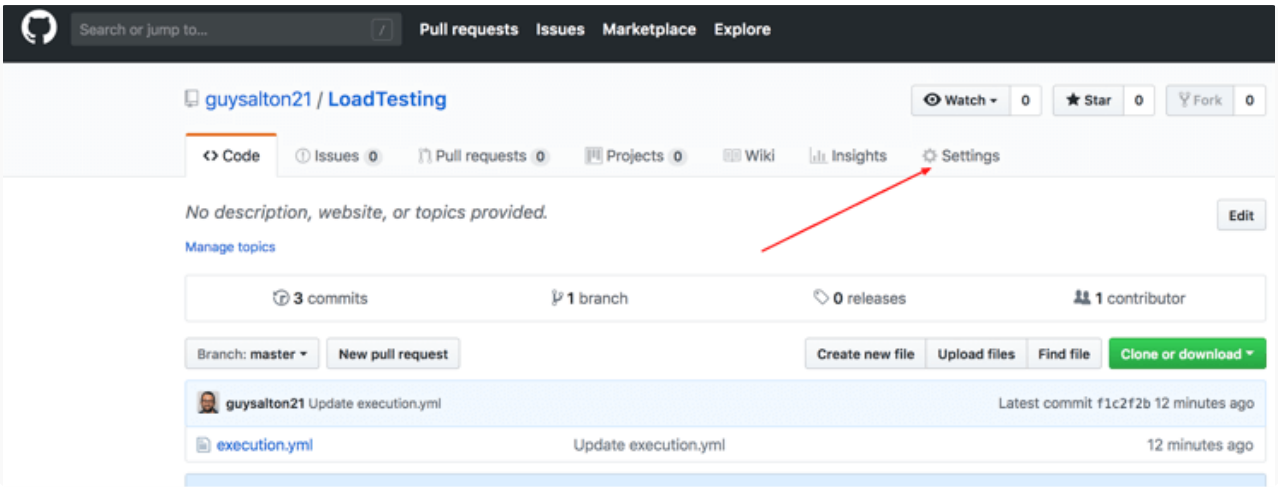
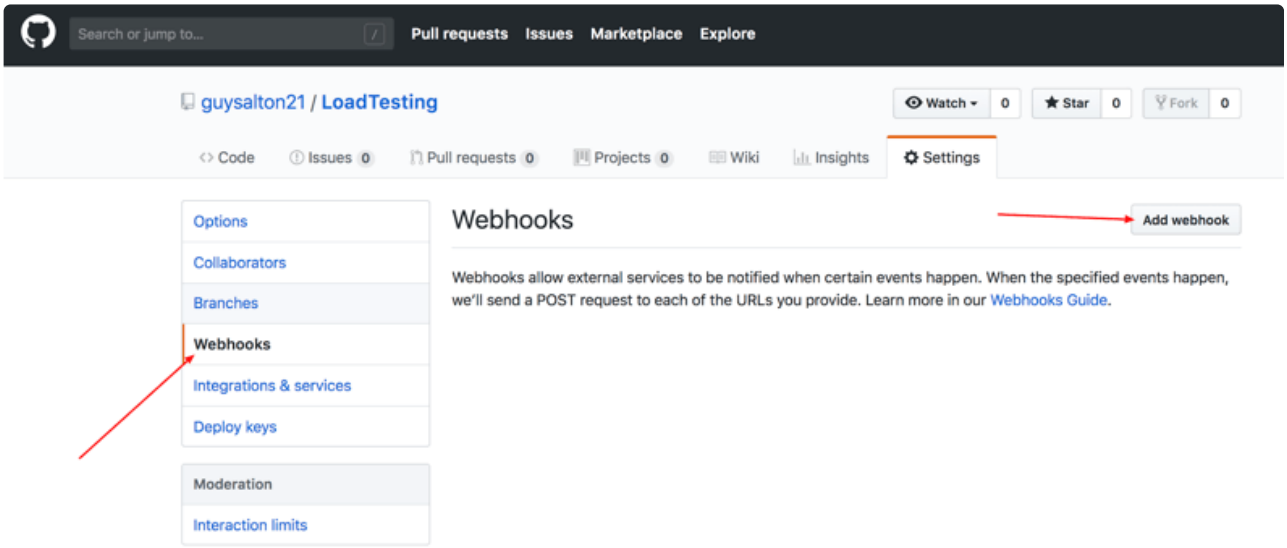
How to Set Up the Jenkins + GitHub Integration

