

## 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

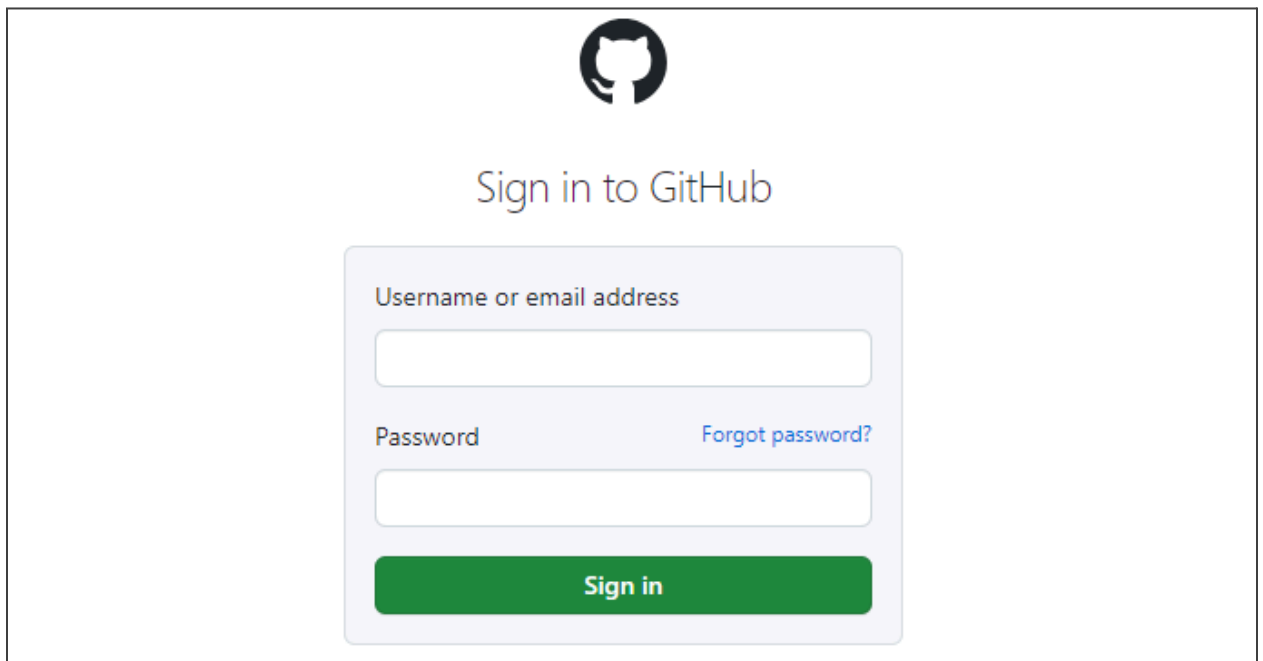
**Prerequisites:** None

Steps to be followed:

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

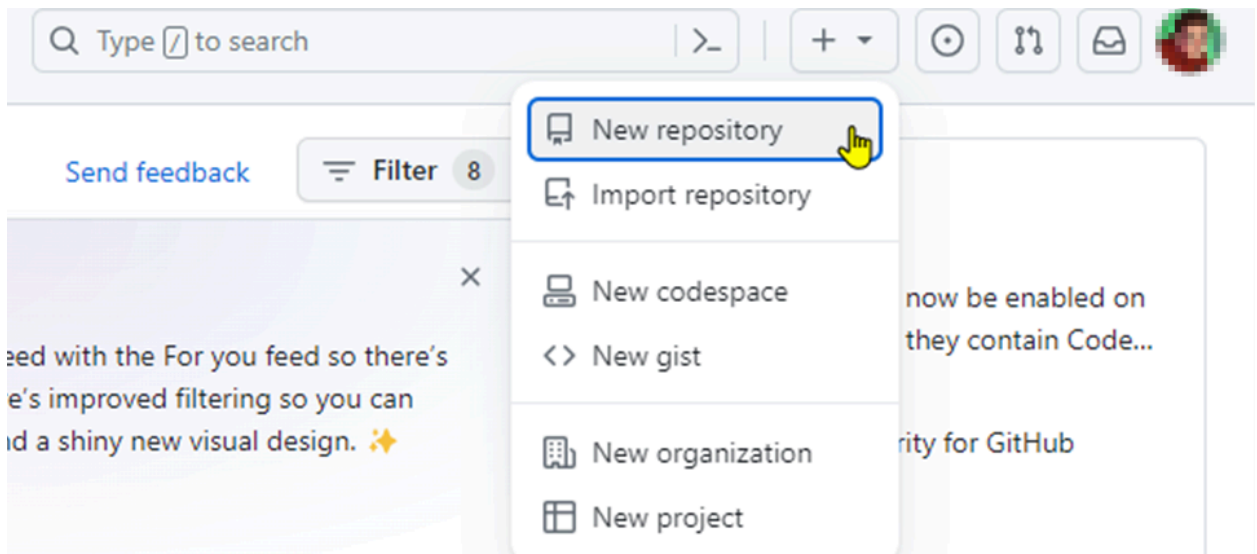
#### Step 1: Create a new GitHub repository

