

Lesson 10 Demo 01

Creating and Executing a Basic GitHub Actions Workflow

Objective: To create and execute a basic GitHub Actions workflow for automated testing and deployment, enhancing the efficiency and reliability of project development processes

Tools required: GitHub

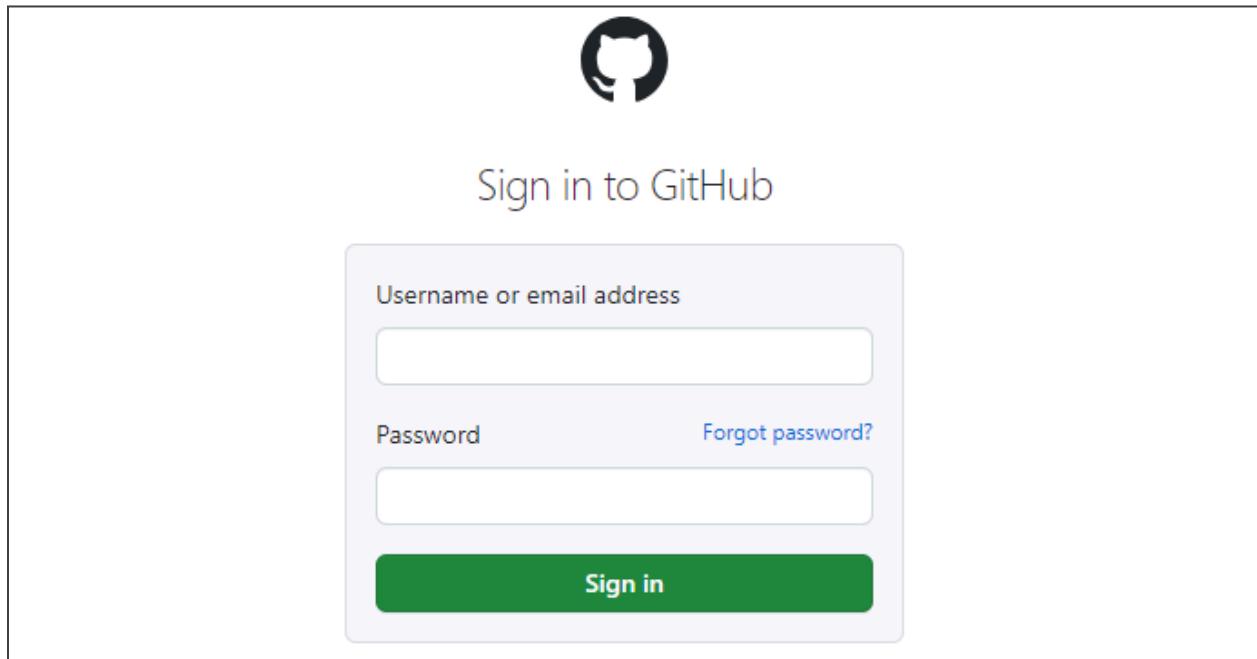
Prerequisites: None

Steps to be followed:

1. Create a new GitHub repository
2. Create and execute a new workflow file

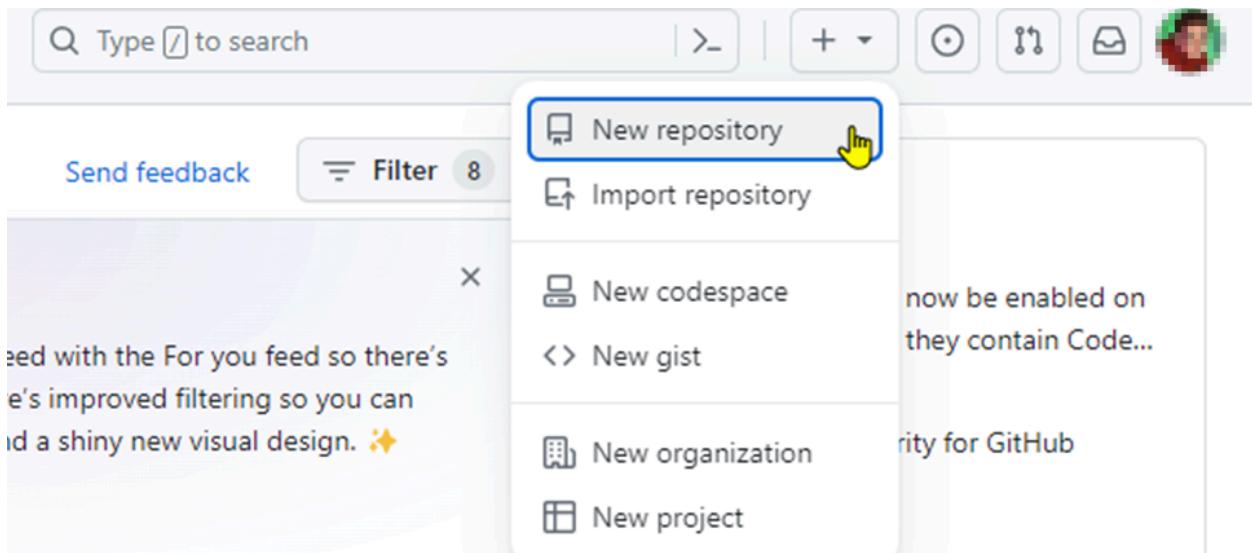
Step 1: Create a new GitHub repository

1.1 Open the browser in your lab, go to github.com, and log in to your account



Note: If you do not have a GitHub account, visit the official website at <https://github.com/signup> and create a new account

1.2 Click on the + icon from the upper-right corner of the page and select **New repository** from the drop-down menu



1.3 Enter the name and description for the GitHub repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

 /

Your new repository will be created as lesson-end-project.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-giggle](#) ?

Description (optional)

This is the lesson-end project for this lesson.

1.4 Choose **Public** for the repository type

Owner * Repository name *

 /

Your new repository will be created as lesson-end-project.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [fuzzy-giggle](#) ?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

 **Private**
You choose who can see and commit to this repository.

1.5 Select **Initialize this repository with a README** to include a README file for the repository

-  **Public**
Anyone on the internet can see this repository. You choose who can commit.
-  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- Add a README file**
 This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None ▾

1.6 Click on the **Create repository** button

Initialize this repository with:

Add a README file
 This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

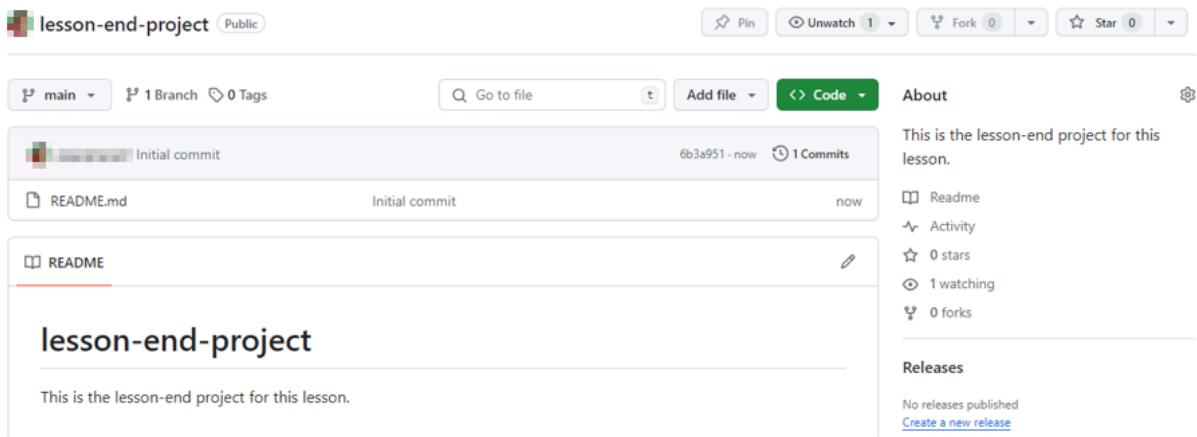
License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

ⓘ You are creating a public repository in your personal account.

