

Build Your First

CI/CD PIPELINE

START

FEB 21 2026



FLORIDA
DRUPAL CAMP

Build Your First CI/CD Pipeline

(And Add a Quality Check While You're At It!)



Amber Matz

Developer Advocate



FLORIDA
DRUPAL CAMP

What will we learn today?

- How I created a sandbox to learn CI/CD concepts
- A process of adding a local CLI tool to a CI/CD check
- 6 essential concepts I learned along the way

What is CI/CD?

CI = Continuous Integration
CD = Continuous Development

In software delivery, the ability to say that the **code that lives on the main** trunk can be **deployed at any time** because **automated checks passed**, or the ability to **automatically ship every change** with confidence because automated checks passed, or **the journey to try and get there**.



A simple CI/CD pipeline with automated checks

The image displays a development environment with two main components: a code editor and a web browser.

Code Editor (VS Code): The editor shows a file named `custom.scss` with the following content:

```
1 /*  
2  * You can add your own custom styles here.  
3  */  
4
```

The terminal at the bottom shows the command being executed:

```
amber@AirOne ~/S/agentolivia.github.io (main)> git checkout amazing-new-wallpaper
```

Web Browser (localhost:1313): The browser displays a personal website for "Not That Kind of Agent" (Amber Matz). The site features a navigation menu with "Home", "Archives", "Search", and "Links". The main content area shows a list of posts:

- New Site, Who Dis** (Monday, February 9, 2026, 5 minutes read)
- Hello World** (Monday, February 9, 2026, 1 minute read)

The footer of the website includes the copyright notice: "© 2026 Not That Kind of Agent", "Built with Hugo", and "Theme Stack designed by Zeno".

Where I Started...

Commit code changes to 'main' and push to repo

Create pull request

Trigger automatic quality checks

Merge pull request

Trigger automatic deployment



My Sandbox

I created and set up a new repo that builds and deploys a Hugo site to GitHub Pages using GitHub Actions.

Used this template

CaiJimmy/hugo-theme-stack-starter

Named the repo

agentolivia.github.io

my username on GitHub

Using GitHub Actions

**Settings > Pages >
Source:
GitHub Actions**

Settings > Pages > Source: GitHub Actions

Actions Projects Wiki Security Insights **Settings** 1

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Models

Webhooks

Copilot

Environments

Codespaces

Pages 2

GitHub Pages

[GitHub Pages](#) is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://agentolivia.github.io/>
Last [deployed](#) by [agentolivia](#) 1 hour ago

[Visit site](#) [Unpublish site](#)

Build and deployment

Source

GitHub Actions

3 **Select GitHub Actions**

Your site was last deployed to the [github-pages](#) environment by the [Pipeline](#) workflow.
[Learn more about deploying to GitHub Pages using custom workflows](#)

Custom domain

Custom domains allow you to serve your site from a domain other than [agentolivia.github.io](#). [Learn more about configuring custom domains.](#)

[Save](#) [Remove](#)

Enforce HTTPS

Set up local

Get the code and your environment set up to build and run the site on your machine.

Code

```
git clone  
...
```

Environment

```
brew install hugo  
brew install go
```

Build and run

```
hugo server
```

Visit localhost:1313

Code editor CLI

git checkout step-1-first-deploy

.github/workflows/deploy.yml

Initial commit · agentolivia/ag... x Not That Kind of Agent x + Gemini

github.com/agentolivia/agentolivia.github.io/commit/b924e34

Filter files...

- .devcontainer
- .github/workflows
 - deploy.yml
 - update-theme.yml
- .gitignore
- .vscode
- LICENSE
- README.md
- assets
- config/_default
- content
- go.mod
- go.sum

35 files changed +831 -0 lines changed

.github/workflows/deploy.yml

```
@@ -0,0 +1,105 @@
1 + name: Build and deploy
2 + on:
3 +   push:
4 +     branches:
5 +       - main
6 +       - master
7 + workflow_dispatch:
8 + permissions:
9 +   contents: read
10 +  pages: write
11 +  id-token: write
12 + concurrency:
13 +   group: pages
14 +   cancel-in-progress: false
15 + defaults:
16 +   run:
17 +     shell: bash
18 + jobs:
19 +   build:
20 +     runs-on: ubuntu-latest
21 +     env:
22 +       DART_SASS_VERSION: 1.97.1
23 +       GO_VERSION: 1.25.5
24 +       HUGO_VERSION: 0.154.2
25 +       NODE_VERSION: 24.12.0
26 +       TZ: Europe/Oslo
27 +     steps:
28 +       - name: Checkout
29 +         uses: actions/checkout@v5
```



Where I Started...After I Got Started

Commit code changes to 'main' and push to repo



Create pull request



Trigger automatic quality checks



Merge pull request to 'main' branch



Trigger automatic deployment



...With a new repo based on the Hugo Stack theme starter template repository.

