

# ANASIEZE IKENNA – CLOUD & AI SOLUTIONS ENGINEER

## Project: CI CD Pipeline with GitHub Actions and AWS

### Overview

I implemented a production grade CI/CD pipeline using GitHub Actions and AWS. The pipeline automates build, test, security scanning, and container publishing for a Java application across multiple environments.

### Problem Statement

Engineering teams needed a CI/CD solution that could:

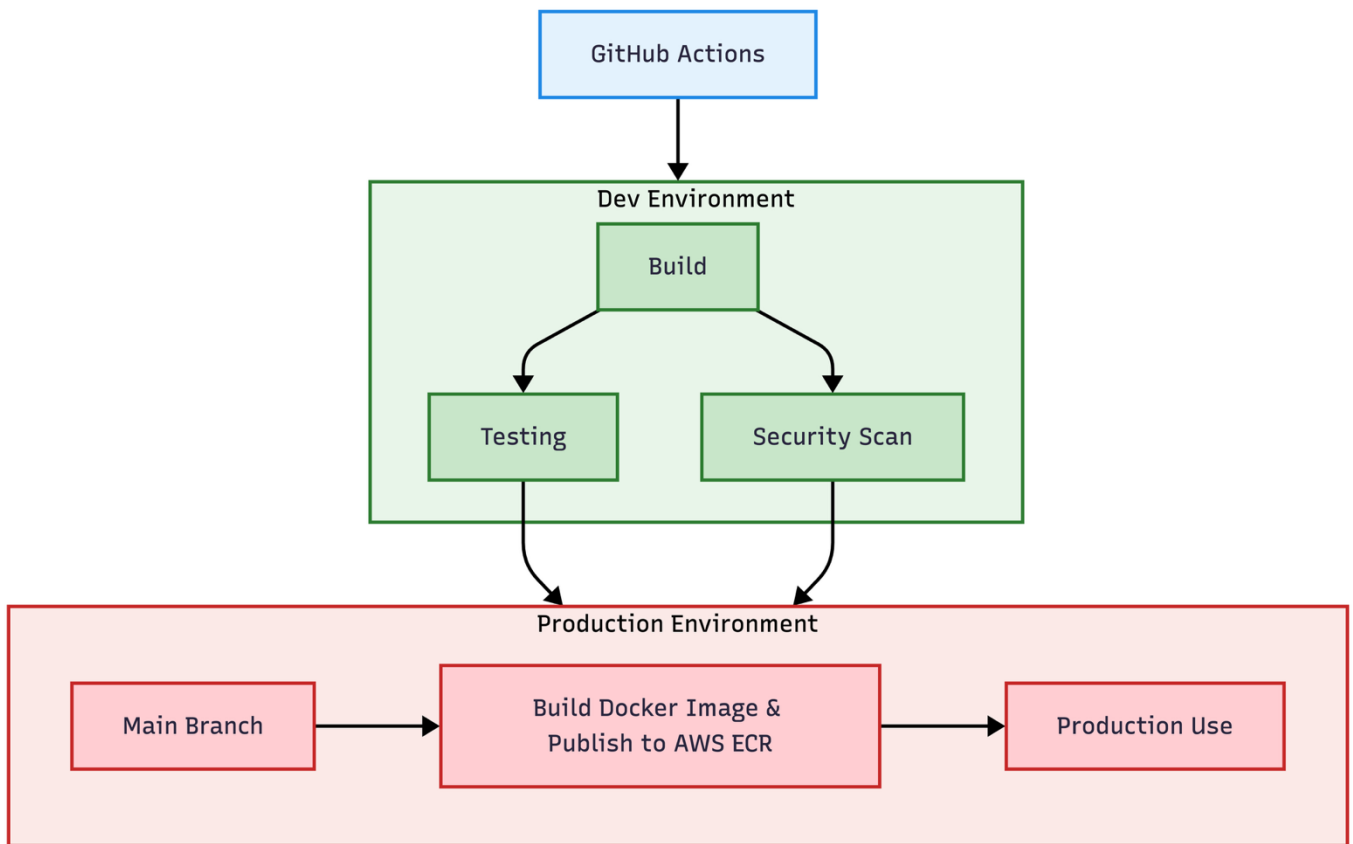
- + Support multiple environments without leaking secrets
- + Prevent accidental production deployments
- + Enforce testing and security checks before release
- + Provide repeatable and auditable deployments
- + Integrate securely with AWS container infrastructure

**Goal:** Responsible for designing and implementing an end to end CI/CD workflow that supports **multiple environments** using GitHub Actions.

