

Создать docker compose с двумя сервисами, чтобы по одной команде они поднимались и останавливались:

Сервис 1: БД Postgres с одной таблицей , куда вставлена одна строка данных.

Сервис 2: Программу на любом языке программирования, которая читает данные из этой БД и печатает эту строку.

Для создания Docker Compose с двумя сервисами — PostgreSQL и приложением, читающим данные из базы данных, — выполню следующие шаги. Я буду использовать Python для приложения, которое будет подключаться к базе данных и выводить данные.

Шаг 1: Создание структуры проекта

Создам следующую структуру каталогов:

```
my_project/
├── docker-compose.yml
├── db/
│   └── init.sql
└── app/
    ├── Dockerfile
    └── main.py
```

Шаг 2: Настройка PostgreSQL

1. В файле db/init.sql добавлю SQL-команды для создания таблицы и вставки данных:

```
CREATE TABLE IF NOT EXISTS test_table (
    id SERIAL PRIMARY KEY,
    message VARCHAR(255) NOT NULL
);
```

```
INSERT INTO test_table (message) VALUES ('Hello from PostgreSQL!');
```

Шаг 3: Настройка приложения на Python

1. В файле app/main.py я напишу код для подключения к базе данных и чтения данных:

```
import psycopg2
import os

def main():
    # Получаем параметры подключения из переменных окружения
    db_host = os.getenv('DB_HOST', 'db')
    db_name = os.getenv('DB_NAME', 'mydatabase')
    db_user = os.getenv('DB_USER', 'postgres')
    db_password = os.getenv('DB_PASSWORD', 'password')

    # Подключаемся к базе данных
    connection = psycopg2.connect(
```

```

    host=db_host,
    database=db_name,
    user=db_user,
    password=db_password
)

cursor = connection.cursor()

# Выполняем запрос к таблице
cursor.execute("SELECT message FROM test_table;")

# Получаем результат
row = cursor.fetchone()

if row:
    print(row[0])
else:
    print("No data found.")

# Закрываем соединение
cursor.close()
connection.close()

if __name__ == "__main__":
    main()

```

2. Создам файл app/Dockerfile с содержимым:

```

# Используем официальный образ Python
FROM python:3.9-slim

# Устанавливаем зависимости
RUN pip install psycorg2-binary

# Копируем код приложения в контейнер
COPY main.py /app/main.py

# Устанавливаем рабочую директорию
WORKDIR /app

# Запускаем приложение
CMD ["python", "main.py"]

```

Шаг 4: Настройка Docker Compose

1. В файле docker-compose.yml добавлю следующее содержимое:

```

version: '3.8'

services:

```

```
db:
  image: postgres:13
  environment:
    POSTGRES_DB: mydatabase
    POSTGRES_USER: postgres
    POSTGRES_PASSWORD: password
  volumes:
    - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql
```

```
app:
  build: ./app
  depends_on:
    - db
  environment:
    DB_HOST: db
    DB_NAME: mydatabase
    DB_USER: postgres
    DB_PASSWORD: password
```

Шаг 5: Запуск проекта

1. Открою терминал и перейду в каталог my_project.
2. Выполню команду для сборки и запуска сервисов:

```
docker-compose up --build
```

Шаг 6: Остановка проекта

Чтобы остановить сервисы, просто нажму Ctrl+C в терминале, где запущен Docker Compose, или выполню команду:

```
docker-compose down
```

Результат

После выполнения команды `docker-compose up --build`, я должен увидеть вывод, содержащий строку "Hello from PostgreSQL!", что означает, что приложение успешно подключилось к базе данных и прочитало данные из таблицы.