Create repository

The screenshot shows the GitHub repository page for 'lesson-end-project'. The repository is public and has 1 branch and 0 tags. It contains a README.md file with the content 'Initial commit'. The repository has 1 commit, 0 forks, and 0 stars. There are sections for About, Releases, and Packages.

The remote GitHub repository is created.

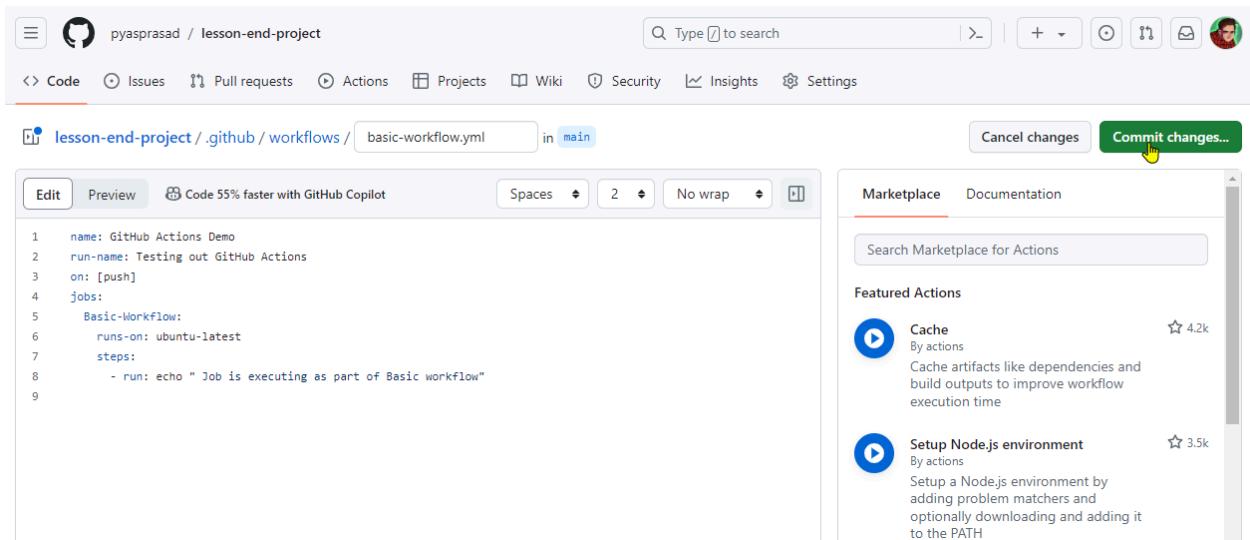
Step 2: Create and execute a new workflow file

2.1 Navigate to the **Actions** tab and click on **set up a workflow yourself** to create a **.github/workflows** directory

The screenshot shows a GitHub repository named "pyasprasad / lesson-end-project". The "Actions" tab is selected. A prominent call-to-action button "Set up a workflow yourself" is highlighted with a yellow arrow. Below it, there's a search bar for workflows and a section titled "Suggested for this repository" featuring a "Simple workflow" card.

2.2 Create a new workflow file **basic-workflow.yml** with the below code, then click on **Commit changes**:

```
name: GitHub Actions Demo
run-name: Testing out GitHub Actions
on: [push]
jobs:
  Basic-Workflow:
    runs-on: ubuntu-latest
    steps:
      - run: echo " Job is executing as part of Basic workflow"
```

