Lesson 08 Demo 02

Setting up Jenkins Pipeline Job from Git

Objective: To set up a Jenkins pipeline job from Git version control system to enable automated CI for building, testing, and potentially deploying software upon code changes

Tools required: Jenkins, Git and Linux

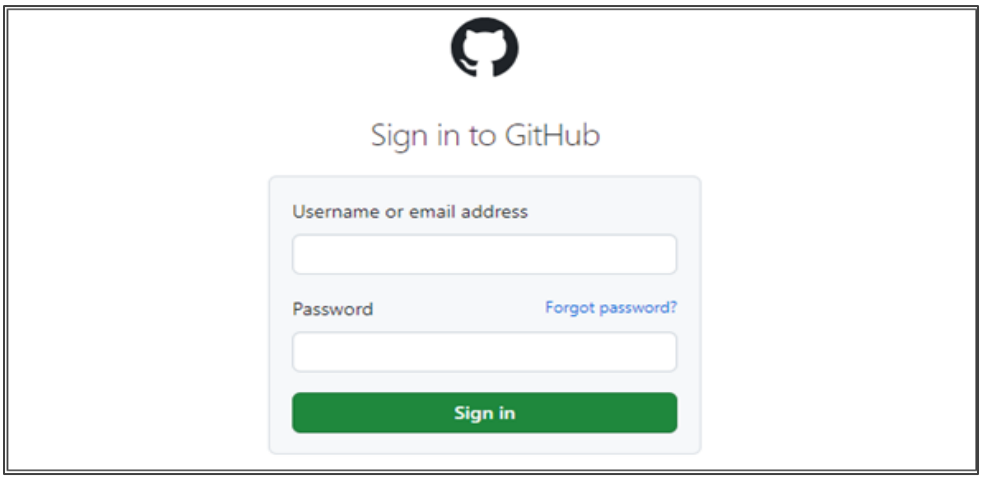
Prerequisites: None

Steps to be followed:

1. Create a Git repository
2. Push the pipeline script into the Git repository
3. Create a pipeline script-based freestyle job

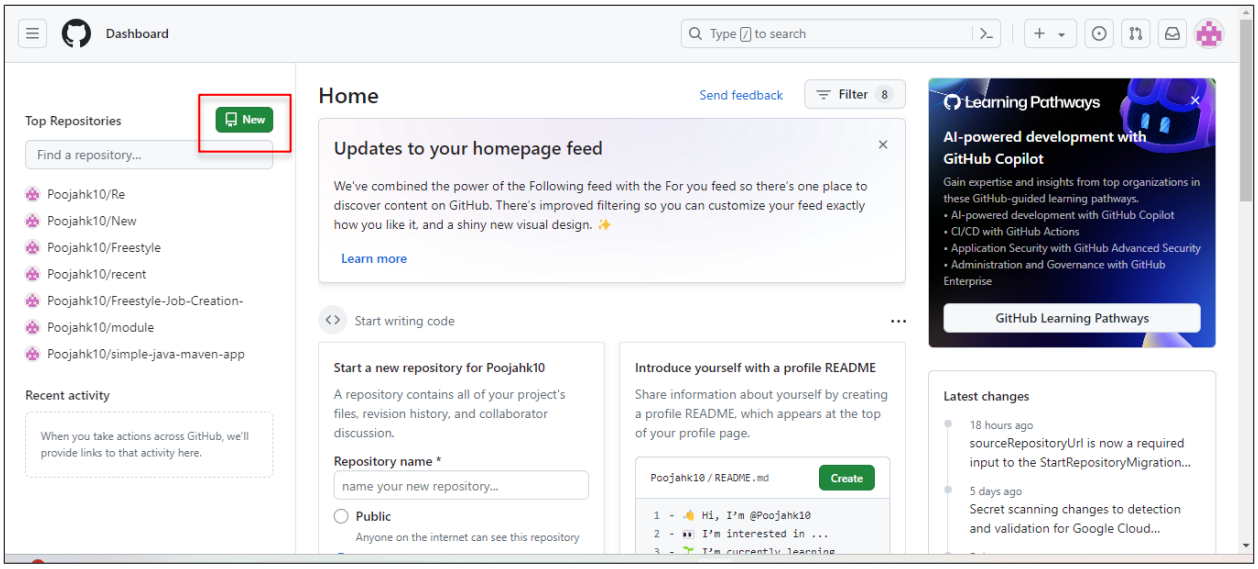
Step 1: Create a Git repository

1.1 Open the browser in your lab, go to **github.com**, and **Sign in** to your account

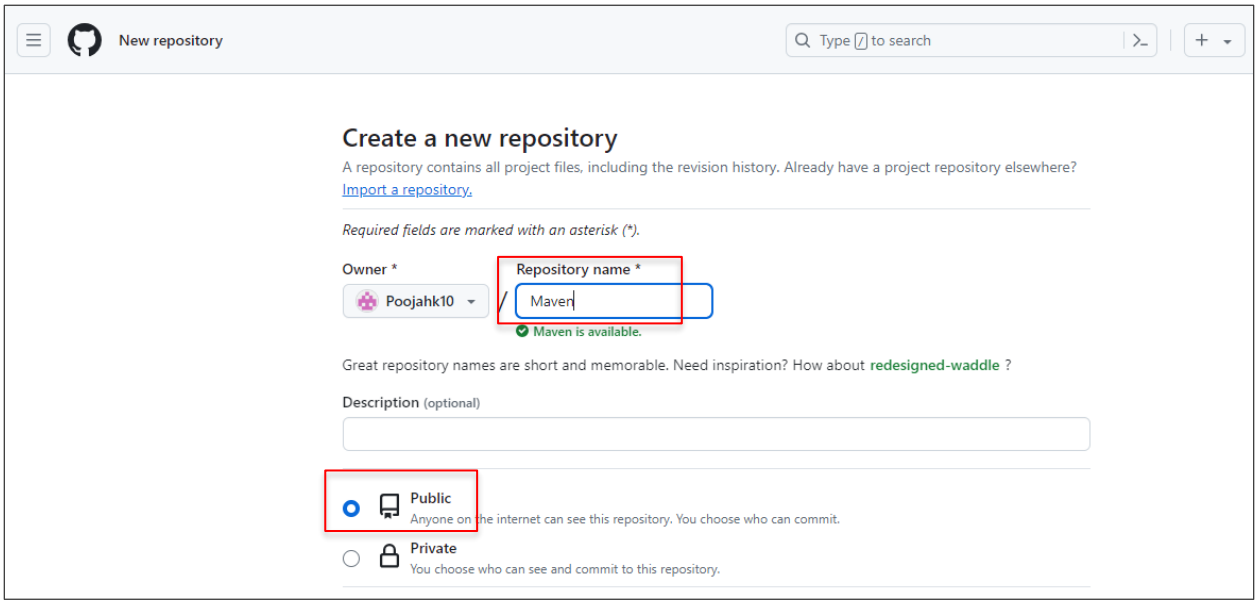


Note: If you do not have a GitHub account, visit the official website at **https://github.com/signup** and create a new account

