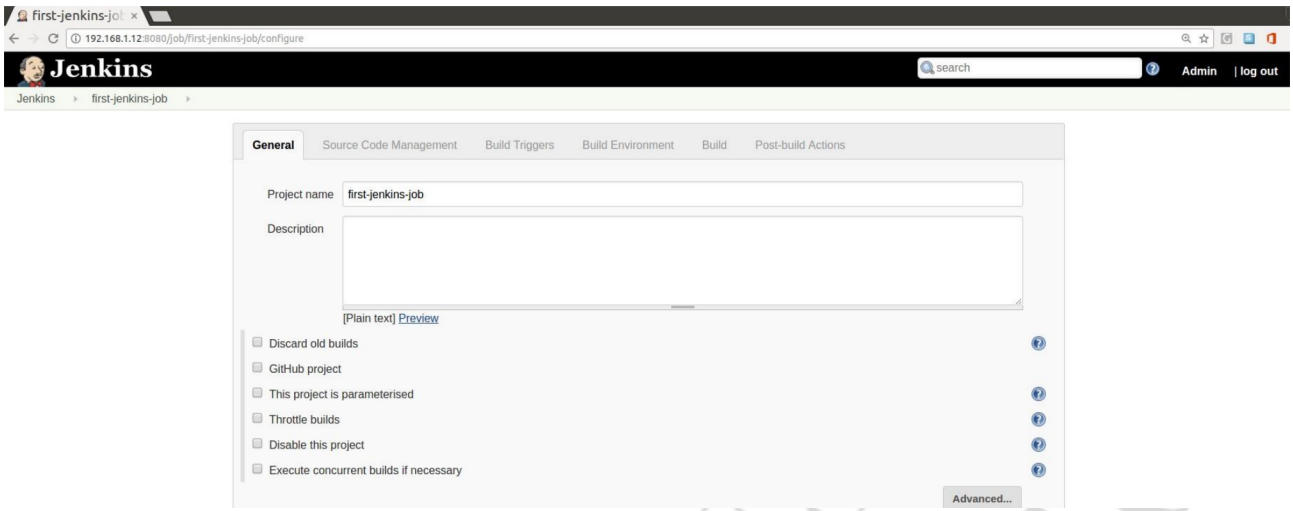
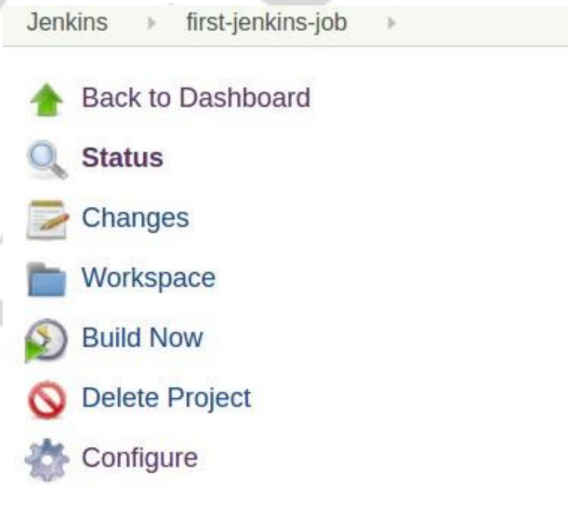


➔ Go to Build section => From drop down select shell => Enter some Linux commands “free -m” ; w => Save.

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



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

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

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.

