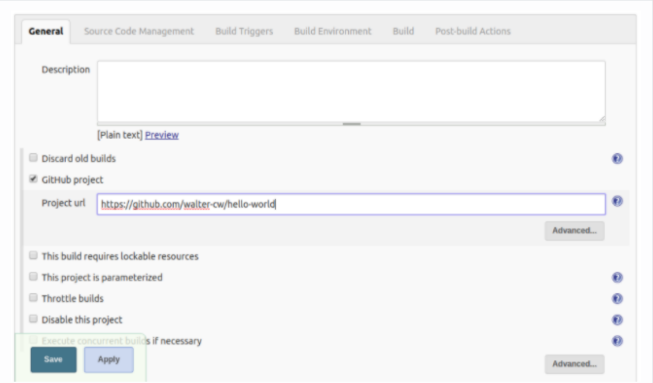
Integrate with GitHub Webhook

Webhook allow external services to be notified when certain events happen within your repository. When the specified events happen we'll send a POST request to each of the URLs you provide.

We will create a new item and then use our own GitHub codes to trigger the build. We will see that when we change our GitHub codes, the build will automatically start without waiting. When our code changes, GitHub's Webhook will tell Jenkins that the code is changed, so the build will occur immediately.

1. Create a new item

2. You can enter the GitHub project.



Description

[Plain text] Preview

□ Discard old builds

☑ GitHub project

Project url: https://github.com/walter-cw/hello-world

□ This build requires lockable resources

□ This project is parameterized

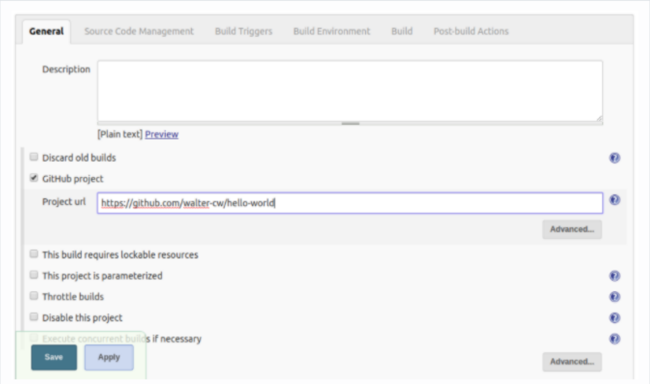
□ Throttle builds

□ Disable this project

□ Execute concurrent builds if necessary

Enter Github

3. Check the GitHub Project from **Source Control Management** then enter your GitHub repository URL.



Description

[Plain text] Preview

□ Discard old builds

☑ GitHub project

Project url: https://github.com/walter-cw/hello-world

□ This build requires lockable resources

□ This project is parameterized

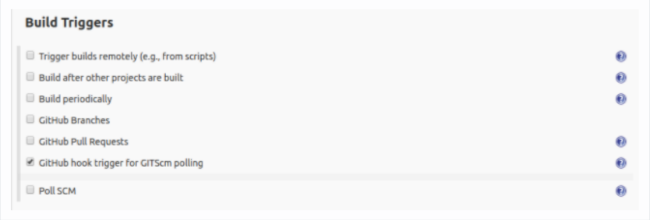
□ Throttle builds

□ Disable this project

□ Execute concurrent builds if necessary

Github Repo Source

4. Then from Build Triggers click on **GitHub hook trigger for GITScm polling**



Build Triggers

□ Trigger builds remotely (e.g., from scripts)

□ Build after other projects are built

□ Build periodically

□ GitHub Branches

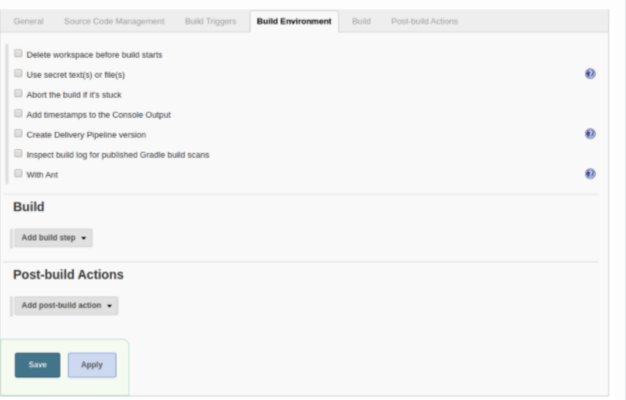
□ GitHub Pull Requests

☑ GitHub hook trigger for GITScm polling

□ Poll SCM

GitHub hook trigger

4. You can add build steps. But for this example, we want to show you how to trigger GitHub Webhook. So we will not add build command. After that please click on **apply** and **save**



