Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ  
ШПиАО Прикладной анализ данных

# ОТЧЕТ

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

# Преподаватель: Кузьмин Денис Иванович Обучающийся группы РИМ–150950: Поварнина Екатерина Дмитриевна

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.

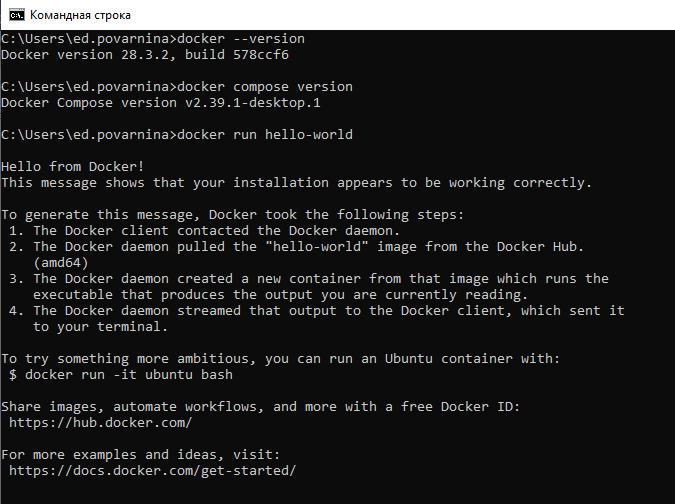
**Задачи:**

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.

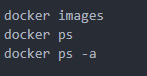
Ход работы

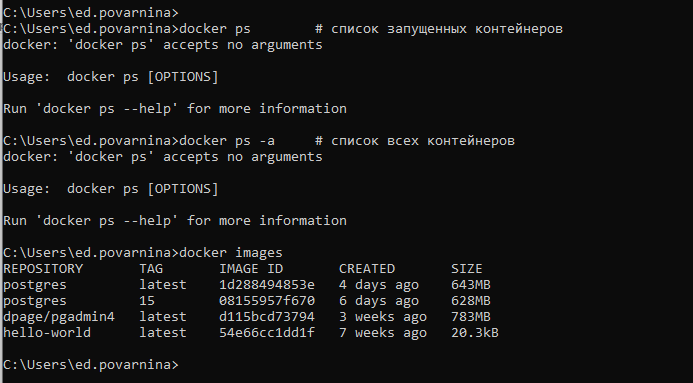
## Часть 0: Установка и проверка Docker

В рамках начального этапа работы был установлен Docker Desktop для Windows, проведена настройка docker compose в соответствии с официальной документацией, и выполнена проверка корректности установки через запуск тестового контейнера hello-world, который успешно скачал образ и вывел подтверждающее сообщение о работоспособности системы.



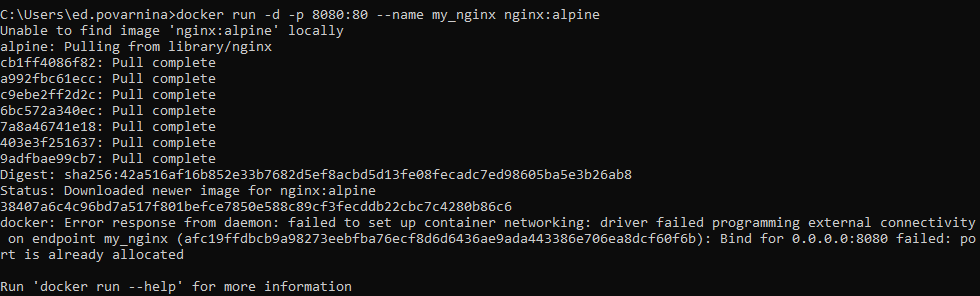
## Часть 1: Базовые команды Docker. Работа с образами и контейнерами

Просмотрела информацию с помощью команд  
****

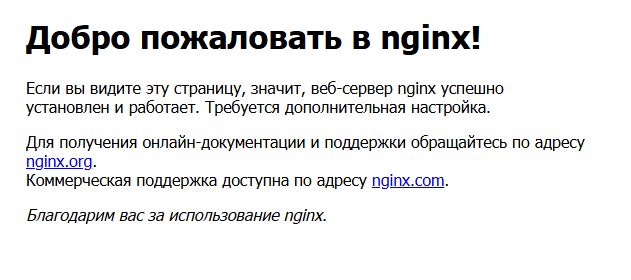
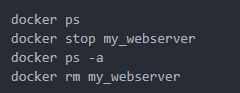
****