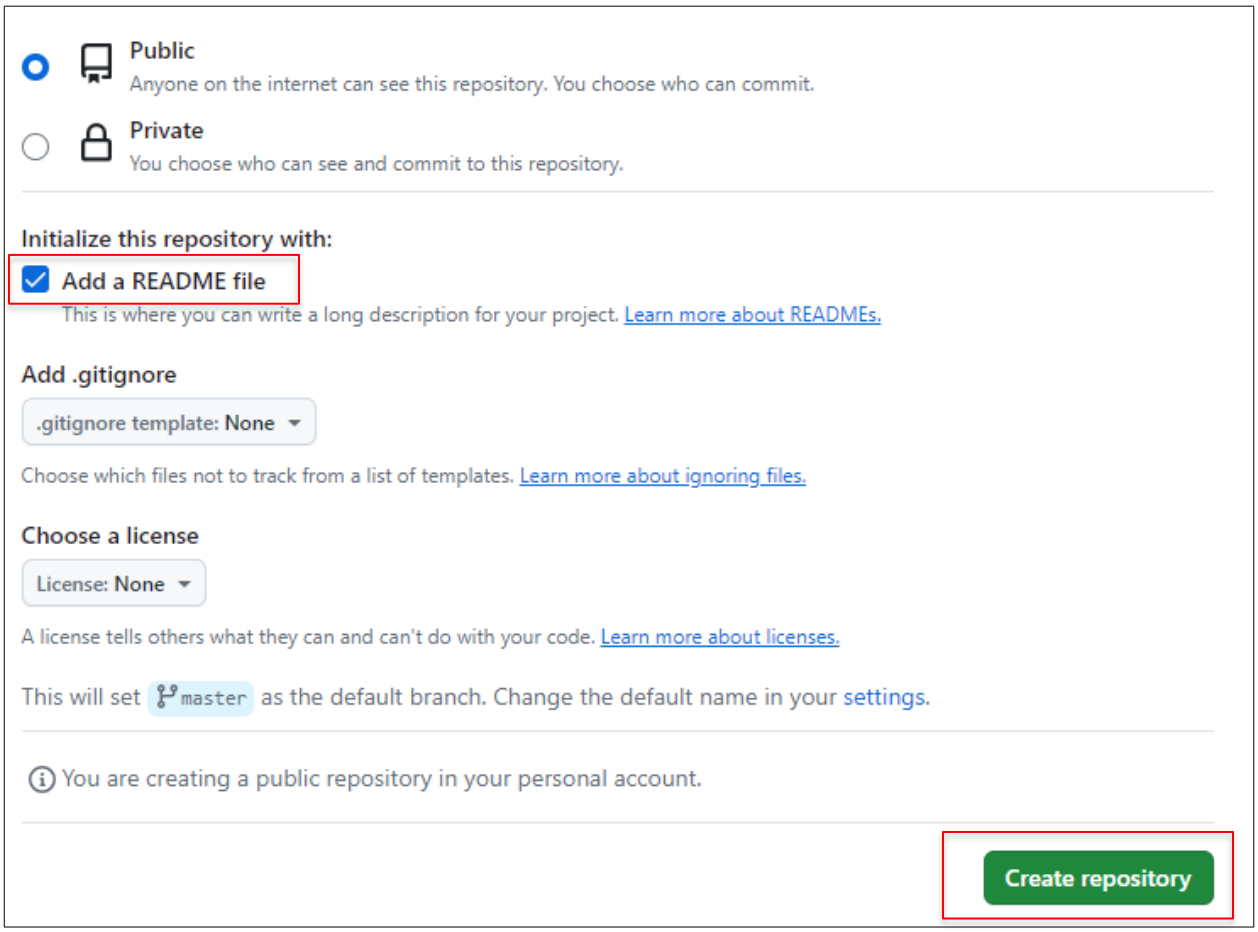
1.2 Click on **New** as shown in the screenshot below:



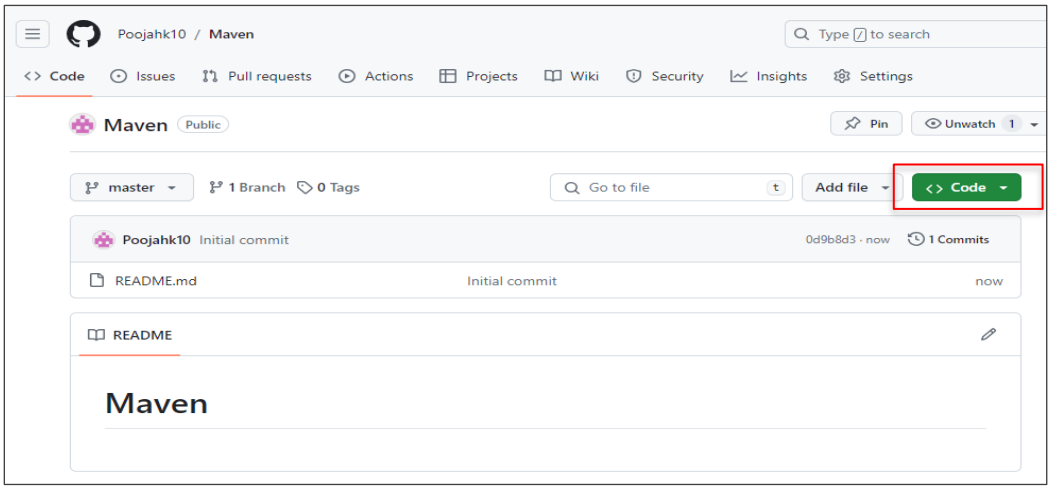
1.3 Enter a desired name for your repository and choose **Public** as shown in the screenshot below:



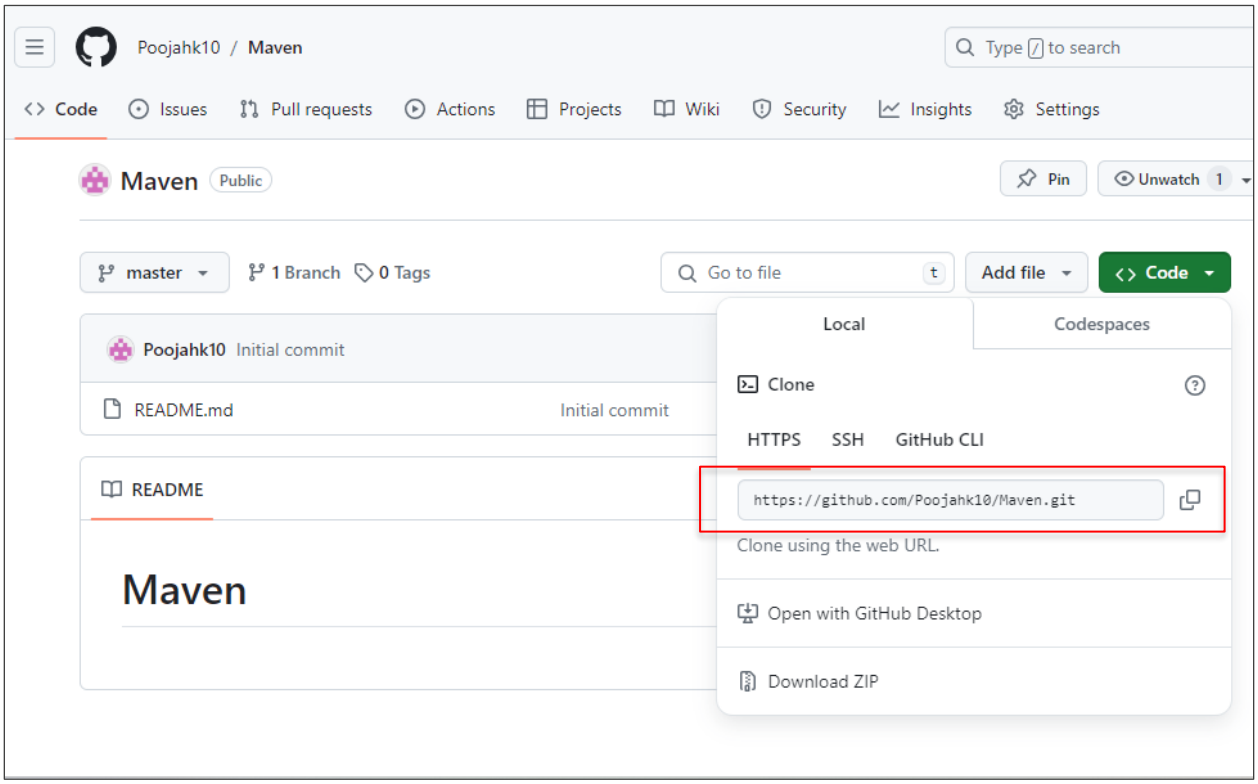
1.4 Click on **Add a README file** and then click on **Create repository** as shown in the screenshot below:



1.5 Click on **Code** as shown in the screenshot below:



1.6 Copy the repository URL as shown in the screenshot below:



Step 2: Push the pipeline script into the Git repository

2.1 Open the Linux terminal in your lab and clone the repository using the below command:
git clone RepositoryURL

```
syedsharozsimpl@ip-172-31-40-171:~$ git clone https://github.com/[redacted]ven.git
Cloning into 'Maven'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
syedsharozsimpl@ip-172-31-40-171:~$
```

2.2 Navigate inside the repository that you had created using the below command:
cd RepositoryName

```
syedsharozsimpl@ip-172-31-40-171:~$ cd Maven
syedsharozsimpl@ip-172-31-40-171:~/Maven$
```

2.3 Initialize the Git using the below command:

git init

```
syedsharozsimpl@ip-172-31-40-171:~/Maven$ git init
Reinitialized existing Git repository in /home/syedsharozsimpl/Maven/.git/
syedsharozsimpl@ip-172-31-40-171:~/Maven$
```

2.4 Create a file using the below command:

nano demofile

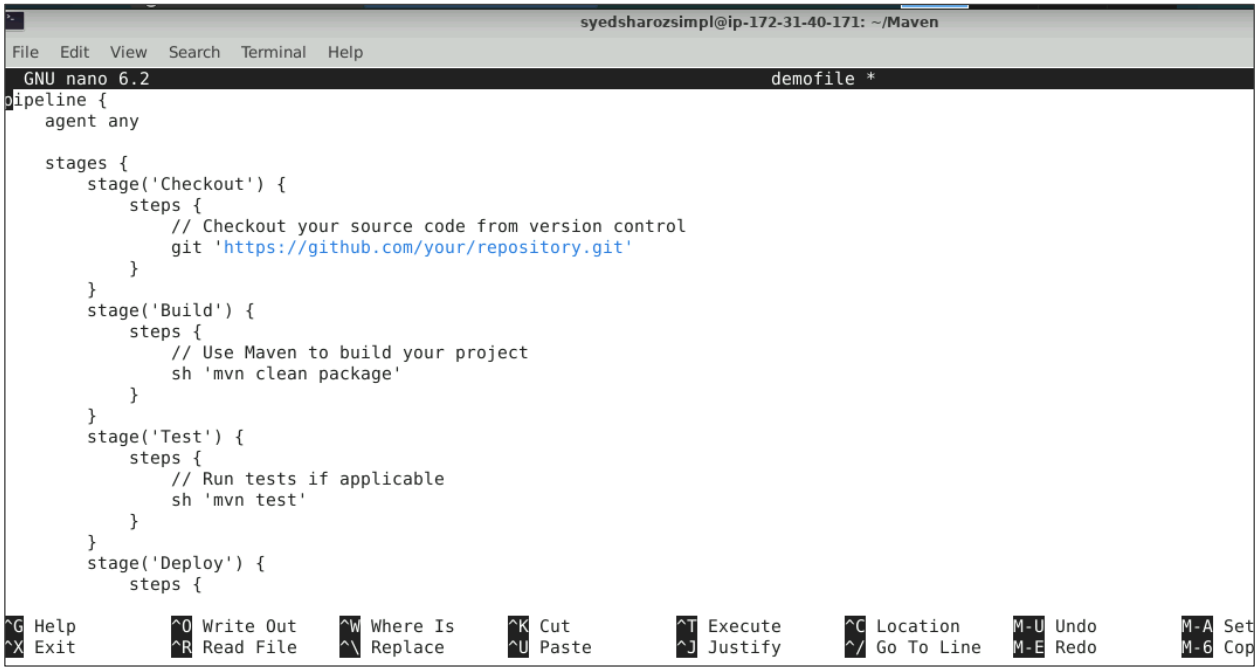
```
syedsharozsimpl@ip-172-31-40-171:~/Maven$ nano demofile
```

2.5 Paste the below pipeline script inside the file as shown in the screenshot below:

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                // Checkout your source code from version control
                git 'https://github.com/your/repository.git'
            }
        }
        stage('Build') {
            steps {
                // Use Maven to build your project
                sh 'mvn clean package'
            }
        }
        stage('Test') {
            steps {
                // Run tests if applicable
                sh 'mvn test'
            }
        }
        stage('Deploy') {
            steps {
                // Deploy your artifact, if necessary
                // Example: sh 'mvn deploy'
            }
        }
    }
```