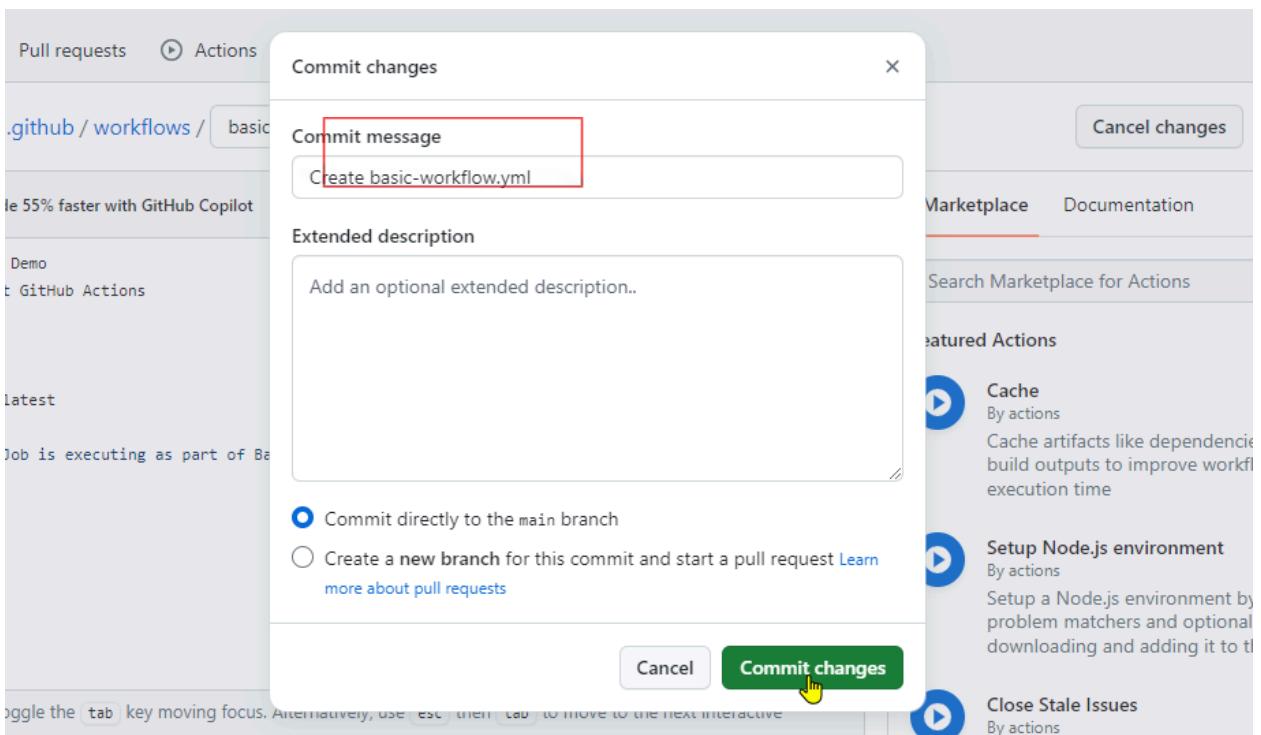


A screenshot of the GitHub Actions workflow editor. The file being edited is `basic-workflow.yml` in the `main` branch of the repository `pyasprasad / lesson-end-project`. The code content is:

```
1 name: GitHub Actions Demo
2 run-name: Testing out GitHub Actions
3 on: [push]
4 jobs:
5   Basic-Workflow:
6     runs-on: ubuntu-latest
7     steps:
8       - run: echo "Job is executing as part of Basic workflow"
9 
```

The right side of the interface shows the Marketplace with featured actions like `Cache` and `Setup Node.js environment`.

2.3 Add the **Commit message** as `Create basic-workflow.yml` and then click on **Commit changes** to save the workflow file in the code repository



2.4 Navigate to the **Actions** tab to access the workflow execution, and under **All workflows**, click on the **Testing out GitHub Actions**

The screenshot shows the GitHub Actions interface. On the left, there's a sidebar with options like 'Actions', 'Workflows', 'GitHub Actions Demo', 'Management', 'Caches', and 'Runners'. The 'Actions' tab is selected. In the main area, it says 'All workflows' and 'Showing runs from all workflows'. A single workflow run is listed: 'Testing out GitHub Actions' (GitHub Actions Demo #1: Commit 3b745ba pushed by pyasprasad). This run is marked as 'Success' and completed '4 minutes ago' with a duration of '13s'. A red box highlights this specific workflow run.

