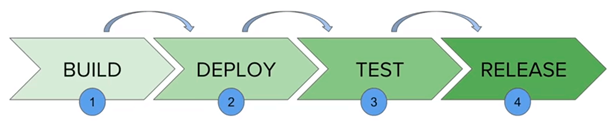
**What is a Jenkins Pipeline?**

Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.

A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

The definition of a Jenkins Pipeline is typically written into a text file (called a Jenkinsfile) which in turn is checked into a project’s source control repository.

## What is Continuous Delivery Pipeline?



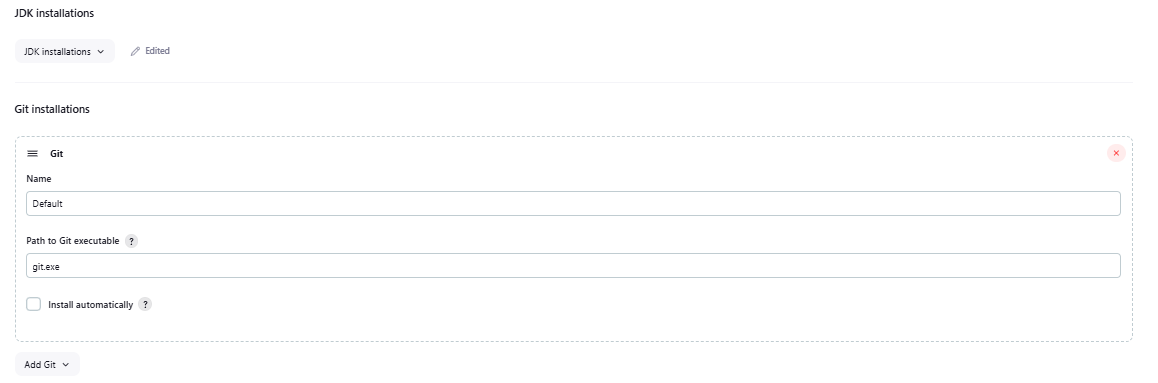
In a Jenkins Pipeline, every job has some sort of dependency on at least one or more jobs or events.

The above diagram represents a continuous delivery pipeline in Jenkins. It contains a collection of states such as build, deploy, test and release. These jobs or events are interlinked with each other. Every state has its jobs, which work in a sequence called a continuous delivery pipeline.

# Git Integration with Jenkins

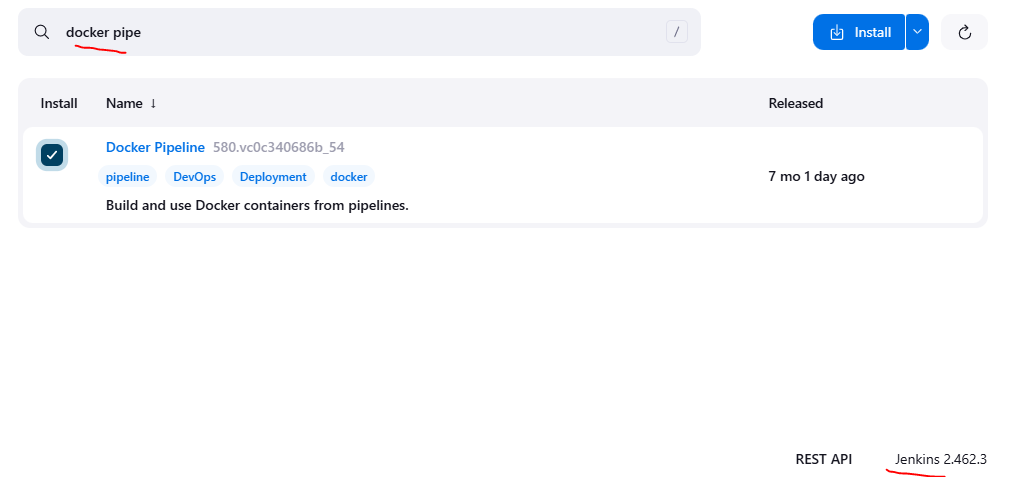
If below Git section is not visible then you need to add “Git Plugin”





