

Siva Jenkins Tutorial

What Is Jenkins?

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks such as building, testing, and deploying software. Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with the Java Runtime Environment installed.

(Or)

Jenkins is a powerful application that allows continuous integration and continuous delivery of projects, regardless of the platform you are working on. It is a free source that can handle any kind of build or continuous integration. You can integrate Jenkins with a number of testing and deployment technologies. In this document, we would explain how you can use Jenkins to build and test your software projects continuously.

Important of Jenkins

This Document is going to help all those software testers who would like to learn how to build and test their projects continuously in order to help the developers to integrate the changes to the project as quickly as possible and obtain fresh builds.

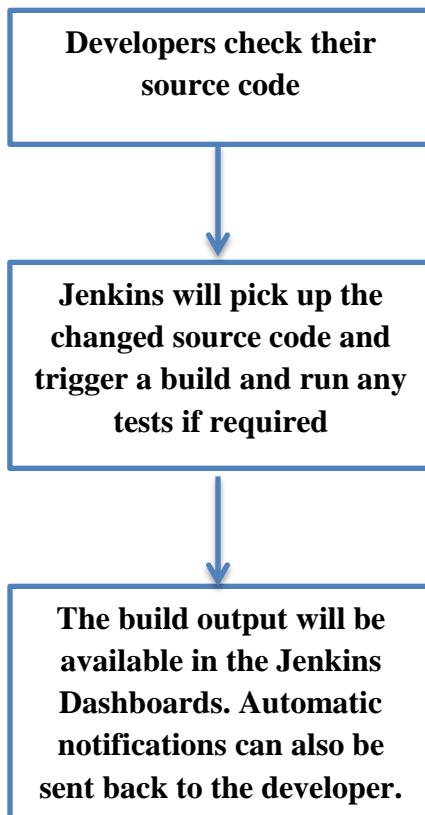
Prerequisites

Jenkins is a popular tool for performing continuous integration of software projects. This is a preliminary documentation that covers the most fundamental concepts of Jenkins. Any software professional having a good understanding of Software Development Life Cycle should benefit from this document

Siva Jenkins Tutorial

Why Jenkins?

Jenkins is software that allows **continuous integration**. Jenkins will be installed on a server where the central build will take place. The following flowchart demonstrates a very simple workflow of how Jenkins works.



Along with Jenkins, sometimes, one might also see the association of **Hudson**. Hudson is a very popular open-source Java-based continuous integration tool developed by Sun Microsystems which was later acquired by Oracle. After the acquisition of Sun by Oracle, a fork was created from the Hudson source code, which brought about the introduction of Jenkins.

What is Continuous Integration (CI)?

Continuous Integration is a development practice that requires developers to integrate code into a shared repository at regular intervals. This concept was meant to remove the problem of finding later occurrence of issues in the build lifecycle. Continuous integration requires the developers to have frequent builds. The common practice is that whenever a code commit occurs, a build should be triggered.

Siva Jenkins Tutorial

System Requirements

JDK	JDK 1.5 or above
Memory	2 GB RAM (recommended)
Disk Space	No minimum requirement. Note that since all builds will be stored on the Jenkins machines, it has to be ensured that sufficient disk space is available for build storage.
Operating System Version	Jenkins can be installed on Windows, Ubuntu/Debian, Red Hat/Fedora/CentOS, Mac OS X, openSUSE, FReeBSD, OpenBSD, and Gentoo.
Java Container	The WAR file can be run in any container that supports Servlet 2.4/JSP 2.0 or later.(An example is Tomcat 5).

Download Jenkins

The official website for Jenkins is [Jenkins](#). If you click the given link, you can get the home page of the Jenkins official website as shown below.

Siva Jenkins Tutorial



By default, the latest release and the Long-Term support release will be available for download. The past releases are also available for download. Click the Long-Term Support Release tab in the download section.

Siva Jenkins Tutorial



Click the link “Older but stable version” to download the Jenkins war file.

Starting Jenkins

Open the command prompt. From the command prompt, browse to the directory where the Jenkins. War file is present. Run the following command

```
D:\>Java -jar Jenkins.War
```

After the command is run, various tasks will run, one of which is the extraction of the war file which is done by an embedded webserver called winstone.

```
D:\>Java -jar Jenkins.war
Running from: D:\jenkins.war
Web root: $user.home/.jenkins
Sep 29, 2015 4:10:46 PM winstone.Logger log Internal
```

Siva Jenkins Tutorial

INFO: Beginning extraction from war file

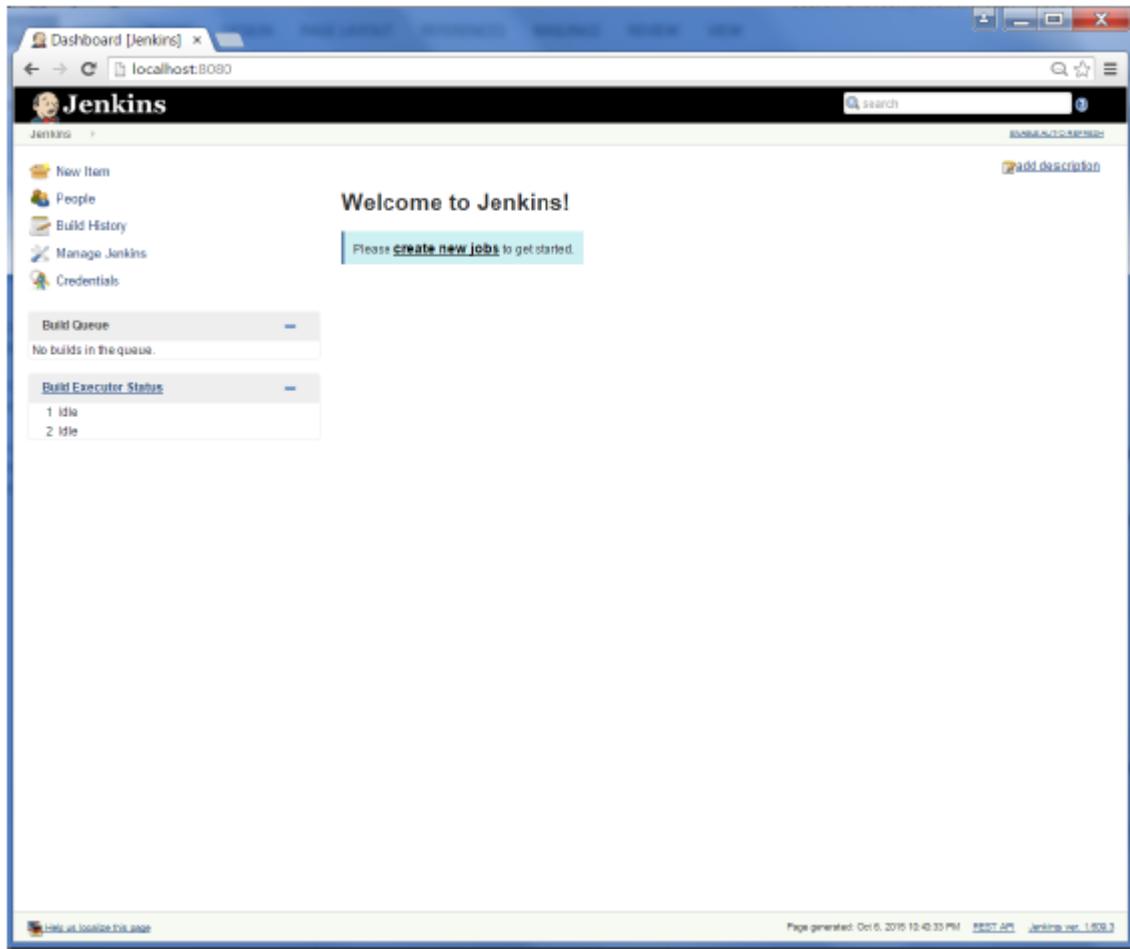
Once the processing is complete without major errors, the following line will come in the output of the command prompt.

INFO: Jenkins is fully up and running

Accessing Jenkins

Once Jenkins is up and running, one can access Jenkins from the link – **http://localhost:8080**

This link will bring up the Jenkins dashboard.



Siva Jenkins Tutorial

Jenkins – Tomcat Setup

The following prerequisites must be met for Jenkins Tomcat setup.

Step 1: Verifying Java Installation

To verify Java installation, open the console and execute the following java command.

OS	Task	Command
Windows	Open command console	\>java -version
Linux	Open command terminal	\$java -version

If Java has been installed properly on your system, then you should get one of the following outputs, depending on the platform you are working on.

OS	Output
Windows	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java Hotspot (TM) 64-bit Server VM (build 24.60-b09, mixed mode)
Linux	java version "1.7.0_25" Open JDK Runtime Environment (rhel-2.3.10.4.el6_4-x86_64) Open JDK 64-Bit Server VM (build 23.7-b01, mixed mode)

Siva Jenkins Tutorial

We assume the readers of this tutorial have Java 1.7.0_60 installed on their system before proceeding for this tutorial.

In case you do not have Java JDK, you can download it from the link [Oracle](#)

Step 2: Verifying Java Installation

Set the JAVA_HOME environment variable to point to the base directory location where Java is installed on your machine. For example,

OS	Output
Windows	Set Environmental variable JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	export JAVA_HOME=/usr/local/java-current

Append the full path of the Java compiler location to the System Path.

OS	Output
Windows	Append the String; C:\Program Files\Java\jdk1.7.0_60\bin to the end of the system variable PATH.
Linux	export PATH=\$PATH:\$JAVA_HOME/bin/

Verify the command java-version from command prompt as explained above.

Step 3: Download Tomcat

The official website for tomcat is [Tomcat](#). If you click the given link, you can get the home page of the tomcat official website as shown below.

Siva Jenkins Tutorial



Browse to the link <https://tomcat.apache.org/download-70.cgi> to get the download for tomcat.

Siva Jenkins Tutorial

The screenshot shows a Windows operating system desktop with a web browser window open to the Apache Tomcat download page at <https://tomcat.apache.org/download-70.cgi>. The browser title bar says "Apache Tomcat - Ap...". The page features a yellow cat logo on the left and a red feather logo on the right. A search bar at the top right has the placeholder "Search the Site..." and a "Search" button. The main content area is titled "Tomcat 7 Downloads" and includes sections for "Quick Navigation" (links to KEYS, 7.0.64, Browse, Archives), "Release Integrity" (instructions about OpenPGP signatures), "Mirrors" (current mirror set to <http://www.us.apache.org/dist/>), and "7.0.64" (link to the README file). The "Binary Distributions" section lists various file types for download, including "Core" (zip, tar.gz, 32-bit Windows zip, 64-bit Windows zip, 64-bit Itenium Windows zip, 32-bit/64-bit Windows Service Installer) and "Full documentation". On the left sidebar, there are links for "Apache Tomcat" (Home, Taglibs, Maven Plugin), "Download" (Which version?, Tomcat 8.0, 7.0, 6.0, Tomcat Connectors, Tomcat Native, Taglibs, Archives), "Documentation" (Tomcat 8.0, 7.0, 6.0, Tomcat Connectors, Tomcat Native, Wiki, Migration Guide), "Problems?" (Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), and "Get Involved" (Overview, SVN Repositories, Buildbot, Reviewboard, Tools).

Go to the 'Binary Distributions' section. Download the 32-bit Windows zip file.

Then unzip the contents of the downloaded zip file.

Step 4: Jenkins and Tomcat Setup

Copy the Jenkis.war file which was downloaded from the previous section and copy it to the webapps folder in the tomcat folder.

Now open the command prompt. From the command prompt, browse to the directory where the tomcat7 folder is location. Browse to the bin directory in this folder and run the start.bat file

```
E:\Apps\tomcat7\bin>startup.bat
```

Once the processing is complete without major errors, the following line will come in the output of the command prompt.

Siva Jenkins Tutorial

INFO: Server startup in 1302 ms

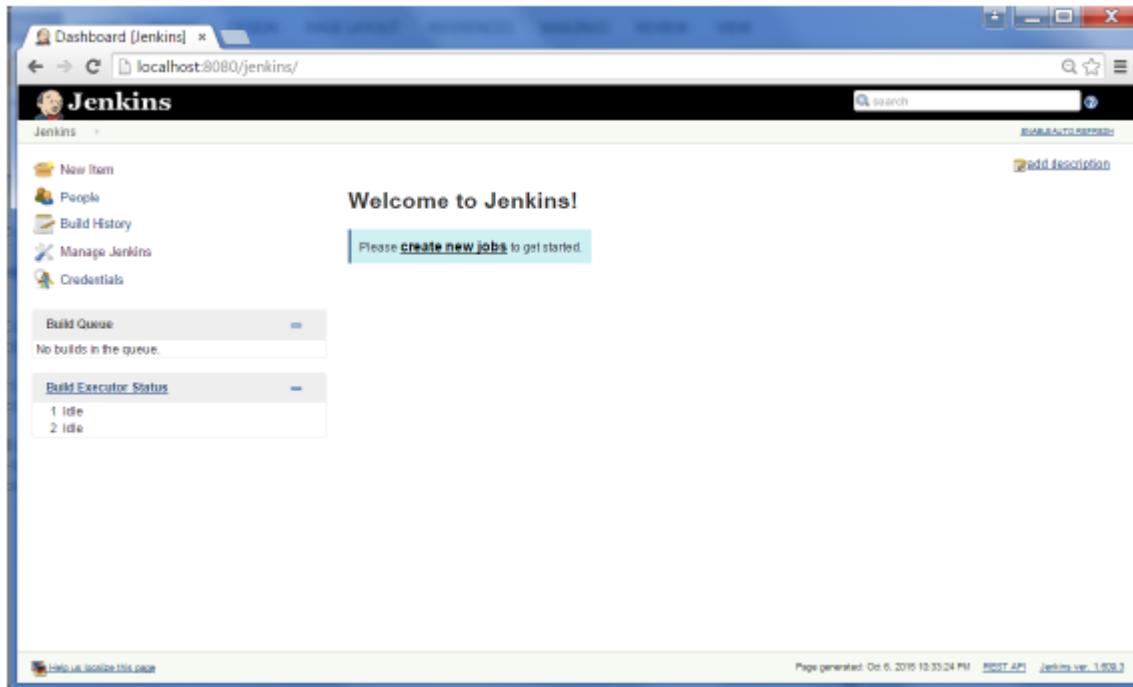
Open the browser and go to the link – <http://localhost:8080/jenkins>. Jenkins will be up and running on tomcat.



Jenkins - Git Setup

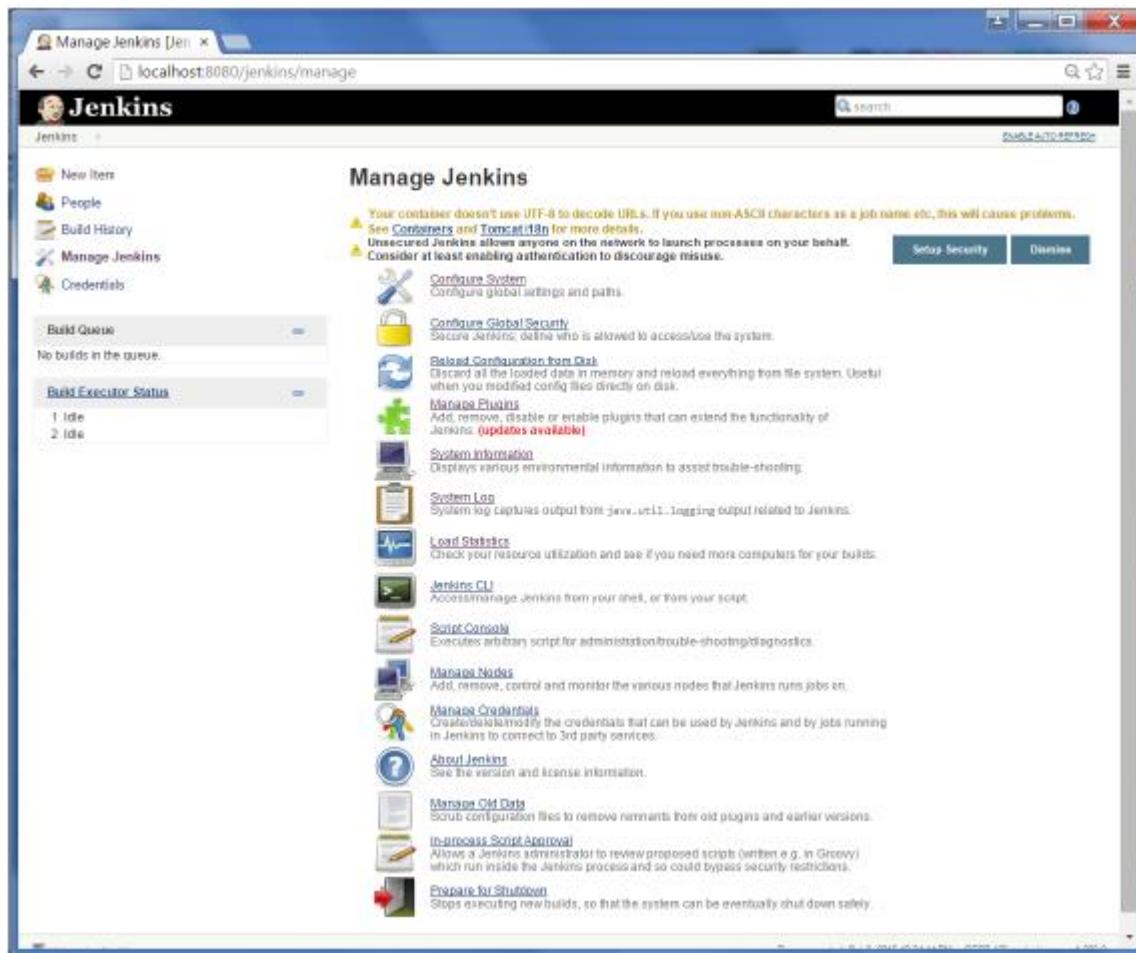
For this exercise, you have to ensure that Internet connectivity is present from the machine on which Jenkins is installed. In your Jenkins Dashboard (Home screen), click the Manage Jenkins option on the left hand side.

Siva Jenkins Tutorial



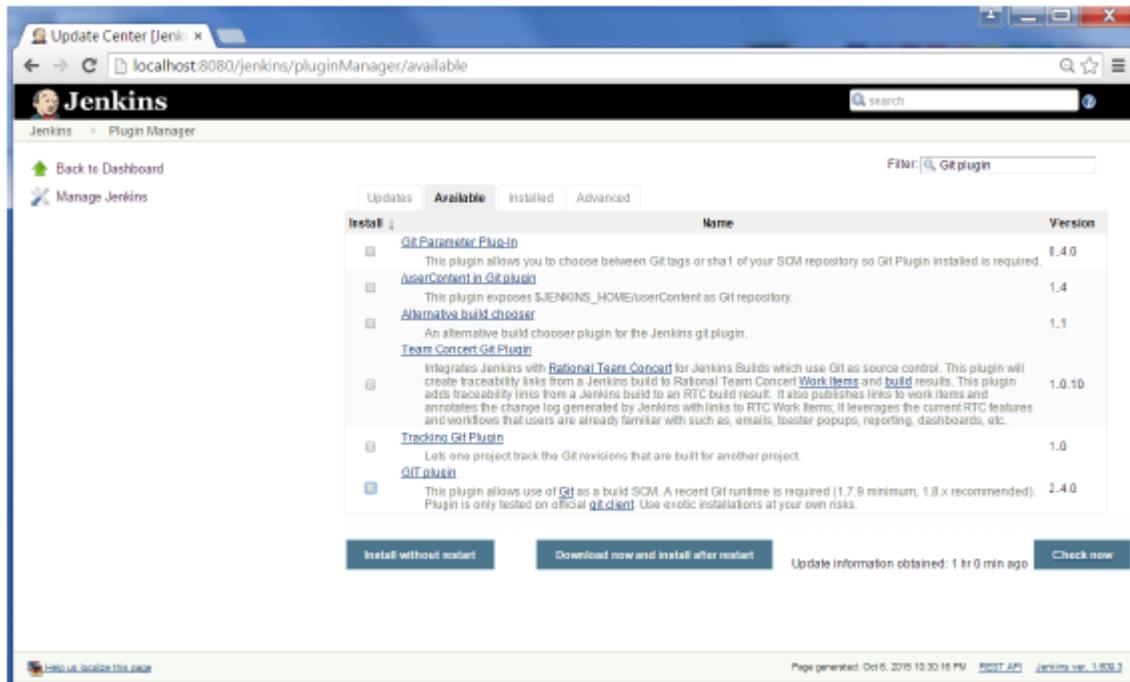
In the next screen, click the ‘Manage Plugins’ option.

Siva Jenkins Tutorial



In the next screen, click the Available tab. This tab will give a list of plugins which are available for downloading. In the 'Filter' tab type 'Git plugin'

Siva Jenkins Tutorial

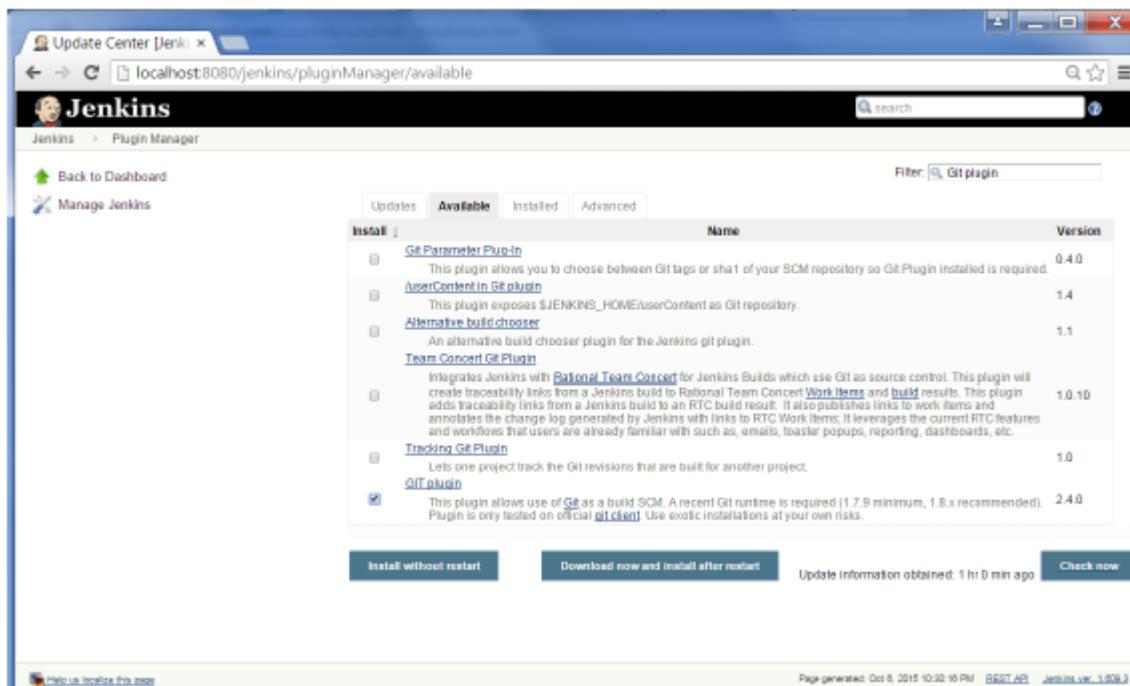


The screenshot shows the Jenkins Plugin Manager interface. The title bar says "Update Center [jenk]" and the address bar shows "localhost:8080/jenkins/pluginManager/available". The main area has a search bar with "search" and a filter bar with "Filter: Git plugin". Below is a table with columns "Name" and "Version". The "Available" tab is selected. A row for "GIT plugin" is highlighted with a blue border. The table contains the following data:

Name	Version
Git Parameter Plug-in	0.4.0
UserContent In Git plugin	1.4
Alternative build chooser	1.1
Team Concert Git Plugin	1.0.10
Tracking Git Plugin	1.0
GIT plugin	2.4.0

At the bottom are buttons: "Install without restart", "Download now and install after restart", "Update information obtained: 1 hr 0 min ago", and "Check now".

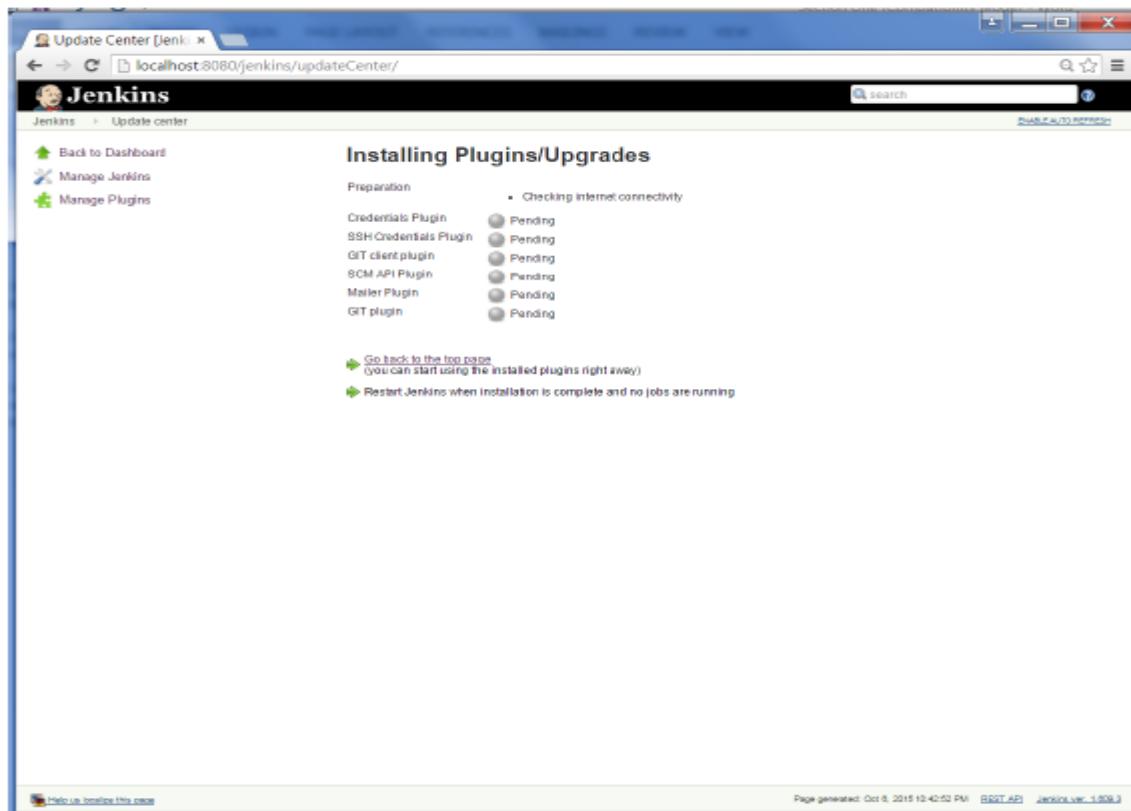
The list will then be filtered. Check the Git Plugin option and click on the button ‘Install without restart’



This screenshot is identical to the one above, showing the Jenkins Plugin Manager with the "Available" tab selected and the "Git plugin" filter applied. The "GIT plugin" row is highlighted with a blue border. The table data is the same as in the previous screenshot.

The installation will then begin and the screen will be refreshed to show the status of the download.

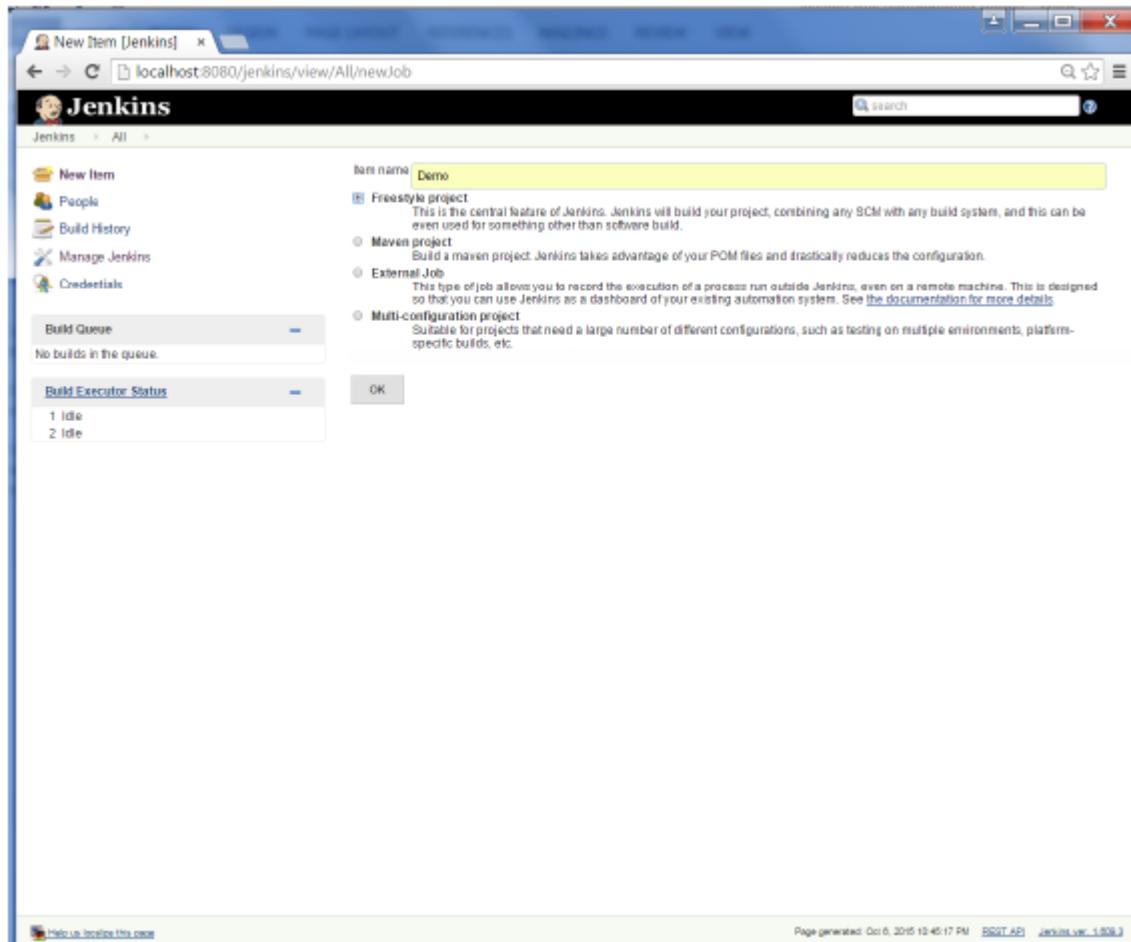
Siva Jenkins Tutorial



Once all installations are complete, restart Jenkins by issue the following command in the browser. <http://localhost:8080/jenkins/restart>

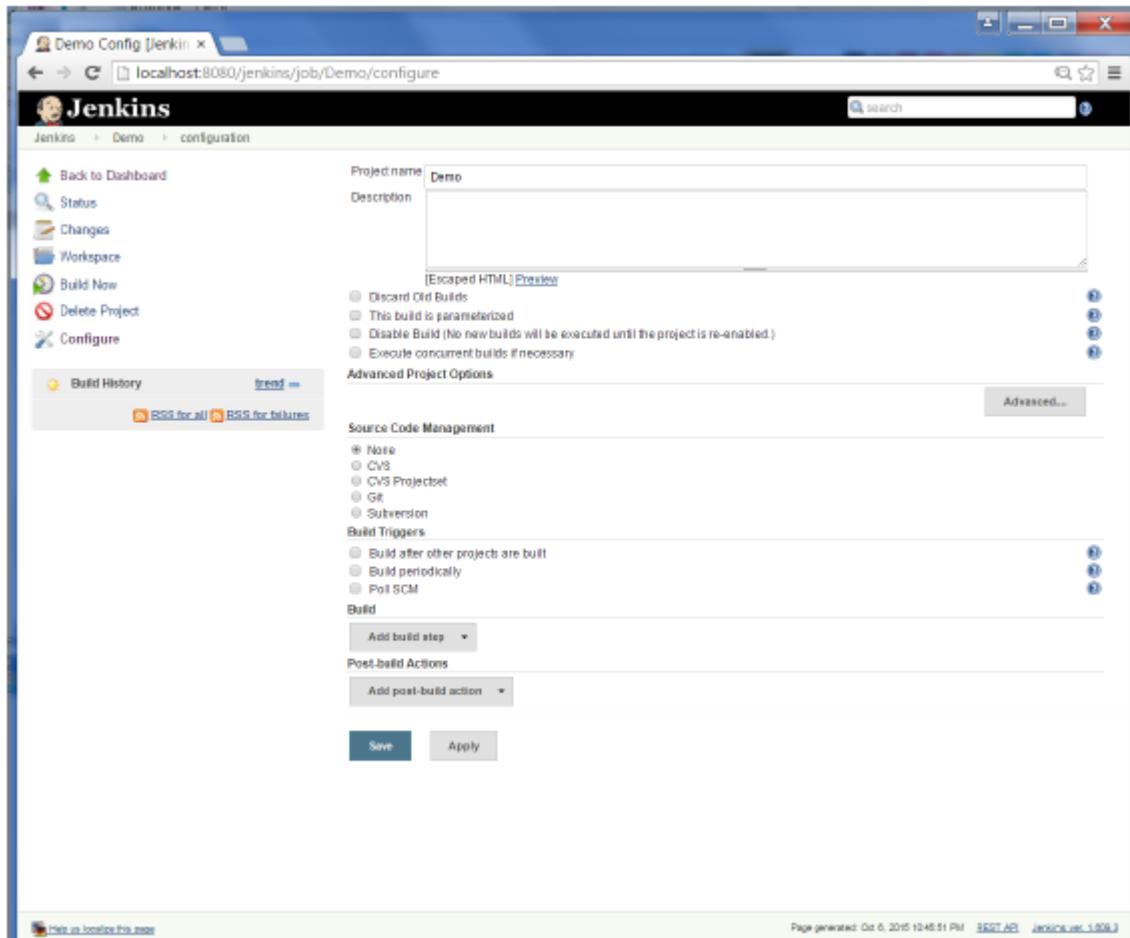
After Jenkins is restarted, Git will be available as an option whilst configuring jobs. To verify, click on New Item in the menu options for Jenkins. Then enter a name for a job, in the following case, the name entered is 'Demo'. Select 'Freestyle project' as the item type. Click the Ok button.

Siva Jenkins Tutorial



In the next screen, if you browse to the Source code Management section, you will now see 'Git' as an option.

Siva Jenkins Tutorial



Jenkins – Maven Setup

Step 1: Downloading and Setting Up Maven

The official website for maven is [Apache Maven](#). If you click the given link, you can get the home page of the maven official website as shown below.

Siva Jenkins Tutorial

The screenshot shows a web browser window displaying the Apache Maven Project download page at <https://maven.apache.org/download.cgi>. The page title is "Apache Maven Project" and the main logo is "Maven™". The left sidebar has a "Download" section selected, containing links for "Install", "Configure", "Run", "IDE Integration", "ABOUT MAVEN", "What is Maven?", "Features", "FAQ", "Support and Training", "DOCUMENTATION", "Maven Plugins", "Index (category)", "Running Maven", "User Centre", "Plugin Developer Centre", "Maven Repository Centre", "Maven Developer Centre", and "Books and Resources". The main content area is titled "Downloading Apache Maven 3.3.3". It states that Maven 3.3.3 is the latest release and recommended version. It includes a note about mirrors and a dropdown for "Other mirrors" set to "http://www.eu.apache.org/di". Below this is a "System Requirements" section with tables for Java Development Kit (JDK), Memory, Disk, and Operating System. At the bottom is a "Files" section with a table for Maven distributions, showing columns for Link, Checksum, and Signature.

	Link	Checksum	Signature
Maven 3.3.3 (binaries)	Download	SHA-1	MD5

While browsing to the site, go to the Files section and download the link to the Binary.zip file.

Siva Jenkins Tutorial

Support and Training
DOCUMENTATION
Maven Plugins
Index (category)
Running Maven
User Centre >
Plugin Developer Centre
Maven Repository Centre
Maven Developer Centre
Books and Resources
Security
COMMUNITY
Community Overview
How to Contribute
Maven Repository
Getting Help
Issue Tracking
Source Repository
The Maven Team
PROJECT DOCUMENTATION
Project Information >
MAVEN PROJECTS
Ant Tasks
Archetype
Doxia
BXD

Memory No minimum requirement

Disk Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.

Operating System No minimum requirement. Start up scripts are included as shell scripts and Windows batch files.

Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [Installation Instructions](#). Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Maven developers.

Link	Checksum	Signature
Binary tar.gz archive	apache-maven-3.3.3-bin.tar.gz	apache-maven-3.3.3-bin.tar.gz.md5
Binary zip archive	apache-maven-3.3.3-bin.zip	apache-maven-3.3.3-bin.zip.md5
Source tar.gz archive	apache-maven-3.3.3-src.tar.gz	apache-maven-3.3.3-src.tar.gz.md5
Source zip archive	apache-maven-3.3.3-src.zip	apache-maven-3.3.3-src.zip.md5

- [Release Notes](#)
- [Reference Documentation](#)
- [Apache Maven Website As Documentation Archive](#)
- All sources (plugins, shared libraries, ...) available at <http://www.apache.org/dist/maven/>
- Distributed under the Apache License, version 2.0

Previous Releases

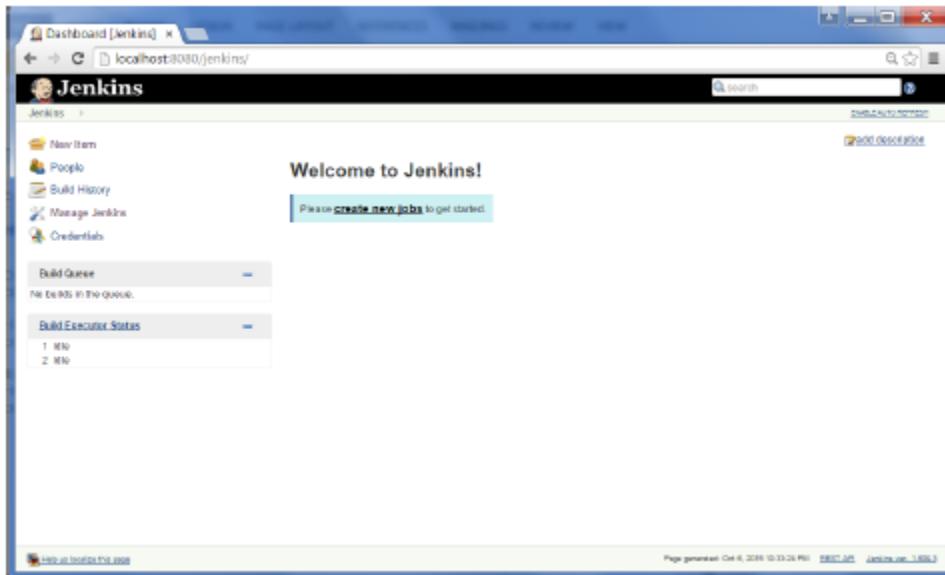
It is strongly recommended to use the latest release version of Apache Maven to take advantage of newest features and bug fixes. If you still want to use an old version you can find more information in the [Maven Releases History](#) and can download files from the [archives](#) for versions 3.0.4+ and [legacy archives](#) for earlier releases.

Once the file is downloaded, extract the files to the relevant application folder. For this purpose, the maven files will be placed in E:\Apps\apache-maven-3.3.3.

