ICTPOWER.IT

Microsoft

#POWERCON2022

Deploy with Github Actions a CI/CD overview

Lorenzo Pieri

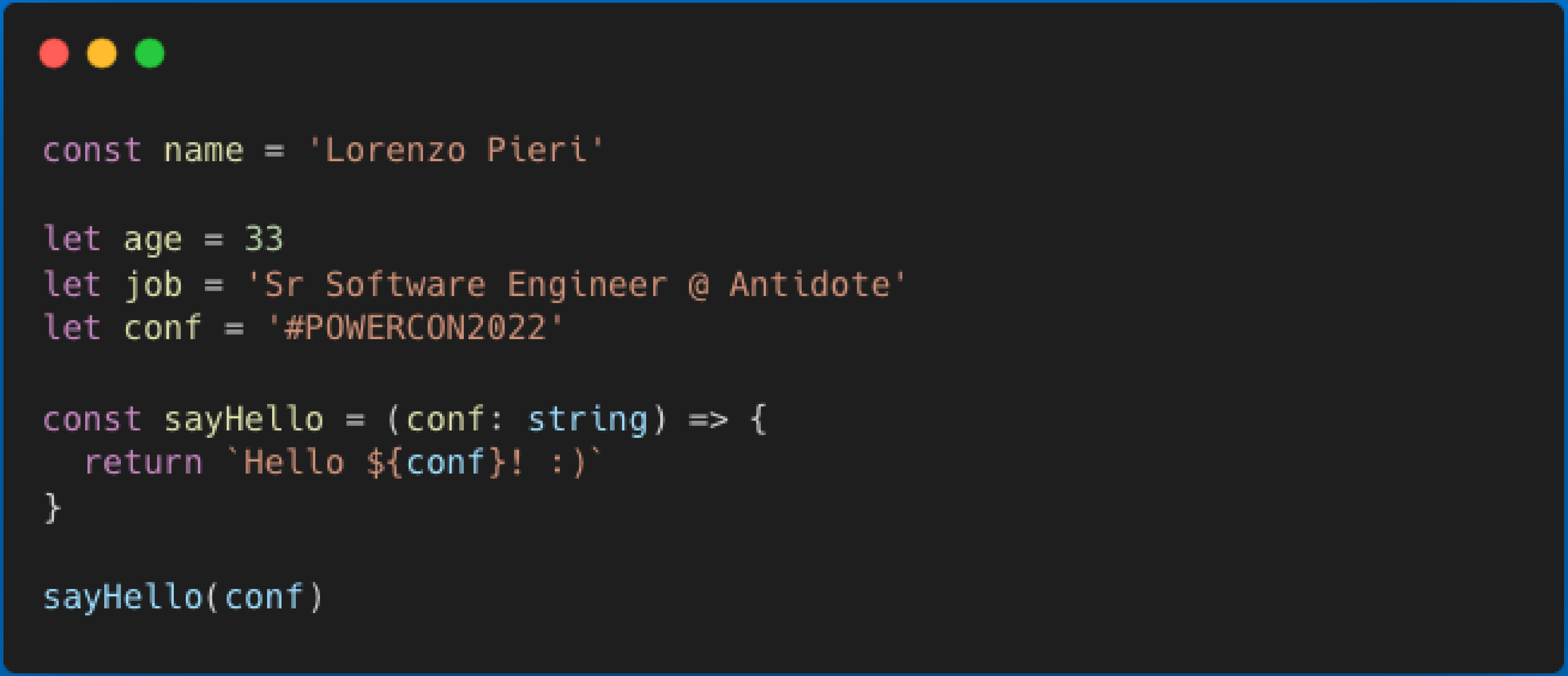
Sr Software Engineer

404answernotfound.public@gmail.com

@404answnotfound

lorenzopieri

Who am I?



```
const name = 'Lorenzo Pieri'

let age = 33
let job = 'Sr Software Engineer @ Antidote'
let conf = '#POWERCON2022'

const sayHello = (conf: string) => {
  return `Hello ${conf}! :)`
}

sayHello(conf)
```

Agenda

- An overview of CI/CD
- What are Github Actions
- Possible use cases
- There's more to it than CI/CD
- Where to go from here

Back to the beginning

An overview of CI/CD

What’s with all the letters

An overview of CI/CD

CI stands for **Continuous Integration**

CD stands for **Continuous Deployment** (or **Delivery**!)

The DevOps methodology is usually showcased as an infinite loop because at the source of its existence, “being in the loop” makes DevOps what it is today. A method for continuity to approach Development in its pitfalls as to avoid Waterfall methodology.

Not all DevOps are born equal

An overview of CI/CD

Are you doing DevOps?
As always, it depends. If your company / team
requires for each step to be pipelined
outside of the scope of continuity, probably not.

You are using a slice of DevOps. You are not doing DevOps.

Show me the tools!

An overview of CI/CD

Docker for image containerization
and artifact management

Github for repository management and workflows

There are many other choices, pick your favourite:

- CircleCI
- Jenkins
- Azure Pipelines
- Google Cloud Build
- BuildBot

The good parts

An overview of CI/CD

The process of CI/CD brings a lot of benefits to the table

- More efficient builds
- Better metrics
- Integration testing on each build
- Faster release times
- More control over fixes and faster knowledge loop
- Powerful tooling
- BYO preferences policy (bring your own)

Enter Github Actions

What are Github Actions

It’s a platform that makes Continuous Integration and Continuous Delivery a developer favourite cookie

We can create:

- Workflows
- Automated pipelines for compiling, testing and distribute
- Scheduled jobs

Take my money!

What are Github Actions

Github Actions has a generous free tier
for public and private repositories
that can run your Actions for up to 2000 minutes.

Action times are counted in minutes as the minimum time usage.

Yet Another Markup Language

What are Github Actions

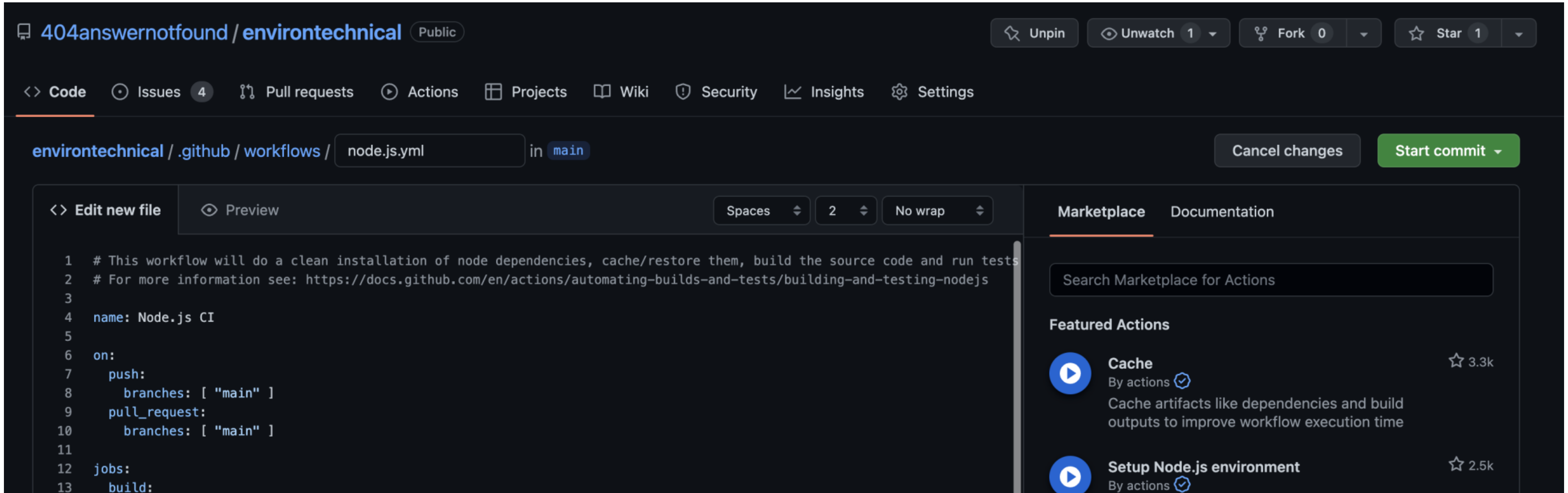
Actions are written in workflow files in the .github/workflows directory

To write them we use a special markup language called YAML with files identified by the .yml or .yaml file extension

