



**CPD # 1195**

## Jenkins CI – User Guide

**Description:** This document demonstrates how Jenkins tool provides an easy-to-use continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

**Authors:** Steena Corriea , Soumya Unnikrishnan

**Reviewer:** Rishi Khanna

**Keywords:** Jenkins, CI

**Date:** April 2015



This document is confidential and contains proprietary information, including trade secrets of CitiusTech. Neither the document nor any of the information contained in it may be reproduced or disclosed to any unauthorized person under any circumstances without the express written permission of CitiusTech.

**Document Control**

Version	Jenkins 2.0
Created by	Steena Corriea , Soumya Unnikrishnan
Reviewed by	Rishi Khanna
Date	08/01/2015



Contents

1.1 Project Summary ..... 4

1.2 Pre-requisites ..... 4

1.3 How to install Jenkins as a Windows service? ..... 4

1.4 Pre-setup to run jobs from Jenkins (Git as a repository and Maven as a build tool) ..... 7

1.5 How to create a new job in Jenkins ? ..... 9

1.6 How to start a job in Jenkins ? ..... 11

2.0 Executing a simple command line using Jenkins ..... 13

2.1 Configuring the HTML reports in Jenkins ..... 16

2.2 Jenkins and its uses ..... 20

2.3 Downloads ..... 20



## 1.1 Project Summary

One of the most important aspects of the software development is the software quality, not just in the final product but also in the process of the development. In order to detect failures early we need a system that can build test and deploy software as and when required. Jenkins is a tool that allows us to do that.

We use this tool even for building iOS or Android applications. Jenkins builds the project with every change made and if the build fails, it gives a warning.

Another feature is the execution of automated tests created for both Web and Android applications. If any of these tests fail, while building applications, it will provide necessary warnings

Jenkins monitors the whole build process and provides reports and notifications to alert developers on success or errors.

## 1.2 Pre-requisites

- Install JDK 1.7 and set JAVA\_HOME environment variable
- Install Git client executables and set path variable for Git
- Download Apache maven 3.2.1 and set M2\_HOME environment variable

## 1.3 How to install Jenkins as a Windows service ?

On Windows operating system Jenkins can be configured to work as a service so it starts up automatically without requiring user log in.

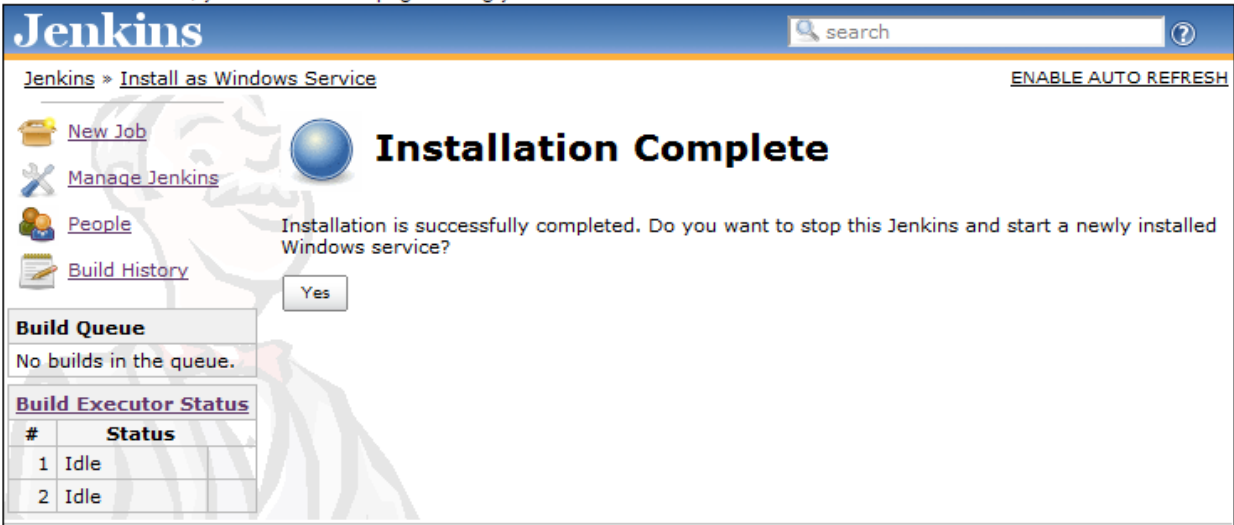
- Download the latest Jenkins war file from Jenkins home page i.e. <http://jenkins-ci.org/>
- Open command prompt and type the following command and enter `>java -jar Jenkins.war`
- Wait until the following information is shown on the command prompt - INFO: Jenkins is fully up and running.
- You should run the command above and then open up a browser and browse to your new running Jenkins's home page at <http://localhost:8080>
- You will see a link for "Manage Jenkins" that looks like:



- Clicking this link shows you the installation screen



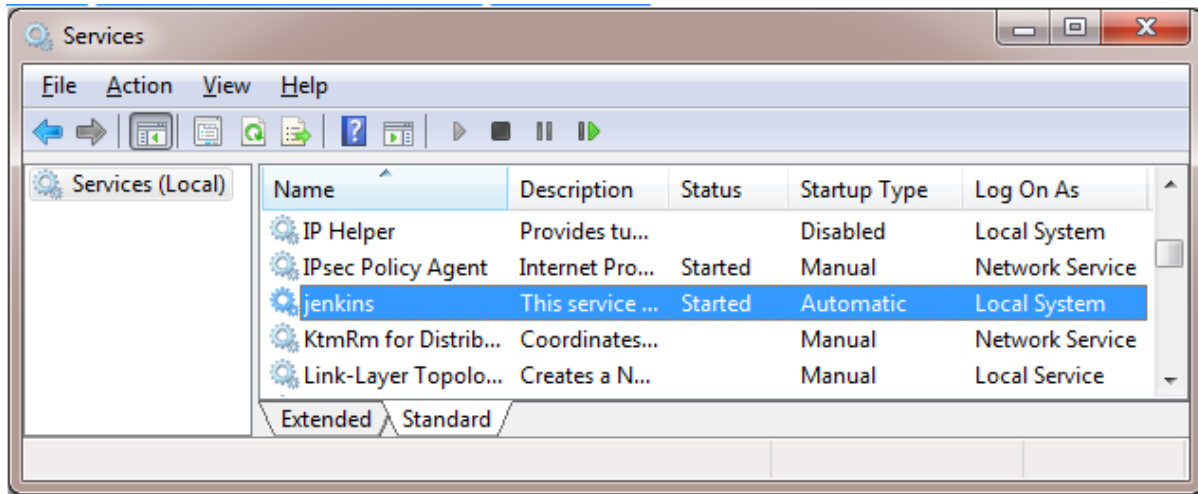
- Choose the directory where Jenkins will be installed (directory must exist. If not create it before hand).
- This will become Jenkins\_HOME and will be used to store data files and programs. Upon successful completion of the installation, you should see a page asking you to restart Jenkins



- This is to re-launch Jenkins as a newly installed Windows service. When you click yes, you'll be asked to wait until restart completes:

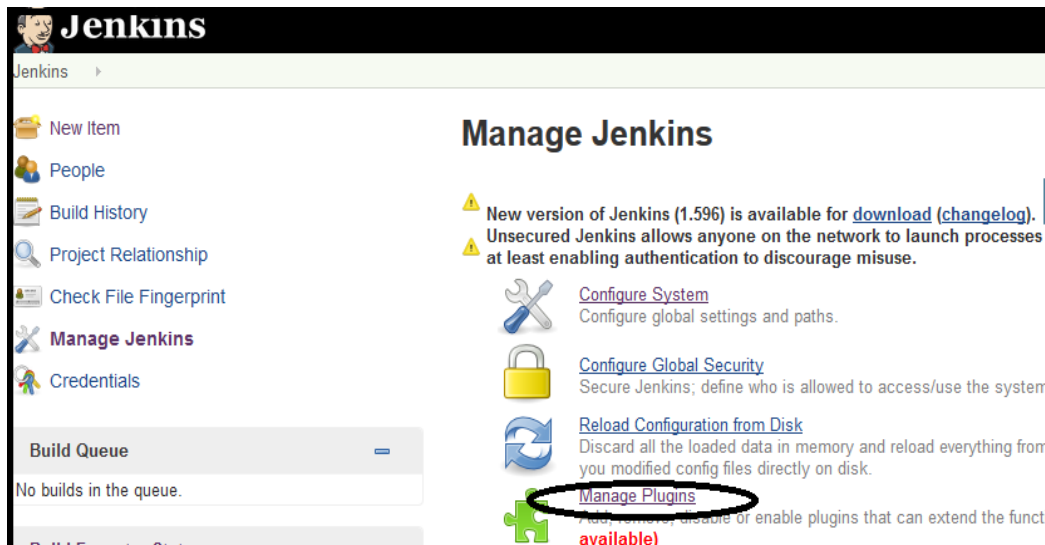


- At this point you can use the service manager to confirm that Jenkins is running as a service.

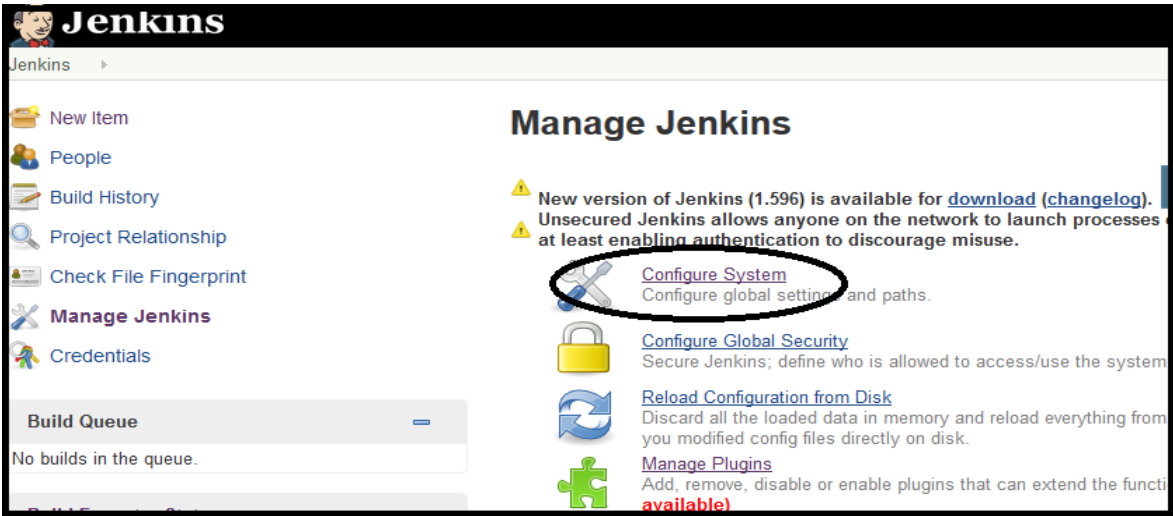


#### 1.4 Pre-setup to run jobs from Jenkins (Git as a repository and Maven as a build tool)

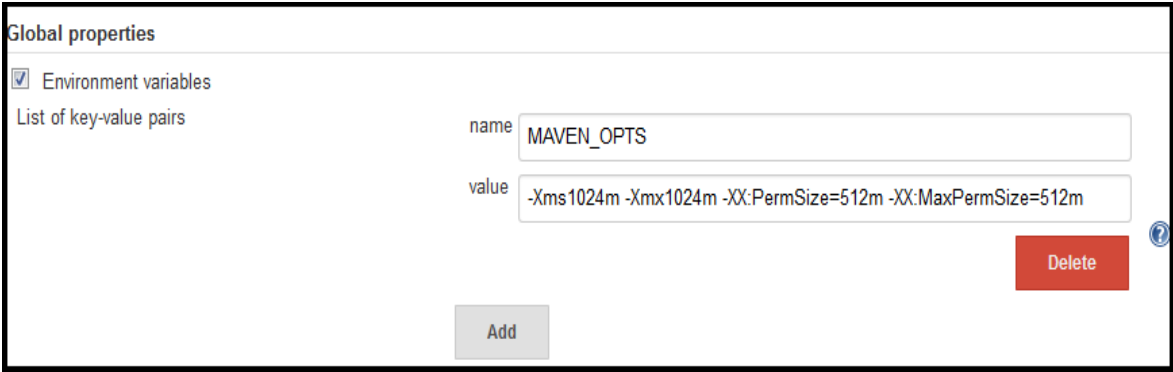
- Install Git plugin for Jenkins - Click on "Manage Plugins" on Manage Jenkins Page. Install GitHub plugin which integrates GitHub to Jenkins.



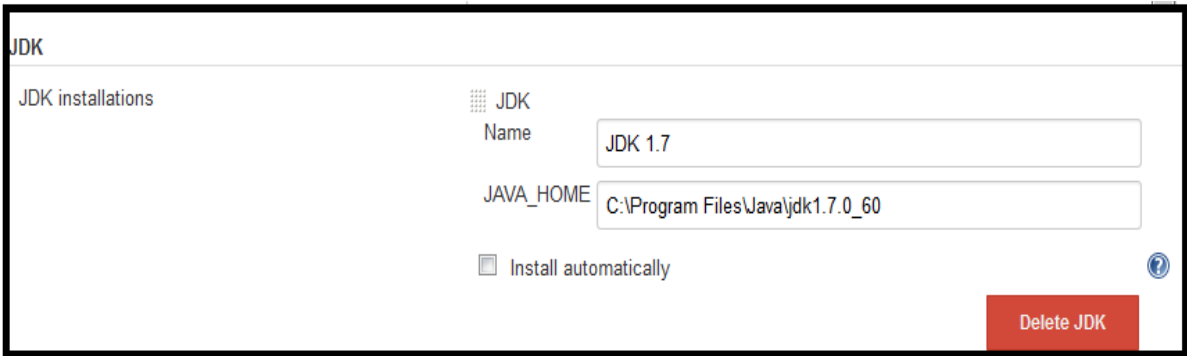
- Configure JDK and Maven to build the project
- Click on manage Jenkins > Configure System



- Setup global variables - MAVEN\_OPTS



- Setup JAVA\_HOME - click on JDK installations -> ADD JDK -> setup JAVA\_HOME to point to JDK



- Setup Path to Git - click on Git installations -> ADD Git -> setup path to Git executables to point to Git exe.