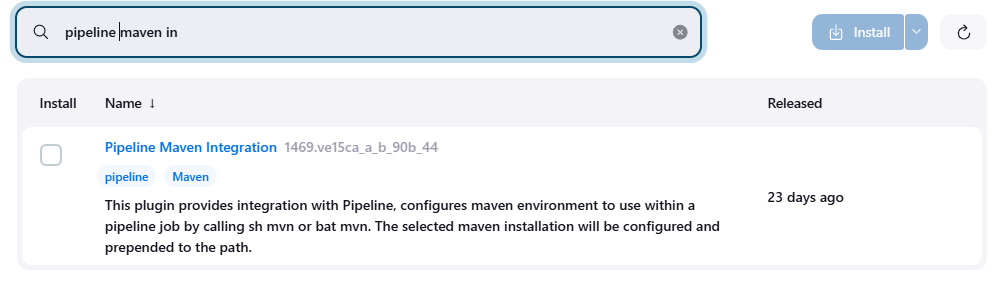
To get started quickly with Pipeline:

1. Install the **[Docker Pipeline plugin](https://plugins.jenkins.io/docker-workflow/)** through the **Manage Jenkins > Plugins** page
2. After installing the plugin, restart Jenkins so that the plugin is ready to use



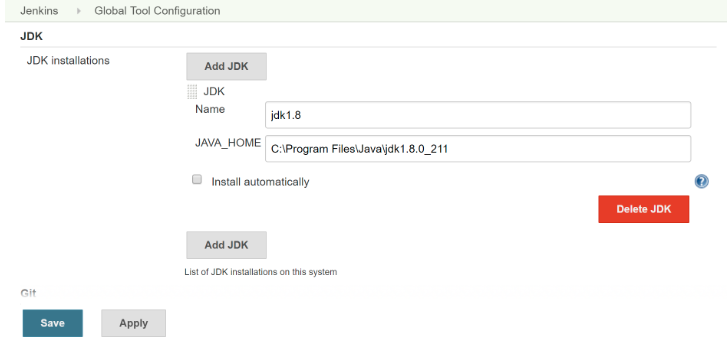
Maven project execution in Jenkins pipeline

<https://www.javatpoint.com/jenkins-maven-setup>



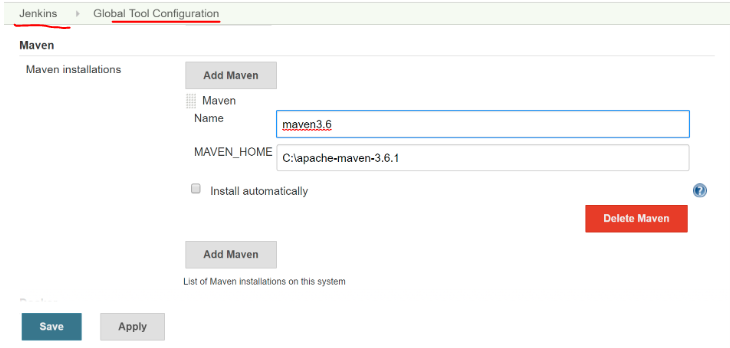
Add JDK path in Jenkins

Dashboard (Home screen) >> manage Jenkins>> "Global Tool Configuration" option.



Add Maven path

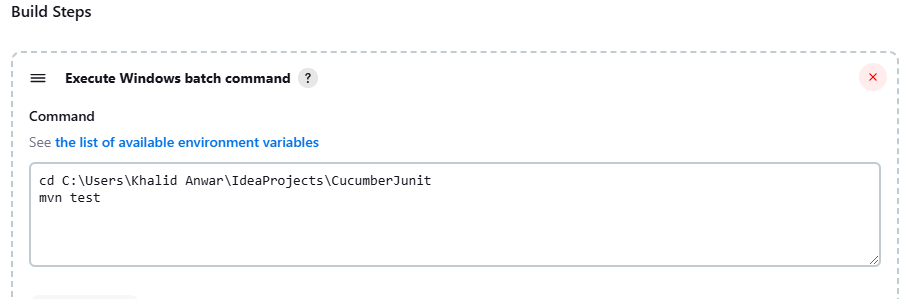
Dashboard (Home screen) >> manage Jenkins>> "Global Tool Configuration" option.



