Module 7: Jenkins Pipeline

Demo Document - 1

edureka!

Demo: Pipeline as Code using Jenkisfile

Problem statement: How to deploy a JAVA application with Jenkins pipeline

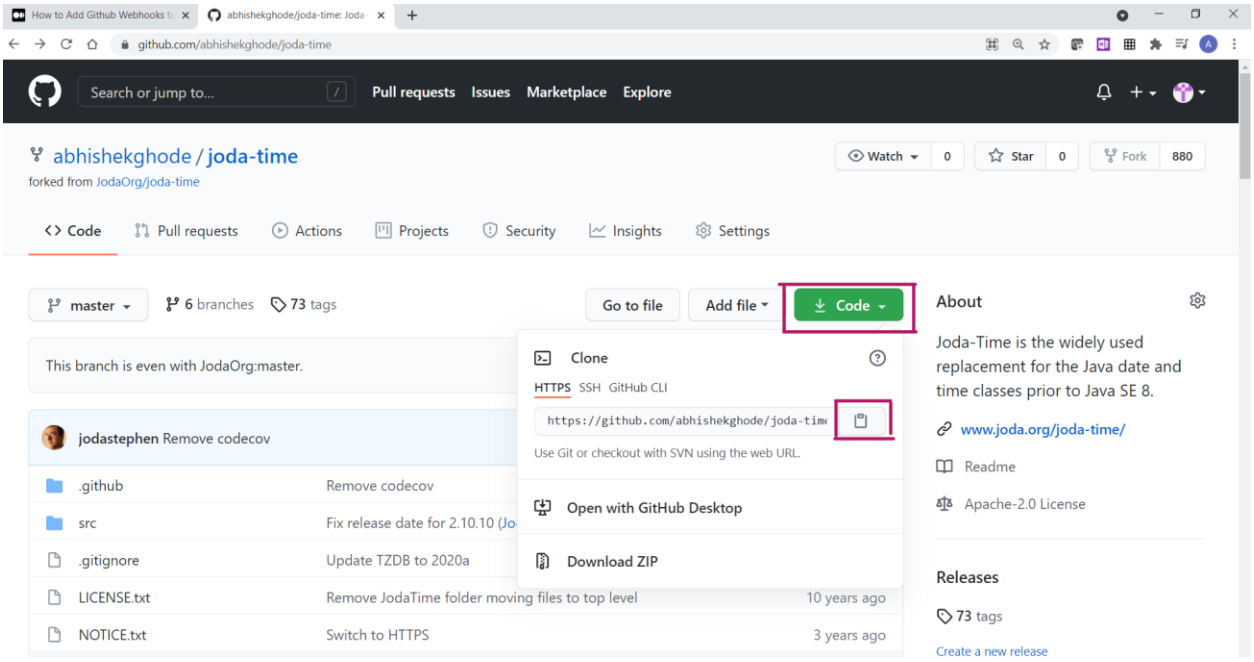
Solution:

Step 1: Adding Jenkinsfile into a git repository

URL to access: github.com

We already have forked joda-time repository in previous lab. Let's clone this repository on local workstation.

Click 'Code' and then copy repository url using copy button.



This forked repository can be cloned on local workstation. We will add a Jenkinsfile. We can commit and push the changes back into this forked repository.

This is called 'Pipeline as Code' which uses continuous delivery pipeline as part of the codebase to be versioned.

Module 7 – Jenkins Pipeline

Step 2: Clone the github repository on local workstation

Tool to access: git bash (you can use any other git client as well)

Commands to run: `git clone https://github.com/abhishekghode/joda-time.git`

ls

cd joda-time

ls

```
MINGW64:/c/AG/repos/joda-time

abghode@LIN51007171 MINGW64 /c/AG/repos
$ git clone https://github.com/abhishekghode/joda-time.git
Cloning into 'joda-time'...
remote: Enumerating objects: 26704, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 26704 (delta 17), reused 28 (delta 9), pack-reused 26652
Receiving objects: 100% (26704/26704), 10.91 MiB | 8.79 MiB/s, done.
Resolving deltas: 100% (13433/13433), done.

abghode@LIN51007171 MINGW64 /c/AG/repos
$ ls
joda-time/

abghode@LIN51007171 MINGW64 /c/AG/repos
$ cd joda-time/

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ ls
LICENSE.txt  NOTICE.txt  pom.xml  README.md  RELEASE-NOTES.txt  src/
```

These steps will help us to clone and view codebase for joda-time library.

