Download Best Udemy Development Course

Download at http://devcourses.co

What is Good Course Look Like ?

- Start from Beginner
- Step by Step Explained Method
- Alot of Example to Make Thing Clear
- Alot of Exercise, Quize
- Real Life Hand on Projects Solve Real Problems
- Boot up to Launch New Carrier

Download Full Version of This Course

https://www.udemy.com/jkcourse/?couponCode=DOWNLOAD

2.5 hours of full HD screencast cover all content of this book.

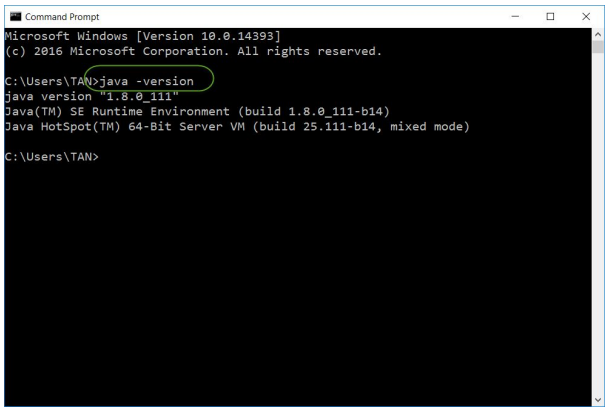
Jenkins Installation

In this chapter, we will jump into how to install and start up Jenkins in 3 environment : Window, Ubuntu and Mac.

In all environment, Jenkins will be installed as service, it mean Jenkins will automatically start up every time you machine power on or restart machine.

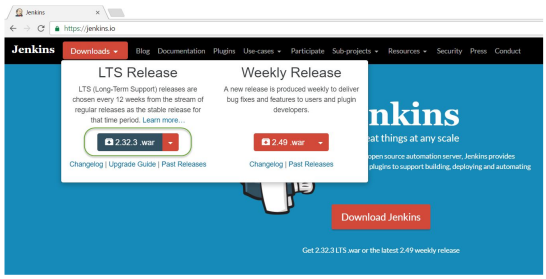
Jenkins on Windows

Check if java is installed : Java should be installed in Windows machine, to check if Java is installed, start a command prompt and typing in ***java -version*** Java version show up as below. Incase java not yet install, please install java before process to installation.

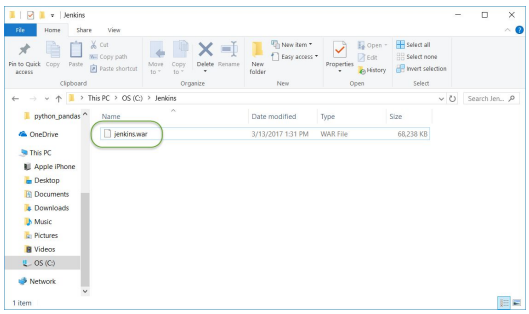


Install process

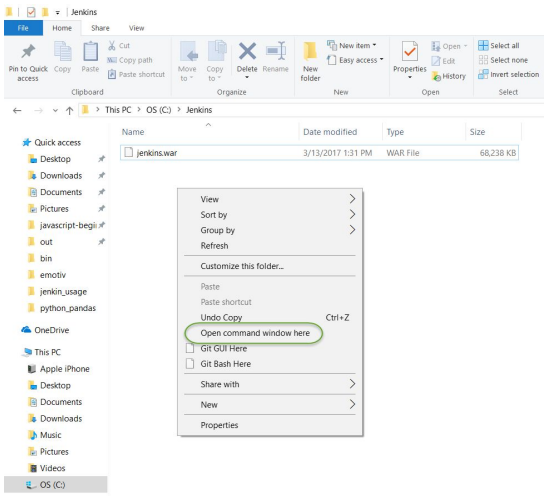
Step 1 : Go to ***https://jenkins.io/*** and download for stable version of Jenkins



Step 2 : Create a new folder for Jenkins (for example ***C:\Jenkins***) and put file ***jenkins.war*** inside this folder

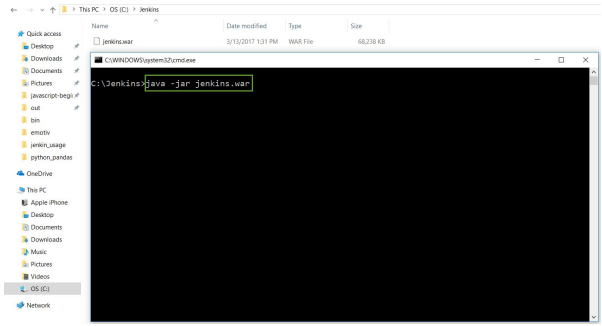


Step 3 : Press to shift and right click at same time to open context menu and select ***Open command window here***

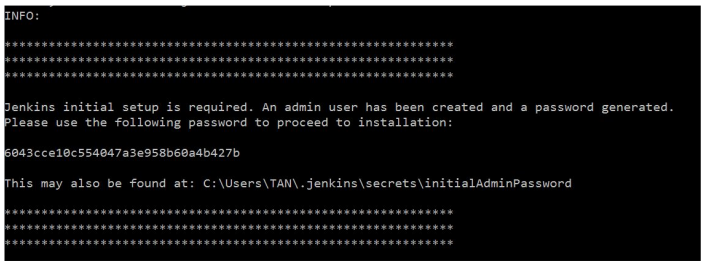


Step 4 : Typing in command prompt following to start installation ***java -jar jenkins.war***

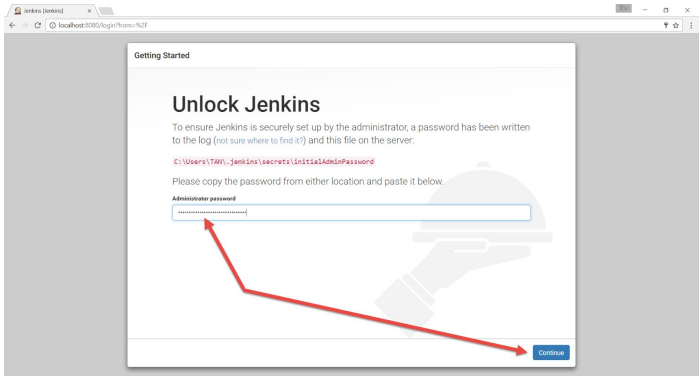
Then wait a while for complete running.



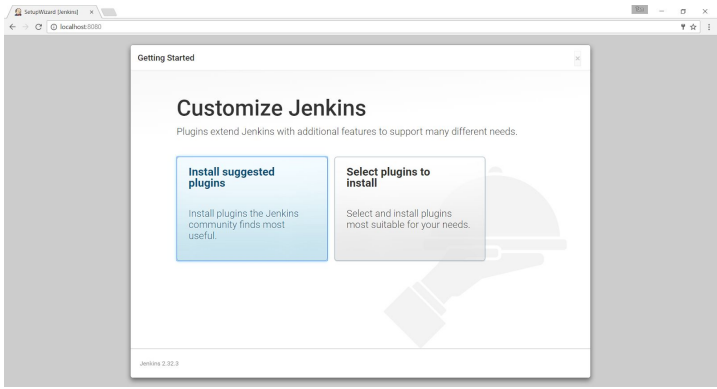
In the output log, you will see default password for first login of admin user, copy this password to notepad and save it for use on next step.



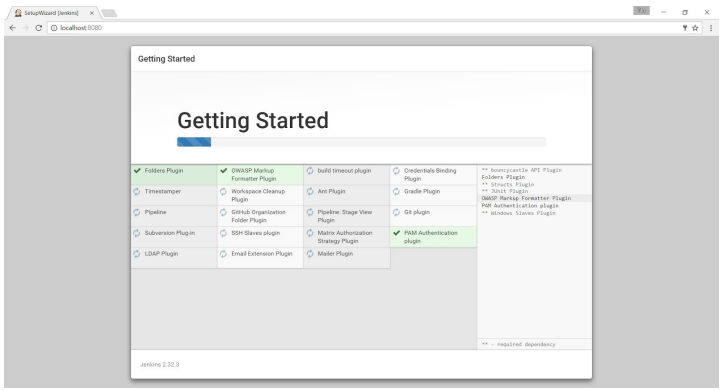
Step 5 : Now open browser and access to ***http://localhost:8080*** and then paste admin password to unlock Jenkins



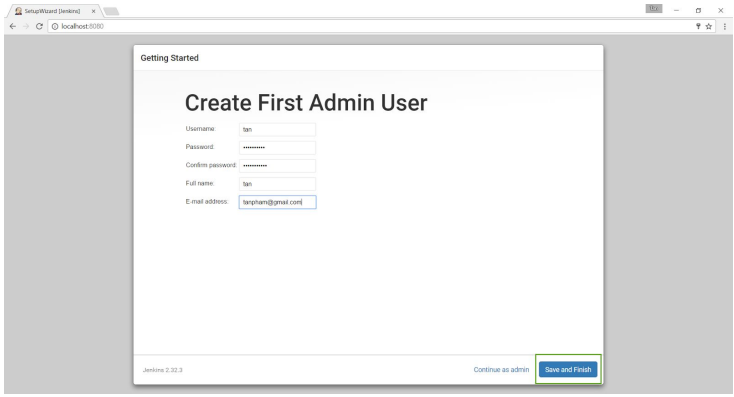
Wait a while and following screen show up, select default option ***Install suggested plugins***



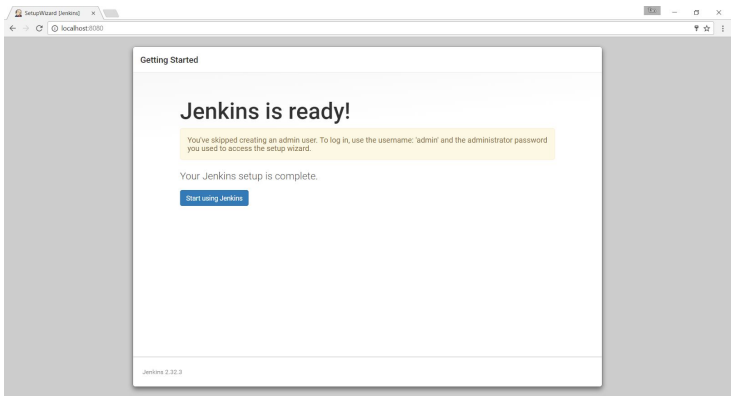
At this step default plugins will be installed.



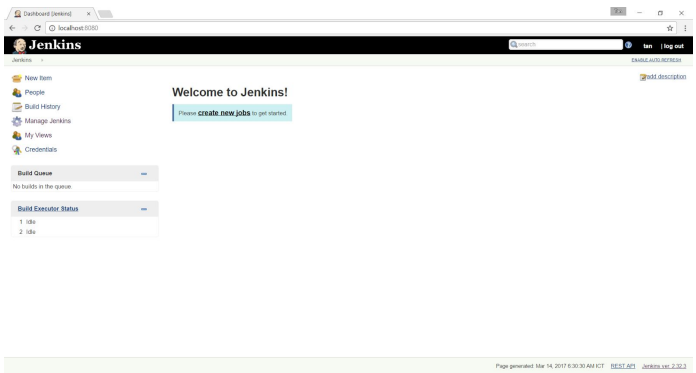
Step 6 : Enter your user name, password for a new admin, then click to ***Save and Finish***



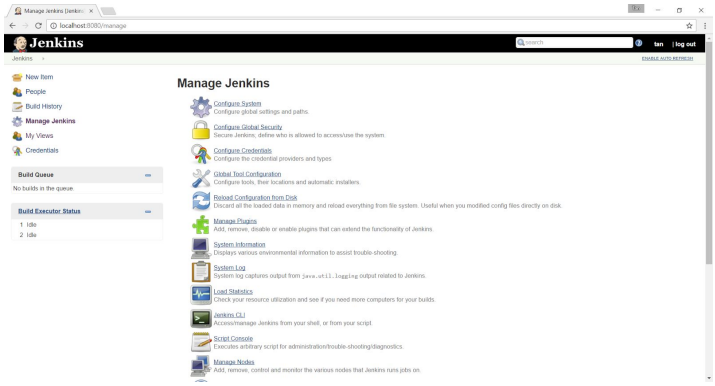
Step 7 : Click to ***Start using Jenkins***, Congregation at this step you already complete for Jenkins installation



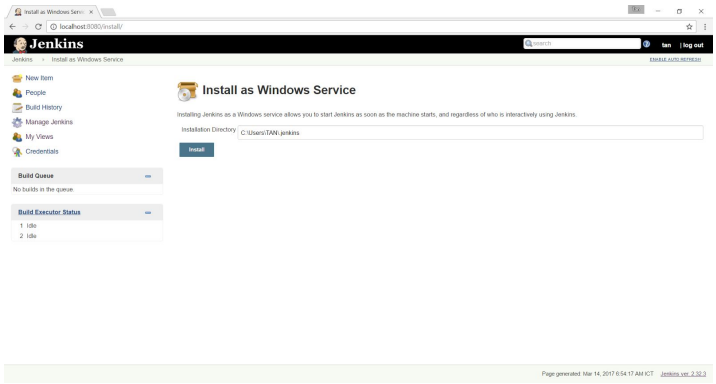
Jenkins auto login by new admin user already created



