Experiment – 4

**Aim:** To perform Jenkins integration with Git.

**Theory:**

### **Jenkins: An Overview**

**1. What is Jenkins?**

* **Definition:** Jenkins is an open-source automation server that enables developers to build, test, and deploy their software applications.
* **Key Features:**
  + **Extensibility:** Supports plugins to integrate with various tools and services.
  + **Distributed Builds:** Can distribute build loads across multiple machines.
  + **User Interface:** Provides a web-based interface for job management and monitoring.

**2. Role in Software Development:**

* **Continuous Integration:** Jenkins helps automate the integration of code changes from multiple contributors into a shared repository.
* **Build Automation:** Automatically compiles and packages code, reducing manual effort.
* **Automated Testing:** Executes test suites automatically after builds, ensuring code quality.
* **Continuous Deployment:** Can automate deployment to production or staging environments, enabling rapid release cycles.

### **Git: An Overview**

**1. What is Git?**

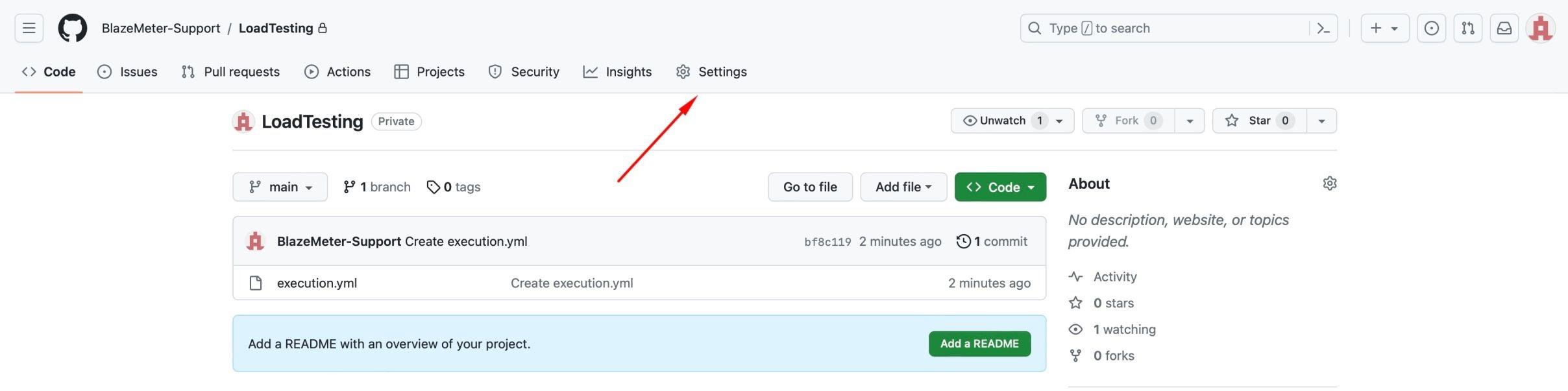
* **Definition:** Git is a distributed version control system designed to track changes in source code during software development.
* **Key Features:**
  + **Branching and Merging:** Supports lightweight branching, allowing developers to work on features in isolation.
  + **Distributed Architecture:** Each developer has a complete local copy of the repository, enabling offline work.
  + **Collaboration:** Facilitates collaboration among multiple developers, providing a robust mechanism for merging changes.

**2. Role in Software Development:**

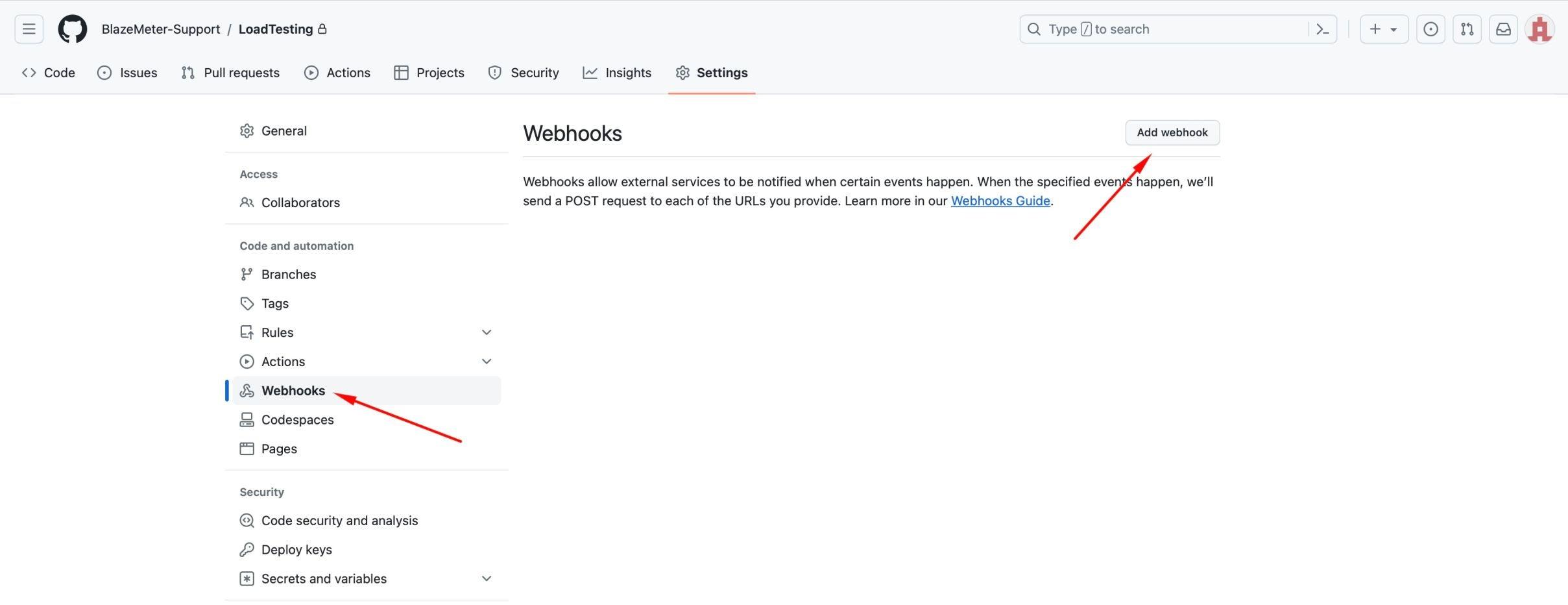
* **Version Control:** Tracks code changes, allowing teams to manage different versions of their projects effectively.
* **Collaboration:** Enables multiple developers to work simultaneously without conflicts.
* **History and Audit Trail:** Maintains a complete history of changes, allowing developers to review or revert changes as needed.

