Jenkins - Installation

Download Jenkins

The official website for Jenkins is [Jenkins](https://jenkins-ci.org/). If you click the given link, you can get the home page of the Jenkins official website as shown below.



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.



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

Webroot: $user.home/ .jenkins

Sep 29, 2015 4:10:46 PM winstone.Logger logInternal

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.



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) |

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](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

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 |

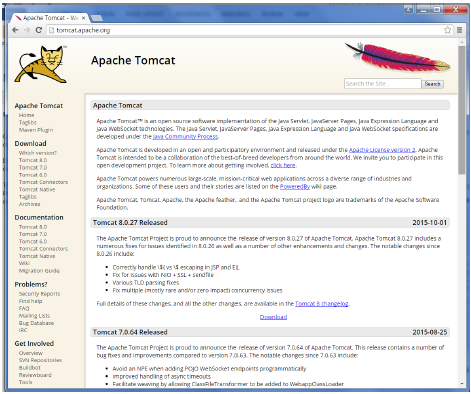
Append the full path of the Java compiler location to the System Path.

|  |  |
| --- | --- |
| **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/ |

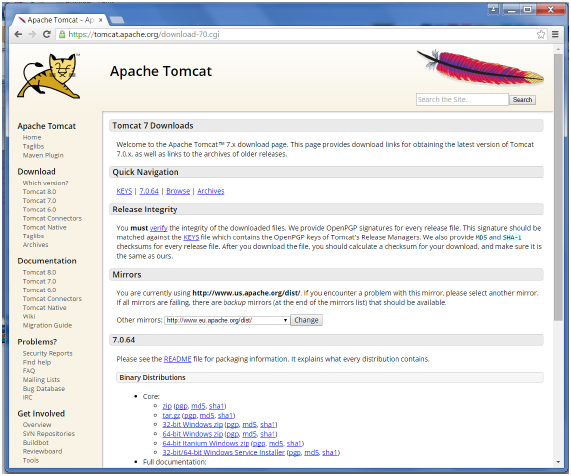
Verify the command java-version from command prompt as explained above.

Step 3: Download Tomcat

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



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



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.

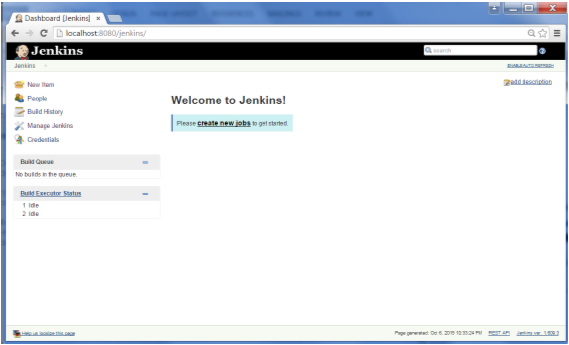
Now open the command prompt. From the command prompt, browse to the directory where the tomcat7 folder is location. Browse to the bin directory in this folder and run the start.bat file

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.

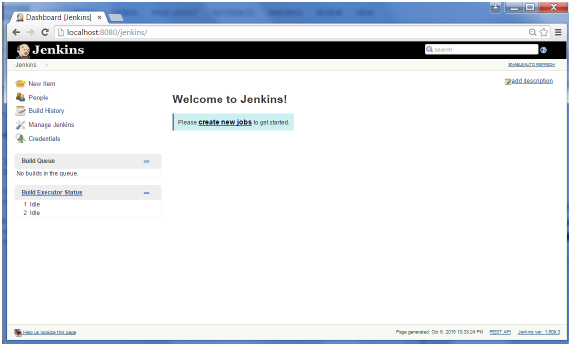
INFO: Server startup in 1302 ms

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.



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



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’



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



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



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.



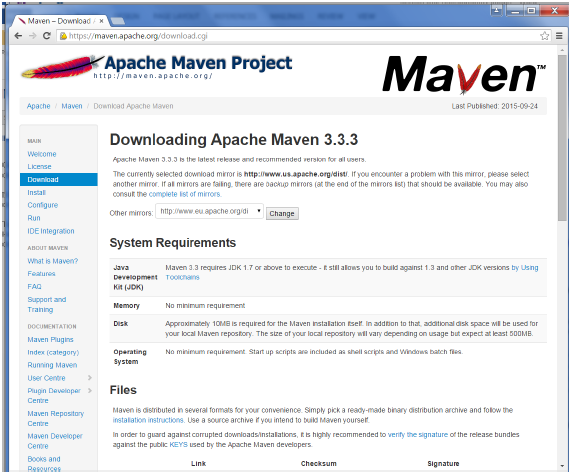
In the next screen, if you browse to the Source code Management section, you will now see ‘Git’ as an option.



