****

Contents

Introduction…………………………………………………………………………………………………………………………………..3

What is the Jenkins………..……………….…………………………………...…………………………………………….3

[How to Use This Test Drive Guide…………..……………….…………………………………...…………………………….3](#_Toc492045154)

Launching the Test Drive………………..................................................................................................................3

[Scenario:…………………………………………….…………………………………………………………………………………..4](#_Toc492045160)

# Introduction

## What is the Jenkins

[Jenkins](http://jenkins-ci.org/) is an open-source [continuous integration](http://searchsoftwarequality.techtarget.com/definition/continuous-integration) software tool written in the [Java](http://searchsoa.techtarget.com/definition/Java) programming language for testing and reporting on isolated changes in a larger code base in real time. The software enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.

**Continuous Integration:**

Continuous Integration is a development practice in which the developers are required to commit changes to the source code in a shared repository several times a day or more frequently. Every commit made in the repository is then built. This allows the teams to detect the problems early. Apart from this, depending on the Continuous Integration tool, there are several other functions like deploying the build application on the test server, providing the concerned teams with the build and test results etc.

## How to Use This Test Drive Guide

This test drive guide features instructions to guide users through a specific use case scenario for Jenkins. These steps may build off of the information show in the video shown on the test drive launch page, which introduces the tools to be used and shows the configuration of the servers we will be using. Please follow the steps as they are laid out here in this guide for the best user experience.

## Launching the Test Drive

Once you have signed up for the Jenkins test drive from the qloudable, you will be taken to a launch page. The test drive should launch in 15 minutes or so.

Please review the documentation and any additional links while you wait.

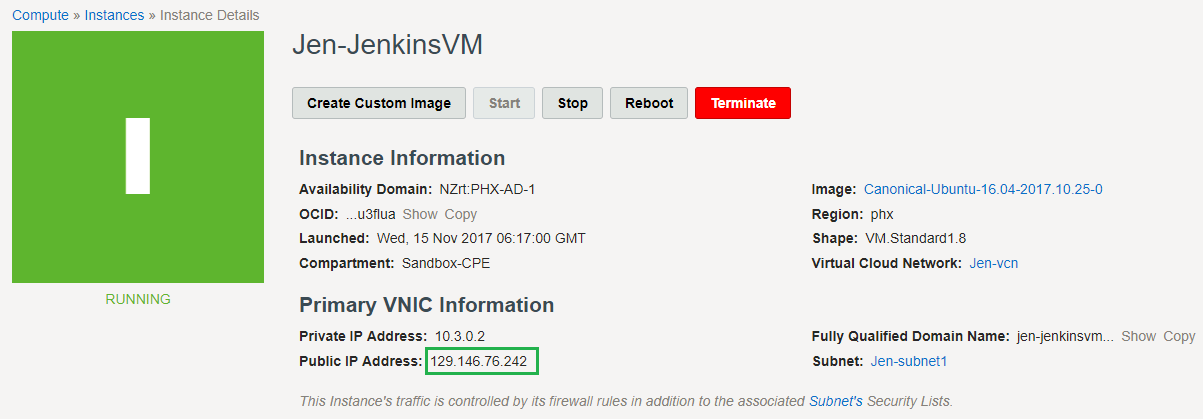
Once the test drive is ready, you should receive the credentials and URL(s) needed to access the test drive in an email sent to the address you used to sign up.

# Scenario:

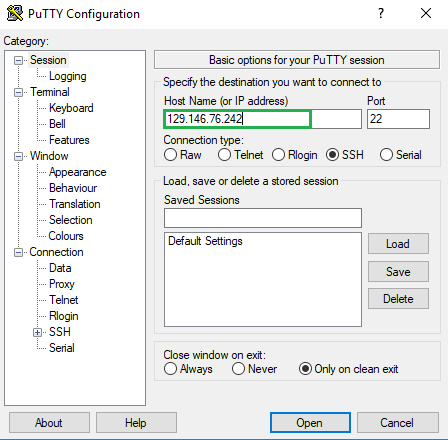
**Objectives:**

* Launch the test drive.
* Setup and configure the GitHub repository.
* Configure Jenkins job.
* See Jenkins in action.

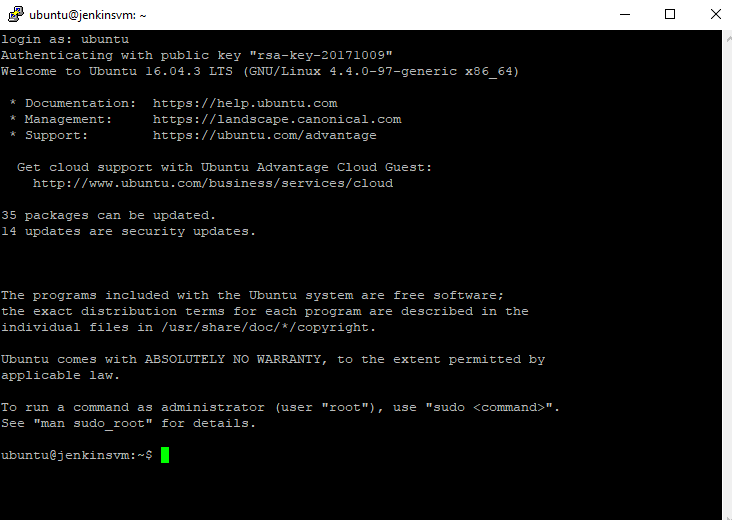
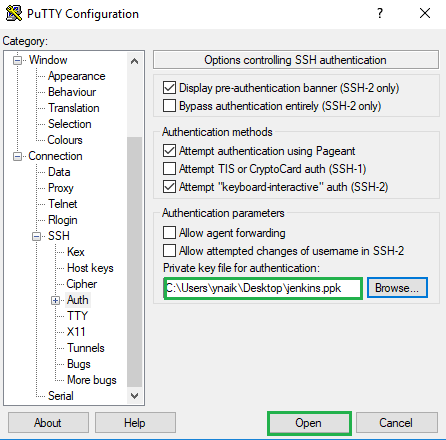
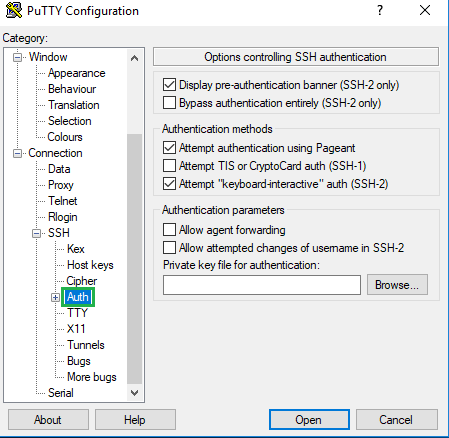
1. Log into your SSH session using client SSH URL provided in test drive launch page.