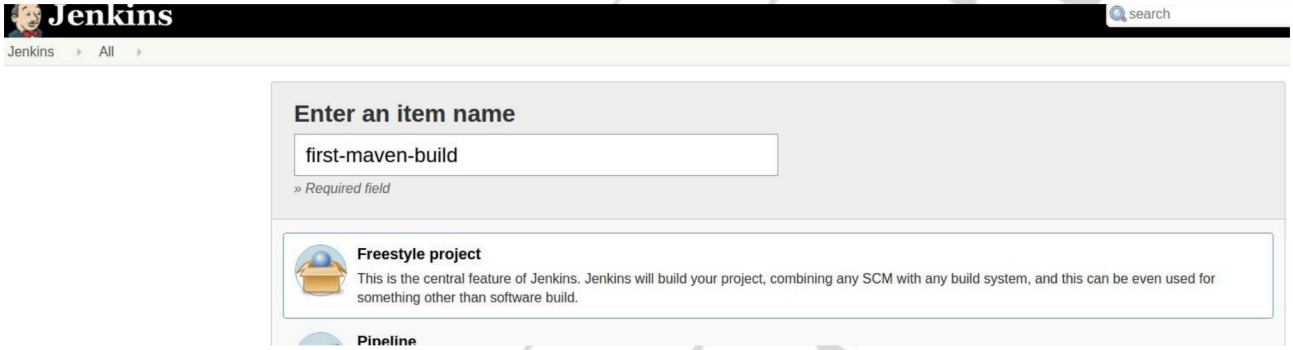
8. Setup a Java build job with Maven

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

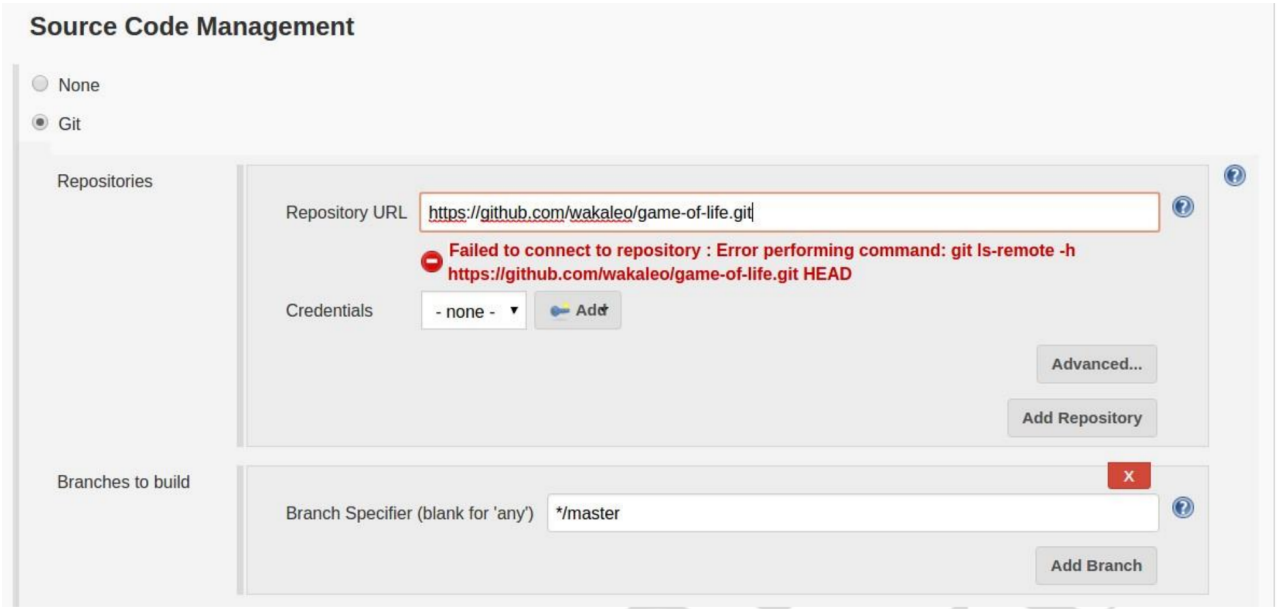


Source Code Managment

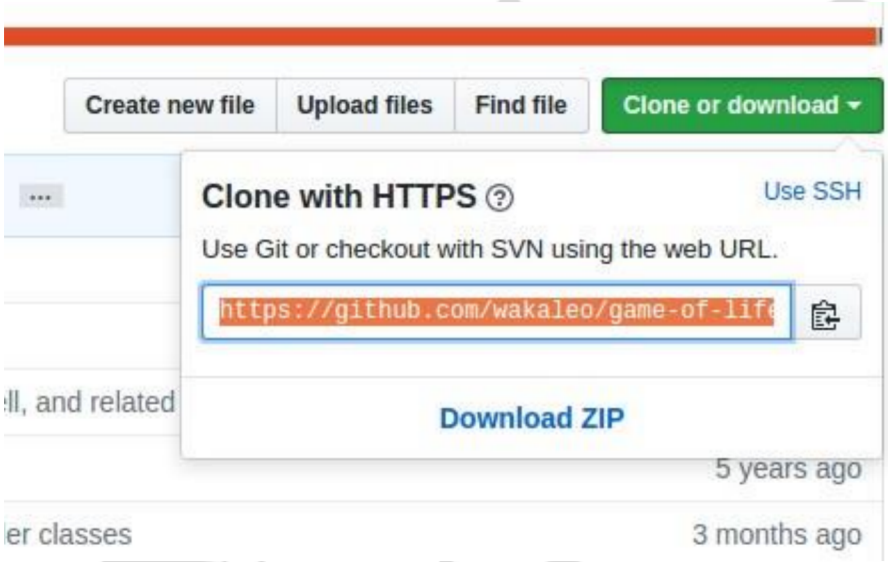
Copy the URL of gameoflife source code from github.

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

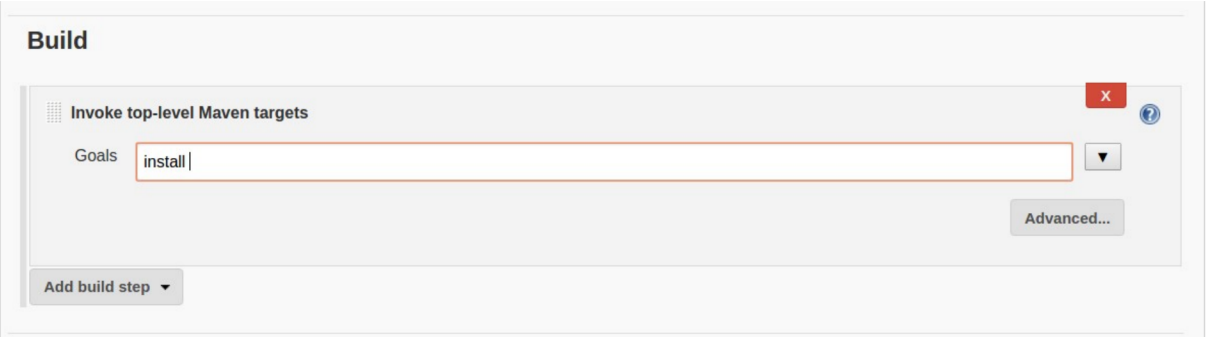
Enter the game of life github url.



