Правительство Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСТИТЕТ

Кафедра компьютерных интеллектуальных технологий

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6 ПО ДИСЦИПЛИНЕ

«СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ»

ТЕМА:

«РАЗВЁРТЫВАНИЕ СУБД POSTGRES С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ АВТОМАТИЗАЦИИ РАЗВЁРТЫВАНИЯ И УПРАВЛЕНИЯ ПРИЛОЖЕНИЯМИ»

Выполнил:

студент группы змИИВТ-231

Образцов Михаил Алексеевич

Подпись:\_\_\_\_\_\_\_\_

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата: \_\_\_\_\_\_\_\_ Подпись:\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

1. Содержание………………………………………………………………….2
2. Введение……………………………………………………………………..3
3. Установка приложения для работы с Docker-контейнерами…………….4
4. Установка программного средства для работы с СУБД DBeaver….…….5
5. Скачивание docker образа для работы с PostgreSQL…………..….………6
6. Создание Dockerfile……………………………….………………………...7
7. Создание образа на основе Dockerfile ……………………………………...8
8. Запуск контейнера для развертывания СУБД Postgres…………………….9
9. Подключение к работающему контейнеру и запуск интерфейса PSQL…10
10. Создание контейнера с томом Volume………………………………………11
11. Создание контейнера с использованием файла Docker-compose.yml……..12
12. Заключение…………………………………………………………………….13
13. Контрольные вопросы и ответы……………………………………………….14

**ВВЕДЕНИЕ**

**Цель лабораторной работы** - изучить основы работы программного средства для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker на примере решения задачи развёртывания СУБД Postgres.

***Основные задачи:***

* установка приложения для работы с Docker-контейнерами;
* установка программного средства для работы с СУБД DBeaver;
* скачивание docker-образа с СУБД postgres;
* создание Dockerfile;
* создание образа на основе Dockerfile;
* запуск контейнера для развёртывания СУБД Postgres;
* подключение к работающему контейнеру и запуск интерфейса psql;
* создание контейнера с томом (volume);
* создание контейнера с использованием файла docker-compose.yml

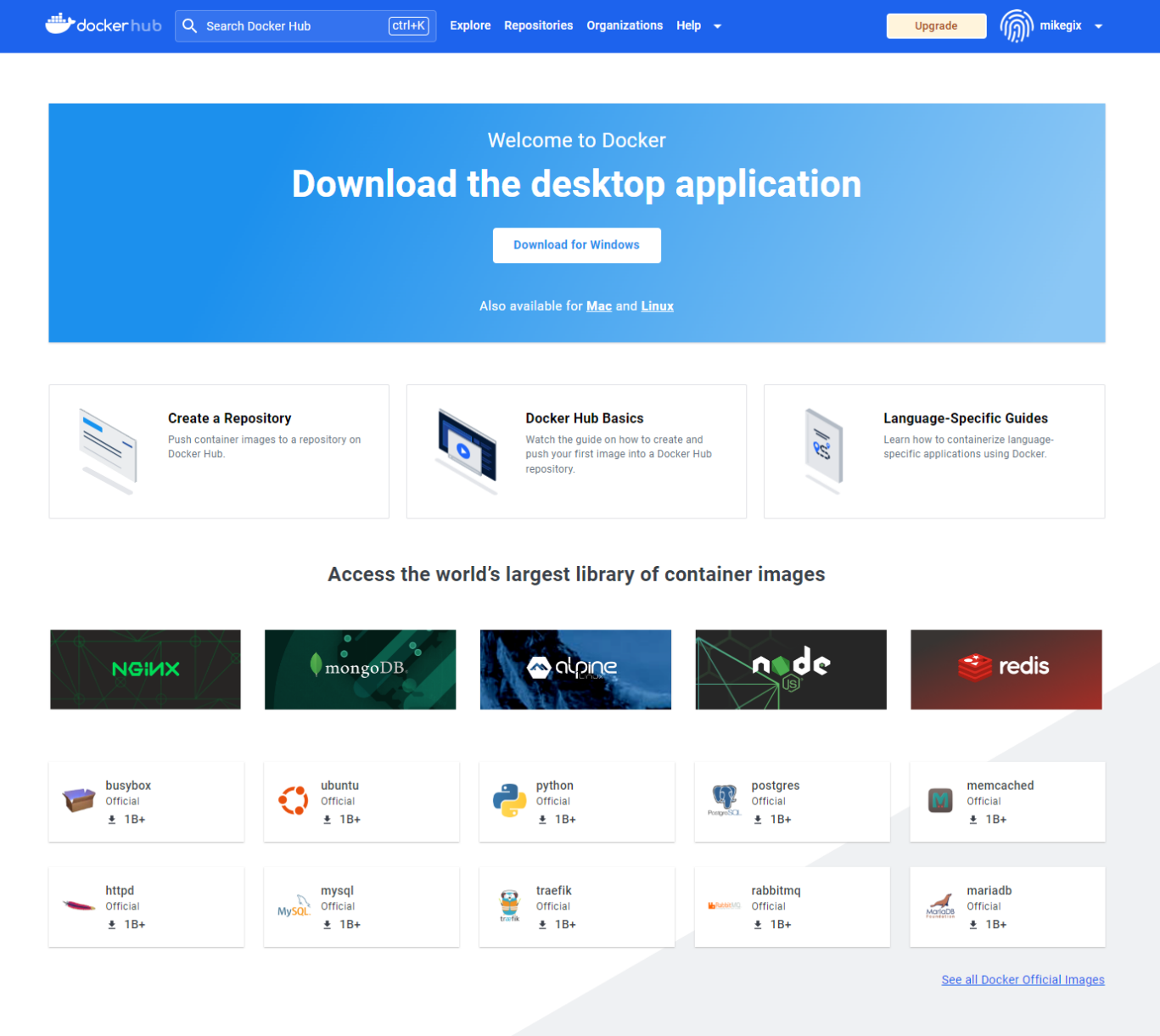
На протяжении выполнения данной лабораторной работы мною использовалась **ОС Windows 11**