My first Github Action

What are Github Actions

Either create a workflow inside .github/workflows directory or create a new workflow from the Github UI directly in your repository ☺



environtechnical / .github / workflows / node.js.yml in main

```
# This workflow will do a clean installation of node dependencies, cache/restore them, build the source code and run tests
# For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-nodejs

name: Node.js CI

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:
```

Cache
By actions
Cache artifacts like dependencies and build outputs to improve workflow execution time

Setup Node.js environment
By actions

My first Github Action

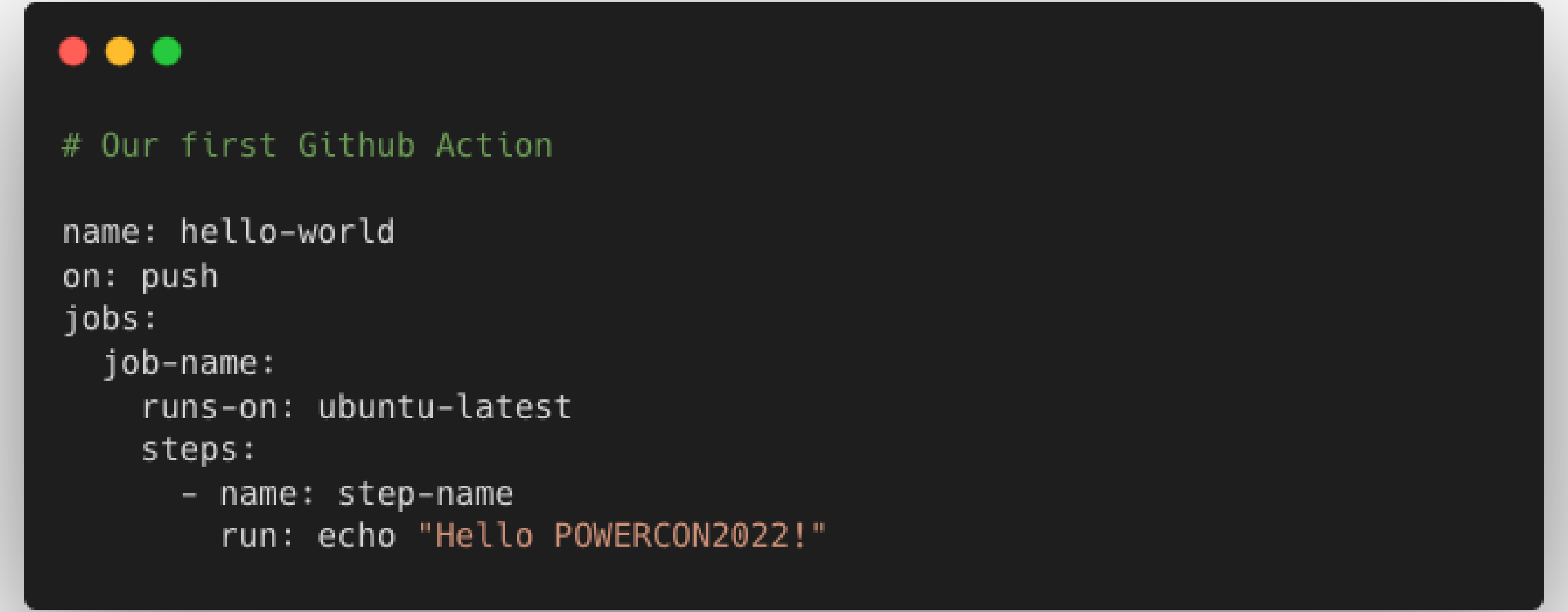
What are Github Actions

An **Action** is a collection of triggers and jobs defined by the user.

A **trigger** binds to an event while a **job** describes all the **steps** that an action will need to fulfill before it’s complete

My first Github Action

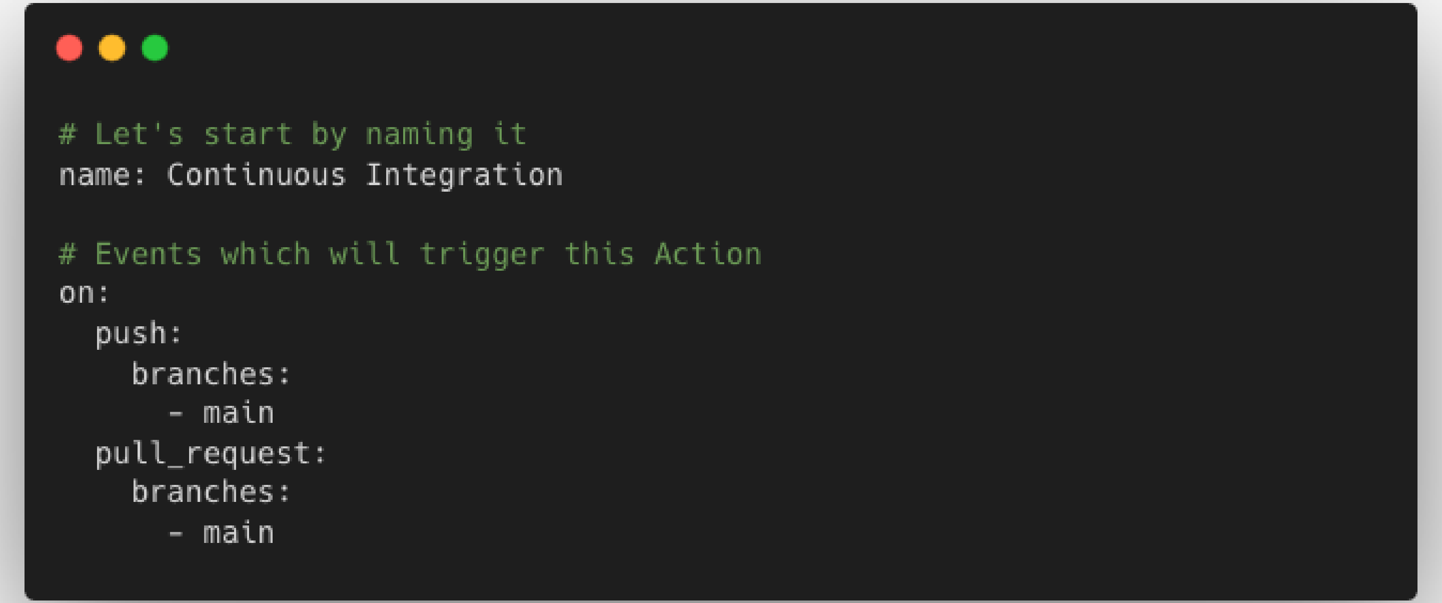
What are Github Actions



```
# Our first Github Action

name: hello-world
on: push
jobs:
  job-name:
    runs-on: ubuntu-latest
    steps:
      - name: step-name
        run: echo "Hello POWERCON2022!"
```

