Lesson 5 Demo 3: Building Continuous Integration Pipelines In Jenkinsfile

This section will guide you to:

* Build Continuous Integration Pipelines in jenkins

This lab has two sub-sections, namely:

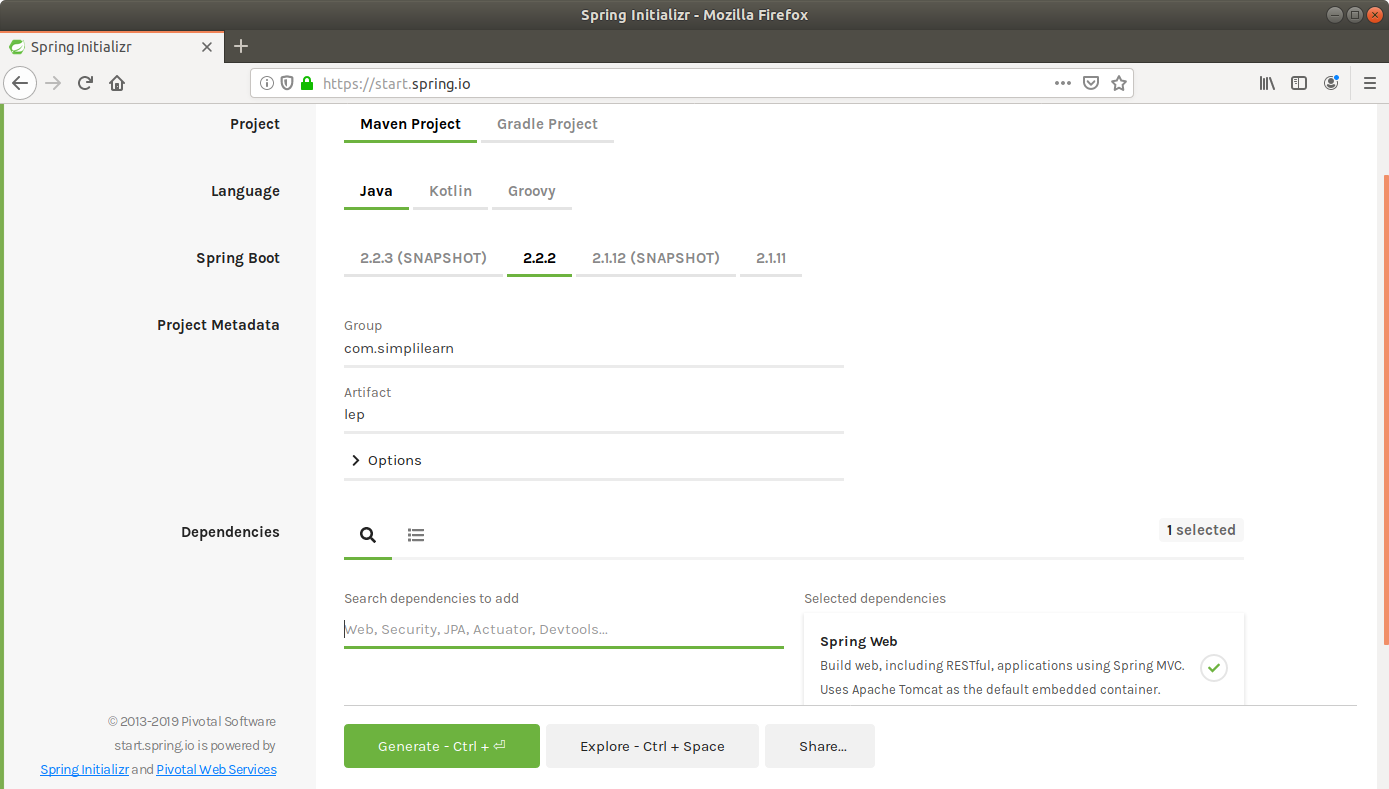
1. Building a Maven project
2. Creating and committing a Jenkinsfile
3. Defining a Pipeline to build the project

**Step 1:** Building a Maven project

* Login to your **Github** account.
* Click on the **plus** icon next to the profile picture and select ***New repository*** from the drop-down menu.



