Jenkins

Continuous integration is a process in which all the development activities (every day activities) are integrated at a given point of time by compiling and building the project and well tested. The basic idea behind the continuous integration is to ensure that there are no compilation issues at the end of the day by numerous check ins made by the developers in a team. Also, this would enable to identify any compilation issues at the early stages of the development process.

In this process, all the developers activities are collaborated and merged at the central system (repository at which all the check ins are made). The main aim again here in this process is to eliminate the “integration problems”. Each and every integration is automatically built, deployed and tested thoroughly.

#### **Advantages of CI**

1. Reduced level of risk.
2. No longer integration process.
3. Lot of bad code smells & bugs can be reduced at the early stages of the development process.
4. Frequent deployment process is easier and faster.
5. Easy to implement the process by using tools like Jenkins (earlier known as Hudson) and cruise control.

#### **What Jenkins does?**

The primary job of the Jenkins is to execute a list of steps configured which is based on a trigger. Below are the list of steps/tasks performed by Jenkins when triggered :

1. Perform the code compilation & build the software with ANT, Maven or Gradle.
2. Run an internal shell script
3. Archive the resultant builds
4. At last start the execution of the integration tests.
5. Monitoring the execution of the above tasks.
6. Provision to stop the build in case of failure in any of the steps.
7. Notifying the user about the success or failure of each build.

#### **What Jenkins focus on?**

Jenkins mainly focuses on 2 important activities :

1. **Continuous build of the software :** Nothing but the creating a continuous integration environment
2. **Monitoring execution of external jobs :** This monitors the jobs defined – CRON jobs.

#### **Features of Jenkins**

1. Easy to install & configure.
2. Permanent link – readable URLs for the build details.
3. Email integration for notifications.
4. JUnit tests reporting.
5. Tagging support for every successful builds.
6. Distributed builds.
7. File fingerprinting for managing dependencies
8. Plugin support

#### **Jenkins Setup**

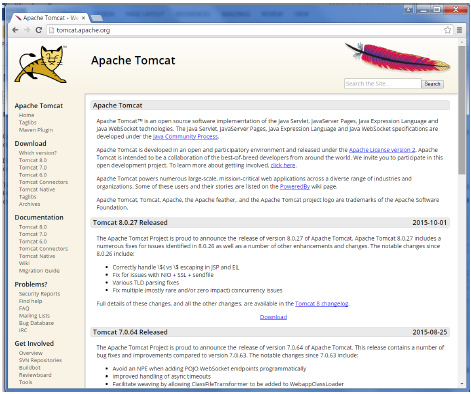
1. Download the latest Jenkins.war from <https://jenkins-ci.org/>. Towards the right top click on the link latest and greatest to download the latest war file.
2. Deploy the **Jenkins.war** file in the local web server of our machine. E.g. Tomcat
3. Start the server and open the browser and hit the URL <http://localhost:8080/jenkins>



Open the command prompt. From the command prompt, browse to the directory where the jenkins.war file is present. Run the following command

D:\>Java –jar Jenkins.war

## Download Tomcat



## Jenkins and Tomcat Setup

Copy the Jenkins.war file which was downloaded from the previous section and copy it to the webapps folder in the tomcat folder.

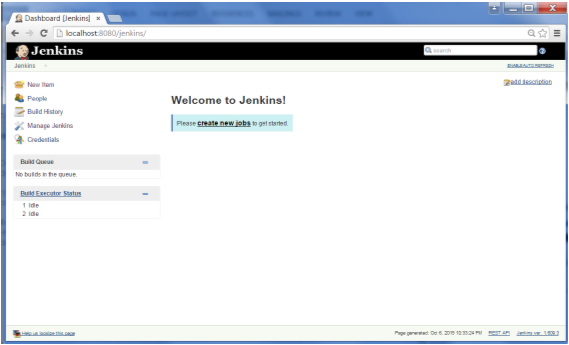
Now open the command prompt. From the command prompt, browse to the directory where the tomcat7 folder is location. Browse to the bin directory in this folder and run the start.bat file

E:\Apps\tomcat7\bin>startup.bat

Once the processing is complete without major errors, the following line will come in the output of the command prompt.