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



➔ 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.

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.

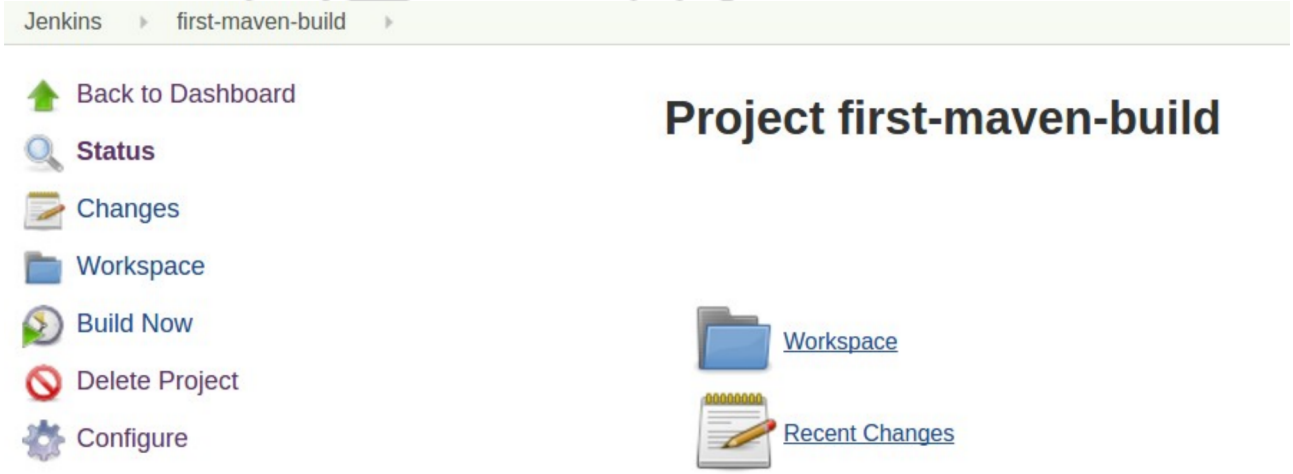
```
[INFO] Reactor Summary:
[INFO]
[INFO] gameoflife ..................................... SUCCESS [2:52.094s]
[INFO] gameoflife-build ............................... SUCCESS [1:04.176s]
[INFO] gameoflife-core ................................ SUCCESS [8.782s]
[INFO] gameoflife-web ................................. 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.



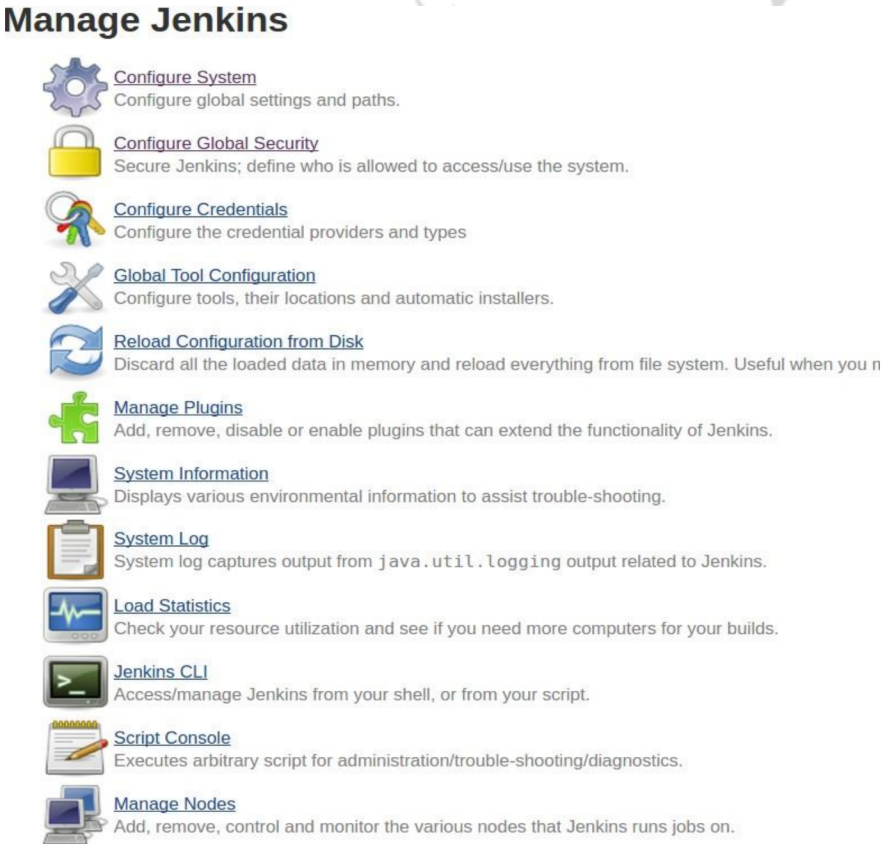
➔ Click on workspace => gameoflife-web => target => gameoflife.war

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.

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.