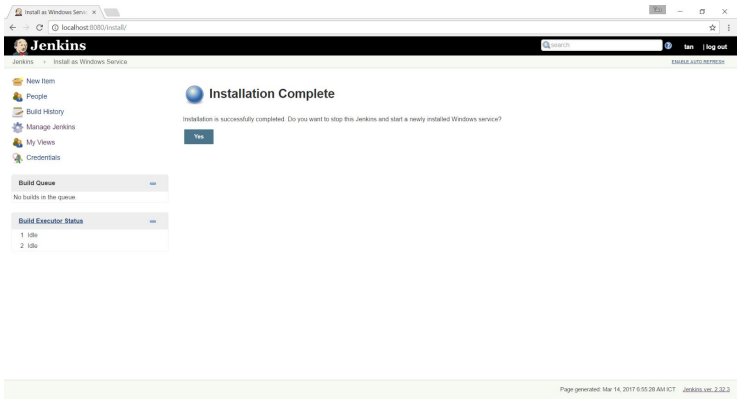
Step 8 : Click to ***Manage Jenkins*** from main screen, scroll down and click to ***Install as Windows Service***



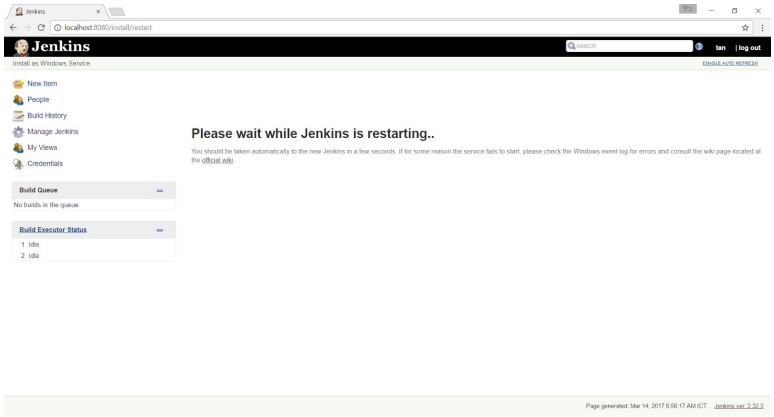
Click to ***Install***

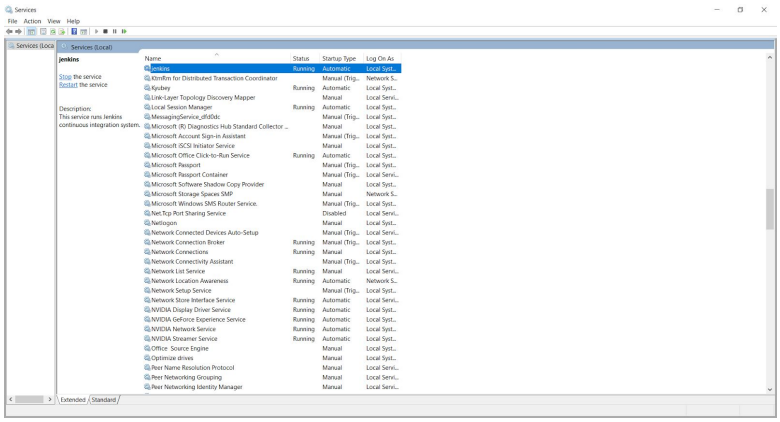


Click to ***Yes***



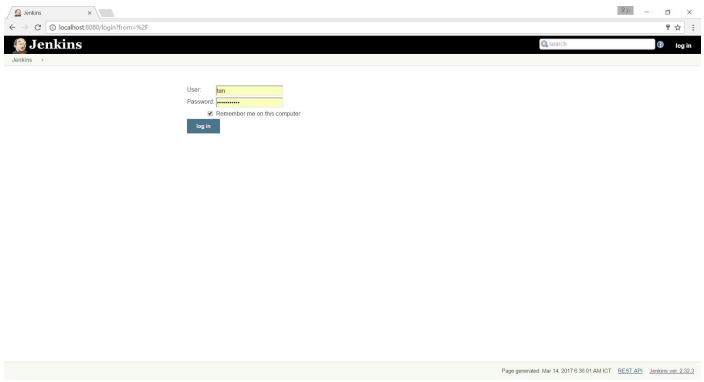
Wait some time and you will see jenkins service is running from windows service manager





That it, You complete install Jenkins on Windows as service. From now on every time Windows power up or restart, Jenkins will ready to access at ***http://localhost:8080***

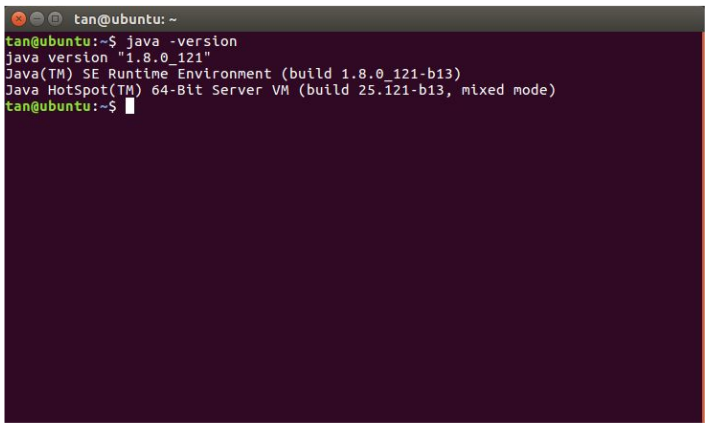
You can try to login Jenkins with above user created.



Jenkins on Ubuntu

Check if java is installed

Java should be installed in Ubuntu machine, to check if Java is installed, start a terminal and typing in ***java -version ,*** If Java already installed, you will see Java version show up.



Install Process

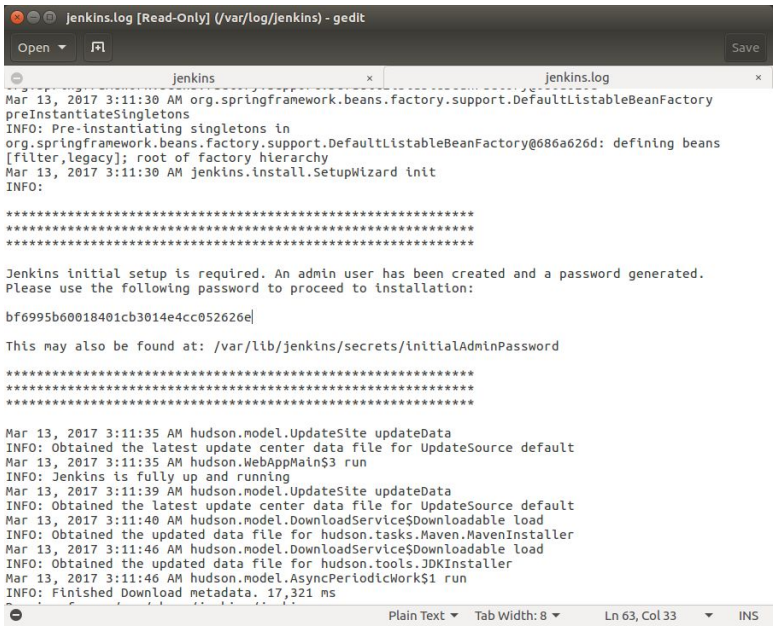
Step 1 : Open terminal and typing in following command

