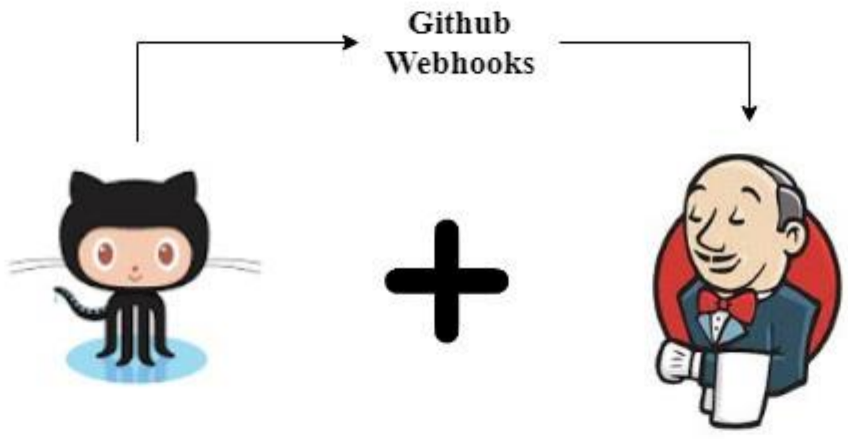
Triggering a Jenkins build on push using GitHub webhooks



d

Jenkins Github Integration

*J**enkins** is a popular open source tool to perform continuous integration and build automation. Let's take a look at how we can integrate **GitHub** with Jenkins for **source code management** and **trigger build on push using web-hooks**.*

Prerequisites :

1. **Jenkins**: Download and install Jenkins as described *here*.

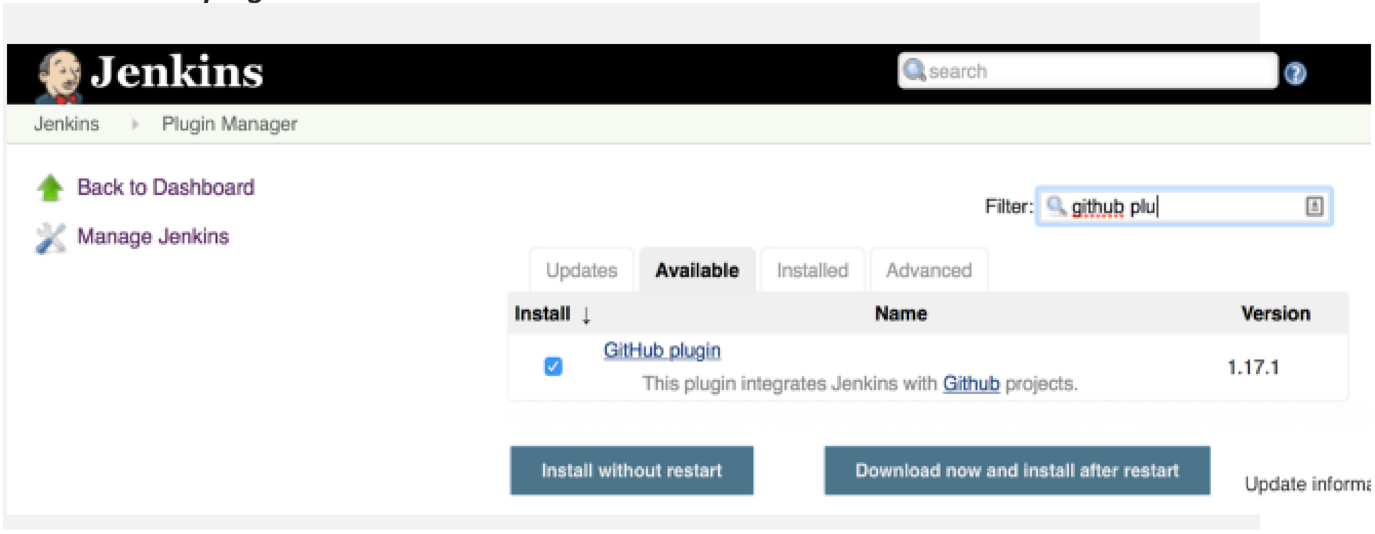
2. **Git**: Install Git. To check whether you have git installed, open a terminal window and type below command.

