Script for Ratatuilay Project and Deployed using Docker and Kubernetes

Members:

รหัสนักศึกษา 64090500405 นางสาวปุณณภา เทียนชัย

รหัสนักศึกษา 64090500443 นายภูมิพัฒน์ กรเจริญพิสุทธิ์

รหัสนักศึกษา 64090500451 นายสิริภัทร นิลประเสริฐ

URL to VDO clip: https://youtu.be/3GVSJVlhU-w

Project GitHub link: o0SoloWolf0o/MiniProject-OOP-DB (github.com)

Docker file:

```
FROM openjdk:19

VOLUME /tmp

ARG JAR_FILE

WORKDIR /app

COPY target/miniproject-0.0.1-SNAPSHOT.jar /app/miniproject.jar

ENTRYPOINT ["java","-jar","/app/miniproject.jar"]
```

Docker-compose.yml:

```
version: '3.8'

services:

  mysql-container:

    container_name: mysql-container

    image: mysql

    ports:

      - 8081:3306

    environment:

      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}

      MYSQL_DATABASE: miniproject

    networks:
```

```
    - springboot-mysql
  healthcheck:
    test: ["CMD", "mysqladmin", "ping", "-h", "localhost", "-u", "root", "--password=password"]
    interval: 10s
    timeout: 5s
    retries: 5
  deploy:
    resources:
      limits:
        cpus: '0.50'
        memory: 512M
  restart: unless-stopped
springboot-container:
  build: .
  container_name: springboot-container
  image: pumipat/miniproject:latest
  ports:
    - 8080:8080
  environment:
    SPRING_DATASOURCE_URL: jdbc:mysql://mysql.my-spring-boot.svc.cluster.local:3306/miniproject
    SPRING_DATASOURCE_USERNAME: ${SPRING_DATASOURCE_USERNAME}
    SPRING_DATASOURCE_PASSWORD: ${MYSQL_ROOT_PASSWORD}
  networks:
    - springboot-mysql
  depends_on:
    mysql-container:
      condition: service_healthy
  deploy:
      resources:
        limits:
          cpus: '0.50'
          memory: 512M
    restart: unless-stopped
networks:
  springboot-mysql:
    driver: bridge
```

my-spring-boot-namespace.yaml:

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-spring-boot
```

mysql-service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: mysql
  namespace: my-spring-boot
spec:
  selector:
    app: mysql
  ports:
  - protocol: TCP
    port: 3306
    targetPort: 3306
```

spring-boot-service.yaml:

```yaml
apiVersion: v1

kind: Service

metadata:

  name: springboot-container-service

  namespace: my-spring-boot

  annotations:

    service.beta.kubernetes.io/azure-load-balancer-internal: "false"

spec:

  type: LoadBalancer

  selector:

    app: springboot-container

  ports:

  - protocol: TCP

    port: 80

    targetPort: 8080
```

mysql-secret.yaml:

```yaml
apiVersion: v1

kind: Secret

metadata:

  name: mysql-secret

  namespace: my-spring-boot

type: Opaque

data:

  MYSQL_ROOT_PASSWORD: cGFzc3dvcmQ= # password
```

spring-boot-secret.yaml:

```yaml
apiVersion: v1

kind: Secret
```

```
metadata:
  name: azure-mysql-secret
  namespace: my-spring-boot
type: Opaque
data:
  # MYSQL_PASSWORD: UGFzc3dvcmQhQA== # UGFzc3dvcmQhQA== (encode to base64 Password!@)
  MYSQL_PASSWORD: cGFzc3dvcmQ= # password
```

mysql-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  namespace: my-spring-boot
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        image: mysql:latest
        ports:
```

```
        - containerPort: 3306
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysql-secret
              key: MYSQL_ROOT_PASSWORD
        - name: MYSQL_DATABASE
          value: miniproject
        resources:
          limits:
            cpu: "0.50"
            memory: 512Mi
```

spring-boot-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: springboot-container
  namespace: my-spring-boot
spec:
  replicas: 3
  selector:
    matchLabels:
      app: springboot-container
  template:
    metadata:
      labels:
        app: springboot-container
    spec:
      containers:
      - name: springboot-container
        image: pumipat/miniproject:latest
        ports:
        - containerPort: 8080
        env:
        - name: SPRING_DATASOURCE_URL
          value: jdbc:mysql://mysql.my-spring-boot.svc.cluster.local:3306/miniproject
        - name: SPRING_DATASOURCE_USERNAME
          value: root
        - name: SPRING_DATASOURCE_PASSWORD
          value: password
        resources:
          limits:
            cpu: "0.50"
            memory: 512Mi
```

spring-boot-configmap.yaml:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: spring-boot-configmap
  namespace: my-spring-boot
data:
  APP_ENV: "production"
  LOG_LEVEL: "info"
```

Apply on K8s

ทำการสร้าง aks Azure Kubernetes Service ใน https://portal.azure.com/#home และทำการกำหนด spec ตามที่ต้องการ

```
# Create the my-spring-boot namespace

kubectl apply -f my-spring-boot-namespace.yaml
```

```
# Create the MySQL resources

kubectl apply -f mysql-secret.yaml

kubectl apply -f mysql-deployment.yaml

kubectl apply -f mysql-service.yaml
```

```
# Create the application resources

kubectl apply -f spring-boot-configmap.yaml

kubectl apply -f spring-boot-secret.yaml

kubectl apply -f spring-boot-deployment.yaml

kubectl apply -f spring-boot-service.yaml
```

```
# Check the status of the MySQL deployment

kubectl get deployments -n my-spring-boot
```

```
# Check the status of the MySQL pod

kubectl get pods -n my-spring-boot
```

```
# Check the status of the MySQL service

kubectl get services -n my-spring-boot
```

```
# Check the status of the Spring Boot deployment

kubectl get deployments -n my-spring-boot
```

```
# Check the status of the Spring Boot pod

kubectl get pods -n my-spring-boot


# Check the status of the Spring Boot service

kubectl get services -n my-spring-boot

kubectl exec -it mysql-<pod-id> -n <namespace> -- mysql -u root -p

kubectl get svc springboot-container-service -n my-spring-boot -w
```

สามารถเข้าไปใช้งาน website ได้ผ่าน: http://20.187.250.73/

หมายเหตุ: ในการทดสอบการใช้งานฟังก์ชั่นหลักโดยสามารถ CRUD ได้ สามารถทำได้ผ่าน Admin account

(สามารถใช้admin ac. นี้ได้ emai: p.kornpisuit@gmail.com, password: asdasd)