```
    post {
        success {
            // This block will be executed if the pipeline runs successfully
            echo 'Pipeline executed successfully!'
        }
        failure {
            // This block will be executed if the pipeline fails
            echo 'Pipeline failed!'
        }
    }
}
```

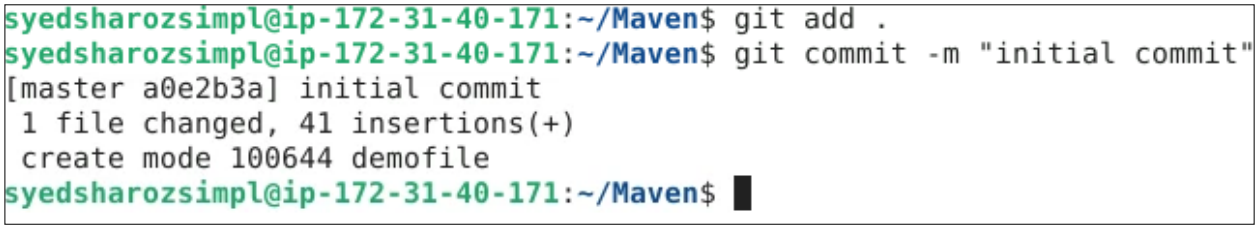
Note: Ensure you provide your Git repository URL on line7, save, and exit the page by clicking on **ctrl+S** to save and **ctrl+X** to exit



2.6 Stage and commit the changes using the below commands:

git add .

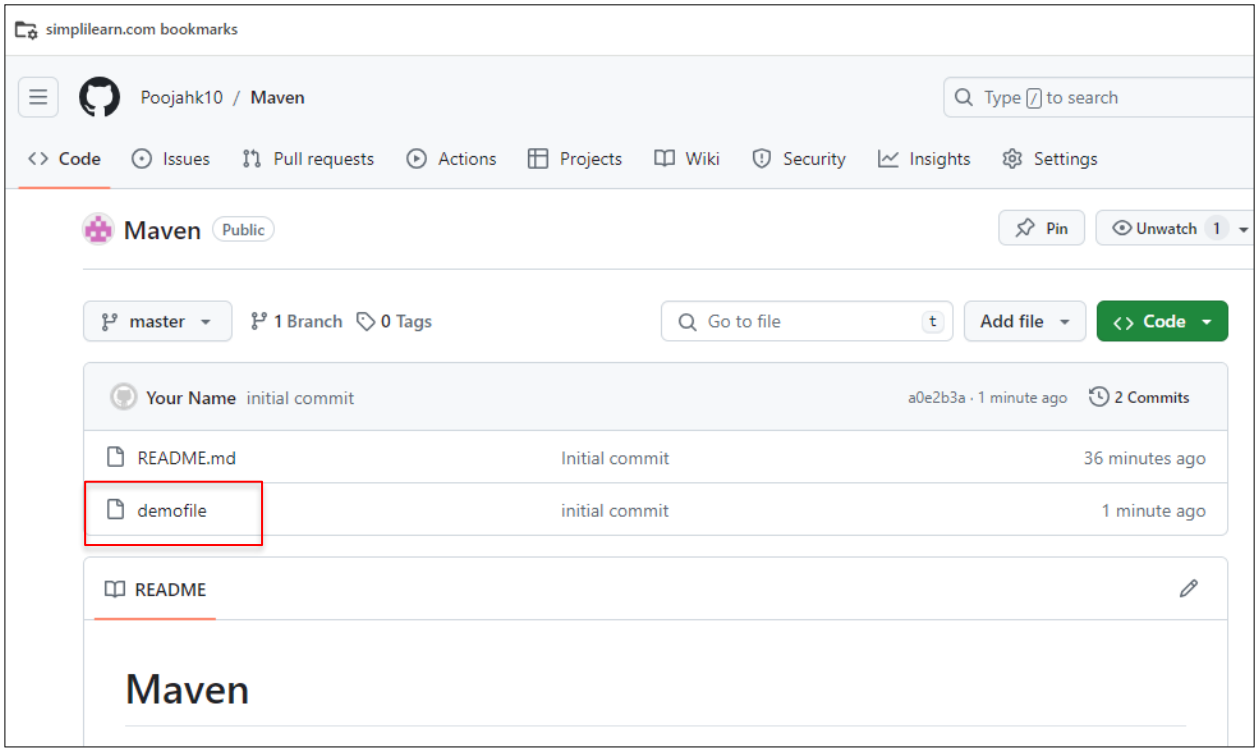
git commit -m “initial commit”