- Setup MAVEN\_HOME - click on Maven installations -> ADD Maven -> setup MAVEN\_HOME to point to apache maven

- Setup Jenkins URL location

- Apply above configurations and save

## 1.5 How to create a new job in Jenkins ?

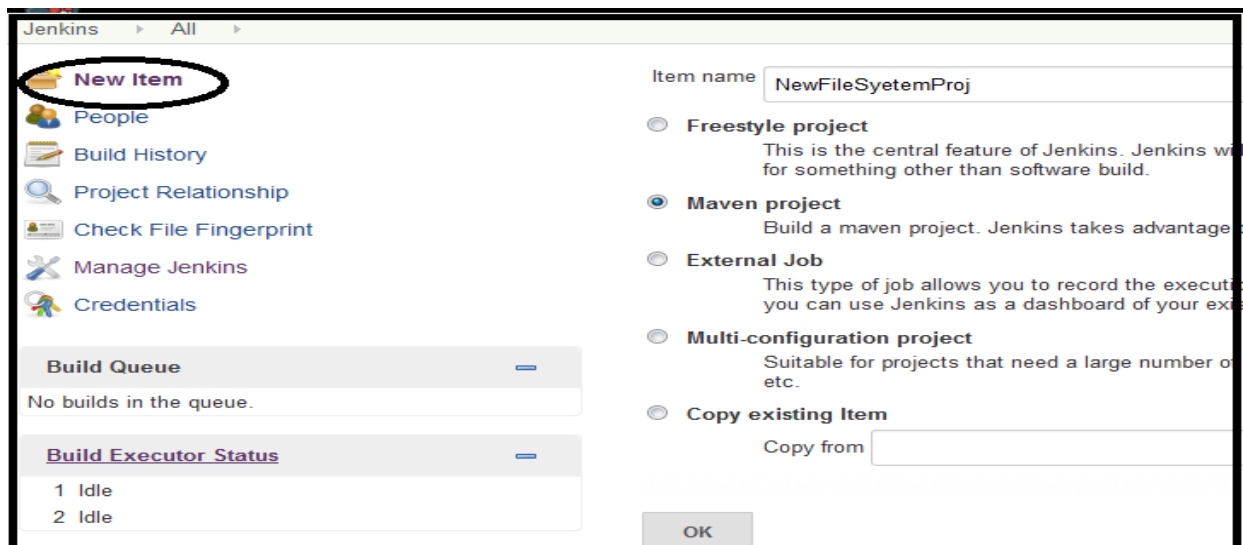
Build jobs are at the heart of the Jenkins build process. You can think of Jenkins build job as a particular task or step in your build process. This may involve simply compiling your source code and running your unit tests. Or you might want a build job to do other related tasks, such as running your integration tests, measuring code coverage or code quality metrics, generating technical documentation, or even deploying your application to a web server. A real project usually requires many separate but related build jobs.

Jenkins supports several different types of build jobs. The two most commonly used are –

- Freestyle builds - allows you to configure just about any sort of build job: they are highly flexible and very configurable.
- Maven 2/3 builds - Maven 2/3 builds understands the Maven project structure, and can be used to set up Maven build jobs with less effort with additional features.

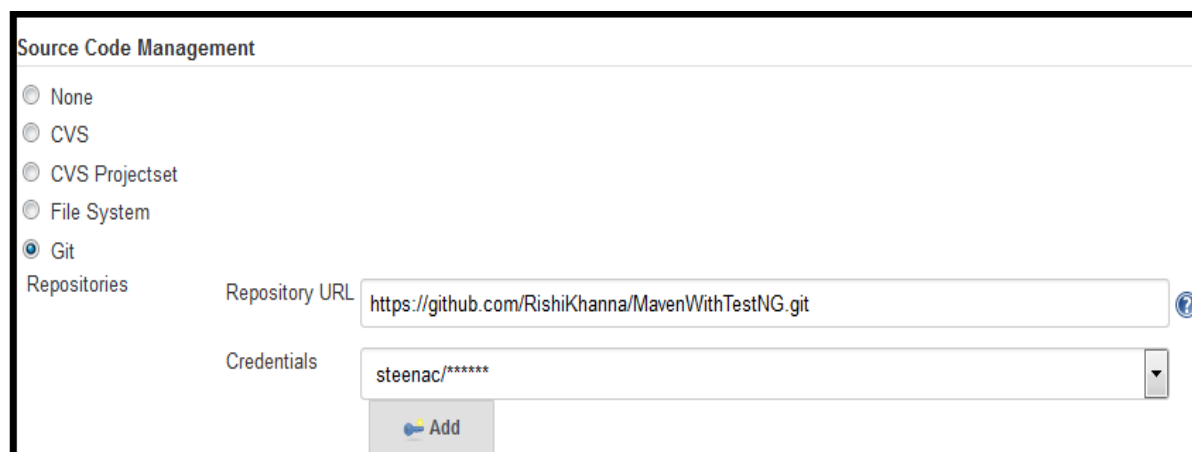
### Steps to create a new job in Jenkins -

- Click on "Create new jobs" link -> Provide a job name and select 'Maven Project' -> Click OK button



The screenshot shows the Jenkins 'New Item' configuration page. The 'New Item' button in the left sidebar is circled. The 'Item name' field is set to 'NewFileSyetemProj'. Under the 'Project type' section, 'Maven project' is selected. Below this, there are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing 1 Idle and 2 Idle executors). An 'OK' button is at the bottom right.

- Update description and scroll down to the "Source code management section"
- Select the source code management tool that is been used, here its Git -> enter Git Repository URL and credentials.



The screenshot shows the 'Source Code Management' section of the Jenkins configuration. The 'Git' radio button is selected. Below it, the 'Repository URL' is set to 'https://github.com/RishiKhanna/MavenWithTestNG.git' and the 'Credentials' dropdown is set to 'steenac/\*\*\*\*\*'. An 'Add' button is at the bottom.

- Enter Root POM value and maven Goals and options under Build.



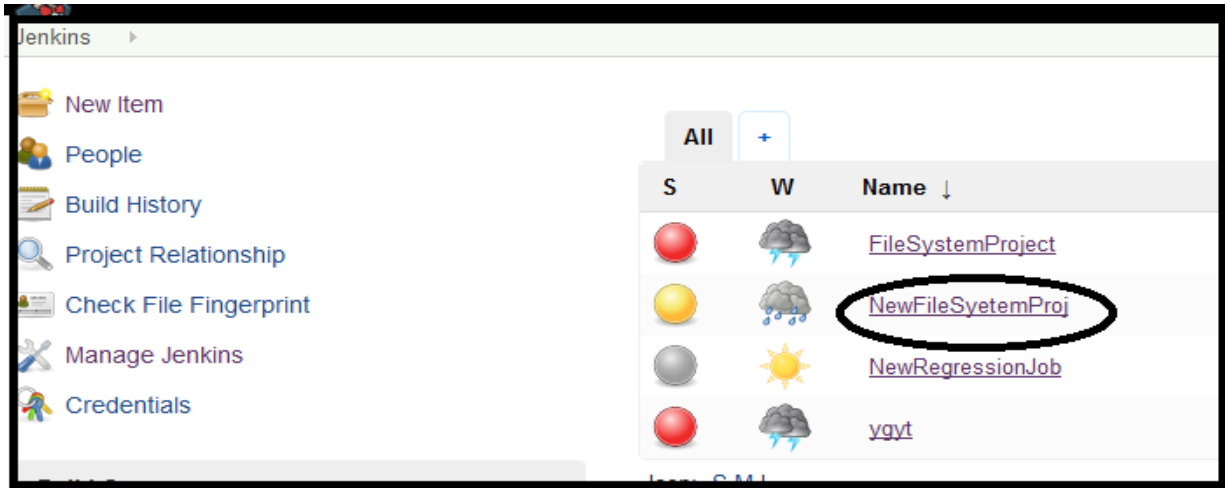
The screenshot shows the 'Build' section of the Jenkins configuration. The 'Root POM' field is set to 'pom.xml' and the 'Goals and options' field is set to 'clean install'. An 'Advanced...' button is at the bottom right.

