Project Documentation: Python Application CI/CD Pipeline with Docker and Security Scanning

Overview

This project implements a CI/CD pipeline for a Python application, automating several key tasks like code linting, unit testing, security scanning, database migration checks, Docker image creation, and vulnerability scanning. The pipeline integrates with GitHub Actions and includes tools like SonarCloud, Snyk, and Trivy for static code analysis and security testing. The application is containerized using Docker and deployed to DockerHub.

Table of Contents

Prerequisites

Before running the CI/CD pipeline, ensure that the following prerequisites are met:

- **Docker**: Docker must be installed and configured on the local machine for building and pushing images.
- **GitHub Secrets**: Ensure that the following GitHub secrets are set up in the repository:
    - `DOCKER_USERNAME`
    - `DOCKER_PASSWORD`
    - `SONAR_TOKEN`
    - `SNYK_TOKEN`
    - `GITHUB_TOKEN`

Project Structure

The project is structured as follows:

```
bash
Copy code
/uni-devops-project
│
├── .editorconfig         # EditorConfig configuration file for consistent
coding styles
├── Dockerfile            # Dockerfile for containerizing the application
├── sonar-project.properties  # SonarCloud configuration file
├── src/
│   ├── app.py            # Python application entry point
│   ├── app_test.py       # Unit tests for the application
│   └── requirements.txt  # List of Python dependencies
└── .github/
    └── workflows/
        ├── ci-cd.yml     # Main CI/CD pipeline workflow
        └── pull-request.yml  # Pull Request workflow for SonarCloud scan
```

CI/CD Pipeline

1. EditorConfig Checker

The pipeline includes an **EditorConfig Checker** to ensure that the coding style across the project is consistent with the rules defined in the `.editorconfig` file.

- **Action**: `editorconfig-checker/action-editorconfig-checker`
- **Run Command**: `editorconfig-checker`

2. Markdown Lint

Markdown files are checked for proper formatting using **markdownlint-cli**.

- **Action**: `markdown-cli`
- **Run Command**: `npx markdown-cli '**/*.md'`

3. Flake8 Lint

The project enforces Python code style using **Flake8**.

- **Action**: `suo/flake8-github-action`
- **Run Command**: `flake8`
- **Dependencies**: `pip install flake8`

4. Unit Testing

Unit tests are run using the built-in `unittest` framework.

- **Run Command**: `python3 -m unittest src/app_test.py`
- **Dependencies**: `pip install -r src/requirements.txt`

5. Secrets Detection

Gitleaks scans the code for hardcoded secrets to ensure that sensitive information is not exposed in the codebase.

- **Action**: `gitleaks/gitleaks-action`
- **Run Command**: `gitleaks scan`

6. Database Migration and Tests

Checks for database migrations and tests them using **Flyway** and a PostgreSQL service.

- **Service**: PostgreSQL
- **Action**: `joshuaavalon/flyway-action`
- **Run Command**: `flyway migrate`

7. SonarCloud Analysis

SonarCloud is used to analyze the code for quality, security vulnerabilities, and code smells.

- **Action**: `sonarsource/sonarcloud-github-action`
- **Run Command**: `sonarcloud scan`
- **Dependencies**: `SONAR_TOKEN`

8. Snyk Security Test

Snyk is used for identifying and fixing security vulnerabilities in the dependencies listed in `requirements.txt`.

- **Action**: `snyk test`
- **Run Command**: `snyk test --file=src/requirements.txt --project-name=uni-devops-project`
- **Dependencies**: `npm install -g snyk`
- **Snyk Auth**: `snyk auth ${{ secrets.SNYK_TOKEN }}`

9. Build and Push Docker Image

The pipeline builds a Docker image for the Python application and pushes it to DockerHub.