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

A screenshot of the GitHub login page. At the top center is the GitHub logo (Octocat). Below it, the text "Sign in to GitHub" is displayed. Underneath is a light gray rounded rectangle containing the login form. The form has two input fields: "Username or email address" and "Password". To the right of the password field is a blue link that says "Forgot password?". At the bottom of the form is a green button with the text "Sign in" in white.

**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 \*

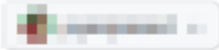
 /

✔ 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)

- 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

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

Choose a license

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

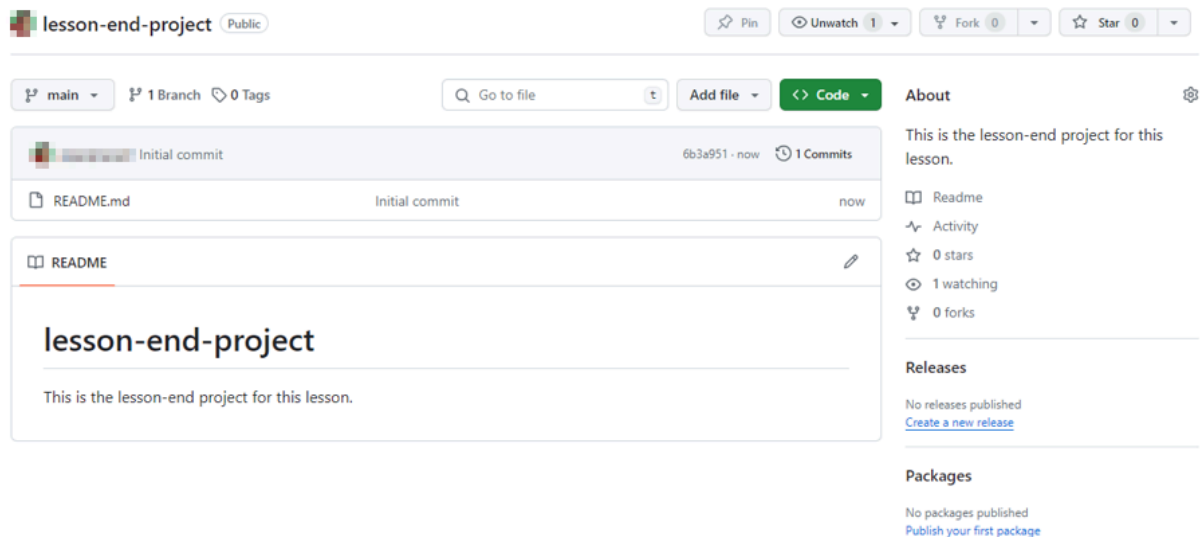
License: None ▾

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

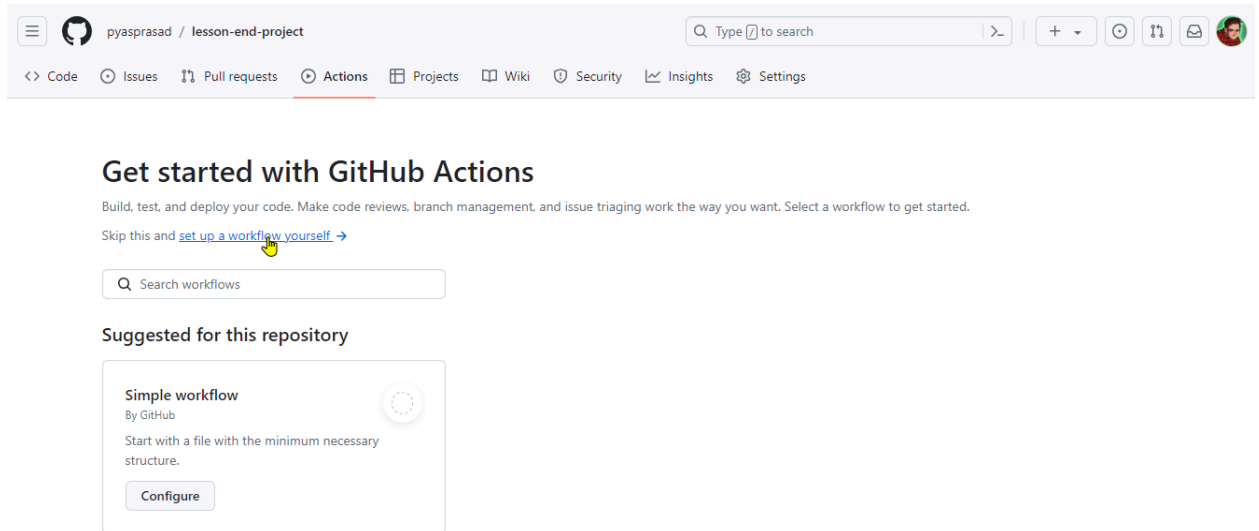