Step 2: Setting up Jenkins and Maven

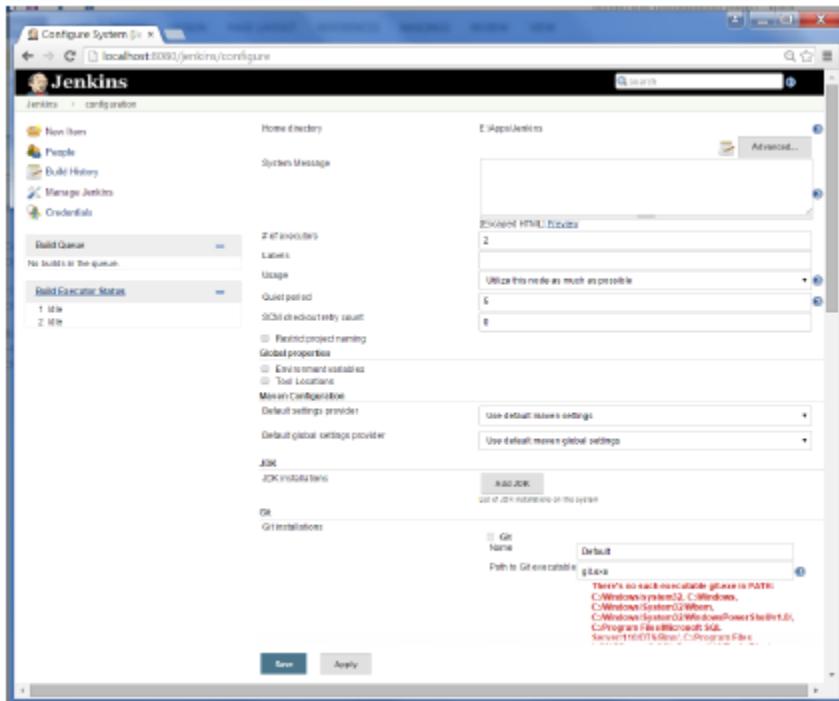
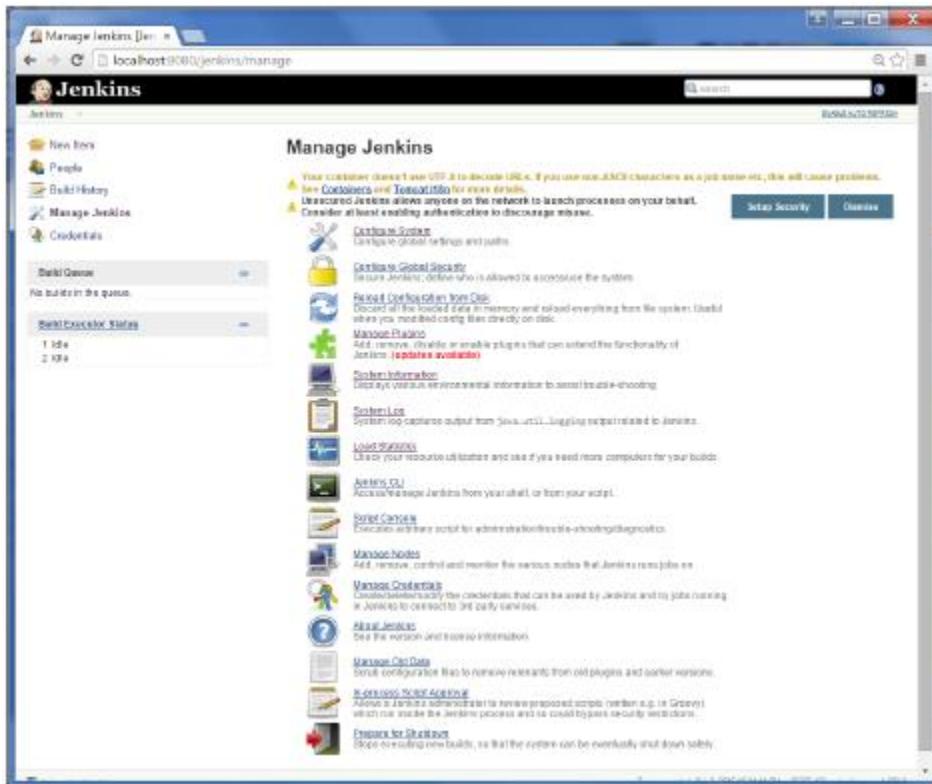
In the Jenkins dashboard (Home screen), click Manage Jenkins from the left-hand side menu.

Siva Jenkins Tutorial



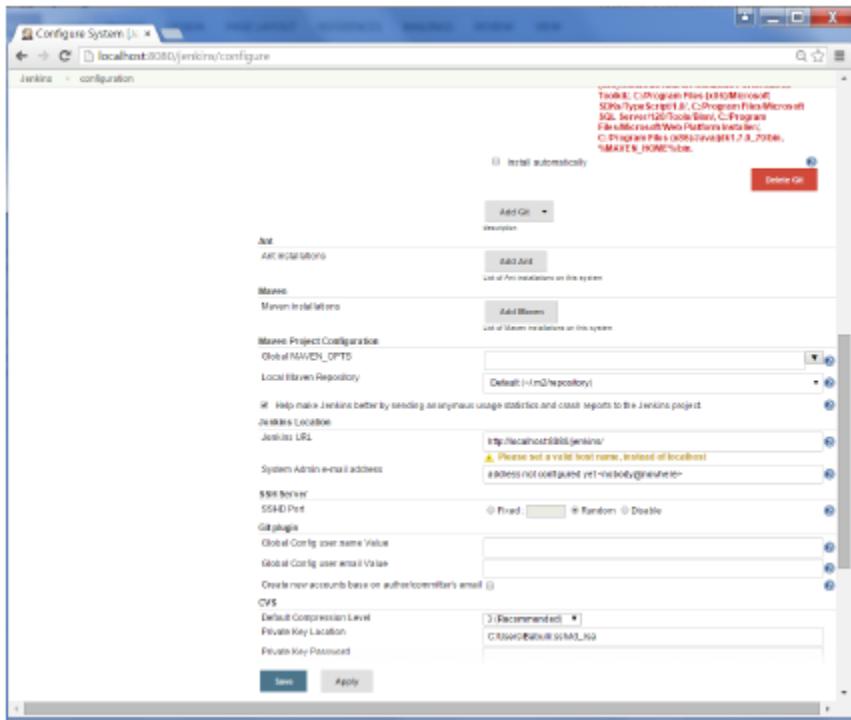
Then, click on ‘Configure System’ from the right hand side.

Siva Jenkins Tutorial



In the Configure system screen, scroll down till you see the Maven section and then click on the 'Add Maven' button.

Siva Jenkins Tutorial

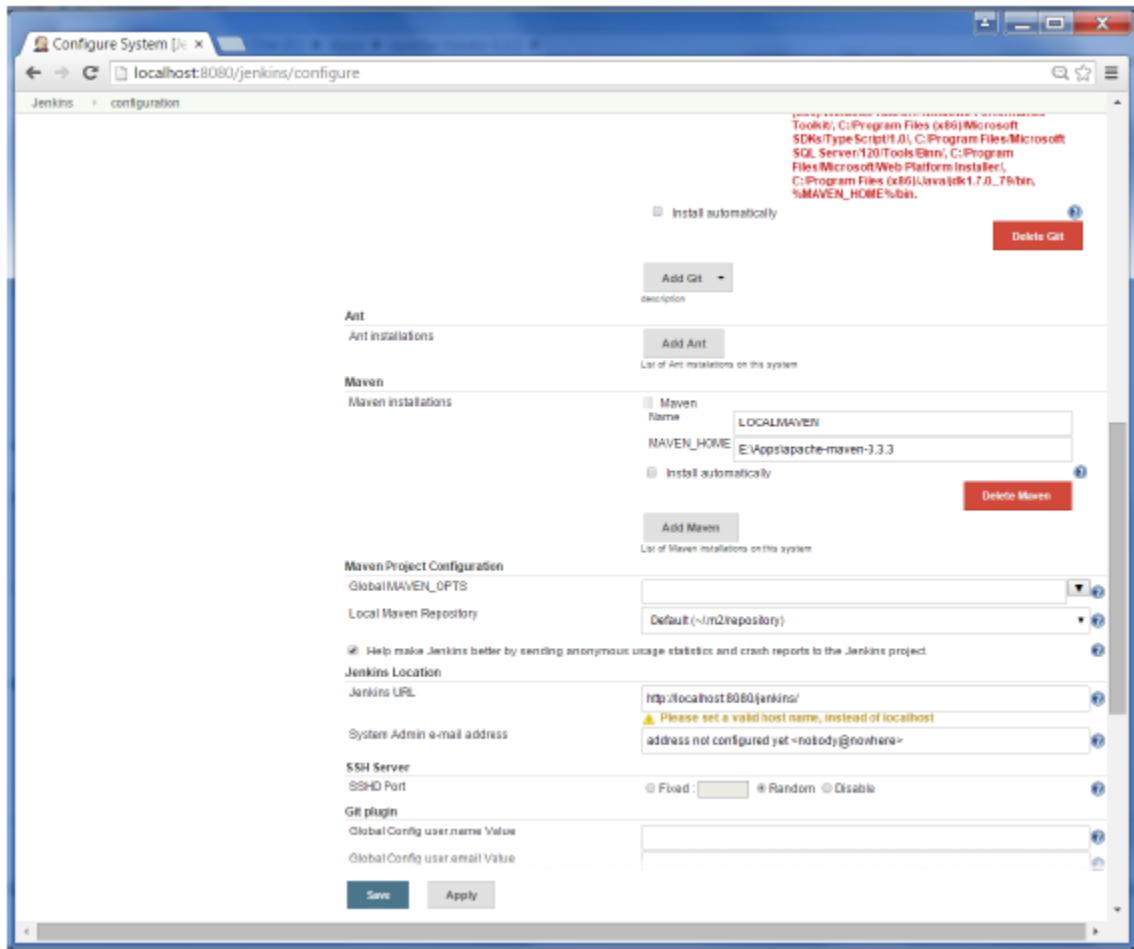


Uncheck the 'Install automatically' option.

Add any name for the setting and the location of the MAVEN_HOME.

Then, click on the 'Save' button at the end of the screen.

Siva Jenkins Tutorial



You can now create a job with the 'Maven project' option. In the Jenkins dashboard, click the New Item option.

Siva Jenkins Tutorial

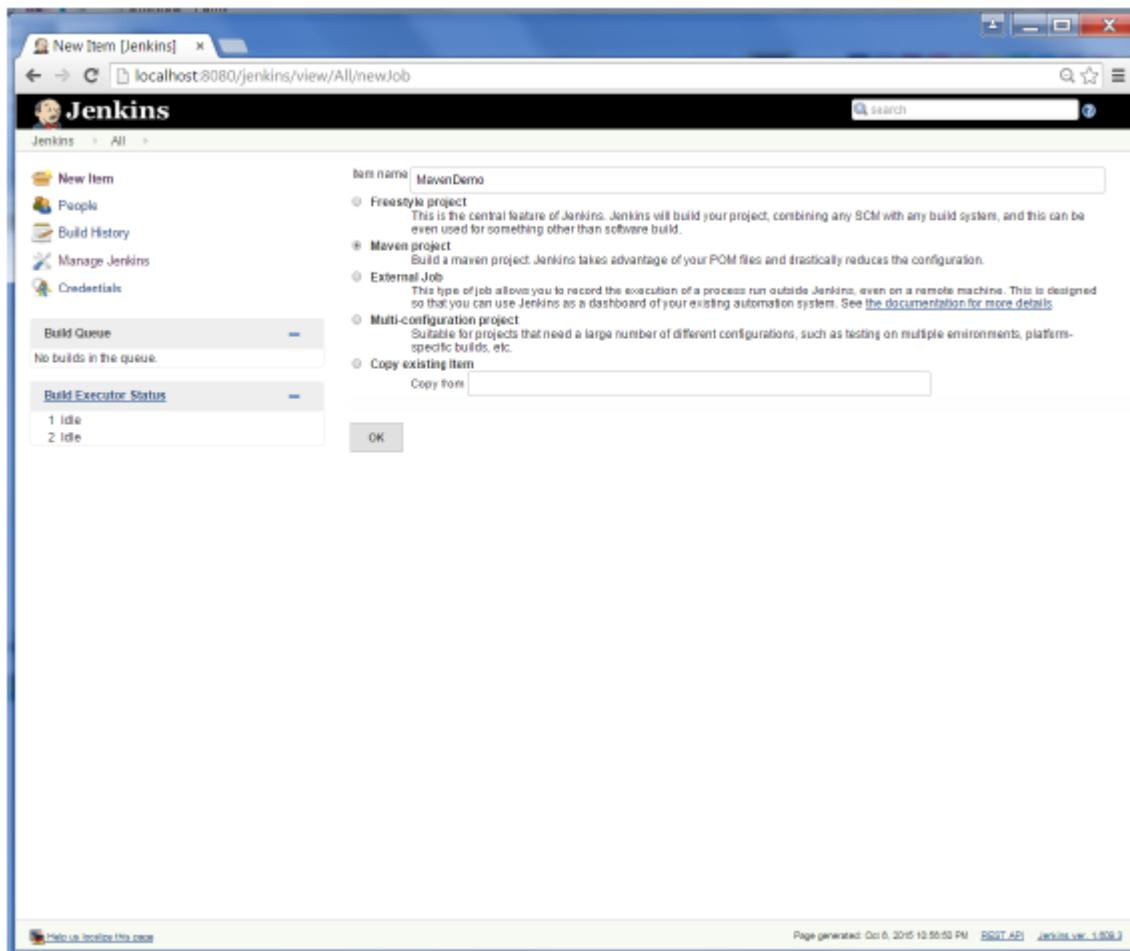
The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. The main interface includes:

- A left sidebar with links: New Item, People, Build History, Manage Jenkins, and Credentials.
- A central table view showing a single job entry:

S	W	Name	Last Success	Last Failure	Last Duration
Idle	Idle	Demo	N/A	N/A	N/A

- Below the table, a legend indicates four RSS feed options: RSS for all, RSS for failures, RSS for just failed builds, and RSS for just latest builds.
- Two expandable sections: "Build Queue" (No builds in the queue) and "Build Executor Status" (1 Idle, 2 Idle).
- Page footer with links: Help Us Improve This Page, Page generated: Oct 6, 2015 12:55:57 PM, REST API, Jenkins ver. 1.608.3.

Siva Jenkins Tutorial

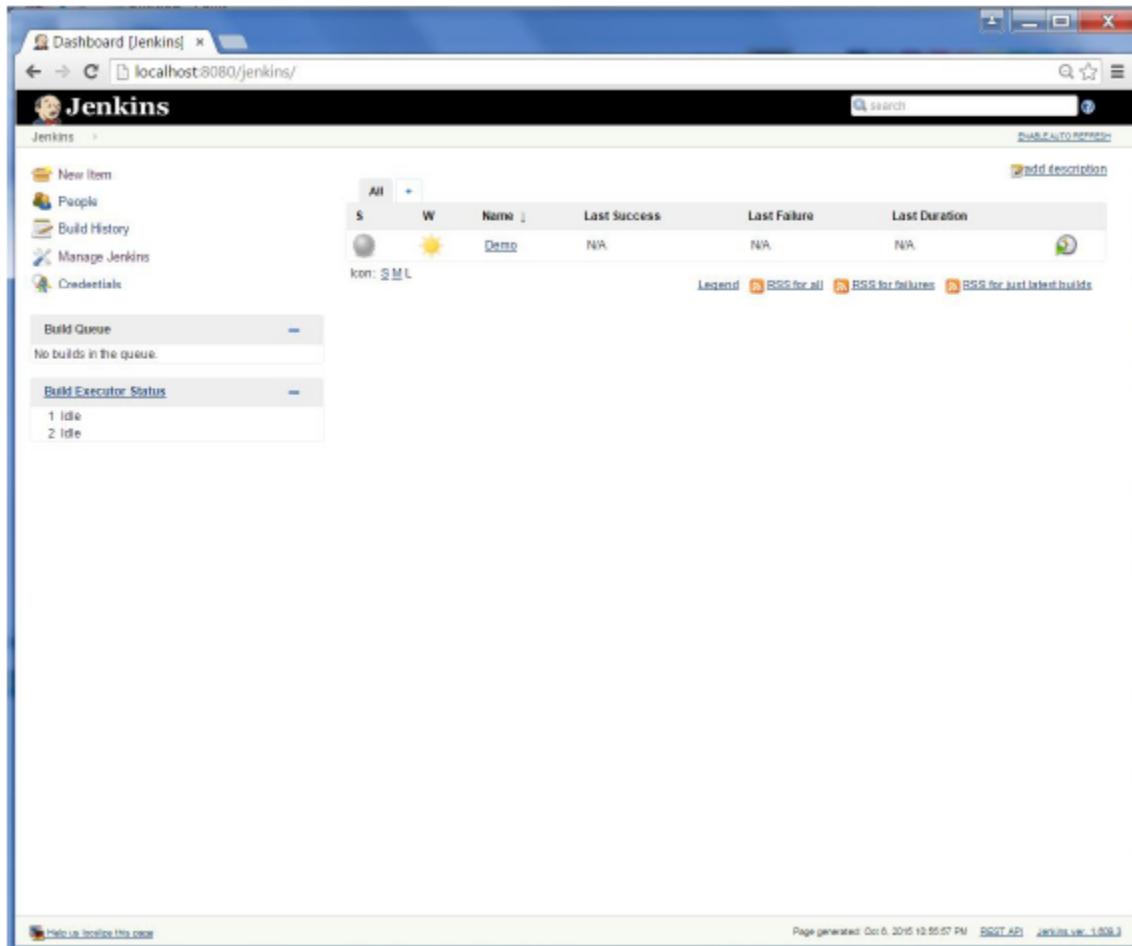


Jenkins – Configuration

You probably would have seen a couple of times in the previous exercises wherein we had to configure options within Jenkins. The following shows the various configuration options in Jenkins.

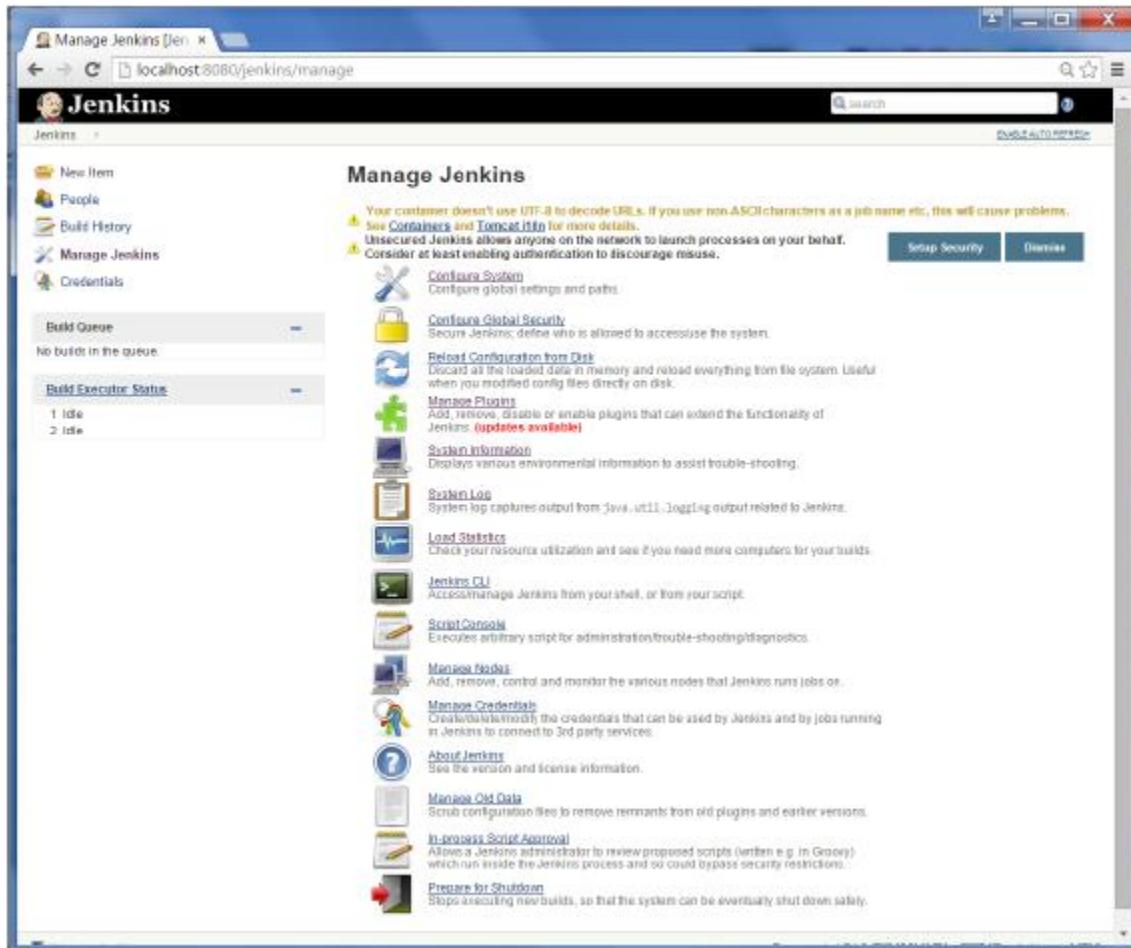
So one can get the various configuration options for Jenkins by clicking the ‘Manage Jenkins’ option from the left hand menu side.

Siva Jenkins Tutorial



You will then be presented with the following screen –

Siva Jenkins Tutorial



Click on Configure system. Discussed below are some of the Jenkins configuration settings which can be carried out.

Jenkins Home Directory

Jenkins needs some disk space to perform builds and keep archives. One can check this location from the configuration screen of Jenkins. By default, this is set to `~/jenkins`, and this location will initially be stored within your user profile location. In a proper environment, you need to change this location to an adequate location to store all relevant builds and archives. Once can do this in the following ways

- Set "JENKINS_HOME" environment variable to the new home directory before launching the servlet container.
- Set "JENKINS_HOME" system property to the servlet container.
- Set JNDI environment entry "JENKINS_HOME" to the new directory.

Siva Jenkins Tutorial

The following example will use the first option of setting the "JENKINS_HOME" environment variable.

First create a new folder E:\Apps\Jenkins. Copy all the contents from the existing ~/.jenkins to this new directory.

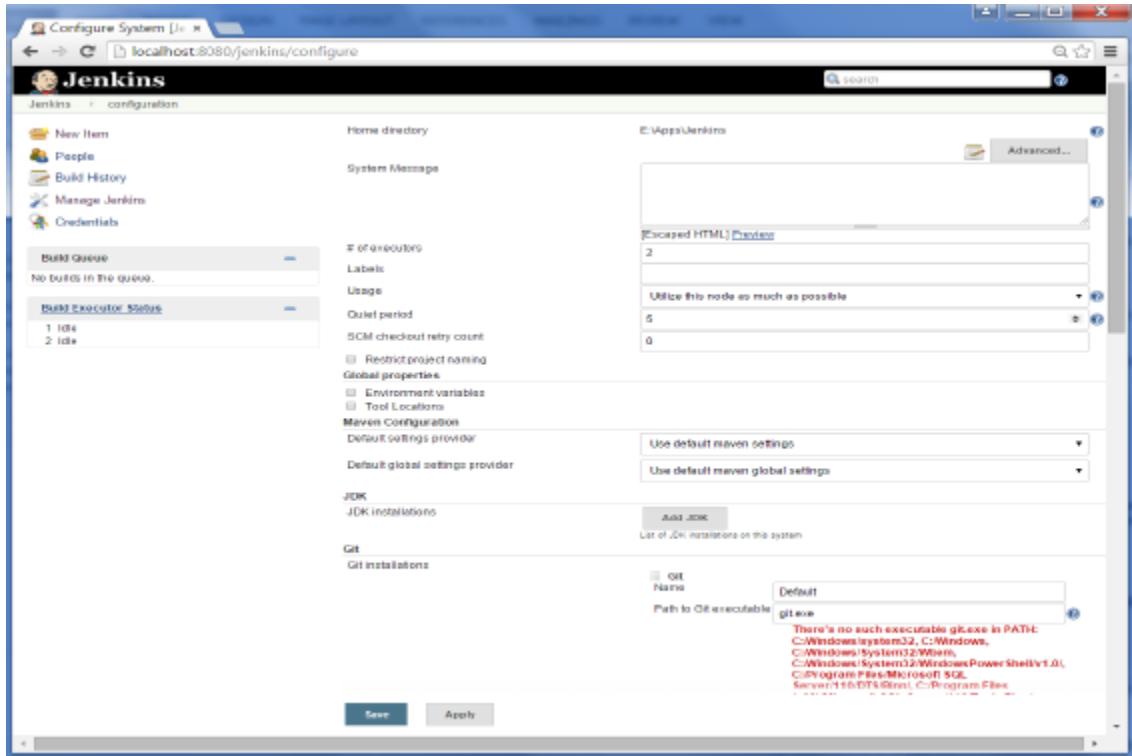
Set the JENKINS_HOME environment variable to point to the base directory location where Java is installed on your machine. For example,

OS	Output
Windows	Set Environmental variable JENKINS_HOME to you're the location you desire. As an example you can set it to E:\Apps\Jenkins
Linux	Export JENKINS_HOME =/usr/local/Jenkins or the location you desire.

In the Jenkins dashboard, click Manage Jenkins from the left hand side menu. Then click on 'Configure System' from the right hand side.

In the Home directory, you will now see the new directory which has been configured.

Siva Jenkins Tutorial



of executors

This refers to the total number of concurrent job executions that can take place on the Jenkins machine. This can be changed based on requirements. Sometimes the recommendation is to keep this number the same as the number of CPU on the machines for better performance.

Environment Variables

This is used to add custom environment variables which will apply to all the jobs. These are key-value pairs and can be accessed and used in Builds wherever required.

Jenkins URL

By default, the Jenkins URL points to local host. If you have a domain name setup for your machine, set this to the domain name else overwrite local host with IP of machine. This will help in setting up slaves and while sending out links using the email as you can directly access the Jenkins URL using the environment variable JENKINS_URL which can be accessed as \${JENKINS_URL}.

Siva Jenkins Tutorial

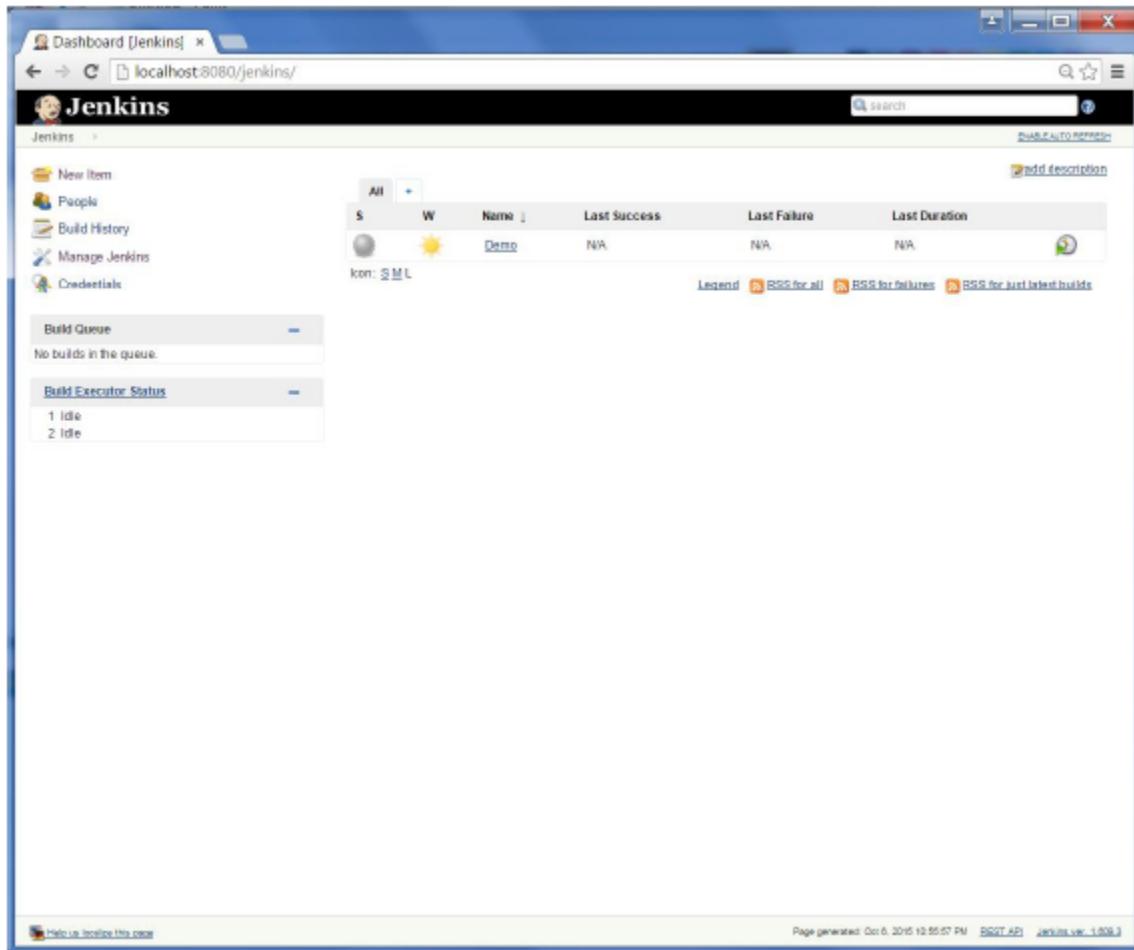
Email Notification

In the email Notification area, you can configure the SMTP settings for sending out emails. This is required for Jenkins to connect to the SMTP mail server and send out emails to the recipient list.

Jenkins – Management

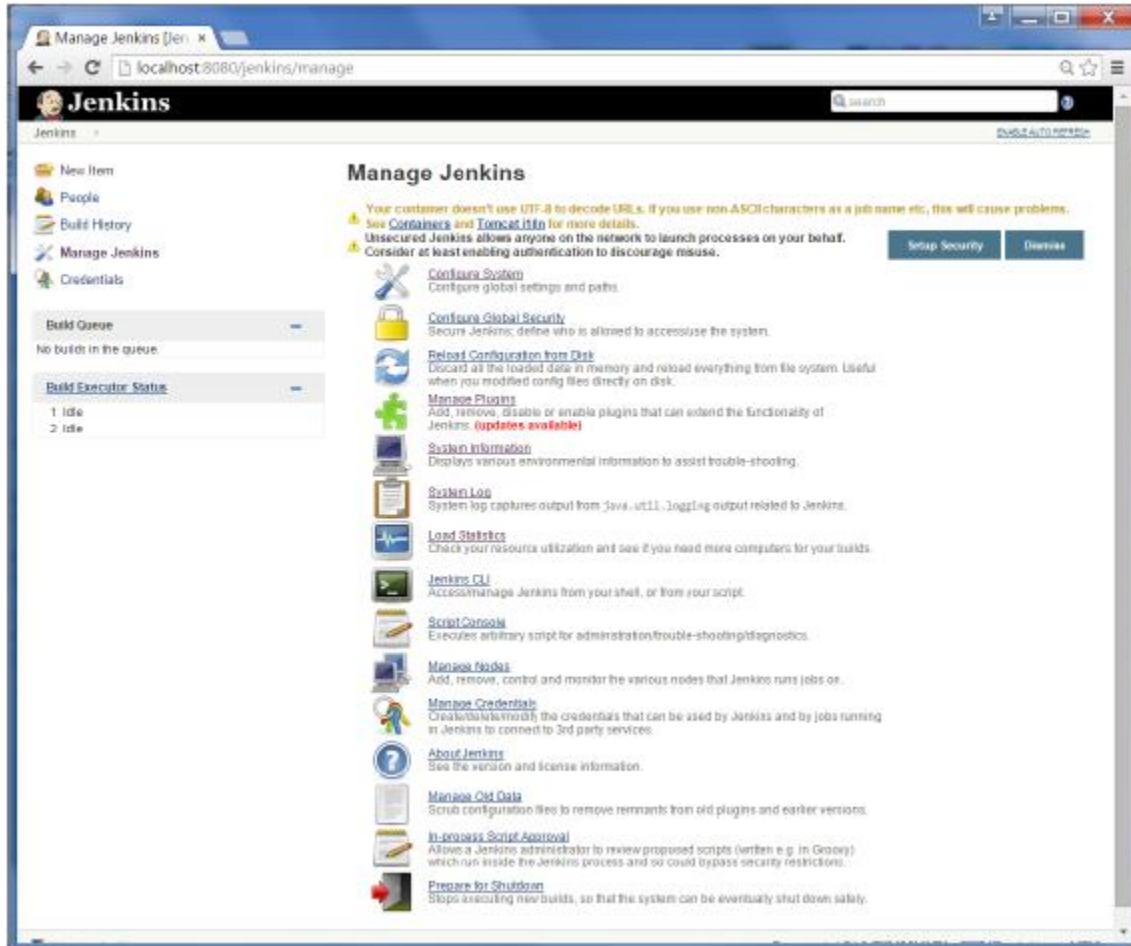
To manage Jenkins, click on the ‘Manage Jenkins’ option from the left hand menu side.

So one can get the various configuration options for Jenkins by clicking the ‘Manage Jenkins’ option from the left hand menu side.



You will then be presented with the following screen –

Siva Jenkins Tutorial



Some of the management options are as follows –

Configure System

This is where one can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed, Jenkins will add the required configuration fields dynamically after the plugins are installed.

Reload Configuration from Disk

Jenkins stores all its system and build job configuration details as XML files which is stored in the Jenkins home directory. Here also all of the build history is stored. If you are migrating build jobs from one Jenkins instance to another, or archiving old build jobs, you will need to add or remove the corresponding build job directories to Jenkins's builds directory. You don't need to take Jenkins offline to do this—you can simply use the "Reload Configuration from Disk" option to reload the Jenkins system and build job configurations directly.

Siva Jenkins Tutorial

Manage Plugin

Here one can install a wide variety of third-party plugins right from different Source code management tools such as Git, Mercurial or Clear Case, to code quality and code coverage metrics reporting. Plugins can be installed, updated and removed through the Manage Plugins screen.

The screenshot shows the Jenkins Manage Plugin interface. At the top, there are tabs for 'Updates', 'Available', 'Installed', and 'Advanced'. The 'Updates' tab is selected. A table lists various plugins:

Name	Version	Installed
CVS Plugin	2.12	2.11
Javadoc Plugin	1.3	1.1
JUnit Plugin	1.9	1.2-beta-4
Matrix Authorization Strategy Plugin	1.2	1.1
Matrix Project Plugin	1.6	1.4.1
Maven Integration plugin	2.12.1	2.7.1
OWASP Markup Formatter Plugin	1.3	1.1
PAM Authentication plugin	1.2	1.1
Script Security Plugin	1.15	1.13
SSH Slaves plugin	1.10	1.9
Subversion Plugin	2.5.3	1.54
Translation Assistance plugin	1.12	1.10
Windows Slaves Plugin	1.1	1.0

At the bottom, there are buttons for 'Download now and install after restart' and 'Check now'. A note says 'Select All. None' and 'This page lists updates to the plugins you currently use.'

System Information

This screen displays a list of all the current Java system properties and system environment variables. Here one can check exactly what version of Java Jenkins is running in, what user it is running under, and so forth.

The following screenshot shows some of the name-value information available in this section.

Siva Jenkins Tutorial

The screenshot shows the Jenkins System Information page at localhost:8080/jenkins/systemInfo. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, and Credentials. Under Build Queue, it says "No builds in the queue". Under Build Executor Status, it shows "1 Idle" and "2 Idle". The main content is a table titled "System Properties" with columns "Name" and "Value". The table lists various Java system properties such as awt.toolkit, catalina.base, catalina.home, catalina.useNaming, common.loader, file.encoding, file.encoding.pkg, file.separator, java.awt.graphicsenv, java.awt.printerjob, java.class.path, java.class.version, java.endorsed.dirs, java.ext.dirs, java.home, and java.io.tmpdir, among others.

Name	Value
awt.toolkit	sun.awt.windows.WToolkit
catalina.base	E:\Appstromcat7
catalina.home	E:\Appstromcat7
catalina.useNaming	true
common.loader	\$[catalina.base]lib;\$[catalina.base]lib*;jar;\$[catalina.home]\lib;\$[catalina.home]\lib*;jar
file.encoding	Cp1252
file.encoding.pkg	sun.jo
file.separator	\
java.awt.graphicsenv	sun.awt.Win32GraphicsEnvironment
java.awt.printerjob	sun.awt.windows.WPrinterJob
java.class.path	E:\Appstromcat7\bin\bootstrap.jar;E:\Appstromcat7\bin\tomcat-juli.jar
java.class.version	51.0
java.endorsed.dirs	E:\Appstromcat7\endorsed
java.ext.dirs	C:\Program Files (x86)\Java\jdk1.7.0_79\jre\lib\ext;C:\Windows\SunJava\lib\ext
java.home	C:\Program Files (x86)\Java\jdk1.7.0_79\jre
java.io.tmpdir	E:\Appstromcat7\temp
java.library.path	C:\Program Files (x86)\Java\jdk1.7.0_79\bin;C:\Windows\SunJava\bin;C:\Windows\system2;C:\Windows;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft SQL Server\Server110\Tools\Binn\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\ManagementStudio\;C:\Program Files\Microsoft SQL Server\100\Tools\Binn\ManagementStudio\;C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\PrivateAssemblies\;C:\Program Files (x86)\Microsoft SQL Server\110\Tools\Binn\;C:\Program Files (x86)\Windows Kits\8.1\Windows Performance Toolkit\;C:\Program Files (x86)\Microsoft SDKs\TypeScript\1.0\;C:\Program Files\Microsoft SQL Server\120\Tools\Binn\;C:\Program Files\Microsoft\Web Platform Installer\;C:\Program Files (x86)\Java\jdk1.7.0_79\bin;%MAVEN_HOME%\bin;
java.naming.factory.initial	org.apache.naming.java.javaURLContextFactory
java.naming.factory.url.pkgs	org.apache.naming
java.runtime.name	Java(TM) SE Runtime Environment
java.runtime.version	1.7.0_79-b15
java.specification.name	Java Platform API Specification
java.specification.vendor	Oracle Corporation
java.specification.version	1.7
java.util.logging.config.file	E:\Appstromcat7\conf\logging.properties
java.util.logging.manager	org.apache.juli.ClassLoaderLogManager
java.vendor	Oracle Corporation
java.vendor.url	http://java.oracle.com/
java.vendor.url_bug	http://bugreport.sun.com/bugreport/
java.version	1.7.0_79
java.vm.info	mixed mode, sharing

System Log

The System Log screen is a convenient way to view the Jenkins log files in real time. Again, the main use of this screen is for troubleshooting.

Load Statistics

This page displays graphical data on how busy the Jenkins instance is in terms of the number of concurrent builds and the length of the build queue which gives an idea of how long your builds need to wait before being executed. These statistics can give a good idea of whether extra capacity or extra build nodes is required from an infrastructure perspective.

Script Console

This screen lets you run Groovy scripts on the server. It is useful for advanced troubleshooting since it requires a strong knowledge of the internal Jenkins architecture.

Siva Jenkins Tutorial

Manage nodes

Jenkins is capable of handling parallel and distributed builds. In this screen, you can configure how many builds you want. Jenkins runs simultaneously, and, if you are using distributed builds, set up build nodes. A build node is another machine that Jenkins can use to execute its builds.

Prepare for Shutdown

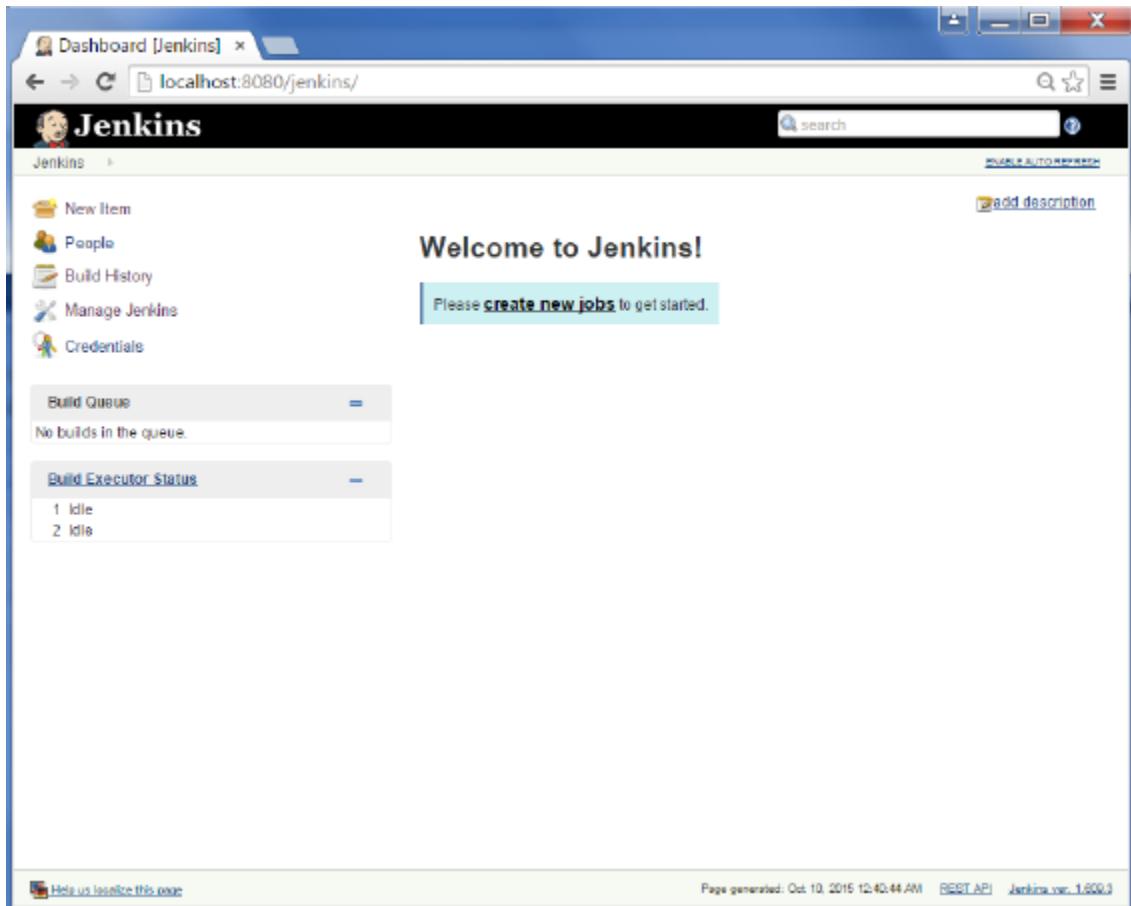
If there is a need to shut down Jenkins, or the server Jenkins is running on, it is best not to do so when a build is being executed. To shut down Jenkins cleanly, you can use the Prepare for Shutdown link, which prevents any new builds from being started. Eventually, when all of the current builds have finished, one will be able to shut down Jenkins cleanly

Jenkins - Setup Build Jobs

For this exercise, we will create a job in Jenkins which picks up a simple HelloWorld application, builds and runs the java program.

