Kevin Vincent Loachamin Almeida A00382106

hello-github-actions

**Step 1: Create a workflow file**

**What is *GitHub Actions*?**: GitHub Actions is a flexible way to automate nearly every aspect of your team's software workflow. You can automate testing, continuously deploy, review code, manage issues and pull requests, and much more. The best part, these workflows are stored as code in your repository and easily shared and reused across teams. To learn more, check out these resources:

* The GitHub Actions feature page, see [GitHub Actions](https://github.com/features/actions).
* The "GitHub Actions" user documentation, see [GitHub Actions](https://docs.github.com/actions).

**What is a *workflow*?**: A workflow is a configurable automated process that will run one or more jobs. Workflows are defined in special files in the .github/workflows directory and they execute based on your chosen event. For this exercise, we'll use a pull\_request event.

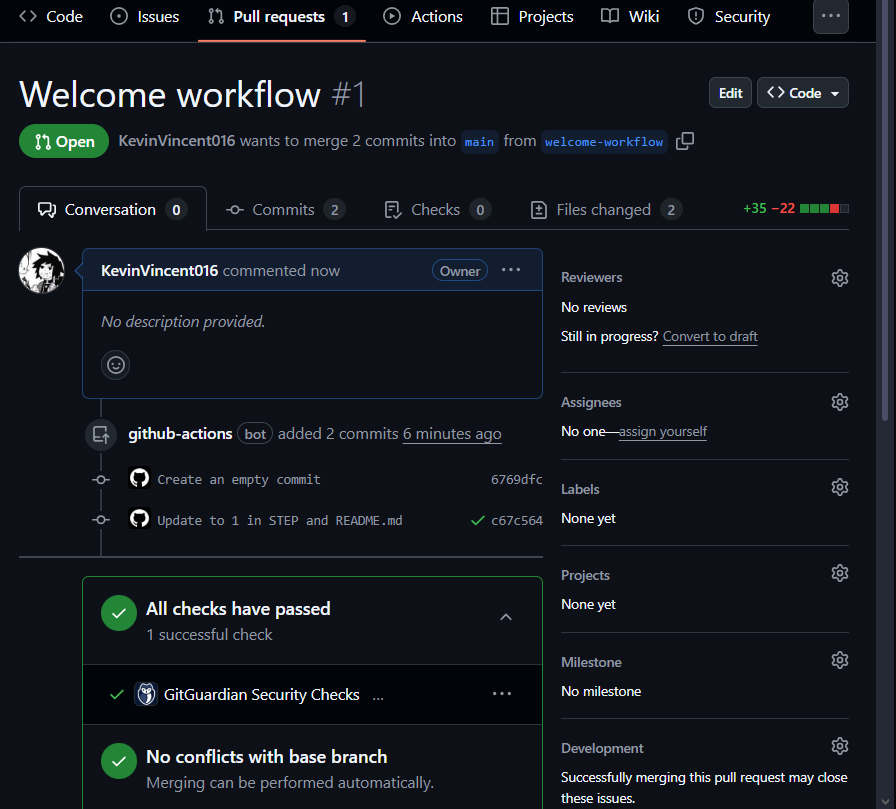
* To read more about workflows, jobs, and events, see "[Understanding GitHub Actions](https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions)".
* If you want to learn more about the pull\_request event before using it, see "[pull\_request](https://docs.github.com/en/developers/webhooks-and-events/webhooks/webhook-events-and-payloads#pull_request)".

To get you started, we ran an Actions workflow in your new repository that, among other things, created a branch for you to work in, called welcome-workflow.

**⌨️ Activity: Create a workflow file**

1. Open a new browser tab, and navigate to this same repository. Then, work on the steps in your second tab while you read the instructions in this tab.
2. Create a pull request. This will contain all of the changes you'll make throughout this part of the course.

