Project Report: CI/CD Pipeline with GitHub Actions

Renzhong Chen

Project Overview

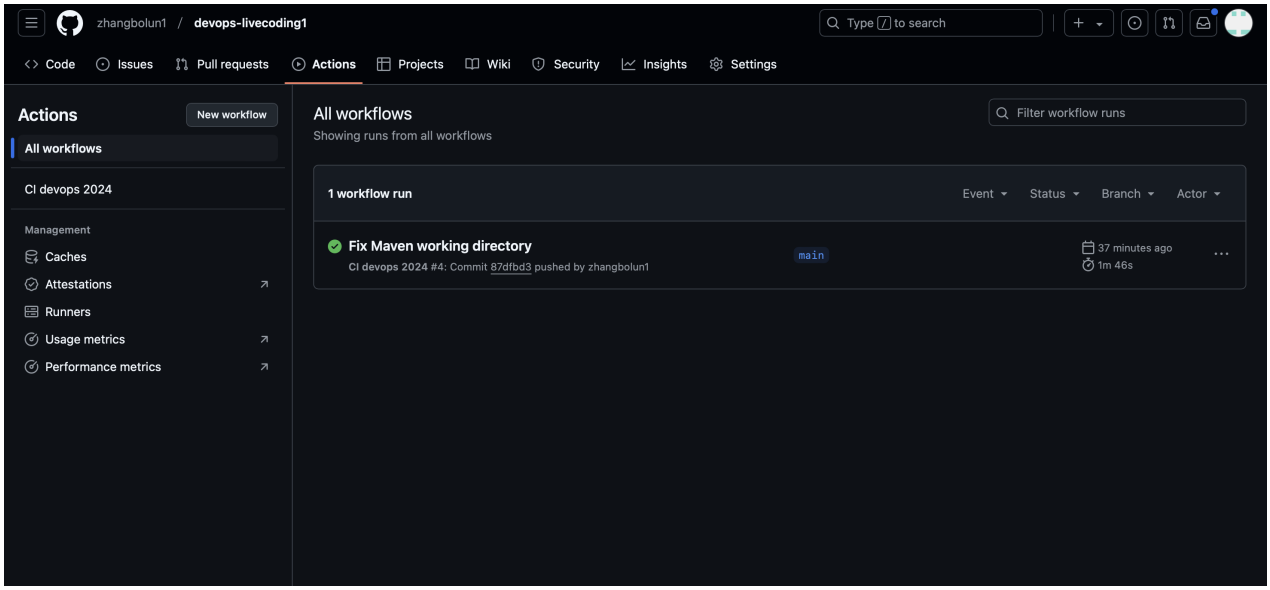
This project aims to implement a **CI/CD pipeline** for a Java-based application using **GitHub Actions**, **Maven**, **Docker**, and **SonarCloud**. The pipeline automates the process of building, testing, analyzing, and deploying the application.

