

The screenshot shows the Jenkins configuration interface for a job named 'first-jenkins-job'. The 'General' tab is selected. The 'Project name' field contains 'first-jenkins-job'. The 'Description' field is empty. Below these fields is a 'Advanced...' button. On the right side of the configuration panel, there is a sidebar with several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. Each checkbox has a question mark icon next to it.

→ Go to Build section => From drop down select shell => Enter some Linux commands “free -m” ; w => Save.

The screenshot shows the 'Build' configuration section. Under the 'Execute shell' heading, the 'Command' field contains the following two lines of shell script:  
# Display memory usage of the system  
free -m  
# Display logged in users  
w

→ After saving the project, we land up into the job's dashboard. Click Build Now to execute this project.

The screenshot shows the Jenkins job dashboard for 'first-jenkins-job'. The top navigation bar includes 'Jenkins' and 'first-jenkins-job'. Below the navigation are several links: 'Back to Dashboard' (with a green arrow icon), 'Status' (with a magnifying glass icon), 'Changes' (with a document icon), 'Workspace' (with a folder icon), 'Build Now' (with a circular progress icon), 'Delete Project' (with a red circle icon), and 'Configure' (with a gear icon).

After the execution is completed we can check the history of the executed jobs and check its output from it.

The screenshot shows the Jenkins 'Build History' page. At the top, there's a search bar labeled 'find'. Below it, a single build entry is listed: '#1' with a blue circular icon, followed by the date '05-Jun-2017 11:56'. At the bottom of the page are two RSS feed links: 'RSS for all' and 'RSS for failures'.

→ Click on the blue ball to see the output of the job.

The screenshot shows the Jenkins 'Console Output' page for build #1. On the left, there's a sidebar with links: 'Back to Project', 'Status', 'Changes', 'Console Output' (which is selected and highlighted in purple), 'View as plain text', 'Edit Build Information', and 'Delete Build'. The main content area has a title 'Console Output' with a blue circular icon. It displays the command-line output of the build. The output starts with 'Started by user Admin' and shows the execution of a shell script. It includes memory usage statistics (free, used, shared, buffers, cached), system load information (load average: 0.21, 0.06, 0.06), and a list of processes (USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT). The build concludes with 'Finished: SUCCESS'.

We have seen from above example that setting up a jenkins job is not such a challenging task. But you need to know what information goes up in the Jenkins job. In the next task, we will create an actual build job.

## 8. Setup a Java build job with Maven

We are going to use a publicly available java source code from github to test the build job.

<https://github.com/wakaleo/game-of-life.git>

This java source code can be built by Maven. It also has the pom.xml file which maven needs to build the code.

- ➔ Go to Jenkins main dashboard => Click New Item => Give a name to your job => Freestyle => OK

Jenkins

search

Enter an item name

first-maven-build

(Required field)

**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

Multibranch Pipeline

### Source Code Management

Copy the URL of gameoflife source code from github.

Go to source code management section in Jenkins and select on Git

Enter the game of life github url.

## Source Code Management

None  
Git

Repositories

Repository URL: <https://github.com/wakaleo/game-of-life.git>

Failed to connect to repository : Error performing command: git ls-remote -h https://github.com/wakaleo/game-of-life.git HEAD

Credentials: - none - Add

Advanced...  
Add Repository

Branches to build

Branch Specifier (blank for 'any'): \*/master

X  
Add Branch

This is a public repository so no credentials required and we are selecting master branch.

Create new file Upload files Find file Clone or download

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

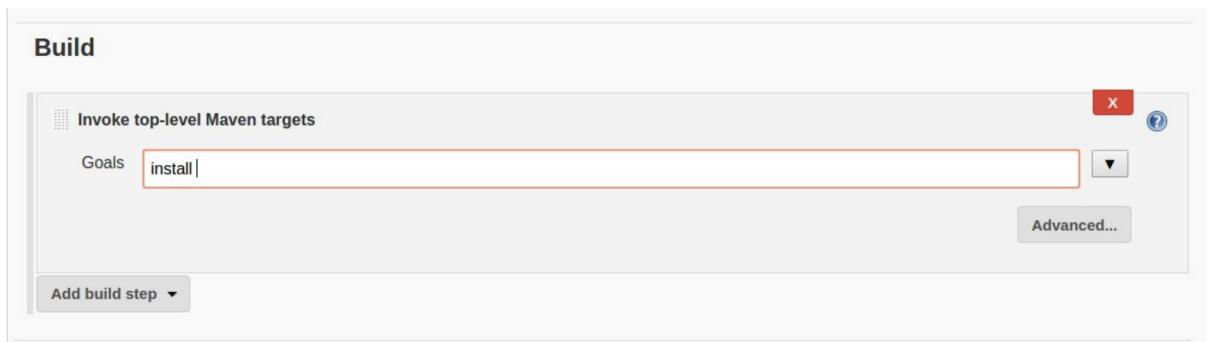
<https://github.com/wakaleo/game-of-life>