Initial commit · agentolivia/ag... x Not That Kind of Agent x | + Gemini

github.com/agentolivia/agentolivia.github.io/commit/b924e34

Filter files... 35 files changed +831 -0 lines changed ↑ Top Search within code

.github/workflows/deploy.yml +105

```
@@ -0,0 +1,105 @@
1 + name: Build and deploy
2 + on:
3 +   push:
4 +     branches:
5 +       - main
6 +       - master
7 + workflow_dispatch:
8 + permissions:
9 +   contents: read
10 +  pages: write
11 +  id-token: write
12 + concurrency:
13 +   group: pages
14 +   cancel-in-progress: false
15 + defaults:
16 +   run:
17 +     shell: bash
18 + jobs:
19 +   build:
20 +     runs-on: ubuntu-latest
21 +     env:
22 +       DART_SASS_VERSION: 1.97.1
23 +       GO_VERSION: 1.25.5
24 +       HUGO_VERSION: 0.154.2
25 +       NODE_VERSION: 24.12.0
26 +       TZ: Europe/Oslo
27 +   steps:
28 +     - name: Checkout
29 +       uses: actions/checkout@v5
```

A Process for Adding Your 1st Quality Check

Set up and test a tool locally before adding it to pipeline

Install a CLI tool
locally

Run it and tune
config/ignore
until it's useful

Add command(s)
to 'scripts' key in
package.json

Fix files until tool
returns 0 failures

Commit changes
to version control

1

2

3

4

5



First Quality Check

Install a CLI tool locally. I chose Prettier.

Install

```
npm install  
--save-dev  
prettier
```

Create Config + Ignore

```
.prettierrc  
.prettierignore
```

Add command to scripts

```
"format:check":  
"prettier --check ."
```

package.json
"scripts"



Test it out locally

Run it. Refine config until it's useful.

Run the command

```
npm run  
format:check
```

Modify tool config files

```
.prettierrc  
.prettierignore
```

What did I need?

```
Node.js
```

Add workflow

Create a workflow that will run the command.

Create the file

.github/workflows
.yml extension req'd

Find an Action that fits

Tool's docs
GitHub Action
Marketplace

Install IDE extension

VS Code = GitHub
Actions extension

Get copypasta for GitHub Actions workflow



Code editor CLI

git checkout step-2-add-prettier-to-lint

← Lint

✓ Quality check updates #19

Re-run all jobs



Summary

All jobs

✓ lint

Run details

Usage

Workflow file

lint

succeeded last week in 8s

Search logs



- > ✓ Set up job 1s
- > ✓ Check out code 0s
- > ✓ Set up Node.js 1s
- > ✓ Install dependencies 2s
- ✓ ✓ Run Prettier check 1s
 - 1 ▶ Run npm run format:check
 - 4
 - 5 > agentolivia.github.io@1.0.0 format:check
 - 6 > prettier --check .
 - 7
 - 8 Checking formatting...
 - 9 All matched files use Prettier code style!
- > ✓ Run Markdown check 0s
- > ✓ Post Set up Node.js 0s
- > ✓ Post Check out code 1s
- > ✓ Complete job 0s

← Build and deploy

✓ Merge pull request #3 from agentolivia/quality-check-updates #18

Re-run all jobs



Summary

All jobs



✓ build

✓ deploy

Run details

🕒 Usage

📄 Workflow file

Triggered via push last week

🌐 agentolivia pushed → `e96d766` `main`

Status

Success

Total duration

43s

Artifacts

1

deploy.yml

on: push

✓ build

22s

✓ deploy

11s



Artifacts

Produced during runtime

Name

Size

Digest

📦 github-pages Expired

687 KB

sha256:46da32ba08a3cb620df7f4c1aa39058a56b5e6...