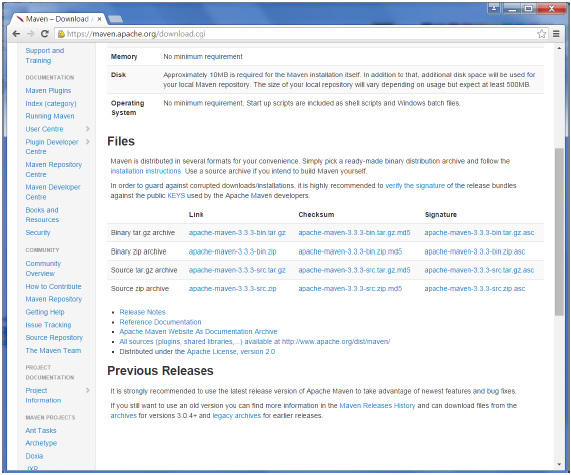
Jenkins – Maven Setup

Step 1: Downloading and Setting Up Maven

The official website for maven is [Apache Maven](https://maven.apache.org/download.cgi). If you click the given link, you can get the home page of the maven official website as shown below.



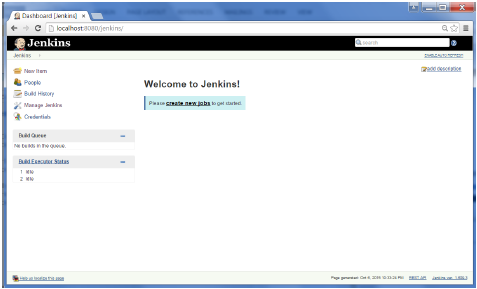
While browsing to the site, go to the Files section and download the link to the Binary.zip file.



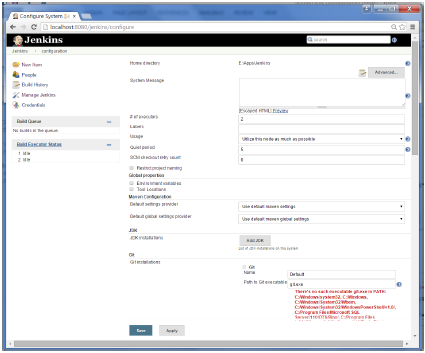
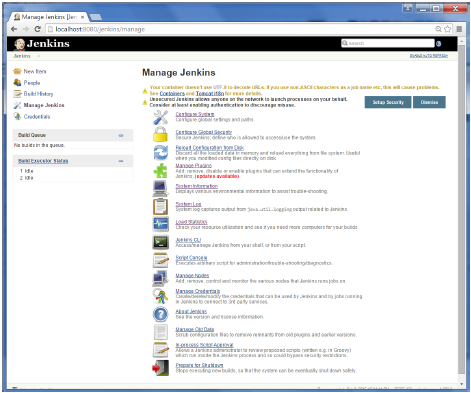
Once the file is downloaded, extract the files to the relevant application folder. For this purpose, the maven files will be placed in E:\Apps\apache-maven-3.3.3.

Step 2: Setting up Jenkins and Maven

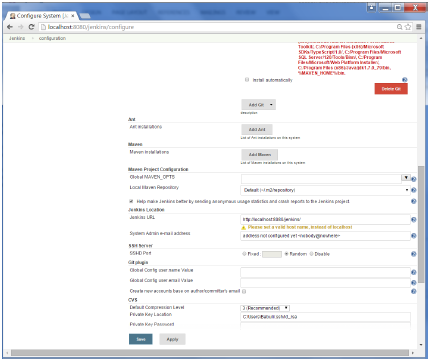
In the Jenkins dashboard (Home screen), click Manage Jenkins from the left-hand side menu.



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



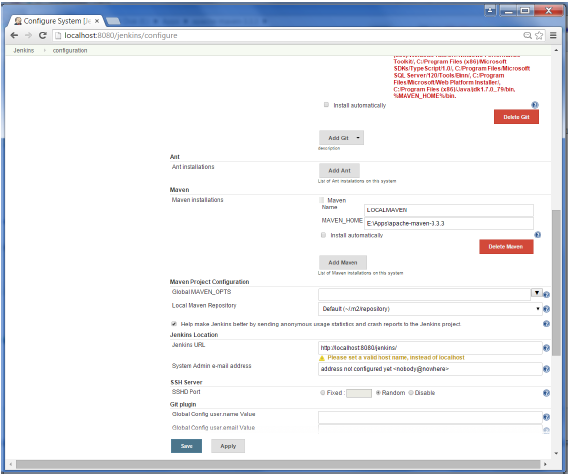
In the Configure system screen, scroll down till you see the Maven section and then click on the ‘Add Maven’ button.



