**Jenkins**

Jenkins is a tool that allows continuous integration and continuous delivery. It can handle any type of build of a software project continuously. This will help developers to integrate the changes to project as quickly as possible and obtain fresh builds.

Jenkins is installed on server where central builds takes place.

Developers check source code

Jenkins pick up changed source code,trigger buildand run tests.



Output will be available in Jenkins dashboard and notifies developer.

**Jenkins workflow**

Continuous integration means building code into shared repository at regular intervals.

**Steps to install Jenkins :**

Download Jenkins 🡪 open cmd 🡪 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.

To access Jenkins: -

<http://localhost:8080> (This link will bring up the Jenkins dashboard.)

**For Jenkins tomcat setup :**

**Step-1 :** Install java. To verify java installed : **\>java -version** (windows)

**$ java –version** (Linux)

**Step-2 :** Setting java home.

**C:\Program Files\java\jdk1.7.0\_60** (windows)

**export JAVA\_HOME=/usr/local/java-current** (Linux)

**Step-3 :** Download Tomcat.

**Step-4 :** Jenkins and tomcat setup :-

* copy Jenkins.war to webapps folder in tomcat folder
* Open cmd -> E:\Apps\tomcat7\bin>startup.bat
* Open [**http://localhost:8080/jenkins**](http://localhost:8080/jenkins)

Jenkins will run on tomcat server.

In localhost Jenkins dashboard:

Manage Jenkins -> Manage plugins -> click available tab -> download Git Plugin.

**Jenkins Management :**

Jenkins dashboard -> manage jenkins

-> Configure System:- Manage paths of tools. Ex. JDK, Ant, Maven

-> Reload config. From disk:- Build jobs configurations are stored in XML file Jenkins Home Directives.

->Manage Plugins:- Install 3rd party plugins such as Git, Mercurial, ClearCase

(install, remove or update)

-> System Info.:- version of java, Jenkins,etc.

-> System Log:- To view Jenkins log file & use for troubleshooting.

-> Load Statistics:-Stats of the build in Jenkins through graphical data. ex. Length of build queue, waiting before executed,etc.

-> Script Console:- Runs groovy script on server.

-> Manage Nodes:- Running many nodes parallely and managing them.

-> Prepare for Shutdown:- When all the builds are finished then only Jenkins

Shut down.

**Jenkins Build Setup:-**

Jenkins Dashboard🡪 New Item🡪 Item Name 🡪Freestyle Project 🡪Specify location of file 🡪 Enter URL of local Git repository 🡪 Add Buil Setup 🡪 Execute windows Batch file🡪Enter command in Command window **(Javac** **Helloworld .java then Java Helloworld)🡪**Click Build Now 🡪Console output(To see details of build.)

**Jenkins Unit Testing:-**

Jenkins provide unit testing.

Click on existing project🡪Choose Configure Option 🡪Add Build Setup🡪Invoke Ant🡪Advance 🡪Build File => location of build.xml🡪Add post–built=>Publish Junit Test Result report🡪Enter location of XML(can be seen later)🡪Click Save🡪Build Now🡪Console Output🡪Test Result

**Jenkins Automated Testing:-**

Go to Manage Plugins🡪Install Hudson Selenium Plugin🡪Configure System🡪Configure Selenium jar file(Click Save)🡪Download Selenium standalone server🡪JD🡪Configure for Project 🡪Add Built Setup=>Selenium HQ html Suite Run🡪Add Details

Click save and execute build.

**Jenkins Notification:-**

This is to add email notification for a build project.

**For configuration SMTP server**:

Manage Jenkins🡪Configure System🡪Go to email verification 🡪Enter SMTP server & user email suffix🡪Add recipients to get email notification

This notification is for, if the build is completed, unstable, broken, etc.

**Jenkins Reporting:-**

There are many reporting plugins available with the simplest one being the reports available for Junit tests.

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 foundat <https://wiki.jenkinsci.org/display/JENKINS/Static+Code+Analysis+Plugins>.

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

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.

