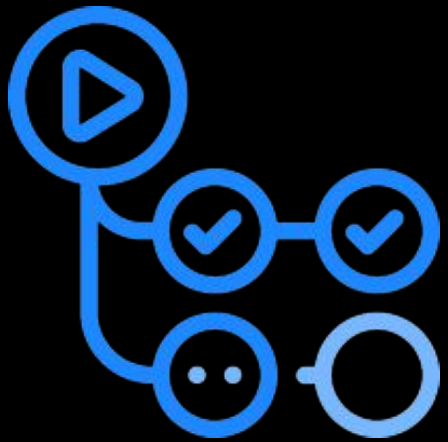
GitHub Partner Engineering 2020

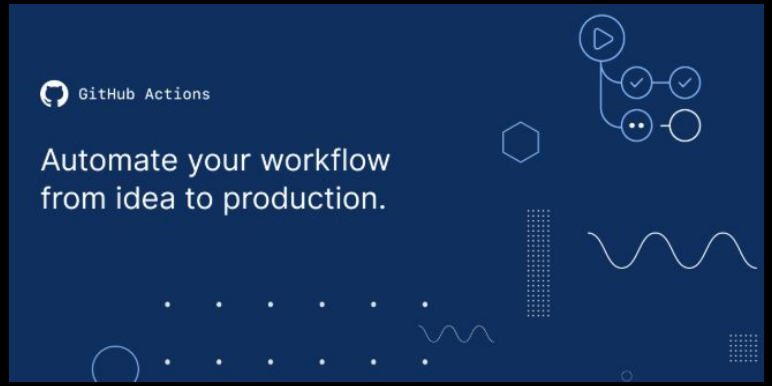
Getting Started with GitHub Actions



Presented by @eekdageek

What is GitHub Actions?

GitHub Actions is a new feature that allows you to customize your workflow on GitHub. Originally released in beta in 2018, the latest version includes powerful CI/CD primitives, a familiar YAML syntax, and the ability to run as a script or in a container! 🎉



Concepts 🌱

GitHub Actions Core Concepts

- GitHub Actions
  - The entire product. GitHub Actions turns your repositories into serverless functions, written by you.
- Action
  - Can be consumed by other repositories on the GitHub graph to automate development workflows
- Workflow
  - Lets you codify useful processes to your liking in your repo, and can utilize Actions
  - Automation for CI/CD, PR/issues management, or anything from your repo
- Runner
  - A GitHub service in each virtual environment that waits for available jobs

Actions Characteristics

- Triggered by events in GitHub
- Triggered by events outside of GitHub via repository dispatch API
- Actions can be:
  - Docker containers
  - JavaScript
- No user interaction
- Best for stateless flows
  - i.e., does not have to remember what happened in the previous run
- Can wrap CLIs or use APIs
- Currently runs only in per repo basis.

Actions Flow

2

Event (such as push)

1

First, check in action workflow file in .github/workflow/<filename>.yml in the repo. (This process can be accomplished by going to the Actions tab as well.)

Workflow file

Branch: master | create-release / .github / workflows / ci.yml

IAmHughes Setup CI

1 contributor

16 lines (13 sloc) 350 Bytes

```
name: "Tests"

on:
  pull_request:
  push:
    branches:          # array of glob patterns matching against refs/heads
      - master         # triggers on pushes that contain changes in master
```

Can trigger from events specified

Go to the Actions tab in your repo to see the runs.

Click on each run to see details.

3

actions / create-release

Code | Issues 3 | Pull requests 2 | Actions | Security | Insights | Settings

Unwatch 5 | Star 20 | Fork

Workflows | New workflow

All workflows

Tests

All workflows

Tests — on: pull_request (becafb2) — csexton:add-body — triggered by csexton — 6 hours ago — 37.0s

Tests — on: pull_request (9167471) — csexton:add-body — triggered by csexton — 10 hours ago — 43.0s

Details of a run

Code | Issues 3 | Pull requests 2 | Actions | Security | Insights | Settings

Include body parameter for release

add-body becafb2

Tests
on: pull_request

test

Tests / test
successful 6 hours ago in 22s

- Set up job
- Run actions/checkout@v1
- Run npm ci
- Run npm test
- Complete job

