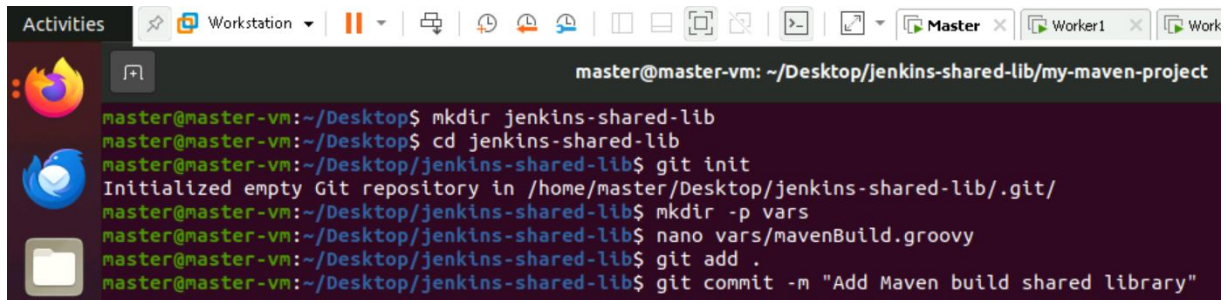


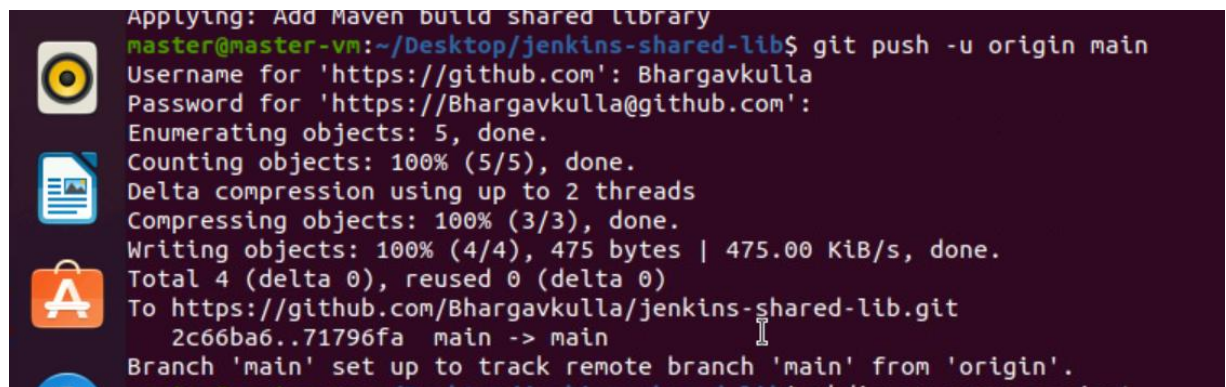
### Q3) Jenkins Shared Library

#### 1.Clone repo



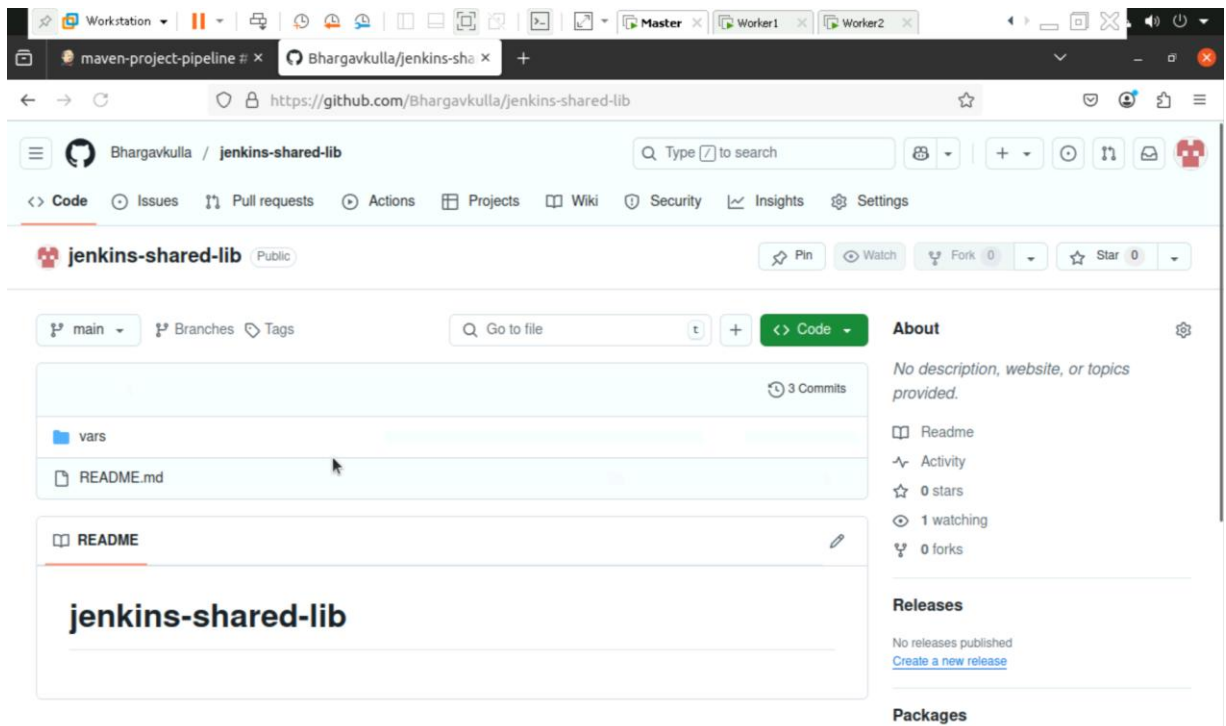
```
master@master-vm: ~/Desktop/jenkins-shared-lib/my-maven-project
master@master-vm:~/Desktop$ mkdir jenkins-shared-lib
master@master-vm:~/Desktop$ cd jenkins-shared-lib
master@master-vm:~/Desktop/jenkins-shared-lib$ git init
Initialized empty Git repository in /home/master/Desktop/jenkins-shared-lib/.git/
master@master-vm:~/Desktop/jenkins-shared-lib$ mkdir -p vars
