July 21, 2022

GitHub + Jenkins: How to Integrate Your GitHub Repository to Your Jenkins Project

OPEN SOURCE AUTOMATION

By Guy Salton

SEND FEEDBACK

Jenkins and GitHub are two powerful tools on their own, but what about using them together? In this blog, learn about the benefits of a Jenkins integration with GitHub and how to set up the integration on your own for GitHub Jenkins harmony.

Table of Contents

Table of Contents

Back to top

Can You Use GitHub + Jenkins?

You can and should use Jenkins with GitHub to save time and keep your project up-to-date.

One of the basic steps of implementing CI/CD is integrating your SCM (Source Control Management) (https://www.blazemeter.com/blog/scm-version-control) tool with your CI tool. This saves you time and keeps your project updated all the time. One of the most popular and valuable SCM tools is GitHub.

📕 ***Related Reading:*** ***Explore Five Advantages Your Peers are Discovering with Continuous Testing >>*** (https://www.blazemeter.com/resources/advantages-continuous-testing-peers)

What is GitHub?

GitHub is a Git-based repository host, commonly used for open-source projects. GitHub enables code collaboration, hosting, and versioning.

What is Jenkins?

Jenkins is an open-source Continuous Integration and Continuous Deployment (CI/CD) tool for automating the software development life cycle (SDLC). With Jenkins testing (https://www.blazemeter.com/solutions/jenkins), teams can automate the

SEND FEEDBACK

building, testing, and deploying of code.

Back to top

Why Do a GitHub-Jenkins Integration?

A Jenkins integration with GitHub will improve the efficiency of building, testing, and deploying your code.

The integration presented in this blog post will teach you how to schedule your build, pull your code and data files from your GitHub repository to your Jenkins machine, and automatically trigger each build on the Jenkins server after each Commit on your Git repository.

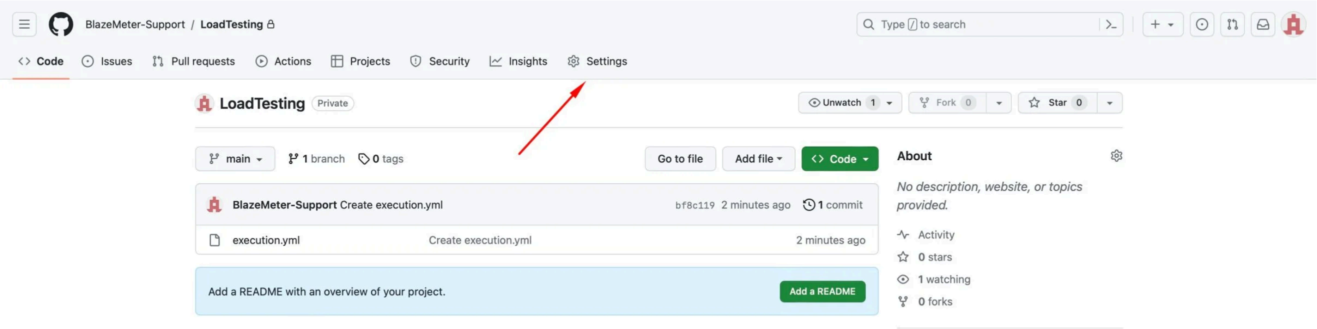
But first, let’s configure the Jenkins and GitHub integration. Let's begin with the GitHub side!