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.

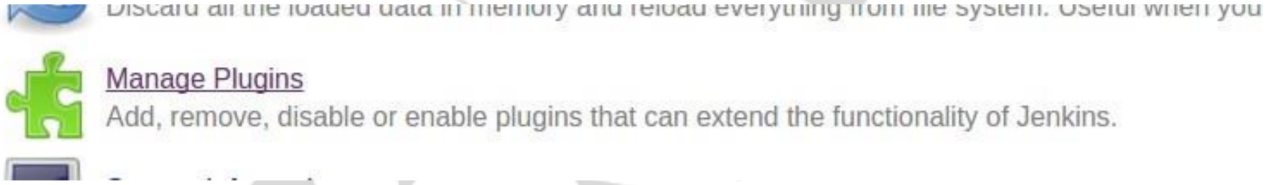
http://<JenkinsIP>:8080/restart

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.

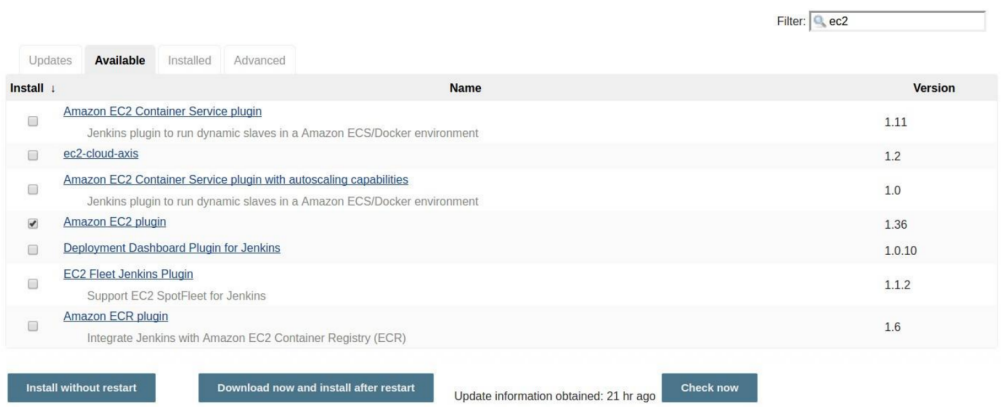
Click Manage plugins



There are four tabs

- **Update**
  If any plugin is outdated or a never version of that plugin is available we can update the plugins from this tab.

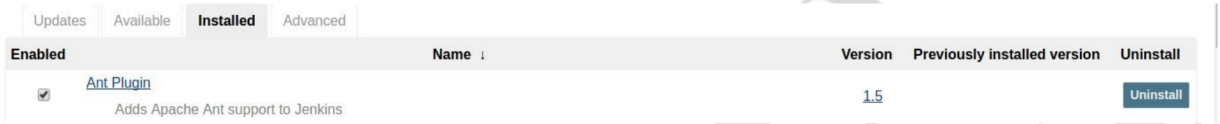
- **Available**
  List of available plugins to install. Find your plugin from filter, just put a checkmark on your favourite plugin and click "Install without restart". If the settings does not take effect restart jenkins server. Every plugin will have a wiki page, click on the plugin to read its wiki.



- **Installed**

