Jenkins Github Integration

Creating Jenkins Job using Git

In this chapter, we will integrate GitHub and Jenkins. We can give instructions to Jenkins about what it should do when we trigger a build. We can take the source code of our project from GitHub. We will tell Jenkins when to start the build. First, we should download the source code of our project from GitHub and build it. So to do that we will learn two options,

Poll SCM: Poll SCM (Source Control Management) periodically polls the SCM to check whether changes were made (i.e. new commits) and builds the project if new commits were pushed since the last build, whereas build periodically builds the project periodically even if nothing has changed.

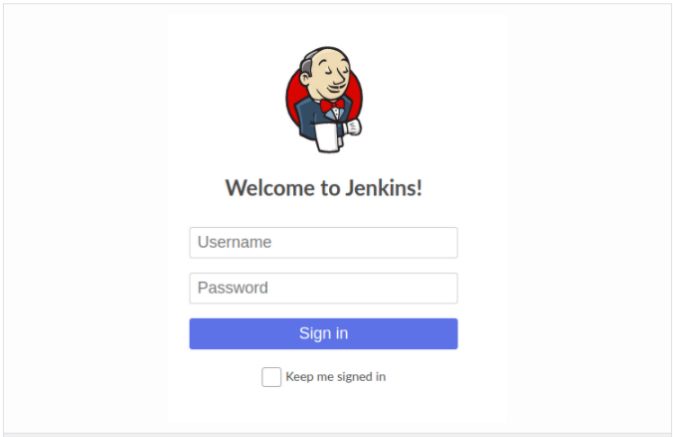
Webhooks: A webhook is a way to deliver real-time data to applications.

Tips:
You can think about webhooks like push notifications on your mobile phone. Rather than burning up the battery on your phone fetching information (polling) from applications to get updates, push notifications (webhooks) automatically send data based on event triggers. And just like push notifications, webhooks are less resource-intensive. Webhooks are far more efficient than polling, from a resource and communication standpoint.

> In the following steps we will create two different projects to show both options.

Integration Github with Poll SCM

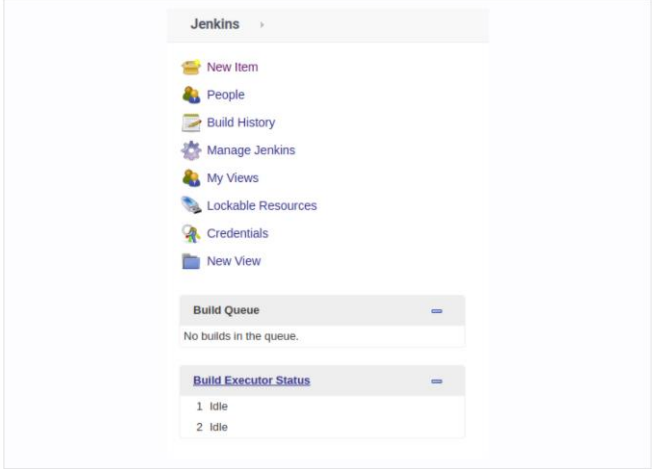
1. Open Jenkins and login to your account.



Login Page

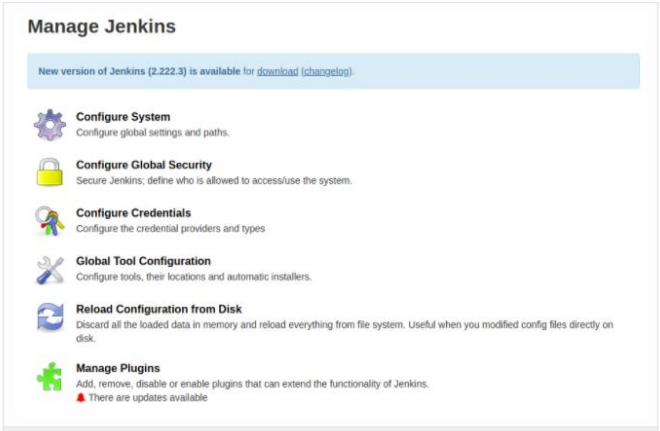
2. Click on "Manage Jenkins"