- **Action**: `docker/setup-buildx-action`, `docker/login-action`
- **Run Command**:
  - `docker build -t gkosteva/uni-devops-project:latest .`
  - `docker push gkosteva/uni-devops-project:latest`

Dockerfile Setup

The Dockerfile used for containerizing the Python application is as follows:

```
Dockerfile
Copy code
# Use a minimal base image to reduce image size
FROM python:3.10-alpine

# Set the working directory in the container
WORKDIR /app

# Copy the requirements file and install dependencies
COPY src/requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# Copy the application files
COPY src/ .

# Run the Python application
CMD ["python3", "src/app.py"]
```

Optimizations:

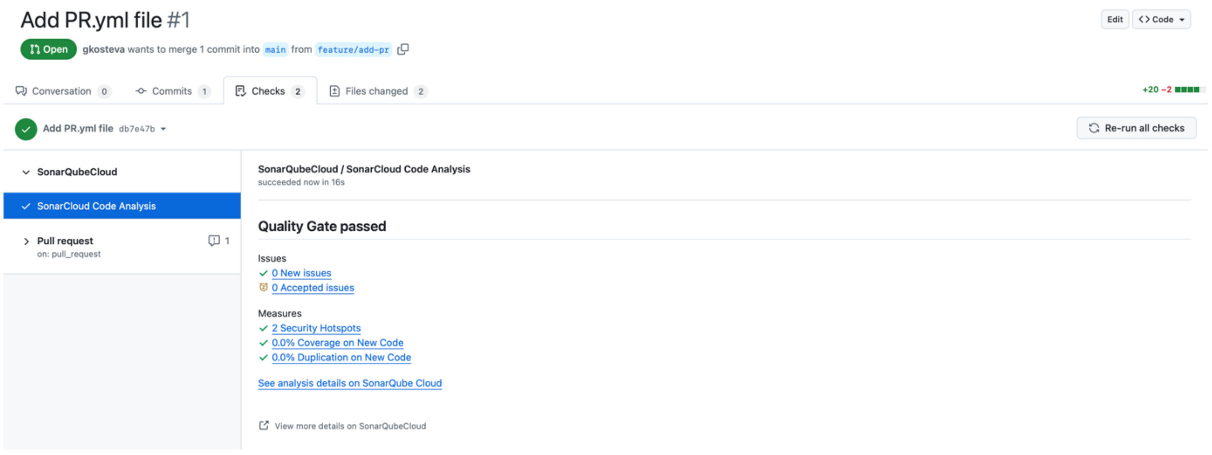
- **Base Image**: The base image is `python:3.10-alpine` for a smaller, security-focused image.

Contributing

1. Fork this repository.
2. Clone your fork to your local machine.
3. Make your changes and commit them.
4. Push your changes to your fork.
5. Open a pull request for review.

Please ensure that your code follows the style guide and passes all tests before submitting a pull request.