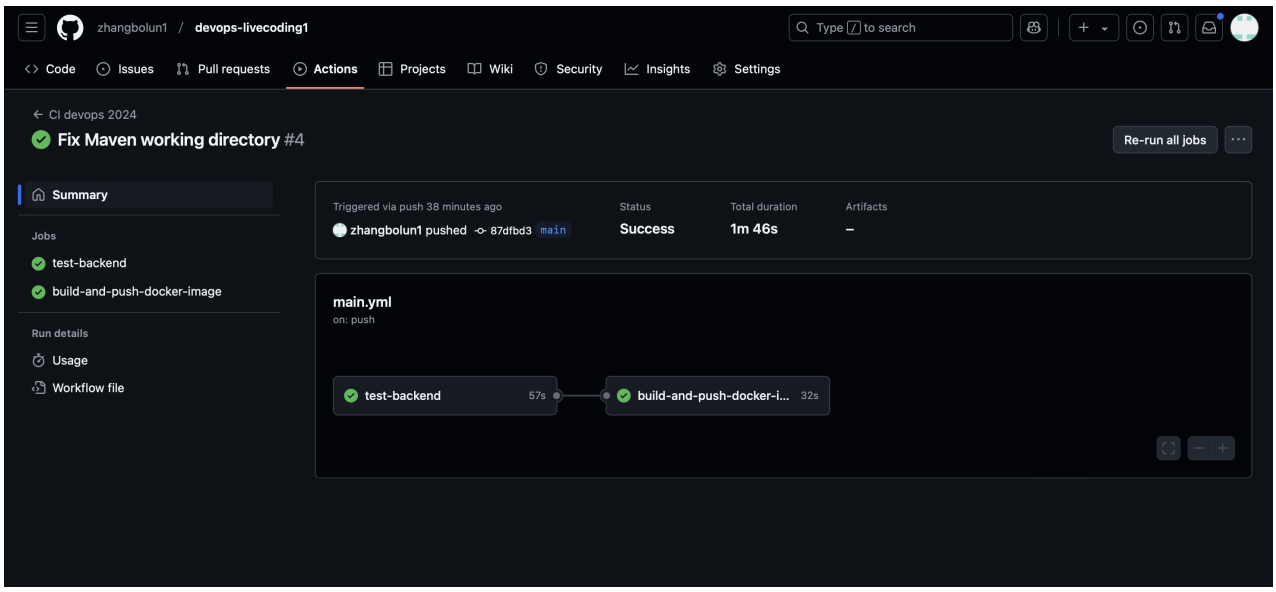
The URL of these sites:

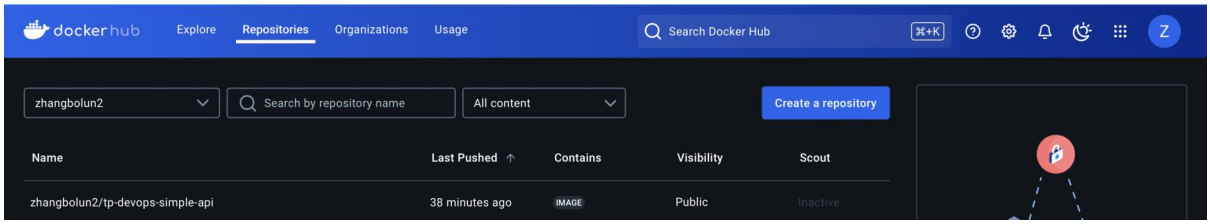
Dockerhub: https://hub.docker.com/repository/docker/zhangbolun2/tp-devops-simple-api/general

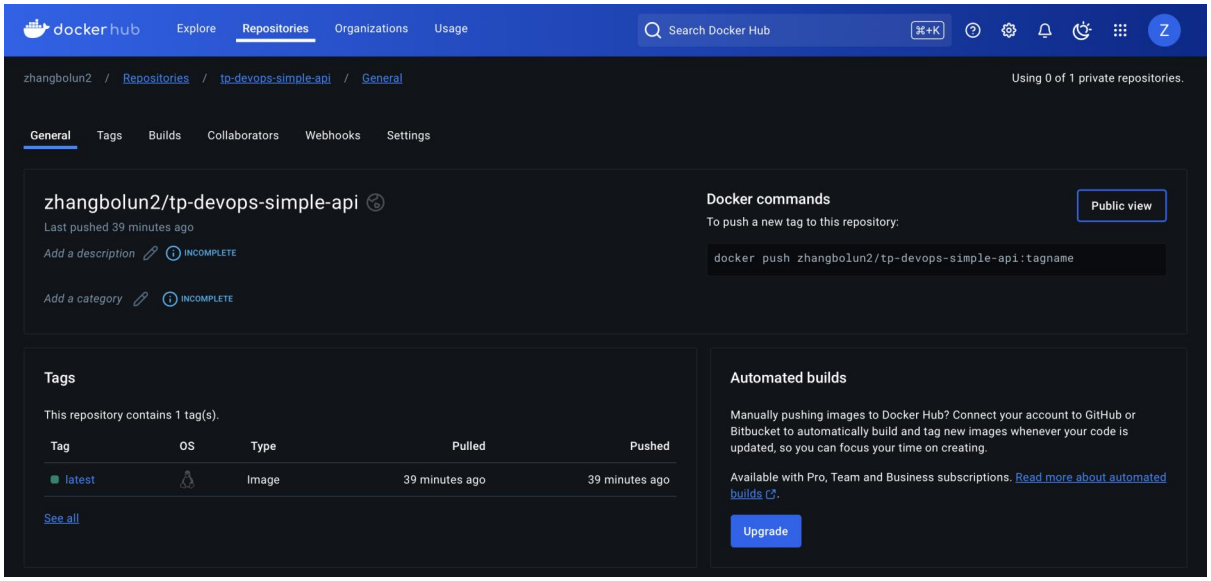
Sonarqube: https://sonarcloud.io/organizations/zhangbolun1/projects