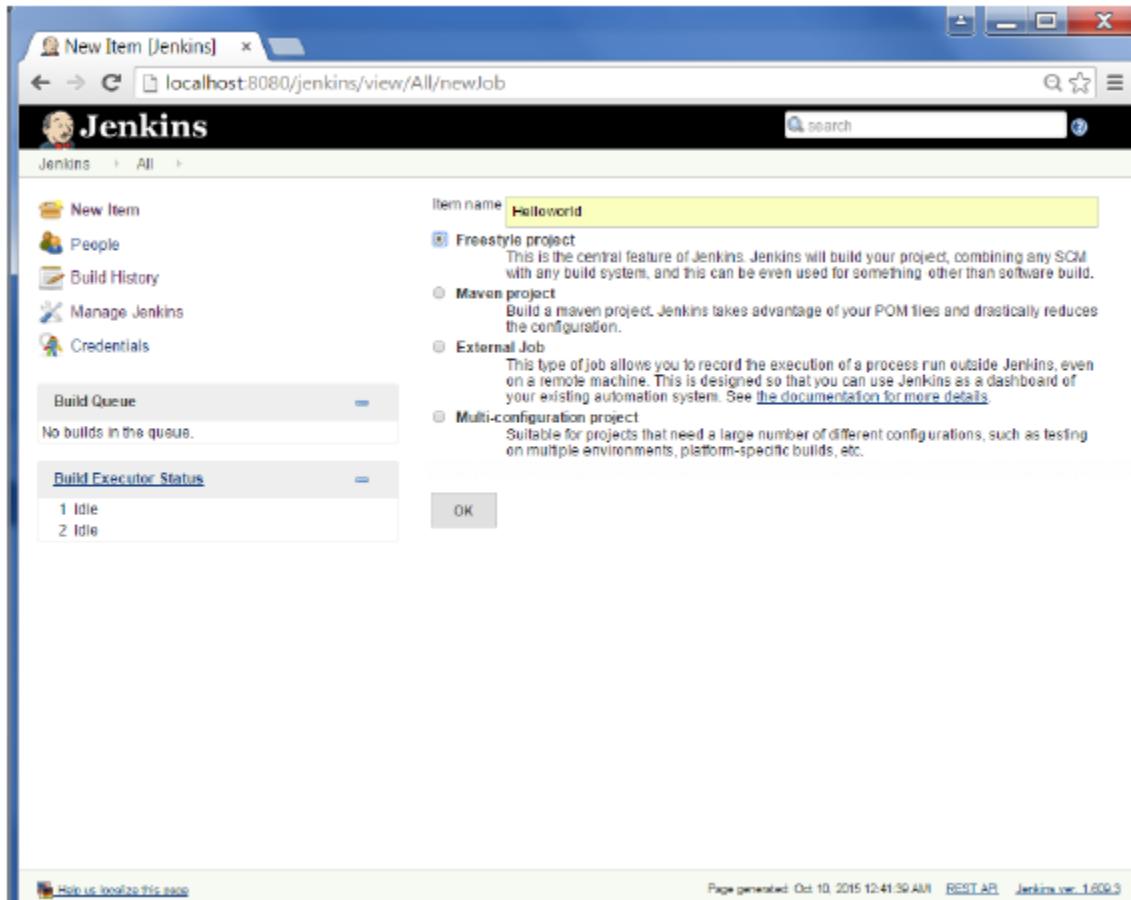
Step 1 – Go to the Jenkins dashboard and Click on New Item

Siva Jenkins Tutorial



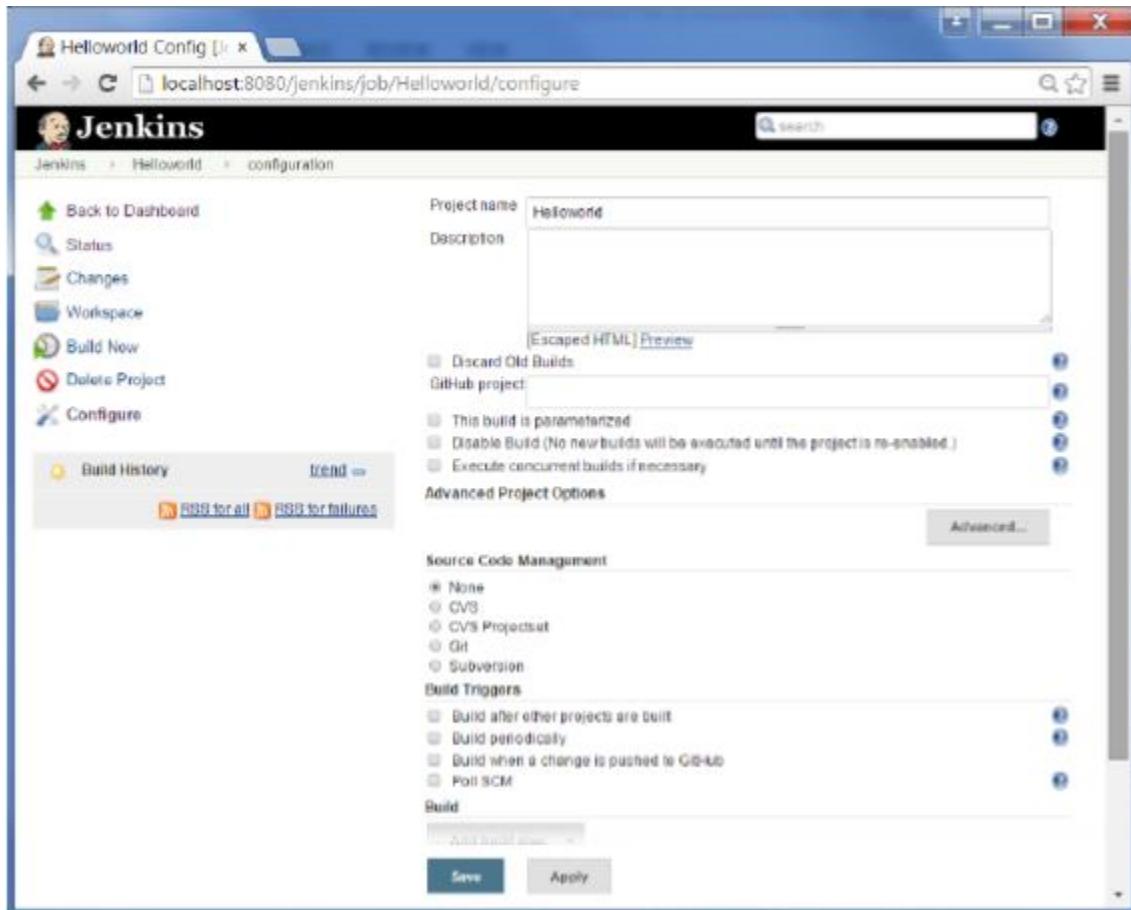
Step 2 – In the next screen, enter the Item name, in this case we have named it Hello world. Choose the ‘Freestyle project option’

Siva Jenkins Tutorial



Step 3 – the following screen will come up in which you can specify the details of the job.

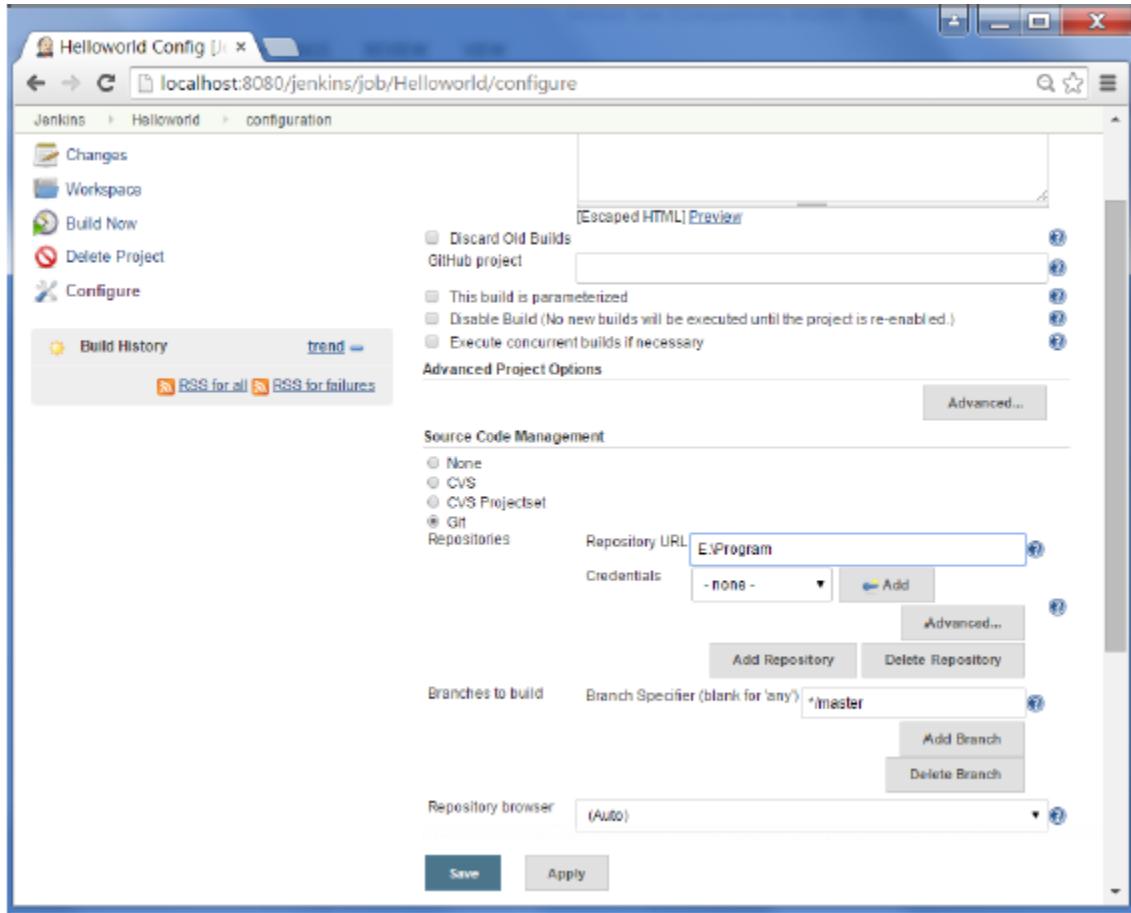
Siva Jenkins Tutorial



Step 4 – we need to specify the location of files which need to be built. In this example, we will assume that a local git repository (E:\Program) has been setup which contains a ‘HelloWorld.java’ file. Hence scroll down and click on the Git option and enter the URL of the local git repository.

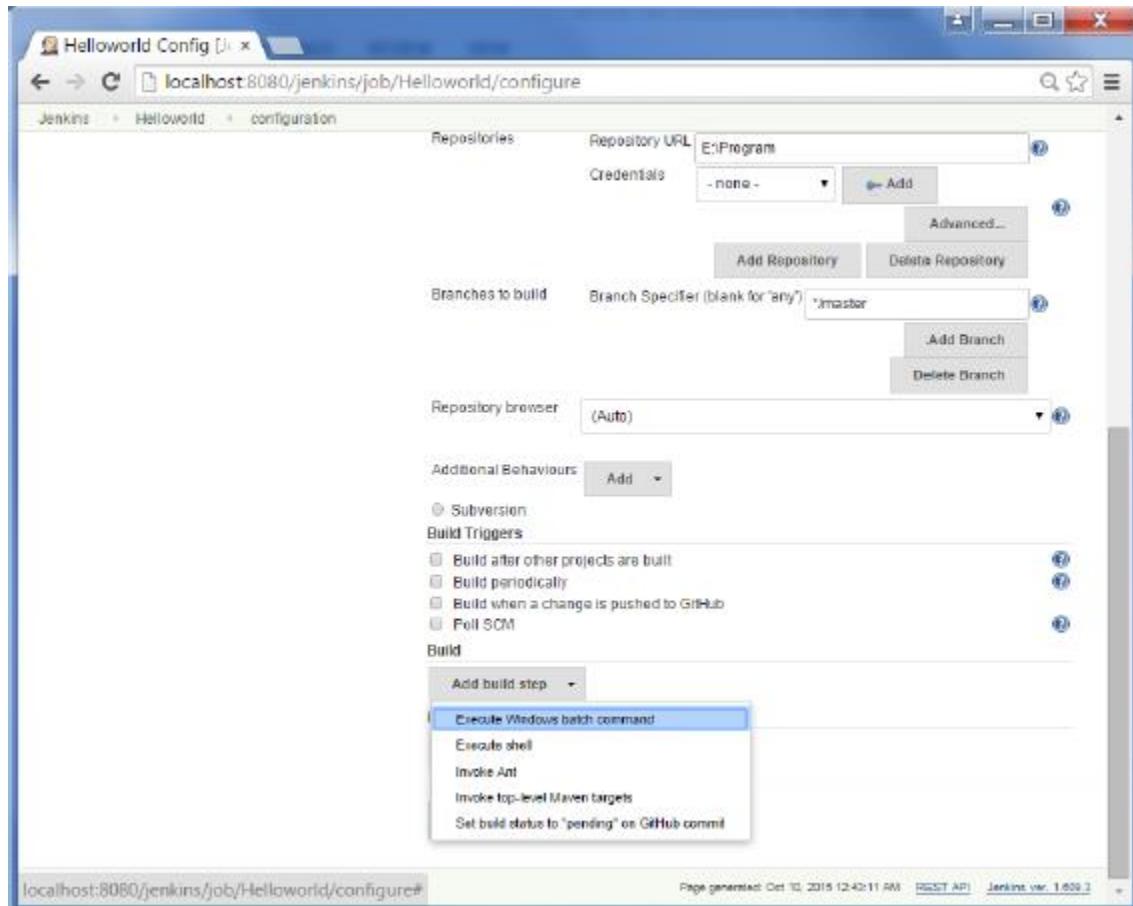
Note – If your repository is hosted on Github, you can also enter the url of that repository here. In addition to this, you would need to click on the Add button for the credentials to add a user name and password to the github repository so that the code can be picked up from the remote repository.

Siva Jenkins Tutorial



Step 5 – Now go to the Build section and click on Add build step → Execute Windows batch command

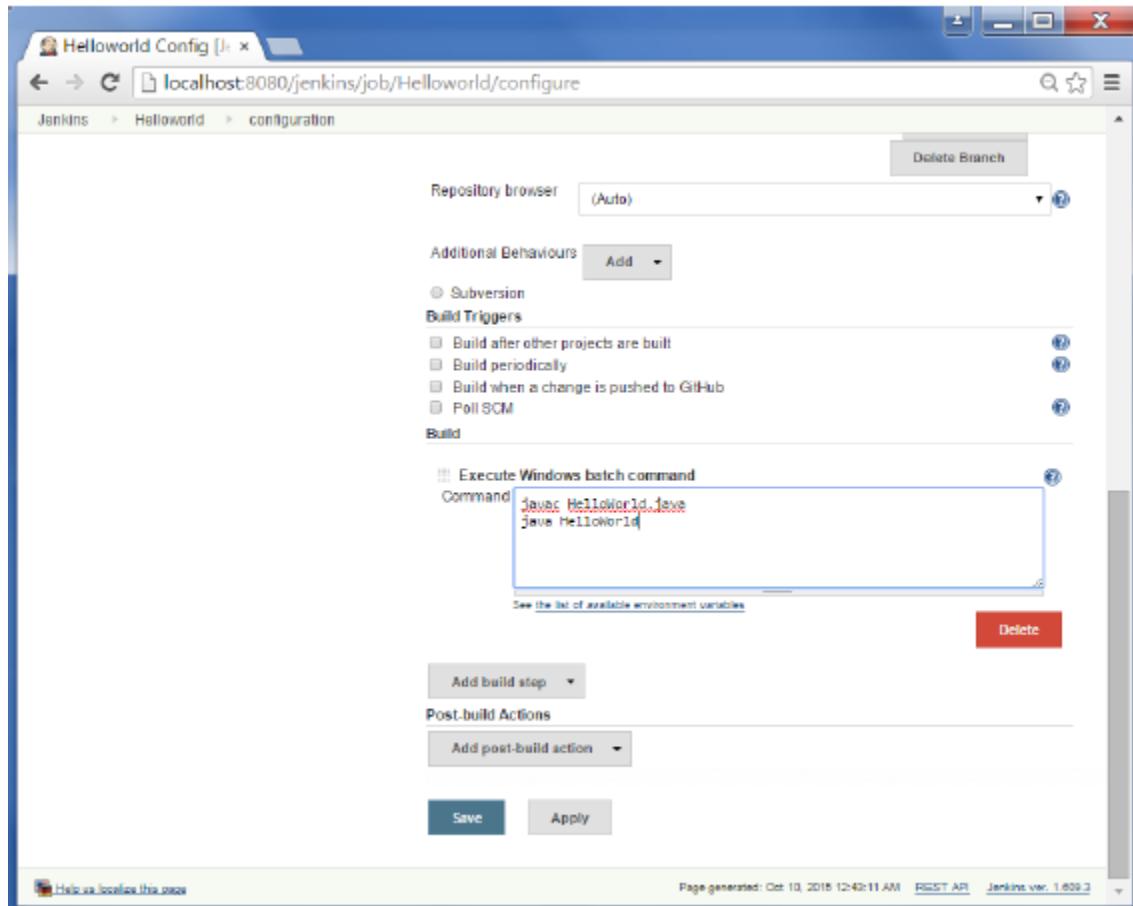
Siva Jenkins Tutorial



Step 6 – In the command window, enter the following commands and then click on the Save button.

```
Javac HelloWorld.java  
Java HelloWorld
```

Siva Jenkins Tutorial



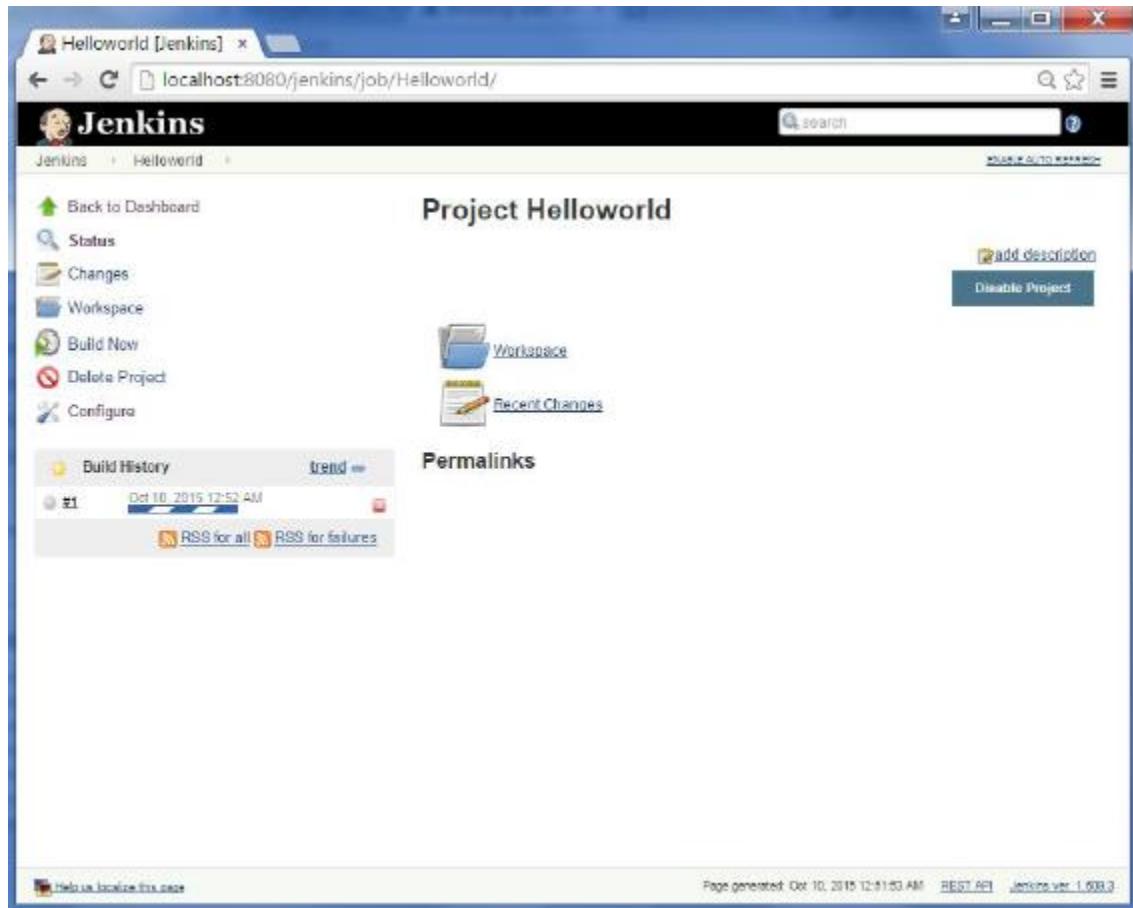
Step 7 – Once saved, you can click on the Build Now option to see if you have successfully defined the job.

Siva Jenkins Tutorial

The screenshot shows a web browser window displaying the Jenkins interface for a project named "Helloworld". The URL in the address bar is "localhost:8080/jenkins/job/Helloworld/". The main title is "Project Helloworld". On the left, there is a sidebar with links: "Back to Dashboard", "Status", "Changes", "Workspace", "Build Now", "Delete Project", and "Configure". Below these are "Build History" and "Permalinks" sections, with "RSS for all" and "RSS for failures" options. On the right, there are buttons for "Add description" and "Disable Project". A "Workspace" link is also present. At the bottom, there is a footer with links for "Help us localize this page", "Page generated: Oct 10, 2015 12:01:03 AM", "REST API", and "Jenkins ver. 1.609.3".

Step 8 – Once the build is scheduled, it will run. The following Build history section shows that a build is in progress.

Siva Jenkins Tutorial



Step 9 – Once the build is completed, a status of the build will show if the build was successful or not. In our case, the following build has been executed successfully. Click on the #1 in the Build history to bring up the details of the build.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for the 'Helloworld' project. At the top, there's a navigation bar with links for 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. Below this is a 'Project Helloworld' header. To the right of the header are buttons for 'add description' and 'Disable Project'. On the left, there's a 'Build History' section showing one build from Oct 10, 2015 at 12:52 AM. To the right of the history are links for 'Workspace' and 'Recent Changes'. At the bottom, there's a 'Permalinks' section with links for 'RSS for all' and 'RSS for failures'. The footer contains links for 'Help us localize this page', 'Page generated: Oct 10, 2015 12:51:53 AM', 'REST API', and 'Jenkins ver. 1.639.3'.

Step 10 – Click on the Console Output link to see the details of the build

Siva Jenkins Tutorial

Helloworld #1 [Jenkins] x localhost:8080/jenkins/job/Helloworld/1/ Jenkins > Helloworld > #1

Back to Project Status Changes Console Output Edit Build Information Delete Build Git Build Data No Tags

Build #1 (Oct 10, 2015 12:52:50 AM) Started 4 min 40 sec ago Took 4.7 sec

No changes.

Started by anonymous user

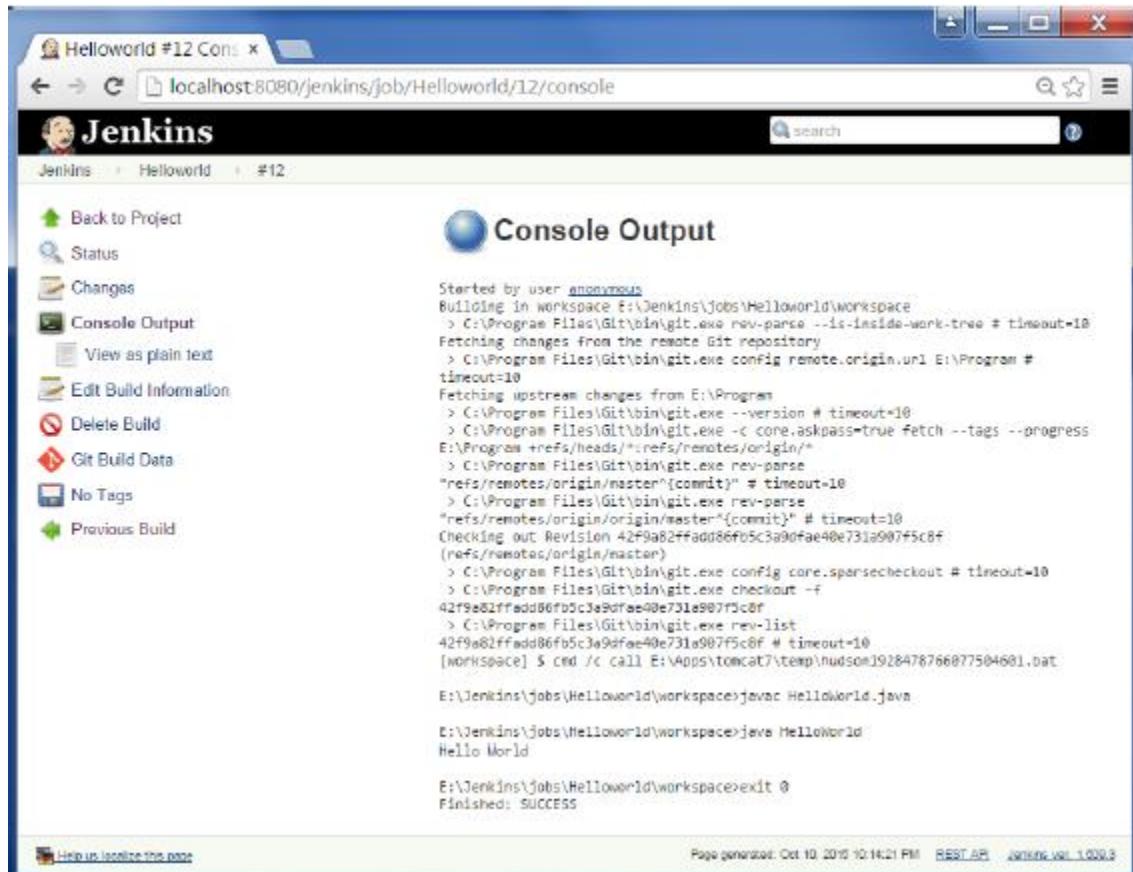
Revision: 429e82ffadd86fb5c3a9d9ae40e731a907f5c0f
• refs/remotes/origin/master

Add description

Help us localize this page Page generated: Oct 10, 2015 12:57:31 AM REST API Jenkins ver. 1.609.3

This screenshot shows the Jenkins interface for a job named 'Helloworld'. The build number is #1, which was run on October 10, 2015, at 12:52:50 AM. The build took 4.7 seconds. There were no changes made to the codebase. The build was started by an anonymous user. The revision of the code is 429e82ffadd86fb5c3a9d9ae40e731a907f5c0f, specifically the 'refs/remotes/origin/master' branch. A link to add a description is available. At the bottom, there's a note about localizing the page and links for page generation details and the Jenkins version.

Siva Jenkins Tutorial



The screenshot shows a Jenkins job named "Helloworld" with a build number of 12. The "Console Output" tab is selected. The output window displays the command-line logs of the build process. The logs show the Jenkins agent connecting to a Git repository at E:\Program Files\Git\bin\git.exe, pulling changes, and then executing Java code to print "Hello World". The build status is listed as "SUCCESS".

```
Started by user anonymous
Building in workspace F:\Jenkins\Jobs\Helloworld\workspace
> C:\Program Files\Git\bin\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url E:\Program #
timeout=10
Fetching upstream changes from E:\Program
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> C:\Program Files\Git\bin\git.exe < core.askpass=true fetch --tags --progress
E:\Program +refs/heads/* refs/remotes/origin/*
> C:\Program Files\Git\bin\git.exe rev-parse
"refs/remotes/origin/master^{commit}" # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse
"refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision 42f9a82ffadd86fb5c3a9dfae40e731a907f5c8f
(refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f
42f9a82ffadd86fb5c3a9dfae40e731a907f5c8f
> C:\Program Files\Git\bin\git.exe rev-list
42f9a82ffadd86fb5c3a9dfae40e731a907f5c8f # timeout=10
[worKspace] $ cmd /c call E:\Apps\tomcat7\temp\hudson1928478766077504601.bat

E:\Jenkins\jobs\Helloworld\workspace>javac HelloWorld.java

E:\Jenkins\jobs\Helloworld\workspace>java HelloWorld
Hello World

E:\Jenkins\jobs\Helloworld\workspace>exit 0
Finished: SUCCESS
```

Apart from the steps shown above there are just so many ways to create a build job, the options available are many, which what makes Jenkins such a fantastic continuous deployment tool.

Jenkins - Unit Testing

Jenkins provides an out of box functionality for Junit, and provides a host of plugins for unit testing for other technologies, an example being MSTest for .Net Unit tests. If you go to the link <https://wiki.jenkins-ci.org/display/JENKINS/xUnit+Plugin> it will give the list of Unit Testing plugins available.

Siva Jenkins Tutorial

xUnit Plugin - Jenkins

https://wiki.jenkins-ci.org/display/JENKINS/xUnit+Plugin

Dashboard > Jenkins > Plugins > xUnit Plugin

Browse Search

xUnit Plugin

Added by Gregory Boissinot, last edited by Gregory Boissinot on Oct 08, 2015 (view change)

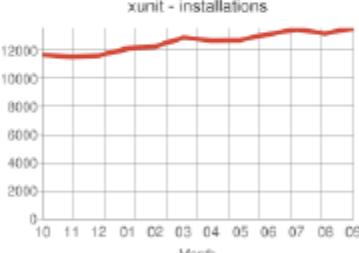
Jenkins

- Home
- Mailing lists
- Source code
- Bugtracker
- Security Advisories
- Events
- Donation
- Commercial Support
- Wiki Site Map

Documents

- Meet Jenkins
- Use Jenkins
- Extend Jenkins
- Plugins
- Servlet Container Notes

Plugin Information

Plugin ID	xunit	Changes	In Latest Release Since Latest Release
Latest Release	1.98 (archives)	Source Code	GitHub
Latest Release Date	Oct 09, 2015	Issue Tracking	Open Issues
Required Core Dependencies	1.580.1 junit (version: 1.6)	Pull Requests	Pull Requests
Usage		Installations	Since Latest Release 2014-Oct 11692 2014-Nov 11557 2014-Dec 11631 2015-Jan 12106 2015-Feb 12262 2015-Mar 12891 2015-Apr 12894 2015-May 12716 2015-Jun 13143 2015-Jul 13470 2015-Aug 13192 2015-Sep 13563

This plugin makes it possible to publish the test results of an execution of a testing tool in Jenkins.

CppUnit output

Siva Jenkins Tutorial

The screenshot shows a web browser window with the title "xUnit Plugin - Jenkins". The URL in the address bar is <https://wiki.jenkins-ci.org/display/JENKINS/xUnit+Plugin>. The page content is as follows:

Features

- Records xUnit tests
- Mark the build unstable or fail according to threshold values

Supported tools

Embedded tools

- * JUnit itself
- * ALUnit
- * MSTest (imported from MSTest Plugin)
- * NUnit (imported from NUnit Plugin)
- * UnitTest++
- * Boost Test Library
- * PHPUnit
- * Free Pascal Unit
- * CppUnit
- * MbUnit
- * GoogleTest
- * EmbUnit
- * gtester/glib
- * QTestLib

Other plugins as an extension of the xUnit plugin:

- * Gallio (Gallio plugin)
- * Parasoft C++Test tool (Cppunit Plugin)
- * JSUnit (JSUnit Plugin)
- * JBehave
- * TestComplete (TestComplete xUnit Plugin)

External contributions

Example of a Junit Test in Jenkins

The following example will consider

- A simple HelloWorldTest class based on Junit.
- Ant as the build tool within Jenkins to build the class accordingly.

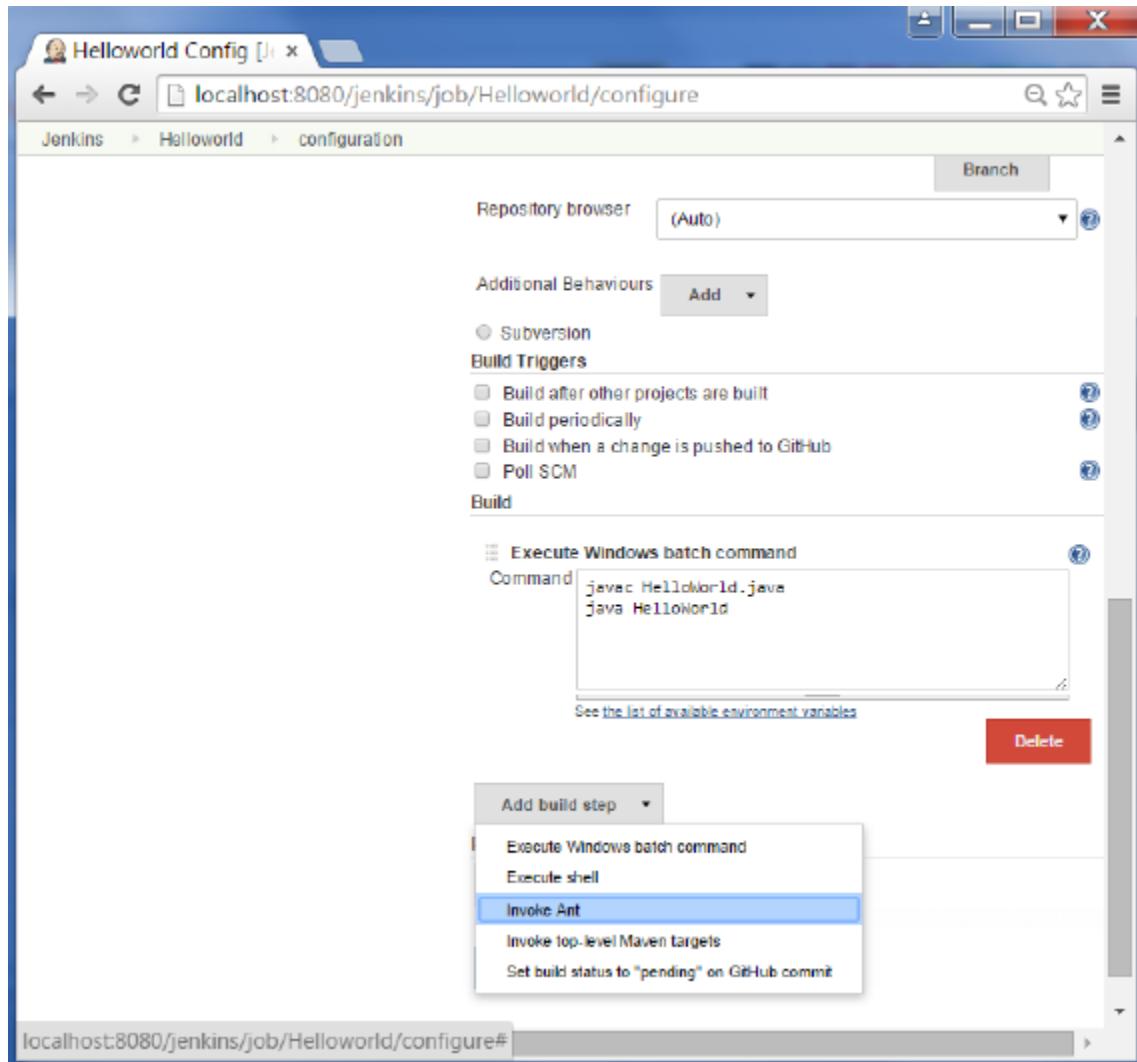
Step 1 – Go to the Jenkins dashboard and Click on the existing Hello World project and choose the Configure option

Siva Jenkins Tutorial

The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. On the left sidebar, there are links for New Item, People, Build History, Manage Jenkins, and Credentials. Below these are sections for Build Queue (No builds in the queue) and Build Executor Status (master has 1 idle and 2 idle executors; build_slave is offline). The main area displays a table of projects. One project, 'Helloworld', is selected, showing its last build information: Last Success (5 sec - #11), Last Failure (2 days 23 hr - #10), and Last Duration (3.2 sec). Below the table are links for Changes, Workspace, Build Now, Delete Project, and Configure. The 'Configure' link is highlighted with a blue background. At the bottom of the page, the URL is localhost:8080/jenkins/job/Helloworld/configure, and the page footer indicates it was generated on Oct 15, 2015 at 10:23:01 PM.

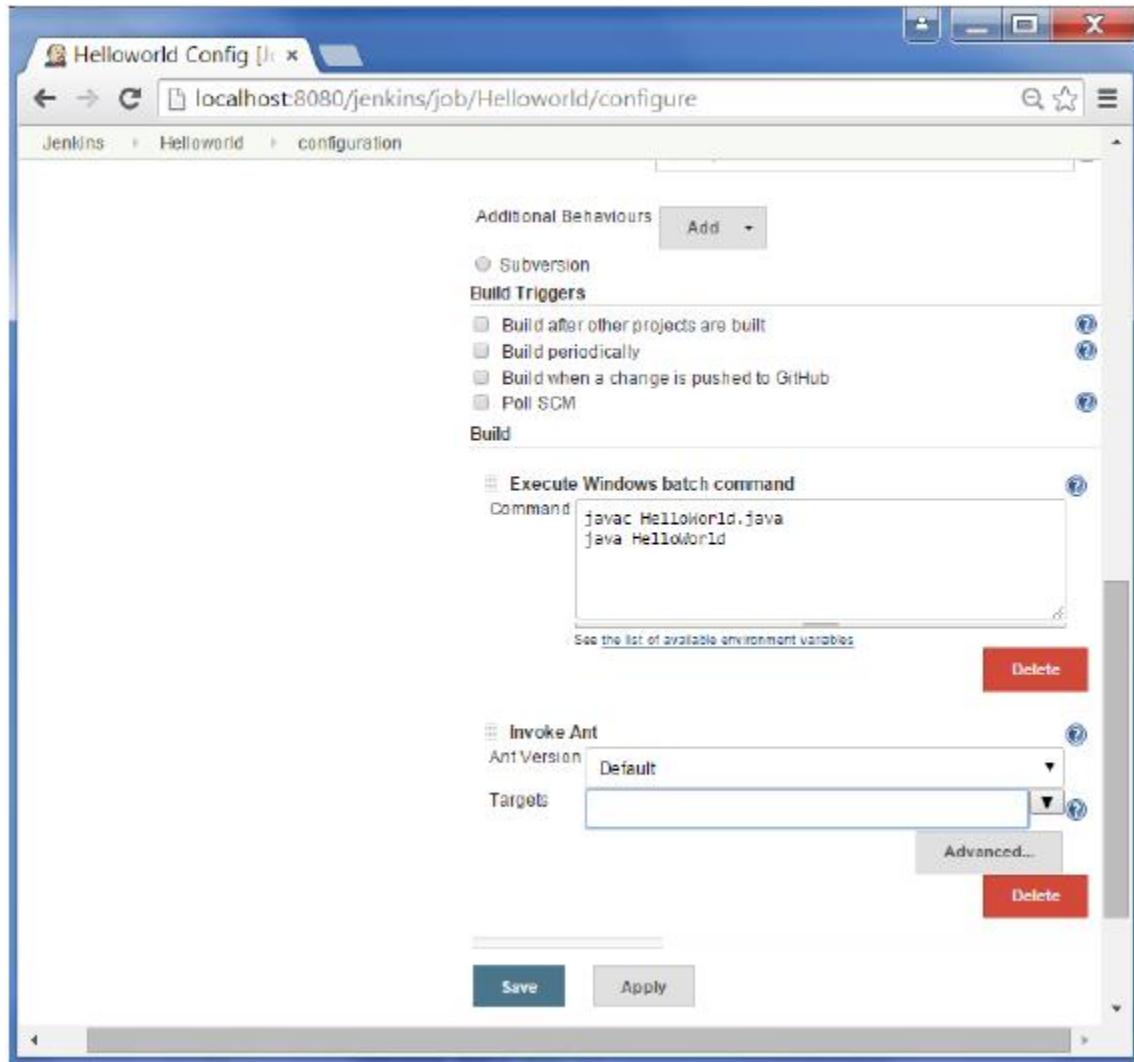
Step 2 – Browse to the section to Add a Build step and choose the option to Invoke Ant.

