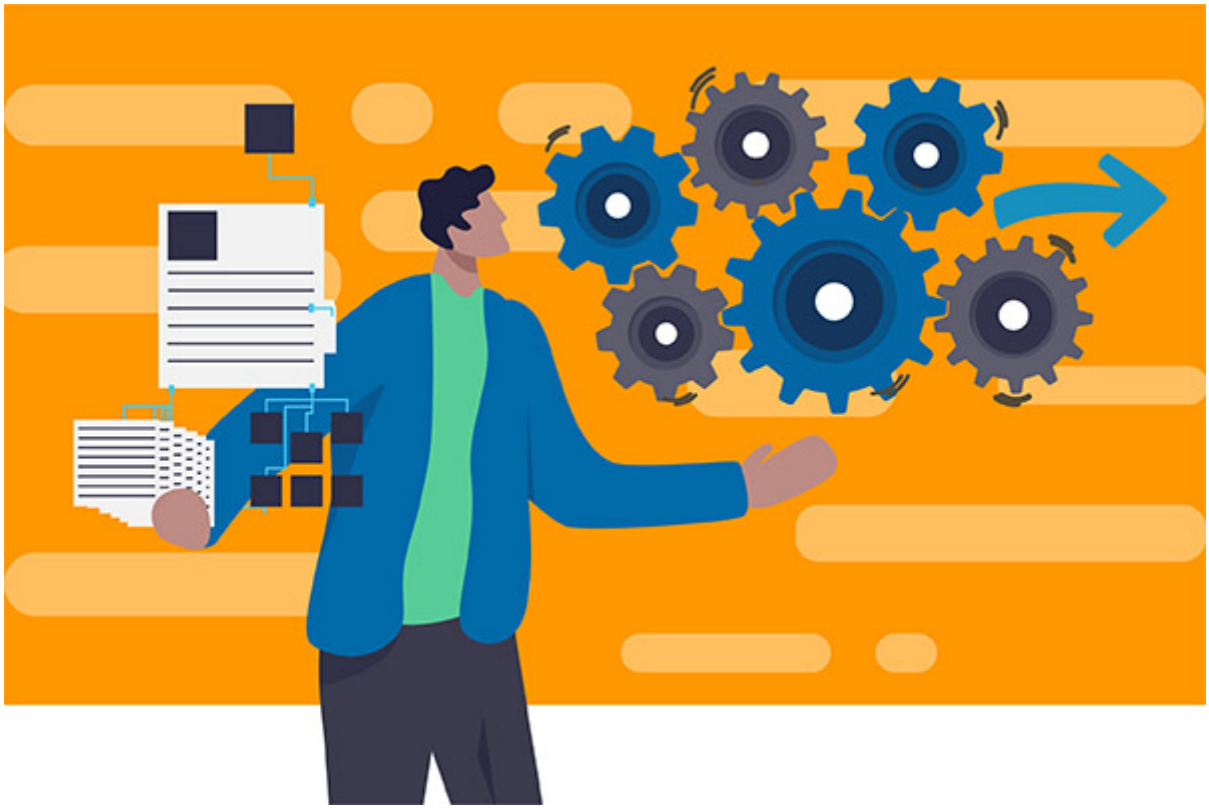


January 12, 2021

How to Integrate Your GitHub Repository to Your Jenkins Project

OPEN SOURCE AUTOMATION

By Guy Salton

GitHub and Jenkins (https://www.blazemeter.com/blog/jenkins-automation) — can you use them together? And can you integrate them? Find out in this blog!

Table of Contents:

Can You Use Jenkins With GitHub?

> 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. In this blog I will explain how to integrate Jenkins with GitHub projects.

Why Integrate GitHub + Jenkins?

You should integrate GitHub and Jenkins to improve efficiency.

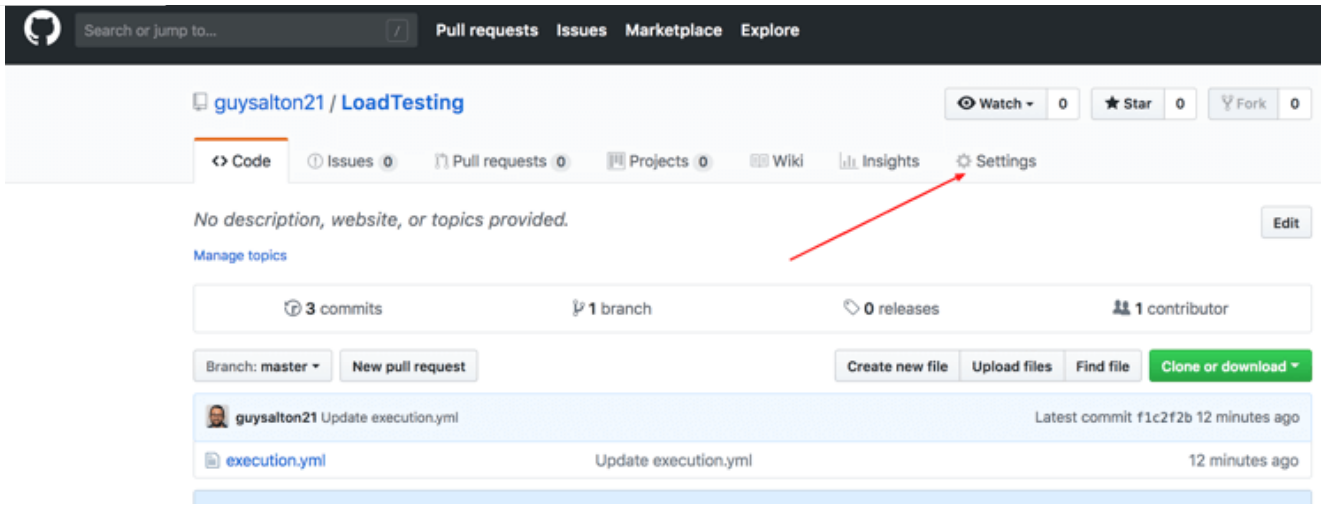
The integration presented in this blog post will teach you how to schedule your build, Pull your code and data files from your GitHub repository to your Jenkins machine, and automatically trigger each build on the Jenkins server after each Commit on your Git repository