**Jenkins and Github integration:-**

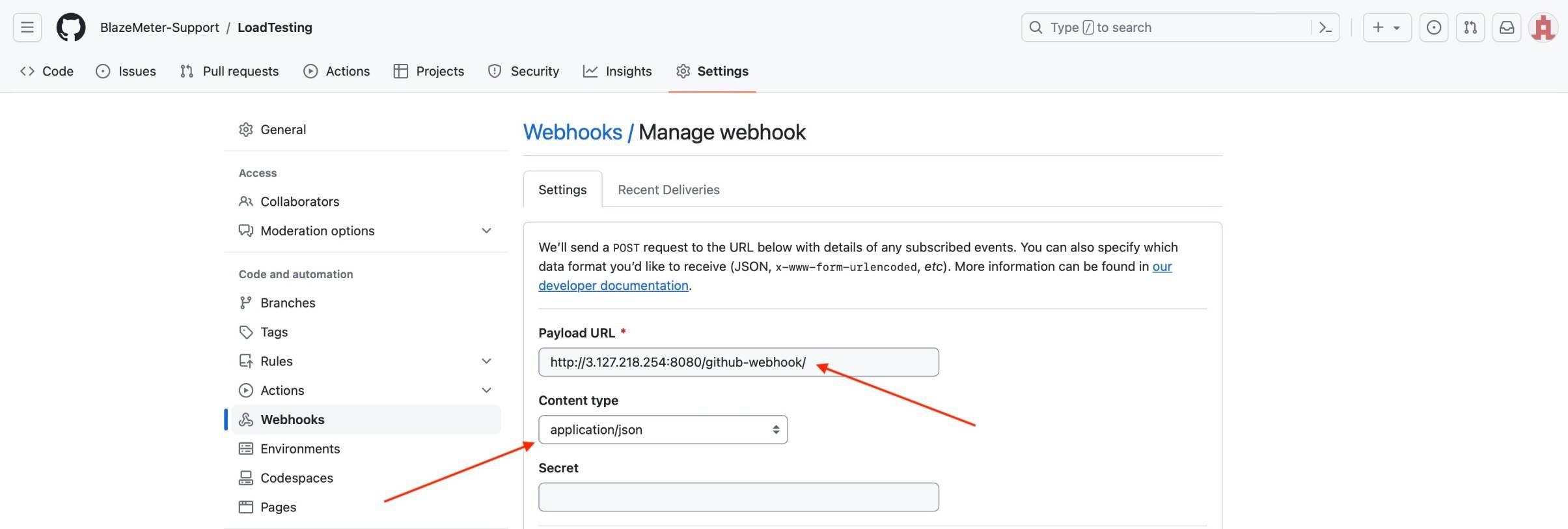
**Step 1:** go to your GitHub repository and click on **Settings**.