Additional images



Add PR.yml file #1

Open gkosteva wants to merge 1 commit into main from feature/add-pr

Add PR.yml file db7e47b

SonarQubeCloud / SonarCloud Code Analysis
succeeded now in 16s

Quality Gate passed

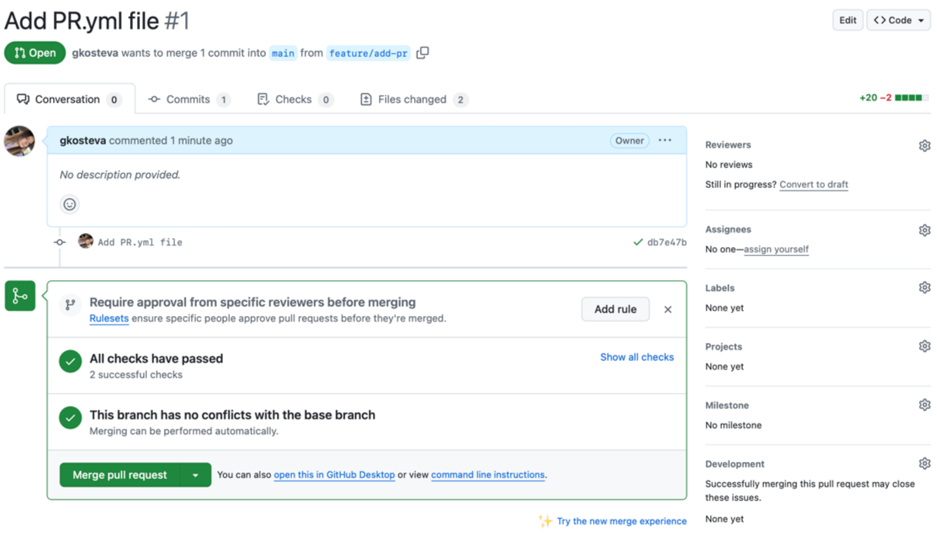
Issues
✓ 0 New issues
0 Accepted issues

Measures
✓ 2 Security Hotspots
✓ 0.0% Coverage on New Code
✓ 0.0% Duplication on New Code

See analysis details on SonarQube Cloud

View more details on SonarQubeCloud

Sonar code analysis output on a PR



Add PR.yml file #1

Open gkosteva wants to merge 1 commit into main from feature/add-pr

gkosteva commented 1 minute ago

No description provided.

Add PR.yml file db7e47b

Require approval from specific reviewers before merging
Rulesets ensure specific people approve pull requests before they're merged.

All checks have passed
2 successful checks

This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one—assign yourself

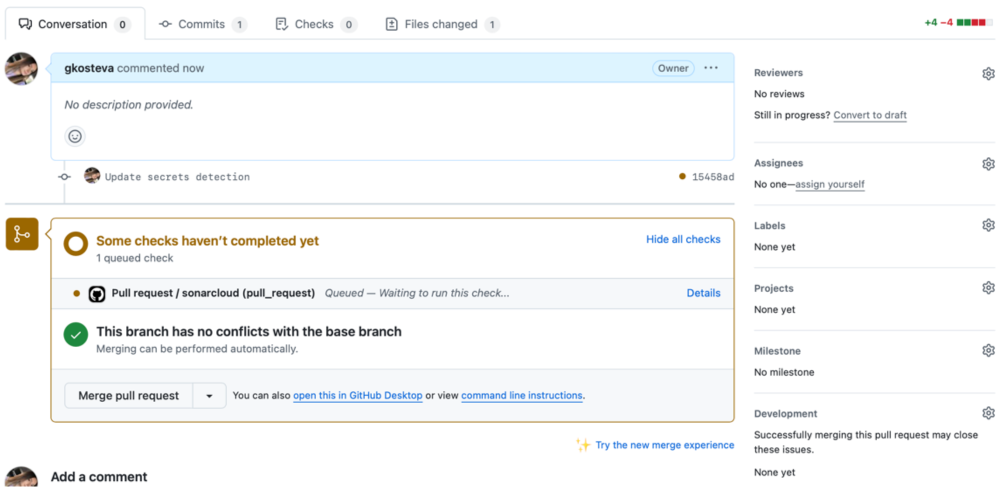
Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.
None yet

Passed pipeline on a PR ready to be merged


Conversation 0 · Commits 1 · Checks 0 · Files changed 1 · +4 −4

gkosteva commented now · Owner

No description provided.

Update secrets detection · 15458ad

Some checks haven't completed yet
1 queued check · Hide all checks

Pull request / sonarcloud (pull_request) *Queued — Waiting to run this check...* Details

This branch has no conflicts with the base branch
Merging can be performed automatically.

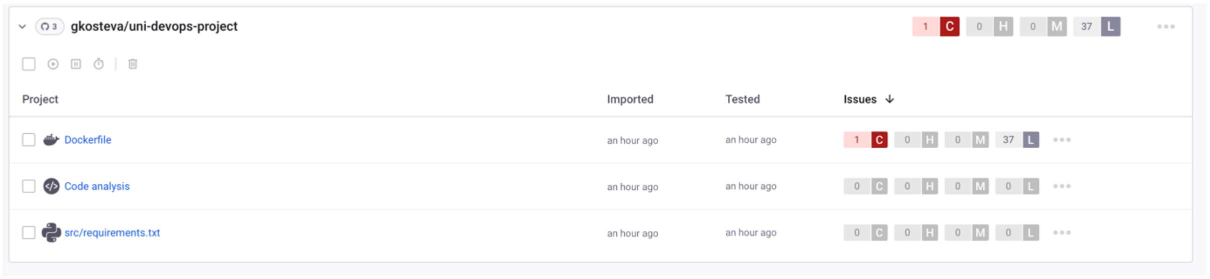
Merge pull request · You can also open this in GitHub Desktop or view command line instructions.

