Jenkins supports three principal build tools: Ant, Maven, and the basic shell-script (or Batch script in Windows). Using Jenkins plugins, you can also add support for other build tools and other languages, such as Gant, Grails, MSBuild, and many more.

Apache ANT

Apache Ant is a Java library and command-line tool that help building software.

Ant is distributed as zip, tar.gz and tar.bz2 archives - the contents are the same.

Jenkins provides excellent build-in support for Ant—you can invoke Ant targets from your build job, providing properties to customize the process as required.

If Ant is available on the system path, Jenkins will find it. However, if you want to know precisely what version of Ant you are using, or if you need to be able to use several different versions of Ant on different build jobs, you can configure as many installations of Ant as required. Just provide a name and installation directory for each version of Ant in the Ant section of the Configure System screen. You will then be able to choose what version of Ant you want to use for each project.

Maven

Maven is a high-level build scripting framework for Java that uses notions such as a standard directory structure and standard life cycles, Convention over Configuration, and Declarative Dependency Management to simplify a lot of the low-level scripting that you find in a typical Ant build script.

Jenkins provides excellent support for Maven, and has a good understanding of Maven project structures and dependencies. You can either get Jenkins to install a specific version of Maven automatically, or provide a path to a local Maven installation. You can configure as many versions of Maven for your build projects as you want, and use different versions of Maven for different projects.

GitHub

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

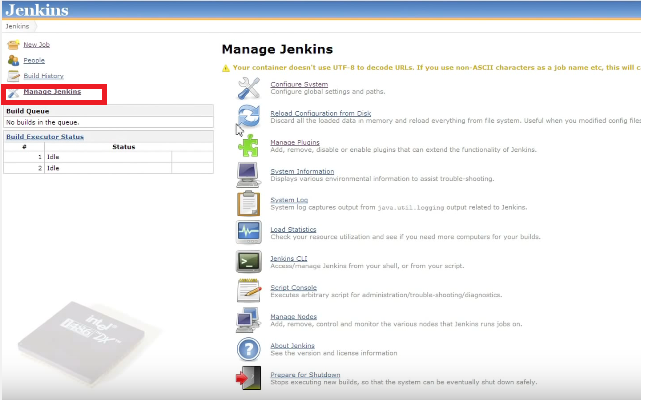
GitHub Plugin integrates Jenkins with Github projects.

The plugin currently has three major functionalities:

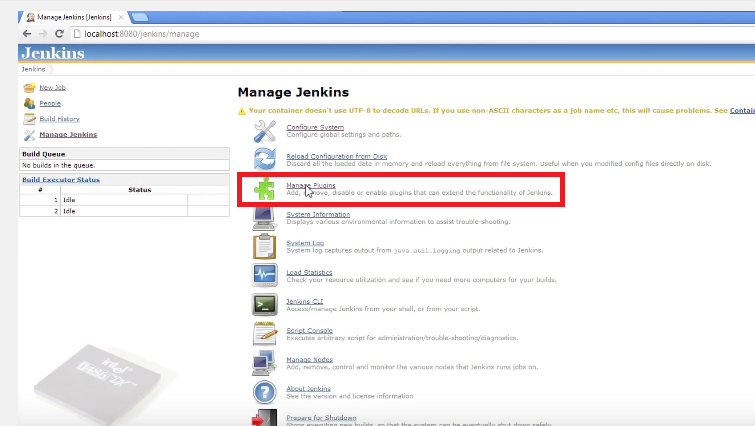
* Create hyperlinks between your Jenkins projects and GitHub
* Trigger a job when you push to the repository by groking HTTP POSTs from post-receive hook and optionally auto-managing the hook setup.
* Report build status result back to github as Commit Status