Where are we at?

Commit changes
on branch
and push to repo

Create pull
request

Trigger automatic
quality checks

Merge pull
request to 'main'
branch

Trigger automatic
deployment



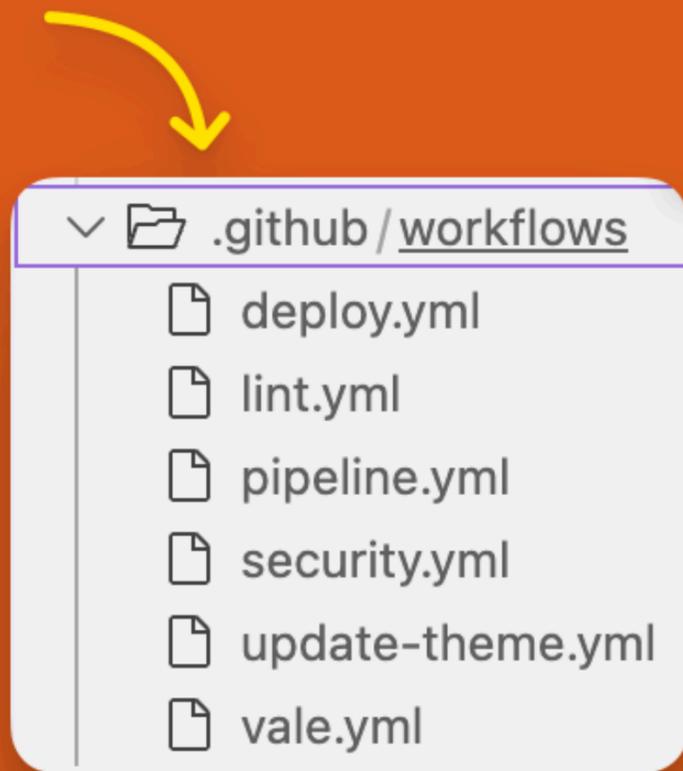
Concept check-in

Concept: The Config File

A version-controlled file in your repo that defines an automation

Describes the *environment*, *triggers*, *jobs*, and *steps* and how they connect into a *pipeline*.

Located and named predictably per platform.



Concept: **The Environment**

```
jobs:  
  build:  
    runs-on: ubuntu-latest  
    env:  
      DART_SASS_VERSION: 1.97.1  
      GO_VERSION: 1.25.7  
      HUGO_VERSION: 0.155.3  
      NODE_VERSION: 24.12.0  
      TZ: America/Los_Angeles
```

`.github/workflows/deploy.yml`

build

succeeded last week in 27s



Set up job

3s

```
1 Current runner version: '2.331.0'
2 ▶ Runner Image Provisioner
9 ▼ Operating System
10 Ubuntu
11 24.04.3
12 LTS
13 ▶ Runner Image
18 ▶ GITHUB_TOKEN Permissions
22 Secret source: Actions
23 Prepare workflow directory
24 Prepare all required actions
25 Getting action download info
26 Download action repository 'actions/checkout@v5' (SHA:93cb6efe18208431cddf8368fd83d5badbf9bfd)
27 Download action repository 'actions/setup-go@v5' (SHA:40f1582b2485089dde7abd97c1529aa768e1baff)
28 Download action repository 'actions/setup-node@v4' (SHA:49933ea5288caeca8642d1e84afbd3f7d6820020)
29 Download action repository 'actions/configure-pages@v5' (SHA:983d7736d9b0ae728b81ab479565c72886d7745b)
30 Download action repository 'peaceiris/actions-hugo@v3' (SHA:75d2e84710de30f6ff7268e08f310b60ef14033f)
31 Download action repository 'actions/cache@v4' (SHA:0057852bf8a89a56745c8a8c7296529d2fc39830)
32 Download action repository 'actions/upload-pages-artifact@v3'
   (SHA:56afc609e74202658d3ffba0e8f6dda462b719fa)
33 Getting action download info
34 Download action repository 'actions/upload-artifact@v4' (SHA:ea165f8d65b6e75b540449e92b4886f43607fa02)
35 Complete job name: build
```

```
jobs:
  build:
    runs-on: ubuntu-latest
    env:
      DART_SASS_VERSION: 1.97.1
      GO_VERSION: 1.25.7
      HUGO_VERSION: 0.155.3
      NODE_VERSION: 24.12.0
      TZ: America/Los_Angeles
```