Step 3: Adding a Jenkinsfile into local repository

Commands to access: nano Jenkinsfile (you can use another text editor as well)

```
MINGW64:/c/AG/repos/joda-time

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ nano Jenkinsfile
```

Module 7 – Jenkins Pipeline

Currently Jenkinsfile is empty. Please add below mentioned content to it.

```
pipeline
{
                agent any

                tools {
                    // Install the Maven version "M3" and add to the path
                    maven "Maven 3"
                }

                stages {
                    stage('Build') {
                        steps {
                            // Get some code from a GitHub repository
                            git 'https://github.com/abhishekghode/joda-time.git'

                            // Run Maven on a Unix agent.
                            //sh "mvn -Dmaven.test.failure.ignore=true clean package"

                            // To run Maven on a Windows agent
                            bat "mvn -Dmaven.test.failure.ignore=true clean package"
                        }

                        post {

                                // Record the test results
                            success {
                                junit '**/target/surefire-reports/TEST-*.xml'
                                  //Archive the jar
                                archiveArtifacts 'target/*.jar'
                            }
                        }
                    }
                }
            }
```

```
MINGW64:/c/AG/repos/joda-time
  GNU nano 2.9.3                                   Jenkinsfile

pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "Maven 3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/abhishekghode/joda-time.git'

                // Run Maven on a Unix agent.
                //sh "mvn -Dmaven.test.failure.ignore=true clean package"

                // To run Maven on a Windows agent, use
                bat "mvn -Dmaven.test.failure.ignore=true clean package"
            }

            post {
                                                    [ Read 33 lines ]
```

You can save the contents of Jenkinsfile using below keyboard shortcuts if you are using nano text editor.

[ctrl] + x; type y; press [enter]

This declarative pipeline script is similar which we have already used in previous module. Please observe the github repository. Its should be owned by you. Otherwise you may not be able to push changes here.

Step 4: Commit changes locally into git repository and push the changes to github repository.

Commands to run: git status

git add ./Jenkinsfile

git status

Module 7 – Jenkins Pipeline

```
MINGW64:/c/AG/repos/joda-time

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Jenkinsfile

nothing added to commit but untracked files present (use "git add" to track)

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git add ./Jenkinsfile
warning: LF will be replaced by CRLF in Jenkinsfile.
The file will have its original line endings in your working directory.

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   Jenkinsfile
```

Commands to run: git commit -m “adding Jenkinsfile”

git remote -v

git push origin master (you may need to authenticate using github credentials)

```
MINGW64:/c/AG/repos/joda-time

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git commit -m "adding Jenkinsfile"
[master c443373e] adding Jenkinsfile
 1 file changed, 33 insertions(+)
 create mode 100644 Jenkinsfile

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git remote -v
origin  https://github.com/abhishekghode/joda-time.git (fetch)
origin  https://github.com/abhishekghode/joda-time.git (push)

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git push origin master
```