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

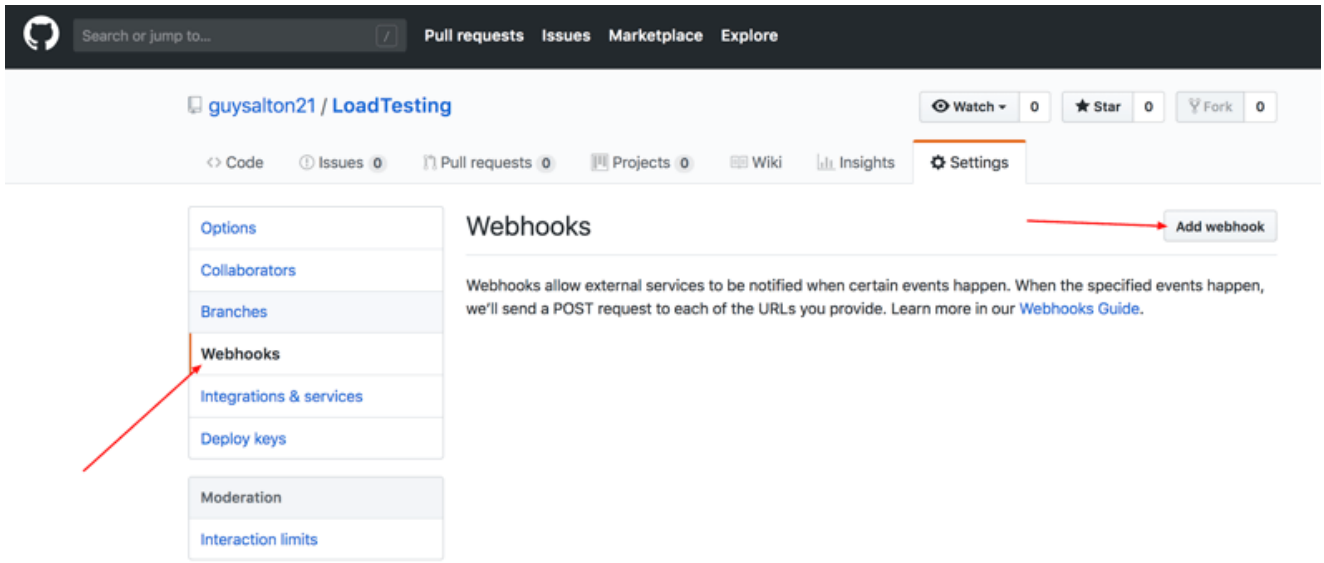
How to Integrate GitHub + Jenkins

Configuring GitHub

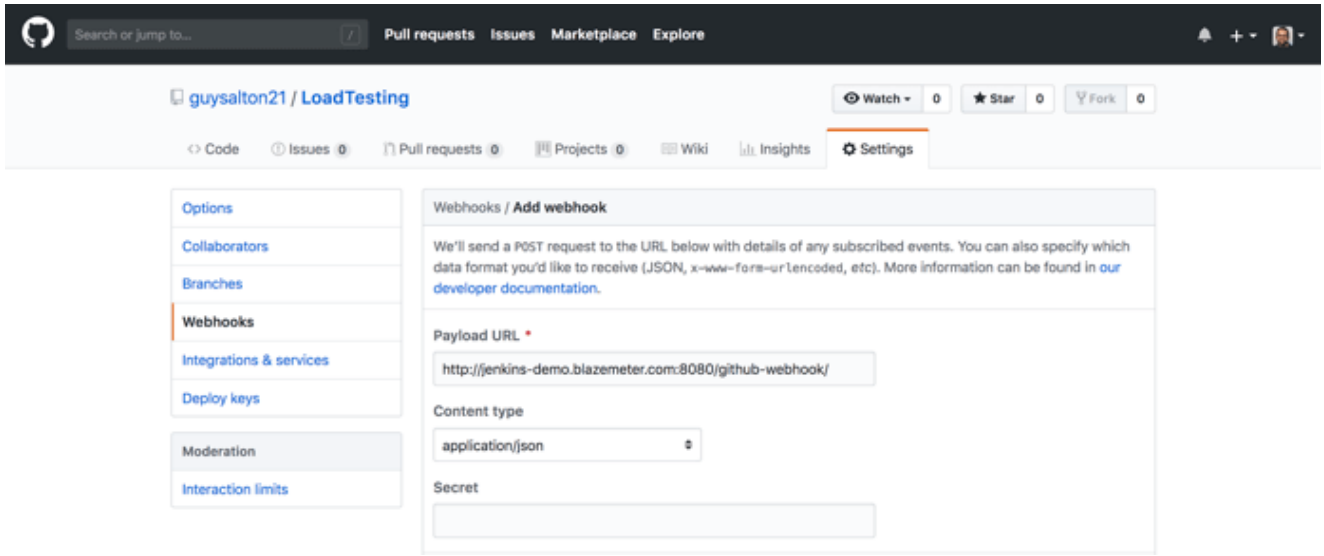
Step 1: go to your GitHub repository and click on **'Settings'.**