Благодаря данной лабораторной работе можно усвоить много полезных навыков, таких как:

* **Навыки работы с Docker Desktop**
* **Навыки работы с DBeaver**
* **Развертывание PostgreSQL**

**УСТАНОВКА ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С DOCKER-КОНТЕЙНЕРАМИ**

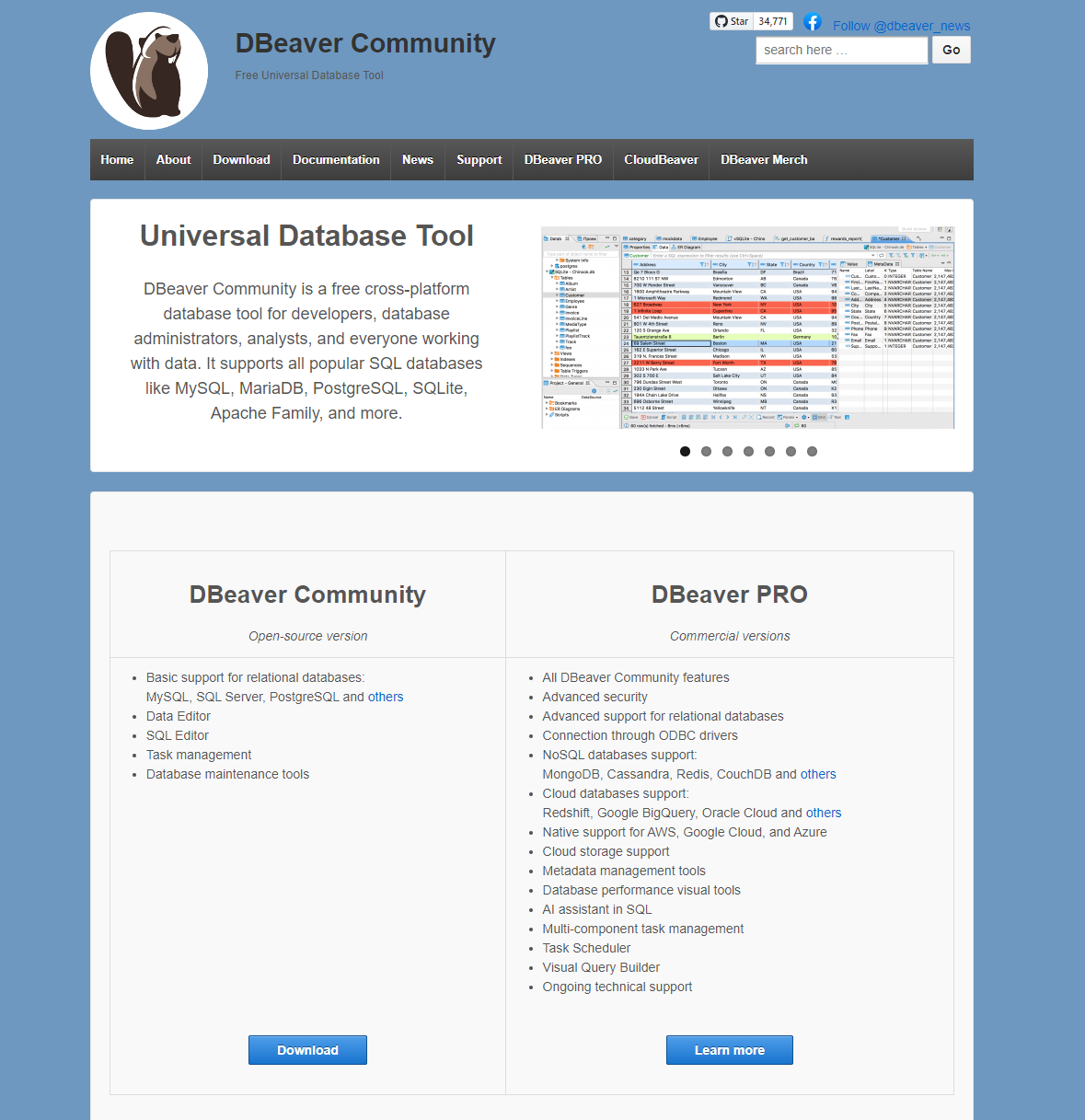
Для начала выполнения работы нам требуется скачать **Docker Desktop** c официального сайта <https://www.docker.com>



После скачивания - устанавливаем следуя инструкциям установщика.

**УСТАНОВКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ РАБОТЫ С СУБД DBEAVER**

Следующим шагом является скачивание **СУБД DBeaver** с сайта <https://dbeaver.io>



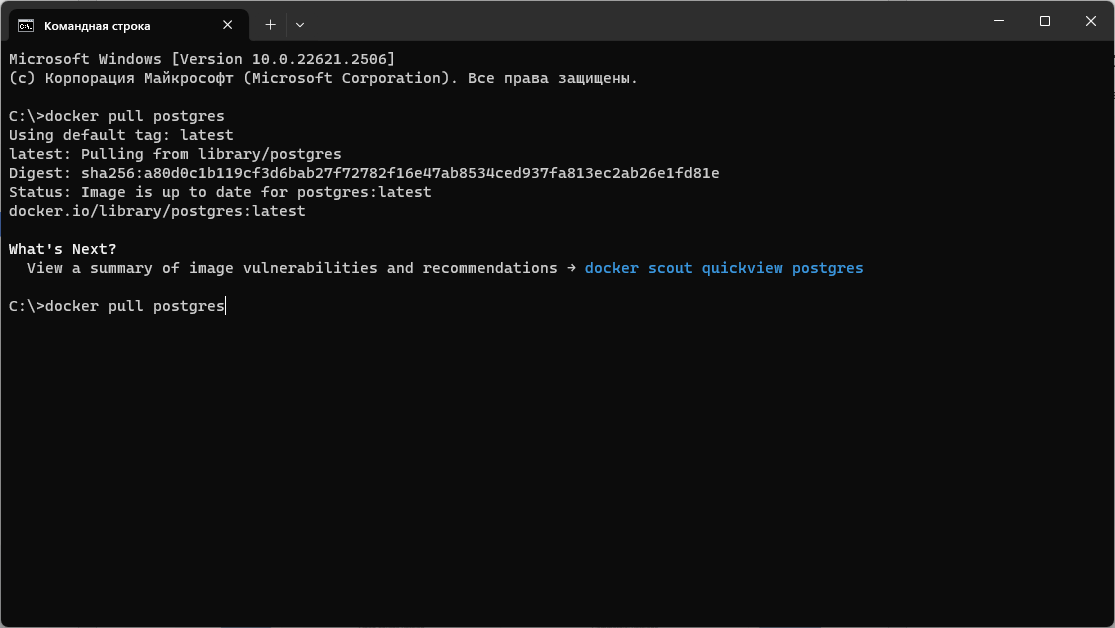
После скачивания - устанавливаем следуя инструкциям установщика.

**В процессе выполнения работы DockerDesktop и DBeaver должны быть запущены**

**СКАЧИВАНИЕ DOCKER-ОБРАЗА С СУБД POSTGRES**

Чтобы скачать образ через терминал необходимо ввести команду «**docker pull postgres**».

**Изображена консоль после ввода команды docker pull postgres**



**СОЗДАНИЕ DOCKERFILE**

Для начала требуется создать каталог(папку) в которой мы создаем файл без расширения с именем Dockerfile.

Далее открываем его c помощью notepad или другого текстового редактора и вводим следующие данные:

**FROM postgres:latest - из последней версии postgres**

**ENV POSTGRES\_PASSWORD=dbpass - Данные для подключения к базе**

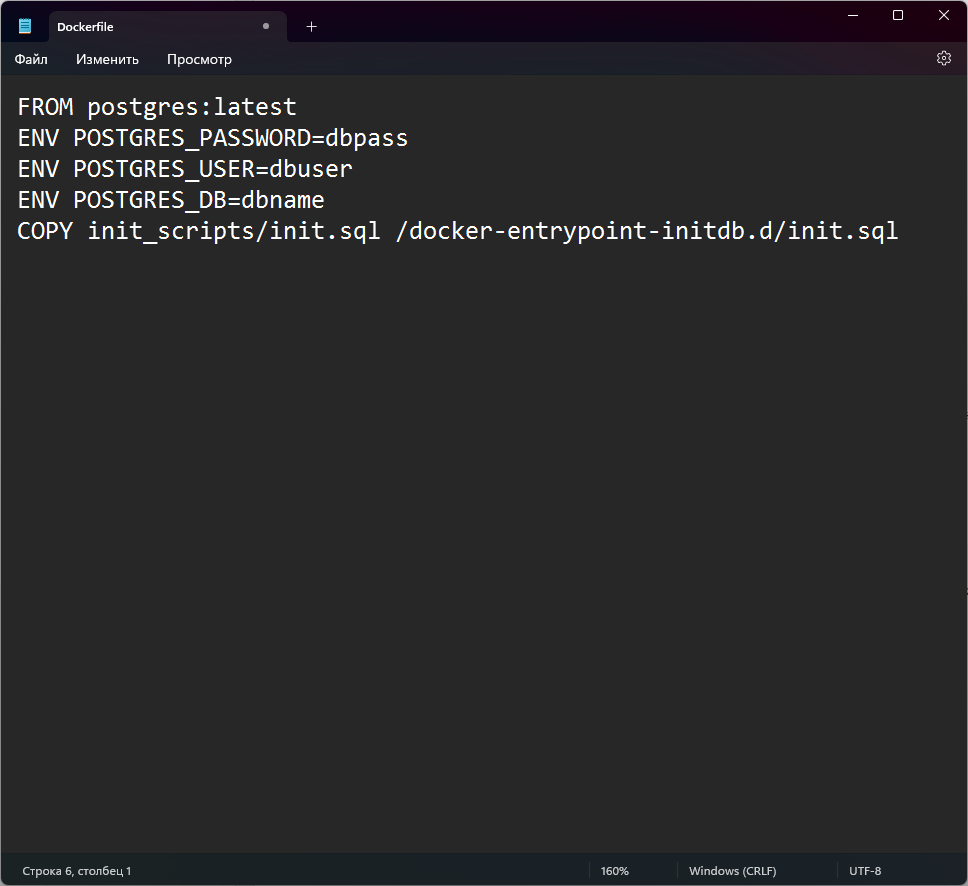
**ENV POSTGRES\_USER=dbuser - Данные для подключения к базе**

**ENV POSTGRES\_DB=dbname - Данные для подключения к базе**

**COPY init\_scripts/init.sql /docker-entrypoint-initdb.d/init.sql**

**- Эта строка копирует файл init.sql из локальной файловой системы внутрь контейнера.**

Далее сохраняем и закрываем.



**СОЗДАНИЕ ОБРАЗА НА ОСНОВЕ DOCKERFILE**

Создать образ с помощью команды «**docker build -t d\_img:latest .**»

(не потеряйте точку в конце команды)

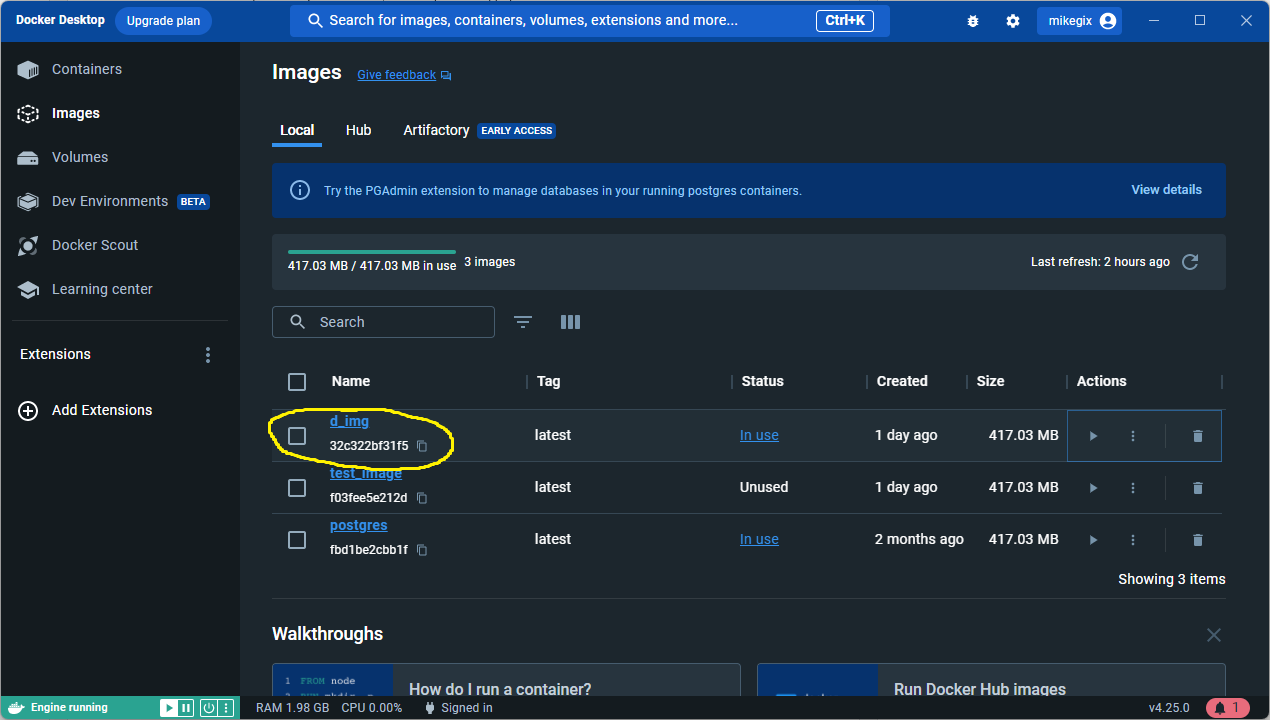
* Команду надо запускать из каталога, в котором находится Dockerfile