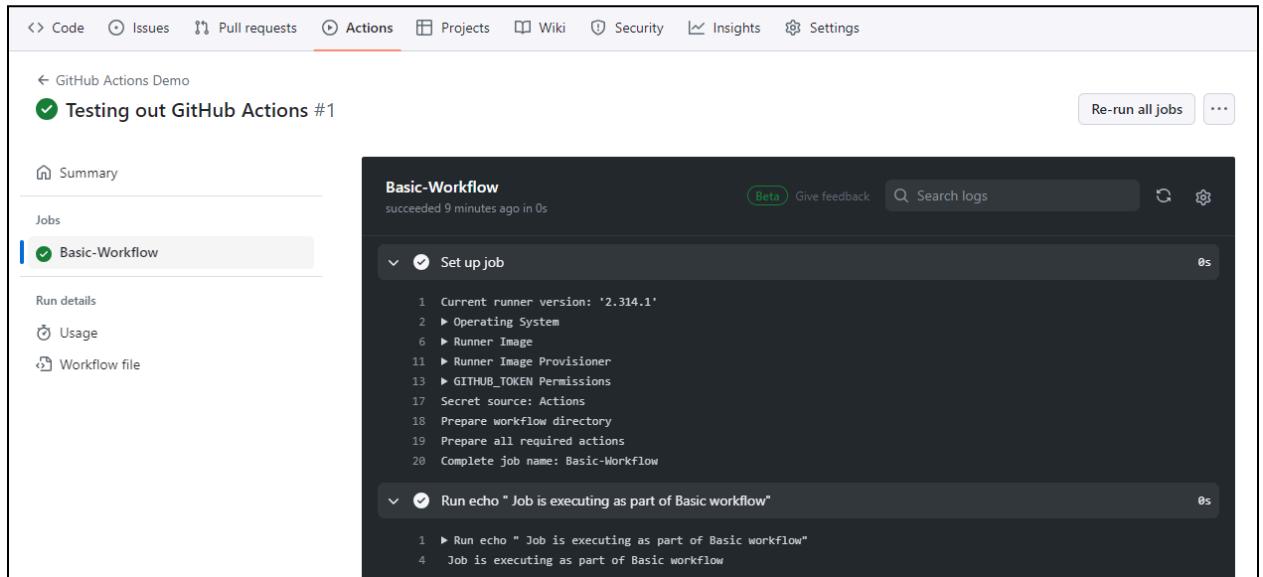
2.5 Click on the **Basic-Workflow** to view the job execution details

This screenshot shows the detailed view of a workflow run. It's for 'Testing out GitHub Actions #1' triggered via push 11 minutes ago. The status is 'Success' with a total duration of '11s'. The workflow file is 'basic-workflow.yml' with the job 'Basic-Workflow'. A red box highlights the 'Basic-Workflow' job card.

The **Basic-Workflow** screen will appear as shown below:

This screenshot shows the execution log for the 'Basic-Workflow'. It lists three steps: 'Set up job', 'Run echo " Job is executing as part of Basic workflow"', and 'Complete job', all of which succeeded. The log was generated 8 minutes ago. A red box highlights the first step, 'Set up job'.

2.6 Expand the output under the **Basic-Workflow**



The screenshot shows the GitHub Actions interface for a workflow named "Testing out GitHub Actions #1". The "Actions" tab is selected. On the left, there's a sidebar with "Summary", "Jobs", and "Basic-Workflow" (which is highlighted with a blue bar). Below the sidebar are links for "Run details", "Usage", and "Workflow file". The main area displays the "Basic-Workflow" logs. The logs show two jobs: "Set up job" and "Run echo". The "Set up job" job has 20 steps, and the "Run echo" job has 4 steps. The logs are displayed in a monospaced font.

```
Basic-Workflow
succeeded 9 minutes ago in 0s
Beta Give feedback Search logs

Set up job
1 Current runner version: '2.314.1'
2 ▶ Operating System
6 ▶ Runner Image
11 ▶ Runner Image Provisioner
13 ▶ GITHUB_TOKEN Permissions
17 Secret source: Actions
18 Prepare workflow directory
19 Prepare all required actions
20 Complete job name: Basic-Workflow

Run echo " Job is executing as part of Basic workflow"
1 ▶ Run echo " Job is executing as part of Basic workflow"
4 Job is executing as part of Basic workflow
```

By following these steps, you have successfully created and executed a basic GitHub Actions workflow for automated testing and deployment, enhancing the efficiency and reliability of project development processes.