Let’s get real

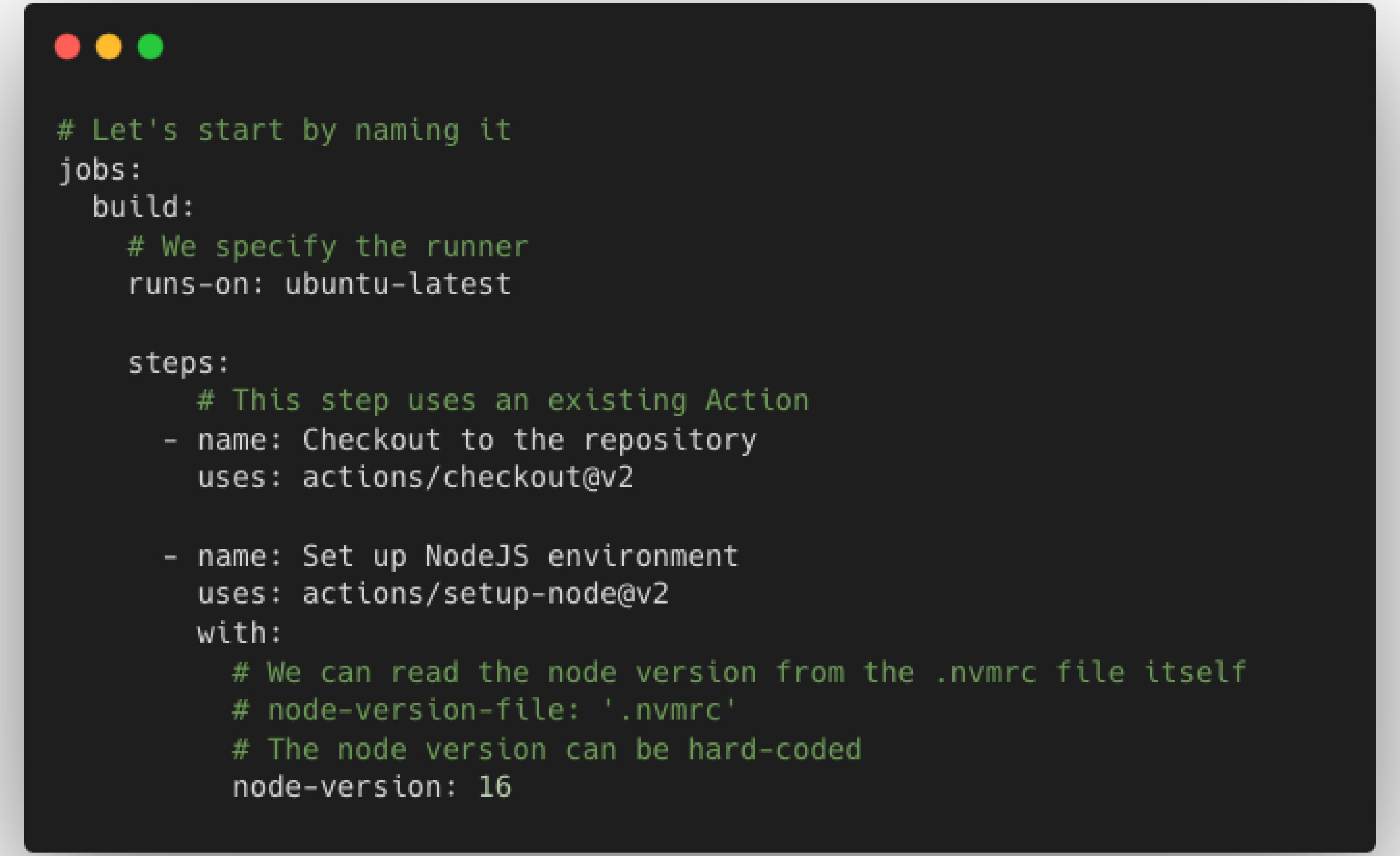
What are Github Actions



```
# Let's start by naming it
name: Continuous Integration

# Events which will trigger this Action
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
```

Let’s get real

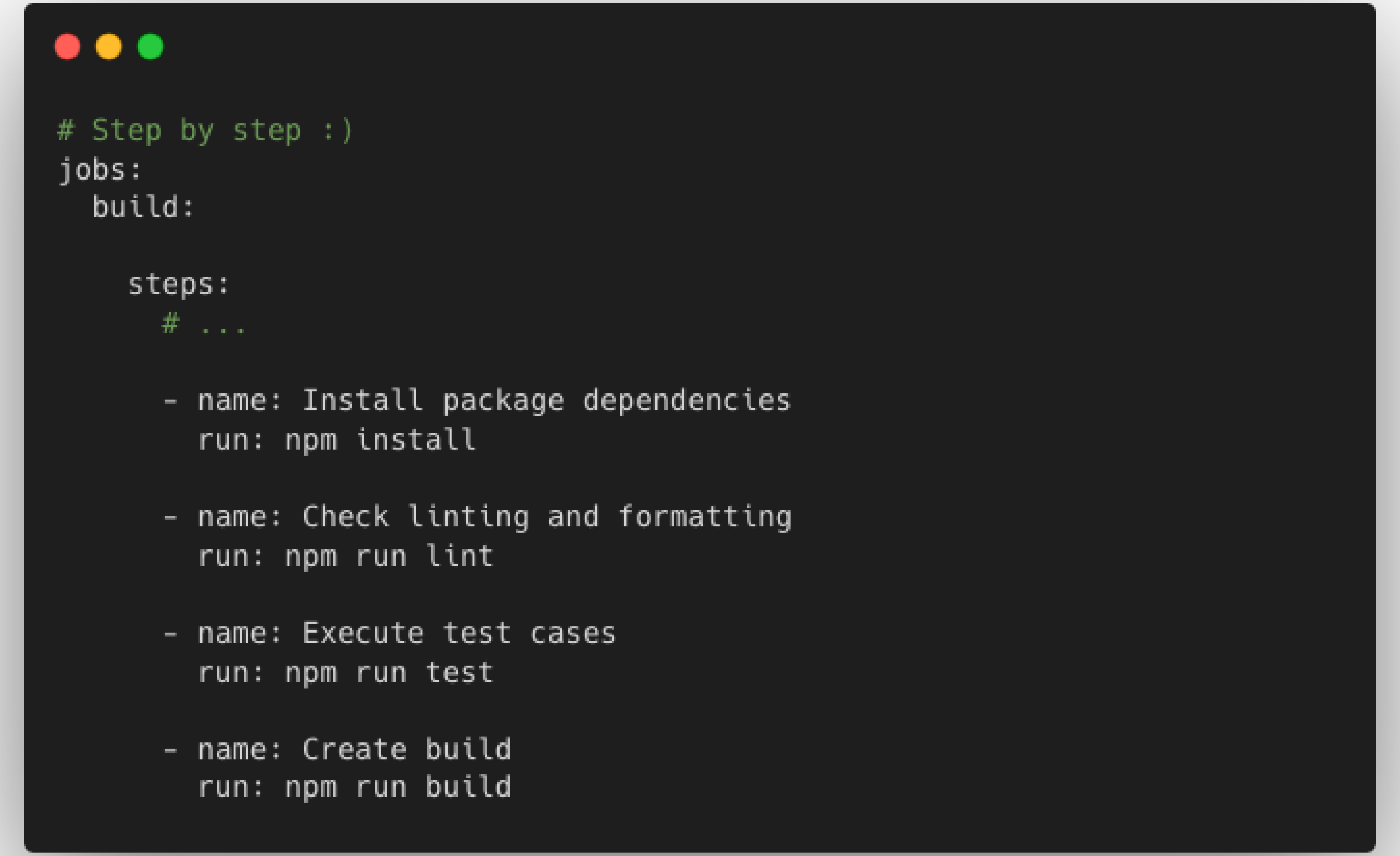
What are Github Actions



```yaml
# Let's start by naming it
jobs:
  build:
    # We specify the runner
    runs-on: ubuntu-latest

    steps:
        # This step uses an existing Action
      - name: Checkout to the repository
        uses: actions/checkout@v2

      - name: Set up NodeJS environment
        uses: actions/setup-node@v2
        with:
          # We can read the node version from the .nvmrc file itself
          # node-version-file: '.nvmrc'
          # The node version can be hard-coded
          node-version: 16
```

Let’s get real

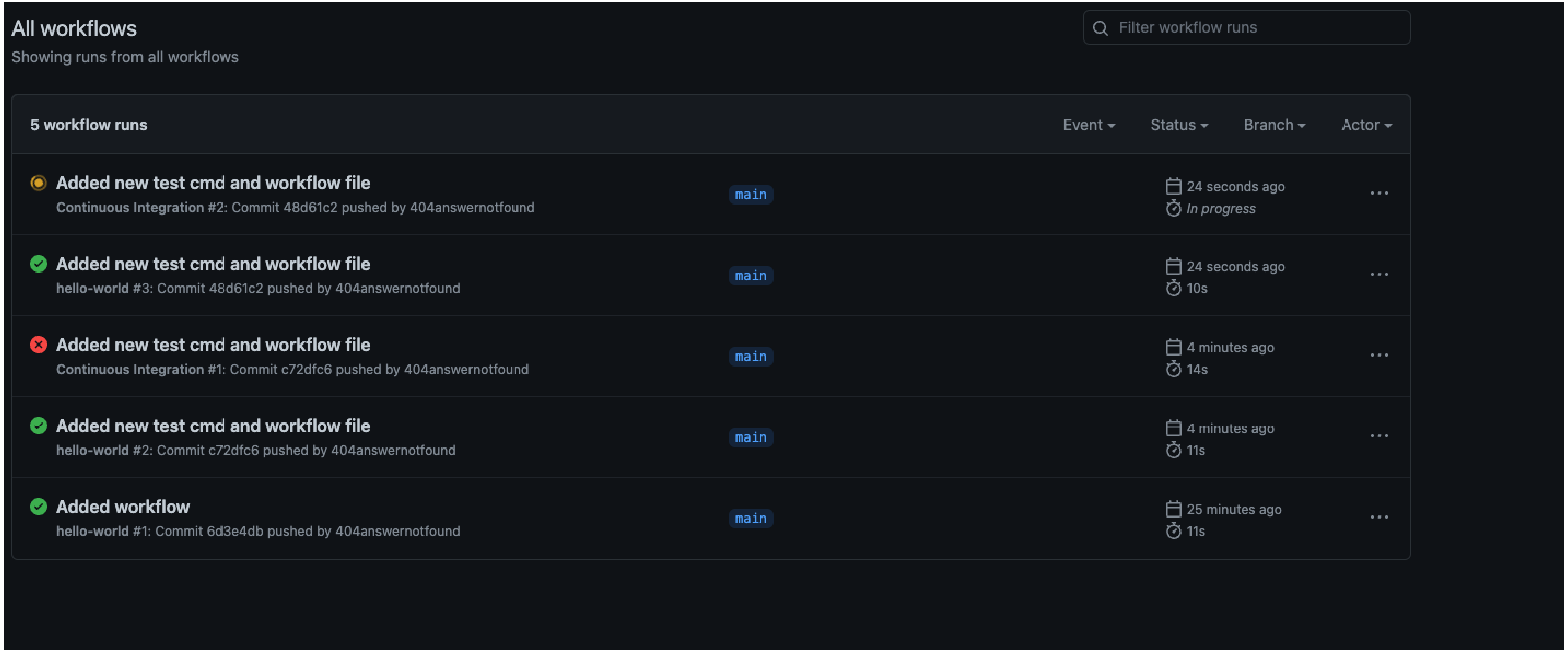
What are Github Actions



```
# Step by step :)
jobs:
  build:

    steps:
      # ...

      - name: Install package dependencies
        run: npm install

      - name: Check linting and formatting
        run: npm run lint

      - name: Execute test cases
        run: npm run test

      - name: Create build
        run: npm run build
```