В моем случае C:\Docker

* В результате запустится Dockerfile и будет создан docker-образ «d\_img»

Созданный образ можно увидеть в приложении Docker Desktop

**На изображении виден ранее созданный образ**

****

**ЗАПУСК КОНТЕЙНЕРА ДЛЯ РАЗВЁРТЫВАНИЯ СУБД POSTGRES**

. **Создание контейнера**

Для создания docker-контейнера необходимо ввести в консоли команду:

«**docker run --rm --name test -e POSTGRES\_PASSWORD=test –p 5432:5432 -v /data:/var/lib/postgresql/data -d postgres** »

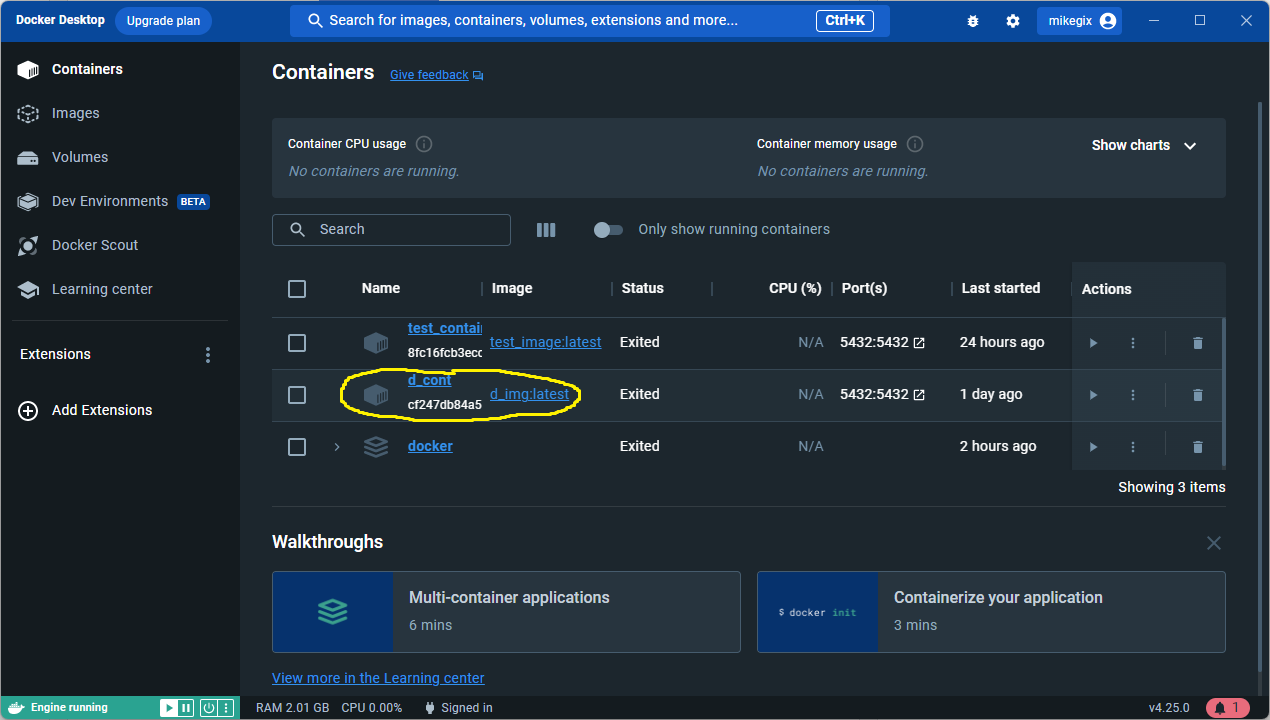
Для создания контейнера без volume используем команду:

«**docker run --rm --name test -e POSTGRES\_PASSWORD=test –p 5432:5432 -d postgres**»

Если не установлен docker-образ (docker-image) для postgres, то после запуска команды он скачается (pull).

После создания контейнера в терминале будет указан номер контейнера (длинный шестнадцатеричный). И в Docker-Desktop созданный контейнер с именем test тоже отобразится.

Чтобы воспользоваться созданной СУБД необходимо запустить контейнер с именем «**d\_cont**» на основе образа «**d\_img:latest**» с помощью команды: **«docker run -d -p 5432:5432 --name d\_cont d\_img:latest».**

