# **Разработка приложений**

# **Лабораторная работа №1**

# **Основы работы с Docker и PostgreSQL**

**Студент:** Шмыголь Александр Петрович

**Группа:** РИМ-150950

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

**Цель работы:** Освоить фундаментальные концепции и базовые операции Docker: создание обра

зов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

**Задачи:**

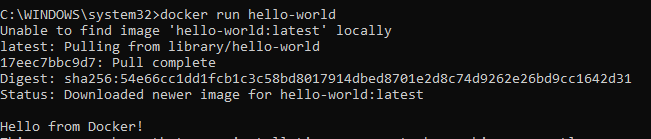
1. Установить и проверить работу Docker.
2. Изучить базовые команды Docker.
3. Запустить контейнер с PostgreSQL в изолированном режиме.
4. Запустить контейнер с pgAdmin и подключить его к контейнеру с БД через сеть Docker.
5. Подключиться к БД из pgAdmin, создать схему и выполнить запросы.
6. Обеспечить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

## **Часть 0: Установка и проверка Docker**

Проверка установленных версий

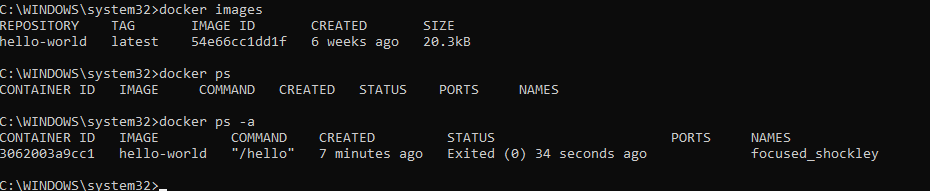




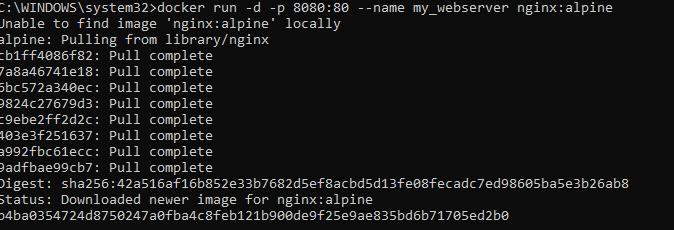
Запуск тестового образа  


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

1. **Просмотр информации:**

****

1. **Запуск простого контейнера (на примере Nginx):**



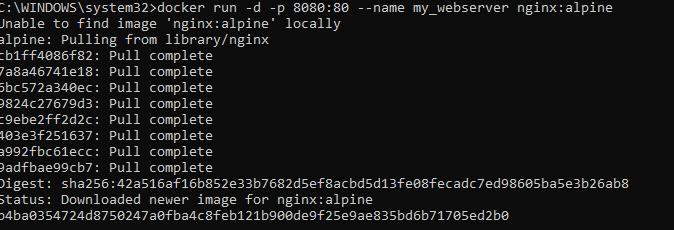
**-d** — запуск контейнера в фоне.

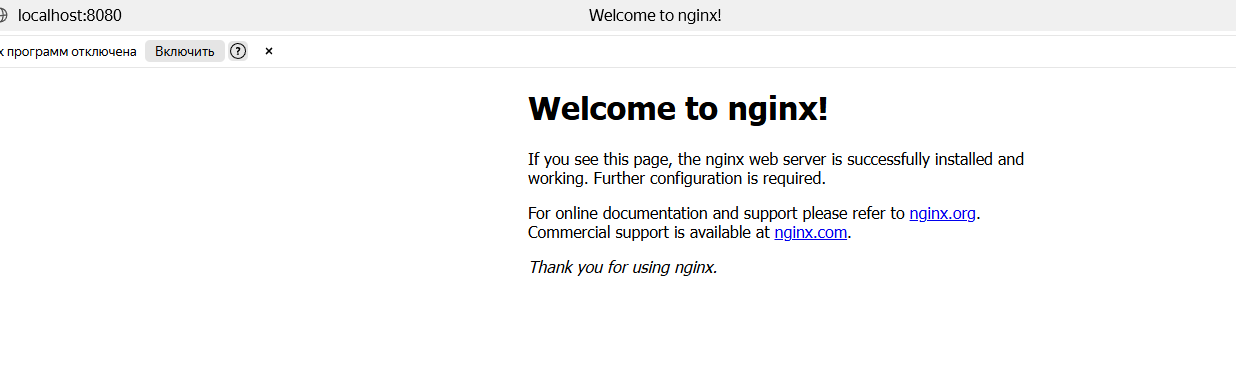
**-p 8080:80** — проброс порта (хост:контейнер).