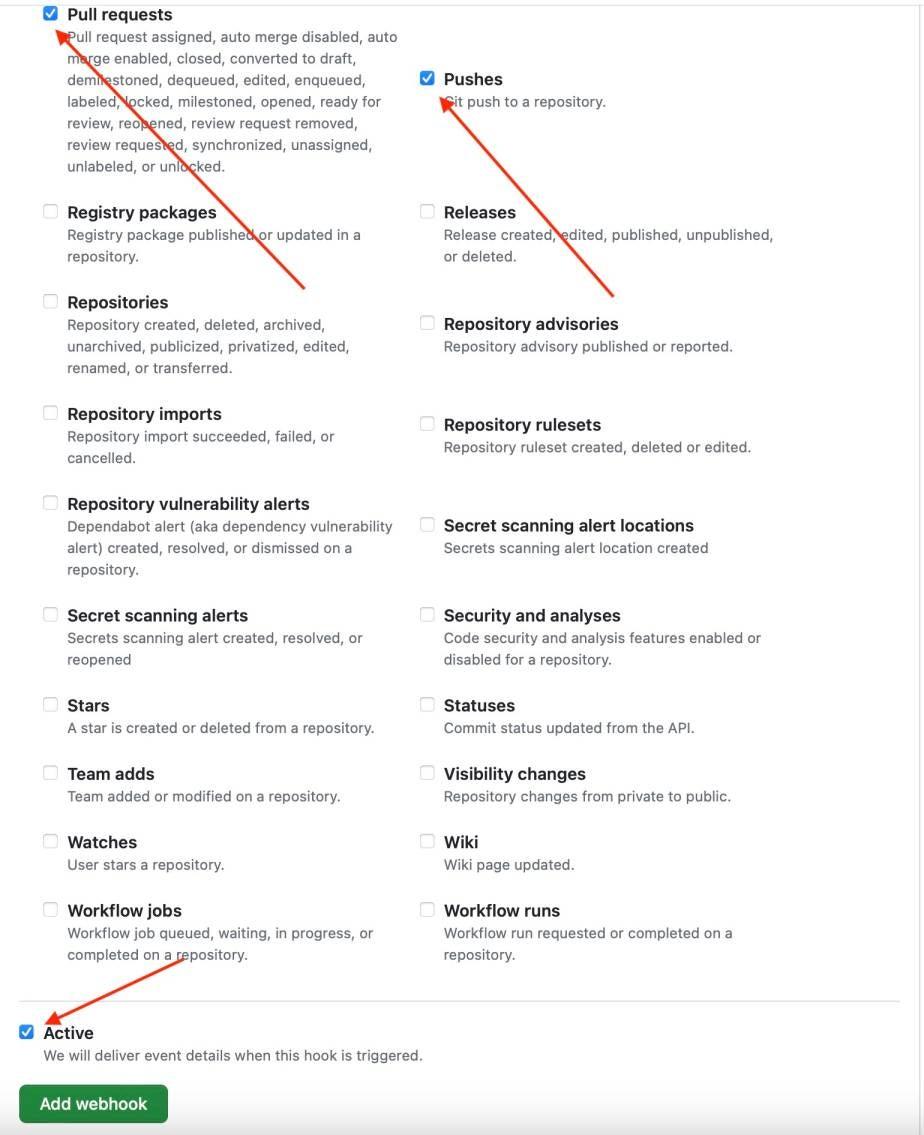
**Step 2:** Click on Webhooks and then click on **Add webhook**.