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo
apt-key add

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins
```

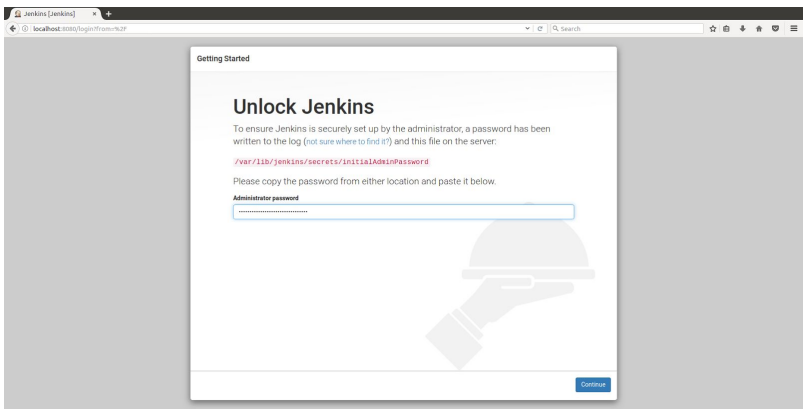
The above code do following job :

- Jenkins will be launched as a daemon up on start. See /etc/init.d/jenkins for more details.
- The 'jenkins' user is created to run this service.
- Log file will be placed in /var/log/jenkins/jenkins.log. Check this file if you are troubleshooting Jenkins.
- /etc/default/jenkins will capture configuration parameters for the launch like e.g JENKINS_HOME
- By default, Jenkins listen on port 8080. Access this port with your browser to start configuration.

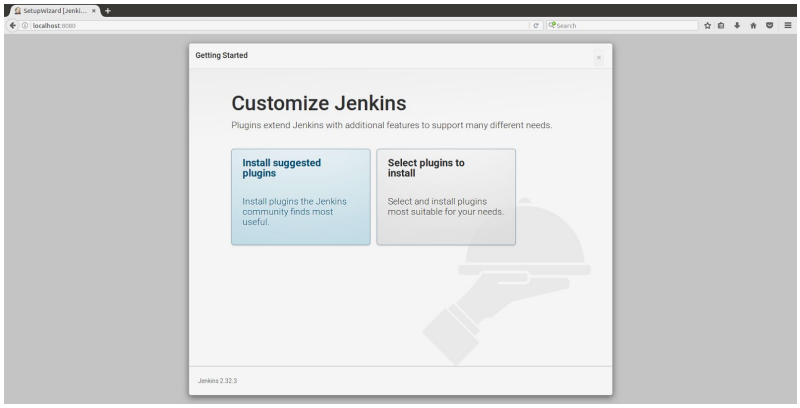
Step 2 : Open the file ***/var/log/jenkins/jenkins.log*** and copy the default password which is automate generated during installation

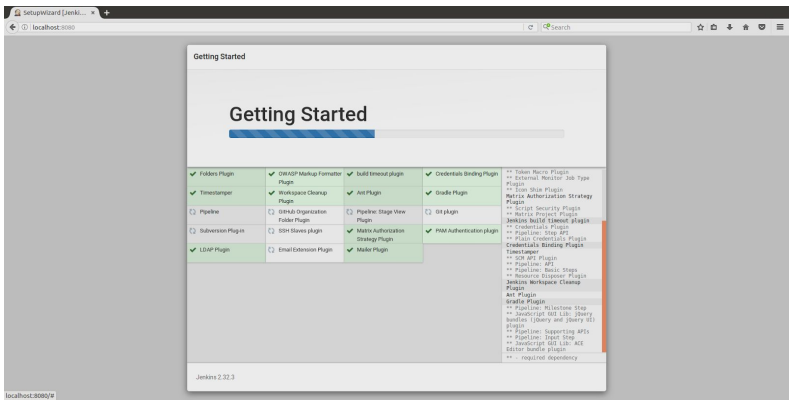


Step 3 : Open browser and access to Jenkins at http://localhost:8080/ then paste the password above to unlock jenkins. Click to ***Continue*** button

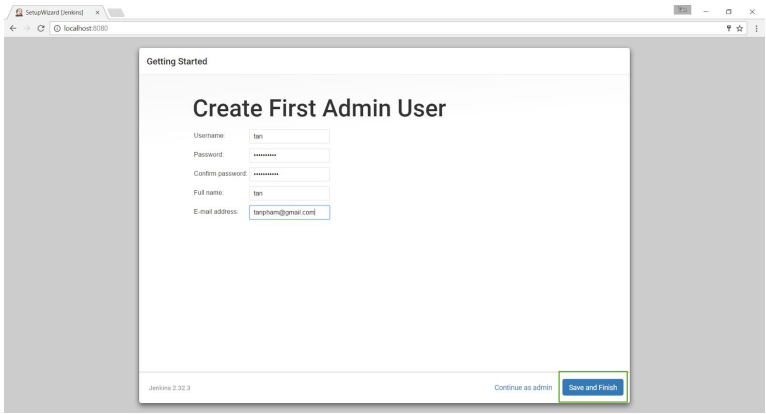


Step 4 : Click on ***Install suggested plugins***

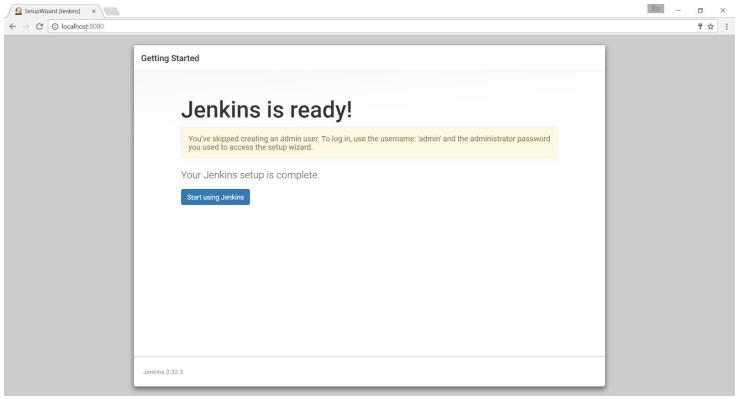




Step 5 : Enter your user name, password for a new admin.



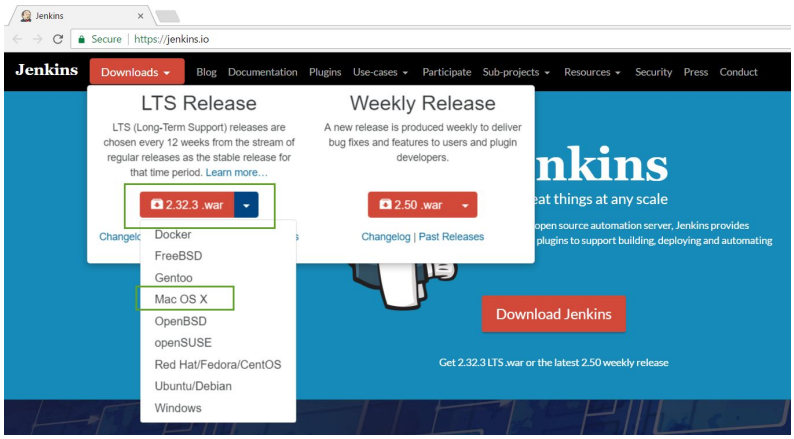
Step 6 : That it, now jenkins ready to use, click to ***Start using Jenkins***



From now Jenkins already installed in Ubuntu as daemon service, so every time you power on machine, Jenkins will start and ready to access at ***http://localhost:8080/***

Jenkins on Mac

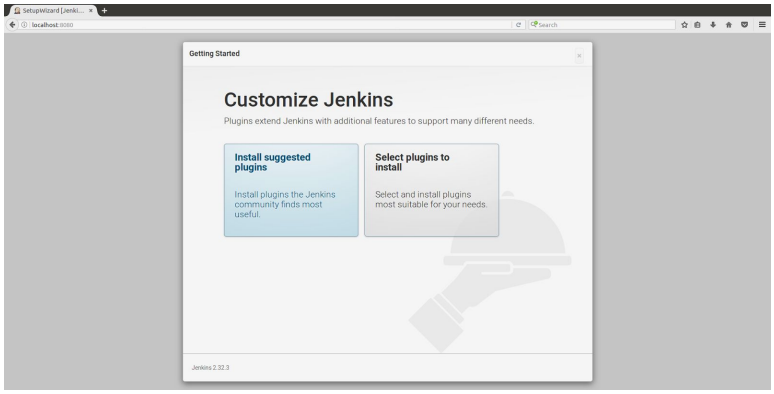
Step 1 : Download installer for Mac from https://jenkins.io/ .Then run the installer file.

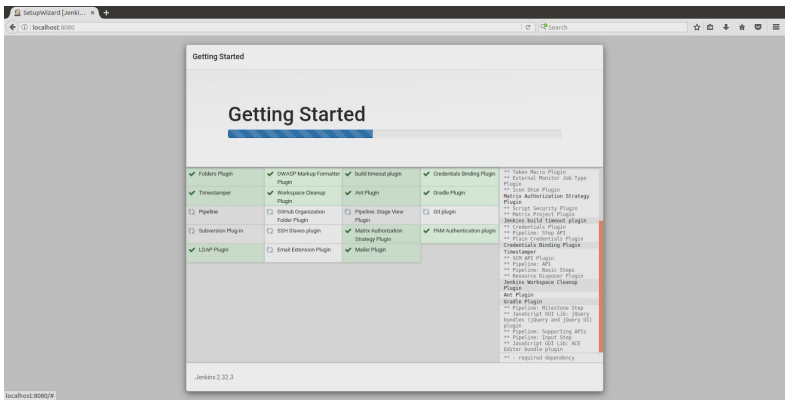


Step 2 : After complete install, you access Jenkins at ***http://localhost:8080/***

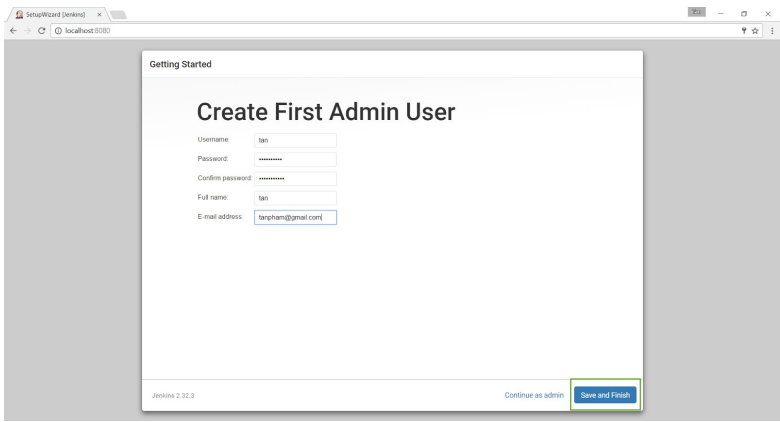
Step 3 : Open file ***/Users/Shared/Jenkins/Home/secrets/initialAdminPassword*** and copy password use for unlock Jenkins

Step 4 : Click on ***Install suggested plugins***

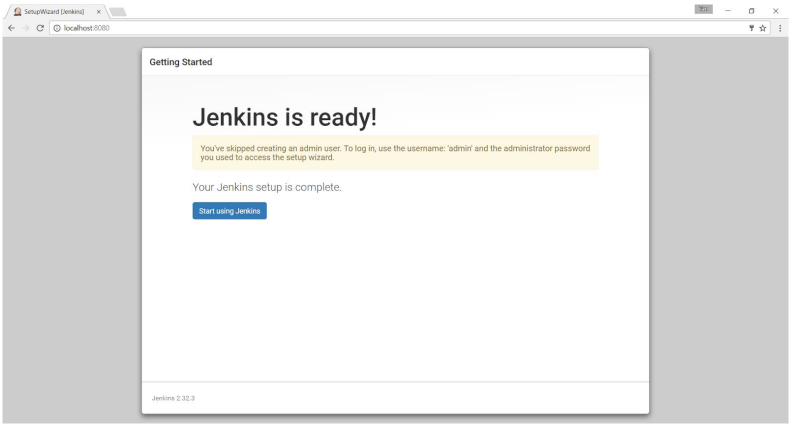




Step 5 : Enter your user name, password for a new admin.



Step 6 : That it, now jenkins ready to use, click to ***Start using Jenkins***



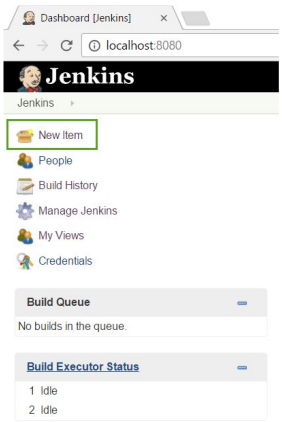
From now Jenkins already installed in Mac as a service, so every time you power on machine, Jenkins will start and ready to access at ***http://localhost:8080/***

Jenkins the Basic

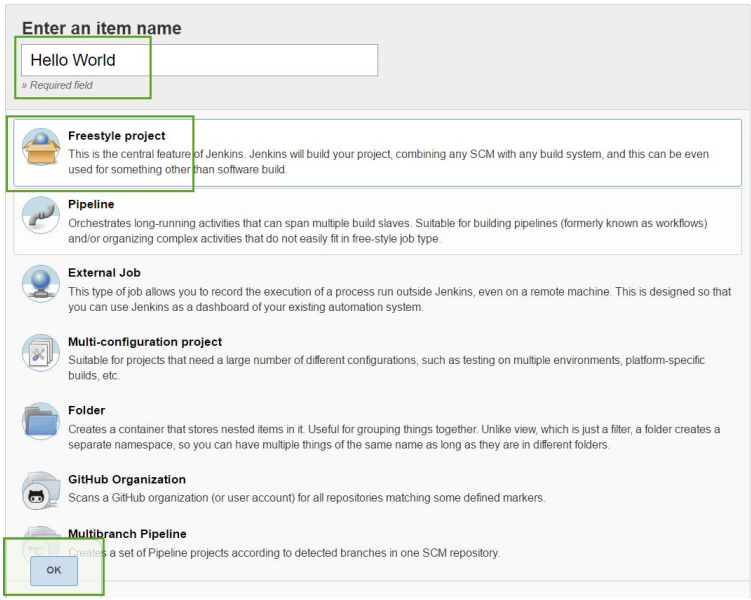
In this section I will explain Jenkins role in the world of continuous integration, continuous delivery and devops. And then we will go through steps in order to create “Hello World” Jenkins job.

Create “Hello world” Job

From Jenkins home page, click to **New Item**

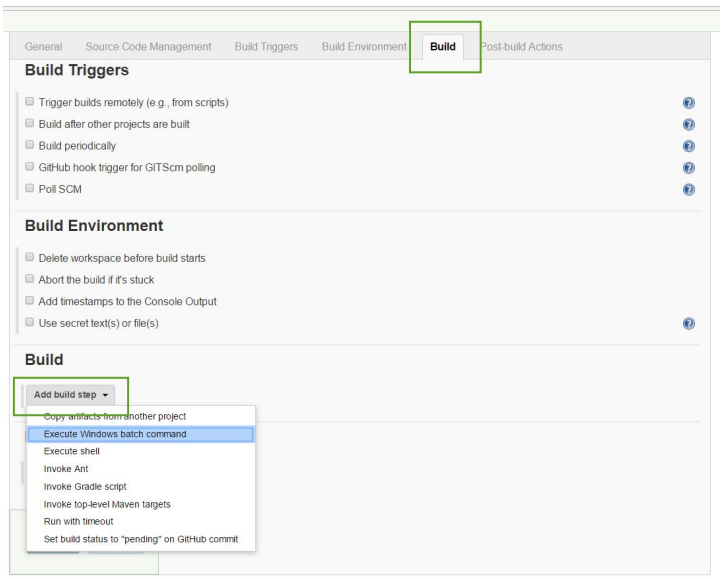


