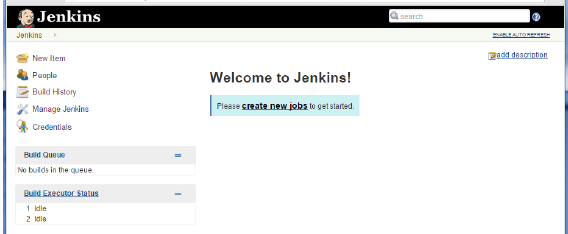
**Jenkins**

**Jobs**

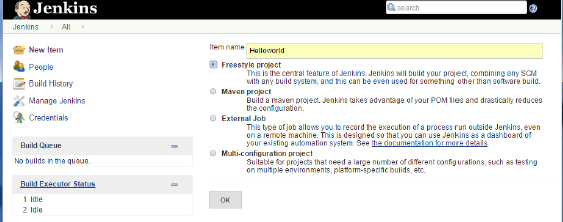
Jenkins job can be used to perform the typical build server work, such as doing continuous/official/nightly builds, run tests, or perform some repetitive batch tasks. This is called "free-style software project" in Jenkins.

**Steps to create a Job**

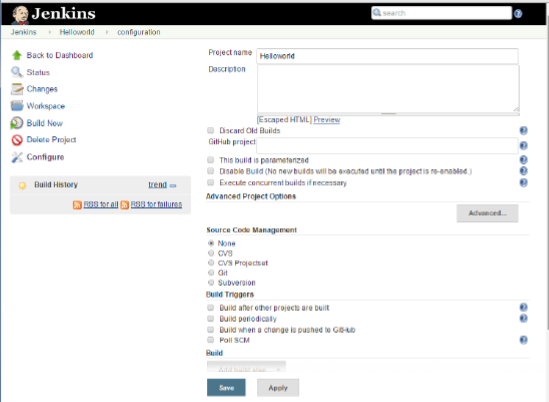
**Step-1** Go to the Jenkins dashboard and Click on New Item

****

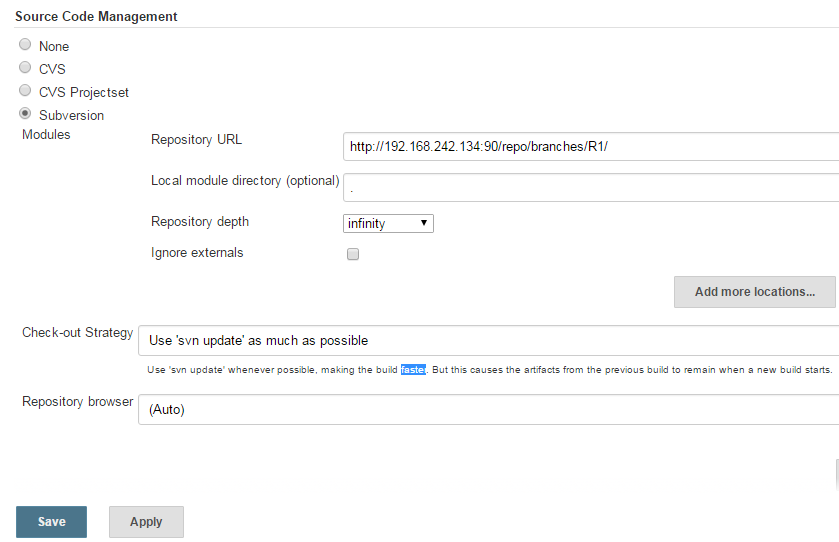
**Step-2** In the next screen, enter the Item name, in this case we have named it Helloworld. Choose the ‘Freestyle project option’

****

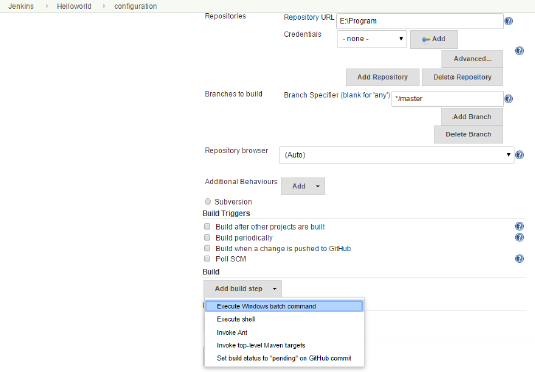
**Step-3** The following screen will come up in which you can specify the details of the job.

****

**Step-4** We need to specify the location of files which need to be built. In this example, we will assume that a local svn or git repository has been setup which contains a ‘HelloWorld.java’ file.