The screenshot shows the GitHub interface for a newly created public repository named "lesson-end-project". At the top, there are buttons for "Pin", "Unwatch" (1), "Fork" (0), and "Star" (0). Below this, the repository name "lesson-end-project" is displayed with a "Public" badge. The main content area shows the "main" branch with 1 commit and 0 tags. A search bar "Go to file" and buttons "Add file" and "Code" are visible. The commit history shows an "Initial commit" with a commit hash "6b3a951" and a timestamp "now". Below the commit history, a file named "README.md" is listed. The README content is displayed, showing the title "lesson-end-project" and the description "This is the lesson-end project for this lesson." On the right side, there is a sidebar with sections: "About" (This is the lesson-end project for this lesson.), "Readme" (with a link to the README), "Activity" (with a link to the activity), "Stars" (0 stars), "Watching" (1 watching), "Forks" (0 forks), "Releases" (No releases published, with a link to "Create a new release"), and "Packages" (No packages published, with a link to "Publish your first package").

The remote GitHub repository is created.

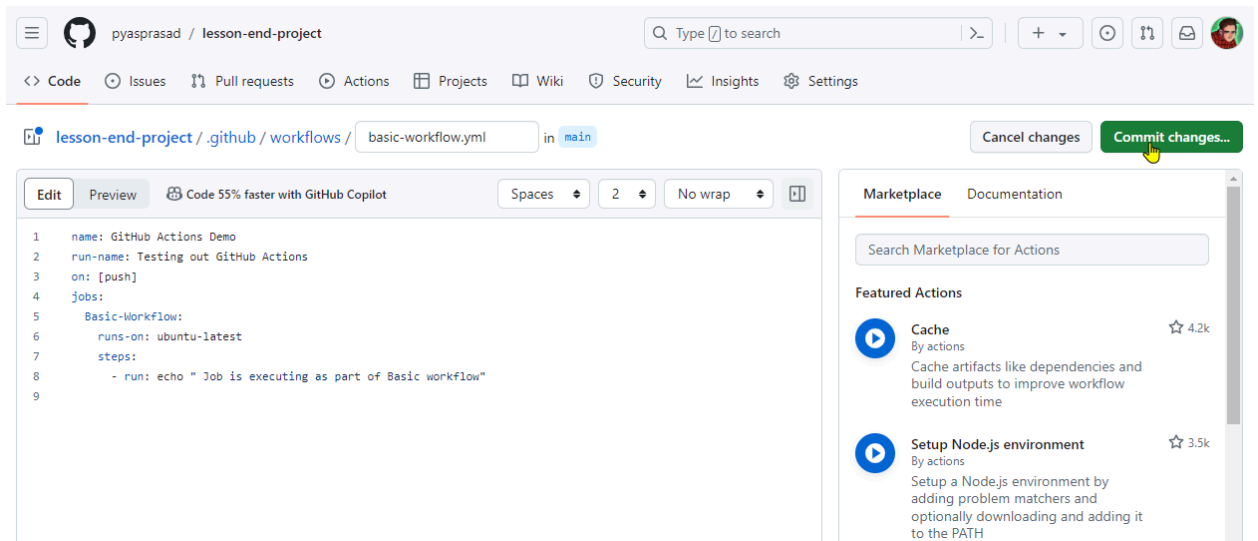
## Step 2: Create and execute a new workflow file

