МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

По дисциплине: «Системы хранения и обработки данных»

Тема: «Развертывание СУБД Postgres с использованием средств автоматизации развертывания и управления приложениями»

Выполнил работу студент группы мИИВТ-241: Боченков А.С.

подпись, дата

Принял: Короленко В.В.

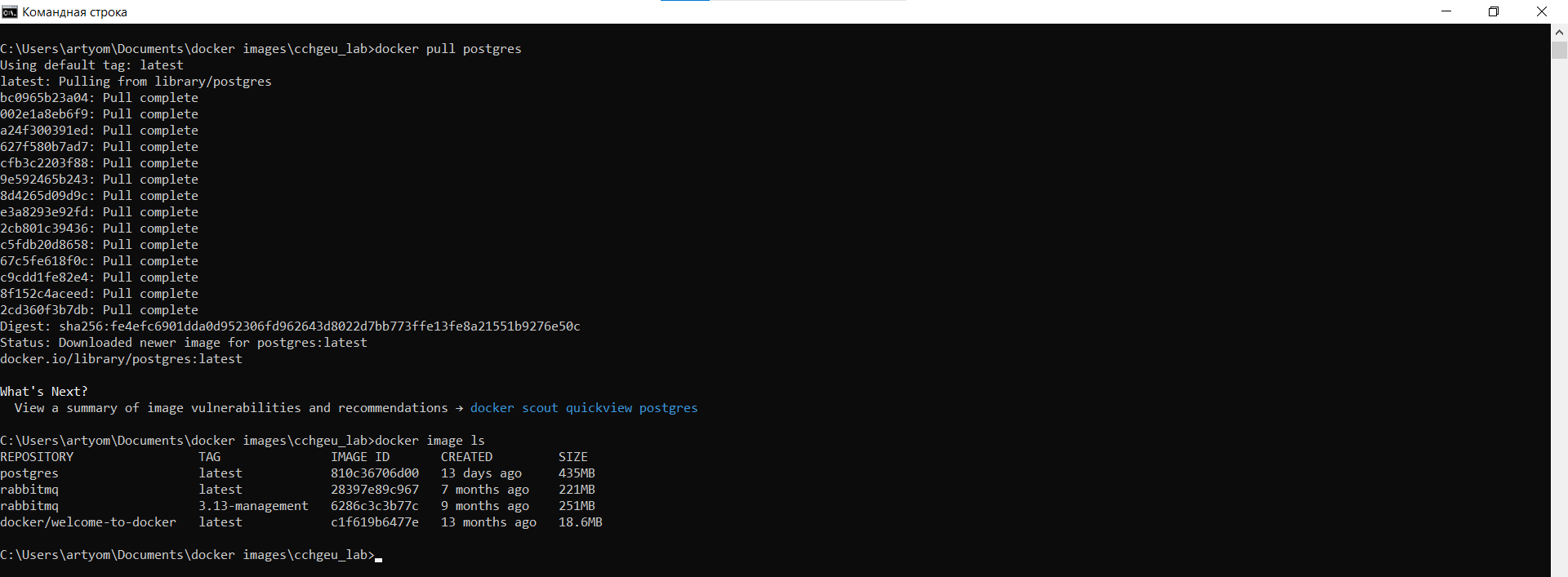
подпись, дата

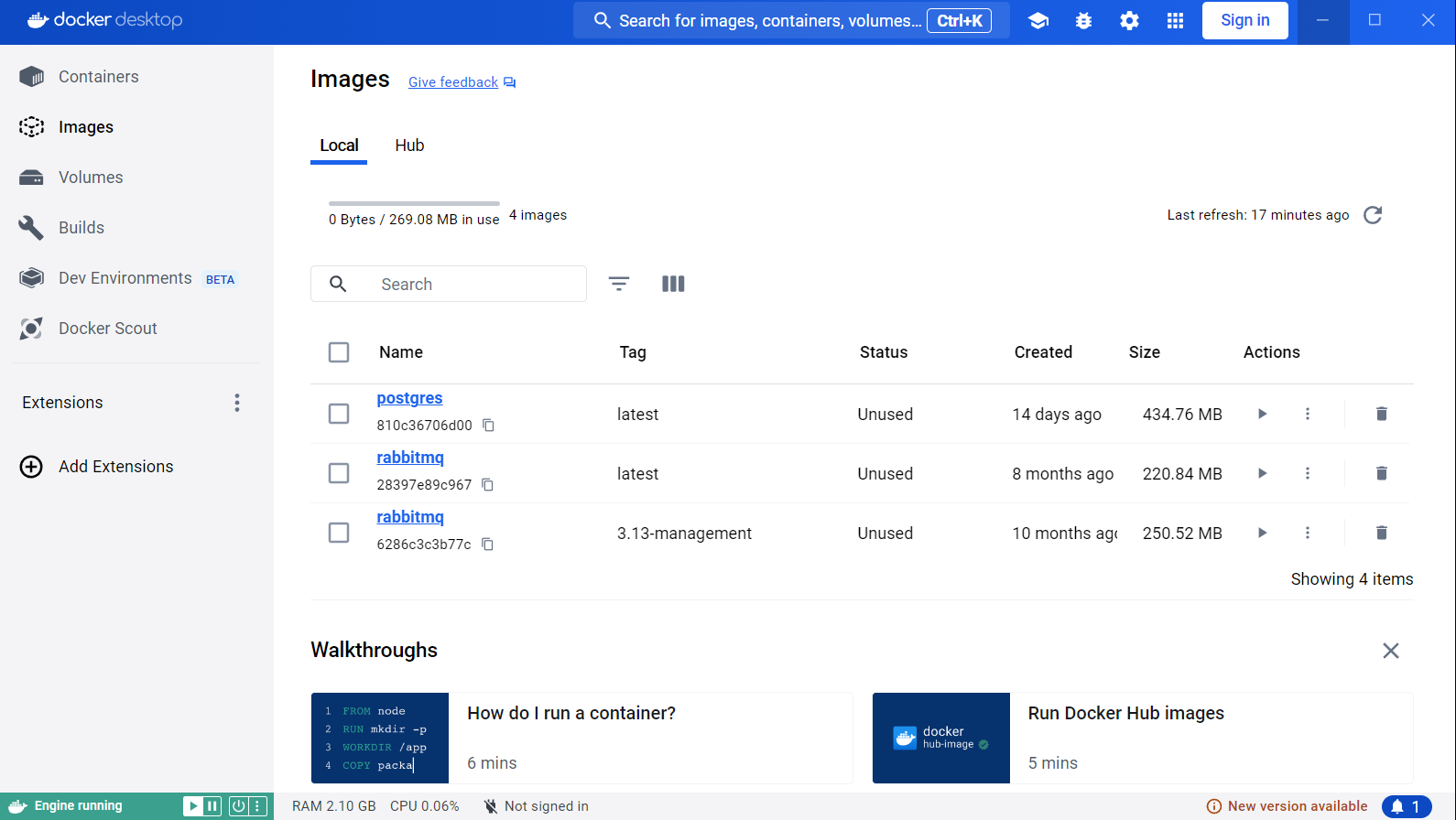
Воронеж 2024

Учебная задача

4. Скачать docker-образ с СУБД postgres с помощью команды для консоли.

