Detail Design Document

Continuous Integration

Votary Softech Solutions Pvt. Ltd.

Plot No: 76, Lumbini layout,   
Near Euro school,   
Gachibowli-I (V), Hyderabad,  
Telangana - 500032,   
India.

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version (x.y) | Date of Revision | Description of Change | Reason for Change | Affected Sections | Approved By |
| 1.0 | 18-07-2017 | New Definition | New | ALL | Mohan |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version (x.y) | Prepared By | Reviewed By/Date | Approved By/Date |
| 1.0 | Meena P | Arun/20-07-2017 | Mohan/20-07-2017 |
|  |  |  |  |
|  |  |  |  |

**Contents**

[1 Continuous Integration(CI) 4](#_Toc481588852)

[1.1 Definition 4](#_Toc481588853)

[1.2 Work flow of CI 4](#_Toc481588854)

[1.3 Advantages of CI 4](#_Toc481588855)

[1.4 Tools Available for CI 4](#_Toc481588856)

[2 Jenkins 4](#_Toc481588857)

[2.1 Definition 4](#_Toc481588858)

[2.2 Why Only Jenkins 5](#_Toc481588859)

[2.3 Jenkins Work flow 5](#_Toc481588860)

[2.4 Advantages of Jenkins 5](#_Toc481588860)

[2.5 Jenkins is integrating various Dev Ops stages 5](#_Toc481588861)

[2.6 Build Status 6](#_Toc481588863)

[3 Jenkins Installation in Ubuntu 6](#_Toc481588865)

[4](#_Toc481588871) [Connect to Jenkins for the initial configuration](http://www.vogella.com/tutorials/Jenkins/article.html" \l "connect-to-jenkins-for-the-initial-configuration) [7](#_Toc481588871)

4.1 Generate ssh key for Jenkins user 7

[4.2 Jenkins Management 7](#_Toc481588873)

[4.3 Setting up a](#_Toc481588874) Jenkins Job [7](#_Toc481588874)

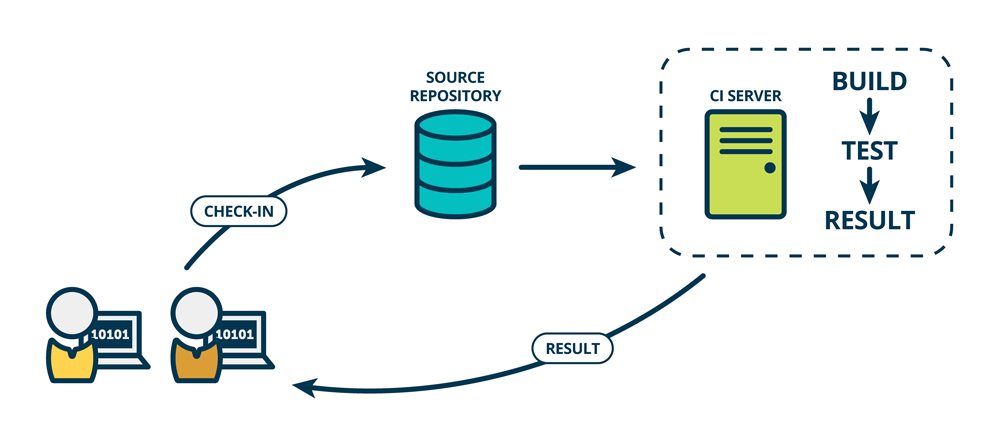
[4.4 Mail Configuration 7](#_Toc481588875)

## **1 Continuous Integration (CI)**

**1.1 Definition**

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

**1.2 Work Flow of CI**



Block Diagram of continuous Integration

By integrating regularly, you can detect errors quickly, and locate them more easily.

**1.3 Advantages of CI**

**Continuous Integration brings multiple benefits :**

* Say goodbye to long and tense integrations
* Increase visibility enabling greater communication
* Catch issues early and nip them in the bud
* Spend less time debugging and more time adding features
* Build a solid foundation
* Stop waiting to find out if your code’s going to work
* Reduce integration problems allowing you to deliver software more rapidly

**1.4 Tools Available for CI**

**Top 8 tools available for Continuous Integration(CI)**

1. Jenkins

2. TeamCity

3. Travis

4. Go

5. Bamboo

6. Git Lab

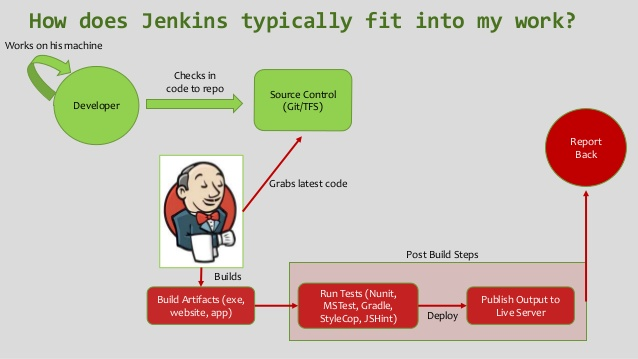
7. CircleCI

8. Code ship

**2 Jenkins**

**2.1 Define**

Jenkins is an open source CI tool written in Java. It originated as the fork of Hudson when the Oracle bought Sun Microsystems. Jenkins is cross-platform tool and it offers configuration both through GUI interface and console commands. What makes Jenkins very flexible is the feature extension through plugins. The Jenkins plugin list is very comprehensive and you can easily add your own. Besides extensibility, Jenkins prides itself on distributing builds and test loads on multiple machines. It is published under MIT license so it is free to use and distribute.



Jenkins is a Java-based tool so you will need Java Runtime.

Environment to run it. All the extensions and plug ins are also to be developed in Java. Given the underlying Java Runtime, Jenkins can be installed on any operating system where Java runs.

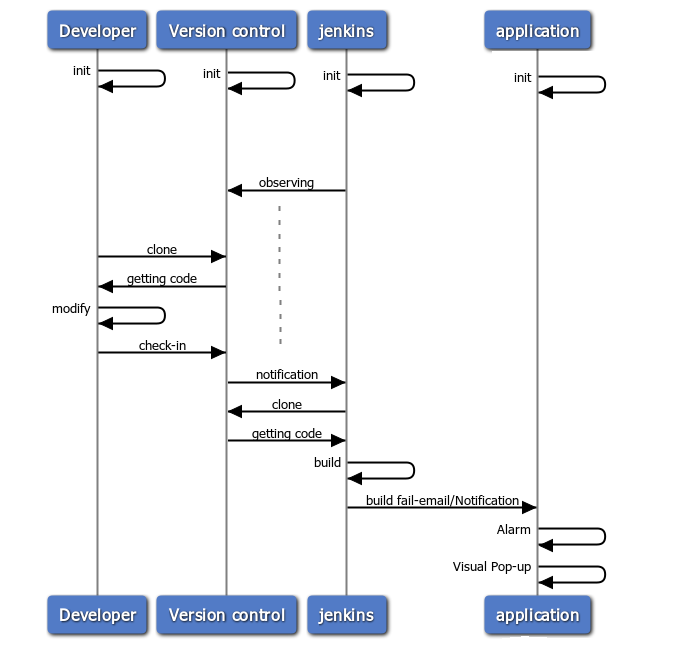
**2.2 Why Only Jenkins**

* **Installation of Jenkins** is easy and user-friendly. You do not require any additional installations or other components or database.
* The **configuration of Jenkins** can be done from its web-based GUI portal. Jenkins also provides detailed on-the-fly error checks, and online help.
* The treasure chest of **plugins available** from Jenkins is quite extensive which can be integrated easily into Jenkins. With these plugins and extensions – Jenkins can be adapted to satisfy virtually every requirement for continuous integration and continuous delivery requirements.
* Jenkins comes with the advantage of extensible as several parts of Jenkins can be modified and extended. This enables us tocustomize Jenkins as per our requirementsand also allow you to create new Jenkins plugins.
* Jenkins is **platform independent** and is available for all operating systems: OS X, Linux, and Windows. Jenkins can also transfer and share test loads to several computers with different operating systems. Therefore, there is no need to keep the same operating systems on all our machines.

Out of the box (or with default plugins, if you want) Jenkins supports the following:

* Version Control Systems:
  + Subversion
  + Git
* Builders
  + Command-line
  + Maven
  + Ant
  + Gradle
* Notifications
  + Git publisher
  + Email Notification
  + Set status for Git hub commit

**2.3 Jenkins Work Flow**

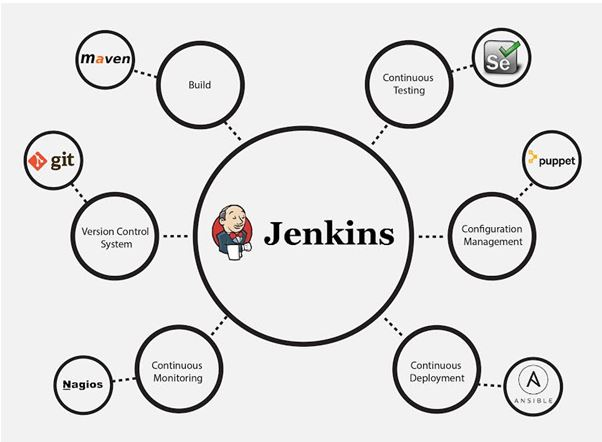
****

**2.4 Advantages of Jenkins**

* Jenkins is an open source tool with much support from its community.
* Installation is easier.
* It has more than 1000 plug-in to make the work easier.
* It is easy to create new Jenkins plugin if one is not available.
* It is a tool which is written in Java. Hence it can be portable to almost all major platforms.
* Jenkins is an open source tool with much support from its community.
* Installation is easier.
* It has more than 1000 plug-in to make the work easier.
* It is easy to create new Jenkins plugin if one is not available.
* It is a tool which is written in Java. Hence it can be portable to almost all major platforms.

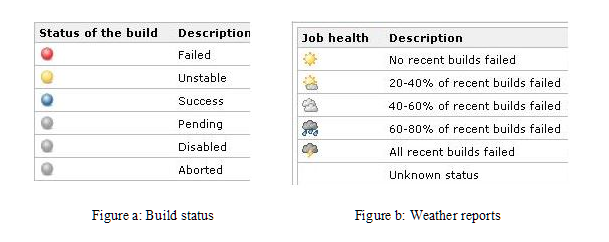
**2.5 Jenkins is integrating various DevOps stages**

**The diagram below depicts that Jenkins is integrating various Dev Ops stages:**



**2.6 Build status**

Once the project is configured in Jenkins then all future builds are automated. It has basic reporting features like status and weather reports (job health).



Most companies who handle continuous integration use their individual cloud-based continuous integration servers built on applications like Jenkins. With Jenkins, organizations can advance the software development process through automation. So overall to say, Jenkins integrates development life-cycle processes of all kinds which include building, documenting, testing, packaging, staging, deploying, static analysis and plenty more.