Back to top

How to Set Up the Jenkins + GitHub Integration

Configuring GitHub

Step 1: go to your GitHub repository and click on **‘Settings’.**

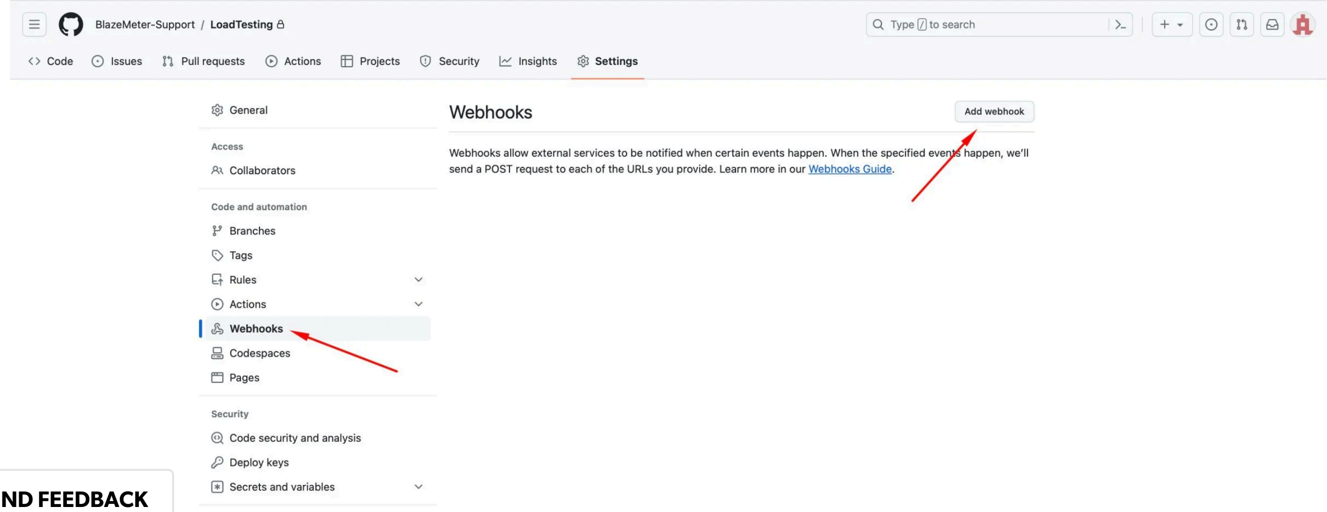


BlazeMeter-Support Create execution.yml bf8c119 2 minutes ago 1 commit

execution.yml Create execution.yml 2 minutes ago

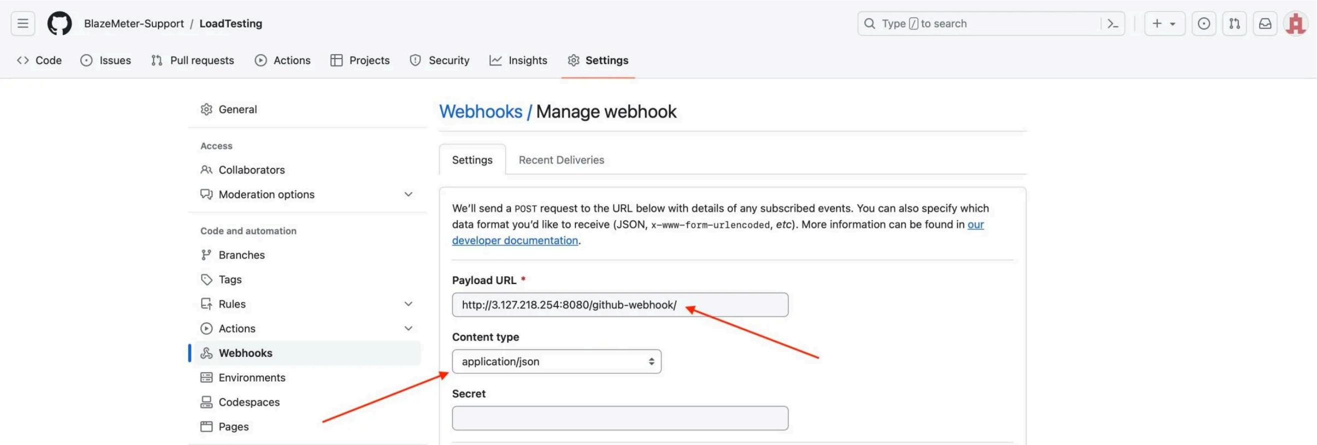
Add a README with an overview of your project. Add a README

Step 2: Click on **Webhooks** and then click on **‘Add webhook’.**



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.

Step 3: In the **'Payload URL'** field, paste your Jenkins environment URL. At the end of this URL add */github-webhook/*. In the **'Content type'** select: *'application/json'* and leave the **'Secret'** field empty.



