# Report: Setting up CI/CD Workflow for a Flutter Project

**Repository Cloning:**

Cloned the repository to initiate the CI/CD setup.

**GitHub Actions Initialization:**

Created an empty main.yml file within the .github/actions/ directory to define the workflow.

**Flutter Build Configuration:**

Faced issues with Flutter version compatibility.

Updated Flutter versions in the workflow to resolve version conflicts.

**SpectralOps Integration:**

Added SpectralOps static analysis to the workflow for enhanced security.

Configured SpectralOps in the YAML file for scan-time options, detector inclusion/exclusion, and output formats.

**Dockerfile Setup:**

Created a Dockerfile for the Flutter project.

Utilized an official Flutter image as the base image.

Copied the project files, installed dependencies, and built the Flutter app in the Dockerfile.

**GitHub Actions Workflow:**

Configured the GitHub Actions workflow to perform the following steps:

Checkout the repository.

Set up Flutter with version 3.16.3.

Run flutter pub get to fetch dependencies.

Set up Node.js.

Executed SpectralOps SAST tool for static analysis.

Built the Docker image using the Dockerfile.

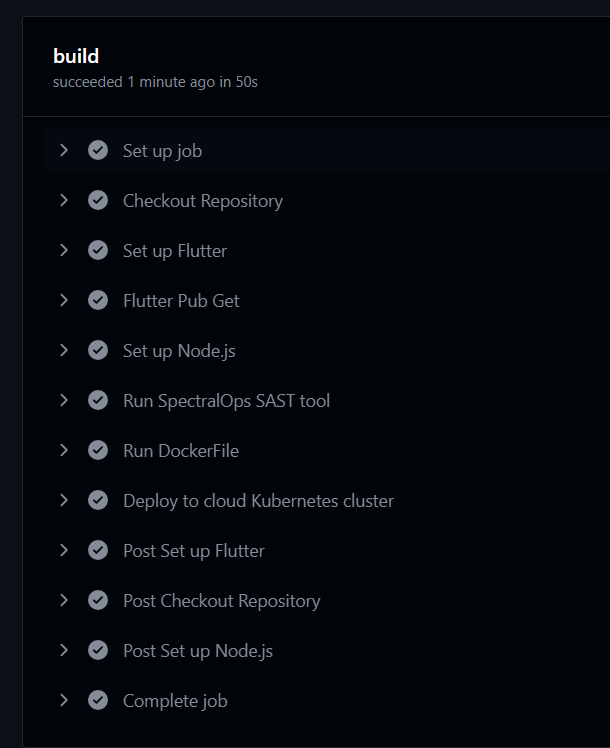
Deployed the Docker image to a cloud Kubernetes cluster.

**Workflow Enhancements:**

Adjusted the Flutter and Node.js versions as needed.

Handled Flutter version compatibility issues.

Ensured that the Dockerfile successfully builds the Flutter app.



**Conclusion:**

The CI/CD workflow is now configured to run static analysis, build the Flutter app, and deploy it to a Kubernetes cluster upon code changes or pushes to the main branch.