Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе № 5

Тема: «»

Дисциплина: «Технологии сетевого программирования»

Выполнили: Зайнуллова А.М.

Тарасов П.Д.

Группа: 6304-010302D

Самара, 2025

Описание структуры контейнеризации и настройки окружения

1. **Сервис PostgreSQL (postgres)**

Образ: Используется официальный образ PostgreSQL версии 15 на базе Alpine Linux (легковесный дистрибутив).

Порты: Порт 5432 пробрасывается на хост для доступа к БД.

Переменные окружения:

POSTGRES\_USER: Логин пользователя (postgres)

POSTGRES\_PASSWORD: Пароль (pa33word)

POSTGRES\_DB: Имя создаваемой БД (EasyShopDocker)

Том данных: postgres\_data для сохранения данных БД между перезапусками контейнера.

Healthcheck: Проверка доступности БД через pg\_isready каждые 5 секунд.

1. **Сервис приложения (easyshop-backend)**

Многоэтапная сборка:

Этап сборки (Builder):

Используется образ eclipse-temurin:23-jdk (JDK для Java 23).

Устанавливается Maven для сборки проекта.

Копируются pom.xml и исходный код, затем выполняется сборка JAR-файла.

Этап запуска (Runtime):

Используется образ eclipse-temurin:23-jre (только JRE для работы).

Копируется собранный JAR-файл и запускается приложение.

Порты: Порт 8080 пробрасывается на хост.

Переменные окружения:

Подключение к БД: URL, логин, пароль.

Hibernate/JPA: Настройки логирования SQL, диалект PostgreSQL, отключение автоматической генерации DDL.

Flyway: Включение миграций, путь к SQL-скриптам (classpath:Database), настройки префиксов/суффиксов.

JWT: Секретный ключ и время жизни токена (24 часа).

Зависимости: Запускается только после успешного старта PostgreSQL (service\_healthy).

Том: Монтирование директории target для доступа к артефактам.

Политика перезапуска: unless-stopped (автоматический перезапуск при падении).

**Архитектура приложения**

**1. Технологический стек**

Backend: Java (сборка через Maven), Spring Boot.

База данных: PostgreSQL.

Миграции БД: Flyway (SQL-скрипты в src/main/resources/Database).

Аутентификация: JWT (JSON Web Tokens).

ORM: Hibernate/JPA для работы с БД.

**2. Компоненты приложения**

Слой данных:

PostgreSQL: Хранение данных.

Flyway: Управление миграциями (версионирование схемы БД).

Hibernate/JPA: Маппинг сущностей БД на Java-классы.

Бизнес-логика:

Spring Boot: Основной фреймворк для обработки запросов.

Сервисы: Логика приложения (например, управление пользователями, товарами и т.д.).

Репозитории: Интерфейсы JPA для доступа к данным.

API и безопасность:

REST-контроллеры: Обработка HTTP-запросов.

JWT: Аутентификация и авторизация (секретный ключ и время жизни токена заданы в переменных окружения).

Контейнеризация:

Docker: Изоляция сервисов (PostgreSQL и приложения).

Docker Compose: Оркестрация многоконтейнерного окружения.

**3. Особенности**

Изолированное тестирование: БД и приложение запускаются в отдельных контейнерах.

Гибкость настроек: Параметры БД, JWT и Flyway задаются через переменные окружения.

Масштабируемость: Контейнеры могут быть развернуты в облаке (например, Kubernetes).

**4. Схема взаимодействия**

Пользователь -> [HTTP:8080] -> easyshop-backend -> [JDBC:5432] -> postgres

Приложение доступно на порту 8080, БД — на порту 5432 (внутри сети Docker).

**5. Безопасность**

Пароли: Хранятся в открытом виде в docker-compose.yml (небезопасно! Для прода лучше использовать секреты Docker или Vault).

JWT: Секретный ключ зашит в переменные окружения.

**6. Логирование**

Hibernate логирует SQL-запросы (SPRING\_JPA\_SHOW\_SQL: "true").