Behind the curtains

What are Github Actions



All workflows

Showing runs from all workflows

5 workflow runs

● **Added new test cmd and workflow file**
Continuous Integration #2: Commit 48d61c2 pushed by 404answernotfound
main
24 seconds ago
In progress

✓ **Added new test cmd and workflow file**
hello-world #3: Commit 48d61c2 pushed by 404answernotfound
main
24 seconds ago
10s

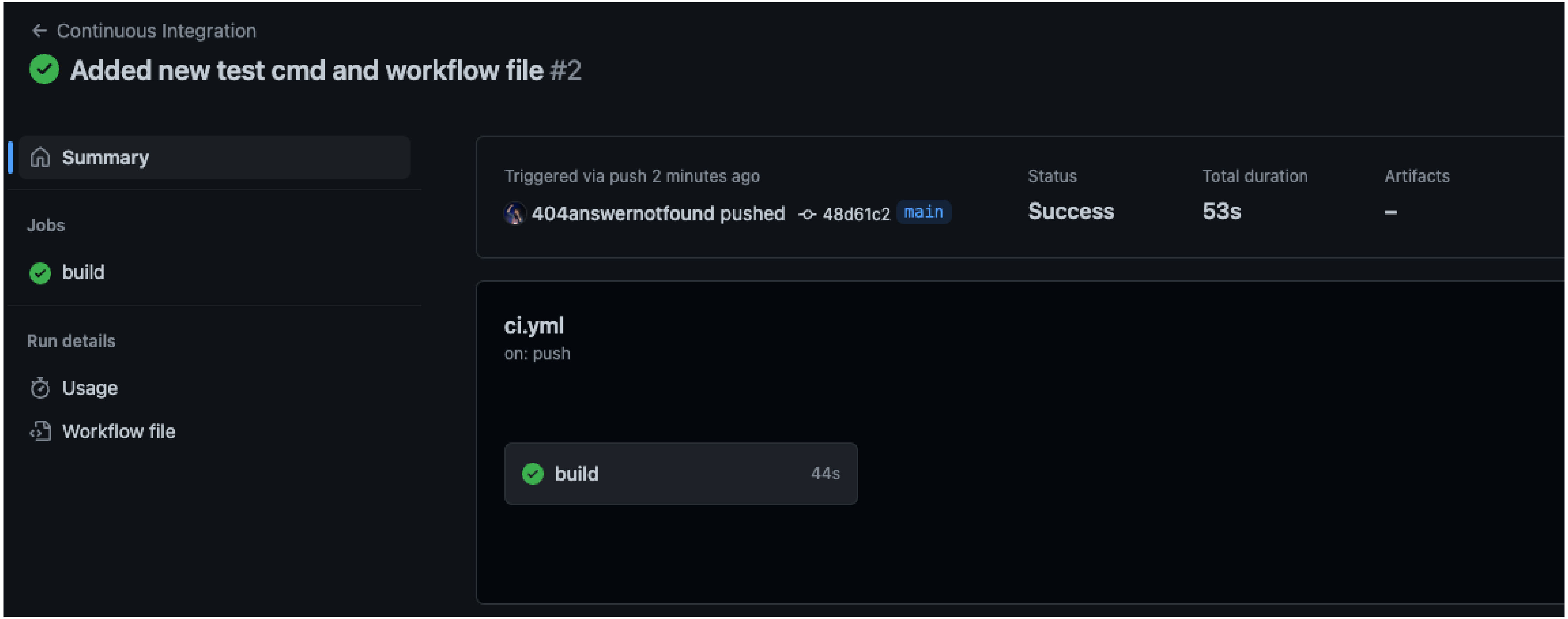
✗ **Added new test cmd and workflow file**
Continuous Integration #1: Commit c72dfc6 pushed by 404answernotfound
main
4 minutes ago
14s

✓ **Added new test cmd and workflow file**
hello-world #2: Commit c72dfc6 pushed by 404answernotfound
main
4 minutes ago
11s

✓ **Added workflow**
hello-world #1: Commit 6d3e4db pushed by 404answernotfound
main
25 minutes ago
11s