**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.

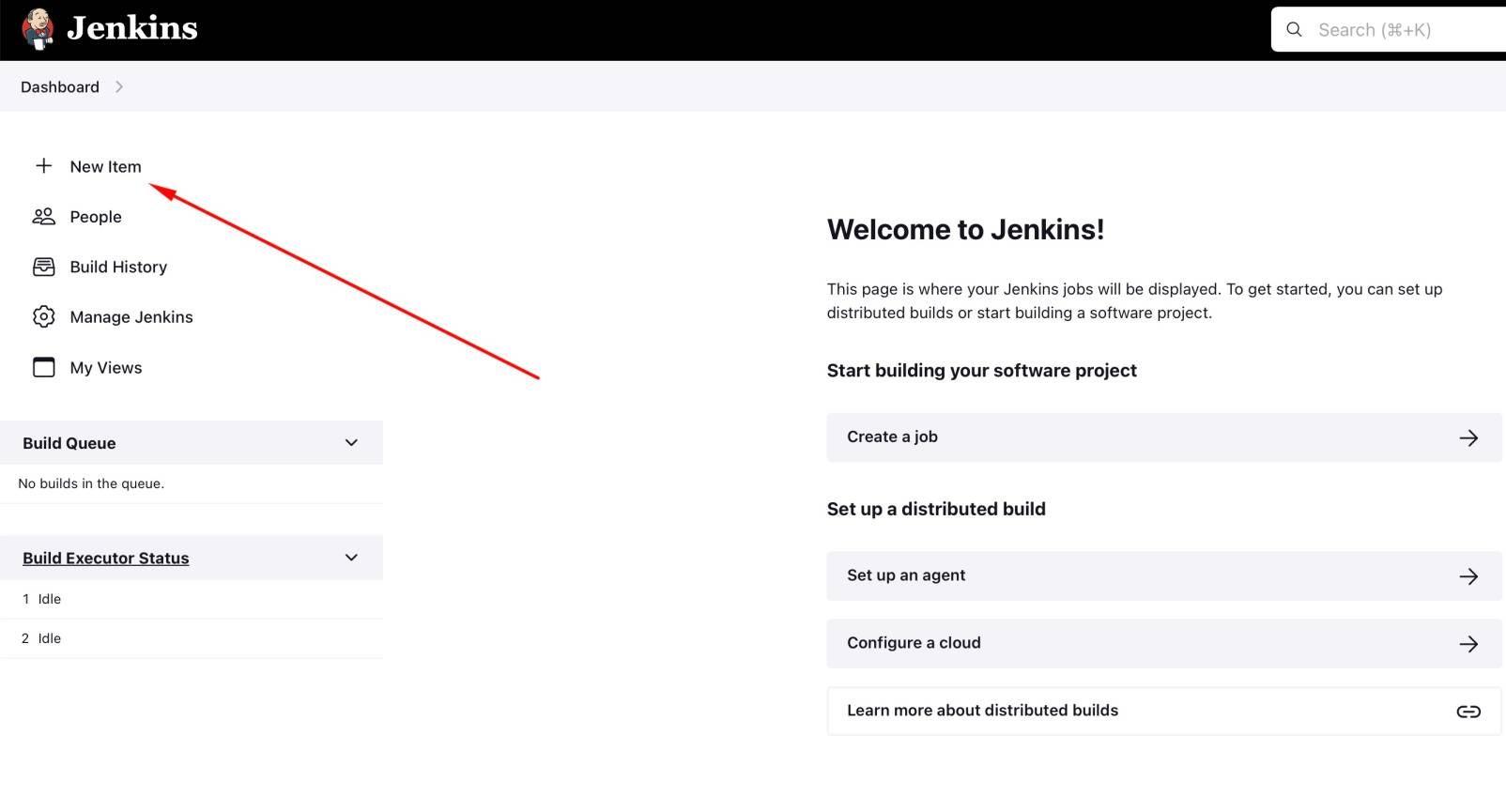


**Step 4:** In the page **Which events would you like to trigger this webhook?** choose ***Let me select individual events.*** Then, check ***Pull Requests*** and ***Pushes***. At the end of this option, make sure that the ***Active*** option is checked and click on ***Add webhook***.