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

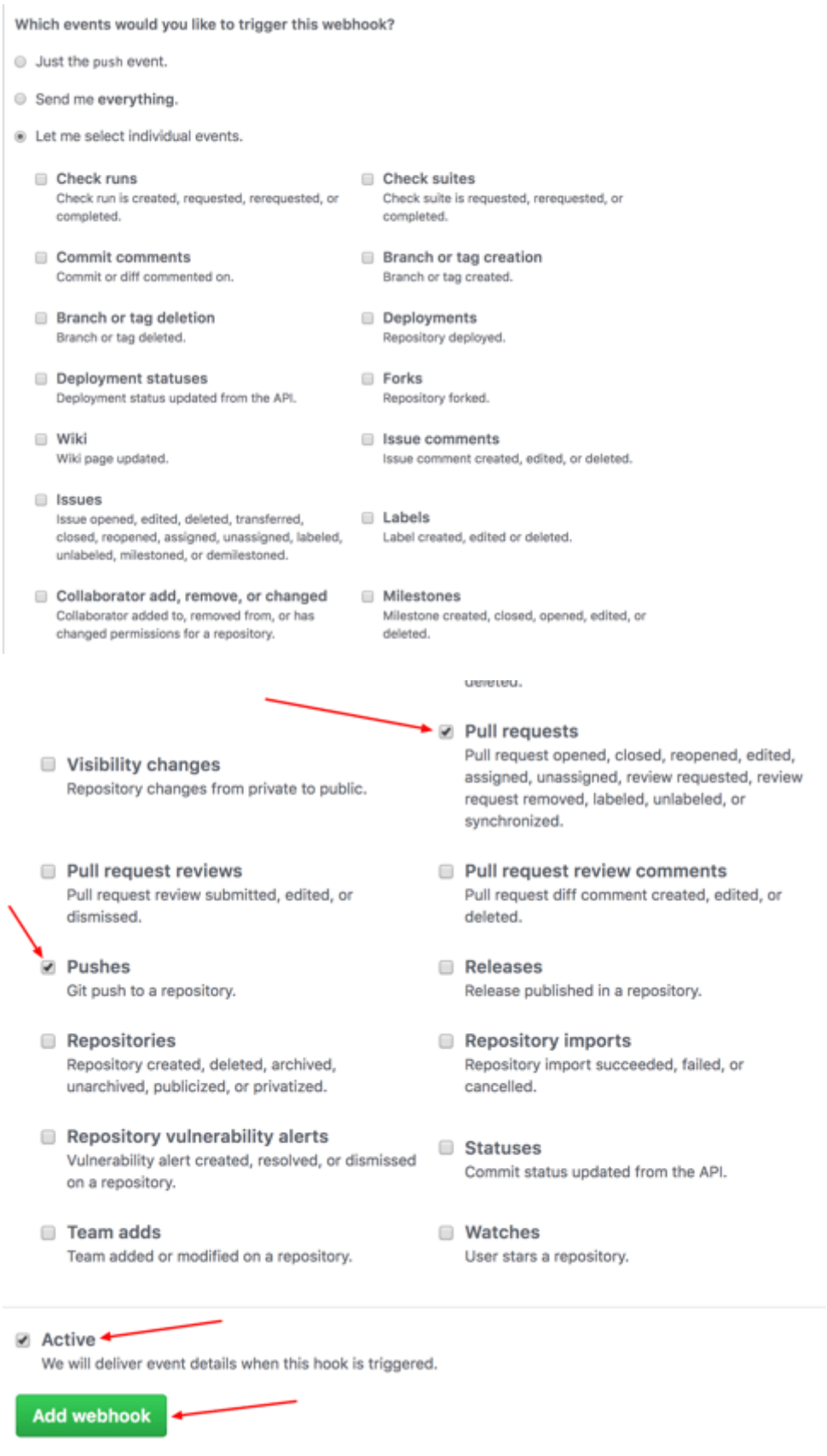


Step 3: In the **'Payload URL'** field, paste your Jenkins environment URL. At the end of this URL add */github-webhook/*. In the **'Content type'** select: *'application/json'* and leave the **'Secret'** field empty.



SEND FEEDBACK

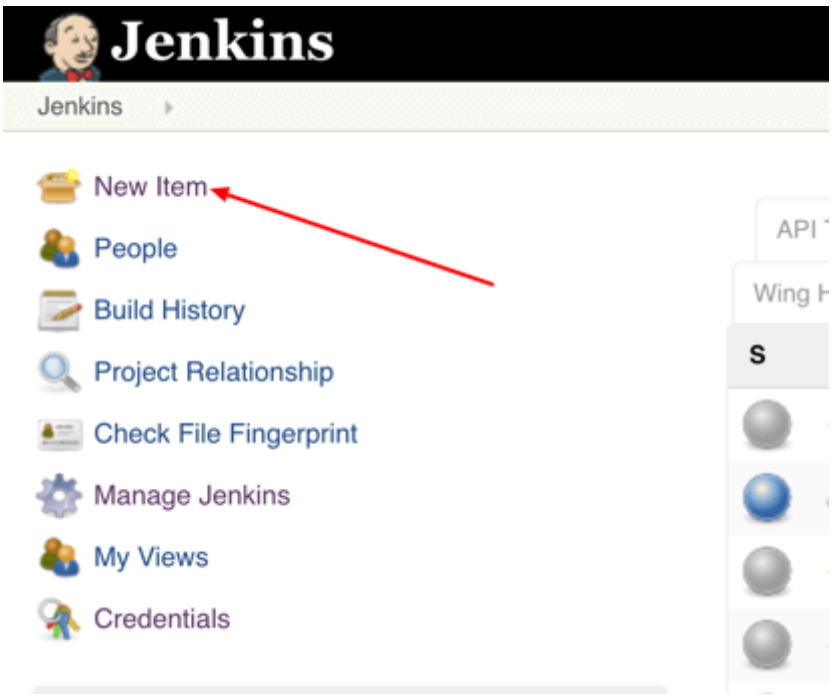
Step 4: In the page **'Which events would you like to trigger this webhook?'** choose *'Let me select individual events.'* Then, check *'Pull Requests'* and *'Pushes'*. At the end of this option, make sure that the *'Active'* option is checked and click on *'Add webhook'*.



