

#### 1. Краток опис на апликацијата

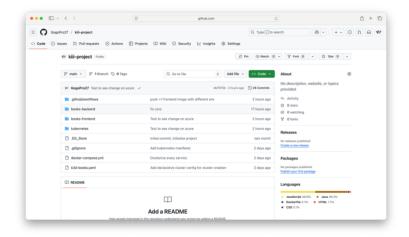
Апликацијата служи за менаџирање на книги и овозможува основни CRUD функционалности за менаџирање со истите. Самите сервиси на оваа апликација се направени со цел да се применат знаењата од предметот "Континуирана Интеграција и Испорака" односно знаењата поврзани со користење на Git и Github, Docker, Kubernetes (преку K3D) како и хостирање на апликација на Azure.

Апликацијата се состои од 3 сервиси имено backend во Spring Boot, frontend во React.js , како и базата на податоци so MongoDB.



### 2. Github репозиториум

Кодот на апликацијата е поставен на мој јавен Github репозиториум и може да се пристапи преку следниот линк: <a href="https://github.com/GogoPro27/kiii-project">https://github.com/GogoPro27/kiii-project</a>.



### 3. Докеризирање на апликацијата

Сите сервиси на оваа веб апликација, backend-от, frontend -от и базата се докеризирани. За оваа цел потребни беа два Dockerfile-ови, за Spring Boot и React сервисите. За кревање на базата на податоци, не ми е потребен таков фајл поради тоа што користам веќе готови слики од Dockerhub.

```
8 Raw □ ± 0 - 0
Code Blame 18 lines (11 loc) · 349 Bytes
   1 FROM node:18-alpine
       ARG VITE_API_BASE=http://localhost:8080
ENV VITE_API_BASE=$VITE_API_BASE
       WORKDIR /app
       COPY package*.json ./
       RUN npm ci
  11
  12
       # Build the app; Vite copies public/* into dist/, so runtime-config.js.template ends up in /app/dist
  13
  14
       RUN npm run build
  15
       EXPOSE 3000
  16
  17
  18 CMD ["npm", "run", "preview", "--", "--port", "3000", "--host"]
 Code Blame 12 lines (6 loc) · 153 Bytes
                                                                                                                 8 Raw □ ± 0 → 0
        FROM maven:3.9-eclipse-temurin-21
        WORKDIR /app
    6 COPY . .
        RUN mvn -B clean package -DskipTests
        EXPOSE 8080
   10
   11
   12 CMD ["sh", "-c", "java -jar target/*.jar"]
```

## 4. Оркестрација со Docker Compose

За поефикасно кревање на контејнерите како и полесно менаџирање и чување на историјата на промени, креиран е следниот docker-compose.yml:

```
Code Blame 37 lines (34 loc) · 735 Bytes
                                                                                                                                                 83 Raw □ ± 0 → 0
       services:
          mongodb:
           container_name: mongodb
           restart: unless-stopped
              - "27017:27017"
           volumes:
              - mongodb_data:/data/db
  10
  11
         books-backend:
            context: ./books-backend
  13
  14
              dockerfile: Dockerfile
           container_name: books-backend
  16
           restart: unless-stopped
          ports:
              - "8080:8080"
  19
          depends_on:
  20
              - mongodb
  21
             - SPRING_DATA_MONGODB_URI=mongodb://mongodb:27017/booksdb
  22
  23
  24
         books-frontend:
          build:
  25
  27
             dockerfile: Dockerfile
  28
           container_name: books-frontend
          restart: unless-stopped
ports:
              - "3000:3000"
  31
  33
             - books-backend
  34
  36
          mongodb_data:
           driver: local
```

## 5. Github Actions и Ci/Cd pipeline

Во прилог се поставени и слики од самиот pipeline со secrets кои се поставени во поставките на github репозиториумот.

Тој се состои од две задачи:

- Build & Push го build-а проектот, креира слика од сервисите и ја постира на dockerhub.
- Deploy се поврзува со виртуелната машина на Azure и користејќи ги претходно поставените слики на dockerhub , прави оркестрација со постоечкиот dockercompose.yml на инстанцата со новите сликисо што новата верзија на проектот е достапнана истата адреса.

```
Code Blame
                                   name: Build, Push, and Deploy (Docker Compose on Azure VM)
                                        on:
push:
branches: [ main ]
                                   permissions:
contents: read
                                teps:
    name: Checkout
    uses: actions/checkout@v4
                                                      - name: Get SHA
id: sha
run: echo *sha=${GITHUB_SHA}* >> $GITHUB_QUTPUT
                                                      - name: Set up QEMU
uses: docker/setup-qemu-action@v3
with:
platforms: arm64,amd64
                                                        - name: Set up Docker Buildx
uses: docker/setup-buildx-action@v3
                                                        - name: Docker Login
uses: docker/login-action@v3
with:
username: ${{ secrets.DOCKER_USERNAME }}
password: ${{ secrets.DOCKER_TOKEN }}
36
37
38
39
40
41
42
43
44
45
50
51
52
53
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
78
                                              - name: Build & Push Backend (multi-arch) uses: docker/build-push-action@v6 with: context: ./books-backend
                                                             push: true
platforms: linux/amd64,linux/arm64
                                                             $\{\secrets.DOCKER_USERNAME \}\/books-backend:latest $\{\secrets.DOCKER_USERNAME \}\/books-backend:\{\{\github.sha}\}\}
                                           - name: Bulld & Push Frontend (multi-arch)
uses: docker/build-push-action@v6
with:
context: //books-frontend
push: true
platforms: linux/am64/linux/arm64
build-args: |
VITE_API_BASE=/api
tass:
                                                         Vile_mrs_uner____

tags: |

${{ secrets.DOCKER_USERNAME }}/books-frontend:latest

${{ secrets.DOCKER_USERNAME }}/books-frontend:${{ github.sha }}}
                                             - name: Build & Push Frontend for KBs (multi-arch, empty base)
uses: docker/build-push-action@v6
with:
context: ./books-frontend
push: true
platforms: linux/am64,linux/arm64
build-arps: |
VITE_API_BASE=
tass: |
                                                           VIII_AMs_Bhace-
tags: |
${{ secrets.DOCKER_USERNAME }}/books-frontend-kBs:latest
${{ secrets.DOCKER_USERNAME }}/books-frontend-kBs:${{ github.sha }}}
}
                            deploy:
runs-on: ubuntu-latest
needs: build-and-push
envi
1MOGE_TAG: ${{ needs.build-and-push.outputs.sha}}
1DOGE_TAG: ${{ needs.build-and-push.outputs.sha}}
1DOGETAG: ${{ 
teps:
- name: Checkout (for possible future file upload)
uses: actions/checkout@v4
                                        - name: Doploy via SBH to Azure VM
uses: appliably/ssh-actionVyl.2.8
with:
host: ${{ secrets.SSH_MDST }}
username: ${{ secrets.SSH_MDST }}
por: ${{ secrets.SSH_MST }}
key: ${{ secrets.SSH_MST }}
script.script.|
script.|
st-e-
cd /opt/books-app
                                                               if [ ! -f docker-compose.yml ]; then
    echo 'docker-compose.yml not found in /opt/books-app'
    exit 1
fi
                                                                # Prepare env for tag and username; you can also maintain a .env on the VM echo "IMAGE_TAG=$(IMAGE_TAG)" > .deploy.env echo "DOCKER_USERNAME=$(DOCKER_USERNAME)" >> .deploy.env
                                                               # Pull and restart with new images
set -a
. .deploy.env
set +a
                                                                docker compose pull
docker compose up -d --remove-orphans
                                                               # Optional: prune old images to free space
docker image prune -f
```

# 6. Кубернетес манифести

## K3d-books.yaml

```
Raw □ ± Ø ▼ ○
Code
       Blame
   1 apiVersion: k3d.io/v1alpha5
   2 kind: Simple
   3 metadata:
       name: books
   5 servers: 1
   6 agents: 1
   7 ports:
       - port: 8080:80
  8
        nodeFilters:
  10

    loadbalancer

       - port: 8443:443
  11
        nodeFilters:
- loadbalancer
  12
 13
 14 options:
 15
       kubeconfig:
        updateDefaultKubeconfig: true
switchCurrentContext: true
  16
  17
```

## Kubernetes/00-namespace.yaml

```
Code Blame

1 apiVersion: v1
2 kind: Namespace
3 metadata:
4 name: books-app
5 labels:
6 name: books-app
```

Kubernetes/01-mongodb.yaml

```
Code Blame 97 lines (94 loc) - 2.89 KB

apiVersion: v1
kind: ConfigMap
sersadars:
anetadars:
anetadars:
anetadars:
anetadars:
anetadars:
anetadars:
apiVersion: v1
kind: Secret

apiVersion: v2

apiVersion: v3
kind: Secret

apiVersion: v4
kind: Secret

apiVersion: v6
data:

apiVersion: v8
apiVersion: appiVersion: appiVersion:
```

## Kubernetes/02-backend.yaml

### Kubernetes/03-frontend.yaml

```
Code Blame 47 lines (46 loc) · 953 Bytes

1 apiVersion: apps/v1 kind: Deployment
2 kind: Deployment
3 metadata:
4 name: books-frontend
5 namespace: books-app
6 spec:
7 restactor:
8 app: books-frontend
11 template:
12 metadata:
13 labels:
14 app: books-frontend
15 spec:
16 containers:
17 - name: books-frontend
18 image: gochot/books-frontend-kBs:latest
19 imagePullPolicy: IfNotPresent
19 ports:
21 - containerPort: 3800
18 name: http
22 port: 3900
25 initialDelaySeconds: 5
28 livenessProbe:
29 tcpSocket:
30 port: 3800
1 initialDelaySeconds: 15
20 port: 3800
21 iname: books-frontend
22 port: 3800
23 namespace: books-app
24 spi-books-frontend
25 periodSeconds: 15
26 periodSeconds: 10
27 periodSeconds: 10
28 app: books-frontend
29 namespace: books-app
30 namespace: books-app
31 names books-frontend
32 namespace: books-frontend
33 namespace: books-app
34 spi-books-frontend
35 app: books-frontend
36 app: books-frontend
37 metadata:
38 namespace: books-app
39 spec:
40 app: books-frontend
41 app: books-frontend
42 app: books-frontend
43 ports: 3800
44 tragetPort: 3800
45 tragetPort: 3800
46 targetPort: 3800
47 type: ClusterIP
```

## Kubernetes/04-ingress.yaml

```
Code
      Blame 26 lines (26 loc) · 612 Bytes
                                                                                 Raw 🗗 🕹 🕖 🕶
      apiVersion: networking.k8s.io/v1
  2
      kind: Ingress
  3
      metadata:
  4
        name: books-ingress
  5
        namespace: books-app
  6
        annotations:
         traefik.ingress.kubernetes.io/router.entrypoints: web
  8
 9
        ingressClassName: traefik
 10
        rules:
 11
         - http:
 12
              paths:
 13
                - path: /api
                  pathType: Prefix
 15
                  backend:
 16
                    service:
 17
                     name: books-backend
 18
                      port:
                       number: 8080
 19
               - path: /
 20
 21
                  pathType: Prefix
 22
                  backend:
 23
                   service:
                      name: books-frontend
 24
 25
                      port:
 26
                        number: 3000
```