□ Delete workspace before build starts

□ Use secret text(s) or file(s)

□ Abort the build if it's stuck

□ Add timestamps to the Console Output

□ Create Delivery Pipeline version

□ Inspect build log for published Gradle build scans

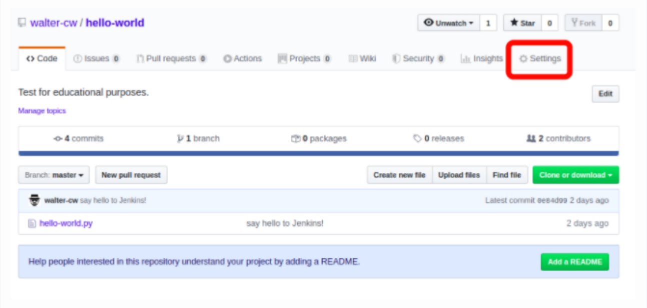
□ With Ant

Build

Post-build Actions

Apply and Save

5. Go to your **GitHub repository page** then click on **Settings**.



Test for educational purposes.

4 commits | 1 branch | 0 packages | 0 releases | 2 contributors

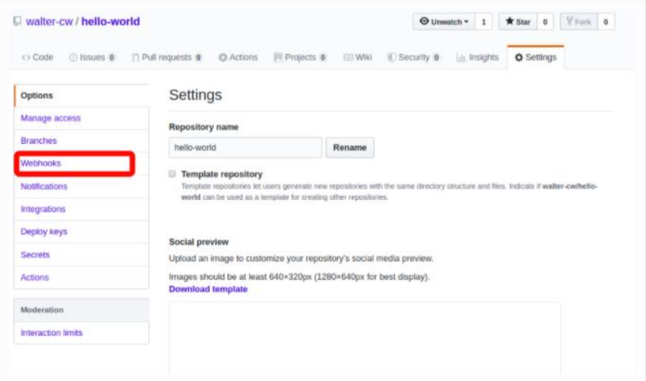
walter-cw say hello to Jenkins! Latest commit 8e84d99 2 days ago

hello-world.py say hello to Jenkins! 2 days ago

Help people interested in this repository understand your project by adding a README.

Github Settings

6. You will see **Webhooks** from settings menu. Click on Webhooks.



Settings

Repository name: hello-world

□ Template repository

Template repositories let users generate new repositories with the same directory structure and files. Indicate if **walter-cw/hello-world** can be used as a template for creating other repositories.

Social preview

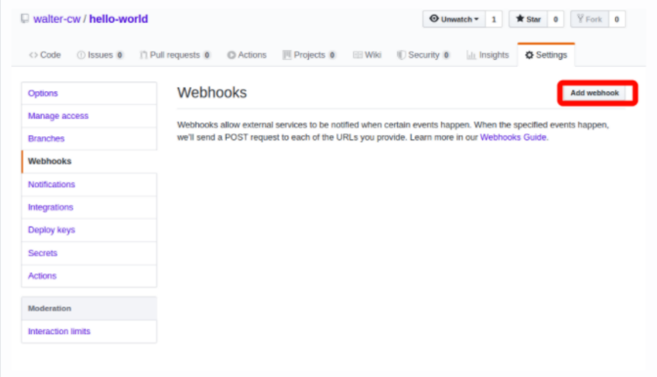
Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Download template

