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

**Студент:** Яськов Антоний Михайлович

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

**Цель работы**

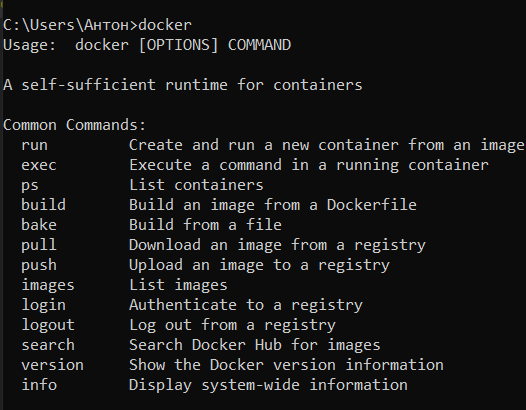
Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

**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. Для удобства конфигурация была вынесена в docker-compose файл.

# **1.Установить и проверить работу Docker**

Скачал Docker desktop. Проверил с помощью команды docker в консоль



# **2.Изучить базовые команды Docker**

Изучил список команд со скриншота выше. Выделю основные.

Для взаимодействия с образами:

docker pull {image} - Скачать образ  
docker push {image} - Отправить образ в registry  
docker build -t {image-name} -f {path-to-Dockerfile} - собрать образ  
docker images - Список образов

Для взаимодействия с контейнерами:

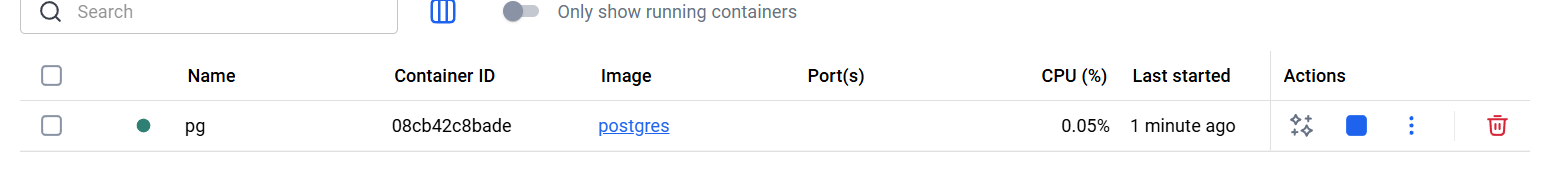
docker run {image} - Запустить контейнер с образом   
docker ps - Список контейнеров  
docker stop {container} - Остановить контейнер  
docker start {container} - Запустить остановленный контейнер  
docker restart {container} - Перезапустить контейнер  
docker rm {container} - Удалить остановленный контейнер

# **3.Запустить контейнер с PostgreSQL в изолированном режиме**

Запускаю с помощью команды

docker run -d --name pg -e POSTGRES\_USER=postgres -e POSTGRES\_PASSWORD=password -e POSTGRES\_DB=softDev postgres

Проверяю, что у него не открыты порты в Docker desktop



# **4.Запустить контейнер с pgAdmin и подключить его к контейнеру с БД через сеть Docker**

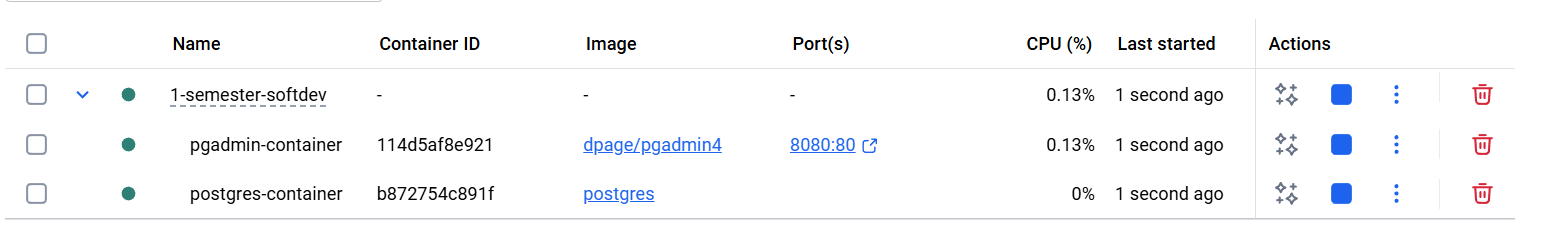
Можно отдельно всё создавать по частям, но я воспользуюсь docker-compose, в котором опишу всю систему сразу.