Configuring GitHub

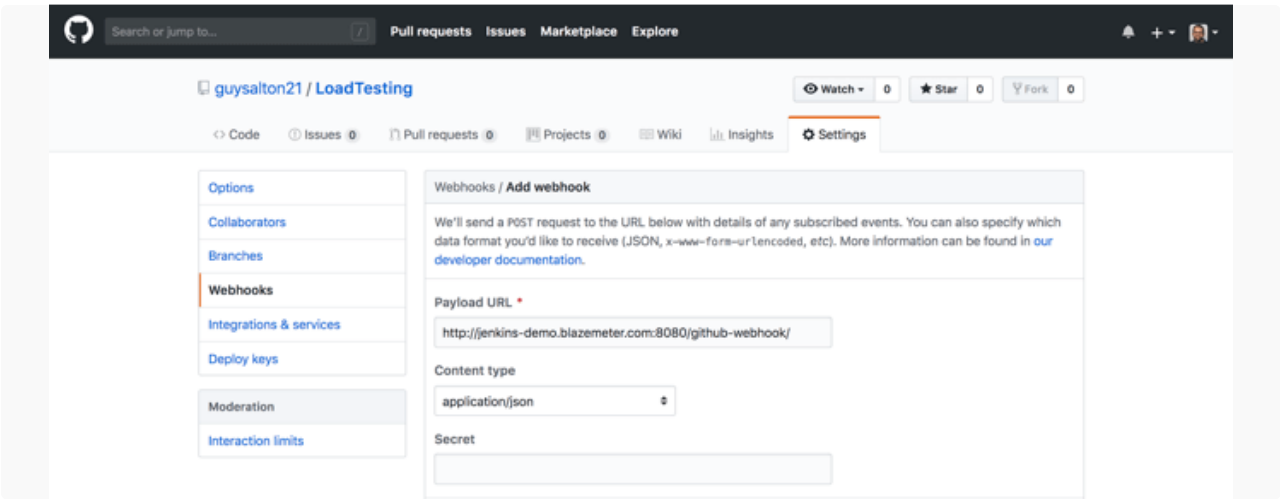
Step 1: go to your GitHub repository and click on **'Settings'.**



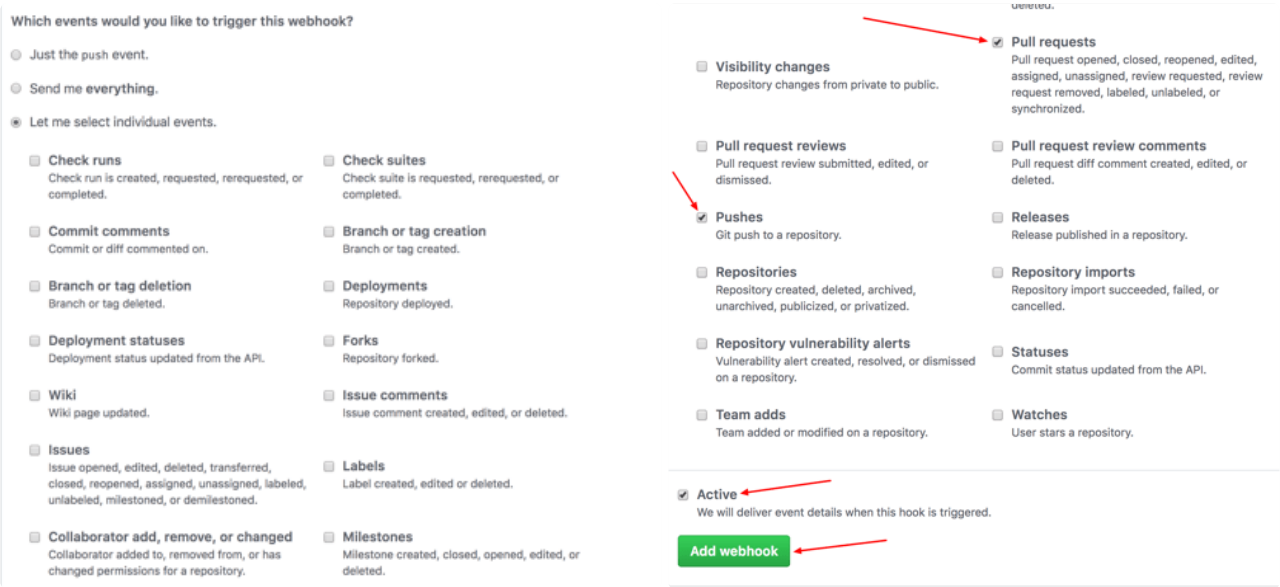
Step 2: Click on **Webhooks** and then click on **'Add webhook'.**



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.



