**GitHub Actions workflows**

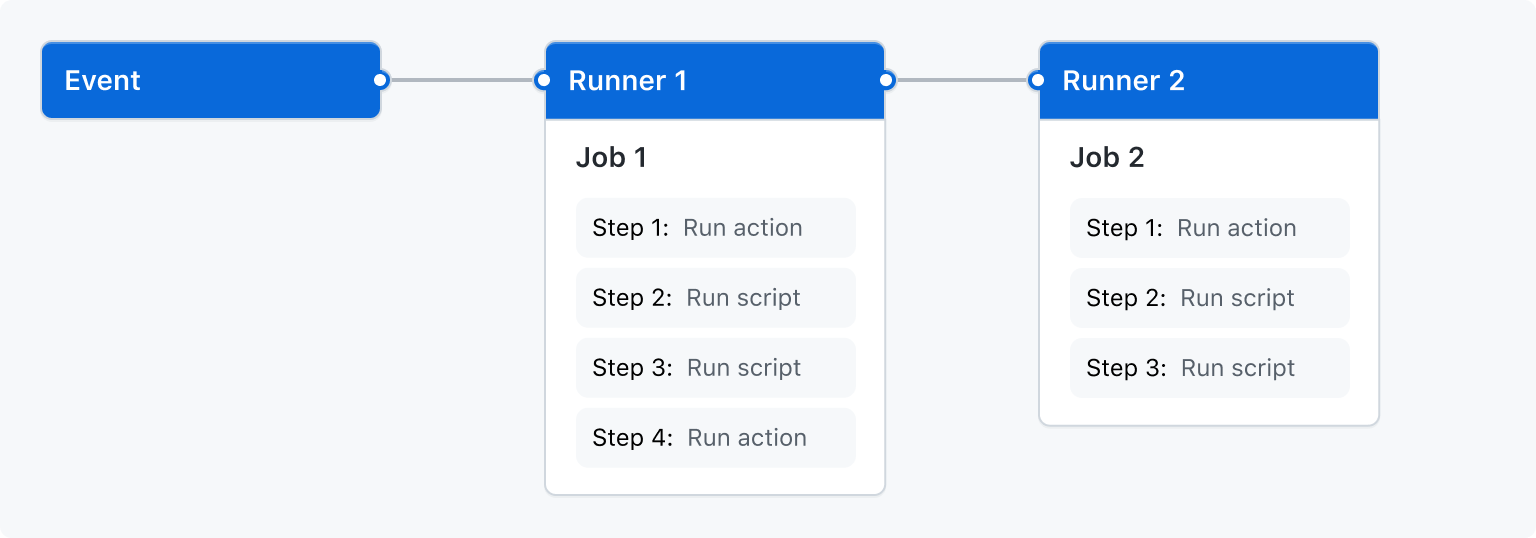
A workflow is a configurable automated process that will run one or more jobs. Workflows are defined by a YAML file checked in to your repository and will run when triggered by an event in your repository, or they can be triggered manually, or at a defined schedule.

Workflows are defined in the .github/workflows directory in a repository, and a repository can have multiple workflows, each of which can perform a different set of tasks. For example, you can have one workflow to build and test pull requests, another workflow to deploy your application every time a release is created, and still another workflow that adds a label every time someone opens a new issue.

## **Workflow basics**

A workflow must contain the following basic components:

1. One or more events that will trigger the workflow.
2. One or more jobs, each of which will execute on a runner machine and run a series of one or more steps.
3. Each step can either run a script that you define or run an action, which is a reusable extension that can simplify your workflow.



## **Triggering a workflow**

Workflow triggers are events that cause a workflow to run. These events can be:

* Events that occur in your workflow's repository
* Events that occur outside of GitHub and trigger a repository\_dispatch event on GitHub
* Scheduled times
* Manual

For example, you can configure your workflow to run when a push is made to the default branch of your repository, when a release is created, or when an issue is opened.

## **Workflow syntax**

Workflow are defined using YAML.

## **Create an example workflow**

GitHub Actions uses YAML syntax to define the workflow. Each workflow is stored as a separate YAML file in your code repository, in a directory named .github/workflows.

You can create an example workflow in your repository that automatically triggers a series of commands whenever code is pushed. In this workflow, GitHub Actions checks out the pushed code, installs the **[bats](https://www.npmjs.com/package/bats)**testing framework, and runs a basic command to output the bats version: bats -v.

In your repository, create the .github/workflows/ directory to store your workflow files.