**На изображении выделен созданный контейнер**

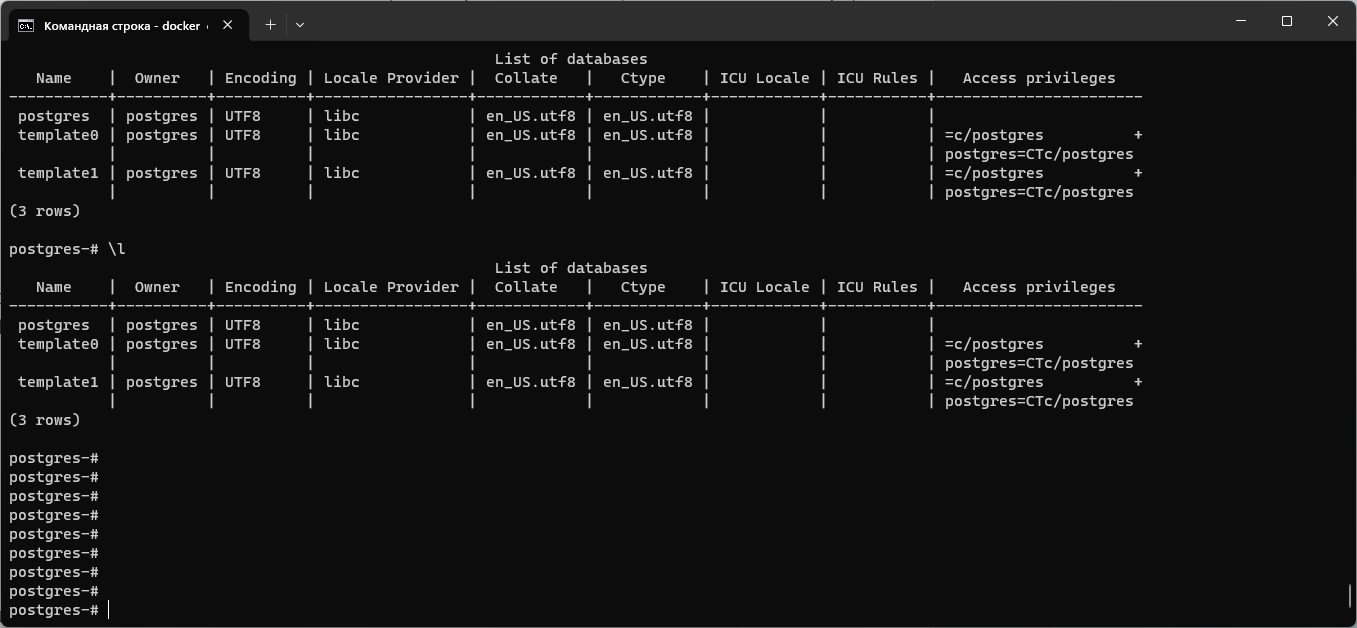
**ПОДКЛЮЧЕНИЕ К РАБОТАЮЩЕМУ КОНТЕЙНЕРУ И ЗАПУСК ИНТЕРФЕЙСА PSQL**

**Подключение к контейнеру**

Команда «***docker exec -it test psql -U user -d database***» - запуск контейнера в интерактивной среде с интерпретатором СУБД postgres под пользователем с именем «postgres»

Подробнее:

* docker exec – подключение к работающему контейнеру.
* флаг -it – для того чтобы работала интерактивная среда прямо в терминале
* test – название контейнера
* psql – команда, которая говорит, что в нашей интерактивной среде (в командной строке) нужно запустить интерпретатор postgres
* -U – флаг указания имени пользователя (user)$
* -d – флаг указания наименования базы данных.

****После входа в интерактивную среду вводим команду «\l» и получаем в терминале список баз данных.

**СОЗДАНИЕ КОНТЕЙНЕРА С ТОМОМ (VOLUME)**

Подключаем контейнер с томом (volume)

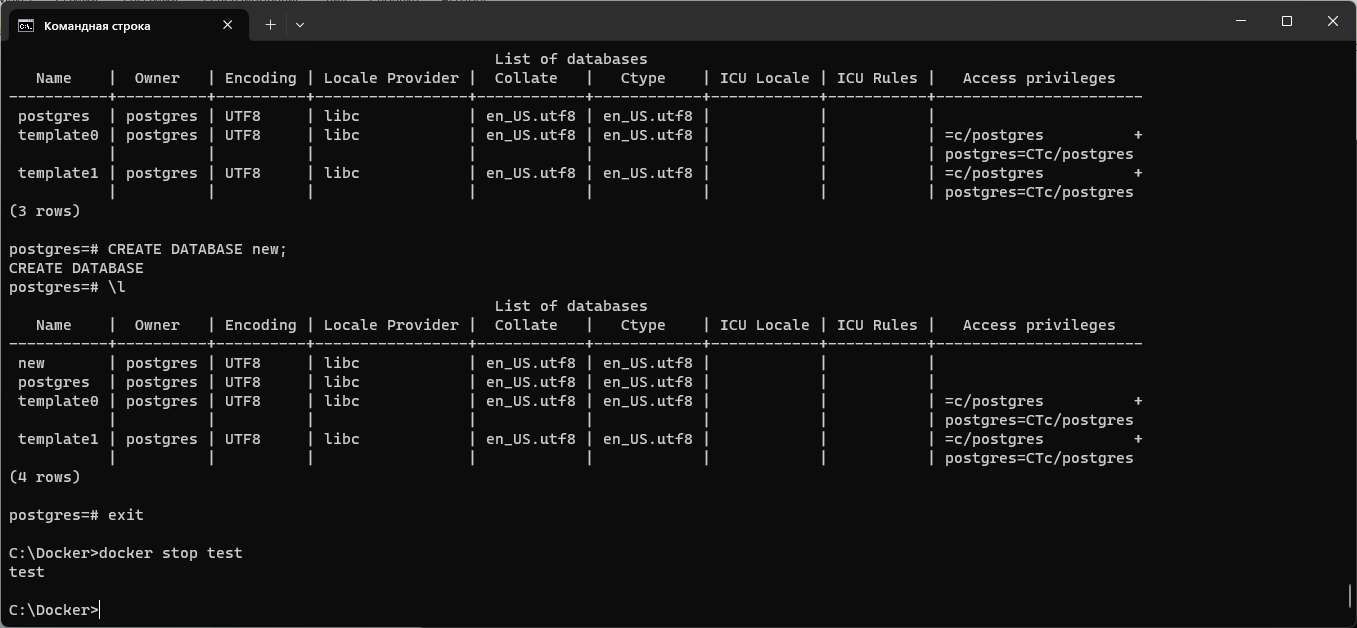
Для создания docker-контейнера необходимо ввести в командной строке (cmd) команду:

«**docker run --rm --name test -e POSTGRES\_PASSWORD=test –p 5432:5432 -v /data:/var/lib/postgresql/data -d postgres**».

*Примечание:*

1. *-v /data:/var/lib/postgresql/data – это как раз место для volume*
2. *Итак, volume подключается с помощью флага -v, за которым следую две папки (два места хранения) через двоеточие (/data:/var/lib/postgresql/data):*

* */data – это папка, которая находится в нашей операционной системе; можно указывать любую папку на компьютере;*
* */var/lib/postgresql/data – это папка, которая находится внутри операционной системы контейнера; эта папка постоянная, она не меняется.*
* *Между этими папками устанавливается синхронизация. То есть новые данные будут сохраняться в папку на компьютере (в нашем случае /data).*

*****Теперь если мы создадим еще одну базу командой CREATE DATABASE new; и далее exit и docker stop test и заного подключимся, то изменения сохранятся, мы увидим ранее созданную базу new*

**СОЗДАНИЕ КОНТЕЙНЕРА С ИСПОЛЬЗОВАНИЕМ ФАЙЛА DOCKER-COMPOSE.YML**

**Освоение Docker-compose**

**Docker Compose** — это инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов.

**Dockerfile** – способ описать логику работы одного контейнера.

**Docker-compose** – это способ описать логику работы сразу нескольких образов.

Для начала нужно создать файл.yml и вставить в него следующие команды:

version: “3.9”

services:

 dbpost:

  image: postgres:latest

  environment:

   POSTGRES\_DB: dbcompose

   POSTGRES\_USER: usrcompose

   POSTGRES\_PASSWORD: passcompose

  volumes:

   - ./init\_scripts/init.sql:/docker-entrypoint-initdb.d/init.sql

  ports:

   - 5432:5432

Далее вводим в консоли:

**$ docker-compose up -d- запуск проекта**

**ЗАКЛЮЧЕНИЕ**

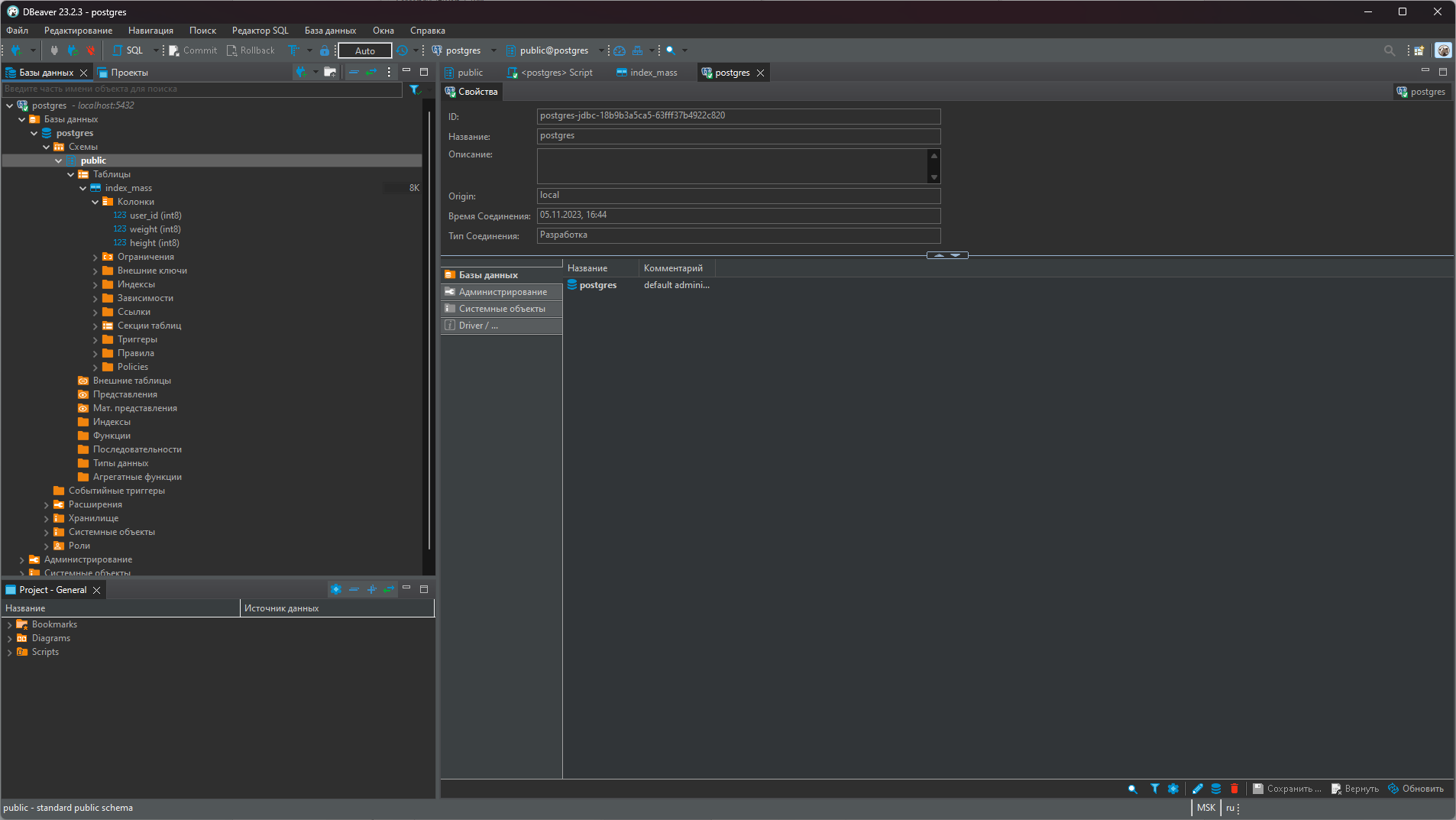
**Итогом выполнения данной лабораторной работы было:**

