

### Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

### Download progress








Preparation	<ul style="list-style-type: none"><li>Checking internet connectivity</li><li>Checking update center connectivity</li><li>Success</li></ul>
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Struts	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
bouncycastle API	Success
Credentials	Success
Plain Credentials	Success
Variant	Success
SSH Credentials	Success
Credentials Binding	Success
SCM API	Success
Pipeline: API	Success
commons-lang3 v3.x Jenkins API	Success
Timestampers	Success
Caffeine API	Success
Script Security	Success
JavaBeans Activation Framework (JAF) API	Success
JAXB	Success

### New Item

Enter an item name

GitMavenPipeline

Select an item type

-  **Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
-  **Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
-  **Pipeline**  
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.
-  **Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
-  **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
-  **Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.
-  **Organization Folder**

OK

## Configure

General

Triggers

Pipeline

Advanced

☐ Trigger builds remotely (e.g., from scripts) ?

## Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

## Definition

Pipeline script

```
1= node {
2  // 1) Checkout from GitHub
3  stage('Checkout') {
4    git url: 'https://github.com/noelmathen/IBM-Consulting.git', branch: 'main'
5  }
6
7  // 2) Build with Maven
8  stage('Build') {
9    // This wrapper will use the Maven installation named "H3"
10   withMaven(maven: 'H3') {
11     // On Linux/Mac:
12     sh 'mvn clean install'
13     // On Windows agents, use:
14     // bat 'mvn clean install'
15   }
16 }
17 }
18 }
```

☒ Use Groovy Sandbox ?

Pipeline Syntax

## Advanced

Advanced

Save

Apply



Status

Changes

Console Output

Edit Build Information

Delete build #4

Timings

Git Build Data

See Fingerprints

Maven

Pipeline Overview

Replay

Pipeline Steps

Workspaces

Previous Build

## Console Output

Download Copy View as plain text

```
started by user Noel Mathen Bidho
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\Users\noebl\jenkins\workspace\gitMavenPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
The recommended git tool is: NONE
no credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\Users\noebl\jenkins\workspace\gitMavenPipeline\.git # timeout=10
Fetching changes from the remote git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/noelmathen/IBM-Consulting.git # timeout=10
Fetching upstream changes from https://github.com/noelmathen/IBM-Consulting.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # 'git version 2.42.0.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/noelmathen/IBM-Consulting.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse --refs/remotes/origin/main[:%s] # timeout=10
Checking out Revision 6883362a18a7135e46f23bc08128c128c2 (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f 6883362a18a7135e46f23bc08128c128c2 # timeout=10
> C:\Program Files\Git\bin\git.exe branch -f -v --no-abbrev # timeout=10
> C:\Program Files\Git\bin\git.exe branch -D main # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -b main 6883362a18a7135e46f23bc08128c128c2 # timeout=10
Commit message: "added pom.xml"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk e65d758c3173e4d6f98812054e4e9d563189ad # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] withMaven
$ C:\Users\noebl\jenkins\tools\hudson.tasks.Maven\mavenInstallation\H3\bin\mvn.cmd --version
[Pipeline] bat
C:\Users\noebl\jenkins\workspace\gitMavenPipeline>mvn clean install
Picked up JAVA_TOOL_OPTIONS: -Dmaven.ext.class.path=C:\Users\noebl\jenkins\workspace\gitMavenPipeline\mp\withMaven\maven-spy.jar
org.jenkinsci.plugins.pipeline.maven.reportsFolder=C:\Users\noebl\jenkins\workspace\gitMavenPipeline\mp\withMaven\maven-spy
[INFO] [jenkins-event-spy] Generate C:\Users\noebl\jenkins\workspace\gitMavenPipeline\mp\withMaven\maven-spy-20250618-205448-41762483752362362270.log.tmp ...
[INFO] Scanning for projects...
[INFO]
[INFO] ----- com.demo:demo-app >-----
[INFO] Building demo-app 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ 3M ]-----
```