2.7 Push the file to the Git repository using the below command:
git push

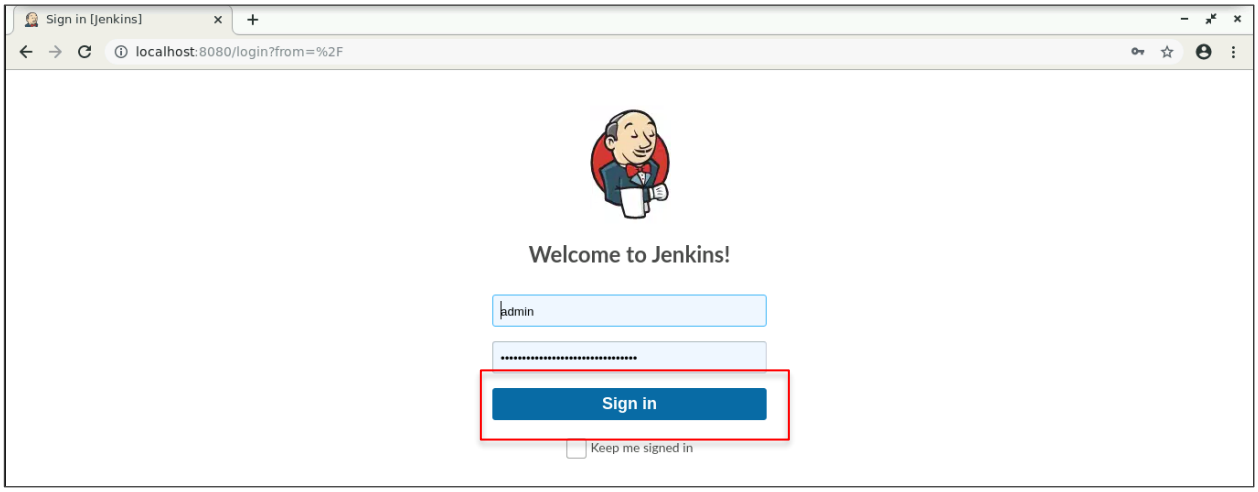
```
syedsharozsimpl@ip-172-31-40-171:~/Maven$ git push
Username for 'https://github.com': Poojahk10
Password for 'https://Poojahk10@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 614 bytes | 614.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Poojahk10/Maven.git
   0d9b8d3..a0e2b3a  master -> master
syedsharozsimpl@ip-172-31-40-171:~/Maven$
```

2.8 Navigate to your Git repository to check for the file that is pushed as shown in the screenshot below:



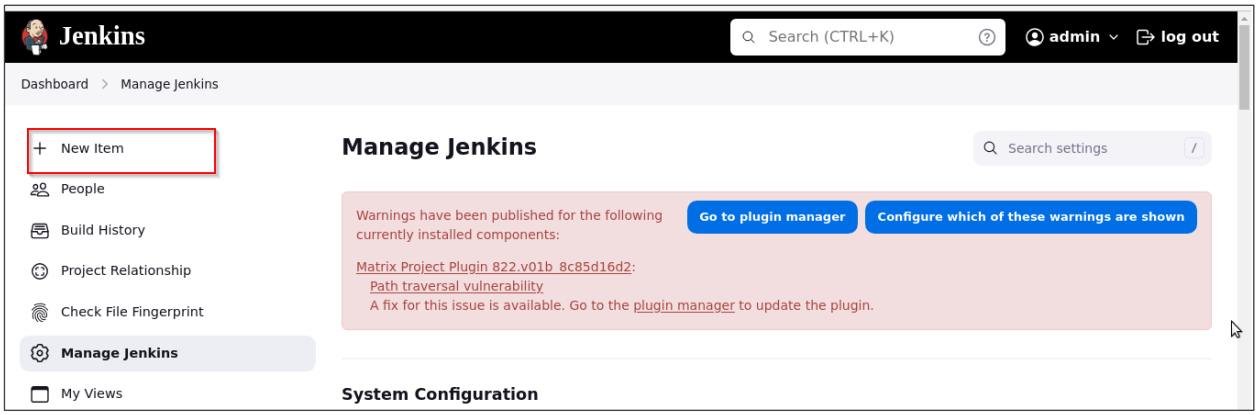