**3 Jenkins installation in Ubuntu**

**Basic Requirements:**

There’s no complex requirement for Jenkins but you must have sufficient superuser privileges to perform some basic installation tasks on your operating system and your system should be properly updated with the latest packages. Furthermore, Jenkins requires to have any basic web server installed on it. So, let’s start the setup of Jenkins on Ubuntu 14.04 LTS operating system with minimum 4GB RAM.

System Update

Run the following command to update your operating system before moving on to other packages installation.

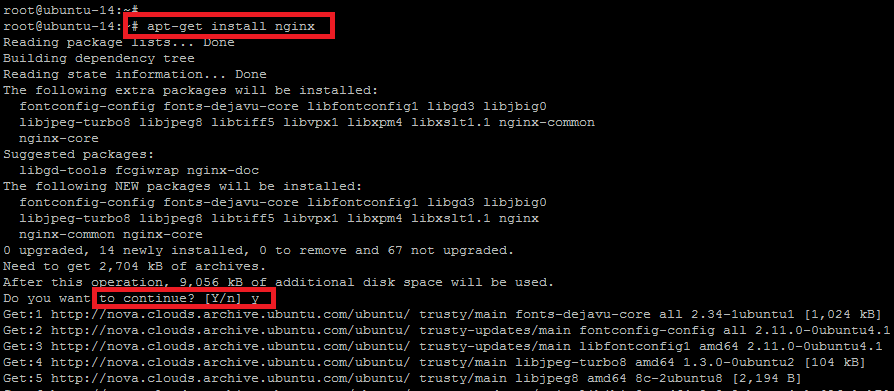
root@ubuntu-14:~# apt-get update

**Basic Web Server:**

Jenkins requires the basic web server installed on your system, so if you don’t have any of your web server installed on your operating system, you can execute the following command to install the web server.

root@ubuntu-14:~# apt-get install nginx

This will be installing Nginx web server. If you are more comfortable with a different web server, you can also choose other web servers to be installed.



Once your Nginx web server is installed check the status of its services with the command below.

root@ubuntu-14:~# service nginx status

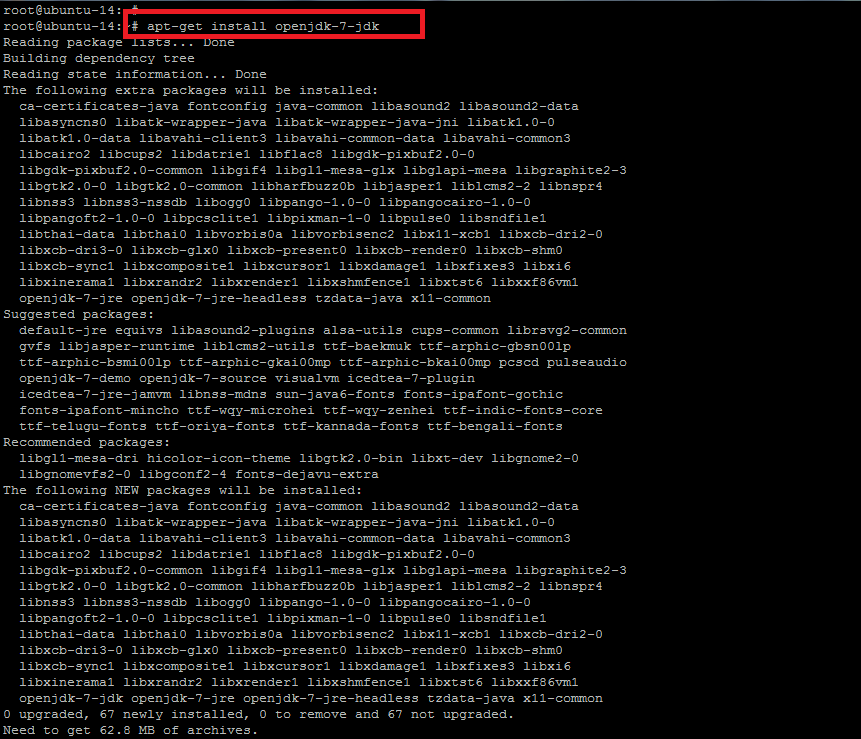
\* nginx is running

It means `Nginx` is working fine, so let’s proceed to next step.

**Java Installation**

Now install the Java first through the `apt-get` package manager. For this purpose, you will need to install open source implementation of JAVA Version 7. This is quite easy to do so by executing the command below.

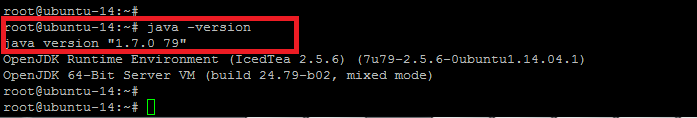
root@ubuntu-14:~# apt-get install openjdk-7-jdk or apt-get install openjdk-8-jdk



When you execute the above command, it shows you the list of other recommended and required packages to be installed with Java `openjdk`. To proceed with those packages, you have to press `Y` or type `N` to exit the installer.

It will take a few minutes to complete the Java installation. Once it is done, run the command below to check the installed version of Java on your system as shown.

root@ubuntu-14:~# java –version



**Jenkins Installation and Configuration**

Now we are ready to install `Jenkins`. So before the package installation, we have to add the key and source list to apt for Jenkins. To do so, issue the following two commands in the terminal.

root@ubuntu-14:~# wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -

OK

You will get the `OK` status after you add the key. For the source list, here is the command to run.

root@ubuntu-14:~# sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'

Now it needs to update `apt’s` cache before moving to Jenkins so that it can refresh the operating system’s repository for the latest packages.