Behind the curtains

What are Github Actions



← Continuous Integration

✓ Added new test cmd and workflow file #2

Summary

Jobs

✓ build

Run details

Usage

Workflow file

Triggered via push 2 minutes ago

404answernotfound pushed -o- 48d61c2 main

Status

Success

Total duration

53s

Artifacts

–

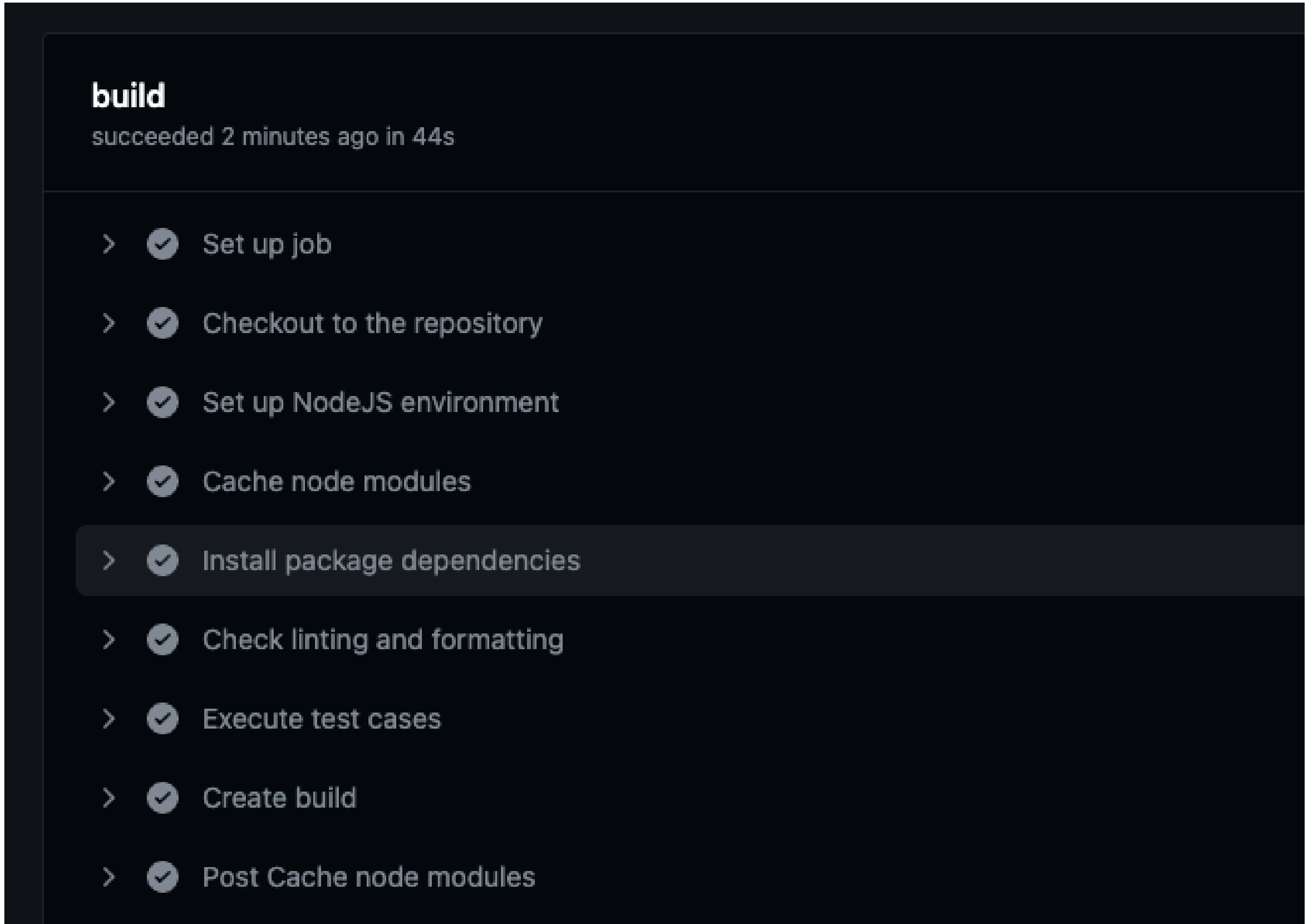
ci.yml

on: push

✓ build 44s

Behind the curtains

What are Github Actions



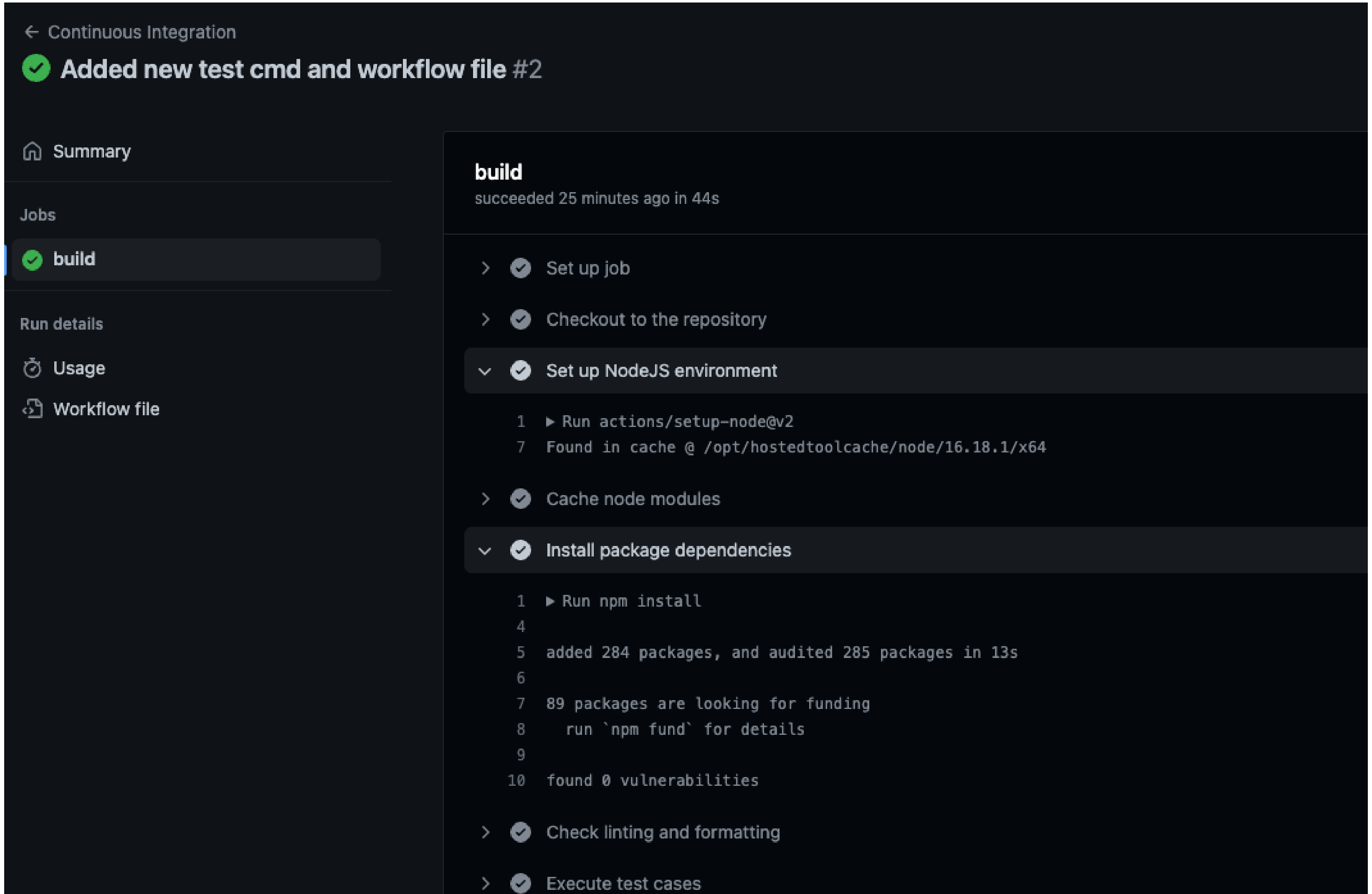
build

succeeded 2 minutes ago in 44s

- Set up job
- Checkout to the repository
- Set up NodeJS environment
- Cache node modules
- Install package dependencies
- Check linting and formatting
- Execute test cases
- Create build
- Post Cache node modules

Behind the curtains

What are Github Actions



← Continuous Integration

✓ Added new test cmd and workflow file #2

build

succeeded 25 minutes ago in 44s

› ✓ Set up job

› ✓ Checkout to the repository

⌄ ✓ Set up NodeJS environment

```
▶ Run actions/setup-node@v2
Found in cache @ /opt/hostedtoolcache/node/16.18.1/x64
```