- Apply job level settings and save.

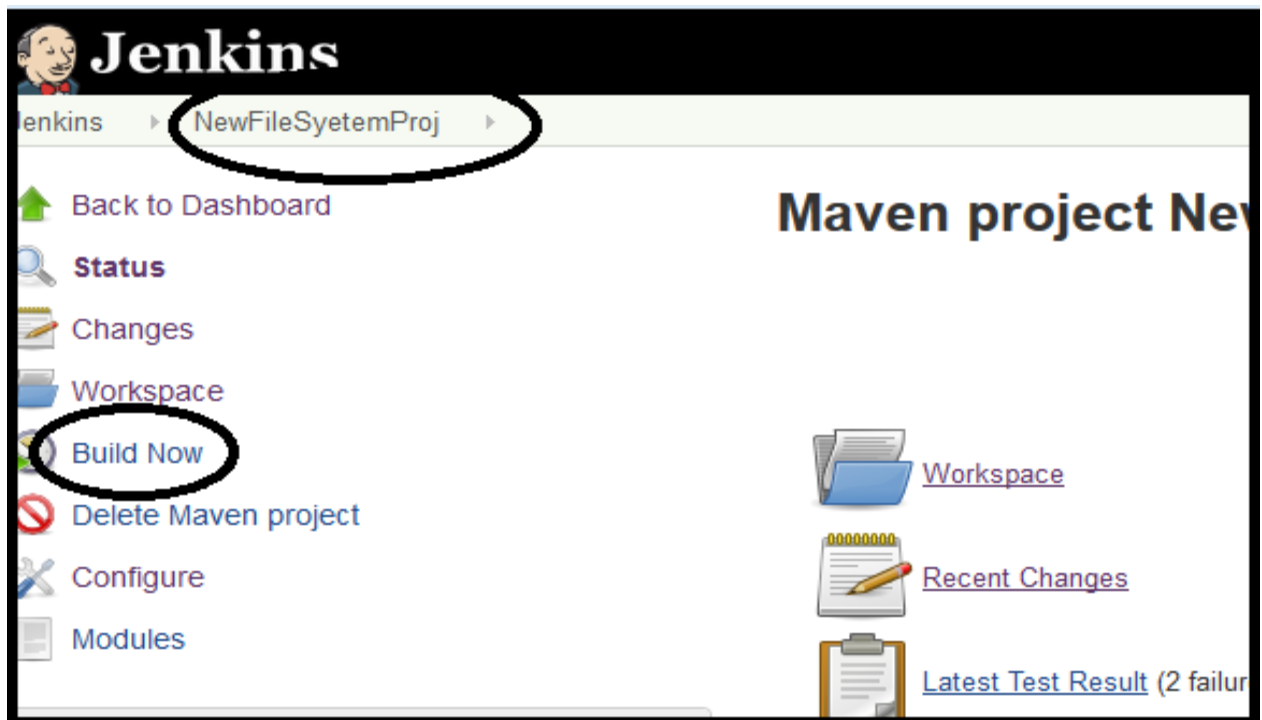


## 1.6 How to start a job in Jenkins ?

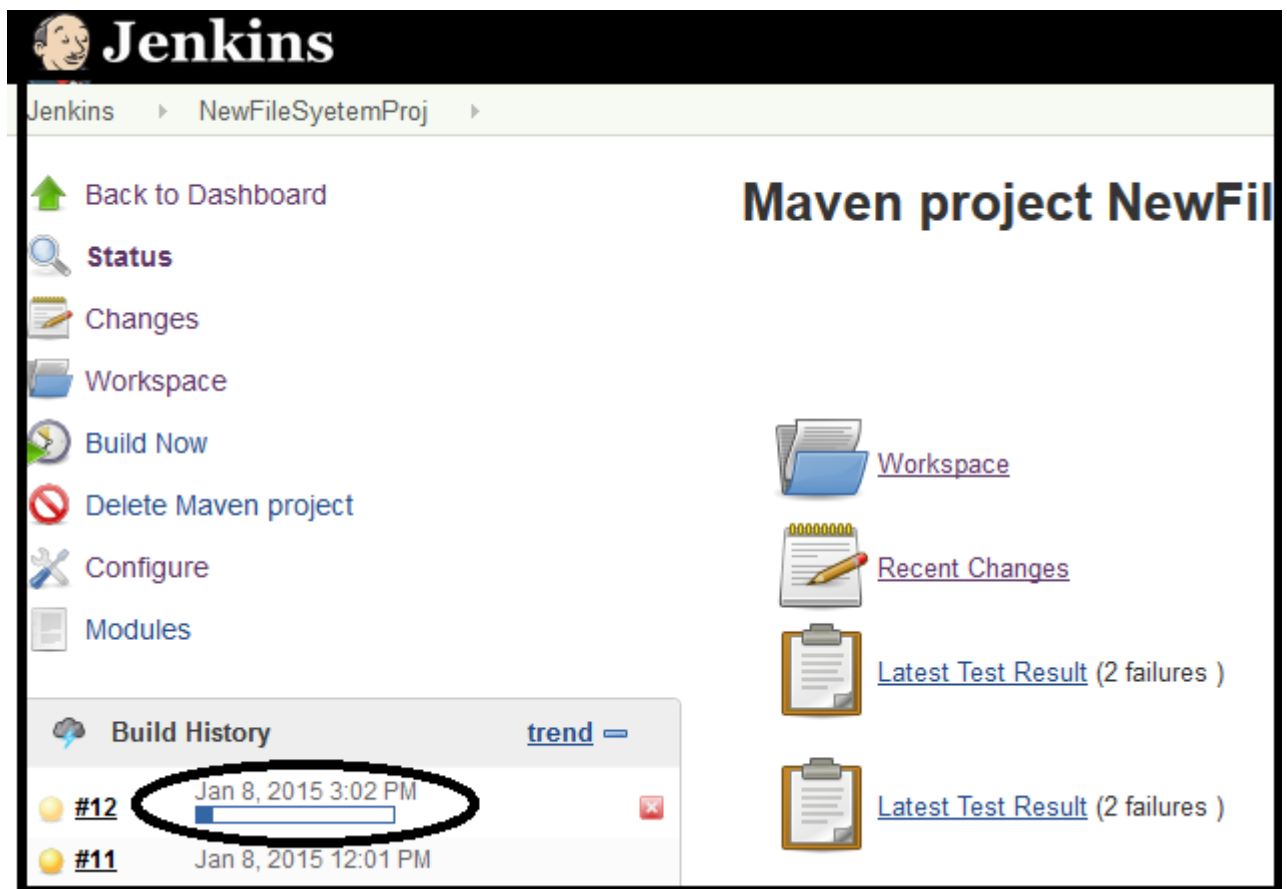
- Go to Jenkins main page and you will see all the available jobs.