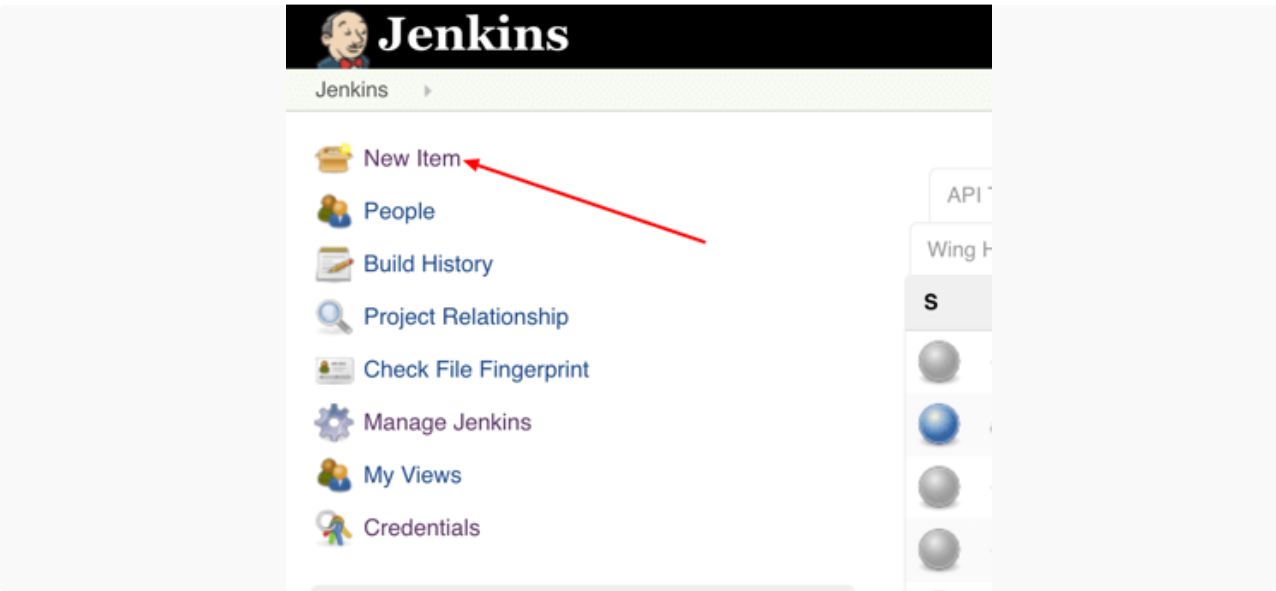
Step 4: In the page **'Which events would you like to trigger this webhook?'** choose *'Let me select individual events.'* Then, check *'Pull Requests'* and *'Pushes'*. At the end of this option, make sure that the *'Active'* option is checked and click on *'Add webhook'*.



