

Building a CI/CD pipeline with GitHub Actions

Welcome back to another article. If you are a cloud engineer, devops engineer or site reliability engineer, chances are you know about CI CD pipeline. Thus, if you are new to cloud engineering you have noticed that there are several DevOps tools, so CI CD tools.

What do you know about CI CD? Have you used Github Actions to build a CI CD pipeline? What is GitHub Actions? Why should you use Actions? When do you use Actions?

In the following guide, I will describe the steps by steps to build a CI/CD pipeline with GitHub Actions.

Background

CI CD refers to Continuous Integration, Continuous Deployment. A **CI/CD pipeline** is the process of building, testing, and deploying code in a SDLC. A CI pipeline runs when code changes, it compiles your code, run tests, and check that it's functional. A CD pipeline deploys the built code into production. **Github Actions** is a CI/CD tool that gives you the ability to implement powerful automations right in your repositories.

Prerequisites

- A Github account, if you don't have an account get one [here for free](#).

Now let's have fun building your CI CD pipeline using Github Actions.

First step: Create a repository or select an existing repo, and a project

- Create or select a repository on GitHub to build a CI pipeline with GitHub Actions.

The screenshot shows the GitHub 'Create repository' form. At the top, there are two fields: 'Owner' with a dropdown menu showing 'djcloudking' and a 'Repository name' field. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about `supreme-winner`?'. There is a 'Description (optional)' text area. Below the description is a section for repository visibility with two options: 'Public' (selected) and 'Private'. The 'Public' option has a description: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a description: 'You choose who can see and commit to this repository.' Below this is a section 'Initialize this repository with:' with a subtext 'Skip this step if you're importing an existing repository.' There is a checkbox 'Add a README file' with a description: 'This is where you can write a long description for your project. [Learn more.](#)'. Below this is a section 'Add .gitignore' with a subtext 'Choose which files not to track from a list of templates. [Learn more.](#)'. There is a dropdown menu for '.gitignore template' currently set to 'None'. Below this is a section 'Choose a license' with a subtext 'A license tells others what they can and can't do with your code. [Learn more.](#)'. There is a dropdown menu for 'License' currently set to 'None'. At the bottom, there is a note: 'You are creating a public repository in your personal account.' and a green 'Create repository' button.

Owner ^{*} Repository name ^{*}

djcloudking /

Great repository names are short and memorable. Need inspiration? How about `supreme-winner`?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

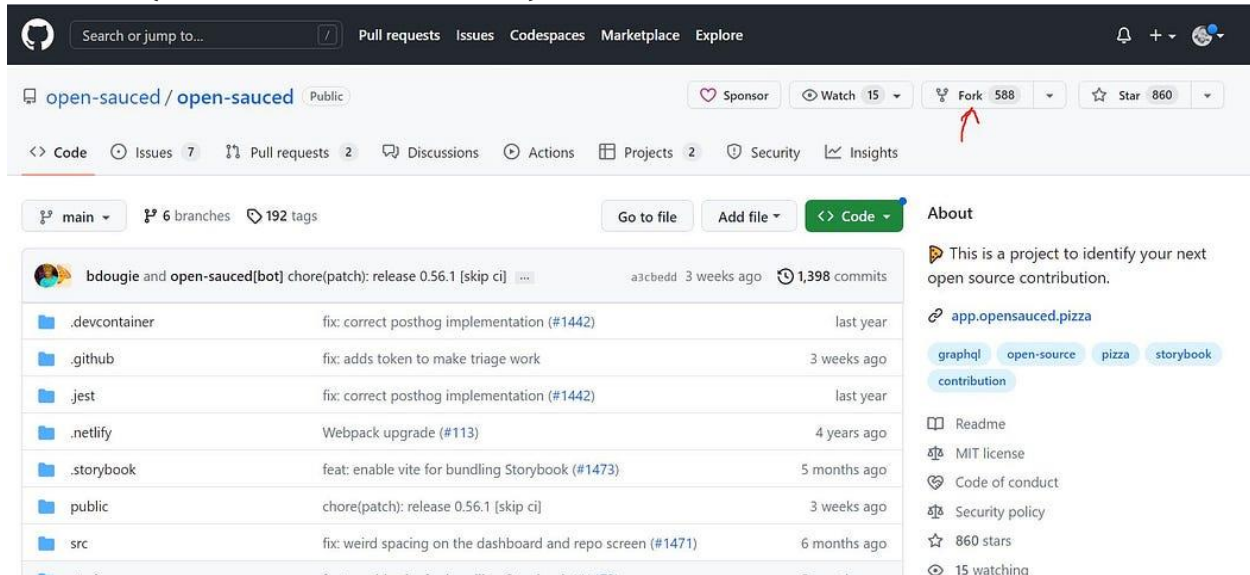
☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