Download ZIP

5 years ago

3 months ago

- Go to build section => From drop down select Invoke top-level Maven targets  
=> In the goals give “install” => Save



→ Click on Build Now => In Build history click on the loading symbol or the blinking ball to see the runtime console output.

```

11/17 KB
14/17 KB
17/17 KB
17/17 KB

Downloaded: http://repo.maven.apache.org/maven2/org/apache/jackrabbit/parent/5/parent-5.pom (17 KB at 24.2 KB/sec)
4/13 KB
6/13 KB
8/13 KB
11/13 KB
13/13 KB

Downloaded: http://repo.maven.apache.org/maven2/org/apache/6/apache-6.pom (13 KB at 20.5 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/org/apache/jackrabbit/jcr-commons/2.2.5/jackrabbit-jcr-commons-2.2.5.pom (3 KB
4.8 KB/sec)

Downloaded: http://repo.maven.apache.org/maven2/org/apache/jackrabbit/jackrabbit-jcr-commons/2.2.5/jackrabbit-jcr-commons-2.2.5.pom (3 KB
4.8 KB/sec)
Download: http://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.6.1/slf4j-api-1.6.1.pom
3/3 KB

Downloaded: http://repo.maven.apache.org/maven2/org/slf4j/slf4j-api/1.6.1/slf4j-api-1.6.1.pom (3 KB at 5.0 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/org/slf4j/slf4j-parent/1.6.1/slf4j-parent-1.6.1.pom
4/10 KB
6/10 KB
8/10 KB
10/10 KB

Downloaded: http://repo.maven.apache.org/maven2/org/slf4j/slf4j-parent/1.6.1/slf4j-parent-1.6.1.pom (10 KB at 15.4 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/commons-httpclient/commons-httpclient/3.0/commons-httpclient-3.0.pom
4/8 KB
6/8 KB
8/8 KB
10/10 KB

Downloaded: http://repo.maven.apache.org/maven2/commons-httpclient/commons-httpclient/3.0/commons-httpclient-3.0.pom (8 KB at 13.0 KB/sec)
Downloaded: http://repo.maven.apache.org/maven2/junit/junit/4.8.2/junit-4.8.2.pom

```

As per the POM.xml of this project, this particular maven job will do below mentioned tasks.

- Download maven dependencies to build the job.
- Build the java source code
- Generate artefact
- Archive artefact
- Run unit test cases in the source code.

```
[INFO] Reactor Summary:  
[INFO]  
[INFO] gameoflife ..... SUCCESS [2:52.094s]  
[INFO] gameoflife-build ..... SUCCESS [1:04.176s]  
[INFO] gameoflife-core ..... SUCCESS [8.782s]  
[INFO] gameoflife-web ..... SUCCESS [2:39.458s]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 7:05.313s  
[INFO] Finished at: Mon Jun 05 12:30:58 UTC 2017  
[INFO] Final Memory: 34M/82M  
[INFO] -----  
Finished: SUCCESS
```

## Artefact of Game of life web application .

After the build process is completed, you can find the artefact for this job in the Workspace of this job.

### Workspace:

Workspace is the place where all the data of the job gets stored, e:g source code, artefacts etc. Every job in jenkins has its own workspace.

The screenshot shows the Jenkins interface for the project 'first-maven-build'. The top navigation bar shows 'Jenkins' and the project name 'first-maven-build'. The left sidebar contains links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. The main content area is titled 'Project first-maven-build' and displays two links: 'Workspace' (with a folder icon) and 'Recent Changes' (with a notepad icon).