How to install Plugin GitHub on Jenkins

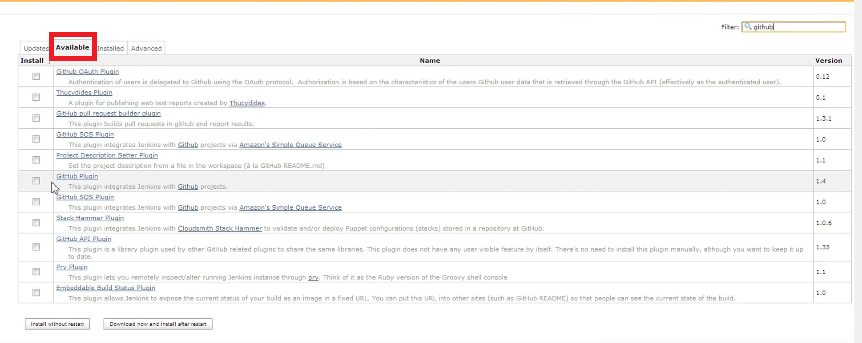
Click manage Jenkins



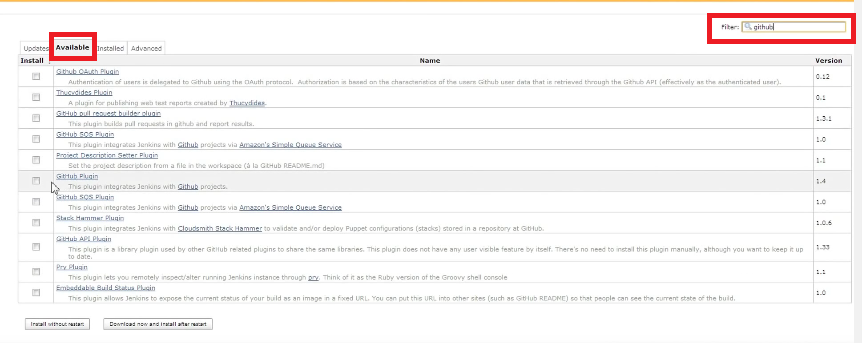
Click Manage Plugins



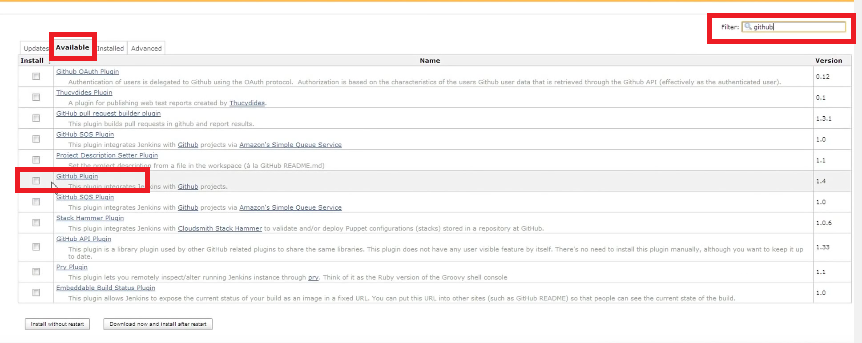
Click the Available Tab



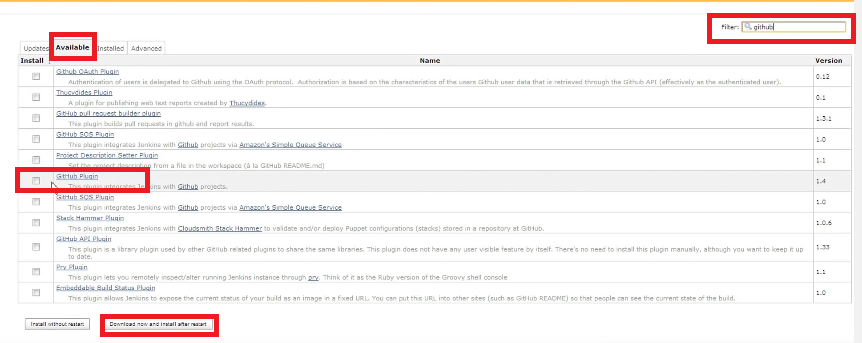
Enter GitHub in the Filter Field



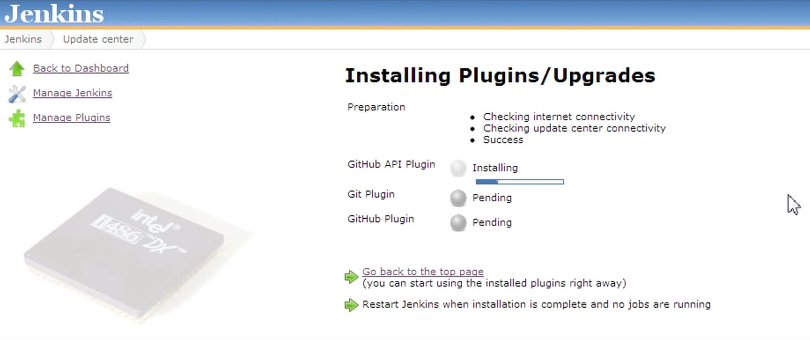
Check the GitHub Plugin for installation