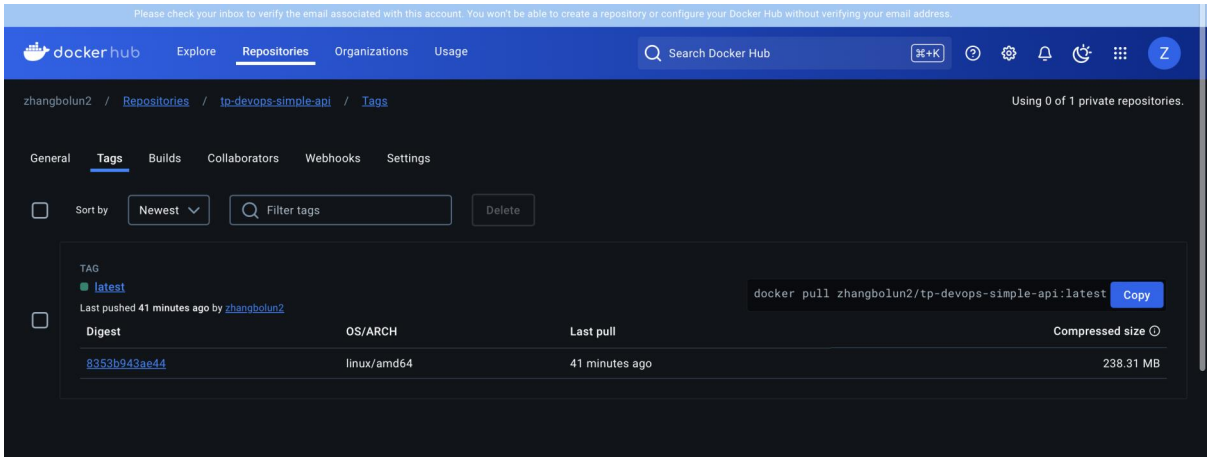
The screenshots of final results:

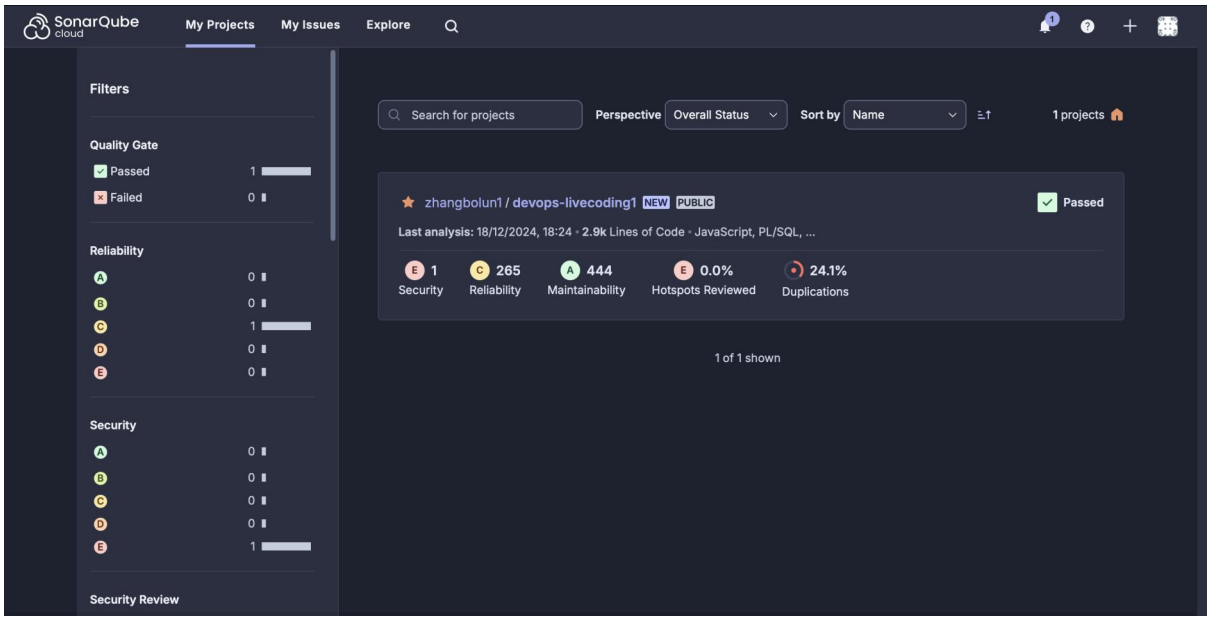

zhangbolun1 / devops-livecoding1
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Actions
New workflow
All workflows
CI devops 2024
Management
Caches
Attestations
Runners
Usage metrics
Performance metrics
All workflows
Showing runs from all workflows
Filter workflow runs
1 workflow run
Event
Status
Branch
Actor
Fix Maven working directory
CI devops 2024 #4: Commit 87dfbd3 pushed by zhangbolun1
main
37 minutes ago
1m 46s