**--name** — имя контейнера.

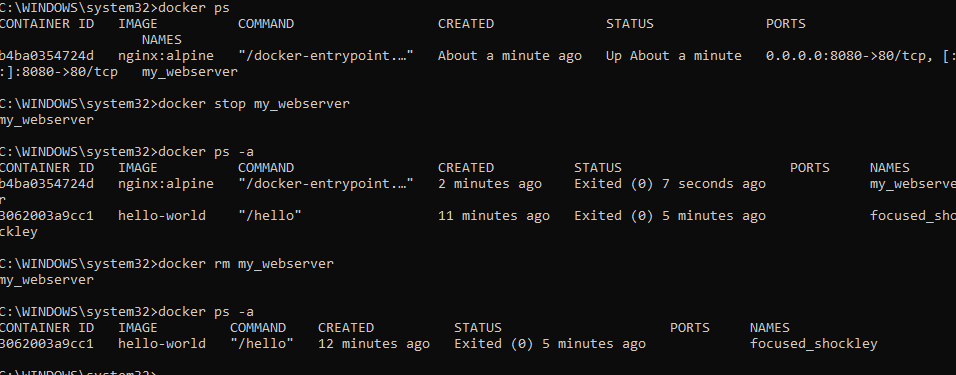
**nginx:alpine** — имя образа и его тег (версия).

1. **Проверка работы**



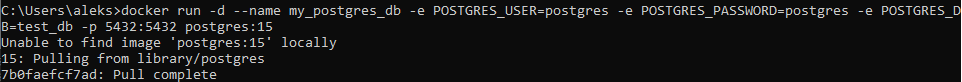


1. **Остановите и удалите контейнер**



## **Часть 2: Запуск PostgreSQL в контейнере**

1. **Запуск контейнера с PostgreSQL:**



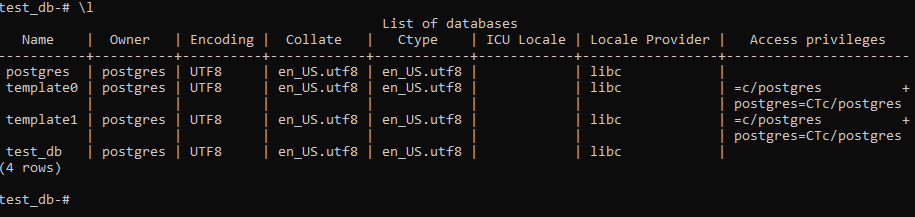
**-e** — установка переменных окружения внутри контейнера.

**-p** – проброс портов для работы с БД внешних систем

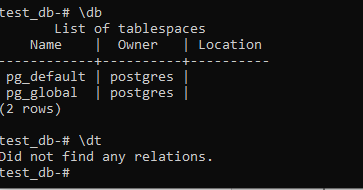
Переменные **POSTGRES\_\*** необходимы для первоначальной настройки СУБД.

1. **Подключение к БД из контейнера (через psql):**

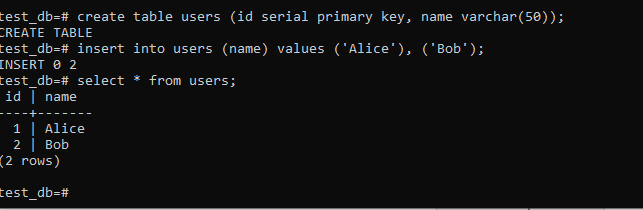
Список всех баз данных



Список таблиц в текущей бд



Создание таблицы и тестовые запросы

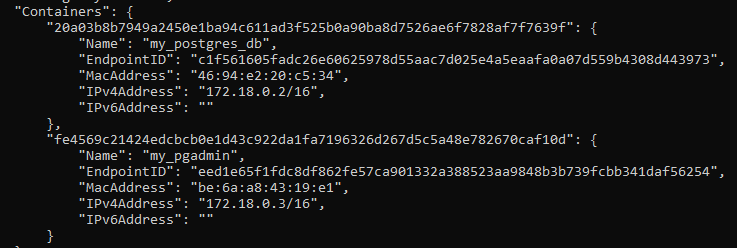


## **Часть 3: Подключение к БД через pgAdmin из второго контейнера**

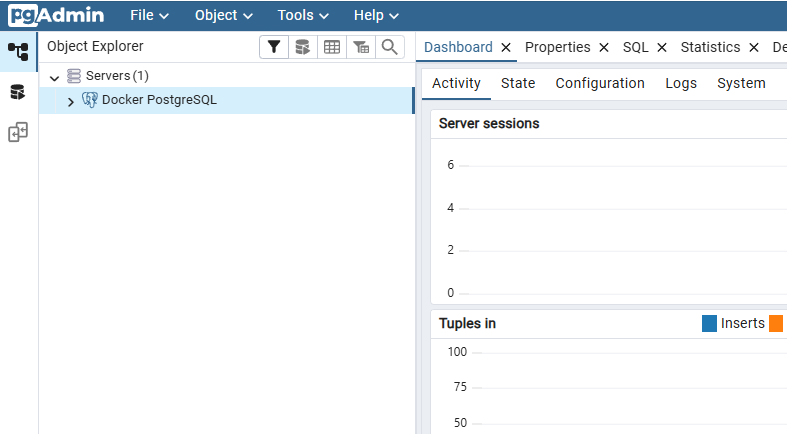
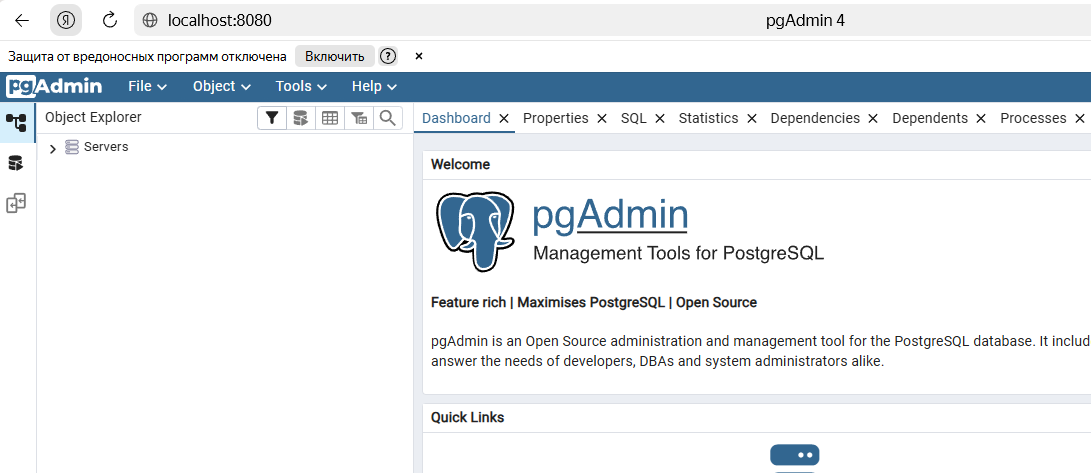
Запуск PgAdmin