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

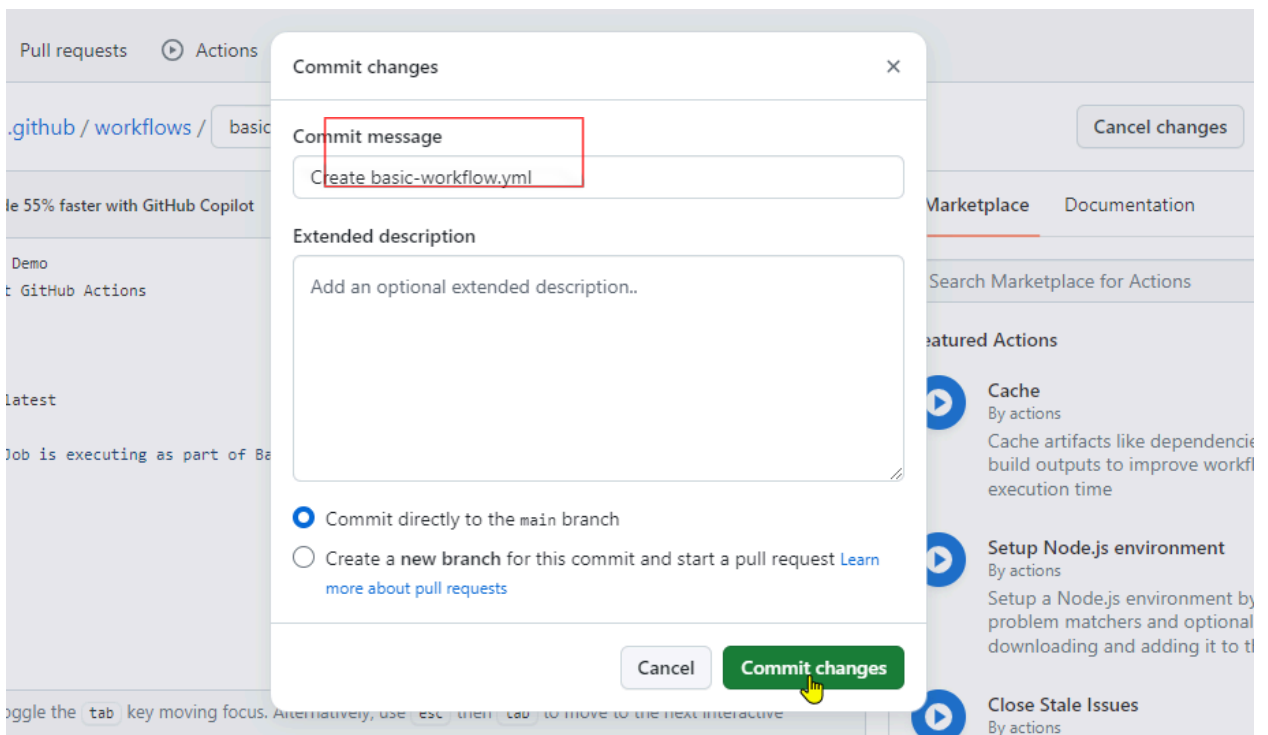


2.2 Create a new workflow file **basic-workflow.yml** with the below code, then click on **Commit changes**:

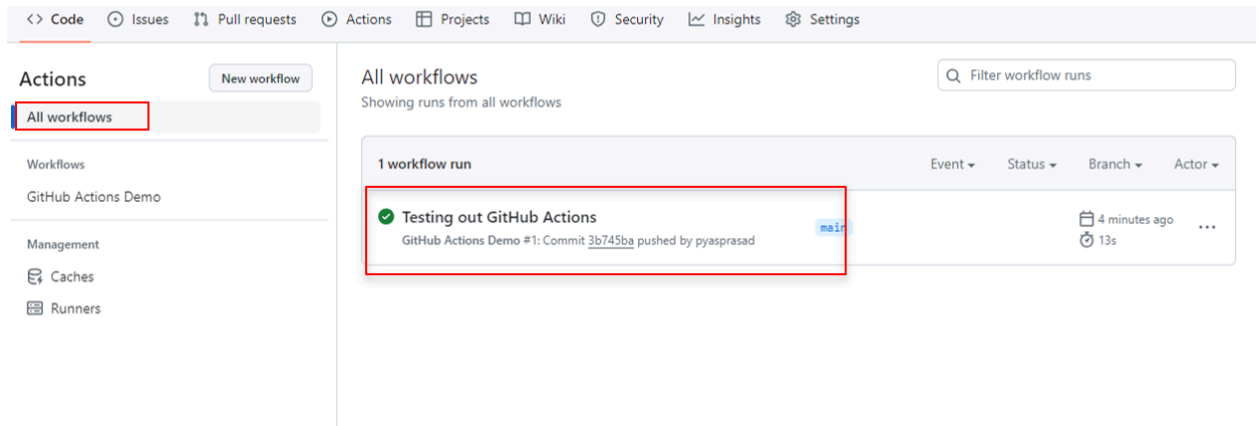
```
name: GitHub Actions Demo
run-name: Testing out GitHub Actions
on: [push]