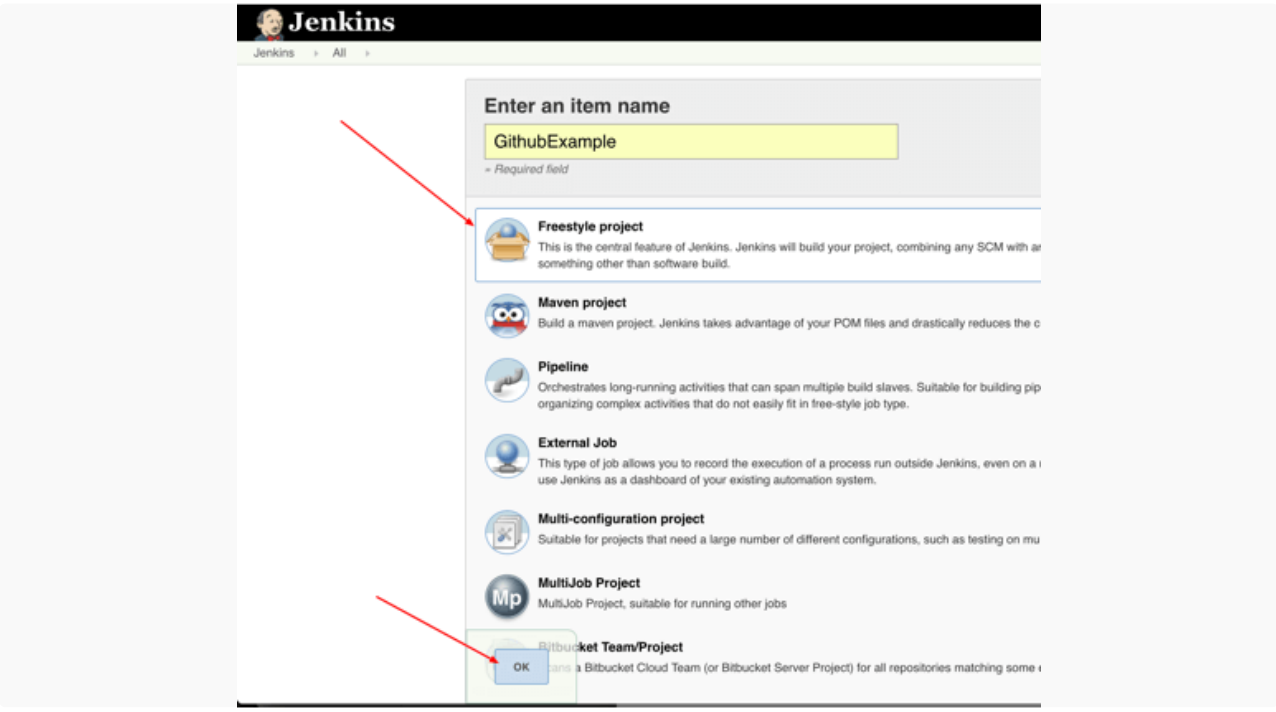
We're done with the configuration on GitHub's side! Now let's move on to Jenkins.

Configuring Jenkins

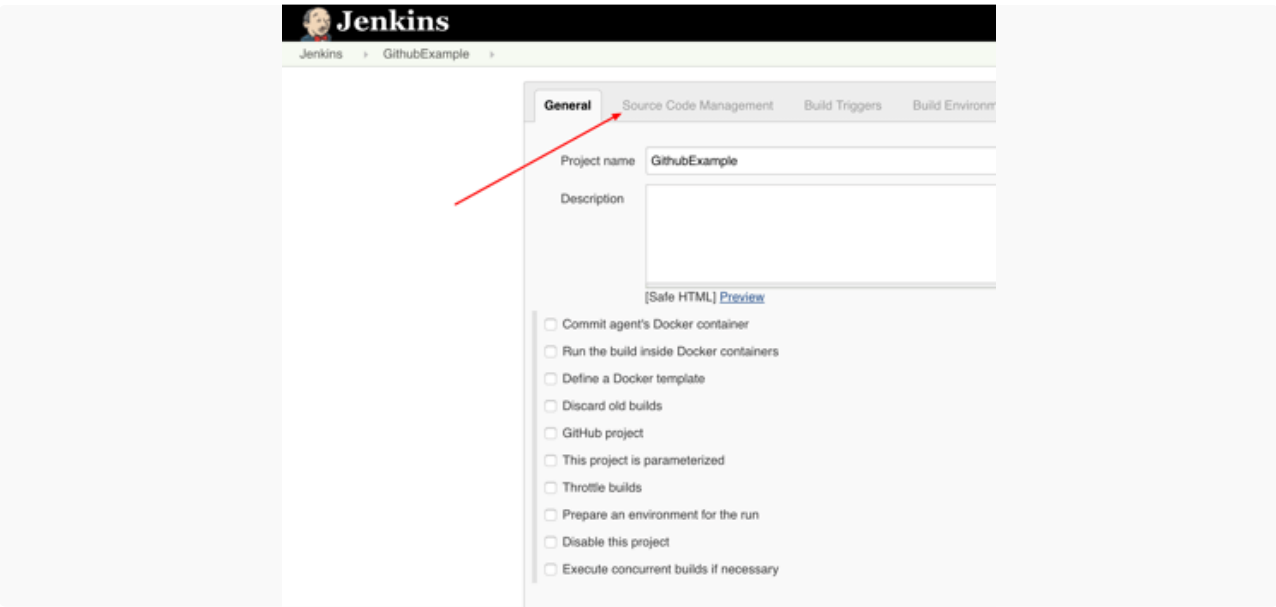
Step 5: In Jenkins, click on **'New Item'** to create a new project.



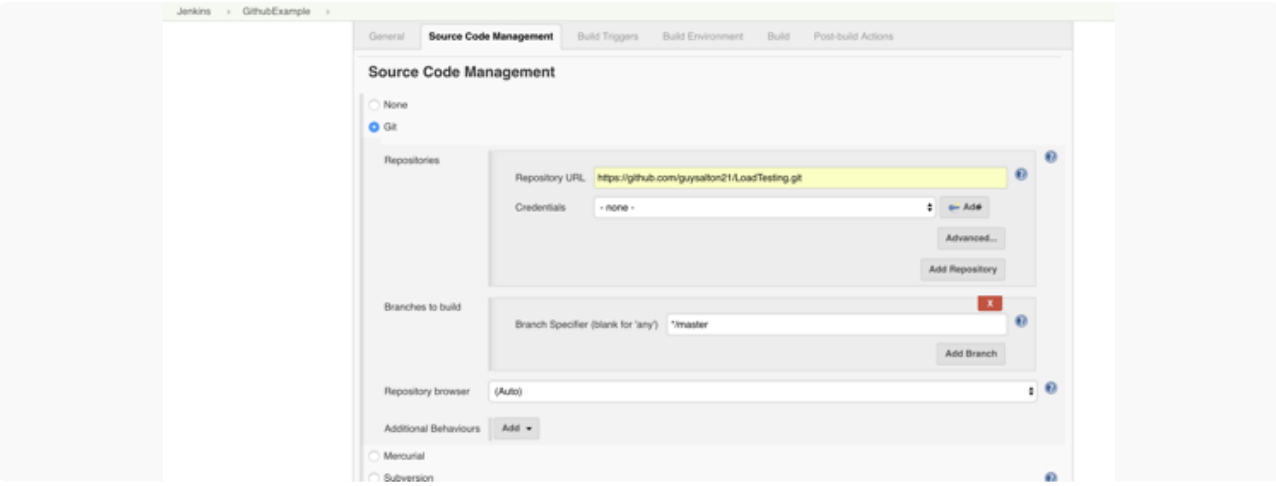
Step 6: Give your project a name, then choose *'Freestyle project'* and finally, click on *'OK'*.



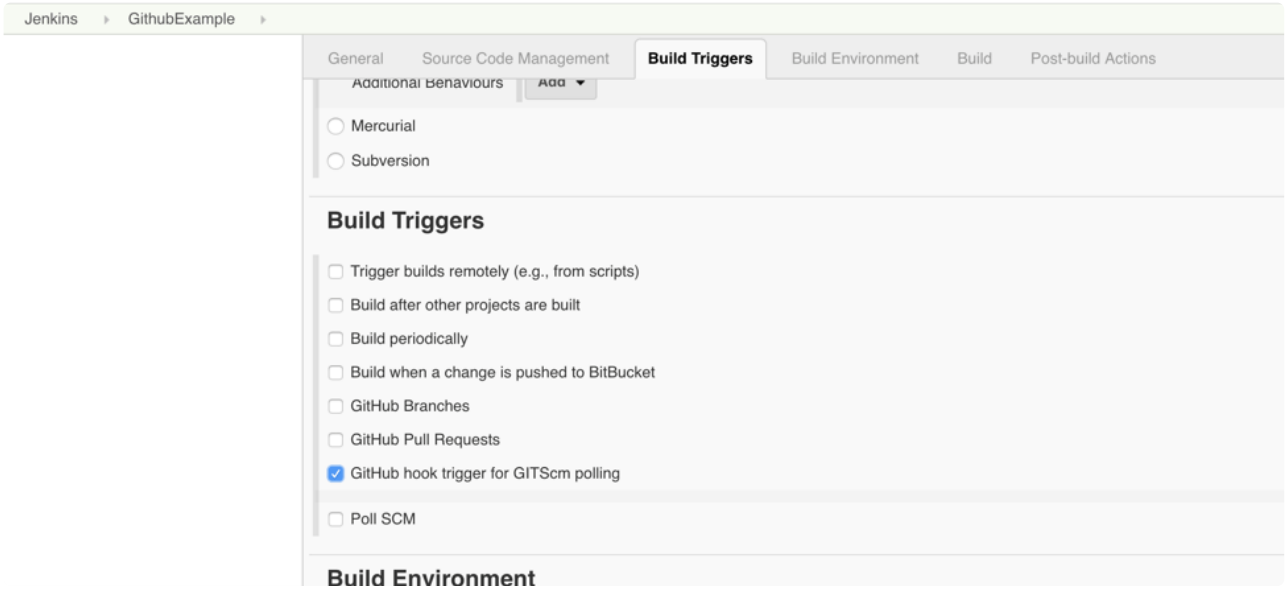
Step 7: Click on the **'Source Code Management'** tab.