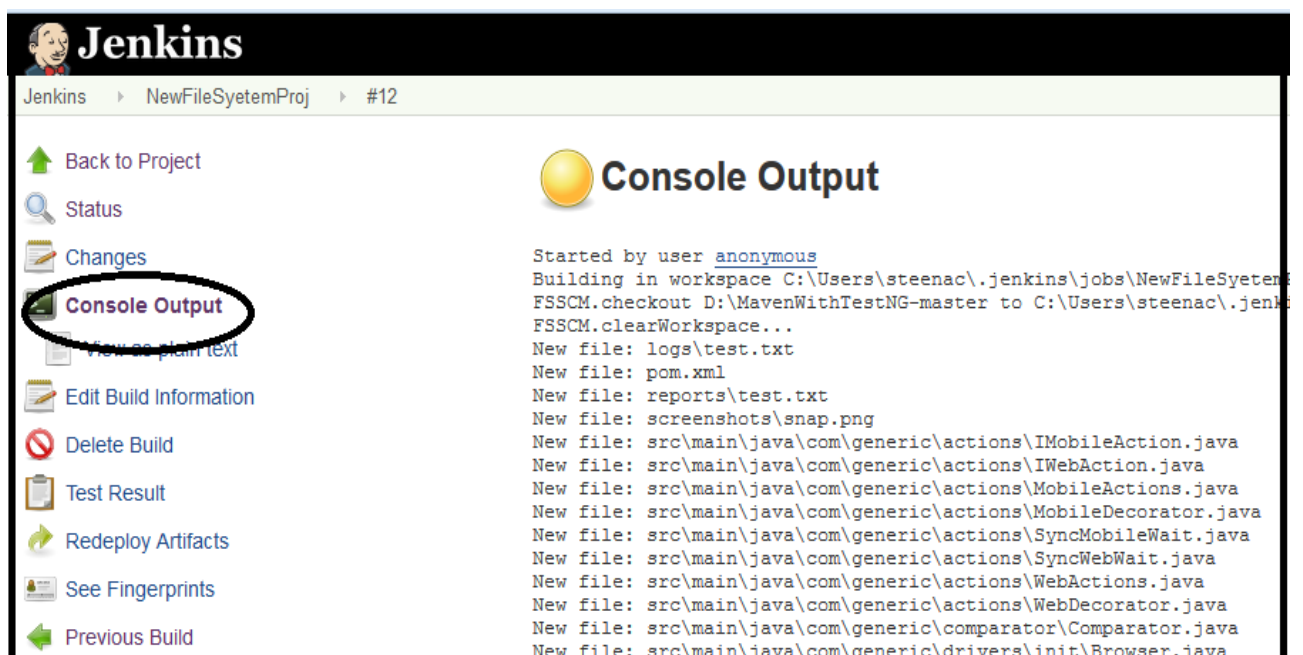
- Click on a job of your choice -> click on "Build Now" option for the respective job (pls. see below screenshot)



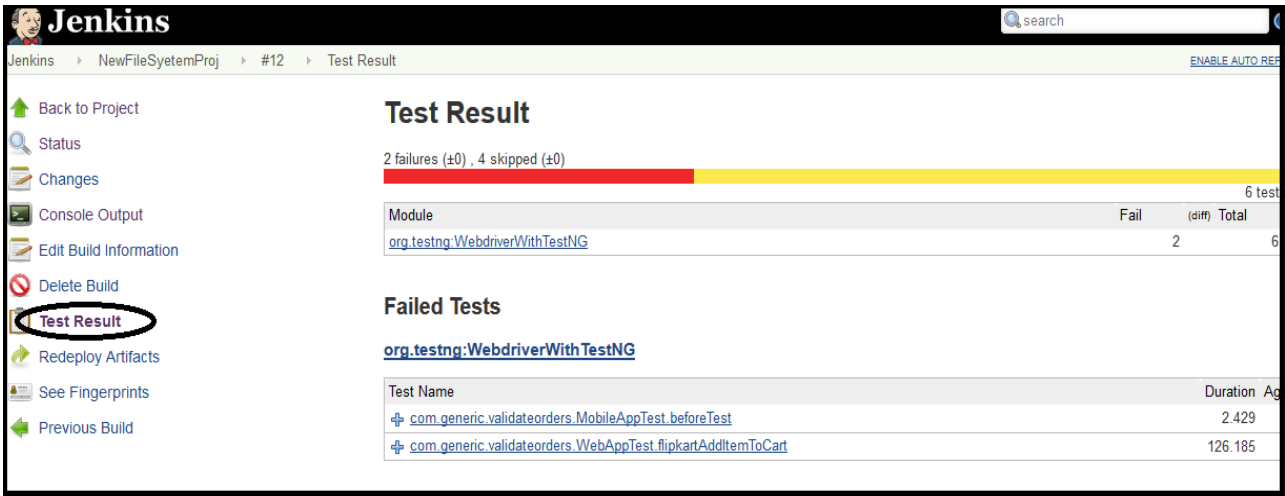
- Once build has started you can see the build progress under Build History section. It will show build number and build start timestamp



- In order to check the build progress Click on the respective build under Build History -> click Console output

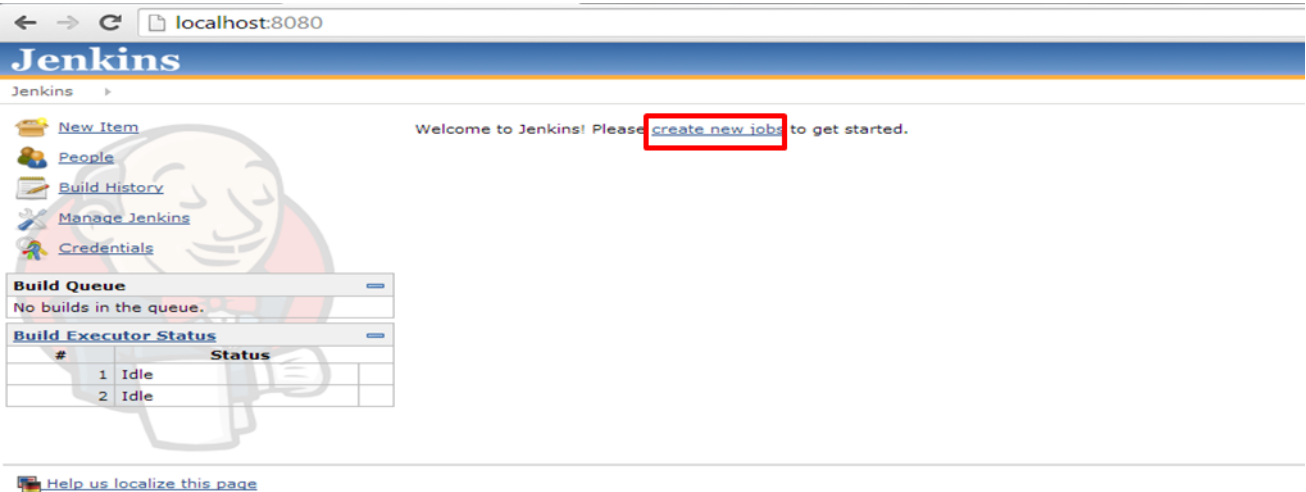


- Once the build is complete, test Results could be seen Under Test Result section.