jobs:
  Basic-Workflow:
    runs-on: ubuntu-latest
    steps:
      - run: echo " Job is executing as part of Basic workflow"
```



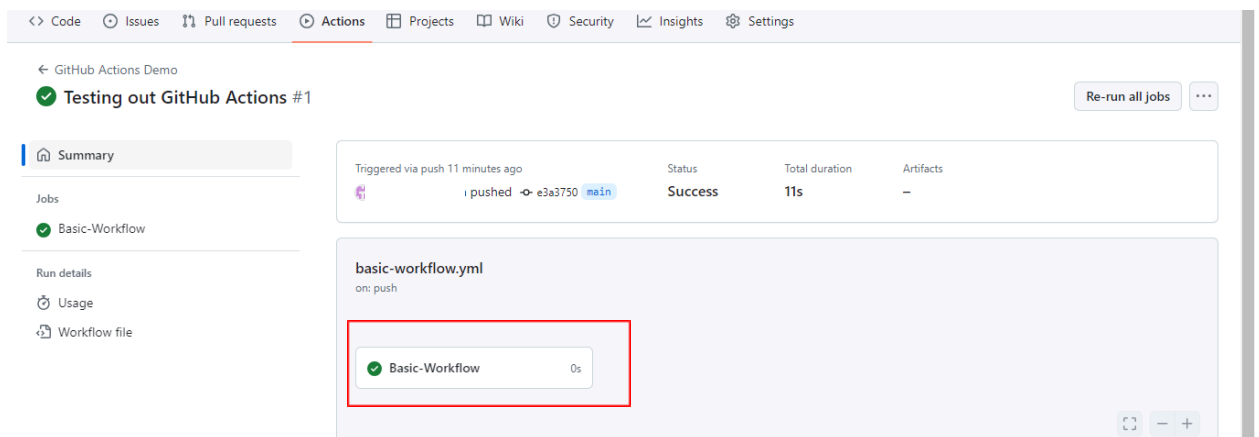
2.3 Add the **Commit message** as **Create basic-workflow.yml** and then click on **Commit changes** to save the workflow file in the code repository