Executing Maven Project using Jenkins FreeStyle Project (**Windows batch project)**

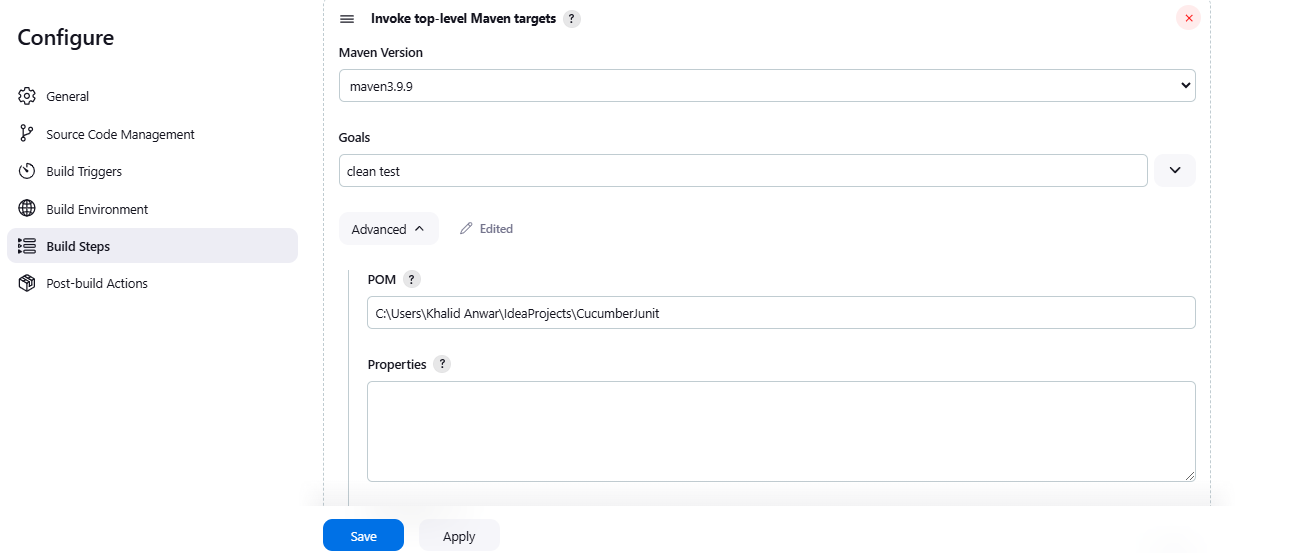
Step1: maven project is available in local system e.g C:\Users\Khalid Anwar\IdeaProjects\CucumberJunit

Step2: using windows batch command , you need to switch to project folder and execute mvn command



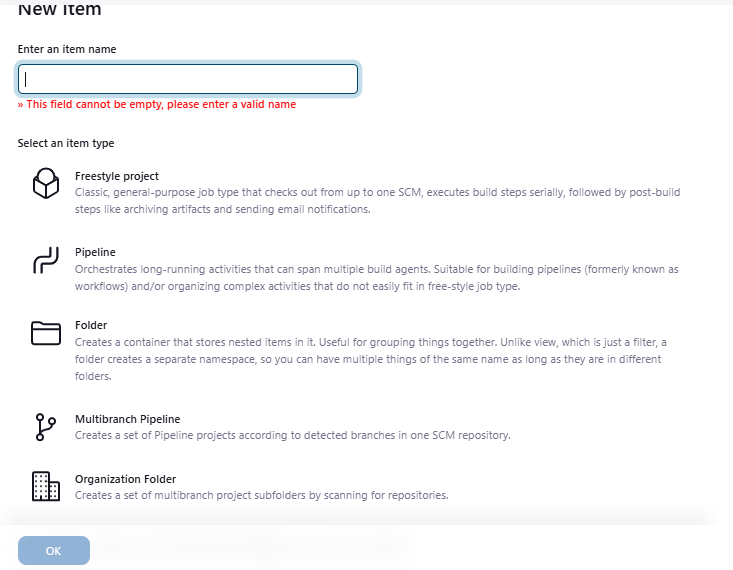
Executing Maven Project using Jenkins FreeStyle Project (**Invoke top level Maven target)**