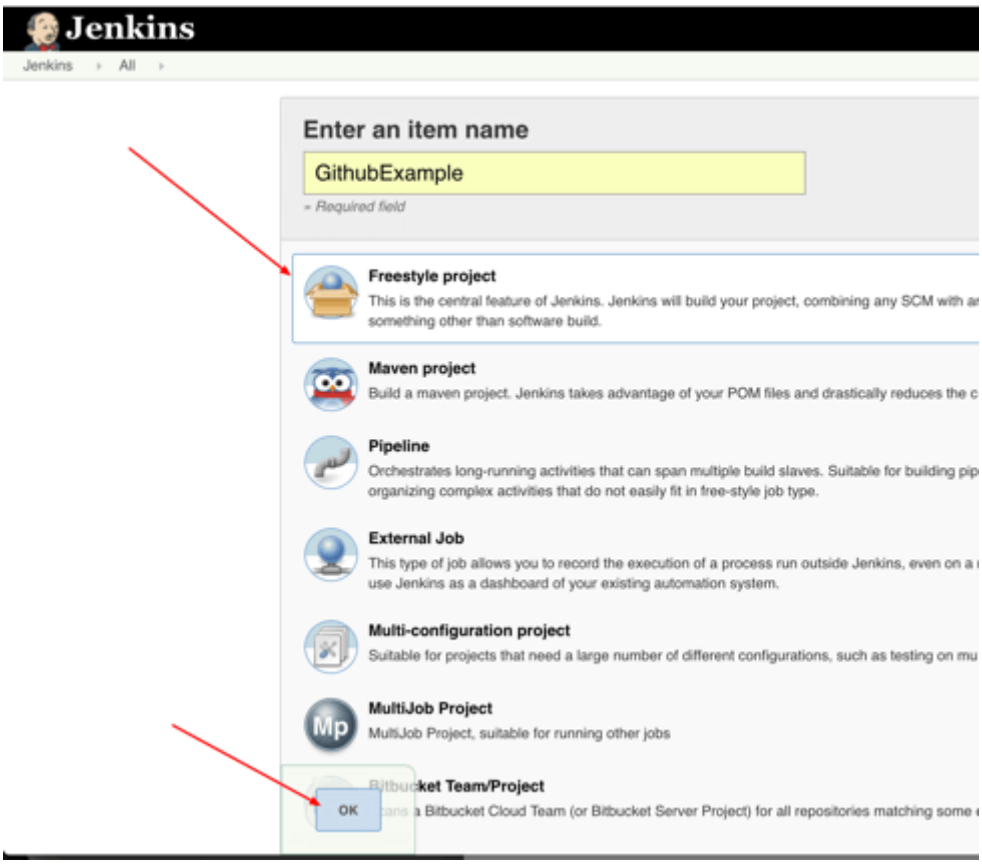
We're done with the configuration on GitHub's side! Now let's move on to Jenkins.

Configuring Jenkins

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

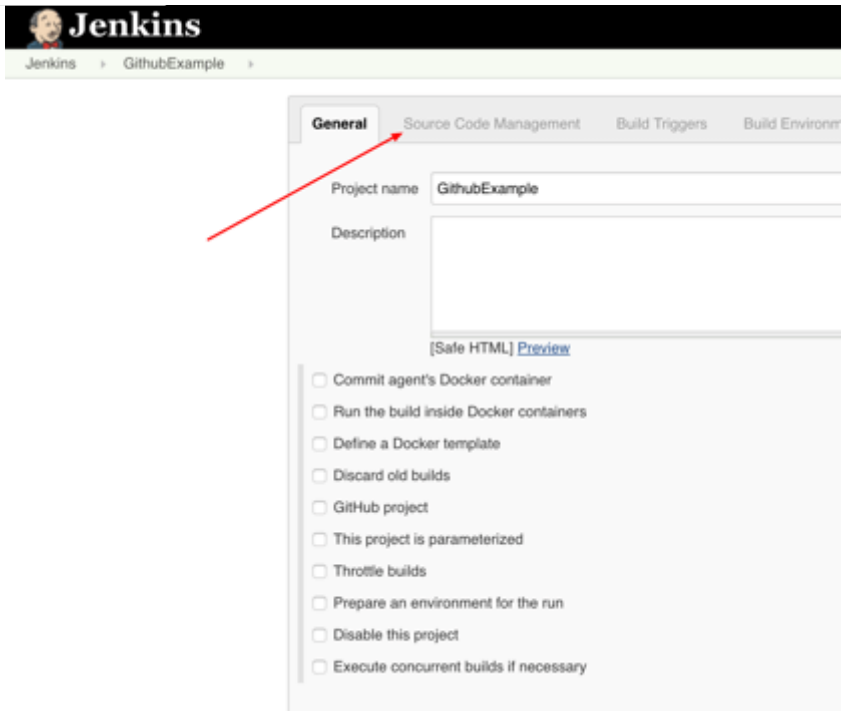


Step 6: Give your project a name, then choose *'Freestyle project'* and finally, click on *'OK'*.

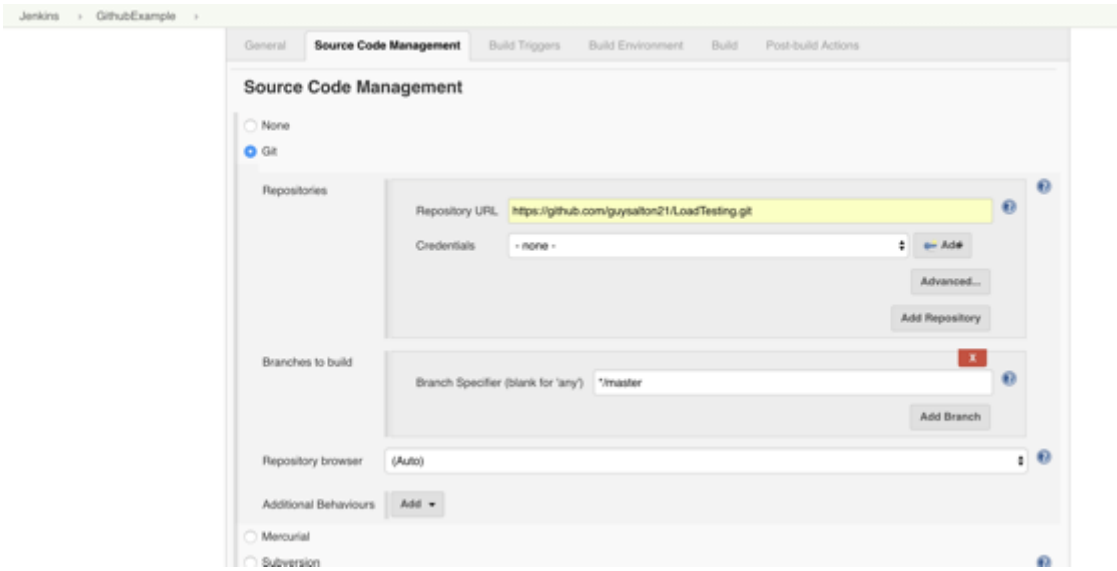


Step 7: Click on the **'Source Code Management'** tab.

SEND FEEDBACK

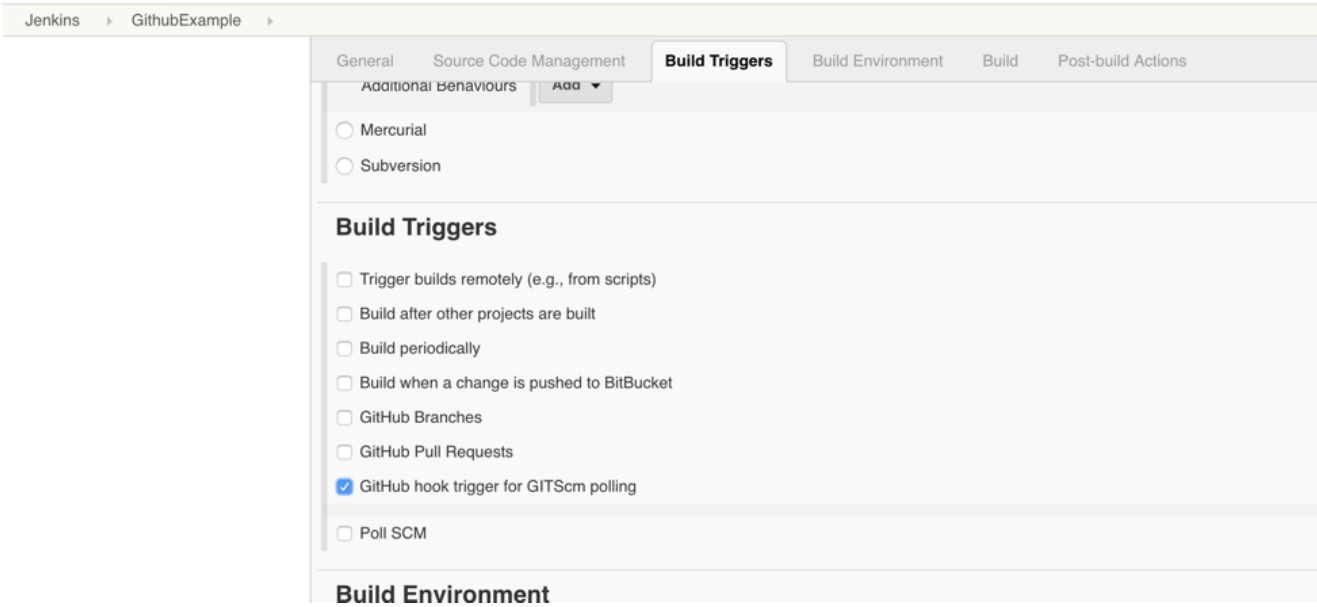


Step 8: Click on Git and paste your GitHub repository URL in the **'Repository URL'** field.



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.