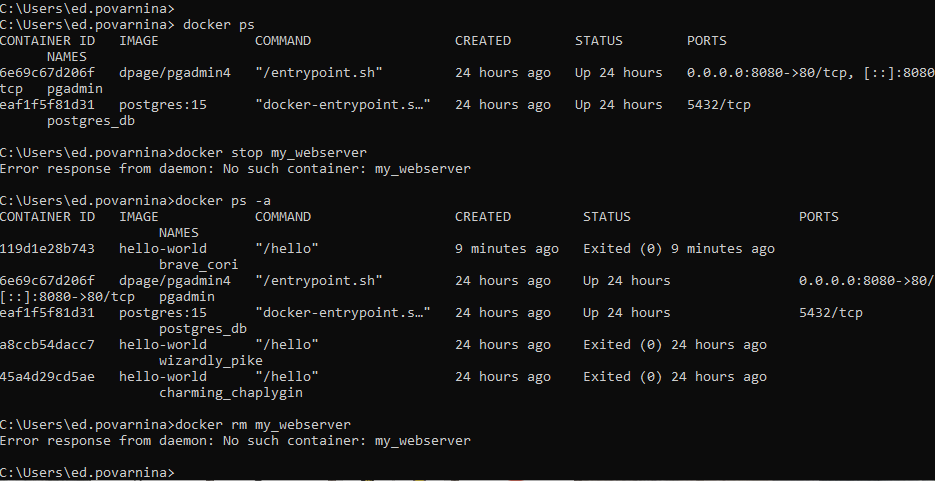
Запустила простой контейнер (на примере Nginx):

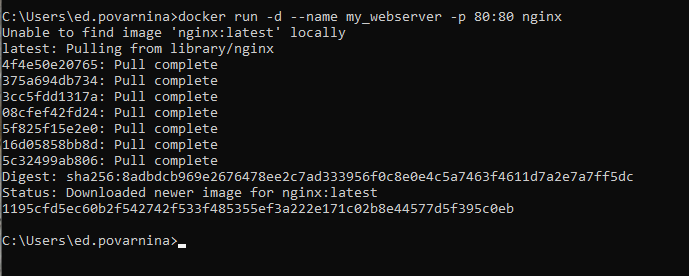


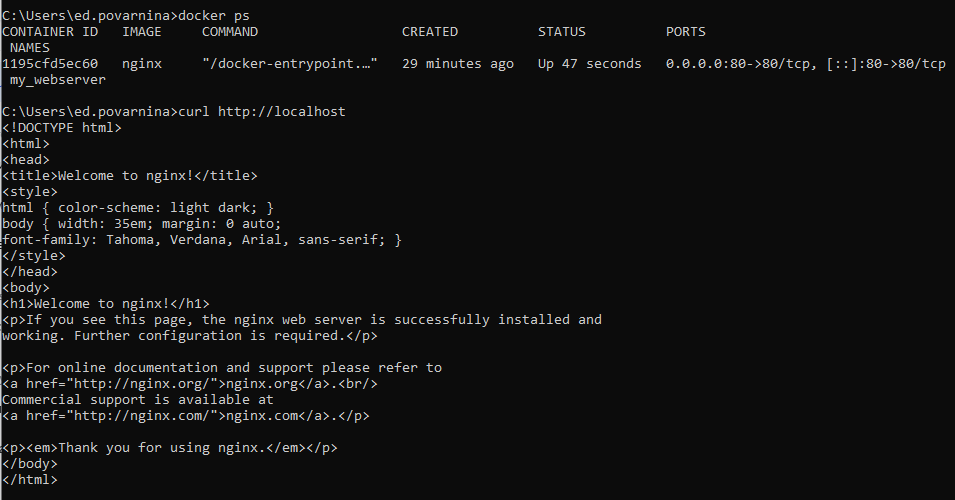


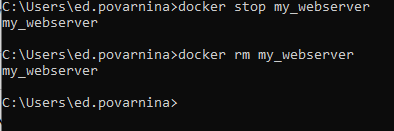
Проверила, что контейнер работает:

  
  
Далее остановила и удалила контейнер при помощи команд:  


