JENKINS INSTALLATION AND CONFIGURATION REFERENCE DOCUMENT

Employee Information	
Employee Name	Ha (Huck) H.K. NGUYEN
Department	Engineering
Date	06-OCT-2015

Table of Contents

1. Jenkins installation

1.1 Environment setting

1.1.1 JDK Installation

For Jenkins environment setting, *openjdk-7-jre* and *openjdk-7-jdk* are suggested.

For Ubuntu 12.04 LTS, you can install *JDK* from Terminal.

1. Open Terminal (Alt + Shift + T)
2. Type commend below:

   sudo apt-get update
3. Check if Java environment has been installed or not:

   java -version
4. If Java hasn't been installed on your system, you can use the command below to install *jre-7*:

   sudo apt-get install openjdk-7-jre
5. If you need *jdk-7* for your project, follow the command below:

   sudo apt-get install openjdk-7-jdk
6. To check if *Java environment* :

   java -version

1.1.2 Change default version of Java

Follow the command below to change Java version:

sudo update-alternatives –config java

Press the number too choose the default version.

1.1.3 Maven Installation

To install Maven from Terminal:

sudo apt-get install maven

Check if installation is successful or not:

mvn –version

The Terminal should show you:

Apache Maven 3.0.4

Maven home: /usr/share/maven

Java version: 1.7.0_80, vendor: Oracle Corporation

Java home: /usr/lib/jvm/java-7-oracle/jre

Default locale: en_US, platform encoding: UTF-8

OS name: "linux", version: "3.11.0-26-generic", arch: "amd64", family: "unix"

2. Jenkins Installation (Debian-based distributions installation)

2.1.1 Installation

1. Open Terminal
2. Paste these commands to Terminal:
   - *wget -q -O - https://jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -*
   - *sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ >/etc/apt/sources.list.d/jenkins.list'*
   - *sudo apt-get update*
   - *sudo apt-get install jenkins*
3. Press *Enter*

2.1.2 Upgrade

Once installed like this, you can update to the later version of Jenkins (when it comes out) by running the following commands:

1. Open Terminal
2. Paste these commands to Terminal:

   sudo apt-get update
   sudo apt-get install jenkins
3. Press *Enter*

2.1.3 Remove service

To remove *Jenkins*, you just need to type below command into *Terminal:*

sudo apt-get remove jenkins

2.1.4 Configuration

- By default, Jenkins listen on port 8080.

- If your */etc/init.d/jenkins* file fails to start jenkins, edit the */etc/default/jenkins* to replace the line

  HTTP_PORT=8080

  by: *HTTP_PORT=<your port>*

For more further information for configuring Jenkins on Ubuntu, please visit site:
https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins+on+Ubuntu

2.1.5 Remote service

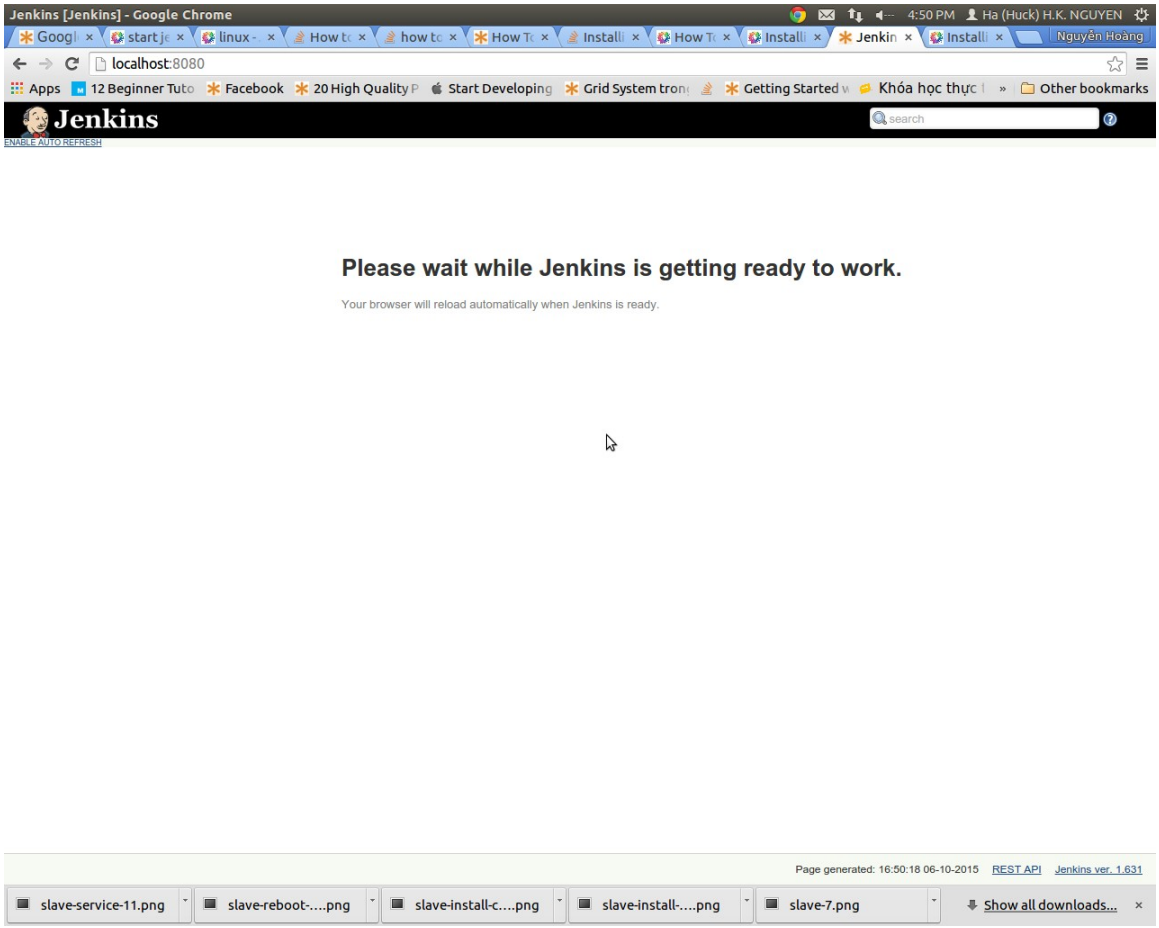
1. To start Jenkins, you can use this command:

   service jenkins start

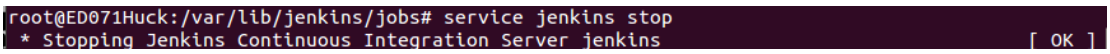
   or

   service jenkins restart

   While processing, the Jenkins page should be like:

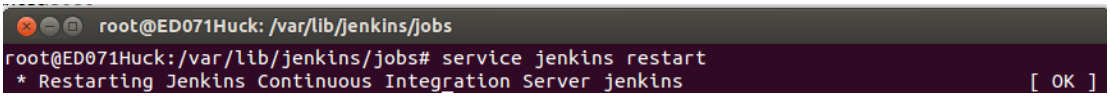


2. To stop Jenkins, you can use the command below:



service jenkins stop

3. To start/restart Jenkins, you can use these commands:



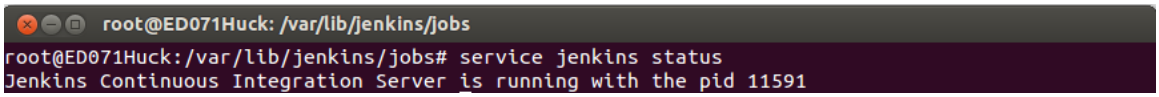
service jenkins start

or

service jenkins restart

4. You can check jenkins service status by using the command below:

service jenkins status



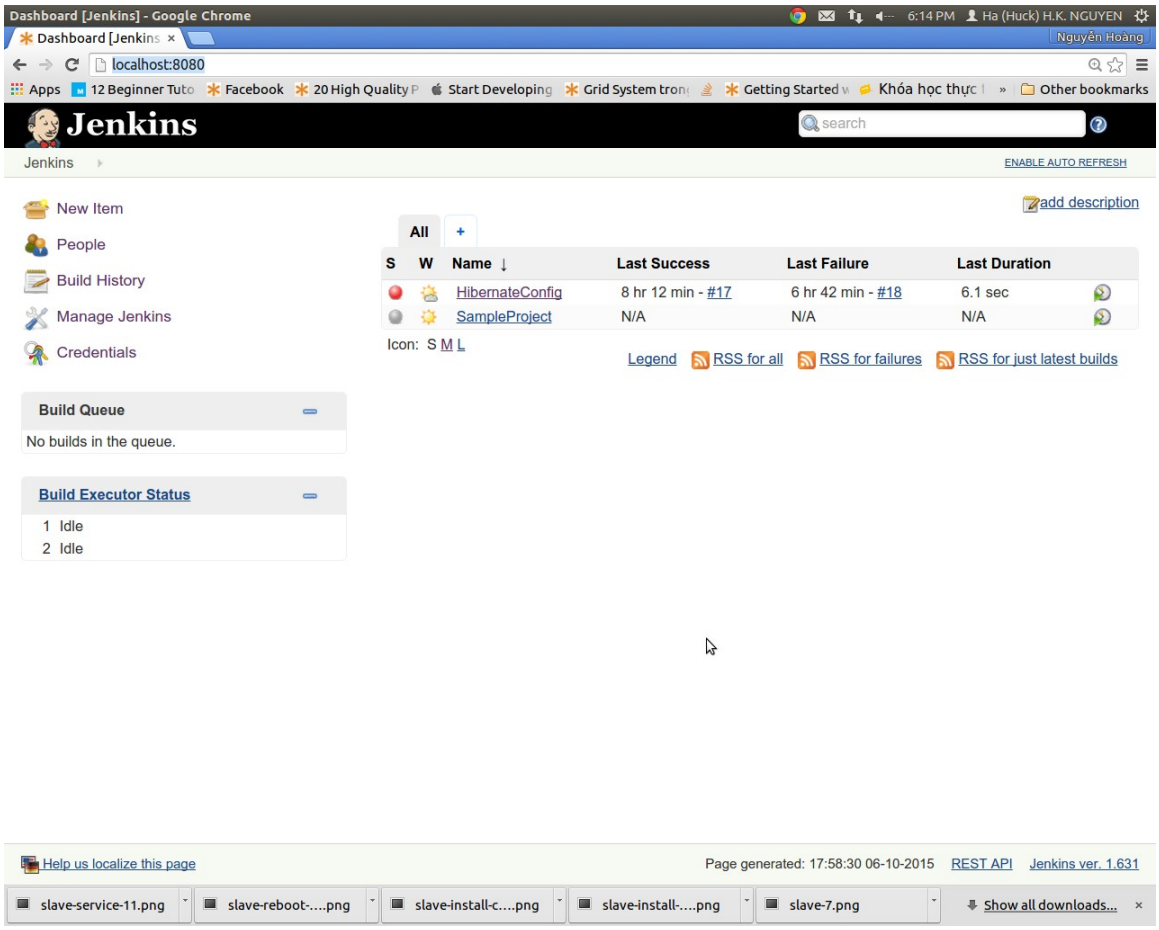
And it could show you *port id* of J*enkins* service.

3. Jenkins configuration

3.1 Overview

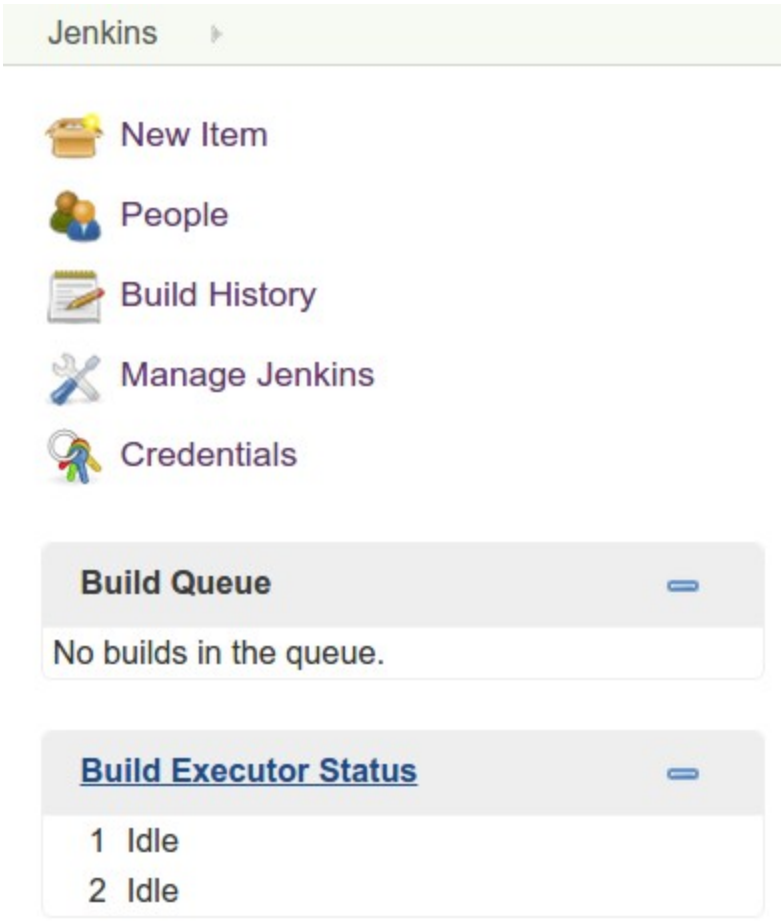
After start service, you can access to *jenkins local web-page* by local address: *http://localhost:8080/*

It should be like the picture below:



This page has two main-component:

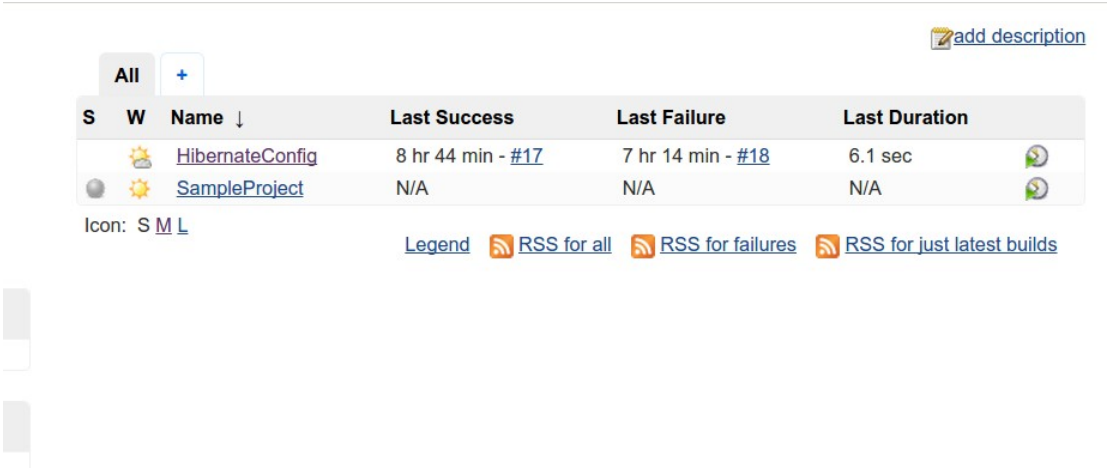
1. **The sidebar on the left**



This sidebar shows options could be used to configure Jenkins's functionalities. As default, it has 3 part:

- *General Configurations*
- *Build Queue*
- *Build Executor Status*

2. The main-content

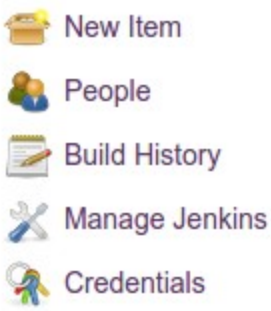


The *Main-content* shows the overview about projects using *Jenkins* every first time you visit this site, including Status (*S*), Weather report (*W: showing aggregated status of recent build*), *Name, Last Success, Last Failure, Last Duration.*

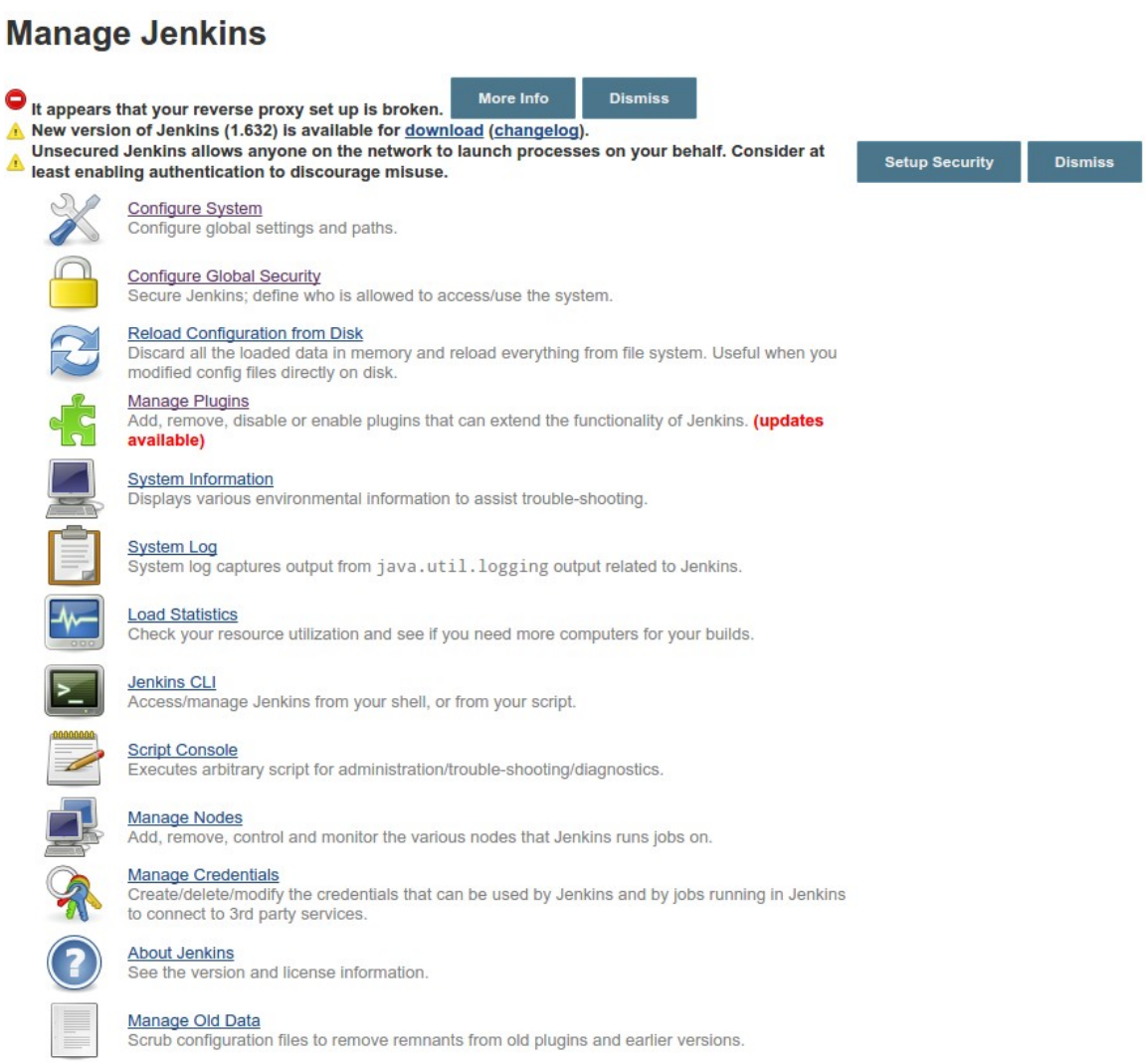
As default, this content is blank.