Siva Jenkins Tutorial



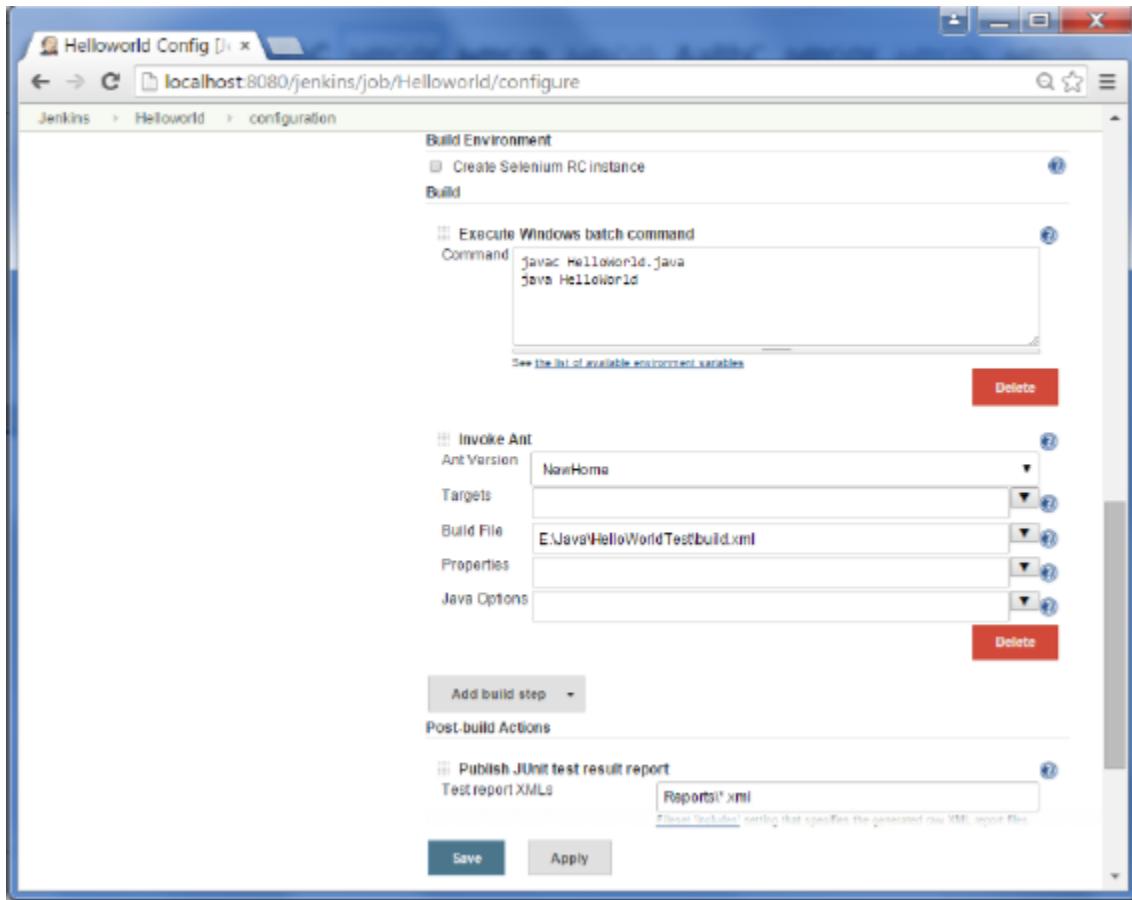
Step 3 – Click on the Advanced button.

Siva Jenkins Tutorial



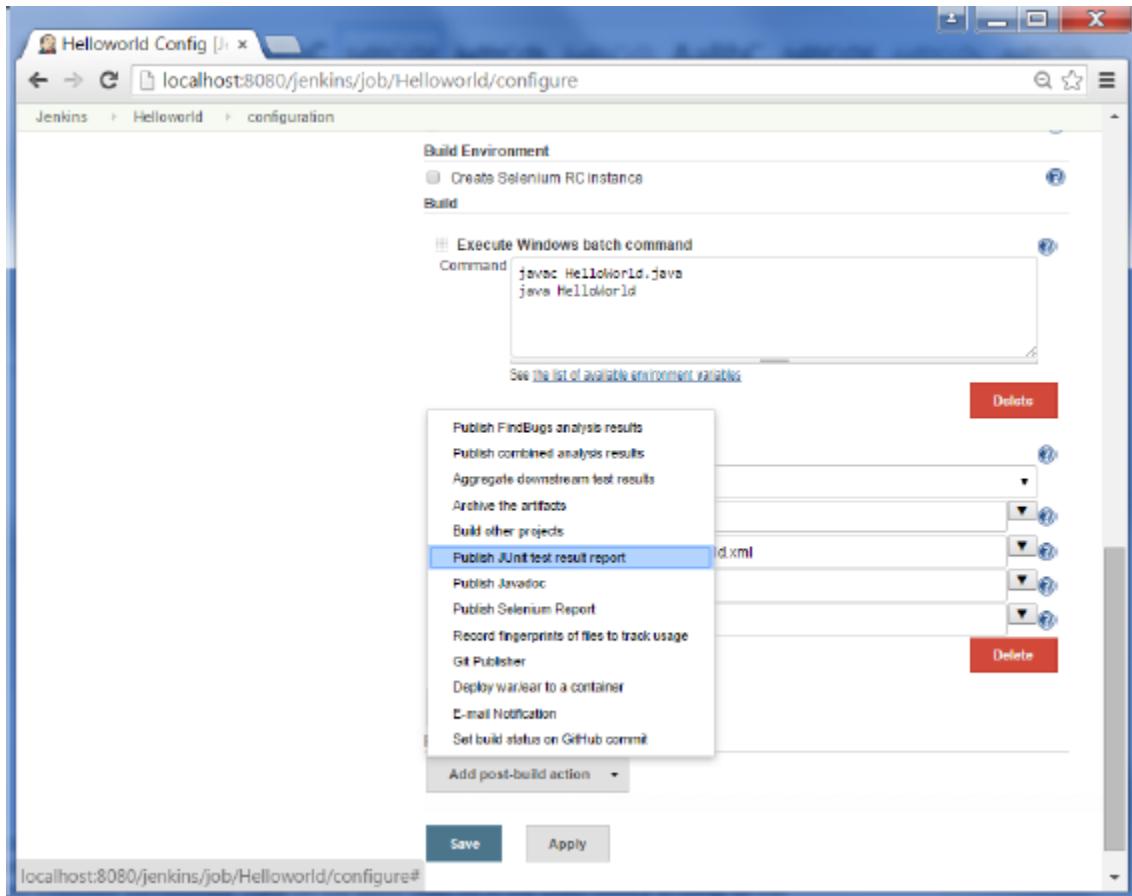
Step 4 – In the build file section, enter the location of the build.xml file.

Siva Jenkins Tutorial



Step 5 – Next click the option to Add post-build option and choose the option of “Publish Junit test result report”

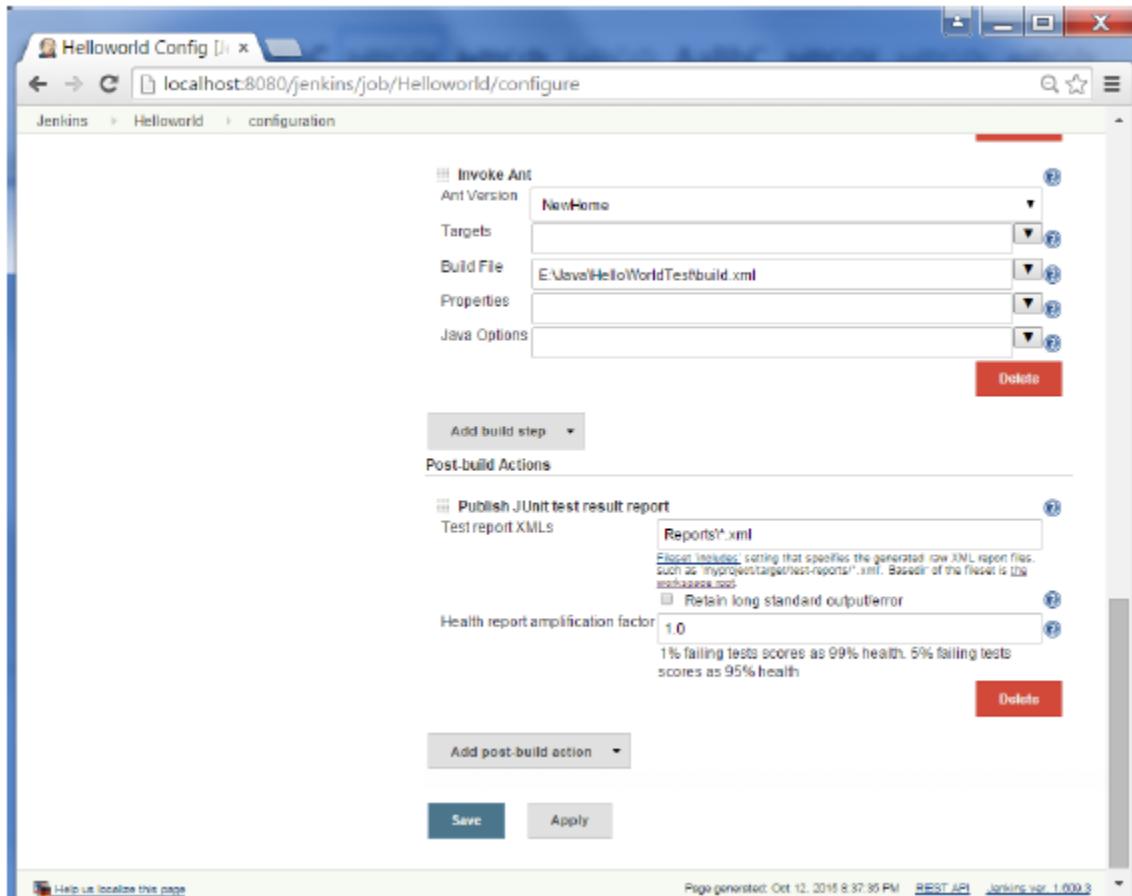
Siva Jenkins Tutorial



Step 6 – In the Test reports XML’s, enter the location as shown below. Ensure that Reports is a folder which is created in the HelloWorld project workspace. The “*.xml” basically tells Jenkins to pick up the result xml files which are produced by the running of the Junit test cases. These xml files which then be converted into reports which can be viewed later.

Once done, click the Save option at the end.

Siva Jenkins Tutorial



Step 7 – Once saved, you can click on the Build Now option.

Once the build is completed, a status of the build will show if the build was successful or not. In the Build output information, you will now notice an additional section called Test Result. In our case, we entered a negative Test case so that the result would fail just as an example.

Siva Jenkins Tutorial

The screenshot shows the Jenkins web interface for a project named "Helloworld". The main title is "Build #4 (Oct 12, 2015) 8:33:16 PM". Key details include "Started 3 days 1 hr ago" and "Took 3.9 sec on master". A sidebar on the left lists options like "Back to Project", "Status", "Changes", "Console Output", "Edit Build Information", "Delete Build", "Git Build Data", "No Tags", "Test Result", "Previous Build", and "Next Build". The "Test Result" section indicates "No changes." and "Started by anonymous user". It also shows a git revision: 42f9a82ffadd86fb5c3a@dtae40e731a807f5c8f, with a note about refs/remotes/origin/master. A link to "Test Result (1 failure)" leads to "HelloWorldTestCase.InitializationError". The bottom of the page includes links for "Help us locate this issue", "Page generated: Oct 15, 2015 10:24:38 PM", "REST API", and "Jenkins ver: 1.609.3".

One can go to the Console output to see further information. But what's more interesting is that if you click on Test Result, you will now see a drill down of the Test results.

Siva Jenkins Tutorial

The screenshot shows a Jenkins test report for a build named "Helloworld #4". The main summary indicates 1 failure. Below this, the "All Failed Tests" section lists a single failed test case: "HelloWorldTestCase InitializationError" which took 10 ms. The "All Tests" section shows a total of 1 test run across the package "root".

Test Name	Duration	Age
HelloWorldTestCase InitializationError	10 ms	1

Package	Duration	Fail	(err)	Skip	(err)	Pass	(err)	Total	(err)
[root]	10 ms	1	+1	0	(err)	0	(err)	1	+1

Jenkins - Automated Testing

One of the basic principles of Continuous Integration is that a build should be verifiable. You have to be able to objectively determine whether a particular build is ready to proceed to the next stage of the build process, and the most convenient way to do this is to use automated tests. Without proper automated testing, you find yourself having to retain many build artifacts and test them by hand, which is hardly in the spirit of Continuous Integration. The following example shows how to use Selenium to run automated web tests.

Step 1 – Go to Manage Plugins.

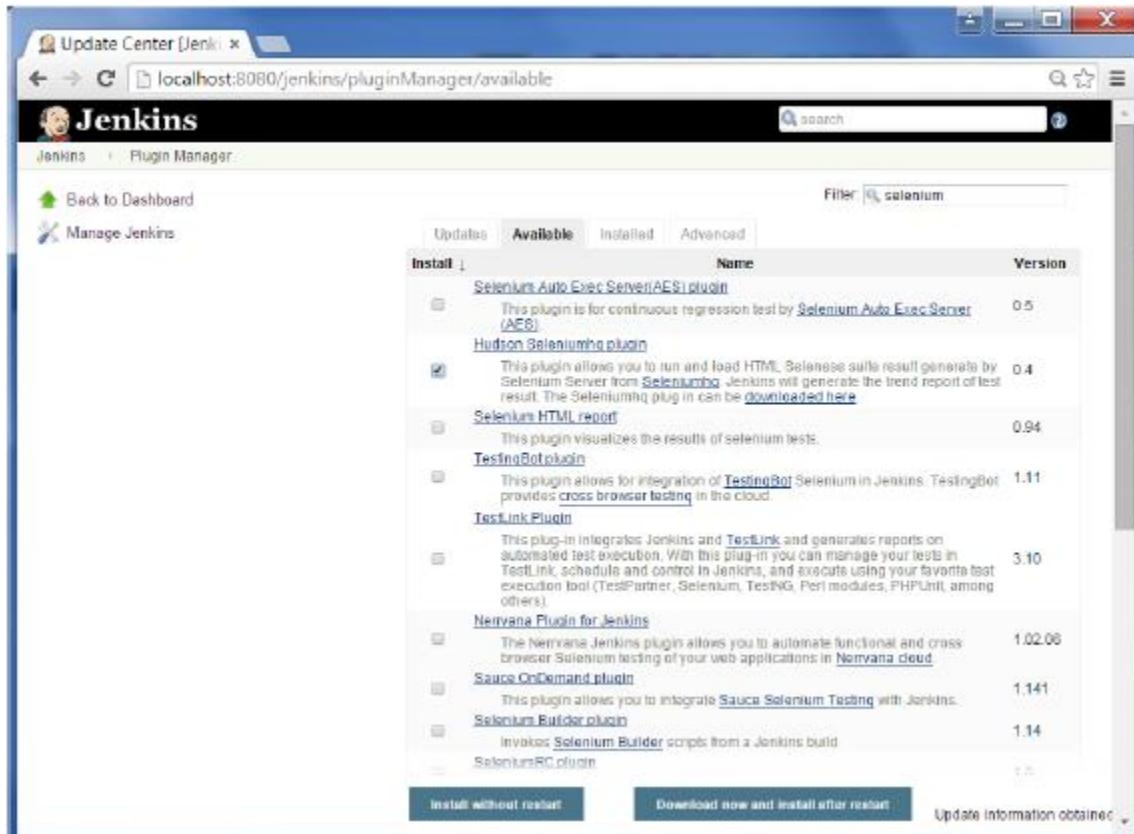
Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/jenkins/manage. The left sidebar includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Under 'Build Queue' and 'Build Executor Status', there are no builds listed. The main content area is titled 'Manage Jenkins' and contains several configuration links:

- Configure System**: Configure global settings and paths.
- Configure Global Security**: Secure Jenkins; define who is allowed to access the system.
- 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. (**Updates available**)
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from `java.util.logging` output related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI**: Access/Manage Jenkins from your shell, or from your script.
- Script Console**: Executes arbitrary script for administration/trouble-shooting/diagnostics.
- Manage Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Manage Credentials**: Create/delete/modify the credentials that can be used by Jenkins and by jobs running in Jenkins to connect to 3rd party services.
- About Jenkins**: Get the version and license information.

Step 2 – Find the Hudson Selenium Plugin and choose to install. Restart the Jenkins instance.

Siva Jenkins Tutorial



The screenshot shows the Jenkins Plugin Manager interface. The URL in the browser is `localhost:8080/jenkins/pluginManager/available`. A search bar at the top right contains the text "selenium". The main table lists several available Jenkins plugins:

Install	Name	Version
<input type="checkbox"/>	Selenium Auto Exec Server(AES) plugin This plugin is for continuous regression test by Selenium Auto Exec Server (AES) .	0.5
<input checked="" type="checkbox"/>	Hudson Seleniumhq plugin This plugin allows you to run and load HTML, Selenease suite result generates by Selenium Server from Selenease . Jenkins will generate the trend report of test result. The Seleniumhq plug-in can be downloaded here .	0.4
<input type="checkbox"/>	Selenium HTML report This plugin visualizes the results of selenium tests.	0.94
<input type="checkbox"/>	TestingBot plugin This plugin allows for integration of TestingBot Selenium in Jenkins. TestingBot provides cross browser testing in the cloud.	1.11
<input type="checkbox"/>	TestLink Plugin This plug-in integrates Jenkins and TestLink and generates reports on automated test execution. With this plug-in you can manage your tests in TestLink, schedule and control in Jenkins, and execute using your favorite test execution tool (TestPartner, Selenium, TestNG, Perl modules, PHPUnit, among others).	3.10
<input type="checkbox"/>	Nemvana Plugin for Jenkins The Nemvana Jenkins plugin allows you to automate functional and cross browser Selenium testing of your web applications in Nemvana cloud .	1.02.06
<input type="checkbox"/>	Sauce OnDemand plugin This plugin allows you to integrate Sauce Selenium Testing with Jenkins.	1.141
<input type="checkbox"/>	Selenium Builder plugin Invokes Selenium Builder scripts from a Jenkins build.	1.14
<input type="checkbox"/>	SeleniumRC plugin	1.2

At the bottom of the page are three buttons: "Install without restart", "Download now and install after restart", and "Update information obtained".

Step 3 – Go to Configure system.

Siva Jenkins Tutorial

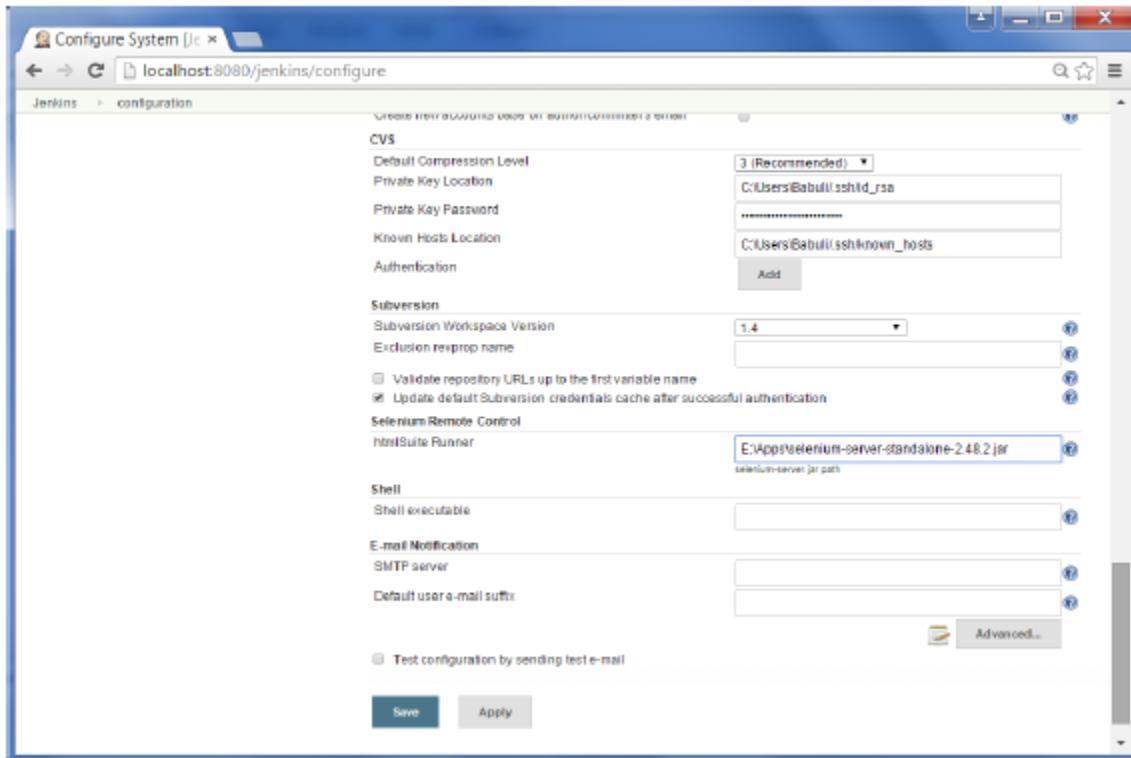
The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/jenkins/manage. The left sidebar includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is selected), and 'Credentials'. Under 'Manage Jenkins', there are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). The main content area is titled 'Manage Jenkins' and contains several configuration links:

- Configure System**: Configure global settings and paths.
- Configure Global Security**: Secure Jenkins; define who is allowed to access the system.
- 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. (**Updates available**)
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from `java.util.logging` related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI**: Access/Manage Jenkins from your shell, or from your script.
- Script Console**: Executes arbitrary script for administration/trouble-shooting/diagnostics.
- Manage Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

At the top right, there are 'Setup Security' and 'Dismiss' buttons. A search bar and an 'ENABLE AUTO REFRESH' checkbox are also present.

Step 4 – Configure the selenium server jar and click on the Save button.

Siva Jenkins Tutorial



Note – the selenium jar file can be downloaded from the location SeleniumHQ

Click on the download for the Selenium standalone server.

Siva Jenkins Tutorial

The screenshot shows a web browser window with the URL www.seleniumhq.org/download/. The page is titled "SeleniumHQ Browser Automation". The main content area is titled "Downloads". It contains sections for "Selenium Downloads", "Selenium Standalone Server", "The Internet Explorer Driver Server", and "Selenium Client & WebDriver Language Bindings". On the left sidebar, there are links for "Selenium Downloads", "Latest Releases", "Previous Releases", "Source Code", "Maven Information", "Donate to Selenium" (with a "Donate" button and payment method icons), and "Selenium Sponsors". A "BrowserStack" logo is also present. A search bar at the top right says "search selenium:".

Selenium Downloads

Below is where you can find the latest releases of all the Selenium components. You can also find a list of [previous releases](#), [source code](#), and additional information for [Maven users](#) (Maven is a popular Java build tool).

Selenium Standalone Server

The Selenium Server is needed in order to run either Selenium RC style scripts or Remote Selenium WebDriver ones. The 2.x server is a drop-in replacement for the old Selenium RC server and is designed to be backwards compatible with your existing infrastructure.

Download version [2.48.2](#)

To use the Selenium Server in a Grid configuration [see the wiki page](#).

The Internet Explorer Driver Server

This is required if you want to make use of the latest and greatest features of the WebDriver InternetExplorerDriver. Please make sure that this is available on your \$PATH (or %PATH% on Windows) in order for the IE Driver to work as expected.

Download version 2.48.0 for (recommended) [32 bit Windows IE](#) or [64 bit Windows IE](#)
[CHangelog](#)

Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for [other languages exist](#), these are the core ones that are supported by the main project hosted on [google code](#).

Step 5 – Go back to your dashboard and click on the Configure option for the Hello World project.

Siva Jenkins Tutorial

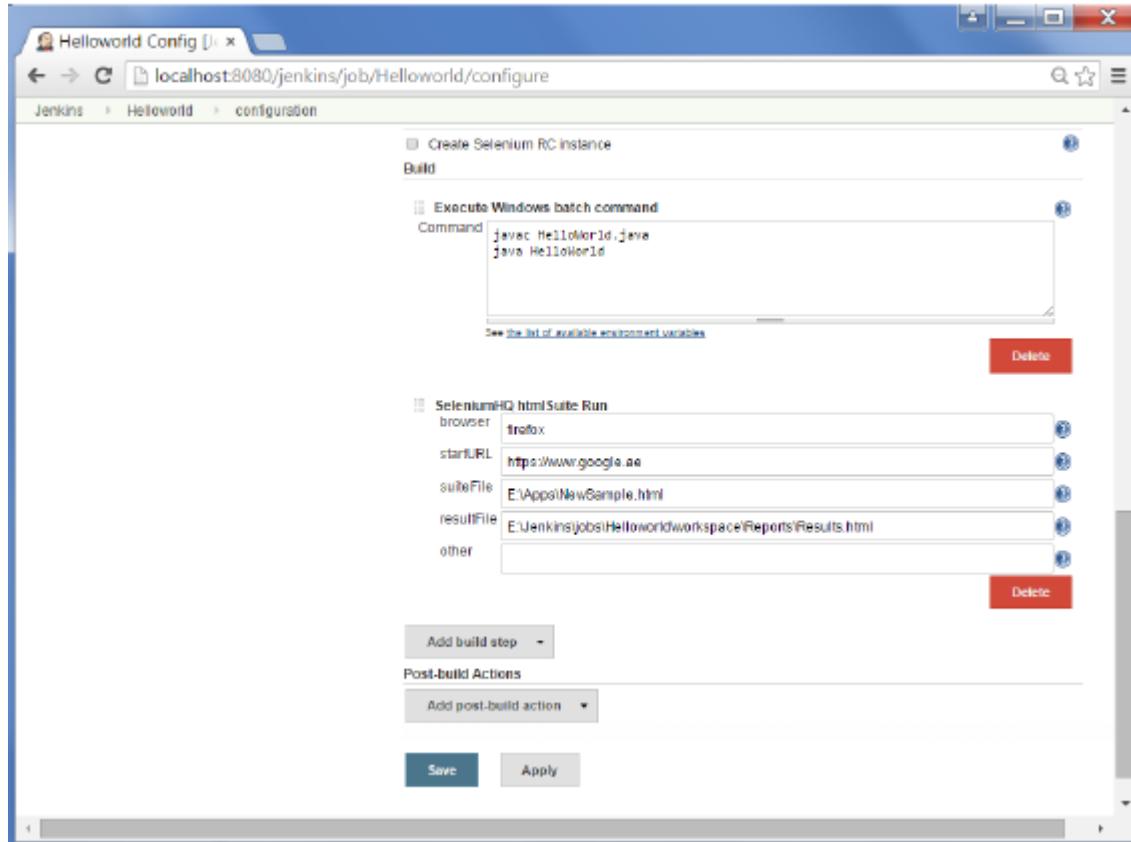
The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. The main area displays a table of jobs. One job, 'Helloworld', is highlighted. Its details are shown in a modal: 'Icon: HML', 'Name: Helloworld', 'Last Success: 23 hr - #12', 'Last Failure: 23 hr - #10', and 'Last Duration: 3.7 sec'. Below the table are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 idle, 2 idle). At the bottom, the URL is localhost:8080/jenkins/job/Helloworld/configure.

Step 6 – Click on Add build step and choose the optin of “SeleniumHQ htmlSuite Run”

The screenshot shows the configuration page for the 'Helloworld' job at localhost:8080/jenkins/job/Helloworld/configure. The 'Build' section contains a 'Execute Windows batch command' step with the command: `javac HelloWorld.java
java HelloWorld`. Below this, a dropdown menu titled 'Add build step' is open, showing options like 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke top-level Maven targets', 'SeleniumHQ htmlSuite Run' (which is highlighted in blue), and 'Set build status to "pending" on GitHub commit'.

Siva Jenkins Tutorial

Step 7 – Add the necessary details for the selenium test. Here the suite File is the Test Suite generated by using the Selenium IDE. Click on Save and execute a build. Now the post build will launch the selenium driver, and execute the html test.

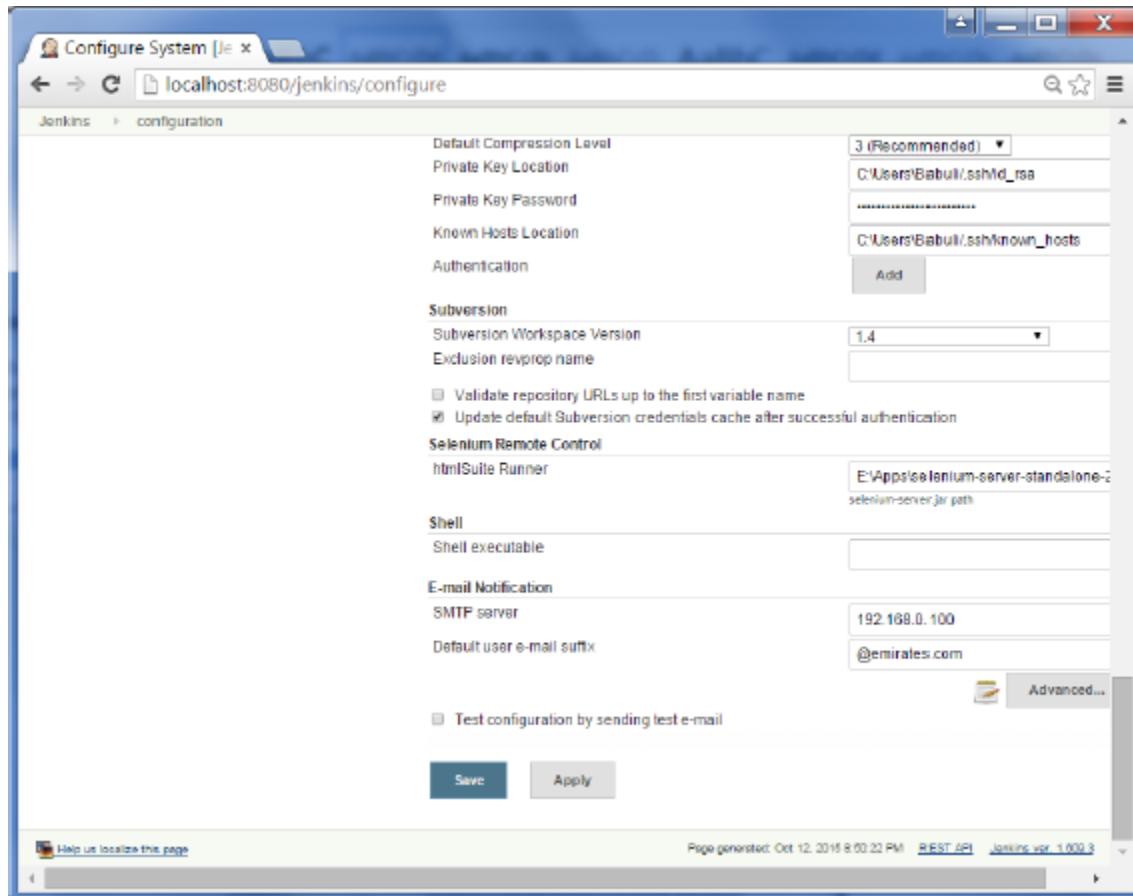


Jenkins - Notification

Jenkins comes with an out of box facility to add an email notification for a build project.

Step 1 – Configuring an SMTP server. Goto Manage Jenkins → Configure System. Go to the E-mail notification section and enter the required SMTP server and user email-suffix details.

Siva Jenkins Tutorial



Step 2 – Configure the recipients in the Jenkins project - When you configure any Jenkins build project, right at the end is the ability to add recipients who would get email notifications for unstable or broken builds. Then click on the Save button.

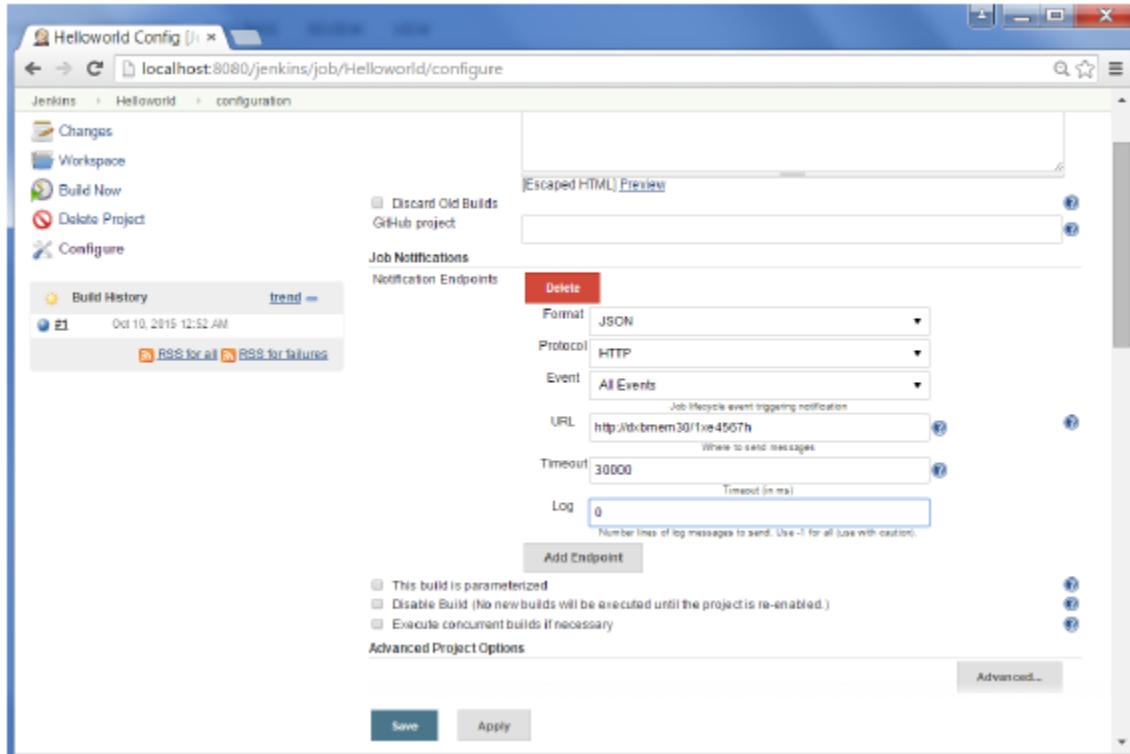
Siva Jenkins Tutorial

The screenshot shows the Jenkins configuration page for the 'Helloworld' job. The top navigation bar includes links for 'Jenkins', 'Helloworld', and 'configuration'. The main content area is titled 'Helloworld Config' and shows the following configuration steps:

- Build Steps:** A section titled 'SeleniumHQ html Suite Run' contains fields for 'browser' (set to 'ieexplorer'), 'startURL' ('https://www.google.es'), 'suiteFile' ('E:\Apps\Selenium\Sample.html'), 'resultFile' ('Result.html'), and 'other' (empty). There are 'Delete' buttons for each row.
- Post-build Actions:** An 'E-mail Notification' step is configured with the recipient 'S112233@domain.com'. It includes options for sending e-mail for every unstable build or for individuals who broke the build. There is also a 'Delete' button for this action.
- Buttons at the bottom:** 'Save' and 'Apply' buttons are located at the bottom left, and a 'Delete' button is located at the bottom right.

Apart from the default, there are also notification plugin's available in the market. An example is the notification plugin from Tikal Knowledge which allows sending Job Status notifications in JSON and XML formats. This plugin enables end-points to be configured as shown below.

Siva Jenkins Tutorial



Here are the details of each option –

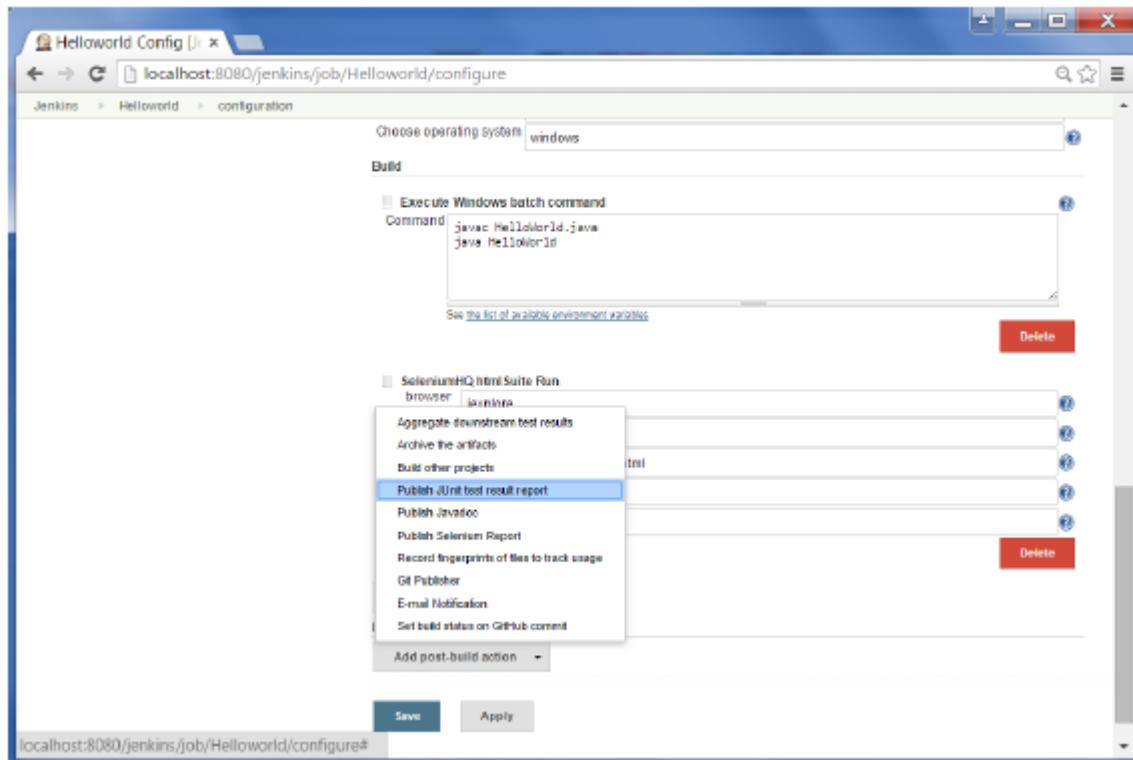
- **"Format"** – This is the notification payload format which can either be JSON or XML.
- **"Protocol"** – protocol to use for sending notification messages, HTTP, TCP or UDP.
- **"Event"** – The job events that trigger notifications: Job Started, Job Completed, Job Finalized or All Events (the default option).
- **"URL"** – URL to send notifications to. It takes the form of "http://host" for HTTP protocol, and "host: port" for TCP and UDP protocols.
- **"Timeout"** – Timeout in milliseconds for sending notification request, 30 seconds by default.

Jenkins – Reporting

As demonstrated in the earlier section, there are many reporting plugins available with the simplest one being the reports available for jUnit tests.

Siva Jenkins Tutorial

In the Post-build action for any job, you can define the reports to be created. After the builds are complete, the Test Results option will be available for further drill-down.



Jenkins - Code Analysis

Jenkins has a host of Code Analysis plugin. The various plugins can be found at <https://wiki.jenkins-ci.org/display/JENKINS/Static+Code+Analysis+Plugins>

Siva Jenkins Tutorial

The screenshot shows the Jenkins Static Code Analysis Plug-ins page. On the left, there's a sidebar with links like Home, Mailing lists, Source code, Bugtracker, Security, Advisories, Events, Donation, Commercial Support, Wiki Site Map, and Documents. Under Documents, it lists Meet Jenkins, Use Jenkins, Extend Jenkins, Plugins, Servlet Container, and Notes. The main content area has a title "Static Code Analysis Plug-ins" with a sub-section "analysis-core". It shows the "Plugin Information" table with details like Plugin ID (analysis-core), Latest Release (1.74), and Dependencies (art, token-magic, maven-plugin, matrix-project, dashboard-view). It also shows a "Usage" chart titled "analysis-core - installations" showing a steady increase from 2010 to 2015. A "Changes" section on the right lists GitHub, Open Issues, Pull Requests, and Maintainer(s) (Ulli Hafner). A "Installations" section lists dates from 2014-Oct to 2015-Sep.

This plugin provides utilities for the static code analysis plugins. Jenkins can parse the results file from various Code Analysis tools such as Check Style, Find Bugs, PMD etc. For each corresponding code analysis tool, a plugin in Jenkins needs to be installed.

Additionally the add-on plugin Static Analysis Collector is available that combines the individual results of these plugins into a single trend graph and view.

The plugins can provide information such as

- The total number of warnings in a job
- A showing of the new and fixed warnings of a build
- Trend Reports showing the number of warnings per build
- Overview of the found warnings per module, package, category, or type
- Detailed reports of the found warnings optionally filtered by severity (or new and fixed)

Jenkins - Distributed Builds

Sometimes many build machines are required if there are instances wherein there are a larger and heavier projects which get built on a regular basis. And running all of these builds on a central machine may not be the best option. In such a scenario, one can configure other Jenkins machines to be slave machines to take the load off the master Jenkins server.