2. Open putty and ssh into the VM using the IP address



Click on browse and select the private key you have stored in your local system

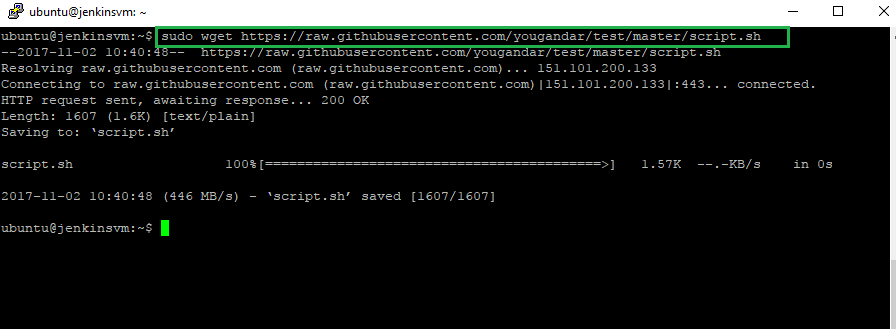


3. Once you login into the VM you need to run a script to configure the job for the Test Drive

**Run the below command**:

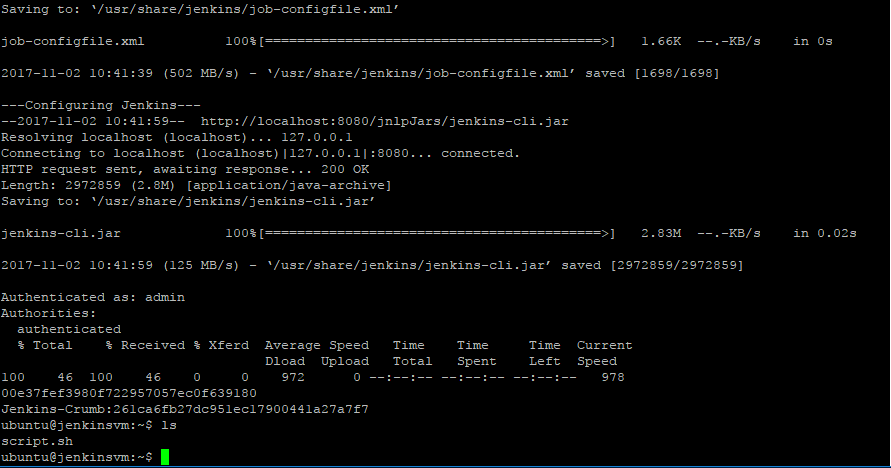
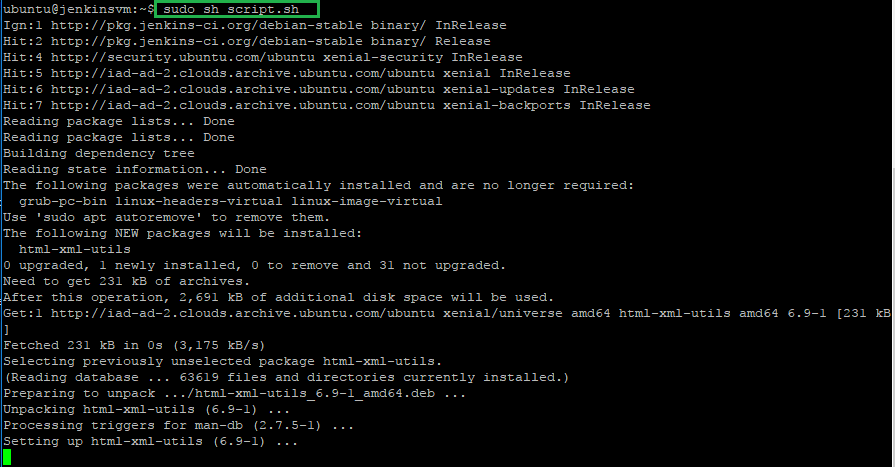
***sudo wget*** [***https://raw.githubusercontent.com/yougandar/test/master/script.sh***](https://raw.githubusercontent.com/yougandar/test/master/script.sh)

This command basically downloads the script from the given location



To run the script run the following command

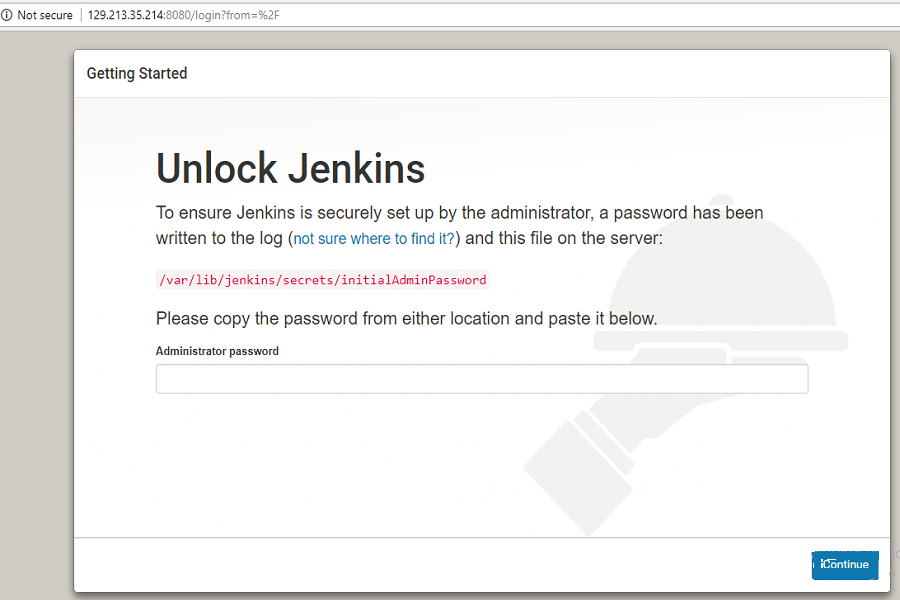
***sudo sh script.sh***



4. Now open any browser and open the Jenkins dashboard using the Jenkins IP address we get from the output section

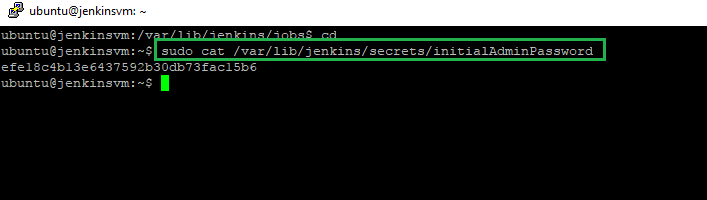
Ex:[***http://JenkinsVMIP:8080***](http://JenkinsVMIP:8080)

***http://129.146.76.242:8080***

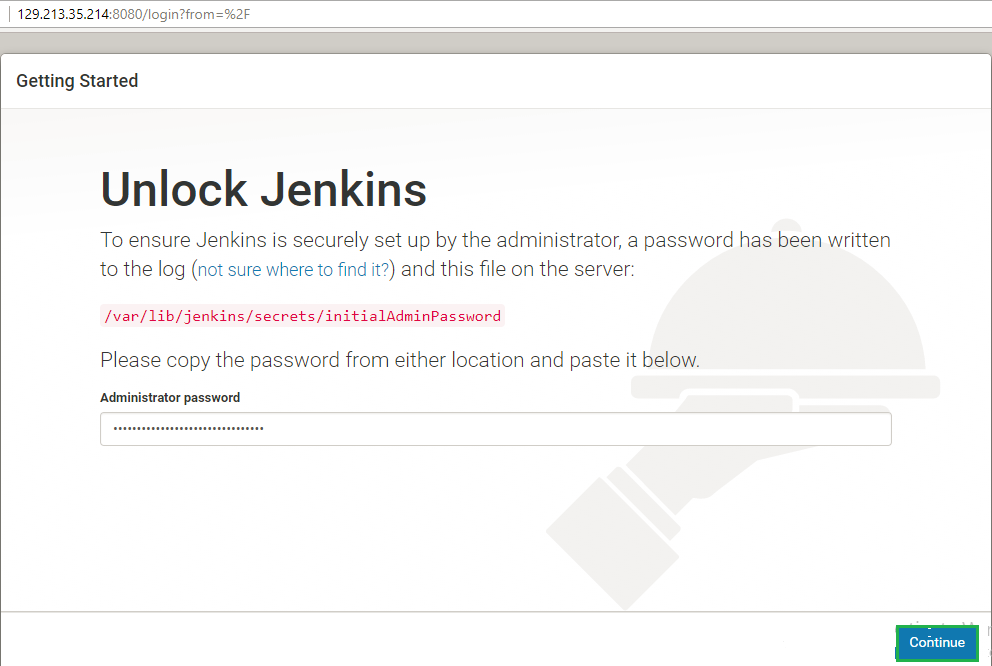


5. To unlock Jenkins you need to login to the Jenkins machine and use the following command to access the initialAdminPassword

