

GitHub Actions

GitHub Actions is a powerful automation tool that allows you to build, test, and deploy your code right from your GitHub repository. Understanding the key building blocks — workflows, jobs, and steps — is essential for effective automation.

- [Workflows](#)
- [Jobs](#)
- [Steps](#)
- [Creating a Simple Workflow](#)

Workflows

A workflow is a customizable, automated process that you can define in your repository. It's typically used for continuous integration (CI), continuous deployment (CD), and other automation tasks. Here are the important aspects of workflows:

- **Trigger:** Workflows are triggered by specific events, such as pushes, pull requests, or scheduled events. You define when and how a workflow runs.
- **YAML Configuration:** Workflows are defined in a YAML file (e.g., `.github/workflows/main.yml`) within your repository. This configuration file specifies the workflow's name, triggers, jobs, and steps.

Jobs

A job is a unit of work within a workflow. You can have multiple jobs in a workflow, and they can run in parallel or sequentially. Here are the key points about jobs:

- **Runs-On:** Each job specifies the runner environment, such as Ubuntu, macOS, or Windows. You choose the environment that best suits your workflow.
- **Parallelism:** You can configure jobs to run concurrently, which can speed up your workflow's execution time.

Steps

A step is an individual task within a job. Steps are the smallest building blocks of a workflow and are where the actual work happens. Here are the important aspects of steps:

- **Name:** Each step has a name that helps identify its purpose. It appears in the GitHub Actions log to provide clarity during execution.
- **Run Commands:** Steps execute commands or scripts. These can include shell commands, script files, or even invoking actions from external sources (GitHub Marketplace, your own custom actions, etc.).
- **Inputs and Outputs:** Steps can have inputs and produce outputs, allowing them to communicate and share data with other steps.

Creating a Simple Workflow

To enable a workflow using the GitHub UI and by creating a directory `.github/workflows`, and to create this simple workflow and test it, follow these steps:

1. Enable Workflow using GitHub UI:

- Navigate to your GitHub repository.
- Click on the "Actions" tab.
- If GitHub Actions is not enabled, you'll see an option to enable it. Click on "Set up this workflow" or "Enable Actions" and follow the prompts.
- Once enabled, you can proceed to create workflows.

2. Create `.github/workflows` Directory:

- In your repository, create a new directory named `.github/workflows`. You can do this directly on GitHub or by cloning the repository to your local machine and creating the directory there.

3. Create the Simple Workflow:

- Navigate to the `.github/workflows` directory.
- Create a new file named `02-01-Simple-Workflow.yml`.
- Copy and paste the following content into the file:

```
name: 02-01-Simple Workflow
on: push
jobs:
  my-first-job:
    runs-on: ubuntu-latest
    steps:
      - name: Print Hello
        run: echo "Hello World"

  my-second-job:
    runs-on: ubuntu-latest
    steps:
      - name: Print Bye
        run: echo "Bye World"
```

4. Testing the Workflow:

- Commit and push the changes made to your GitHub repository.
- Go to the "Actions" tab on your repository in GitHub.
- You should see your workflow listed there with a status indicating its execution.
- If there are any errors, GitHub will provide feedback in the workflow's logs.

5. Validation:

- Ensure that your workflow is properly triggered on a push event.
- Verify that both jobs (`my-first-job` and `my-second-job`) execute successfully.
- Check the logs of each job to confirm that the expected output ("Hello World" and "Bye World") is printed.