**In below screen, in POM section, you need to provide project directory wherein pom.xml exist**

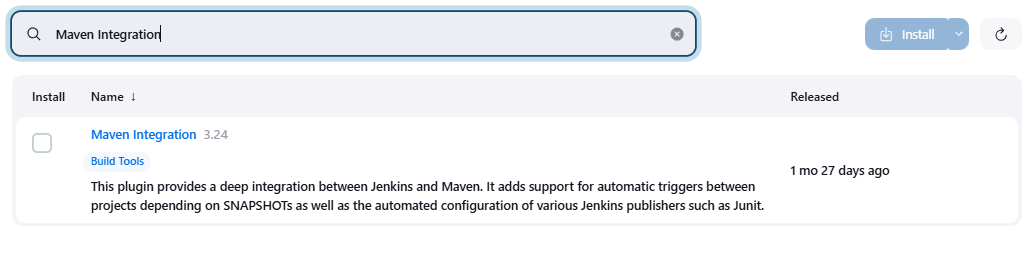


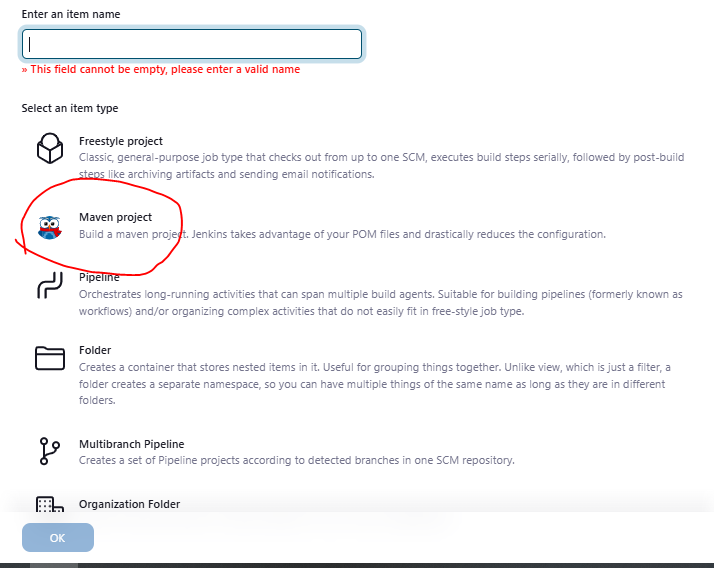
Executing Maven Project using Jenkins FreeStyle Project (**Using Maven Integration)**

