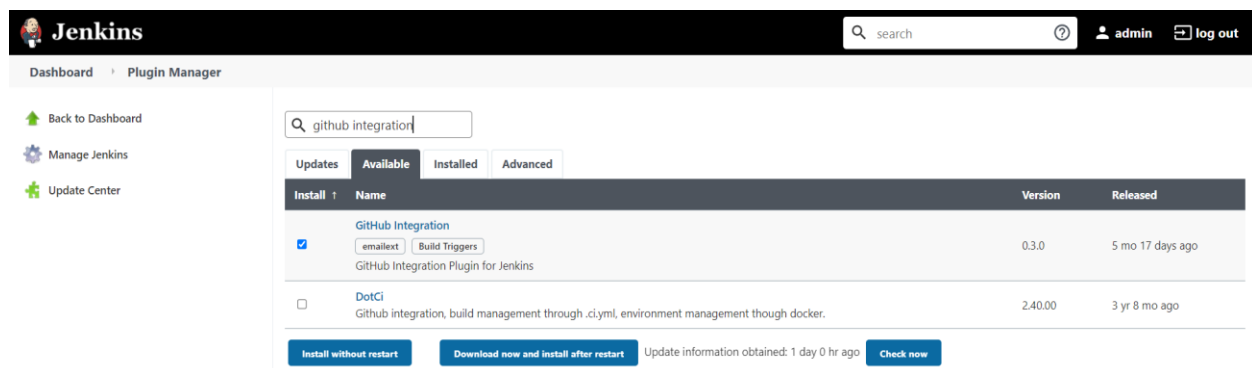


# Triggering a Jenkins build from a push to Github

## 1. Install Github Integration Plugin

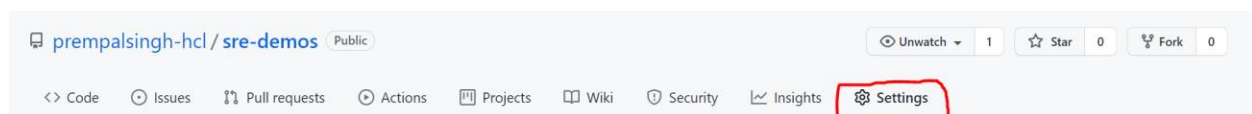
First we need to install the [GitHub Integration Plugin](#), this will give us the ability to configure Jenkins to use our Github repository.

