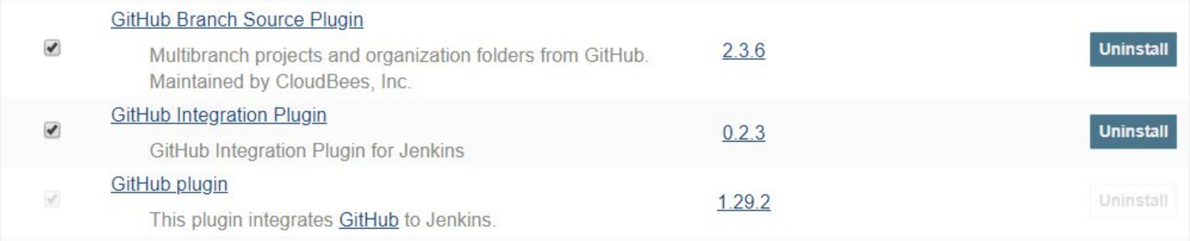
Integrating Jenkins with Github

We first want to trigger a Jenkins build from a push to Github. We configure a Jenkins build that will be initiated on every push made to the repository.

First, we Install the Github Integration Plugin

I already have it installed as seen below:



☑	GitHub Branch Source Plugin Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.3.6	Uninstall
☑	GitHub Integration Plugin GitHub Integration Plugin for Jenkins	0.2.3	Uninstall
☑	GitHub plugin This plugin integrates GitHub to Jenkins.	1.29.2	Uninstall

Now, we prepare our Github repository.

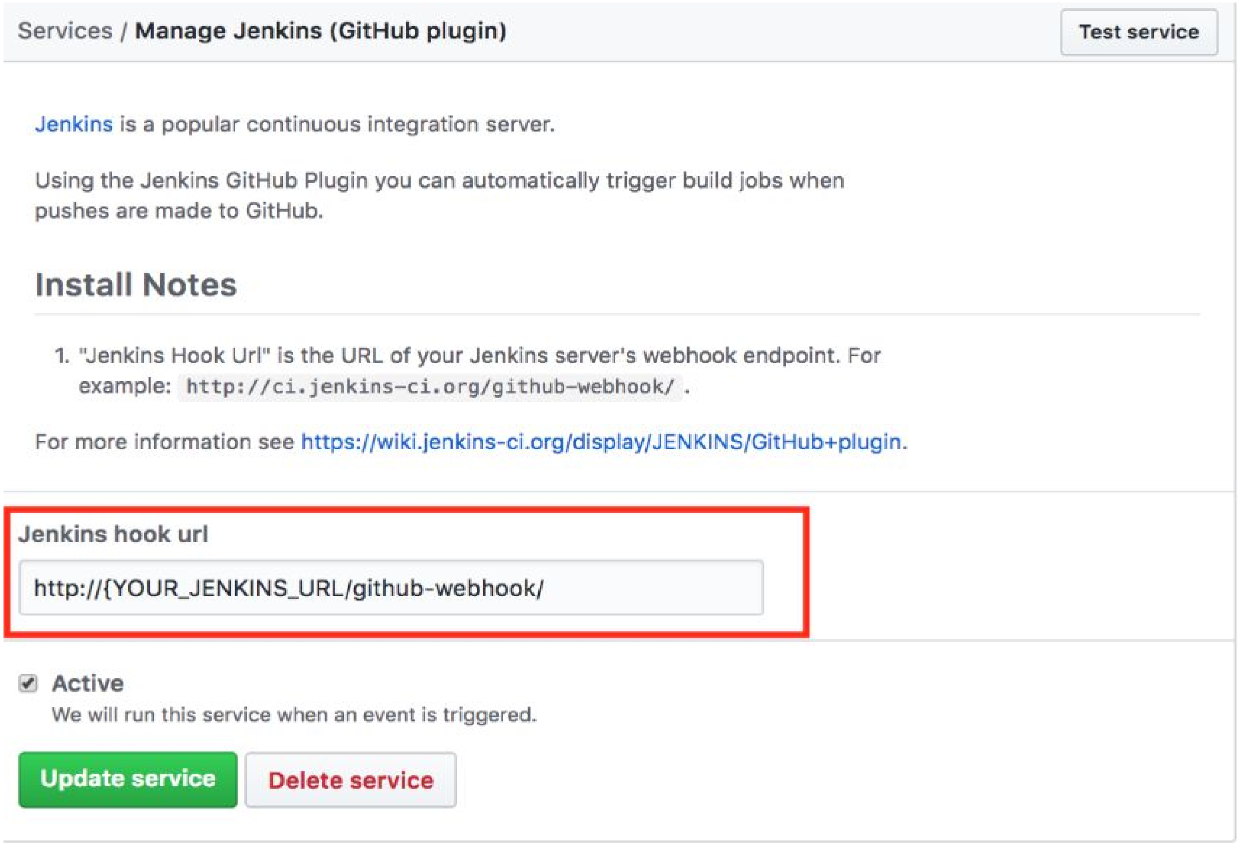
We add a service so that on every push, Jenkins Github Webhook is called.