Step 3: Create a pipeline script-based freestyle job

3.1 Open the browser, go to the Jenkins **Dashboard** by typing **localhost:8080** in your browser, provide the credentials, and click the **Sign in** button

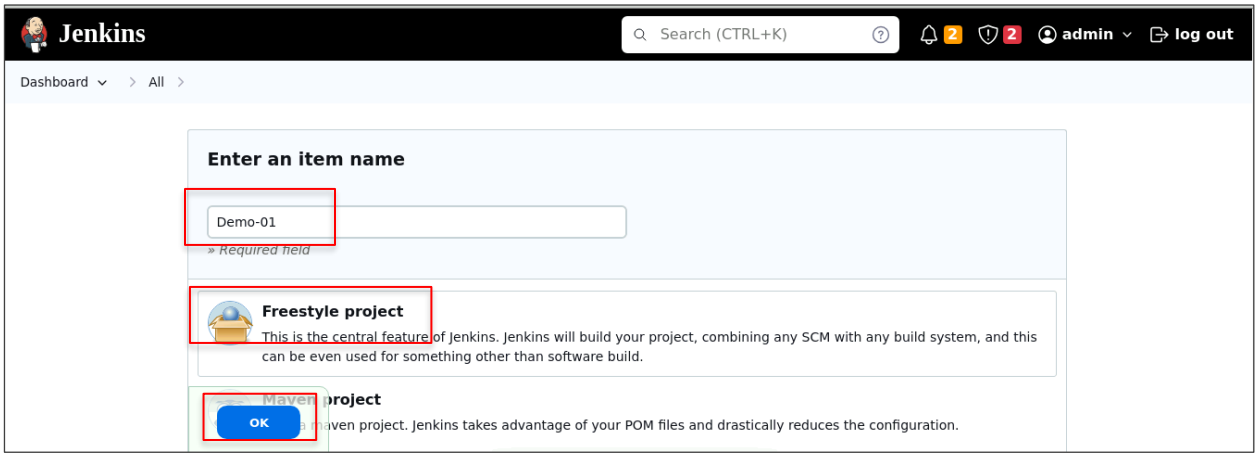


Note: Use the given credentials to access Jenkins in the lab: **Username** is admin and **Password** is admin

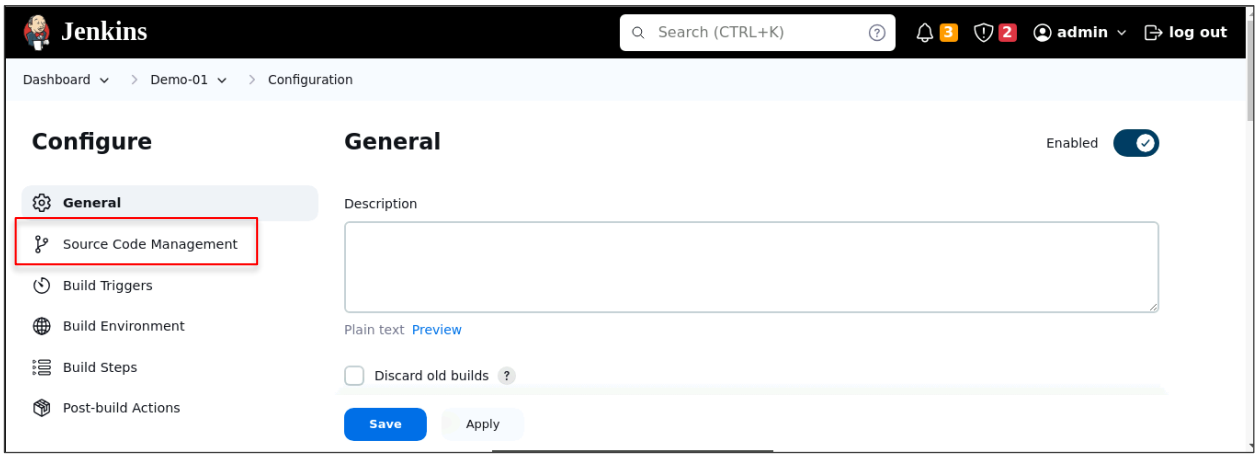
3.2 Click on **New Item** as show in the screenshot below:



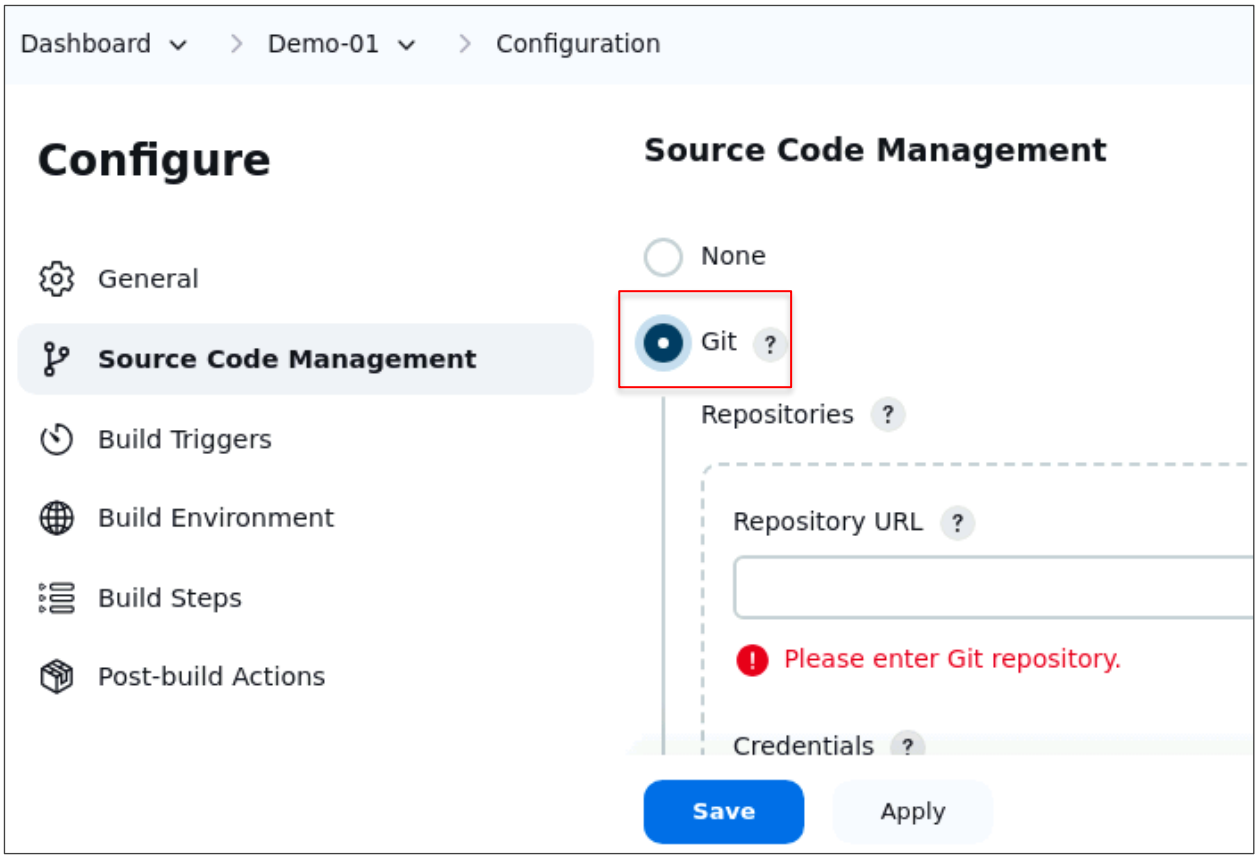
3.3 Enter a desired name for the project, select **Freestyle project**, and then click on **OK** as shown in the screenshot below:



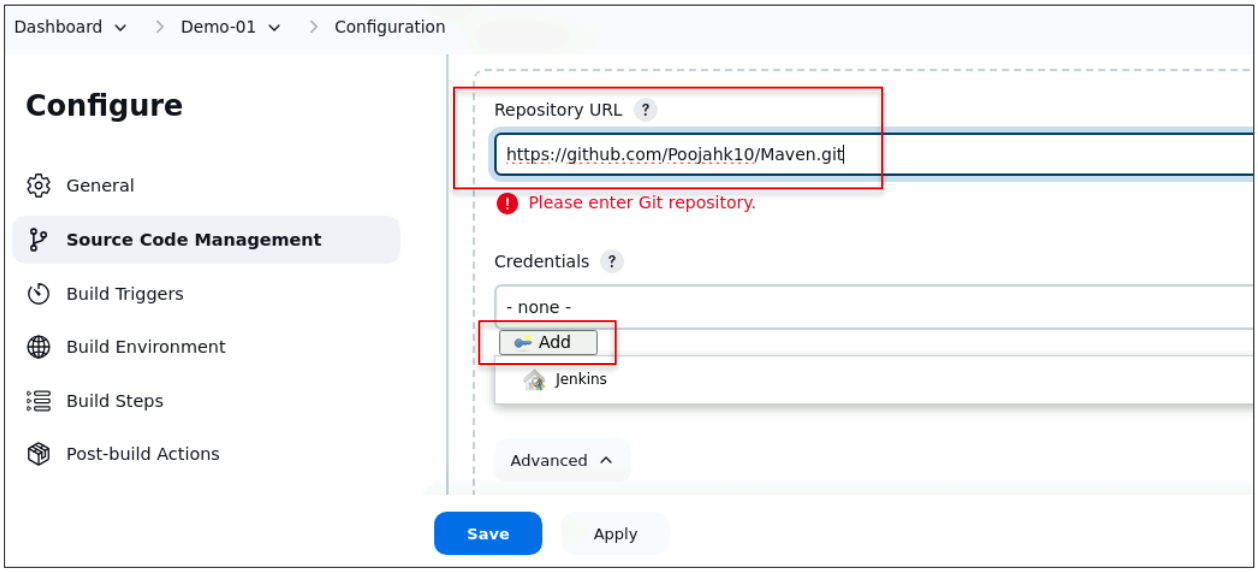
3.4 Click on **Source Code Management** as shown in the screenshot below:



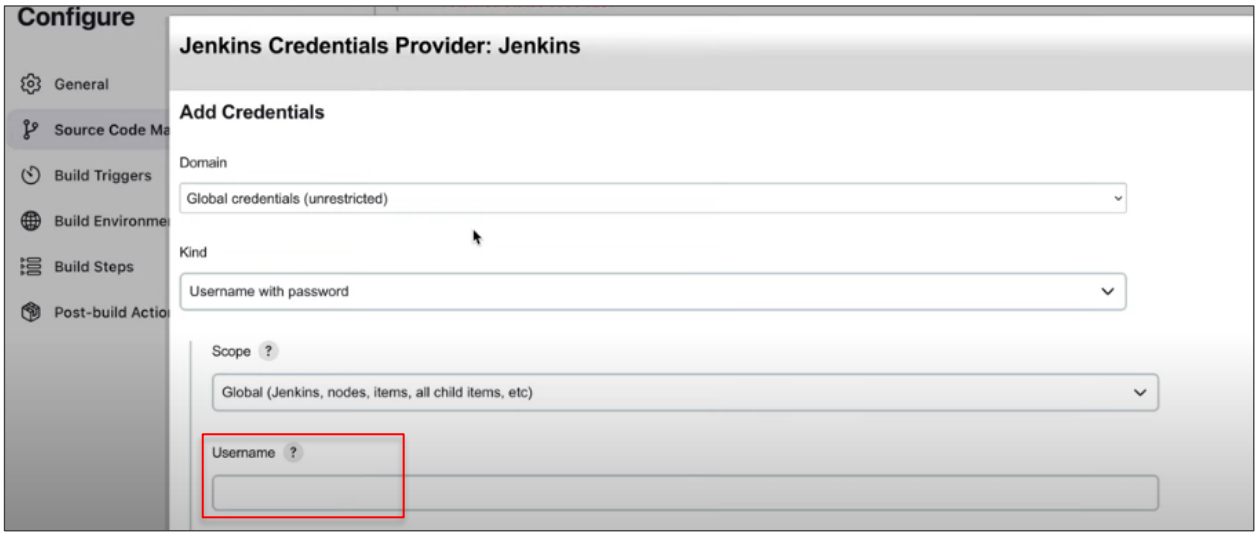
3.5 Click on **Git** as shown in the screenshot below:



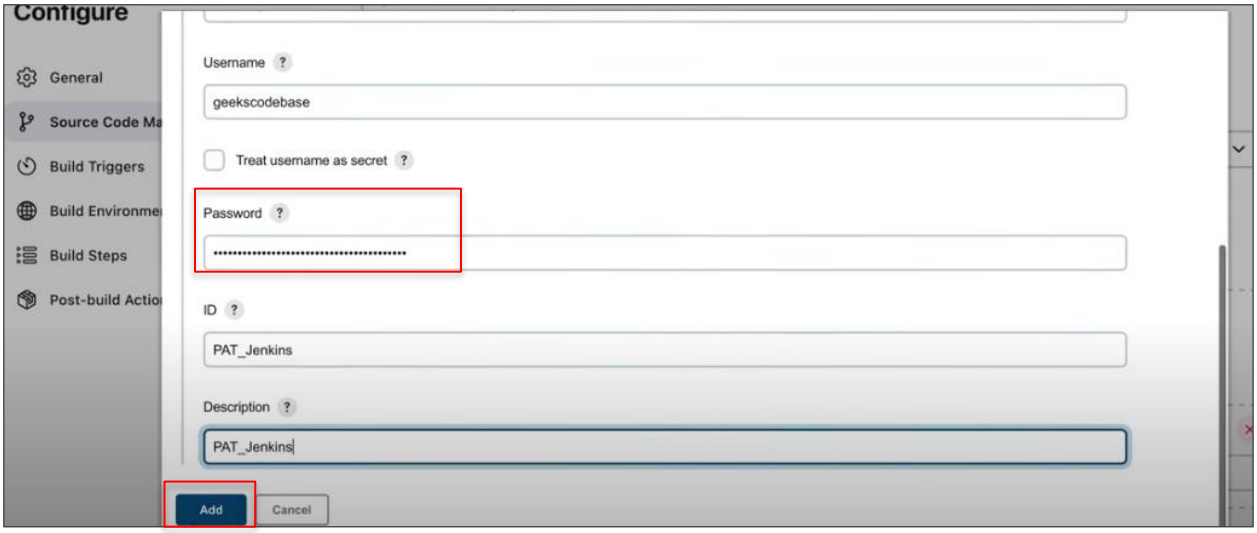
3.6 Enter the repository URL and click on **Add** as shown in the screenshot below:



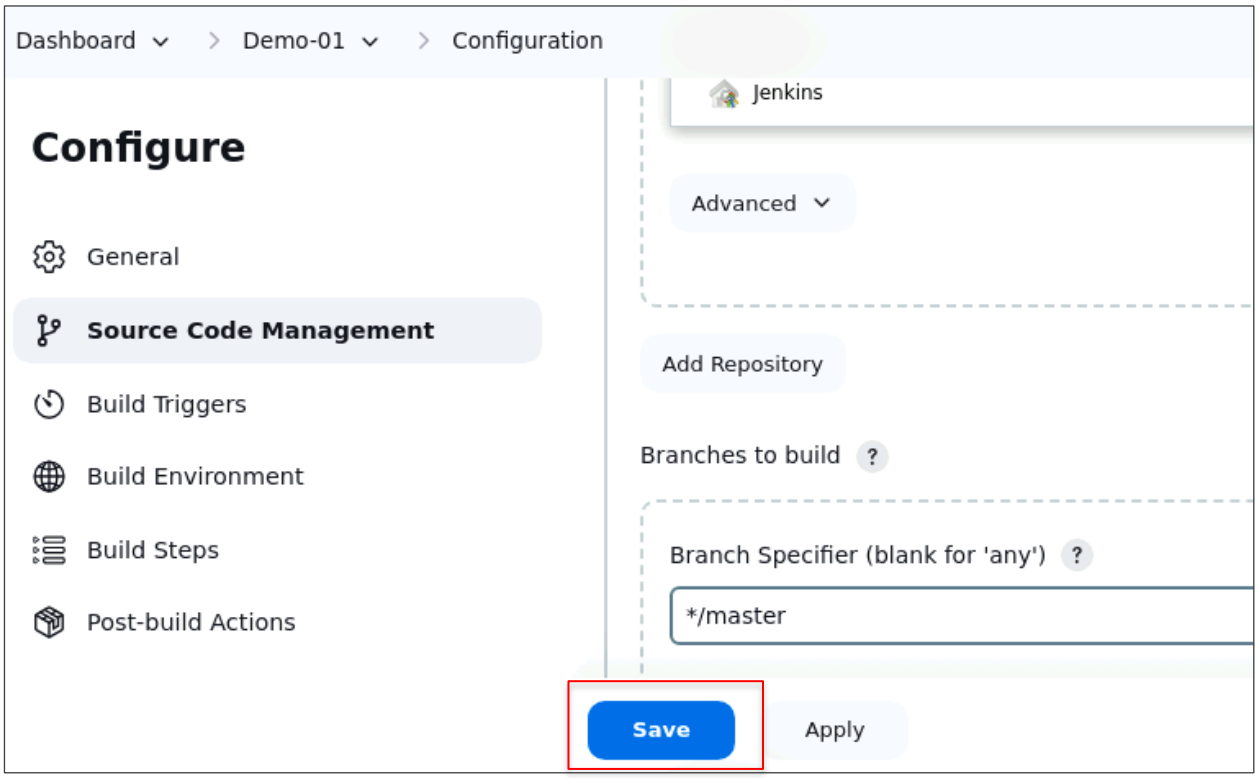
3.7 Enter a desired name for the **Username** as shown in the screenshot below:



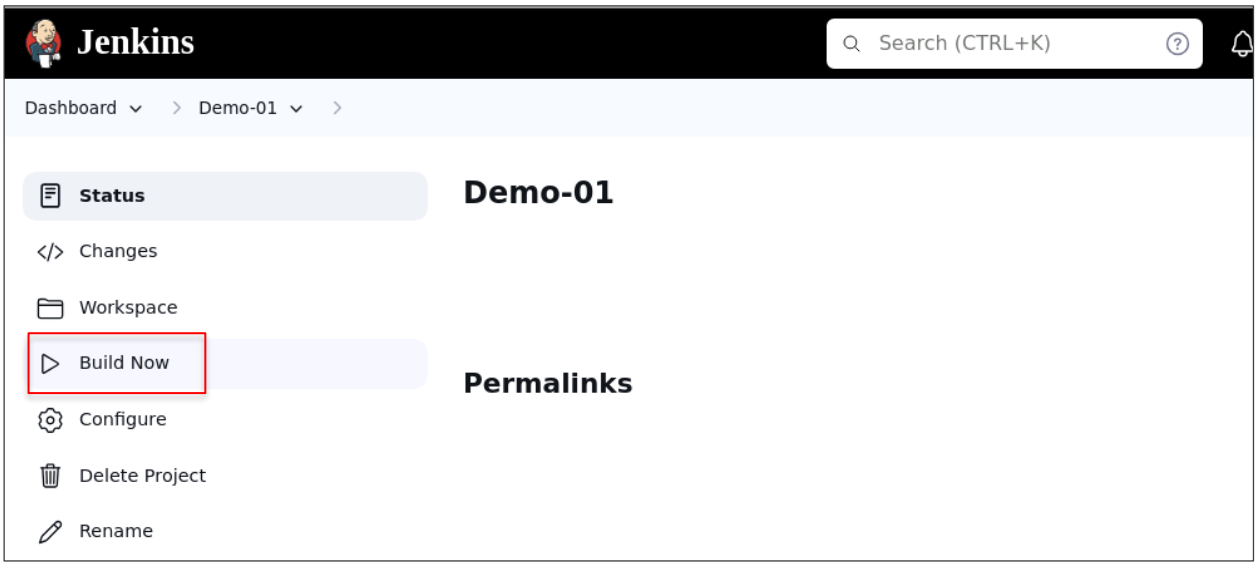
3.8 Paste the Git token under **Password** section and then click on **Add** as shown in the screenshot below**:**



