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

## Цель работы

Научиться работать с Docker на практике: уметь запускать контейнеры, управлять ими, работать с сетями и томами, и на конкретном примере развернуть PostgreSQL с возможностью подключения из внешнего инструмента (pgAdmin).

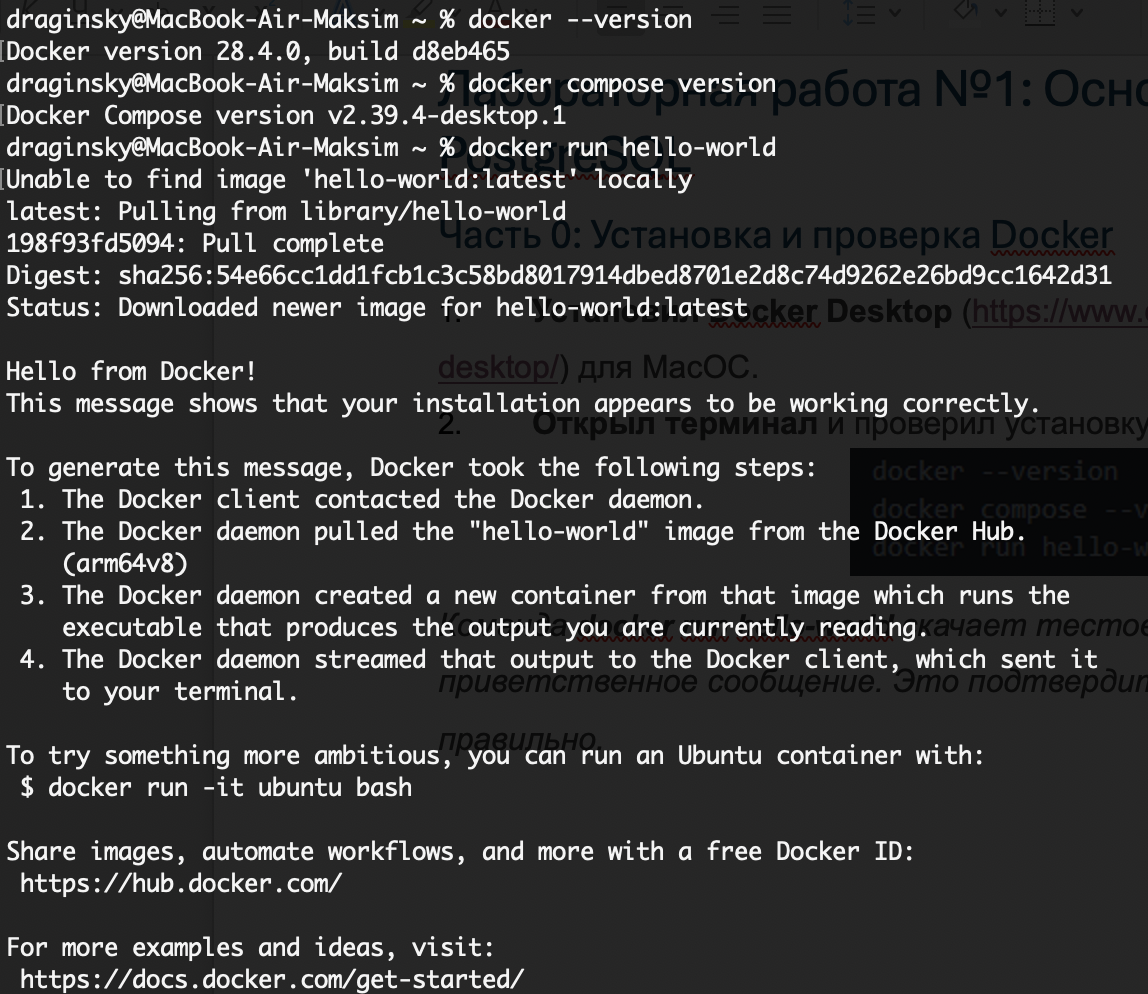
## Описание задачи

Нужно было установить Docker, разобраться с базовыми командами, запустить контейнер с PostgreSQL, подключиться к нему через pgAdmin, научиться сохранять данные с помощью томов, а в конце оформить всё это через docker-compose.