SEND FEEDBACK

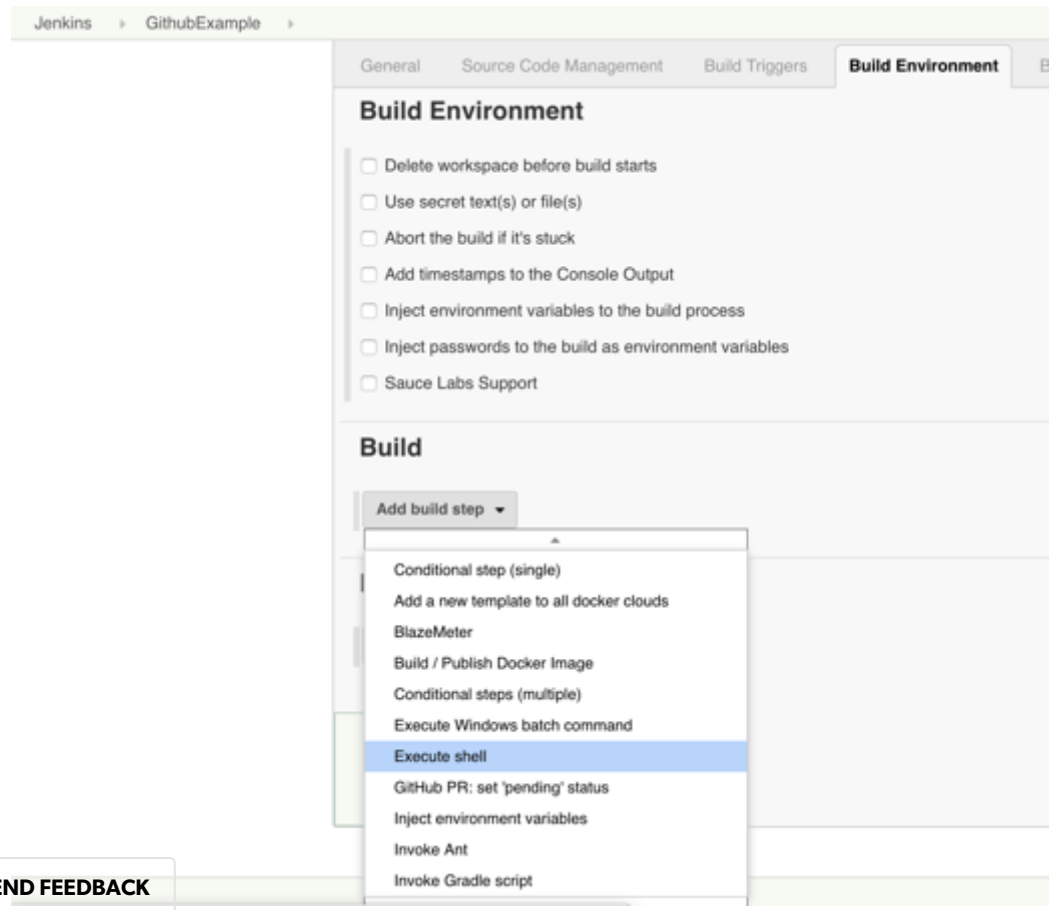


That's it! Your GitHub repository is integrated with your Jenkins project. You can now use any of the files found in the GitHub repository and trigger the Jenkins job to run with every code commit.

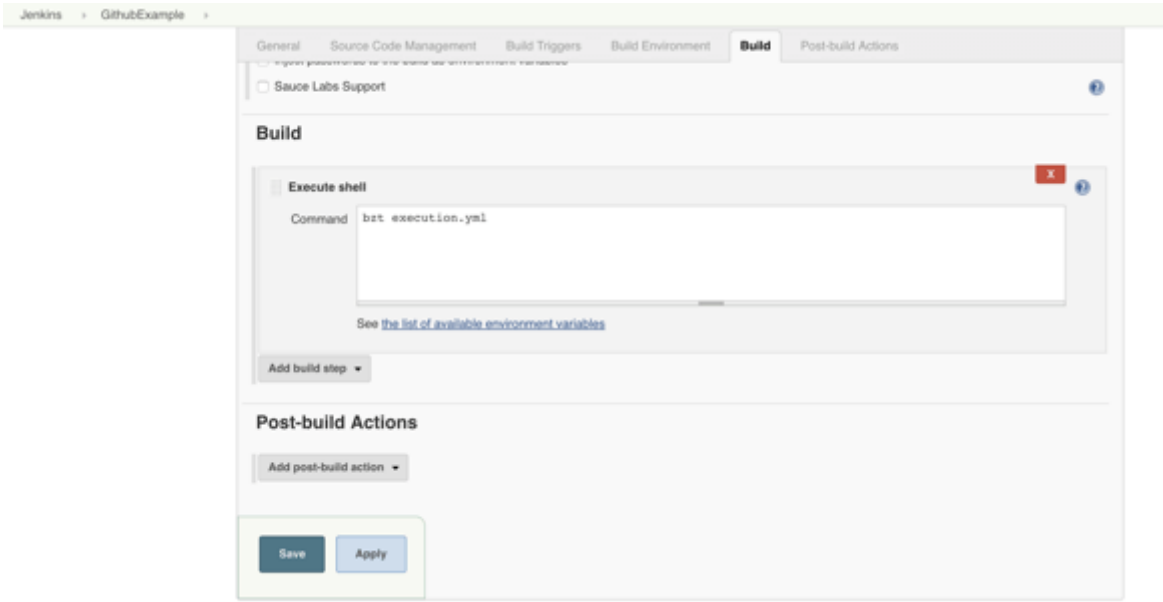
For example, I will show you how to run a Taurus script that I uploaded to my GitHub repository from my Jenkins project.