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

  Put a check mark and click uninstall

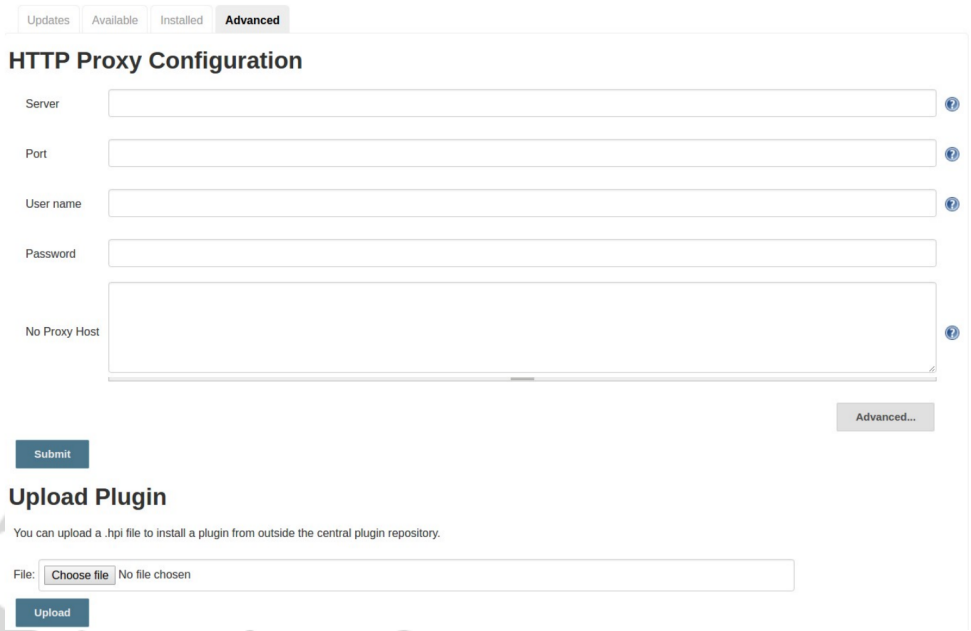


- **Advanced**

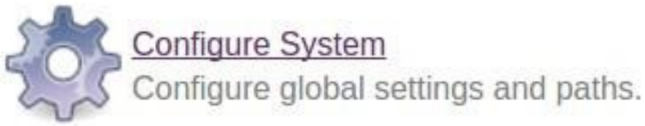
  Some time you sit behind a proxy server and don't have a direct internet connection. That time you won't be able to see the list of Available plugin and won't be able to install it.

  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.



Configure System



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

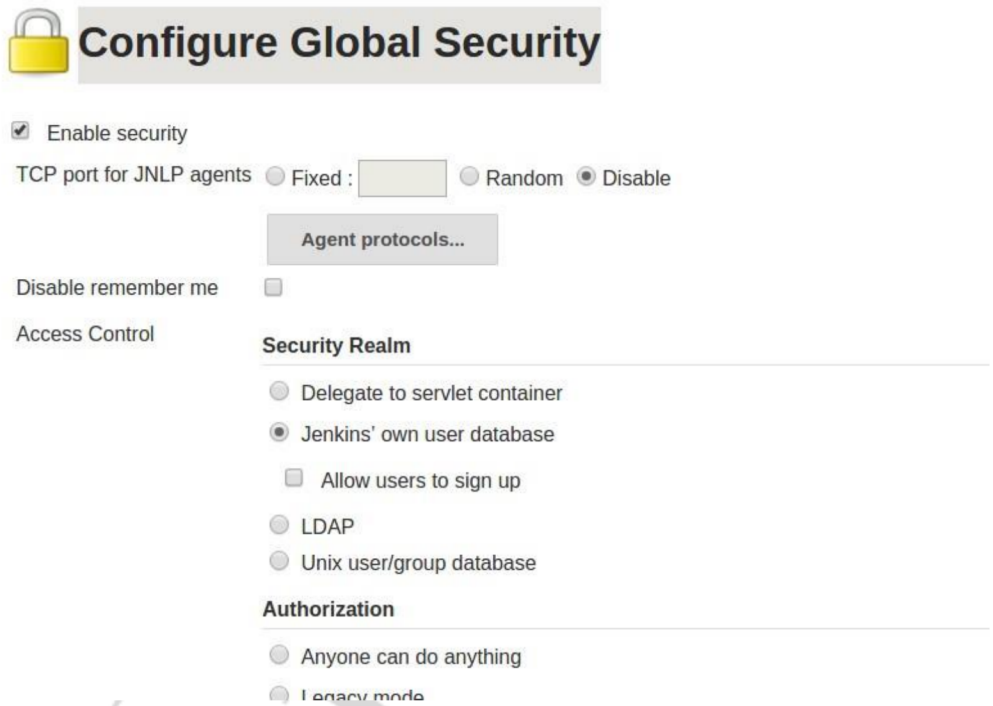
- **Number of Executor:** Number of jobs that can parallelly run in Jenkins

  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.



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

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:

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

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.

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.

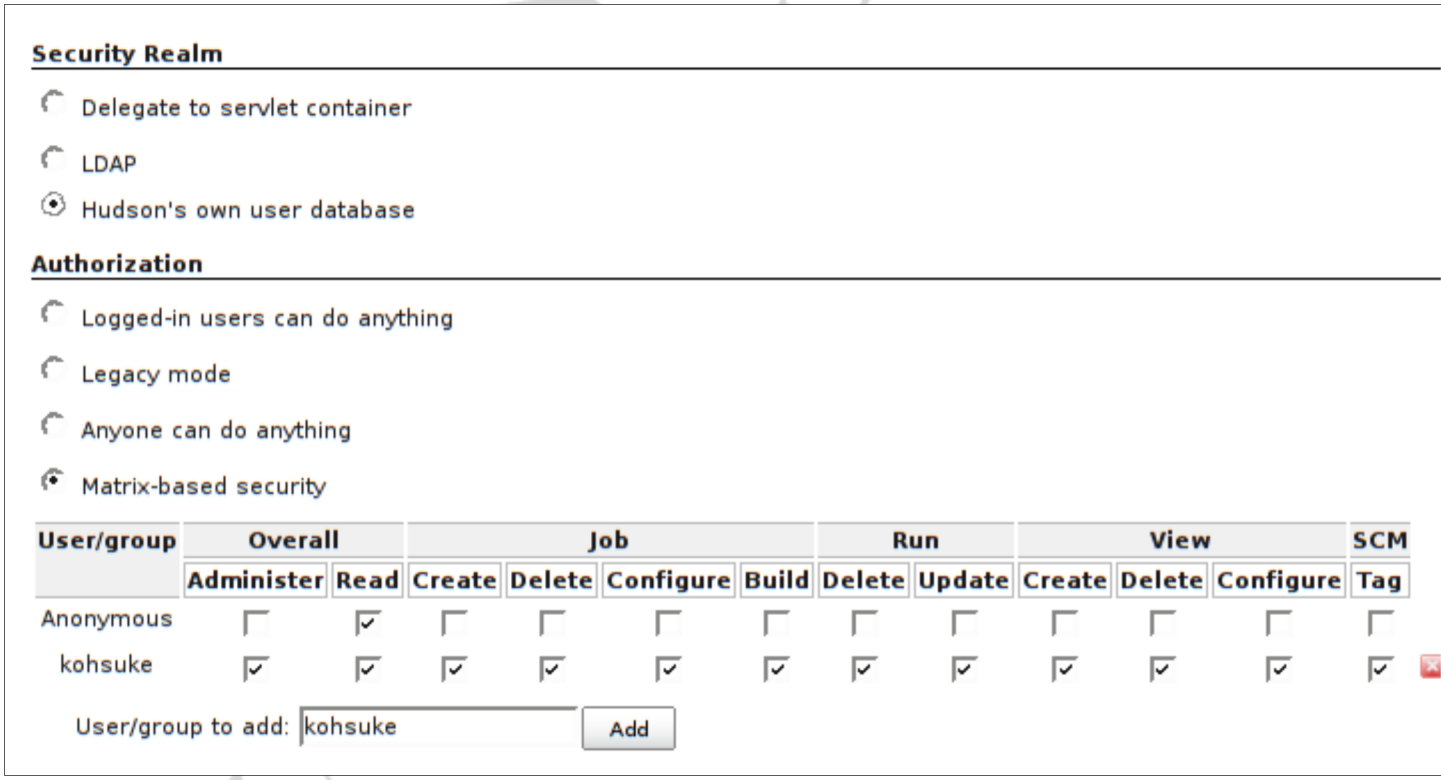
1.Select Matrix-based security as the Authorization

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:



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

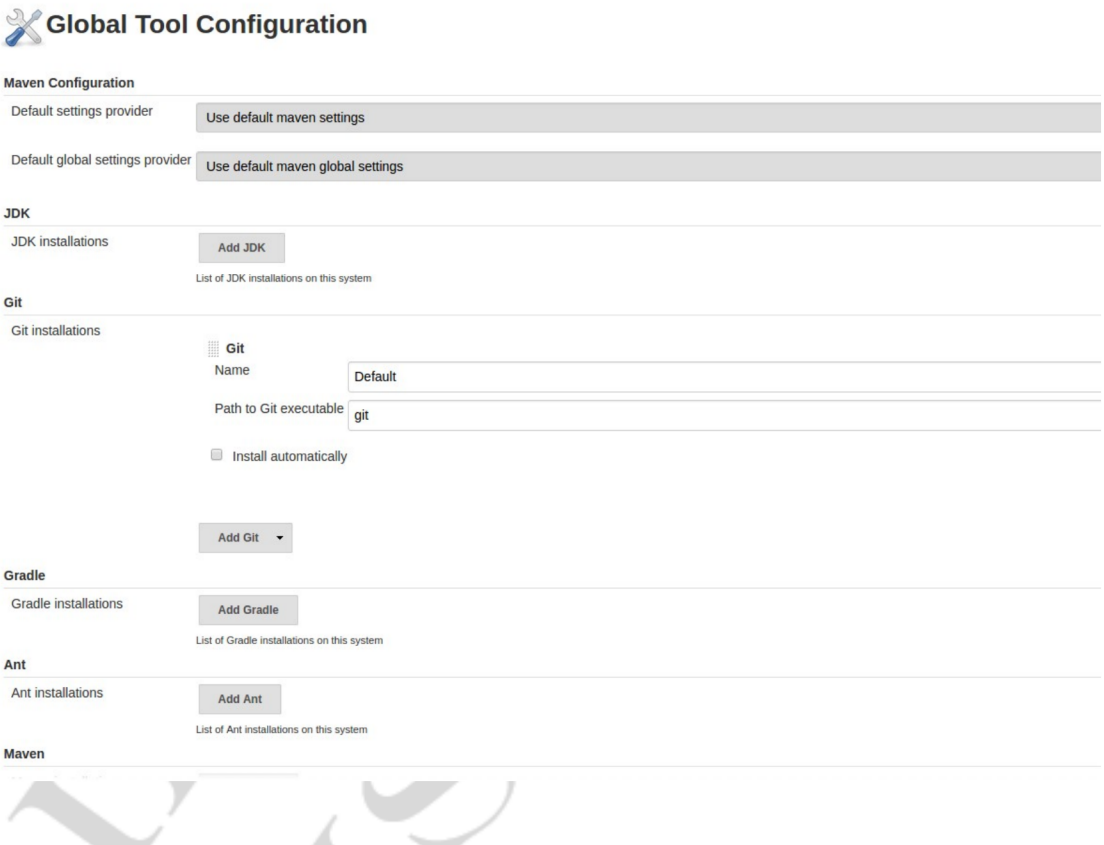
2.Choose Create an account

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

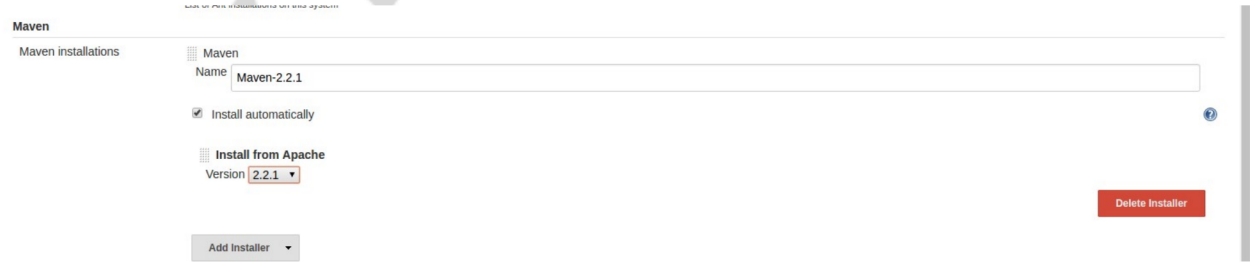
If everything works smoothly, you are now logged on as yourself with full permissions. If something goes wrong, follow this to reset the security setting.

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.