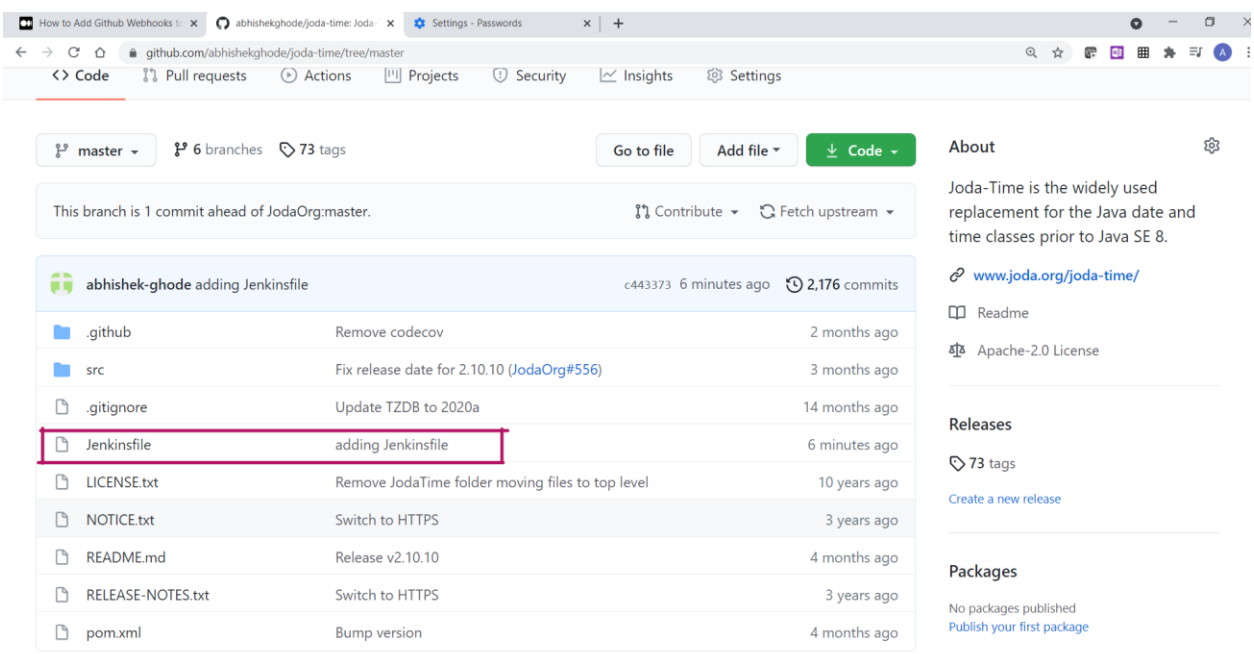
Module 7 – Jenkins Pipeline

Please ensure that ‘git push’ command executed successfully.

```
MINGW64:/c/AG/repos/joda-time

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git commit -m "adding Jenkinsfile"
[master c443373e] adding Jenkinsfile
 1 file changed, 33 insertions(+)
 create mode 100644 Jenkinsfile

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git remote -v
origin  https://github.com/abhishekghode/joda-time.git (fetch)
origin  https://github.com/abhishekghode/joda-time.git (push)

abghode@LIN51007171 MINGW64 /c/AG/repos/joda-time (master)
$ git push origin master
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': abhishekghode
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 704 bytes | 704.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/abhishekghode/joda-time.git
   27edfffa..c443373e  master -> master
```

Step 5: Check Jenkinsfile on github repository

URL to access: https://github.com/abhishekghode/joda-time



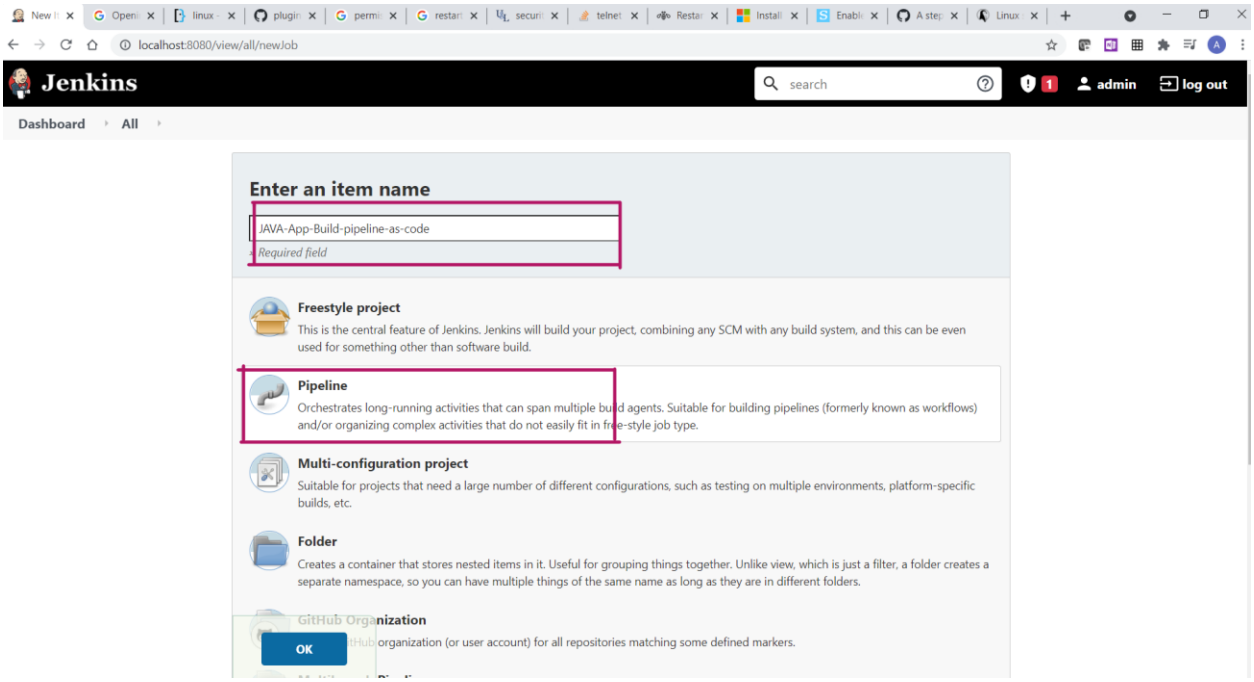
Now, your github repository is ready to be used for ‘pipeline as code’ since it has a Jenkisfile available in the codebase.