3.2 Jenkins general configuration

Go to *Jenkins – local page,* click on the *Manage Jenkins* on the *left sidebar*

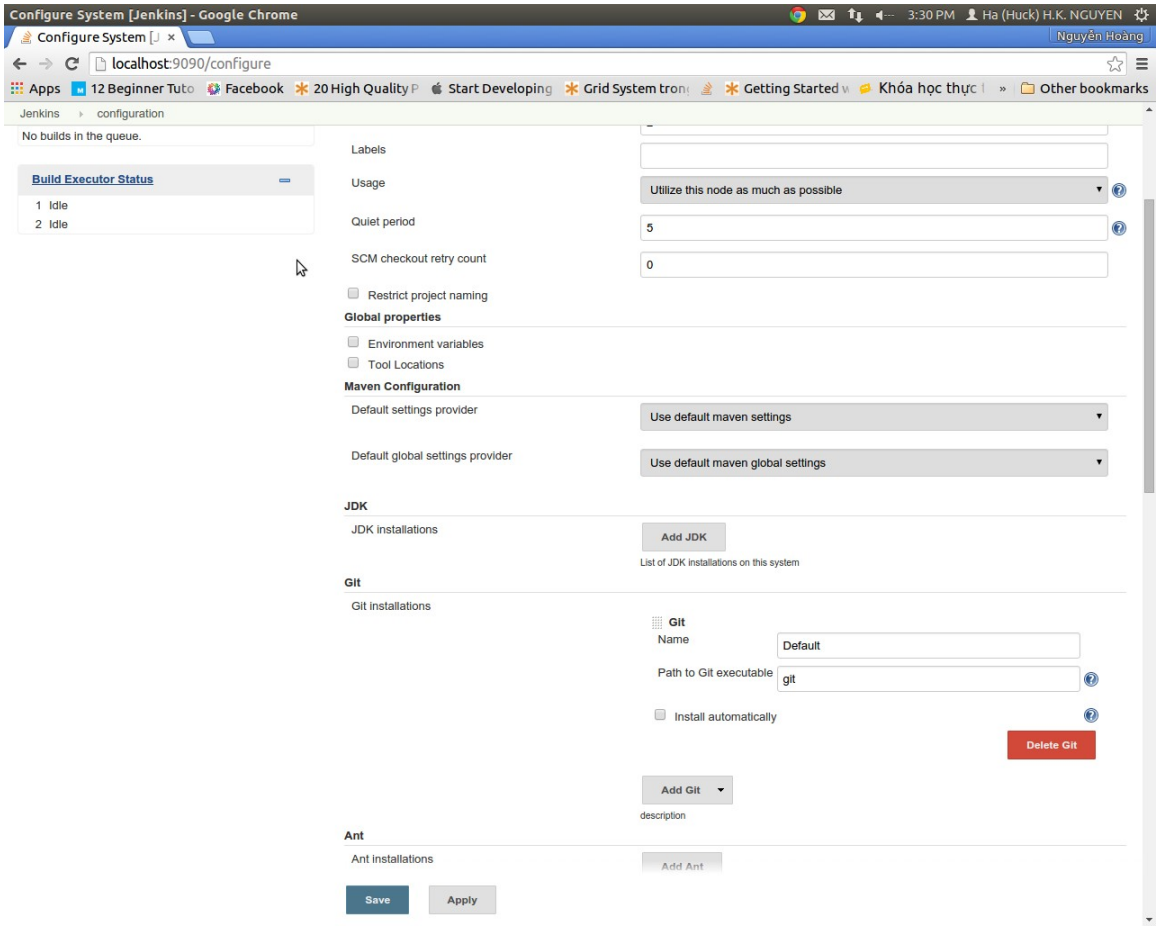


It should change your main-content in the right to the interface of *Manage Jenkins* functionality.



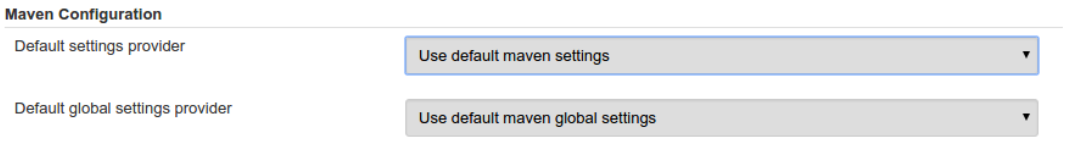
Click *Configure System* to go to Configuration functionalities.

The site will change to show you *system configuration* functionality.



By default, *Jenkins* let you some fill are not blank, but you can customize them as long as you want.

- **Maven configuration:**

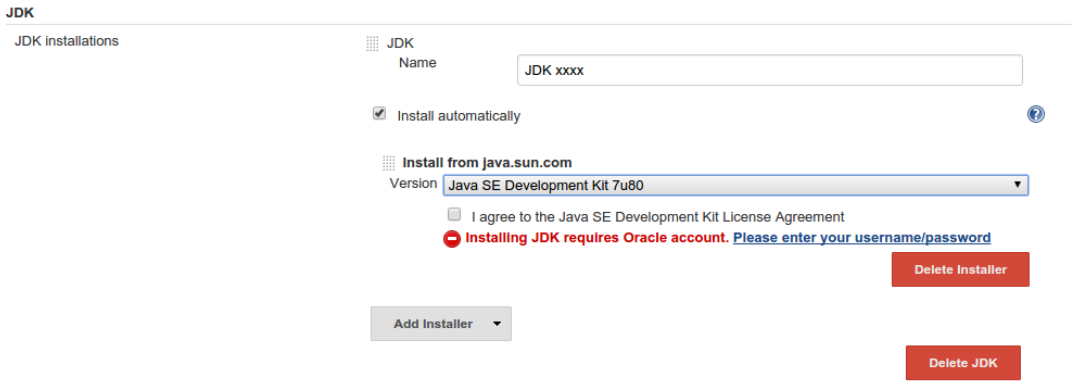


You can define where is *setting.xml* file located, by relative path of every project or relative to system. By default, it's depend on project setting.

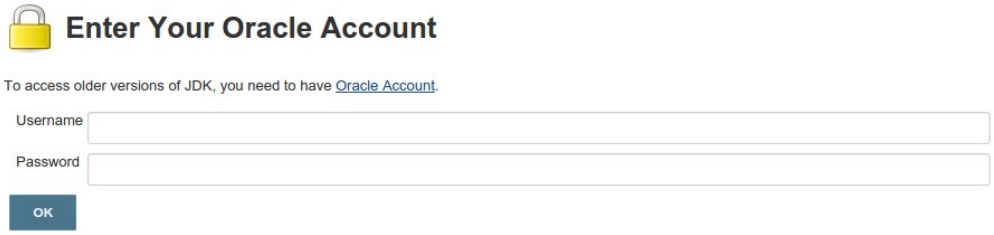
- **JDK configuration:**

By default, *Jenkins* uses current Java version.

By clicking on *add JDK*, it should show you the form.

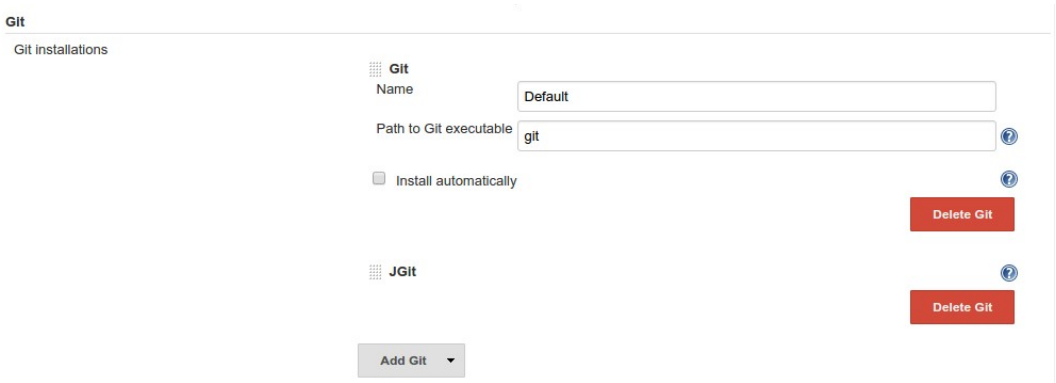


You can install JDK from your local machine by *Add Installer*, or tchoose the version of Java from Oracle and install. Which the second option, click on the blue line *(Please enter your username/password)* below the option of version and check I agree to the Java SE Development Kit License Agreement.



- **Git installation (After you installing Git plug-in package)**

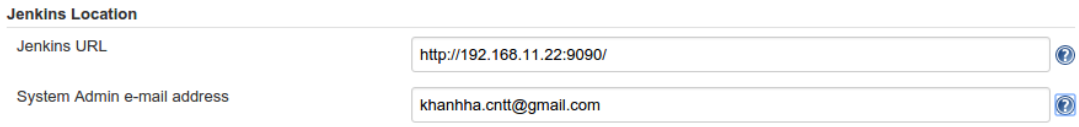
Figure where git-executable file is. You can figure it out by completed path or just fill "git" as default.



If you have demand to use *JGit*, just click on select tag *"Add Git"* and choose *Jgit*.

- **Jenkins Location Configuration**

To other people can access from another machine, you'd like to deploy it on your host.

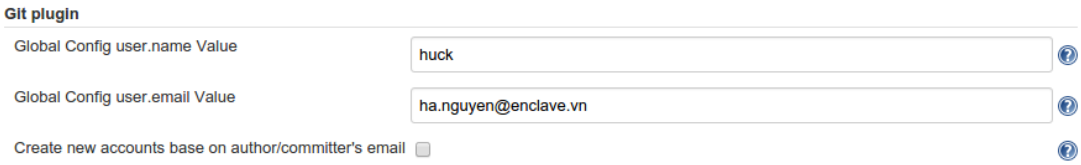


You can use your host by pattern: *http://<your host>:<port>/*

Fill in the *System Admin e-mail address* to figure out the email of project owner.

- **Git - plug-in configuration**

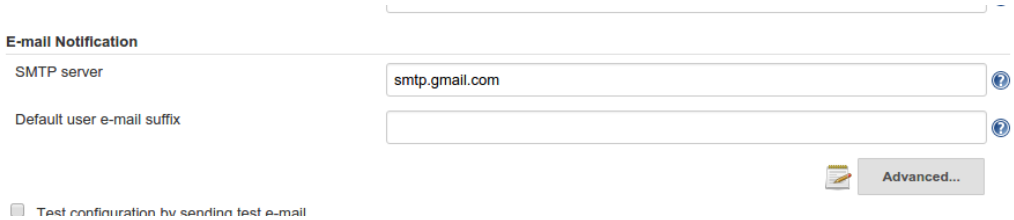
  In order to use *Git plug-in* you need to define your account by completing these fields below



You can add your *Git user name* and *email* stand for your *Git* account

- **Email Notification**

Go to *Email notification*.



Define mail server, for specific definition, click on *Advanced tag.*

Click-on *Use SMTP Authentication to* validate your account and password to send mail. (*Account* and *Password* field are below check box).

Click-on *Use SSL* to use *SSL* protocol on mail.