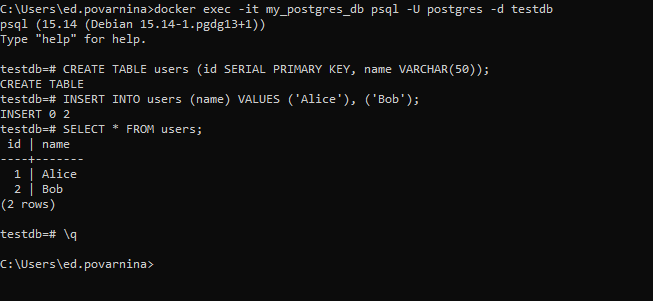
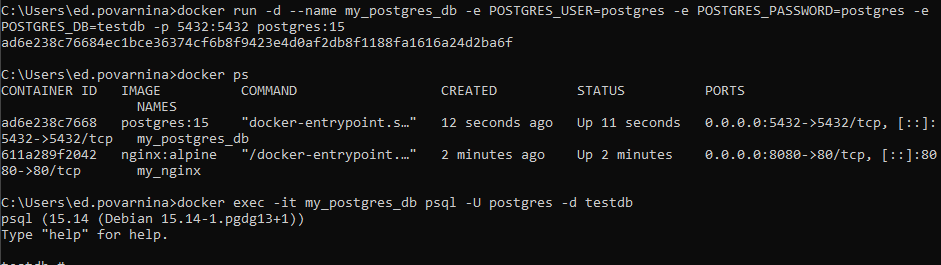








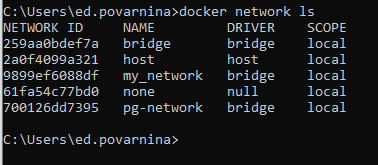
## Часть 2: Запуск PostgreSQL в контейнере

Запустили контейнер с PostgreSQL 15, указав необходимые переменные окружения для настройки СУБД (пользователь, пароль и база данных) и пробросив порт 5432 для внешнего подключения. Убедились в успешном запуске контейнера командой docker ps, после чего подключились к нему в интерактивном режиме через psql для проверки работоспособности и выполнения тестовых SQL-запросов.

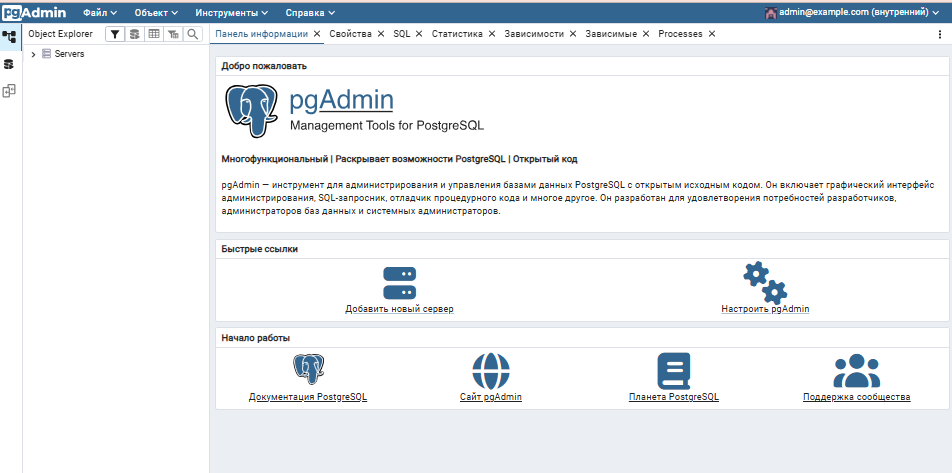
## Часть 3: Подключение к БД через pgAdmin из второго контейнера

Создали пользовательскую сеть Docker и запустили в ней контейнер с pgAdmin, подключив его к уже работающему контейнеру PostgreSQL через общую сеть. После запуска pgAdmin выполнили настройку подключения к базе данных через веб-интерфейс, указав имя контейнера PostgreSQL в качестве хоста, и успешно протестировали соединение, выполнив SQL-запросы к ранее созданной таблице users, что подтвердило корректную работу связки контейнеров через Docker-сеть.









**Добавьте новый сервер:**

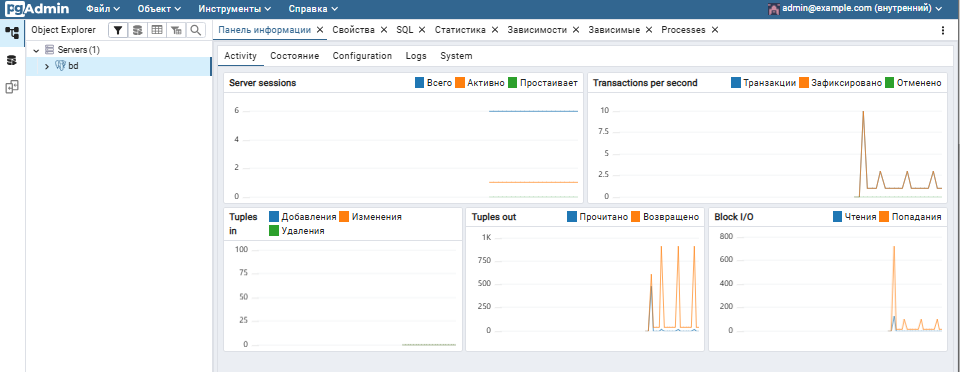
**General -> Name:** **Docker PostgreSQL** (bd/bd2)