In the .github/workflows/ directory, create a new file called learn-github-actions.yml and add the following code.

name: learn-github-actionson: [push]jobs:

check-bats-version:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v3

- uses: actions/setup-node@v3

with:

node-version: '14'

- run: npm install -g bats

- run: bats -v

Commit these changes and push them to your GitHub repository.

Your new GitHub Actions workflow file is now installed in your repository and will run automatically each time someone pushes a change to the repository.

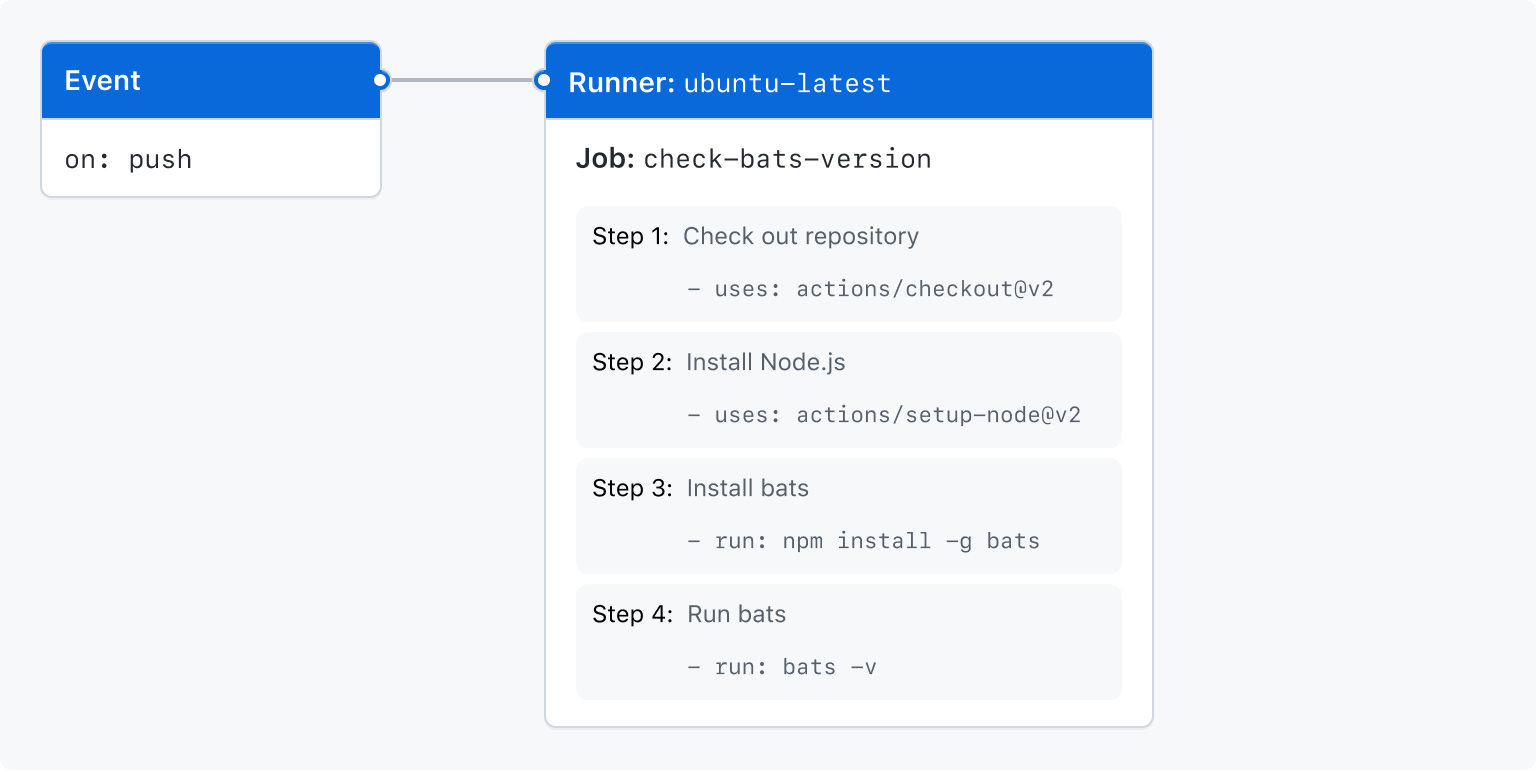
## **Understanding the workflow file**

To help you understand how YAML syntax is used to create a workflow file, this section explains each line of the introduction's example:

|  |  |
| --- | --- |
| name: learn-github-actions | Optional - The name of the workflow as it will appear in the Actions tab of the GitHub repository. |
| on: [push] | Specifies the trigger for this workflow. This example uses the push event, so a workflow run is triggered every time someone pushes a change to the repository or merges a pull request. This is triggered by a push to every branch. |
| jobs: | Groups together all the jobs that run in the learn-github-actions workflow. |
| check-bats-version: | Defines a job named check-bats-version. The child keys will define properties of the job. |
| runs-on: ubuntu-latest | Configures the job to run on the latest version of an Ubuntu Linux runner. This means that the job will execute on a fresh virtual machine hosted by GitHub. |
| steps: | Groups together all the steps that run in the check-bats-version job. Each item nested under this section is a separate action or shell script. |
| - uses: actions/checkout@v3 | The uses keyword specifies that this step will run v3 of the actions/checkout action. This is an action that checks out your repository onto the runner, allowing you to run scripts or other actions against your code (such as build and test tools). You should use the checkout action any time your workflow will run against the repository's code. |
| - uses: actions/setup-node@v3  with:  node-version: '14' | This step uses the actions/setup-node@v3 action to install the specified version of the Node.js (this example uses v14). This puts both the node and npm commands in your PATH. |
| - run: npm install -g bats | The run keyword tells the job to execute a command on the runner. In this case, you are using npm to install the bats software testing package. |
| - run: bats -v | Finally, you'll run the bats command with a parameter that outputs the software version. |

### **Visualizing the workflow file**

In this diagram, you can see the workflow file you just created and how the GitHub Actions components are organized in a hierarchy. Each step executes a single action or shell script. Steps 1 and 2 run actions, while steps 3 and 4 run shell scripts.

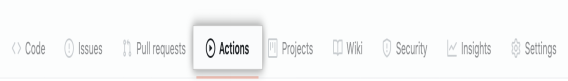


## **Viewing the activity for a workflow run**

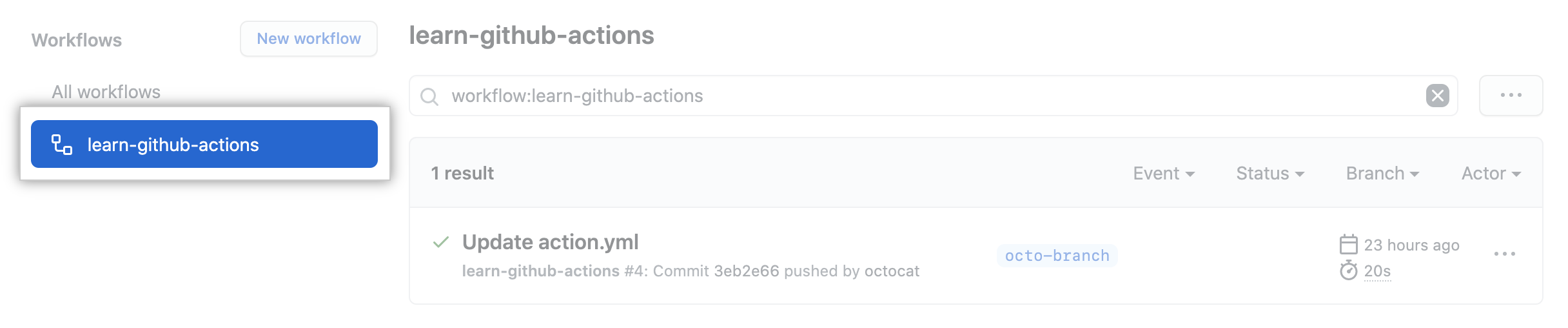
When your workflow is triggered, a workflow run is created that executes the workflow. After a workflow run has started, you can see a visualization graph of the run's progress and view each step's activity on GitHub.

On GitHub.com, navigate to the main page of the repository.