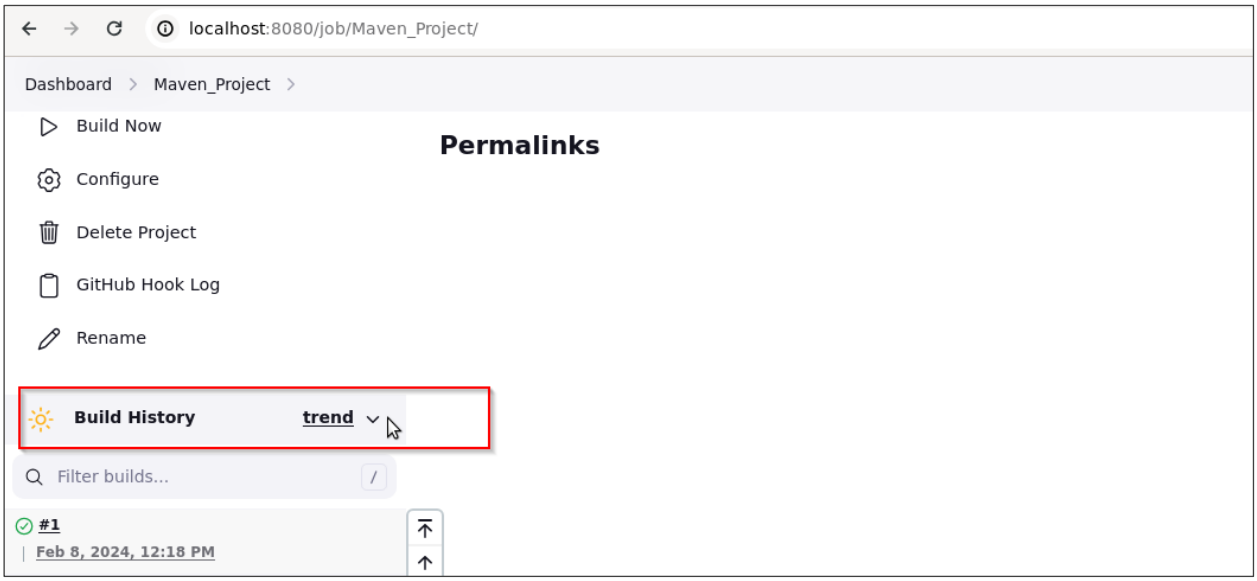
3.9 Click on **Save** as shown in the screenshot below:



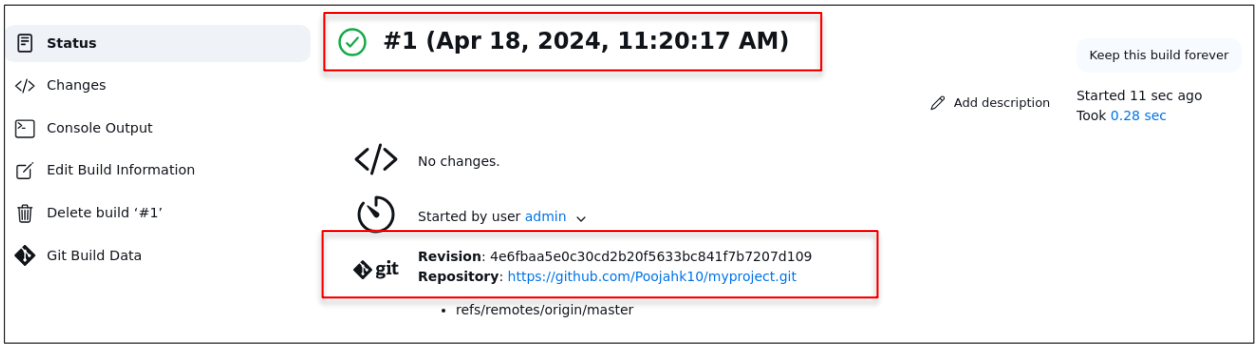
3.10 Click on **Build Now** in the left section after the creation of the job as shown in the below screenshot:



3.11 Click on **Build History** as shown in the screenshot below:



3.12 Verify that the output indicates the status (successful) as shown in the screenshot below:



By following these steps, you have successfully set up the Jenkins freestyle job from the Git version control system to enable automated CI for building, testing, and potentially deploying software upon code changes.