→ Click on workspace => gameoflife-web => target => gameoflife.war

gameoflife.war is the artefact that got generated and archived by maven build process. This artefact can be deployed to the java web application server like tomcat or jboss server.

 <a href="#">gameoflife.war</a>	3.04 MB <a href="#">view</a>
 <a href="#">iacon exec</a>	17.95 KB <a href="#">view</a>

## 9. Jenkins Administration.

Once we had a little taste of jenkins and how to run jobs, we can understand how to administer Jenkins. Jenkins gives amazing features and its very flexible, we can setup jenkins as per our need. Jenkins can do variety of tasks apart from just being a CI server but we need to configure it as per our need. In coming section, we will see how jenkins is flexible and extensible.

You can open Jenkins admin settings by clicking on Manage Jenkins from main dashboard.



The screenshot shows the Jenkins main dashboard with a 'Manage Jenkins' link highlighted. Below it, the 'Manage Jenkins' page is displayed, listing various administrative options:

-  [Configure System](#)  
Configure global settings and paths.
-  [Configure Global Security](#)  
Secure Jenkins; define who is allowed to access/use the system.
-  [Configure Credentials](#)  
Configure the credential providers and types
-  [Global Tool Configuration](#)  
Configure tools, their locations and automatic installers.
-  [Reload Configuration from Disk](#)  
Discard all the loaded data in memory and reload everything from file system. Useful when you n
-  [Manage Plugins](#)  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
-  [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.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

There are variety of settings you can tinker with. We will dig into it one by one.

**Note:** Sometimes after making any config change you may need to restart Jenkins. In the browser, you can use below url to restart jenkins.

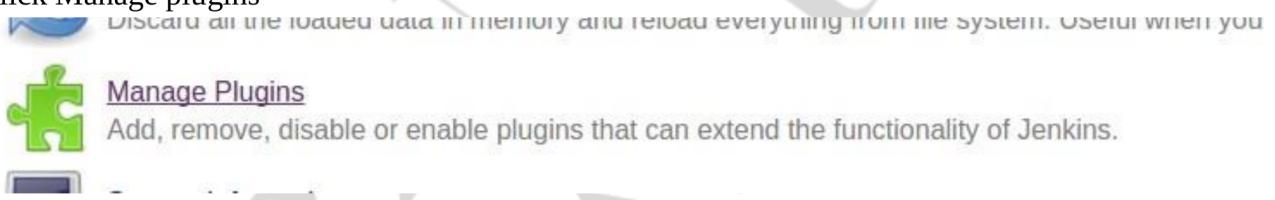
**http://<JenkinsIP>:8080/restart**

## Manage Plugins

Plugins are the most powerful feature of jenkins. You can customize jenkins as per your need by installing and setting up plugins. You can use jenkins to automate almost anything, it's just matter of the plugins you setup and there are wide variety of choices in plugins. We have already used some plugins in our build job like Git SCM, Invoke top level Maven target etc.

Some plugin comes by default installed in Jenkins and then you can install any plugin as per your choice and need.

Click Manage plugins



There are four tabs

### ➤ Update

If any plugin is outdated or a newer version of that plugin is available we can update the plugins from this tab.

### ➤ Available

List of available plugins to install. Find your plugin from filter, just put a checkmark on your favourite plugin and click "Install without restart". If the settings does not take effect restart jenkins server. Every plugin will have a wiki page, click on the plugin to read its wiki.

Install	Name	Version
<input type="checkbox"/>	Amazon EC2 Container Service plugin	1.11
<input type="checkbox"/>	Jenkins plugin to run dynamic slaves in a Amazon ECS/Docker environment	
<input type="checkbox"/>	ec2-cloud-axis	1.2
<input type="checkbox"/>	Amazon EC2 Container Service plugin with autoscaling capabilities	1.0
<input checked="" type="checkbox"/>	Amazon EC2 plugin	1.36
<input type="checkbox"/>	Deployment Dashboard Plugin for Jenkins	1.0.10
<input type="checkbox"/>	EC2 Fleet Jenkins Plugin	1.1.2
<input type="checkbox"/>	Support EC2 SpotFleet for Jenkins	
<input type="checkbox"/>	Amazon ECR plugin	1.6
	Integrate Jenkins with Amazon EC2 Container Registry (ECR)	

Install without restart   Download now and install after restart   Update information obtained: 21 hr ago   Check now

9, 961 824 5689 E-

Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## ➤ Installed

List of Installed plugin, if you choose to uninstall a plugin.