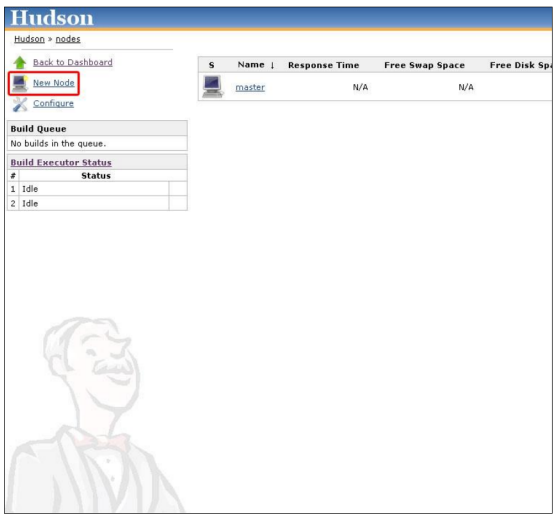


Installing Maven 2.2.1



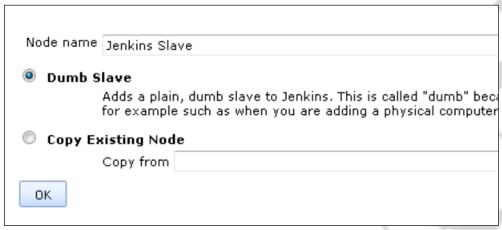
Adding a Windows slave node.

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



2. New Node --> Enter Node Name.

3. Select Dumb Slave --> Press OK.



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.

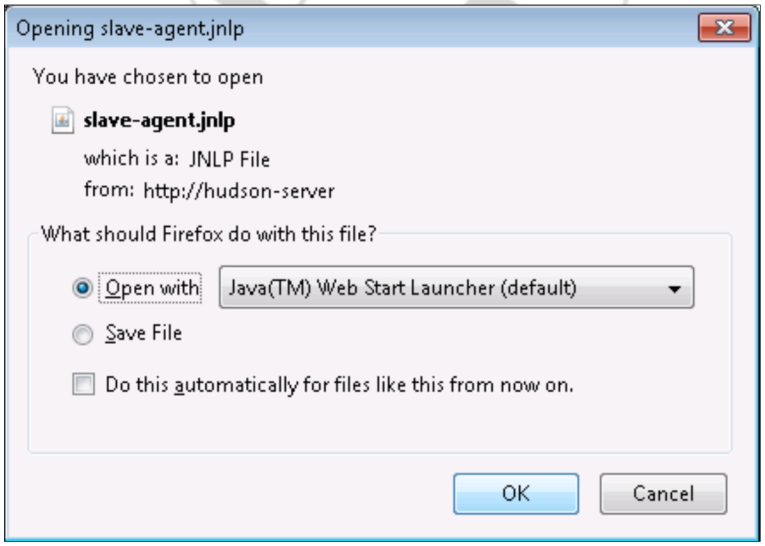
5. Now you need to connect your slave machine to the master using the following steps.

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, ...)

1. 

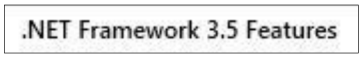
e. Now you should see the Slave machine connected under Nodes.

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"



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.

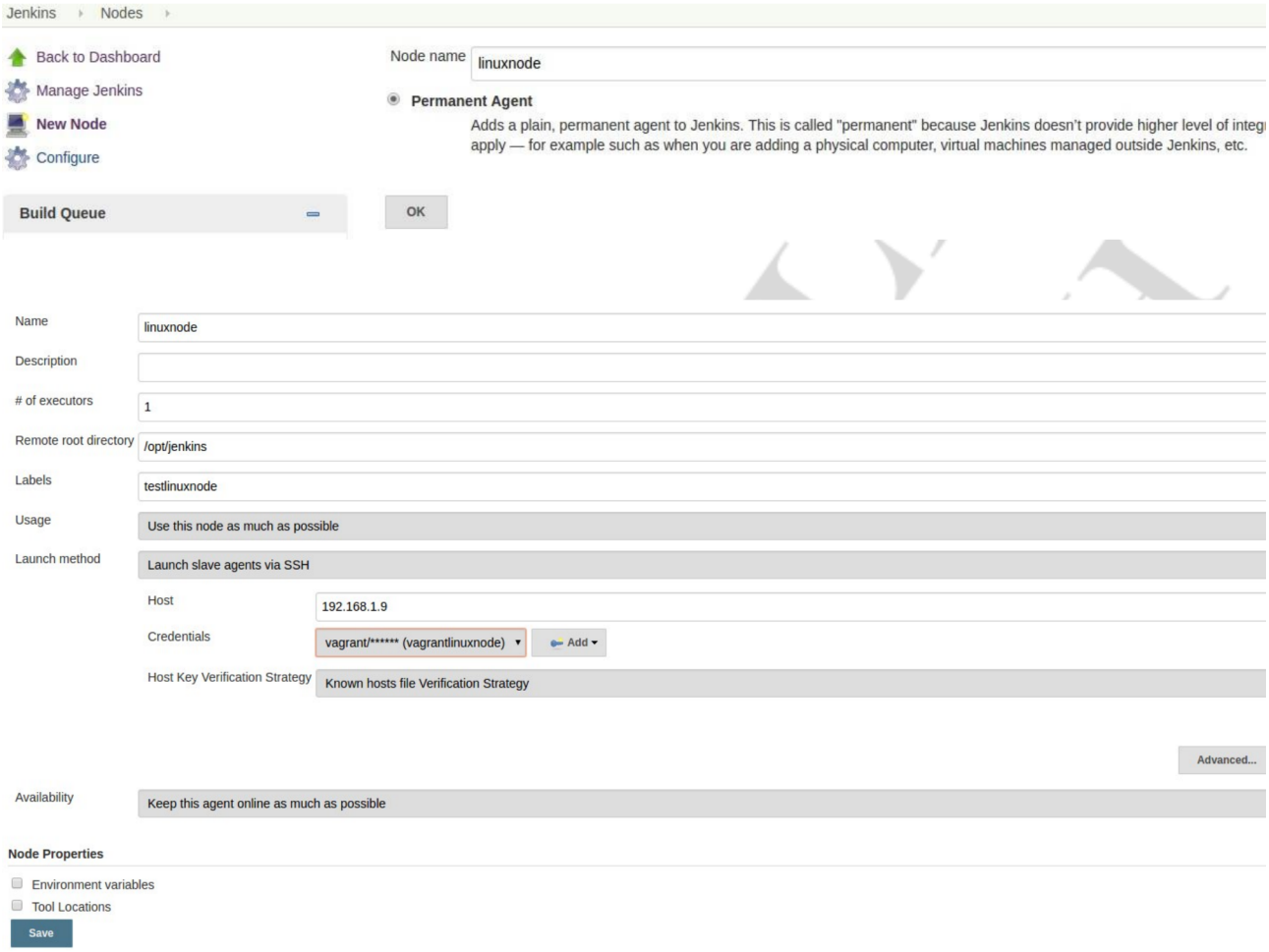
First setup the node.