Triggering the Jenkins GitHub Integration With Every Code Commit

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

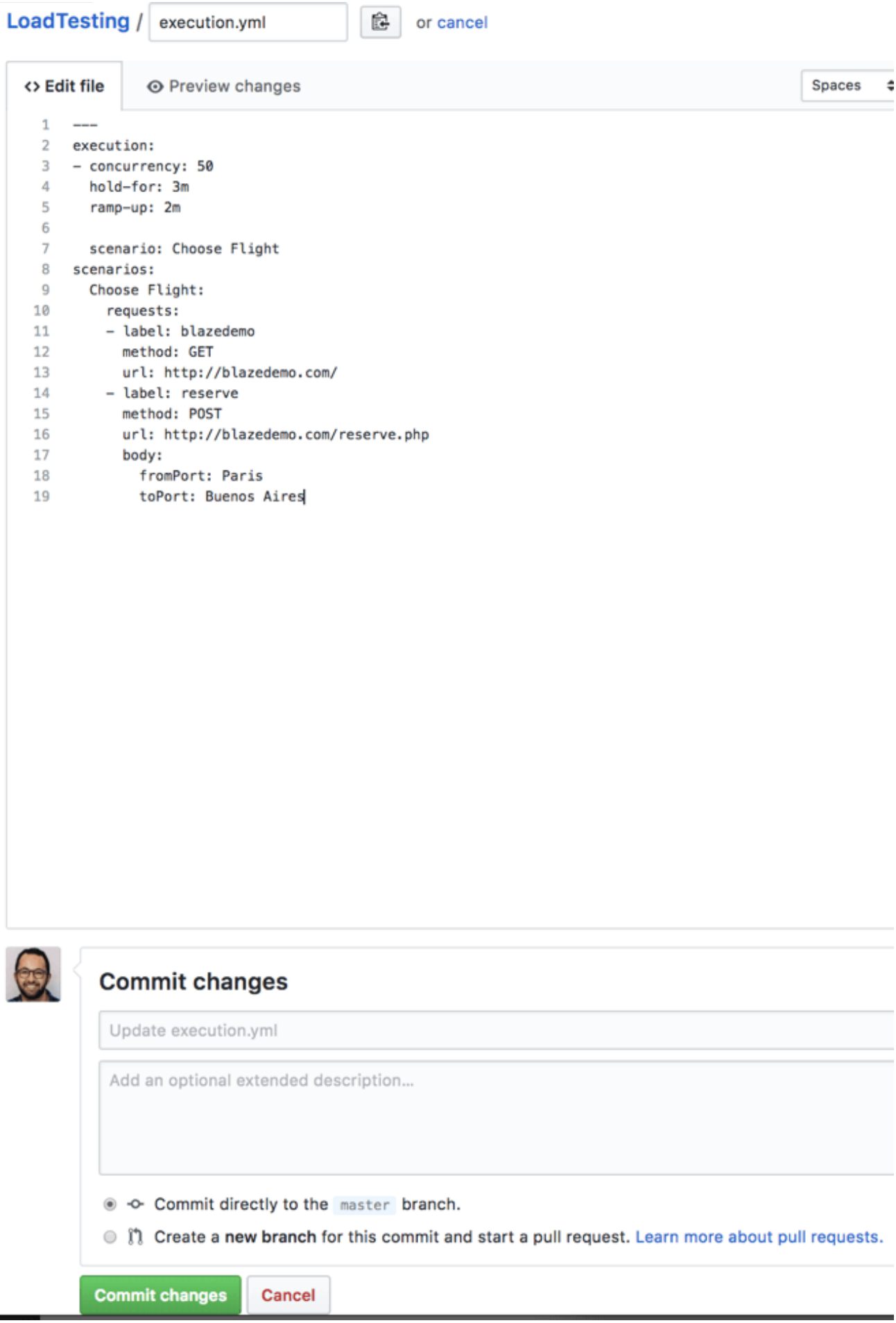


Step 11: To run a Taurus test, simply use the **'bzt'** command, followed by the name of your YML file and click on **'Save'.**