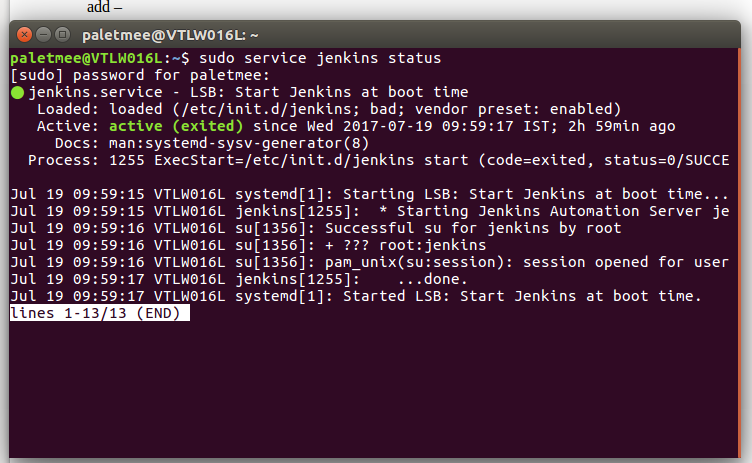
root@ubuntu-14:~# apt-get update

Once your system is updated, execute the command below for the installation of Jenkins and type the `Y` key to proceed with the installation process.

root@ubuntu-14:~# apt-get install Jenkins

The installation process will end up with starting its daemon. To check the status of Jenkins service, the command below will show you its status with running process ID. Jenkins service run with its default user name `jerkin`.

root@ubuntu-14:~# service jenkins status



Jenkins Continuous Integration Server is running with the pid of **some number.**

If you need to update the configurations of Jenkins as per your requirements, then you can find its configuration file under the `/etc/default/` directory and can make the changes.

root@ubuntu-14:~# vim /etc/default/Jenkins

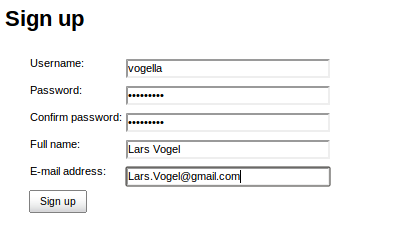
Jenkins runs on port 8080 by default for HTTP connector.

### **4 [Connect to Jenkins for the initial configuration](http://www.vogella.com/tutorials/Jenkins/article.html" \l "connect-to-jenkins-for-the-initial-configuration)**

After installation, open a browser and connect to it. The default port of Jenkins is :8080, therefore on your local machine you find it under the following URL:

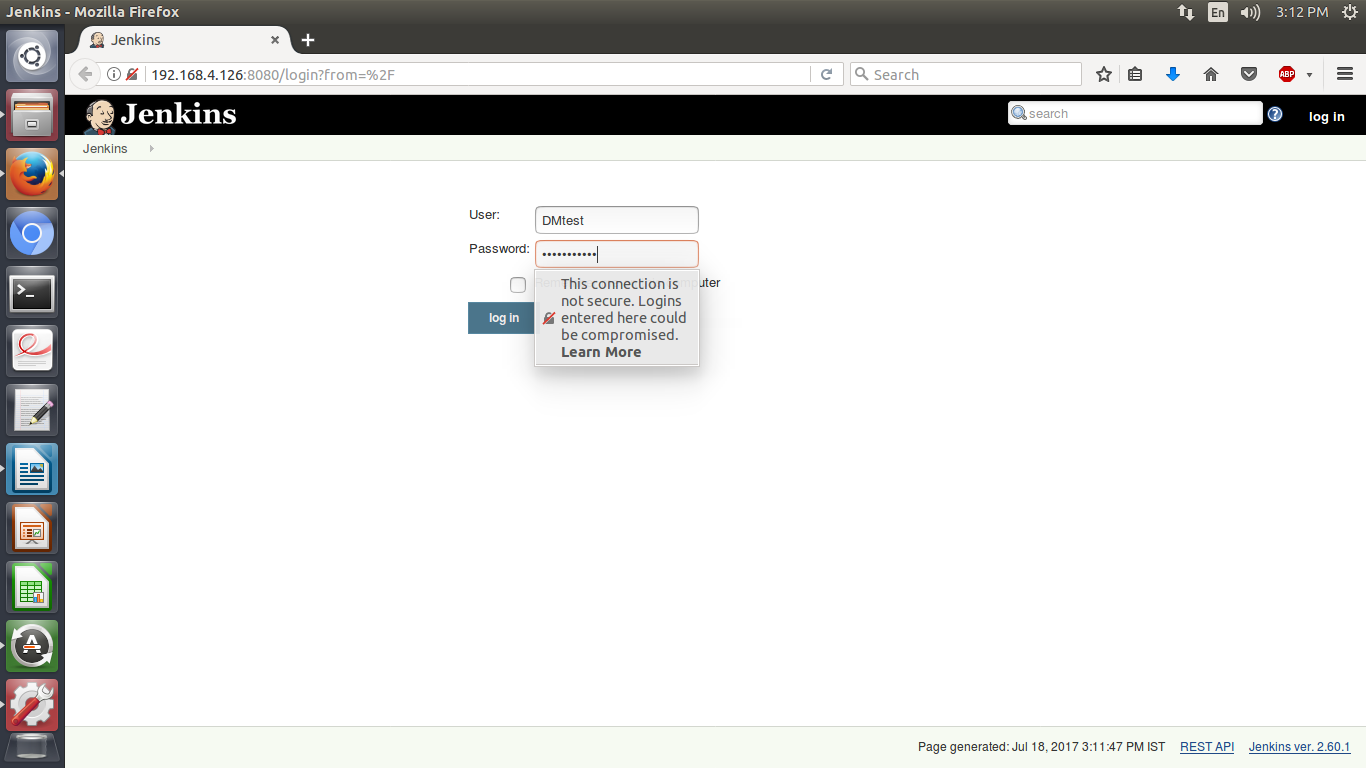
http://localhost:8080/

On the login page, select Create an account to create the users you just gave access.