You are creating a public repository in your personal account.

Cheat Sheet: you can either use an existing project code base, fork a project you like on GitHub (see screenshot below), or start from scratch (see screenshot above).



Open sauced repository

- For this guide I decided to fork this [Open Sauced repository](#).

Cheat Sheet: A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks](#).

Owner *

 djcloudking ▾

Repository name *

/ open-sauced ✓

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

 This is a project to identify your next open source contribution.

☒ Copy the `main` branch only

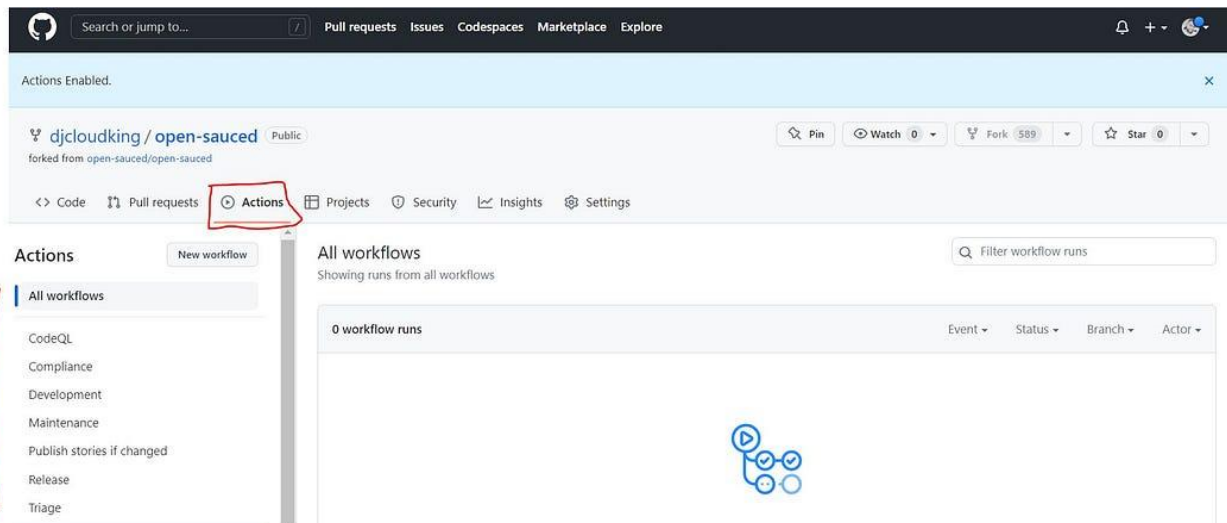
Contribute back to open-sauced/open-sauced by adding your own branch. [Learn more](#).

 You are creating a fork in your personal account.

Create fork

Second Step : Open GitHub Actions in your repository to start building your CI/CD workflow

- Go to your repository's top navigation bar.
- Open the GitHub Actions tab



- From a list of CI/CD and workflow automation templates click on **New workflow** (see screenshot below) or select one in **All workflows** list (see screenshot above).

Choose a workflow

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

[Skip this and set up a workflow yourself →](#)

Q Search workflows

Suggested for this repository

Docker image
By GitHub Actions

Build a Docker image to deploy, run, or push to a registry.

[Configure](#)

Dockerfile

Deno
By GitHub Actions

Test your Deno project

[Configure](#)

JavaScript

Grunt
By GitHub Actions

Build a NodeJS project with npm and grunt.

[Configure](#)

JavaScript

Jekyll using Docker image
By GitHub Actions

Package a Jekyll site using the jekyll/builder Docker image.