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.

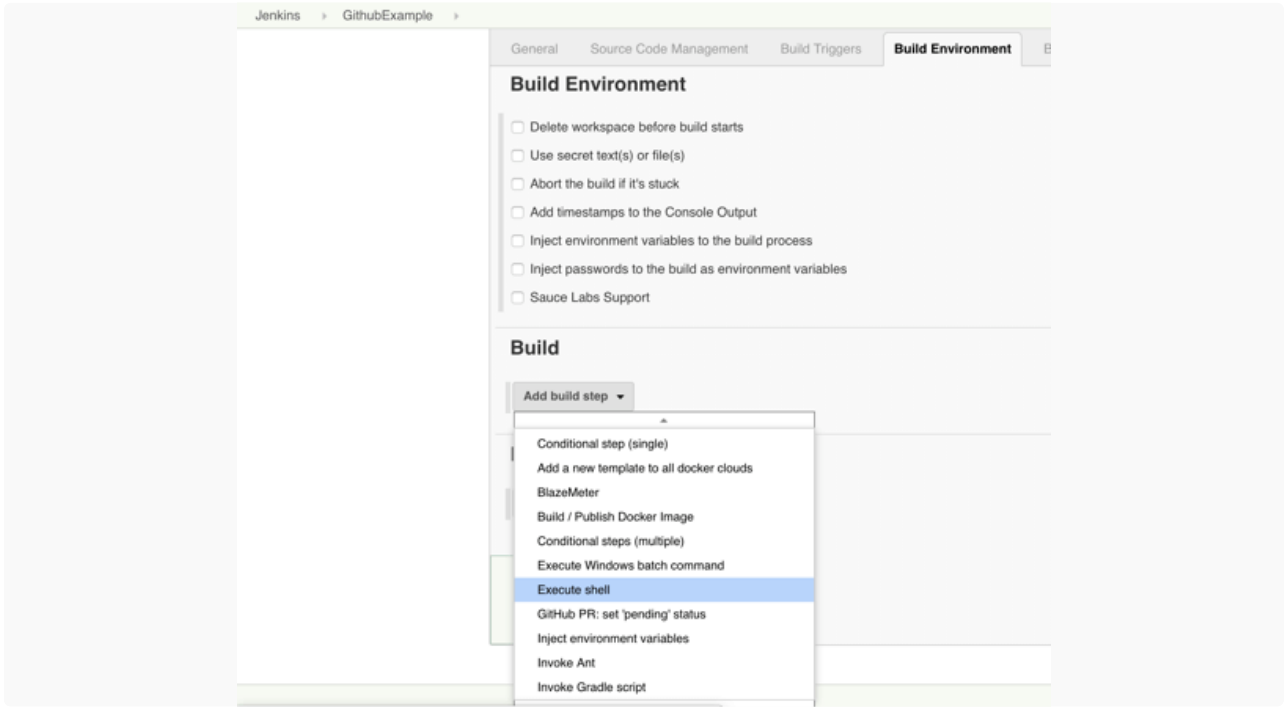


That's it! Your GitHub repository is integrated with your Jenkins project. With this Jenkins GitHub integration, you can now use any file found in the GitHub repository and trigger the Jenkins job to run with every code commit.