Sometimes you might also need several different environments to test your builds. In this case using a slave to represent each of your required environments is almost a must.

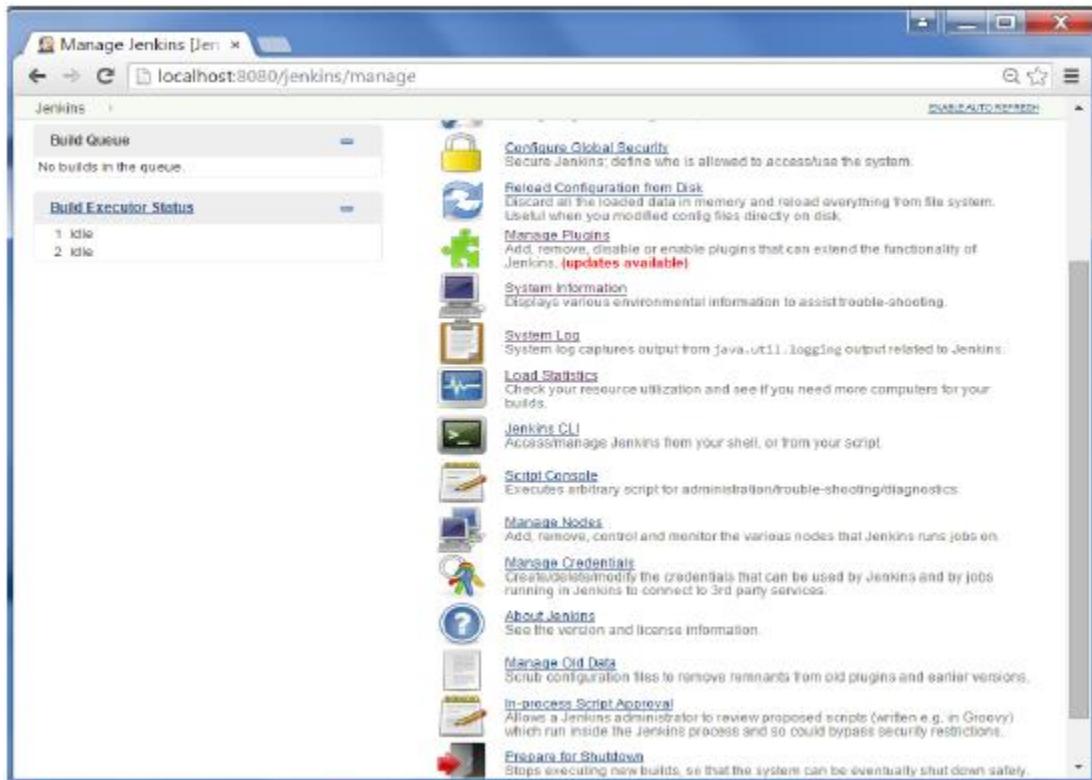
A slave is a computer that is set up to offload build projects from the master and once setup this distribution of tasks is fairly automatic. The exact delegation behavior depends on the configuration of each project; some projects may choose to "stick" to a particular machine for a build, while others may choose to roam freely between slaves.

Since each slave runs a separate program called a "slave agent" there is no need to install the full Jenkins (package or compiled binaries) on a slave. There are various ways to start slave agents, but in the end the slave agent and Jenkins master needs to establish a bi-directional communication link (for example a TCP/IP socket.) in order to operate.

To set up slaves/nodes in Jenkins follow the steps given below.

Step 1 – Go to the Manage Jenkins section and scroll down to the section of Manage Nodes.

Siva Jenkins Tutorial



Step 2 – Click on New Node

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Nodes' page. At the top, there's a navigation bar with icons for Back to Dashboard, Manage Jenkins, New Node, and Configure. Below this is a search bar and a 'ENABLE AUTO REFRESH' button. The main content area has a table titled 'Nodes' with one row. The table columns are S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, and Free. The single row shows 'master' as the name, 'Windows 7 (x86)' as the architecture, 'In sync' as the clock difference, '229.89 GB' as free disk space, '12.13 GB' as free swap space, and '3 min 12 sec' for both clock difference and free. Below the table is a 'Refresh status' button. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). At the bottom, there are links for Help us localize this page, Page generated: Oct 12, 2015 9:22:44 PM, REST API, and Jenkins ver. 1.609.2.

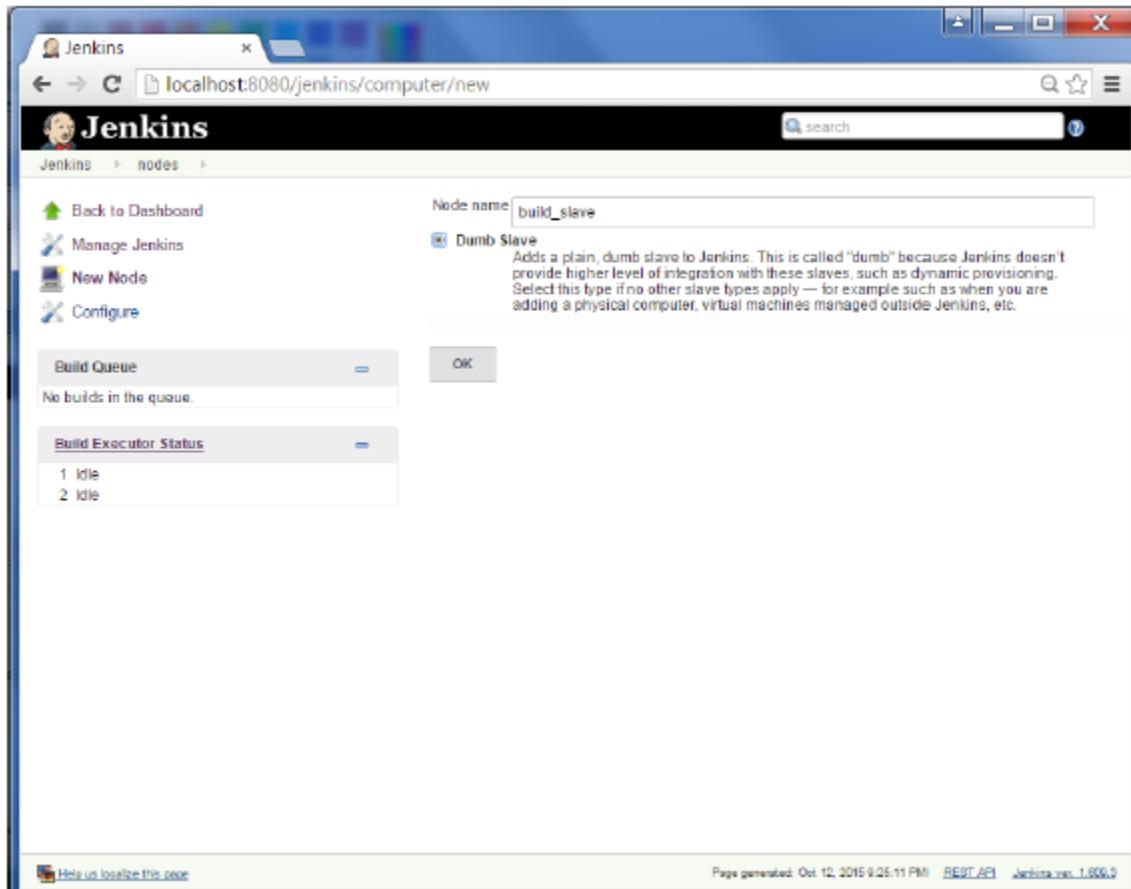
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free
	master	Windows 7 (x86)	In sync	229.89 GB	12.13 GB	3 min 12 sec

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle.

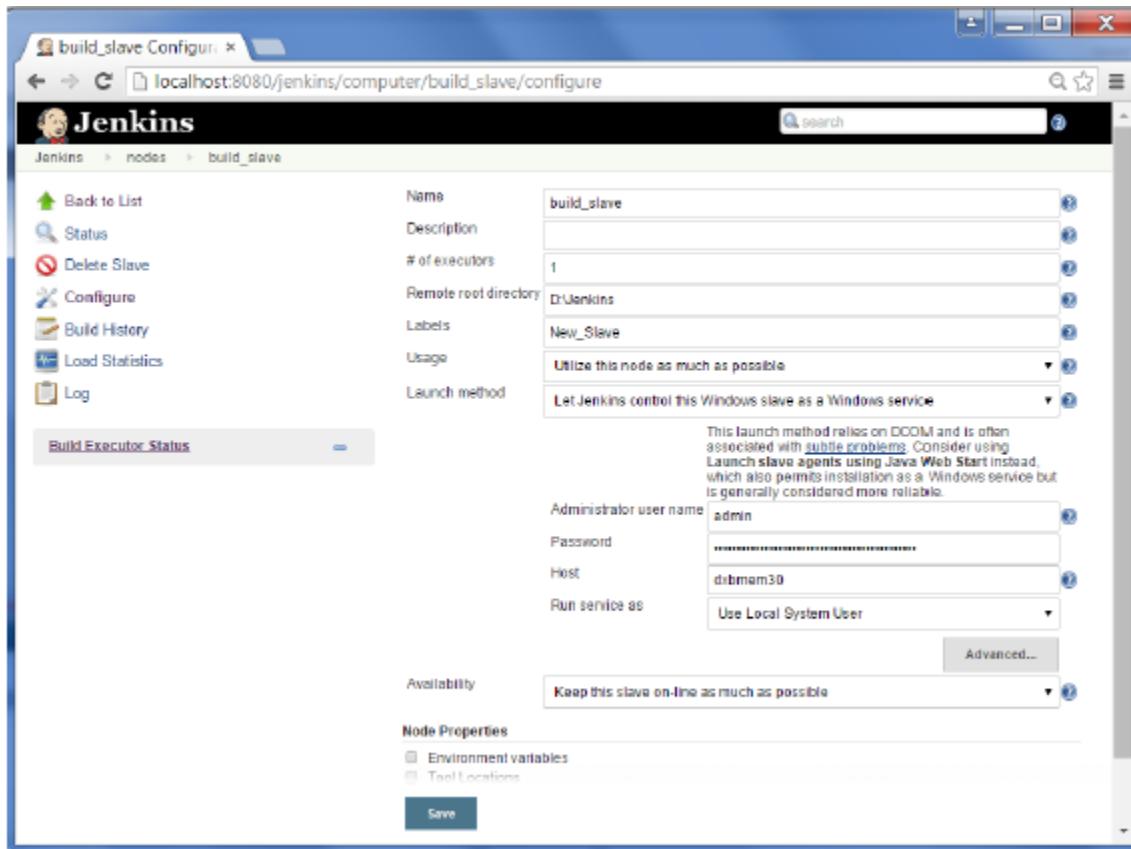
Step 3 – Give a name for the node, choose the Dumb slave option and click on Ok.

Siva Jenkins Tutorial



Step 4 – Enter the details of the node slave machine. In the below example, we are considering the slave machine to be a windows machine, hence the option of “Let Jenkins control this Windows slave as a Windows service” was chosen as the launch method. We also need to add the necessary details of the slave node such as the node name and the login credentials for the node machine. Click the Save button. The Labels for which the name is entered as “New_Slave” is what can be used to configure jobs to use this slave machine.

Siva Jenkins Tutorial



Once the above steps are completed, the new node machine will initially be in an offline state, but will come online if all the settings in the previous screen were entered correctly. One can at any time make the node slave machine as offline if required.

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with links like 'Back to Dashboard', 'Manage Jenkins', 'New Node', and 'Configure'. Below that are sections for 'Build Queue' (empty) and 'Build Executor Status'. The main area displays a table of nodes:

\$	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp S...
	build_slave		N/A	N/A	N/A	
	master	Windows 7 (x86)	In sync	229.89 GB	12.13 GB	229.8...
		Data obtained	3 ms	2 ms	1 ms	11 min

A blue button labeled 'Refresh status' is at the bottom right of the table. At the very bottom of the page, there are links for 'Help us localize this page', 'Page generated: Oct 12, 2016 9:31:43 PM', 'REST API', and 'Jenkins ver. 1.600.3'.

Sometimes many build machines are required if there are instances wherein there are a larger and heavier projects which get built on a regular basis. And running all of these builds on a central machine may not be the best option. In such a scenario, one can configure other Jenkins machines to be slave machines to take the load off the master Jenkins server.

Sometimes you might also need several different environments to test your builds. In this case using a slave to represent each of your required environments is almost a must.

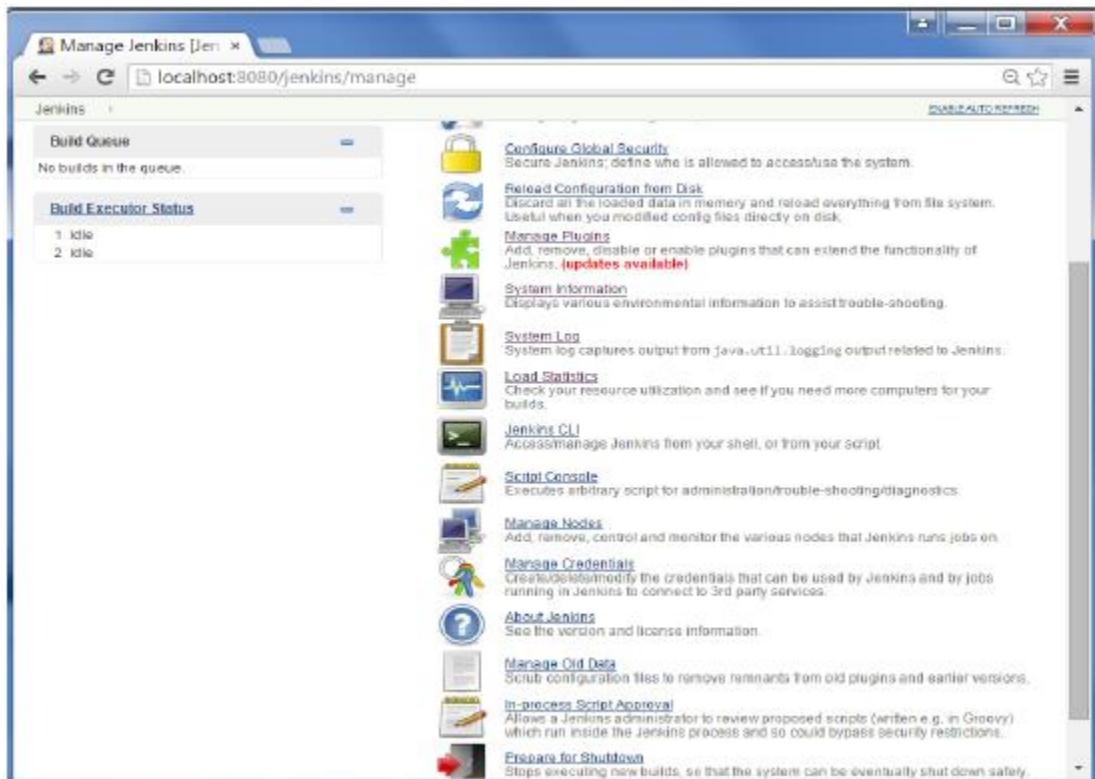
A slave is a computer that is set up to offload build projects from the master and once setup this distribution of tasks is fairly automatic. The exact delegation behavior depends on the configuration of each project; some projects may choose to "stick" to a particular machine for a build, while others may choose to roam freely between slaves.

Since each slave runs a separate program called a "slave agent" there is no need to install the full Jenkins (package or compiled binaries) on a slave. There are various ways to start slave agents, but in the end the slave agent and Jenkins master needs to establish a bi-directional communication link (for example a TCP/IP socket.) in order to operate.

Siva Jenkins Tutorial

To set up slaves/nodes in Jenkins follow the steps given below.

Step 1 – Go to the Manage Jenkins section and scroll down to the section of Manage Nodes.



Step 2 – Click on New Node

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Nodes' page. At the top, there's a navigation bar with icons for Back to Dashboard, Manage Jenkins, New Node, and Configure. Below this is a search bar and a 'ENABLE AUTO REFRESH' button. The main content area has a table titled 'Nodes' with one row. The table columns are S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, and Free. The single row shows 'master' as the name, 'Windows 7 (x86)' as the architecture, 'In sync' as the clock difference, '229.89 GB' as free disk space, '12.13 GB' as free swap space, and '3 min 12 sec' for both clock difference and free. Below the table is a 'Refresh status' button. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). At the bottom, there are links for Help us localize this page, Page generated: Oct 12, 2015 9:22:44 PM, REST API, and Jenkins ver. 1.609.2.

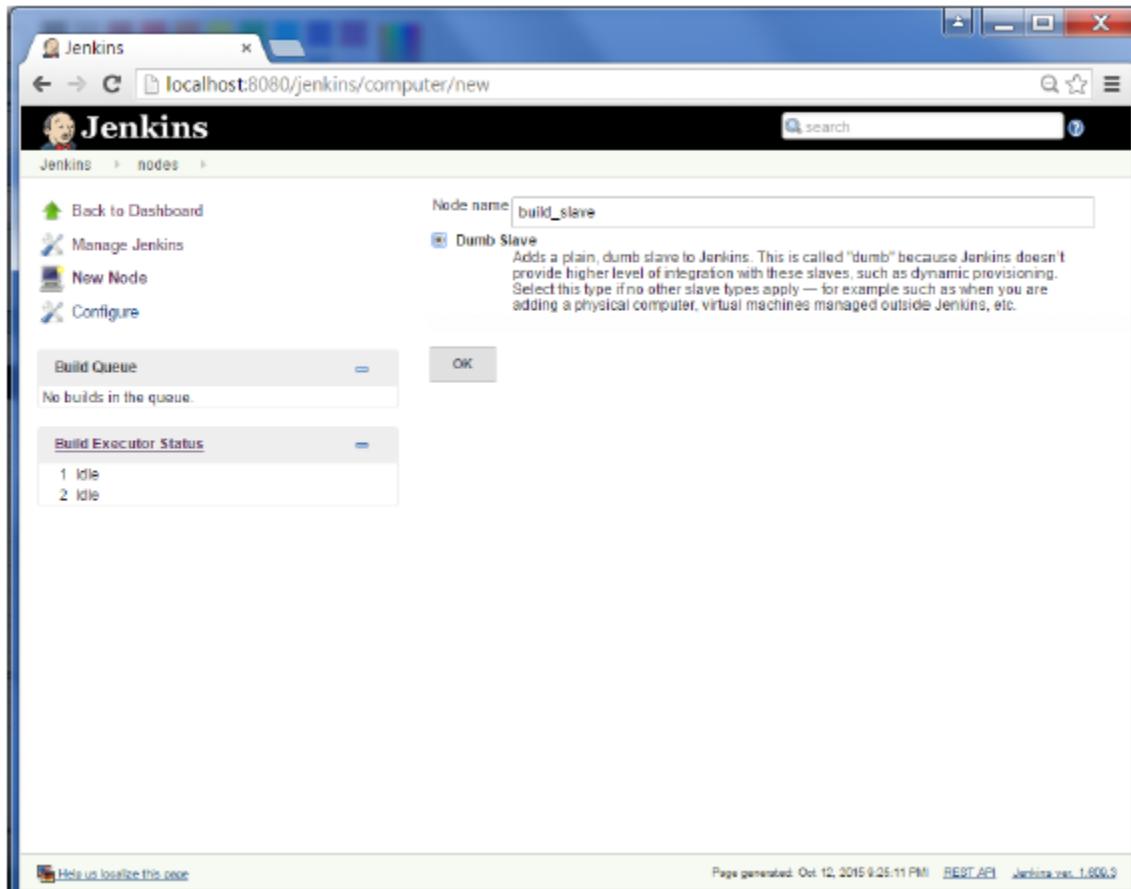
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free
	master	Windows 7 (x86)	In sync	229.89 GB	12.13 GB	3 min 12 sec

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle.

Step 3 – Give a name for the node, choose the Dumb slave option and click on Ok.

Siva Jenkins Tutorial



Step 4 – Enter the details of the node slave machine. In the below example, we are considering the slave machine to be a windows machine, hence the option of “Let Jenkins control this Windows slave as a Windows service” was chosen as the launch method. We also need to add the necessary details of the slave node such as the node name and the login credentials for the node machine. Click the Save button. The Labels for which the name is entered as “New Slave” is what can be used to configure jobs to use this slave machine.

Siva Jenkins Tutorial

The screenshot shows the Jenkins configuration interface for a slave node named 'build_slave'. The left sidebar includes links for Back to List, Status, Delete Slave, Configure, Build History, Load Statistics, and Log. The main form fields are:

- Name: build_slave
- Description: (empty)
- # of executors: 1
- Remote root directory: D:\Jenkins
- Labels: New_Slave
- Usage: Utilize this node as much as possible
- Launch method: Let Jenkins control this Windows slave as a Windows service

A note below the launch method states: "This launch method relies on DCOM and is often associated with subtle problems. Consider using Launch slave agents using Java Web Start instead, which also permits installation as a Windows service but is generally considered more reliable."

Administrator user name: admin

Host: dbmem30

Run service as: Use Local System User

Availability: Keep this slave on-line as much as possible

Node Properties sections: Environment variables, Tool Locations

Save button at the bottom.

Once the above steps are completed, the new node machine will initially be in an offline state, but will come online if all the settings in the previous screen were entered correctly. One can at any time make the node slave machine as offline if required.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for managing nodes. On the left, there's a sidebar with links like 'Back to Dashboard', 'Manage Jenkins', 'New Node', and 'Configure'. Below that are sections for 'Build Queue' (empty) and 'Build Executor Status' (showing 'master' with 1 idle and 2 idle executors, and 'build_slave' which is offline). The main area displays a table of nodes:

\$	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp S
	build_slave		N/A	N/A	N/A	
	master	Windows 7 (x86)	In sync	229.89 GB	12.13 GB	229.89 GB

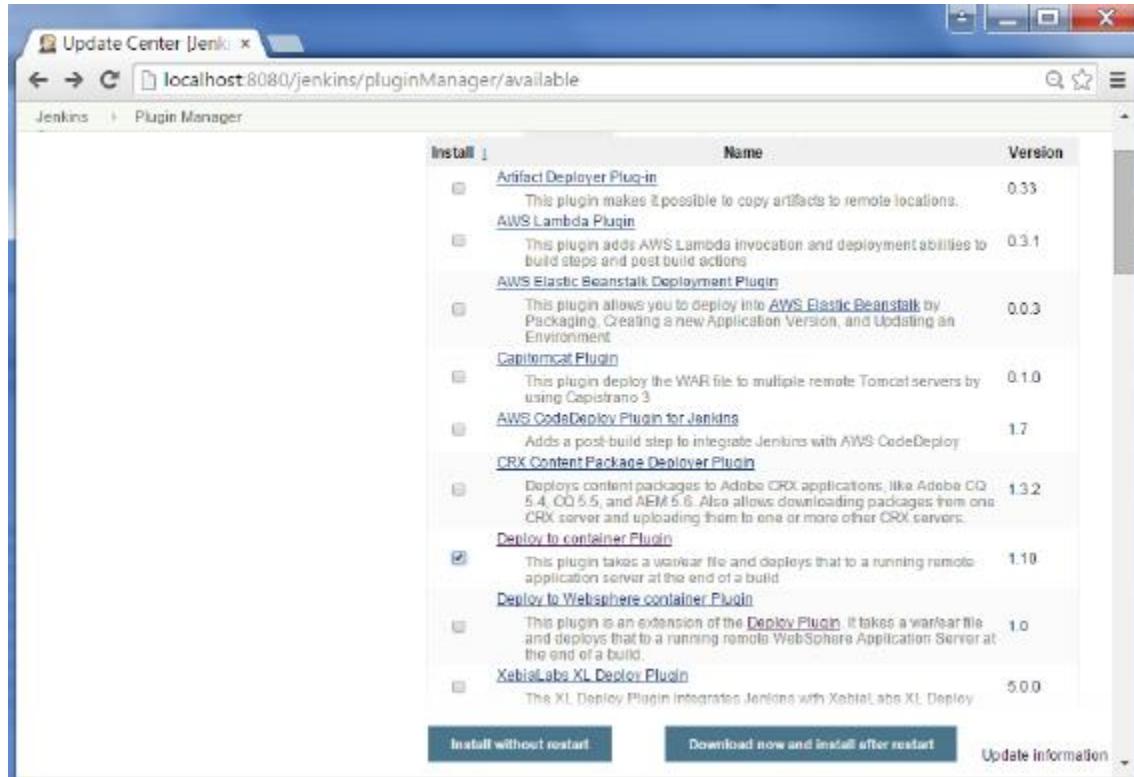
At the bottom, there's a 'Refresh status' button and some footer text.

Jenkins - Automated Deployment

There are many plugins available which can be used to transfer the build files after a successful build to the respective application/web server. One example is the “Deploy to container Plugin”. To use this follow the steps given below.

Step 1 – Go to Manage Jenkins → Manage Plugins. Go to the Available section and find the plugin “Deploy to container Plugin” and install the plugin. Restart the Jenkins server.

Siva Jenkins Tutorial



The screenshot shows the Jenkins Plugin Manager interface. The title bar says "Update Center [Jenkins]". The address bar shows "localhost:8080/jenkins/pluginManager/available". The left sidebar has "Jenkins" and "Plugin Manager" options. The main area is titled "Install 1" and lists several plugins. One plugin, "Deploy to container Plugin", is highlighted with a blue selection bar at the bottom. Other listed plugins include "Artifact Deployer Plug-in", "AWS Lambda Plugin", "AWS Elastic Beanstalk Deployment Plugin", "Capiflycat Plugin", "AWS CodeDeploy Plugin for Jenkins", "CRX Content Package Deployer Plugin", and "Xebialabs XL Deploy Plugin". At the bottom of the list are three buttons: "Install without restart", "Download now and install after restart", and "Update information".

This plugin takes a war/ear file and deploys that to a running remote application server at the end of a build.

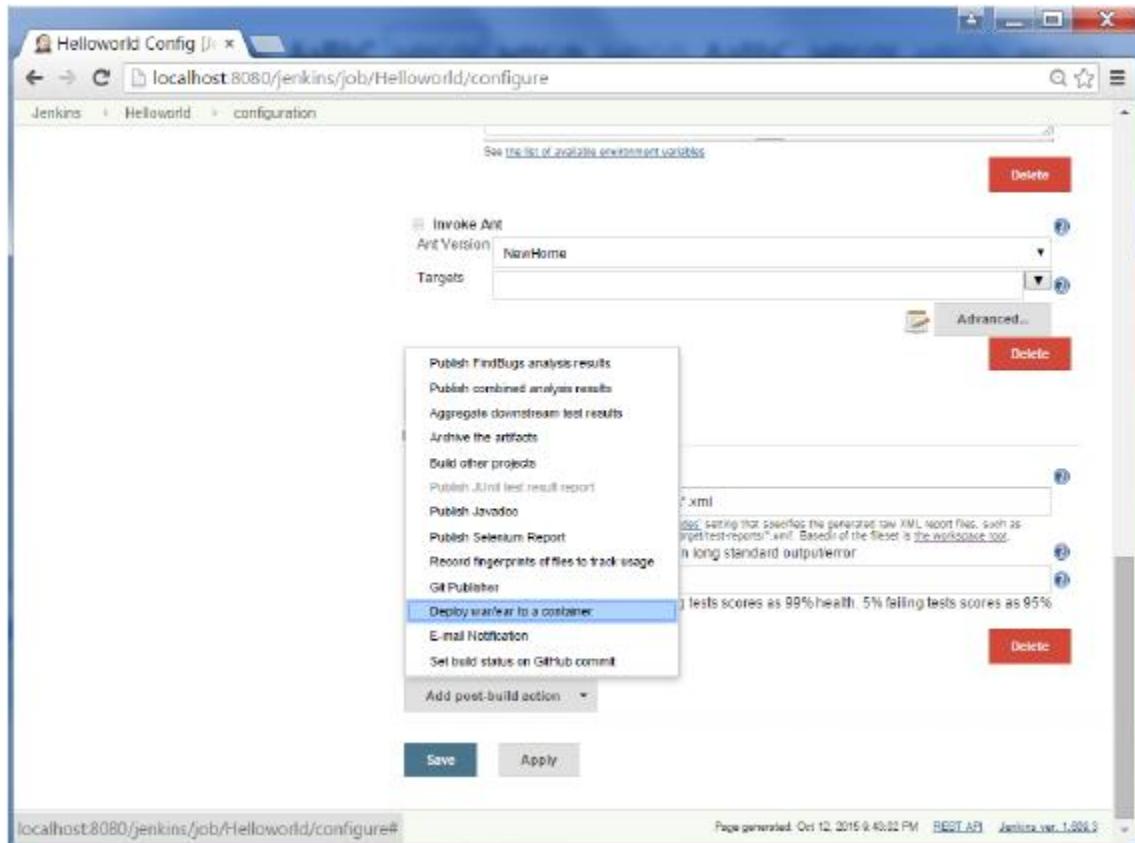
Tomcat 4.x/5.x/6.x/7.x

JBoss 3.x/4.x

Glassfish 2.x/3.x

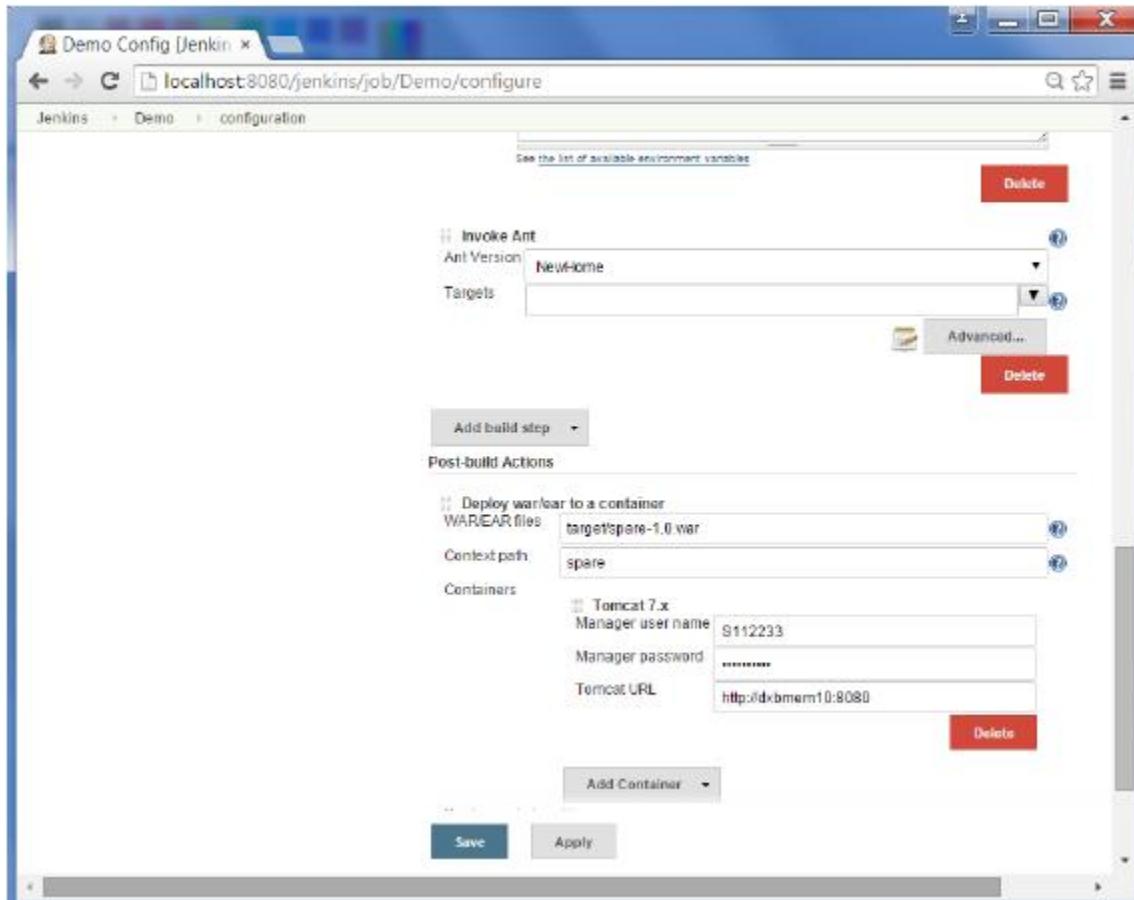
Step 2 – Go to your Build project and click the Configure option. Choose the option “Deploy war/ear to a container”

Siva Jenkins Tutorial



Step 3 – In the Deploy war/ear to a container section, enter the required details of the server on which the files need to be deployed and click on the Save button. These steps will now ensure that the necessary files get deployed to the necessary container after a successful build.

Siva Jenkins Tutorial



Jenkins - Metrics & Trends

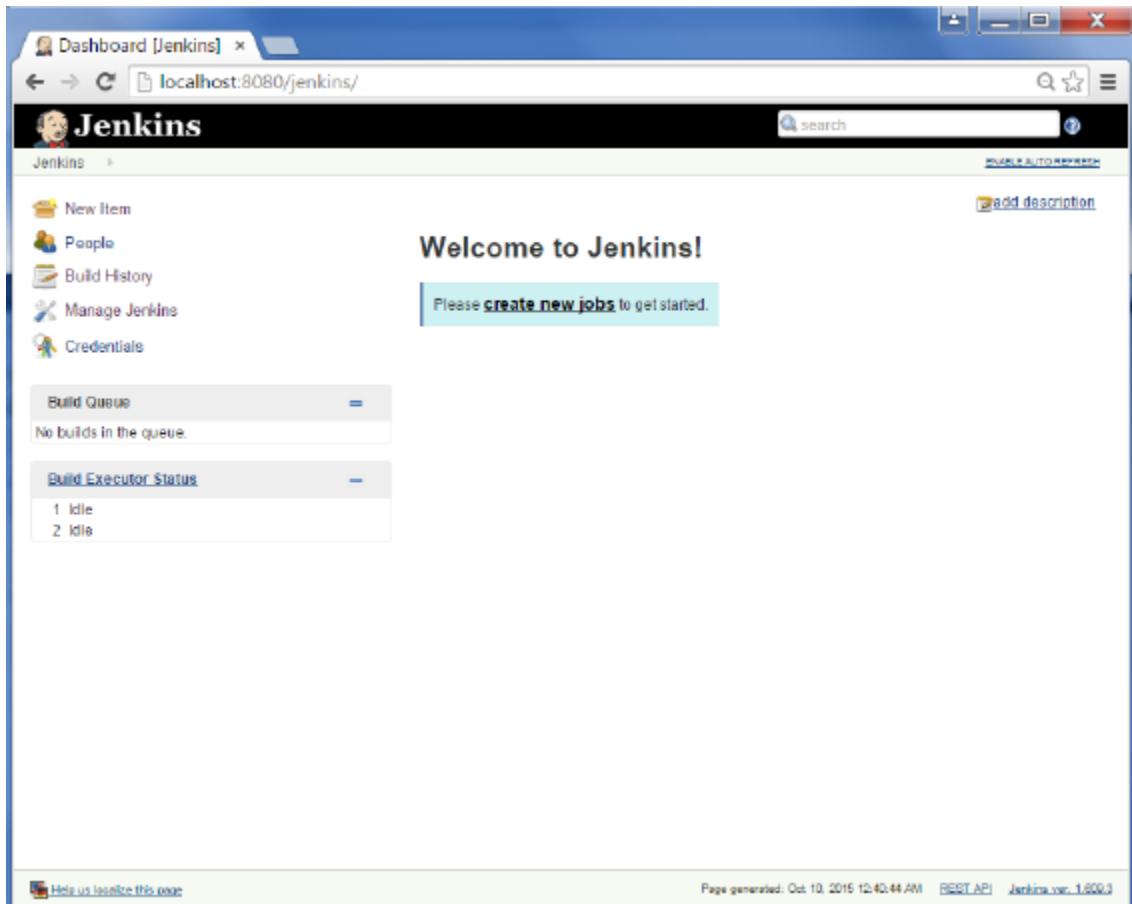
There are various plugins which are available in Jenkins to showcase metrics for builds which are carried out over a period of time. These metrics are useful to understand your builds and how frequently they fail/pass over time. As an example, let's look at the 'Build History Metrics plugin'.

This plugin calculates the following metrics for all of the builds once installed

- Mean Time To Failure (MTTF)
- Mean Time To Recovery (MTTR)
- Standard Deviation of Build Times

Step 1 – Go to the Jenkins dashboard and click on Manage Jenkins

Siva Jenkins Tutorial



Step 2 – Go to the Manage Plugins option.

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/jenkins/manage. The left sidebar includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Under 'Build Queue' and 'Build Executor Status', there are no builds listed. The main content area is titled 'Manage Jenkins' and contains several management options:

- Configure System**: Configure global settings and paths.
- Configure Global Security**: Secure Jenkins, define who is allowed to access the system.
- 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. **(Updates available)**
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from `java.util.logging` related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI**: Access or manage Jenkins from your shell, or from your script.
- Script Console**: Executes arbitrary script for administration/trouble-shooting/diagnostics.
- Manage Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Manage Credentials**: Create/delete/modify the credentials that can be used by Jenkins and by jobs running in Jenkins to connect to 3rd party services.
- About Jenkins**: Get this version and license information.

Step 3 – Go to the Available tab and search for the plugin ‘Build History Metrics plugin’ and choose to ‘install without restart’.

Siva Jenkins Tutorial

The screenshot shows the Jenkins Plugin Manager interface. The title bar says "Update Center [Jenkins]". The address bar shows "localhost:8080/jenkins/pluginManager/available". The main header has tabs for "Updates", "Available" (which is selected), "Installed", and "Advanced". A search bar at the top right contains the text "build-history-metrics-plugin". Below the tabs is a table with columns "Name" and "Version". One row in the table is highlighted, showing "Build History Metrics plugin" and "1.2". To the left of the table is a description: "Provides build metrics that encompass the history of all the runs". At the bottom of the table are three buttons: "Install without restart" (highlighted in blue), "Download now and install after restart", and "Update information obtained: 2 mi".

Step 4 – The following screen shows up to confirm successful installation of the plugin. Restart the Jenkins instance.

Siva Jenkins Tutorial

The screenshot shows a web browser window for the Jenkins Update Center. The URL in the address bar is `localhost:8080/jenkins/updateCenter/`. The main content area is titled "Installing Plugins/Upgrades". On the left sidebar, there are links for "Back to Dashboard", "Manage Jenkins", and "Manage Plugins". The central panel displays the status of the "Build History Metrics plugin" under the "Preparation" section, which includes three items: "Checking Internal connectivity", "Checking update center connectivity", and "Success". Below this, a "Success" message is shown with a blue circular icon. At the bottom of the page, there are two green checkmark icons with instructions: "Go back to the top page (you can start using the installed plugins right away)" and "Restart Jenkins when installation is complete and no jobs are running". The footer of the page includes the Jenkins logo, the URL "HTTP://localhost:8080", and the page generation information "Page generated: Oct 24, 2015 3:53:07 PM - REST API - Jenkins ver. 1.609.3".

When you go to your Job page, you will see a table with the calculated metrics. Metric's are shown for the last 7 days, last 30 days and all time.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for the 'Helloworld' project. On the left, there's a sidebar with links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. Below these are sections for 'Build History' (listing builds #12 to #1), 'MTTR' (Mean Time To Recovery) tables, 'MTTF' (Mean Time To Failure) tables, and 'Standard Deviation' tables. On the right, there are 'Recent Changes' and 'Disable Project' buttons. At the bottom, there's a 'Permalinks' section with a list of build links and RSS feeds for all and failures.