   ```
   git --version
   ```

3. **Plugins:** Add Git and GitHub Plugins.

Go to Manage-Jenkins-> Manage Plugin

search ***Github plugin*** and install without restart



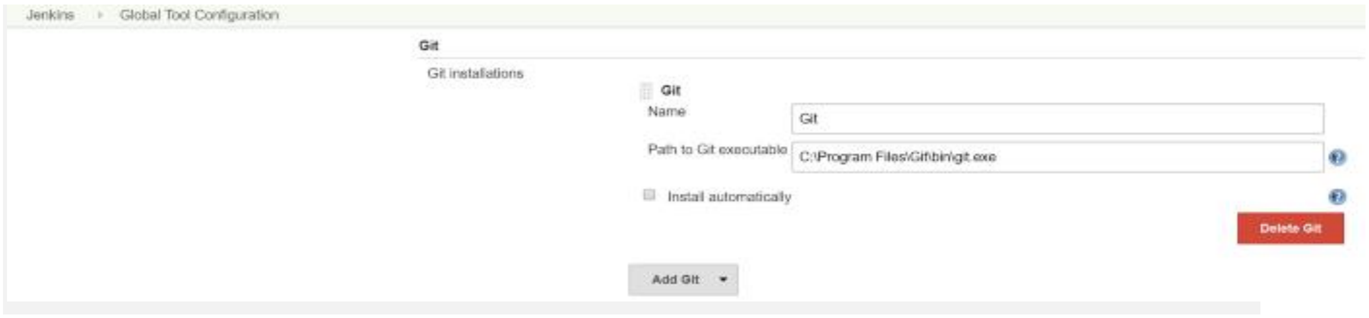
Filter: github plu

Install ↓	Name	Version
✓	GitHub plugin This plugin integrates Jenkins with Github projects.	1.17.1

Plugin Manager

4. Go to Manage Jenkins -> Global Tool Configuration -> Git

Add git executable path to Global Tool Configuration.



Name: Git

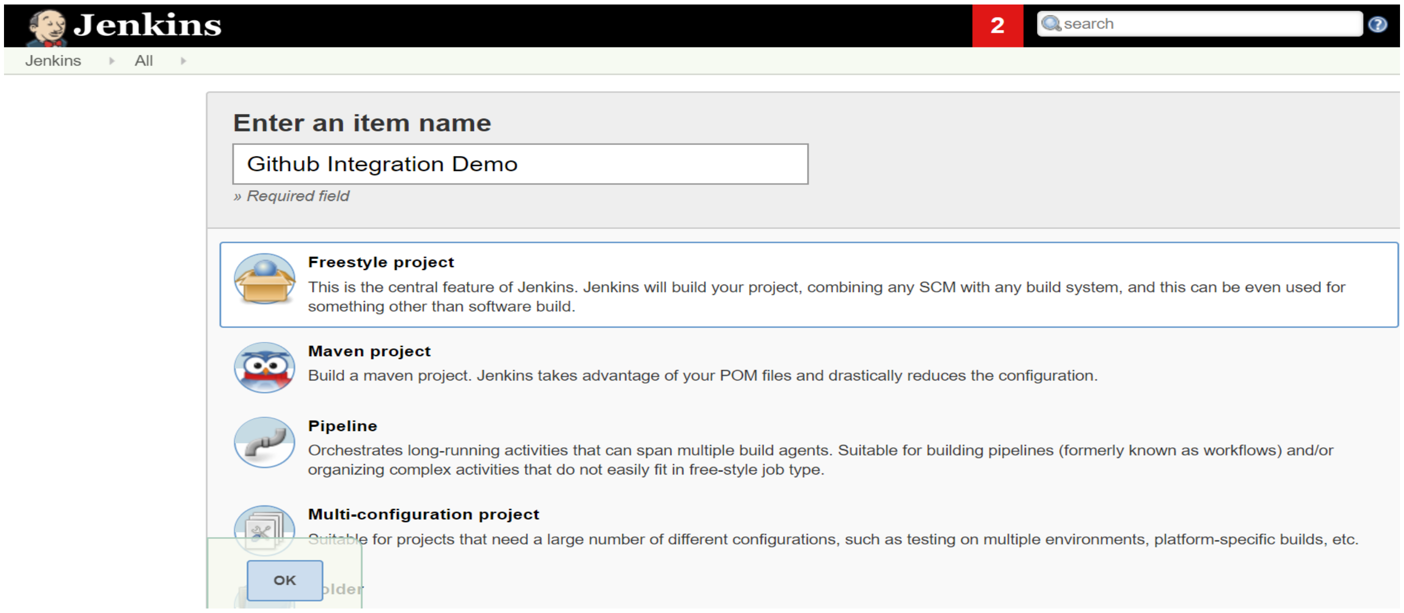
Path to Git executable: C:\Program Files\Git\bin\git.exe