[Configure](#)

HTML

Gulp
By GitHub Actions

Build a NodeJS project with npm and gulp.

[Configure](#)

JavaScript

Webpack
By GitHub Actions

Build a NodeJS project with npm and webpack.

[Configure](#)

JavaScript

Deployment

[View all](#)

Deploy a container to an Azure Web App

By Microsoft

Deploy to Amazon ECS
By Amazon Web Services

aws

Build and Deploy to GKE
By Google Cloud

Google Cloud

Deploy to Alibaba Cloud ACK
By Alibaba Cloud

Alibaba Cloud

Security

[View all](#)

CodeQL Analysis
By GitHub
Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby and Kotlin developers.
[Configure](#) Code scanning

SonarCloud
By Sonar
Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!
[Configure](#) Code scanning

SonarQube
By Sonar
Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!
[Configure](#) Code scanning

Fortify on Demand Scan
By Micro Focus
Integrate Fortify's comprehensive static code analysis (SAST) for 27+ languages into your DevSecOps workflows to build secure software faster.
[Configure](#) Code scanning

Continuous integration

[View all](#)

Publish Docker Container
By GitHub Actions
Build, test and push Docker image to GitHub Packages.
[Configure](#) Dockerfile

SLSA Generic generator
By Open Source Security Foundation (OpenSSF)
Generate SLSA3 provenance for your existing release workflows
[Configure](#) Go

Datadog Synthetics
By Datadog
Run Datadog Synthetic tests within your GitHub Actions workflow
[Configure](#) JavaScript

Node.js
By GitHub Actions
Build and test a Node.js project with npm.
[Configure](#) JavaScript

Automation

Stale
By GitHub Actions
Checks for stale issues and pull requests
[Configure](#) Automation

Manual workflow
By GitHub Actions
Simple workflow that is manually triggered.
[Configure](#) Automation

Labeler
By GitHub Actions
Labels pull requests based on the files changed
[Configure](#) Automation

Greetings
By GitHub Actions
Greets users who are first time contributors to the repo
[Configure](#) Automation

- Remember we forked an existing repository for this exercise. Go back to the repo and click on workflow. You are able to see all the existing workflows.

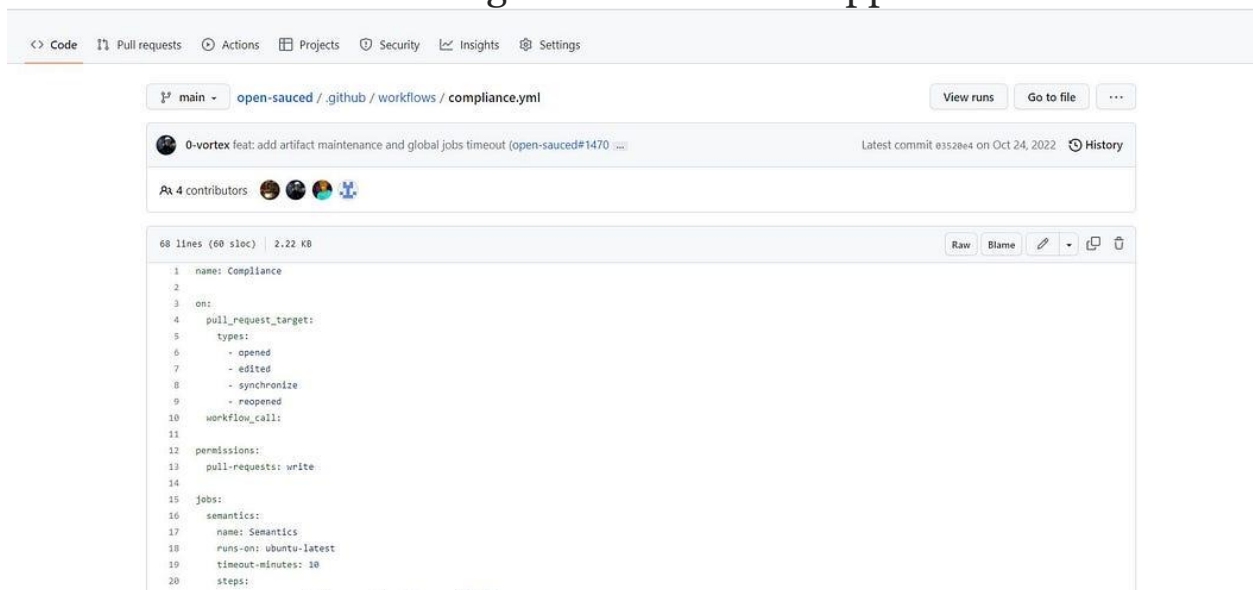
The screenshot shows the GitHub repository page for `djcloudking/open-sauced`, which is a fork of `open-sauced/open-sauced`. The repository is public and has 589 forks and 0 stars. The navigation bar includes links for Code, Pull requests, Actions, Projects, Security, Insights, and Settings. The main content area displays the `.github/workflows/` directory, indicating that the branch is up to date with the upstream repository. A commit by `bdougie` is shown, titled "fix: adds token to make triage work". Below the commit, a list of workflow files is displayed, including `codeql-analysis.yml`, `compliance.yml`, `development.yml`, `maintenance.yml`, `release.yml`, `storybook.yml`, and `triage.yml`, each with a brief description and a timestamp.

