Github Actions

James Henstridge <james@jamesh.id.au>

Github Actions

Yet another continuous integration system for projects hosted on Github

This time from Github itself

Free for Open Source projects, and a free tier for private projects

Hosted x86 runners for Linux, Windows, and MacOS

Can provide your own runners for additional platform support

Workflows

A project can have one or more workflows

YAML files stored in the `./.github/workflows` directory of the repository

Consist of one or more events to trigger the workflow, and one or more jobs to run in response.

Trigger events

Workflows are triggered by any of Github’s web hook events. To run a workflow on code check-ins and pull requests, you might use:

```
on:
  push:
    branches: [“master”]
  pull_request:
```

Triggers

- Pushing to a branch
- Creating a tag
- Creating or updating a pull request
- Creating or updating a bug report
- Manipulating “project” cards (Github’s kanban/trello-like feature)
- Cron-style scheduled workflows
- External triggering via a webhook

Other events are documented here:
https://docs.github.com/en/actions/reference/events-that-trigger-workflows

Jobs

Each job represents work performed on an independent VM.

- Linux (Ubuntu), Windows, MacOS

Linux jobs can run in a Docker container for non-Ubuntu environments.

Jobs run in parallel by default, but you can be ordered via dependencies

A matrix feature allows parameterised jobs

Example job

```
jobs:
  tests:
    name: Run tests
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v1
    - run: |
        make
        make check
```

Matrix

```
strategy:
  matrix:
    var1: [val1, val2, …]
    var2: [val1, val2, …]
```

Matrix variables can be substituted into other job configuration.

A copy of the job is made for the cartesian product of all variables values.

Steps

Each job consists of one or more steps

A step can run some shell commands:

```
- run: |
    # commands go here
```

Or invoke an action:

```
- uses: username/repo@version
```

Actions

Actions are convenient ways to include canned logic to your job.

Even basic functionality like checking out a repository is provided as an action.

Actions can have input parameters and generate outputs that can be used by later steps.

Actions can either be Node.js applications or Docker containers.

Github Provided Actions

Check out a Git repository:

- actions/checkout@v2

Upload or download artifacts:

- actions/upload-artifact@v2
- actions/download-artifact@v2

Cache data between runs:

- actions/cache@v2

Github Provided Actions (2)

Install common tools:

- actions/setup-go@v2
- actions/setup-node@v1
- actions/setup-python@v2
- …

Manipulate issues or pull requests:

- actions/labeler@v2
- actions/stale@v3
- actions/first-interaction@v1

Github Provided Actions (3)

Make releases

- actions/create-release@v1
- actions/upload-release-asset@v1

Writing Actions

Docker Actions:

- Write in any language: you provide your runtime in the container
- Only supports Linux runners
- Slower to set up

Node.js Actions

- Can run on Windows and MacOS runners too
- Runs directly on VM, without a container
- More efficient

Possible Workflows

Categorise issues and pull requests:

- Use actions/labeler@v2 to label new pull requests based on what files they change.
- E.g. “docs” label for PRs changing files in the documentation directory

Possible Workflows (2)

Build a non Jekyll Github Pages website

- Trigger on pushes to master
- Checkout repo
- Build website
- Commit built version of website to “gh-pages” branch
- Push “gh-pages” branch back to Github

Possible Workflows (3)

Create a release on request

- Trigger via “workflow_dispatch” with the version number as an input.
- Checkout repository, and tag HEAD with version number.
- Run tests
- Create packages
- Push version tag back to repository
- Publish release on Github and other services (e.g. PyPI)

Possible Workflows (4)

Check for outdated dependencies

- Check out repository
- Check for new versions upstream
- Compare to what versions you are currently using
- Fail if dependencies are out of date

Alternatively:

- Update dependencies to match upstream and commit to new branch
- Create pull request to merge changes to master

Resources

Workflow syntax:

https://docs.github.com/en/actions/reference/workflow-syntax-for-github-actions

Github published actions:

https://github.com/actions

Actions marketplace:

https://github.com/marketplace?type=actions