## Ход работы

### Часть 0: Установка и проверка Docker

1. **Установил Docker Desktop** (<https://www.docker.com/products/docker-desktop/>) для MacОС.
2. **Открыл терминал** и проверил установку:



Команда **docker run hello-world** скачала тестовый образ и запустила его, выведя приветственное сообщение.

Я убедился, что Docker работает правильно.

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

1. **Просмотрел информацию:**

**Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.**

1. **Запустил простой контейнер с Nginx:**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Проверил, что контейнер работает:** В браузере по [**http://localhost:8080**](http://localhost:8080)открылась стартовая страница Nginx.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

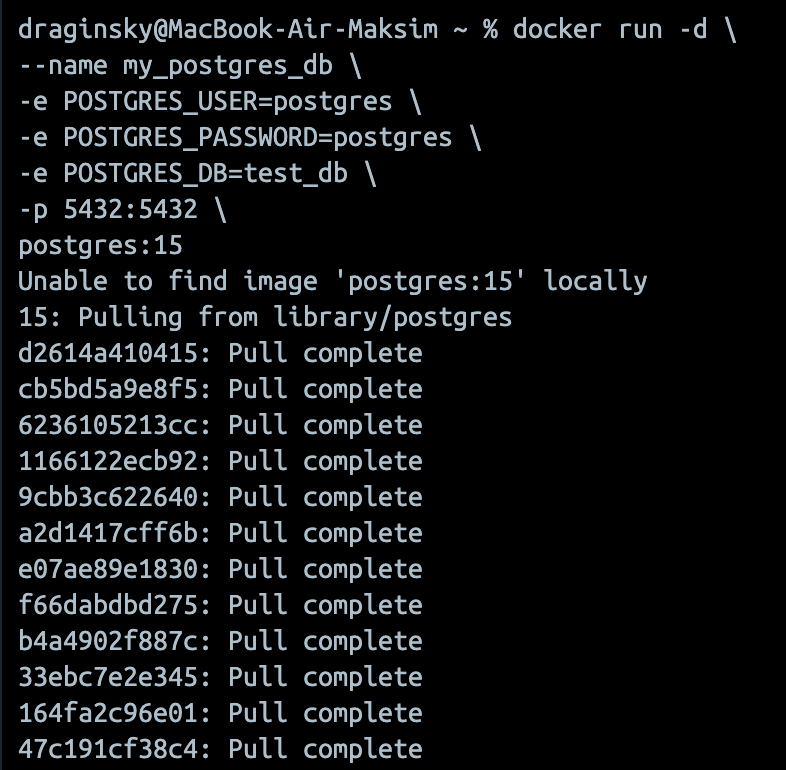
1. **Остановил и удалил контейнер:**

Изображение выглядит как текст, снимок экрана, Шрифт, черный

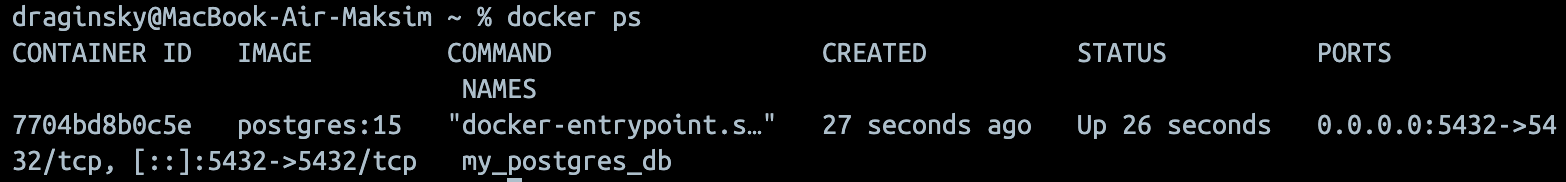
Контент, сгенерированный ИИ, может содержать ошибки.

### Часть 2: Запуск PostgreSQL в контейнере

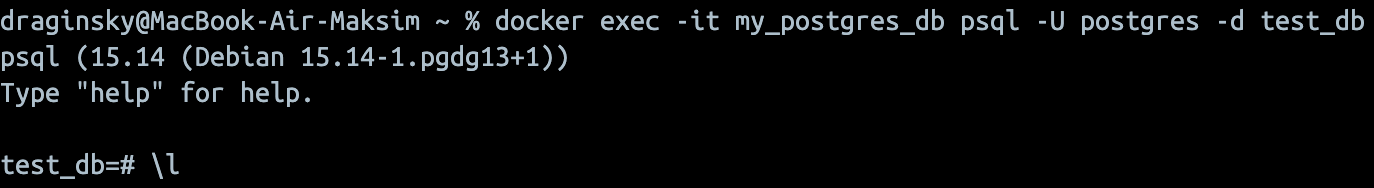
1. **Запустил контейнер с PostgreSQL:**



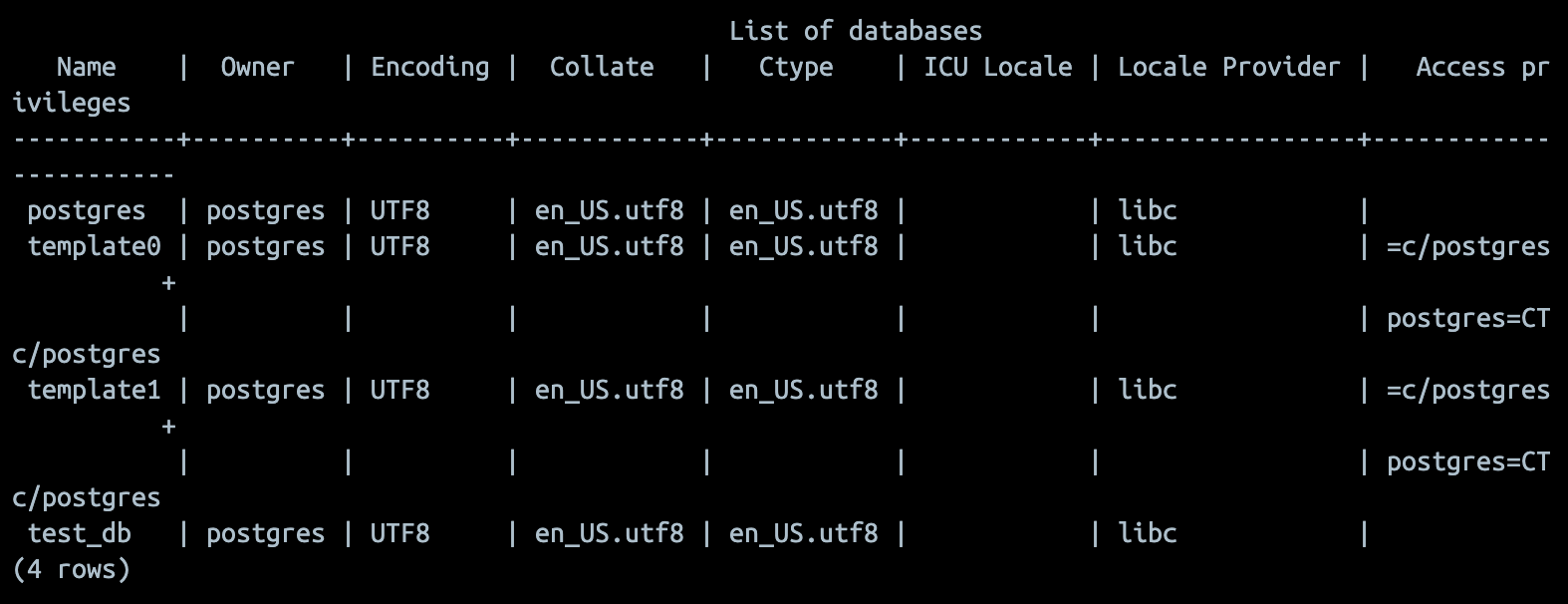
1. **Проверил, что контейнер запущен и слушает порт:**



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



Выполнил тестовые запросы. Список баз данных вывел следующие базы:



Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

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

1. **Создал сеть Docker:**  Чтобы контейнеры "увидели" друг друга по именам.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Подключил контейнер с PostgreSQL к сети:**



1. **Запустил pgAdmin в той же сети:**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.Изображение выглядит как текст, снимок экрана, Шрифт, черный

Контент, сгенерированный ИИ, может содержать ошибки.

4. Настроил подключение в pgAdmin:

* Открыл http://localhost:8080 в браузере.
* Войдите под admin@example.com / admin.
* Добавил новый сервер:

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки. Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Сохранил. Подключение прошло успешно.

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

Контент, сгенерированный ИИ, может содержать ошибки.

Через Query Tool в pgAdmin выполнил запрос:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Увидел таблицу users, созданную ранее через консоль, и данные в ней.

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

1. **Остановил и удалил текущий контейнер с БД:**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Создал том для хранения данных БД:**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Запустил новый контейнер с PostgreSQL, подключив том:**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Проверил сохранность данных:**

1. Через pgAdmin (**http://localhost:8080**) снова выполнил запрос **SELECT \* FROM users;**

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

Контент, сгенерированный ИИ, может содержать ошибки.

Таблицы с данными нет, добавим их:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

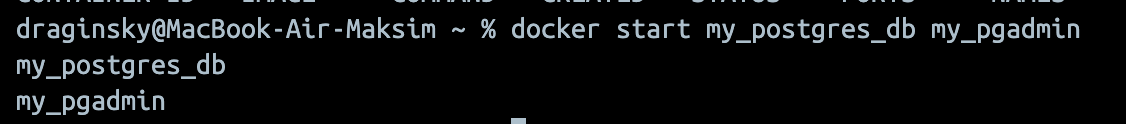
Контент, сгенерированный ИИ, может содержать ошибки.

1. Остановил контейнеры с БД и pgAdmin:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Запустил контейнеры:



1. Проверил таблицу users:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Таблица с данными есть.

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

После настройки pgAdmin и PostgreSQL настраил docker-compose.yaml. Ссылка на него лежит рядом в репозитории.

## Вопросы

**Что такое Docker?**  
Это такая штука, которая позволяет запускать программы в «коробочках» — контейнерах, которые не мешают остальной системе. В контейнере всё своё: система, программы, библиотеки.

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

* Тома — чтобы данные не терялись, если контейнер удалить.
* Сети — чтобы контейнеры могли «видеть» друг друга и общаться по именам.

**Как подключиться к контейнеру и выполнить в нём команды?**  
Через docker exec -it <имя\_контейнера> <команда>. Например, для PostgreSQL:

docker exec -it my\_postgres\_db psql -U postgres

**Для чего нужен pgAdmin?**  
Это веб-интерфейс для работы с PostgreSQL. Можно легко смотреть таблицы, выполнять запросы, создавать базы и пользователей, не дергаясь с командной строки.

## Вывод

В ходе работы я разобрался, как запускать контейнеры, работать с PostgreSQL внутри Docker, подключаться к БД через pgAdmin, использовать тома для сохранения данных и объединять контейнеры в одну сеть. Также научился оформлять конфигурацию через docker-compose — теперь можно быстро поднимать и выключать весь стек.