master@master-vm:~/Desktop/jenkins-shared-lib$ nano vars/mavenBuild.groovy
master@master-vm:~/Desktop/jenkins-shared-lib$ git add .
master@master-vm:~/Desktop/jenkins-shared-lib$ git commit -m "Add Maven build shared library"
```

#### Push

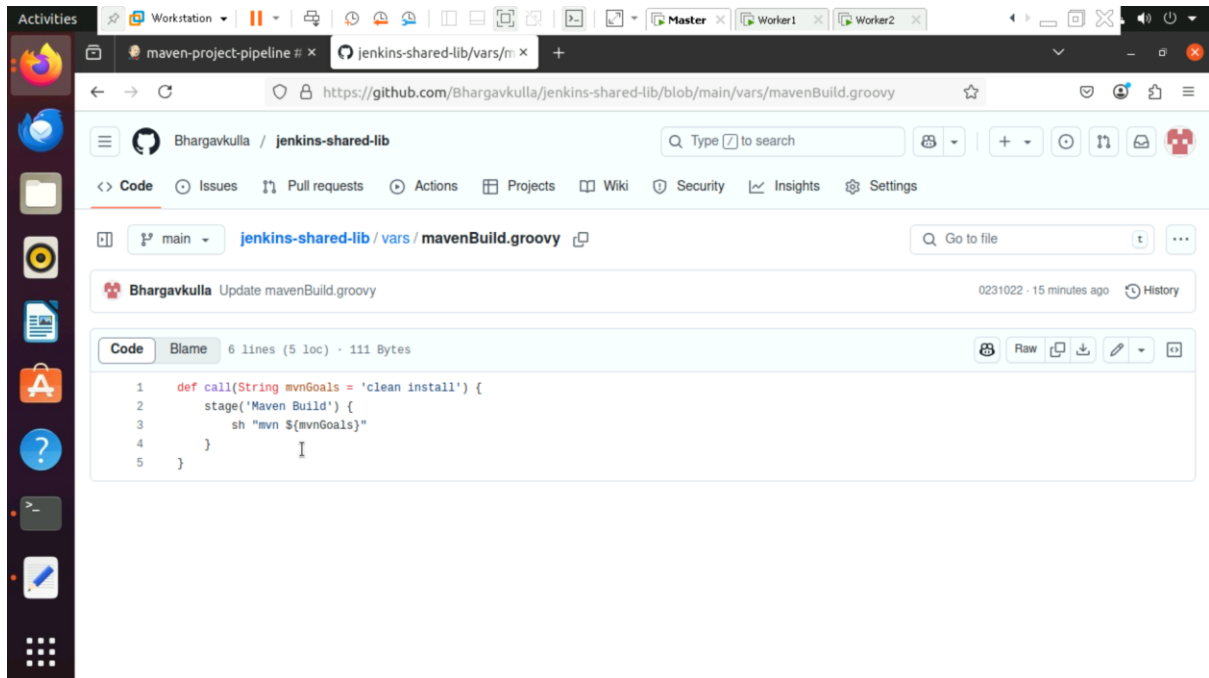


```
Applying: Add Maven build shared library
master@master-vm:~/Desktop/jenkins-shared-lib$ git push -u origin main
Username for 'https://github.com': Bhargavkulla
Password for 'https://Bhargavkulla@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 475 bytes | 475.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/Bhargavkulla/jenkins-shared-lib.git
