GitHub Action Introduction

Tuesday, April 23, 2024 10:57 PM

Overview

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.
We can configure a GitHub Actions workflow to be triggered when an event occurs in your repository, such as a pull request being opened or an issue being created.
Your workflow contains one or more jobs which can run in sequential order or in parallel. Each job will run inside its own virtual machine runner, or inside a container, and has one or more steps that either run a script that you define or run an action, which is a reusable extension that can simplify your workflow.

Things you can do with GHA

You can create workflows that

- Run tests whenever you push a change to your repository
- Build and test pull requests
- Deploy your application every time a release is created
- Deploy merged pull requests to production
- Add the appropriate labels whenever someone creates a new issue in your repository

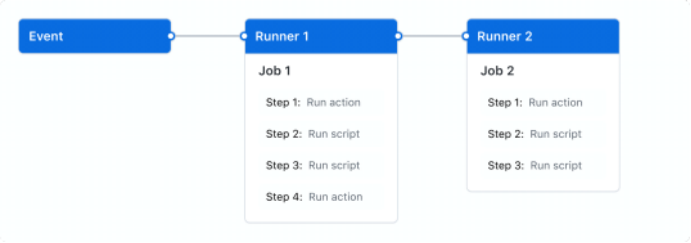
Runners

A runner is a server that runs your workflows when they're triggered.
Each runner can run a single job at a time.
Each workflow run executes in a fresh, newly-provisioned virtual machine.
Workflow can be ran on GitHub provides virtual machines,

- Linux
- Windows
- macOS
- Or our own self-hosted runners in our own data center or cloud infrastructure

Workflows

A workflow is a configurable automated process that will run one or more jobs.
Workflows are defined by a YAML file checked in to your repository defined in the `.github/workflows` directory in a repository.
A repository can have multiple workflows, each of which can perform a different set of tasks.
You can reference a workflow within another workflow.



Triggers

- Event in your repository
- Manually
- At a defined schedule
- Rest API

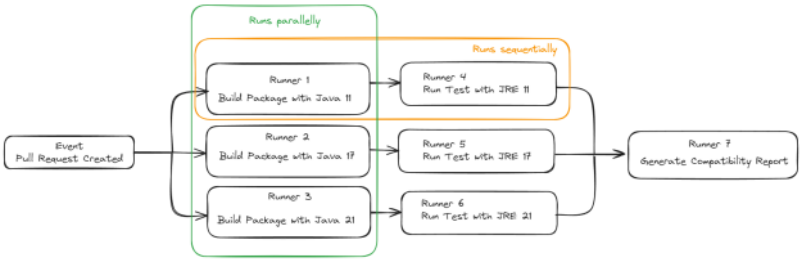
Events

- When someone creates a pull request
- Opens an issue
- Pushes a commit to a repository

More Events can be explored here : Events that trigger workflows
Note some Event have multiple activity type

Jobs

A job is a set of steps in a workflow that is executed on the same runner.
Each step is either a shell script that will be executed, or an action that will be run.
Steps are executed in order and are dependent on each other.
Since each step is executed on the same runner, you can share data from one step to another.
For example, you can have a step that builds your application followed by a step that tests the application that was built.



You can configure a job's dependencies with other jobs; by default, jobs have no dependencies and run in parallel with each other. When a job takes a dependency on another job, it will wait for the dependent job to complete before it can run.
For example, you may have multiple build jobs for different architectures that have no dependencies, and a packaging job that is dependent on those jobs. The build jobs will run in parallel, and when they have all completed successfully, the packaging job will run.

Actions

A reusable extension that can simplify your workflow
An action is a custom application for the GitHub Actions platform that performs a complex but frequently repeated task. Use an action to help reduce the amount of repetitive code that you write in your workflow files.

- You can write your own actions : Creating actions
- Find actions to use in your workflows in the GitHub Marketplace.