File	Description	Timestamp
<code>codeql-analysis.yml</code>	feat: add artifact maintenance and global jobs timeout (open-sauced#1470)	6 months ago
<code>compliance.yml</code>	feat: add artifact maintenance and global jobs timeout (open-sauced#1470)	6 months ago
<code>development.yml</code>	feat: add artifact maintenance and global jobs timeout (open-sauced#1470)	6 months ago
<code>maintenance.yml</code>	ci: make maintenance callable in forks	6 months ago
<code>release.yml</code>	feat: add artifact maintenance and global jobs timeout (open-sauced#1470)	6 months ago
<code>storybook.yml</code>	feat: add artifact maintenance and global jobs timeout (open-sauced#1470)	6 months ago
<code>triage.yml</code>	fix: adds token to make triage work	3 weeks ago

Cheat Sheet: building a CI pipeline can be a simple process. You can use your workflow and move quickly, or it can be more complex when building a big enterprise software.

Third Step: Change your code to trigger your CI/CD pipeline

- For the easy route, you will use **compliance workflow** or `compliance.yml`.
- Open it and read through, you will see this workflow is triggered whenever someone makes their first pull request and sends a message to the owner for approval.



The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links for Code, Pull requests, Actions, Projects, Security, Insights, and Settings. Below this, the repository path is shown as 'open-sauced / github / workflows / compliance.yml'. The file name 'compliance.yml' is highlighted. Below the file name, there's a commit history section showing the latest commit by '0-vortex' with the message 'feat: add artifact maintenance and global jobs timeout (open-sauced#1470)' and a link to 'History'. The main part of the screenshot shows the content of the 'compliance.yml' file, which is a GitHub Actions workflow. The workflow is named 'Compliance' and is triggered on pull request events (opened, edited, synchronize, reopened). It includes a 'permissions' section for 'pull-requests: write' and a 'jobs' section with a job named 'Semantics' that runs on 'ubuntu-latest' and has a timeout of 10 minutes. The workflow is 60 lines long, 60 sloc, and 2.22 KB.

```
1 name: Compliance
2
3 on:
4   pull_request_target:
5     types:
6       - opened
7       - edited
8       - synchronize
9       - reopened
10    workflow_call:
11
12 permissions:
13   pull-requests: write
14
15 jobs:
16   semantics:
17     name: Semantics
18     runs-on: ubuntu-latest
19     timeout-minutes: 10
20     steps:
```

- Now it's time to change the code in compliance workflow to trigger your CI CD pipeline.
- Check `compliance.yml`, focus on the part of the code below:

In order to be considered for merging, the pull request description must refer to a specific issue number. This is described in our [Contributing Guide](https://docs.opensauced.pizza/contributing/introduction-to-contributing/).

This check is looking for a phrase similar to: "Fixes #XYZ" or "Resolves #XYZ" where XYZ is the issue number that this PR is meant to address.

protected-branch-comment: >

Protected Branch

In order to be considered for merging, the pull request changes must not be implemented on the "%branch%" branch. This is described in our [Contributing Guide](https://docs.opensauced.pizza/contributing/introduction-to-contributing/). We would suggest that you close this PR and implement your changes as described in our Contributing Guide and open a new pull request.

- Add **“IMPORTANT”** on top of this part and **start commit**.

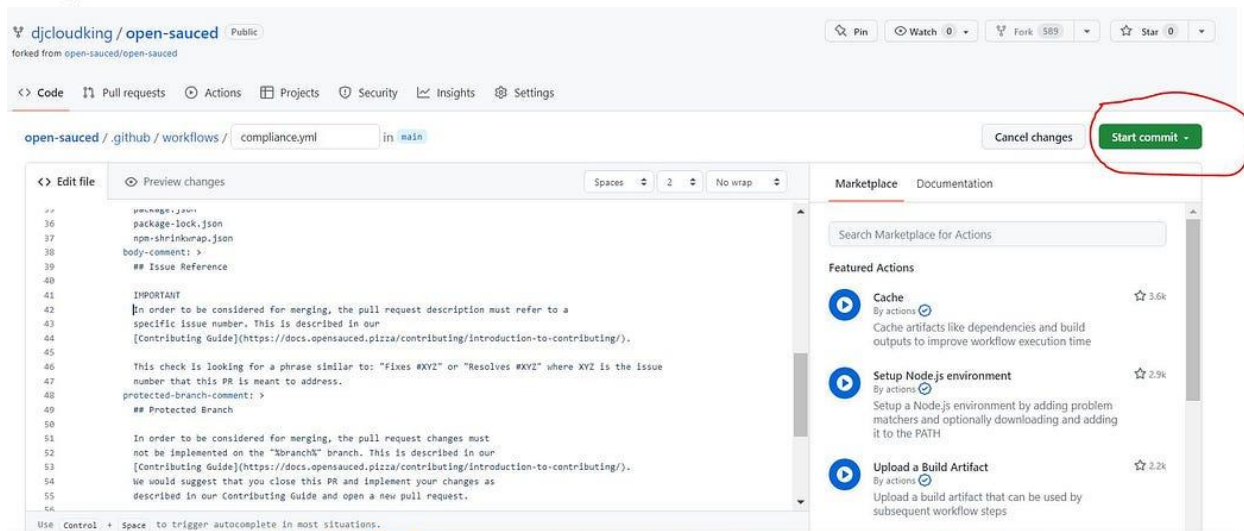
Issue Reference

IMPORTANT

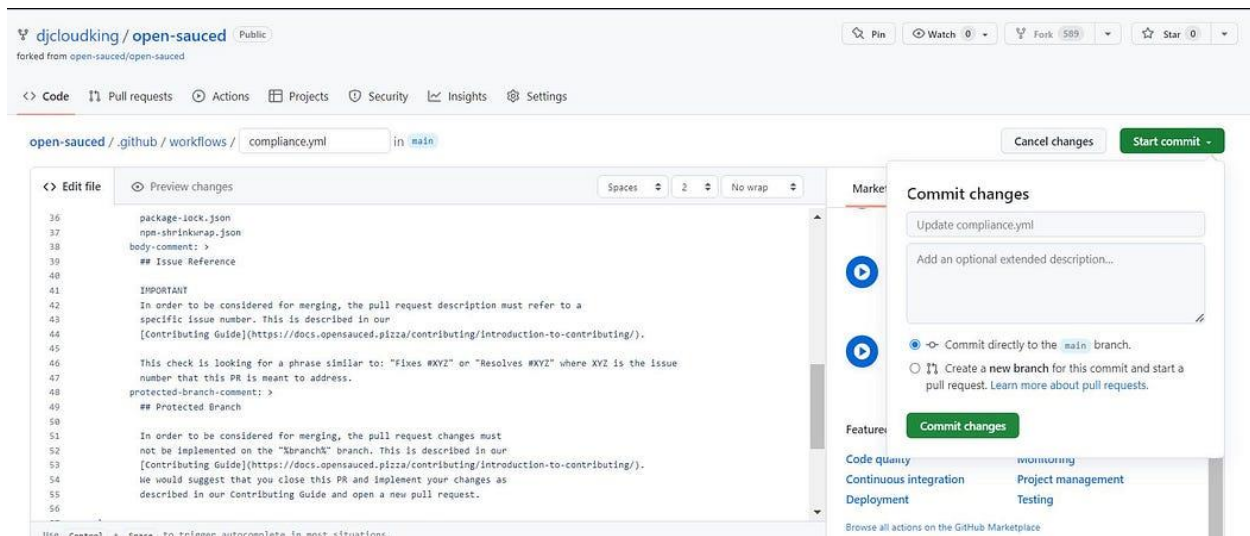
In order to be considered for merging, the pull request description must refer to a specific issue number. This is described in our [Contributing Guide](https://docs.opensauced.pizza/contributing/introduction-to-contributing/).

This check is looking for a phrase similar to: "Fixes #XYZ" or "Resolves #XYZ" where XYZ is the issue number that this PR is meant to address.

protected-branch-comment: >

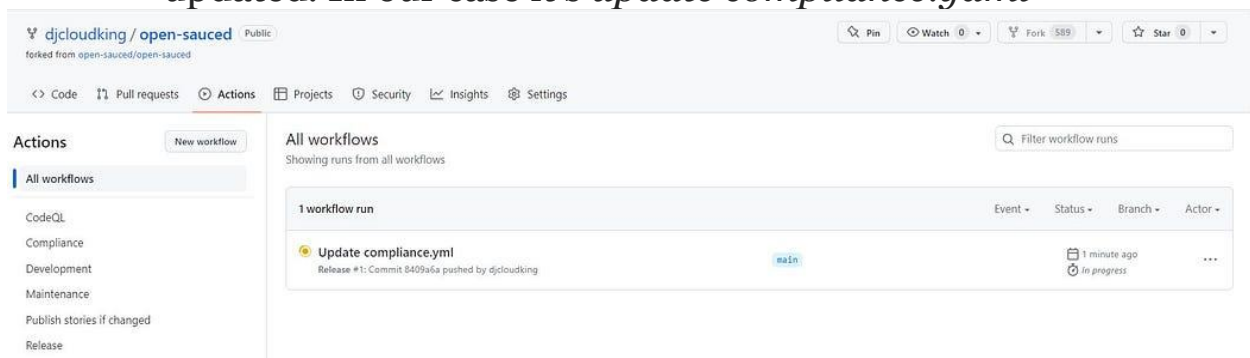


- Choose to attach the change to the main branch or create a new branch, then **commit changes**.



Forth Step: Check your pipeline

- Now look at the workflow visualizer and live logs.
- Go back to **all workflows**, and click on the one you just updated. In our case it's *update compliance.yaml*



Cheat Sheet: In the visualizer, it's important to see which job in the workflow works (green check mark), fails (red sign) and if something is in progress (a yellow sign).

The screenshot shows the GitHub Actions interface for a workflow named 'Update compliance.yml #1'. The workflow is triggered by a push to the 'main' branch by user 'djcloudking'. The status is 'Success' with a total duration of '2m 38s'. The workflow consists of several jobs: 'Test and lint', 'Build container', 'Build application', 'Semantic release', and 'Cleanup actions'. The 'Semantic release' job is currently running, showing a progress bar for 'Deploying to production'. The 'Test and lint' job is also shown with its sub-jobs: 'Test application' (59s), 'Code standards' (50s), 'Build container' (1m 34s), and 'Build application' (57s).

Navigation: <> Code | Pull requests | **Actions** | Projects | Security | Insights | Settings

Release
Update compliance.yml #1 [Re-run all jobs] [More options]

Summary

Jobs

- Test and lint
- Build container
- Build application
- Semantic release
- Cleanup actions

Run details

- Usage
- Workflow file

Triggered via push 3 minutes ago

Triggered by: djcloudking pushed · 8409a6a · main

Status: **Success**

Total duration: 2m 38s

Artifacts: -

release.yml
on: push

Test and li... / Test application 59s

Test and li... / Code standards 50s

Build container 1m 34s

Build application 57s

Semantic release 1m 5s

Cleanup actions 4s

Deploying to production

Cheat Sheet: These logs and the timestamps are very useful if you're debugging a time-sensitive error. A live log makes it possible to immediately address the problem.

Voilà! You have your guide to build a CI/CD pipeline using GitHub Actions.