In create project page, typing in “Hello World” to project name, select **Freestyle project** and then click to save button.



Add Build Step

In configure job page, select **Build** tab, click to **Add build step** and select **Execute Windows batch command** (in case you are using Linux please select **Execute shell**)

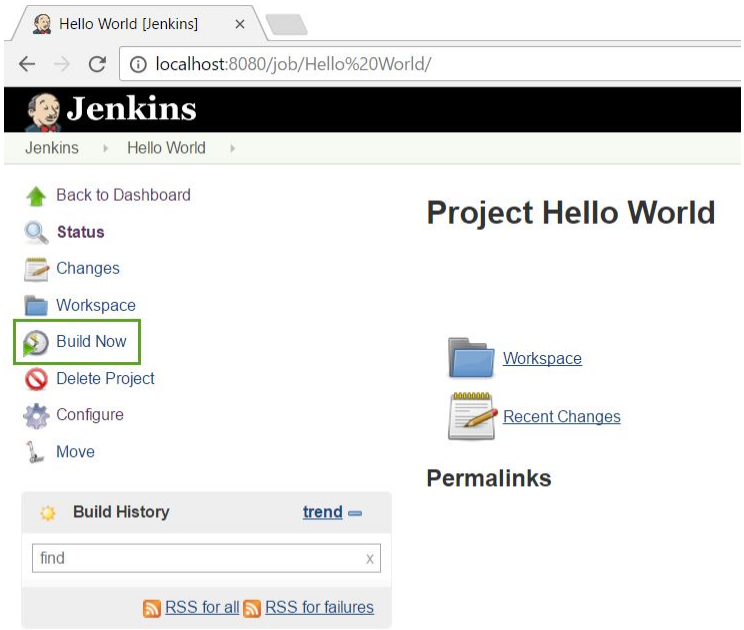


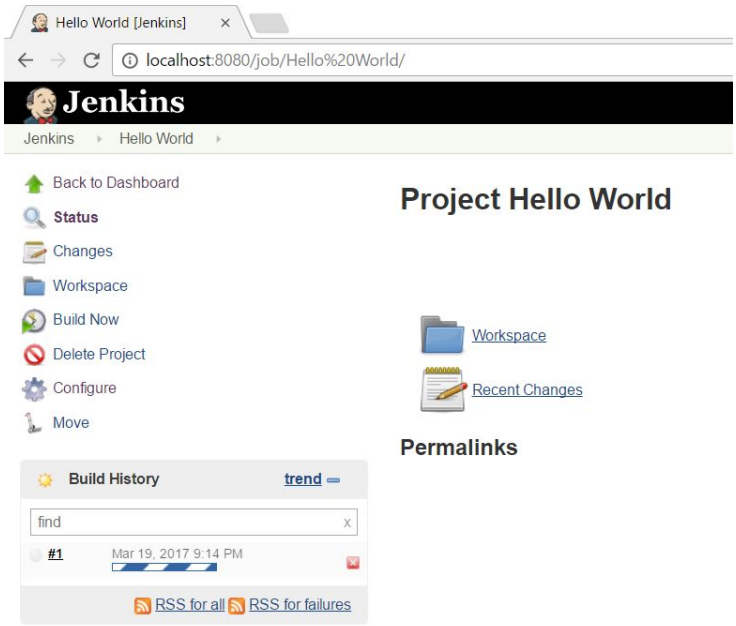
Typing in command “echo Hello World”. The purpose of this job is just echo to log message “Hello World” text.

Click **Save** button and we complete the first job!

Start the First Build

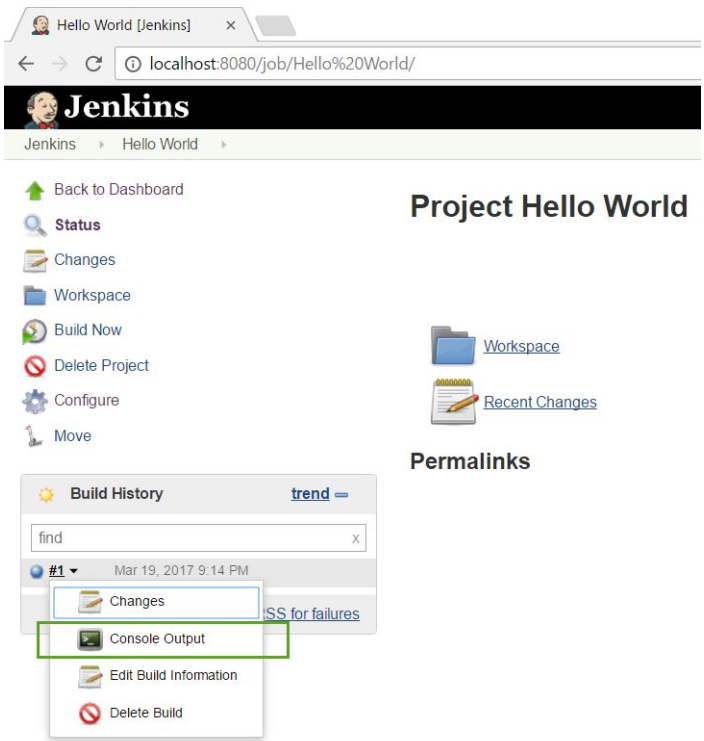
From job dashboard, click **Build Now** and you will see job run. This job just do one thing, invoke the command “echo hello world”





Console Output

Now job run completely, click to build number and select **Console Output.** You will see message “Hello World” show up inside build log. That it you just complete the first Jenkins job.

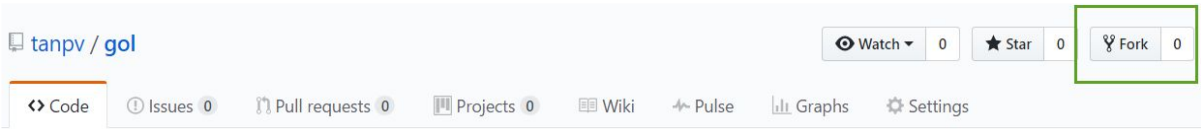


Intro GOL Project (or World without Jenkins)

In this book, project called GOL (game of life) is used to express the features related to Jenkins. So it will be helpful to review some information related to this project.

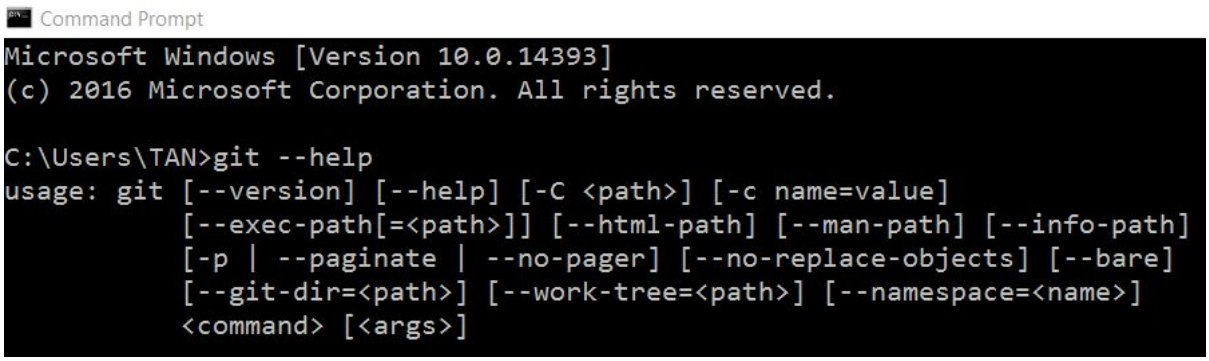
Git Installation and Clone GOL Repository

GOL project is hosted on Github at https://github.com/tanpv/gol. Please **fork** this repo to your Github account so you can try Jenkins features by yourself later.

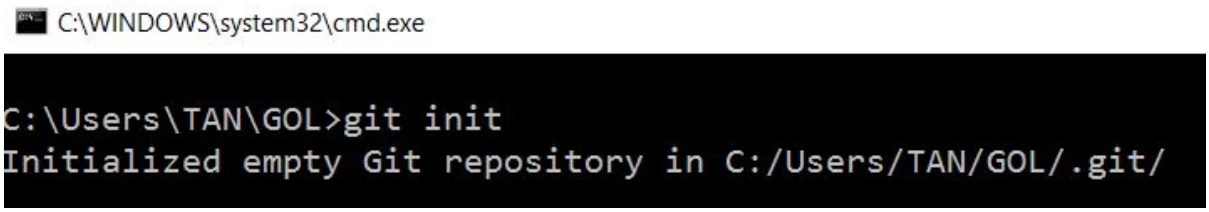


To install Git, download software at https://git-scm.com/ , and install with default option.

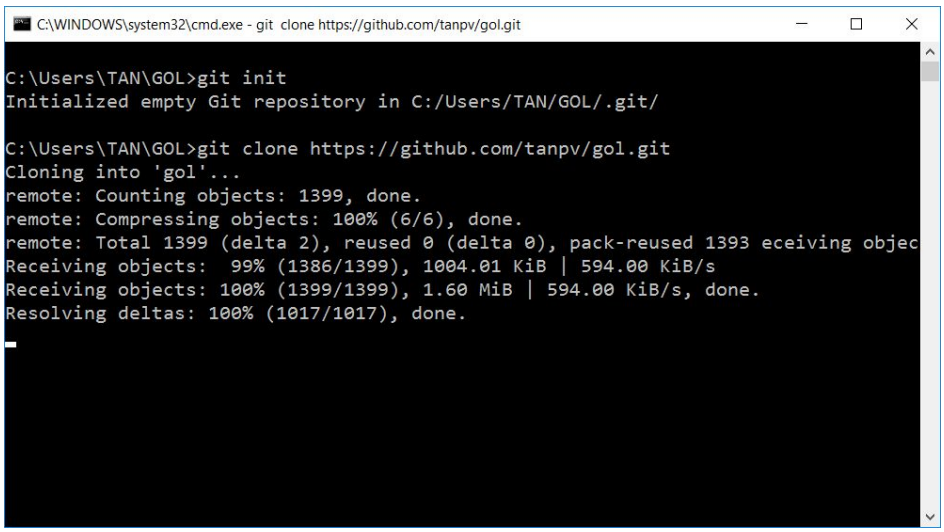
After install, open a cmd prompt and typing in **git --help** to check if install successfully



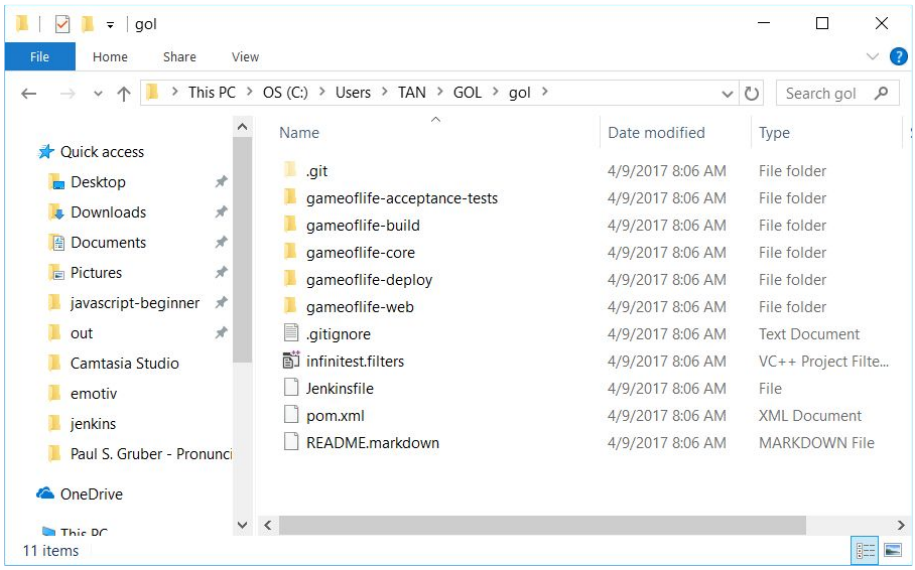
Now create a folder name GOL and init it with command **git init**



And finally, clone the repo from Github to local with command **git clone https://github.com/tanpv/gol.git**

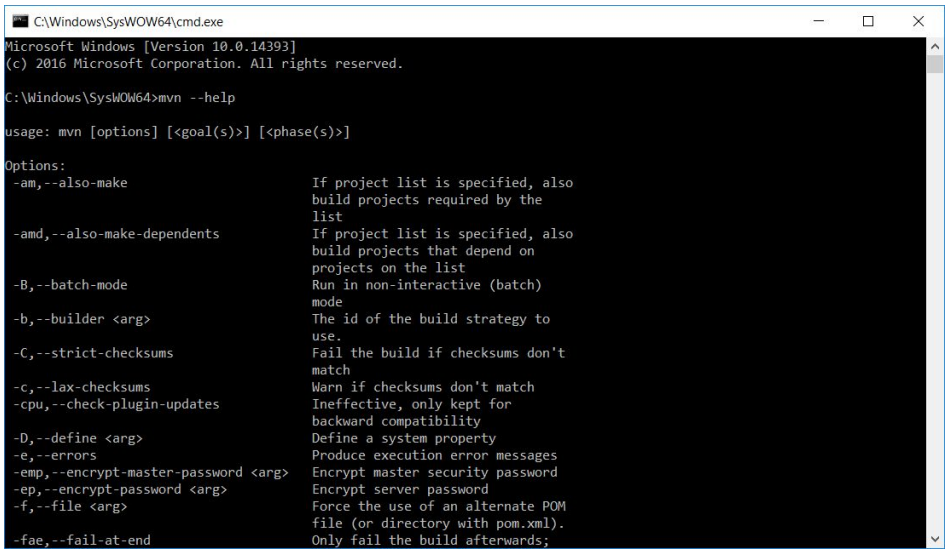


And finally, you will see GOL project is completely clone to local



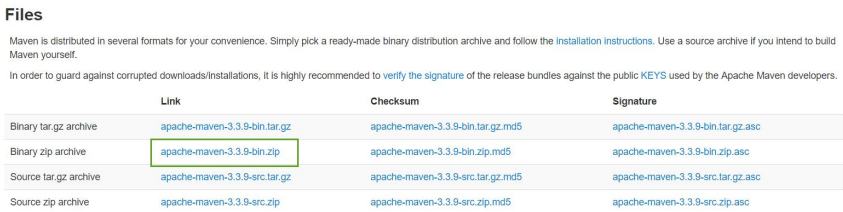
Maven Installation

First step, maven should be installed in machine which running Jenkins job. To check this, just open command prompt and typing in ***mvn --help ,*** you will some all available maven command.



In case maven not yet install follow these steps to do maven installation (for Window)

- Download maven from link http://maven.apache.org/download.cgi



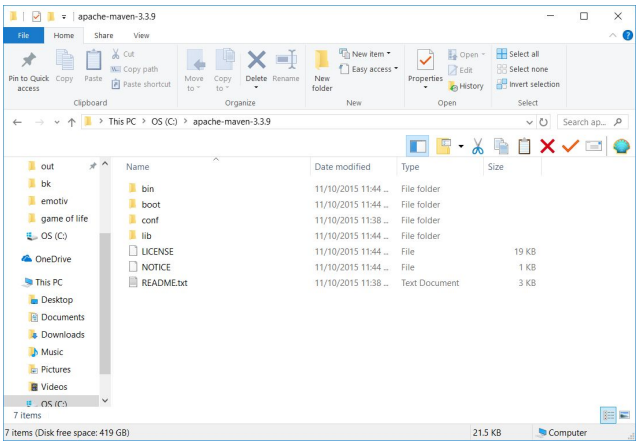
Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

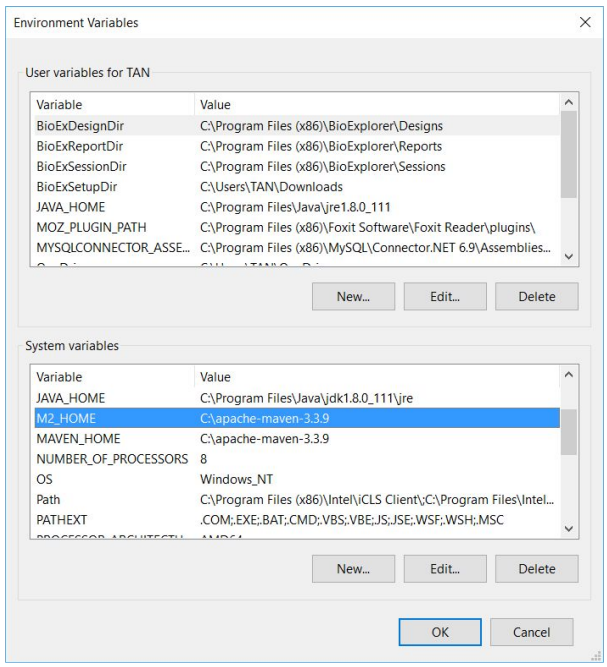
In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

	Link	Checksum	Signature