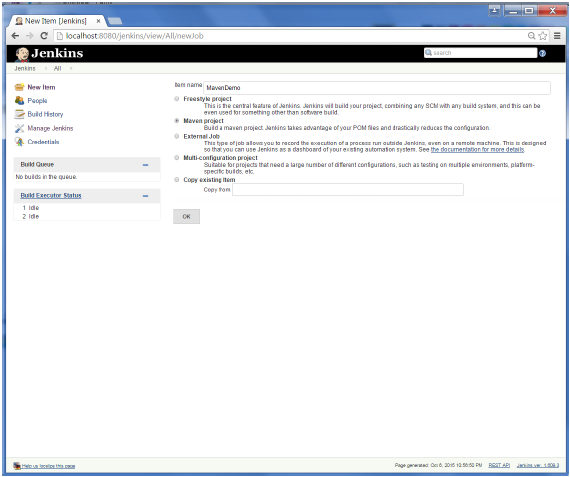
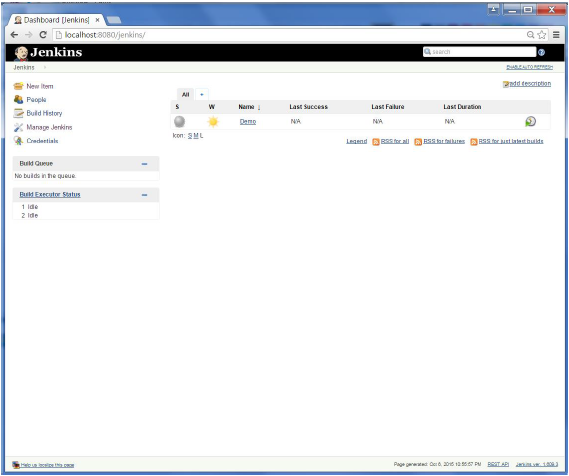
Uncheck the ‘Install automatically’ option.

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.



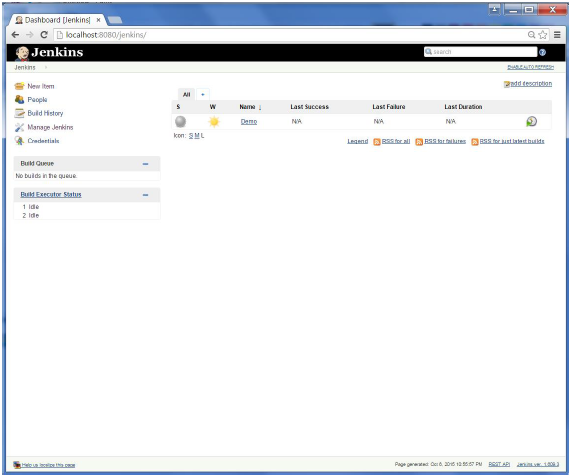
You can now create a job with the ‘Maven project’ option. In the Jenkins dashboard, click the New Item option.



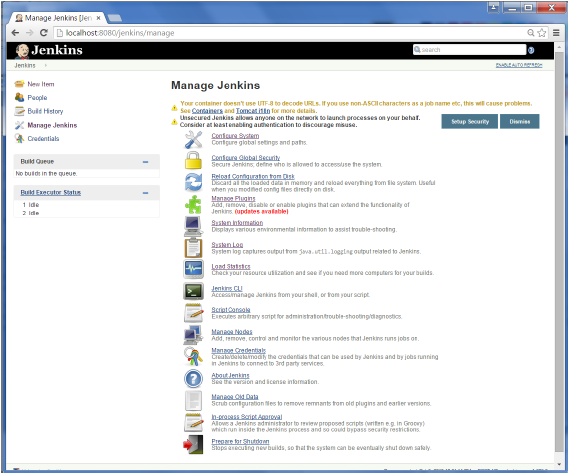
Jenkins - Configuration

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.



You will then be presented with the following screen −



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.

The following example will use the first option of setting the "JENKINS\_HOME" environment variable.

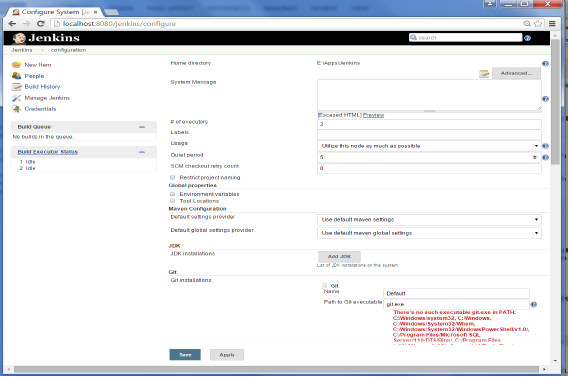
First create a new folder E:\Apps\Jenkins. Copy all the contents from the existing ~/.jenkins to this new directory.