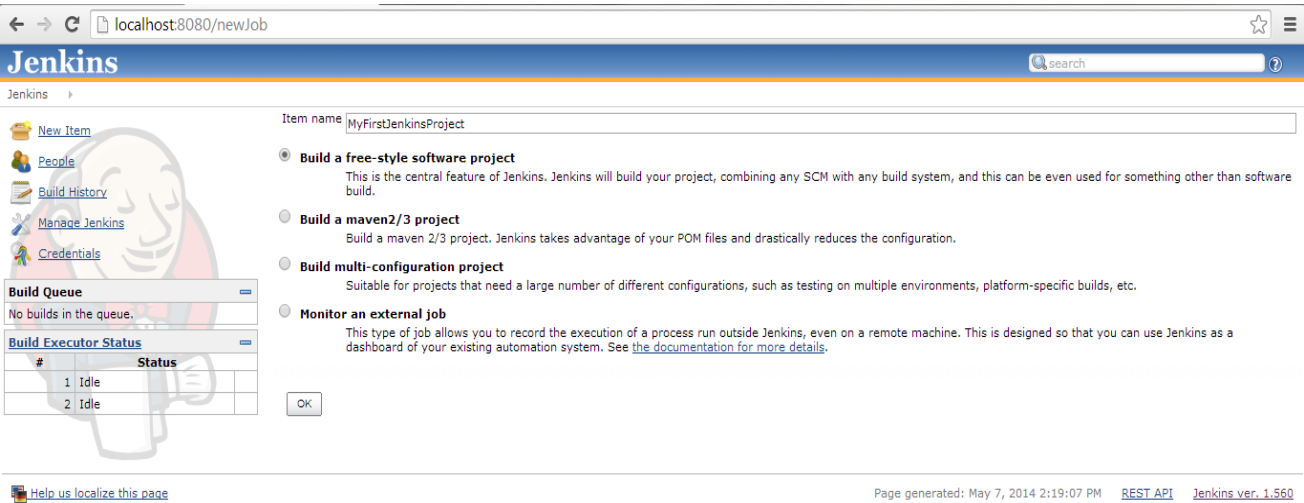


2.0 Executing a simple command line using Jenkins

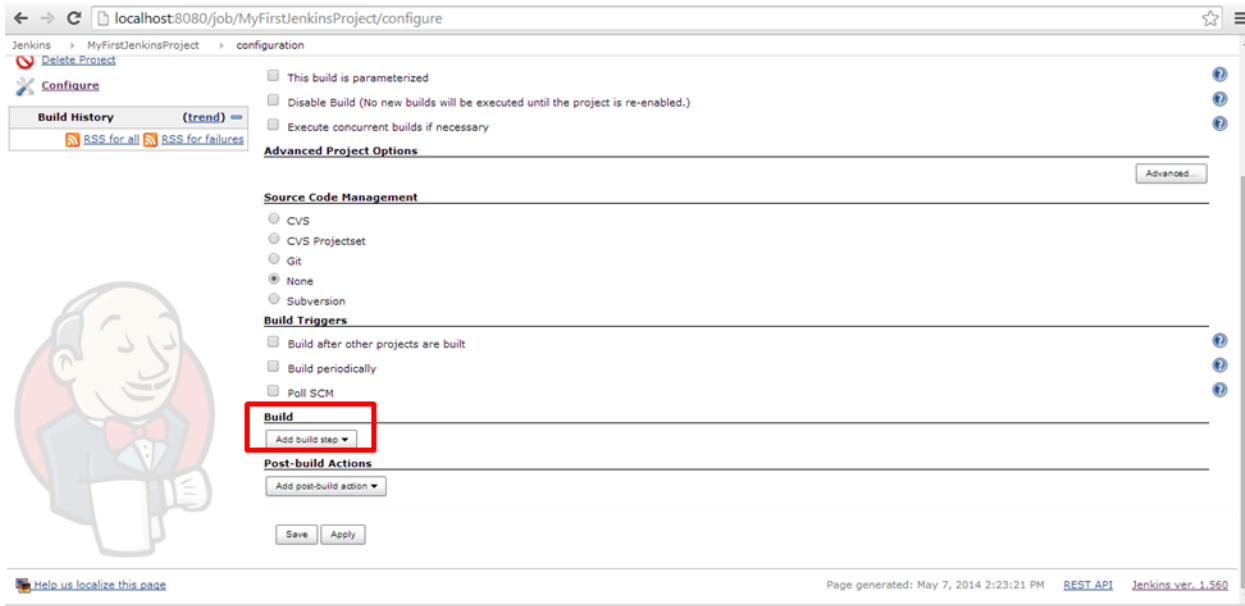
- Click on create new jobs link



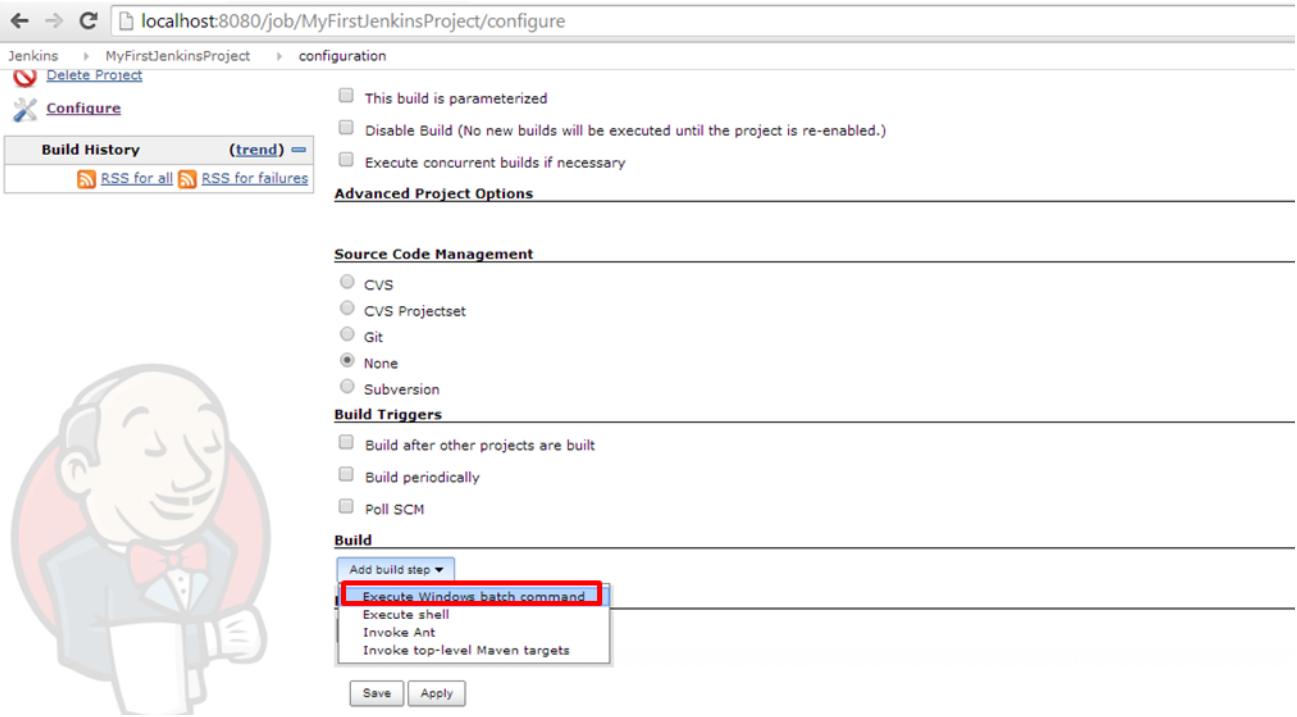
- Provide an Item name (e.g. MyFirstJenkinsProject) and select Build a free-style software project. Click on OK button.