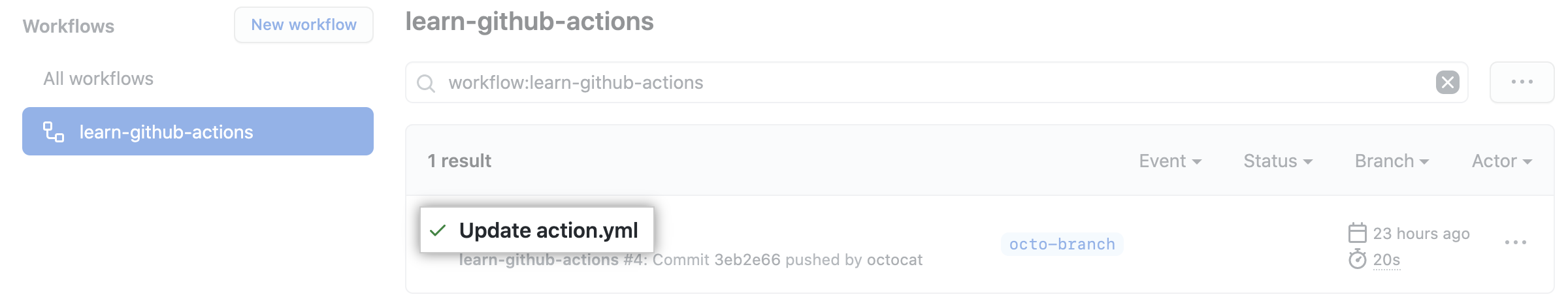
Under your repository name, click ****Actions****.



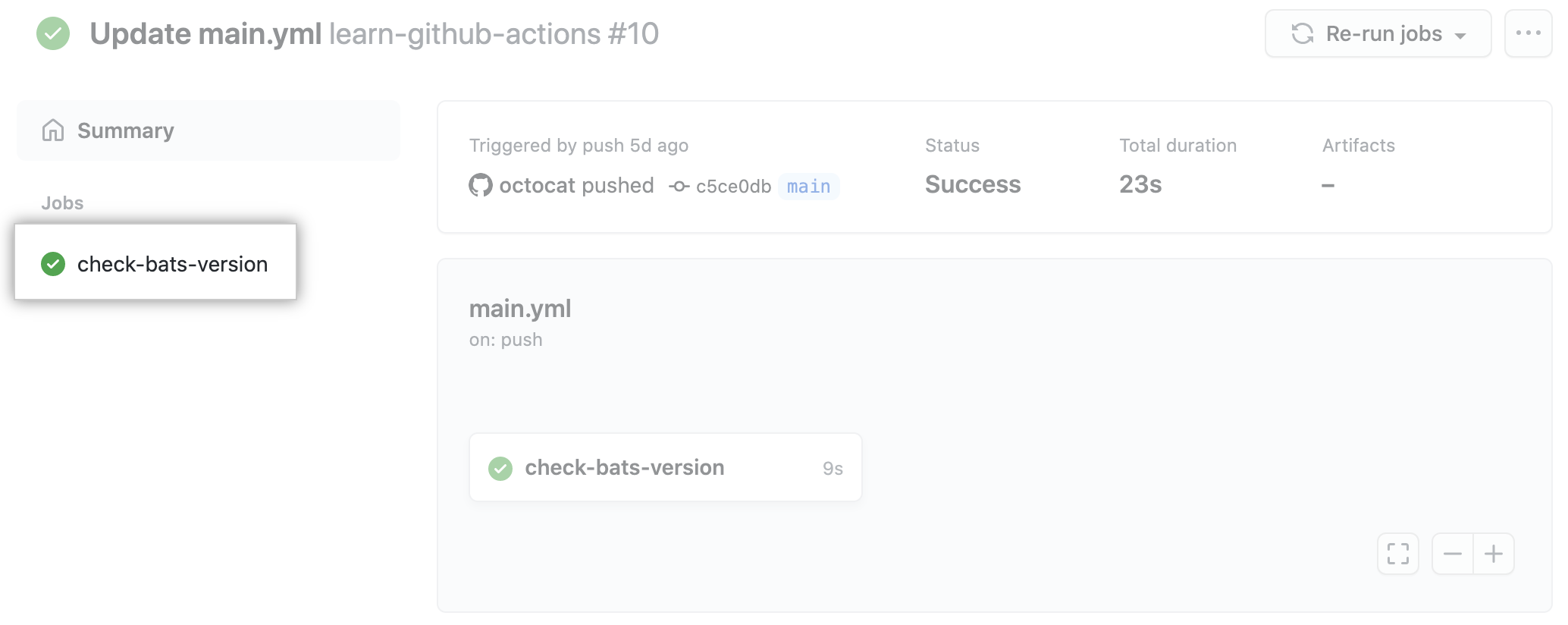
In the left sidebar, click the workflow you want to see.



Under "Workflow runs", click the name of the run you want to see.



Under ****Jobs**** or in the visualization graph, click the job you want to see.



View the results of each step.