Binary tar.gz archive	apache-maven-3.3.9-bin.tar.gz	apache-maven-3.3.9-bin.tar.gz.md5	apache-maven-3.3.9-bin.tar.gz.asc
Binary zip archive	apache-maven-3.3.9-bin.zip	apache-maven-3.3.9-bin.zip.md5	apache-maven-3.3.9-bin.zip.asc
Source tar.gz archive	apache-maven-3.3.9-src.tar.gz	apache-maven-3.3.9-src.tar.gz.md5	apache-maven-3.3.9-src.tar.gz.asc
Source zip archive	apache-maven-3.3.9-src.zip	apache-maven-3.3.9-src.zip.md5	apache-maven-3.3.9-src.zip.asc

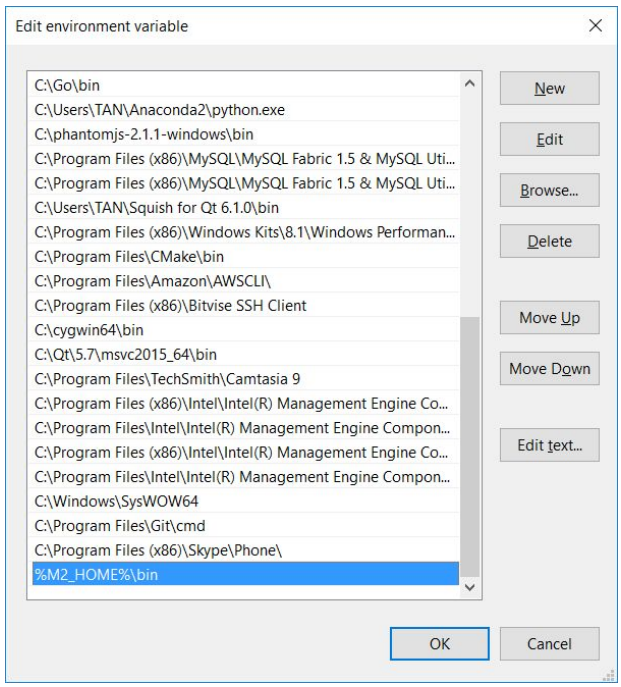
- Unzip zip file to one folder (for example I unzip to C:\apache-maven-3.3.9)



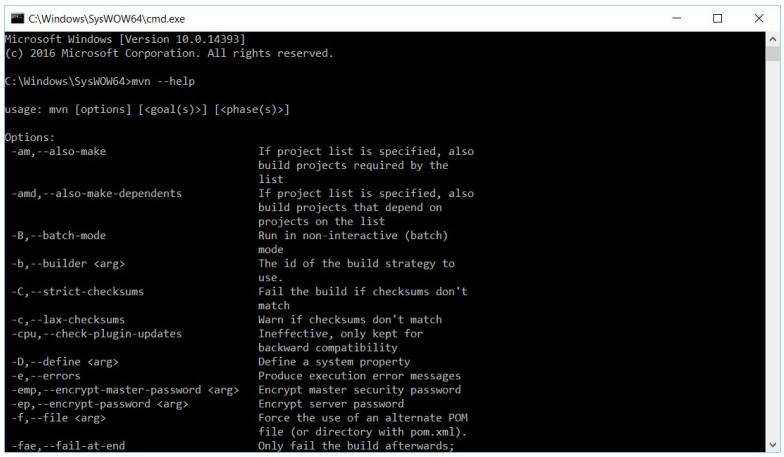
- Add M2_HOME and MAVEN_HOME variable point to above foder



- Add mavin bin to Path variable so you can call maven from any place



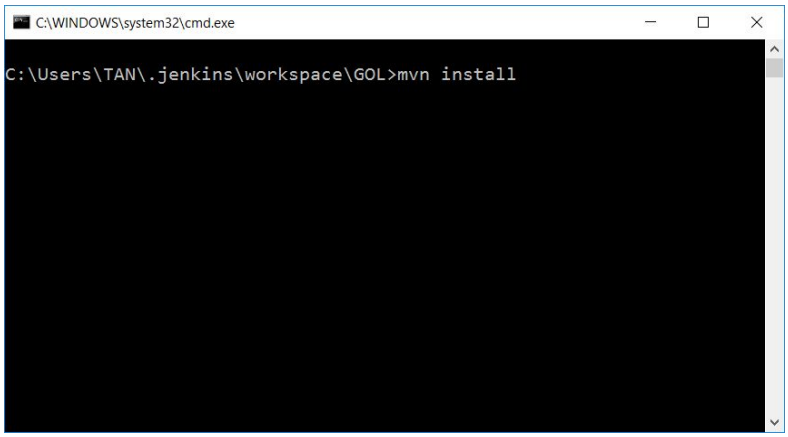
That it, to check maven installed successfully, open cmd and typing in ***mvn --help***
Help content for mvn should show up correctly



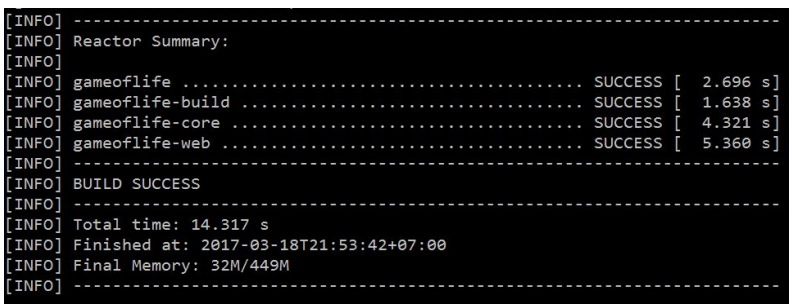
Run Maven Build Manually

Now maven is ready, we can try to run the build for source code which we just clone from Github repository at **source code management** step.

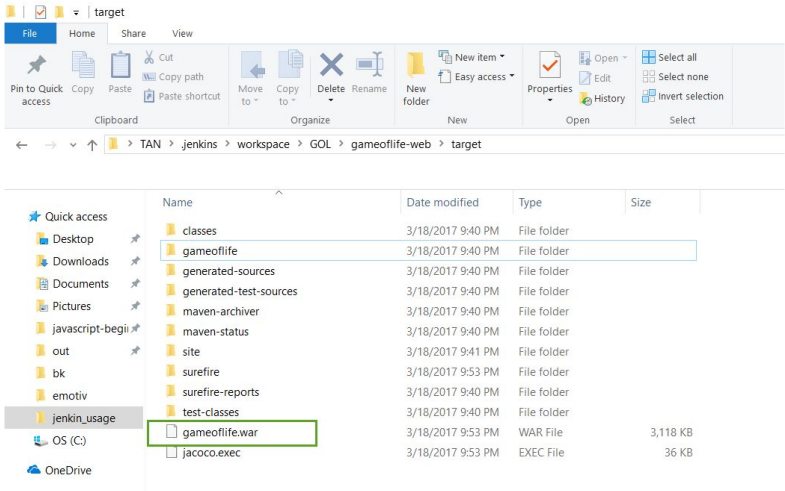
Open **cmd** and change directory to repository GOL folder. And then typing in **mvn install** This command will run java source unit test and then build this web application to a .war file



You will build result from command prompt



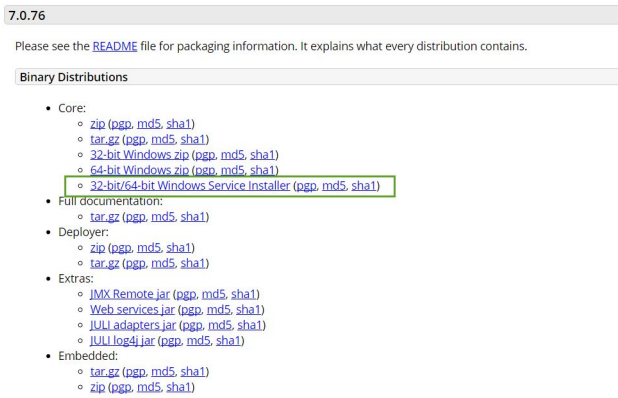
And can see actual .war file inside target folder of **gameoflife-web.** This folder could be use to deploy with web application server.



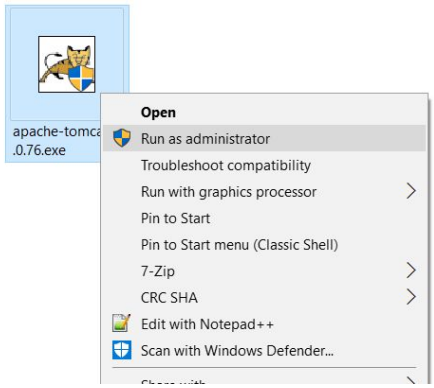
Install Tomcat Server as Service

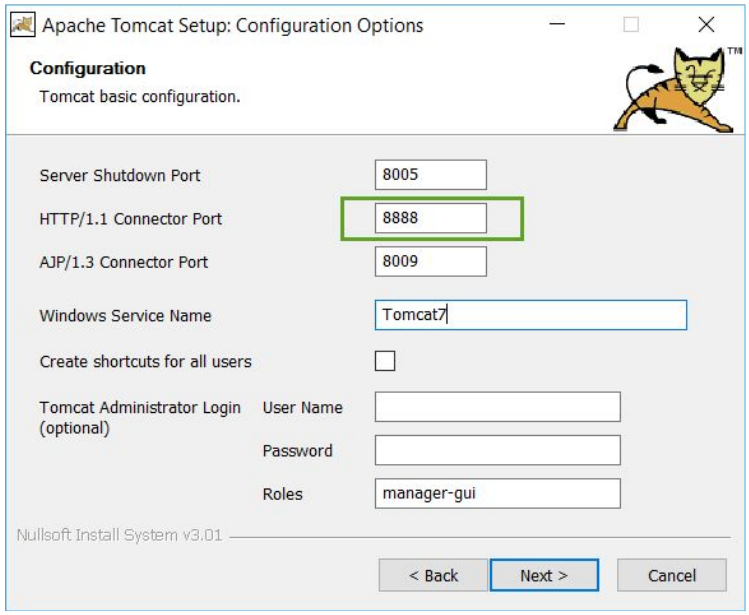
To install Tomcat 7 server, you just following these steps:

- Download Tomcat 7 from this page http://tomcat.apache.org/download-70.cgi , note that we will download version Windows Service Installer so Tomcat will be installed as window service

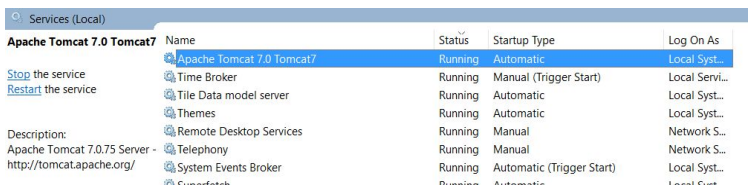


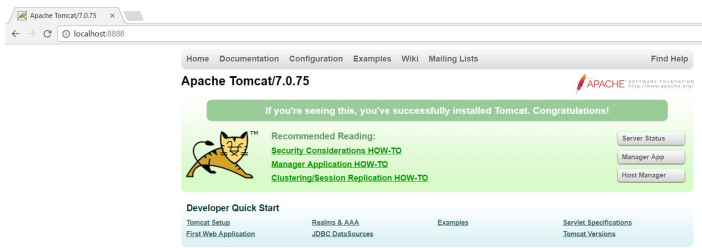
- Run installer and select port 8888 for running Tomcat (We already use 8080 for Jenkins installation)



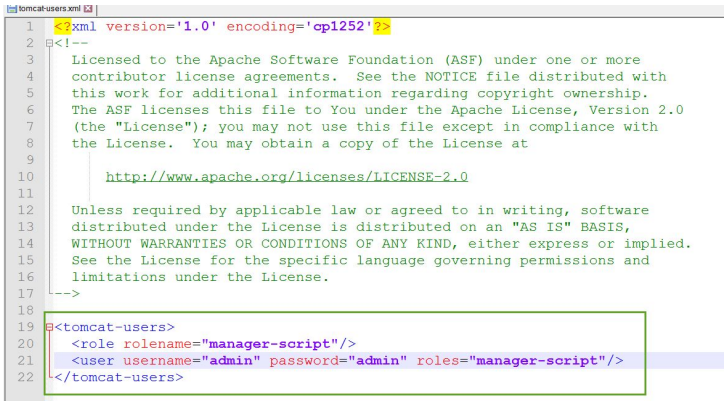


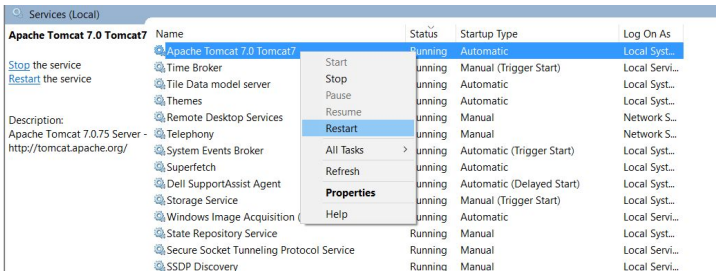
- After install successfully, a Tomcat service will running and you can access Tomcat server from localhost



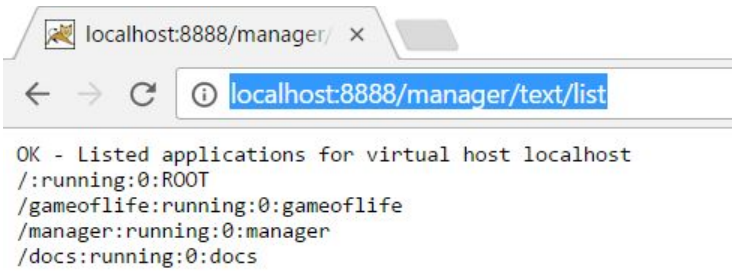


- To control Tomcat from Jenkins, one user with role “manager-script” should be added to file **conf\tomcat-users.xml .** Then restart Tomcat server by service.



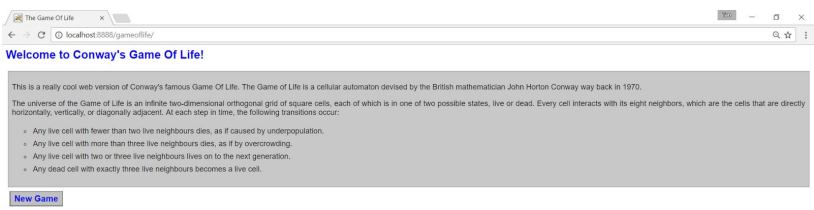


To check if admin user working, access to http://localhost:8888/manager/text/list and then put on user / password you just specify



Deploy GOL to Tomcat Server