* Fill the required fields in the create repository form.
* Click on the **Create Repository** button.
* Click on the **Clone or download** button and copy the URL.
* Go to ​start.​spring.​io/​



* Select **Maven** as the project type.
* Fill Group and Artifact with appropriate values. For example, ***com.simplilearn*** and ***Calculator1.***
* Add **Web (Spring Web)** to Dependencies.
* Select Packaging as: **jar**
* Select Java: **8**
* Click on **Generate Project.**
* The generated skeleton project should be downloaded as a zip file.
* Open the terminal
* Run **git clone [URL]** to clone the repository.
* Unzip the downloaded spring boot project to the cloned repository.
* (cd Downloads  
  unzip Calculator1.zip

Copy the contents of Calculator1 folder present in downloads and paste it into your repository folder)

* Commit the changes to the remote SCM.
* Run **git add .**
* Run **git commit -m “Add logic and test”**
* Run **git push -u origin master**

**Step 2**: Creating and committing a Jenkinsfile

* Navigate to the root directory where the pom.xml is.
* Open a new text file and add the following script to it.

**pipeline {**

**agent any**

**stages {**

**stage("Compile") {**

**steps {**

**sh "mvn compile"**

**}**

**}**

**stage("Unit test") {**

**steps {**

**sh "mvn test"**

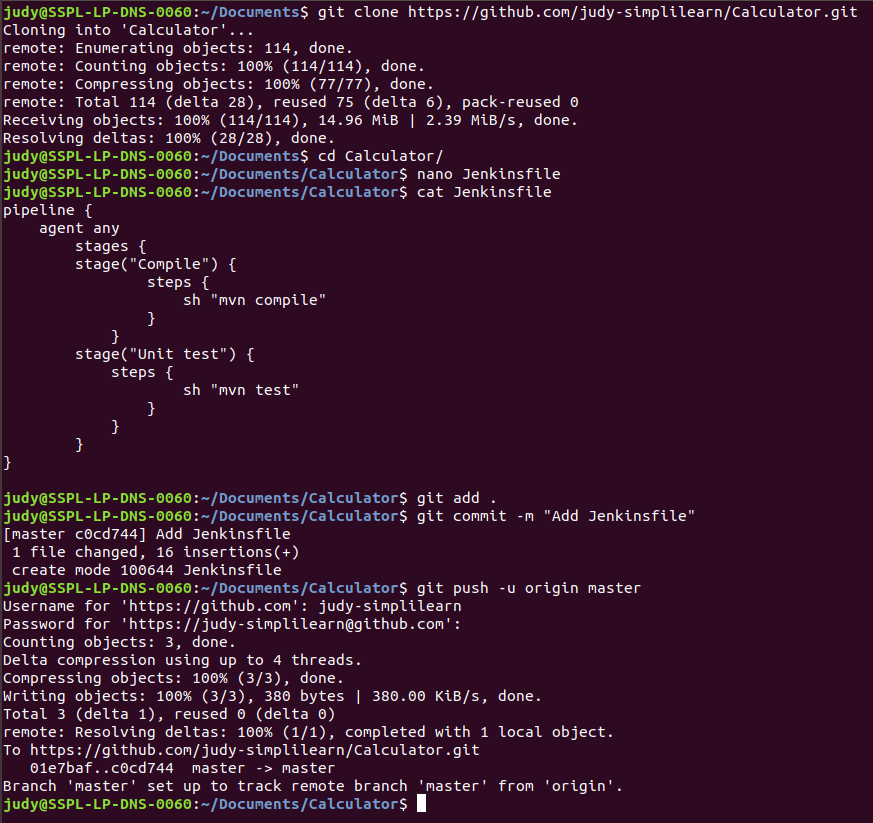
**}**

**}**

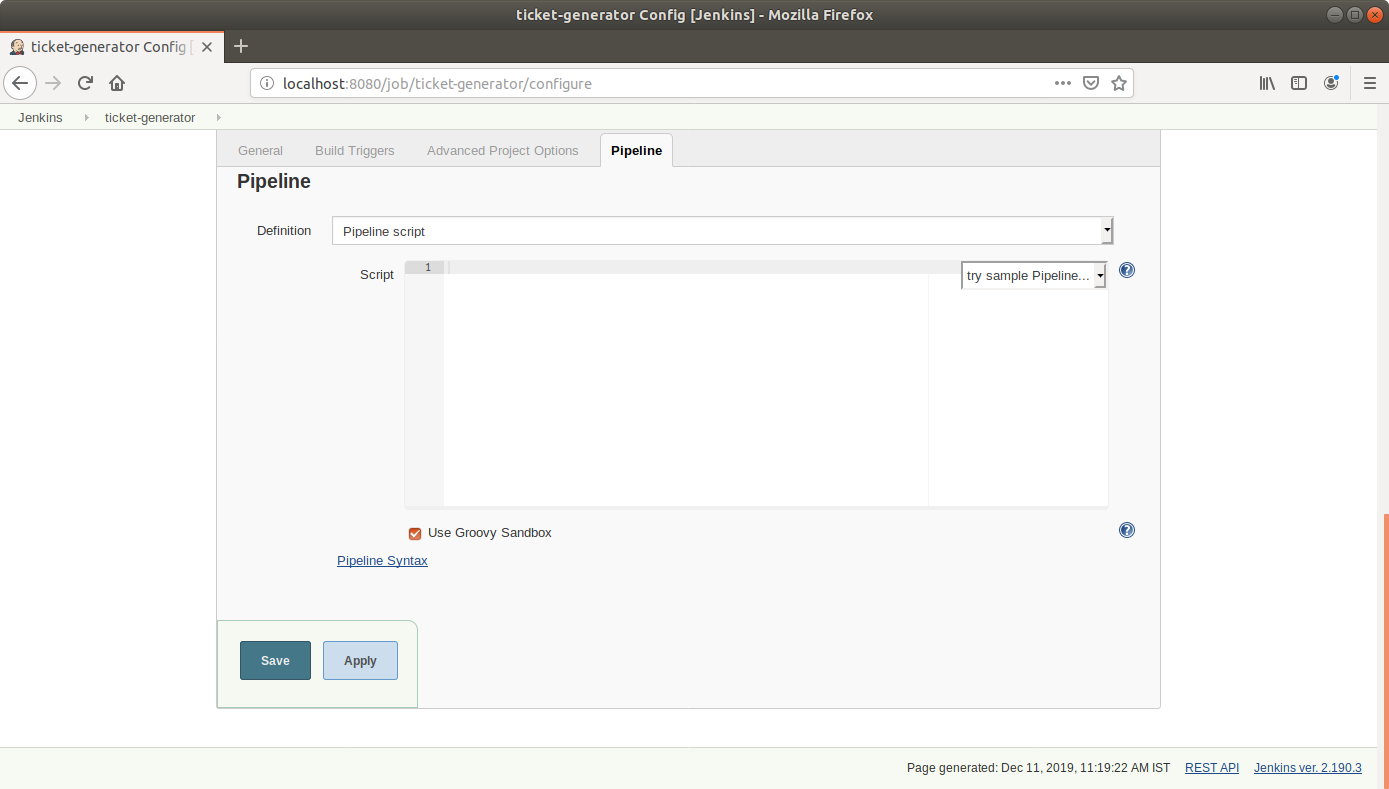
**}**

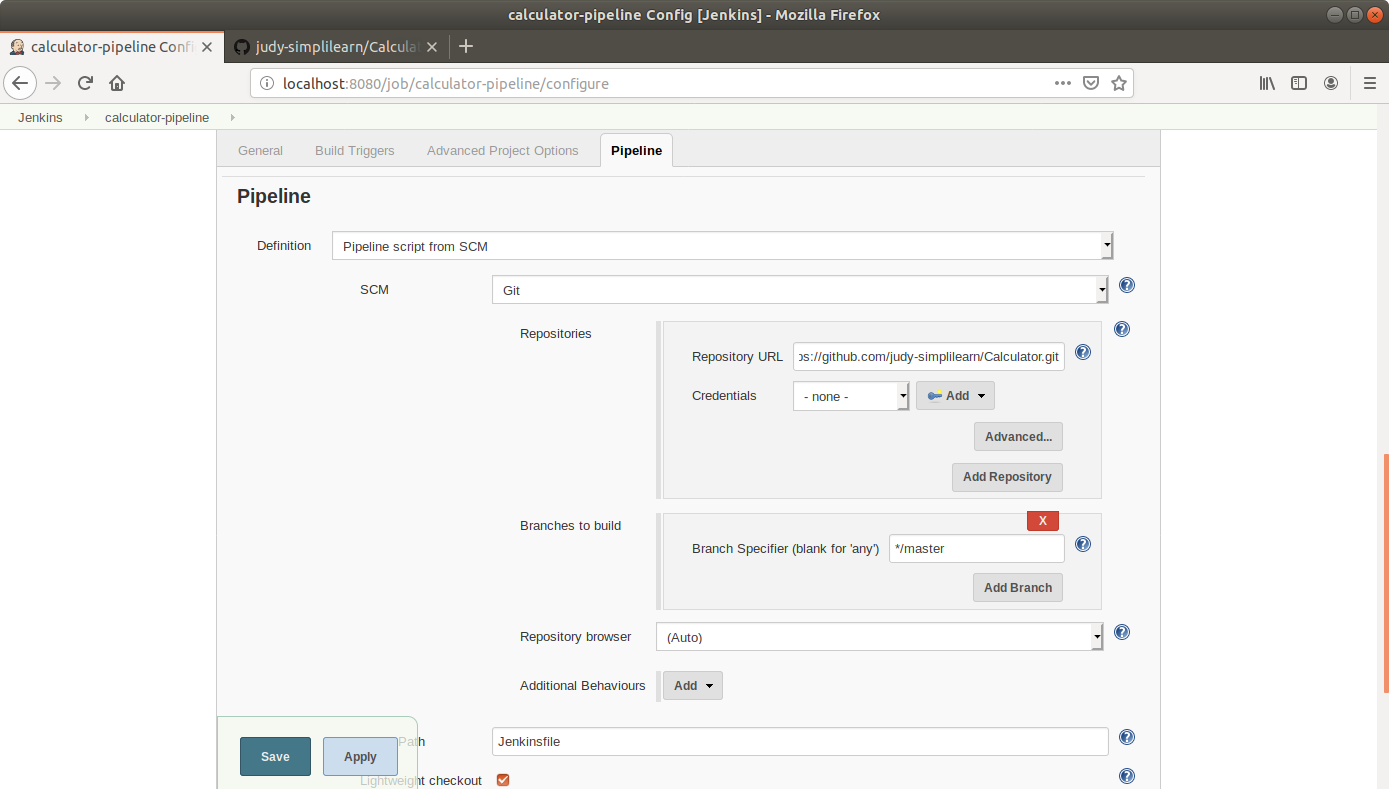
**}**

* Save the file as **Jenkinsfile** with no extension.
* Commit the changes to the remote SCM.
* Run **git add .**
* Run **git commit -m “Jenkinsfile”**
* Run **git push -u origin master**



**Step 3:** Creating a Pipeline

* Go to Jenkins **dashboard**.
* Click on ***New Item***.
* Enter a **name** for your build job.
* Select ***Pipeline***as the build job type.
* Click **OK**.
* On the configuration page, scroll down to the **Pipeline** section.
* Change *Definition* from *Pipeline script* to ***Pipeline script from SCM****.*
* Select *Git in* ***SCM****.*
* Add the repository URL.



* Click **Save**.
* Click on ***Build Now***in the project window.
* Jenkins will now build your pipeline and output the logs.

