



Универзитет Св. Кирил и Методиј - Скопје
Факултет за информатички науки и
компјутерско инженерство



Проектна задача по предметот

Континуирана интеграција и испорака

Апликација за менаџирање на курсеви

Наставници

Д-р Милош Јовановиќ

Д-р Панче Рибарски

Изработил

Хана Хасаницај 191511

Скопје, Јули 2023

Апликацијата која е користена за овој проект е Spring Boot апликација која користи PostgreSQL база на податоци. Проектот е дел од предметот Веб Програмирање, поточно ги содржу лабораториските вежби по предметот. Овозможува CRUD операции врз студенти, професори и курсеви (кои се посебни ентитети).

Користени алатки се:

Spring Boot (<https://spring.io/projects/spring-boot>)

PostgreSQL (<https://www.postgresql.org/>)

Docker (<https://www.docker.com/>)

Kubernetes (<https://kubernetes.io/>)

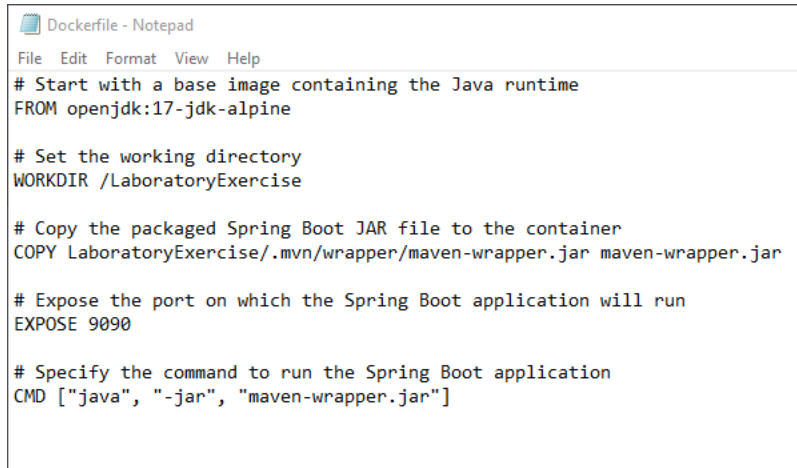
GitHub Actions

1. Креирање на јавен репозиториум (<https://github.com/vitruvianarendt/devops-project>)

```
git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/vitruvianarendt/devops-project.git
git push -u origin main
```

2. Docker

```
docker build -t projectkiii:latest .
docker images
```



The screenshot shows a Notepad window titled 'Dockerfile - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The content of the Dockerfile is as follows:

```
# Start with a base image containing the Java runtime
FROM openjdk:17-jdk-alpine

# Set the working directory
WORKDIR /LaboratoryExercise

# Copy the packaged Spring Boot JAR file to the container
COPY LaboratoryExercise/.mvn/wrapper/maven-wrapper.jar maven-wrapper.jar

# Expose the port on which the Spring Boot application will run
EXPOSE 9090

# Specify the command to run the Spring Boot application
CMD ["java", "-jar", "maven-wrapper.jar"]
```

```
H:\Projects\Dev
λ docker build -t projectkiii:latest .
[+] Building 25.4s (4/7)
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 500B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/openjdk:17-jdk-alpine 0.6s
=> [1/3] FROM docker.io/library/openjdk:17-jdk-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e 24.7s
=> => resolve docker.io/library/openjdk:17-jdk-alpine@sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9 0.0s
=> => sha256:4b6abae565492dbe9e7a894137c966a7485154238902f2f25e9dbd9784383d81 319B / 319B 0.0s
=> => sha256:a996cdcc040704ec6badaf5fecf1e144c096e00231a29188596c784bcf858d05 951B / 951B 0.0s
=> => sha256:264c9bdce361556ba6e685e401662648358988c01151c3d977f0fd777f7c26ab 3.48kB / 3.48kB 0.0s
=> => sha256:5843afab387455b37944e709ee8c78d7520df80f8d01cf7f861aae63beeddb6b 2.81MB / 2.81MB 2.0s
=> => sha256:53c9466125e464fed5626bde7b7a0f91aab09905f0a07e9ad4e930ae72e0fc63 928.44kB / 928.44kB 2.5s
=> => sha256:d8d715783b80cab158f5bf9726bcada5265c1624b64ca2bb46f42f94998d4662 47.19MB / 186.80MB 24.6s
=> => extracting sha256:5843afab387455b37944e709ee8c78d7520df80f8d01cf7f861aae63beeddb6b 0.5s
=> => extracting sha256:53c9466125e464fed5626bde7b7a0f91aab09905f0a07e9ad4e930ae72e0fc63 0.3s
=> [internal] load build context 0.0s
=> => transferring context: 204B 0.0s
```