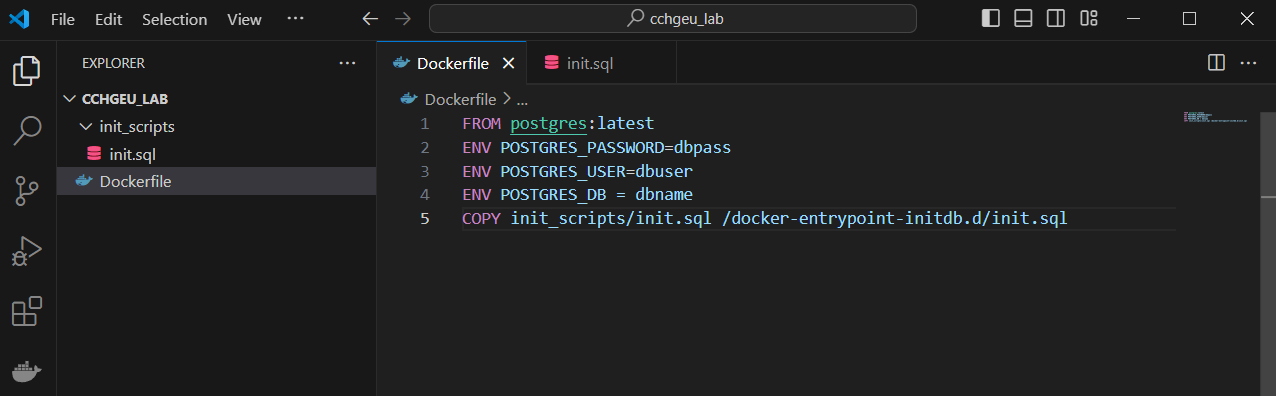
Скачаем docker-образ с СУБД postgres с помощью команды в терминале «docker pull postgres».





5. Создать Dockerfile

Создадим Dockerfile со следующим содержанием



Описание команд:

- FROM postgres:latest указывает какой docker образ (image) взять за основу для создания нового образа. :latest указывает, что нужно взять последнюю версию.

- ENV указывает значение переменной окружения.

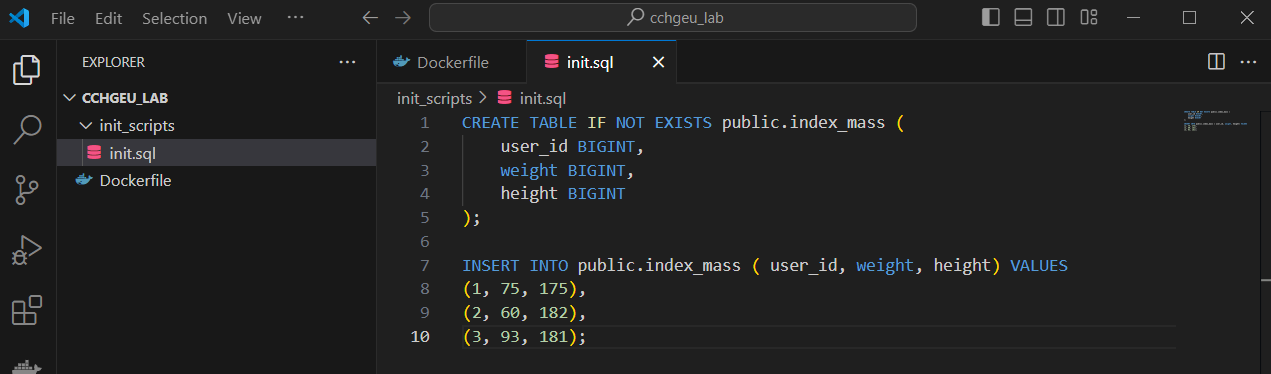
ENV POSTGRES\_PASSWORD = dbpass - создать пароль dbpass для входа.

ENV POSTGRES\_USER = dbuser – создает нового пользователя с названием dbuser, к которому сразу же прикрепляется пароль dbpass. Если пользователь не указан, то прикрепляется по умолчанию к пользователю postgres.

ENV POSTGRES\_DB = dbname – создает новую базу данных с названием dbname.

COPY init\_script/init.sql /docker-entrypoint-initdb.d/init.sql – копирует с локального диска файл init.sql, который находится в директории init\_script, в директорию /docker-entrypoint-initdb.d находящимся в новом образе.

Теперь создадим файл init.sql



CREATE TABLE IF NOT EXISTS public.index\_mass создает таблицу в с названием index\_mass в схеме public, если таблица с таким названием не существовала ранее. Это делается для того, чтобы случайно не стереть уже имеющуюся таблицу с данными.

Создаются атрибуты в домене (таблице) с названием user\_id, weight, height тип данных которых BIGINT.

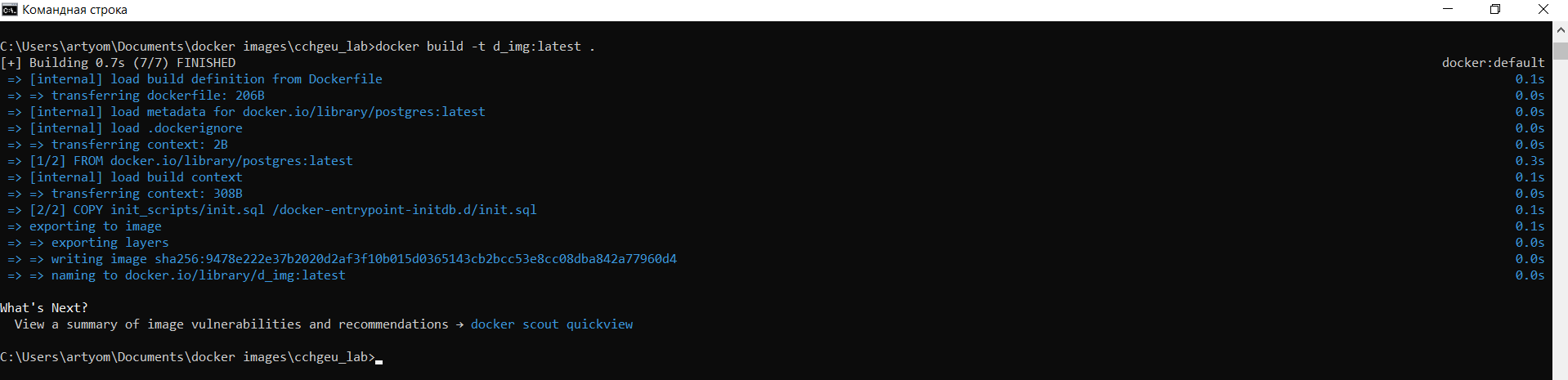
INSERT INTO public.index\_mass( user\_id, weight, height) VALUES () означается что мы добавляем значение атрибутов в таблицу index\_mass именно указанным в скобках порядке. При этом порядок атрибутов и их следования не обязаны совпадать с порядком и количеством в имеющейся таблице.