### Tech Stack

- + GitHub Actions – CI/CD orchestration
- + Maven – Java build and packaging
- + Docker – Application containerization
- + Trivy – Security and vulnerability scanning
- + Amazon ECR – Container image registry
- + AWS IAM – Environment scoped access control

## Architecture Overview



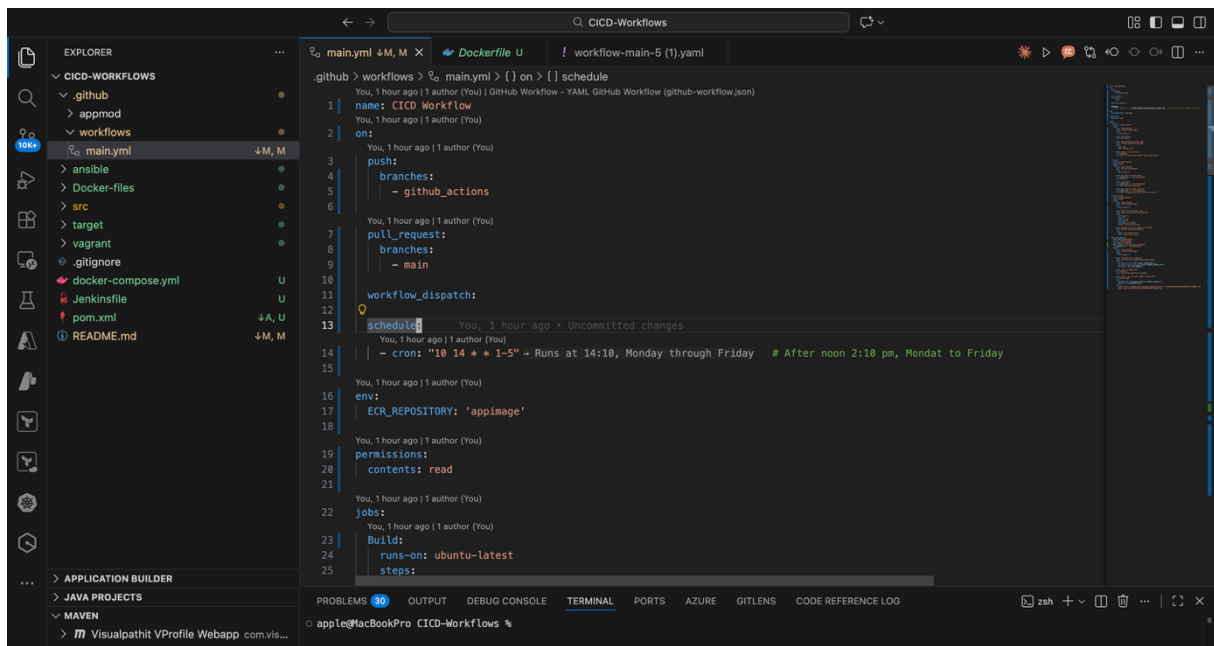
### Flow of Pipeline

- + Code changes are pushed to feature branches or submitted via pull requests
- + CI jobs run automatically to build the application and generate artifacts
- + Security scanning runs in parallel with build validation
- + Tests and static analysis run when changes reach the main branch
- + Production release is gated and triggered only from main
- + Docker images are built and published to Amazon ECR using production scoped credentials