Put a check mark and click uninstall

Updates	Available	Installed	Advanced	
Enabled	Name	Version	Previously installed version	Uninstall
	Ant Plugin <input checked="" type="checkbox"/> Adds Apache Ant support to Jenkins	1.5		<button>Uninstall</button>

## ➤ Advanced

Some time you sit behind a proxy server and don't have a direct internet connection. That time you won't be able to see the list of Available plugin and won't be able to install it.

You can mention proxy settings in this page, restart Jenkins and then you will see the list of plugins to choose from.

If you are java programmer and have written your own plugin, you can upload your plugin to Jenkins in .hpi format.

Updates Available Installed **Advanced**

### HTTP Proxy Configuration

Server:

Port:

User name:

Password:

No Proxy Host:

**Submit** **Advanced...**

### Upload Plugin

You can upload a .hpi file to install a plugin from outside the central plugin repository.

File:  Choose file No file chosen

**Upload**

## Configure System



### Configure System

Configure global settings and paths.

Settings option in this page depends on the number of plugin you have. Many plugins need some global configuration settings which can be modified from this page. More number of plugins more settings you will see in this page.

From this page, we can change few Jenkins global settings like

- **Number of Executor:** Number of jobs that can parallelly run in Jenkins
  - If you have number of executor 2 and has initiated 3 jobs at the same time, then the third job will go in queue.
- **Environment variables & tools path can also be set.**
- **Email Notification:** SMTP server address and account details to send emails from Jenkins.

## Configure Global Security

Adding, removing and updating user and its permission can be handled from this page.

Configure Global Security

Enable security

TCP port for JNLP agents  Fixed :   Random  Disable

**Agent protocols...**

Disable remember me

**Access Control**

**Security Realm**

Delegate to servlet container  
 Jenkins' own user database  
 Allow users to sign up  
 LDAP  
 Unix user/group database

**Authorization**

Anyone can do anything  
 Legacy mode

### Security Realm

First, establish the user authentication method. For smaller, more informal installations, you can use Jenkins' own user database. For enterprise installations, you will want to use your corporate service, which allows users to log in to Jenkins with their usual username and password.

### Jenkins' Own User Database

#### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

This is the simplest authentication scheme--Jenkins maintains its own independent user database. People can sign up for their own accounts, and you as the administrator decide who can do what in Jenkins.

1.Select Jenkins' own user database

2>Place a check mark next to Allow users to sign up

3.Continue with Authorization, below. In particular, do not forget to press the Save button at the bottom of the page

### Active Directory on Linux Server

If Jenkins is running on a Windows server then it is better to install the Active Directory plugin.

On a Linux host you have an option to either use the Active Directory plugin or an LDAP based authentication. To configure the LDAP to work with Active Directory, provide the following:

<b>Server</b>	<i>mydomaincontroller.mycompnay.com:389</i>
<b>Root DN</b>	<i>dc=mycompnay,dc=com</i>
<b>User Search Filter</b>	<i>sAMAccountName={0}</i>
<b>Manager DN</b>	<i>cn=mymanageruser,ou=users,ou=na,ou=mycompany,dc=mycompany,dc=com</i>
<b>Manager Password</b>	<i>*****</i>

Note that the correct Manager DN value can vary greatly depending on your Active Directory set up.

### UNIX NIS

To set up Network Information System:

1.Go to the Jenkins dashboard, usually `http://_server_:8080` or `http://_server_/jenkins:8080`, where server is the host on which Jenkins is running

2>Select Manage Jenkins, then Configure Global Security

3.Click Enable Security. The page will expand to offer a choice of access control

4.Select Unix user/group database#\* Push the Test button (on the extreme right)

•If Success is displayed, everything is set up properly

•If not, follow the instructions to fix the problem and repeat

•If you still do not succeed, push the Advanced button and specify Service Name sshd and repeat

5.Continue with Authorization, below. In particular, do not forget to press the Save button at the bottom of the page.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

## LDAP

See LDAP Plugin. Then continue with Authorization, below. In particular, do not forget to press the Save button at the bottom of the page.