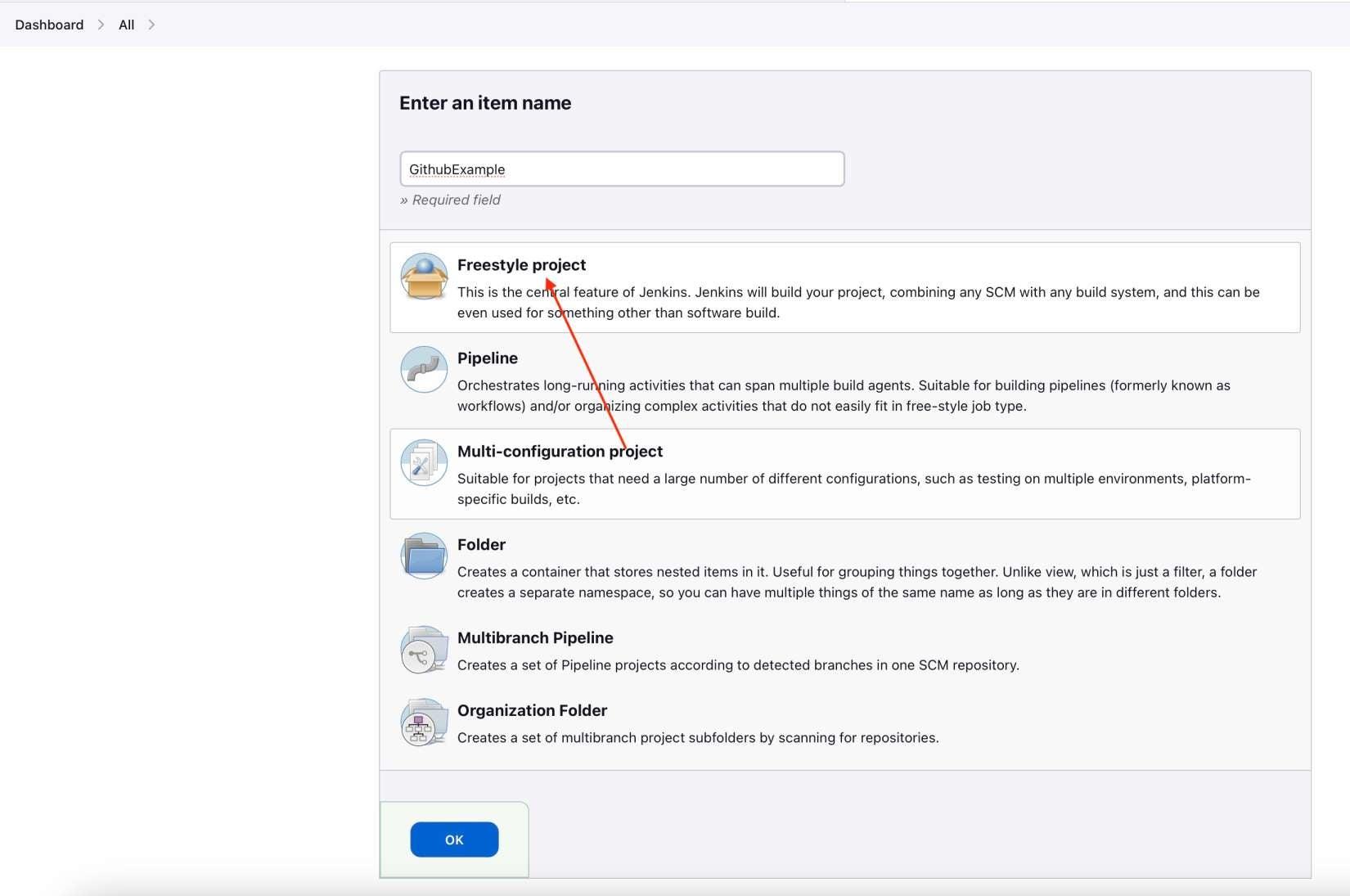


### **Configuring Jenkins**

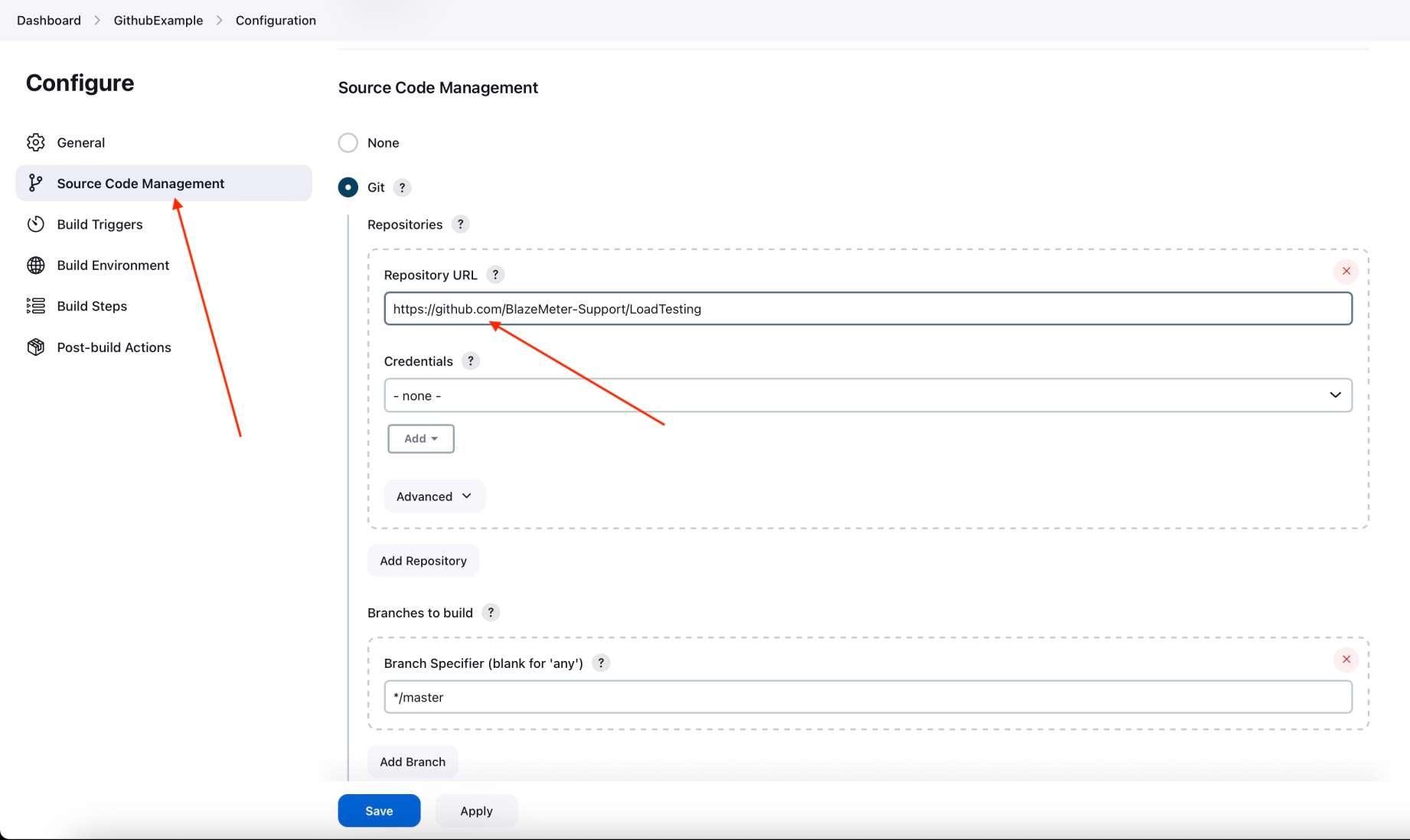
**Step 5:** In Jenkins, click on **New Item** to create a new project.