Была скачена и развернута СУБД Postgres, посредством сборки образа с volume

Получение базовых навыков работы с docker, а также затрагивание темы docker compose

Получение умения развертывать СУБД Postgres, посредством сборки образа

Получение навыка создания контейнеров, образов, а также взаимодействия с ними



**КОНТРОЛЬНЫЕ ВОПРОСЫ**

***Контрольные вопросы***

1. Что такое Docker?
2. Зачем нужен Docker?
3. Что такое docker-образ?
4. Что такое docker-контейнер?
5. Что такое volume и зачем он нужен?
6. Что такое docker-compose?
7. В чем разница между dockerfile и docker-compose?
8. Какая команда позволяет отправлять различные задания в запущенный докер-контейнер?
9. С помощью какого инструмента можно сохранить важные данные после аварийного отключения контейнера?

***Ответы на контрольные вопросы:***

**1)** Docker - это платформа с открытым исходным кодом, которая позволяет упаковывать, развертывать и запускать приложения с использованием контейнеризации.

**2)** Docker полезен по множеству причин:

- Упрощает развертывание и масштабирование приложений. Docker позволяет создавать состояние, независимое от окружения, и запускать его на любой системе.

- Облегчает совместную работу. Docker-контейнеры могут быть переданы другим разработчикам, что упрощает совместную разработку приложений.

- Улучшает надежность и безопасность. Docker предоставляет изолированное окружение, которое позволяет избежать конфликтов между приложениями и обеспечивает дополнительный уровень безопасности.

**3)** Docker-образ - это запускаемый пакет, который включает в себя исполняемый код, библиотеки, зависимости и настройки, необходимые для работы приложения. Он представляет собой шаблон для создания контейнеров. Образы создаются на основе Dockerfile - специального текстового файла, который содержит инструкции для сборки образа.

**4)** Docker-контейнер - это запущенный экземпляр Docker-образа. Контейнеры являются изолированными и легковесными, что позволяет запускать несколько контейнеров на одном хосте. Каждый контейнер имеет свою собственную файловую систему, процессы и сетевые ресурсы.

**5)** Volume в Docker представляет собой механизм для постоянного хранения данных в контейнере. Volume - это специальная директория или файл внутри контейнера, которая может быть использована для сохранения данных, которые переживают жизненный цикл контейнера. Он позволяет сохранить данные между запусками контейнера и обмениваться данными между контейнерами.

**6)** Docker-compose - это инструмент Docker, который позволяет определить и управлять множеством контейнеров как единое приложение. С помощью docker-compose можно описать конфигурацию приложения в YAML-файле, указать зависимости и настройки контейнеров, а затем запустить все контейнеры с помощью одной команды.

**7)** Dockerfile и docker-compose - два разных инструмента, но используются в связке. Dockerfile - это текстовый файл, в котором описаны все инструкции для сборки Docker-образа. Он определяет, какой базовый образ использовать, как скопировать файлы и настройки, как установить зависимости и многое другое.

При этом docker-compose - это файл конфигурации, который описывает различные сервисы и их зависимости, используя YAML-синтаксис. Он позволяет определить и управлять множеством контейнеров, как единое приложение.

**8)** Чтобы отправить задание в запущенный Docker-контейнер, необходимо использовать команду docker exec.