## Authorization

The Authorization section of the Configure Global Security page allows you to configure what users are allowed to do once authenticated.

### Matrix-based Security

Matrix-based security offers the most precise control over user privileges.

1. Select Matrix-based security as the Authorization
2. Give the Anonymous user only Overall Read access
3. In the text box below the matrix, type your user name (or the user name you plan to use when you register as a new Jenkins user) and click Add
4. Give yourself full access by checking the entire row for your user name
5. Repeat for other users who deserve full access. The configuration should look like the picture below:

User/group	Overall	Job				Run			View			SCM
	Administer	Read	Create	Delete	Configure	Build	Delete	Update	Create	Delete	Configure	Tag
Anonymous	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kohsuke	<input checked="" type="checkbox"/>											

User/group to add:

6. Click Save at the bottom of the page. You will be taken back to the top page. Now Jenkins is successfully secured.

7. Restart Jenkins (service Jenkins restart on Linux)

If you set up a service like NIS, Active Directory or LDAP, you can now log in to Jenkins using your network credentials. If you are using Jenkins' own user database, create a user account for yourself:

1.Click the Login link at the top right portion of the page

2.Choose Create an account

3.Specify the user name you used in the above step, and fill in the rest

If everything works smoothly, you are now logged on as yourself with full permissions. If something goes wrong, follow this to reset the security setting.

## Global Tool Configuration

Jenkins gets integrated with variety of tools. In the build job, we have seen it gets integrated with git and maven. Similarly, you can have other tools in your OS installed or you can install it from this page. Also, sometimes we need different versions of Java or Maven or Git etc. We can manage multiple versions of the tools from this page as well.

### Global Tool Configuration

**Maven Configuration**

Default settings provider

Default global settings provider

**JDK**

JDK installations  List of JDK installations on this system

**Git**

Git installations

<input type="checkbox"/> Git	Name <input type="text" value="Default"/>
	Path to Git executable <input type="text" value="git"/>
<input type="checkbox"/> Install automatically	

**Gradle**

Gradle installations  List of Gradle installations on this system

**Ant**

Ant installations  List of Ant installations on this system

**Maven**

## Installing Maven 2.2.1

**Maven**

Maven installations

<input type="checkbox"/> Maven	Name <input type="text" value="Maven-2.2.1"/>
<input checked="" type="checkbox"/> Install automatically	
<input type="checkbox"/> Install from Apache	Version <input type="text" value="2.2.1"/>

## Adding a Windows slave node.

1. On your master machine go to Manage Jenkins > Manage Nodes.

The screenshot shows the 'Manage Nodes' page in Hudson. At the top, there's a navigation bar with 'Hudson > nodes'. Below it is a toolbar with 'Back to Dashboard', 'New Node' (which is highlighted with a red box), and 'Configure'. To the right is a table with columns 'S', 'Name', 'Response Time', 'Free Swap Space', and 'Free Disk Space'. A single row is present with 'master' in the 'Name' column. Below the table is a 'Build Queue' section stating 'No builds in the queue.' and a 'Build Executor Status' table showing two entries, both marked as 'Idle'.

2. New Node --> Enter Node Name.

3. Select Dumb Slave --> Press OK.

The screenshot shows the 'New Node' configuration dialog. It has a 'Node name' field containing 'Jenkins Slave'. Below it is a radio button group where 'Dumb Slave' is selected (indicated by a blue circle). A tooltip explains: 'Adds a plain, dumb slave to Jenkins. This is called "dumb" because for example such as when you are adding a physical computer'. There is also an unselected option 'Copy Existing Node' with a 'Copy from' field. At the bottom is an 'OK' button.

4. Fill out the following:

- a. Set a number of executors (one or more) as needed.
- b. Set a Remote FS Root, a home directory for the master on the slave machine.
  - i. For a Windows slave, use something like: "C:\Jenkins\"
  - ii. TODO: add details.
- c. Select the appropriate Usage setting:
  - i. For an additional worker: Utilize this slave as much as possible
  - ii. For specialized jobs: Leave this machine for tied jobs only
- d. Launch Method:
  - i. An easy way to control a Windows slave is by using Launch slave agents via Java Web Start (Recommended for Windows)
  - ii. TODO: add steps for other methods.
- e. Availability --> Keep this slave online as much as possible
  - i. TODO: add details for each option.

f. Press OK.

Name	Jenkins Slave
Description	
# of executors	2
Remote FS root	C:\Jenkins\
Labels	
Usage	Leave this machine for tied jobs only
Launch method	Launch slave agents via Java Web Start
Availability	Keep this slave on-line as much as possible

**Node Properties**

Environment variables  
 Prepare jobs environment  
 Tool Locations

**Save**

5. Now you need to connect your slave machine to the master using the following steps.

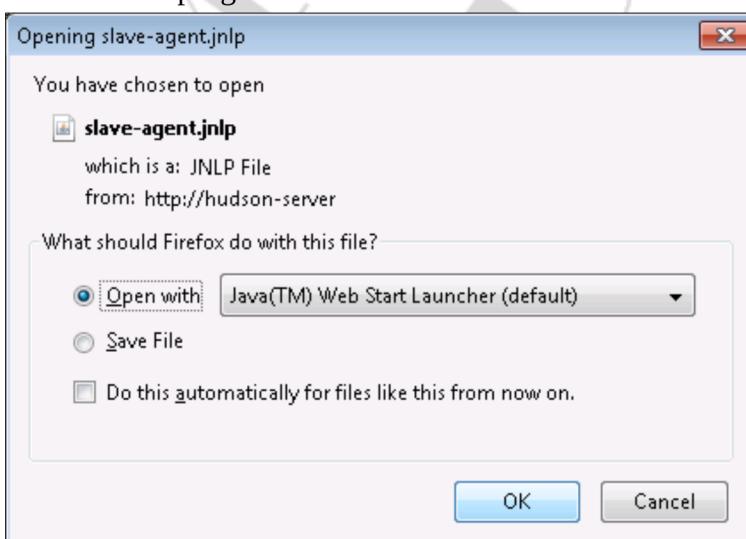
a. Open a browser on the slave machine and go to the Jenkins master server url (<http://yourjenkinsmaster:8080>).

b. Go to Manage Jenkins > Manage Nodes, Click on the newly created slave machine. You will need to login as someone that has the "Connect" Slave permission if you have configured global security.

c. Click on the Launch button to launch agent from browser on slave.



d. Run the program.



### If you encounter connection issue, then you could enlarge the popup windows to see the master port used and check your network configuration (firewall, port forward, ...)

#### Visualpath Training & Consulting.

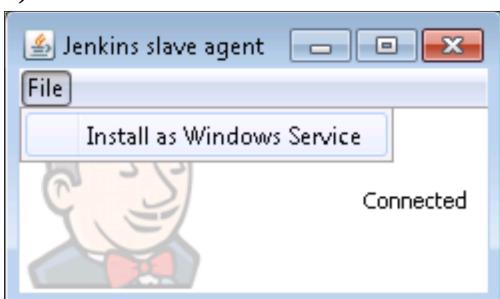
Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).



1. e. Now you should see the Slave machine connected under Nodes.

6. If you want the service to run on start-up of the slave machine do the following (Windows only directions):

- b) In the Slave agent program running on your slave machine,
- c) click File --> Install as Windows Service.



Note that this feature requires ".Net Framework 3.5"

#### .NET Framework 3.5 Features

- d) Start, type Services and Select the Services program.
- e) Find Jenkins Slave in the list, double click to open.
- f) Select Startup type --> Automatic.
- g) Go to the Log On tab, change the Log on as to a user of your choice (Special user account Jenkins recommended).
- h) Make sure that auto login is set for the slave machine for the user account, then the VM (or physical computer) should connect and be available when needed.

## Adding a Linux slave node.

First setup the node.

```
sudo apt-get install default-jre  
sudo mkdir /opt/jenkins  
sudo chown <username>:<groupname> /opt/jenkins -R
```

Manage nodes => New Node => node name => Permanent Agent/Dumb slave => OK

The screenshot shows the Jenkins 'Nodes' configuration page. A new node named 'linuxnode' is being created. The 'Permanent Agent' option is selected. The configuration includes:

- Name: linuxnode
- Description: (empty)
- # of executors: 1
- Remote root directory: /opt/jenkins
- Labels: testlinuxnode
- Usage: Use this node as much as possible
- Launch method: Launch slave agents via SSH
- Host: 192.168.1.9
- Credentials: vagrant\*\*\*\*\* (vagrantlinuxnode) - selected from a dropdown.
- Host Key Verification Strategy: Known hosts file Verification Strategy

At the bottom, there are 'Save' and 'Advanced...' buttons.