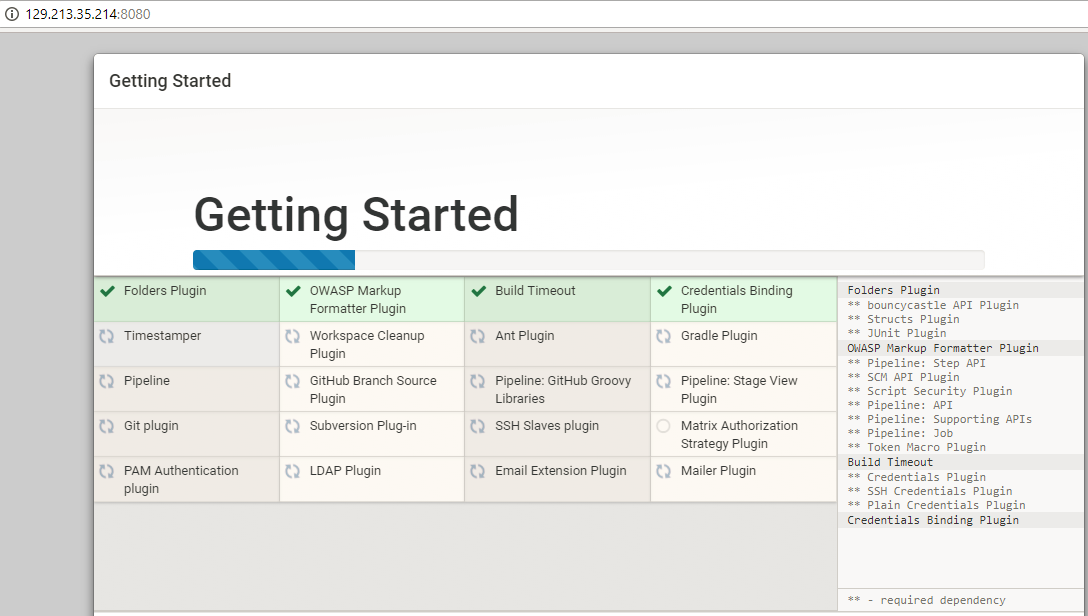
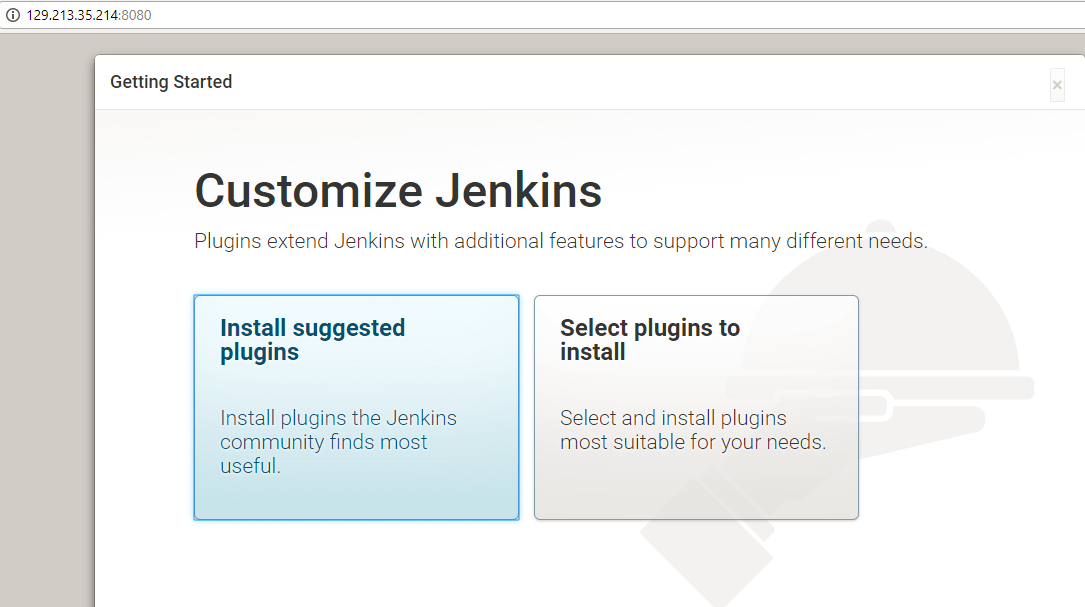
***sudo cat/var/lib/Jenkins/secrets/initialAdminPassword***



Paste the password you get as shown below

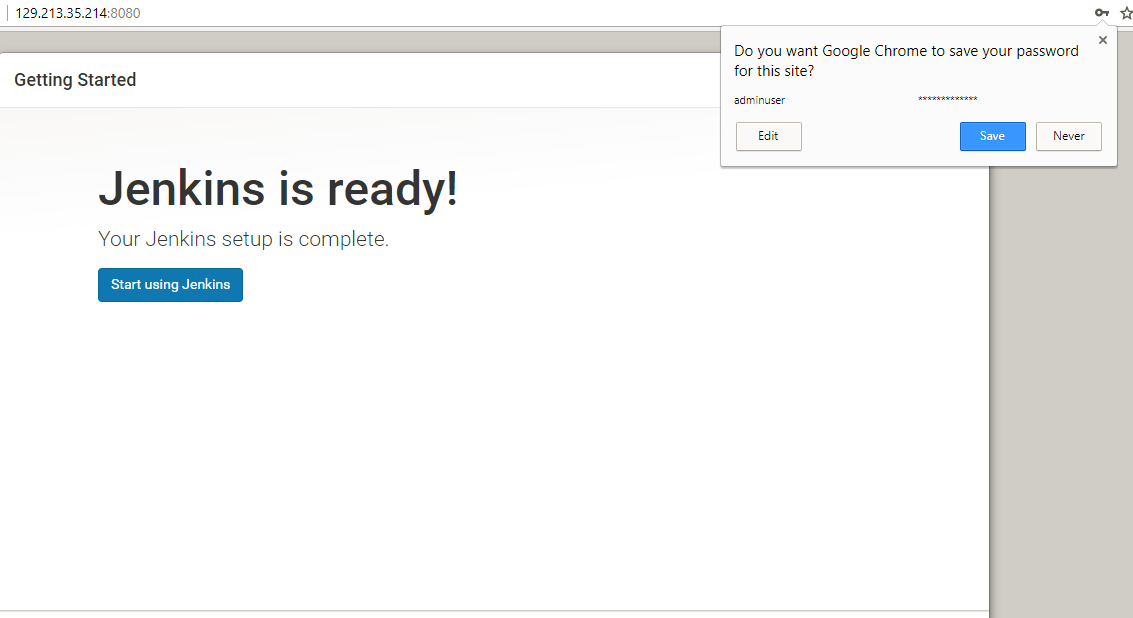
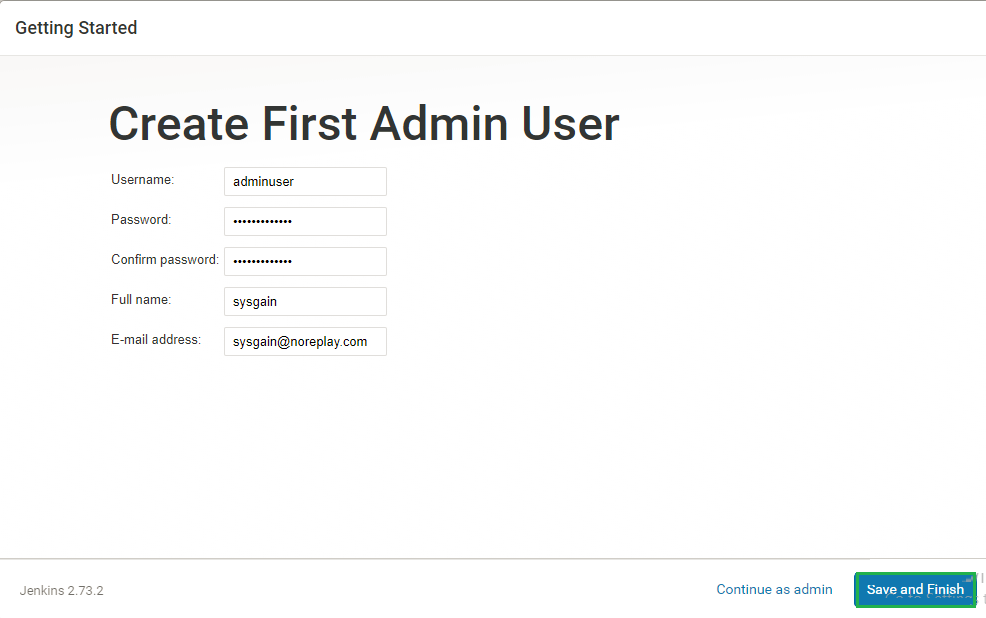


