QA
POWERING
POTENTIAL

GitHub Actions for CI/CD

WHAT IS GITHUB ACTIONS?

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline. You can create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.

GITHUB ACTIONS TERMINOLOGY

- Workflow
  → An automated process that will run one or more *jobs*, defined by a YAML file in your repository and triggered by an *event*. (or manually, or on a schedule)
- Event
  → An activity in a repository that triggers a workflow run
- Job
  → A set of *steps* in a workflow that execute on the same *runner*. Either a shell script or an *action*.
- Action
  → A custom application for the platform that performs a complex but frequently-repeated task. You can write your own custom actions, or use pre-defined ones
- Runner
  → A server that runs your workflows when triggered. Runs a single job at a time. Ubuntu, Windows and macOS *runners* are provided

HOW TO CREATE A NEW GITHUB ACTIONS WORKFLOW

- Create a `.github/workflows` directory in your repository
- Add a YAML file describing your workflow
- Many example workflows can be found at: https://github.com/actions/starter-workflows

WORKFLOW STATUS

- Success
  - → The workflow succeeded
  - → If all the build *steps* complete successfully, this will be the workflow status.
- Failure
  - → The workflow failed
  - → If any of the *steps* exit with a non-zero status (if they throw an error), then the workflow status will go to failed.

RUNS

- A run is a an execution of a workflow in GitHub Actions
- Runs can be seen in the Actions section of the repository

BASIC SYNTAX

```yaml
name: Name of workflow
on: [list, of, events]
jobs:
  name-of-job: # shows up in build logs
    runs-on: operating-system
    steps:
      - uses: actions/actionname@version
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: some shell script
      - run: another shell script
```

EXAMPLE

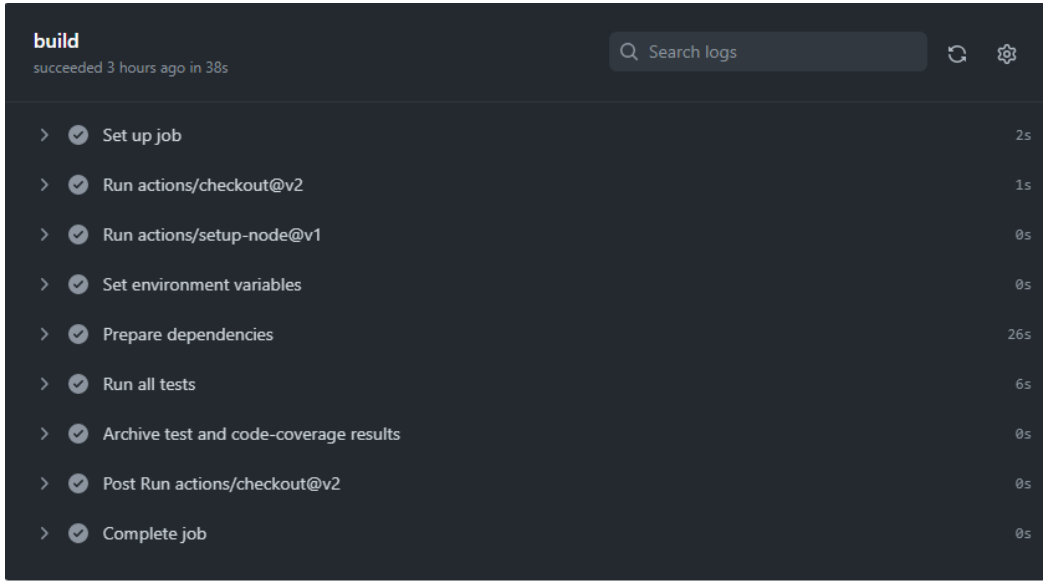
```yaml
name: learn-github-actions
on: [push]
jobs:
  check-bats-version:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '14'
      - run: npm install -g bats
      - run: bats -v
```

BUILD AND TEST EXAMPLE

```yaml
name: Node.js CI

on:
  push:
    branches: [ $default-branch ]
  pull_request:
    branches: [ $default-branch ]

jobs:
  build:

    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [12.x, 14.x, 16.x]

    steps:
    - uses: actions/checkout@v3
    - name: Use Node.js ${{ matrix.node-version }}
      uses: actions/setup-node@v3
      with:
        node-version: ${{ matrix.node-version }}
        cache: 'npm'
    - run: npm ci
    - run: npm run build --if-present
    - run: npm test
```

BUILD LOGS OUTPUT

- The output is likely one of the main parts of a build that you'll be checking for information and debugging purposes
- This section includes the output for any shell scripts and plugins that have been executed in the build step for a workflow



ACTIVATE GITHUB ACTIONS

A newly-forked repository won't have GitHub actions activated, so you'll need to activate them. In your repository, click on the "Actions" tab and then click the green button