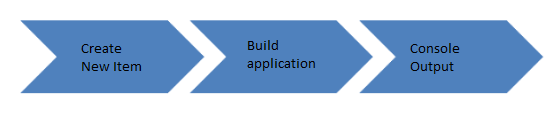
**OJAS BUSINESS SUITE CI&CD IN JENKINS**

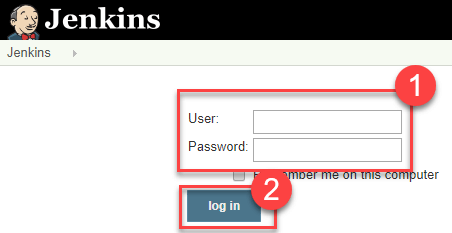
**What is a Jenkins Maven project** : Maven uses **archetype** plugins to create projects. To create a simple java application, we'll use maven-archetype-quickstart plugin. In example below, we'll create a maven based java application project.



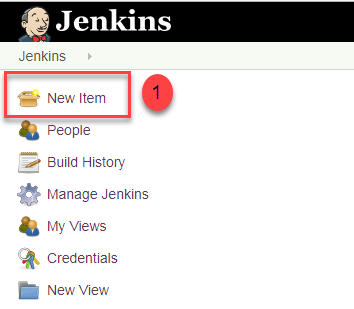
## Creating a Maven Build Job

The Maven build job is a highly flexible and easy-to-use option. You can use it only maven projects is easy to set up, and many of its options appear in other build jobs.

**Step 1)**To create a Jenkins Maven job, log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at [http://localhost:8080](http://localhost:8080/) If you have installed Jenkins in another path, use the appropriate URL to access your dashboard.

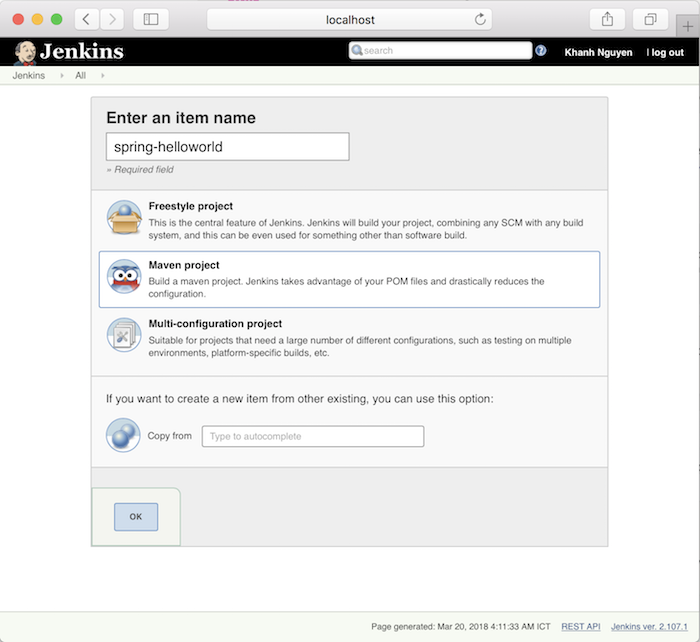


**Step 2)**Click on "**New Item**" at the top left-hand side of your dashboard.

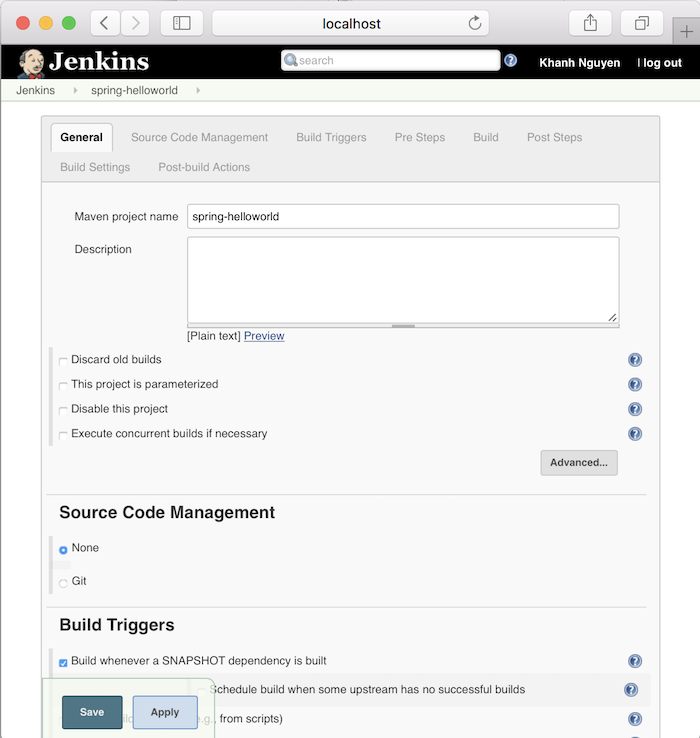


**Step 3)**In the next screen,

1. As you can see, in this step, we will select the type of project we want to build with Jenkins. There are many different options, we just focus on Maven project, so after entering the name, called Jenkins job, in the Enter an item name field, then select the Maven project.
2. Myself as follows:



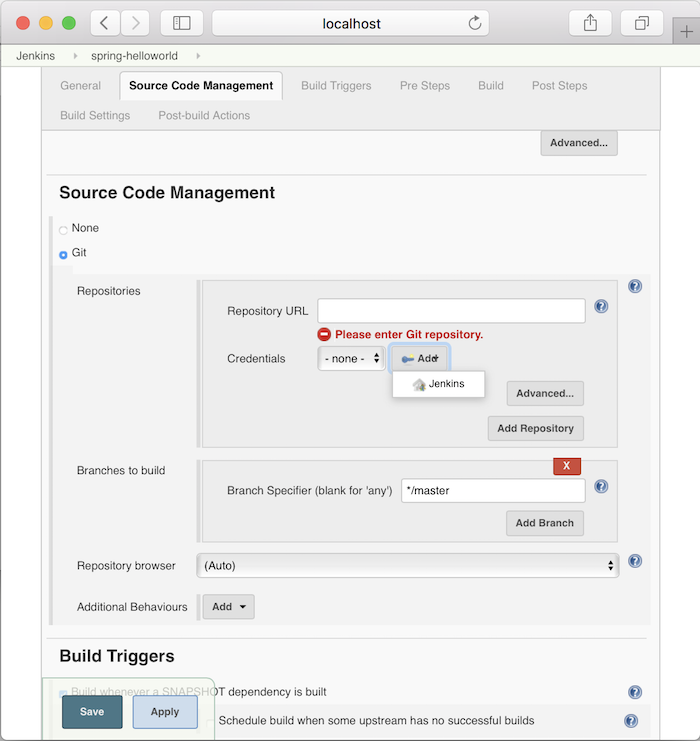
Click OK to continue.



**Next step, we will configure Git SCM information.**

This step allows Jenkins to clone the code from Git systems.

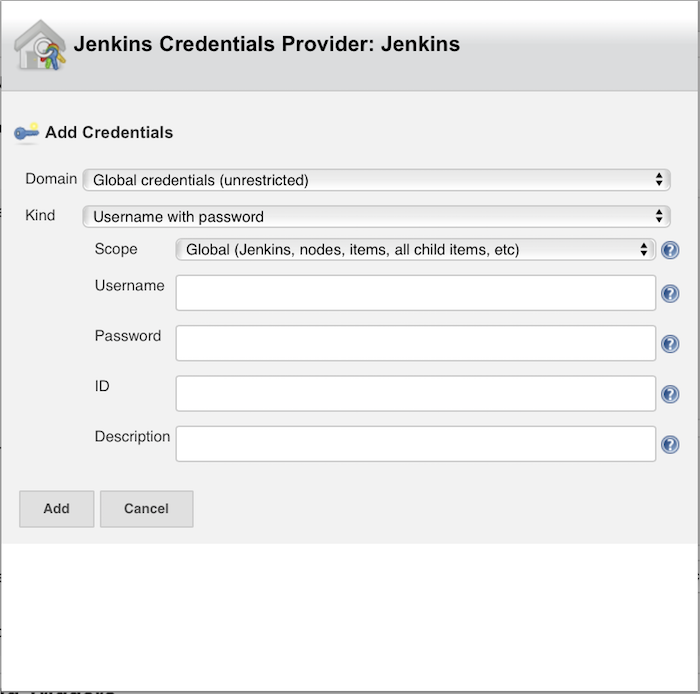
Go to the Source Code Management and click on Git as follows:



There are 3 things we need to configure for this Git SCM:

* The first is the URL to clone the code in the Repository URL section
* Second is the credentials login to the Git system we have declared in the Repository URL
* And finally, the Git branch we want to build.

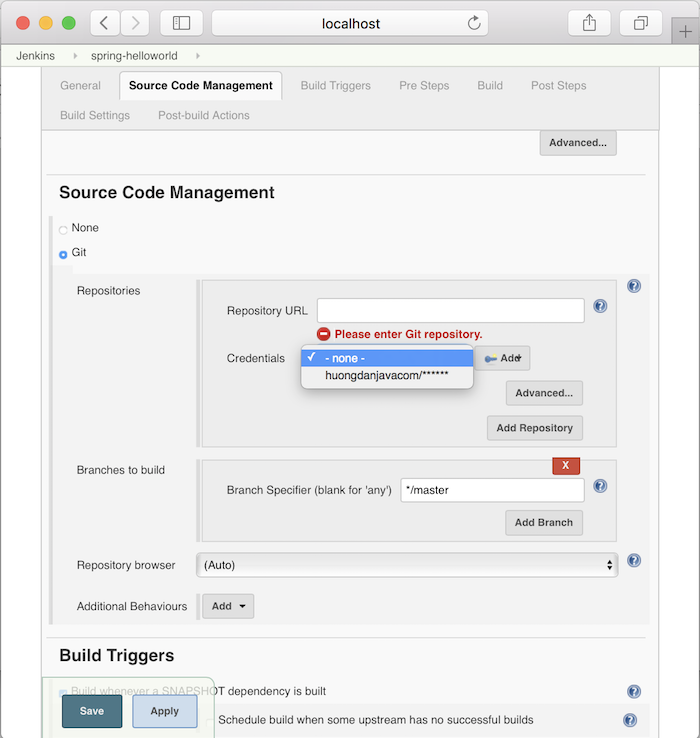
In the Credentials login information, please click the Add button