Module 7 – Jenkins Pipeline

Step 6: Create a pipeline project on Jenkins

URL to access: http://localhost:8080/view/all/newJob

Please ensure to start Jenkins server prior to accessing this url.

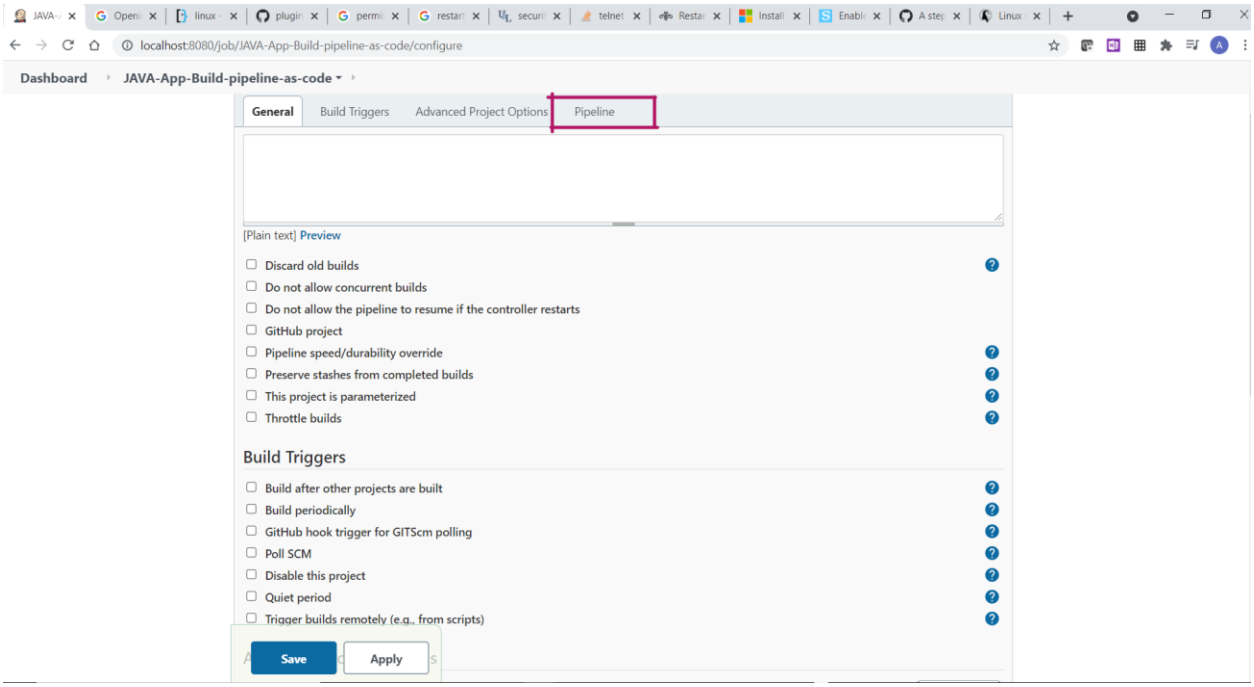


Step 6: Configuring pipeline project to specify pipeline script

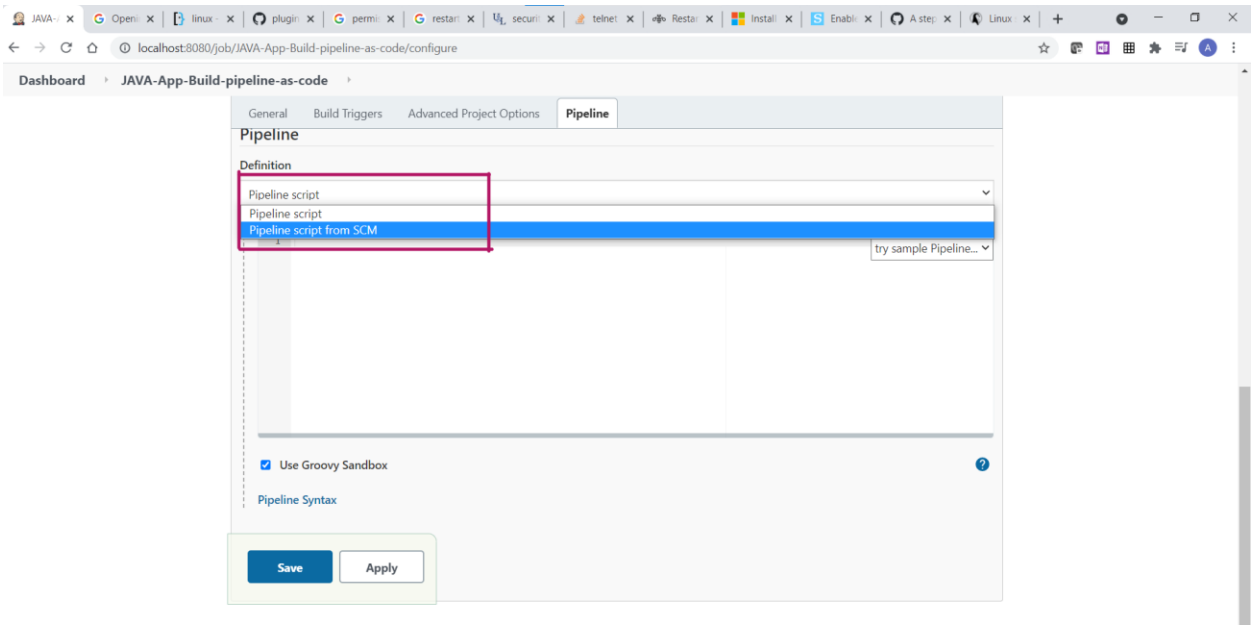
URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-as-code/configure

Click on 'Pipeline'