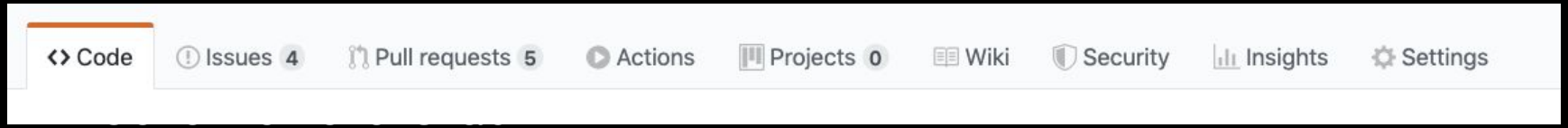
Creating Your First Workflow

Workflows

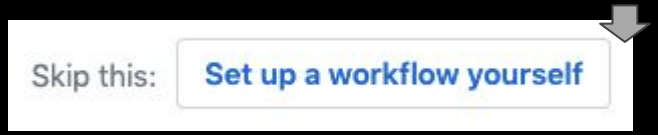
- Workflows let you codify useful processes to your liking in your repo
- Automation

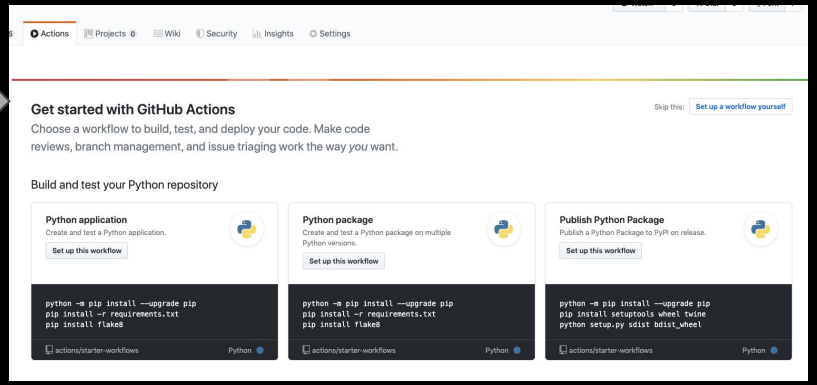
Enabling GitHub Actions in a Repo

- Make sure the repo you are working in now has an Actions tab.



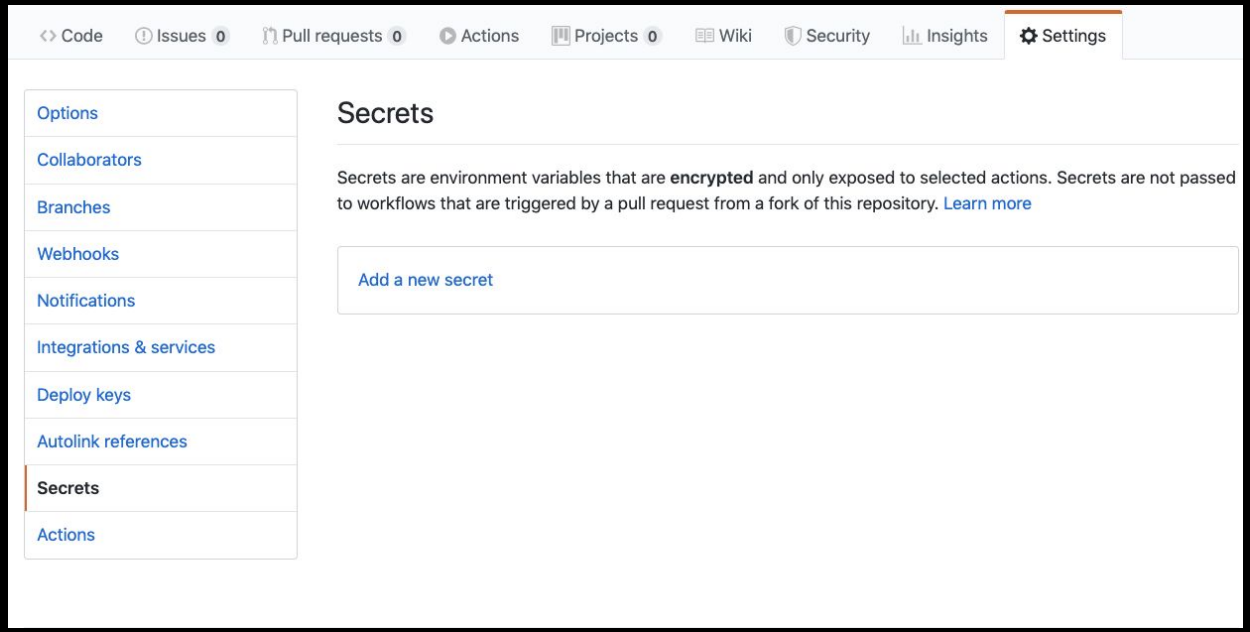
- Once you are in the Actions tab, set up the workflow using existing actions
- Or, set up a workflow yourself