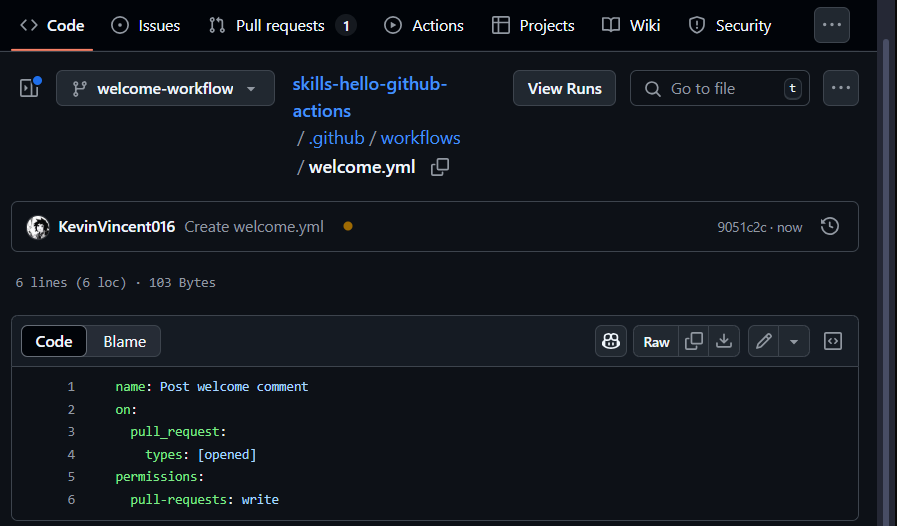
Click the **Pull Requests** tab, click **New pull request**, set base: main and compare:welcome-workflow, click **Create pull request**, then click **Create pull request** again.



1. Navigate to the **Code** tab.
2. From the **main** branch dropdown, click on the **welcome-workflow** branch.
3. Navigate to the .github/workflows/ folder, then select **Add file** and click on **Create new file**.
4. In the **Name your file** field, enter welcome.yml.
5. Add the following content to the welcome.yml file:
6. name: Post welcome comment
7. on:
8. pull\_request:
9. types: [opened]
10. permissions:

pull-requests: write

1. To commit your changes, click **Commit changes**.



**Step 2: Add a job to your workflow file**

Here's what the entries in the welcome.yml file, on the welcome-workflow branch, mean:

* name: Post welcome comment gives your workflow a name. This name will appear in the Actions tab of your repository.
* on: pull\_request: types: [opened] indicates that your workflow will execute whenever someone opens a pull request in your repository.
* permissions assigns the workflow permissions to operate on the repository
* pull-requests: write gives the workflow permission to write to pull requests. This is needed to create the welcome comment.

Next, we need to specify jobs to run.

**What is a *job*?**: A job is a set of steps in a workflow that execute on the same runner (a runner is a server that runs your workflows when triggered). Workflows have jobs, and jobs have steps. Steps are executed in order and are dependent on each other. You'll add steps to your workflow later in the course. To read more about jobs, see "[Jobs](https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions#jobs)".

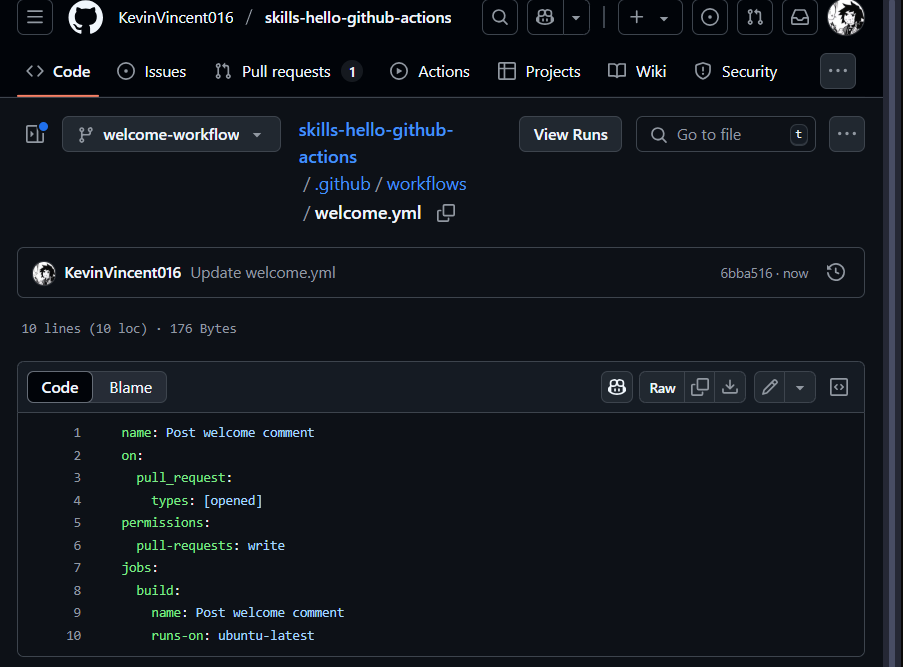
In the following activity, you'll add a "build" job to your workflow. You'll specify ubuntu-latest as the fastest, and cheapest, job runner available. If you want to read more about why we'll use that runner, see the code explanation for the line runs-on: ubuntu-latest in the "[Understanding the workflow file](https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions#understanding-the-workflow-file)" article.

**⌨️ Activity: Add a job to your workflow file**

1. In a separate browser tab, make sure you are on the welcome-workflow branch and open your .github/workflows/welcome.yml file.
2. Edit the file and update its contents to:
3. name: Post welcome comment
4. on:
5. pull\_request:
6. types: [opened]
7. permissions:
8. pull-requests: write
9. jobs:
10. build:
11. name: Post welcome comment

runs-on: ubuntu-latest

1. Click **Commit changes** in the top right of the workflow editor.
2. Type a commit message and commit your changes directly to the welcome-workflow branch.



**Step 3: Add a step to your workflow file**

*Nice work adding a job to your workflow! 💃*

Workflows have jobs, and jobs have steps. So now we'll add a step to your workflow.

**What are *steps*?**: Actions steps run - in the order they are specified, from the top down - when a workflow job is processed. Each step must pass for the next step to run.

Each step consists of either a shell script that's executed, or a reference to an action that's run. When we talk about an action (with a lowercase "a") in this context, we mean a reusable unit of code. You can find out about actions in "[Finding and customizing actions](https://docs.github.com/en/actions/learn-github-actions/finding-and-customizing-actions)," but for now we'll use a shell script in our workflow step.

Update your workflow to make it post a comment on new pull requests. It will do this using a [bash](https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29) script and [GitHub CLI](https://cli.github.com/).

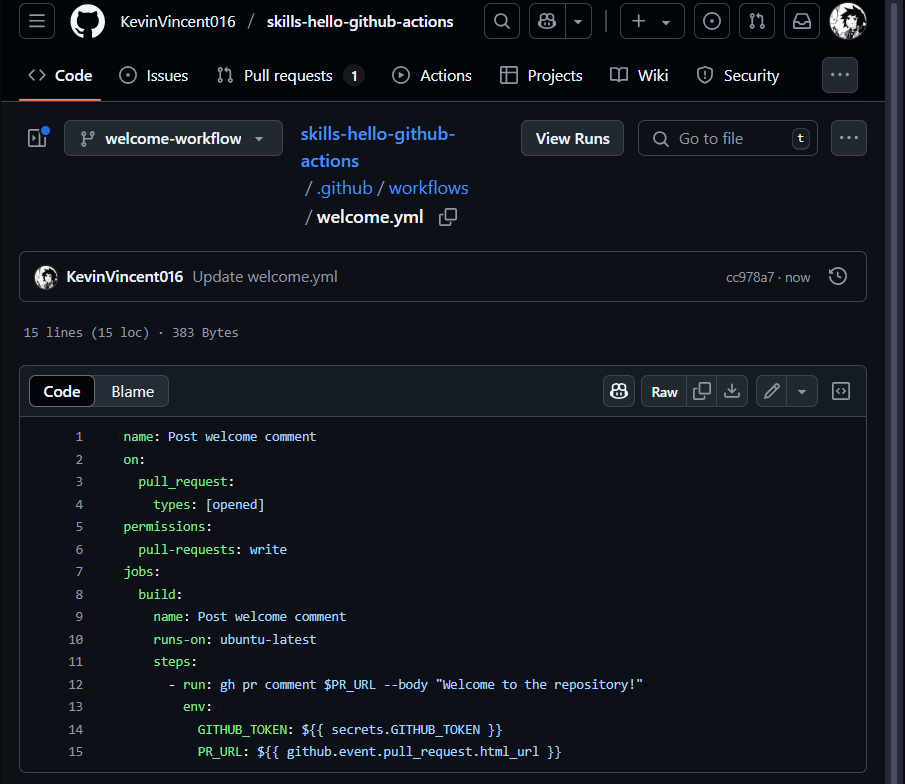
**⌨️ Activity: Add a step to your workflow file**

1. Still working on the welcome-workflow branch, open your welcome.yml file.
2. Update the contents of the file to:
3. name: Post welcome comment
4. on:
5. pull\_request:
6. types: [opened]
7. permissions:
8. pull-requests: write
9. jobs:
10. build:
11. name: Post welcome comment
12. runs-on: ubuntu-latest
13. steps:
14. - run: gh pr comment $PR\_URL --body "Welcome to the repository!"
15. env:
16. GITHUB\_TOKEN: ${{ secrets.GITHUB\_TOKEN }}

PR\_URL: ${{ github.event.pull\_request.html\_url }}

**Note:** The step you've added uses GitHub CLI (gh) to add a comment when a pull request is opened. To allow GitHub CLI to post a comment, we set the GITHUB\_TOKEN environment variable to the value of the GITHUB\_TOKEN secret, which is an installation access token, created when the workflow runs. For more information, see "[Automatic token authentication](https://docs.github.com/en/actions/security-guides/automatic-token-authentication)." We set the PR\_URL environment variable to the URL of the newly created pull request, and we use this in the gh command.

1. Click **Commit changes** in the top right of the workflow editor.
2. Type your commit message and commit your changes directly to your branch.



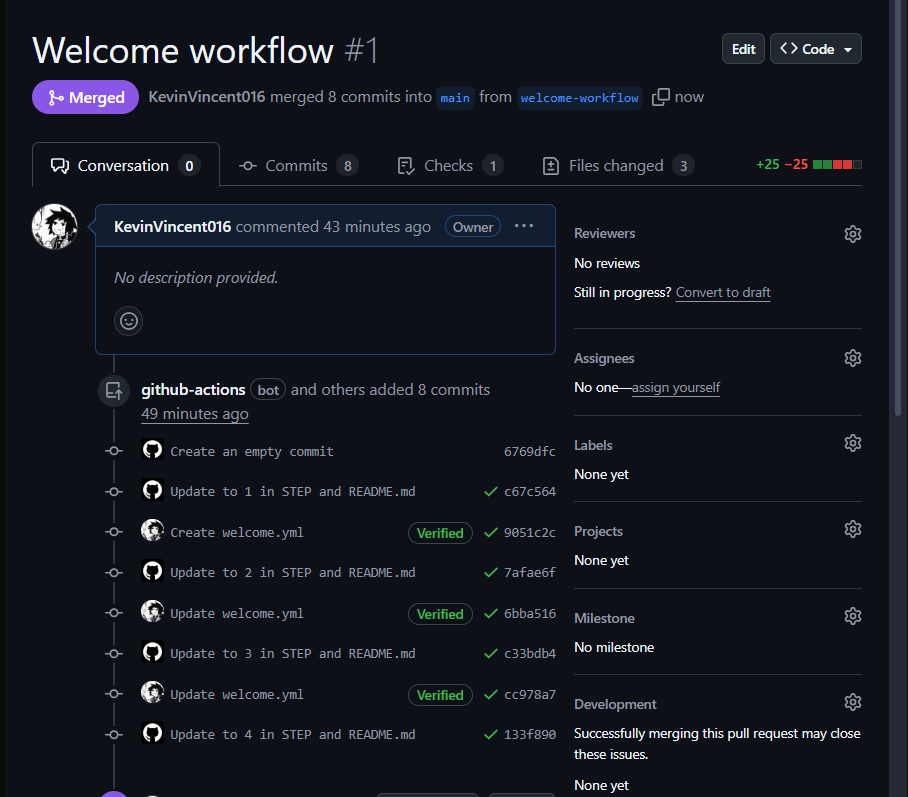
**tep 4: Merge your workflow file**

*You're now able to write and run an Actions workflow! ✨*

Merge your changes so the action will be a part of the main branch.