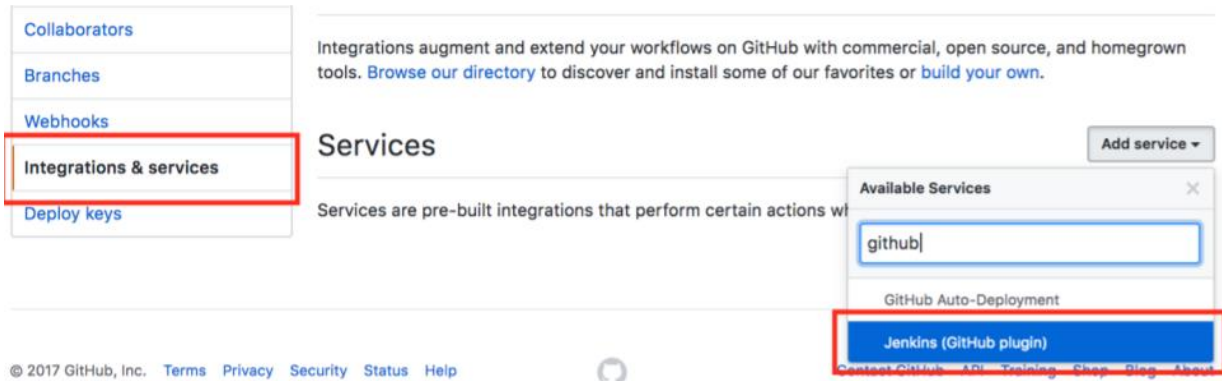


Installing Github integration plugin

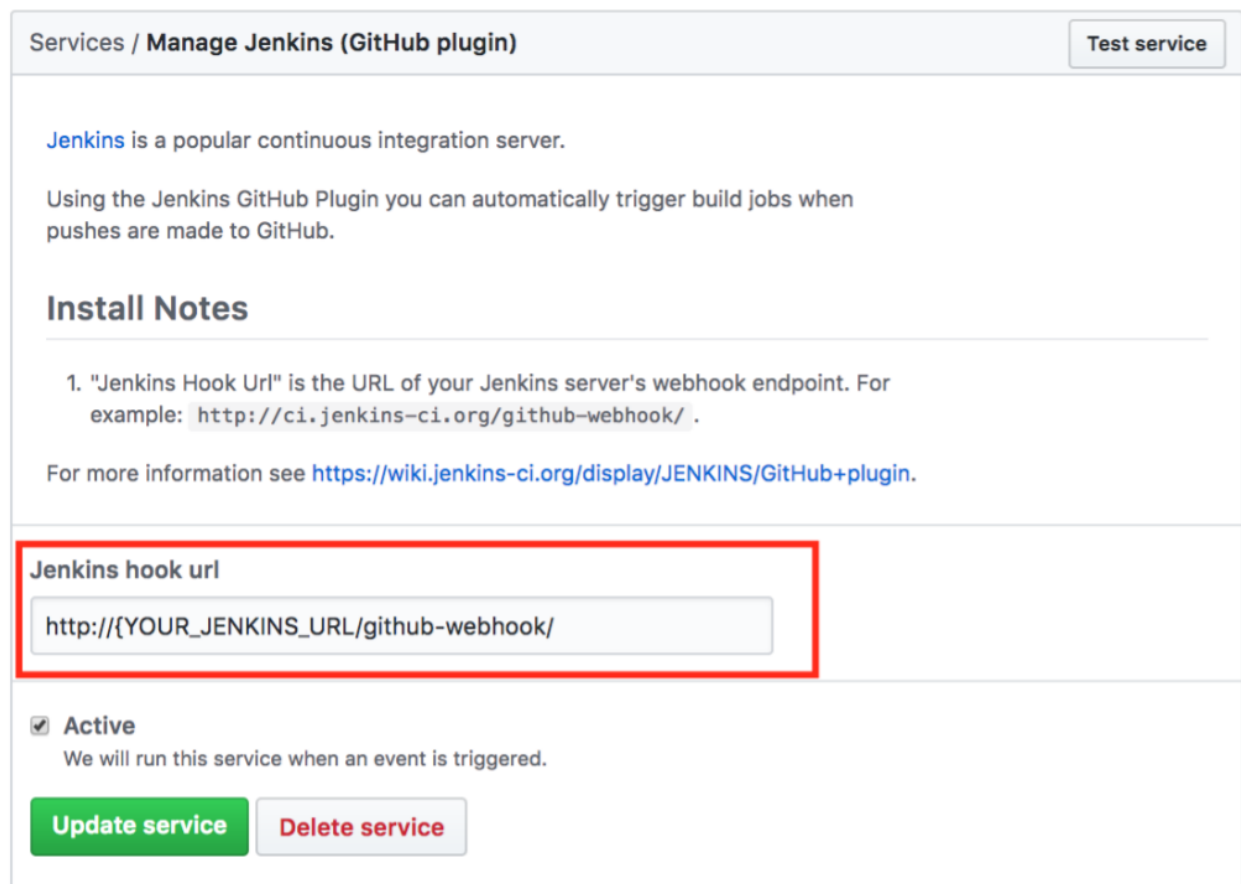
## 2. Prepare Github repository

We need to add a service to call the Jenkins Github webhook on a push, to do this go to settings -> integrations & Services and add a new service. The Jenkins Github plugin service should be in the list of available services.





Enter the URL of your Jenkins instance followed by `/github-webhook/`



### 3. Giving the Jenkins user access to the Github repository

We need to give the Jenkins user access to our repository by adding a deploy key in the Github settings.

The first step is generating SSH keys for the Jenkins user if they do not already exist

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

Depending on where the key was created, you need to copy the public key so that it can be added to Github

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

Options  
Collaborators  
Branches  
Webhooks  
Integrations & services  
Deploy keys

### Deploy keys

Add deploy key

There are no deploy keys for this repository

**Title**  
Jenkins

**Key**  
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQCoD1zZ6nhwzXyQF1p7wiHM1dBDAS1IFycfFgCGuh0R4dHGb  
bObfvi3wU3ddPk3FNbY5Khzkwc+ZEquJBZvp0wQ/ZU7QtVP8JaQN22sJZ1ijzNcdKu6wS51+qQizKdc+4  
w6nGoSf3L/a2kpg6Yb47YfaiwTPZsS0MOFBCuel9Yw+O9+rverd4U8D3cDr/5yghbJJuvzsrHtvx9VYmRT  
k8nofRFom2IDbALyk4cvOAtI7ofz7UVE0dkqcufowndDa8FrDqPr/luM8EcnUEhCo6fnALelxdPegvr0OMw  
LnUkpxQIBDalqfaRL2DApH5fpqf0jXiZs2vCsYCTDJ71b0p/p jenkins@ip-172-31-22-109

☐ Allow write access  
Can this key be used to push to this repository? Deploy keys always have pull access.