**If this pluginn is not installed then new item in Jenkins will have below screen. We don’t find Maven Project**

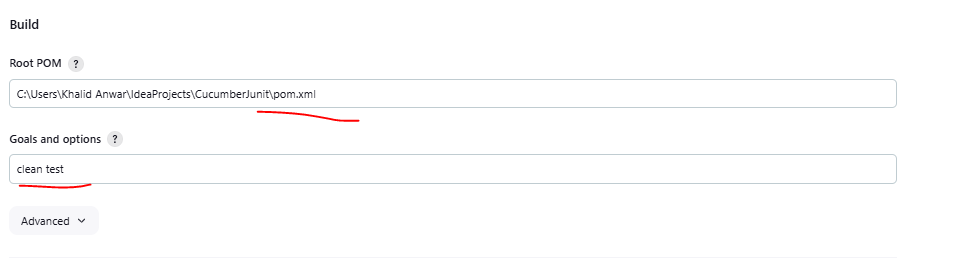


After installation of **Maven Integration plugins**

****



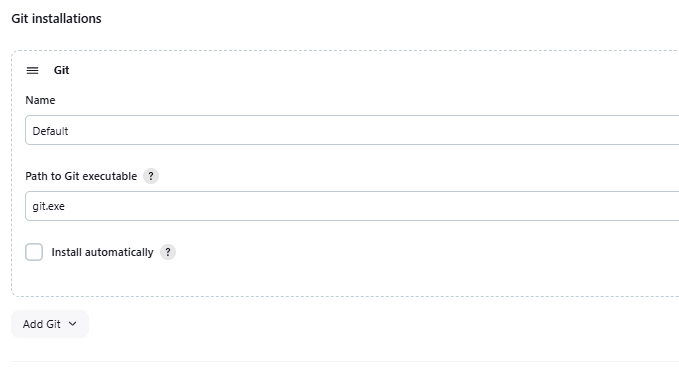
While creating job, select “Maven project” and add below configuration



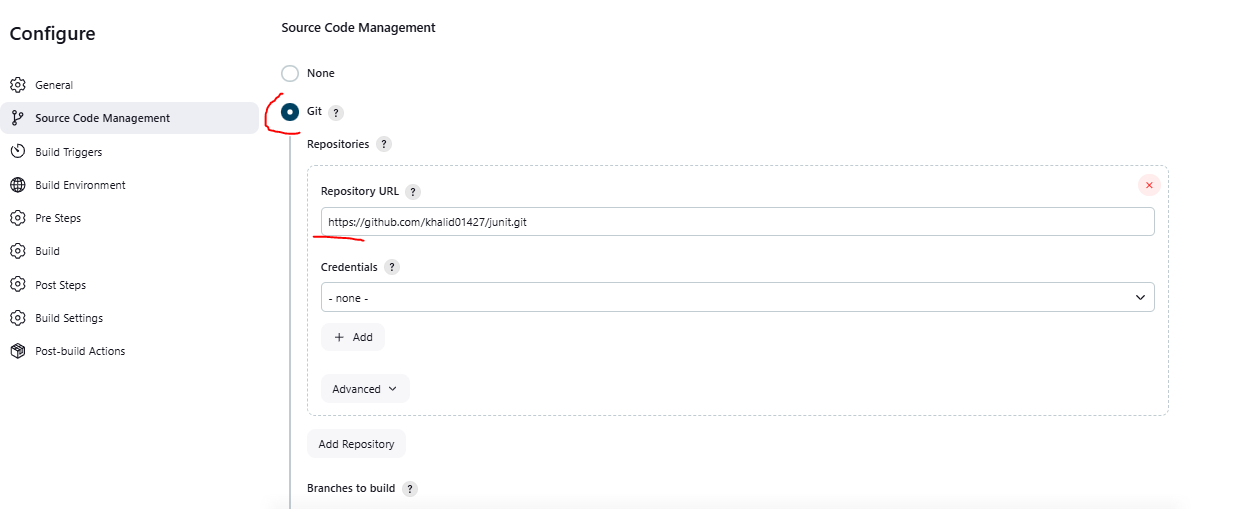
# Executing Maven Project from gitub using Jenkins Maven Project

**Step1: check git installation in below section in**

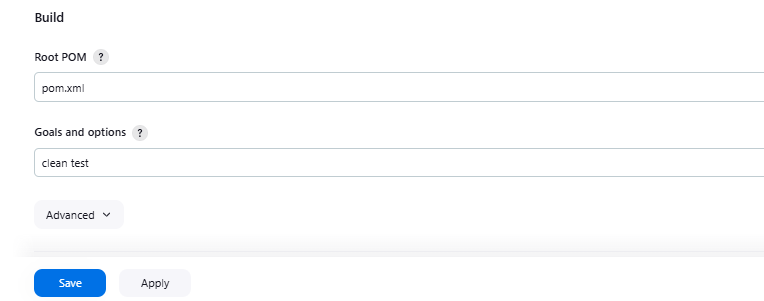
(Home screen) >> manage Jenkins>> "Global Tool Configuration" option