Click the Download Now and install after restart Button

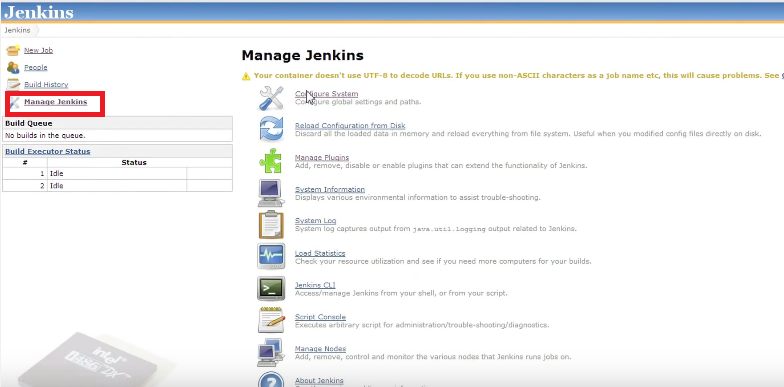


Wait until all plugins are downloaded and after that Restart the Server

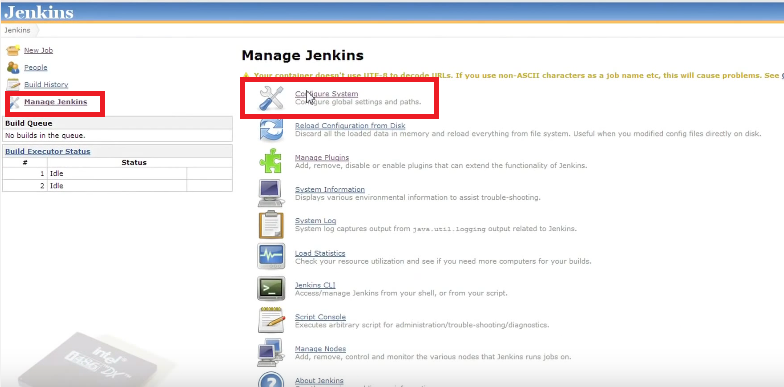


How to configure Jenkins to use Maven, Ant and GitHub

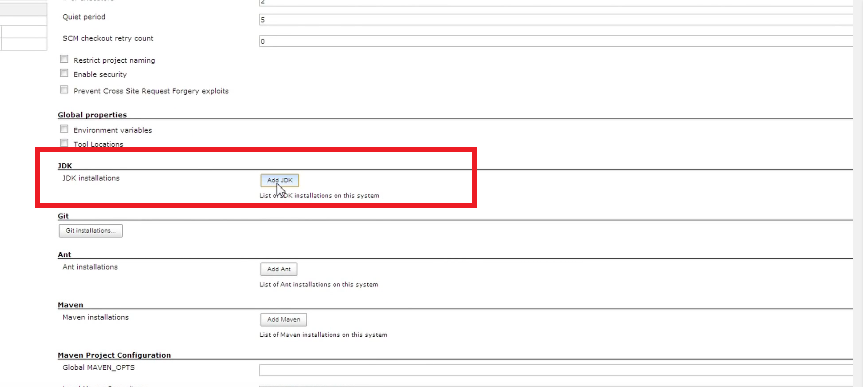
Click Manage Jenkins

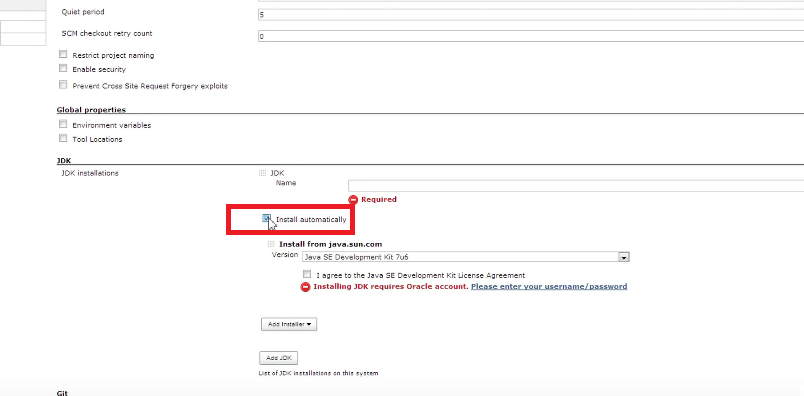


Click Configure System/Global tool configuration



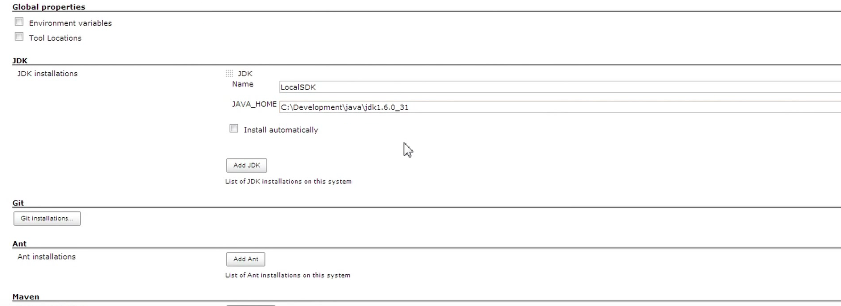
In the JDK Section click Add JDK Button

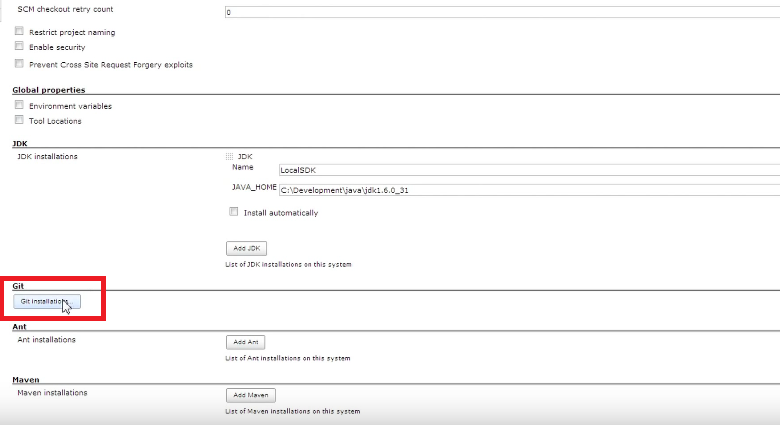


Uncheck the install automatically option 

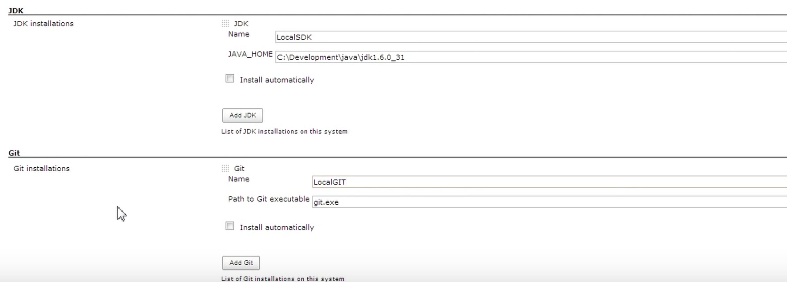
Enter a name for the JDK. i.e LocalJDK

Enter the JAVA\_HOME variable which is pointing to the JDK directory