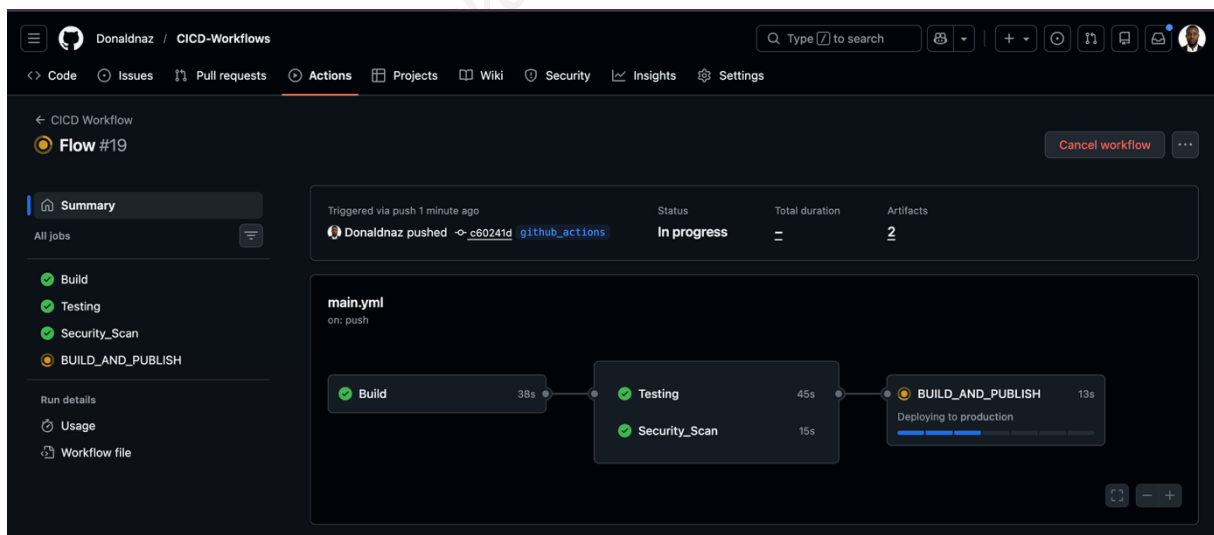
This design ensures production remains protected while allowing fast iteration in lower environments.

## Deployment Steps

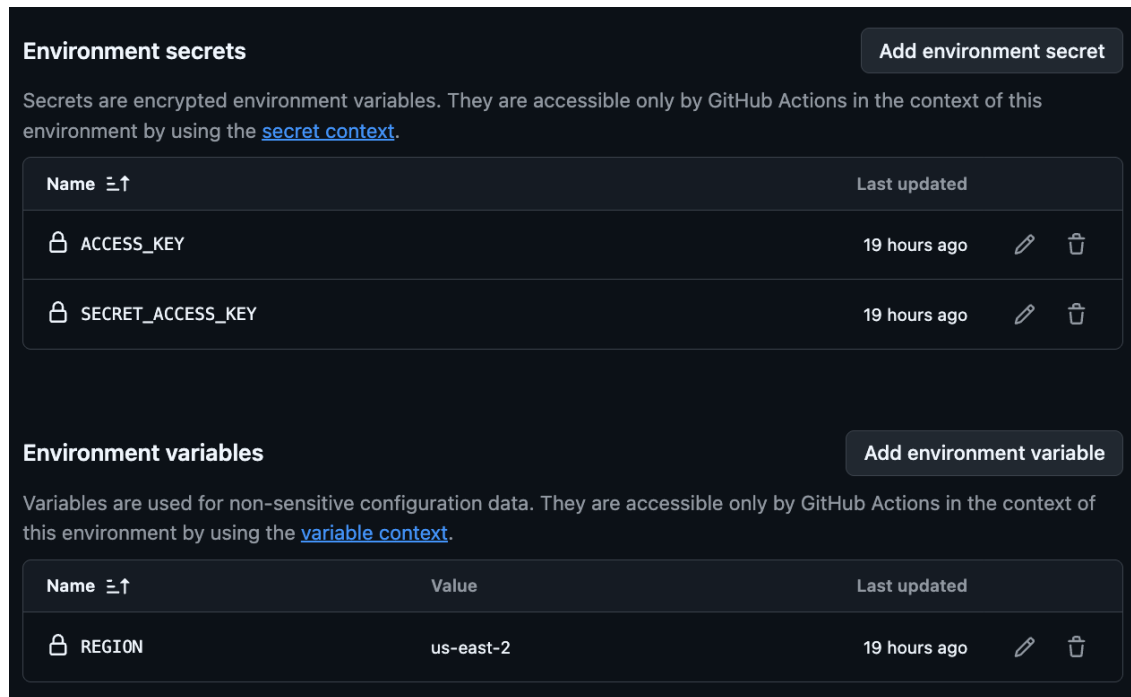
- + Created a GitHub Actions workflow with job dependencies



- + Configured build, test, and security scan stages



## + Protected production deployment using a GitHub environment



The screenshot shows the GitHub 'Environment secrets' and 'Environment variables' configuration page. The 'Environment secrets' section has a table with two entries: 'ACCESS\_KEY' and 'SECRET\_ACCESS\_KEY', both updated 19 hours ago. The 'Environment variables' section has a table with one entry: 'REGION' with the value 'us-east-2', also updated 19 hours ago. Both sections include 'Add environment secret' and 'Add environment variable' buttons respectively.

