtained Jenkinsfile from git https://github.com/sumyak-3ain/Final-Lab-Exp.git'. It details the pipeline stages: 'node' (running on Jenkins), 'stage' (Checkout 50%), 'checkout' (no credentials specified), 'git' (rev-parse --is-inside-work-tree # timeout=10), 'git' (version # timeout=10), 'git' (fetch --tags --force --progress -- https://github.com/sumyak-3ain/Final-Lab-Exp.git --refs/heads/\*:refs/remotes/origin/\* # timeout=10), 'git' (rev-parse refs/remotes/origin/master:commit# # timeout=10), 'git' (rev-parse refs/remotes/origin/master:commit# # timeout=10), 'git' (checkout -f 61c81d0f0439760b2c58f0ab891ede4e46c23 # timeout=10), 'git' (rev-list --no-walk 6027584961b0808a38de7be58f8c52e48f6 # timeout=10), 'stage' (withEnv), 'stage' (Clean), and 'bat'. The log then shows 'C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp>mvn clean' and 'Scanning for projects...'. It identifies the project as 'org.example:Assignment2' and proceeds with 'Building Assignment2 1.0-SNAPSHOT'. The build process includes 'maven-clean-plugin:2.5:clean (default-clean) @ Assignment2 ---', 'Deleting C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target', 'maven-resources-plugin:2.6:resources (default-resources) @ Assignment2 ---', 'Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!', 'skip non existing resourceDirectory C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\src\main\resources', 'maven-compiler-plugin:3.1:compile (default-compile) @ Assignment2 ---', 'Changes detected - recompiling the module!', 'File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!', 'Compiling 1 source file to C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\classes', 'maven-resources-plugin:2.6:testResources (default-testResources) @ Assignment2 ---', 'Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!', 'skip non existing resourceDirectory C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\src\test\resources', 'maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Assignment2 ---', 'No sources to compile', 'maven-surefire-plugin:2.12.4:test (default-test) @ Assignment2 ---', 'No tests to run.', 'maven-jar-plugin:2.4:jar (default-jar) @ Assignment2 ---', 'Building jar: C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\Assignment2-1.0-SNAPSHOT.jar', 'maven-install-plugin:2.4:install (default-install) @ Assignment2 ---', 'Installing C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\Assignment2-1.0-SNAPSHOT.jar to C:\Users\GAURAV\.m2\repository\org\example\Assignment2\1.0-SNAPSHOT\Assignment2-1.0-SNAPSHOT.jar', 'Installing C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\pom.xml to C:\Users\GAURAV\.m2\repository\org\example\Assignment2\1.0-SNAPSHOT\Assignment2-1.0-SNAPSHOT.pom', 'BUILD SUCCESS', 'Total time: 1.435 s', 'Finished at: 2020-12-08T11:25:49+05:30', 'The requested profile "profile" could not be activated because it does not exist.', 'stage' (Deploy), 'stage' (withEnv), 'stage' (Clean), and 'bat'. The log concludes with 'Finished: SUCCESS'.

```
Started by user sumyak
Obtained Jenkinsfile from git https://github.com/sumyak-3ain/Final-Lab-Exp.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout 50%)
[Pipeline] checkout
No credentials specified
> D:\Git\bin\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> D:\Git\bin\git.exe config remote.origin.url https://github.com/sumyak-3ain/Final-Lab-Exp.git # timeout=10
Fetching upstream changes from https://github.com/sumyak-3ain/Final-Lab-Exp.git
> D:\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.27.0.windows.1'
> D:\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/sumyak-3ain/Final-Lab-Exp.git --refs/heads/*:refs/remotes/origin/* # timeout=10
> D:\Git\bin\git.exe rev-parse refs/remotes/origin/master:commit# # timeout=10
> D:\Git\bin\git.exe rev-parse refs/remotes/origin/master:commit# # timeout=10
Checking out Revision 61c81d0f0439760b2c58f0ab891ede4e46c23 (refs/remotes/origin/master)
> D:\Git\bin\git.exe config core.sparsecheckout # timeout=10
> D:\Git\bin\git.exe checkout -f 61c81d0f0439760b2c58f0ab891ede4e46c23 # timeout=10
Commit message: "update Jenkinsfile"
> D:\Git\bin\git.exe rev-list --no-walk 6027584961b0808a38de7be58f8c52e48f6 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Clean)
[Pipeline] bat

C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp>mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----> org.example:Assignment2 >-----
[INFO] Building Assignment2 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ Assignment2 ---
[INFO] Deleting C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Assignment2 ---
[INFO] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Assignment2 ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Assignment2 ---
[INFO] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Assignment2 ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Assignment2 ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Assignment2 ---
[INFO] Building jar: C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\Assignment2-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ Assignment2 ---
[INFO] Installing C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\target\Assignment2-1.0-SNAPSHOT.jar to C:\Users\GAURAV\.m2\repository\org\example\Assignment2\1.0-SNAPSHOT\Assignment2-1.0-SNAPSHOT.jar
[INFO] Installing C:\Users\GAURAV\.jenkins\workspace\Final Lab Exp\pom.xml to C:\Users\GAURAV\.m2\repository\org\example\Assignment2\1.0-SNAPSHOT\Assignment2-1.0-SNAPSHOT.pom
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.435 s
[INFO] Finished at: 2020-12-08T11:25:49+05:30
[INFO]
[WARNING] The requested profile "profile" could not be activated because it does not exist.
[INFO]
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
deploy
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] end of Pipeline
Finished: SUCCESS
```

STEP 5) view your pipeline as dashboard