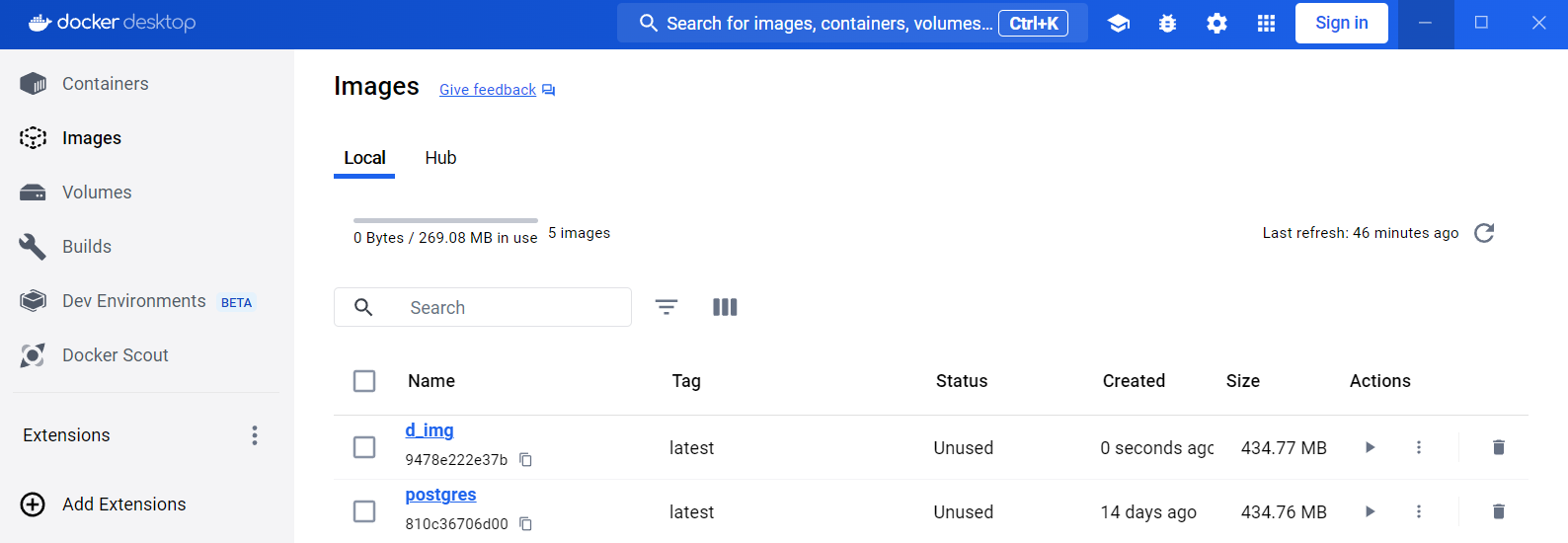
6. Создать образ с помощью команды «docker build –t d\_img:latest . »

Для того, чтобы создать новый образ с помощью Dockerfile необходимо находиться в директории, где находится данный файл и ввести следующую команду «docker build –t d\_img:latest . ».

docker build – команда создания образа;

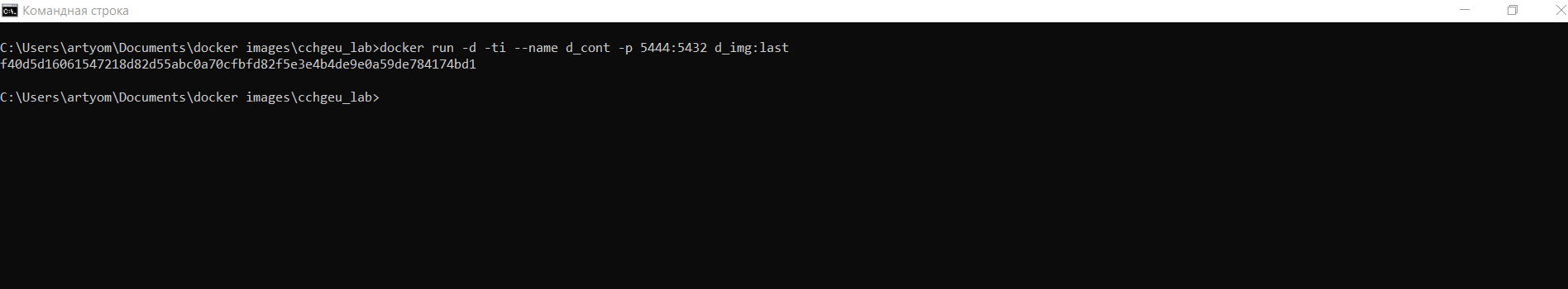
-t – флаг t(tag) указывает имя образа его создании d\_img, а :latest указывает название (версию) тега. В данном случае, он стандартный, но указан явно.

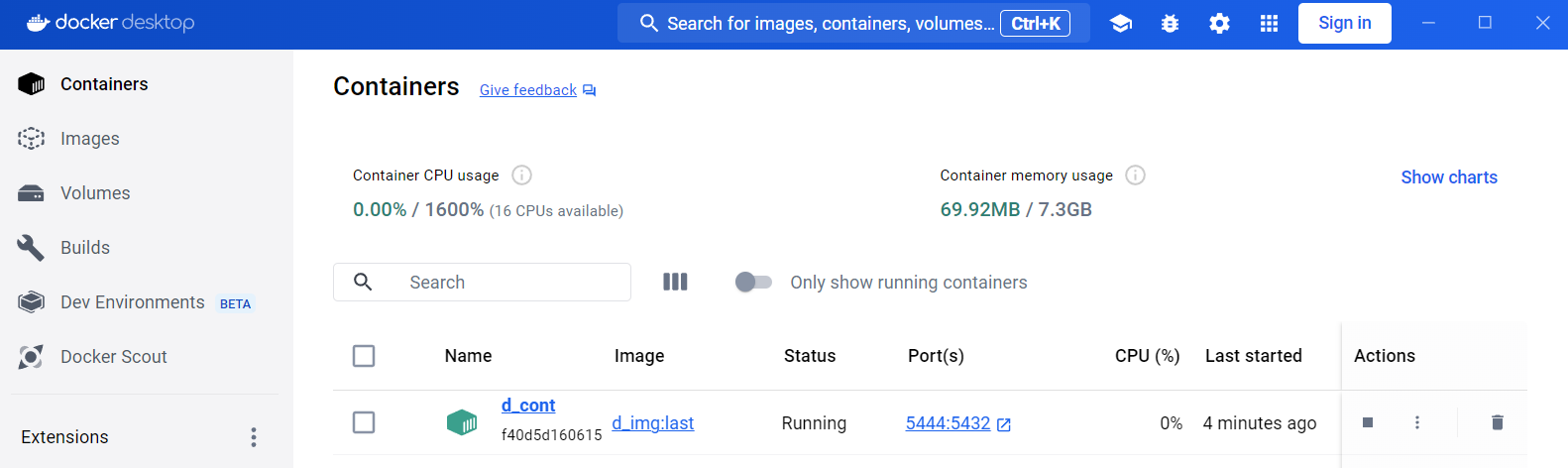




7. Чтобы воспользоваться созданной СУБД необходимо запустить контейнер с именем «d\_cont» на основе образа «d\_img:latest» с помощью команды: «docker run -d -p 5432:5432 --name d\_cont d\_img:latest».

Запустим команду в терминале «docker run -d -ti --name d\_cont -p 5444:5432 d\_img:latest»





docker run – запускает (создает) контейнер.

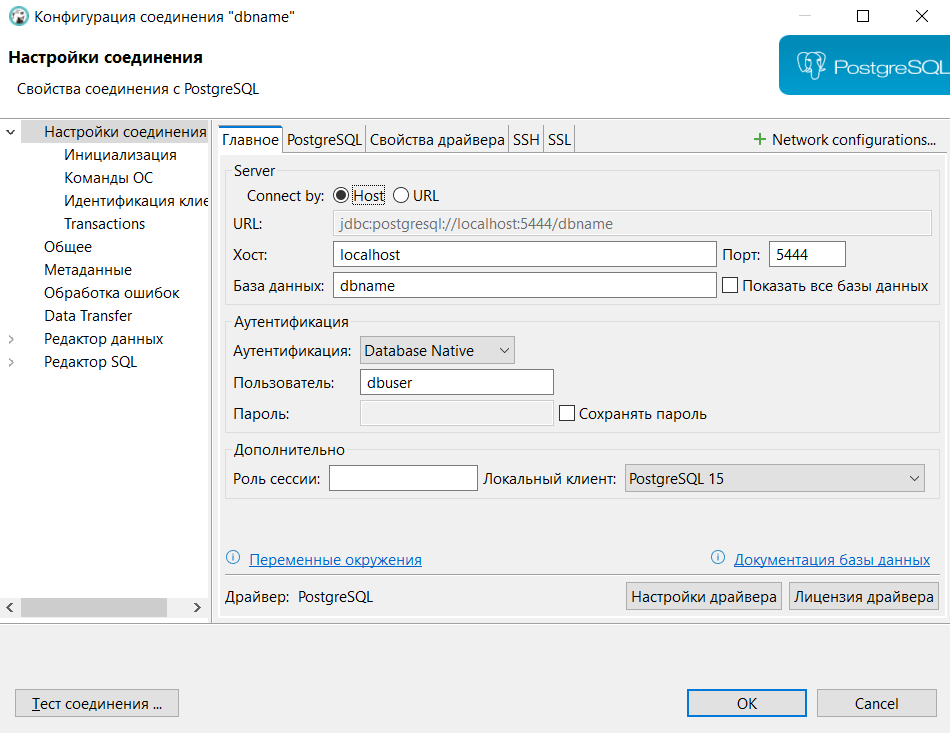
-d – флаг (demon) создает процесс как демон, т.е. выполняется в фоновом режиме и отдает контроль на введение новых команд в терминале.

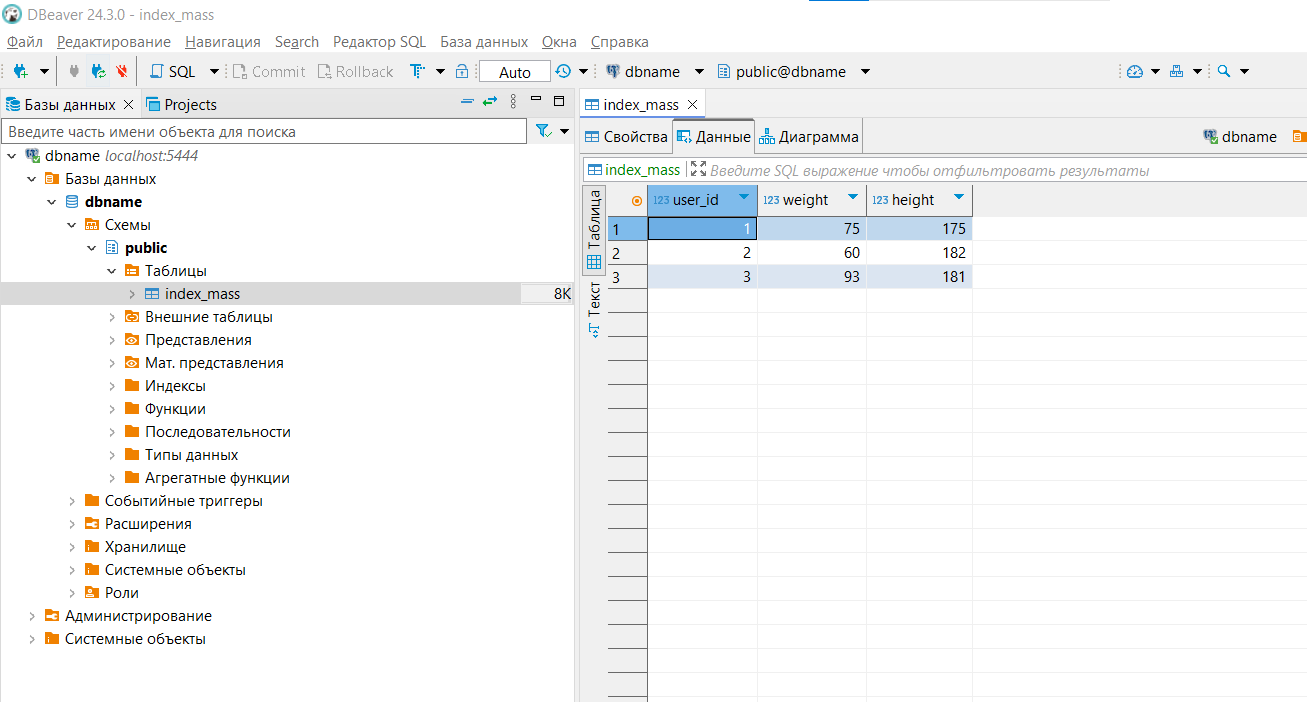
-t, -i – флаг -t (-tty) вызывает псевдо tty, -i (--interactive) – держит поток ввода (STDIN) открытым. Так как флаги с одиночной буквой, то мы имеем право их объединить в любом порядке -t -i = -ti = -it .

--name – указываем имя контейнера (d\_cont).