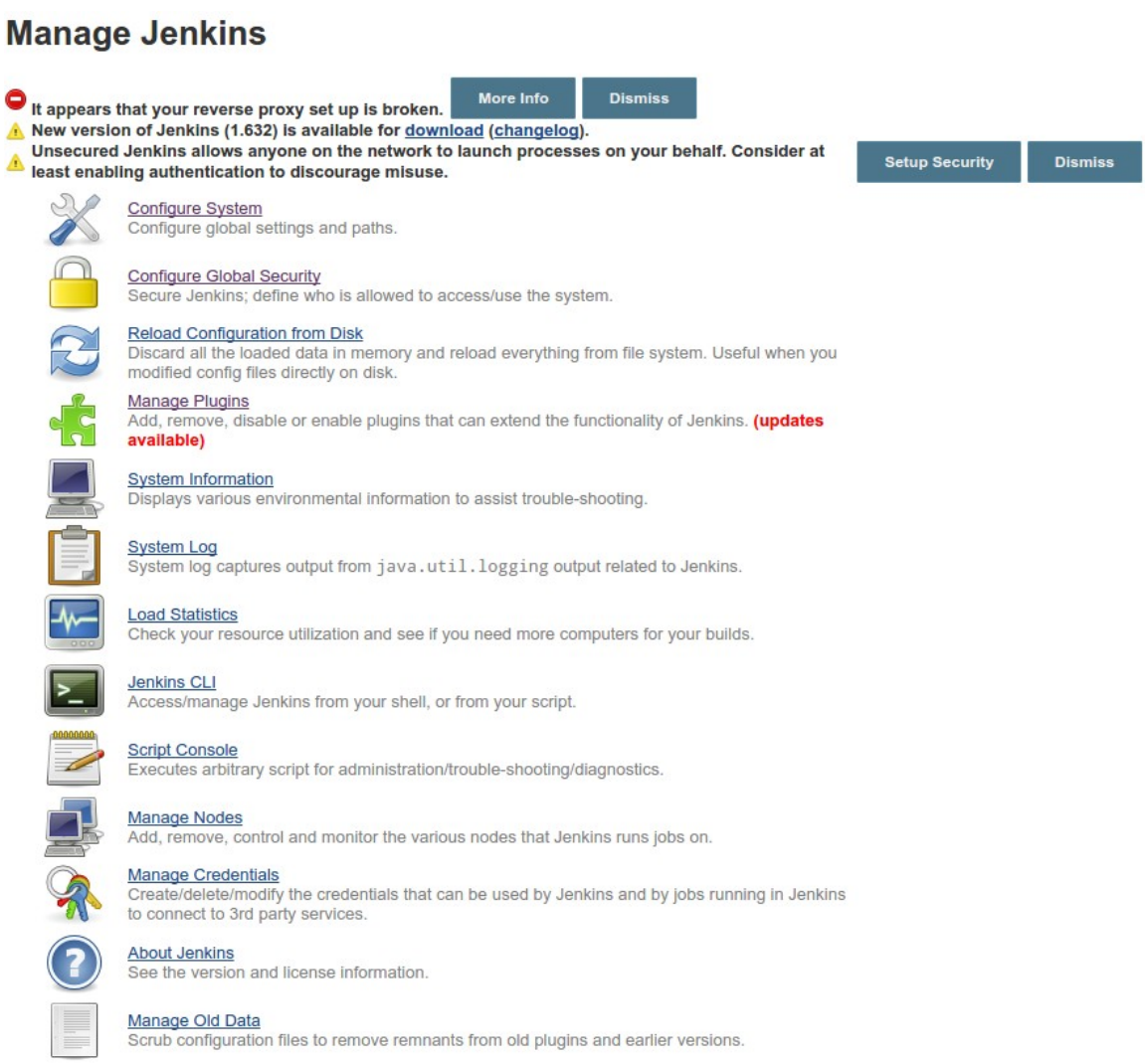
Define port of mail server (for example: *Gmail* uses port 465 for *SSL mail-service*).

Checking on *Test configuration by sending test email* to test function.

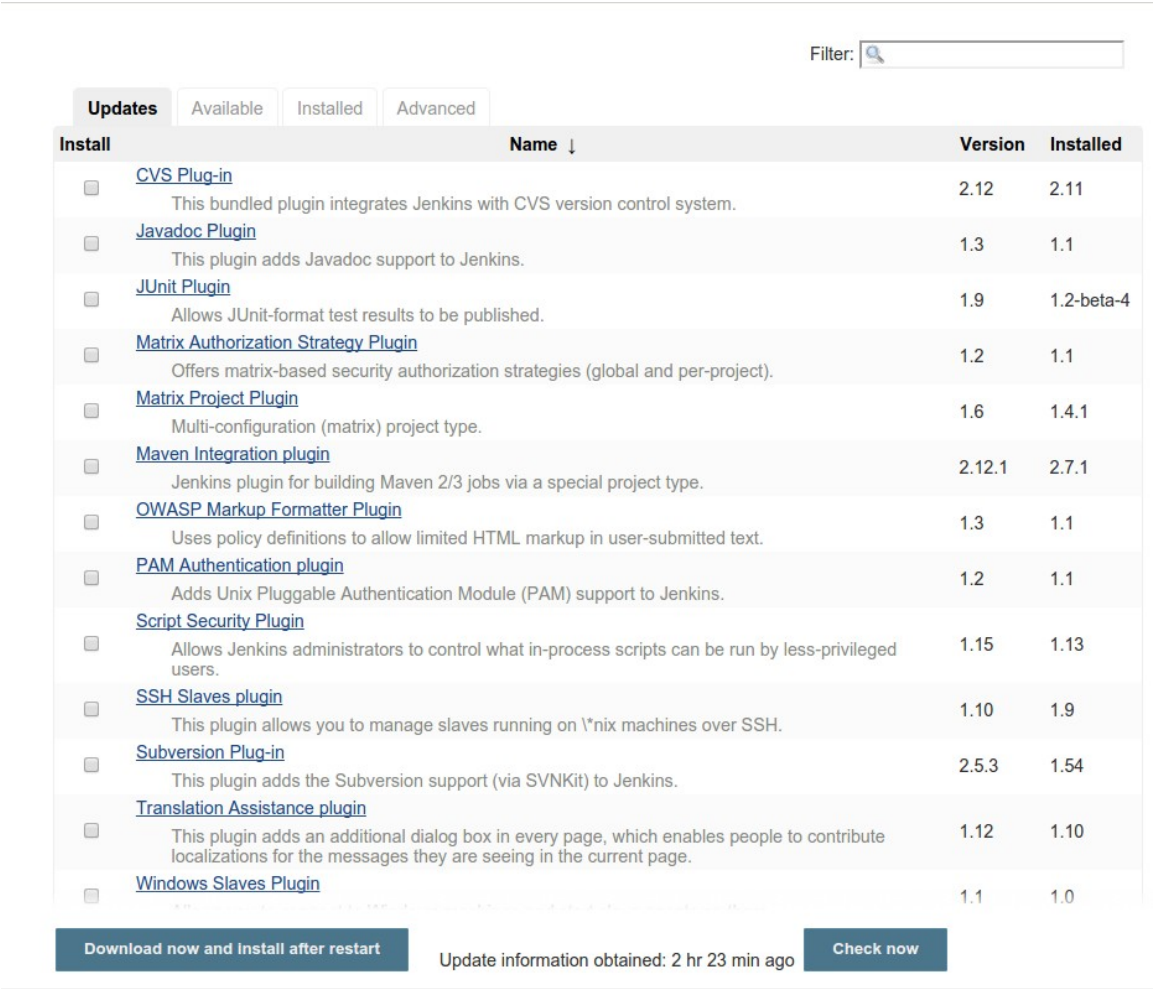
3.3 Plugin management

Go to *Jenkins – local page,* click on the *Manage Jenkins* on the *left sidebar*

Click on *Manage Jenkins* functionality.



Click on *Manage Plugin* to go to *Plugin - management site*



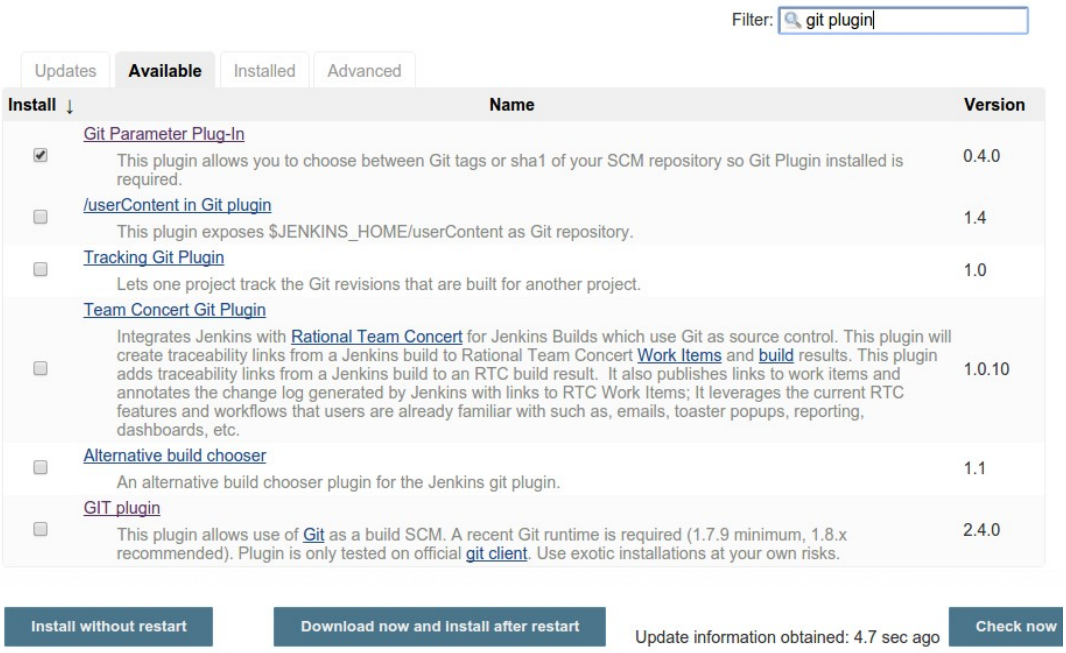
At the main content, you have the list of *Jenkins Plugin* which you *installed*, *available* on market place, can be *updated*, or something you want to add specifically at *Advanced* tag.

3.3.1 Install Git plug-in

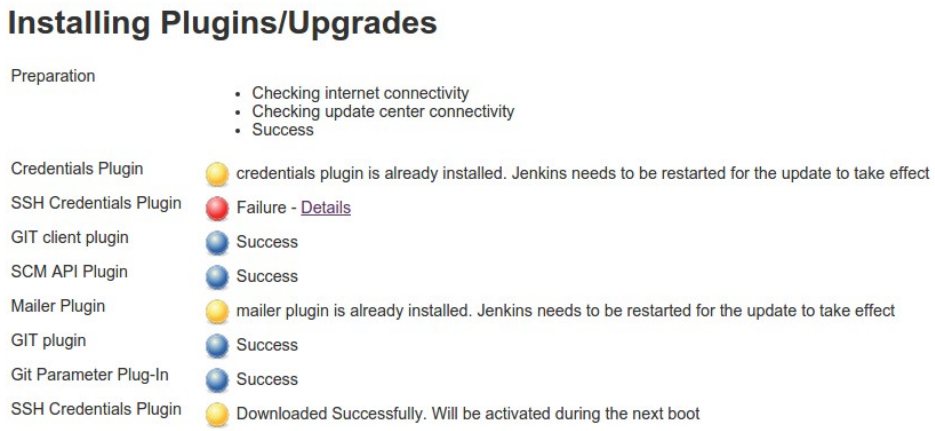
Go to *Plugin – management-site to* check if *Git Plug-in* was installed or not.