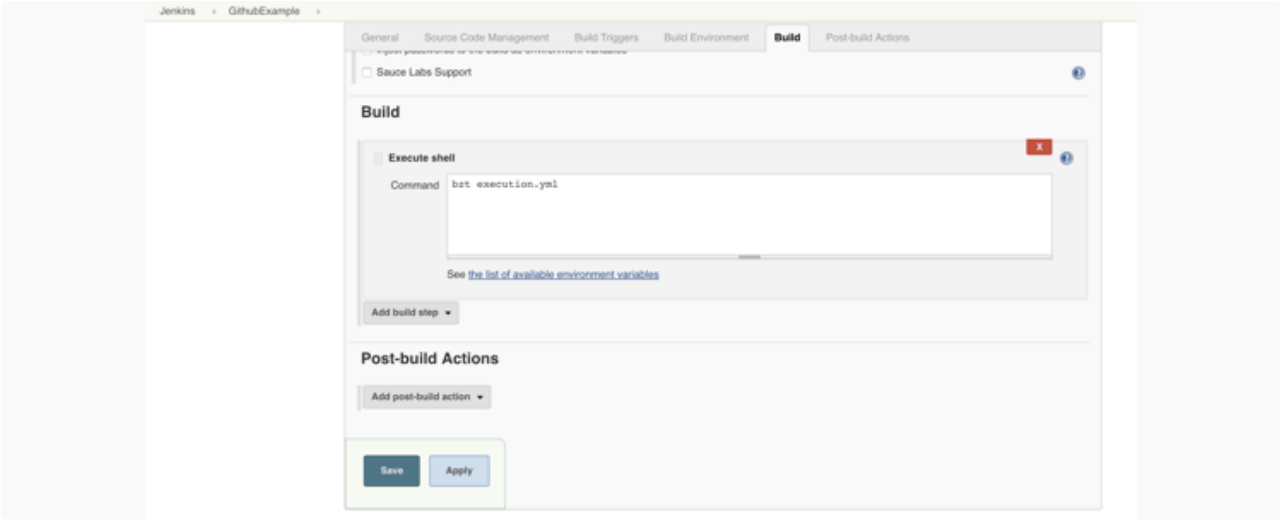
For example, I will show you how to run a Taurus script that I uploaded to my GitHub repository from my Jenkins project. Taurus is an open source load testing solution, enabling developers to run load testing scripts from sophisticated platforms like JMeter and Selenium, but with a simple YAML code.

Triggering the Jenkins GitHub Integration With Every Code Commit

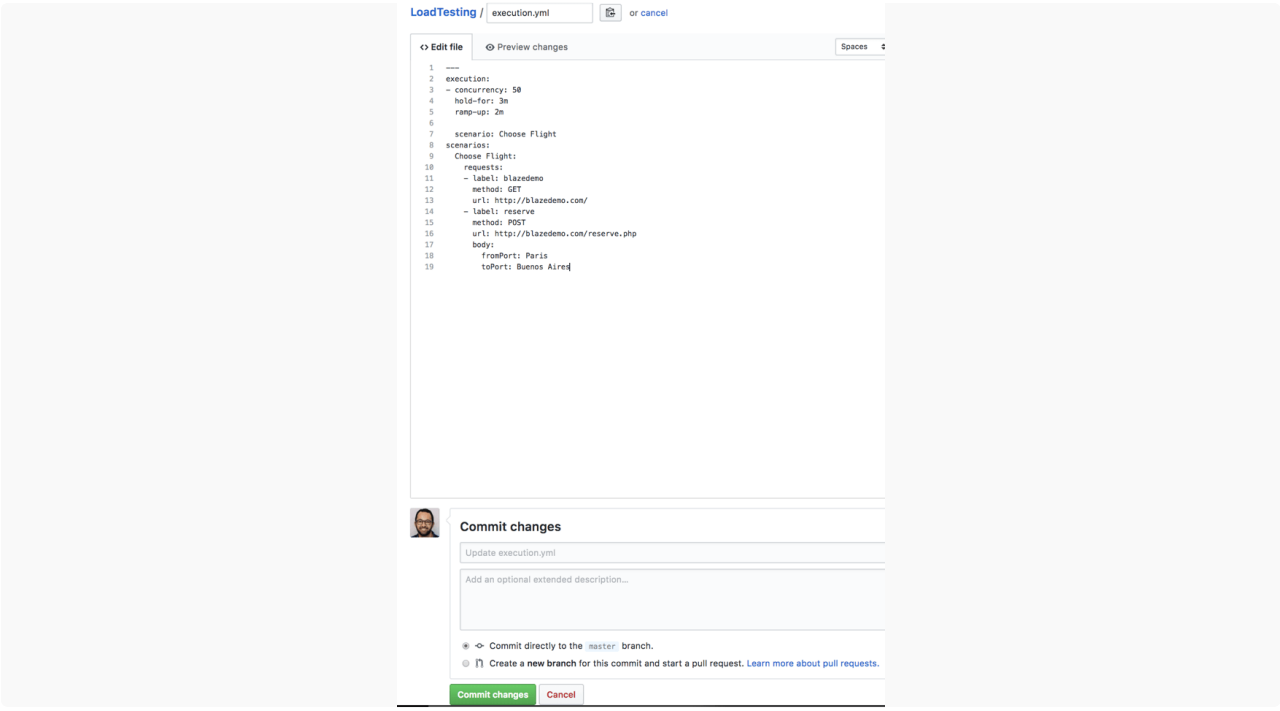
Step 10: Click on the **'Build'** tab, then click on _'Add build step'_and choose *'Execute shell'*.