Webhooks / Manage webhook

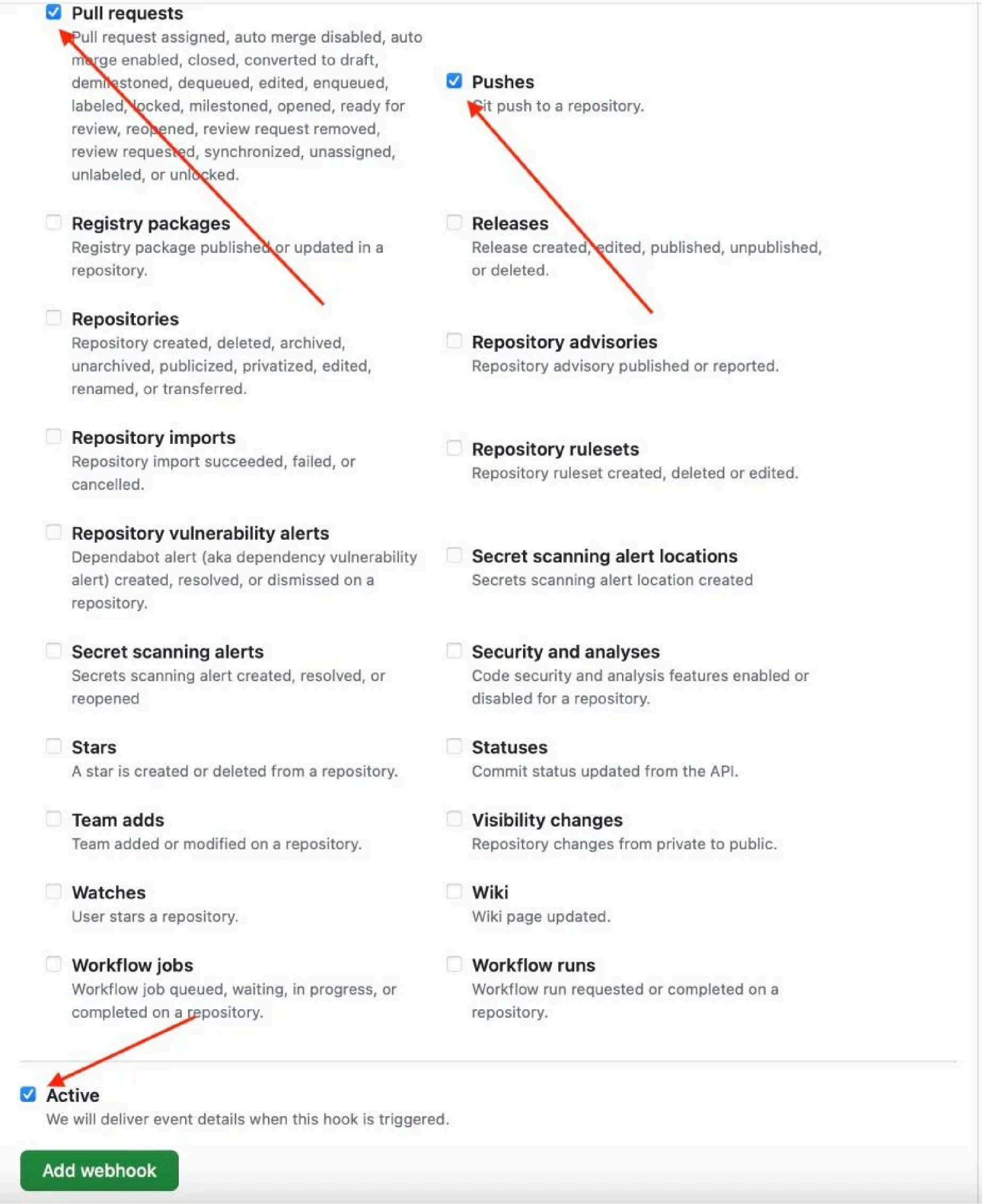
We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *: http://3.127.218.254:8080/github-webhook/

Content type: application/json

Secret:

Step 4: In the page **'Which events would you like to trigger this webhook?'** choose *'Let me select individual events.'* Then, check *'Pull Requests'* and *'Pushes'*. At the end of this option, make sure that the *'Active'* option is checked and click on *'Add webhook'*.



☑ **Pull requests**
Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

☑ **Pushes**
Git push to a repository.

□ **Registry packages**
Registry package published or updated in a repository.

□ **Releases**
Release created, edited, published, unpublished, or deleted.

□ **Repositories**
Repository created, deleted, archived, unarchived, publicized, privatized, edited, renamed, or transferred.

□ **Repository advisories**
Repository advisory published or reported.

□ **Repository imports**
Repository import succeeded, failed, or cancelled.

□ **Repository rulesets**
Repository ruleset created, deleted or edited.

□ **Repository vulnerability alerts**
Dependabot alert (aka dependency vulnerability alert) created, resolved, or dismissed on a repository.

□ **Secret scanning alert locations**
Secrets scanning alert location created

□ **Secret scanning alerts**
Secrets scanning alert created, resolved, or reopened

□ **Security and analyses**
Code security and analysis features enabled or disabled for a repository.

□ **Stars**
A star is created or deleted from a repository.