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:**

Manage Jenkins🡪Manage Nodes🡪Click on New Node🡪 Give a name for the node🡪choose the Dumb slave option🡪Click Ok🡪Enter the details of the node slave machine🡪 Click Save button

**Jenkins Automated Deployment:-**

Plugins can be used to transfer the build files after a successful build to the respective application/web server. On example is the “Deploy to container Plugin”. To use this follow the steps given below.

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(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 , Glassfish 2.x/3.x) , JBoss 3.x/4.x

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

**Jenkins Metrics & Trends:-**

 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

JD🡪Manage Jenkins🡪Manage Plugins🡪 Go to the Available tab and search for the plugin ‘Build History Metrics plugin’ and choose to ‘install without restart’🡪 Restart the Jenkins instance

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.

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’.

Steps for this are:-

JD🡪Manage Jenkins🡪Manage Plugins🡪 Go to the Available tab and search for the plugin ‘Hudson global-build-stats plugin’ and choose to ‘install without restart’🡪  Restart the Jenkins instance

Now to see global stats:-

JD🡪Manage Jenkins🡪’Global Build Stats’ Click the link🡪 Click on the ‘Initialize stats’(It gathers all the existing records for build which have already been carried out and charts can be created based on these results.)🡪Click on ’Create New Chart’ 🡪  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

A chart will be displayed and it will give you a drill down of the details of the job and their builds.

**Jenkins Server Maintenance:-**

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:

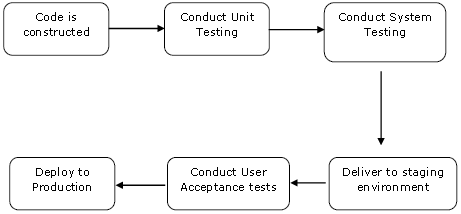
The Jenkins Home directory is the location on your drive where Jenkins stores all information for the jobs, builds etc. The location of your home directory can be seen when you click on Manage Jenkins → Configure system.

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.

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.



Software Development through Deployment

The main part of Continuous deployment is to ensure that the entire process is automated. Jenkins achieves all of this via various plugins, one of them being the “Deploy to container Plugin”.

 Create a simple project which emulates the QA stage, and does a test of the Helloworld application.

**Step-1:**

JD🡪 Click on New Item

**Step 2 :**

Choose ‘Freestyle Project’ & enter project name as ’QA’🡪 Click Ok🡪 Execute the program

**Step-3 :**

After setup build it to see if its builds properly🡪 Go to Helloworld project & click on Configure Option

**Step-4 :**

Choose ‘Add post-build action’🡪 Choose ‘Build other projects’

**Step-5 :**

In ‘Project to build’ section, enter QA as the project name to build. Leave option as default of ‘Trigger only if build is stable’🡪 Click on 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.)

**Step-7**

Now go to Manage Jenkins🡪Manage plugins🡪Search ‘Delivery Pipeline Plugins’ and 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.

**Step-9 :**

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

**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.
* 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.

**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**](https://wiki.jenkins-ci.org/display/JENKINS/Plugins)

**Uninstalling Plugins:**

Manage Jenkins🡪 Manage plugins🡪Click on the Installed tab🡪Click on uninstall option🡪Ensure to restart Jenkins instance.

**Installing another version of a plugin:**

Use **Upload** option to manually upload the plugin.

**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:-

**Step 1**:

Click on Manage Jenkins and choose the ‘Configure Global Security’ option

**Step 2:**

Click on Enable Security option. As an example, let’s assume that we want Jenkins to maintain it’s 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.

**Step 3**:

You will be prompted to add your first user.

**Step-4:**

Click on ‘Manage Users’.

**Step-5:**

Setup your authorizations🡪 Go to Manage Jenkins🡪Configure Global Security🡪 In Authorization section, click on ‘Matrix based security’.

**Step-6:**

Give appropriate permission to users🡪Click on Save.

Your Jenkins security is now setup.

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

**Jenkins Backup Plugins:**

Jenkins has a backup plugin which can used to backup critical configuration settings related to Jenkins.

Steps to backup plugins:-

**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.

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

**Step 4** − Click on Setup.

**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.

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

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

🡪  Click on Launch Restore to begin the restoration of the backup.