After sing up Install the plugins

**Sign In**



### 4.1 [Generate ssh key for Jenkins user](http://www.vogella.com/tutorials/Jenkins/article.html" \l "jenkinsconfiguration_ssh)

If you want to access a private Git repository, you need to generate an ssh key-pair. Create a SSH key with the following command.

$sudo -u jenkins ssh-key gen

The public key must be uploaded to the service you are using.

jenkins@VTLW016L:~$ scp id\_rsa.pub [gitrepository@serverIP](mailto:gitrepository@serverIP):~/.ssh

## 4.2 [Jenkins management](http://www.vogella.com/tutorials/Jenkins/article.html" \l "jenkinsmanagement)

### 4.2.1 [Plug-in management](http://www.vogella.com/tutorials/Jenkins/article.html" \l "jenkins_pluginmanagement)

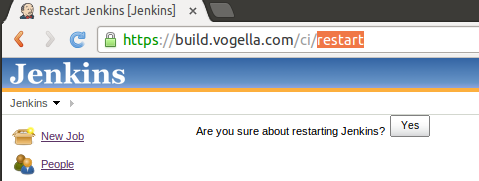
To install plugins in Jenkins select use the Manage Jenkins ▸ Manager Plugins link and search for the plugin you want to install. Select it from the list and select to install it and restart Jenkins.

If you are working with git repository ,install git plugin

For Audio Notification ,Install HTML Audio Notifier plugin

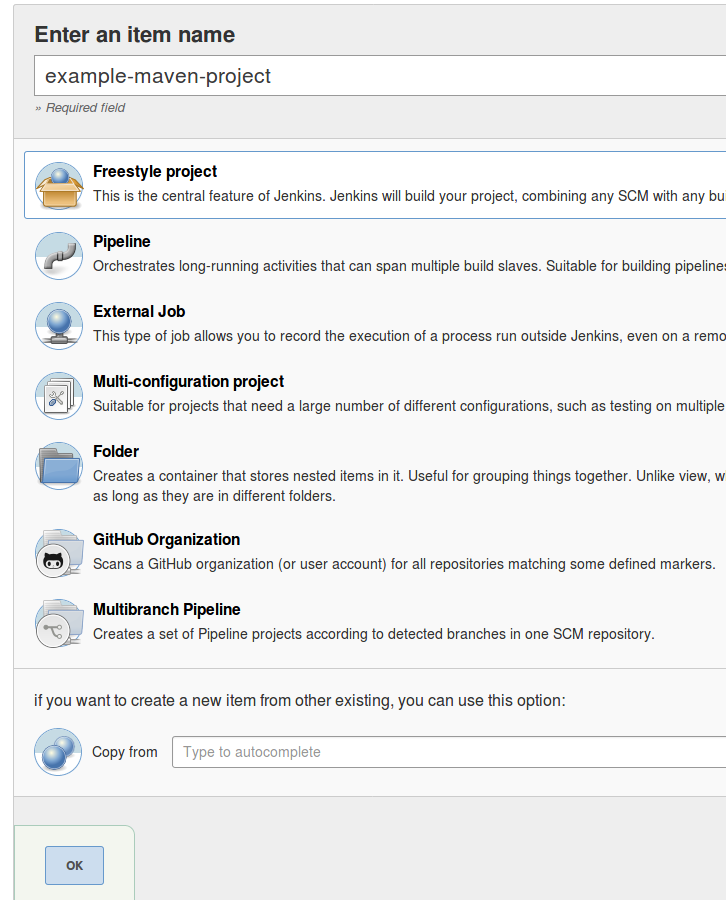
for Email Notification ,Install Email Extended plugin

### 4.2.2 [Restart your Jenkins](http://www.vogella.com/tutorials/Jenkins/article.html" \l "jenkins_pluginmanagement_restart)