□ **Statuses**
Commit status updated from the API.

□ **Team adds**
Team added or modified on a repository.

□ **Visibility changes**
Repository changes from private to public.

□ **Watches**
User stars a repository.

□ **Wiki**
Wiki page updated.

□ **Workflow jobs**
Workflow job queued, waiting, in progress, or completed on a repository.

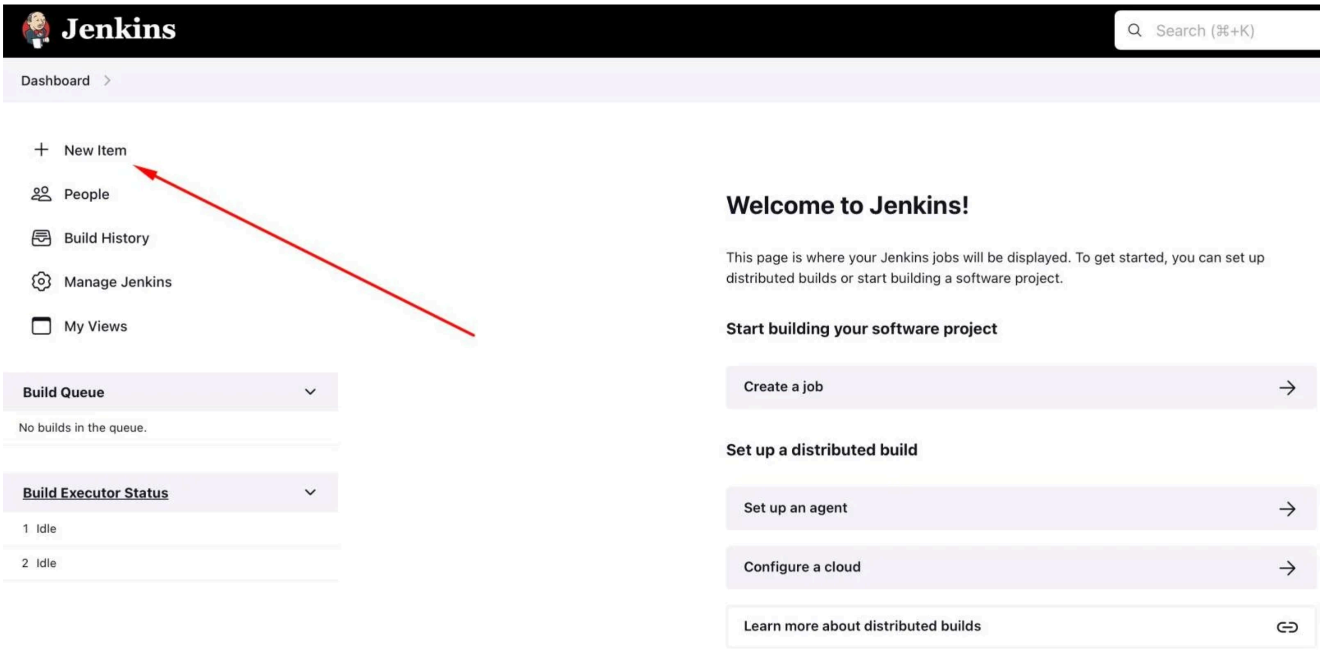
□ **Workflow runs**
Workflow run requested or completed on a repository.

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

We're done with the configuration on GitHub's side! Now let's move on to Jenkins.

Configuring Jenkins

Step 5: In Jenkins, click on **'New Item'** to create a new project.



Jenkins

Search (⌘+K)

Dashboard >

- + New Item
- People
- Build History
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