When prompted please click on install suggested plugins, so Jenkins automatically installs the necessary plugins

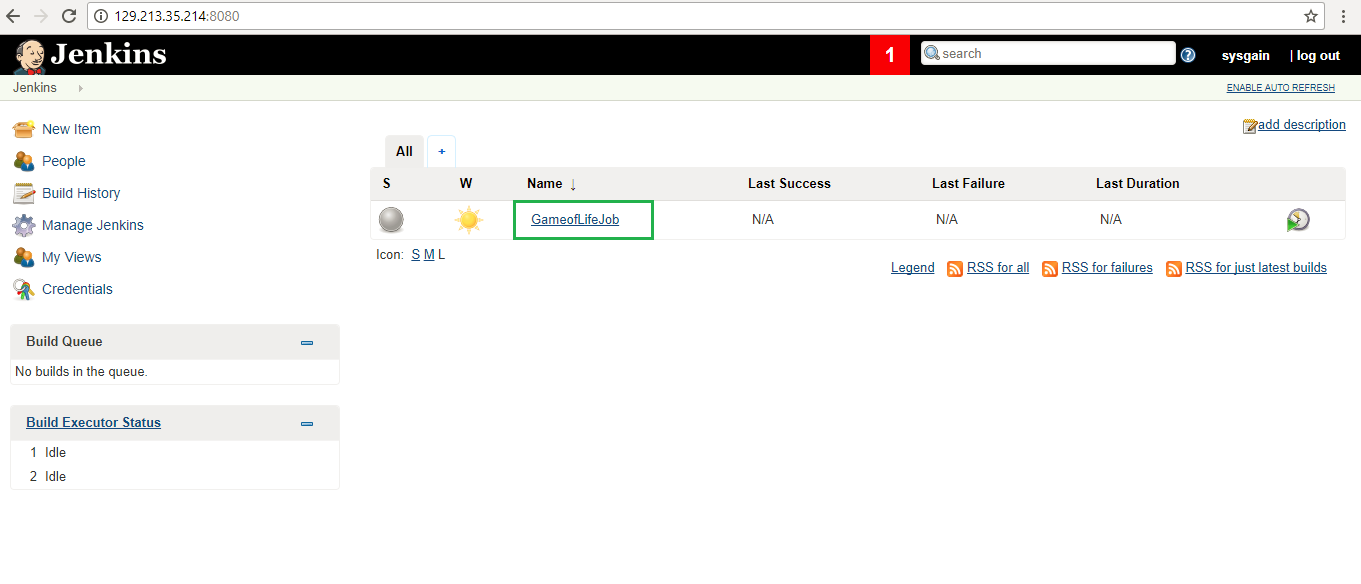


Please wait while the plugins get installed

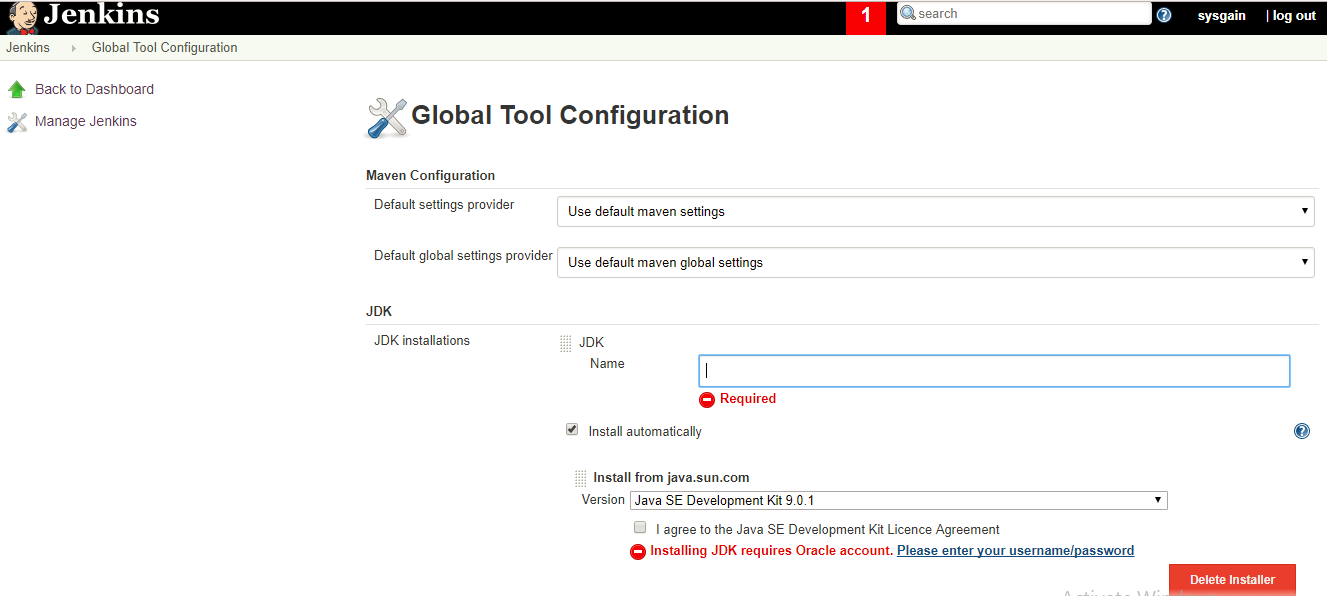
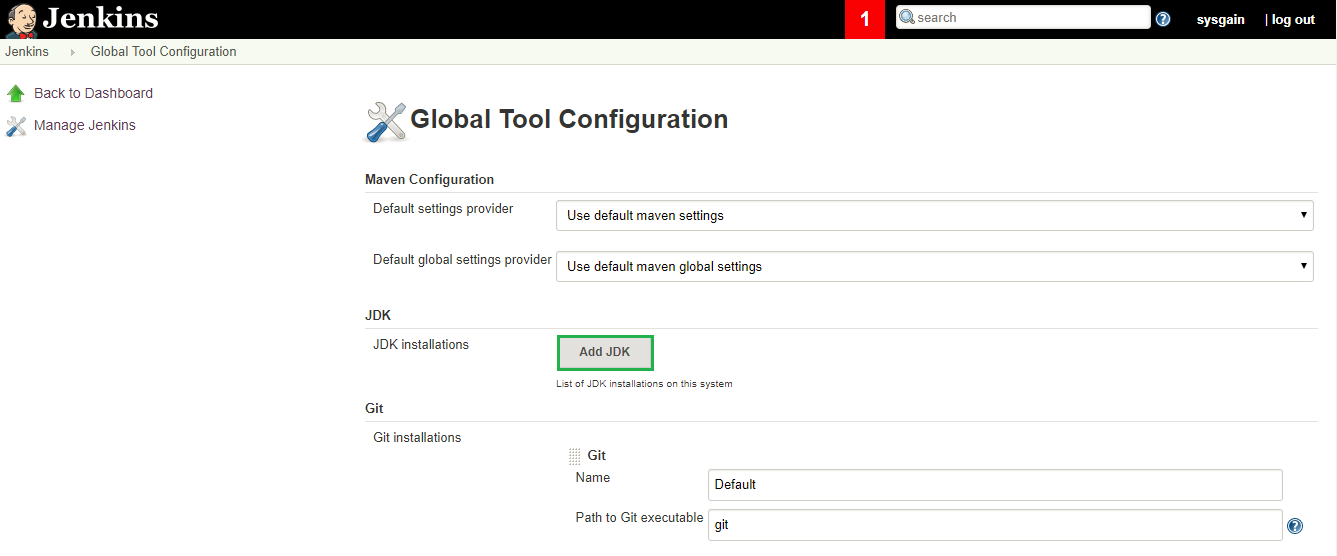
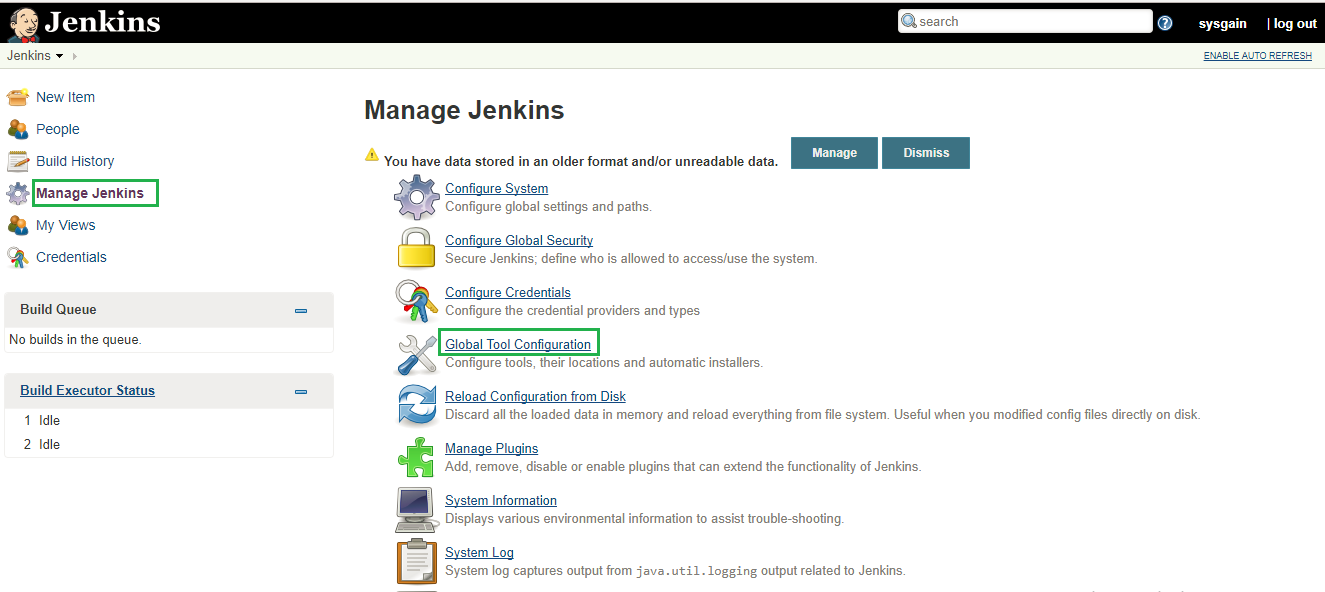