Deploy this application to Tomcat server, and you will see some the app from browser and actually start a new game !.



Continuous Integration with Jenkins

This chapter will show you how to automate software build with Jenkins.

Jenkins in the Big Picture of CI, CD and DevOps

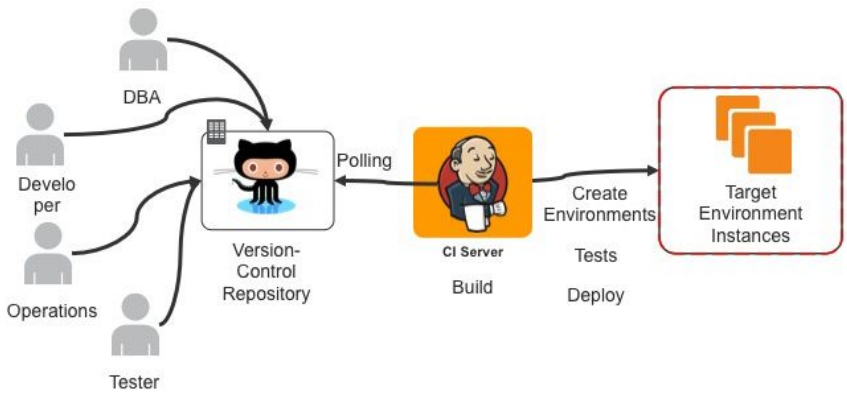
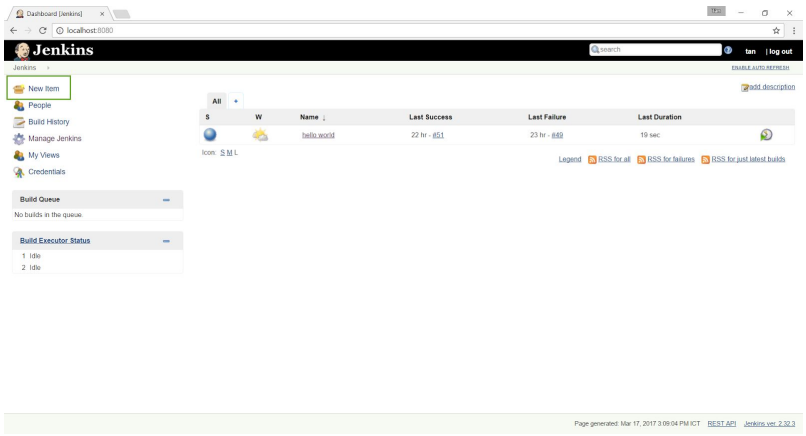


Image above show quite clear about Jenkins role in software development process.

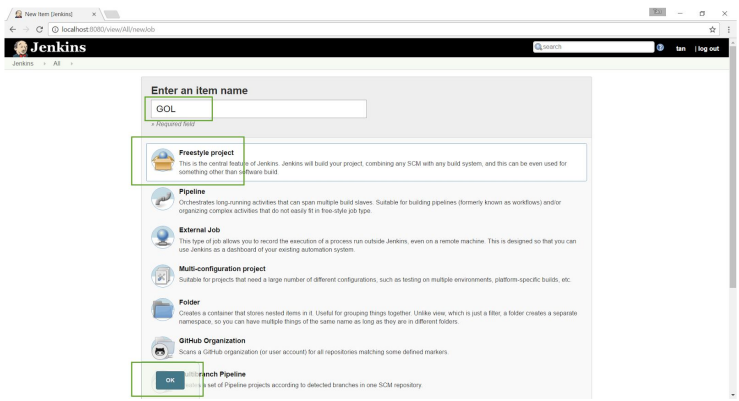
- Developer do coding and commit their code to a version control repository like Github
- Jenkins actively check Github to see if have any change, in case has code change, Jenkins will pull the last code from Github to build machine (machine which run Jenkins job)
- Jenkins build software artifact from pulled source code (for example war file from .java source file)
- Jenkins run and show up unit test report
- Jenkins deploy artifact to target environment (for example a web server like : Tomcat, JBoss, WebSphere...)

Create GOL Job

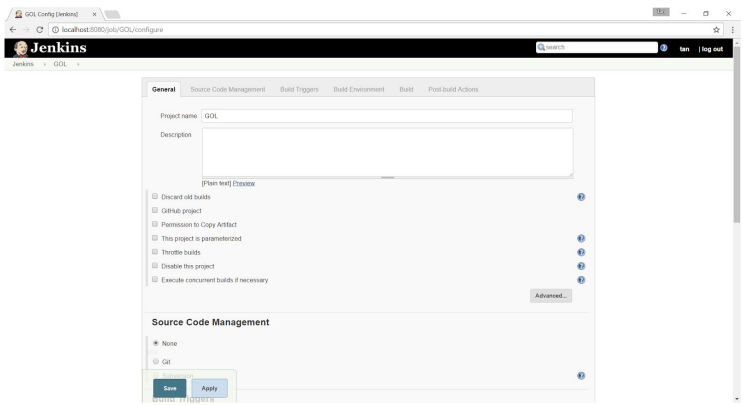
From home page, click to ***New Item*** link



Enter ***GOL*** for project name and then chose ***Freestyle project*** and finally click to ***OK*** button to create a new job with name is ***GOL***



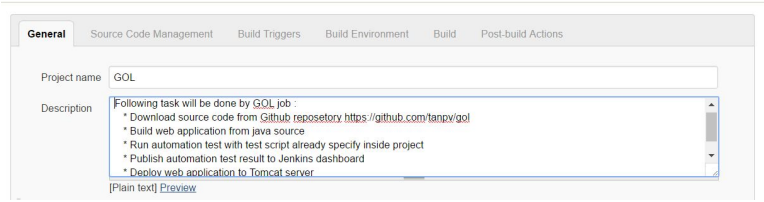
Then a new screen will show up allow us to configure everything for **GOL** job.



Description is place to describe what mission of this job and steps inside the job. Put follow text into ***Description*** session.

“ Following task will be done by GOL job :

- Check change on Github repo and automatically download source code from Github repository https://github.com/tanpv/gol
- Build web application from java source
- Run automation test with test script already specify inside project
- Publish automation test result to Jenkins dashboard
- Deploy web application to Tomcat server ”

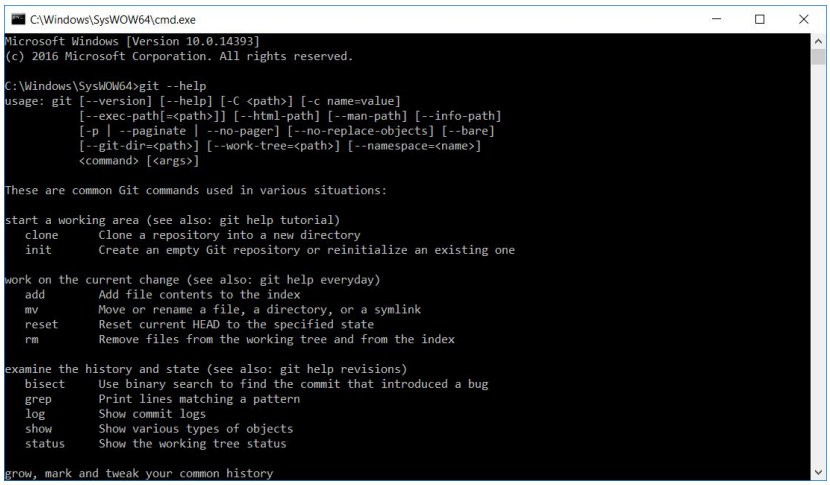


Source Code Management

Purpose of this part is answer for question ***Where to get source code ?*** for our software build. From job configure, click into ***Source Code Management*** tab

Check if git is installed in local machine

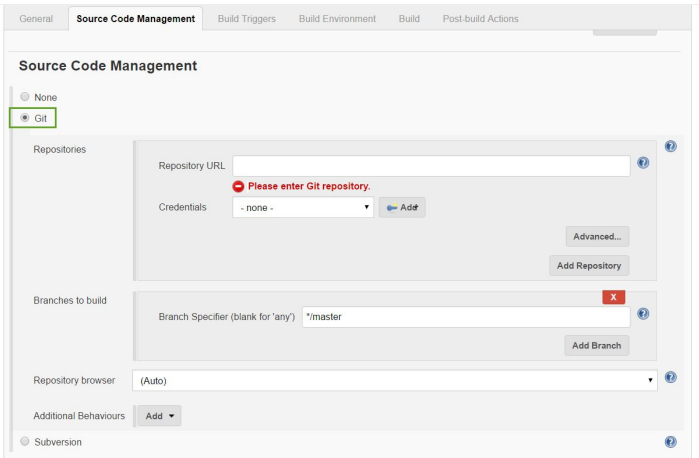
In order to clone source code from Github, local machine should install Git.
To check if local machine already install **git** or not, open command prompt and typing **git --help**



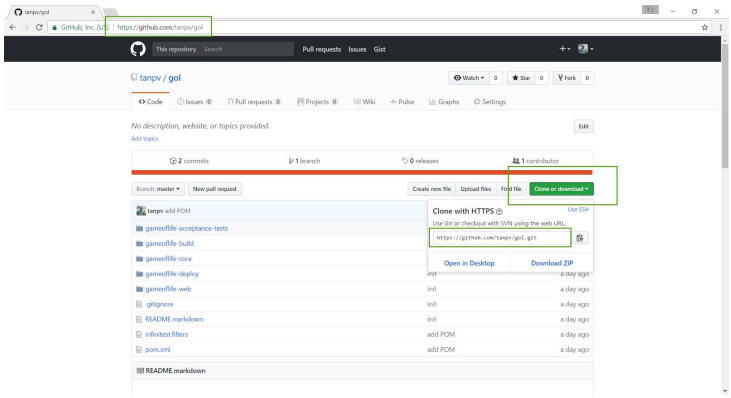
To download and install git, please refer to this link https://git-scm.com/downloads

Specify source code repository

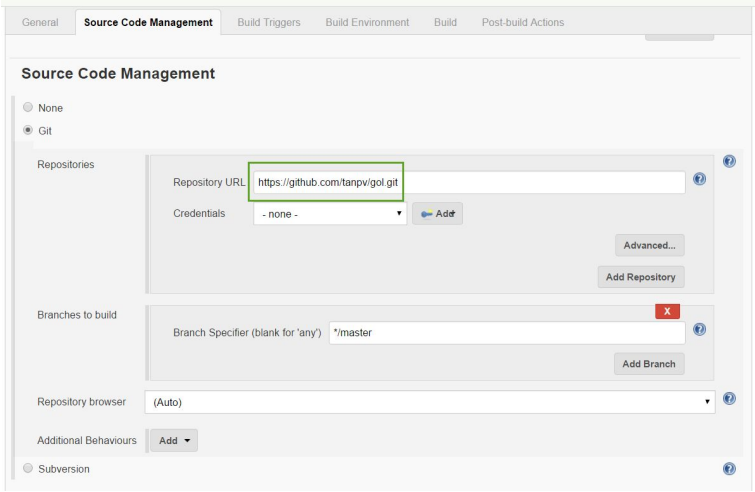
Because we host project source code on Github, so you will choose ***Git*** as below.



To get repo link, go to Github https://github.com/tanpv/gol then click to ***Clone or download*** button, you will see the repo link. Copy this link.