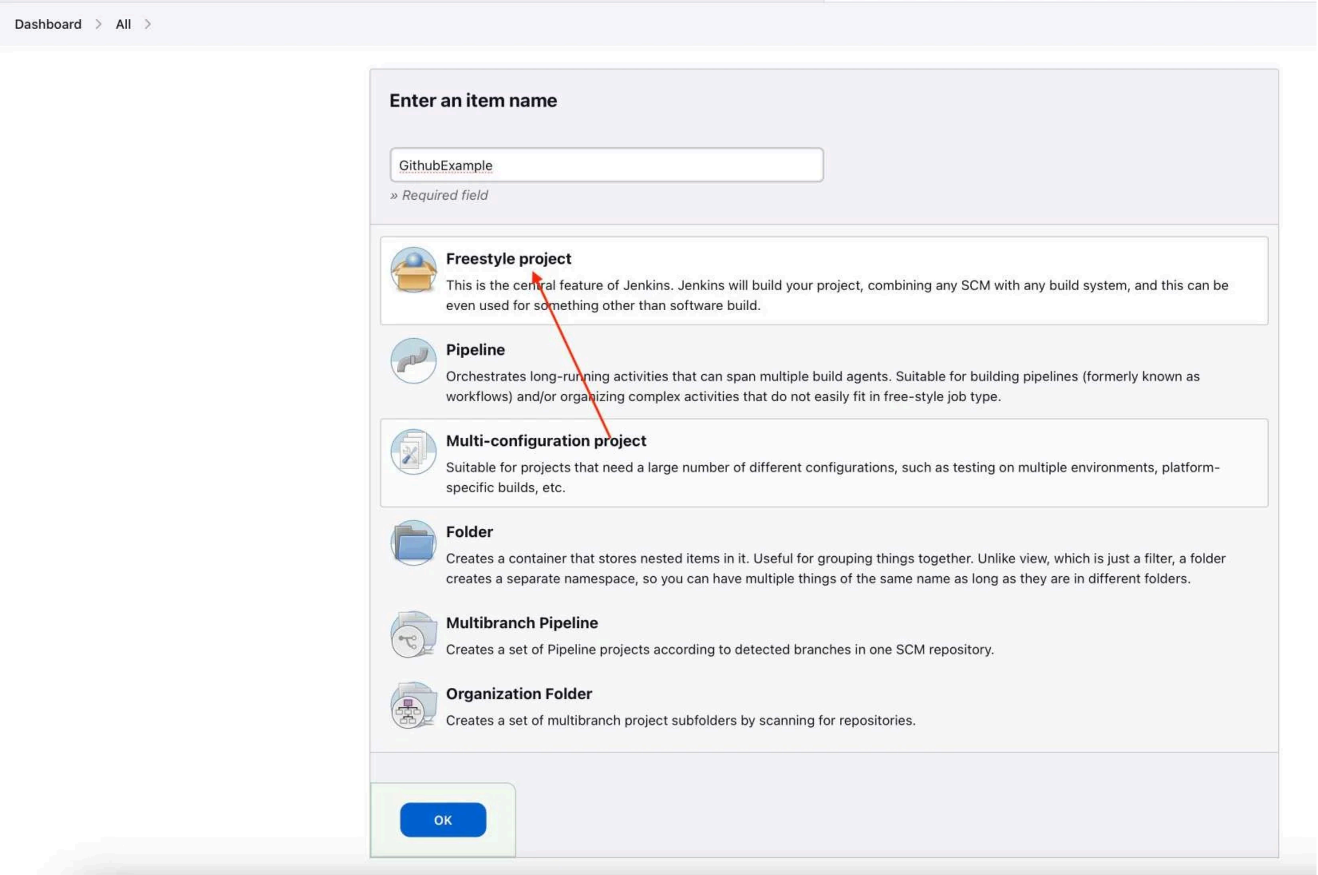
Set up a distributed build

Set up an agent →

Configure a cloud →

Learn more about distributed builds

Step 6: Give your project a name, then choose *'Freestyle project'* and finally, click on *'OK'.*



Dashboard > All >

Enter an item name

GithubExample

» 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 creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

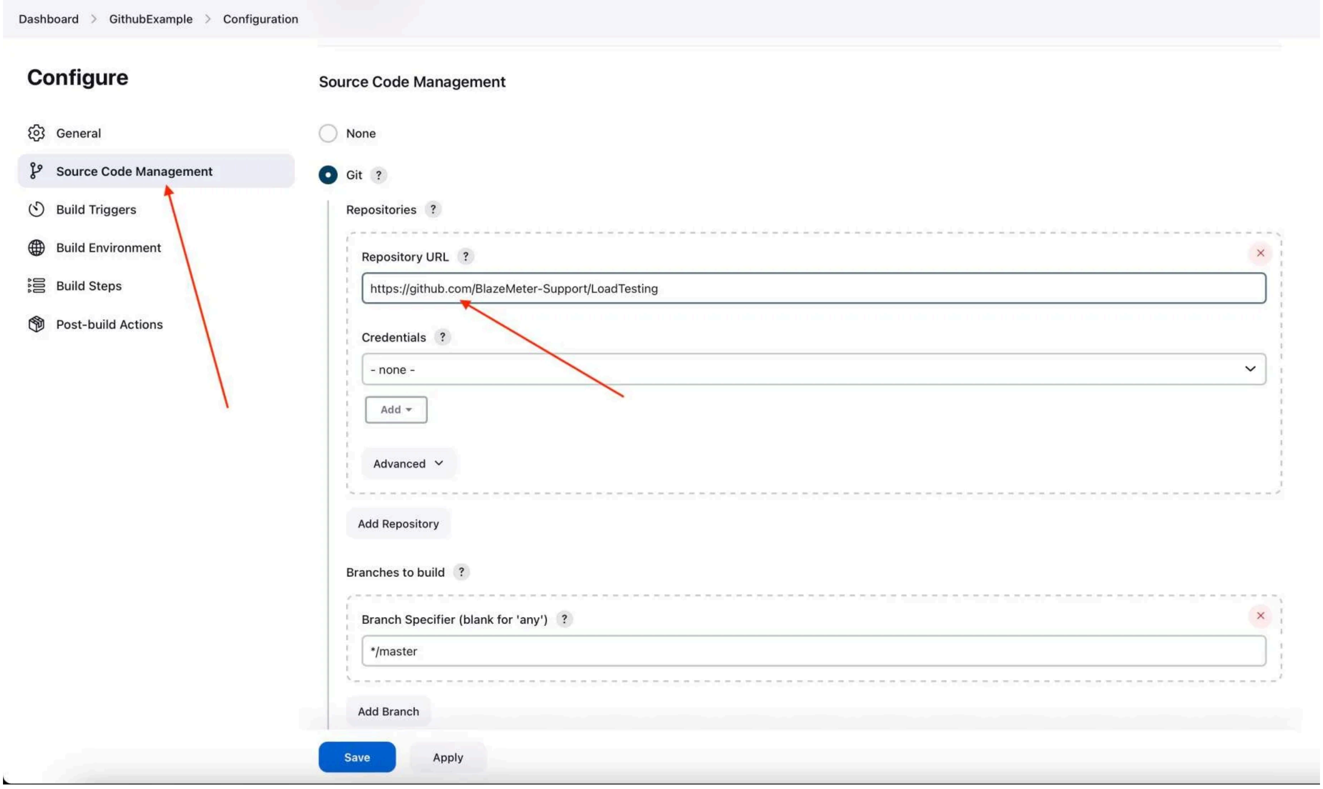
Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.

OK

Step 7: Click on the **'Source Code Management'** tab.

SEND FEEDBACK



Dashboard > GithubExample > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/BlazeMeter-Support/LoadTesting

Credentials ?