6. When prompted create a new admin username and password which you can use to access the dashboard.



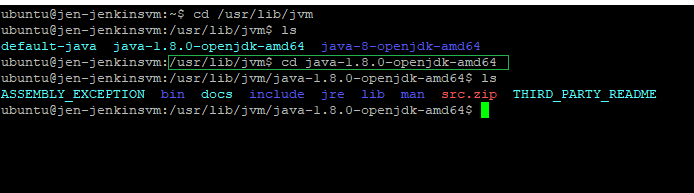
7. When you access the dashboard you can see a job configured: i.e. **GameofLifeJob**

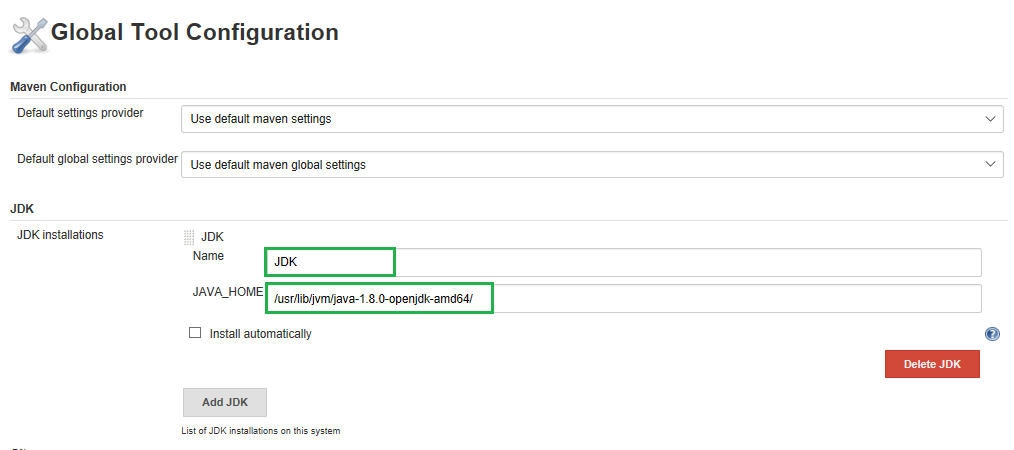


8. Go to manage Jenkins click on global tool configuration, here you need to add the JDK path as shown in the below screenshot