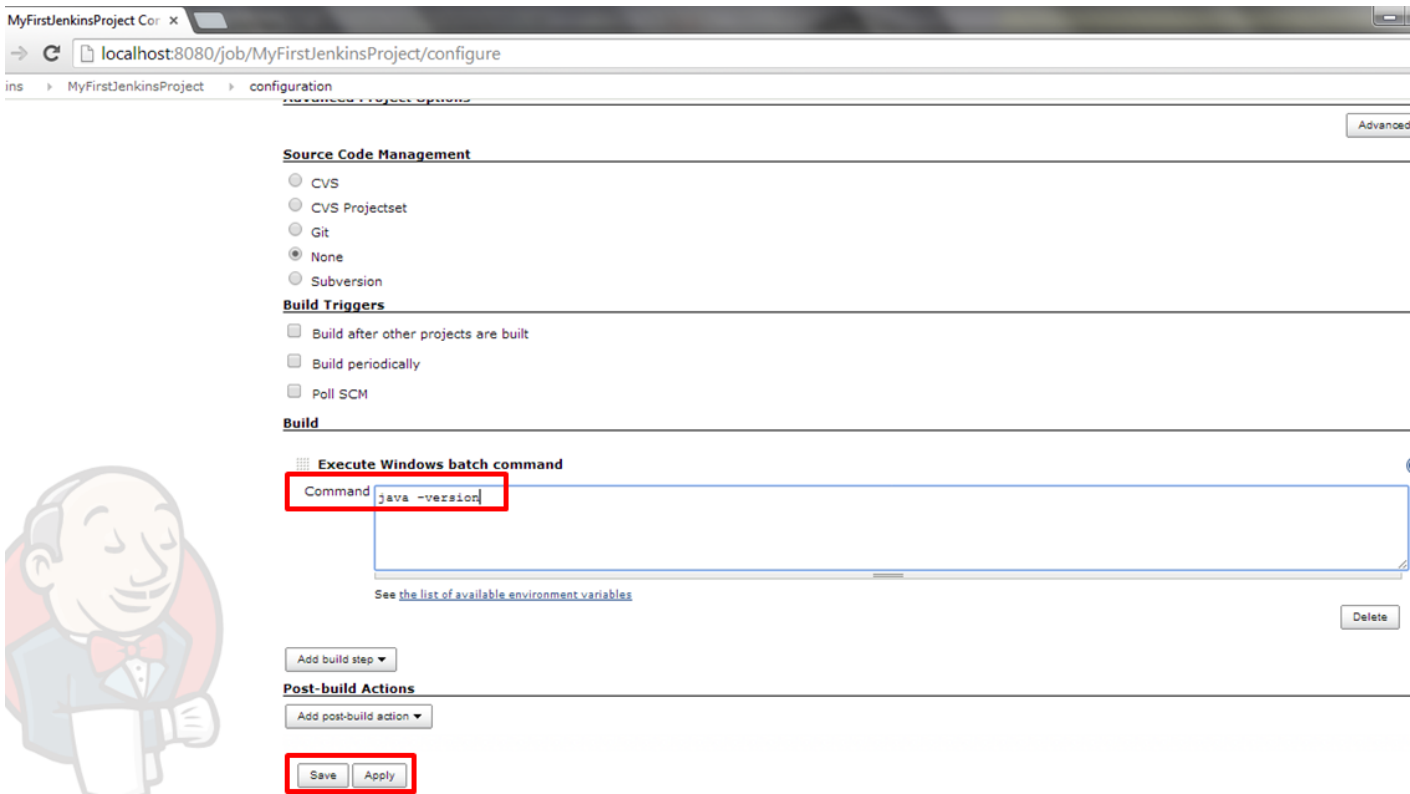
- Scroll down to the Build section and click on Add build step button



- From the menu list, select option "Execute Windows batch command"



- Enter the command (java -version) under the “Execute Windows batch command” box. Click on Apply and Save button.





2.1 Configuring the HTML reports in Jenkins

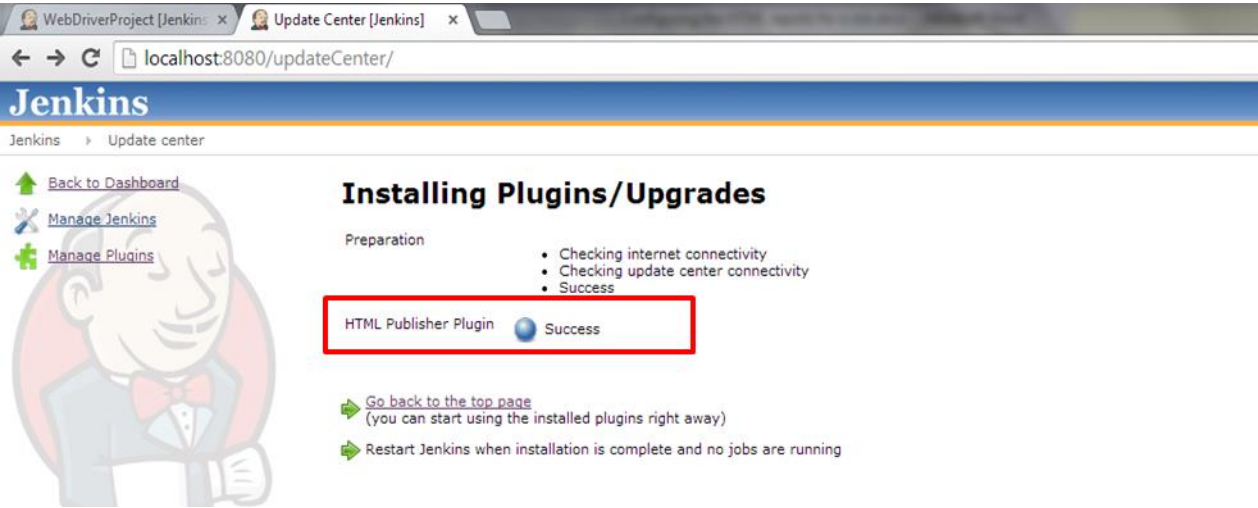
- Navigate to URL: `http://localhost:8080/pluginManager/available` and search for HTML Publisher Plugin.



- Click on the Install without restart button.



- Wait until the status of the HTML Publisher Plugin shows: Success



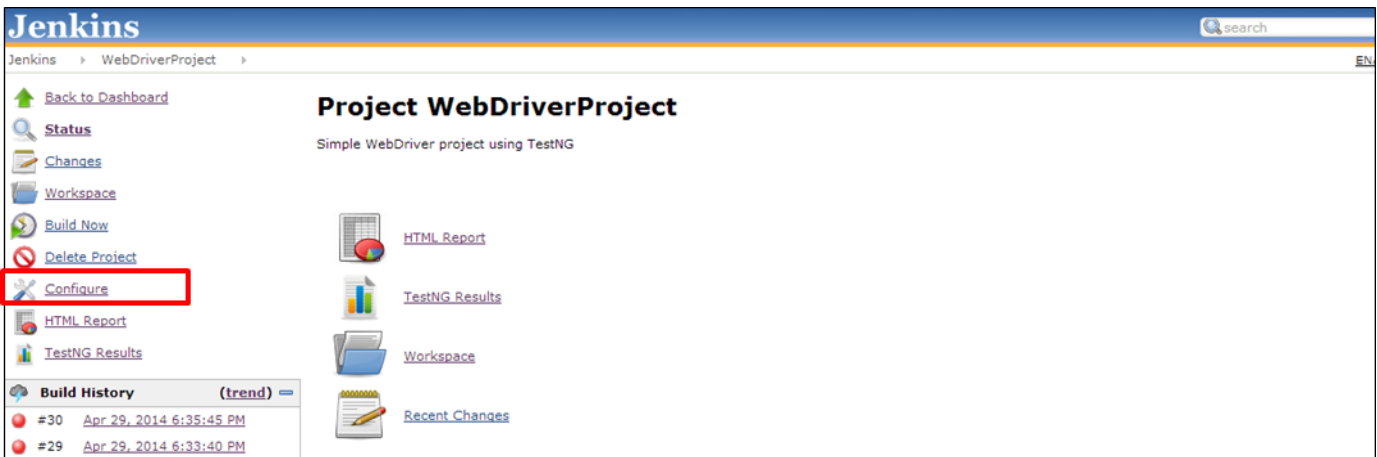
- Open the Job which is to be configured for HTML reports; by clicking on the name of the item in the Dashboard