- none -

Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

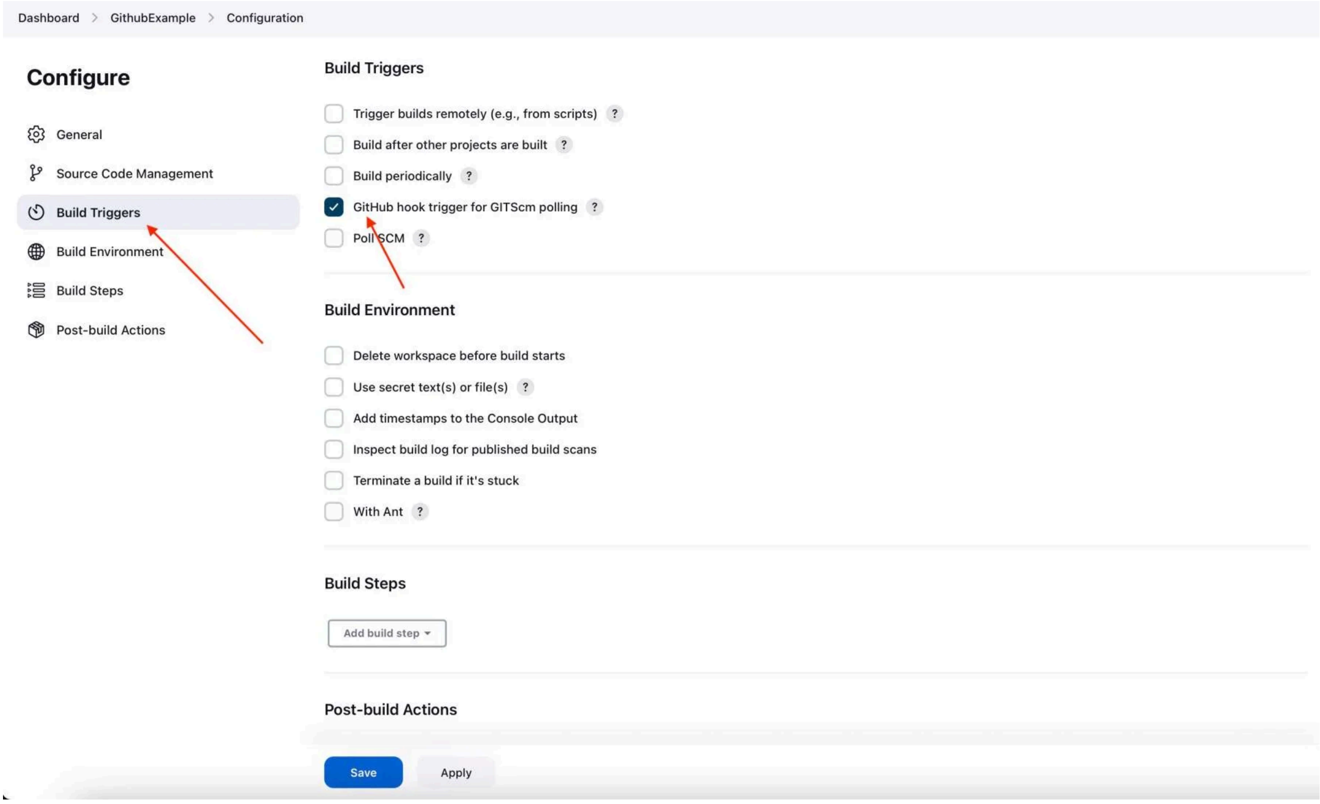
*/master

Add Branch

Save Apply

Step 8: Click on Git and paste your GitHub repository URL in the **'Repository URL'** field.

Step 9: Click on the **'Build Triggers'** tab and then on the *'GitHub hook trigger for GITScm polling'*. Or, choose the trigger of your choice.



Dashboard > GithubExample > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-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 ?