2c66ba6..71796fa main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

#### Shared library repo



mavenBuild.groovy

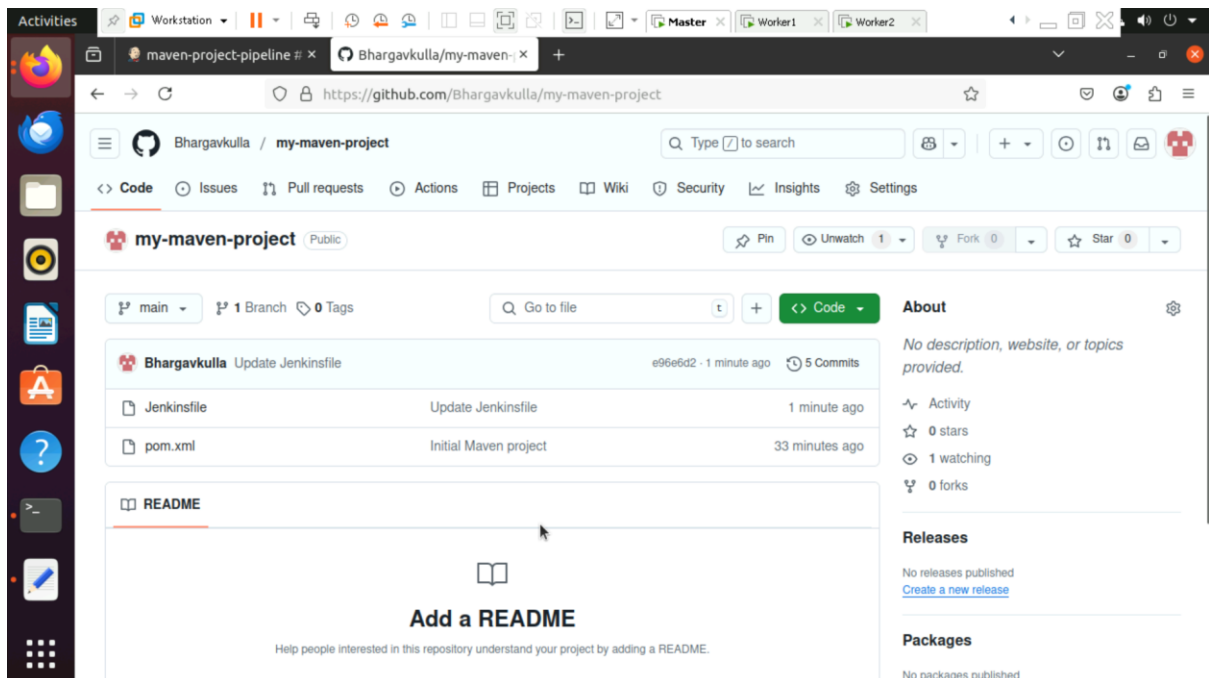


The screenshot shows a web browser displaying the GitHub repository 'jenkins-shared-lib' by user 'Bhargavkulla'. The file 'mavenBuild.groovy' is selected, showing its code. The code is a Groovy script for a Jenkins stage named 'Maven Build'. It calls a function 'call' with 'clean install' as the goal and runs a shell command 'mvn' with the goal as an argument.

```
1 def call(String mvnGoals = 'clean install') {  
2     stage('Maven Build') {  
3         sh "mvn ${mvnGoals}"  
4     }  
5 }
```

Repo(my-maven-project)

Repo: <https://github.com/Bhargavkulla/my-maven-project>



The screenshot shows the GitHub repository page for 'my-maven-project' by user 'Bhargavkulla'. The repository is public and has 1 branch and 0 tags. The commit history shows 5 commits. The file list includes 'Jenkinsfile' and 'pom.xml'. The README section is empty, with a prompt to 'Add a README'. The right sidebar shows the repository's activity, including 0 stars, 1 watching, and 0 forks.

**my-maven-project** Public

1 Branch 0 Tags

Go to file

Code

**Commit History**

Commit	Author	Message	Time
e96e6d2	Bhargavkulla	Update Jenkinsfile	1 minute ago
		Jenkinsfile	Update Jenkinsfile
		pom.xml	Initial Maven project

**README**

**Add a README**

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

**About**

No description, website, or topics provided.

**Activity**

- 0 stars
- 1 watching
- 0 forks

**Releases**

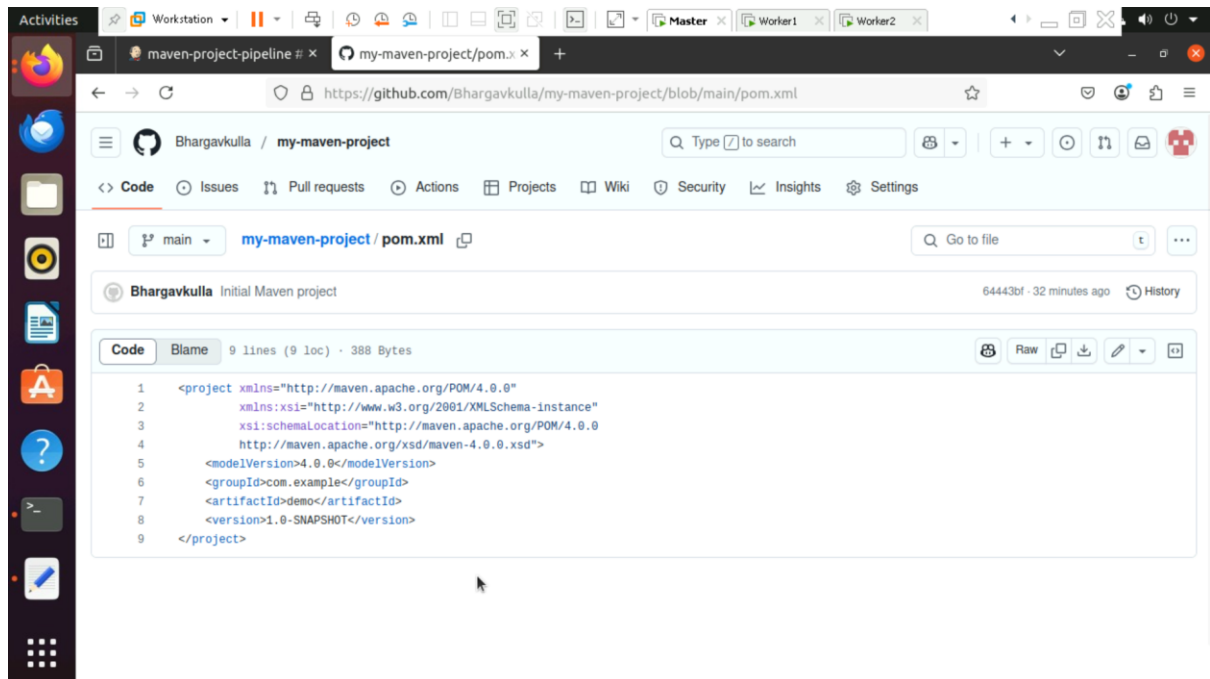
No releases published

[Create a new release](#)

**Packages**

No packages published

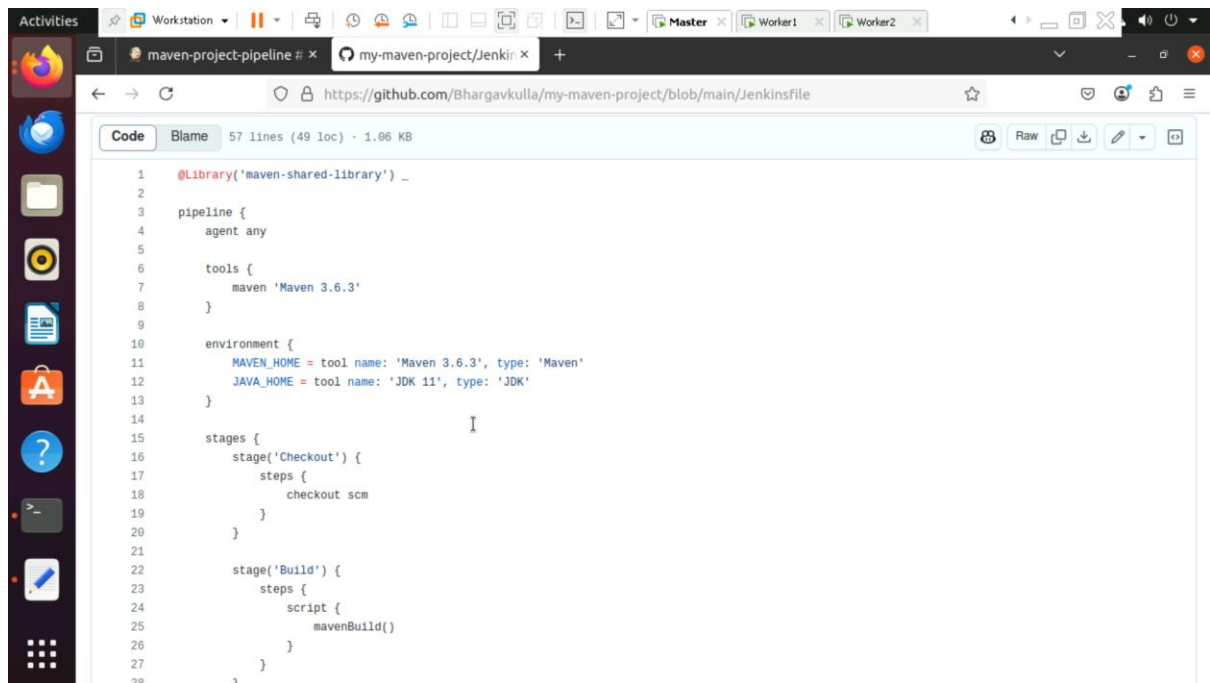
## Pom.xml



The screenshot shows a web browser window displaying the GitHub repository page for 'my-maven-project/pom.xml'. The browser's address bar shows the URL 'https://github.com/Bhargavkulla/my-maven-project/blob/main/pom.xml'. The repository name 'Bhargavkulla / my-maven-project' is visible at the top. The file 'pom.xml' is selected, and its content is displayed in a code editor. The code is a Maven POM file with the following content:

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6     <groupId>com.example</groupId>
7     <artifactId>demo</artifactId>