From your Github repository, go to Settings and select ‘Integrations & Services’

Add a new service

The Jenkins Github Plugin will be available in the list of services

You have to enter your Jenkins instance URL followed by ‘/github-webhook/’



Services / **Manage Jenkins (GitHub plugin)**

Jenkins is a popular continuous integration server.

Using the Jenkins GitHub Plugin you can automatically trigger build jobs when pushes are made to GitHub.

Install Notes

1. "Jenkins Hook Url" is the URL of your Jenkins server's webhook endpoint. For example: `http://ci.jenkins-ci.org/github-webhook/` .

For more information see https://wiki.jenkins-ci.org/display/JENKINS/GitHub+plugin.

Jenkins hook url

http://{YOUR_JENKINS_URL/github-webhook/

☑ **Active**
We will run this service when an event is triggered.

Now, grant the Jenkins user an access to the Github repository by adding a deploy key from Github settings.

If the SSH Keys for the Jenkins user do not exist already, we will generate them

```
jenkins@ip:/home/ubuntu$ ssh-keygen
```

Copy the public key to the location where the key was created

```
jenkins@ip:/home/ubuntu$ cat /var/lib/jenkins/.ssh/id_rsa.pub
```

Now add the copied key to Github from Repository Settings>Deploy Keys

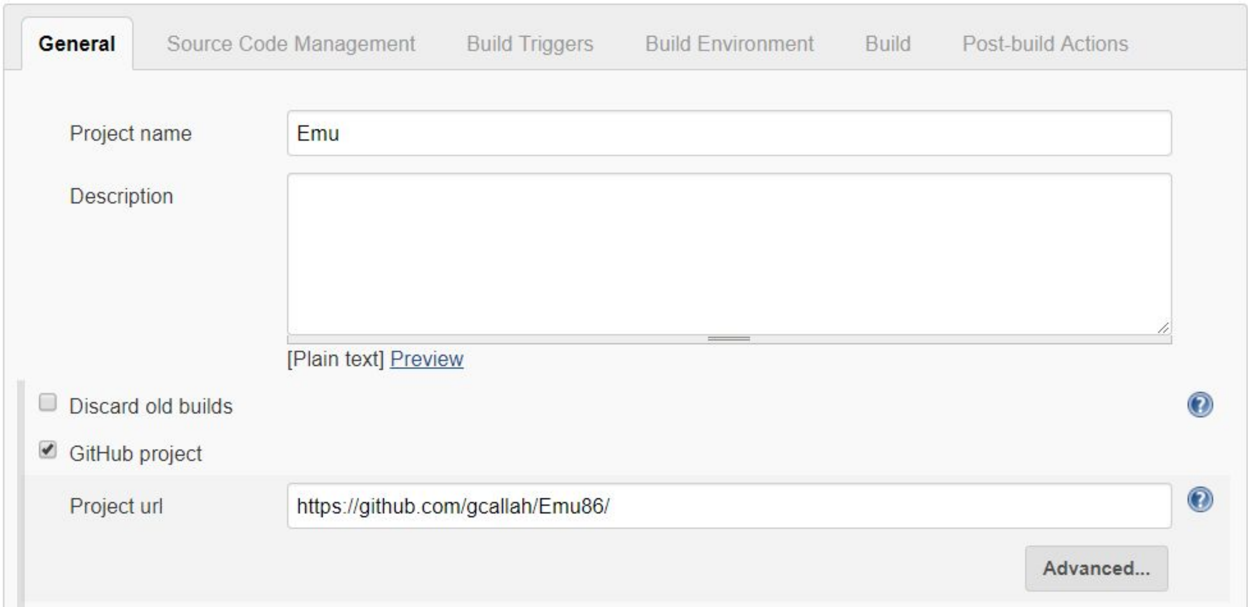
Finally, to check if everything is working as per our expectations, we run

```
jenkins@ip:~/.ssh$ ssh git@github.com
```

Now, we enter the Github configuration for the Jenkins job

In ‘Job Configuration’, go to the ‘General’ section and select ‘Github’.