****

**Need to configure “source code management” as below**

****

**And in build section , configure as below**

****

# Executing Maven Project from gitub using Jenkins pipeline

## Pipeline script

<https://www.youtube.com/watch?v=AqITZLJ5eZ4>



**Pipeline script Content-**

pipeline {

agent any

stages {

// Stage1------------

stage('view') {

steps {

echo 'Hello World'

}

}

// Stage2---------

stage('Build') {

steps {

echo 'Building application'

}

}

// Stage3---------

stage('Test') {

steps {

echo 'Testing application'

}

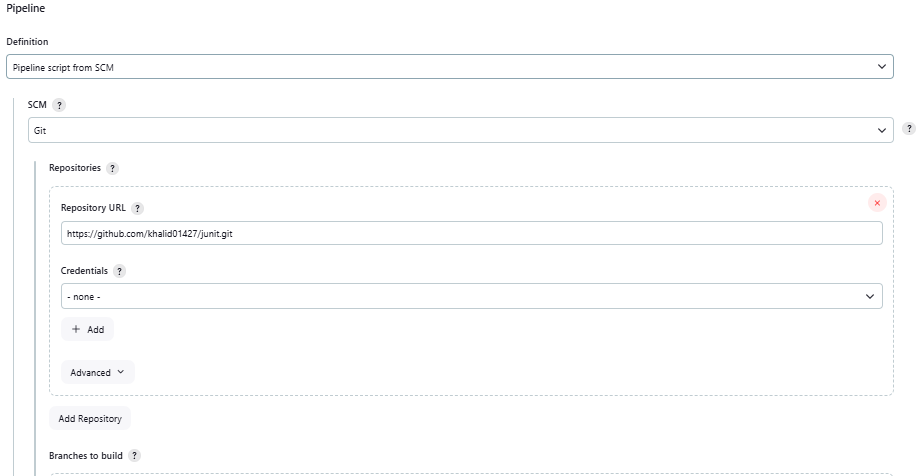
}

}

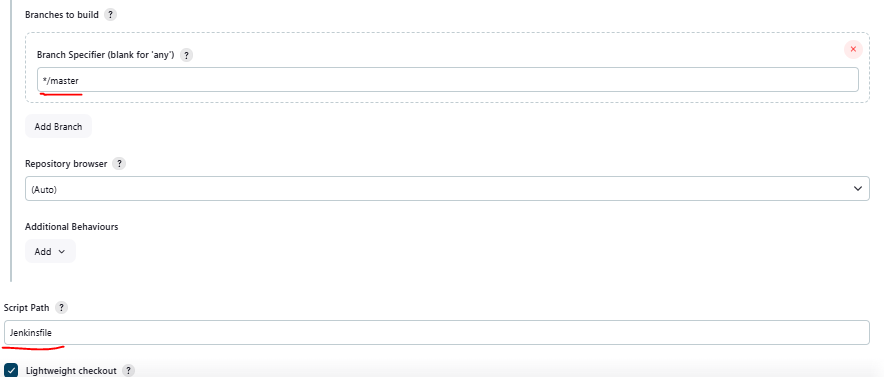
}

## Pipeline script from scm

Screen1--



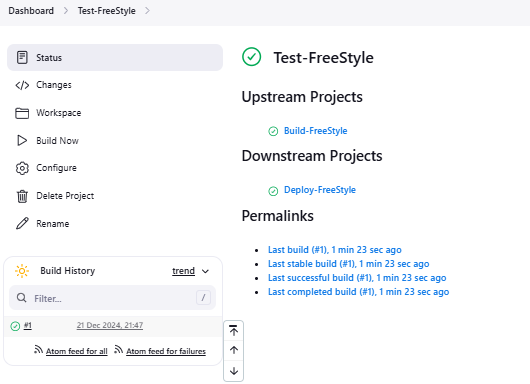
Screen2---



# Jenkins job-Upstream and downstream job

<https://www.youtube.com/watch?v=_1EgoteXRoc>

Job “Test-Freestyle have 1 upstream and 1 downstream job



Step1: you need to configure “post-build Actions”

Step2: Select “Build other projects” from list

Step3: enter other existing job which you want to run post after completion of current job

Step4: need to select radio button whether other job should run after successful current job

