ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд.тех.наук |  |  |  | П.А. Степанов |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАЗРАБОТКА МИКРОСЕРВИСА |
| по курсу: ТЕХНОЛОГИИ РАЗРАБОТКИ СЕРВЕРНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4136 |  |  |  | Н.С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Задание (Вариант 3)**

Расписание занятий в институте

**Описание разрабатываемого продукта**

1 Подготовьте Ваше приложение к разворачиванию в облачном сервисе или  
компоненте Docker.  
2 Реализуйте файл docker-compose.xml, который будет содержать все необходимые для  
работы Вашего приложения ресурсы  
3 Внимание. В виду того, что далеко не на каждом компьютере можно запустить систему виртуализации, данная лабораторная работа сдается в электронном виде, без демонстрации преподавателю.

**Текст программы**

**Dockerfile**

FROM docker pull eclipse-temurin:17.0.9\_9-jdk  
RUN mkdir -p /usr/src/myapp  
COPY target/Lab3.jar /usr/src/myapp  
RUN mkdir -p /usr/src/myapp/target  
COPY target/keystore.p12 /usr/src/myapp/target  
WORKDIR /usr/src/myapp  
VOLUME /tmp  
ARG JAR\_FILE  
ADD ${JAR\_FILE} app.jar  
ENTRYPOINT ["java","-jar","/Lab3.jar"]

**docker-compose.yml**

version: "3.9"  
services:  
 zookeeper:  
 image: confluentinc/cp-zookeeper:latest  
 ports:  
 - 22181:2181  
 - 2888:2888  
 - 3888:3888  
 environment:  
 ZOOKEEPER\_CLIENT\_PORT: 2181  
 ZOOKEEPER\_TICK\_TIME: 2000  
 kafka:  
 image: confluentinc/cp-kafka:latest  
 depends\_on:  
 - zookeeper  
 ports:  
 - 29092:9092  
 environment:  
 KAFKA\_BROKER\_ID: 1  
 KAFKA\_ZOOKEEPER\_CONNECT: zookeeper:2181  
 KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://kafka:9092,PLAINTEXT\_HOST://localhost:29092  
 KAFKA\_LISTENER\_SECURITY\_PROTOCOL\_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT\_HOST:PLAINTEXT  
 KAFKA\_INTER\_BROKER\_LISTENER\_NAME: PLAINTEXT  
 KAFKA\_OFFSETS\_TOPIC\_REPLICATION\_FACTOR: 1  
 schema-registry:  
 image: confluentinc/cp-schema-registry:7.0.1  
 ports:  
 - 8181:8181  
 - 8081:8081  
 environment:  
 - SCHEMA\_REGISTRY\_KAFKASTORE\_BOOTSTRAP\_SERVERS=kafka:9092  
 - SCHEMA\_REGISTRY\_HOST\_NAME=schema-registry  
 - SCHEMA\_REGISTRY\_LISTENERS=http://schema-registry:8081  
 links:  
 - zookeeper  
 consumer:  
 build:  
 context: Lab3  
 environment:  
 - KAFKA\_BOOTSTRAPADDRESS=kafka:9092  
 - KAFKA\_TOPIC=test  
 - KAFKA\_GROUPID=7655  
 links:  
 - kafka  
 depends\_on:  
 - kafka  
 kafka-ui:  
 image: provectuslabs/kafka-ui  
 container\_name: kafka-ui  
 ports:  
 - "8090:8080"  
 restart: always  
 environment:  
 - KAFKA\_CLUSTERS\_0\_NAME=local  
 - KAFKA\_CLUSTERS\_0\_BOOTSTRAPSERVERS=kafka:9092  
 - KAFKA\_CLUSTERS\_0\_ZOOKEEPER=zookeeper:2181  
 - KAFKA\_CLUSTERS\_0\_SCHEMAREGISTRY=http://schema-registry:8081  
 links:  
 - kafka  
 - zookeeper

**pom.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.1.4</version>  
 <relativePath/>  
 </parent>  
 <groupId>com.example</groupId>  
 <artifactId>Lab3</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <name>Lab3</name>  
 <description>Demo project for Spring Boot</description>  
 <properties>  
 <docker.image.prefix>springio</docker.image.prefix>  
 <java.version>17</java.version>  
 </properties>  
 <dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>6.0.9</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 </dependencies>  
 </dependencyManagement>  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>com.spotify</groupId>  
 <artifactId>dockerfile-maven-plugin</artifactId>  
 <version>1.3.6</version>  
 <configuration>  
 <repository>${docker.image.prefix}/${project.artifactId}</repository>  
 <buildArgs>  
 <JAR\_FILE>target/${project.build.finalName}.jar</JAR\_FILE>  
 </buildArgs>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>