INFO: Server startup in 1302 ms

Open the browser and go to the link − **http://localhost:8080/jenkins**. Jenkins will be up and running on tomcat.

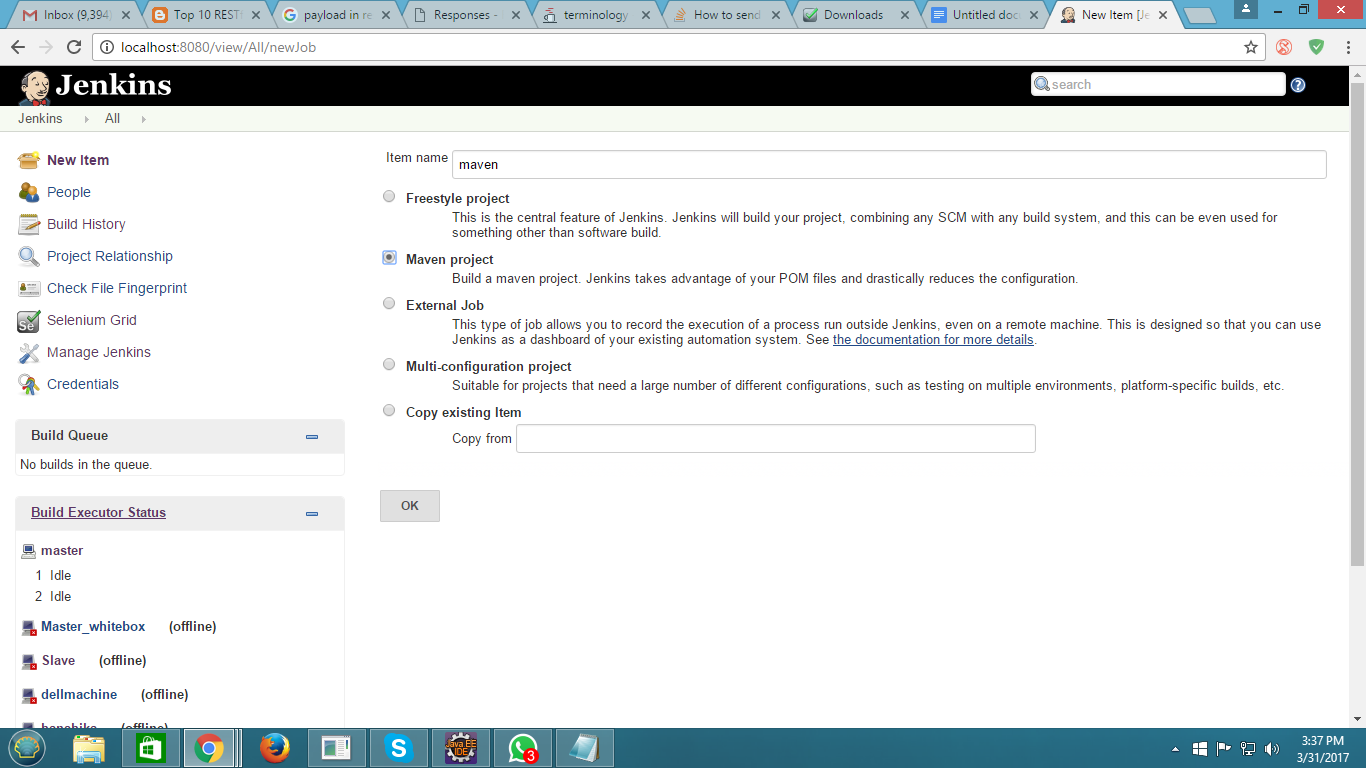


Master machine -source code -git

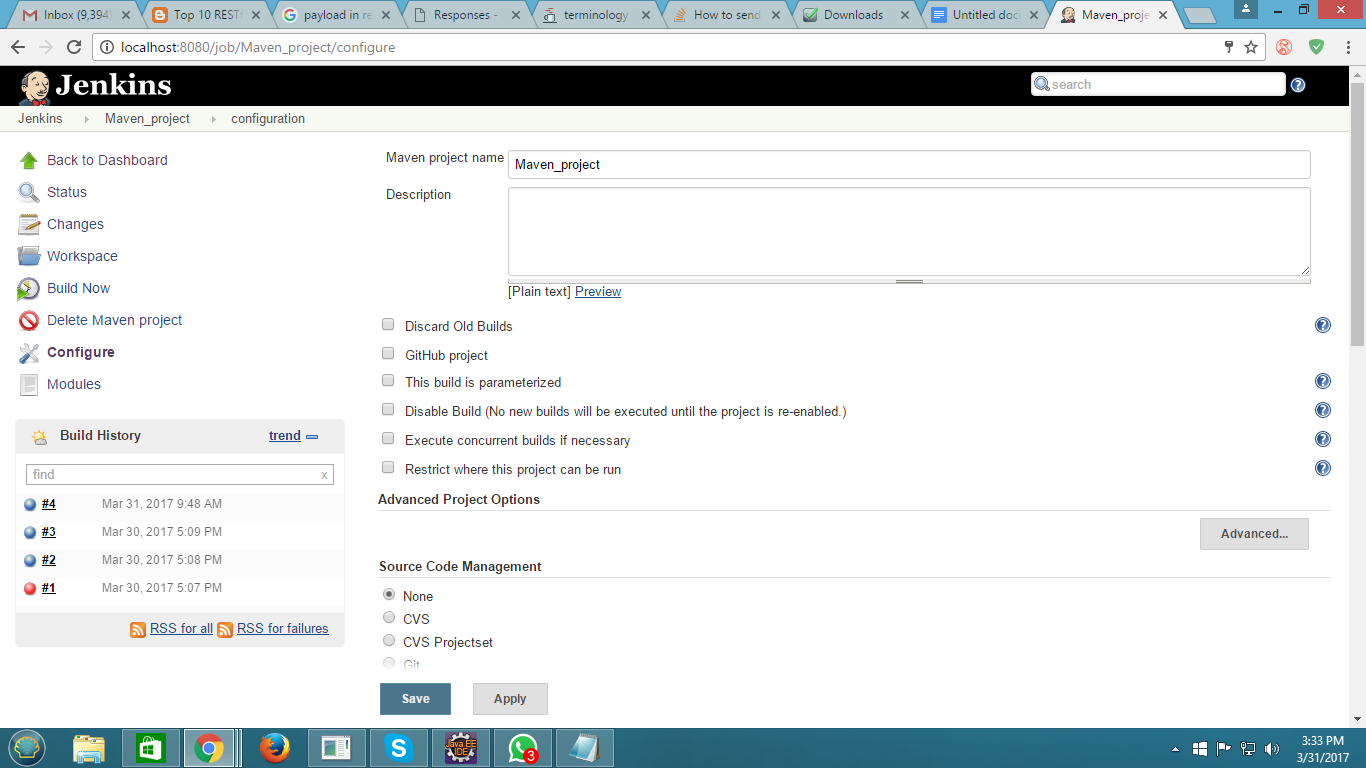
Slave machine -compile-maven

Create new job:

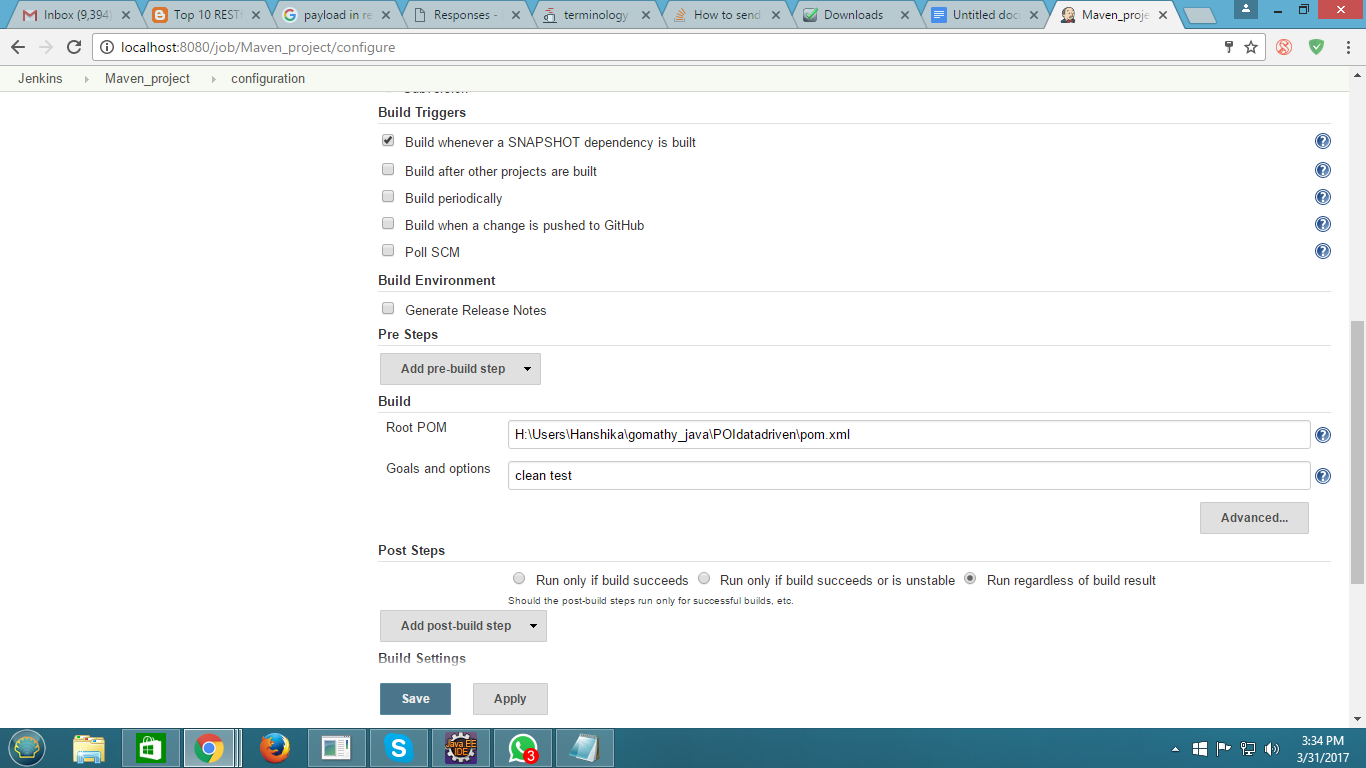
By clicking new item in jenkins dashboard



