

- 1) Clone the maven project

git clone <https://github.com/amtopenwriteup/mvnproj.git>

```
root@amit:~# git clone https://github.com/amtopenwriteup/mvnproj.git
Cloning into 'mvnproj'...
remote: Enumerating objects: 144, done.
remote: Counting objects: 100% (144/144), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 144 (delta 64), reused 144 (delta 64), pack-reused 0
Receiving objects: 100% (144/144), 31.70 KiB | 400.00 KiB/s, done.
Resolving deltas: 100% (64/64), done.
root@amit:~#
```

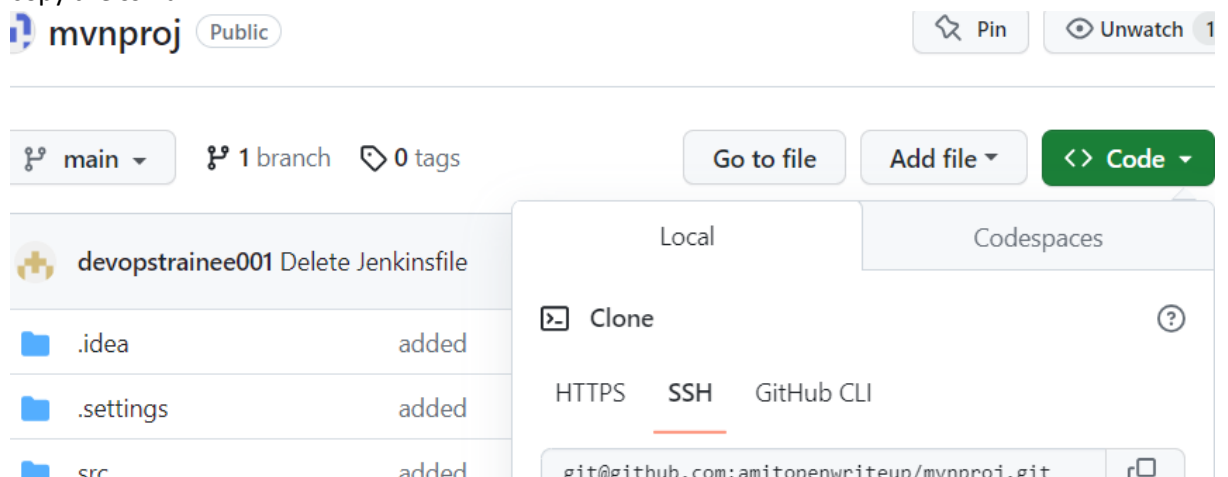
- 2) cd mvnproj/
git remote

```
root@amit:~# cd mvnproj/
root@amit:~/mvnproj# git remote
origin
root@amit:~/mvnproj#
```

- 3) git remote remove origin

```
root@amit:~/mvnproj# git remote remove origin
root@amit:~/mvnproj#
```

- 4) create a new repository in github named mvnproj
- 5) copy the ssh url



- 6) add the url
git remote add origin <url>
- 7) run the command git push

```

root@amit:~/java-mvn-proj# git push -u origin main
Enumerating objects: 144, done.
Counting objects: 100% (144/144), done.
Delta compression using up to 4 threads
Compressing objects: 100% (71/71), done.
Writing objects: 100% (144/144), 31.70 KiB | 272.00 KiB/s, done.
Total 144 (delta 64), reused 139 (delta 59), pack-reused 0
remote: Resolving deltas: 100% (64/64), done.
To github.com:amitopenwriteup/mvnproj.git
 * [new branch]      main -> main

```

8) vi Jenkinsfile (J is in caps) and copy this code

```

pipeline {
  agent any
  stages {
    stage('Compiling and Running Test Cases') {
      steps {
        sh 'mvn clean'
        sh 'mvn compile'
        sh 'mvn test'
      }
    }
    stage('Creating Package') {
      steps {
        sh 'mvn package'
      }
    }
  }
}

```

```

root@amit:~/mvnproj# git add Jenkinsfile
root@amit:~/mvnproj# git commit -m "adding jenkinsfile"
[main 9ee9ba1] adding jenkinsfile
1 file changed, 23 insertions(+)
create mode 100644 Jenkinsfile


```

```

root@amit:~/mvnproj# git push origin main
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), 440 bytes | 440.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:amitopenwriteup/mvnproj.git
   39aed82..9ee9ba1  main -> main

```

11) Create pipeline



The screenshot shows the Jenkins job configuration page. At the top, there is a text input field containing 'pipelinemvn' and a 'Required field' message. Below this, there are two main options: 'Freestyle project' and 'Pipeline'. The 'Pipeline' option is selected and highlighted. It includes a description: '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.'

12) Configure below option

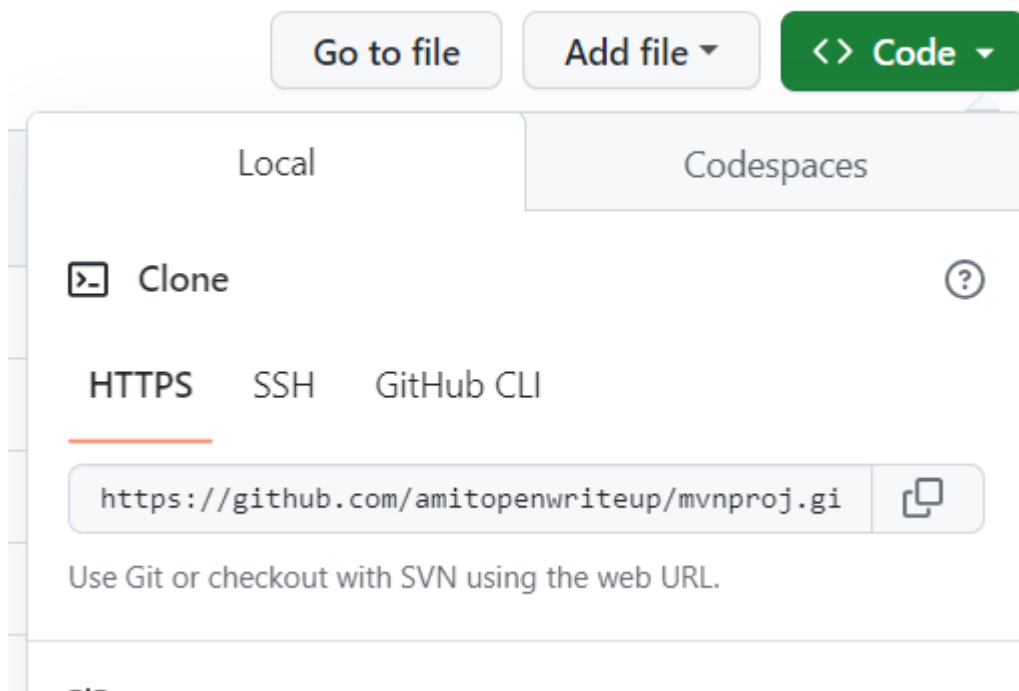
Pipeline

Definition



The screenshot shows the Jenkins Pipeline configuration page. Under the 'Definition' section, the 'Pipeline script from SCM' option is selected. Below this, there is a section for 'SCM' with a dropdown menu set to 'Git'. Under the 'Repositories' section, there is a 'Repository URL' field containing 'https://github.com/amitopenwriteup/mvnproj.git' and a 'Credentials' dropdown menu set to '- none -'. There is a red 'X' icon next to the 'Repository URL' field, indicating an error.

Repository url



Check the branch

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?