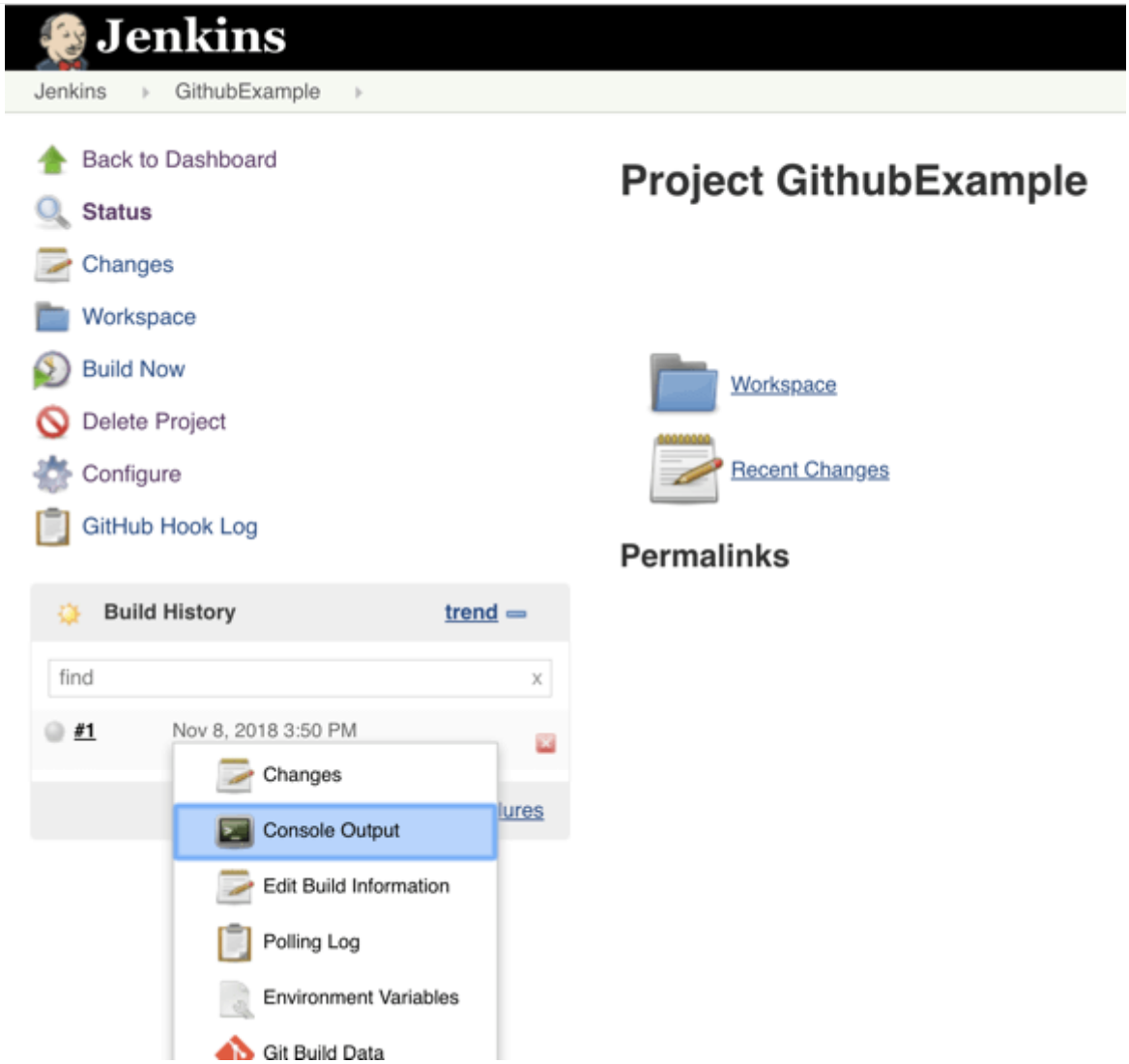


Step 12: Go back to your GitHub repository, edit the Taurus script and commit the changes. We will now see how Jenkins ran the script after the commit.

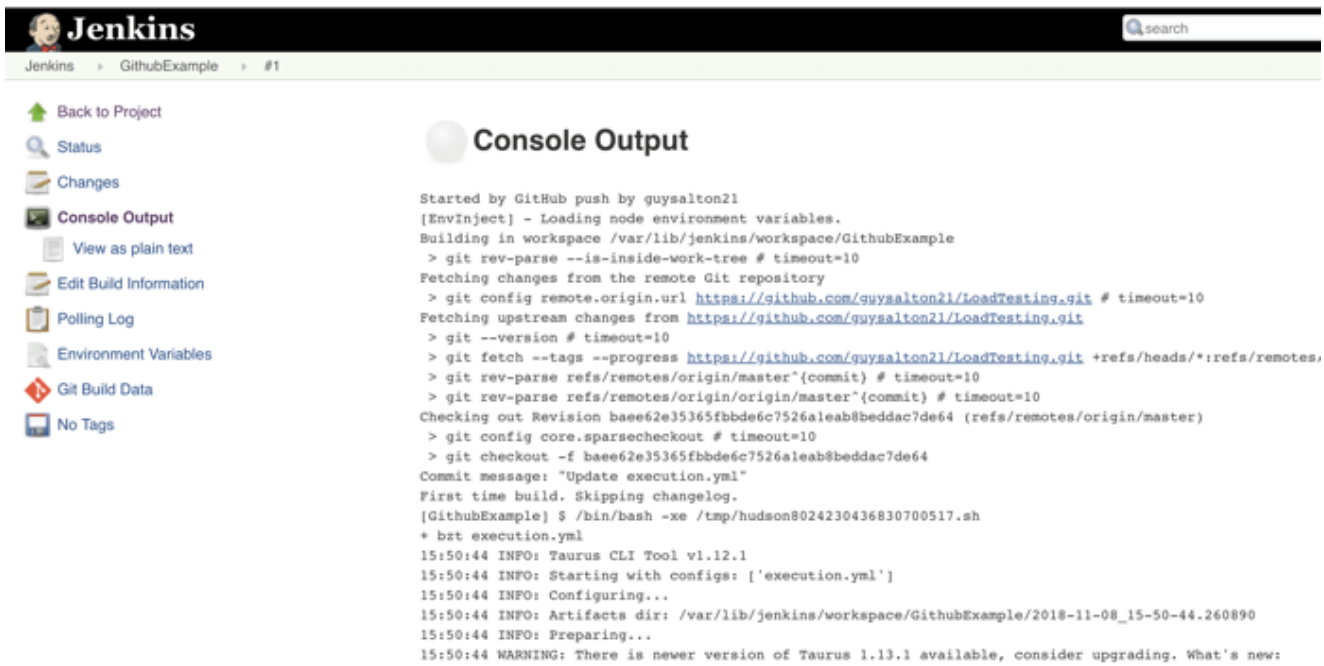
SEND FEEDBACK



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!



Get Started With Jenkins, GitHub, and BlazeMeter

SEND FEEDBACK

Congratulations! Every time you publish your changes to Github, GitHub will trigger your new 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.

START TESTING NOW (HTTPS://A.BLAZEMETER.COM/APP/SIGN-UP?__HSTC=46213176.284FBB8EBB81C4E3CC139F4B91BE38F3.1655221191797.1656

Guy Salton (/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.

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

(/Resources/Loadrunner-Vs-Blazemeter)
Black Friday (/Solutions/Black-Friday)
Mainframe (/Solutions/Mainframe)

PRICING >

Utm_medium=BM_Blog&Utm_source=Blog&Utm_campaign=Advanced-Load-Testing-Jmeter-1-Correlations)
Status (Https://Status.Blazemeter.Com/)
Changelog (/Release-Notes/Blazemeter)

Contact Us (/Contact-Us)
(/Resources/Recorded-Webinars)
Videos (Https://Www.Youtube.Com/User/BlazeMeterSupp

UNIVERSITY >

BLOG >

(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