	Last 7 Days	0 ms
MTTR	Last 30 Days	23 hr
	All Time	23 hr
	Last 7 Days	0 ms
MTTF	Last 30 Days	2 days 4 hr
	All Time	2 days 4 hr
	Last 7 Days	0 ms
Standard Deviation	Last 30 Days	52 sec
	All Time	52 sec

	Last 7 Days	0 ms
MTTR	Last 30 Days	23 hr
	All Time	23 hr
	Last 7 Days	0 ms
MTTF	Last 30 Days	2 days 4 hr
	All Time	2 days 4 hr
	Last 7 Days	0 ms
Standard Deviation	Last 30 Days	52 sec
	All Time	52 sec

	Last 7 Days	0 ms
MTTR	Last 30 Days	23 hr
	All Time	23 hr
	Last 7 Days	0 ms
MTTF	Last 30 Days	2 days 4 hr
	All Time	2 days 4 hr
	Last 7 Days	0 ms
Standard Deviation	Last 30 Days	52 sec
	All Time	52 sec

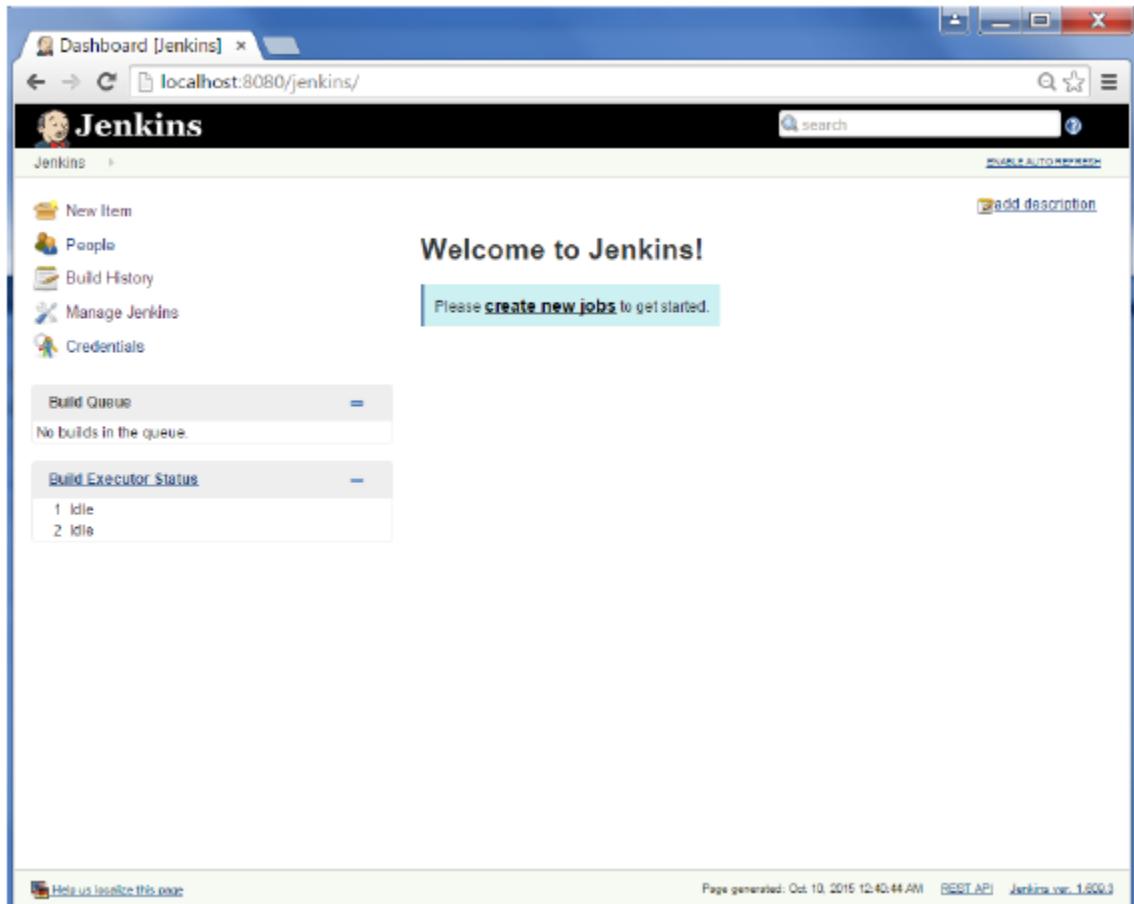
Permalinks

- [Last build \(#12\), 5.5 sec ago](#)
- [Last stable build \(#11\), 8 days 17 hr ago](#)
- [Last successful build \(#11\), 8 days 17 hr ago](#)
- [Last failed build \(#12\), 5.5 sec ago](#)
- [Last unstable build \(#4\), 11 days ago](#)
- [Last unsuccessful build \(#12\), 5.5 sec ago](#)

To see overall trends in Jenkins, there are plugins available to gather information from within the builds and Jenkins and display them in a graphical format. One example of such a plugin is the ‘Hudson global-build-stats plugin’. So let’s go through the steps for this.

Step 1 – Go to the Jenkins dashboard and click on Manage Jenkins

Siva Jenkins Tutorial



Step 2 – Go to the Manage Plugins option

Siva Jenkins Tutorial

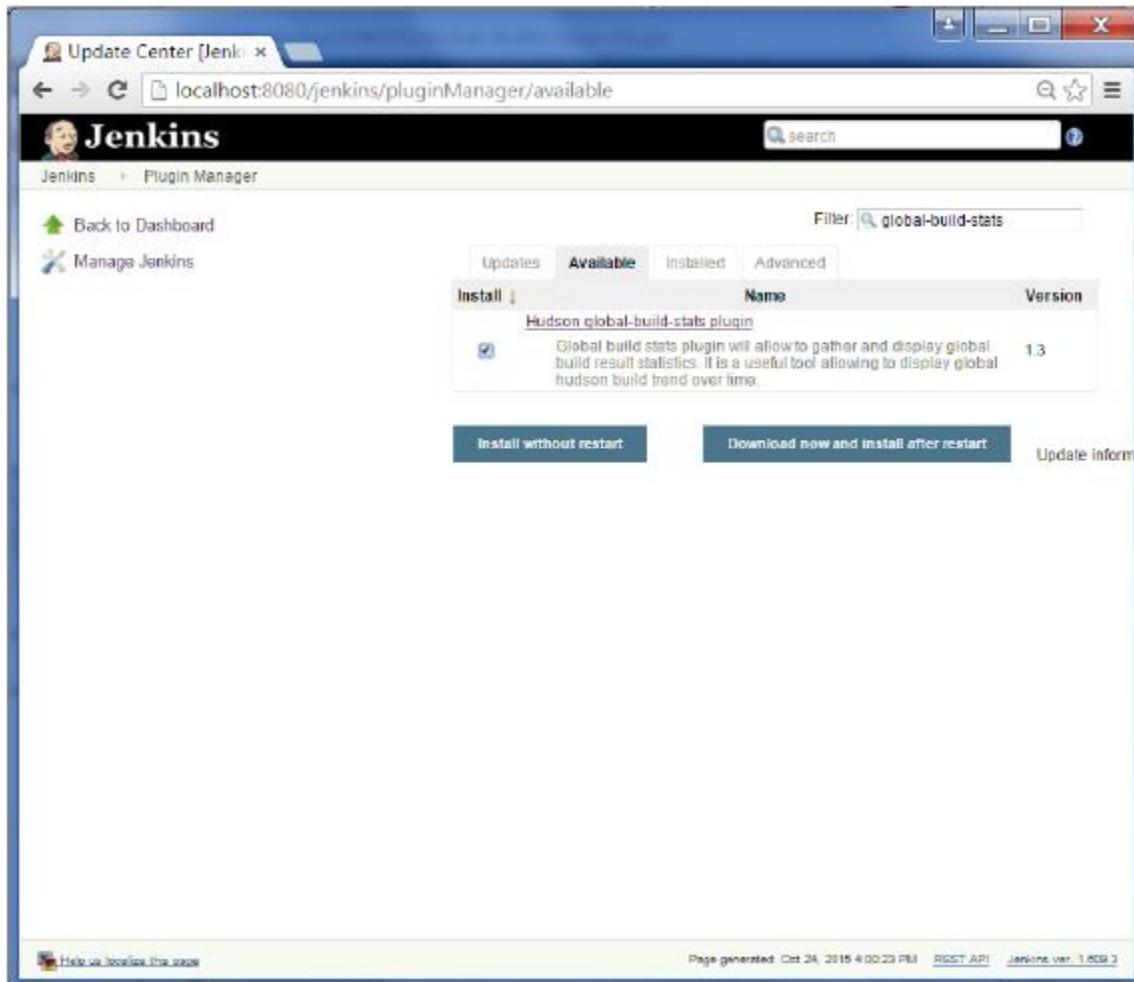
The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/jenkins/manage. The left sidebar includes links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Under 'Manage Jenkins', there are sections for 'Build Queue' (empty) and 'Build Executor Status' (2 Idle). The main content area is titled 'Manage Jenkins' and contains several management options:

- Configure System**: Configure global settings and paths.
- Configure Global Security**: Secure Jenkins; define who is allowed to access the system.
- 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. **(Updates available)**
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from java.util.logging output related to Jenkins.
- Load Statistics**: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI**: Access/Manage Jenkins from your shell, or from your script.
- Script Console**: Executes arbitrary script for administration/trouble-shooting/diagnostics.
- Manage Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Manage Credentials**: Create/delete/modify the credentials that can be used by Jenkins and by jobs running in Jenkins to connect to 3rd party services.
- About Jenkins**: Get this version and license information.

A message at the top right states: 'Your container doesn't use UTF-8 to decode URLs. If you use non-ASCII characters as a job name etc, this will cause problems. See [Containers](#) and [Tomcat 1.8](#) for more details.' It includes 'Setup Security' and 'Dismiss' buttons.

Step 3 – Go to the Available tab and search for the plugin ‘Hudson global-build-stats plugin’ and choose to ‘install without restart’.

Siva Jenkins Tutorial



Step 4 – The following screen shows up to confirm successful installation of the plugin. Restart the Jenkins instance.

Siva Jenkins Tutorial

The screenshot shows a web browser window for the Jenkins Update Center. The URL is `localhost:8080/jenkins/updateCenter/`. The main title is "Installing Plugins/Upgrades". On the left sidebar, there are links for "Back to Dashboard", "Manage Jenkins", and "Manage Plugins". The central content area is titled "Installing Plugins/Upgrades" and shows a "Preparation" section with three items: "Checking internet connectivity", "Checking update center connectivity", and "Success". Below this, it shows the "Hudson global-build-stats plugin" with a "Success" status. There are two buttons at the bottom: "Go back to the top page (you can start using the installed plugins right away)" and "Restart Jenkins when installation is complete and no jobs are running". At the bottom of the page, there is a link "Help us localize this page" and footer text "Page generated: Oct 24, 2015 4:01:04 PM REST API Jenkins ver. 1.809.3".

To see the Global statistics, please follow the Step 5 through 8.

Step 5 – Go to the Jenkins dashboard and click on Manage Jenkins. In the Manage Jenkins screen, scroll down and now you will now see an option called ‘Global Build Stats’. Click on this link.

Siva Jenkins Tutorial



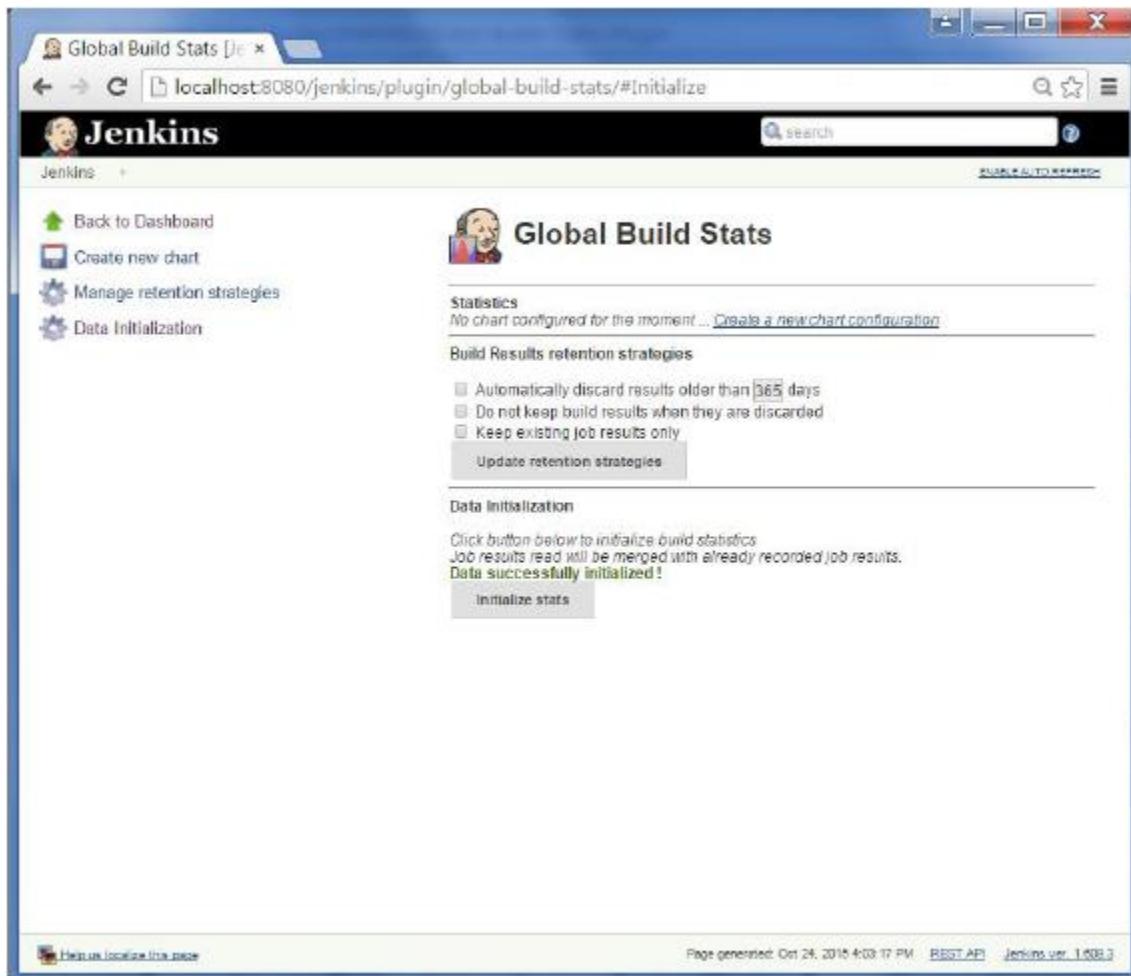
Step 6 – Click on the button ‘Initialize stats’. What this does is that it gathers all the existing records for builds which have already been carried out and charts can be created based on these results.

Siva Jenkins Tutorial

The screenshot shows a web browser window with the title 'Global Build Stats [1e x]' and the URL 'localhost:8080/jenkins/plugin/global-build-stats/'. The page is titled 'Jenkins' and features a sidebar with links: 'Back to Dashboard', 'Create new chart', 'Manage retention strategies', and 'Data Initialization'. The main content area is titled 'Global Build Stats' and includes sections for 'Statistics' (with a note 'No chart configured for the moment ... Create a new chart configuration!'), 'Build Results retention strategies' (with three options: 'Automatically discard results older than 30 days', 'Do not keep build results when they are discarded', and 'Keep existing job results only', with 'Update retention strategies' as a button), and 'Data initialization' (with a note 'Click button below to initialize build statistics. Job results read will be merged with already recorded job results.' and an 'Initialize stats' button). At the bottom, there is a link 'Help us localize this page' and footer text 'Page generated: Oct 24, 2015 4:05:17 PM | REST API | Jenkins ver. 1.809.3'.

Step 7 – Once the data has been initialized, it's time to create a new chart. Click on the ‘Create new chart’ link.

Siva Jenkins Tutorial



Step 8 – A pop-up will come to enter relevant information for the new chart details. Enter the following mandatory information

- Title – Any title information, for this example is given as ‘Demo’
- Chart Width – 800
- Chart Height – 600
- Chart time scale – Daily
- Chart time length – 30 days

The rest of the information can remain as it is. Once the information is entered, click on Create New chart.

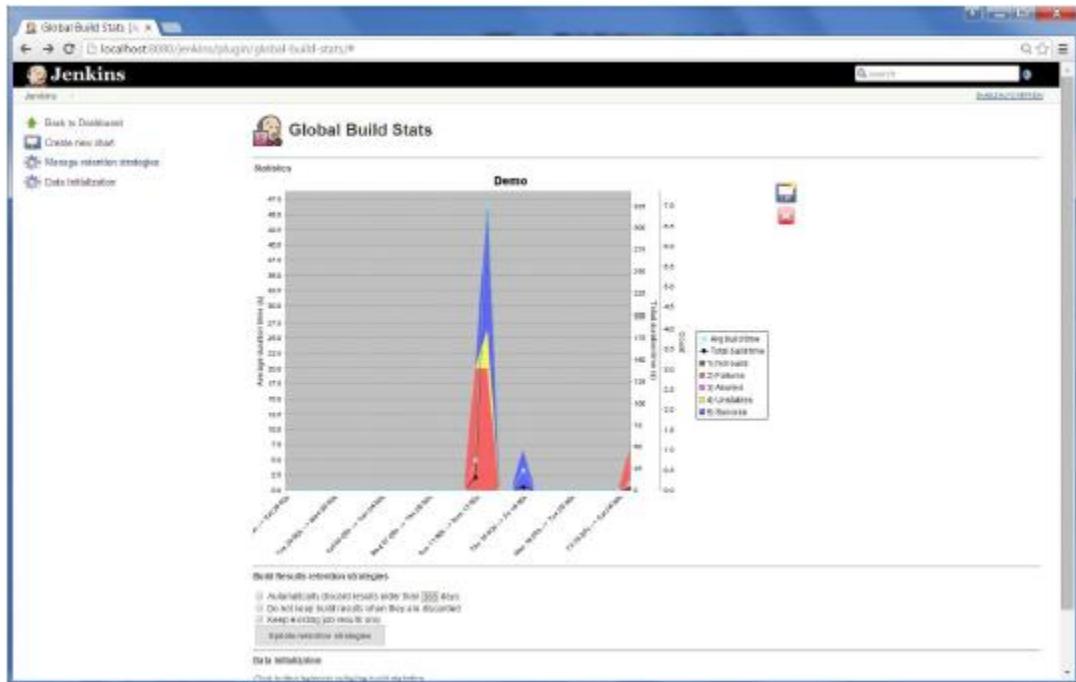
Siva Jenkins Tutorial

The screenshot shows a web browser window for 'Global Build Stats' on the Jenkins plugin. The URL is `localhost:8080/jenkins/plugin/global-build-stats/#`. The main header says 'Jenkins' with a search bar and an 'ENABLE AUTO REFRESH' link. On the left, there's a sidebar with links: 'Back to Dashboard', 'Create new chart', 'Manage retention strategies', and 'Data Initialization'. The main content area is titled 'Global Build Stats' with a sub-section 'Statistics' containing the message 'No chart configured for the moment ... Create a new chart configuration'. Below this is a 'Build Retention strategies' section. A large central box is titled 'Adding new chart' with the following fields:

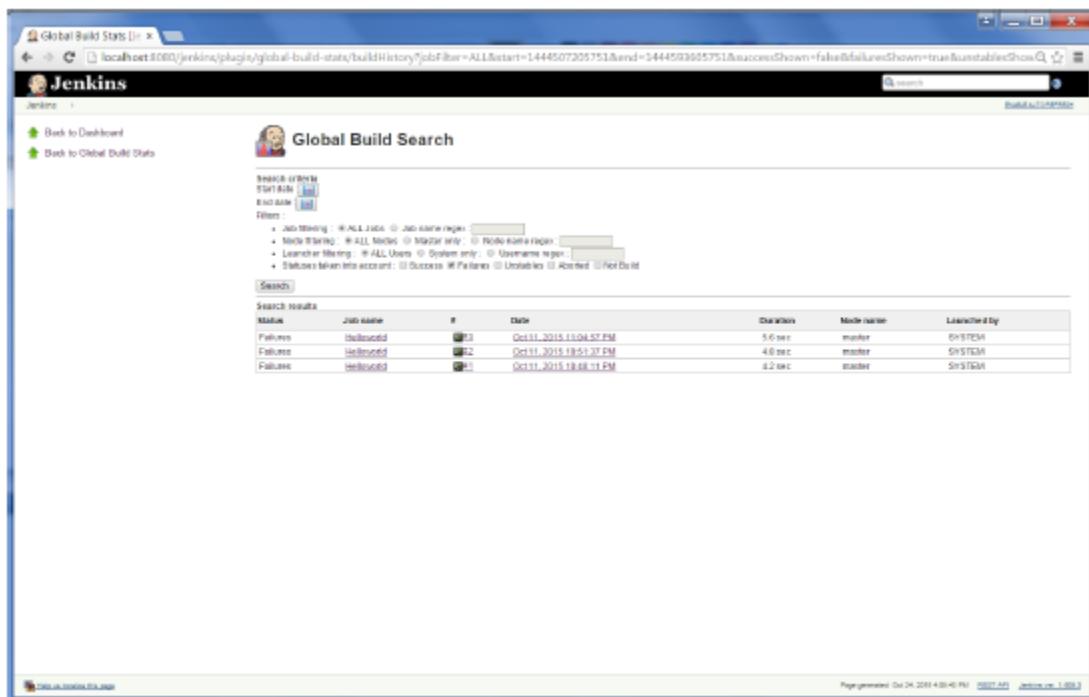
- Title: Demo
- Chart Width * Height: 800 * 600
- Chart time scale: Daily
- Chart time length: 30 days
- Filters:
 - Job filtering: ALL Jobs Job name regex: []
 - Node filtering: ALL Nodes Master only Node name regex: []
 - Launcher filtering: ALL Users System only Username regex: []
 - Statuses taken into account: Success Failures Unstables Aborted Not Build
- Elements displayed on chart:
 - Build statuses with Y Axis type: Count
 - Total build time
 - Average build time

You will now see the chart which displays the trends of the builds over time.

Siva Jenkins Tutorial



If you click on any section within the chart, it will give you a drill down of the details of the job and their builds.



Jenkins - Server Maintenance

The following are some of the basic activities you will carry out, some of which are best practices for Jenkins server maintenance

Siva Jenkins Tutorial

URL Options

The following commands when appended to the Jenkins instance URL will carry out the relevant actions on the Jenkins instance.

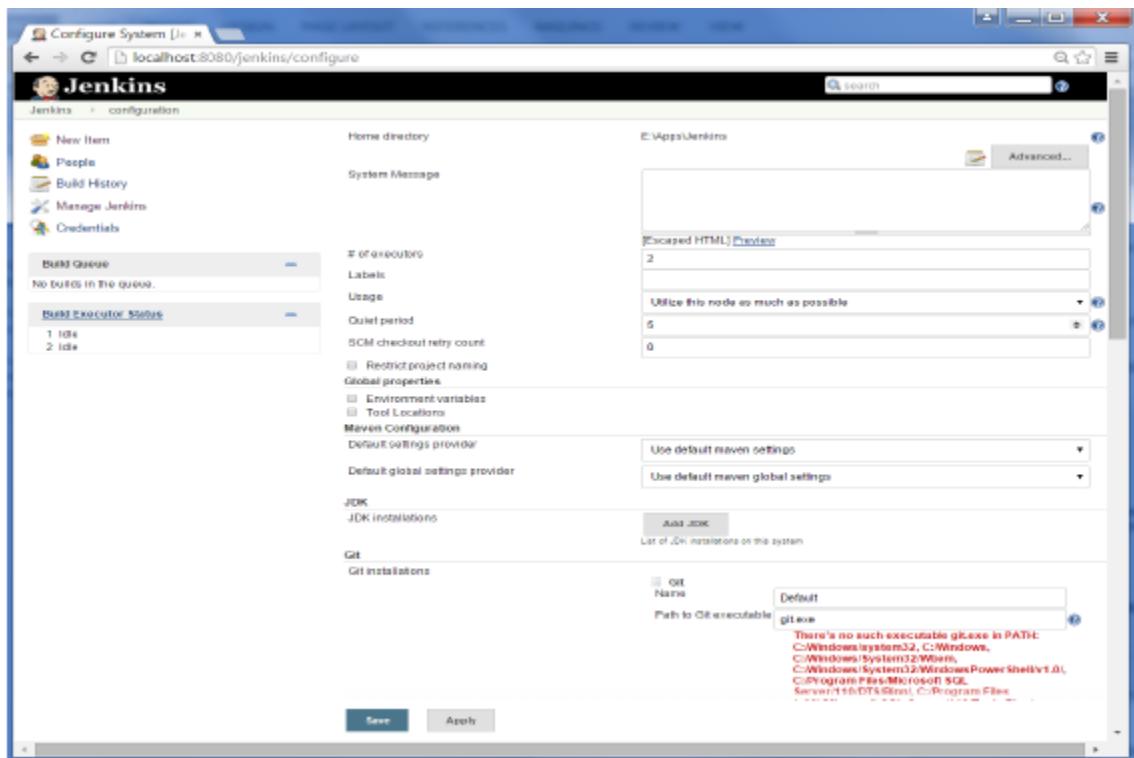
http://localhost:8080/jenkins/exit – shutdown jenkins

http://localhost:8080/jenkins/restart – restart jenkins

http://localhost:8080/jenkins/reload – to reload the configuration

Backup Jenkins Home

The Jenkins Home directory is nothing but the location on your drive where Jenkins stores all information for the jobs, builds etc. The location of your home directory can be seen when you click on Manage Jenkins → Configure system.



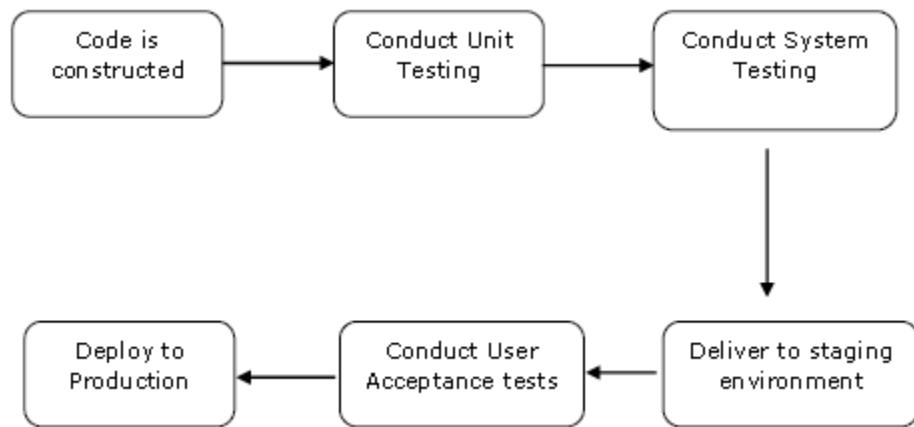
Set up Jenkins on the partition that has the most free disk-space – Since Jenkins would be taking source code for the various jobs defined and doing continuous builds, always ensure that Jenkins is setup on a drive that has enough hard disk space. If you hard disk runs out of space, then all builds on the Jenkins instance will start failing.

Siva Jenkins Tutorial

Another best practice is to write cron jobs or maintenance tasks that can carry out clean-up operations to avoid the disk where Jenkins is setup from becoming full.

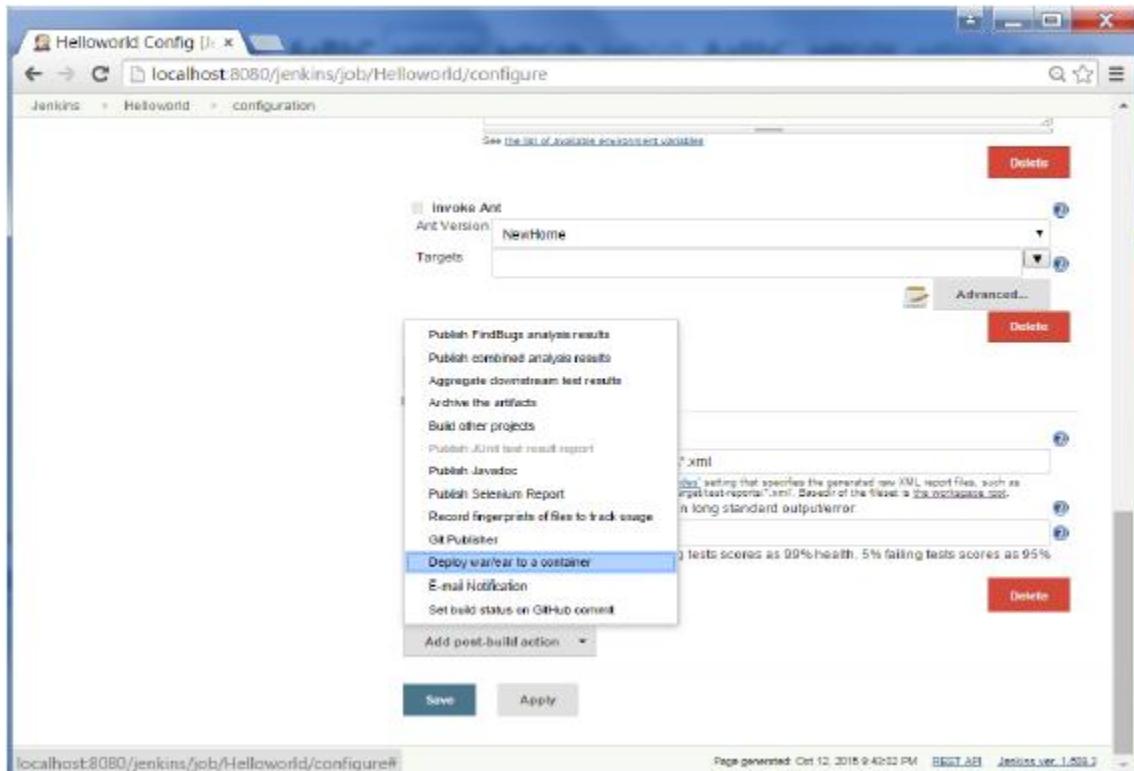
Jenkins - Continuous Deployment

Jenkins provides good support for providing continuous deployment and delivery. If you look at the flow of any software development through deployment, it will be as shown below.



The main part of Continuous deployment is to ensure that the entire process which is shown above is automated. Jenkins achieves all of this via various plugins, one of them being the “Deploy to container Plugin” which was seen in the earlier lessons.

Siva Jenkins Tutorial

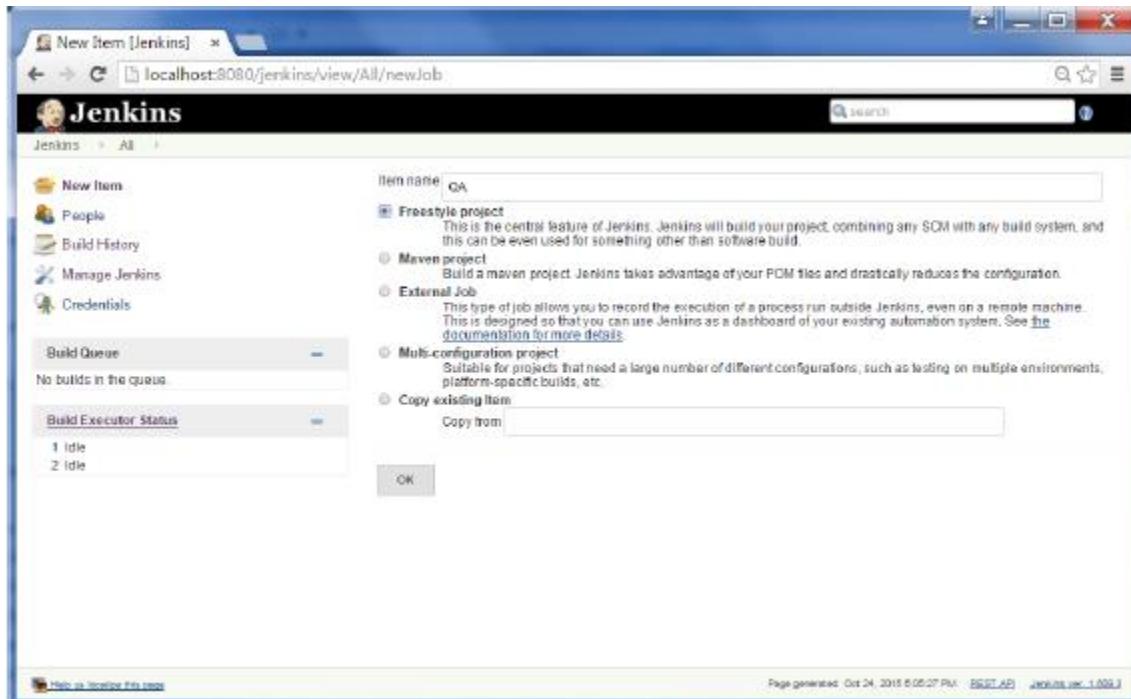


There are plugins available which can actually give you a graphical representation of the Continuous deployment process. But first lets create another project in Jenkins, so that we can see best how this works.

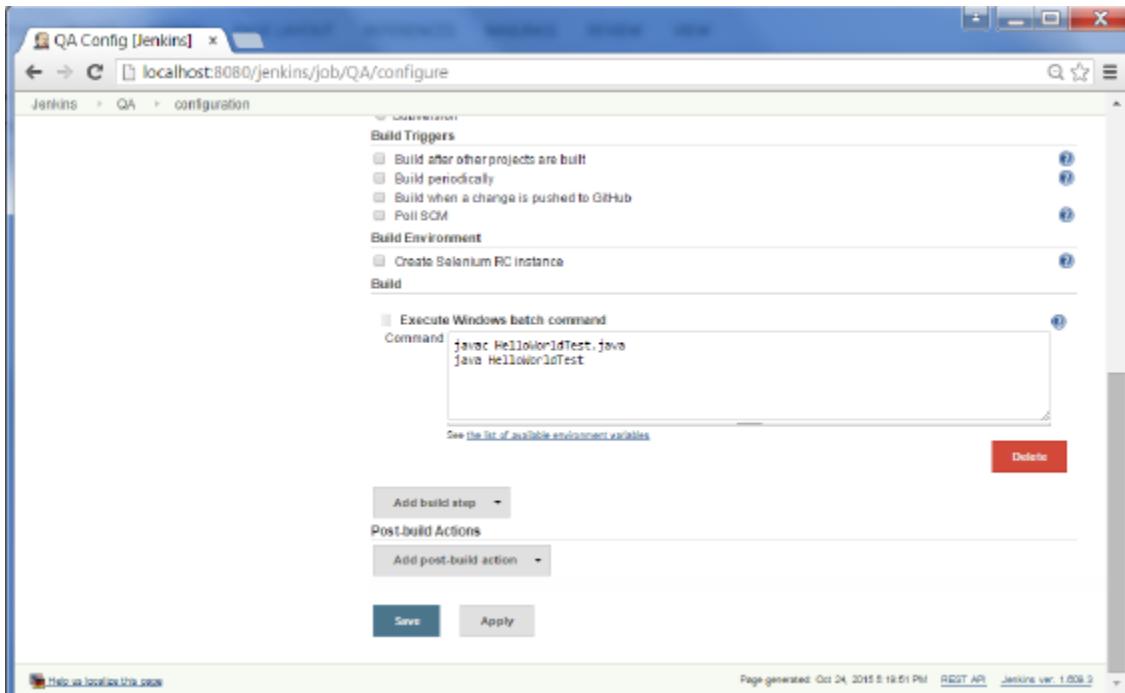
Let's create a simple project which emulates the QA stage, and does a test of the Helloworld application.

Step 1 – Go to the Jenkins dashboard and click on New Item. Choose a ‘Freestyle project’ and enter the project name as ‘QA’. Click on the Ok button to create the project.

Siva Jenkins Tutorial



Step 2 – In this example, we are keeping it simple and just using this project to execute a test program for the Helloworld application.



So our project QA is now setup. You can do a build to see if it builds properly.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for the 'QA' project. On the left, there's a sidebar with links like 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. The main area is titled 'Project QA' and contains a 'Recent Changes' section with a 'Workspace' icon. Below it is a table showing MTTR, MTTF, and Standard Deviation metrics for different time periods. At the bottom, there are 'Permalinks' for the last build and stable build.

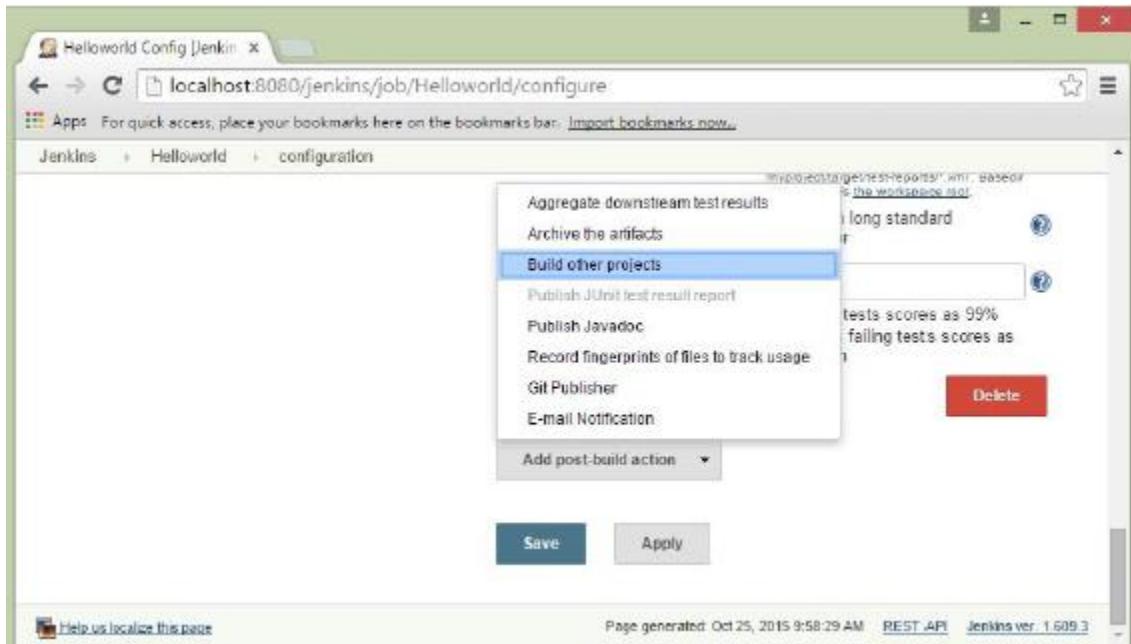
	Last 7 Days	Last 30 Days	All Time
MTTR	2 min 28 sec	2 min 28 sec	2 min 28 sec
MTTF	0 ms	0 ms	0 ms
Standard Deviation	75 ms	75 ms	75 ms

Step 3 – Now go to your Helloworld project and click on the Configure option

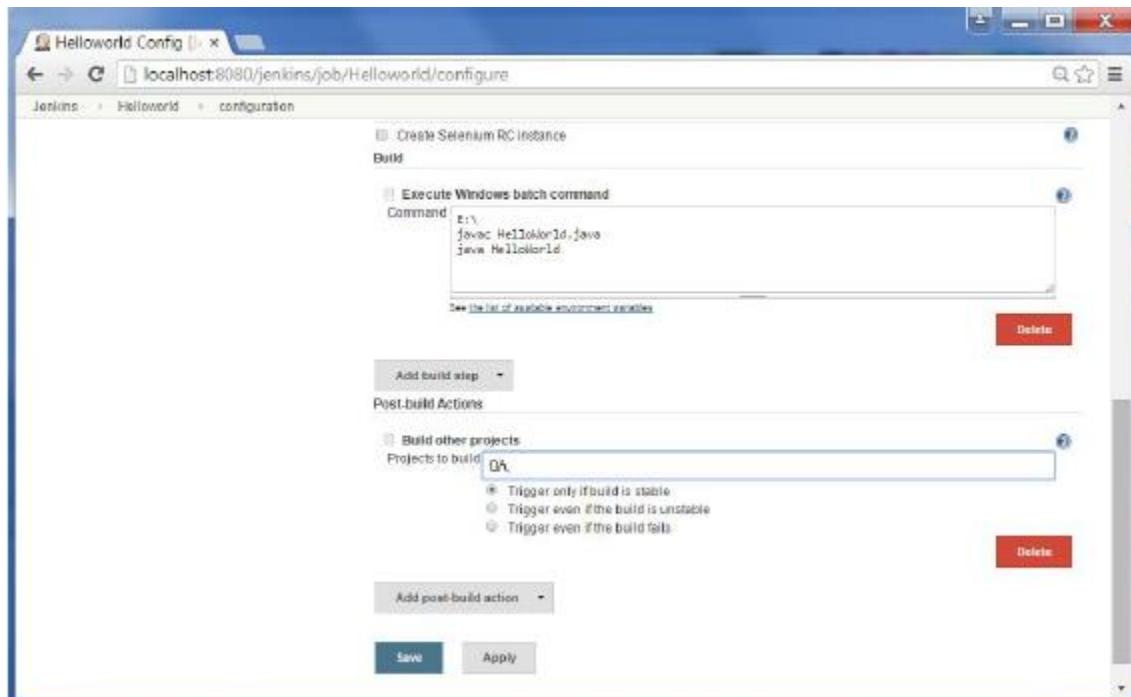
The screenshot shows the Jenkins dashboard with the 'Helloworld' project selected. The sidebar includes 'New item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. The main area displays the 'Build Queue' (empty) and 'Build Executor Status' (idle). The 'Helloworld' project card shows its status (12 days - #15), last success (12 days - #14), last failure (N/A), and last duration (6.6 sec). A 'Configure' button is highlighted at the bottom of the project card.

Step 4 – In the project configuration, choose the ‘Add post-build action’ and choose ‘Build other projects’

Siva Jenkins Tutorial



Step 5 – In the ‘Project to build’ section, enter QA as the project name to build. You can leave the option as default of ‘Trigger only if build is stable’. Click on the Save button.