docker-compose services:  
 postgres:  
 image: postgres  
 container\_name: postgres-container  
 environment:  
 POSTGRES\_USER: postgres  
 POSTGRES\_PASSWORD: password  
 POSTGRES\_DB: softDev  
 networks:  
 - pg-network  
  
 pgadmin:  
 image: dpage/pgadmin4  
 container\_name: pgadmin-container  
 environment:  
 PGADMIN\_DEFAULT\_EMAIL: [p@p.p](mailto:p@p.p) PGADMIN\_DEFAULT\_PASSWORD: password  
 ports:  
 - "8080:80"  
 networks:  
 - pg-network  
 depends\_on:  
 - postgres  
  
networks:  
 pg-network:  
 driver: bridge

Запускаю docker-compose командой

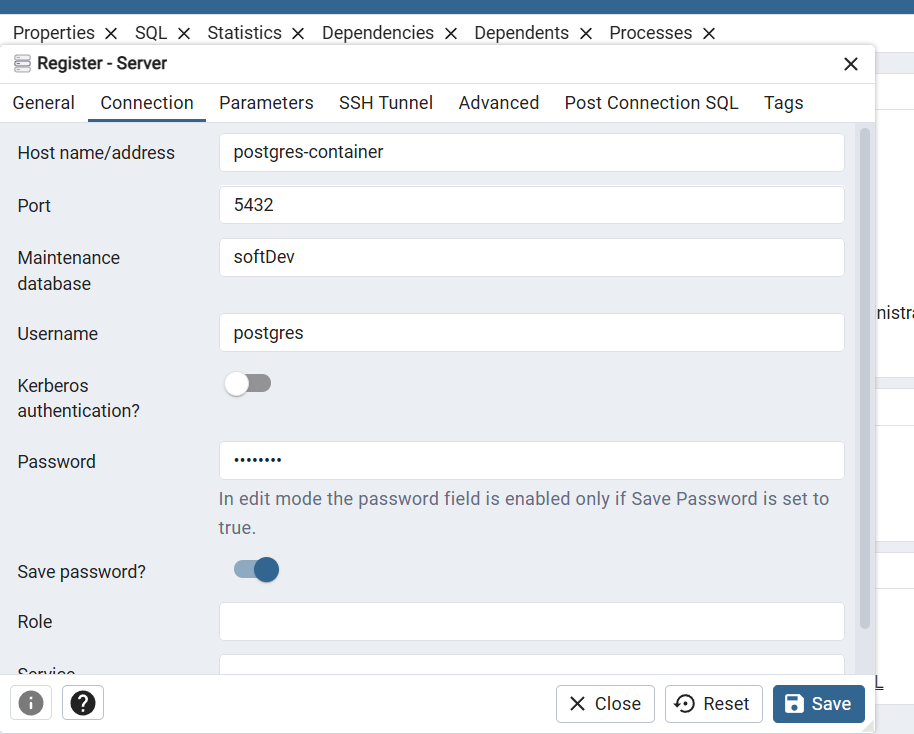
docker compose -f docker-compose\_4.yaml up