[.github/workflows/deploy.yml](#)

Concept: Jobs

```
.github > workflows > deploy.yml
1  name: Build and deploy
2  on:
3    workflow_call:
4  permissions:
5    contents: read
6    pages: write
7    id-token: write
8  concurrency:
9    group: pages
10   cancel-in-progress: false
11 defaults:
12   run:
13     shell: bash
14 jobs:
15   build:
16     runs-on: ubuntu-latest
17     > env: ...
23     > steps: ...
91   deploy:
92     > environment: ...
95     runs-on: ubuntu-latest
96     needs: build
97     > steps: ...
101
```

✓ Merge pull request #3 from agentolivia/quality-check-updates #18

Re-run all jobs ⋮

Summary

All jobs ☰

- ✓ build
- ✓ deploy

Run details

- 🕒 Usage
- 📄 Workflow file

Triggered via push last week

🌐 agentolivia pushed [e96d766](#) `main`

Status

Success

Total duration

43s

Artifacts

1

deploy.yml

on: push



🗒 − +

Artifacts

Produced during runtime

Name	Size	Digest
📦 github-pages Expired	687 KB	sha256:46da32ba08a3cb620df7f4c1aa39058a56b5e6... 📄

```
.github > workflows > lint.yml
1 name: Lint
2
3 on:
4   workflow_call:
5
6 jobs:
7   lint:
8     runs-on: ubuntu-latest
9
10    steps:
11      1 - name: Check out code
12         uses: actions/checkout@v4
13
14      2 - name: Set up Node.js
15         uses: actions/setup-node@v4
16         with:
17           node-version: '20'
18
19      3 - name: Install dependencies
20         run: npm ci
21
22      4 - name: Run Prettier check
23         run: npm run format:check
24
25      5 - name: Run Markdown check
26         run: npm run lint:markdown
27
```

Concept: Steps

lint

succeeded last week in 11s

Search logs



- > Set up job 0s
- > Check out code 2s
- Set up Node.js 0s
 - ▶ Run actions/setup-node@v4
 - 7 Found in cache @ /opt/hostedtoolcache/node/20.20.0/x64
 - 8 ▶ Environment details
- > Install dependencies 3s
- > Run Prettier check 0s
- > Run Markdown linting 1s
- > Post Set up Node.js 0s
- > Post Check out code 0s
- > Complete job 0s

```
.github > workflows > lint.yml
1  name: Lint
2
3  on:
4    workflow_call:
5
6  jobs:
7    lint:
8      runs-on: ubuntu-latest
9
10   steps:
11     1 - name: Check out code
12       uses: actions/checkout@v4
13
14     2 - name: Set up Node.js
15       uses: actions/setup-node@v4
16       with:
17         node-version: '20'
18
19     3 - name: Install dependencies
20       run: npm ci
21
22     4 - name: Run Prettier check
23       run: npm run format:check
24
25     5 - name: Run Markdown check
26       run: npm run lint:markdown
27
```