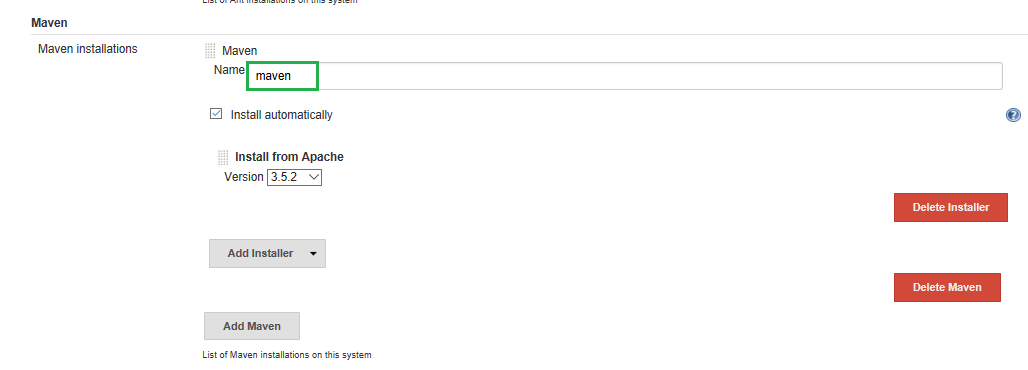


9. Please paste the path as **/usr/lib/jvm/java-1.8.0-openjdk-amd64.**Choose any name

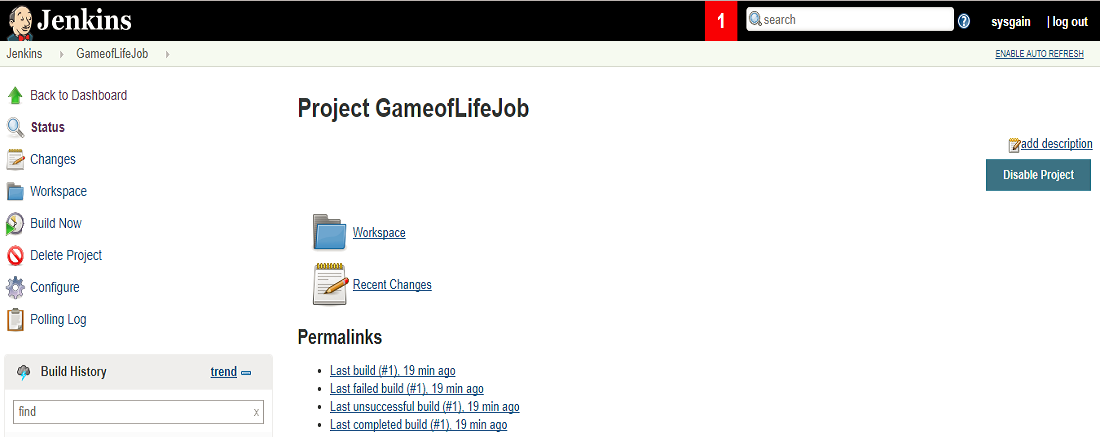




10. Under maven give any name and select the checkbox to install automatically



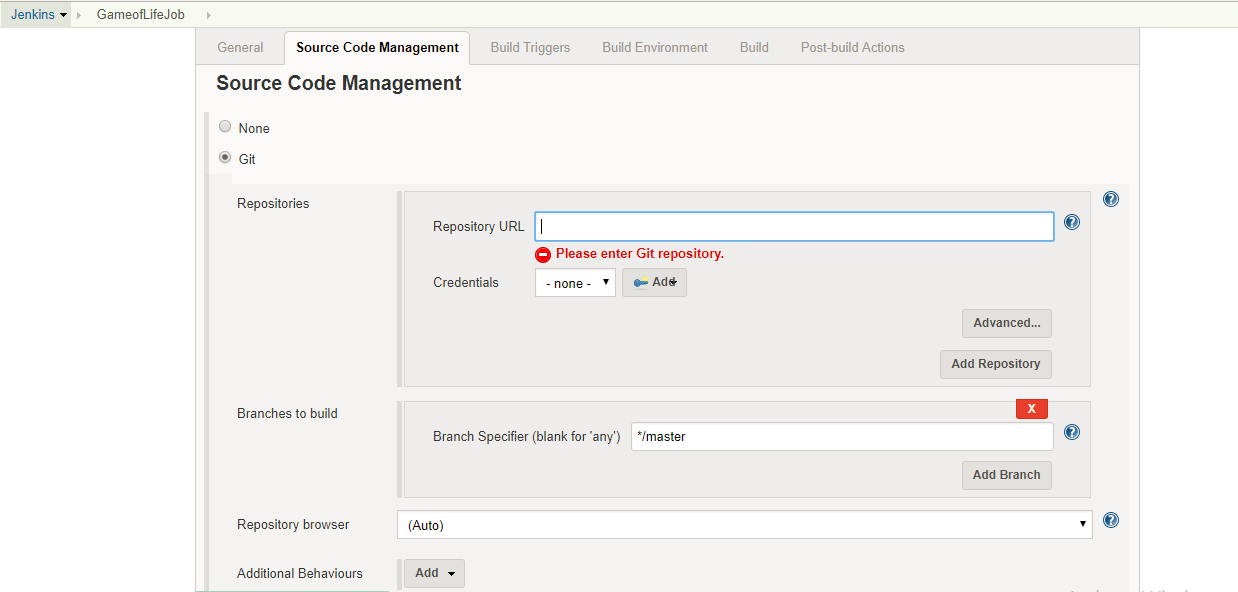
11. Now click on the job and click on configure.

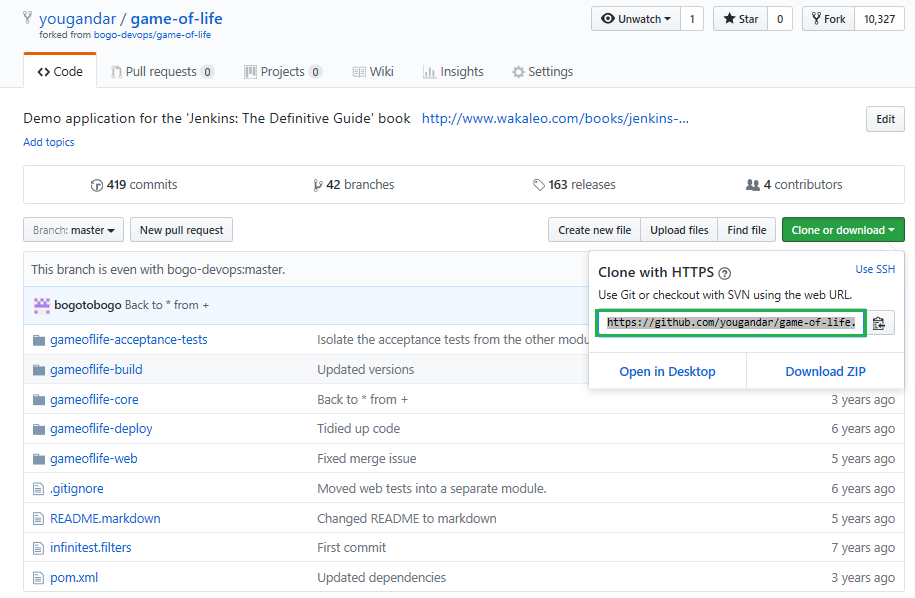


12. Under SourceCodeManagement give the GitHub repository URL (the game of life repository which you have forked earlier)