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.



# 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 localhost. If you have a domain name setup for your machine, set this to the domain name else overwrite localhost 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}.

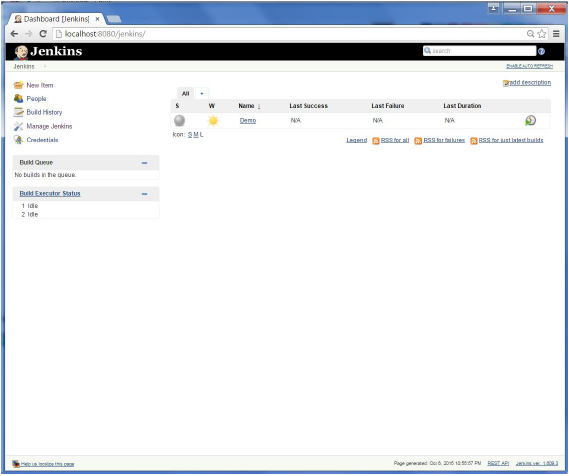
Email Notification

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.

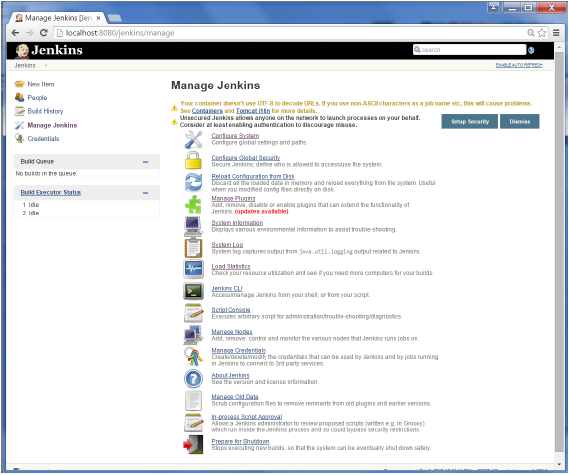
Jenkins - Management

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 −



Some of the management options are as follows −

Configure System

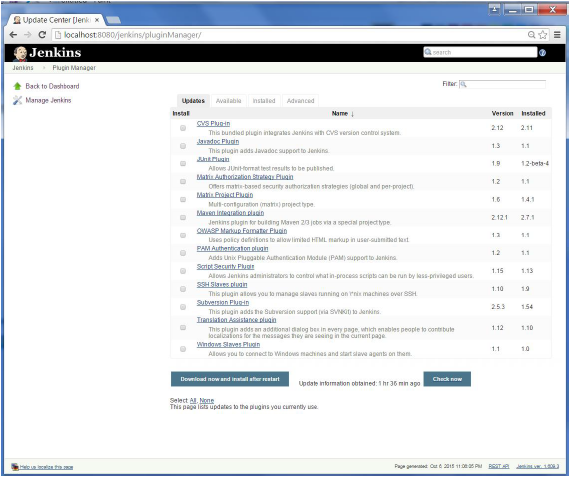
This is where one can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed. Jenkins will add the required configuration fields dynamically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

Manage Plugin

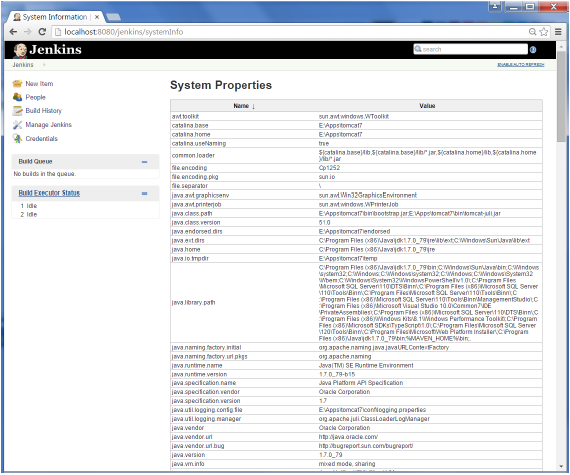
Here one can install a wide variety of third-party plugins right from different Source code management tools such as Git, Mercurial or ClearCase, to code quality and code coverage metrics reporting. Plugins can be installed, updated and removed through the Manage Plugins screen.



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.



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 pages 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.

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.