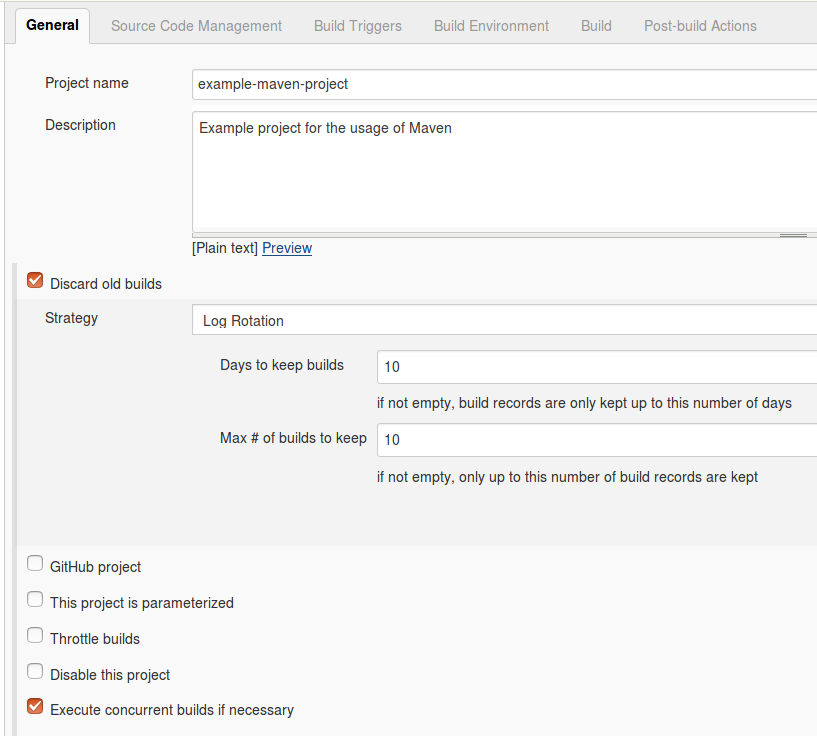


## 4.3 [Setting up a Jenkins job](http://www.vogella.com/tutorials/Jenkins/article.html" \l "setting-up-a-jenkins-job)

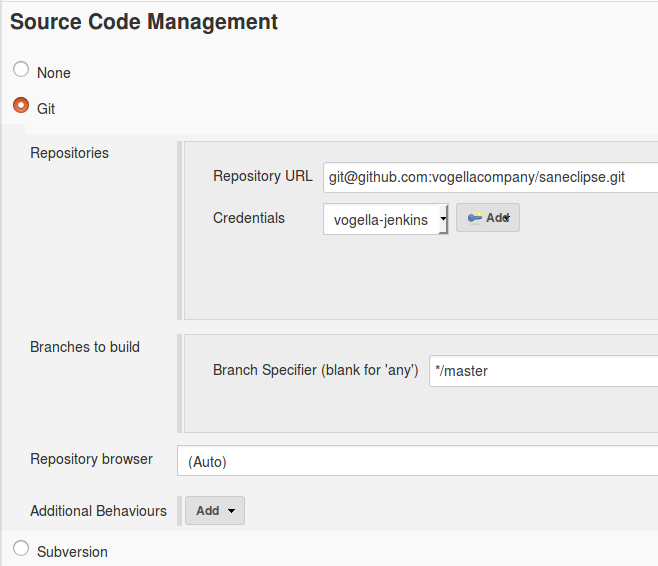
The build of a project is handled via jobs in Jenkins. Select New Item. Afterwards, enter a name for the job and select Freestyle Job and press **OK**.



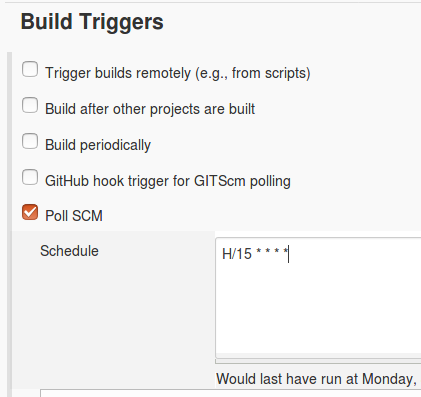
Enter a description for the job and configure how many old jobs should be retained.



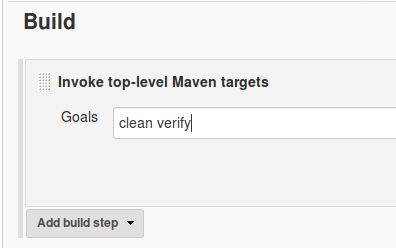
Configure how the source code can be retrieved. If you for example using Git, enter the URL to the Git repository. If the repository is not public, you may also need to configure the credentials if the jenkins pub key is not added to the repository.



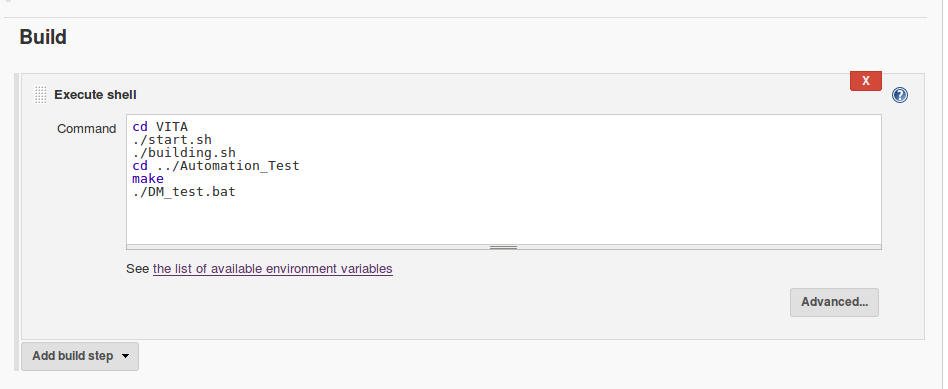
Specify when and how your build should be triggered. The following example polls the Git repository every 15 min. It triggers a build, if something has changed in the repo.

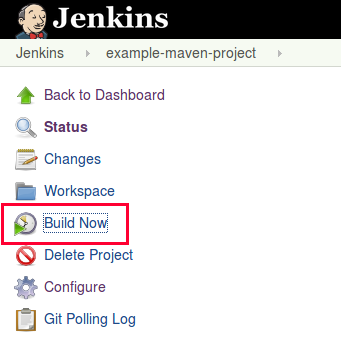


I typically delete the workspace before a build to avoid any side-effect. In the Build section you can add a build step, e.g., a Maven build.