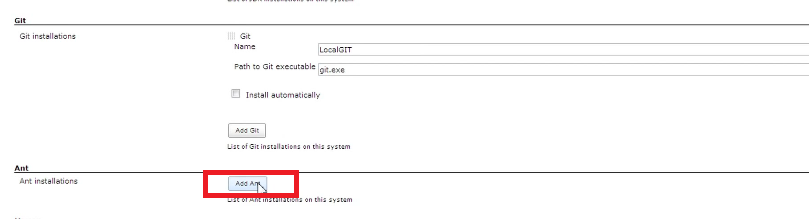
Next in the GIT section click in the Git Installations Button 

Leave Path to Git executable with the value: git.exe

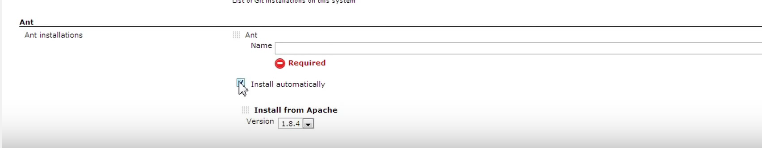


Next in the ANT section

Click add ANT



Uncheck the Install Automatically option

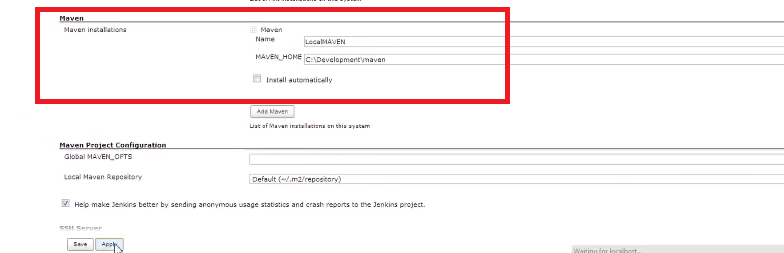


Enter the Ant Name Value. i.e LocalAnt

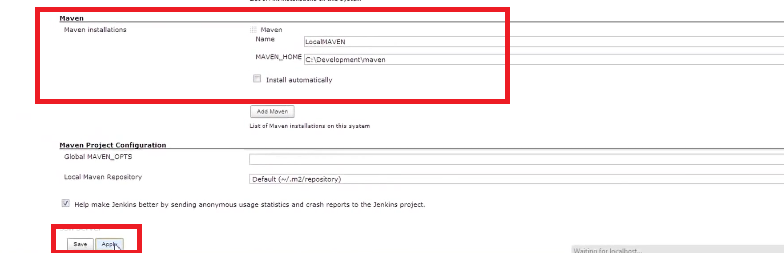


Enter the Path to your ANT installation in the ANT\_HOME field

In The MAVEN section clic Add Maven and configure the same things



Uncheck the install Automatically option and enter the MAVEN name and path to your Local Maven installation



Then click Apply and then click Save.