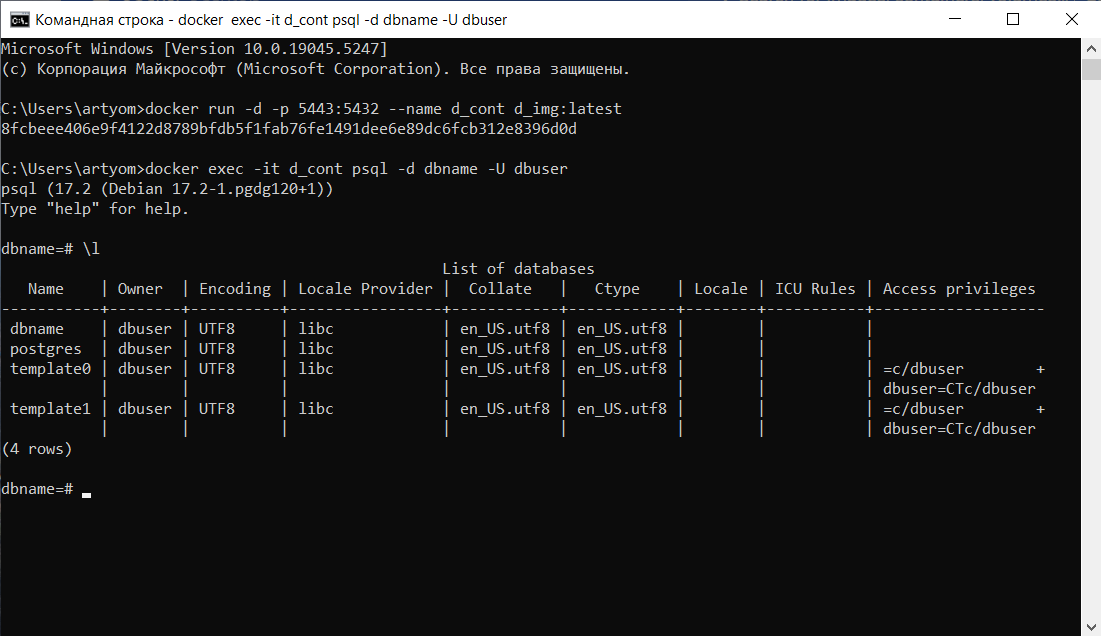
-p (--expose) – указывает номер порта локальной системы, который соединяется с номером порта в контейнере. Порт 5444 на локальной системе выбран для избегания конфликтов с имеющимся postgres в системе.

8. Проверим работоспособность нашего контейнера через DBeaver.





9. Написать специальную команду docker, которая позволит подключаться к работающему контейнеру, запускать интерфейс psql и вносить новые данные «на лету»: «docker exec -it d\_cont psql -d -U usr dbn».

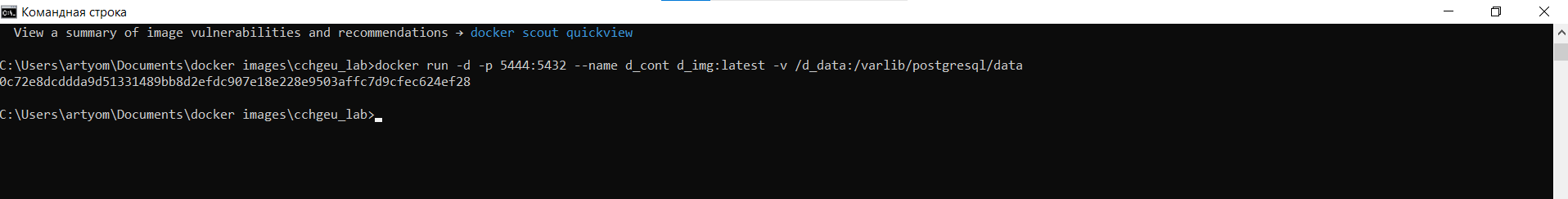


10-11. Чтобы данные, измененные в процессе работы контейнера, были доступны после удаления контейнера (во вновь созданном контейнере) необходимо создать том (volume).

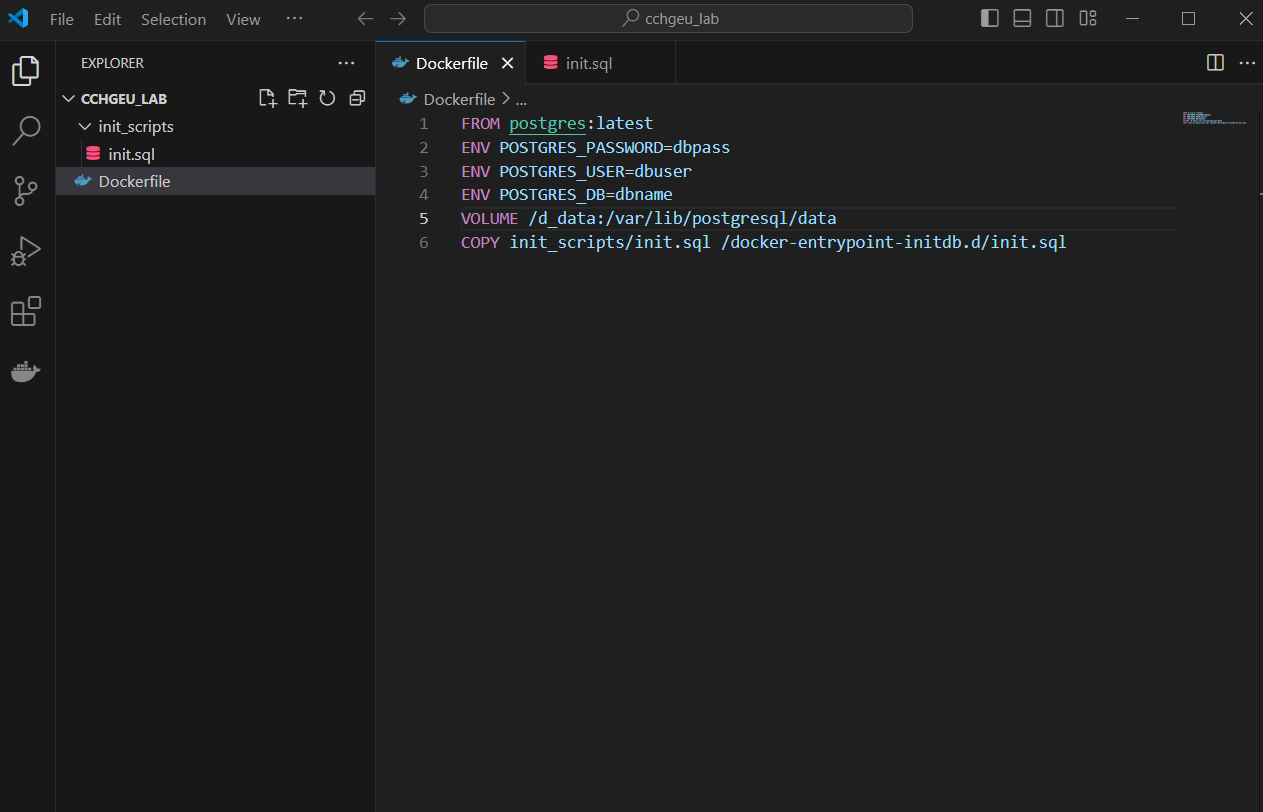
Для того, чтобы добавить том (volume) удалим контейнер с помощью команды «docker container rm d\_cont:latest». После этого добавить том в контейнер можно двумя путями: введя дополнительный флаг –v при создании контейнера, или прописать это в Dockerfile. Покажем оба пути, они аналогичны.