Enter the commands to run Batch files to run

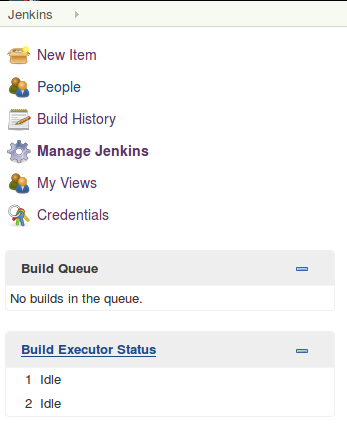
Press **Apply** and **Save** to finish the job definition. Press **Build Now** on the job page to validate the job works as expected.



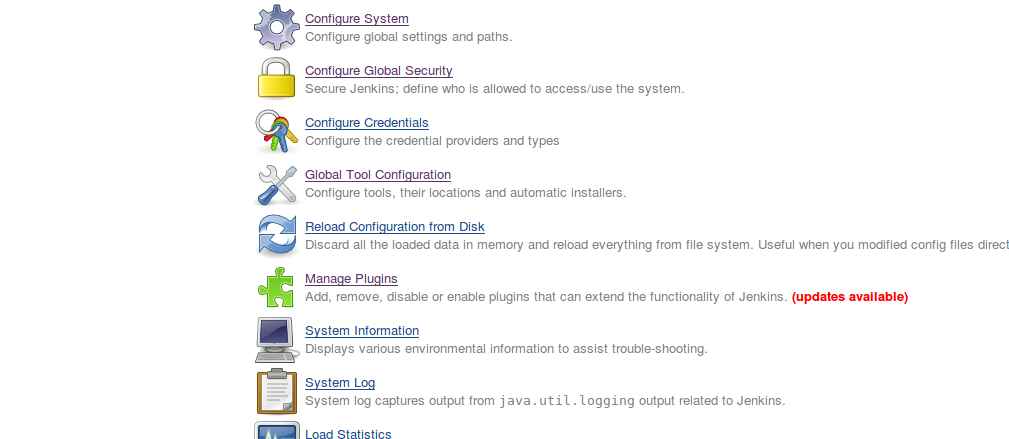
After a while the job should go to green or blue (depending on your configuration), if successful. Click on the job and afterwards on Console Output to see the log file. Here you can analyze the build errors.

**4.4 Mail configuration**

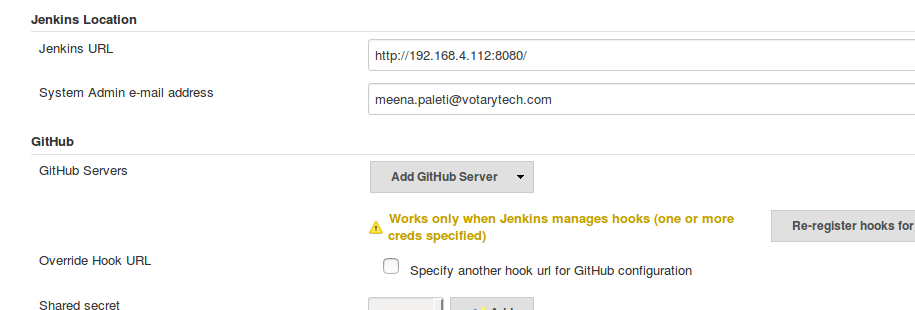
Select Manage Jenkins from Jenkins home page