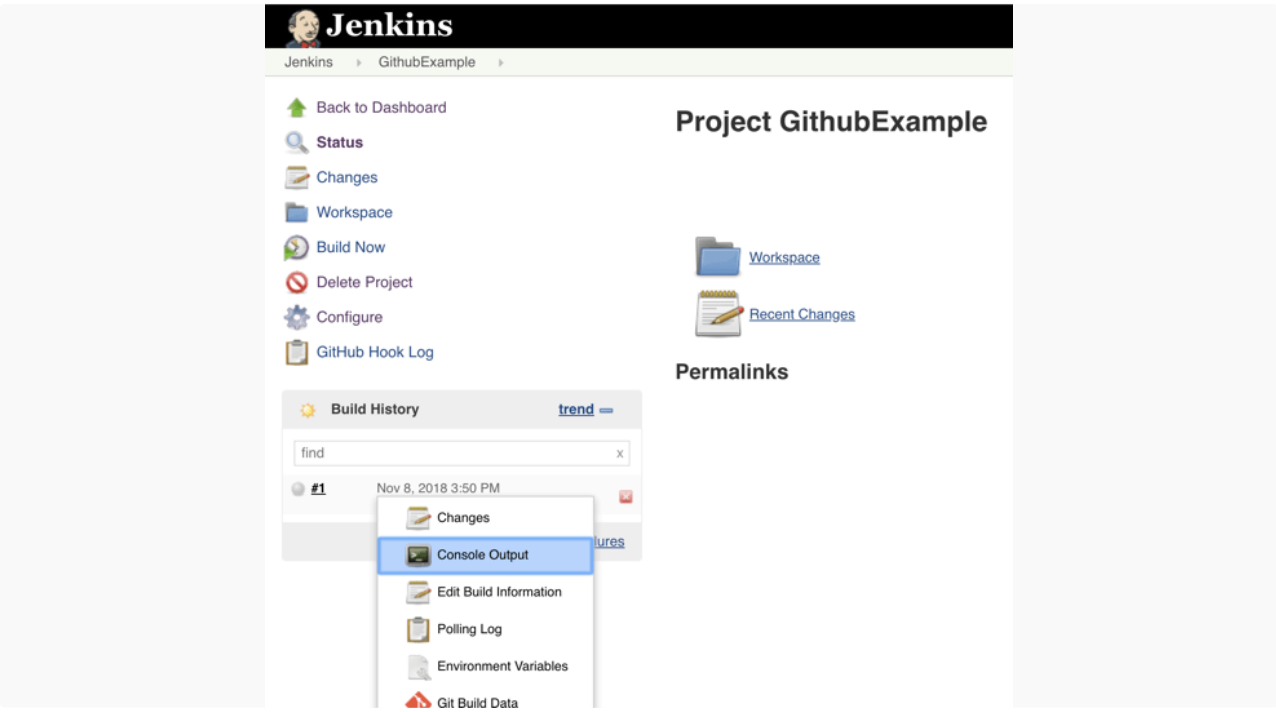
Step 11: To run a Taurus test, simply use the **'bzt'** command, followed by the name of your YML file and click on **'Save'.**



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.



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!