**Лабораторная работа №1: Основы работы с Docker и PostgreSQL**

**Студент:** Осьминин Никита Борисович

**Группа:** Прикладной анализ данных, РИМ-150950

**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.

**Описание задачи**

Необходимо было установить Docker, запустить контейнер с Postgres и подключиться к нему через pgAdmin. Для сохранения данными между запусками контейнеров был добавлен том. В конце работы для удобства вся конфигурация была вынесена в compose-файл.

**Ход работы**

**Часть 0: Установка и проверка Docker**

Docker у меня уже был установлен

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Для того, чтобы убедиться в этом, я использовал команду **docker run hello-world** , которая скачивает тестовый имадж и выводит запись в консоль.

**Часть 1: Базовые команды Docker. Работа с образами и контейнерами**

Изображение выглядит как текст, программное обеспечение, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Я проверил список всех имаджей, находящихся в системе командой **docker images**, а также проверил список запущенных контейнеров (команда **docker ps**) и список всех контейнеров (**команда docker ps -a**)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Далее я запулил образ nginx и запустил его на порту 8080

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

После проверки я удалил контейнер с вебсервером

**Часть 2: Запуск PostgreSQL в контейнере**

Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Я запустил контейнер с образом postgres и проверил, что он действительно активен

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Далее я напрямую подключился к СУБД в контейнере из терминала

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Для того, чтобы подключиться из контейнера с pgAdmin к СУБД, находящуюся в другом контейнере, я создал новую сеть

Изображение выглядит как текст, снимок экрана, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

В сеть я добавил уже имеющийся контейнер с Postgres и запустил новый с pgAdmin, подключённый к той же сети

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

На порту 8080 теперь находится pgAdmin, данные для входа в который были заданы переменными окружения при запуске контейнера

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Для подключения к СУБД я указал данные из параметров запуска контейнера с Postgres

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Теперь из админки можно напрямую писать SQL-запросы к СУБД

**Часть 4: Сохранение данных с помощью Томов (Volumes)**

При перезапуске контейнера, все данные будут утеряны, т.к файловая система контейнера никак не привязан к файловой системе ПК

Изображение выглядит как текст, Шрифт, Цвет электрик, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Для того, чтобы избежать этой проблемы, я остановил контейнер с СУБД

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

А далее создал новый том, в котором и будут сохраняться данные

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Новый том находится в самом низу списка и называется postgres\_data

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

В дальнейшем я запускал контейнер с СУБД, привязывая к ней созданный том

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.

Т.к в перезапущенном контейнере ещё не была создана таблица users, то том является пустым и любые SQL-запросы к прежней таблице будут падать в ошибку

Изображение выглядит как текст, Шрифт, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Пересоздадим таблицу в СУБД

Изображение выглядит как текст, снимок экрана, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.

Проверим, что данные возвращаются через запрос

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Изображение выглядит как текст, программное обеспечение, снимок экрана, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

А также то, что они сохраняются при перезапуске контейнеров

**Часть 5: Перенос конфигурации контейнеров в docker-compose.yaml**

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

После мануального запуска контейнеров с pgAdmin и Postgres, я конфигурировал docker-compose.yaml файл для автоматизации развёртывания контейнеров

**Вопросы**

**Что такое Docker?**  
Это кроссплатформенная программа, позволяющая запускать в изолированном окружении различные образы, объединять их в единые сети и привязывать к ПК пользователя.

**Для чего нужны тома и сети Docker?**

Тома нужны для сохранения данных при перезапуске контейнеров. Сети нужны для коммуникации контейнеров между собой.

**Как подключиться к контейнеру и выполнить в нём команды?**  
Через docker exec -it <имя\_контейнера> <команда>

**Для чего нужен pgAdmin?**  
Это интерфейс для работы с PostgreSQL. Можно визуализировать запросы, таблицы, список колонок и т.д без нужды работать в терминале.

**Вывод**

В ходе работы я научился разворачивать окружение в изолированных контейнерах Docker, объединять их в общие сети и использовать тома для сохранения данных. Также научился оформлять конфигурацию через docker-compose для автоматизации развёртывания