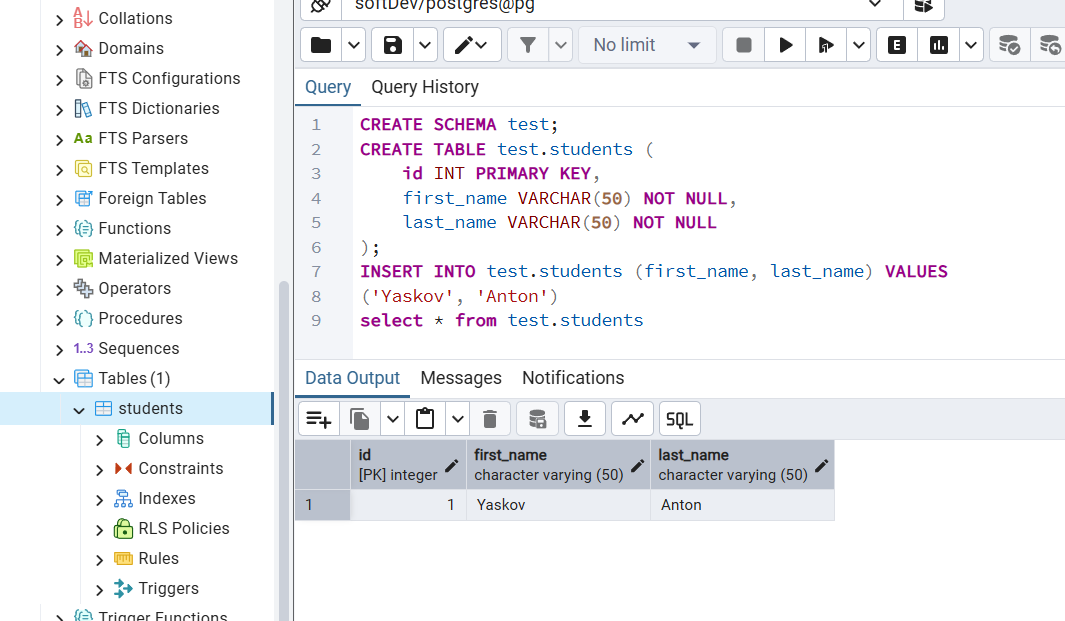
Проверяю работоспособность



# **5. Подключиться к БД из pgAdmin, создать схему и выполнить запросы**

Захожу в pgAdmin по его порту 8080, вхожу по данным которые указал в docker-compose.





Регистрирую новый сервер со следующими параметрами подключения:

Создаю схему, выполняю запросы

# **6. Обеспечить сохранность данных БД с помощью томов Docker.**

Для этого нужно добавить volumes в docker-compose.

Также, добавлю файл .env и вынесу туда переменные окружения. В файле буду использовать их в виде ${ENV\_VAR}

Итоговый docker-compose с volumes services:  
 postgres:  
 image: postgres  
 container\_name: postgres-container  
 environment:  
 POSTGRES\_USER: $POSTGRES\_USER  
 POSTGRES\_PASSWORD: $POSTGRES\_PASSWORD  
 POSTGRES\_DB: $POSTGRES\_DB  
 networks:  
 - pg-network  
 volumes:  
 - postgres\_data:/var/lib/postgresql/data <-- Нужно добавить сюда  
  
 pgadmin:  
 image: dpage/pgadmin4  
 container\_name: pgadmin-container  
 environment:  
 PGADMIN\_DEFAULT\_EMAIL: $PGADMIN\_DEFAULT\_EMAIL  
 PGADMIN\_DEFAULT\_PASSWORD: $PGADMIN\_DEFAULT\_PASSWORD  
 ports:  
 - "8080:80"  
 networks:  
 - pg-network  
 depends\_on:  
 - postgres  
  
networks:  
 pg-network:  
 driver: bridge  
  
volumes:   
 postgres\_data: <-- И сюда

**Вопросы**

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

Тома нужны для сохранности данных контейнеров.   
Сети нужны для связи контейнеров между собой.

1. **Как подключиться к контейнеру и выполнить в нём команды?**  
   Через docker exec -it {имя\_контейнера} {команда}
2. **Для чего нужен pgAdmin?**  
   Это интерфейс для работы с PostgreSQL. Используется для визуализации взаимодействия с БД, без использования терминала.

**Вывод**

В ходе данной лабораторной работы были успешно освоены базовые принципы работы с Docker. В практической части был развернут контейнер с сервером PostgreSQL и веб-интерфейс для управления им — pgAdmin. Контейнеры были связаны с помощью сетей Docker. Также, их данные были сохранены с помощью Volumes. Итоговая конфигурация развёртывания была записана в docker-compose файл для автоматизации процесса.