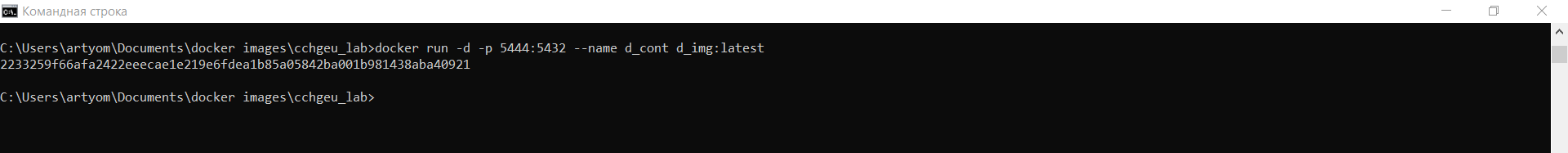
«docker run –d –p 5444:5432 --name d\_cont d\_img:latest –v /d\_data:/varlib/postgresql/data».

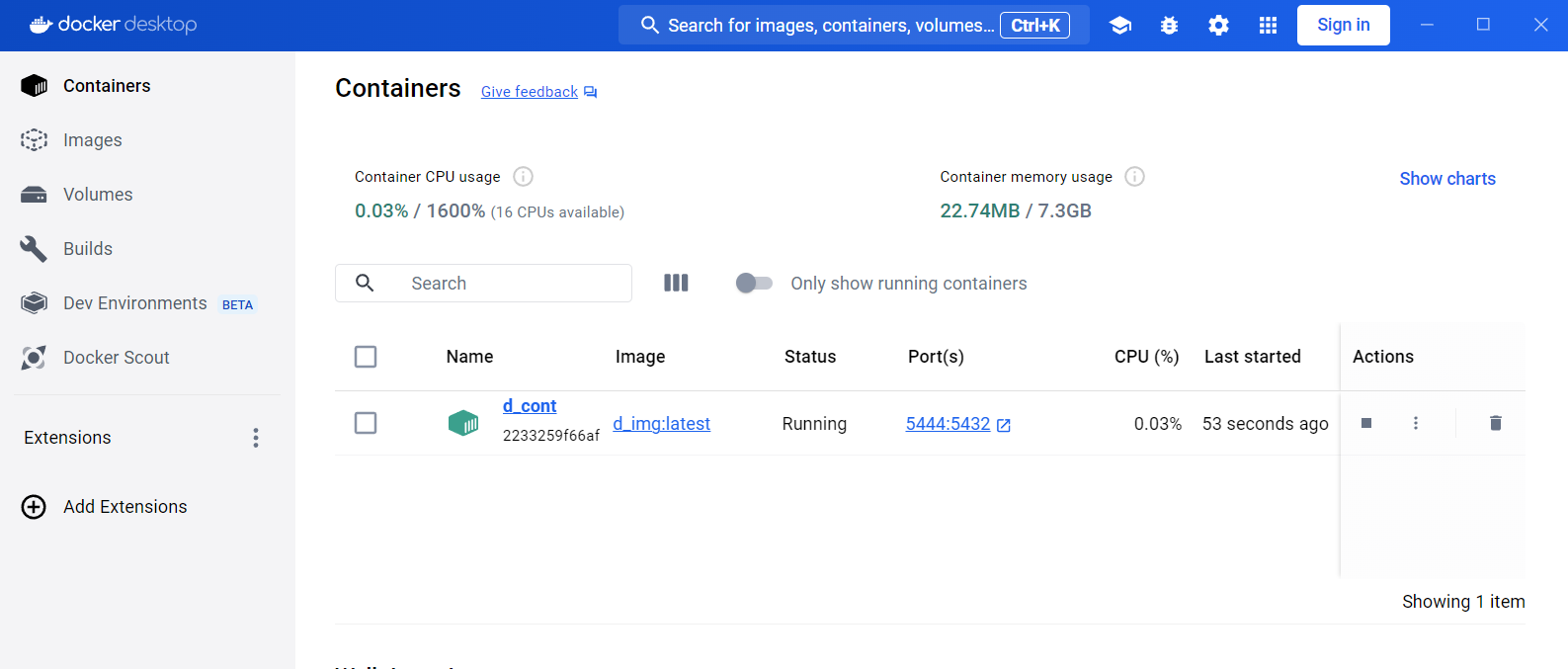
–v /d\_data:/varlib/postgresql/data – Локально данные будут храниться в директории /d\_data, а в контейнере в директории /varlib/postgresql/data.