Try the new merge experience

Add a comment

Reviewers: No reviews. Still in progress? Convert to draft
Assignees: No one—assign yourself
Labels: None yet
Projects: None yet
Milestone: No milestone
Development: Successfully merging this pull request may close these issues. None yet

Waiting to pass all checks


gkosteva/uni-devops-project · 1 C · 0 H · 0 M · 37 L

Project	Imported	Tested	Issues
Dockerfile	an hour ago	an hour ago	1 C · 0 H · 0 M · 37 L
Code analysis	an hour ago	an hour ago	0 C · 0 H · 0 M · 0 L
src/requirements.txt	an hour ago	an hour ago	0 C · 0 H · 0 M · 0 L

Snyk gkosteva/uni-devops-project

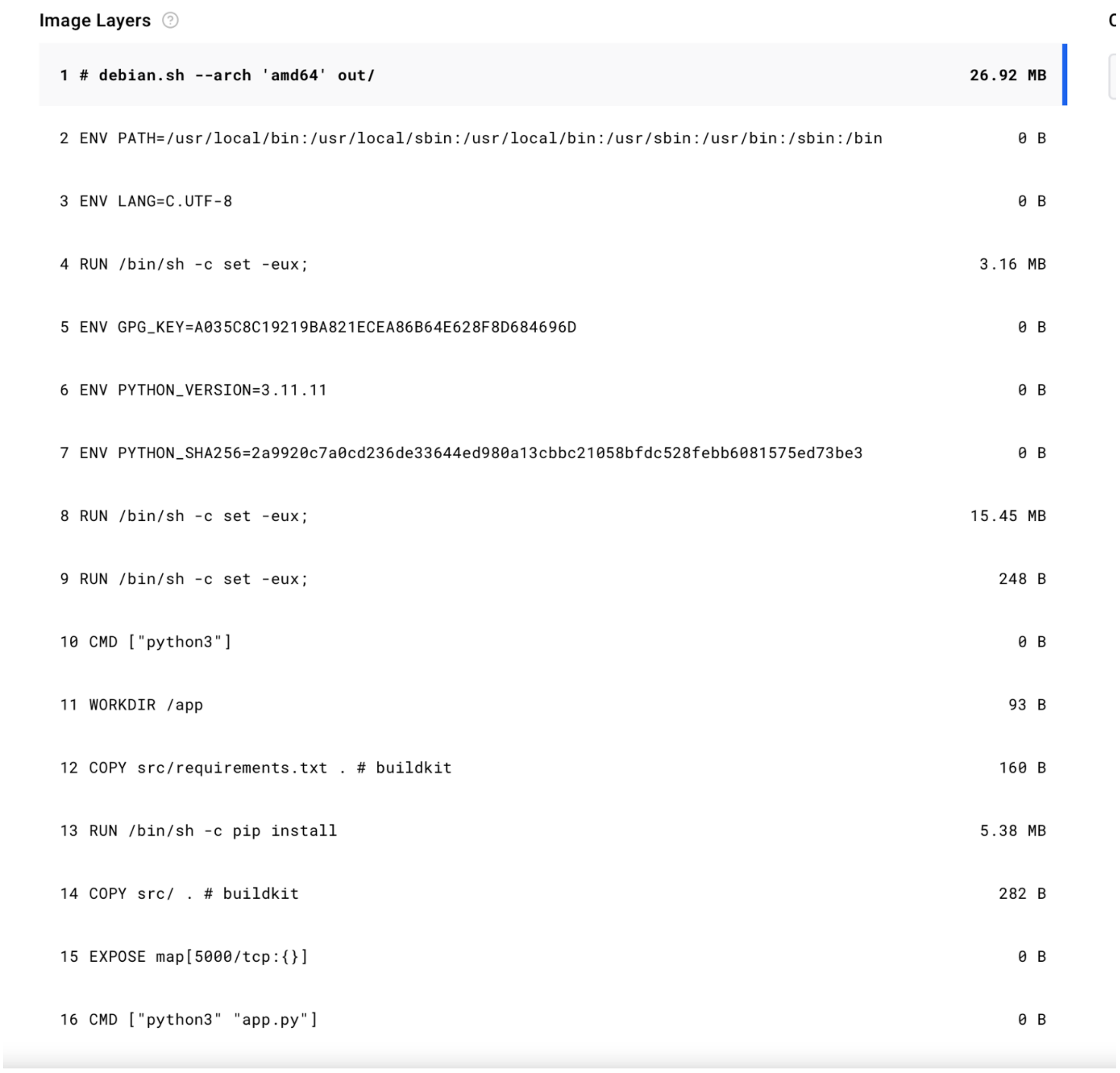
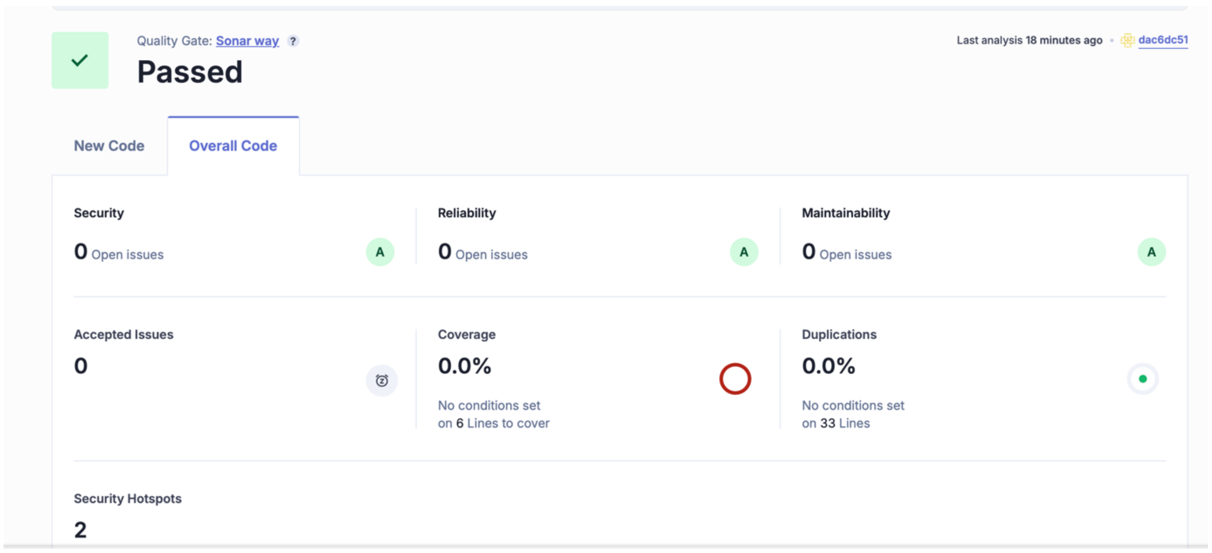


Image Layers

```
1  # debian.sh --arch 'amd64' out/                                                          26.92 MB
2  ENV PATH=/usr/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin          0 B
3  ENV LANG=C.UTF-8                                                                              0 B
4  RUN /bin/sh -c set -eux;                                                                   3.16 MB
5  ENV GPG_KEY=A035C8C19219BA821ECEA86B64E628F8D684696D                                          0 B
6  ENV PYTHON_VERSION=3.11.11                                                                    0 B
7  ENV PYTHON_SHA256=2a9920c7a0cd236de33644ed980a13cbbc21058bfdc528febb6081575ed73be3            0 B
8  RUN /bin/sh -c set -eux;                                                                  15.45 MB
9  RUN /bin/sh -c set -eux;                                                                    248 B
10 CMD ["python3"]                                                                               0 B
11 WORKDIR /app                                                                                 93 B
12 COPY src/requirements.txt . # buildkit                                                      160 B
13 RUN /bin/sh -c pip install                                                                 5.38 MB
14 COPY src/ . # buildkit                                                                      282 B
15 EXPOSE map[5000/tcp:{}]                                                                       0 B
16 CMD ["python3" "app.py"]                                                                      0 B
```