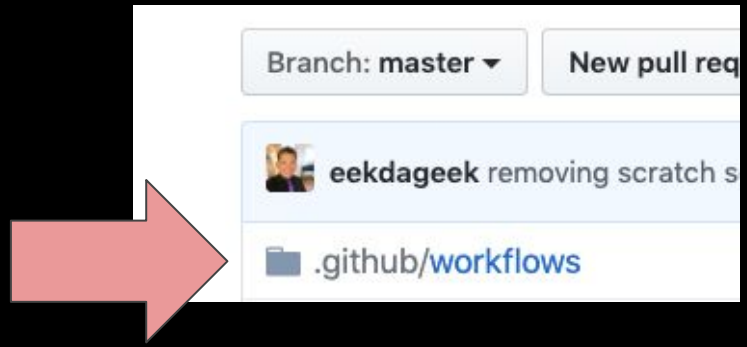
Secrets

- Secrets can be set under Settings-> Secrets
- Information about secrets can be found here.
- GITHUB_TOKEN is already available without adding to secrets



Enabling Actions Workflow in a Repo

- You can also just create a .yml file in the .github/workflows directory in a repo.



Workflow File

In the “run” section of the workflow file, you can run multiple commands for the virtual host machine you specify.

```yaml
# .github/workflows/test.yml
on:
  push:
    branches:         # array of glob patterns matching against refs/heads. Optional; defaults to all
    - master          # triggers on pushes that contain changes in master
    - feature/*

name: Test

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v1       # this is an action
    - uses: actions/setup-node@v1     # this is another action
    - name: running basic commands on ubuntu # just showing how we can run commands
      run: |
          ls -lstra # shell command!
          pwd # another shell command
    - name: npm install, test         # this is a script
      run: |
          npm install
          npm test
```

Virtual host machine specified ([Available virtual hosts](#))

Can run multiple commands to accomplish your tasks

Modularize

```yaml
jobs:
  build_and_release:
    name: Build and Release
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v1.0.0
      - name: Lint code
        uses: actions/linting@v1.0.0
      - name: Build project
        run: |
          npm build
      - name: Run tests
        run: |
          npm test
      - name: Create draft release
        id: create_draft_release
        uses: actions/create-draft-release@v1.0.0
        env:
          # Access the `GITHUB_TOKEN` secret from the repository
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        with:
          # Access the `ref` from the `github` payload object
          tag_name: ${{ github.ref }}
          release_name: Release ${{ github.ref }}
          draft: true
          prerelease: false
```

Modularize as much as possible

Sample code found here.

Creating Your First Action 

Actions

- Reusable units of code
- Can be as simple as the example on the right

```yaml
# action.yml
name: 'Hello World'
description: 'Print greeting message'
author: 'GitHub'
inputs:
  greeting:
    description: 'Who to greet'
    default: 'world'
runs:
  using: 'node12'
  main: 'index.js'
```

```javascript
// index.js
const core = require('@actions/core'); // npm install this

async function run() {
  try {
    const greeting = core.getInput('greeting');
    console.log(`Hello, ${greeting}!`);
  }
  catch (error) {
    core.setFailed(error.message);
  }
}

run();
```

Hello World Actions!

- Javascript
  - Javascript Hello World Action
- Container
  - Docker Hello World Action

Container vs. JavaScript Actions

	JavaScript Action	Container Action
Virtual Environment	Linux, macOS, Windows	Linux
Language	Anything that compiles JavaScript	Any
Speed	👍👍	👍

JavaScript-based Actions are preferred:

- They run on all virtual environments (Linux, macOS, Windows)
- The user experience is improved
- We have actions/toolkit and actions/javascript-action available as a good way to get started building JS-based actions.

References 📌

Example GitHub Actions

- setup-node
- create-release
- upload-release-asset
- javascript-action

Other GitHub Actions can be found here

Quick Links

- Core concepts for GitHub Actions
- GitHub Actions Documentation
- Usage limits
- Workflow syntax
- Authenticating with the GITHUB_TOKEN
- Current supported virtual environments
- Events that trigger workflows