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

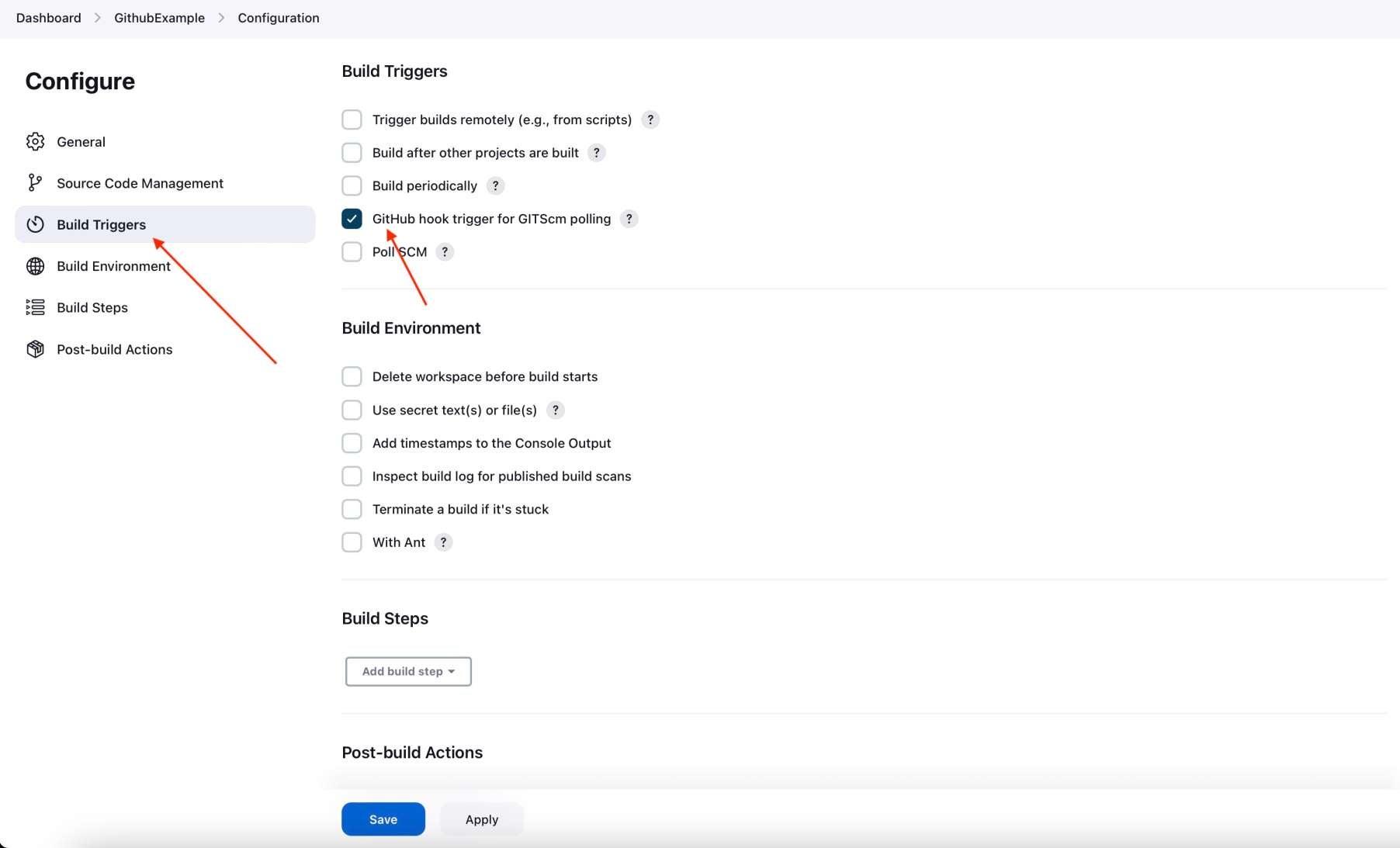
**

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



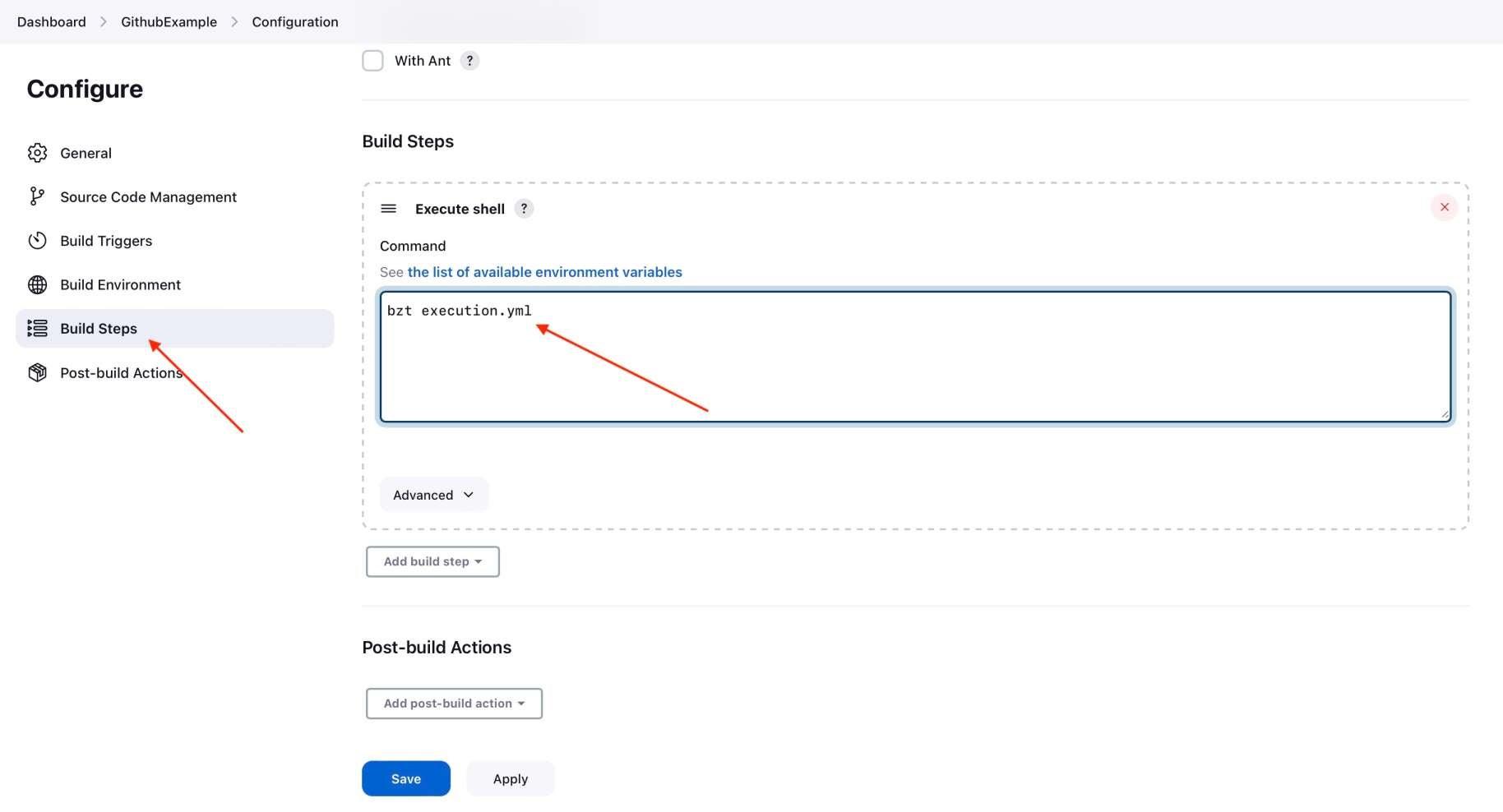
**Step 8:** Click on Git and paste