Webhooks

7. Click on **Add webhook**.

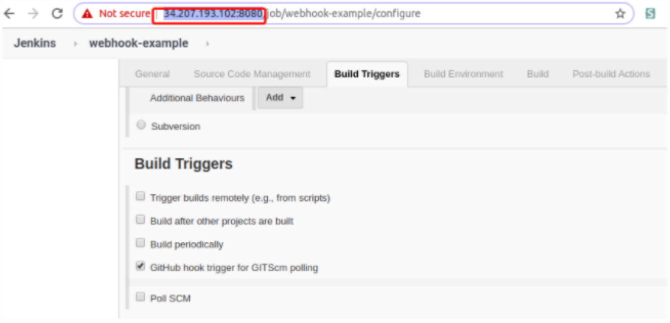


Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

Add Webhooks

8. Copy to your Jenkins URL and paste into **Payload URL** section. Be careful, you can't use localhost:8080. But you can use your EC2 machine's Jenkins URL. You must add **/github-webhook/** at the end of your URL. Then click on **Add Webhook**



Not secure | 34.207.193.102:8080/job/webhook-example/configure

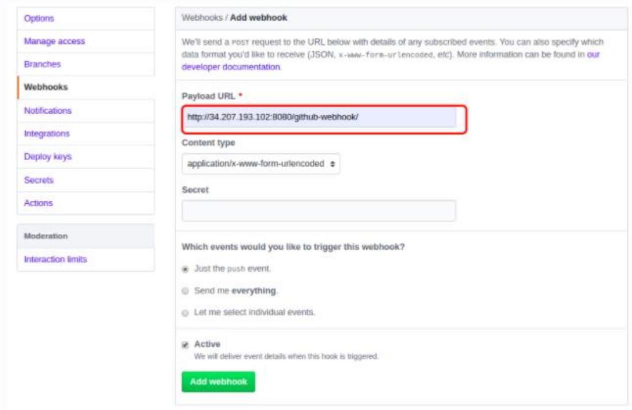
Jenkins › webhook-example ›

○ Subversion

Build Triggers

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

Take Jenkins URL



Webhooks / **Add 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 *

http://34.207.193.102:8080/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

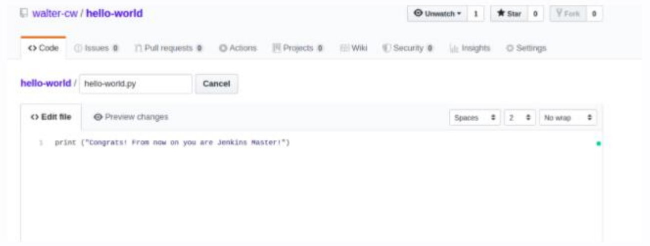
Which events would you like to trigger this webhook?

- ● Just the `push` event.
- ○ Send me **everything**.
- ○ Let me select individual events.

☑ **Active**
We will deliver event details when this hook is triggered.

Write URL to Payload URL

9. Go back to your repository and make changes at your code (make commit).



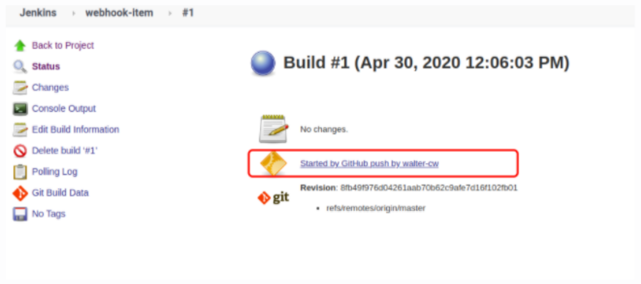
walter-cw / **hello-world**

hello-world / hello-world.py

```
print ("Congrats! From now on you are Jenkins Master!")
```

Edit your code

10. When you change your repository. Github webhook understands the changes and immediately tells your Jenkins to update the project. Then Jenkins will automatically start a new build. You will see that the build is started by GitHub.



Jenkins › webhook-item › #1

Build #1 (Apr 30, 2020 12:06:03 PM)

No changes.

Started by GitHub push by walter-cw

Revision: 8fb49f976d04261aab70b62c9afe7d16f102fb01

- refs/remotes/origin/master

New Build Console Output

Complementary Lesson about Build Your First Job

Integrate with GitHub: build after each commit (Get started with Jenkins, part 13)