8     <version>1.0-SNAPSHOT</version>
9 </project>
```

## Jenkins file

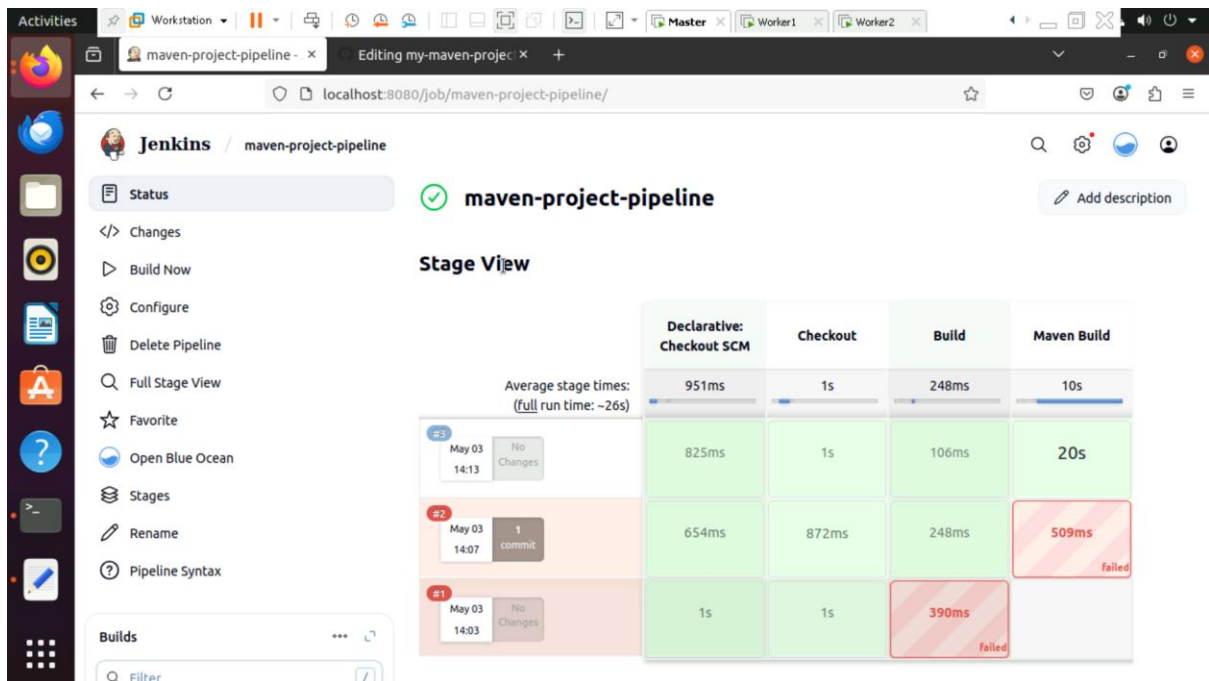


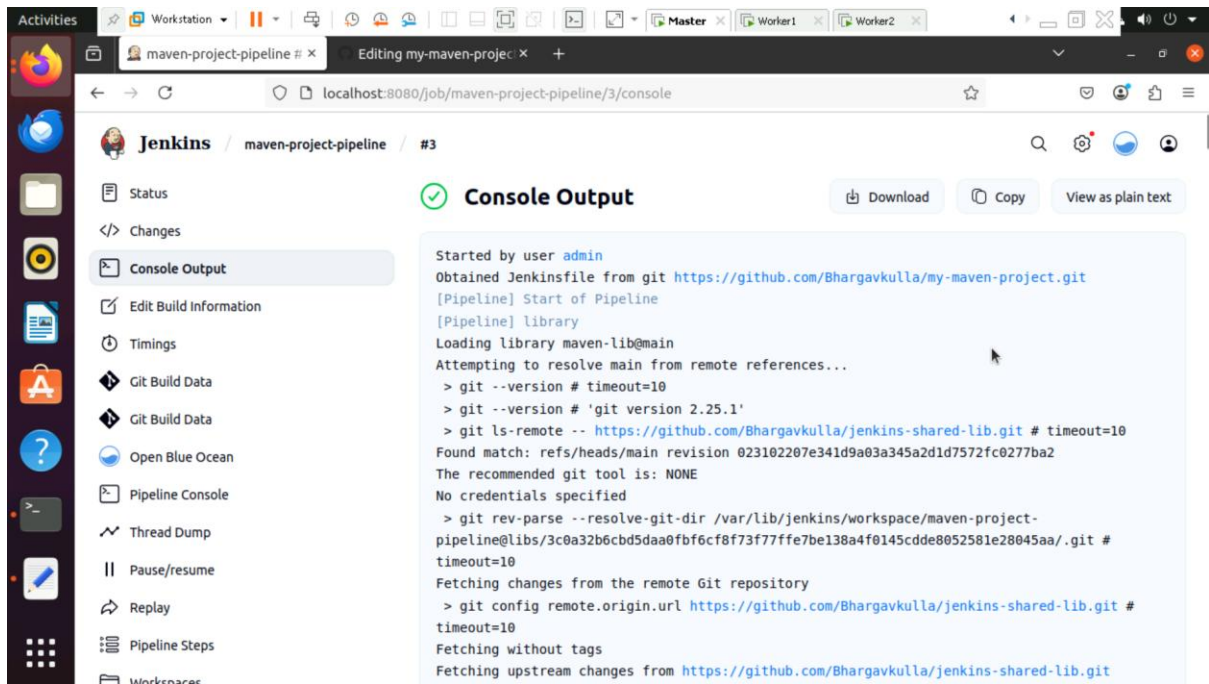
The screenshot shows a web browser window displaying the GitHub repository page for 'my-maven-project/Jenkinsfile'. The browser's address bar shows the URL 'https://github.com/Bhargavkulla/my-maven-project/blob/main/Jenkinsfile'. The repository name 'Bhargavkulla / my-maven-project' is visible at the top. The file 'Jenkinsfile' is selected, and its content is displayed in a code editor. The code is a Jenkinsfile with the following content:

```
1 @Library('maven-shared-library') _
2
3 pipeline {
4     agent any
5
6     tools {
7         maven 'Maven 3.6.3'
8     }
9
10    environment {
11        MAVEN_HOME = tool name: 'Maven 3.6.3', type: 'Maven'
12        JAVA_HOME = tool name: 'JDK 11', type: 'JDK'
13    }
14
15    stages {
16        stage('Checkout') {
17            steps {
18                checkout scm
19            }
20        }
21
22        stage('Build') {
23            steps {
24                script {
25                    mavenBuild()
26                }
27            }
28        }
29    }
30}
```

```
29
30     stage('Test') {
31         steps {
32             script {
33                 echo 'Running tests...'
34                 sh 'mvn test'
35             }
36         }
37     }
38
39     stage('Deploy') {
40         steps {
41             script {
42                 echo 'Deploying project...'
43                 sh 'mvn deploy'
44             }
45         }
46     }
47
48     post {
49         success {
50             echo 'Build and deployment succeeded!'
51         }
52         failure {
53             echo 'Build or deployment failed!'
54         }
55     }
```

## Output





## Success output