3. Docker compose

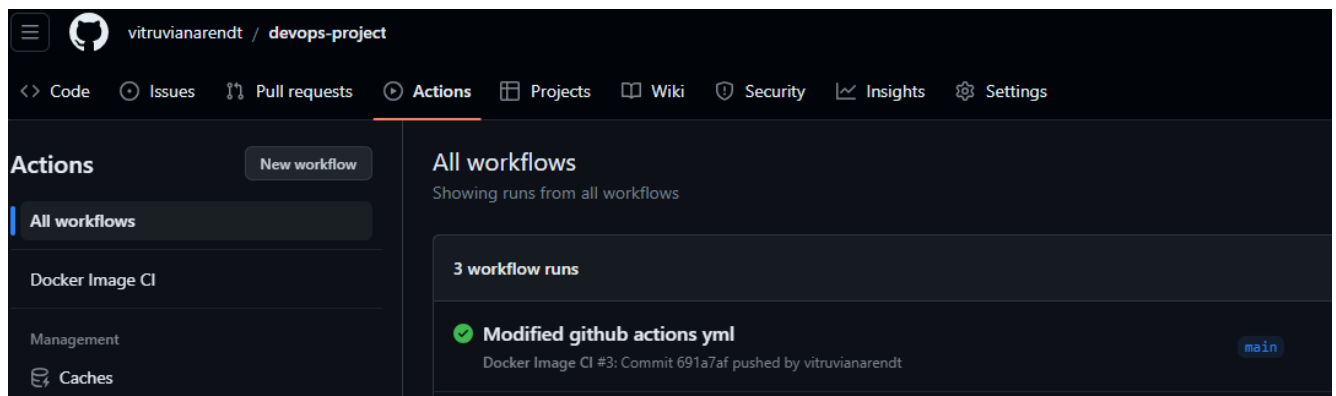
docker-compose up -d
docker-compose config

```
H:\Projects\Dev
λ docker-compose up -d
[+] Running 2/2
- Container dev-db-1 Started 2.8s
- Container dev-app-1 Started 4.4s

H:\Projects\Dev
λ docker-compose config
name: dev
services:
  app:
    build:
      context: H:\Projects\Dev
      dockerfile: Dockerfile
    depends_on:
      db:
        condition: service_started
    environment:
      POSTGRES_DB: e-shop
      POSTGRES_PASSWORD: root
      POSTGRES_USER: postgres
      SPRING_DATASOURCE_PASSWORD: root
      SPRING_DATASOURCE_URL: jdbc:postgresql://localhost:5432/e-shop
      SPRING_DATASOURCE_USERNAME: postgres
    networks:
      default: null
    ports:
      - mode: ingress
        target: 9091
```

```
docker-compose.yml - Notepad
File Edit Format View Help
version: '3'
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - 9091:9091
    depends_on:
      - db
    env_file:
      - .env
  db:
    image: postgres
    restart: always
    ports:
      - 5432:5432
    env_file:
      - .env
```

4. GitHub Actions



```
docker-image.yml - Notepad
File Edit Format View Help
name: Docker Image CI

on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v2

      - name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${ secrets.DOCKERHUB_USERNAME }
          password: ${ secrets.DOCKERHUB_TOKEN }

      - name: Build Docker Image
        run: docker build . --file Dockerfile --tag vitruvianarendt/projectkiii:latest

      - name: Push Docker Image
        run: docker push vitruvianarendt/projectkiii:latest
```

5. Kubernetes

k3d cluster create -a 1 -s 1 kubeproject1

k3d cluster create kubeproject2 -p "80:80@loadbalancer" -s 1 -a 1

kubectl apply -f namespace.yaml

kubectl get namespaces

kubectl apply -f service.yaml

kubectl get services -n devops

```
namespace.yaml - Notepad
File Edit Format View Help
apiVersion: v1
kind: Namespace
metadata:
  name: devops
```

```
service.yaml - Notepad
File Edit Format View Help
apiVersion: v1
kind: Service
metadata:
  name: devops-project-service
  namespace: devops
spec:
  selector:
    app: devops-project
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

```
kubectl apply -f database.yaml
kubectl get statefulsets -n devops
kubectl apply -f deployment.yaml
kubectl get pods -n devops
kubectl apply -f ingress.yaml
```

deployment.yaml - Notepad

```
File Edit Format View Help
apiVersion: v1
kind: ConfigMap
metadata:
  name: devops-project-postgres-config
  namespace: devops
data:
  POSTGRES_DB: e-shop
  POSTGRES_USER: postgres
---
apiVersion: v1
kind: Secret
metadata:
  name: devops-project-postgres-secret
  namespace: devops
type: Opaque
data:
  password: cm9vdA==
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: devops-project-deployment
  namespace: devops
spec:
  replicas: 5
  selector:
    matchLabels:
      app: devops-project
  template:
    metadata:
      labels:
        app: devops-project
    spec:
      containers:
        - name: projectkiii
          image: vitruvianarendt/projectkiii:latest
          imagePullPolicy: Always
```

ingress.yaml - Notepad

```
File Edit Format View Help
apiVersion: traefik.containo.us/v1alpha1
kind: Middleware
metadata:
  name: mcu-all
spec:
  stripPrefix:
    forceSlash: false
    prefixes:
      - /ver1
---
apiVersion: networking.k8s.io/v1
kind: Ingress
```

database.yaml - Notepad

```
File Edit Format View Help
apiVersion: v1
kind: ConfigMap
metadata:
  name: devops-project-postgres-config
  namespace: devops
data:
  POSTGRES_DB: e-shop
  POSTGRES_USER: postgres
---
apiVersion: v1
kind: Secret
metadata:
  name: devops-project-postgres-secret
  namespace: devops
type: Opaque
data:
  password: cm9vdA==
---
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: devops-project-postgres-pvc
  namespace: devops
  labels:
    app: devops-project
spec:
  storageClassName: local-path
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: devops-project-postgres
  namespace: devops
spec:
  serviceName: devops-project-postgres-service
  selector:
    matchLabels:
```