```
sudo spt-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



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.

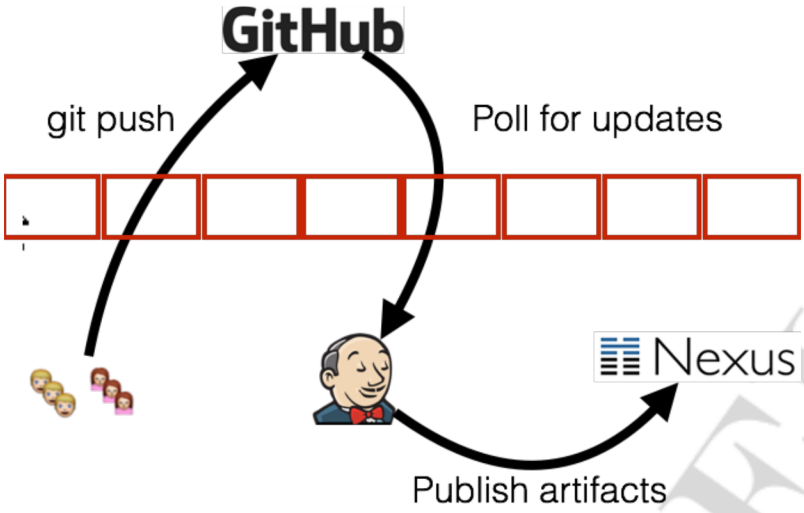
Nexus

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.



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

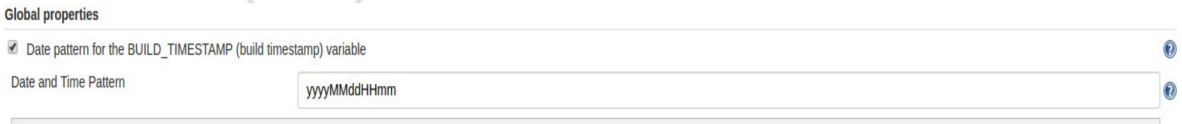
http://<jenkinsIP>:8080

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.

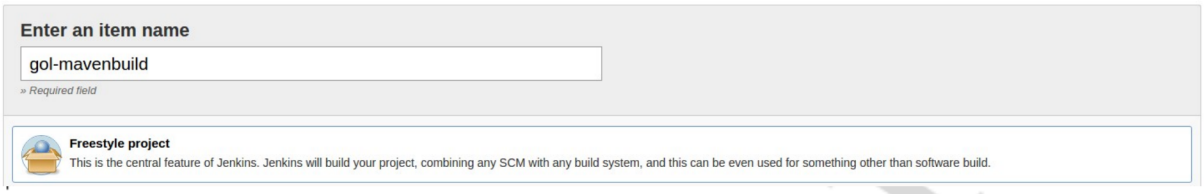
  After installing the plugin, we have to set its value from Configure System page.
  Manage Jenkins => Configure System => Global properties.



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

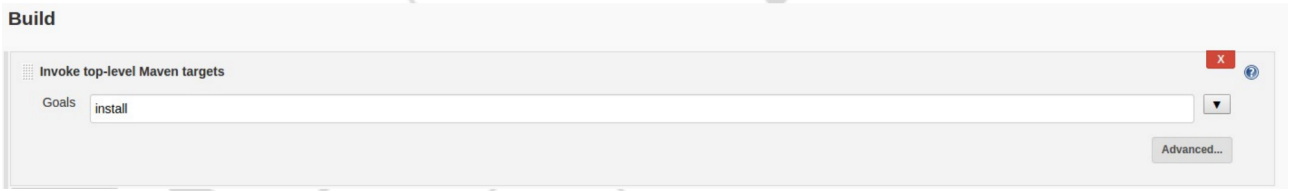
Create new build job

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



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

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



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

```
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