

Работа с docker compose

Начальный композ файл:

```
# 1. Создать новую папку под проект
# 2. Установить docker compose: sudo apt install docker-compose
# 3. Создать этот docker-compose.yml
# 4. Создать подпапку db
# 5. Выполнить docker compose up -d
# 6. Открыть в браузере localhost:8080 — должна появиться веб-админка БД
# 7. Войти в админку под юзером postgres и паролем example

# Use postgres/example user/password credentials

services:

  db:
    image: postgres
    # set shared memory limit when using docker-compose
    shm_size: 128mb
    volumes:
      - ./db:/var/lib/postgresql/data
    environment:
      POSTGRES_PASSWORD: example

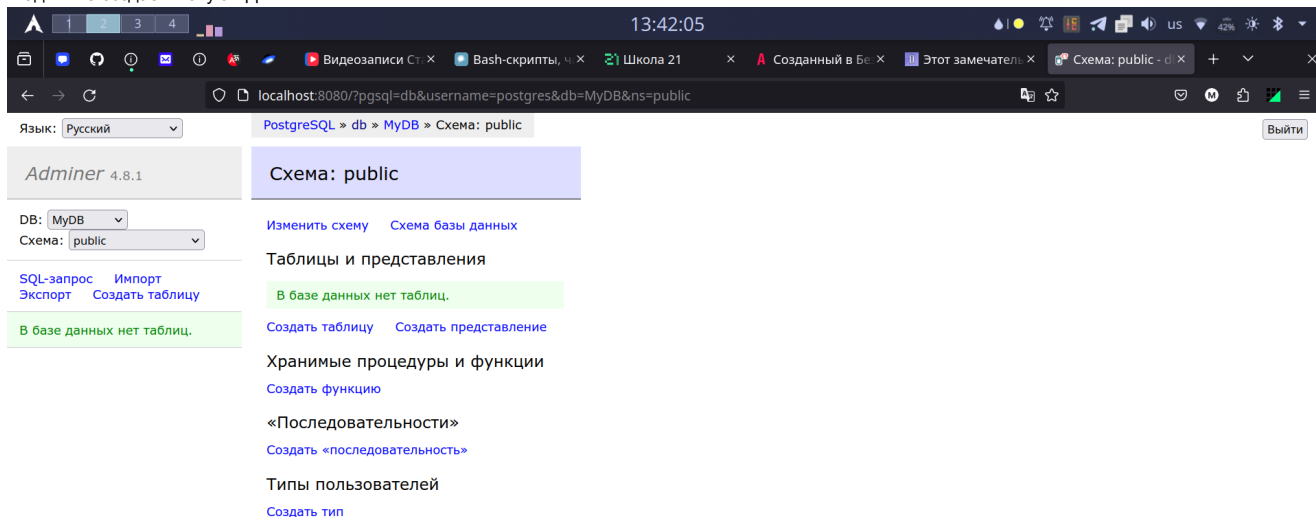
  adminer:
    image: adminer
    ports:
      - 8080:8080
```

Нужно применить `docker-compose up -d` для создания контейнера с постгресом и админкой для него.

Создание нового собственного образа

Суть в том, чтобы из питона подключиться к запущенной в докере БД и получить от туда какие-то данные.

1. В админке создаем новую БД:



2. Создаем в ней новую таблицу:

13:45:14

localhost:8080/?pgsql=db&username=postgres&db=MyDB&ns=public&create=

Язык: Русский PostgreSQL > db > MyDB > public > Создать таблицу Выйти

Adminer 4.8.1

DB: MyDB
Схема: public

SQL-запрос Экспорт Создать таблицу

В базе данных нет таблиц.