Add key

The last step is to check that everything is working as expected, as the Jenkins user in your console enter the below to check the connection to Github.

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

If successful you should see the following message

```
Warning: Permanently added the RSA host key for IP address  
'{YOUR_SERVER_IP}' to the list of known hosts.
```

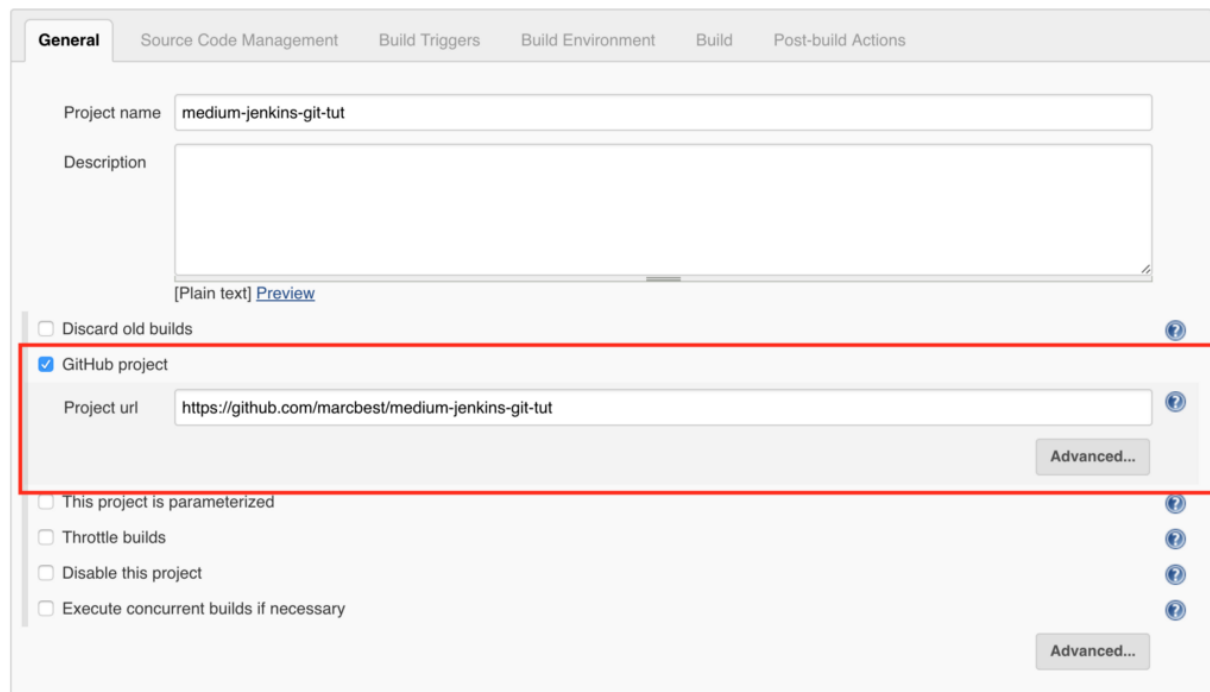
```
PTY allocation request failed on channel 0
```

```
Hi You've successfully authenticated, but GitHub does not provide  
shell access.
```

```
Connection to github.com closed.
```

## 4. Update Jenkins job with Github configuration

In the 'General' section of the job configuration check the Github project tick box and enter the URL to the repository that you configured in step 2.



The screenshot shows the Jenkins job configuration interface with the 'General' tab selected. The 'Project name' field is 'medium-jenkins-git-tut'. The 'Description' field is empty. The 'Discard old builds' checkbox is unchecked. The 'GitHub project' checkbox is checked and highlighted with a red box. The 'Project url' field is 'https://github.com/marcbest/medium-jenkins-git-tut'. Below this, there are several other checkboxes: 'This project is parameterized', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary', all of which are unchecked. There are 'Advanced...' buttons on the right side of the 'GitHub project' and the bottom section.

Next update the Source Code Management section, first set the repository URL (note the format `git@github.com:{YOUR_REPO}`). You can also specify the branch you would like to use.

**Source Code Management**

☐ None  
☒ Git

**Repositories**

Repository URL

Credentials

**Branches to build**

Branch Specifier (blank for 'any')

The last step is to tell Jenkins to build when the Github hook is called, select the highlighted option below in the Build Triggers section.

**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