## 10. Continuous Integration Project.

We will setup a project that will build the artefact, version it and upload the versioned artefact to a Software Repository.

### Nexus

Software repository or repository managers are becoming very central part of Continuous Integration and Continuous Delivery projects.

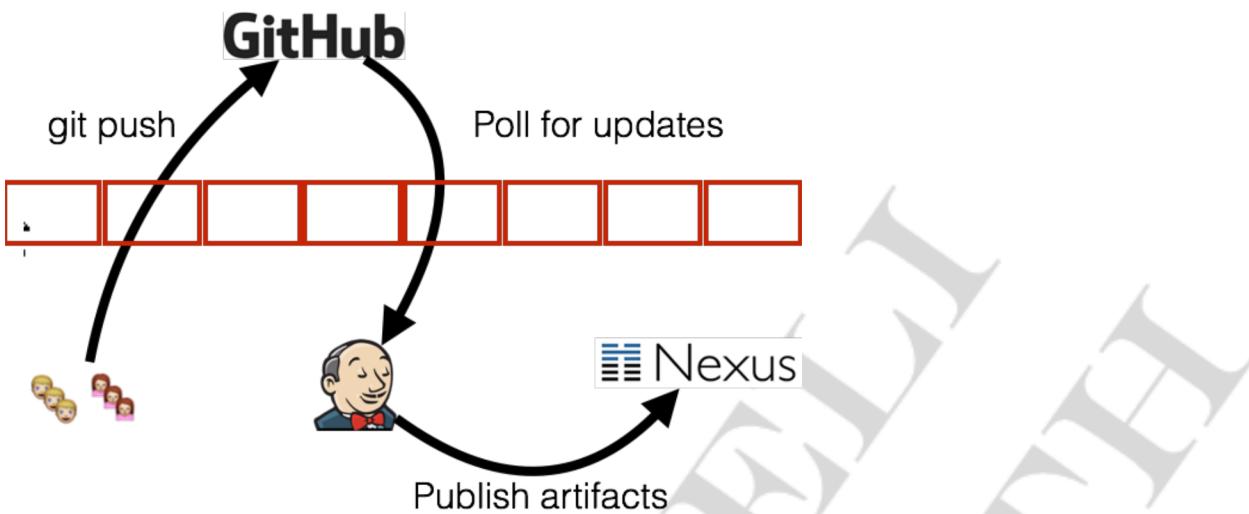
We have seen in our second build job, whenever we run the build job it will create gameoflife.war artefact. This artefact will get replaced every time we run the job.

If we generate an artefact that does not work or have any issues with it then we may need to go back to the previous version of the artefact. If we start versioning artefacts in Jenkins then we may fill up Jenkins disk space very quickly as these jobs runs several times in a day. For this we should have a mechanism of versioning and storing our versioned artefact to some centralised place.

For that very purpose we can use Nexus Repository Manager.

### Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).



There are other benefits to it. It gives a hosted repository so anybody with right credentials can download the artefact.

For example, from our deployment scripts we can select our artefact from Nexus and download it to a target location like tomcat server.

## Project Setup

### Jenkins Plugin Setup

#### Access Jenkins server from browser

<http://<jenkinsIP>:8080>

Install plugins:-

- Git plugin  
Checkout source code from github. Integrates Jenkins with git
- Zentimestamp plugin  
Creates variable named \$BUILD\_TIMESTAMP which can be used for versioning/naming our artifact.

After installing the plugin, we have to set its value from Configure System page.

Manage Jenkins => Configure System => Global properties.

Global properties	
<input checked="" type="checkbox"/> Date pattern for the BUILD_TIMESTAMP (build timestamp) variable	<input type="button" value="?"/>
Date and Time Pattern	<input type="text" value="yyyyMMddHHmm"/> <input type="button" value="?"/>

- Nexus plugin  
Uploads our versioned artifact to Nexus repository. Integrates Nexus with Jenkins.

## Create new build job

→ New item --> Enter project name --> Select freestyle project

The screenshot shows the Jenkins 'Create new build job' interface. In the 'Enter an item name' field, the value 'gol-mavenbuild' is entered. Below it, under 'Project type', 'Freestyle project' is selected. A tooltip explains: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.'

→ Select Git --> Enter GameOfLife git project URL  
(<https://github.com/wakaleo/game-of-life.git>)

The screenshot shows the 'Source Code Management' configuration for the 'gol-mavenbuild' job. Under 'Repositories', 'Git' is selected, and the 'Repository URL' is set to '<https://github.com/wakaleo/game-of-life.git>'. Under 'Branches to build', the 'Branch Specifier' is set to '\*/master'. There are 'Advanced...' and 'Add Repository' buttons.

→ Add Build step --> Invoke top level maven project --> In Goals enter "install"

The screenshot shows the 'Build' configuration for the 'gol-mavenbuild' job. It contains a single step: 'Invoke top-level Maven targets' with 'Goals' set to 'install'. There is an 'Advanced...' button.

Save --> Build Now.

## Build verification.

In your project's dashboard => Go to the workspace => [gameoflife-web](#) => [target](#)

You should see gameoflife.war.

## Nexus setup

We will setup nexus server on Centos in this tutorial.

Create a centos vm or cloud instance and login to it.

Follow below steps to setup Nexus

## Visualpath Training & Consulting.

Flat no: 205, Nilgiri Block, Aditya Enclave, Ameerpet, Hyderabad, Phone No: - +91-970 445 5959, 961 824 5689 E-Mail ID : [online.visualpath@gmail.com](mailto:online.visualpath@gmail.com), Website : [www.visualpath.in](http://www.visualpath.in).

```

sudo -i

yum install -y java-1.8.0-openjdk.x86_64 vim wget

export RUN_AS_USER=root

wget http://www.sonatype.org/downloads/nexus-latest-bundle.tar.gz

sudo cp nexus-latest-bundle.tar.gz /usr/local/

cd /usr/local

sudo tar xvzf nexus-latest-bundle.tar.gz

sudo ln -s <nexus directory name> nexus

/usr/local/nexus/bin/nexus start

service iptables stop

```

## Accessing Nexus dashboard

- ➔ From browser hit URL <Nexus server IP>:8081/nexus.
- ➔ Click login button and enter the credentials. (admin/admin123)
- ➔ Create hosted repository named “gol-repo” with all default settings

The screenshot shows the Nexus Repository Manager OSS interface. The main window displays a list of repositories under 'User Managed Repositories'. The columns include Repository, Type, Health Check, Format, Policy, Repository Status, and Repository Path. The list includes entries for Public Repositories, 3rd party, Apache Snapshots, Central, Central M1 shadow, Releases, and Snapshots. Below the main window, a modal dialog is open with the title 'Add...'. It contains four options: 'Hosted Repository' (which is selected and highlighted in blue), 'Proxy Repository', 'Virtual Repository', and 'Repository Group'. The background of the main window shows a large watermark reading 'VISUALPATH' diagonally across the screen.

Repository	Type	Health Check	Format	Policy	Repository Status	Repository Path
Public Repositories	group	<span>ANALYZE</span>	maven2	Release	In Service	http://192.168.1.9:8081/nexus/content/groups/public
3rd party	hosted	<span>ANALYZE</span>	maven2	Release	In Service	http://192.168.1.9:8081/nexus/content/repositories/thirdparty
Apache Snapshots	proxy	<span>ANALYZE</span>	maven2	Snapshot	In Service	http://192.168.1.9:8081/nexus/content/repositories/apache-snapshots
Central	proxy	<span>ANALYZE</span>	maven2	Release	In Service	http://192.168.1.9:8081/nexus/content/repositories/central
Central M1 shadow	virtual	<span>ANALYZE</span>	maven1	Release	In Service	http://192.168.1.9:8081/nexus/content/shadows/central-m1
Releases	hosted	<span>ANALYZE</span>	maven2	Release	In Service	http://192.168.1.9:8081/nexus/content/repositories/releases
Snapshots	hosted	<span>ANALYZE</span>	maven2	Snapshot	In Service	http://192.168.1.9:8081/nexus/content/repositories/snapshots