



Микросервисная архитектура и Docker Compose

Создание микросервисных проектов

Микросервисная архитектура — это стиль проектирования, в котором приложение разбивается на маленькие, независимые компоненты (сервисы), которые:

- выполняют отдельную бизнес-задачу (а не всё сразу),
- разворачиваются, масштабируются и обновляются независимо,
- общаются между собой по сети

В отличие от монолита:

Монолит	Микросервисы
Весь код в одном сервисе	Каждый сервис отвечает за свою часть
Один процесс	Много процессов, каждый со своим кодом
Сложно масштабировать частями	Масштабируется по отдельным сервисам
Обновление = перезапуск всего	Можно обновлять кусками

Пример: три контейнера (PostgreSQL + Adminer + Python backend)

Создать общую сеть

```
docker network create mynet
```

Запустить PostgreSQL

```
docker run -d \  
  --name db \  
  --network mynet \  
  -e POSTGRES_USER=user \  
  -e POSTGRES_PASSWORD=password \
```

```
-e POSTGRES_DB=mydb \  
-v pgdata:/var/lib/postgresql/data \  
postgres:15
```

Запустить Adminer

```
docker run -d \  
  --name adminer \  
  --network mynet \  
  -p 8080:8080 \  
  adminer
```

Запустить backend

```
docker run -d \  
  --name backend \  
  --network mynet \  
  -p 8000:8000 \  
  -e DB_HOST=db \  
  -e DB_PORT=5432 \  
  -e DB_NAME=mydb \  
  -e DB_USER=user \  
  -e DB_PASS=password \  
  my-backend
```

Все три связаны через сеть **mynet**. Связь между сервисами — по DNS-именам контейнеров, благодаря Docker-сети.

Docker Compose

Docker Compose — это инструмент, позволяющий описывать и запускать многоконтейнерные приложения через один YAML-файл (**docker-compose.yml**).

docker-compose.yml

```
version: '3.8'
```

```
services:
  db:
    image: postgres:15
    container_name: my_postgres
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
      POSTGRES_DB: mydb
    volumes:
      - pgdata:/var/lib/postgresql/data
    networks:
      - backend-net

  adminer:
    image: adminer
    container_name: my_adminer
    ports:
      - "8080:8080"
    depends_on:
      - db
    networks:
      - backend-net

  backend:
    build: ./backend
    container_name: my_backend
    ports:
      - "8000:8000"
    environment:
      DB_HOST: db
      DB_PORT: 5432
      DB_NAME: mydb
      DB_USER: user
      DB_PASS: password
    depends_on:
      - db
    networks:
      - backend-net
```

```
volumes:
```

```
  pgdata:
```

```
networks:
```

```
  backend-net:
```