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

Add build step

Post-build Actions

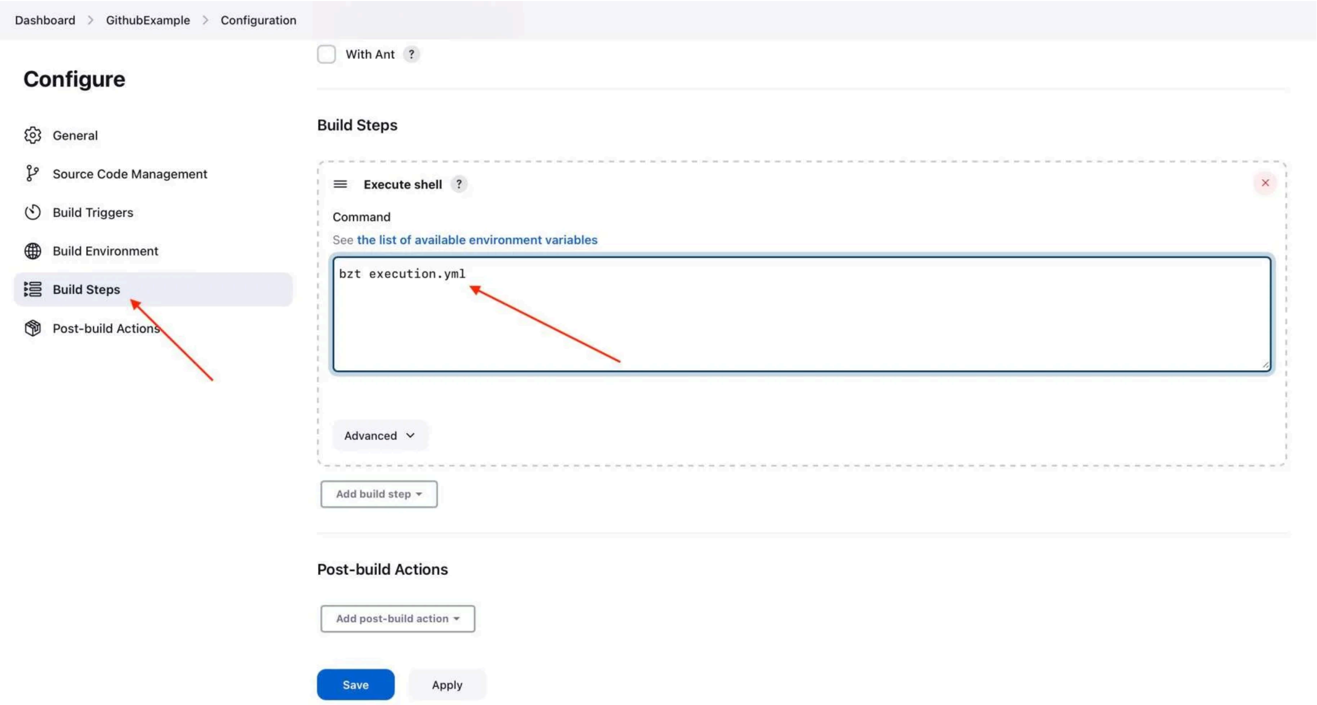
Save Apply

That's it! Your GitHub repository is integrated with your Jenkins project. With this Jenkins GitHub integration, you can now use any file found in the GitHub repository and trigger the Jenkins job to run with every code commit.

For example, I will show you how to run a Taurus script that I uploaded to my GitHub repository from my Jenkins project. Taurus (http://gettaurus.org) is an open source load testing solution, enabling developers to run load testing scripts from sophisticated platforms like JMeter and Selenium, but with a simple YAML code.

Triggering the GitHub Jenkins Integration With Every Code Commit

Step 10: Click on the **'Build'** tab, then click on *'Add build step'* and choose *'Execute shell'*.



```
bzt execution.yml
```

Step 11: To run a Taurus test, simply use the **'bzt'** command, followed by the name of your YML file and click on **'Save'.**

Step 12: Go back to your GitHub repository, edit the Taurus script and commit the changes. We will now see how Jenkins ran the script after the commit.

Step 13: Go back to your Jenkins project and you'll see that a new job was triggered automatically from the commit we made at the previous step. Click on the little arrow next to the job and choose **'Console Output'.**

Step 14: You can see that Jenkins was able to pull the Taurus script and run it!

Back to top

SEND FEEDBACK

Bottom Line

Congratulations! You have now set up a fully functioning Jenkins GitHub integration. Every time you publish your changes to GitHub, your newly set up GitHub-Jenkins integration will trigger your latest Jenkins job.

Once the GitHub plugin is fully installed and integrated into your Jenkins project, you have completed a very crucial step towards the full CI process. Now, you can proceed to the testing phase. To complete the full CI process, integrate your load testing into your CI tool. Using the BlazeMeter integration with Jenkins, incorporating load testing into your CI/CD pipeline has never been simpler.

START TESTING NOW (https://a.blazemeter.com/app/sign-up?__hstc=46213176.284fbb8ebb81c4e3cc139f4b91be38f3.1655221191797.1656097415722

Back to top

Guy Salton (/author/guy-salton)

Testing Expert

Guy Salton is a technical expert on the whole performance testing ecosystem - load testing tools, monitoring tools, CI tools, Networking and Infrastructure. His expertise is helping with POCs and special technical projects for strategic customers. Guy talks at conferences and meetups around the world, writes blog posts and gives webinars.

Functional
(/Product/Blazemeter/Functional-
Testing)
Performance
(/Product/Blazemeter/Load-
Testing)
APIs
(/Product/Blazemeter/Api-
Monitoring)
Mock Services
(/Product/Blazemeter/Mock-
Services)
Test Data
(/Product/Blazemeter/Test-
Data)

Jenkins CI
(/Solutions/Jenkins)
LoadRunner Vs.
BlazeMeter
(/Resources/Loadrunner-
Vs-Blazemeter)
Black Friday
(/Solutions/Black-
Friday)
Mainframe
(/Solutions/Mainframe)

PRICING >

Services)
Documentation
(Https://Help.Blazemeter.Com)
Status
(Https://Status.Blazemeter.Com/)
Changelog (/Release-
Notes/Blazemeter)

Webinars)
Papers & Videos
(/Resources/Papers-
And-Videos)
Recorded Webinars
(/Resources/Recorded-
Webinars)
Videos
(Https://Www.Youtube.Com/User/BlazeMeterSup

UNIVERSITY >

BLOG >

News (/Press/Press-
Releases)

CONTACT >

Contact Us (/Contact-
Us)

Produ
Blazei
Load
(Https
Testin
Perfor
Testin
(Https
Testin
API Te
Monit
(Https
Monit

Blazemeter by Perforce
(https://www.perforce.com)

Terms & Conditions
(https://www.perforce.com/sites/default/files/pdfs/BlazeMeter-Software-as-a-Service-Agreement_October-29-2021.pdf) | Privacy Policy
(https://www.perforce.com/privacy-policy) | Sitemap
(/sitemap)

BlazeMeter® (/)

SEND FEEDBACK