

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.

Welcome to Jenkins!

Username

Password

Sign in

☐ Keep me signed in

Login Page

2. Click on "Manage Jenkins"

Jenkins

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

Credentials

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Manage Plugins

3. Manage plugins

Manage Jenkins

New version of Jenkins (2.222.3) is available for [download](#) ([changelog](#)).

Configure System

Configure global settings and paths.

Configure Global Security

Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials

Configure the credential providers and types

Global Tool Configuration

Configure tools, their locations and automatic installers.

Reload Configuration from Disk

Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
There are updates available

Manage Plugins

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

UpdatesAvailableInstalledAdvanced

Install

	Name	Version	Released
<input type="checkbox"/>	GitHub Pipeline for Blue Ocean External Site/Tool Integrations User Interface BlueOcean GitHub organization pipeline creator	1.23.0	6 days 14 hr ago
<input type="checkbox"/>	DotCi GitHub integration, build management through .ci.yml, environment management through docker.	2.40.00	2 yr 2 mo ago
<input type="checkbox"/>	GitBucket Source Code Management Build Triggers This plugin integrates GitBucket to Jenkins.	0.8	4 yr 7 mo ago
<input checked="" type="checkbox"/>	GitHub Integration emailxst Build Triggers GitHub Integration Plugin for Jenkins	0.2.8	1 mo 24 days ago
<input type="checkbox"/>	GitHub SOS Build Trigger aws External Site/Tool Integrations github Build Triggers This plugin allows you to trigger builds using Amazons Simple Queue Service.	2.1	2 yr 3 mo ago
<input type="checkbox"/>	Kubernetes agent Cloud Providers Cluster Management and Distributed Build kubernetes	1.25.3	9 days 21 hr ago

Install without restart

Download now and install after restart

Update information obtained: 9 min 46 sec ago

Check now

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.

Filter: git plugin

UpdatesAvailableInstalledAdvanced

Enabled

	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Credentials Plugin This plugin allows you to store credentials in Jenkins.	2.3.7		Uninstall
<input checked="" type="checkbox"/>	Git client plugin Utility plugin for Git support in Jenkins	3.2.1		Uninstall
<input checked="" type="checkbox"/>	Git plugin This plugin integrates Git with Jenkins.	4.2.2		Uninstall
<input checked="" type="checkbox"/>	GitHub Branch Source Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.	2.6.0		Uninstall
<input checked="" type="checkbox"/>	GitHub Integration Plugin GitHub Integration Plugin for Jenkins	0.2.8		Uninstall
<input checked="" type="checkbox"/>	GitHub plugin This plugin integrates GitHub to Jenkins.	1.29.5		Uninstall
<input checked="" type="checkbox"/>	Mailer Plugin This plugin allows you to configure email notifications for build results	1.32		Uninstall
<input checked="" type="checkbox"/>	Pipeline: GitHub Groovy Libraries Allows Pipeline Groovy libraries to be loaded on the fly from GitHub.	1.0		Uninstall
	Pipeline: SCM Step			

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.

GitHub plugin

Success

GitHub Branch Source Plugin

Success

Pipeline: GitHub Groovy Libraries

Success

Pipeline: Stage View Plugin

Success

Git plugin

Success

MapDB API Plugin

Success

Subversion Plug-in

Success

SSH Slaves plugin

Success

Matrix Authorization Strategy Plugin

Success

PAM Authentication plugin

Success

LDAP Plugin

Success

Email Extension Plugin

Success

Mailer Plugin

Success

GitHub Integration Plugin

Pending

[Go back to the top page](#)

(you can start using the installed plugins right away)

☐

Restart Jenkins when installation is complete and no jobs are running

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

Enter an item name

trigger-github

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.

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.

Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder is a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Create a project

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

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Source Code Management

None

Git

Subversion

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.

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Source Code Management

None

Git

Repositories

Repository URL

https://github.com/walter-cwhello-world.git

Credentials

none

Add

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*master

Add Branch

Repository browser

(Auto)

Additional Behaviours

Add

Save

Apply

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.

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

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

Schedule

* * * * *

Build Environment

Save

Apply

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

Add build step

Conditional step (single)

Conditional steps (multiple)

Execute Windows batch command

Execute shell

GitHub PR: set 'pending' status

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

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

Execute shell

Command

python hello-world.py

See the list of available environment variables

Advanced...

Save

Apply

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.

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Move

Rename

Project HelloMaster

Hello World, this is my first Jenkins Job!

Workspace

Recent Changes

Permalinks

Build History

find

X

#2

Apr 25, 2020 1:52 PM

#1

Apr 25, 2020 1:52 PM

Atom feed for all

Atom feed for failures

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.

Back to Project

Status

Changes

Console Output

Edit Build Information

Delete build #2

Previous Build

Build #2 (Apr 25, 2020 1:52:43 PM)

No changes.

Started by user [walter](#)

Click on Console Output

So our output will be like:

Console Output

Started by user [walter](#)
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/trigger-github
No credentials specified

Git Commands

```
git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/walter-cw/hello-world # timeout=10
Fetching upstream changes from https://github.com/walter-cw/hello-world
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/walter-cw/hello-world +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision ea81f704f62aaf176ada18e9f1551e463d32902b (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f ea81f704f62aaf176ada18e9f1551e463d32902b # timeout=10
Commit message: "hello world"
> git rev-list --no-walk ea81f704f62aaf176ada18e9f1551e463d32902b # timeout=10
python hello-world.py
Hello from the other side!
Finished: SUCCESS
```

Python code runs and gives output

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.

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Git Polling Log

Rename

Project trigger-github

Workspace

Recent Changes

Permalinks

Build History

find

X

#4

Apr 28, 2020 11:03 AM

#3

Apr 28, 2020 11:00 AM

#2

Apr 28, 2020 10:59 AM

#1

Apr 28, 2020 10:56 AM

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.

Console Output

Started by an SCM change
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/trigger-github
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/walter-cw/hello-world # timeout=10
Fetching upstream changes from https://github.com/walter-cw/hello-world
> git --version # timeout=10
> git fetch --tags --progress -- https://github.com/walter-cw/hello-world +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 0e84d999f6dcbed54c76d8f69a960fe75aa8ca31 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 0e84d999f6dcbed54c76d8f69a960fe75aa8ca31 # timeout=10
Commit message: "say hello to Jenkins!"
> git rev-list --no-walk ea81f704f62aaf176ada18e9f1551e463d32902b # timeout=10
[trigger-github] \$ /bin/sh -xe /tmp/jenkins6677226394732053284.sh
+ python hello-world.py
Hello Jenkins Learner!
Finished: SUCCESS

You can see "Hello Jenkins Learner!" from output

Jenkins understood GitHub changes in a minutes and re-build