Concept: Triggers

Triggered via push now ↖

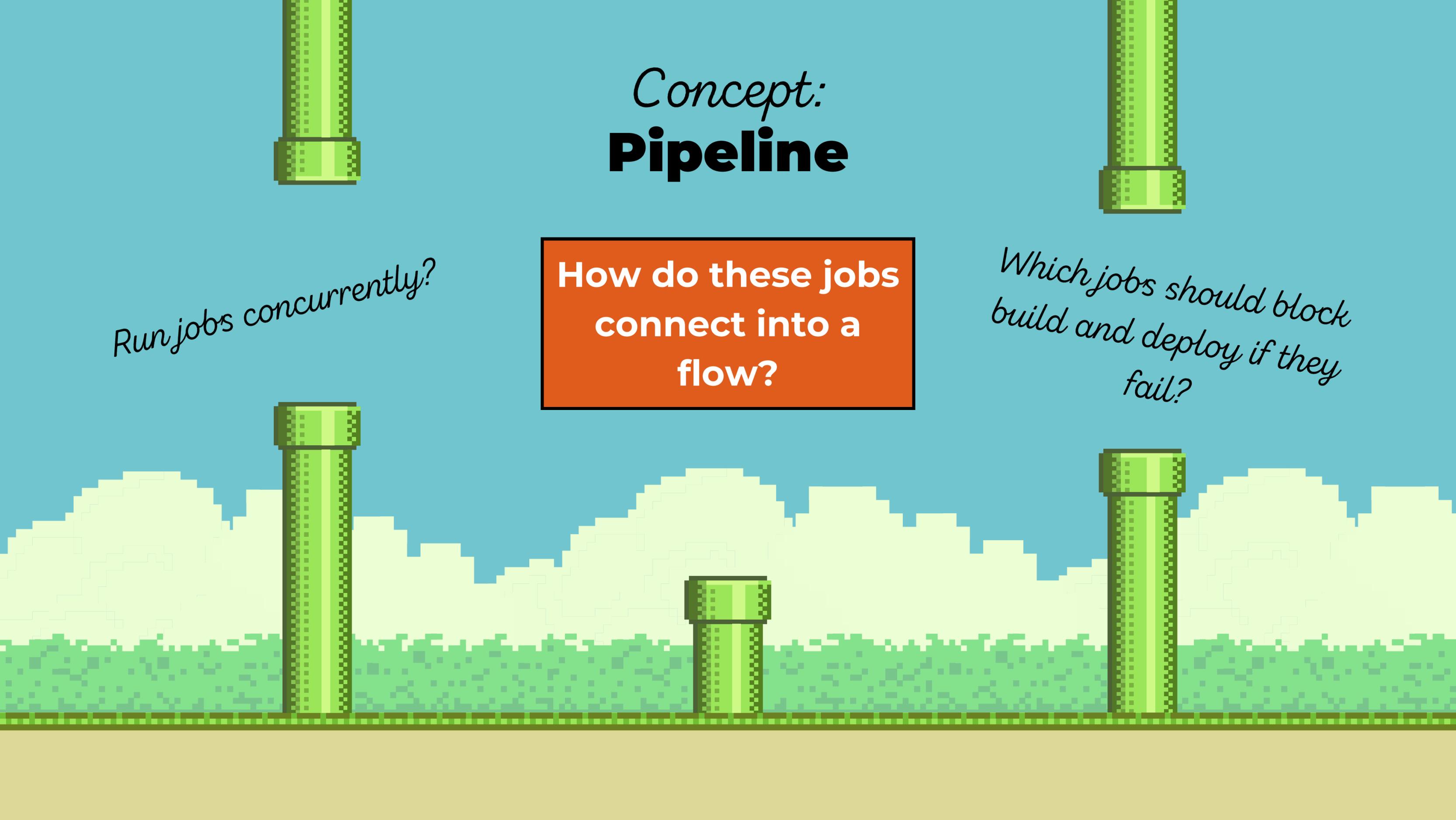
	Status	Total duration	Artifacts
agentolivia pushed main	In progress	=	-

pipeline.yml
on: push

```
graph LR; A[lint / lint 10s] --> B[security / security 9s]; B --> C[deploy / build 8s]; C --> D[deploy / deploy];
```

lint / lint 10s
security / security 9s
deploy / build 8s
deploy / deploy

```
on:  
  push:  
    branches:  
      - main  
  pull_request:  
    branches:  
      - main
```



Concept: **Pipeline**

**How do these jobs
connect into a
flow?**

Run jobs concurrently?

*Which jobs should block
build and deploy if they
fail?*

On GitHub:
Update pipeline and test lint failure#7

<https://github.com/agentolivia/agentolivia.github.io/pull/7>

**On GitHub:
Refactored security check and updated package
to remove vulnerability#8**

<https://github.com/agentolivia/agentolivia.github.io/pull/8>

Code editor CLI

git checkout step-3-pipeline-refactor

So much quality to check!