Go to *Available* tag to find out the plug in

Put name of plug in to *Filter*



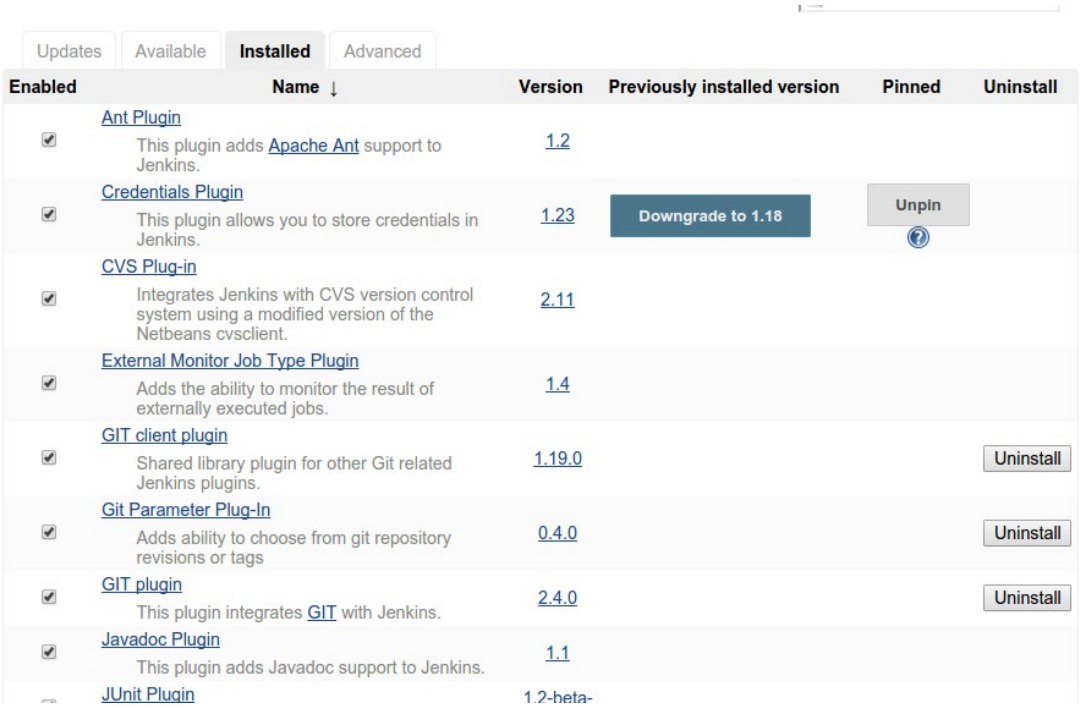
Checking the plug-in and hitting at *"Install without restart"*.



Balls show the status processes, the *blue* is success, the *red* one is failure and the *yellow* is waiting for restart service or reboot system. The *gray* one shows that element is in processing.

***Note**: if any failure element, you just need to copy name of that one and find again in list of plug-ins, don't need to install the whole plug-in.*

You can check if the plug – in is installed successfully or not by taking a look at *Installed* tag in *list of Plugin.*

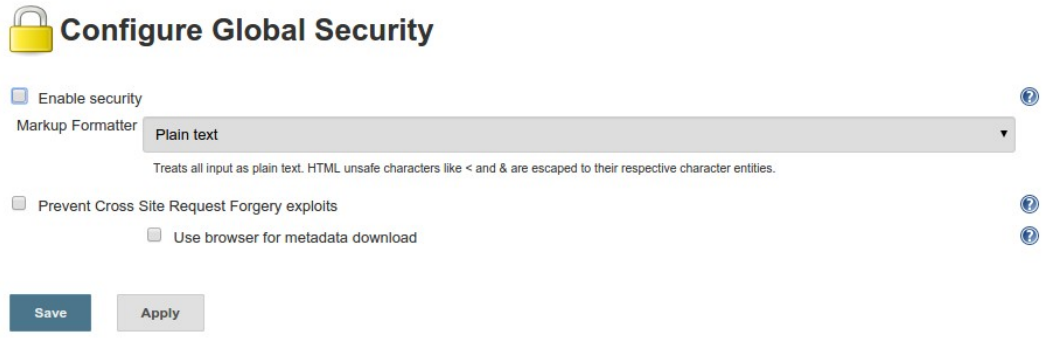


To remove plug in, Click on *Uninstall* button and wait for the end of process.

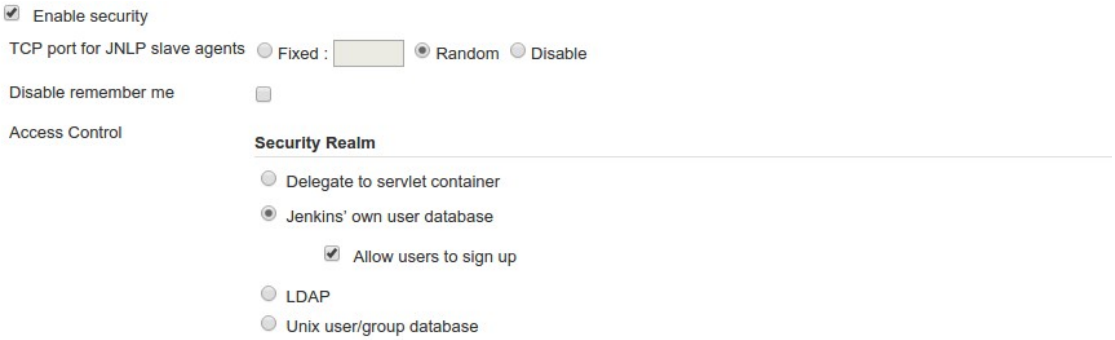
Note: *To update plug-in, you just need to do the same thing when installing, but with Updates tag*

3.4 Security in Jenkins

Take a look at *Configure Global Security* at *Manage Jenkins functionality,* then, you can see the content is like:



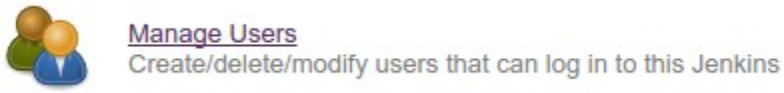
Checking at *Enable security*:



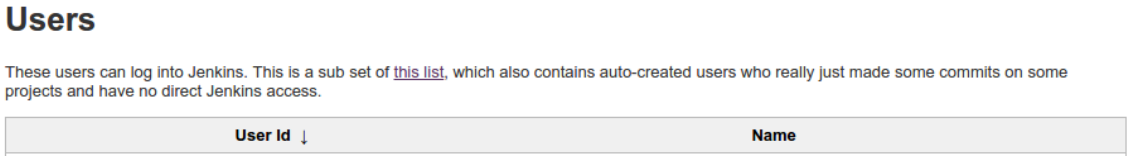
Choose *Jenkins' own user* database (automatically checking at *Allow users to sign up*).

Checking at *Project-based Matrix Authorization Strategy, then,* enable all permissions for this *Anonymous*. *Save.*

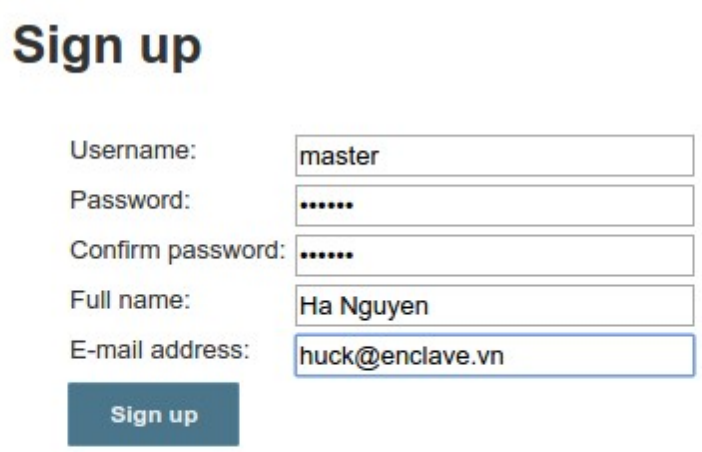
Back to *Manage Jenkins,* you should take a new functionality *"Manage User"*.



Click on this one to go to *Manage-User functionality.*

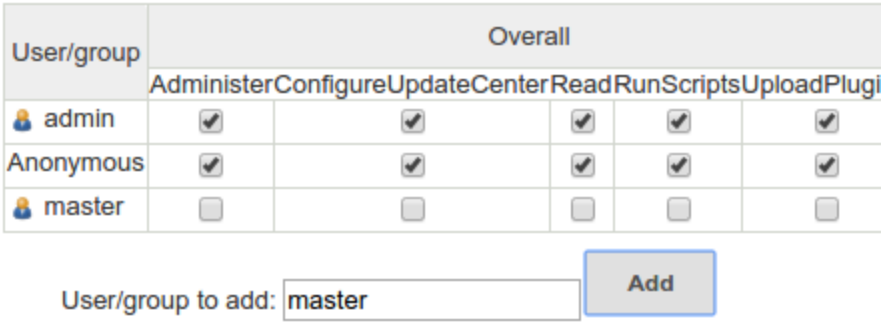


Click on *"Create User"* at the left sidebar, it will show you the form to create new user.



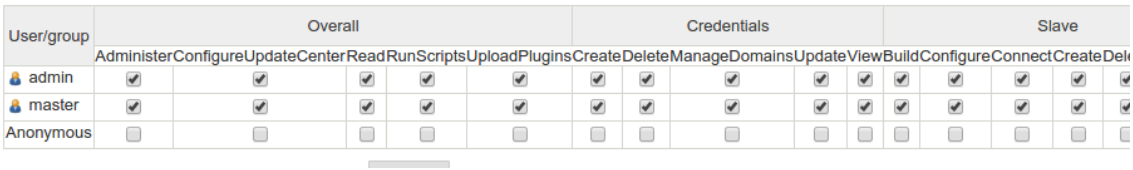
Fill in all fields and click *Sign up*. Back to *Configure Global Security* .

Below the table of user/group, Enter user-name of the account you've just created.



Provide permissions for that account**.**

Note*: In order to start User management, we gave anonymous account all permissions. Remember to remove permissions of anonymous account*



Click *Save*.

Note: *In case you forget to set permissions for anonymous, go to:*

/var/lib/jenkins/config.xml

then set:

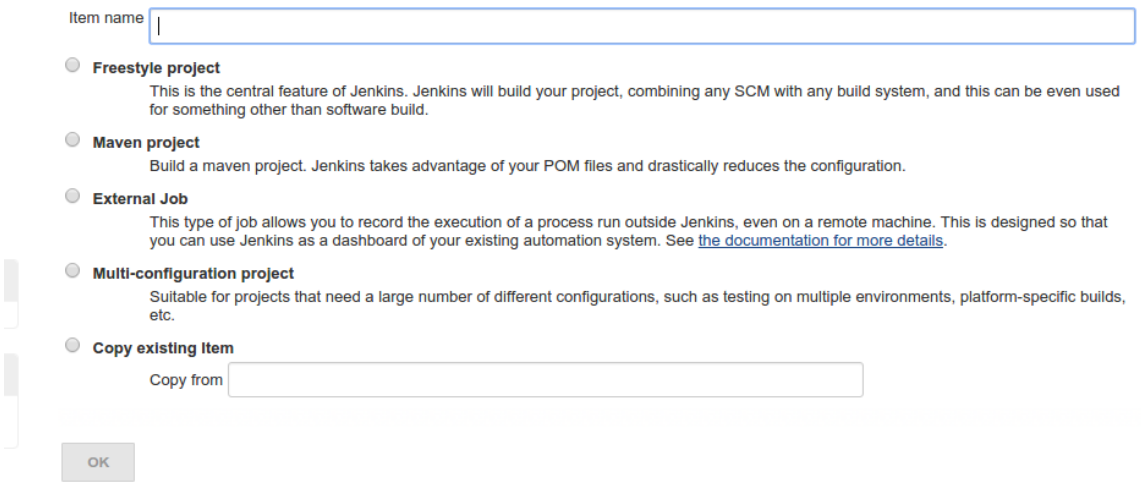
*<useSecurity>**true**</useSecurity>*

to

*<useSecurity>**false**</useSecurity>*

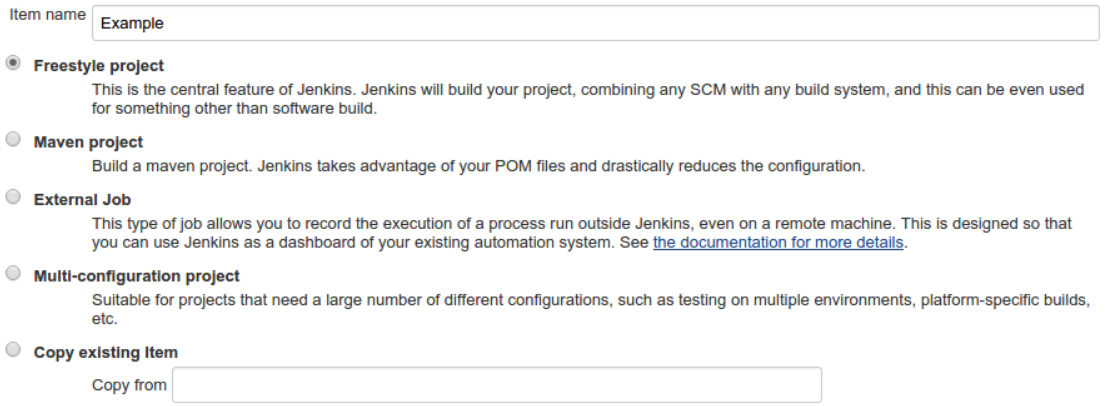
3.5 Project management

Click at New Item int left sidebar from *Jenkins' Dashboard* (Homepage).



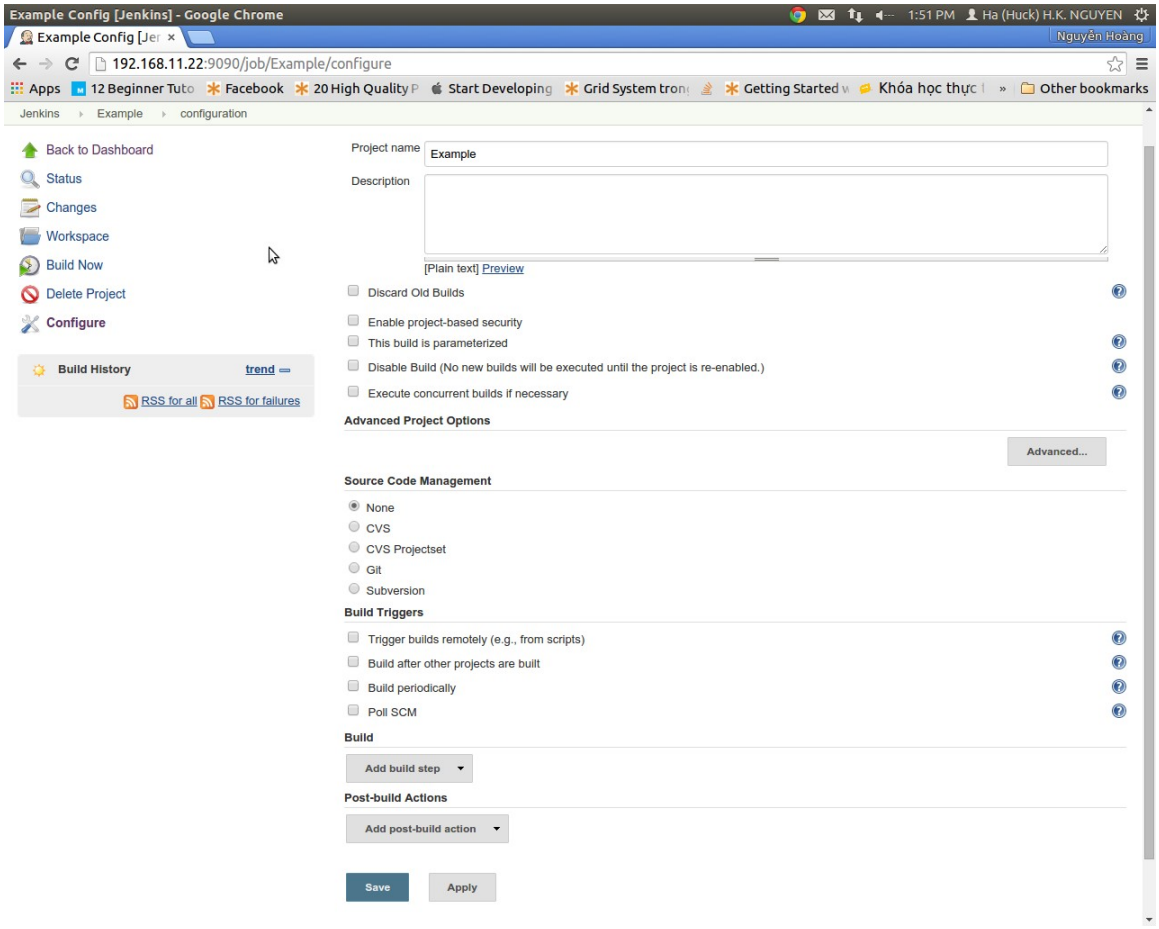
Fill in the name of project, then, choose one of options below the name field (As default, you have 4 options).

Choose *Freestyle project*

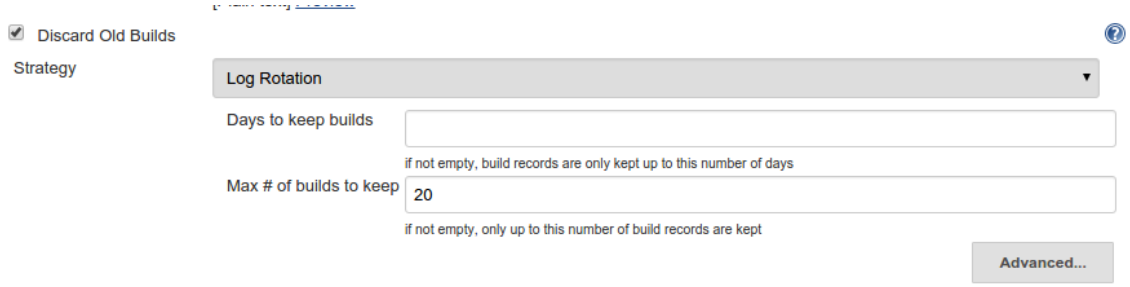


Click *OK*.

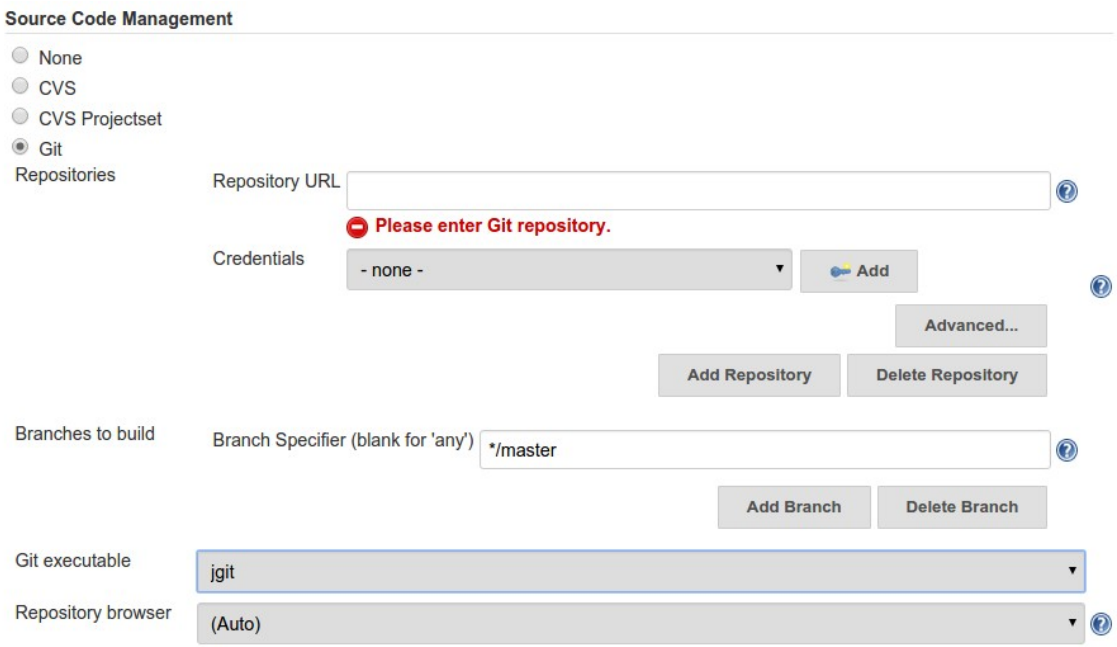
3.5.1 Freestyle project



Check at *Discard Old Builds,* then fill in *Max # of builds to keep* by the max number of build-process you want to keep.

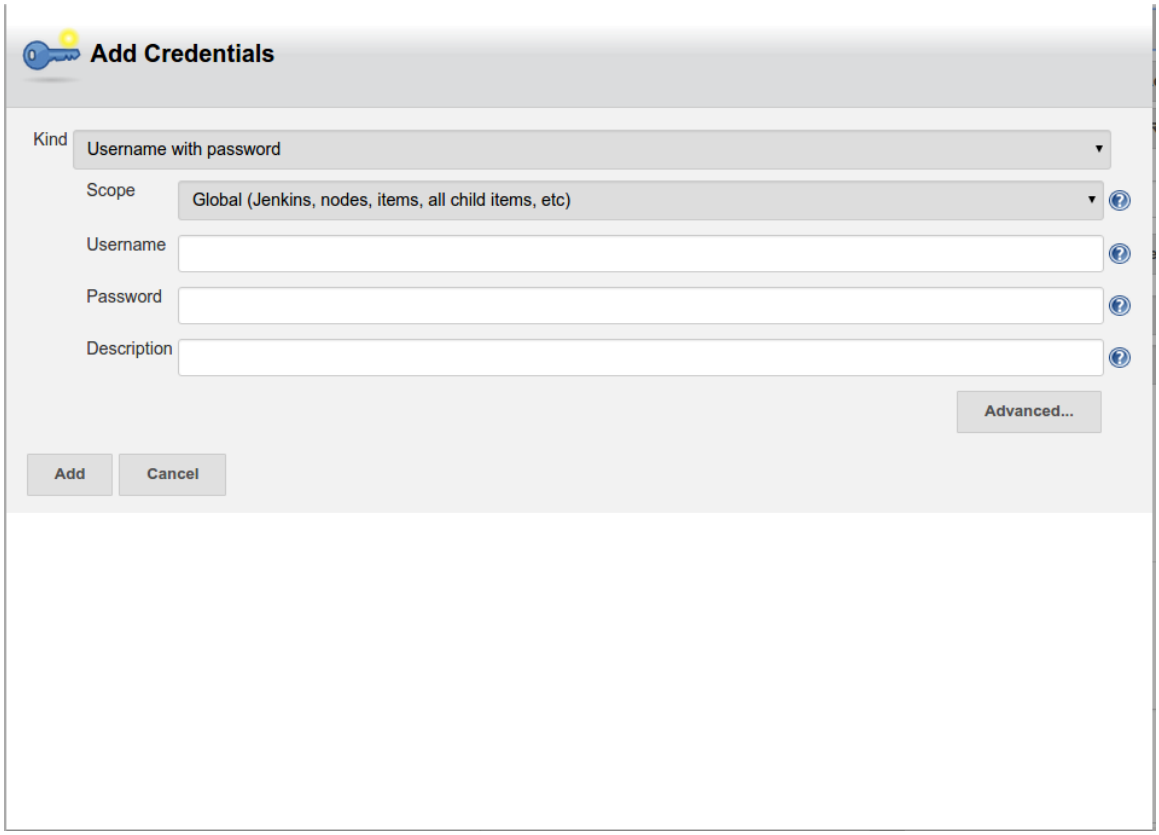


After adding *Git plug-in* (shown above), you can use git-repository to save your source. Checking at *Git*.



Please fill in *Repository URL* by your Git repository address. If you're familiar with *Jgit*, please choose *jgit* at *Git executable* (make sure that you did choose Jgit at *Jenkins' Configure System*).

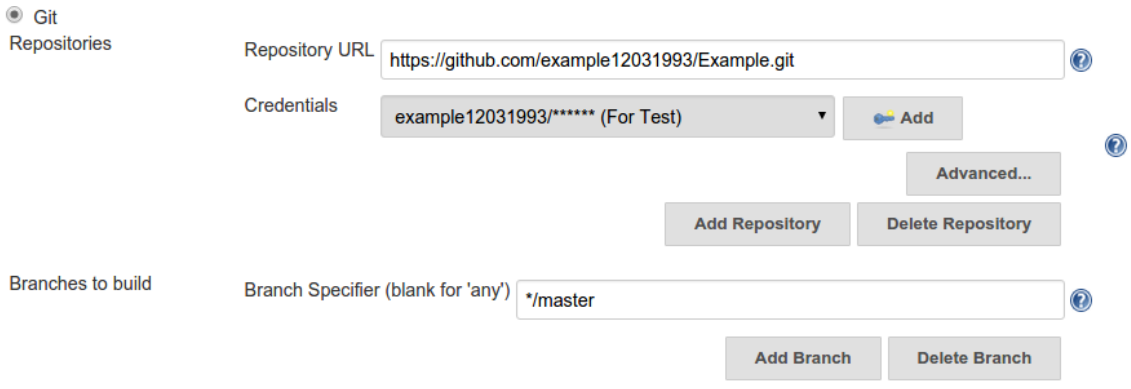
Click on *Add* button at *Credentials* field.



Fill the form with your *Git Username/Password*, click *Add*.

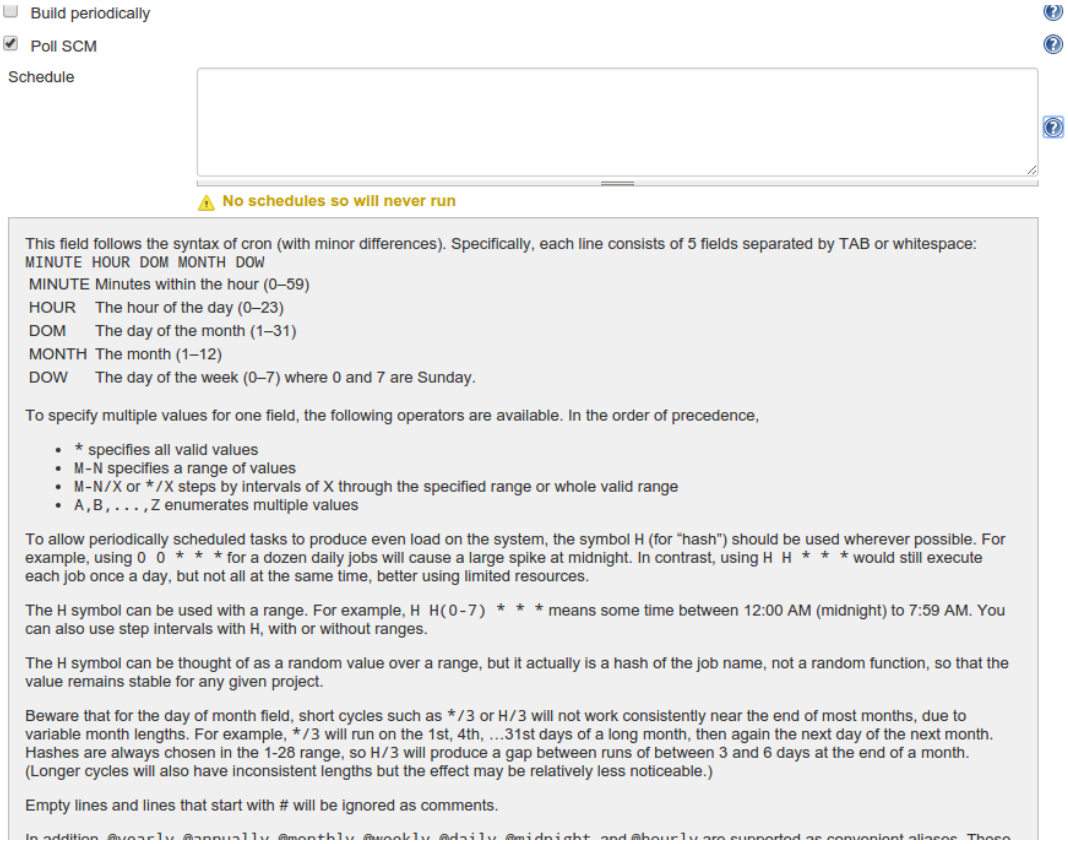
***Note**: you can use ssh protocol by changing Kind.*

Choose your *Credentials* bypass.



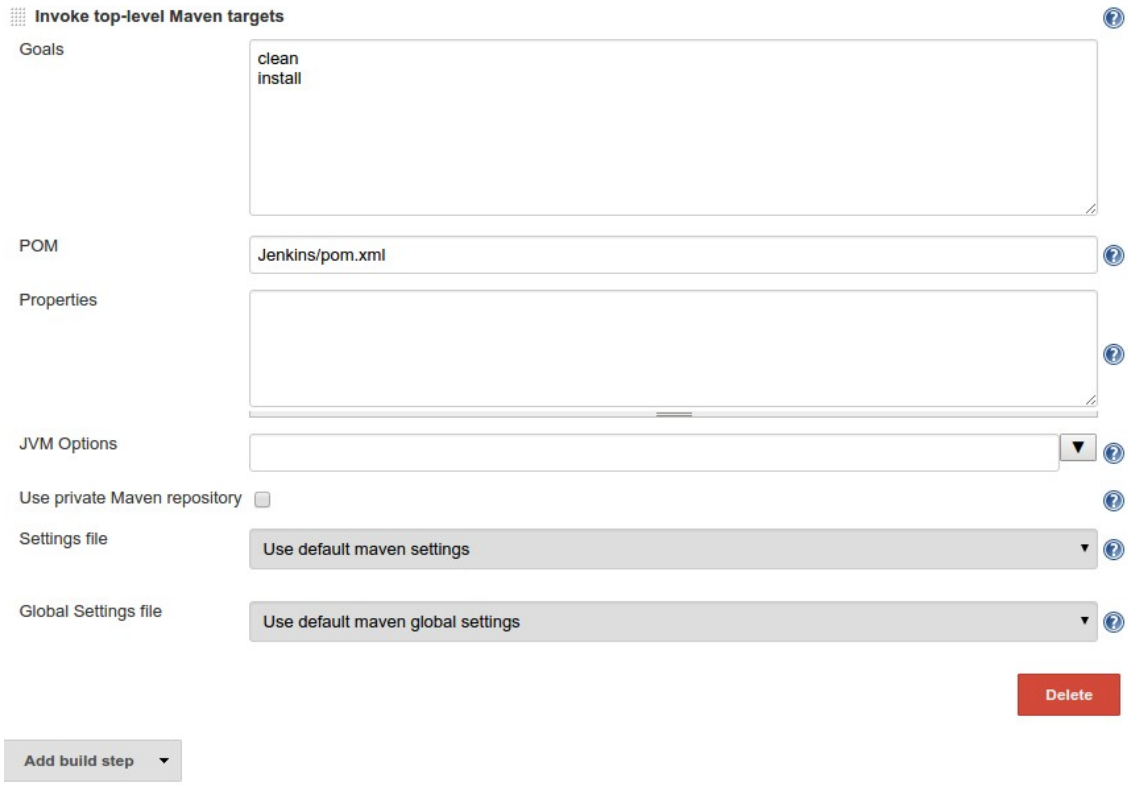
***Note**: in case you'd have many branches in your Git, please specify your need-to-build branch. If you let that field free, Jenkins would build whole branches from your git.*

Check at *Poll SCM* to build by your schedule. You can find out the syntax to define schedule by clicking onto the question mark (?) on the right of the field.