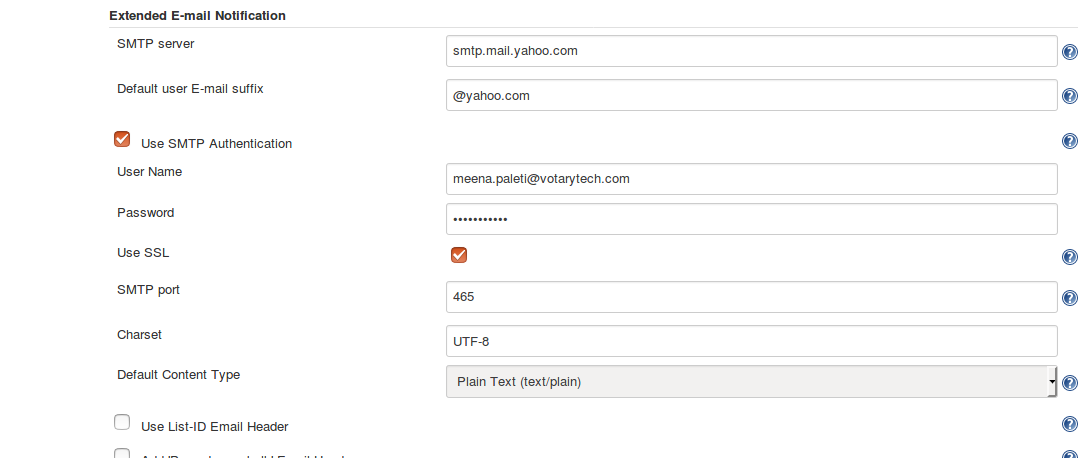
Go for Configure System

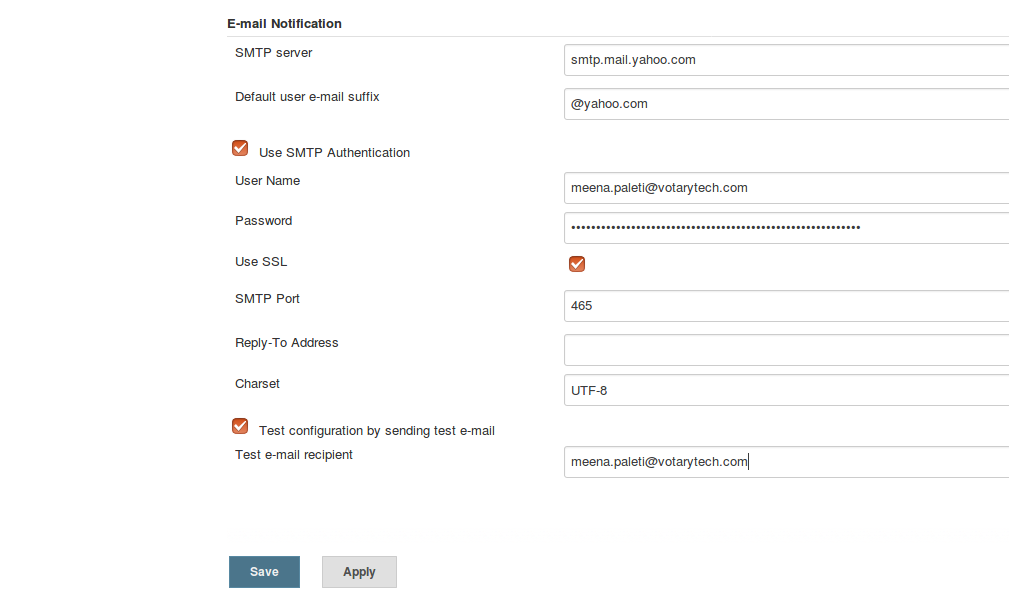


Enter system Admin e-mail address



Find the Extended e-mail Notification ,E-mail Notification and enter the required details.





Enter your mail id and password ,you will find in Advance settings

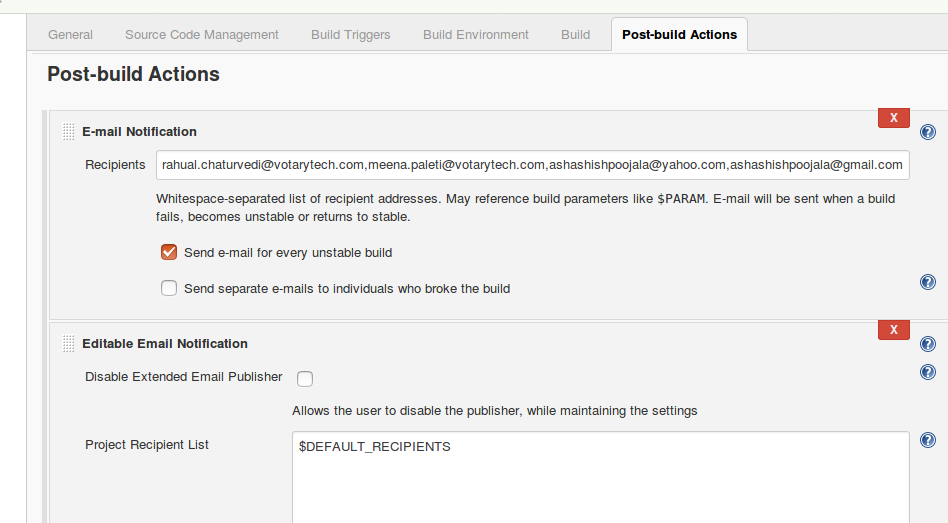
Press **Apply** and **Save**

When Builds Failure/Unstable/Success to get email notification to the Developers /Testers/Reception List ,you need to add their mail-id .

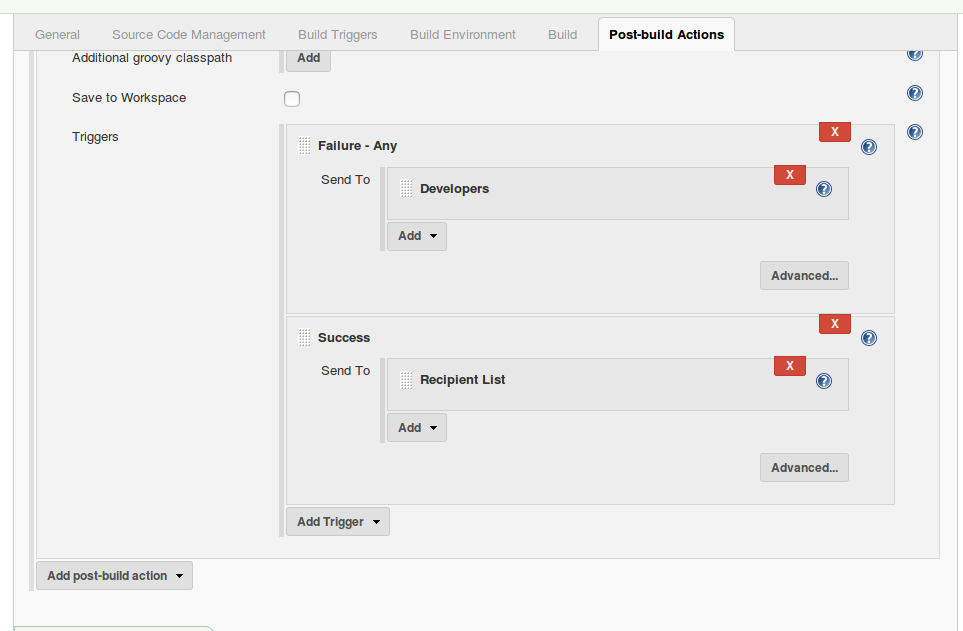
To add email-id,follow the steps

Go back to project

select configure ->post build Actions->add post build action->select Email notification and Editable email notification.



Select add trigger to add more triggering options and go for advance settings to add mail-id for different domain(Developer /Tester /reception list etc)



Press **Apply** and **Save**