your GitHub repository URL in the **Repository URL** field.

**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.



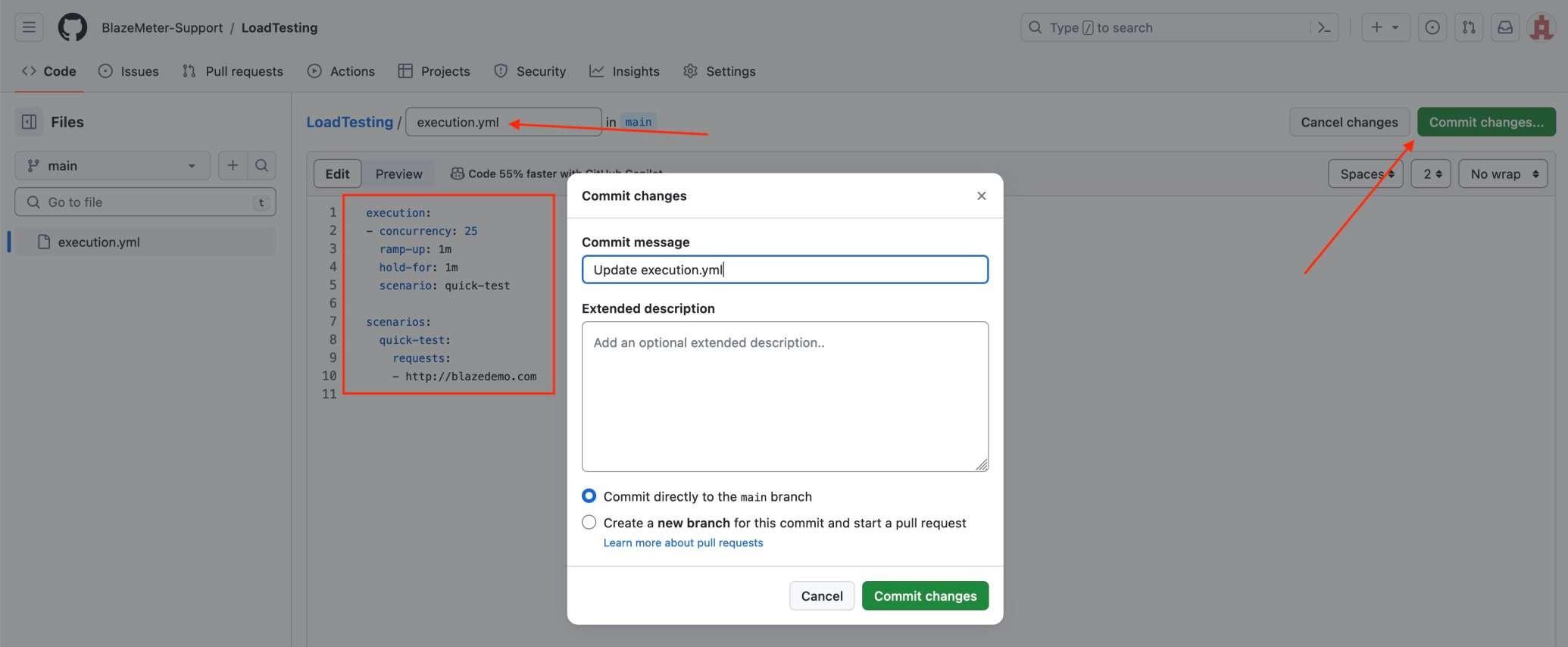
### **Triggering the GitHub Jenkins Integration With Every Code Commit**