Enter the URL of the repository in the text box



Project name: Emu

Description:

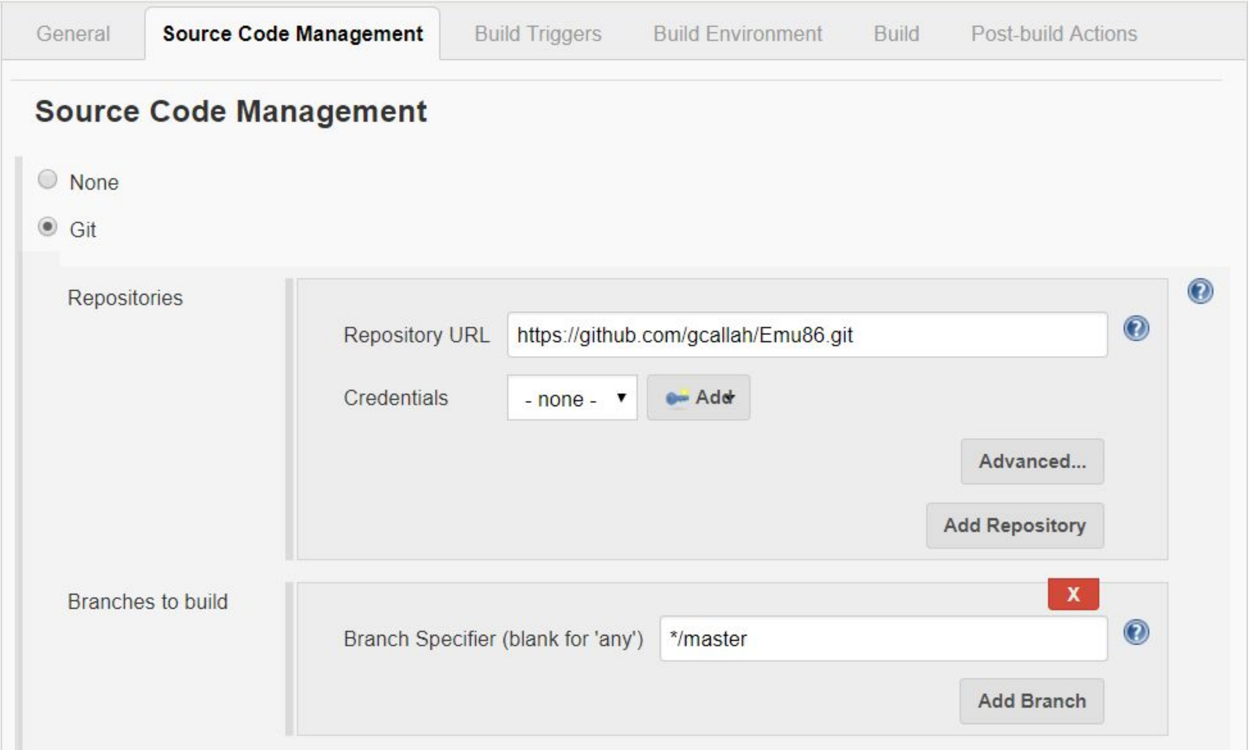
[Plain text] Preview

☐ Discard old builds

☑ GitHub project

Project url: https://github.com/gcallah/Emu86/

Now,set the priority of the repository URL under Source Code Management



Source Code Management

○ None

● Git

Repositories

Repository URL: https://github.com/gcallah/Emu86.git

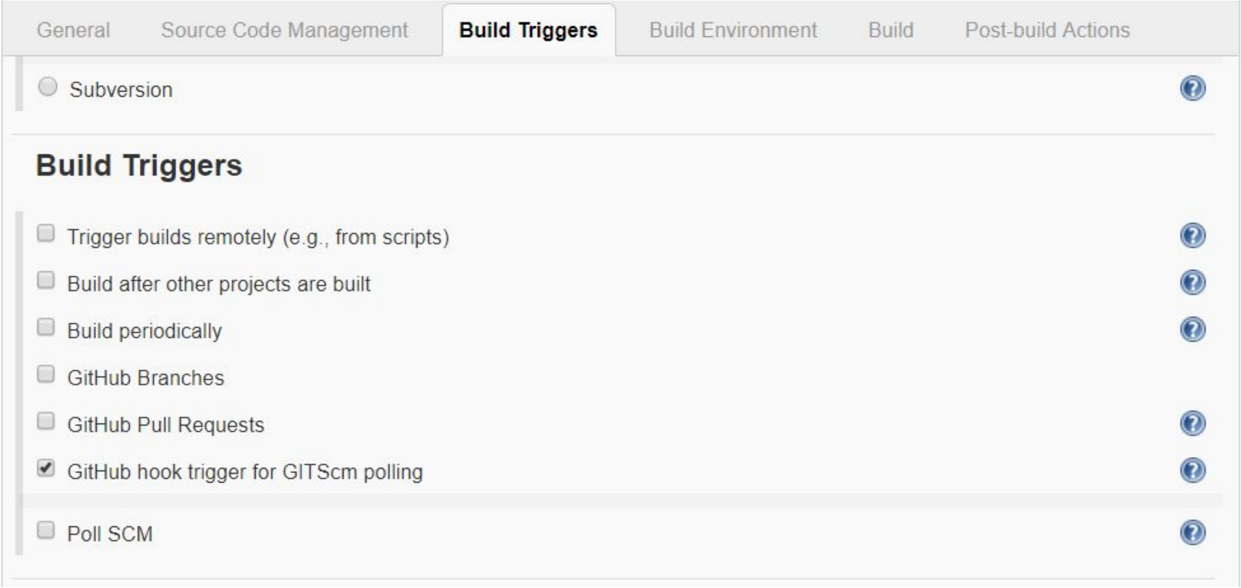
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