Docker latest image layers on project uni-devops-project



Quality Gate: Sonar way

Last analysis 18 minutes ago · dac6dc51

Passed

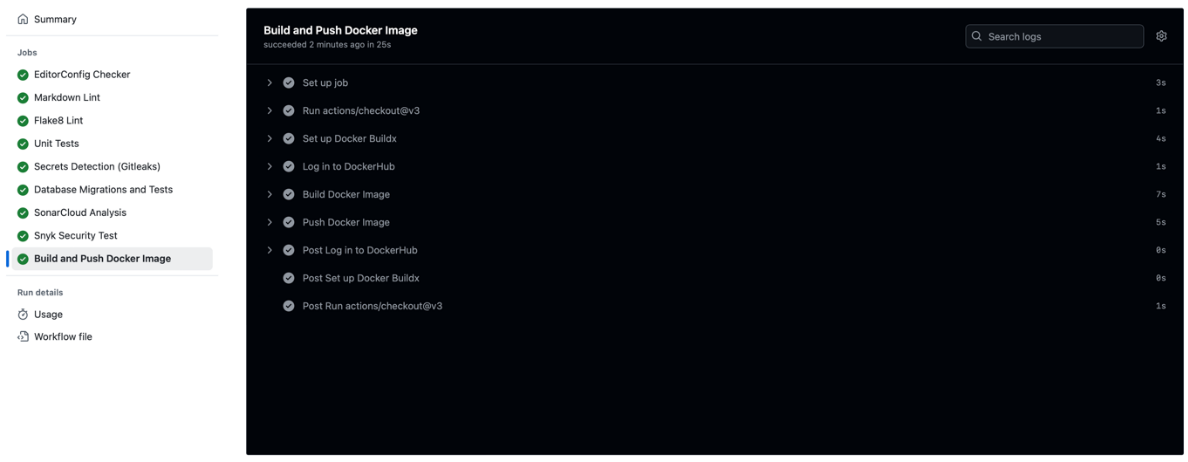
New Code | Overall Code

Security	Reliability	Maintainability
0 Open issues (A)	0 Open issues (A)	0 Open issues (A)

Accepted Issues	Coverage	Duplications
0	0.0% No conditions set on 6 Lines to cover	0.0% No conditions set on 33 Lines

Security Hotspots
2

Sonar code analysis on ***main*** *branch*



- ✓ EditorConfig Checker
- ✓ Markdown Lint
- ✓ Flake8 Lint
- ✓ Unit Tests
- ✓ Secrets Detection (Gitleaks)
- ✓ Database Migrations and Tests
- ✓ SonarCloud Analysis
- ✓ Snyk Security Test
- ✓ Build and Push Docker Image

Build and Push Docker Image
succeeded 2 minutes ago in 25s

```
✓ Set up job                      3s
✓ Run actions/checkout@v3         1s
✓ Set up Docker Buildx            4s
✓ Log in to DockerHub             1s
✓ Build Docker Image              7s
✓ Push Docker Image               5s
✓ Post Log in to DockerHub        0s
✓ Post Set up Docker Buildx       0s
✓ Post Run actions/checkout@v3    1s
```

*Fully passed pipeline on **main** branch*

License

This project is licensed under the MIT License.

This document provides a concise overview of the CI/CD pipeline, Docker containerization, and security scanning processes implemented in the project. It should guide both contributors and users in understanding and using the system.

GitHub repo -> https://github.com/gkosteva/uni-devops-project