Maven-Jenkins-Github_Integration
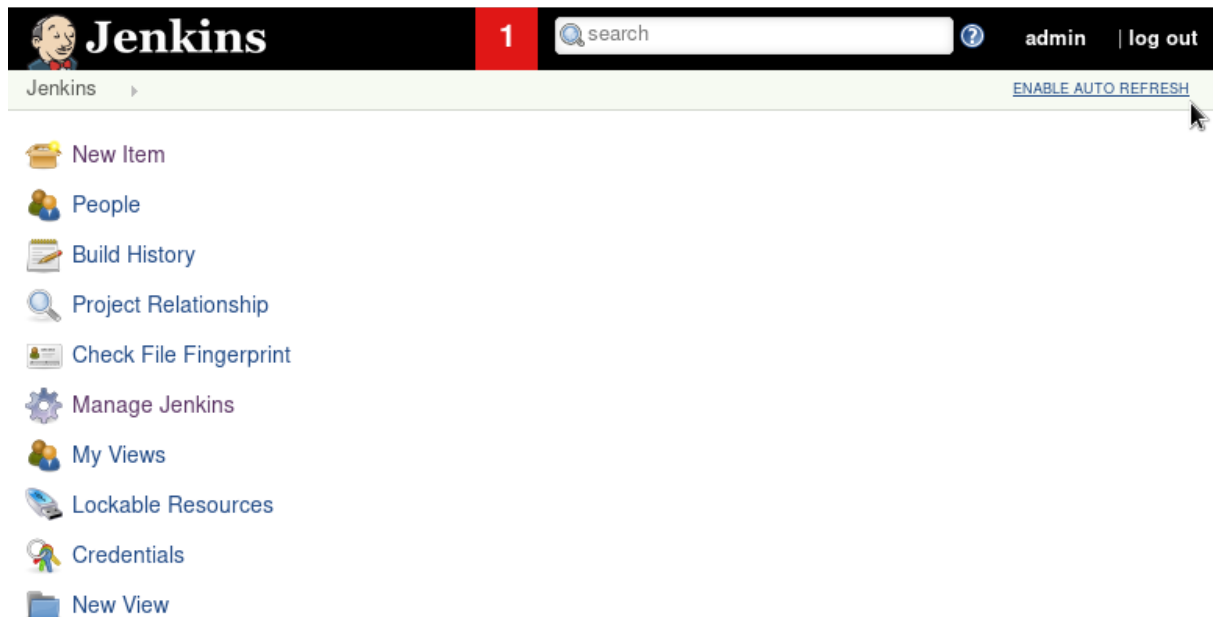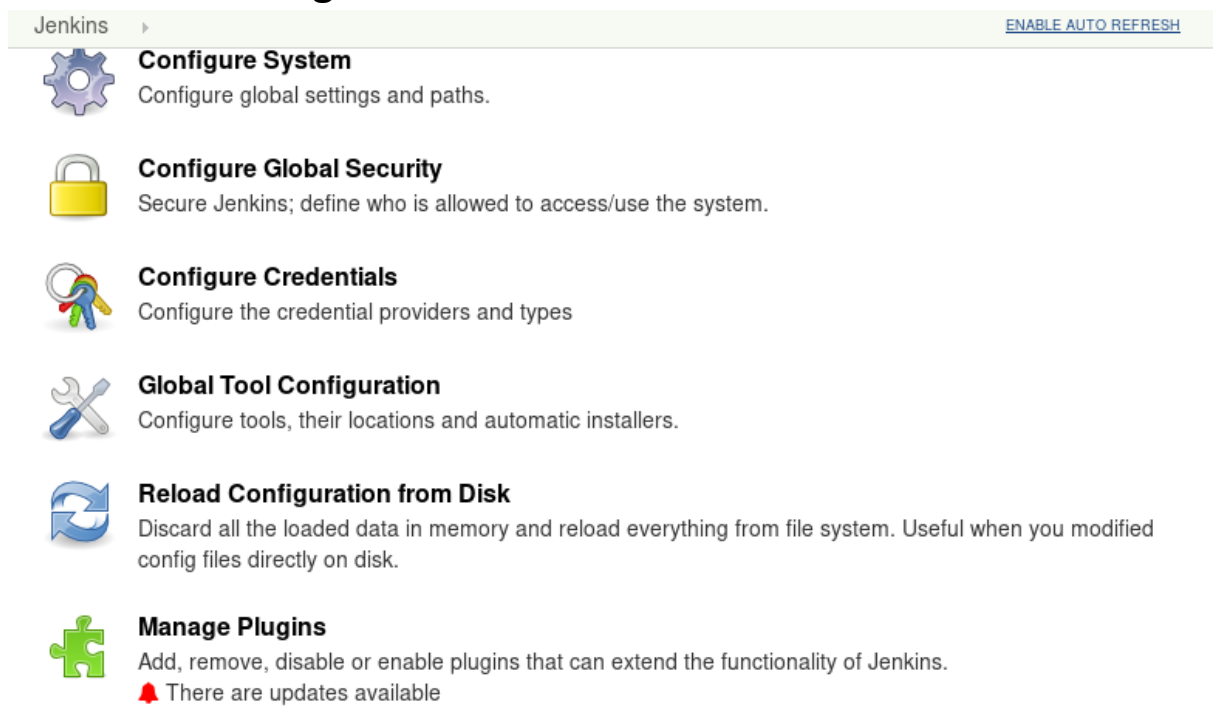
1. Login to your Jenkins.

2. Go inside Manage Jenkins.

3. Click on Manage Plugins.

4. Download Maven-Integration plugin.

"Extensible Choice Plugin" which is able to read Artifact information from a Nexus repository.

Maven Dependency Update Trigger
This plugin will check if any SNAPSHOT dependencies (or optionally plugins SNAPSHOT) have been updated during your project's dependencies resolution and trigger a build.   1.5   Uninstall

Maven Integration plugin
This plug in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTs, automated configuration of various Jenkins publishers (Junit, ...).   3.4   Uninstall

Maven Metadata Plugin for Jenkins CI server
Adds a build parameter that presents versions of an artifact from a maven repository as a drop down list.   2.0.0   Uninstall

Maven SNAPSHOT Check Plugin
This plugin is used to check if pom.xml contains SNAPSHOT.   1.5   Uninstall

Oracle Java SE Development Kit Installer Plugin
Allows the Oracle Java SE Development Kit (JDK) to   1.4   Uninstall

5. Now go to Configure global tool.

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
🔺 There are updates available

6. Scroll down to maven and click on add maven. Write the name and Path of your maven. In my case /usr/share/maven.

7. Click on apply and save. Now create a new job and select maven project.

8. Click on git-hub project and add the url of your git-hub
   repository where your pom.xml and Jenkins file are
   stored.
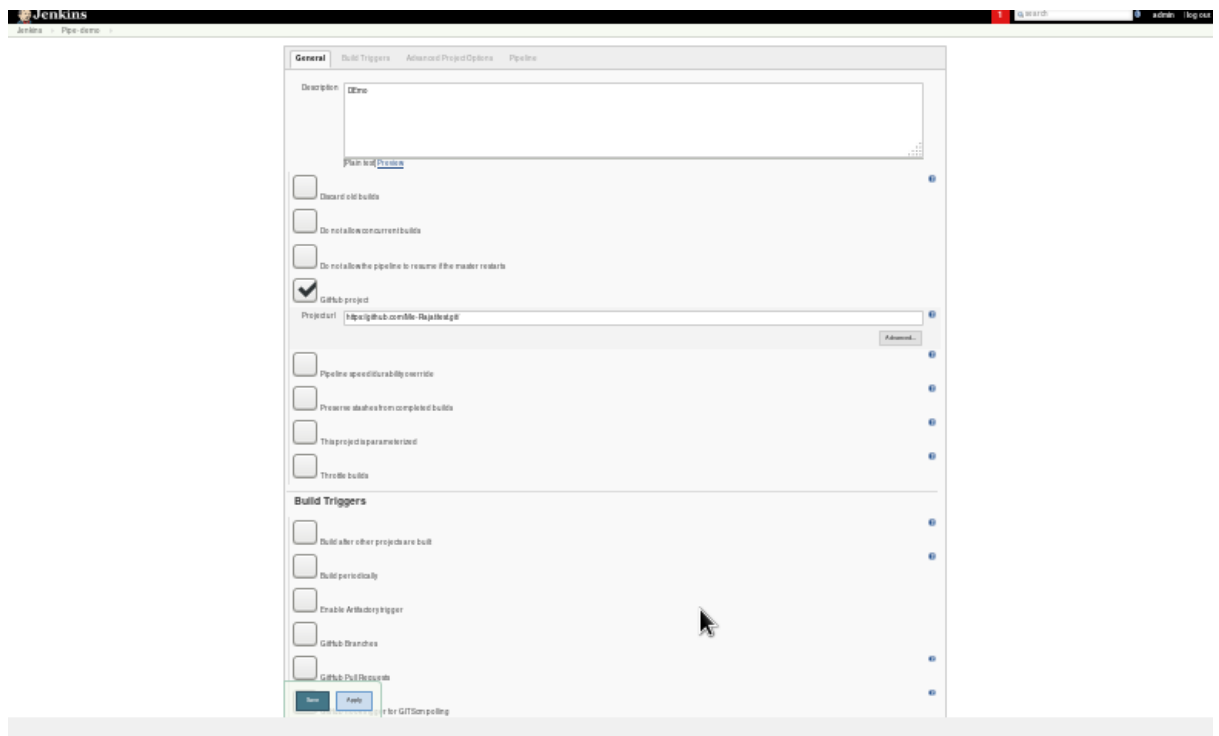
9. Make sure in your Jenkinsfile doc you must include the
   maven build and test steps.
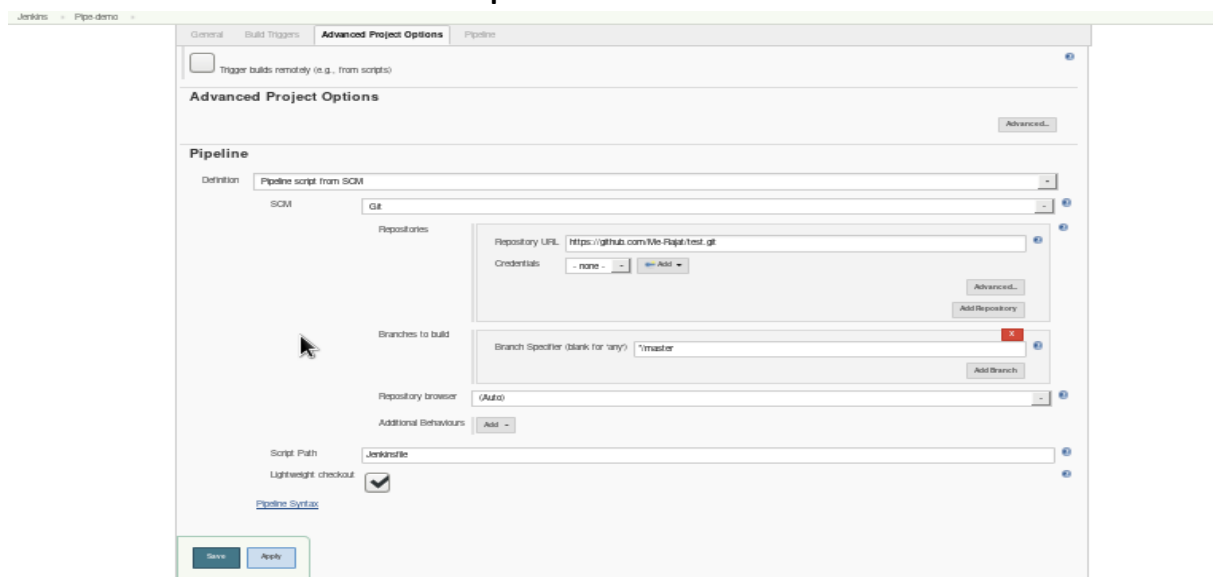
10.   Click on save and run the build.