Finally, we tell Jenkins to build every time the Github hook is called

Check the appropriate option from Build Triggers to do this



○ Subversion

Build Triggers

□ Trigger builds remotely (e.g., from scripts)
□ Build after other projects are built
□ Build periodically
□ GitHub Branches
□ GitHub Pull Requests
☑ GitHub hook trigger for GITScm polling
□ Poll SCM

We are done! Every time there is a push to the Github repository, there will be a build on Jenkins.

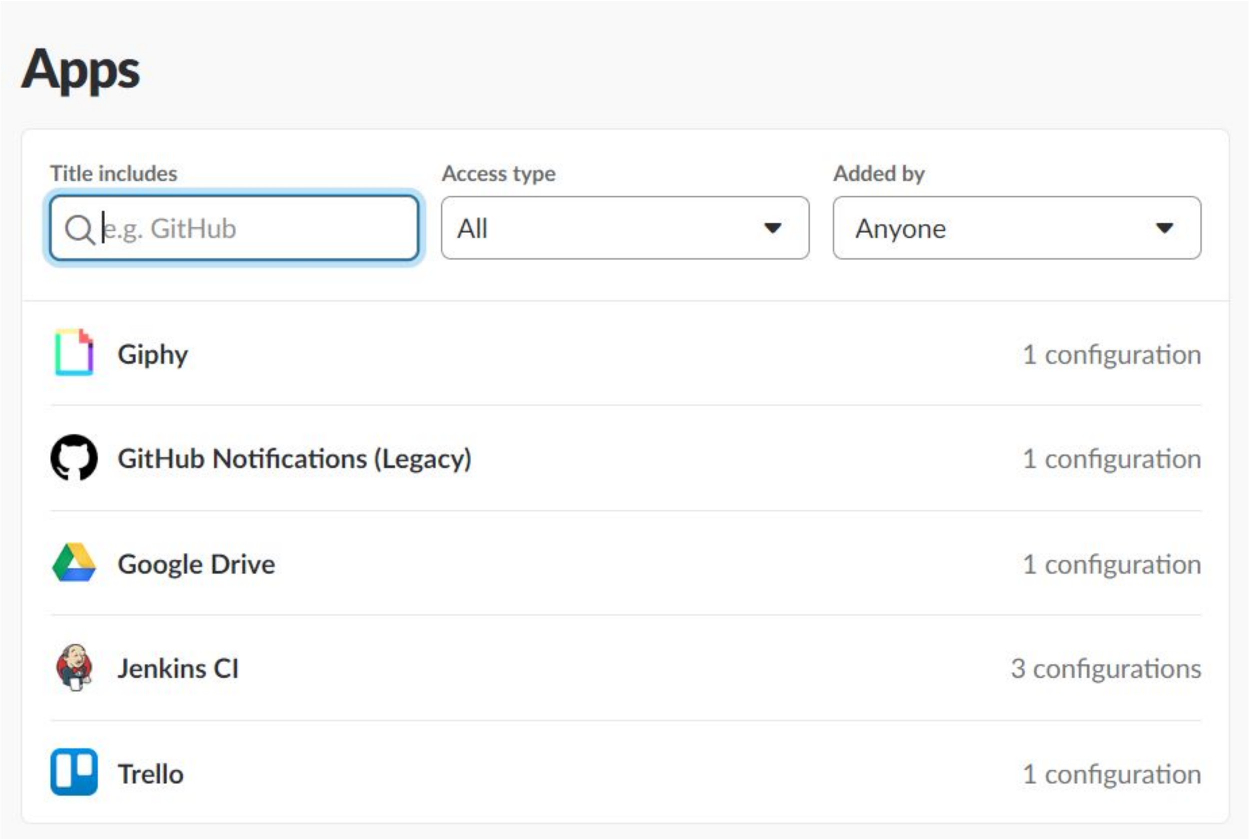
Sending Slack notifications for Jenkins build

To implement this,we need to install Slack Notification and Git plugins

Now, from the Slack workspace, we will install the Jenkins app

This can be done by going to Administration>Manage Apps and then selecting ‘Install’

We have already installed it as seen below



Apps

Title includes: e.g. GitHub

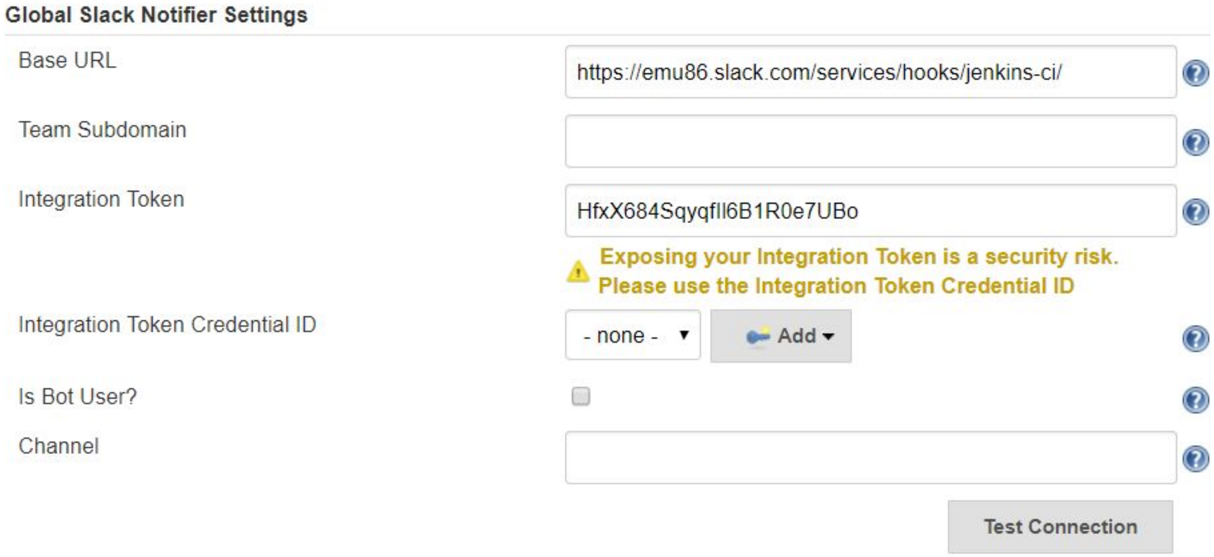
Access type: All

Added by: Anyone

Giphy	1 configuration
GitHub Notifications (Legacy)	1 configuration
Google Drive	1 configuration
Jenkins CI	3 configurations
Trello	1 configuration

Now, select the channel to which you want to send the notifications

Next, from Manage Jenkins>Configure System>Global Slack Notifier Settings, configure the plugin as shown below



Global Slack Notifier Settings

Base URL: https://emu86.slack.com/services/hooks/jenkins-ci/

Team Subdomain:

Integration Token: HfxX684SqyqfIl6B1R0e7UBo

⚠ Exposing your Integration Token is a security risk.
Please use the Integration Token Credential ID

Integration Token Credential ID: - none - ▾

Is Bot User? □

Channel:

Now, we can test our connection by building the system. We should get a Slack notification on occasion of a new build.