**⌨️ Activity: Merge your workflow file**

1. In your repo, click on the **Pull requests** tab.
2. Click on the pull request you created in step 1.
3. Click **Merge pull request**, then click **Confirm merge**.
4. Optionally, click **Delete branch** to delete your welcome-workflow branch.

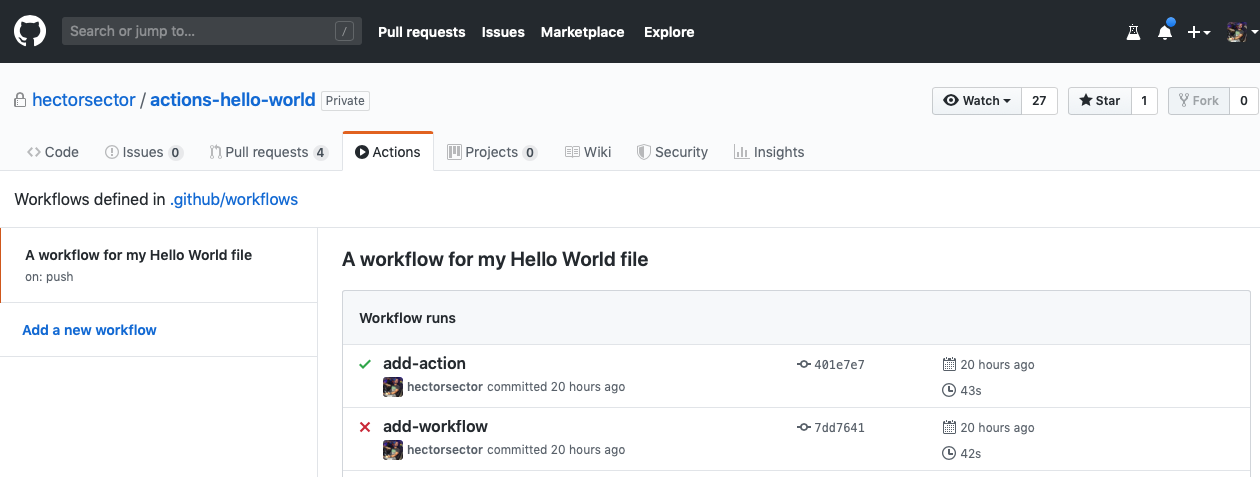


**Step 5: Trigger the workflow**

*You've now added a fully functioning workflow to your repository! 😄*

The shell script in the workflow will run whenever a new pull request is opened.

**Seeing your *action* in action**: The status of each workflow run that's triggered is shown in the pull request before it's merged: look for **All checks have passed** when you try out the steps below. You can also see a list of all the workflows that are running, or have finished running, in the **Actions** tab of your repository. From there, you can click on each workflow run to view more details and access log files.

[](https://user-images.githubusercontent.com/16547949/62388049-4e64e600-b52a-11e9-8bf5-db0c5452360f.png)

**⌨️ Activity: Trigger the workflow**

1. Make a new branch named test-workflow.
2. Make a change, such as adding an emoji to your README.md file, and commit the change directly to your new branch.+
3. In the **Pull requests** tab, create a pull request that will merge test-workflow into main.
4. Watch the workflow running in the checks section of the pull request.
5. Notice the comment that the workflow adds to the pull request.