Code quality & formatting

Linters

Formatters

Visual diffs

SEO

Performance

Accessibility

Manual QA

User Acceptance Testing

Peer code review

Automated testing

Unit tests

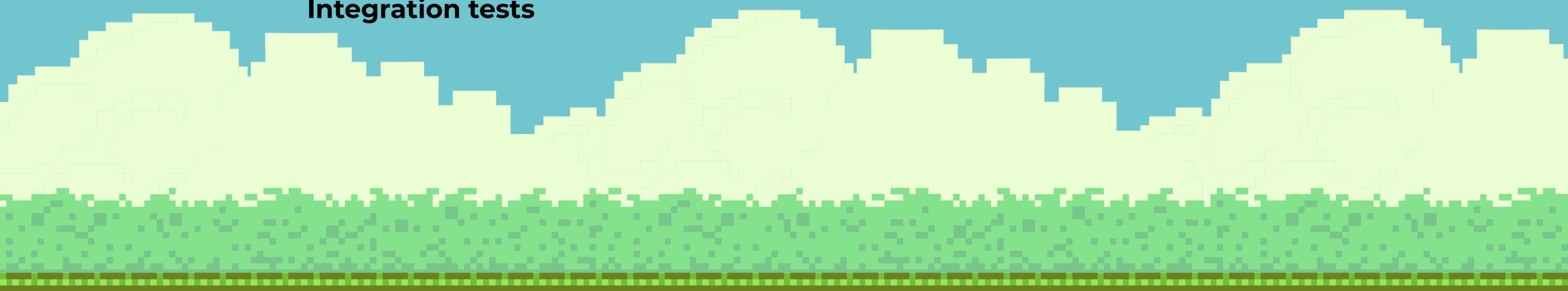
End-to-end (E2E)

Integration tests

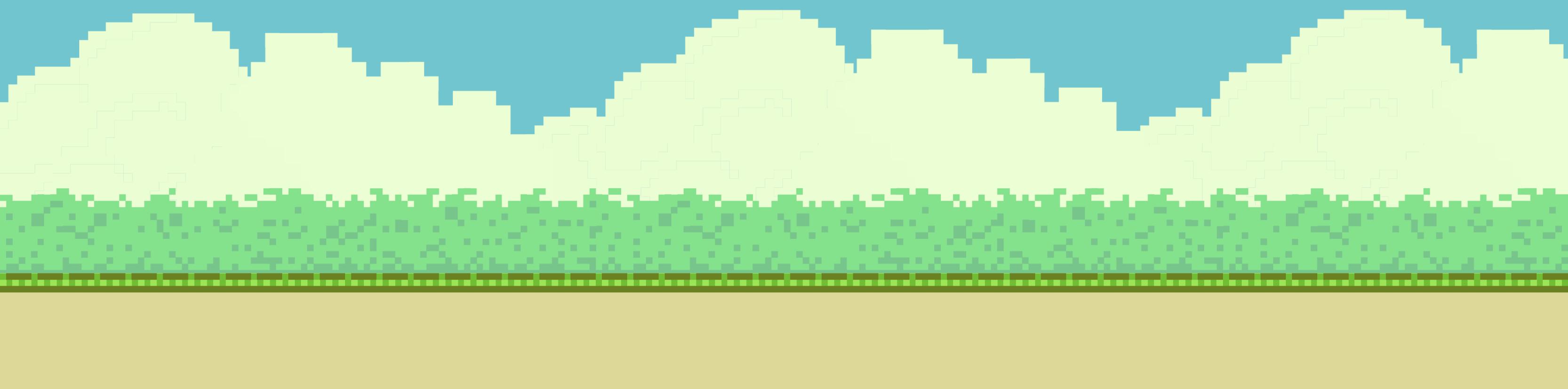
Supply chain security

Dependency updates

Design review



What is **one tool** you **love to use** in your IDE that **you could add to a CI/CD pipeline** that would run **automatically** on every pull request?



final boss demo



Where am I at now?

Commit changes
on branch
and push to repo

Create pull
request

Trigger automatic
quality checks

Merge pull
request to 'main'
branch

Trigger automatic
deployment



Now with more pipeline!

What did we learn?

- How I created a sandbox to learn CI/CD concepts
- A process of adding a local CLI tool to a CI/CD check
- 6 essential concepts

Config File!

Environment!

Jobs!

Steps!

Triggers!

The Pipeline FTW!

Now Go and Build Your First CI/CD Pipeline

(And Add a Quality Check While You're At It!)



Amber Matz

Developer Advocate



tugboatqa.com



FLORIDA
DRUPAL CAMP

Resources

My repo:

<https://github.com/agentolivia/agentolivia.github.io>

GitHub Actions Marketplace

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

GitHub Actions Docs

<https://docs.github.com/en/actions>