Install automatically: ☐

Global tool configurations

Let us start with creating a Freestyle Project :

Step 1: Go to New Item -> create a freestyle project.



Enter an item name

Github Integration Demo

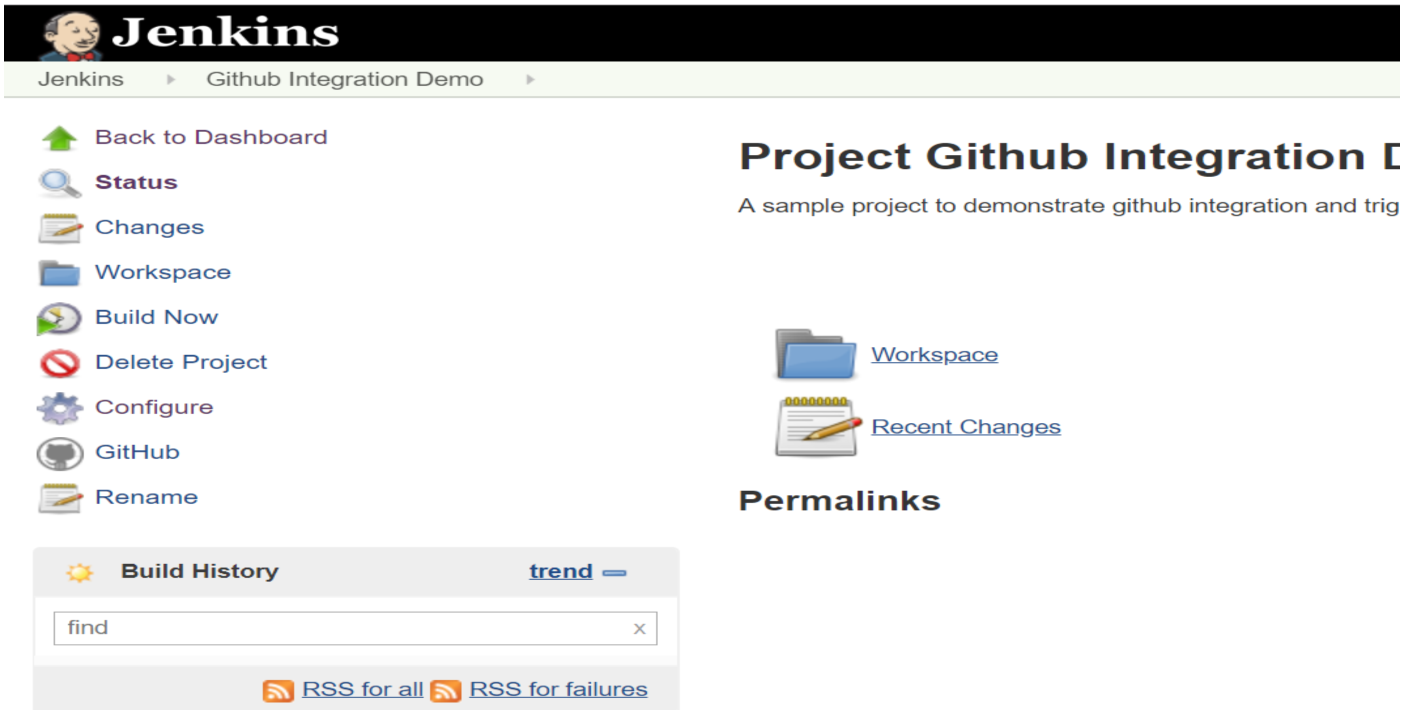
» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Project Github Integration D