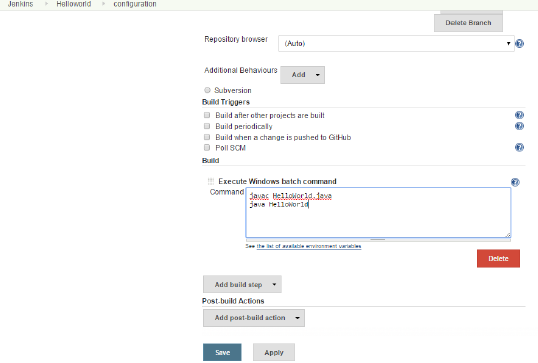
**Step 5** − Now go to the Build section and click on Add build step → Execute Windows batch command



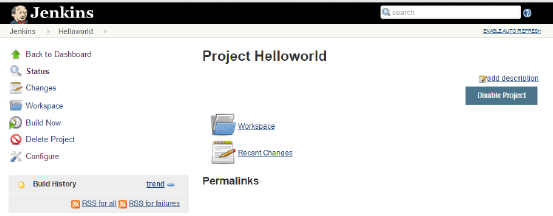
**Step 6** − In the command window, enter the following commands and then click on the Save button.

Javac HelloWorld.java

Java HelloWorld



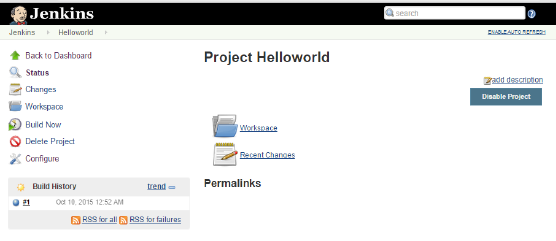
**Step 7** − Once saved, you can click on the Build Now option to see if you have successfully defined the job.



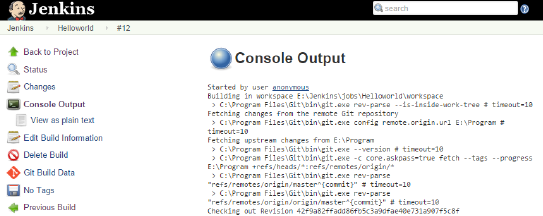
**Step 8-** Click on the build now button to run the job, then we can see the following Build history section shows that a build is in progress.



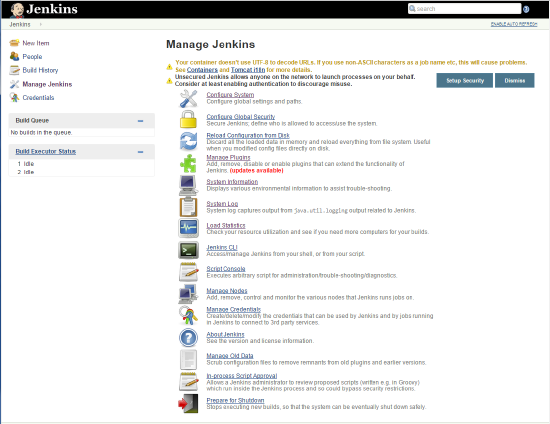
**Step 9** − Once the build is completed, a status of the build will show if the build was successful or not. In our case, the following build has been executed successfully. Click on the #1 in the Build history to bring up the details of the build.



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



**Jenkins Management**

**Step 1-** 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**

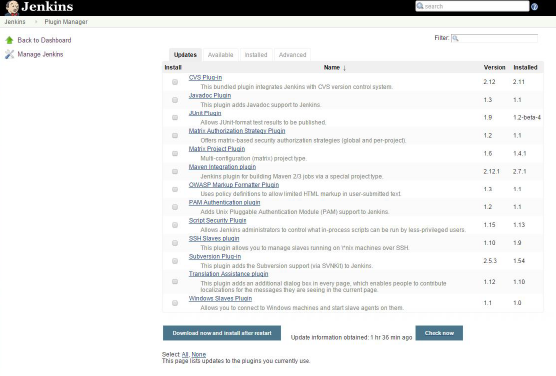
This is where one can manage paths to the various tools to use in builds, such as the JDKs, the versions of Ant and Maven, as well as security options, email servers, and other system-wide configuration details. When plugins are installed. Jenkins will add the required configuration fields dynamically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

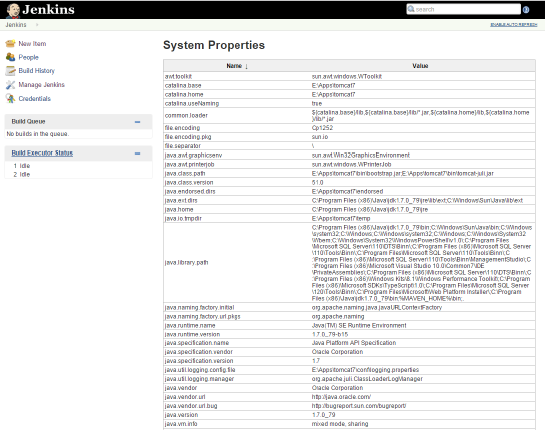
**Manage Plugin**

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.