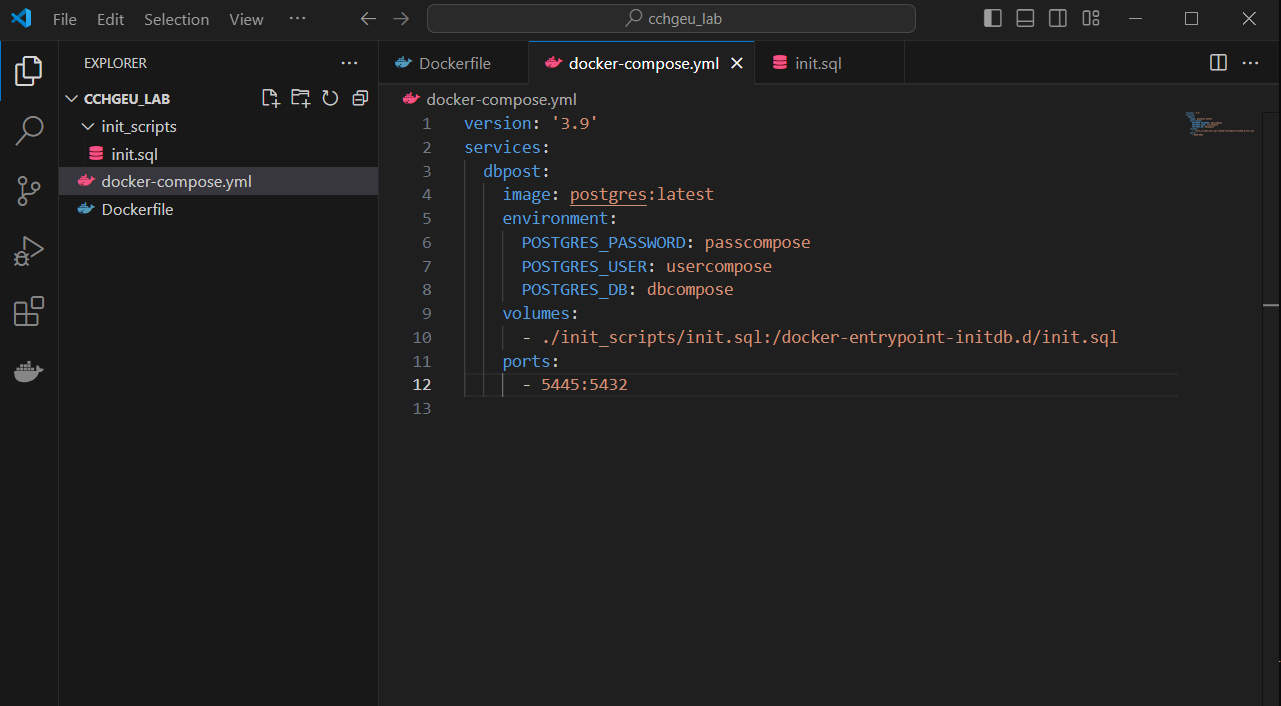
Второй путь



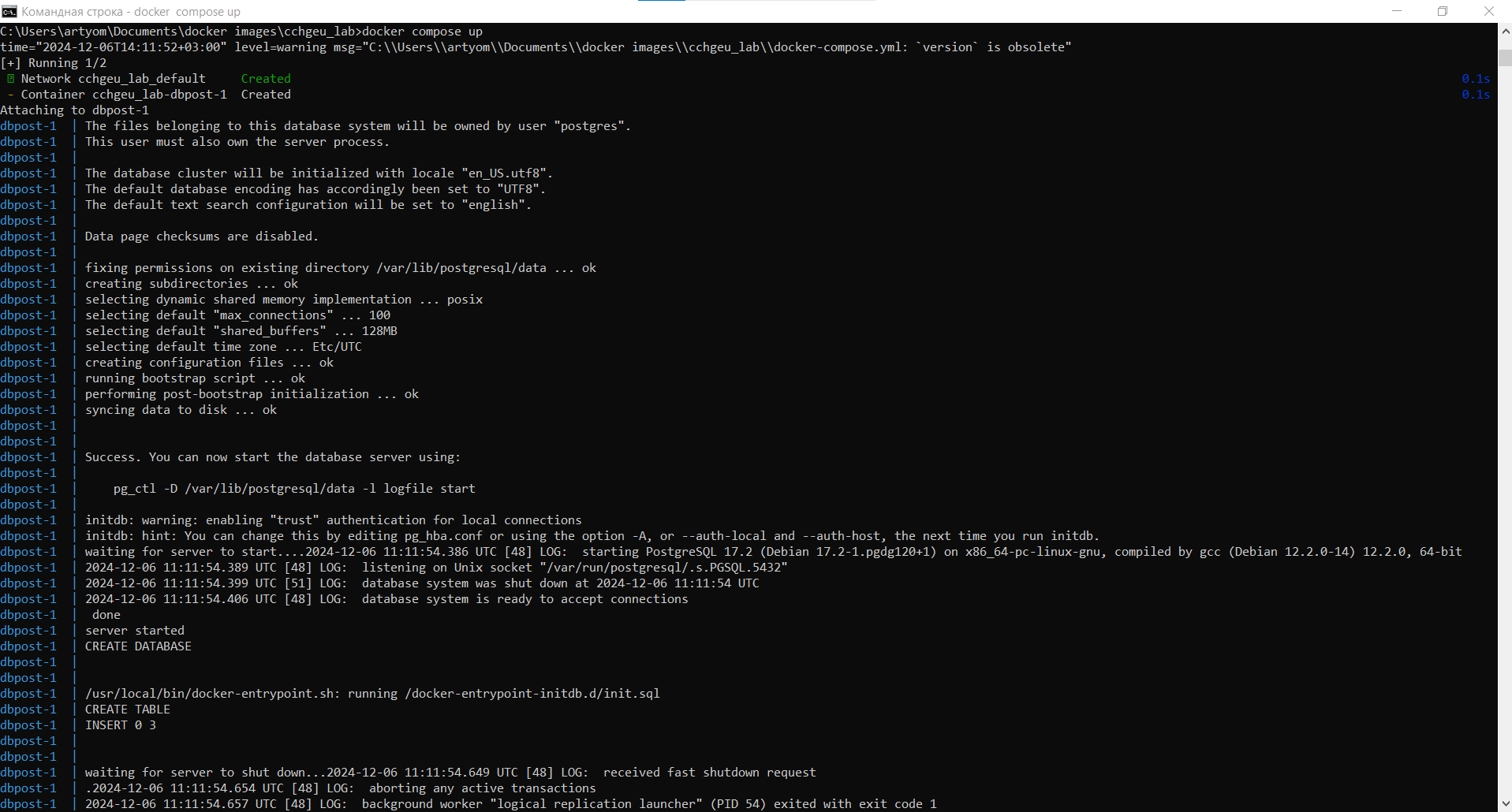


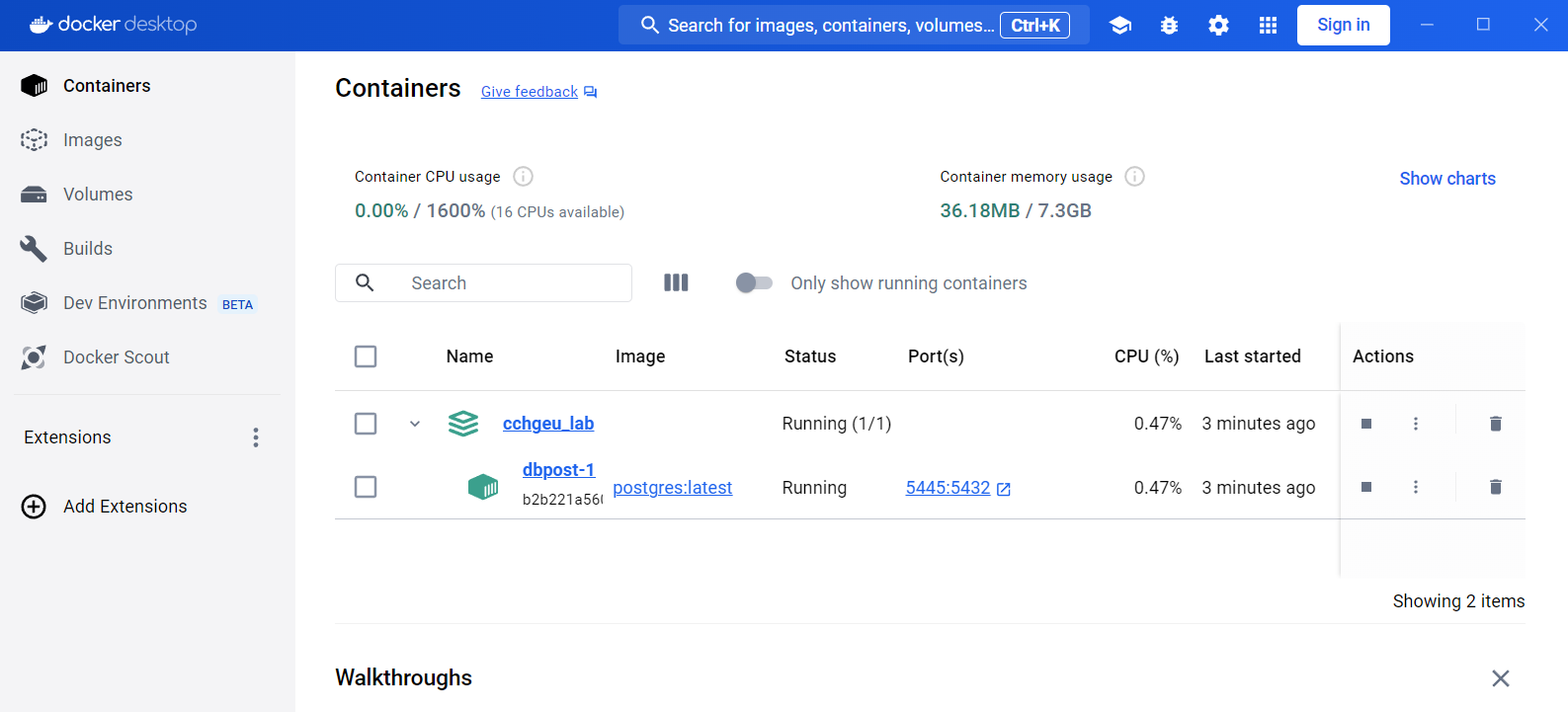


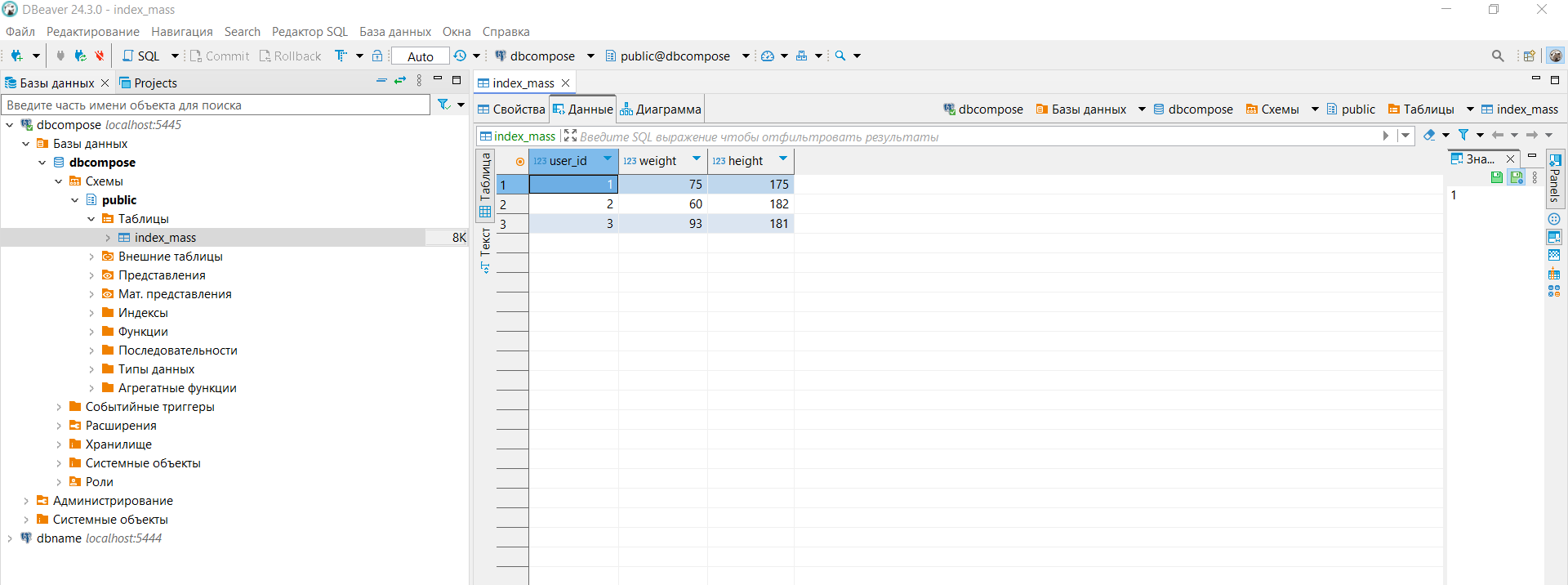
13. Создать контейнер с использованием файла docker-compose.yml



Затем командой «docker compose up» запустим







Контрольные вопросы

1. Что такое Docker?

Docker это открытая платформа для разработки, развертывания и запуска приложений, которая доступна на Windows, Linux, Mac. Так же Docker это название компании, которая создает и поддерживает этот продукт, а так предоставляет инфраструктуру Docker Hub с готовыми бесплатными образами и дополнительные платные услуги компаниям по поддержанию.

2. Зачем нужен Docker?

Docker можно быстро разворачивать приложения независимые от инфраструктуры, на которой они выполняются, очень быстро. Так же это позволяет удобно тестировать, обновлять приложение, а собственное окружение, поможет избежать проблем с зависимостями.

3. Что такое docker-образ?

Docker-образ это некоторая конфигурация, которая задает параметры, включает в себя файлы, программы, библиотеки на основе чего создается docker-контейнер. Существуют два главных принципа образа:

* Неизменность. Нельзя вносить изменение в созданный образ. Можно создали лишь новый образ или внести новую версию изменения, но предыдущий изменить нельзя.
* Контейнер разделен на слои. Каждый слой определяет установки файловой системы, которая в свою очередь может добавлять, удалять и изменять файлы.

4. Что такое docker-контейнер?

В упрощенном объяснении контейнер запускает каждый компонент системы в изолированном процессе со своей окружающей средой, файловой системой, файлами и настройками. В связи с этим контейнер:

* Самодостаточен и не зависит от машины, на который он запущен.