› ✓ Cache node modules

⌄ ✓ Install package dependencies

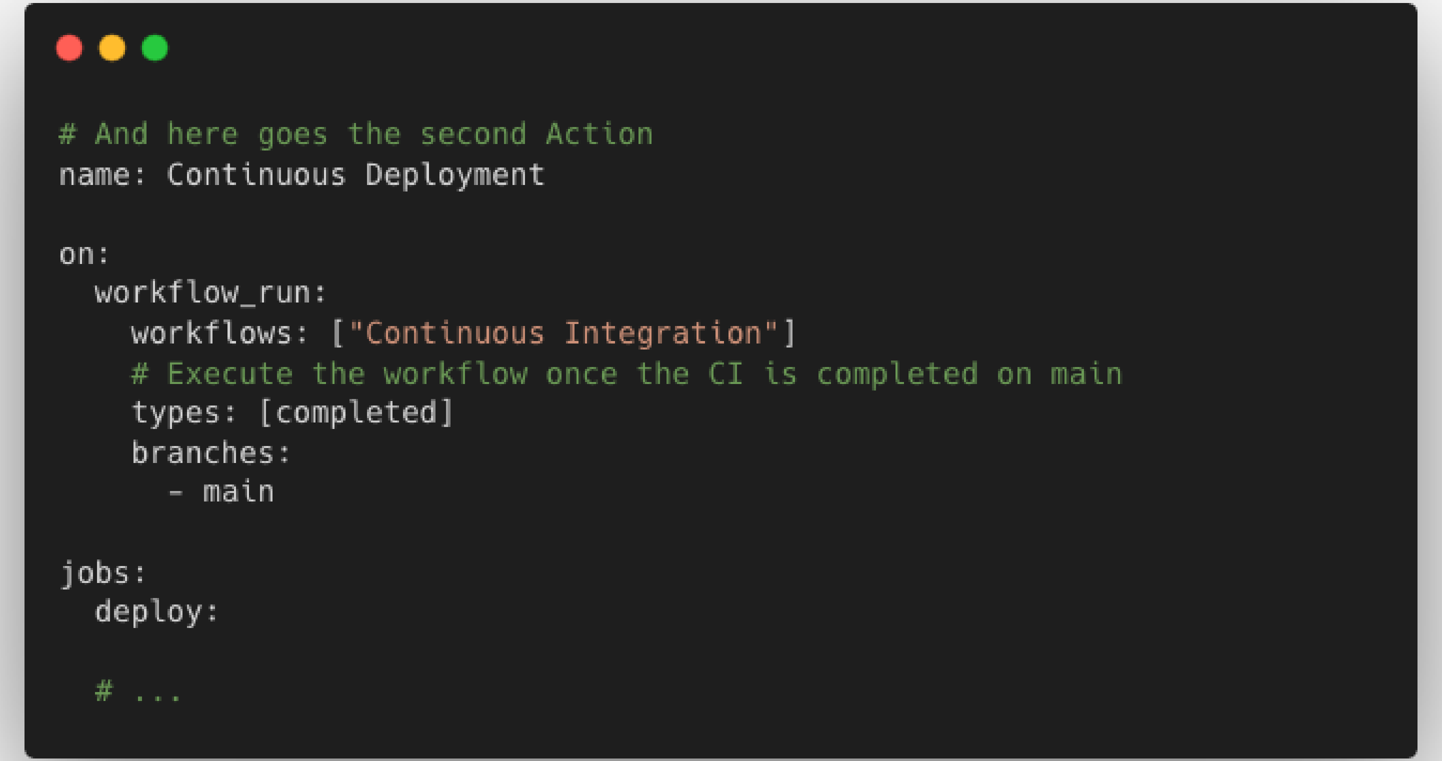
```
▶ Run npm install

added 284 packages, and audited 285 packages in 13s

89 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

› ✓ Check linting and formatting

› ✓ Execute test cases

The more the merrier

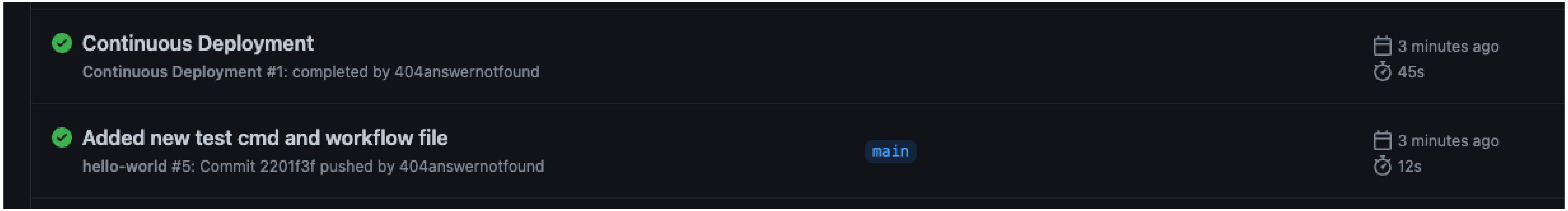
What are Github Actions



```yaml
# And here goes the second Action
name: Continuous Deployment

on:
  workflow_run:
    workflows: ["Continuous Integration"]
    # Execute the workflow once the CI is completed on main
    types: [completed]
    branches:
      - main

jobs:
  deploy:

  # ...
```

The more the merrier

What are Github Actions

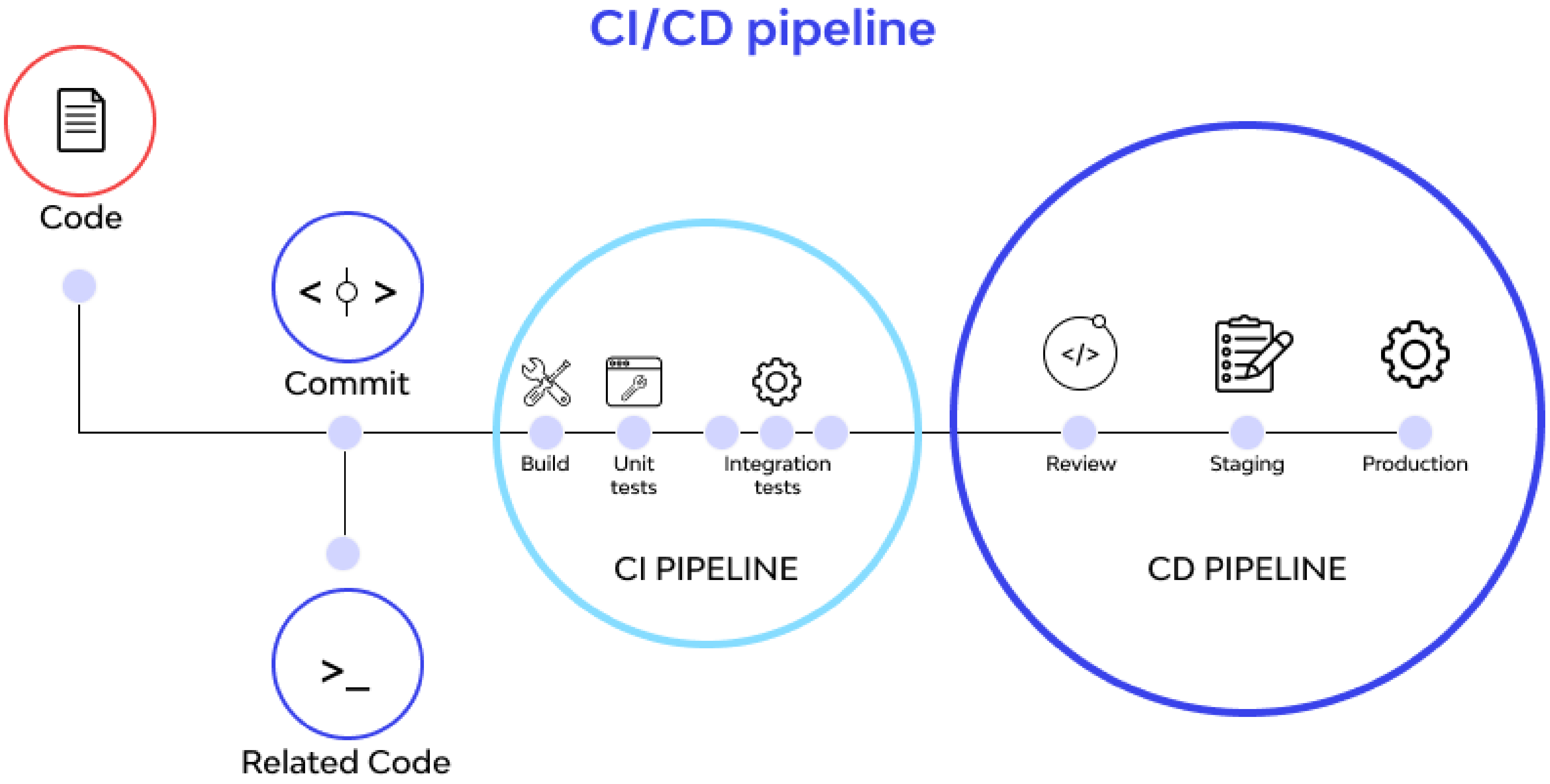


✓ **Continuous Deployment**
Continuous Deployment #1: completed by 404answernotfound
3 minutes ago
45s

✓ **Added new test cmd and workflow file**
hello-world #5: Commit 2201f3f pushed by 404answernotfound
main
3 minutes ago
12s

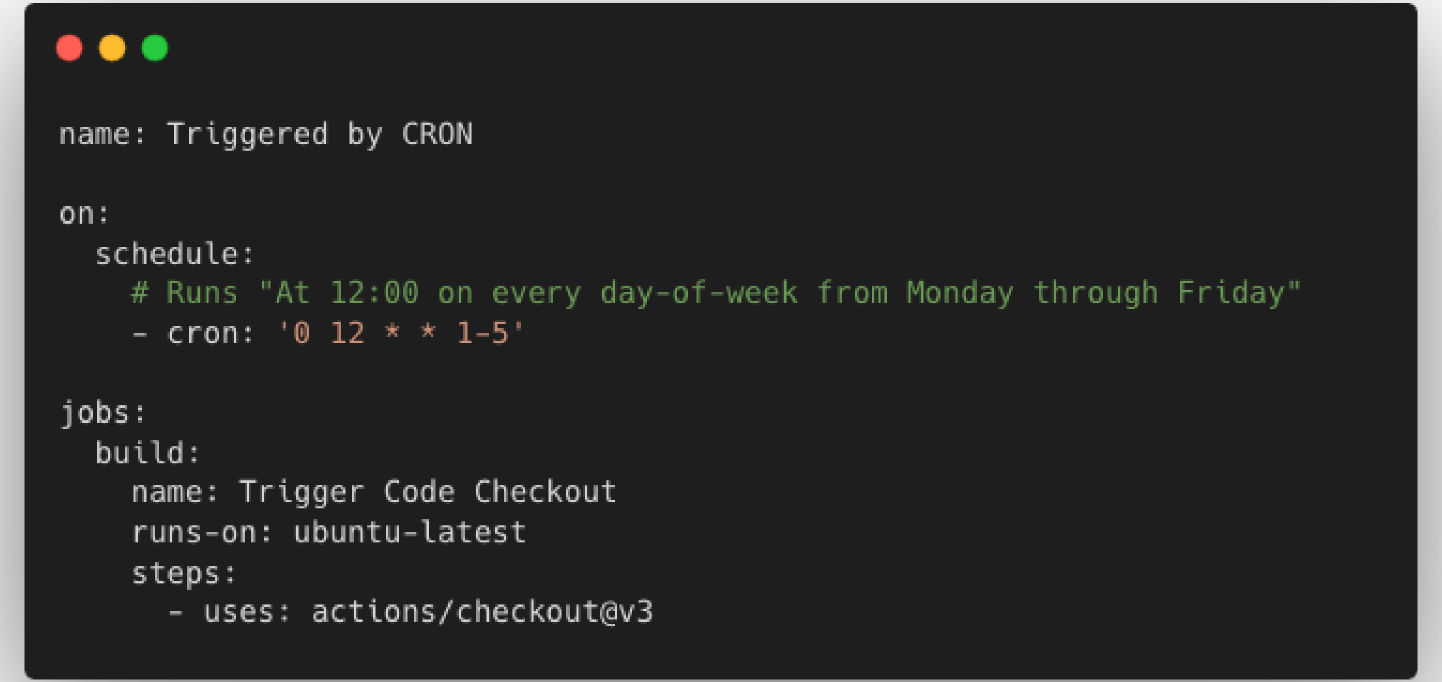
The CI/CD pipeline

Possible use cases



Schedule all the things

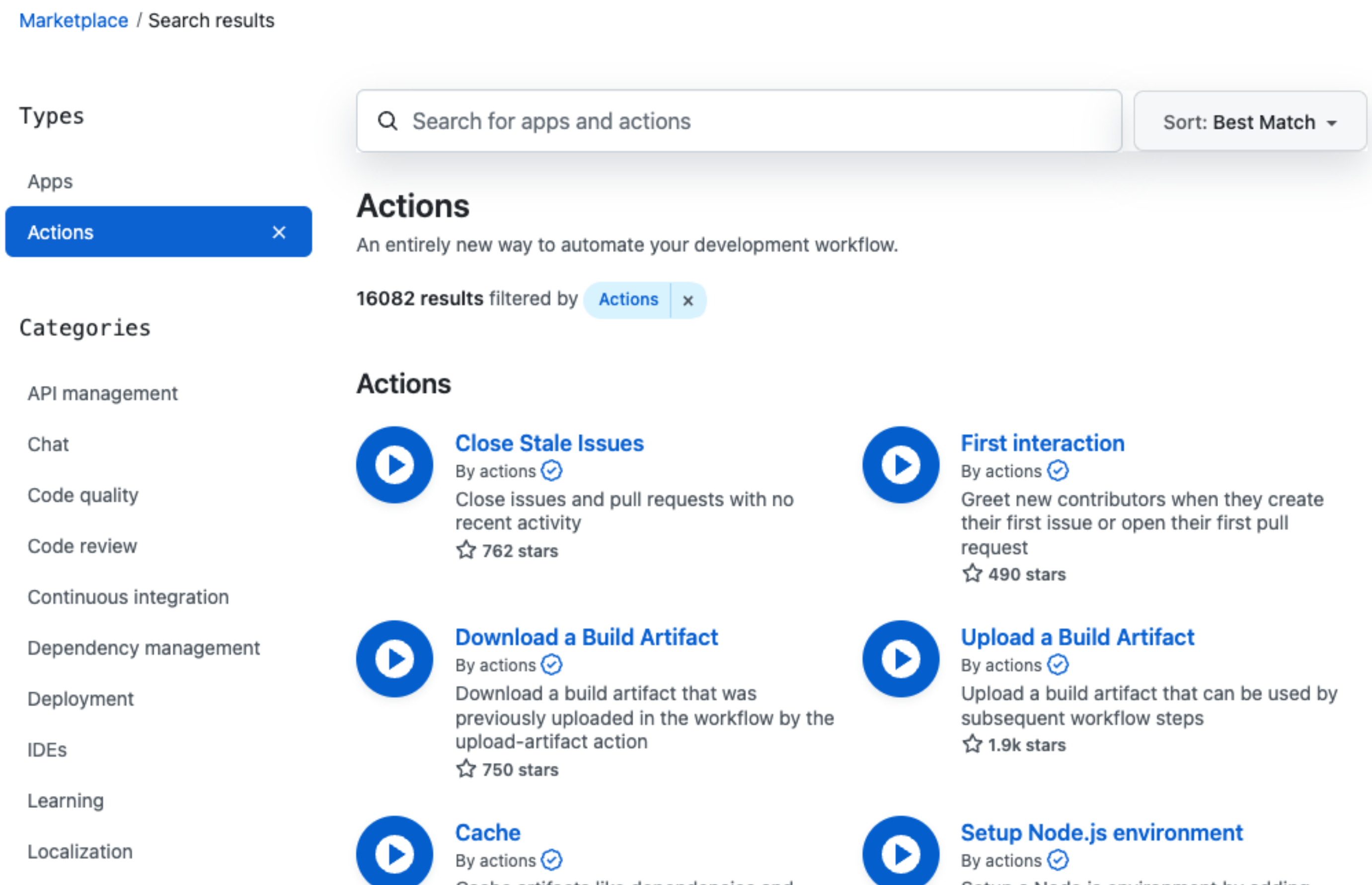
Possible use cases



```yaml
name: Triggered by CRON

on:
  schedule:
    # Runs "At 12:00 on every day-of-week from Monday through Friday"
    - cron: '0 12 * * 1-5'

jobs:
  build:
    name: Trigger Code Checkout
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
```

Is that all?

There's more to it than CI/CD



Marketplace / Search results

Types

- Apps
- Actions ×

Categories

- API management
- Chat
- Code quality
- Code review
- Continuous integration
- Dependency management
- Deployment
- IDEs
- Learning
- Localization

Search for apps and actions

Sort: Best Match

Actions

An entirely new way to automate your development workflow.

16082 results filtered by Actions ×

Actions

Close Stale Issues
By actions
Close issues and pull requests with no recent activity
☆ 762 stars

First interaction
By actions
Greet new contributors when they create their first issue or open their first pull request
☆ 490 stars

Download a Build Artifact
By actions
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action
☆ 750 stars

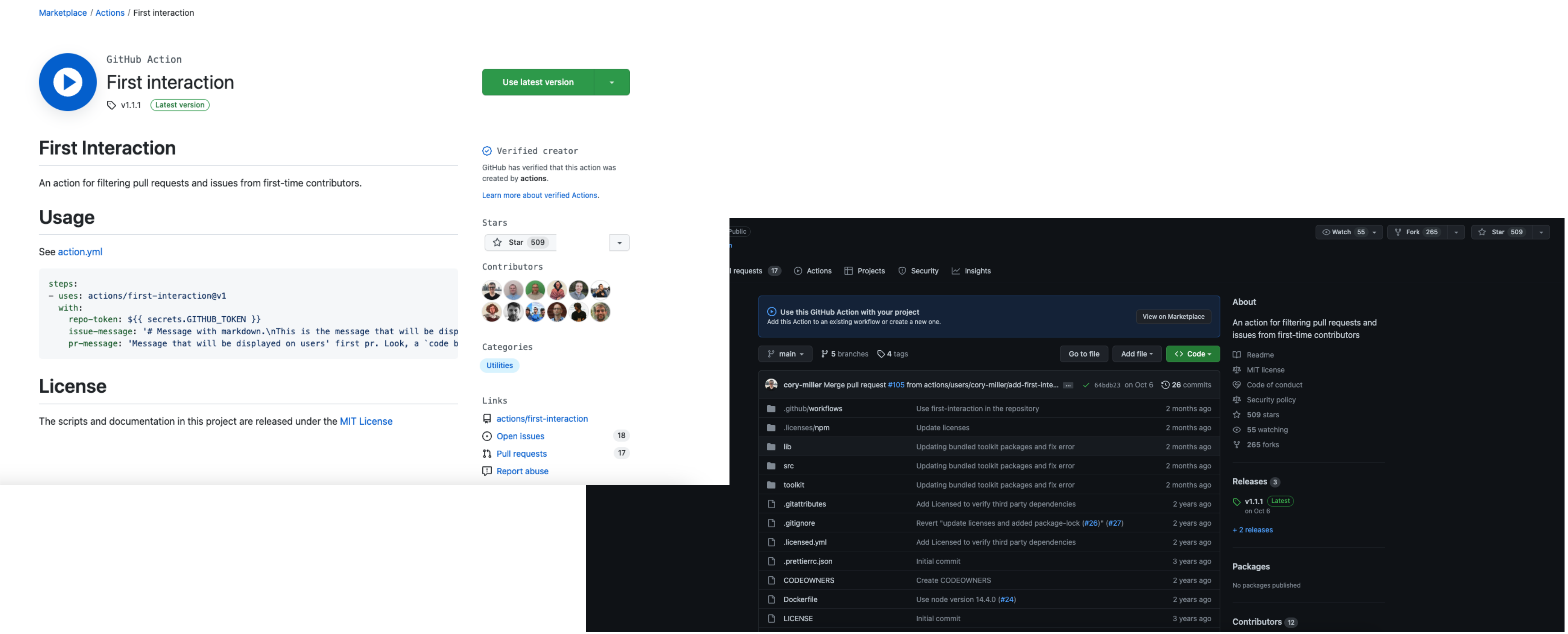
Upload a Build Artifact
By actions
Upload a build artifact that can be used by subsequent workflow steps
☆ 1.9k stars

Cache
By actions

Setup Node.js environment
By actions

Cinderella’s style, the perfect fit

There’s more to it than CI/CD



First interaction

v1.1.1 Latest version

First Interaction

An action for filtering pull requests and issues from first-time contributors.

Usage

See action.yml

```
steps:
- uses: actions/first-interaction@v1
  with:
    repo-token: ${{ secrets.GITHUB_TOKEN }}
    issue-message: '# Message with markdown.\nThis is the message that will be disp
    pr-message: 'Message that will be displayed on users' first pr. Look, a `code b
```

License

The scripts and documentation in this project are released under the MIT License

GitHub has verified that this action was created by **actions**.

Learn more about verified Actions.

actions/first-interaction

Open issues 18

Pull requests 17

Report abuse

Use this GitHub Action with your project

Add this Action to an existing workflow or create a new one.

cory-miller Merge pull request #105 from actions/users/cory-miller/add-first-inte... ✓ 64bdb23 on Oct 6 26 commits

.github/workflows	Use first-interaction in the repository	2 months ago
.licenses/npm	Update licenses	2 months ago
lib	Updating bundled toolkit packages and fix error	2 months ago
src	Updating bundled toolkit packages and fix error	2 months ago
toolkit	Updating bundled toolkit packages and fix error	2 months ago
.gitattributes	Add Licensed to verify third party dependencies	2 years ago
.gitignore	Revert "update licenses and added package-lock (#26)" (#27)	2 years ago
.licensed.yml	Add Licensed to verify third party dependencies	2 years ago
.prettierrc.json	Initial commit	3 years ago
CODEOWNERS	Create CODEOWNERS	2 years ago
Dockerfile	Use node version 14.4.0 (#24)	2 years ago
LICENSE	Initial commit	3 years ago

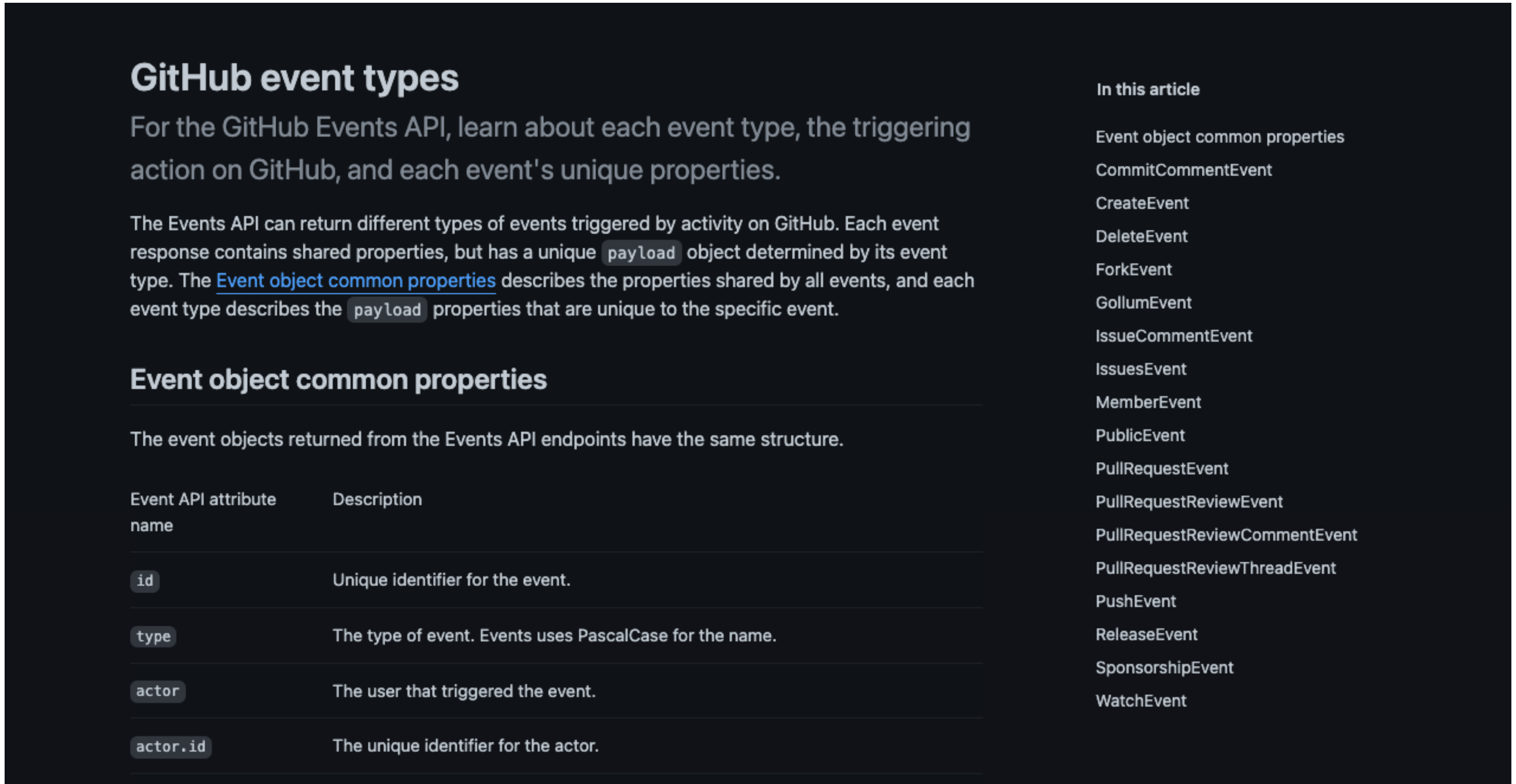
About

An action for filtering pull requests and issues from first-time contributors

No packages published

Automate all the things

There's more to it than CI/CD



GitHub event types

For the GitHub Events API, learn about each event type, the triggering action on GitHub, and each event's unique properties.

The Events API can return different types of events triggered by activity on GitHub. Each event response contains shared properties, but has a unique `payload` object determined by its event type. The Event object common properties describes the properties shared by all events, and each event type describes the `payload` properties that are unique to the specific event.

Event object common properties

The event objects returned from the Events API endpoints have the same structure.

Event API attribute name	Description
`id`	Unique identifier for the event.
`type`	The type of event. Events uses PascalCase for the name.
`actor`	The user that triggered the event.
`actor.id`	The unique identifier for the actor.

In this article

Where to go from here

There are so many more features to **Github Actions** left to learn!

- Understanding Github Actions
- Finding Actions
- Github Actions course

Git is quite important, so:

- Git the simple guide
- Oh shit git!

Bing, Google, Ecosia, DuckDuckGo. Pick your favorite Search Engine and look for great learning material!

ICT POWER.IT

Microsoft

Where to find me

ICTPower blog https://www.ictpower.it/

My personal blog https://404answernotfound.eu/

Medium @404answernotfound

Spotify 404AnswerNotFound Podcast

Community https://osday.dev by SH

@404answnotfound

lorenzopieri

Thank you

Lorenzo Pieri
Sr Software Engineer
404answernotfound.public@gmail.com

@404answnotfound

lorenzopieri