Module 7 – Jenkins Pipeline



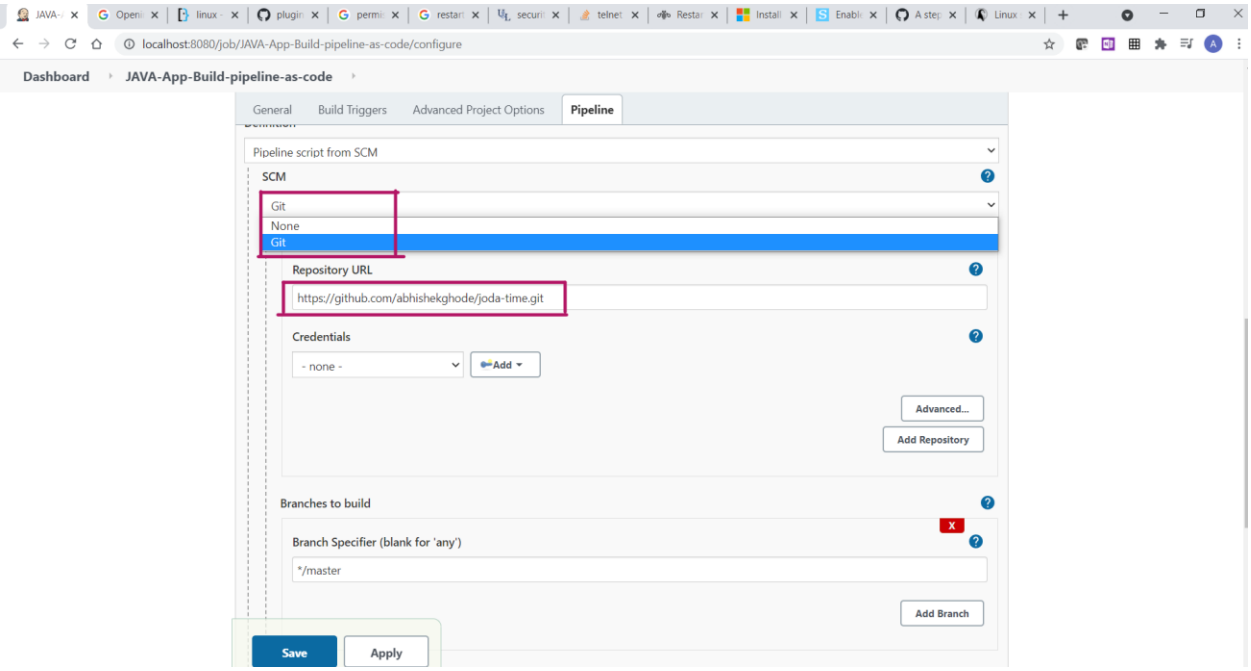
Please select 'Pipeline script from SCM' option from drop-down widget



Module 7 – Jenkins Pipeline

SCM should be selected as ‘Git’

Also please enter gihub repository url. And click on ‘Save’



Module 7 – Jenkins Pipeline

Step 7: Build the pipeline project

URL to access: http://localhost:8080/job/JAVA-App-Build-pipeline-as-code/

Click on 'Build Now'