* Изолирован. Работа внутри контейнера имеет самое минимальное влияние на запускающую машину и другие контейнеры. Для взаимодействия контейнеров между собой нужно прописывать дополнительные настройки.
* Независим. Выступает как отдельная единица. Изменение и удаление контейнера не повлечет за собой изменение других контейнеров.
* Портативный. Может запускаться на любой операционной системе, как на локальной машине, так и на удаленной.

5. Что такое volume и зачем от нужен?

Volume (том) это постоянно место для сохранения данных для контейнера. По умолчанию он не задается, поэтому необходимо задавать его отдельно. С помощью него сохраняются данные во время изменения программы внутри контейнера. Без него при удалении и создании контейнера изменения пропадут, создается чистая система, но благодаря данным в томе, мы может новый контейнер соединить с ним и увидим все изменения данных.

6. Что такое docker-compose?

Docker-compose это дополнение над Docker, которая помогает задать и создать не один контейнер, а множество прописав все их настройки и дополнительное соединения организовав сеть между ними и прописать маршрутизацию потоков данных.

7. В чем разница между docker и docker-compose?

Docker позволяет задать настройки только одного контейнера, когда docker-compose сразу нескольких, связав их в сеть и задать доп настройки сети для коммуникации между процессами.

8. Какая команда позволяет отправлять различные задания в запущенный докер-контейнер?

docker exec (cmd) позволяет запустить различные задания в запущенный контейнер.

9. С помощью какого инструмента можно сохранить важные данные после аварийного отключения контейнера?

Volume (том) позволяет сохранить данные контейнера, которые не будут зависеть от существования контейнера.