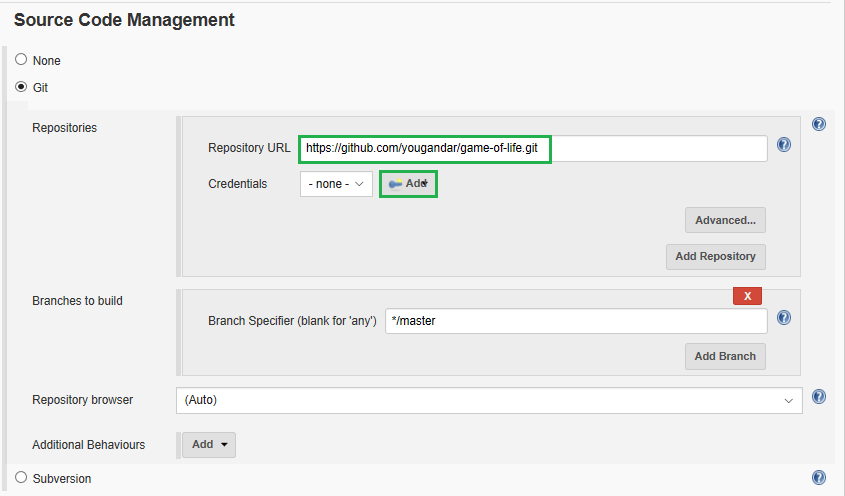
EX: [***https://github.com/yougandar/game-of-life.git***](https://github.com/yougandar/game-of-life.git)

Under branch specifier select \*/master

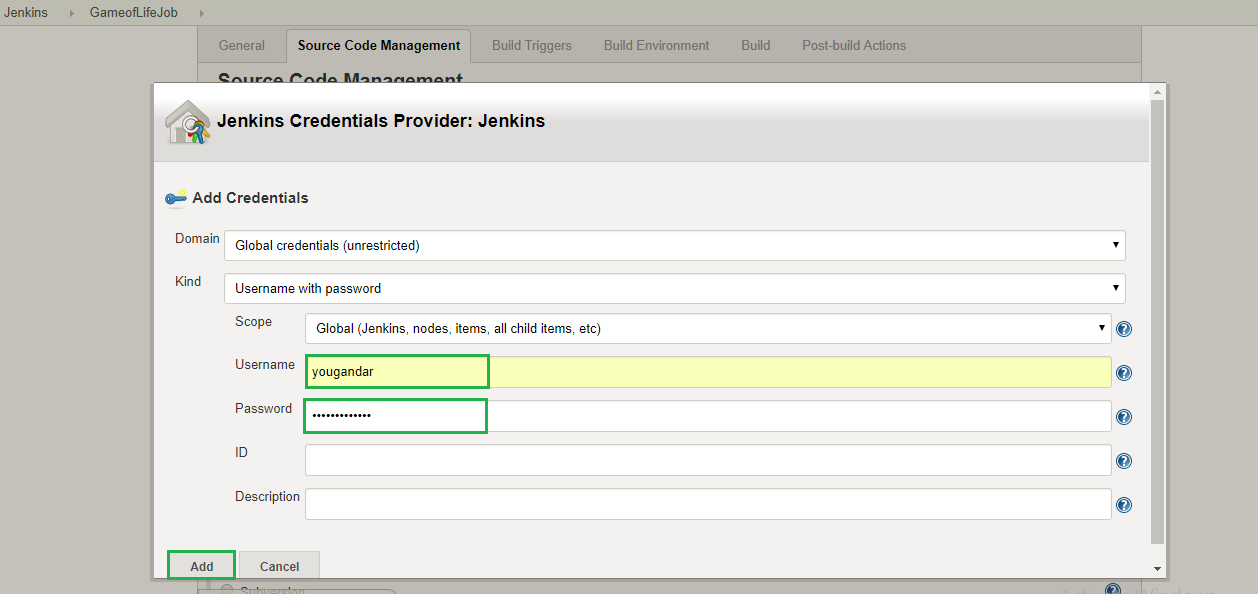


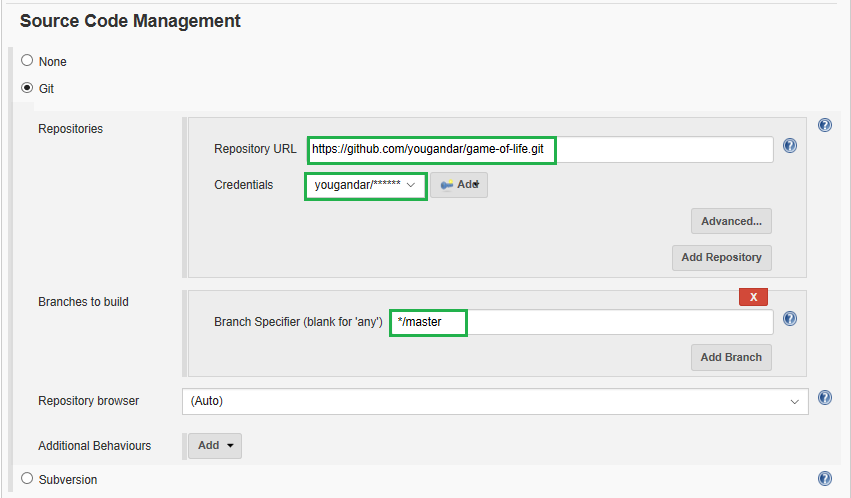


Click on Add -> Jenkins ->It will open the Jenkins Credentials provider

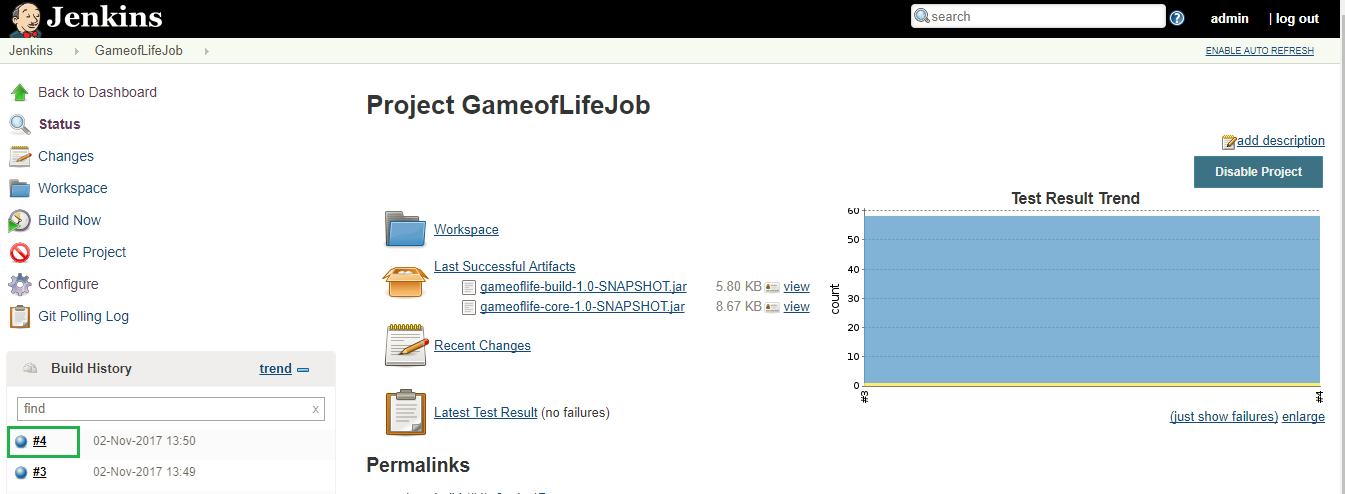
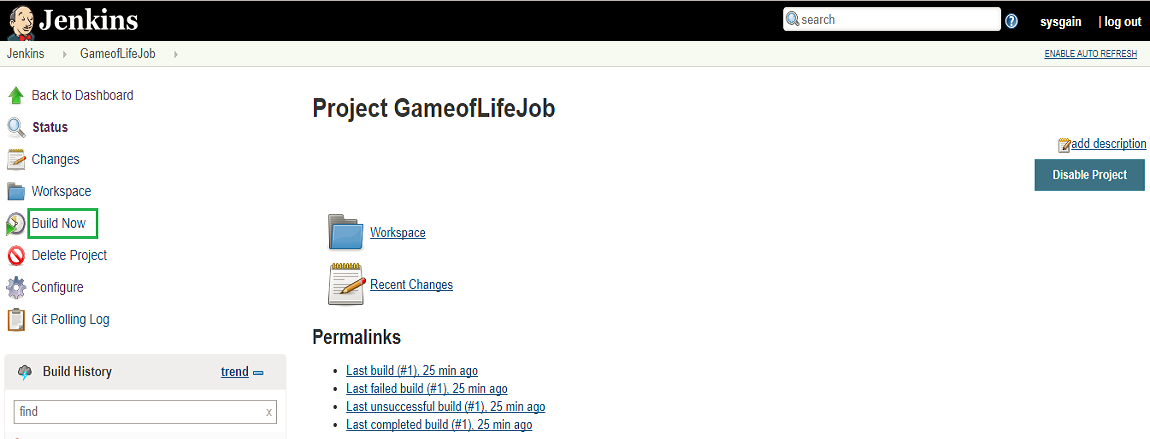