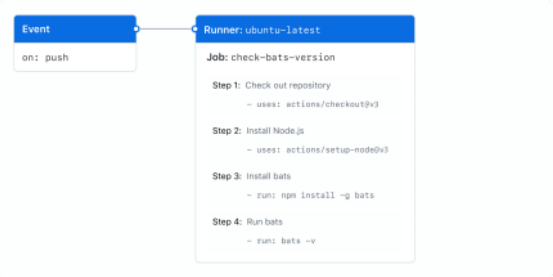
Example :

- Pull your git repository from GitHub --> actions/checkout
- Set up the correct toolchain for your build environment --> actions/setup-node
- Set up the authentication to your cloud provider --> azure/login

EXAMPLE

YAML

```yaml
name: learn-github-actions
run-name: ${{ github.actor }} is learning GitHub Actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: '20'
      - run: npm install -g bats
      - run: bats -v
```

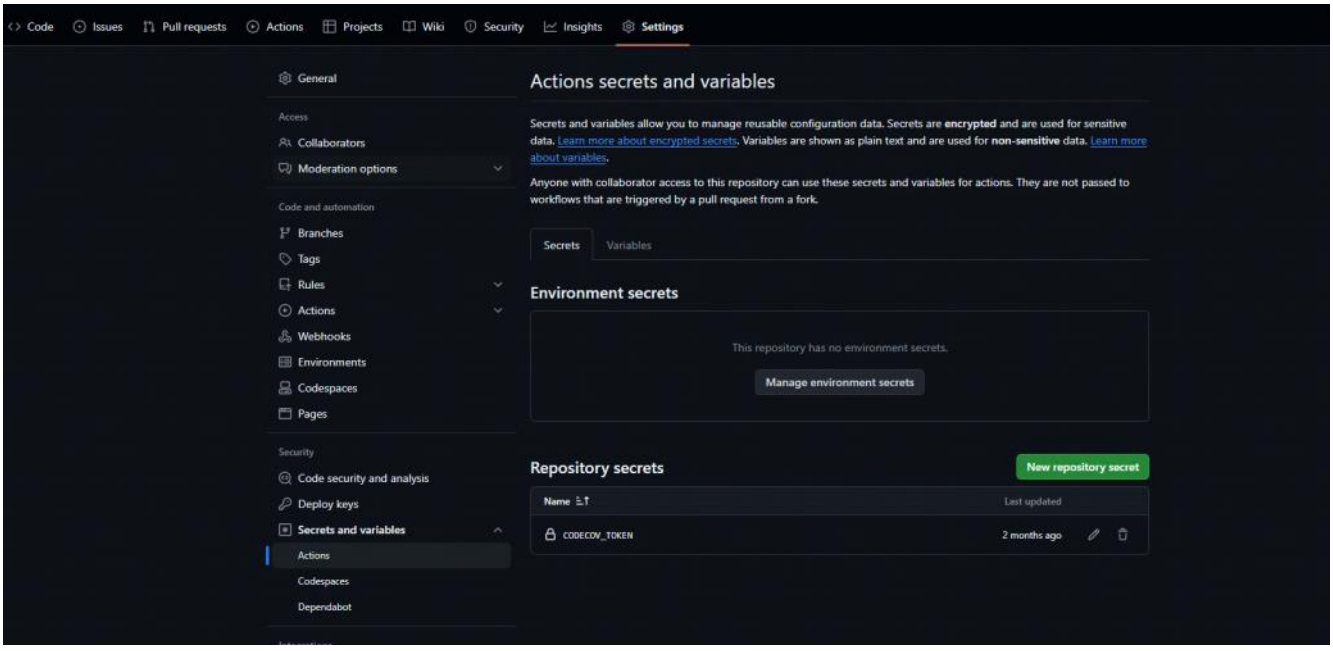


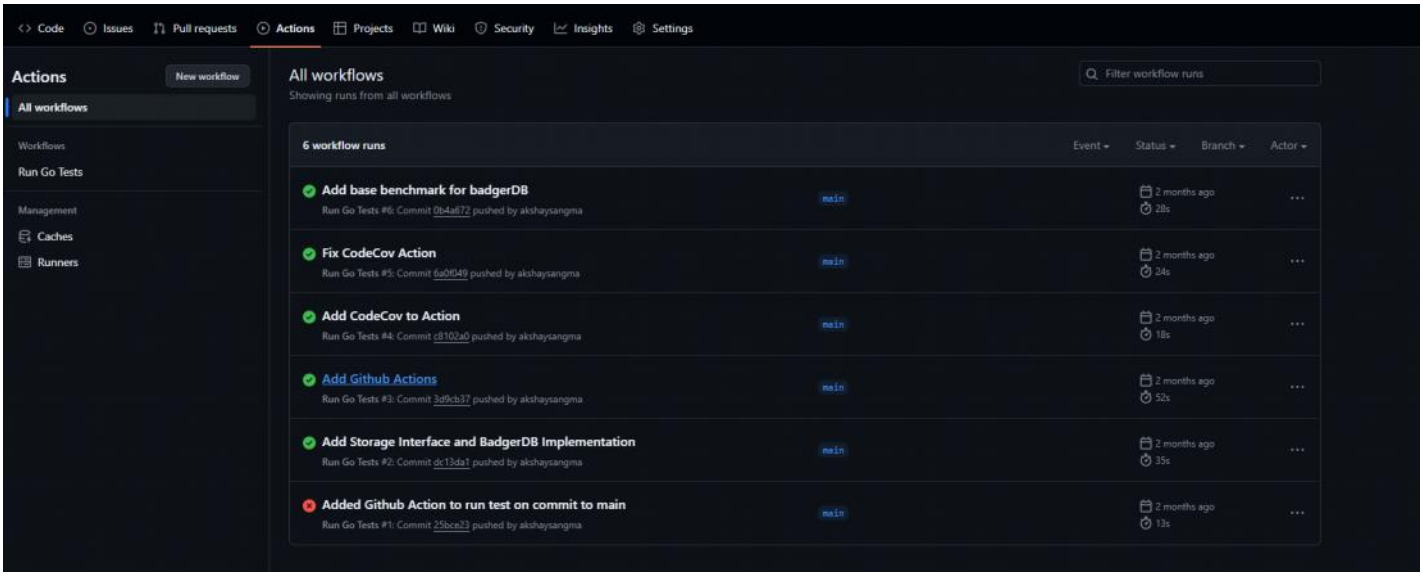
Example 2:

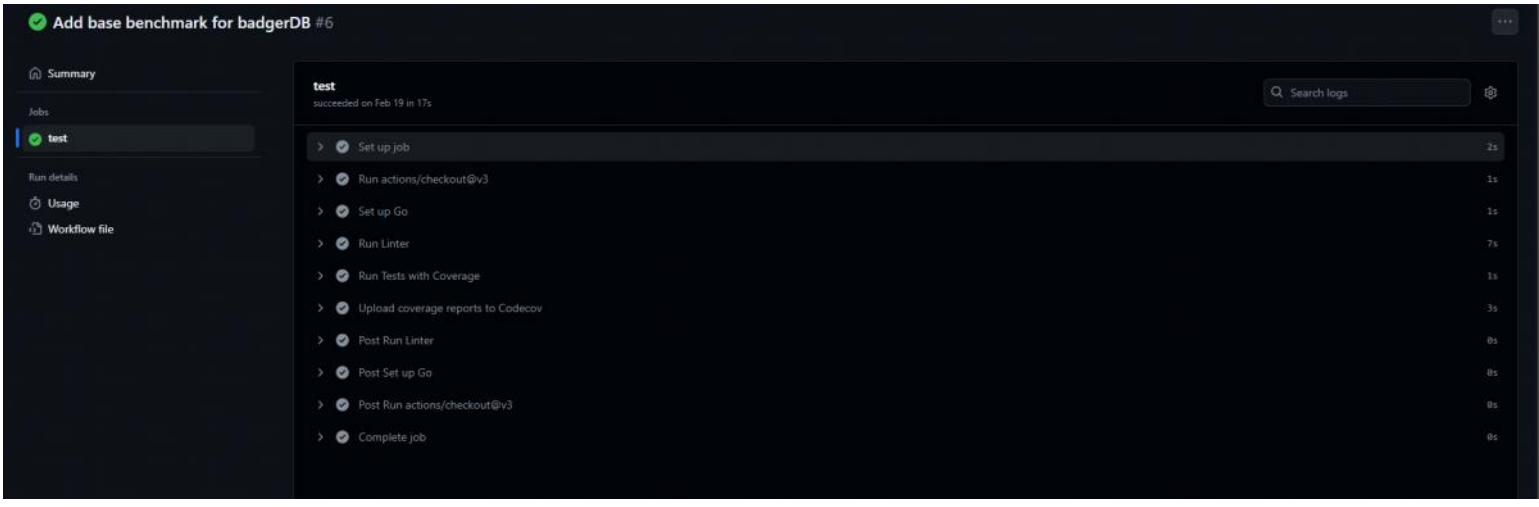
https://github.com/akshaysangma/hydrastore/blob/main/.github/workflows/go.yml