**Environment secrets**

Secrets are encrypted environment variables. They are accessible only by GitHub Actions in the context of this environment by using the [secret context](#).

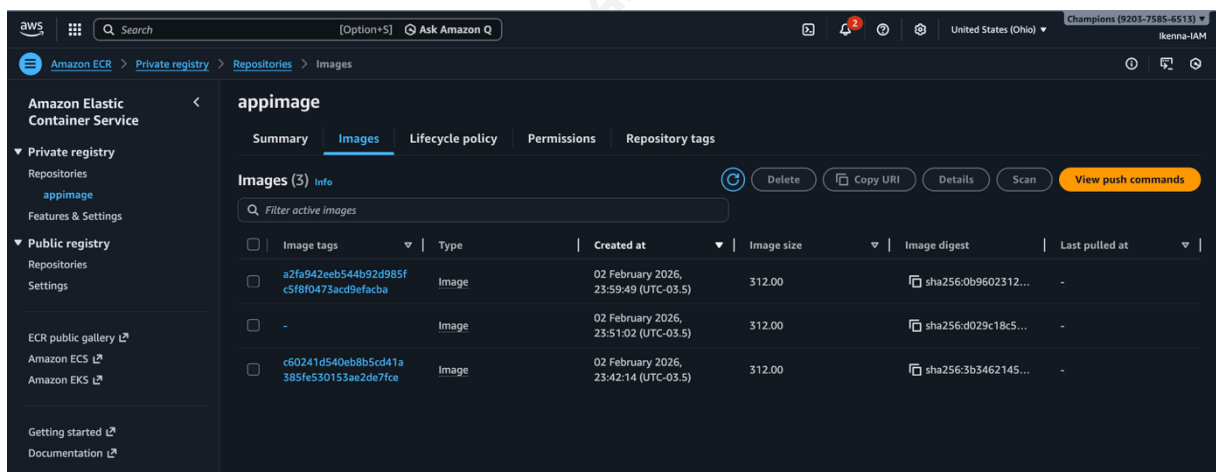
Name	Last updated
ACCESS_KEY	19 hours ago
SECRET_ACCESS_KEY	19 hours ago

**Environment variables**

Variables are used for non-sensitive configuration data. They are accessible only by GitHub Actions in the context of this environment by using the [variable context](#).

Name	Value	Last updated
REGION	us-east-2	19 hours ago

## + Built and published Docker images to Amazon ECR from main



The screenshot shows the Amazon ECR console for the 'appimage' repository. The 'Images' tab is selected, showing a table of three Docker images. The first image has a tag 'a2fa942eeb544b92d985fc5f8f0473acd9efacba' and was created on 02 February 2026. The second image is unnamed and was created at 23:51:02. The third image has a tag 'c60241d540eb8b5cd41a385fe530153ae2de7fce' and was created at 23:42:14. All images are 312.00 MB in size.

Amazon Elastic Container Service

Private registry

Repositories

appimage

Features & Settings

Public registry

Repositories

Settings

ECR public gallery

Amazon ECS

Amazon EKS

Getting started

Documentation

**appimage**

Summary Images Lifecycle policy Permissions Repository tags

Images (3)

Filter active images

Image tags	Type	Created at	Image size	Image digest	Last pulled at
a2fa942eeb544b92d985fc5f8f0473acd9efacba	Image	02 February 2026, 23:59:49 (UTC-03:5)	312.00	sha256:0b9602312...	-
-	Image	02 February 2026, 23:51:02 (UTC-03:5)	312.00	sha256:d029c18c5...	-
c60241d540eb8b5cd41a385fe530153ae2de7fce	Image	02 February 2026, 23:42:14 (UTC-03:5)	312.00	sha256:3b3462145...	-

## Outcome

- A fully operational multi environment CI/CD pipeline
- Safe promotion of code from development to production
- Secure and repeatable Docker image publishing to ECR

## Key Takeaway

CI/CD is not about YAML files. It is about **control, safety, and repeatability at scale**.