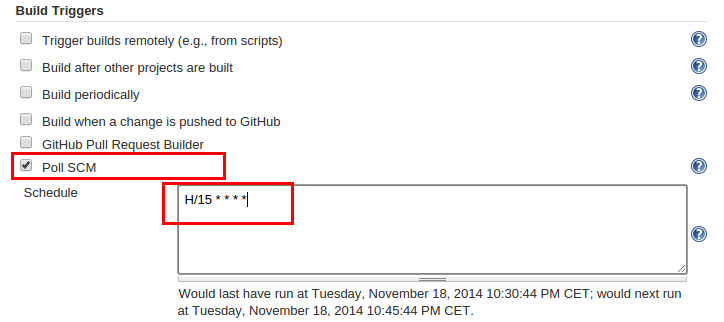
Then configure ur job below

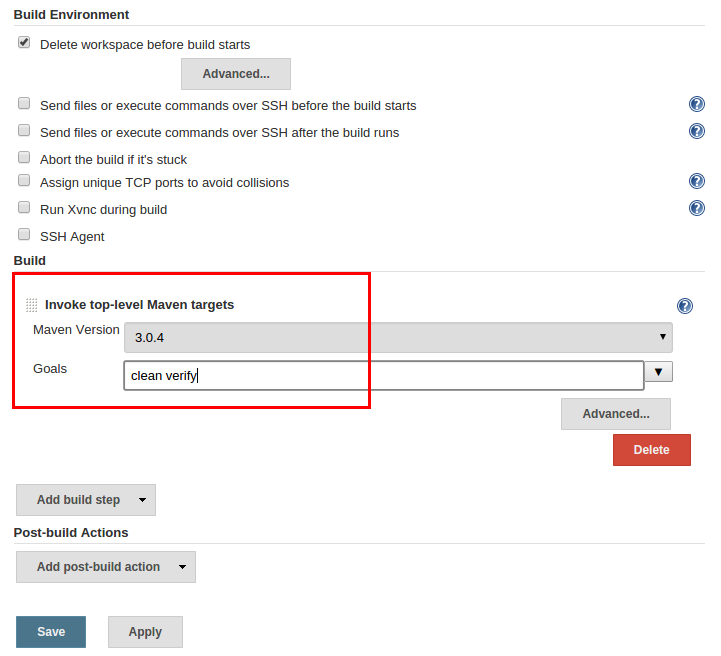


Build the job (give pom.xml file path) below like that

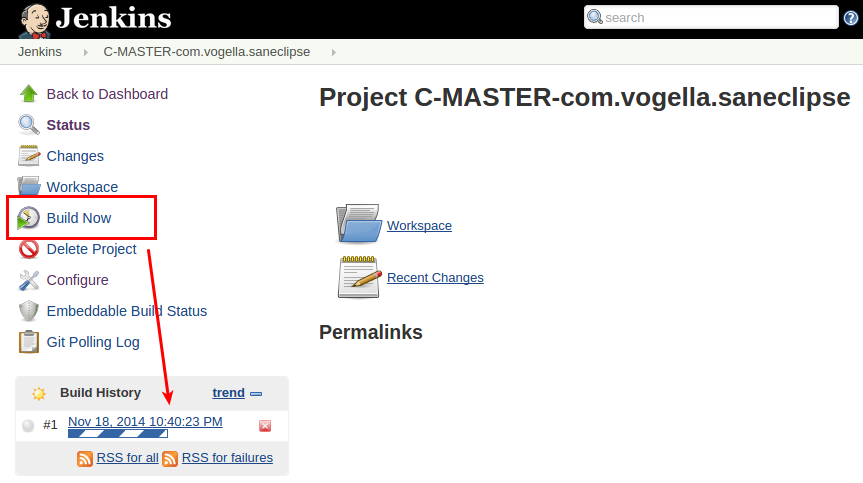


Schedule the job below:

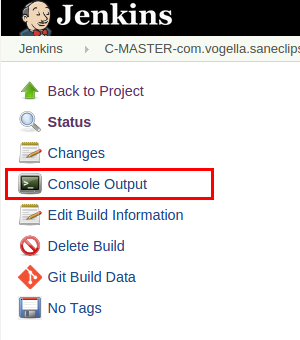




Now here you go to build the job:



See the output



Master-slave in jenkins

Goto------->manage jenkins------>manage nodes click it

Then able to see new nodes on left side in your dashboard

Then click new nods----> create a new node >

Master will appear on dashboard (initially it will be in offline)

Double click---> master node--->then launch agent---->you will see download file --->click keep button ----> java windows dialog box will appear then click ---->run

Then master will connected (its will appear small window says it connected )