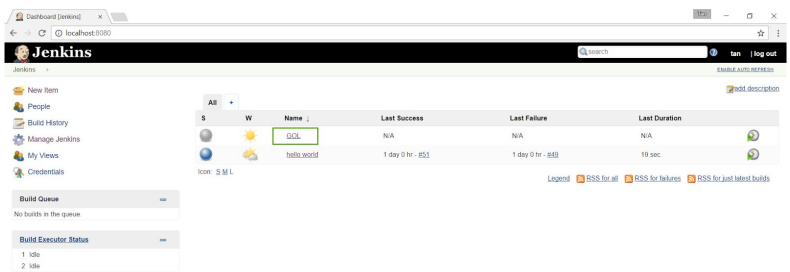
Put the repo link ***https://github.com/tanpv/gol.git*** in to ***Repository URL*** section. The branch will be default as ****/master***



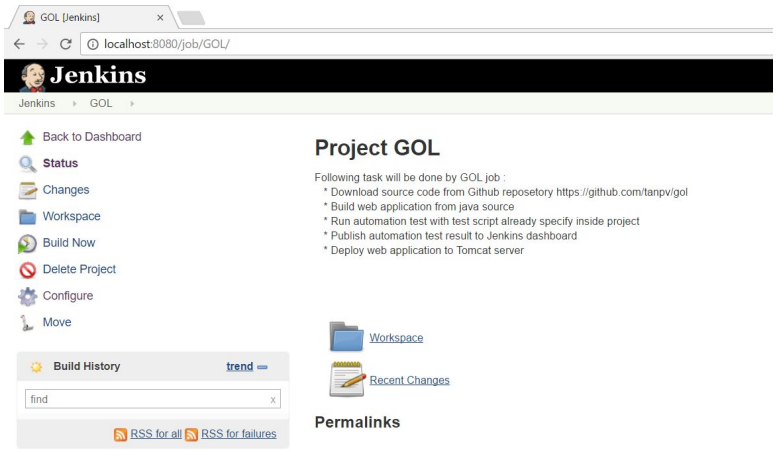
Click to ***Save*** button and that it, we already tell to Jenkins where to get source code.

Check if setting working fine

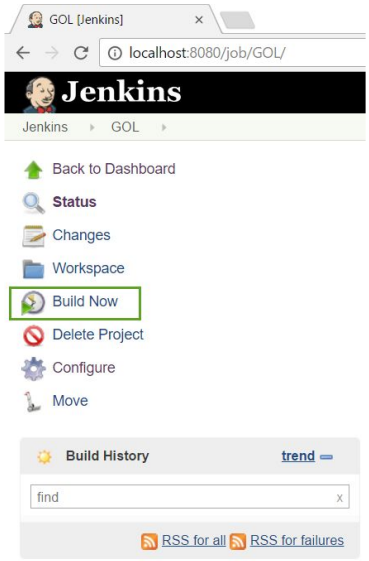
This is the time to try for first running and see if source code from Github is clone or not. Go back to home page then click to GOL job



The dashboard for GOL will show up, you will see describe which we just added before

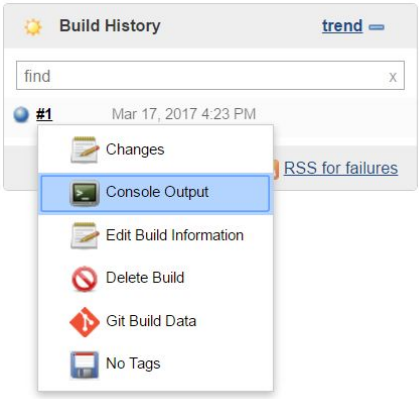


Click in to ***Build Now*** to activate job run immediately.

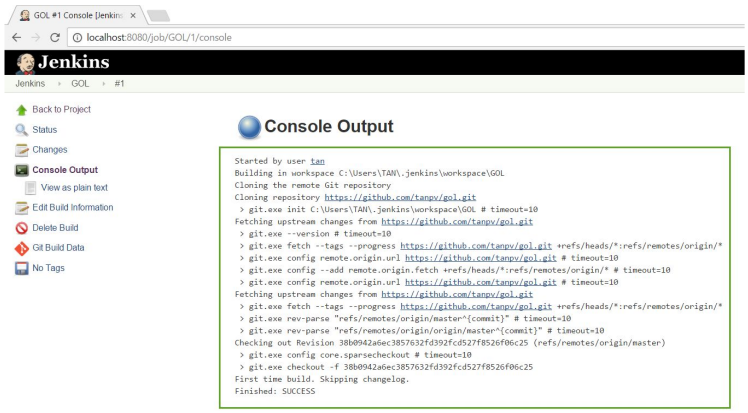


You will see GOL start running as expected

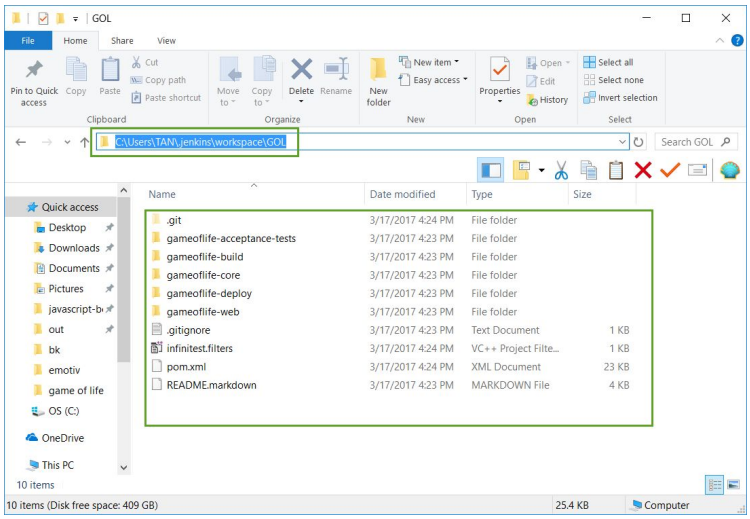
To see how job running, select ***Console Output*** from context menu while you click to build number



Then the job log will show up every thing happen in order to clone source code from Github repository. You can see that the build already success running.

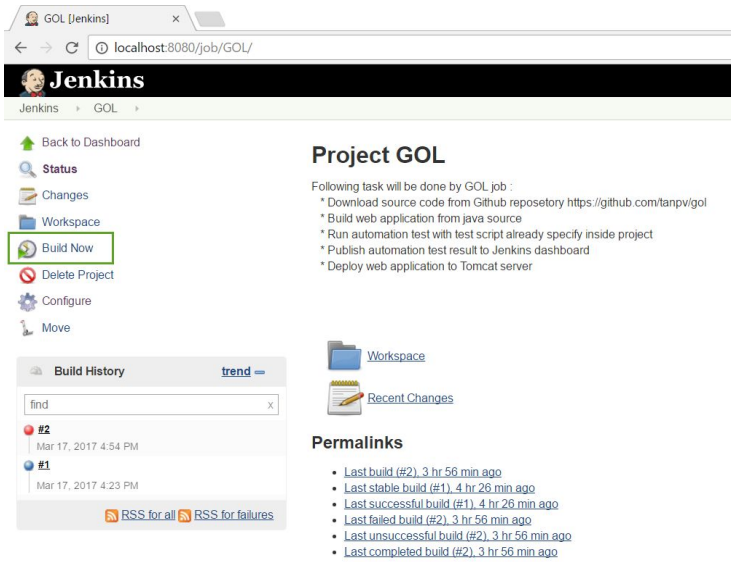


Now we could go to local repository to see the source code already downloaded. Each job will have it's own folder inside **workspace** folder

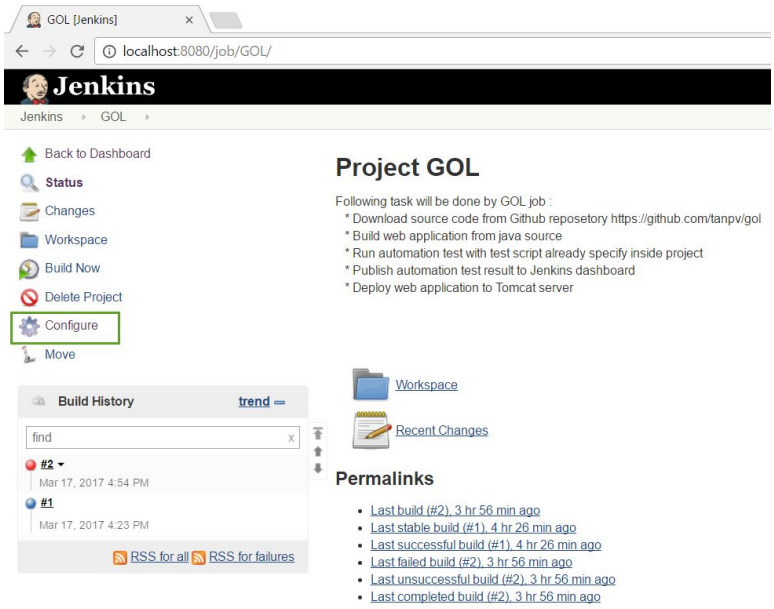


That it, we complete the first step of this job.

Build Triggers

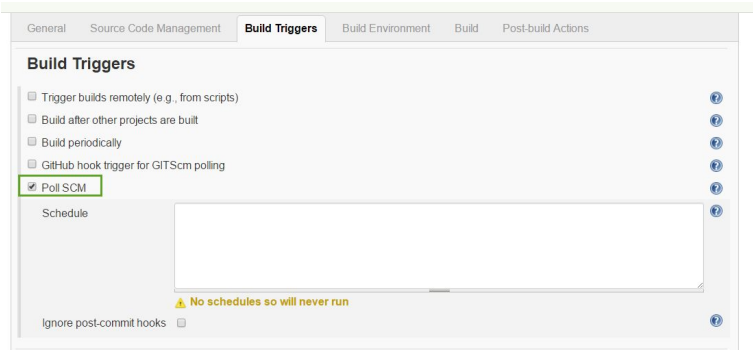


The first way and most simple way to start LOG job is just click in to ***Build Now*** button. But this way is manually and not so cool. We want to trigger Jenkins job running automatically, so to do this, click to ***Configure*** from LOG dashboard.



Then select to ***Build Triggers*** tab you will some option to configure so job could trigger automatically

Have 5 ways to automate trigger Jenkins job as show above. In this book I will focus on **Poll SCM,** the good way in configure for a continuous system. Now you click on **Poll SCM,** you will see a text box show up for setting schedule



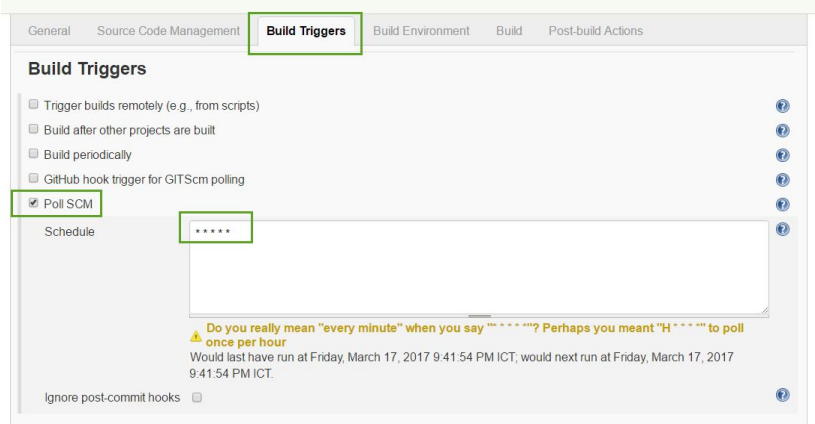
So following are steps show up how ***Poll SCM*** work :

- Base on schedule setting, Jenkins will actively check Github repository to see if have any change from repository
- If have any change on Github repo Jenkins job will trigger

Entry	Description	Equivalent To
`@yearly (or @annually)`	Run once a year at midnight in the morning of January 1	`0 0 1 1 *`
`@monthly`	Run once a month at midnight in the morning of the first of the month	`0 0 1 * *`
`@weekly`	Run once a week at midnight in the morning of Sunday	`0 0 * * 0`
`@daily`	Run once a day at midnight	`0 0 * * *`
`@hourly`	Run once an hour at the beginning of the hour	`0 * * * *`
`@reboot`	Run at startup	`@reboot`