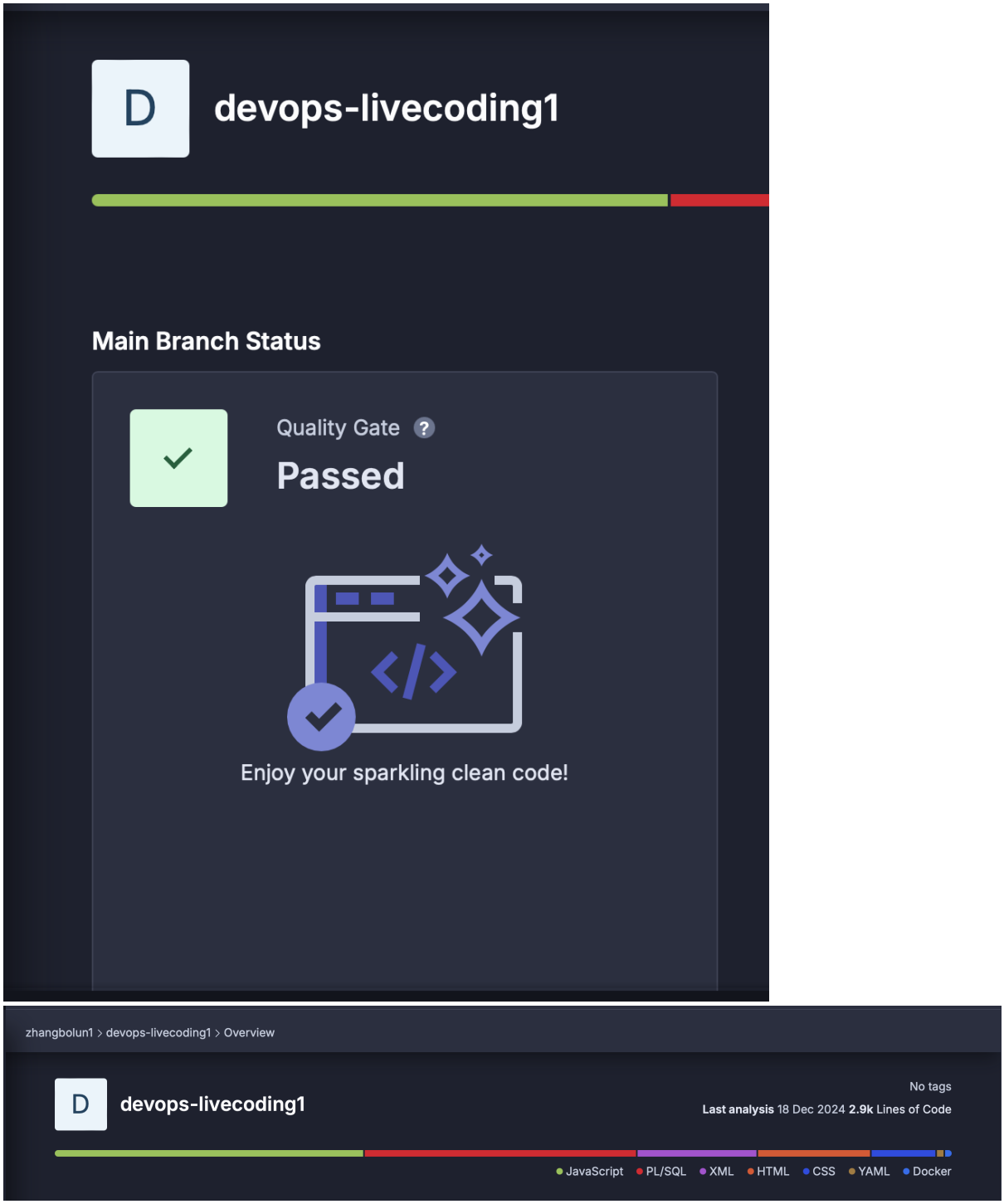

zhangbolun1 / devops-livecoding1
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
CI devops 2024
Fix Maven working directory #4
Re-run all jobs
Summary
Jobs
test-backend
build-and-push-docker-image
Run details
Usage
Workflow file
Triggered via push 38 minutes ago
zhangbolun1 pushed 87dfbd3 main
Status
Success
Total duration
1m 46s
Artifacts
–
main.yml
on: push
test-backend 57s
build-and-push-docker-i... 32s


docker hub
Explore
Repositories
Organizations
Usage
Search Docker Hub
⌘+K
zhangbolun2
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
zhangbolun2/tp-devops-simple-api
38 minutes ago
IMAGE
Public
Inactive


docker hub
Explore
Repositories
Organizations
Usage
Search Docker Hub
⌘+K
zhangbolun2 / Repositories / tp-devops-simple-api / General
Using 0 of 1 private repositories.
General
Tags
Builds
Collaborators
Webhooks
Settings
zhangbolun2/tp-devops-simple-api
Last pushed 39 minutes ago
Add a description
INCOMPLETE
Add a category
INCOMPLETE
Docker commands
To push a new tag to this repository:
Public view
docker push zhangbolun2/tp-devops-simple-api:tagname
Tags
This repository contains 1 tag(s).
Tag
OS
Type
Pulled
Pushed
latest
Image
39 minutes ago
39 minutes ago
See all
Automated builds
Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds.
Upgrade


Please check your inbox to verify the email associated with this account. You won't be able to create a repository or configure your Docker Hub without verifying your email address.
docker hub
Explore
Repositories
Organizations
Usage
Search Docker Hub
⌘+K
zhangbolun2 / Repositories / tp-devops-simple-api / Tags
Using 0 of 1 private repositories.
General
Tags
Builds
Collaborators
Webhooks
Settings
Sort by
Newest
Filter tags
Delete
TAG
latest
Last pushed 41 minutes ago by zhangbolun2
docker pull zhangbolun2/tp-devops-simple-api:latest
Copy
Digest
OS/ARCH
Last pull
Compressed size
8353b943ae44
linux/amd64
41 minutes ago
238.31 MB


SonarQube cloud
My Projects
My Issues
Explore
Filters
Quality Gate
Passed 1
Failed 0
Reliability
A 0
B 0
C 1
D 0
E 0
Security
A 0
B 0
C 0
D 0
E 1
Security Review
Search for projects
Perspective
Overall Status
Sort by
Name
1 projects
zhangbolun1 / devops-livecoding1
NEW
PUBLIC
Passed
Last analysis: 18/12/2024, 18:24 • 2.9k Lines of Code • JavaScript, PL/SQL, ...
E 1 Security
C 265 Reliability
A 444 Maintainability
E 0.0% Hotspots Reviewed
24.1% Duplications
1 of 1 shown



Goals Achieved

1. **Continuous Integration (CI):**

- Automatically build and test the application on every commit to main and develop branches.

- Tools used: **GitHub Actions** and **Maven**.

2. **Continuous Delivery (CD):**

- Build a Docker image for the application.
- Push the Docker image to **Docker Hub** only when the code is pushed to the main branch.

3. **Code Quality Analysis:**

- Integrated **SonarCloud** to analyze code quality, including:
- Bugs and security vulnerabilities.
- Code smells.
- Code duplication percentage.

CI/CD Pipeline Workflow

The CI/CD pipeline consists of two main jobs:

1. **Backend Testing (CI):**

- Checkout code using actions/checkout.
- Set up Java 17 using actions/setup-java.
- Build and test the application using Maven:

```
mvn clean verify
```

2. **Docker Build and Push (CD):**

- Build a Docker image using docker/build-push-action.
- Push the image to Docker Hub:
- Tags used: latest for the main branch.

3. **SonarCloud Analysis:**

- Integrated SonarCloud analysis using Maven's Sonar plugin:

```
mvn -B verify sonar:sonar \
  -Dsonar.projectKey=zhangbolun1 \
  -Dsonar.organization=zhangbolun1 \
  -Dsonar.host.url=https://sonarcloud.io \
  -Dsonar.login=${{ secrets.SONAR_TOKEN }}
```

Key Results

1. **GitHub Actions**:
   - Successfully automated building, testing, and deploying the application.
   - CI pipeline status: **Passed**.
2. **SonarCloud Analysis**:
   - **Security issues**: 1
   - **Reliability issues**: 265
   - **Maintainability**: A (444 points)
   - **Duplications**: 24.1%
3. **Docker Image**:
   - Image successfully pushed to Docker Hub:

```
zhangbolun2/tp-devops-simple-api:latest
```

Tools Used

- **GitHub Actions**: CI/CD workflow automation.
- **Maven**: Build and test Java applications.
- **SonarCloud**: Code quality and security analysis.

- **Docker**: Containerization and delivery.
- **Docker Hub**: Image storage and distribution.

Conclusion

The project successfully implements a **CI/CD pipeline** that:

1. Ensures code reliability through automated tests.
2. Improves maintainability and code quality via SonarCloud.
3. Streamlines deployment with Docker image delivery.

This pipeline can now be extended for further features such as **deployment to production environments** or **splitting pipelines into multiple workflows**.

Future Improvements

1. Optimize code to reduce duplication (24.1%).
2. Address identified security and reliability issues.
3. Implement multi-stage Docker builds for smaller image sizes.

This report summarizes the project's achievements and highlights areas for improvement. The implemented pipeline demonstrates modern DevOps practices using widely adopted tools.