Click on *Add Build Step* to define the steps to build.

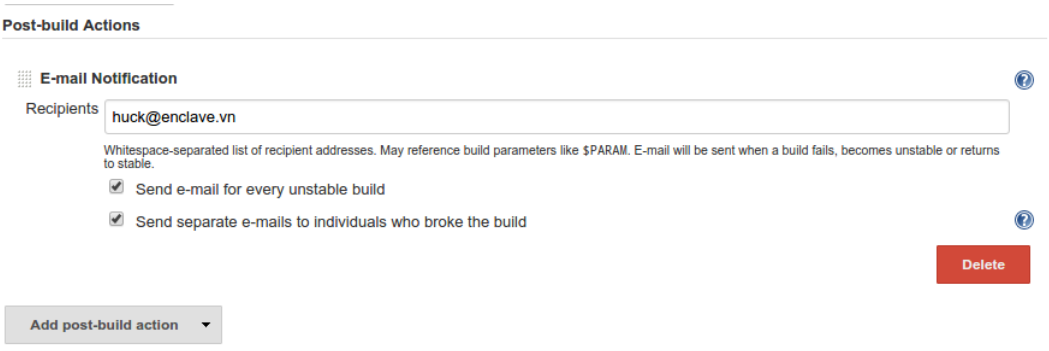
For Maven project only, choose *Invoke top-level Maven install,* then click on *Advanced* button to show all functions.



You need to define some fields.

- ***Goals***: The maven command you need to run when Jenkins build your project.
- ***POM***: you can define relative link to POM.xml file in Jenkins project (for example, in my case is: */var/lib/jenkins/jobs/Example/workspace/**Jenkins/pom.xml***).

Look at *Post-build Action,* click at *Add post-build action* and choose *Email notification*

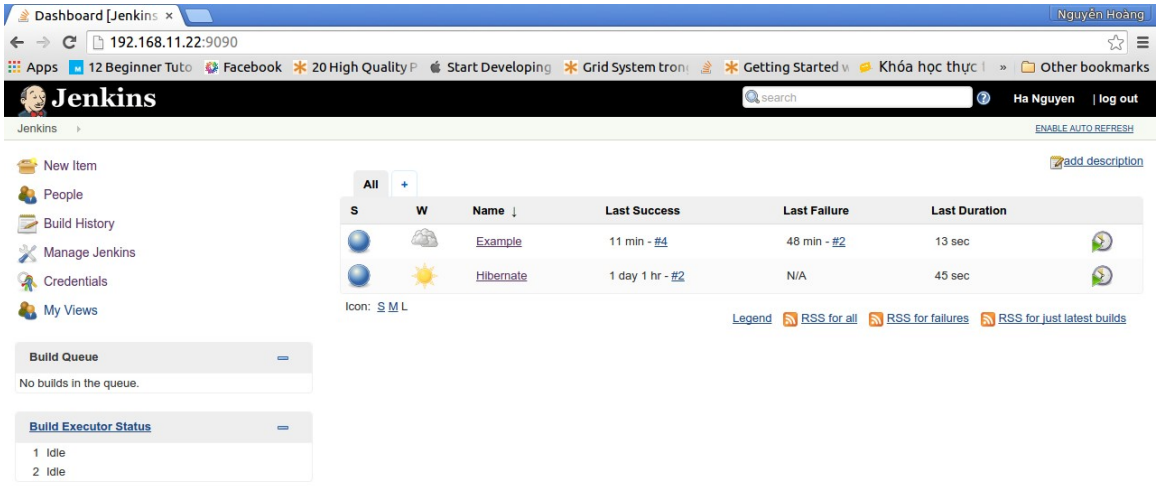


- Fill in the *Recipients* by the emails of people who you want to send to.
- Check in Send email for every unstable build: to send mail every unstable version of project was build.

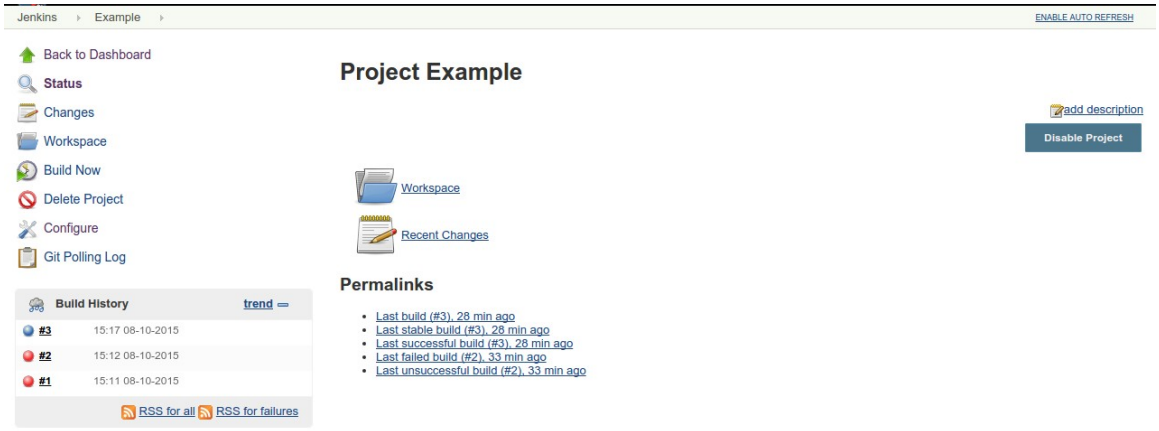
- Check in Send separate e-mails to individuals who broke the build: to send email to send mail to the guy who broke the system.

3.5.2 Build Project

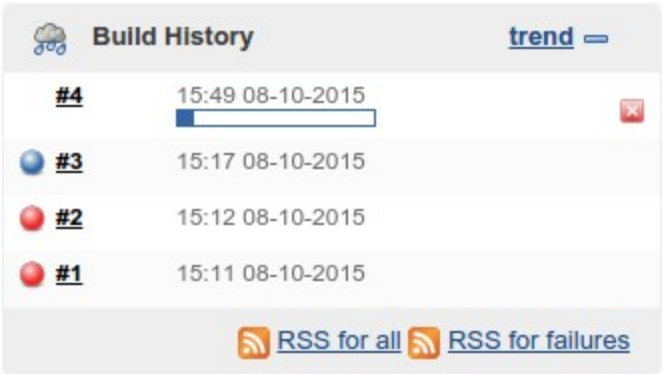
There are many place to build your project. One of them, click on your own from Project table in Jenkins homepage.



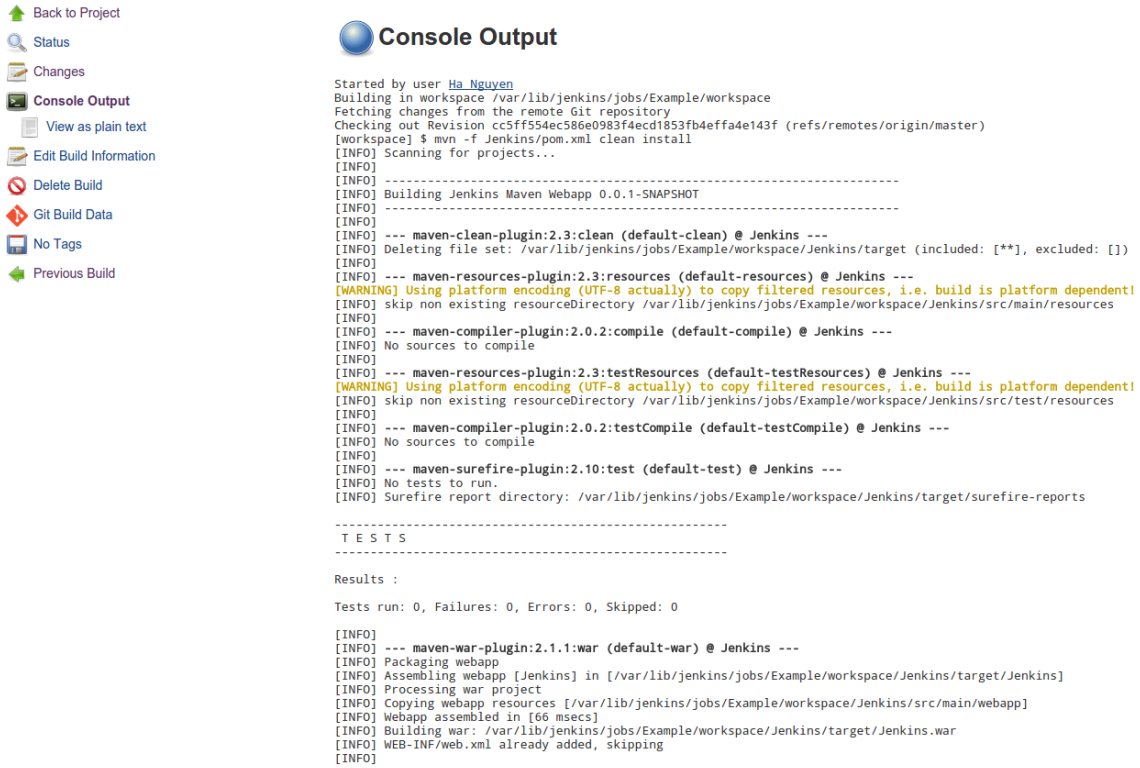
Go to Project page.



Click on *Build Now,* then look at the Build History on the left sidebar to track process.



You can click at the colorful ball at *Build History* to follow build-processing.



Click on *Console Output* on the left sidebar to show the console of that build.

4. Summary