2. Click on "Manage Jenkins"



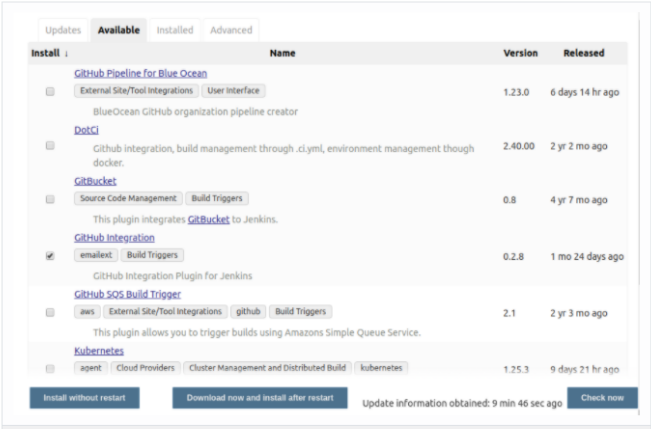
Manage Plugins

3. Manage plugins



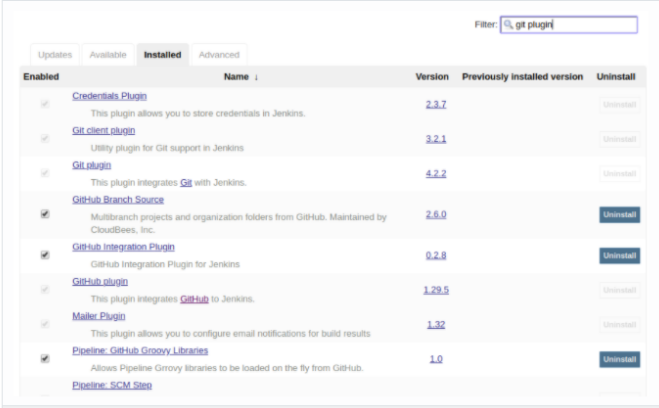
Manage Plugins

4. There will be a list of all available plugins for Jenkins. See available plugins to install from "Available" tab.



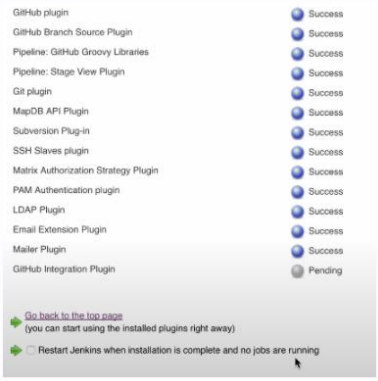
Available Plugins

5. Search and select "Git plugin" and "GitHub Integration Plugin" then click to "Install without restart". While finding plugins you can use Filter from the top right corner. If these plugins not shown at the search results. They are probably already installed, you can check it from installed tab.



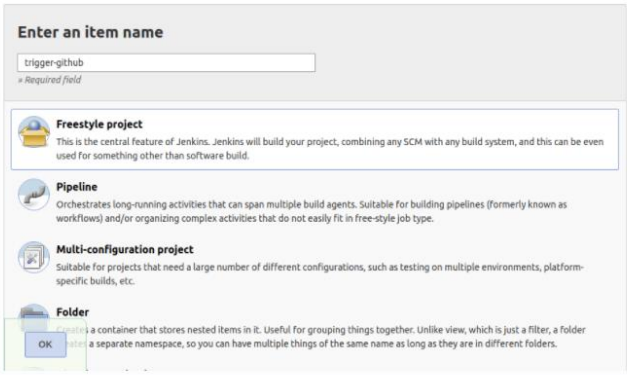
Available Plugins

6. Plugins will be installed with their dependencies. After the installation has finished, click on to the "Go back to the top page" link.



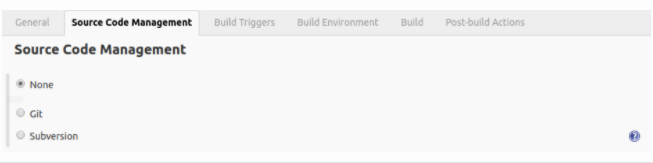
Go back to top page

7. Create a new job and give a name to your job, select "freestyle project" then click "OK". Then you can write a description about your item



Create a project

8. Go down to the **Source code management** and click on **Git**.

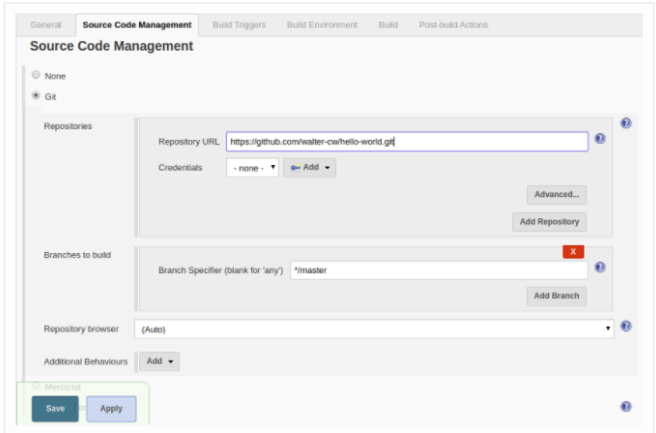


Select git

9. We should tell Jenkins where our Git Repository is. So you can give the link to one of your public repositories on GitHub. Please use your own repository. If your repository is private, you should use your credentials. But in this example, our repository is public so that's not needed. Also, you can tell Jenkins which branch you should build. You can leave it as default. If your all code is in the master branch.