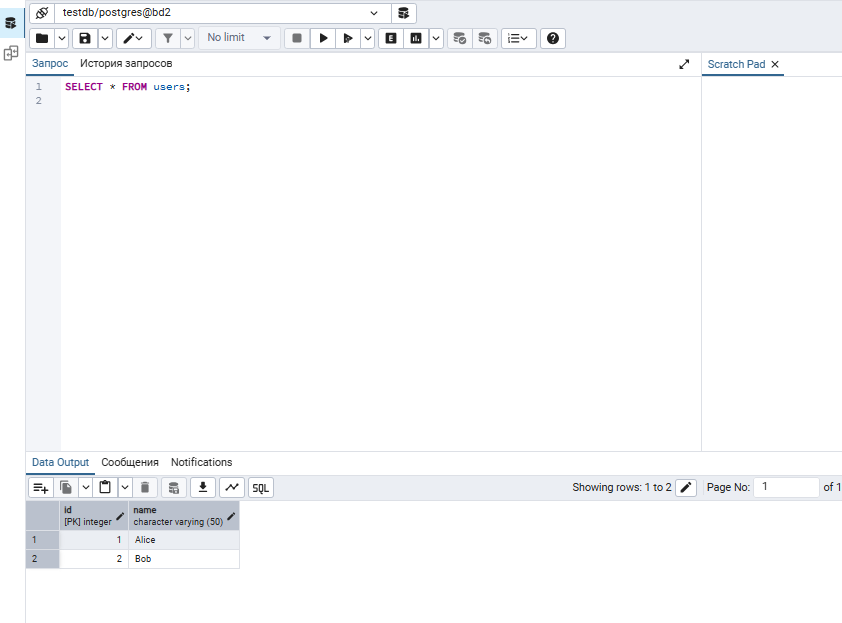
**Connection -> Host name/address:** **my\_postgres\_db** (имя контейнера testbd)

**Connection -> Username:** **postgres**

**Connection -> Password:** **postgres**

Подключение успешно.

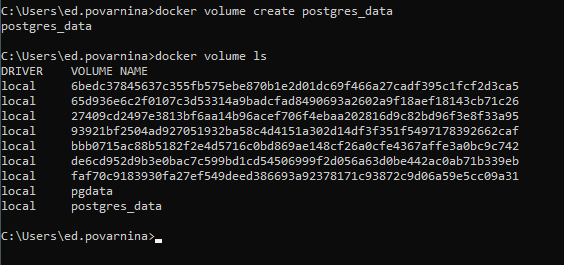
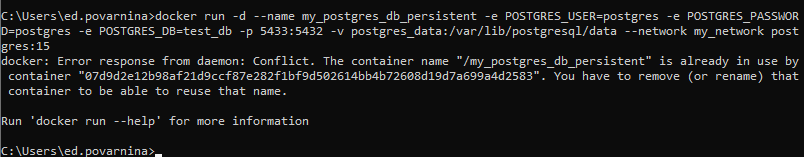
**Через Query Tool в pgAdmin выполните запрос:**