Step 6 – Build the Helloworld project. Now if you see the Console output, you will also see that after the Helloworld project is successfully built, the build of the QA project will also happen.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for a build named 'Helloworld #14'. The left sidebar has links for Back to Project, Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete Build, and Previous Build. The main area is titled 'Console Output' and displays the following log:

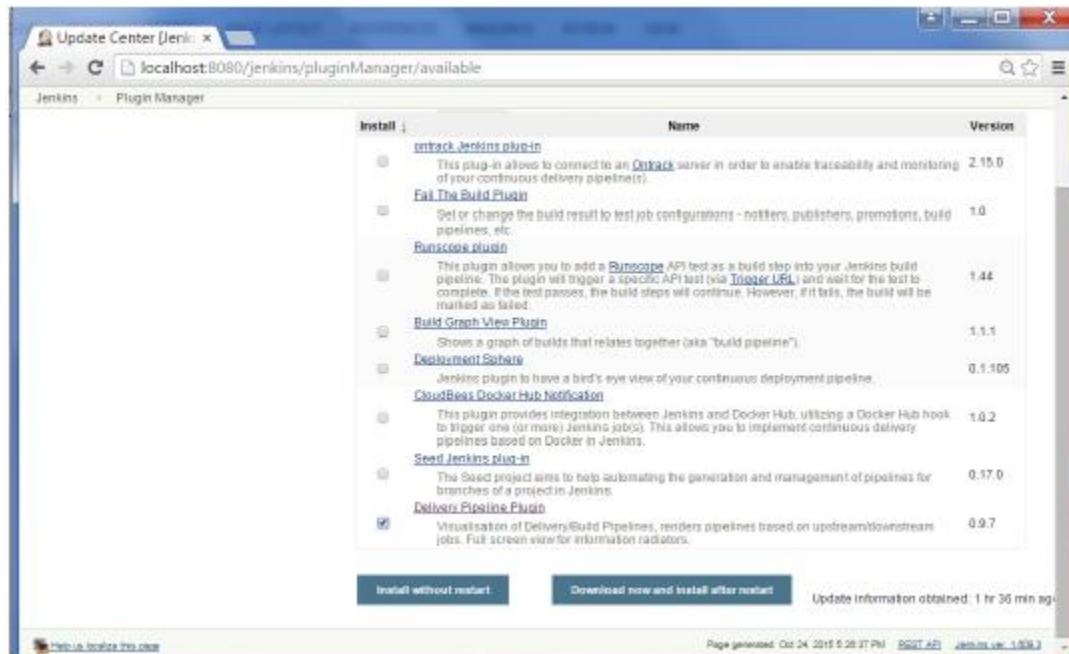
```
Started by user anonymous
Building in workspace E:\Jenkins\jobs\Helloworld\workspace
[workspace] $ cmd /c call E:\Apache\tomcat7\temp\hudson397097412396969963.bat
E:\Jenkins\jobs\Helloworld\workspace>%
'cmd' is not recognized as an internal or external command,
operable program or batch file.

E:\Jenkins\jobs\Helloworld\workspace>java -jar HelloWorld.jar
E:\Jenkins\jobs\Helloworld\workspace>java HelloWorld
Hello World

E:\Jenkins\jobs\Helloworld\workspace>exit 0
warning: you have no plugins providing access control for builds, so falling back to legacy behavior of permitting any downstream builds to be triggered
Triggering a new build of QA
Finished: SUCCESS
```

At the bottom, there are links for Help us localize this page, Page generated: Oct 24, 2015 5:22:02 PM, REST API, and Jenkins ver. 1.809.3.

Step 7 – Let now install the Delivery pipeline plugin. Go to Manage Jenkins → Manage Plugin's. In the available tab, search for ‘Delivery Pipeline Plugin’. Click On Install without Restart. Once done, restart the Jenkins instance.



Step 8 – To see the Delivery pipeline in action, in the Jenkins Dashboard, click on the + symbol in the Tab next to the ‘All’ Tab.

Siva Jenkins Tutorial

The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, and Credentials. The main area displays a table of build history with columns: S (Status), W (Workstation), Name, Last Success, Last Failure, and Last Duration. Two builds are listed: 'HelloWorld' and 'Q8'. A legend at the bottom right indicates colors for success (green), failure (red), and latest build (blue).

Step 9 – Enter any name for the View name and choose the option ‘Delivery Pipeline View’.

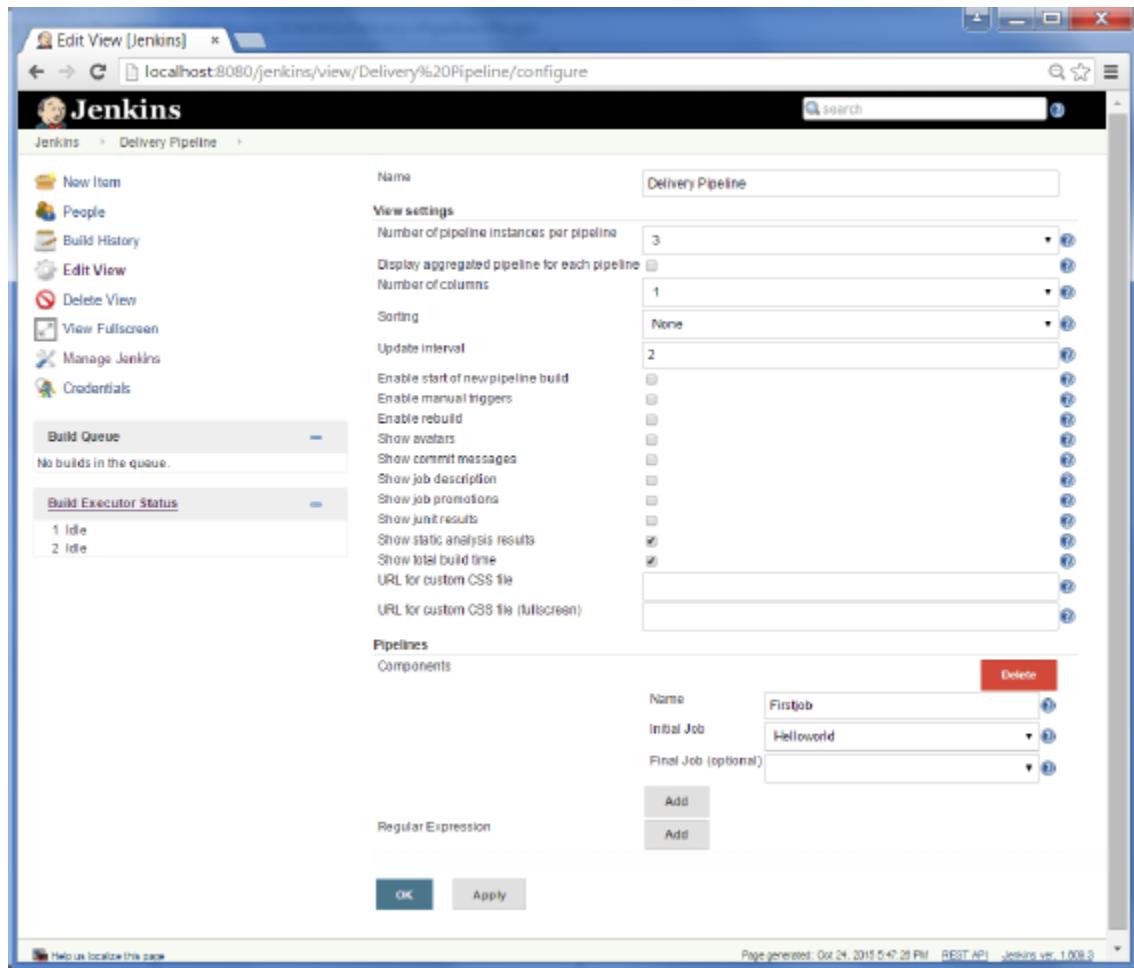
The screenshot shows the 'New View' configuration page at localhost:8080/jenkins/newView. The 'View name' field is set to 'Delivery Pipeline'. Below it, there are three options: 'Build Pipeline View', 'Delivery Pipeline View' (which is selected), and 'List View'. The 'OK' button is visible at the bottom.

Step 10 – In the next screen, you can leave the default options. One can change the following settings –

- Ensure the option ‘Show static analysis results’ is checked.
- Ensure the option ‘Show total build time’ is checked.

Siva Jenkins Tutorial

- For the Initial job – Enter the Helloworld project as the first job which should build.
- Enter any name for the Pipeline
- Click the OK button.



You will now see a great view of the entire delivery pipeline and you will be able to see the status of each project in the entire pipeline.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for a 'Delivery Pipeline'. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', etc. The main area is titled 'Firstjob' and shows three stages in a pipeline:

- Stage 1: Helloworld**
Triggered by user anonymous 29 minutes ago. Total build time: 2 sec.
Status: Success (green bar).
Details: Helloworld 29 minutes ago 1sec.
- Stage 2: QA**
Triggered by user anonymous an hour ago. Total build time: 0 sec.
Status: Success (green bar).
Details: QA 29 minutes ago 1sec.
- Stage 3: QA**
Triggered by user anonymous 2 hours ago. Total build time: 1 sec.
Status: Success (green bar).
Details: QA 2 hours ago 1sec.

At the bottom, there are links for 'Help Us Improve This Page', 'Page generated: Oct 24, 2015 5:49:35 PM', 'REST API', and 'Jenkins ver. 1.632.3'.

Another famous plugin is the **build pipeline plugin**. Let's take a look at this.

Step 1 – Go to Manage Jenkins → Manage Plugin's. In the available tab, search for ‘Build Pipeline Plugin’. Click On Install without Restart. Once done, restart the Jenkins instance.

Siva Jenkins Tutorial

The screenshot shows the Jenkins Plugin Manager interface. The URL in the browser is `localhost:8080/jenkins/pluginManager/available`. The search bar at the top right contains the text "Build pipeline". Below the search bar, there are tabs: "Updates", "Available" (which is selected), "Installed", and "Advanced". A filter bar below the tabs contains the text "Build pipeline". The main area displays a list of available plugins:

Install	Name	Version
<input checked="" type="checkbox"/>	Build Pipeline Plugin	1.4.8
<input type="checkbox"/>	Fail The Build Plugin	1.0
<input type="checkbox"/>	Runscope plugin	1.44
<input type="checkbox"/>	Build Graph View Plugin	1.1.1
<input type="checkbox"/>	Delivery Pipeline Plugin	0.9.7

Below the table are three buttons: "Install without restart", "Download now and install after restart", and "Update info". At the bottom left, there is a link "Help us localize this page". At the bottom right, it says "Page generated: Oct 24, 2015 4:49:09 PM REST API Jenkins ver. 1.809.3".

Step 2 – To see the Build pipeline in action, in the Jenkins Dashboard, click on the + symbol in the Tab next to the ‘All’ Tab.

Siva Jenkins Tutorial

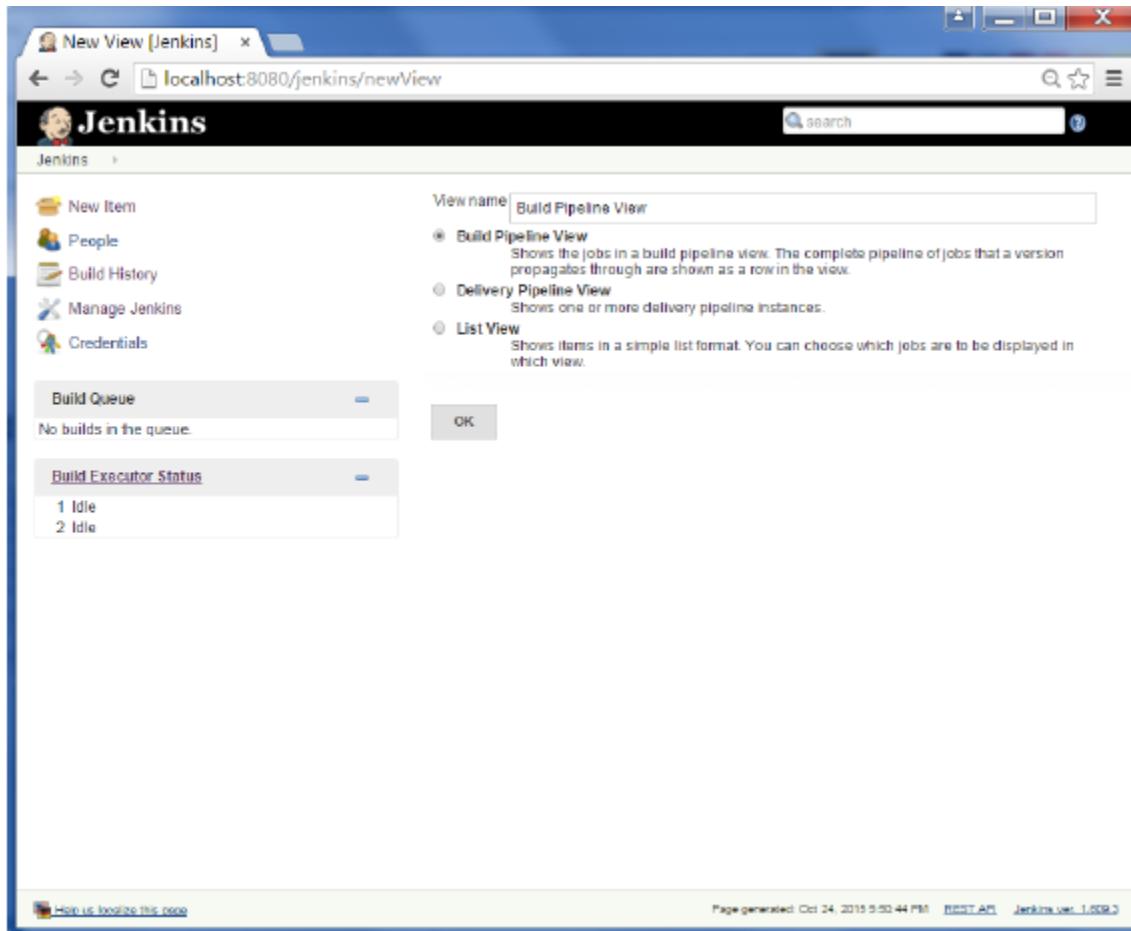
The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, and Credentials. The main area displays a table of builds with columns: S, W, Name, Last Success, Last Failure, and Last Duration. Two builds are listed: 'HelloWorld' (last success 25 min - #14, last failure 1 hr 48 min - #12, duration 1.4 sec) and 'QA' (last success 25 min - #5, last failure 28 min - #2, duration 1.4 sec). Below the table, there are sections for 'Build Queue' (empty) and 'Build Executor Status' (1 idle, 2 idle). At the bottom, there are links for Help, Feedback, and Logout, along with page generation information.

S	W	Name	Last Success	Last Failure	Last Duration
●	●	HelloWorld	25 min - #14	1 hr 48 min - #12	1.4 sec
●	●	QA	25 min - #5	28 min - #2	1.4 sec

Legend: ■ RSS for all ■ RSS for failures ■ RSS for successful builds

Step 3 – Enter any name for the View name and choose the option ‘Build Pipeline View’.

Siva Jenkins Tutorial



Step 4 – Accept the default settings, just in the Selected Initial job, ensure to enter the name of the Helloworld project. Click on the Ok button.

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Edit View [Jenkins]' configuration interface for a 'Build Pipeline View'. The left sidebar includes links for New Item, People, Build History, Edit View, Delete View, Manage Jenkins, and Credentials. Under 'Build Queue', it says 'No builds in the queue.' Under 'Build Executor Status', there are two entries: '1 Idle' and '2 Idle'. The main configuration area has sections for Name, Description, Filter build queue, Filter build executors, Build Pipeline View Title, Layout, and Select Initial Job ('HelloWorld'). There are also checkboxes for restrict triggers, show pipeline headers, and show pipeline parameters. At the bottom are OK and Apply buttons.

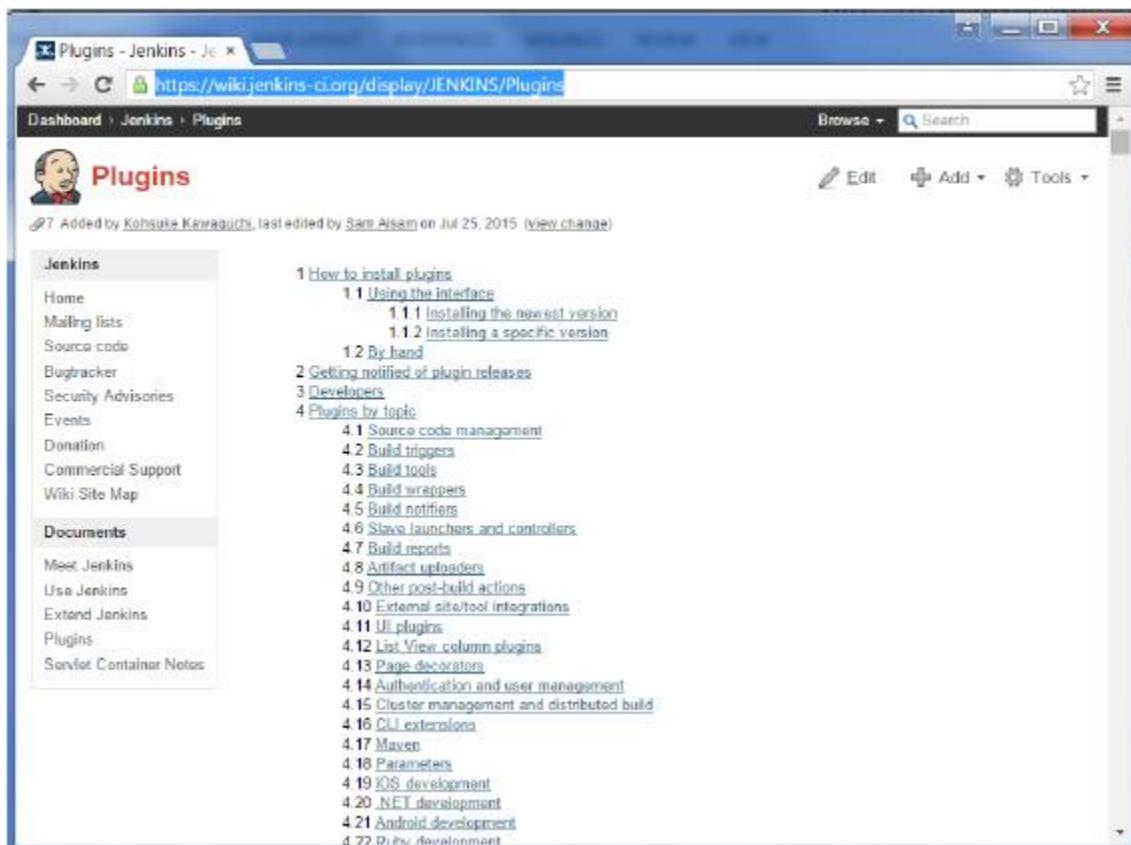
You will now see a great view of the entire delivery pipeline and you will be able to see the status of each project in the entire pipeline.

The screenshot shows the Jenkins 'Build Pipeline View' status page. It displays a horizontal timeline of pipeline stages. From left to right: a grey box labeled 'Pipeline #14', a green box labeled '#14 HelloWorld', and another green box labeled '#14 HelloWorld'. Each box contains a small Jenkins logo and a progress bar. Above the boxes are icons for Map, History, Configure, Add Step, Clone, and Delete. The URL in the browser is 'localhost:8080/jenkins/view/Build%20Pipeline%20View/'.

Siva Jenkins Tutorial

Jenkins - Managing Plugins

To get the list of all plugins available within Jenkins, one can visit the link – <https://wiki.jenkins-ci.org/display/JENKINS/Plugins>

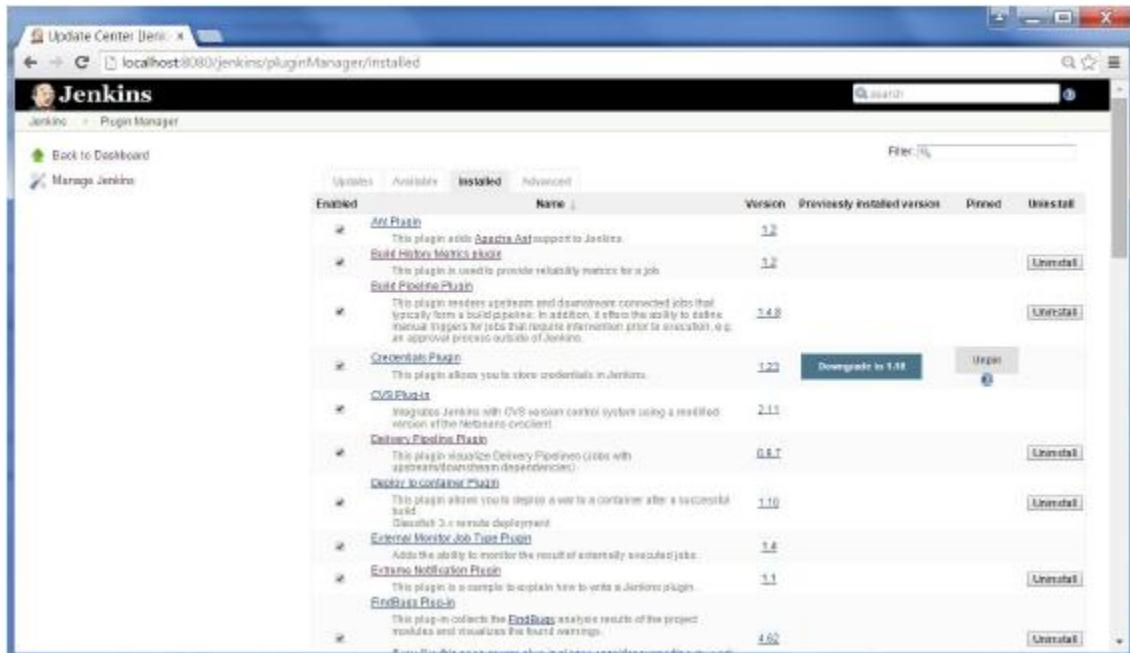


We've already seen many instances for installing plugins, let's look at some other maintenance tasks with regards to plugins

Uninstalling Plugins

To uninstall a plugin, Go to Manage Jenkins → Manage plugins. Click on the Installed tab. Some of the plugins will have the Uninstall option. You can click these buttons to uninstall the plugins. Ensure to restart your Jenkins instance after the uninstallation.

Siva Jenkins Tutorial



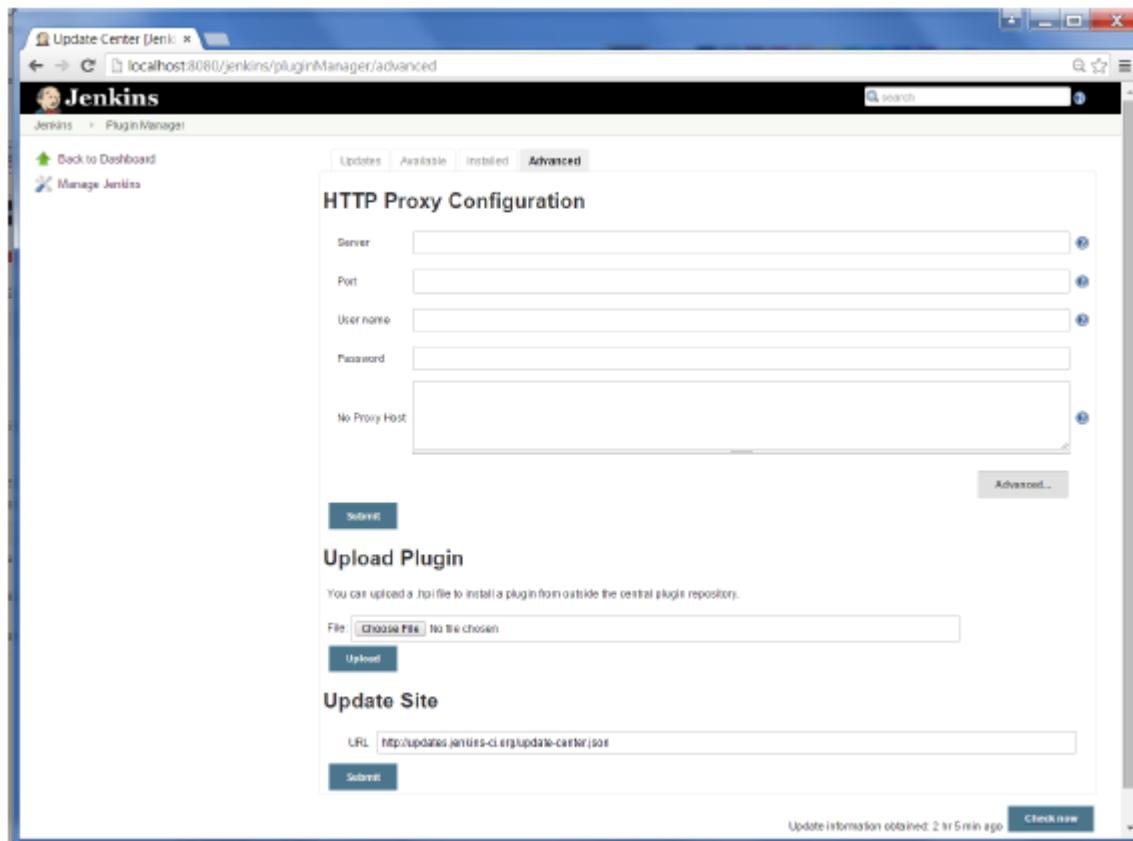
The screenshot shows the Jenkins Plugin Manager interface. The title bar says "Update Center [Beta] - Jenkins - localhost:8080/jenkins/pluginManager/installed". The main area has tabs for "Available", "Installed", and "Advanced". The "Installed" tab is selected. It lists several Jenkins plugins:

Name	Version	Previously installed version	Pinned	Uninstall
Ansible Plugin	1.2			Uninstall
Build History Metrics Plugin	1.2			Uninstall
Build Pipeline Plugin	1.4.8			Uninstall
Credentials Plugin	1.23	Downgrade to 1.19	Unpin	Uninstall
CVS Plugin	2.11			Uninstall
Delivery Pipeline Plugin	0.9.7			Uninstall
Deploy to container Plugin	1.10			Uninstall
External Monitor Job Type Plugin	1.8			Uninstall
Extreme Notification Plugin	1.1			Uninstall
EuroBrics Plugin	4.00			Uninstall

Installing another Version of a Plugin

Sometimes it may be required to install an older version of a plugin, in such a case, you can download the plugin from the relevant plugin page on the Jenkins web site. You can then use the **Upload** option to upload the plugin manually.

Siva Jenkins Tutorial



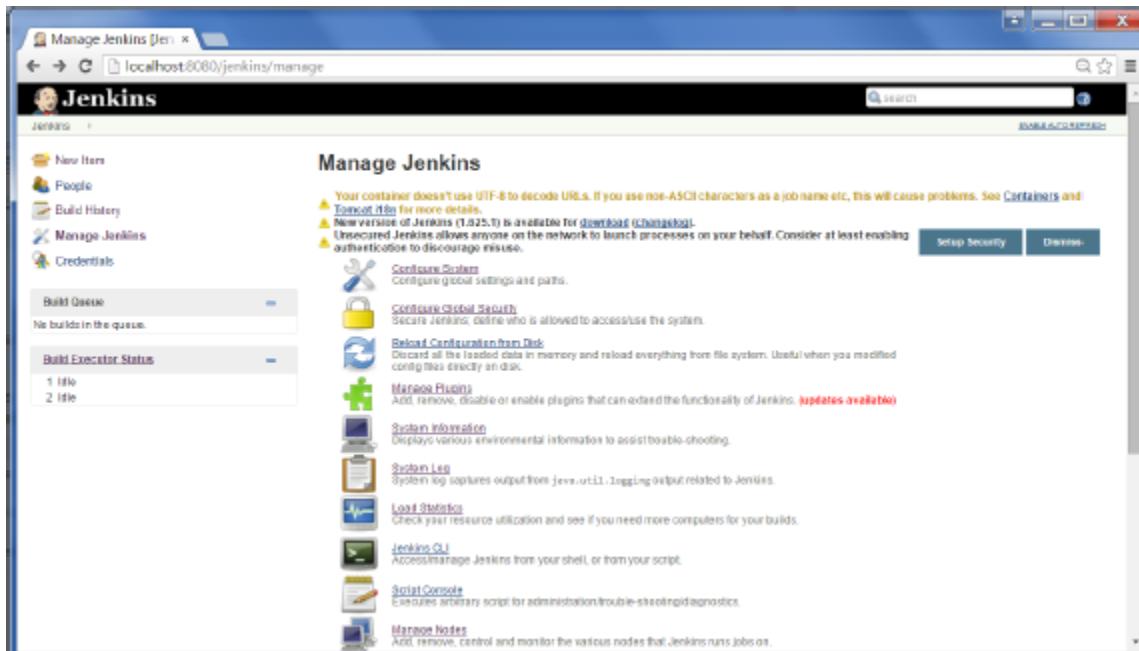
Jenkins - Security

In Jenkins you have the ability to setup users and their relevant permissions on the Jenkins instance. By default you will not want everyone to be able to define jobs or other administrative tasks in Jenkins. So Jenkins has the ability to have a security configuration in place.

To configure Security in Jenkins, follow the steps given below.

Step 1 – Click on Manage Jenkins and choose the ‘Configure Global Security’ option.

Siva Jenkins Tutorial



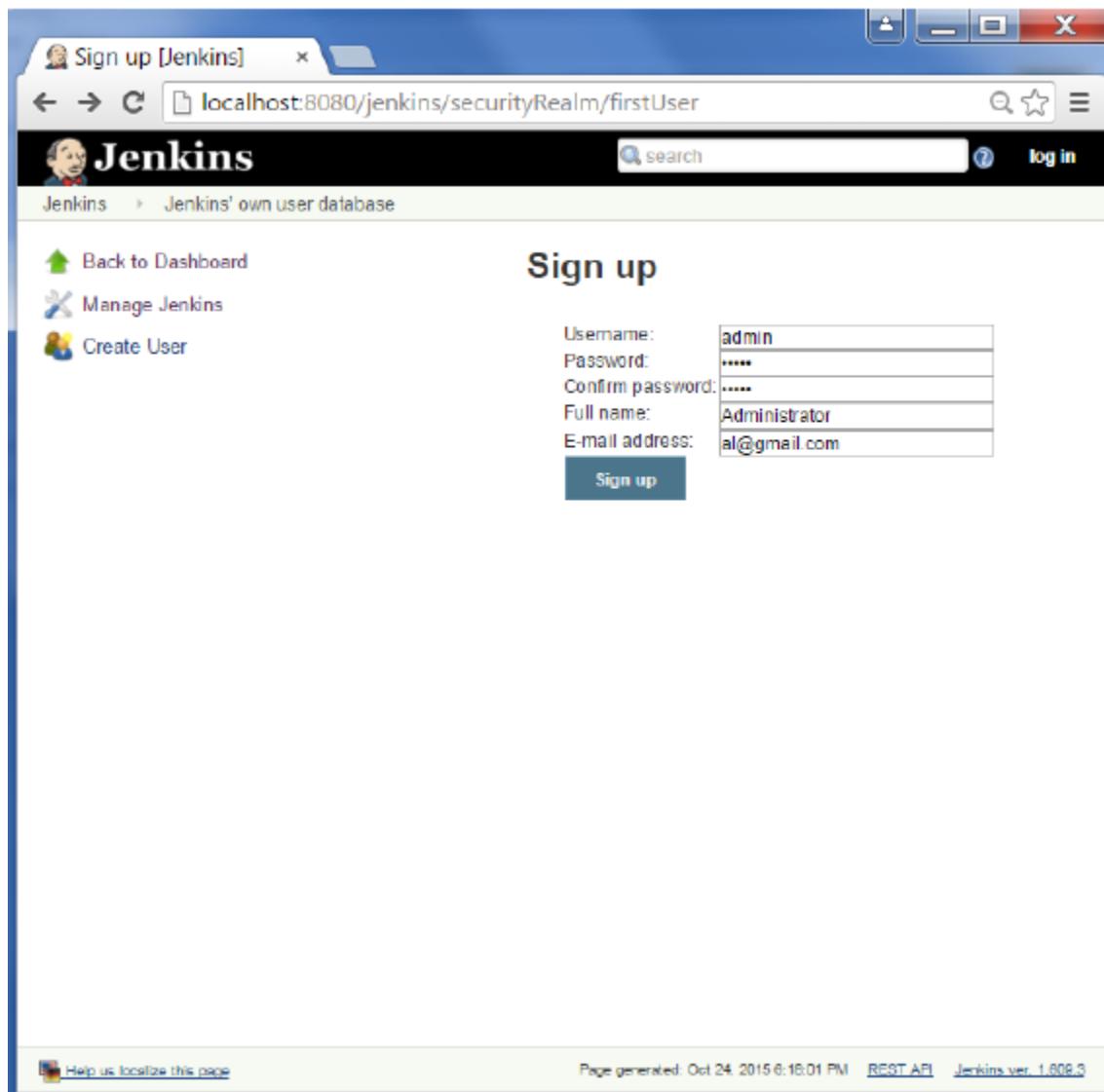
Step 2 – Click on Enable Security option. As an example, let's assume that we want Jenkins to maintain its own database of users, so in the Security Realm, choose the option of ‘Jenkins’ own user database’.

By default you would want a central administrator to define users in the system, hence ensure the ‘Allow users to sign up’ option is unselected. You can leave the rest as it is for now and click the Save button.



Siva Jenkins Tutorial

Step 3 – You will be prompted to add your first user. As an example, we are setting up an admin users for the system.



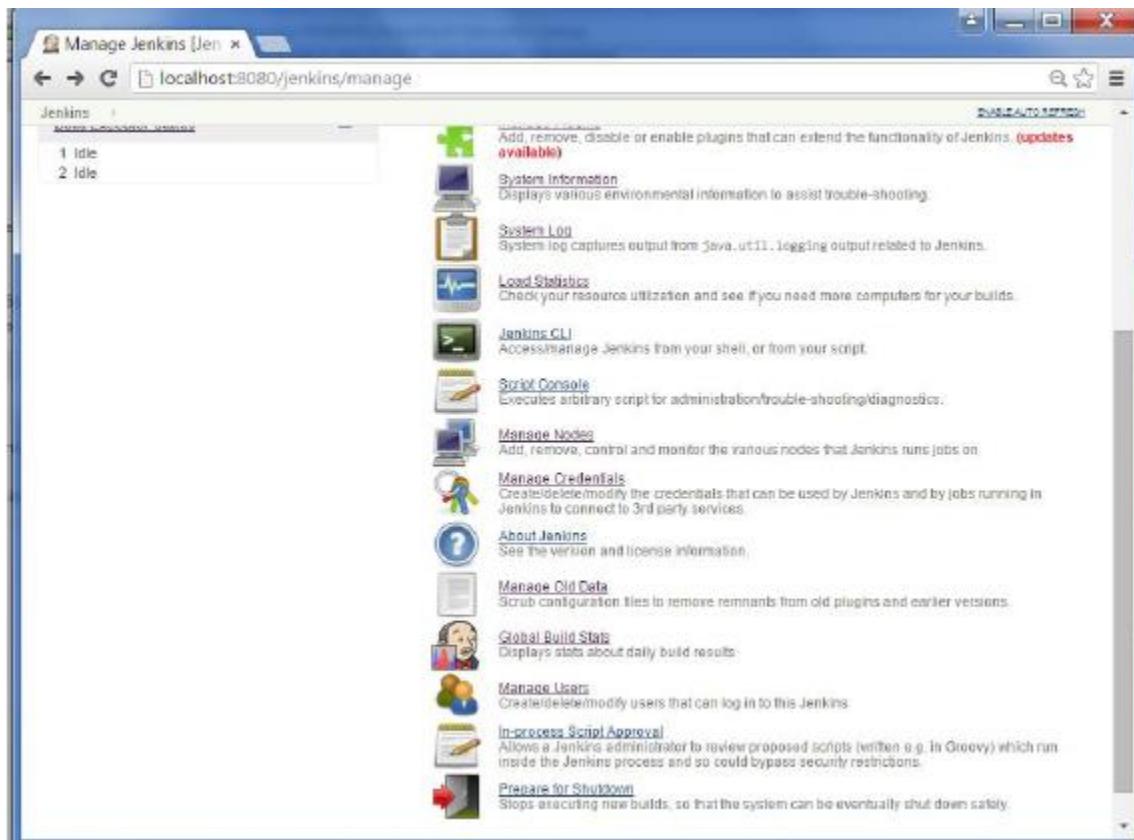
The screenshot shows the Jenkins 'Sign up' page. The URL in the browser is `localhost:8080/jenkins/securityRealm/firstUser`. The page title is 'Sign up [Jenkins]'. On the left sidebar, there are links: 'Back to Dashboard', 'Manage Jenkins', and 'Create User'. The main content area is titled 'Sign up' and contains the following form fields:

Username:	<input type="text" value="admin"/>
Password:	<input type="password" value="*****"/>
Confirm password:	<input type="password" value="*****"/>
Full name:	<input type="text" value="Administrator"/>
E-mail address:	<input type="text" value="al@gmail.com"/>

A blue 'Sign up' button is located at the bottom of the form.

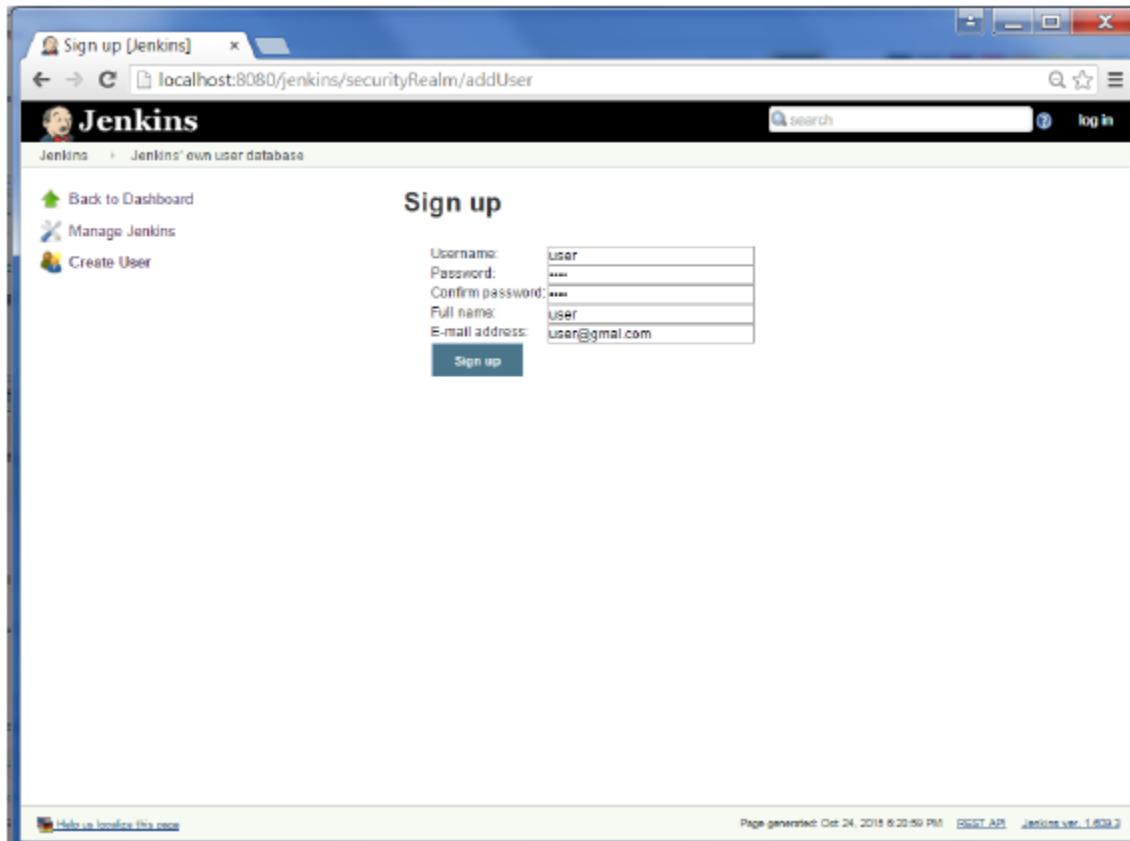
Step 4 – It's now time to setup your users in the system. Now when you go to Manage Jenkins, and scroll down, you will see a 'Manage Users' option. Click this option.

Siva Jenkins Tutorial



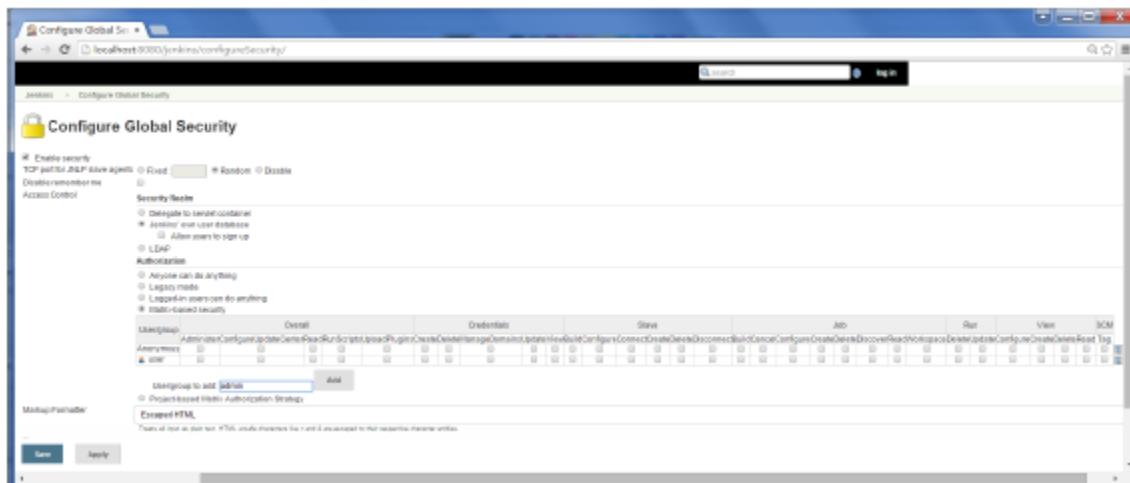
Step 5 – Just like you defined your admin user, start creating other users for the system. As an example, we are just creating another user called ‘user’.

Siva Jenkins Tutorial



Step 6 – Now it's time to setup your authorizations, basically who has access to what. Go to Manage Jenkins → Configure Global Security.

Now in the Authorization section, click on ‘Matrix based security’



Step 7 – If you don't see the user in the user group list, enter the user name and add it to the list. Then give the appropriate permissions to the user.

Siva Jenkins Tutorial

Click on the Save button once you have defined the relevant authorizations.

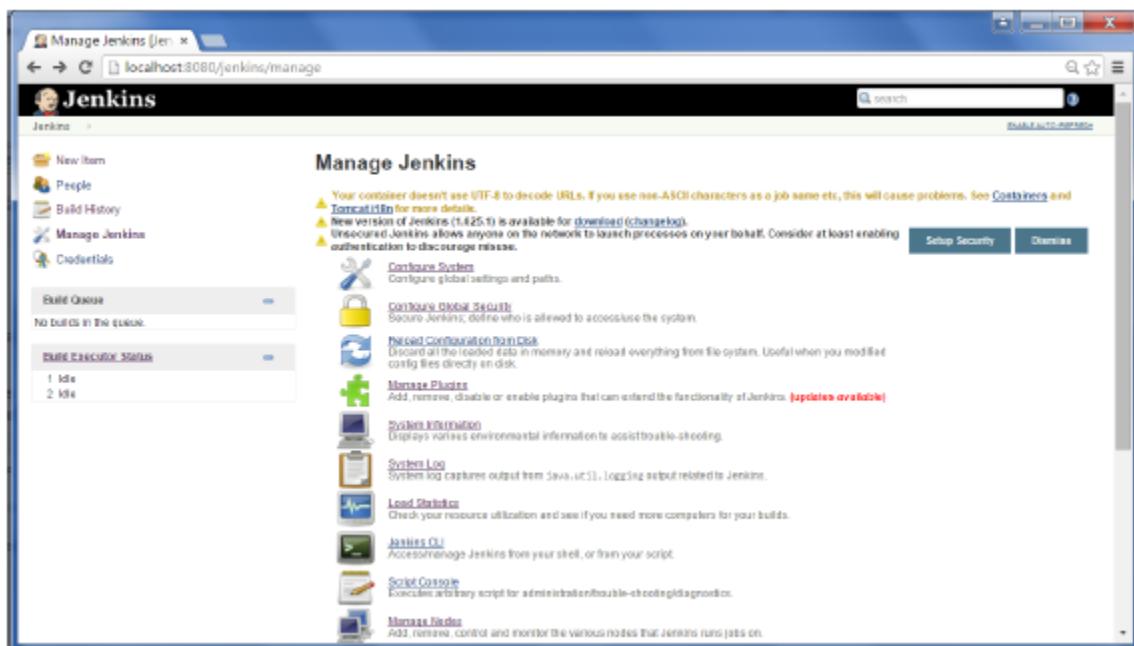
Your Jenkins security is now setup.

Note – For Windows AD authentication, one has to add the Active Directory plugin to Jenkins.

Jenkins - Backup Plugin

Jenkins has a backup plugin which can be used to backup critical configuration settings related to Jenkins. Follow the steps given below to have a backup in place.

Step 1 – Click on Manage Jenkins and choose the ‘Manage Plugins’ option.



Step 2 – In the available tab, search for ‘Backup Plugin’. Click On Install without Restart. Once done, restart the Jenkins instance

Siva Jenkins Tutorial

The screenshot shows the Jenkins Update Center interface. The URL is `localhost:8080/jenkins/pluginManager/available`. The 'Available' tab is selected. A search bar at the top right contains the text 'backup'. The results table lists several plugins:

Install	Name	Version
<input checked="" type="checkbox"/>	Backup plugin	1.6.1
<input type="checkbox"/>	Backup and monitor job plugin	1.0
<input type="checkbox"/>	Instal CloudBees Jenkins Enterprise	15.06.1
<input type="checkbox"/>	CloudBees Free Enterprise Plugins	5.0
<input type="checkbox"/>	Periodic Backup	1.3
<input type="checkbox"/>	ThinBackup	1.7.4

Below the table are three buttons: 'Install without restart', 'Download now and install after restart', and 'Update information of ...'.

The screenshot shows the Jenkins Update Center interface. The URL is `localhost:8080/jenkins/updateCenter/`. The title is 'Installing Plugins/Upgrades'. The 'Preparation' section lists:

- Checking internet connectivity
- Checking up date center connectivity
- Success

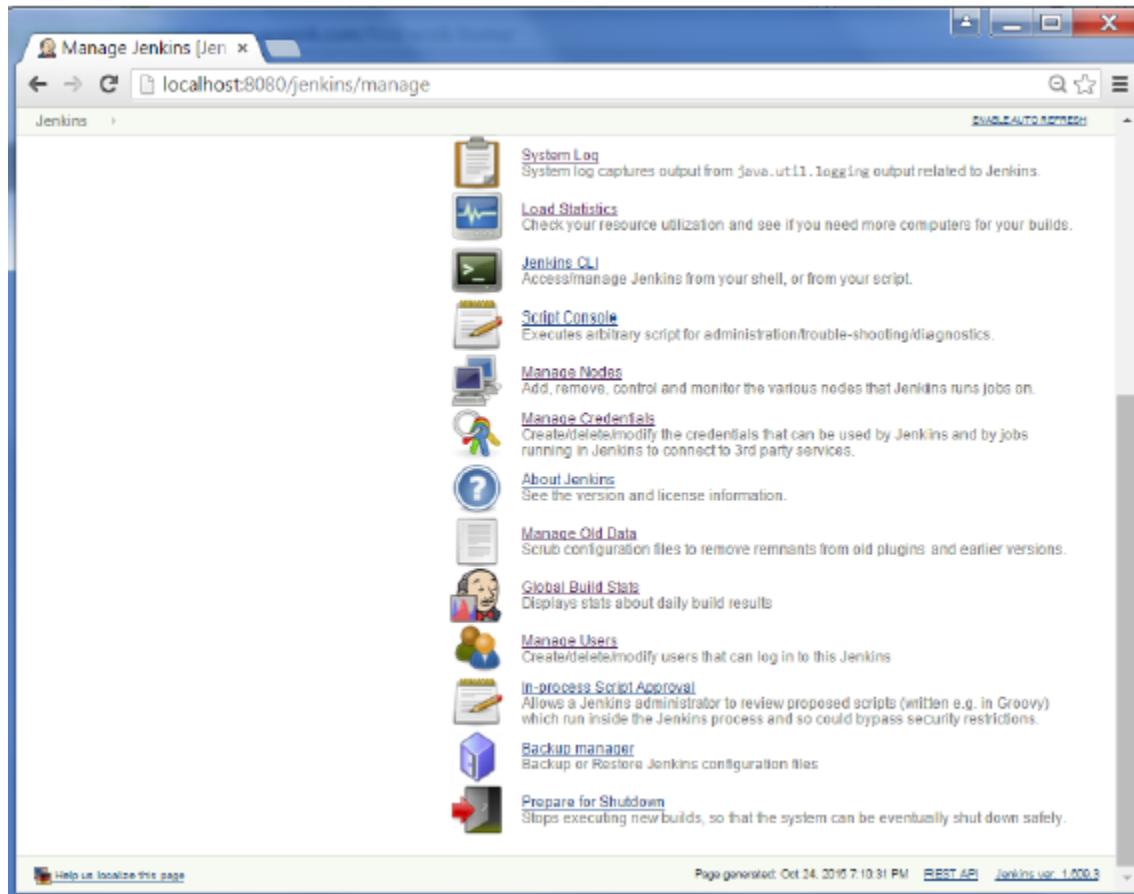
The 'Backup plugin' status is shown as 'Success' with a blue circular icon. Below the preparation section, there are two green checkmark icons with instructions:

- Go back to the top page (you can start using the installed plugins right away)
- Restart Jenkins when installation is complete and no jobs are running

At the bottom of the page, there is a link 'Help us localize this page' and footer text 'Page generated: Oct 26, 2015 8:26:36 PM REST API Jenkins ver. 1.651.3'.

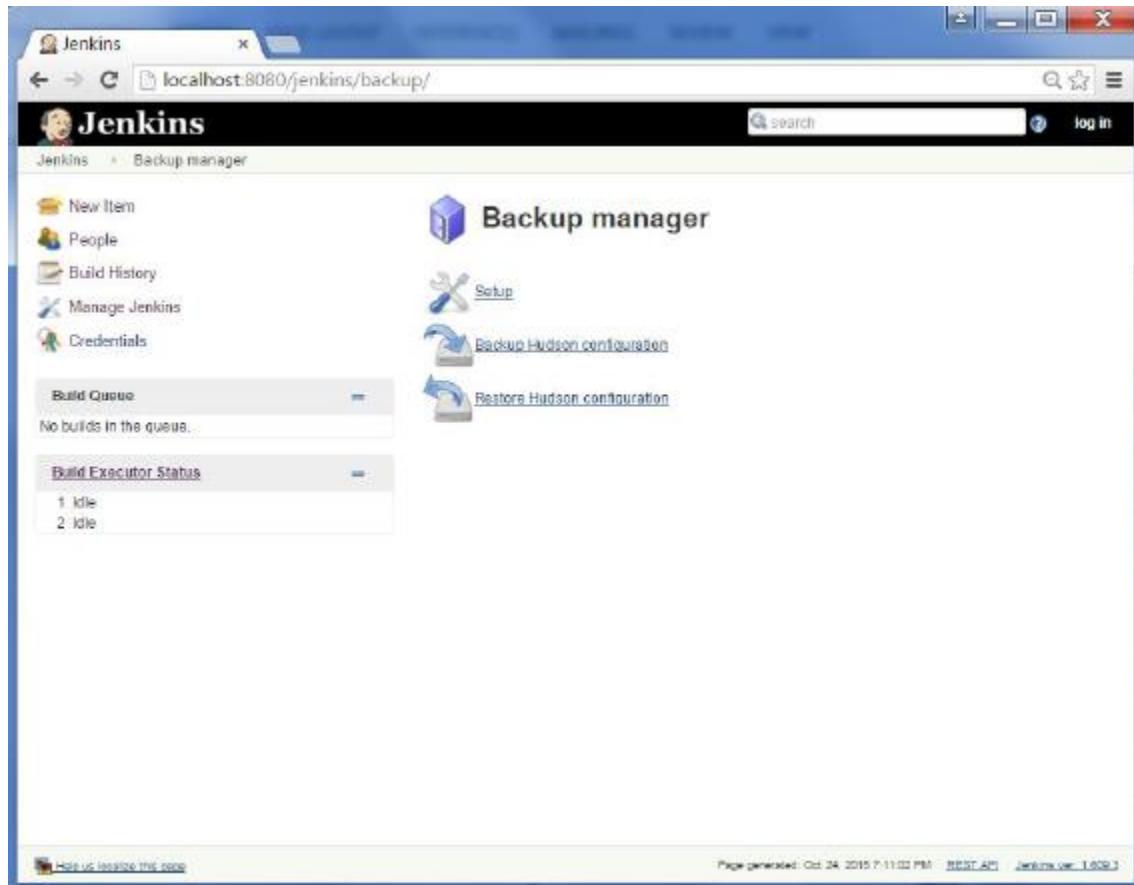
Step 3 – Now when you go to Manage Jenkins, and scroll down you will see ‘Backup Manager’ as an option. Click on this option.

Siva Jenkins Tutorial



Step 4 – Click on Setup.

Siva Jenkins Tutorial



Step 5 – Here, the main field to define is the directory for your backup. Ensure it's on another drive which is different from the drive where your Jenkins instance is setup. Click on the Save button.

Siva Jenkins Tutorial

The screenshot shows the Jenkins Backup manager interface. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Below these are two expandable sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area is titled 'Backup config files' and contains 'Backup configuration' settings. It includes fields for 'Hudson root directory' (E:\Jenkins), 'Backup directory' (D:\Backup), 'Format' (Zip), 'File name template' (backup_@date@.extension@), and 'Custom exclusions'. There are also checkboxes for 'Verbose mode', 'Configuration files (.xml) only', and 'No shutdown'. Under 'Backup content', there are four checkboxes: 'Backup job workspace', 'Backup builds history', 'Backup maven artifacts archives', and 'Backup fingerprints'. A 'Save' button is at the bottom right of the configuration section. At the very bottom of the page, there are links for 'Help us localize this page', 'Page generated: Oct 24, 2015 7:17:45 PM', 'REST API', and 'Jenkins ver. 1.609.3'.

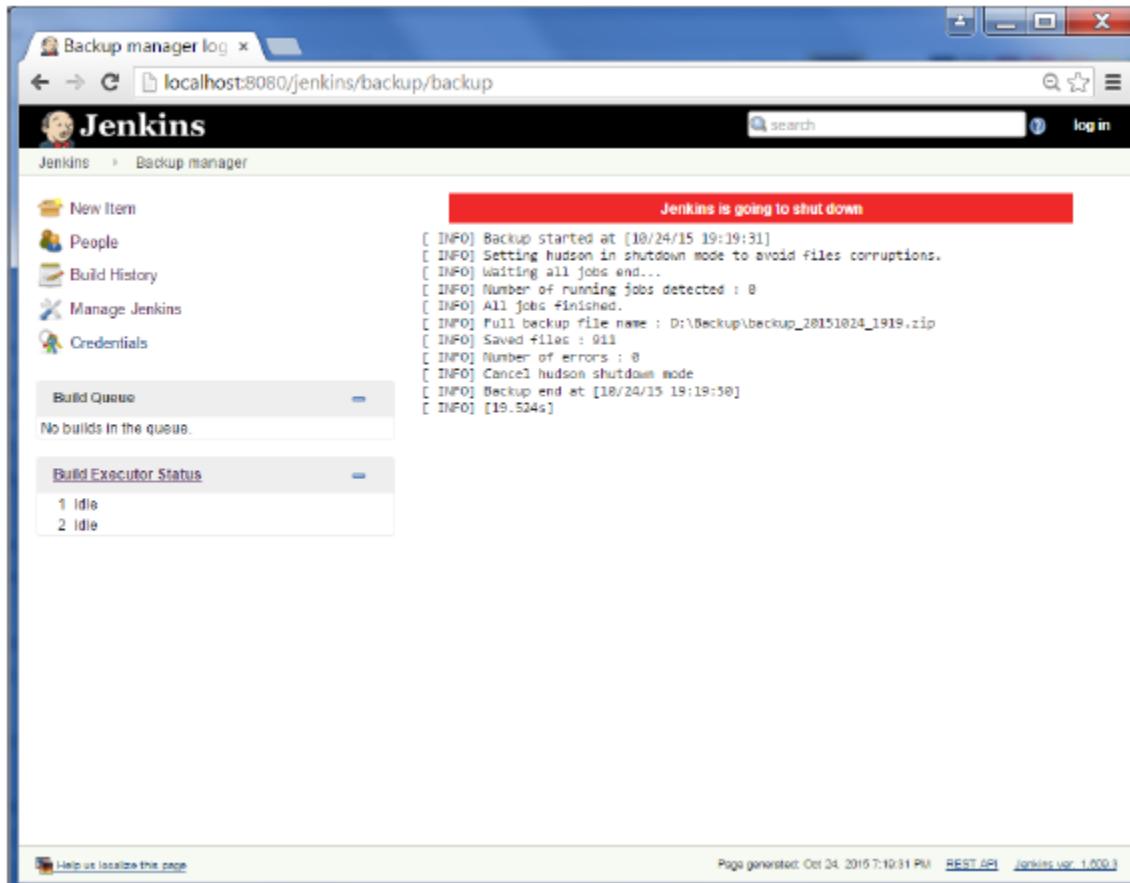
Step 6 – Click on the ‘Backup Hudson configuration’ from the Backup manager screen to initiate the backup.

Siva Jenkins Tutorial

The screenshot shows the Jenkins interface for managing backups. At the top, there's a navigation bar with links for 'Jenkins', 'Backup manager', 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Below this is a search bar and a 'log in' link. The main content area is titled 'Backup manager' and contains three main sections: 'Setup' (with a gear icon), 'Backup Hudson configuration' (with a blue arrow icon), and 'Restore Hudson configuration' (with a red arrow icon). On the left, there are two dropdown menus: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle'). At the bottom, there are links for 'Help us localize this page' and 'Page generated: Oct 24, 2015 7:11:02 PM - REST API - Jenkins ver. 1.608.3'.

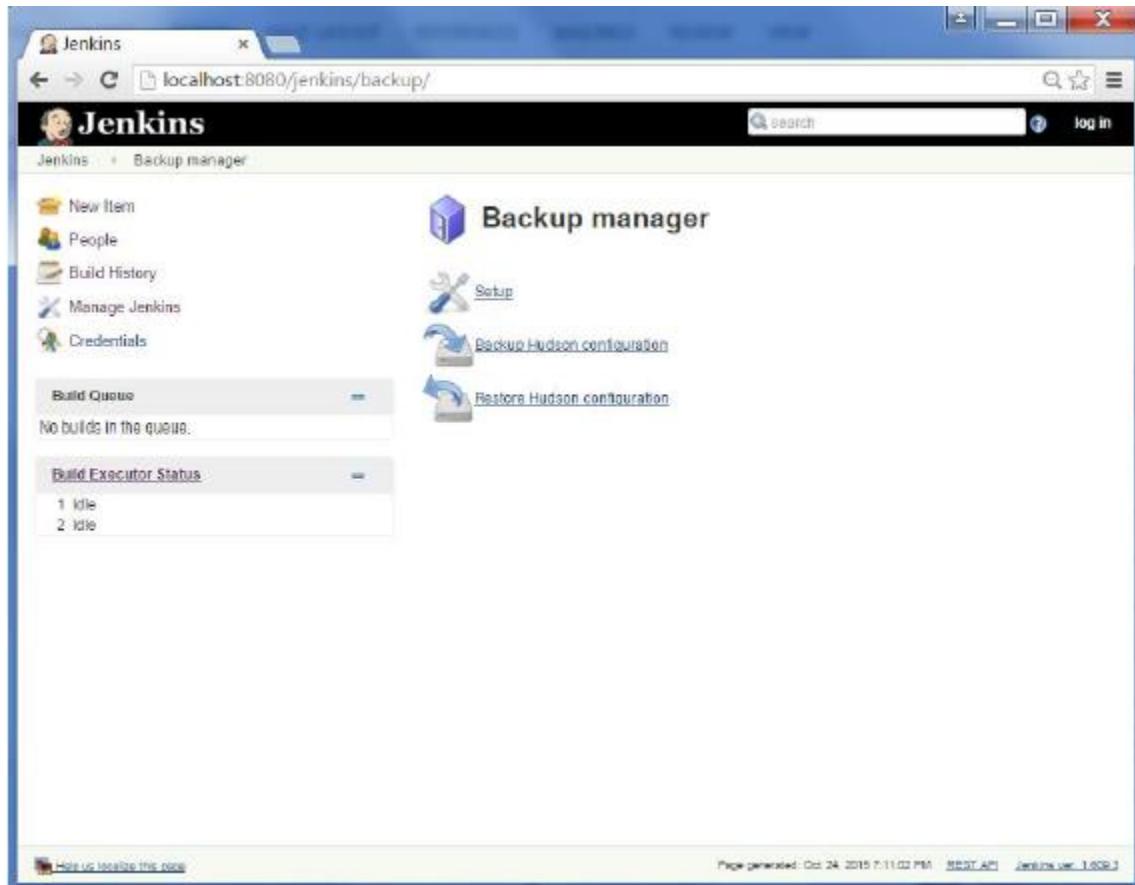
The next screen will show the status of the backup

Siva Jenkins Tutorial



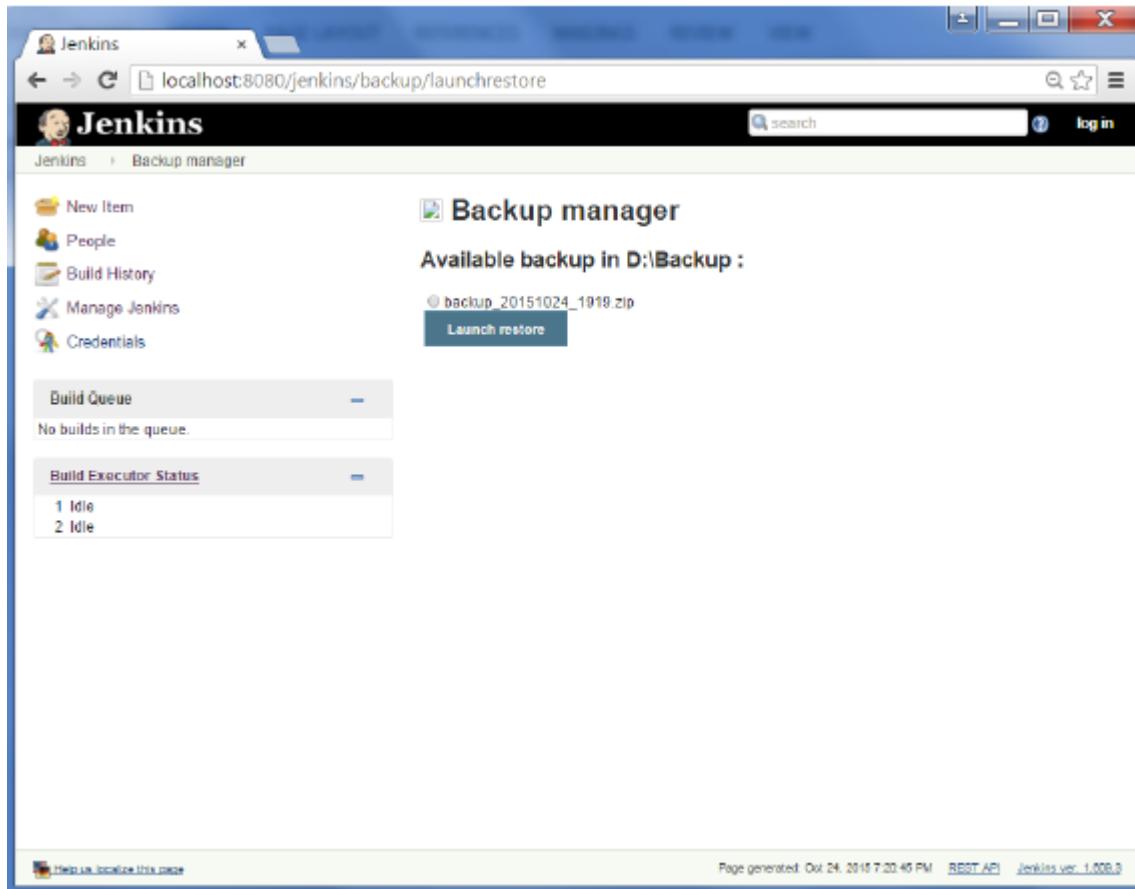
To recover from a backup, go to the Backup Manager screen, click on Restore Hudson configuration.

Siva Jenkins Tutorial



The list of backup's will be shown, click on the appropriate one to click on Launch Restore to begin the restoration of the backup.

Siva Jenkins Tutorial

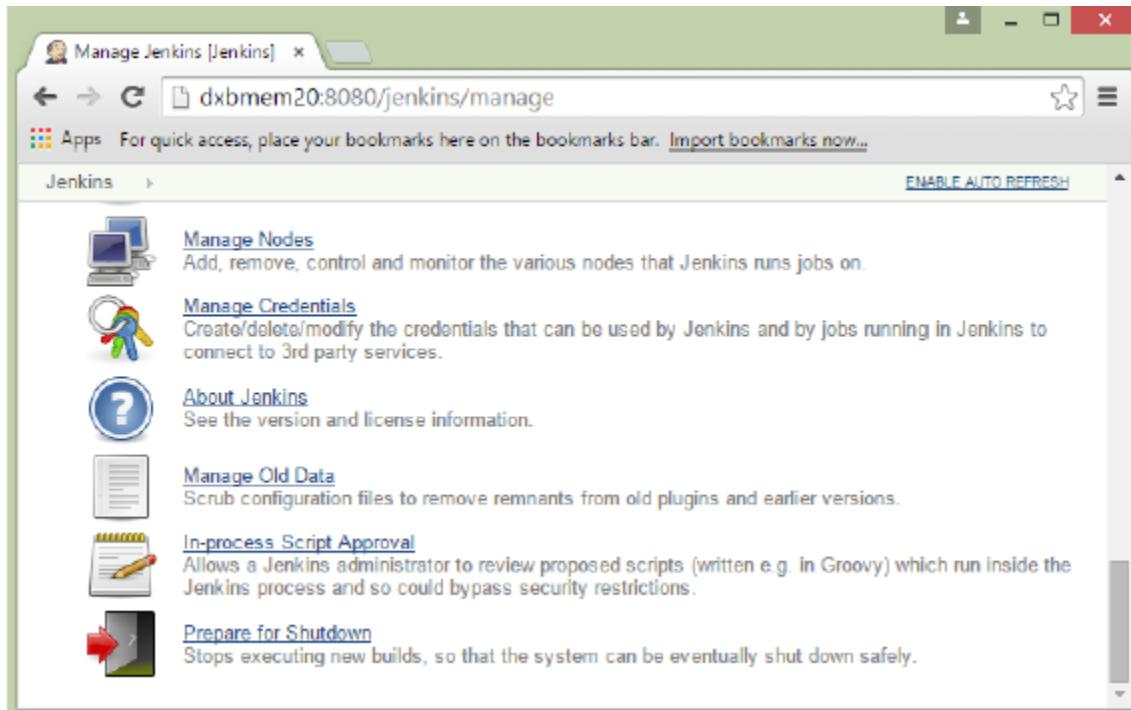


Jenkins - Remote Testing

Web tests such as selenium tests can be run on remote slave machines via the master slave and selenium suite plugin installation. The following steps show how to run remote tests using this configuration.

Step 1 – Ensuring your master slave configuration is in place. Got to your master Jenkins server. Go to Manage Jenkins → Manage Nodes.

Siva Jenkins Tutorial



In our node list, the DXBMEM30 label is the slave machine. In this example, both the master and slave machines are windows machines.

The screenshot shows the Jenkins Nodes list. It displays two nodes: 'DXBMEM30' (1 Idle) and 'master' (2 Idle). Below the list is a table providing detailed information about each node.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Sp
1	DXBMEM30	Windows Server 2012 (x86)	In sync	112.79 GB	4.54 GB	13 min
2	master	Windows Server 2012 (x86)	In sync	94.20 GB	3.85 GB	94.20
	Data obtained	13 min	13 min	13 min	13 min	13 min

Step 2 – Click on configure for the DXBMEM30 slave machine.

Siva Jenkins Tutorial

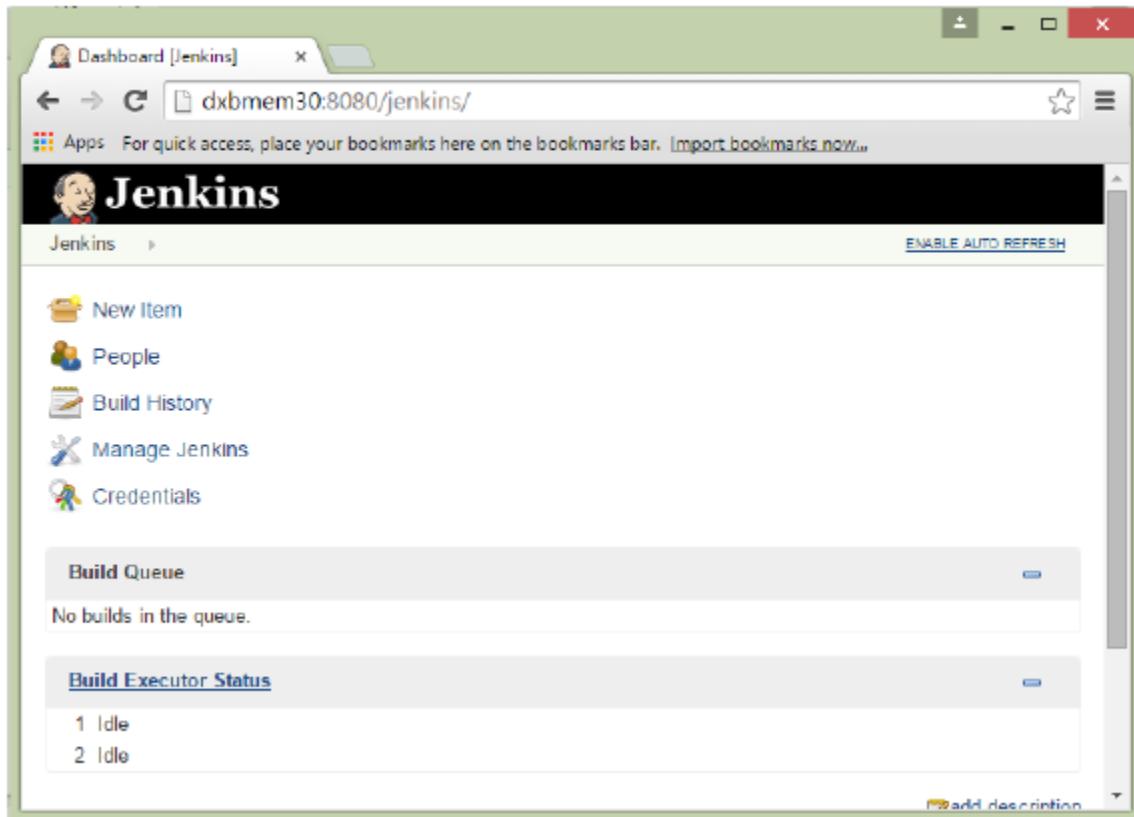
The screenshot shows the Jenkins 'Nodes' page. There are two nodes listed: 'DXBMEM30' and 'Idle'. The 'DXBMEM30' node is highlighted. A context menu is open over the 'DXBMEM30' node, with the 'Configure' option selected. Other options in the menu include 'Delete Slave' and 'Build History'. At the bottom right of the table, there is a 'Refresh status' button.

Step 3 – Ensure the launch method is put as ‘Launch slave agents via Java Web Start’

The screenshot shows the 'DXBMEM30 Configuration' page. The 'Launch method' dropdown is set to 'Launch slave agents via Java Web Start'. Other configuration options shown include Name (DXBMEM30), Description, # of executors (1), Remote root directory (C:\users\administrator EMIRATES\jenkins), Labels, Usage (Utilize this node as much as possible), and an Advanced... button. A Save button is at the bottom left.

Step 4 – Now go to your slave machine and from there, open a browser instance to your Jenkins master instance. Then go to Manage Jenkins → Manage Nodes. Go to DXBMEM30 and click on

Siva Jenkins Tutorial



Step 5 – Click on the DXBMEMP30 instance.

Siva Jenkins Tutorial

The screenshot shows the Jenkins 'Nodes' page. At the top, there's a header bar with icons for user profile, minimize, maximize, and close. Below it is a browser-style address bar with the URL 'dxbmem20:8080/jenkins/computer/'. The main content area has a breadcrumb navigation: 'Jenkins > nodes >'. To the right of the breadcrumb is a link 'ENABLE AUTO REFRESH' and a three-dot menu icon. A message 'No builds in the queue.' is displayed. Below this is a section titled 'Build Executor Status' with a collapse arrow. It lists two entries: 'master' and 'DXBMEM30 (offline)'. The 'master' entry shows '1 Idle' and '2 Idle'. The 'DXBMEM30' entry is marked as '(offline)'. Below this is a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, and Free Swap Space. The table contains two rows: one for 'DXBMEM30' (Windows Server 2012 (x86)) and one for 'master' (Windows Server 2012 (x86)). Both rows show 'In sync' under Clock Difference, and '45 min' under Data obtained. The 'Free Disk Space' and 'Free Swap Space' columns also show values for each node. At the bottom right of the table is a blue button labeled 'Refresh status'.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space
	DXBMEM30	Windows Server 2012 (x86)	In sync	112.79 GB	4.54 GB
	master	Windows Server 2012 (x86)	In sync	94.20 GB	3.85 GB
	Data obtained	45 min	45 min	45 min	45 min

Step 6 – Scroll down and you will see the Launch option which is the option to Start ‘Java Web Start’

Siva Jenkins Tutorial

The screenshot shows a web browser window with the URL `dxbmem20:8080/jenkins/computer/DXBMEM30/`. The page title is "DXBMEM30 [Jenkins]".

Stack Trace:

```
at org.jenkinsci.remoting.nio.NioChannelHub.run(NioChannelHub.java:561)
... 6 more
```

Connect slave to Jenkins one of these ways:

- Launch agent from browser on slave
- Run from slave command line:
`javaws http://localhost:8080/jenkins/computer/DXBMEM30/slave-agent.jnlp`
- Or if the slave is headless:
`java -jar slave.jar -jnlpUrl http://localhost:8080/jenkins/computer/DXBMEM30/slave-agent.jnlp`

Created by anonymous user

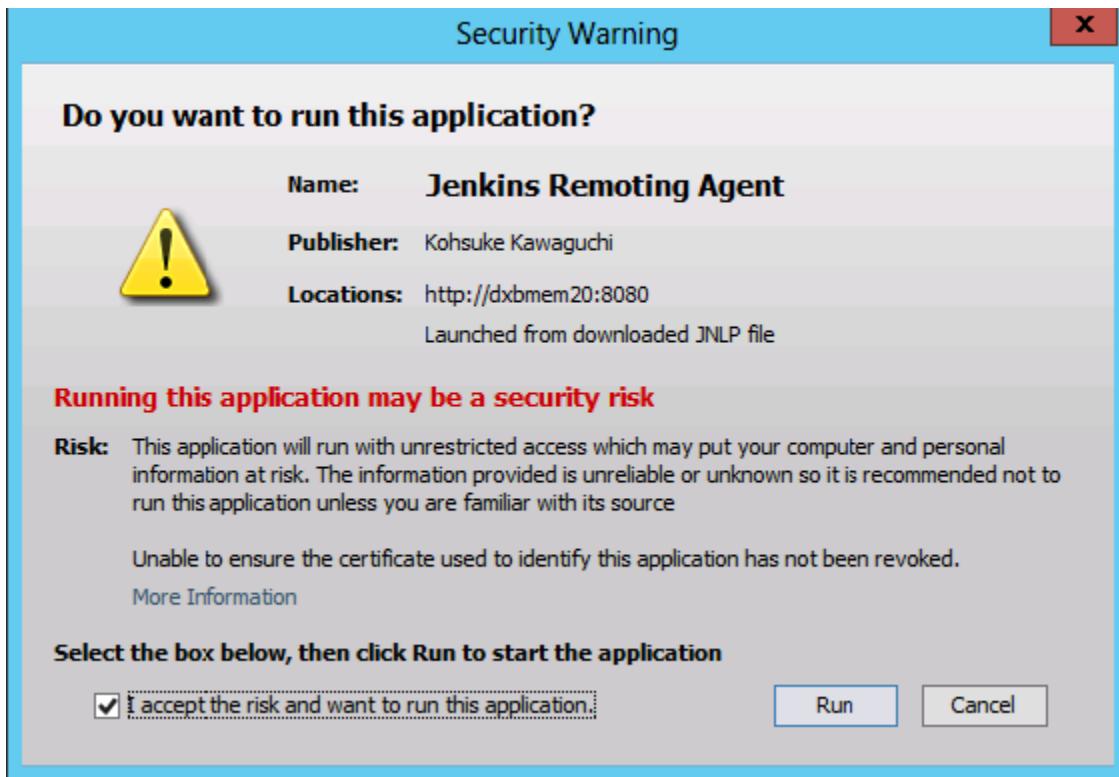
Projects tied to DXBMEM30

S	W	Name ↓	Last Success	Last Failure	Last Duration
		HelloWorld	43 min - #12	41 min - #13	7.3 sec

Icon: S M L [Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Step 7 – You will be presented with a Security Warning. Click on the Acceptance checkbox and click on run.

Siva Jenkins Tutorial



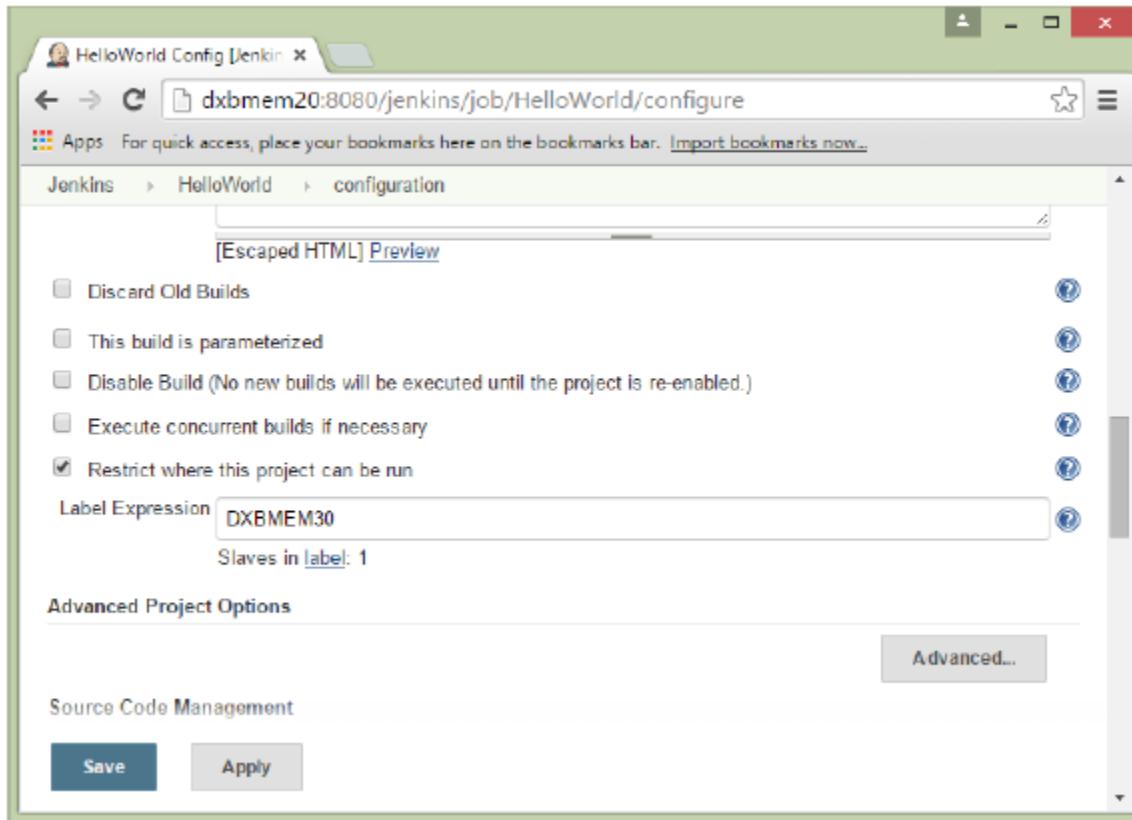
You will now see a Jenkins Slave window opened and now connected.



Step 8 – Configuring your tests to run on the slave. Here, you have to ensure that the job being created is meant specifically to only run the selenium tests.

In the job configuration, ensure the option ‘Restrict where this project can be run’ is selected and in the Label expression put the name of the slave node.

Siva Jenkins Tutorial



Step 9 – Ensure the selenium part of your job is configured. You have to ensure that the Sample.html file and the selenium-server.jar file is also present on the slave machine.

Siva Jenkins Tutorial

The screenshot shows the Jenkins configuration interface for the 'HelloWorld' job. The browser title is 'HelloWorld Config [Jenkins]'. The URL in the address bar is 'dxbmem20:8080/jenkins/job/HelloWorld/configure'. The page displays the 'SeleniumHQ html Suite Run' build step configuration. The configuration fields are:

- browser: firefox
- startURL: http://localhost:8080
- suiteFile: C:\Selenium\Sample.html
- resultFile: C:\Users\administrator.EMIRATES\jenkins\jobs\HelloWorld\workspace\Reports\Results.html
- other: (empty)

At the bottom, there are 'Delete', 'Save', and 'Apply' buttons.

Once you have followed all of the above steps, and click on Build, this project will run the Selenium test on the slave machine as expected.