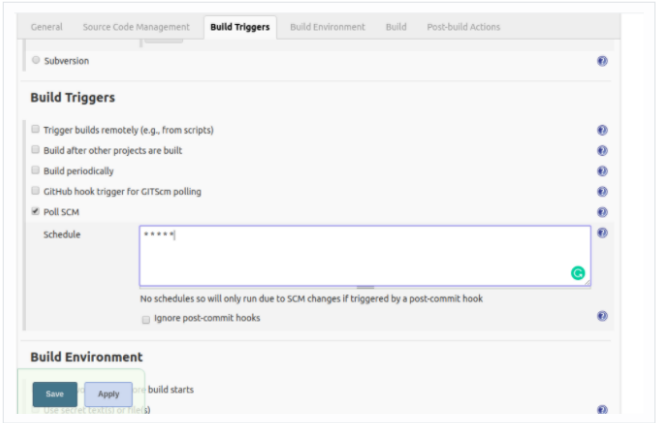
⚠ **Avoid:**

- Be sure that you have git installed on the machine that you are running Jenkins and your git repository is public.



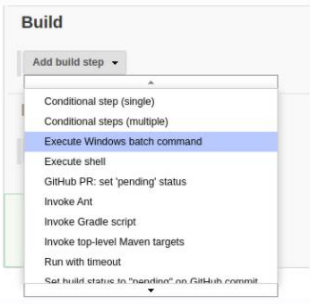
Select git

10. Then go to Build Triggers and select Poll SCM. You can give instructions that how often and what intervals you want to check the source code repository changes. Based on your requirements, you can give different expressions. For this example, we will check it every minute with "* * * * *" expression.



Poll SCM

11. You can run your code from **build**. But first you should add your build step. If you are Windows user you can select **Execute Windows batch command**. But if you are using Jenkins from your MacOS or Linux (maybe ec2 Linux machine) you can select **Execute shell**.

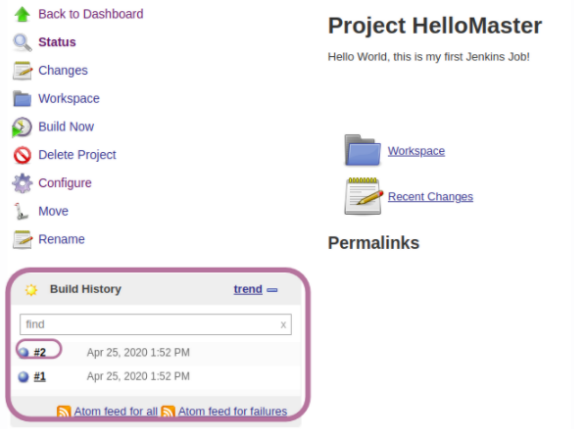


Build Steps

12. Now we will write our build command. In our example we have simple python code at GitHub and we will call it. Then **Apply** and **Save**

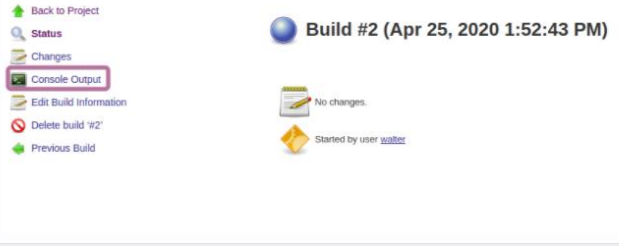
Build Steps

13. After you saved the project you will see the project menu. From project menu click on **Build Now**. A few seconds later you will see **Build History**. Click on the last build number.



Build History

14. You can see your Build's result from **console output**. In our example we are printing "Hello from the other side!" with python code.

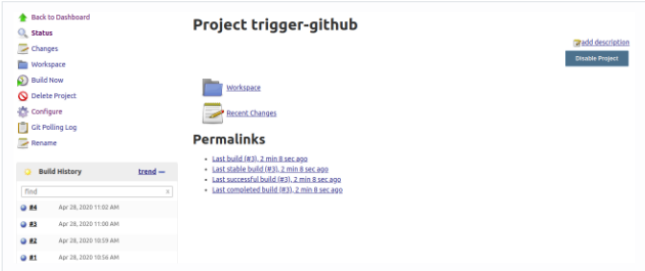


Click on Console Output

So our output will be like:

Output

15. Now please go to your GitHub account and change your code. In our example, we changed the inside of the print function from Github or you can change your code from your own machine and push it, it's up to you. Our new code should print "Hello Jenkins Learner!". When we make our change Jenkins will understand the changes in a minute and build it itself. Because we set **Poll SCM** while creating the item. You can see new build number from your build history.



Jenkins understood GitHub changes in a minutes and re-build

15. When you go to last build's console output you can see your changes result.

You can see "Hello Jenkins Learner!" from output