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

**

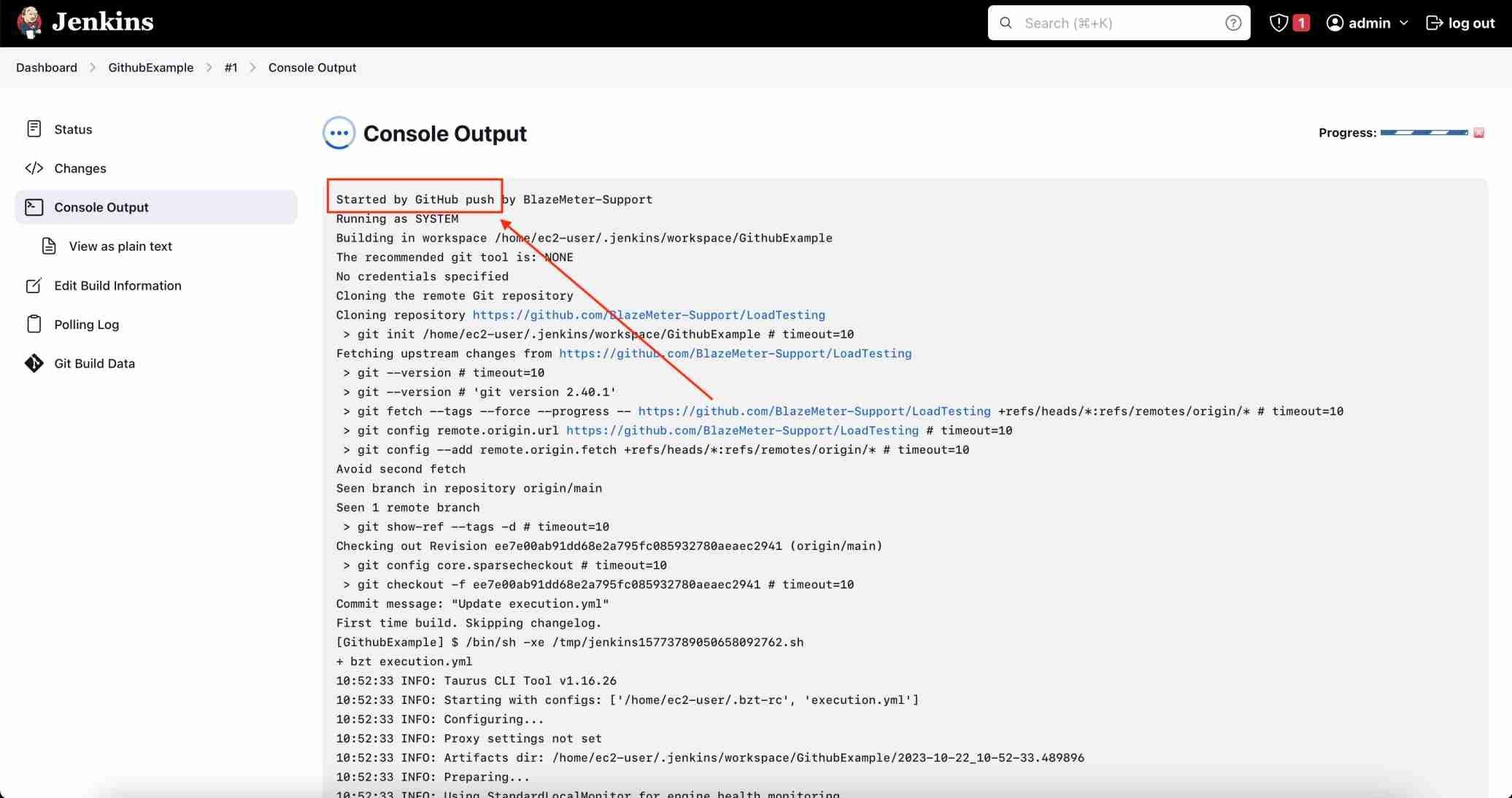
**Step 11:** To run a Taurus test, simply use the **bzt** command, followed by the name of your YML file and click on **Save**.

**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!



**Experiment:**