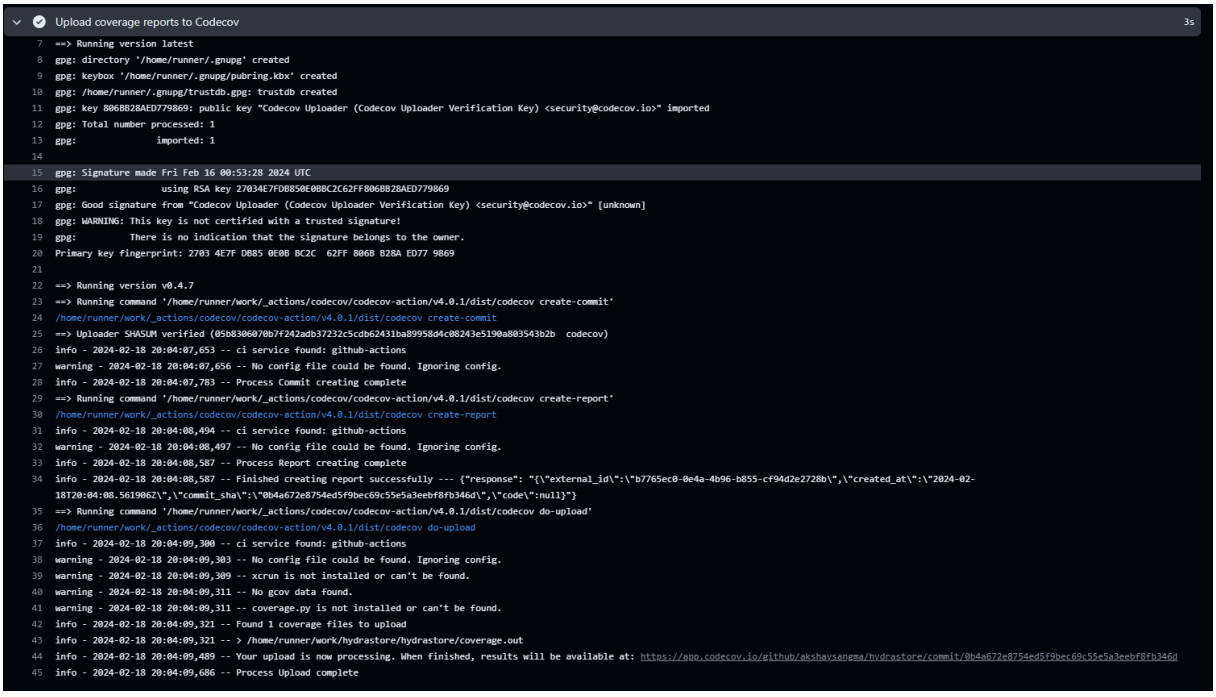
Code Blame 28 lines (21 loc) · 616 Bytes Code 55% faster with GitHub Copilot Raw

```yaml
name: Run Go Tests

on:
  push:
    branches: [ main ]

jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3

      - name: Set up Go
        uses: actions/setup-go@v3
        with:
          go-version: '^1.22' # Specify your desired Go version

      - name: Run Linter
        uses: golangci/golangci-lint-action@v3

      - name: Run Tests with Coverage
        run: go test ./... -coverprofile=coverage.out

      - name: Upload coverage reports to Codecov
        uses: codecov/codecov-action@v4.0.1
        env:
          CODECOV_TOKEN: ${{ secrets.CODECOV_TOKEN }}
```


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security and analysis
Deploy keys
Secrets and variables
Actions
Codespaces
Dependabot
Actions secrets and variables
Secrets and variables allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for non-sensitive data. Learn more about variables.
Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.
Secrets
Variables
Environment secrets
This repository has no environment secrets.
Manage environment secrets
Repository secrets
New repository secret
Name
Last updated
CODECOV_TOKEN
2 months ago


Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
Workflows
Run Go Tests
Management
Caches
Runners
All workflows
Showing runs from all workflows
Filter workflow runs
6 workflow runs
Event
Status
Branch
Actor
Add base benchmark for badgerDB
Run Go Tests #6: Commit 0b4a672 pushed by akshaysangma
main
2 months ago
28s
Fix CodeCov Action
Run Go Tests #5: Commit 6a0f049 pushed by akshaysangma
main
2 months ago
24s
Add CodeCov to Action
Run Go Tests #4: Commit c8102a0 pushed by akshaysangma
main
2 months ago
18s
Add Github Actions
Run Go Tests #3: Commit 3d9cb37 pushed by akshaysangma
main
2 months ago
52s
Add Storage Interface and BadgerDB Implementation
Run Go Tests #2: Commit dc13da1 pushed by akshaysangma
main
2 months ago
35s
Added Github Action to run test on commit to main
Run Go Tests #1: Commit 25bce23 pushed by akshaysangma
main
2 months ago
13s


Add base benchmark for badgerDB #6
Summary
Jobs
test
Run details
Usage
Workflow file
test
succeeded on Feb 19 in 17s
Search logs
Set up job
2s
Run actions/checkout@v3
1s
Set up Go
1s
Run Linter
7s
Run Tests with Coverage
1s
Upload coverage reports to Codecov
3s
Post Run Linter
0s
Post Set up Go
0s
Post Run actions/checkout@v3
0s
Complete job
0s


Upload coverage reports to Codecov
3s
==> Running version latest
gpg: directory '/home/runner/.gnupg' created
gpg: keybox '/home/runner/.gnupg/pubring.kbx' created
gpg: /home/runner/.gnupg/trustdb.gpg: trustdb created
gpg: key 806BB28AED779869: public key "Codecov Uploader (Codecov Uploader Verification Key) <security@codecov.io>" imported
gpg: Total number processed: 1
gpg: imported: 1
gpg: Signature made Fri Feb 16 00:53:28 2024 UTC
gpg: using RSA key 27034E7FDB850E0BC2C62FF806BB28AED779869
gpg: Good signature from "Codecov Uploader (Codecov Uploader Verification Key) <security@codecov.io>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: 2703 4E7F DB85 0E0B BC2C 62FF 806B B28A ED77 9869
==> Running version v0.4.7
==> Running command '/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov create-commit'
/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov create-commit
==> Uploader SHASUM verified (05b8306070b7f242adb37232c5cdb62431ba89958d4c08243e5190a803543b2b codecov)
info - 2024-02-18 20:04:07,653 -- ci service found: github-actions
warning - 2024-02-18 20:04:07,656 -- No config file could be found. Ignoring config.
info - 2024-02-18 20:04:07,783 -- Process Commit creating complete
==> Running command '/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov create-report'
/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov create-report
info - 2024-02-18 20:04:08,494 -- ci service found: github-actions
warning - 2024-02-18 20:04:08,497 -- No config file could be found. Ignoring config.
info - 2024-02-18 20:04:08,587 -- Process Report creating complete
info - 2024-02-18 20:04:08,587 -- Finished creating report successfully --- {"response": "{\"external_id\":\"b7765ec0-0e4a-4b96-b855-cf94d2e2728b\",\"created_at\":\"2024-02-18T20:04:08.561906Z\",\"commit_sha\":\"0b4a672e8754ed5f9bec69c55e5a3eebf8fb346d\",\"code\":null}"}
==> Running command '/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov do-upload'
/home/runner/work/_actions/codecov/codecov-action/v4.0.1/dist/codecov do-upload
info - 2024-02-18 20:04:09,300 -- ci service found: github-actions
warning - 2024-02-18 20:04:09,303 -- No config file could be found. Ignoring config.
warning - 2024-02-18 20:04:09,309 -- xcrun is not installed or can't be found.
warning - 2024-02-18 20:04:09,311 -- No gcov data found.
warning - 2024-02-18 20:04:09,311 -- coverage.py is not installed or can't be found.
info - 2024-02-18 20:04:09,321 -- Found 1 coverage files to upload
info - 2024-02-18 20:04:09,321 -- > /home/runner/work/hydrastore/hydrastore/coverage.out
info - 2024-02-18 20:04:09,489 -- Your upload is now processing. When finished, results will be available at: https://app.codecov.io/github/akshaysangma/hydrastore/commit/0b4a672e8754ed5f9bec69c55e5a3eebf8fb346d
info - 2024-02-18 20:04:09,686 -- Process Upload complete