All						
S	W	Name	Last Success	Last Failure	Last Duration	
		WebDriverProject	4 hr 28 min - #28	4 hr 24 min - #30	25 sec	

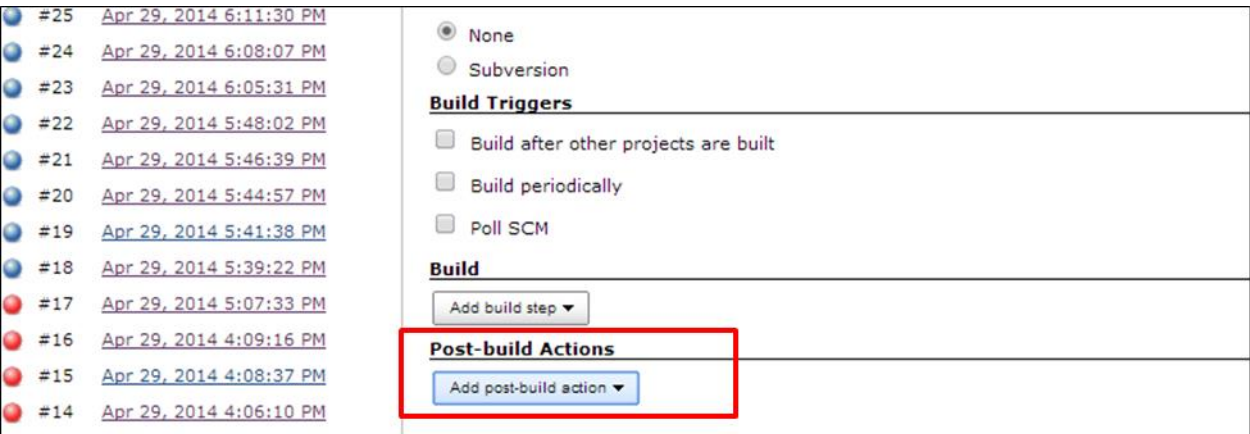




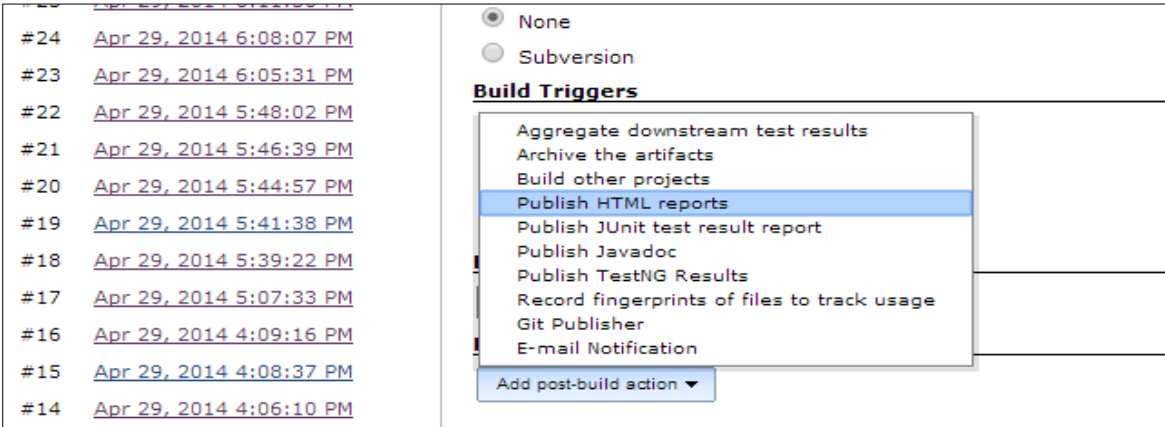
- Click on the Configure link on the left side of the screen



- Navigate to the Post-build Actions section and click on Add post-build action drop down



- Select Publish HTML reports option from the drop-down.





- Click on Add button under Publish HTML reports

- Enter the following details:
  - HTML Directory to archive** - Path to the directory containing the html reports in the "HTML directory to archive" field.
  - Index Pages** – HTML Pages to display (default index.html); one can specify multiple comma-separated and each will be a tab on the report page.
  - Report title** – Name provided in the report title will be used to provide a link to the report.

**Note:** By default, only the most recent HTML report will be saved, but if you'd like to be able to view HTML reports for each past build, select

"Keep past HTML reports."

- Click on Apply and then Save button.

- After saving the configuration, run build once. The published HTML reports are available to view from within Jenkins with convenient links in the dashboard.



# Jenkins

[Jenkins](#) > [WebDriverProject](#) >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[HTML Report](#)

[TestNG Results](#)

## Project WebDriverProject

Simple WebDriver project using TestNG

[HTML Report](#)

[TestNG Results](#)

[Workspace](#)

[Recent Changes](#)

**Build History** [\(trend\)](#)

#32 [Apr 30, 2014 12:14:37 PM](#)

#31 [Apr 30, 2014 12:12:22 PM](#)

Sample Reports:

localhost:8080/job/WebD...HTML Publisher Plugin - x

localhost:8080/job/WebDriverProject/25/HTML\_Report/

[Back to WebDriverProject](#) [index](#) [Zip](#)

Test results				
Suite	Passed	Failed	Skipped	testng.xml
Total	1	0	0	
<a href="#">TestAll</a>	1	0	0	<a href="#">Link</a>

localhost:8080/job/WebD...HTML Publisher Plugin - x

localhost:8080/job/WebDriverProject/28/HTML\_Report/

[Back to WebDriverProject](#) [index](#) [Zip](#)

Test results

1 suite

org.rishik.jenkins.Application

MyTestVlass

All suites

TestAll

Info

- C:\Users\rishik\workspace\TestForJenkins\testng.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 1 method, 1 passed
- Passed methods (hide)
- MyTestVlass

accelerating innovation in healthcare

CitiusTech



## 2.2 Jenkins and its uses

- It is proven technology that is deployed at a large scale in many organizations.
- It is an open source technology, so the code is open to review and has no licensing costs.
- It has a simple configuration through a web-based GUI, which speeds up Job creation, improves consistency, and decreases the maintenance costs.
- It is a master slave topology that distributes the build and testing effort over slave servers with the results automatically accumulated on the master. This topology ensures a scalable, responsive, and stable environment.
- There is no fuss in installation, as installation is as simple as running only a single download file named jenkins.war.
- It has over 400 plugins supporting communication, testing, and integration to numerous external applications (<https://wiki.jenkins-ci.org/display/JENKINS/Plugins>).
- Though highly supportive of Java, Jenkins also supports other languages.
- Jenkins is an agile project; you can see numerous releases in the year, pushing improvements rapidly (<http://jenkins-ci.org/changelog>). There is also a highly stable long-term support release for the more conservative hence; there is a rapid pace of improvement.
- Jenkins pushes up code quality by automatically testing within a short period after code commit, and then shouting loudly if build failure occurs.

## 2.3 Downloads

- Jenkins war file can be downloaded from <http://jenkins-ci.org/>
- Java installation - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Maven installation - <http://maven.apache.org/download.cgi>