2.2 Build and Configure CI/CD Pipeline with Maven Project

This section will guide you:

* To understand CI/CD Pipeline
* To set up a Pipeline
* To write Jenkins file
* To build a CI/CD Pipeline with Maven

**Development Environment:**

* Jenkins

This guide has three sub-sections, namely:

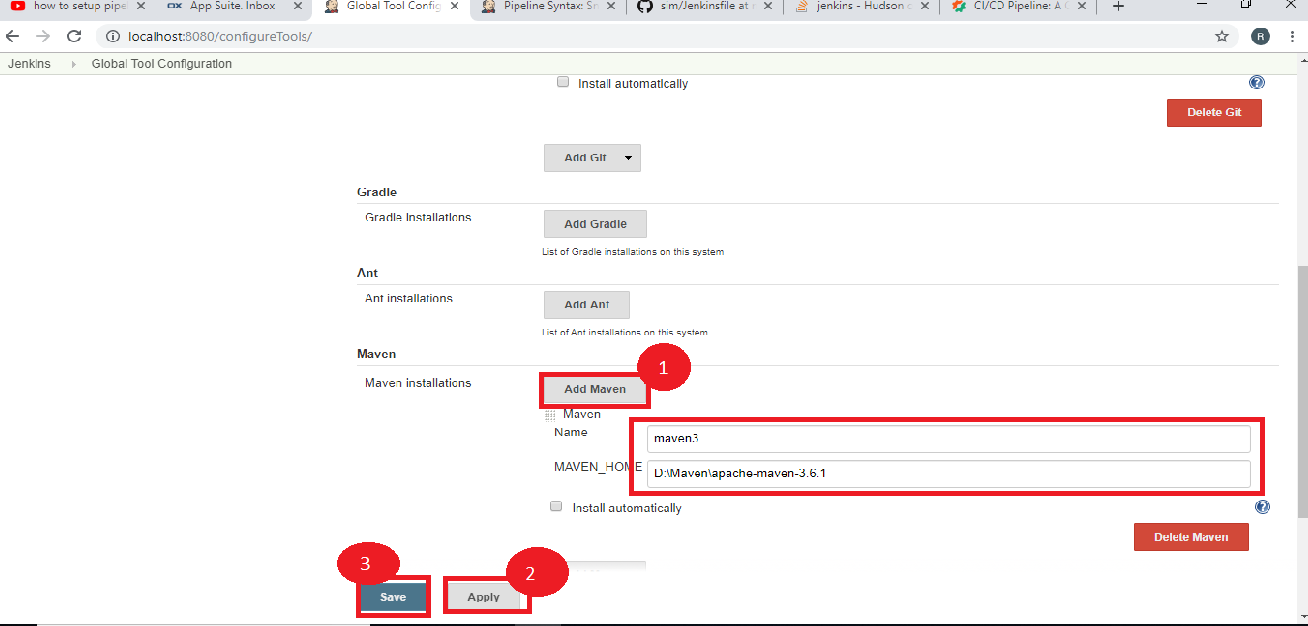
2.2.1 Setting up a Pipeline

2.2.2 Writing a Jenkins file

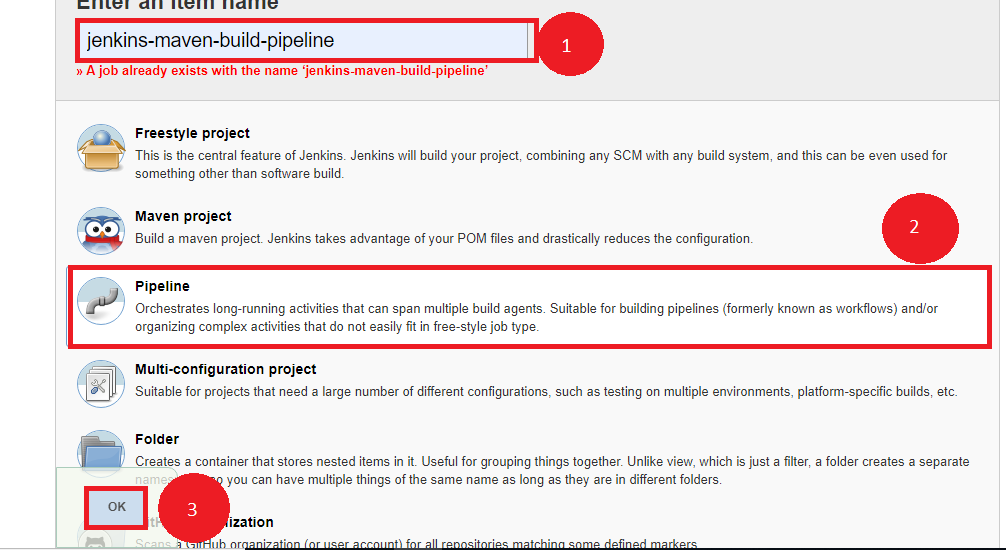
2.2.3 Pushing the code into GitHub repositories

**Step 2.2.1:** Setting up a Pipeline

* Jenkins is already set up in your practice lab. Refer to the lab guide for phase 2 for more information.
* Open Jenkins- locahhost:8080.
* Click on **Manage Jenkins.**
* Select **Global Tool Configuration.**
* Set the path of **Java and Maven.**
* Then, **Apply and Save.**



* Select a **New Item.**
* Name the job: **Jenkins-maven-build-pipeline.**
* Select the **Pipeline.**
* Now, click on **OK.**



**Step 2.2.2:** Writing a Jenkins file

* Create a job that needs to be configured.
* Click on **Pipeline.**
* Select the definition as- **Pipeline script from SCM.**
* Pass the GitHub URL.
* Now, we need to write the Jenkins file inside that git file.

node{

stage('SCM Checkout'){

git 'https://github.com/Autamation/sim.git'

}

stage('Compile-Package'){

def mvnHome = tool name: 'maven3', type: 'maven'

bat "${mvnHome}/bin/mvn package"

bat 'mvn package'

}

}

* Now, click on ‘Apply’ and ‘Save.’
* Click on **Build Now.**

**Step 2.2.3:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master