# TP2 MMMDE4IN21 DevOps

## Rastislav Kmeť

2-1 What are testcontainers?

Java library that provides lightweight instances of common databases, anything that can run in a docker container.

Setting up the Github actions consists of adding DOCKER\_USERNAME and DOCKER\_PASSWORD secrets in your GitHub repository settings under Secrets and variables > Actions

ci.yml file configuration:

name: CI/CD Pipeline

#

on:

  push:

    branches:

      - main

  pull\_request:

    branches:

      - main

jobs:

  build:

    runs-on: ubuntu-latest

    steps:

    - name: Checkout code

      uses: actions/checkout@v2

    - name: Set up JDK 17

      uses: actions/setup-java@v4

      with:

        distribution: 'temurin'

        java-version: '17'

    - name: Build and test with Maven

      run: mvn -B verify sonar:sonar -Dsonar.projectKey=rastokme\_tp1 -Dsonar.organization=rastokme -Dsonar.host.url=https://sonarcloud.io -Dsonar.login=${{ secrets.SONAR\_TOKEN }}  --file ./simple-api/pom.xml

  build-and-push-docker-image:

    needs: build

    runs-on: ubuntu-22.04

    steps:

    - name: Checkout code

      uses: actions/checkout@v2

    - name: Log in to Docker Hub

      uses: docker/login-action@v2

      with:

        username: ${{ secrets.DOCKER\_USERNAME }}

        password: ${{ secrets.DOCKER\_PASSWORD }}

    - name: Build and push simple-api image

      uses: docker/build-push-action@v3

      with:

        context: ./simple-api

        push: true

        tags: ${{ secrets.DOCKER\_USERNAME }}/tp-devops-simple-api:latest

    - name: Build and push httpd image

      uses: docker/build-push-action@v3

      with:

        context: ./httpd

        push: true

        tags: ${{ secrets.DOCKER\_USERNAME }}/tp-devops-httpd:latest

    - name: Build and push database image

      uses: docker/build-push-action@v3

      with:

        context: ./database

        push: true

        tags: ${{ secrets.DOCKER\_USERNAME }}/tp-devops-database:latest

This is our address:

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

After disconnecting from the school wifi which is problematic.

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

CI/CD Pipeline Configuration

This image shows the CI/CD pipeline configuration in GitHub Actions.Obrázok, na ktorom je text, snímka obrazovky, jedálny lístok

Automaticky generovaný popis

API Endpoints Verification

This image shows the API endpoint verification using curl command.Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Job is complete

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis

Result of the process

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis

Secured Variables, why?

Information like passwords, API keys, and tokens are not easily accessible in the project. This is crucial for maintaining security, especially in public repositories.

Quality Gate Results in SonarCloud

This image shows the quality gate results and analysis report in SonarCloud.

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis

For the sonarcloud setup it was necessary to update GitHub Actions for SonarCloud

Update the ci.yml file to include it. This is the element of the code:

- name: Build and test with Maven

      run: mvn -B verify sonar:sonar -Dsonar.projectKey=rastokme\_tp1 -Dsonar.organization=rastokme -Dsonar.host.url=https://sonarcloud.io -Dsonar.login=${{ secrets.SONAR\_TOKEN }}  --file ./simple-api/pom.xml

Documentation of Docker Configuration

Dockerfile:

Obrázok, na ktorom je text, snímka obrazovky, písmo, softvér

Automaticky generovaný popis

Docker-Compose Configuration:

version: '3.7'

services:

  db:

    container\_name: db

    build: ./database

    environment:

      POSTGRES\_DB: db

      POSTGRES\_USER: usr

      POSTGRES\_PASSWORD: pwd

    volumes:

      - pgdata:/var/lib/postgresql/data

    networks:

      - app-network

  simple-api:

    container\_name: simple-api

    build: ./simple-api

    environment:

      SPRING\_DATASOURCE\_URL: jdbc:postgresql://db:5432/db

      SPRING\_DATASOURCE\_USERNAME: usr

      SPRING\_DATASOURCE\_PASSWORD: pwd

    networks:

      - app-network

    depends\_on:

      - db

  httpd:

    container\_name: httpd

    build: ./httpd

    ports:

      - "80:80"

    networks:

      - app-network

    depends\_on:

      - simple-api

networks:

  app-network:

volumes:

  pgdata:

Access and Verify API Endpoints

I used curl to access API endpoints and verify the response.

Obrázok, na ktorom je text, snímka obrazovky, písmo, číslo

Automaticky generovaný popis

The quality gate results are satisfactory.

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis