Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Уральский Федеральный Университет имени первого президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Отчет по лабораторной работе №1

«Основы работы с Docker и PostgreSQL»

по дисциплине «Разработка приложений»

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_ Федоров А. Д.

Группа РИМ-150950

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмин Д. И.

Екатеринбург, 2025

Цель работы

Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

Задачи

1. Установить и проверить работу Docker.
2. Изучить базовые команды Docker.
3. Запустить контейнер с PostgreSQL в изолированном режиме.
4. Запустить контейнер с pgAdmin и подключить его к контейнеру с БД через сеть Docker.
5. Подключиться к БД из pgAdmin, создать схему и выполнить запросы.
6. Обеспечить сохранность данных БД с помощью томов Docker.

# Установка и проверка Docker

Установив Docker Desktop, WSL запустим проверку версии и тестовый образ (Рисунок 1).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 - Проверка работы Docker

Приветственное сообщение вывелось правильно, следовательно Docker установлен и корректно работает.

# Базовые команды Docker. Работа с образами и контейнерами

## Просмотр информации

Опробуем базовые команды (Рисунок 2).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 - Базовые команды Docker

## Запуск простого контейнера

На примере Nginx запустим новый контейнер (Рисунок 3).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 - Запуск контейнера

Проверим работоспособность контейнера (Рисунок 4).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 - Работа Nginx

## Остановка и удаление контейнера

Проверив работоспособность контейнера, остановим его и удалим (Рисунок 5).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 - Остановка и удаление контейнера

# Запуск PostgreSQL в контейнере

## Запуск контейнера с PostgerSQL

Запустим новый контейнер с установленными переменными окружения (Рисунок 6).

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 - Запуск контейнера с PostgresSQL

Проверим запуск (Рисунок 7).

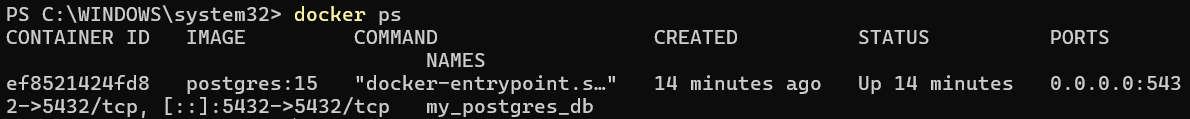


Рисунок 7 - Проверка запуска контейнера

## Подключение к БД из контейнера

Подключимся к БД прямо из контейнера через psql (Рисунок 8).

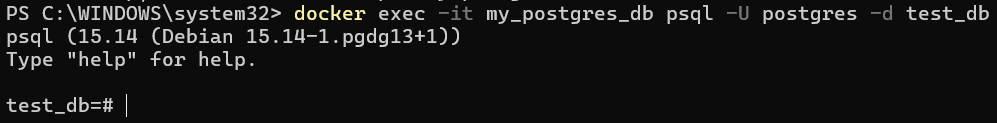


Рисунок 8 - Подключение к БД

Проверим список всех баз данных через команду \l (Рисунок 9).

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 9 - Список баз данных

Проверим список таблиц в текущей БД (Рисунок 10).

Изображение выглядит как текст, Шрифт, черный, Графика

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 10 - Список таблиц в текущей БД

Создадим новую таблицу и добавим туда данные (Рисунки 11-12).



Рисунок 11 - Создание таблицы



Рисунок 12 - Добавление данных

Проверим заполнение таблицы (Рисунок 13).

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 13 - Заполненная таблица

# Подключение к БД через pgAdmin из второго контейнера

Создадим сеть чтобы контейнеры могли видеть друг друга (Рисунок 14).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 14 - Создание сети

Подключим наш контейнер с PostgresSQL к сети (Рисунок 15).



Рисунок 15 - Подключение контейнера к сети

Запустим pgAdmin в той же сети (Рисунок 16).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 16 - Запуск pgAdmin

Настроим подключение в pgAdmin, и проверим подключение с помощью запроса к БД (Рисунок 17).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 17 - Проверка запроса

Подключение прошло успешно.

# Сохранение данных с помощью Томов (Volumes)

Остановим и удалим ранее созданные контейнеры (Рисунок 18).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 18 - Остановка и удаление контейнеров

Запустим новый контейнер с PostgreSQL, подключив том (Рисунок 19).

Запуск нового контейнера (Рисунок 19).

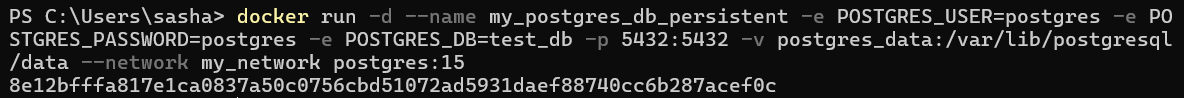


Рисунок 19 - Новый контейнер

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 20 - Данные не сохранились

Добавление новых данных (Рисунок 21).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 21 - Новые данные

# Перенос конфигурации контейнеров в docker-compose.yaml

Был создан файл docker-compose.yaml. Далее была проверка работоспособности данных из файла (Рисунок 22).

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.



Рисунок 22 – Проверка работоспособности

Ответы на вопросы

1. Docker — платформа для контейнеризации приложений (упаковка программы со всеми её зависимостями в изолированную среду)

2. Тома – средства постоянного хранения информации, сети – для соединения докер-контейнеров между собой.

3. Через команды docker exec\docker attach

4. pgAdmin нужен для администрирования и настройки СУБД PostgreSQL

Вывод

В ходе лабораторной работы удалось познакомиться с Docker. Получилось создать контейнеры и подключить их в одну сеть.

Был запущен PostgreSQL в контейнере, а также было проведено подключение к базе данных через pgAdmin из второго контейнера.

Был создан файл docker-compose.yaml – текстовый файл конфигурации. В контексте Docker он описывает, какие контейнеры нужно поднять, с какими настройками, сетями и томами.