A sample project to demonstrate github integration and trig

Permalinks

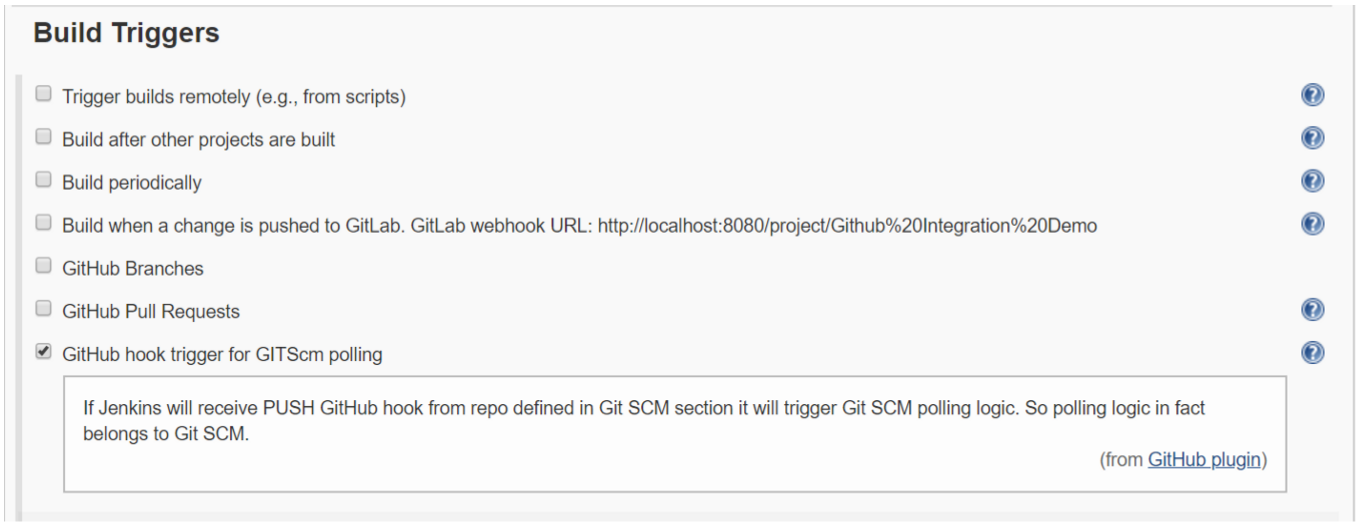
Creating a freestyle project

Step 2: Go to **Configure,** add a project description, and Github project URL.

Step 3: In **Source Code Management** tab select on **Git**, add your Github repository URL and click Add button to save your GitHub credentials.

Save and click build to make sure everything is right till here and your project is successfully building.

step 4: In **Build Triggers** select **GitHub hook trigger for GITScm polling.** When Jenkins will receive PUSH GitHub hook, it will trigger Git SCM polling logic which will start a new Jenkins build, with the updated code.



Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] Build when a change is pushed to GitLab. GitLab webhook URL: http://localhost:8080/project/Github%20Integration%20Demo
- [ ] GitHub Branches
- [ ] GitHub Pull Requests
- [✓] GitHub hook trigger for GITScm polling

If Jenkins will receive PUSH GitHub hook from repo defined in Git SCM section it will trigger Git SCM polling logic. So polling logic in fact belongs to Git SCM.

(from GitHub plugin)

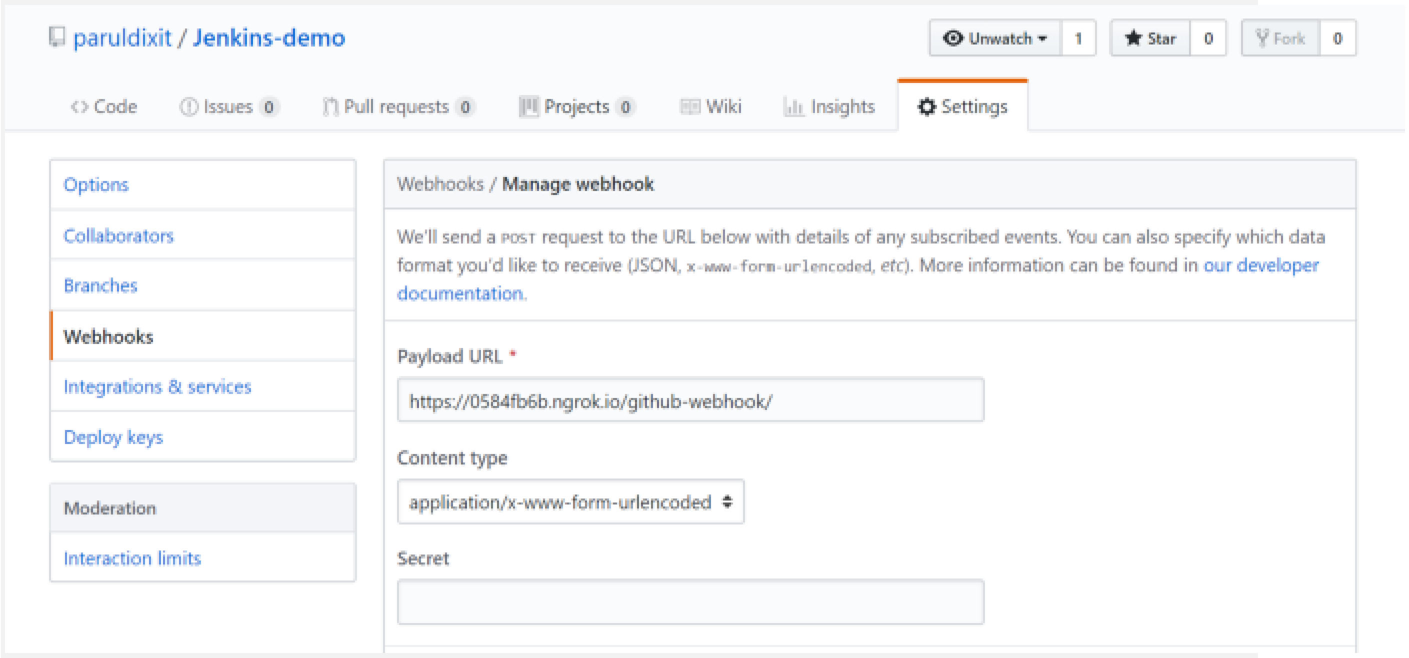
Build Triggers

We use ***ngrok*** *to expose local jenkins to the internet, so that github can send the webhooks.*

Step 5: Go to your Github repo -> settings -> webhooks

Add public URL of your tunnel as **Payload URL,** it will tell Github where to send the webhooks as below:

https://0584fb6b.ngrok.io/github-webhook/



We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *
https://0584fb6b.ngrok.io/github-webhook/

Content type
application/x-www-form-urlencoded

Secret

step 6: Finally add your build steps in the **Build** tab and save.

That’s all! Now whenever any change is pushed, a new Jenkins build will be triggered.