Создать таблицу

Название таблицы: Сохранить

Название поля	Тип	Длина	Действие	NULL	AI?	+
id	integer			<input type="checkbox"/>	<input checked="" type="radio"/>	✕
first_name	character var	125		<input type="checkbox"/>	<input type="radio"/>	✕
age	integer			<input type="checkbox"/>	<input type="radio"/>	✕
last_name	character var	125		<input type="checkbox"/>	<input type="radio"/>	✕
	integer			<input type="checkbox"/>	<input type="radio"/>	✕

Автоматическое приращение: ☐ Значения по умолчанию ☐ Комментарий

Сохранить

13:46:03

localhost:8080/?pgsql=db&username=postgres&db=MyDB&ns=public&table=users

Язык: Русский PostgreSQL > db > MyDB > public > Таблица: users Выйти

Adminer 4.8.1

DB: MyDB
Схема: public

SQL-запрос Экспорт Создать таблицу

выбрать users

Таблица: users

Таблица была создана, 10:45:50 SQL-запрос

Выбрать Показать структуру Изменить таблицу Новая запись

поле	Тип	Комментарий
id	bigint Автоматическое приращение [nextval('users_id_seq')]	
first_name	character varying(125)	
age	integer	
last_name	character varying(125)	

Индексы

PRIMARY id

Изменить индексы

Внешние ключи

Добавить внешний ключ

Триггеры

Добавить триггер

3. Создать новую запись:

4. Создать папку src в проекте и в ней python скрипт :

5. Записать в скрипт:

```
import psycopg2

db_config = {
    "host": "db",
    "database": "mydb",
    "user": "manager",
    "password": "12345678",
}

def get_data():
    """ Retrieve data from the vendors table """
    try:
        with psycopg2.connect(**db_config) as conn:
            with conn.cursor() as cur:
                cur.execute("SELECT * FROM students")
                rows = cur.fetchall()

                print("The number of rows: ", cur.rowcount)
                for row in rows:
```

```

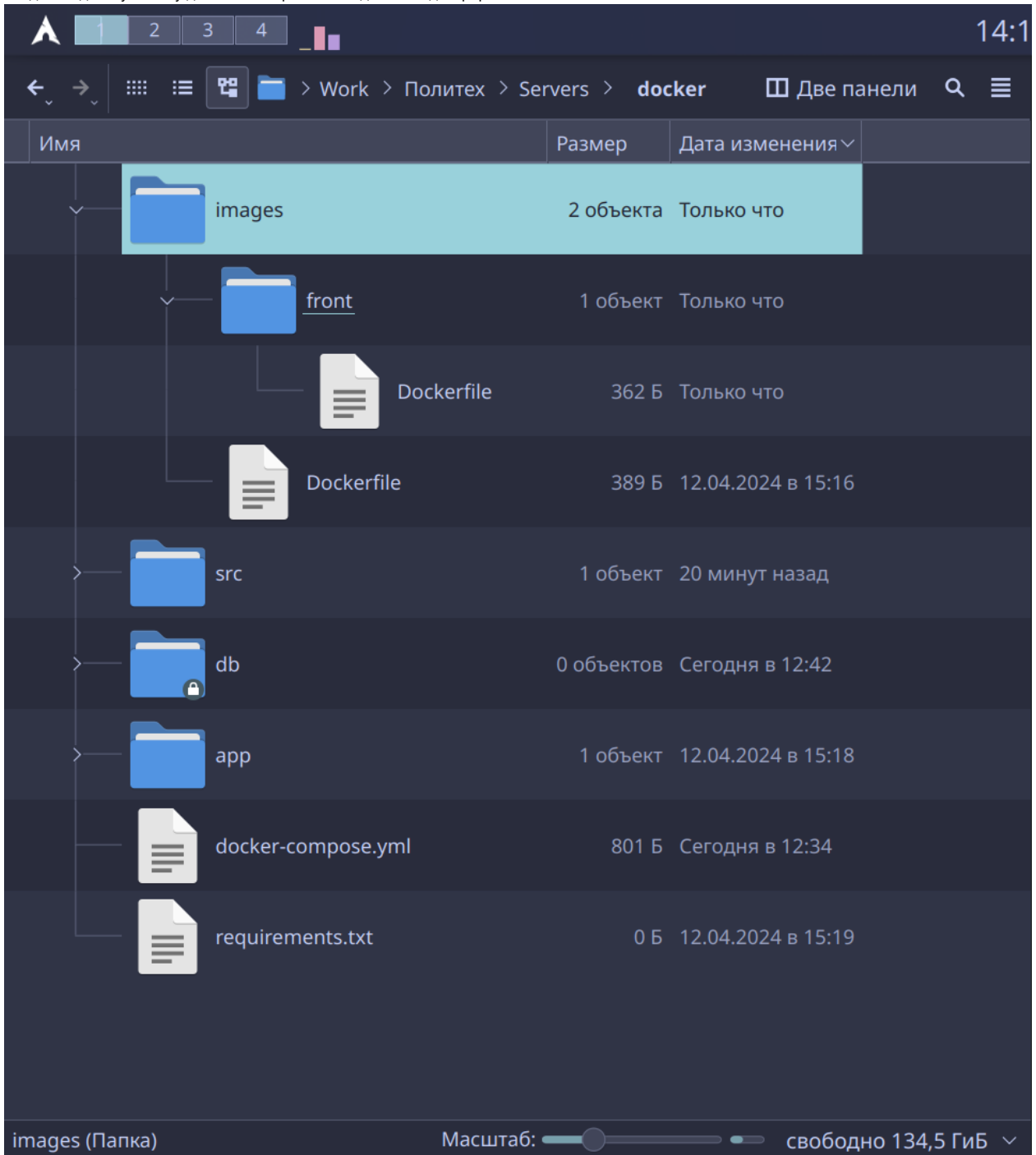
        print(row)

    except (Exception, psycopg2.DatabaseError) as error:
        print(error)

if __name__ == '__main__':
    get_data()