****

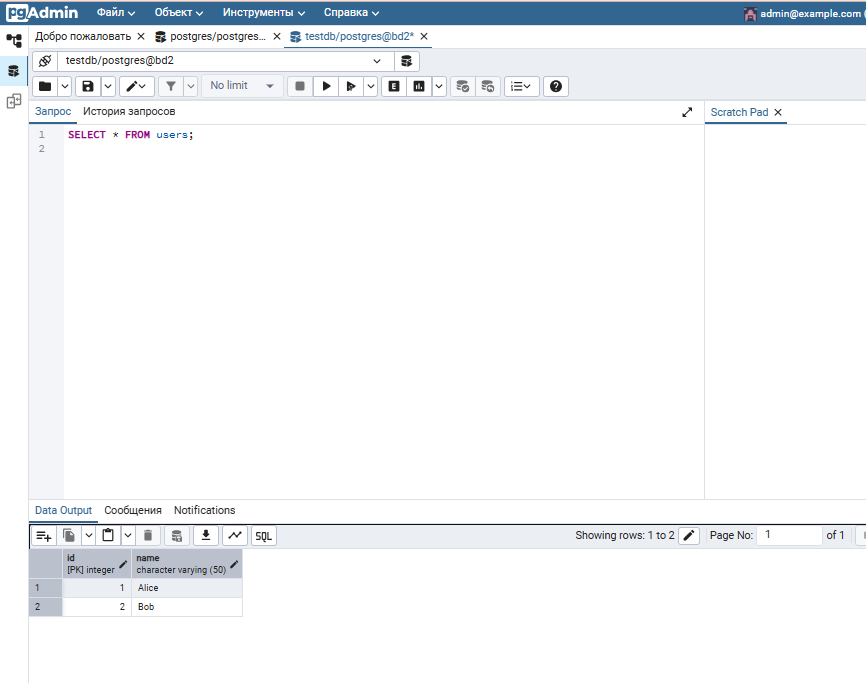
*Видем таблицу****users****, созданную ранее через консоль, и данные в ней.*

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

Для обеспечения сохранности данных создали том Docker и перезапустили контейнер PostgreSQL с примонтированным томом к директории хранения данных СУБД. После добавления тестовых данных через pgAdmin провели цикл остановки и повторного запуска контейнеров, в ходе которого убедились, что все созданные таблицы и записи сохраняются после перезапуска, что подтвердило корректную работу механизма томов для постоянного хранения данных.

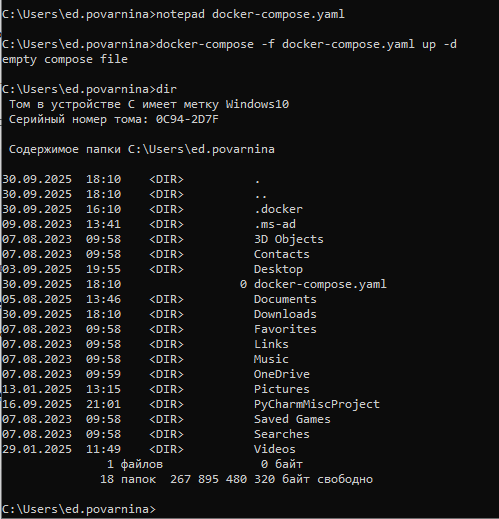
   
использовала свободный хост -p 5433:5432

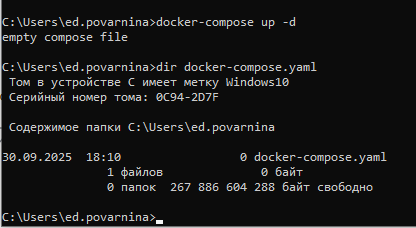




## Часть 5: Перенос конфигурации контейнеров в docker-compose.yaml







## Вопросы

1. Что такое docker

**Docker** - это платформа для разработки, доставки и запуска приложений в **контейнерах**. Контейнеры представляют собой изолированные среды, которые содержат все необходимое для работы приложения

1. Для чего нужны тома и сети docker?

Тома (Volumes) в Docker используются для сохранения данных после удаления контейнеров. Они обеспечивают сохраняемость данных, обмен данными между контейнерами, резервное копирование и миграцию данных, разделение данных и кода приложения

Сети (Networks) обеспечивают изоляцию групп контейнеров, безопасное взаимодействие между контейнерами, возможность обращения к контейнерам по именам, контроль доступа к сетевым ресурсам

1. Как подключится к контейнеру и выполнить в нём команды?

docker exec имя\_контейнера команда - выполнение одной команды

docker exec -it имя\_контейнера /bin/bash - интерактивное подключение через bash

docker exec -it имя\_контейнера /bin/sh - подключение через shell

1. Для чего нужен pgAdmiт

pgAdmin - это веб-интерфейс для управления базами данных PostgreSQL, который предоставляет:

**- Визуальное управление**    
**-Выполнение SQL-запросов**    
**-Мониторинг производительности**  
**-Управление пользователями**    
**-Резервное копирование**    
**-Отладка**

Вывод: В результате выполнения лабораторной работы были успешно освоены ключевые аспекты работы с Docker: управление контейнерами, создание сетей и использование томов. Практическая часть включала развертывание связки PostgreSQL и pgAdmin в изолированных контейнерах, их взаимодействие через внутреннюю сеть и обеспечение сохранности данных. Освоенные навыки позволяют уверенно применять Docker для изолированного развертывания приложений.  
  
Ссылка на гит <https://github.com/pokatyusha/bd25>