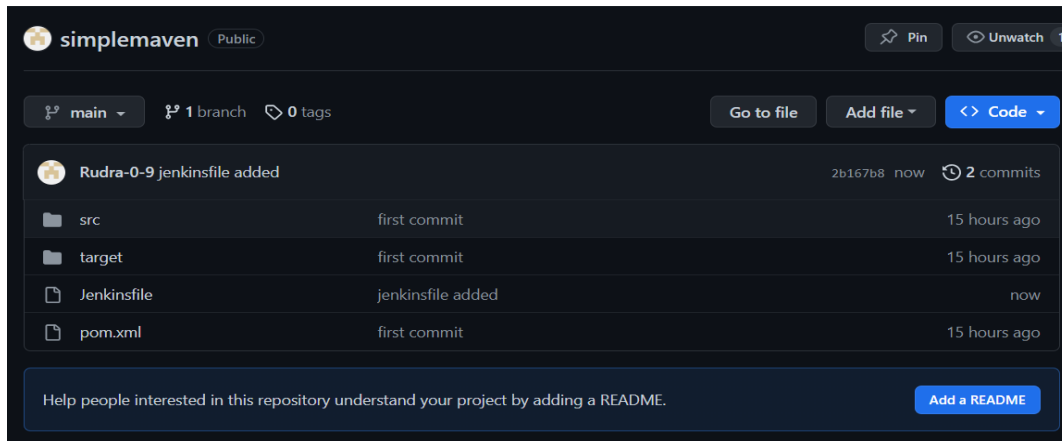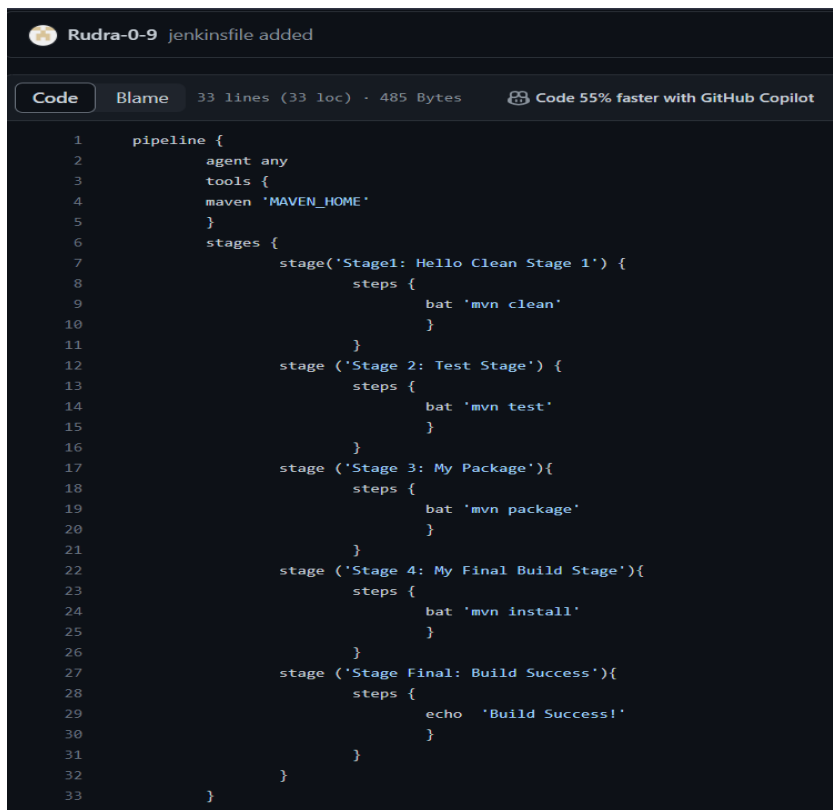# Experiment -2

(Creating a Jenkins Pipeline with a Jenkins File)

Objective: Create a Jenkins pipeline using a Jenkins file that builds a simple project, runs tests, and deploys the project to a designed environment.

1. Create a simple maven Project (the same one used in Experiment -1) and push the project to GitHub.

2. Create a Jenkins file and push it to GitHub repo (in the root dir) Note: the following Jenkins file includes stages associated with maven project


```
1    pipeline {
2        agent any
3        tools {
4        maven 'MAVEN_HOME'
5        }
6        stages {
7            stage('Stage1: Hello Clean Stage 1') {
8                steps {
9                    bat 'mvn clean'
10               }
11           }
12           stage ('Stage 2: Test Stage') {
13               steps {
14                   bat 'mvn test'
15               }
16           }
17           stage ('Stage 3: My Package'){
18               steps {
19                   bat 'mvn package'
20               }
21           }
22           stage ('Stage 4: My Final Build Stage'){
23               steps {
24                   bat 'mvn install'
25               }
26           }
27           stage ('Stage Final: Build Success'){
28               steps {
29                   echo  'Build Success!'
30               }
31           }
32       }
33   }
```

3. Create a Jenkins Pipeline CONFIGURATION:

Enter an item name

Lab2

» *Required field*

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, comb
for something other than software build.

Maven project

Build a maven project. Jenkins takes advantage of your POM files and dra

Pipeline

Orchestrates long-running activities that can span multiple build agents.
and/or organizing complex activities that do not easily fit in free-style job

Multi-configuration project

Suitable for projects that need a large number of different configurations
builds, etc.

Folder

Creates a container that stores nested items in it. Useful for grouping thir
separate namespace, so you can have multiple things of the same name

OK

Multibranch Pipeline

4. Select pipeline Script from SCM (git) and provide GitHub repo URL.

Pipeline

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Rudra-0-9/simplemaven

Credentials ?

- none -

+ Add ▾

5. Provide the Jenkins file path (path in the repo) in the Script Path.

Script Path ?

Jenkinsfile

✓ Lightweight checkout ?

Pipeline Syntax

6. Now if we make any changes to the GitHub repo and push it, it will trigger a new pipeline build.

Declarative: Checkout SCM	Declarative: Tool Install	Stage 1 : Clean Stage	Stage 2 : test Stage	Stage 3 : Install stage	Stage Final : Build Sucess
2s	172ms	4s	4s	4s	255ms
2s	172ms	4s	4s	4s	255ms