```

7. Создать отдельную папку для наших образов и создаем там докер файл:



8. В этой же папке нужно создать файл `requirements.txt` и записываем в него:

```
psycopg2
```

3. редактируем файл `Dockerfile` в папке `images/front`:

```

FROM python:3

WORKDIR /usr/src

COPY requirements.txt ./
RUN pip install --no-cache-dir -r requirements.txt

CMD [ "python", ]

```

9. Изменяем файл docker-compose.yml:

```
# 1. Создать новую папку под проект
# 2. Установить docker compose: sudo apt install docker-compose
# 3. Создать этот docker-compose.yml
# 4. Создать подпапку db
# 5. Выполнить docker compose up -d
# 6. Открыть в браузере localhost:8080 – должна появиться веб-админка БД
# 7. Войти в админку под юзером postgres и паролем example

# Use postgres/example user/password credentials

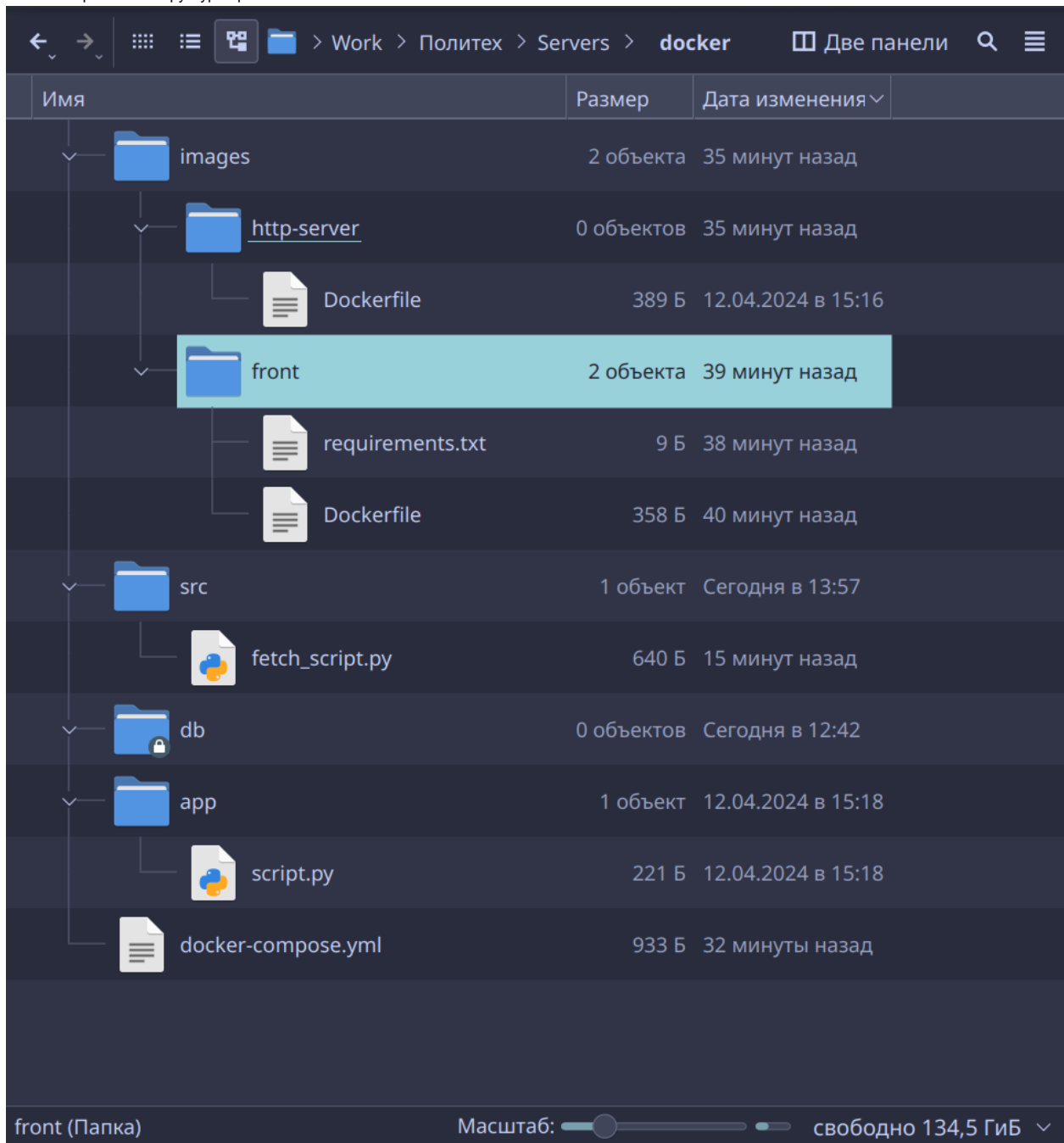
services:

  db:
    image: postgres
    # set shared memory limit when using docker-compose
    shm_size: 128mb
    volumes:
      - ./db:/var/lib/postgresql/data
    environment:
      POSTGRES_PASSWORD: example

  adminer:
    image: adminer
    ports:
      - 8080:8080

  front:
    build: images/front
    volumes:
      - ./src:/usr/src
    command:
      - python3
      - /usr/src/fetch_script.py
```

10. Конечная файловая структура проекта:



Имя	Размер	Дата изменения
images	2 объекта	35 минут назад
http-server	0 объектов	35 минут назад
Dockerfile	389 Б	12.04.2024 в 15:16
front	2 объекта	39 минут назад
requirements.txt	9 Б	38 минут назад
Dockerfile	358 Б	40 минут назад
src	1 объект	Сегодня в 13:57
fetch_script.py	640 Б	15 минут назад
db	0 объектов	Сегодня в 12:42
app	1 объект	12.04.2024 в 15:18
script.py	221 Б	12.04.2024 в 15:18
docker-compose.yml	933 Б	32 минуты назад

front (Папка) Масштаб: свободно 134,5 ГиБ

11. Создать нового пользователя в админке и дать ему права доступа к базе:

```
create role manager password '12345678';
alter role manager with login;
grant all privileges on database mydb to manager;
alter table students owner to manager;
```

12. Запускаем докер композ: `docker compose up -d`

13. запускаем нужный нам контейнер со скриптом: `docker compose run --rm front`