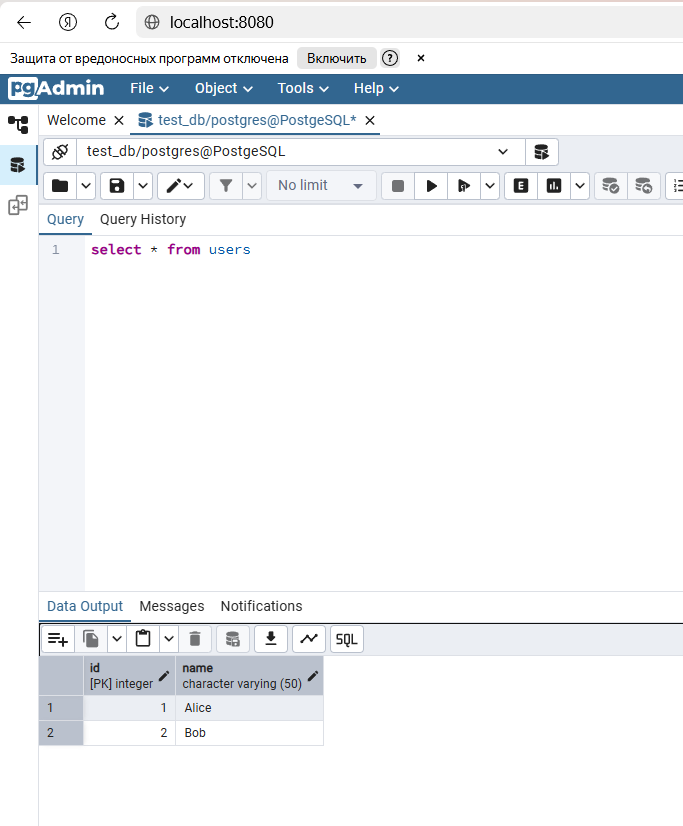
Подключение контейнеров к сети



Настройка подключения в pgAdmin

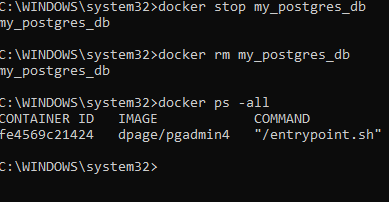


Выполнение запроса через Query Tool

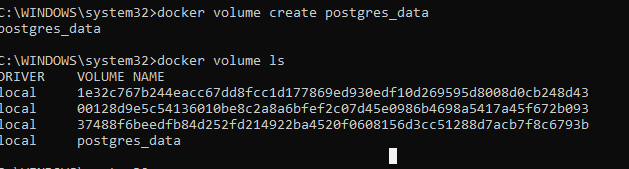


## **Часть 4: Сохранение данных с помощью Томов (Volumes)**

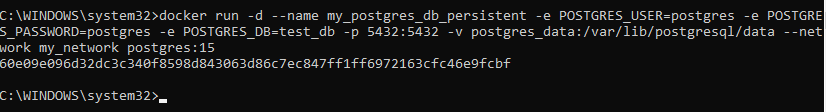
Остановка и удаление текущего контейнера с БД:



Создание тома для хранения данных БД:



**З**апуск нового контейнера с PostgreSQL с подключением тома:

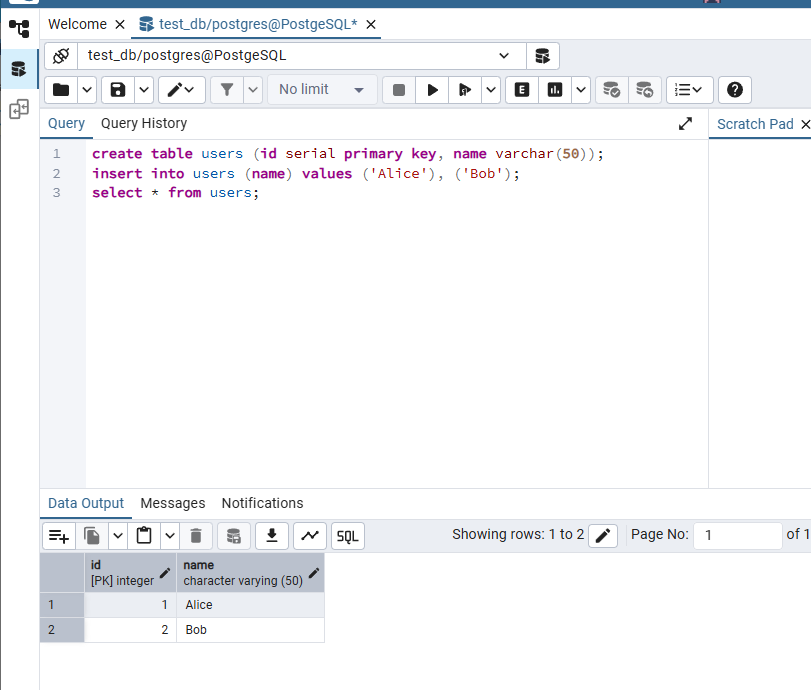


**-v postgres\_data:/var/lib/postgresql/data** — монтирует том **postgres\_data** в директорию внутри контейнера, где PostgreSQL хранит все свои данные.

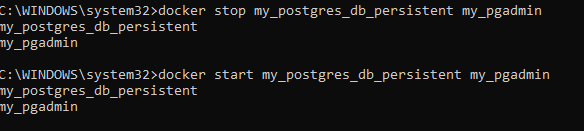
После пересоздания контейнера данные были утеряны:



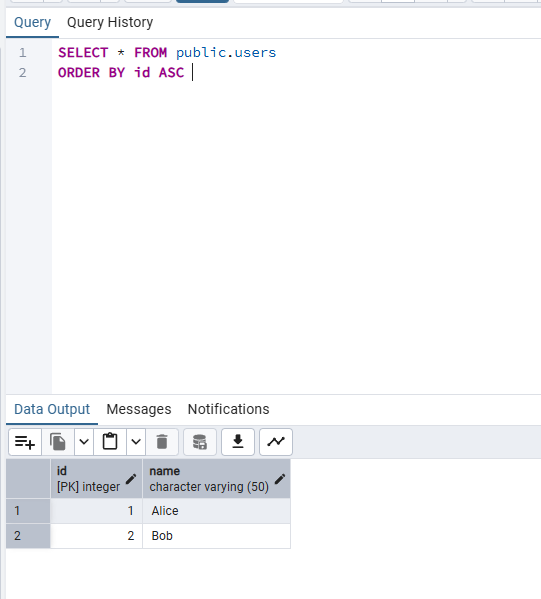
Добавление данных в таблицу



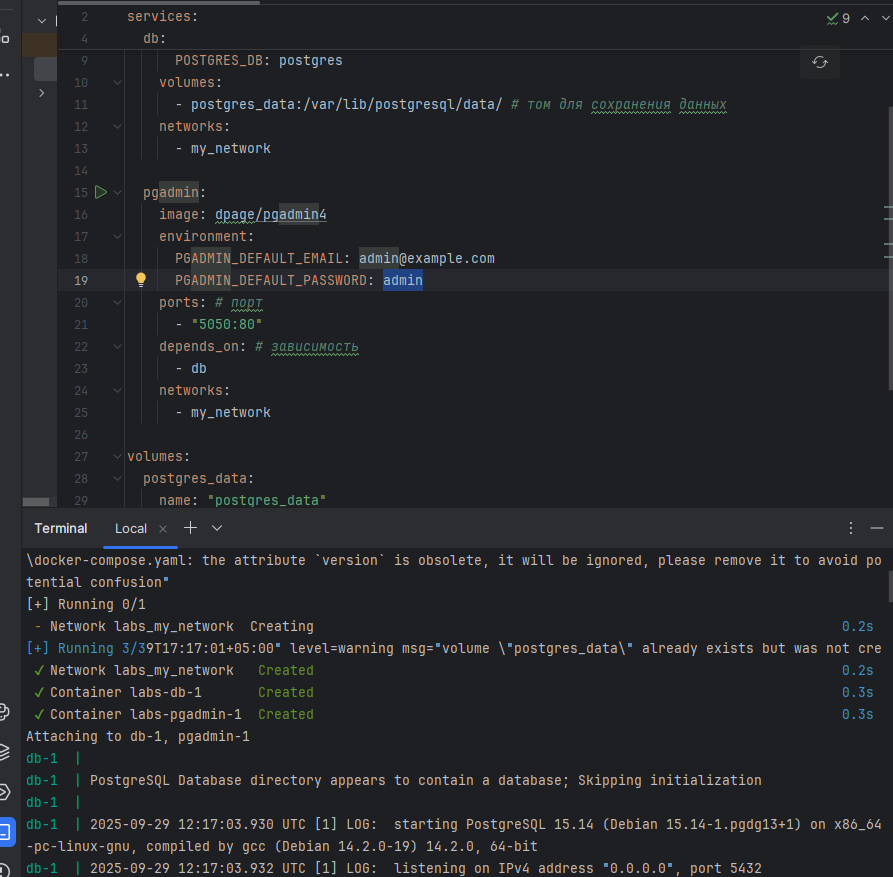
Остановка и запуск контейнеров

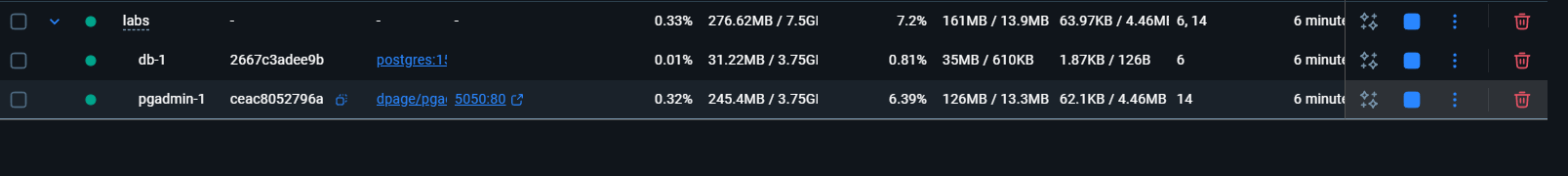


Проверка сохранности данных



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





<https://github.com/saschaschmygol/AppDevelop.git>

## Вопросы

1. Что такое docker? Docker – система контейнеризации. Это система для создания, доставки и запуска приложений внутри изолированных контейнеров.
2. Для чего нужны тома и сети docker? Тома (volumes) в Docker нужны для хранения данных вне контейнера, чтобы они сохранялись при перезапуске или удалении контейнера, обеспечивая персистентность
3. Как подключится к контейнеру и выполнить в нём команды? Через docker exec с ключами -it и указанием имени/id контейнера и команды.
4. Для чего нужен pgAdmin? Это графический инструмент для администрирования баз данных PostgreSQL, позволяющий управлять таблицами, запросами и пользователями через удобный интерфейс.