13. Provide the credentials of your git account so Jenkins can access your repository

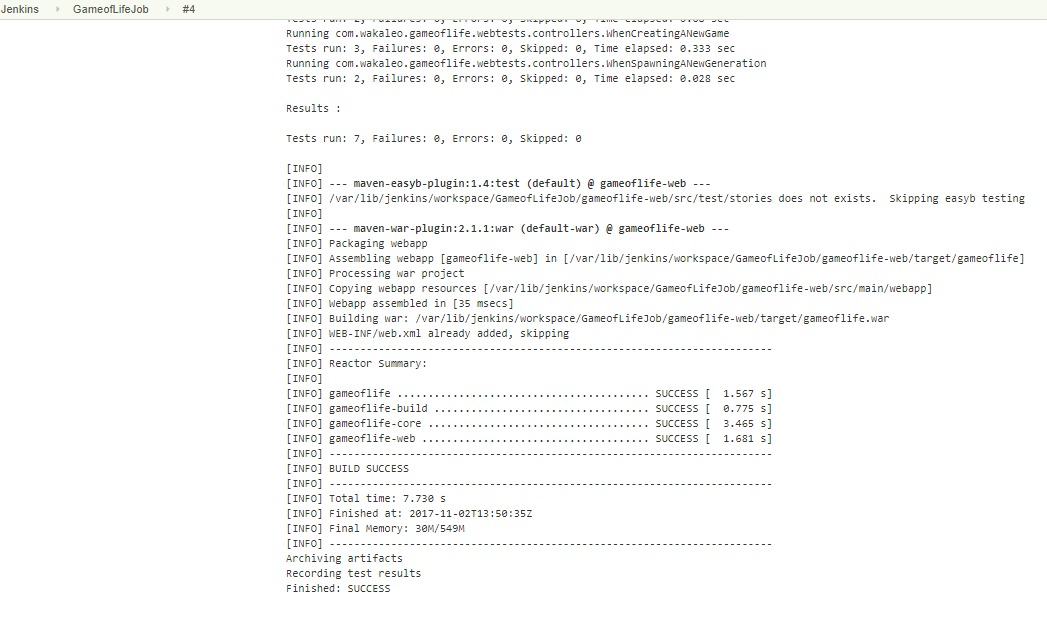




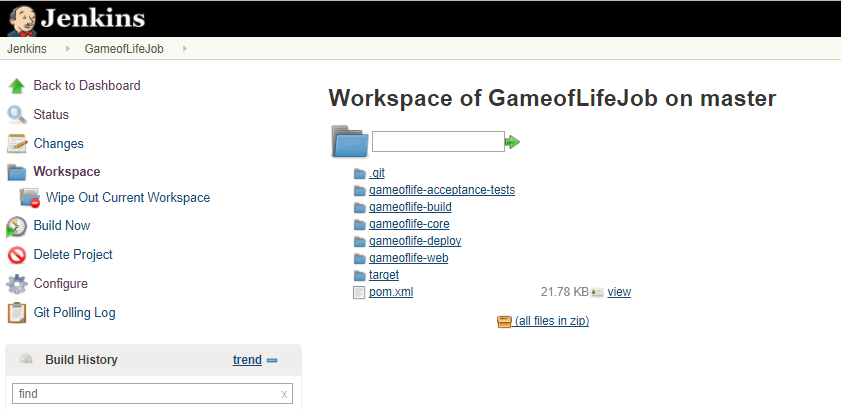
14. Now when you build the project Jenkins performs the tests and give the test results accordingly indicating success or failure.



In the below screenshot we can see that the build was successful in the console output.

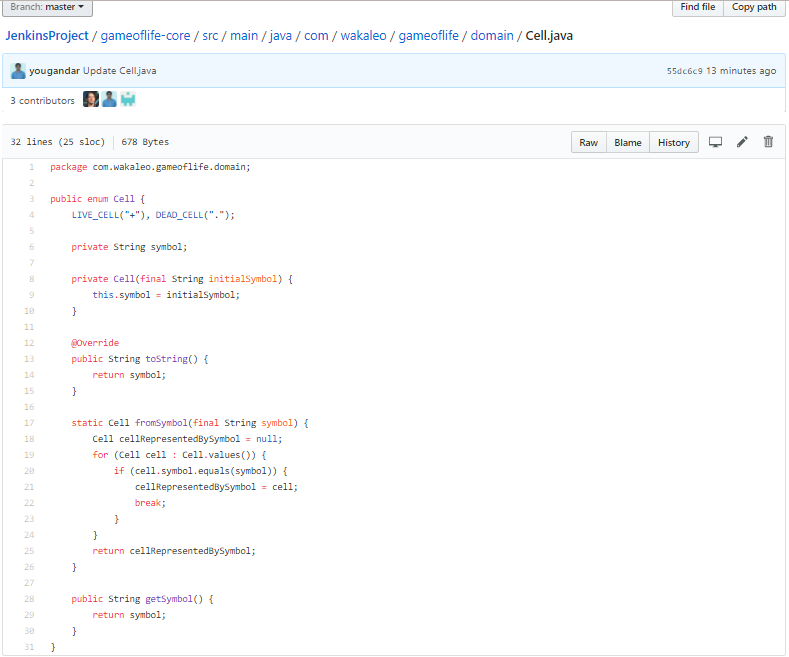


15. Click on the Workspace of project as show in the below screenshot.



16. Now, what we can do is change to source code to demonstrate Jenkins failed test results.

In the source code if we change the LIVE\_CELL (\*) to LIVE\_CELL (+) and build the job again.



17. Here we can see that the build has failed and also review the results as what has caused the failure.



18. Now once we revert back to the original source code, we can see that build is successful.

And also we can see the test results trend.