```
*    *    *    *    *  command to be executed
┬    ┬    ┬    ┬    ┬
│    │    │    │    │
│    │    │    │    └─── day of week (0 - 7) (0 or 7 are Sunday, or use names)
│    │    │    └──────── month (1 - 12)
│    │    └───────────── day of month (1 - 31)
│    └────────────────── hour (0 - 23)
└─────────────────────── min (0 - 59)
```

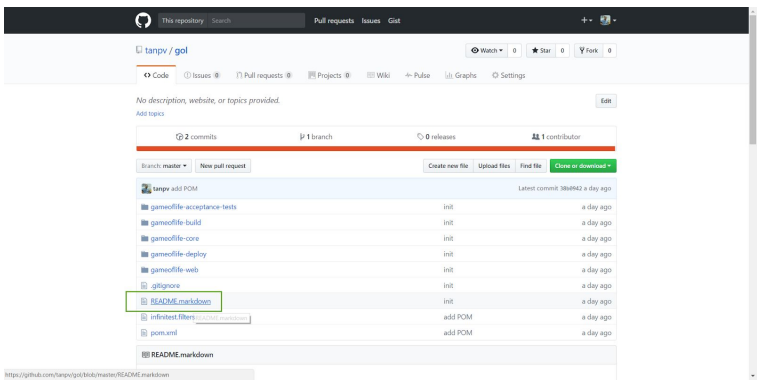
Schedule with ***Poll SCM*** work follow cron schedule rule, above image show some cron schedule example. Basically, we want to catch the change on Github repository as soon as possible, so follow text will be add to ***Schedule*** * * * * * . This mean we want to check change on Github every minute !

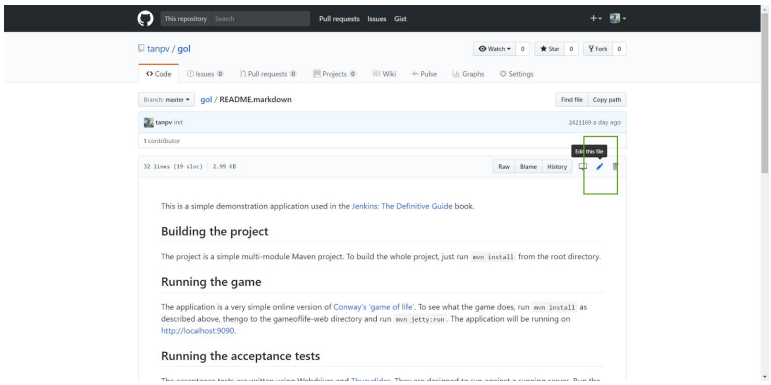


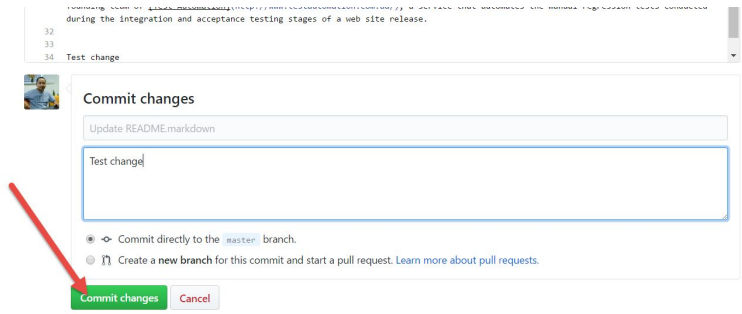
Click to **Save** and Jenkins will move you back to job dashboard

Now is the time to check if this setting really work ?

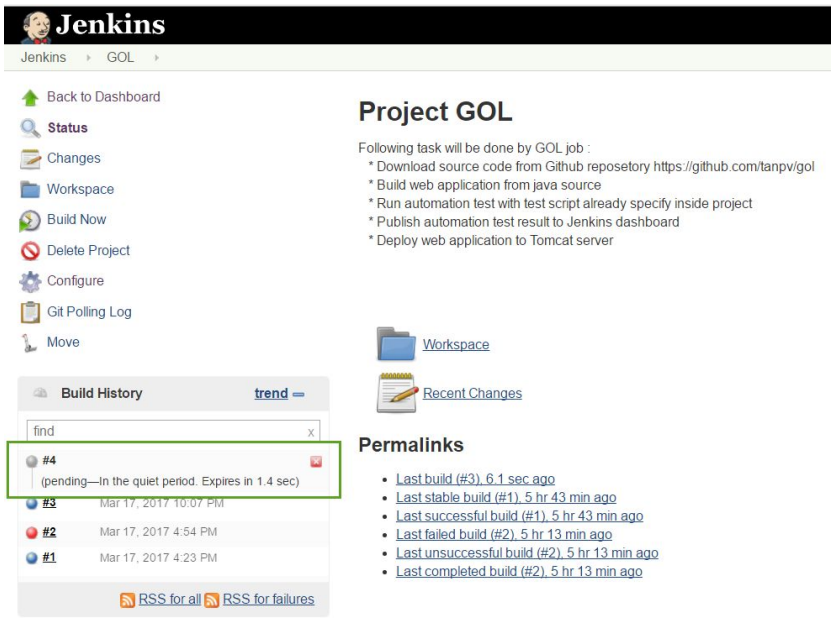
I will go to my GOL project repo on Github and change content of file ***README.markdown*** right from web browser (Github support commit change right from web browser)



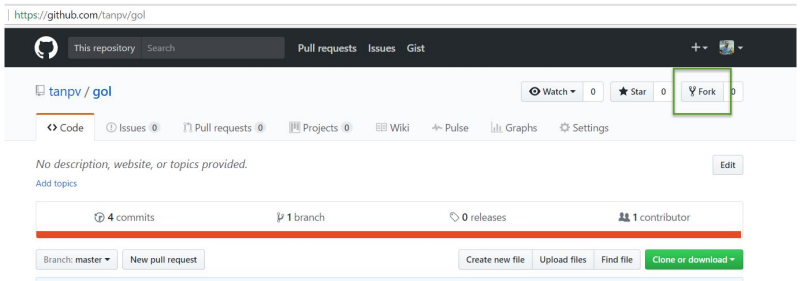


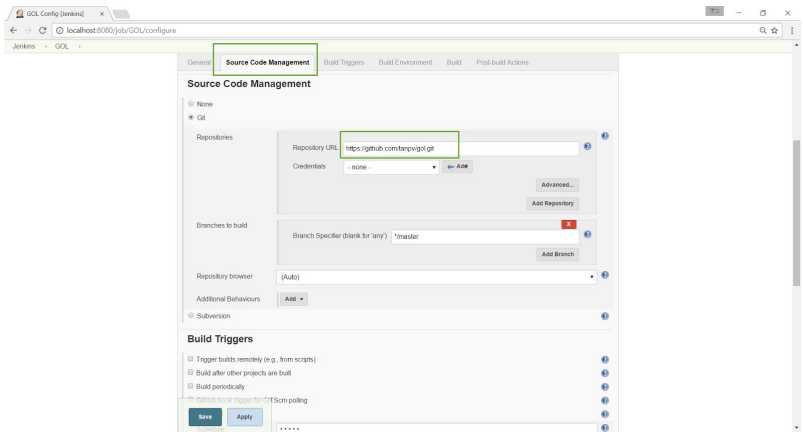


Now come back to job dash board, wait for about 1 minute, you will see new job is planning and run.



That it, it work !!!, to practice by yourself, just clone GOL project to your account by **Fork** button and then change the GOL configuration point to your github account as show below.



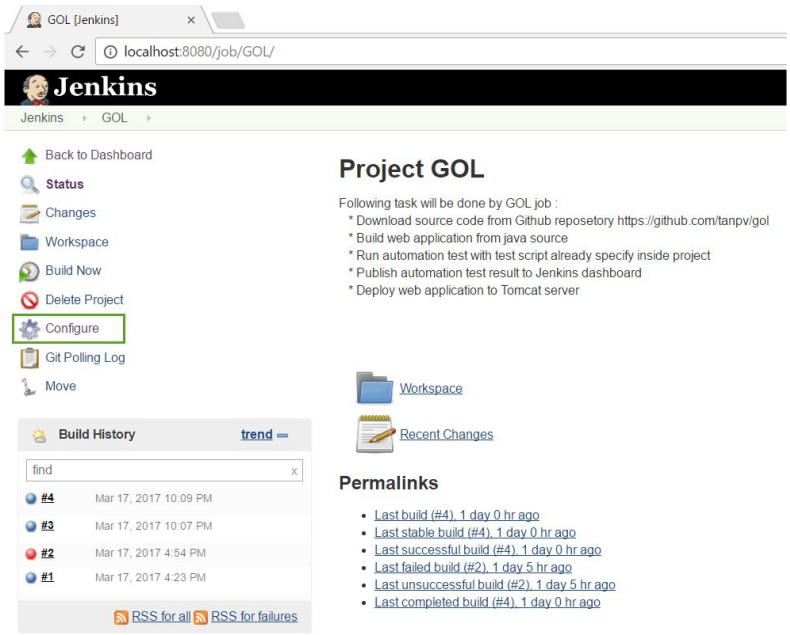


Build

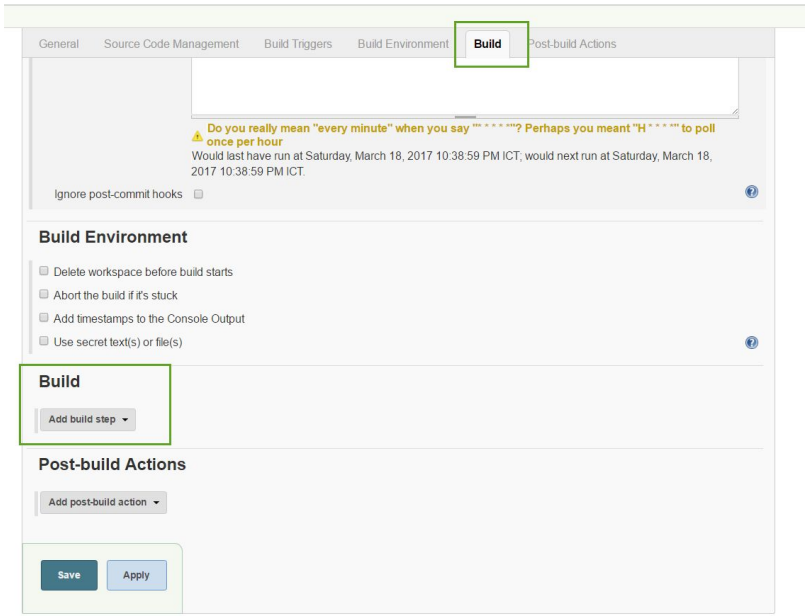
In this section, We will setting to build java web app with maven.

Configure Build Step

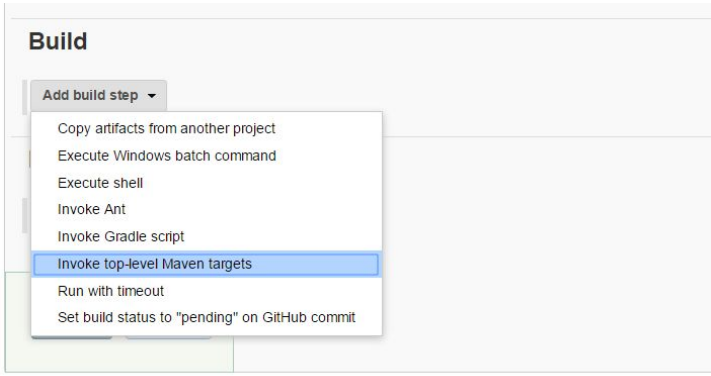
From GOL job dashboard, click to **Configure** link, the configure page will show up



From configure page, select **Build** tab, page will scroll down to **Add build step**

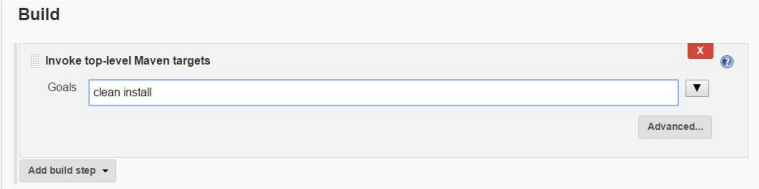


Click to **Add build step** and select **Invoke top-level Maven targets**



Section to adding maven command will show up as below

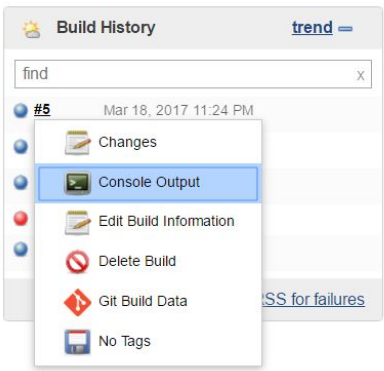
Put **Goals** maven command **clean install**



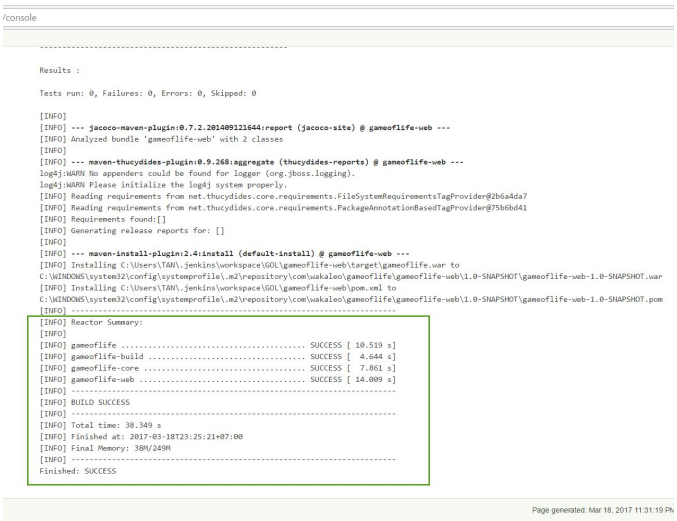
Then finally click in to **Save** button. That it we already complete configure for build maven step.

Check Log and Workspace

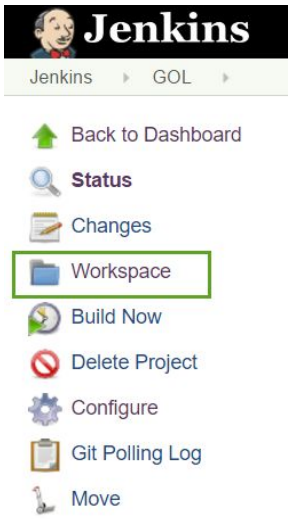
Now from dashboard of GOL, just click to **Build Now** to see how Jenkins job work by open console log



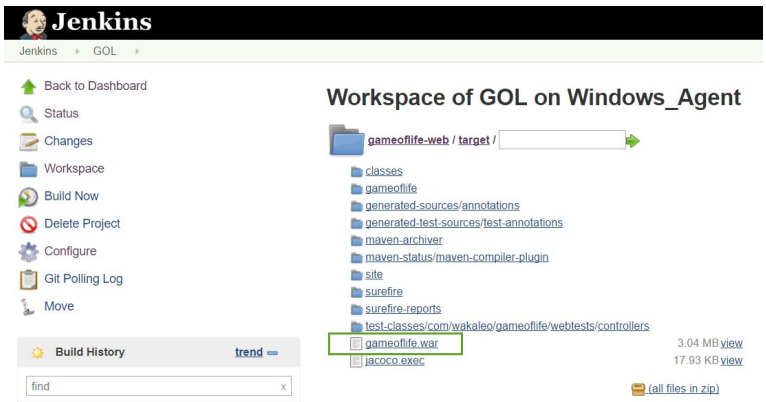
Scroll this log to bottom you will see build job run successfully



From GOL home page, click to **Workspace**



You will access directory where build happen and actually could see the war file inside **Gameoflife-web/target/.** You could actually download war file from here.



That it, We already finish setting up build step.