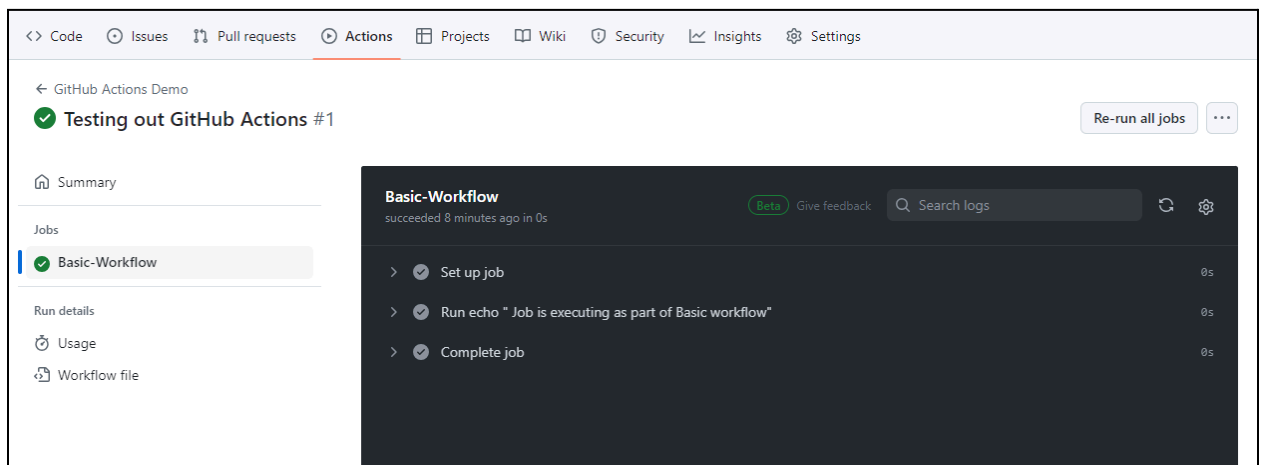
## 2.4 Navigate to the **Actions** tab to access the workflow execution, and under **All workflows**, click on the **Testing out GitHub Actions**



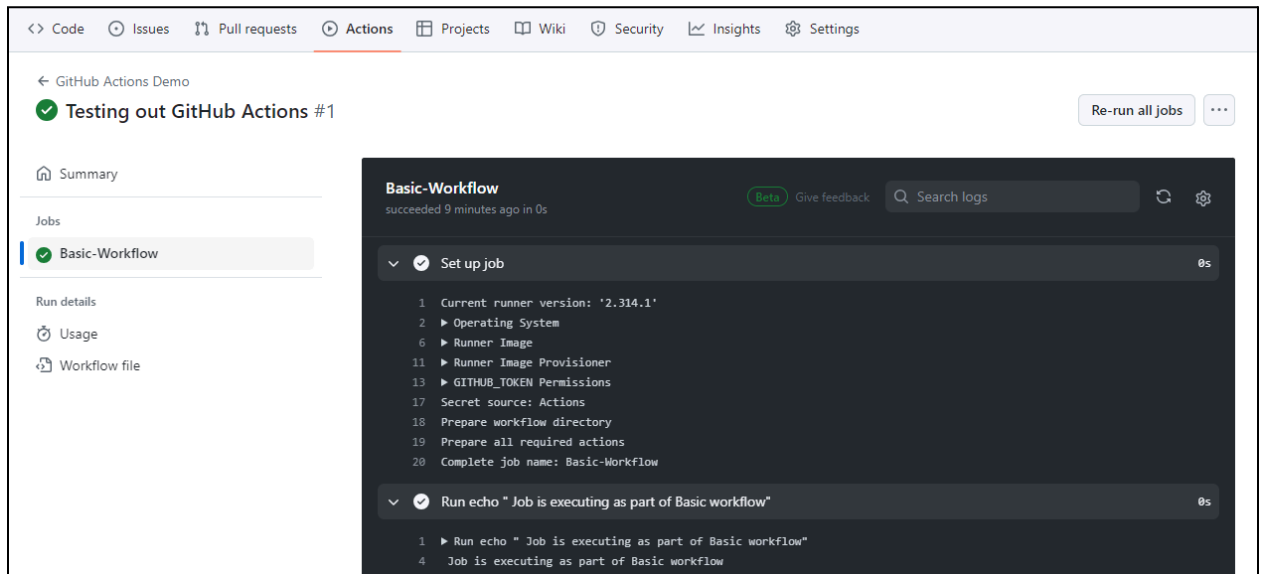
## 2.5 Click on the **Basic-Workflow** to view the job execution details



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



## 2.6 Expand the output under the **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.