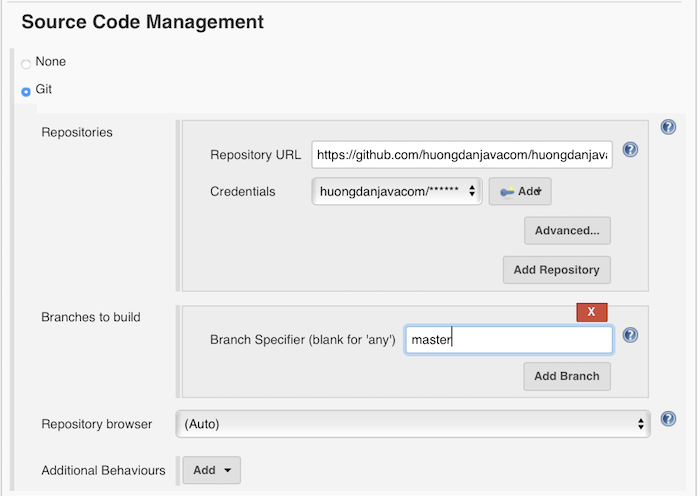


to add information about username and password.

In the window above, you only need to enter Username, Password and then click the Add button is okay.

Then the credentials login information will appear information about the Username that we just added.





Please go to the Build section:

**The thing we need to do is to configure the pom.xml file and the Maven statement that we want to run when our build project.**

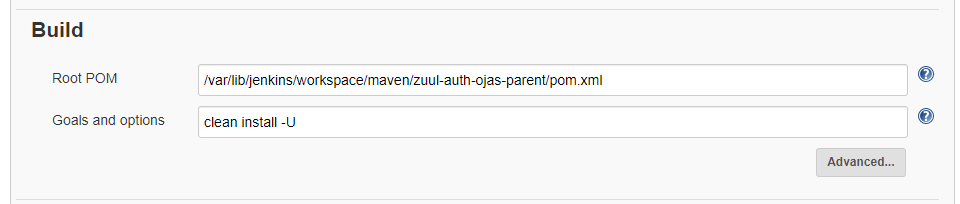


Root POM is the location of the pom.xml file. Let’s imagine when the clone code comes in, we will have a directory containing all our code. The path of the pom.xml file will start from this directory.

In our example, when clone code returns, in order to get to the pom.xml file of the spring-helloworld project, we have to go through the spring-hellworld directory first. Therefore, the location of the pom.xml file in this example will be spring-helloworld/pom.xml, right ?

The Goals and options section is for configuring the Maven statement.

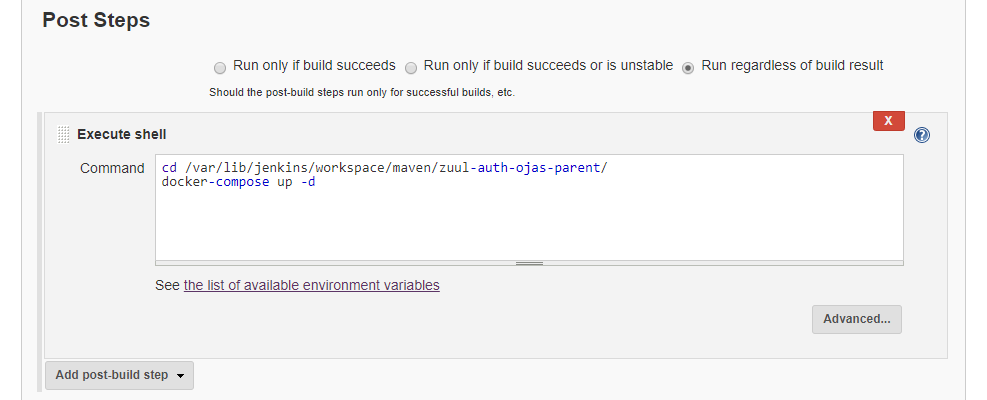
Here, I will configure the Build as follows:



Project Deployment Step :

In postbuild action you need to specify the docker compose file PATH for

Running the containers.



That’s it. You can click Save or Apply button to save.

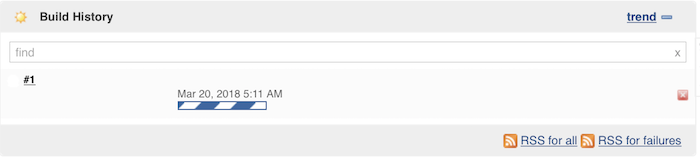
Now, let’s try to build the Maven project.

After you click Save, you will go to the current Jenkins job page.



Now click the Build Now button to start building this project.

A Build History section will appear :



Build successful:



You can click on # 1 in the Build History section, then in the Console Output section to see how the build progresses.

Myself as follows:

