Quickstart for GitHub Actions

In this article

Try out the features of GitHub Actions in 5 minutes or less.

Note: GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the GitHub public roadmap.

Introduction

You only need a GitHub repository to create and run a GitHub Actions workflow. In this guide, you'll add a workflow that demonstrates some of the essential features of GitHub Actions.

The following example shows you how GitHub Actions jobs can be automatically triggered, where they run, and how they can interact with the code in your repository.

Creating your first workflow

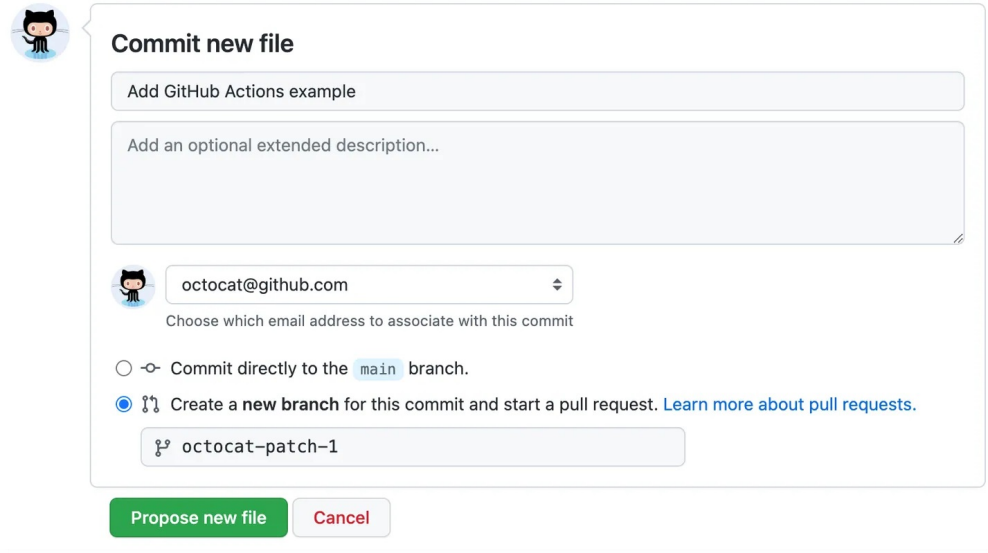
1. Create a `.github/workflows` directory in your repository on GitHub if this directory does not already exist.
2. In the `.github/workflows` directory, create a file named `github-actions-demo.yml`. For more information, see "Creating new files."
3. Copy the following YAML contents into the `github-actions-demo.yml` file:

YAML

```yaml
name: GitHub Actions Demo
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo " The job was automatically triggered by a ${{ github.event_r
      - run: echo " This job is now running on a ${{ runner.os }} server hosted
      - run: echo " The name of your branch is ${{ github.ref }} and your repos
```

```
      - name: Check out repository code
        uses: actions/checkout@v2
      - run: echo " The ${{ github.repository }} repository has been cloned to
      - run: echo "□ The workflow is now ready to test your code on the runner.
      - name: List files in the repository
        run: |
          ls ${{ github.workspace }}
      - run: echo " This job's status is ${{ job.status }}."
```

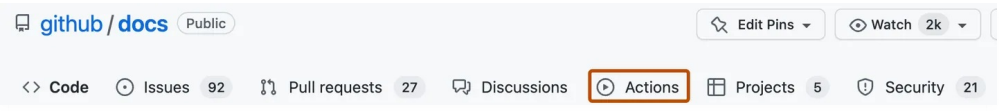
4. Scroll to the bottom of the page and select **Create a new branch for this commit and start a pull request**. Then, to create a pull request, click **Propose new file**.



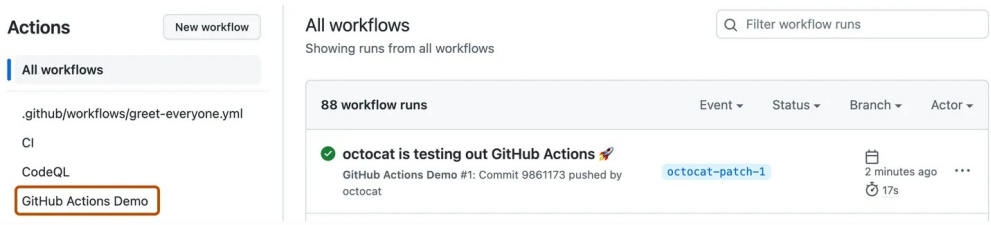
Committing the workflow file to a branch in your repository triggers the `push` event and runs your workflow.

Viewing your workflow results

1. On your GitHub Enterprise Server instance, navigate to the main page of the repository.
2. Under your repository name, click **Actions**.

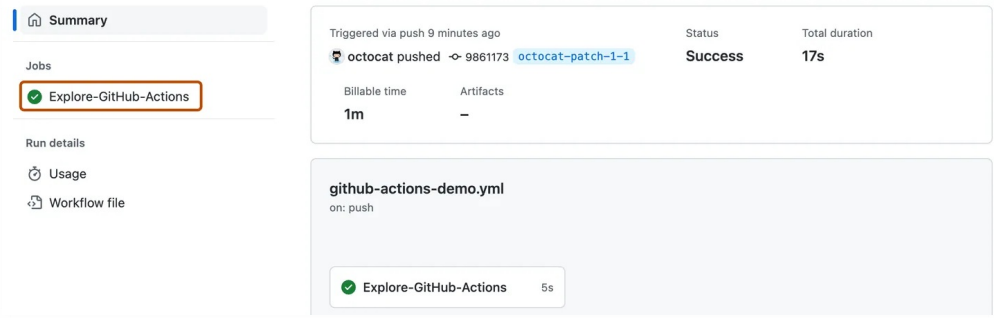


3. In the left sidebar, click the workflow you want to display, in this example "GitHub Actions Demo."

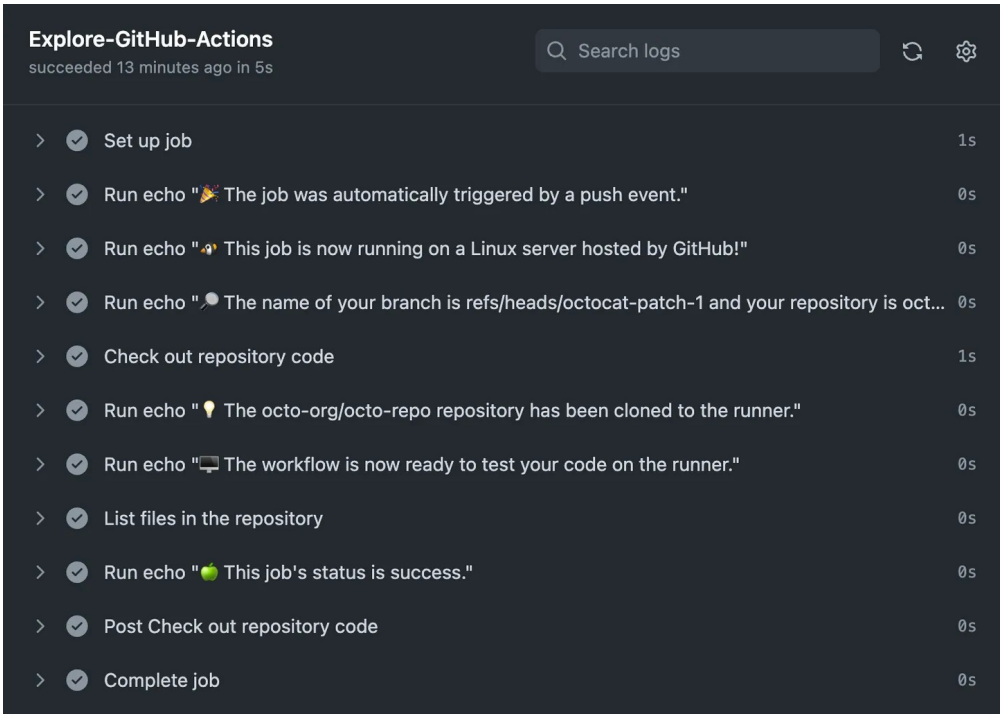


4. From the list of workflow runs, click the name of the run you want to see, in this example "USERNAME is testing out GitHub Actions."

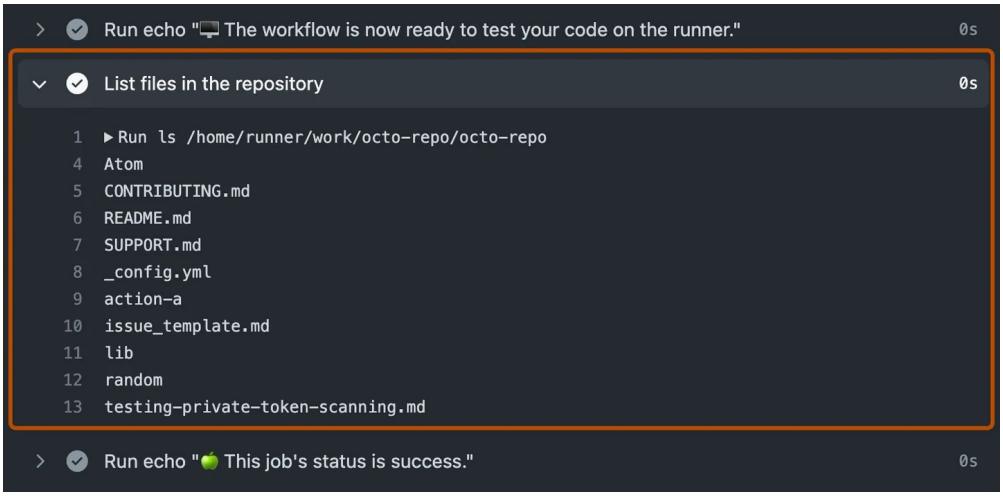
5. In the left sidebar of the workflow run page, under **Jobs**, click the **Explore-GitHub-Actions** job.



6. The log shows you how each of the steps was processed. Expand any of the steps to view its details.



For example, you can see the list of files in your repository:



The example workflow you just added is triggered each time code is pushed to the branch, and shows you how GitHub Actions can work with the contents of your

repository. For an in-depth tutorial, see "Understanding GitHub Actions."

More starter workflows

GitHub provides preconfigured starter workflows that you can customize to create your own continuous integration workflow. GitHub Enterprise Server analyzes your code and shows you CI starter workflows that might be useful for your repository. For example, if your repository contains Node.js code, you'll see suggestions for Node.js projects. You can use starter workflows as a starting place to build your custom workflow or use them as-is.

You can browse the full list of starter workflows in the `actions/starter-workflows` repository on your GitHub Enterprise Server instance.

Next steps

GitHub Actions can help you automate nearly every aspect of your application development processes. Ready to get started? Here are some helpful resources for taking your next steps with GitHub Actions:

- For continuous integration (CI) workflows to build and test your code, see "Automating builds and tests."
- For building and publishing packages, see "Publishing packages."
- For deploying projects, see "Deployment."
- For automating tasks and processes on GitHub, see "Managing issues and pull requests."
- For examples that demonstrate more complex features of GitHub Actions, including many of the above use cases, see "Examples." You can see detailed examples that explain how to test your code on a runner, access the GitHub CLI, and use advanced features such as concurrency and test matrices.