

# Docker-compose

## Лучшие практики





## Что будем делать:

A vertical list icon consisting of three horizontal lines, each ending in a small circle.

Узнаем best practice по работе с Docker

Посмотрим как работать с Docker compose



## Best practice

- Тюнинг Dockerfile
- .dockerignore
- Размер образа
- Кеширование
- Multi-stage сборка

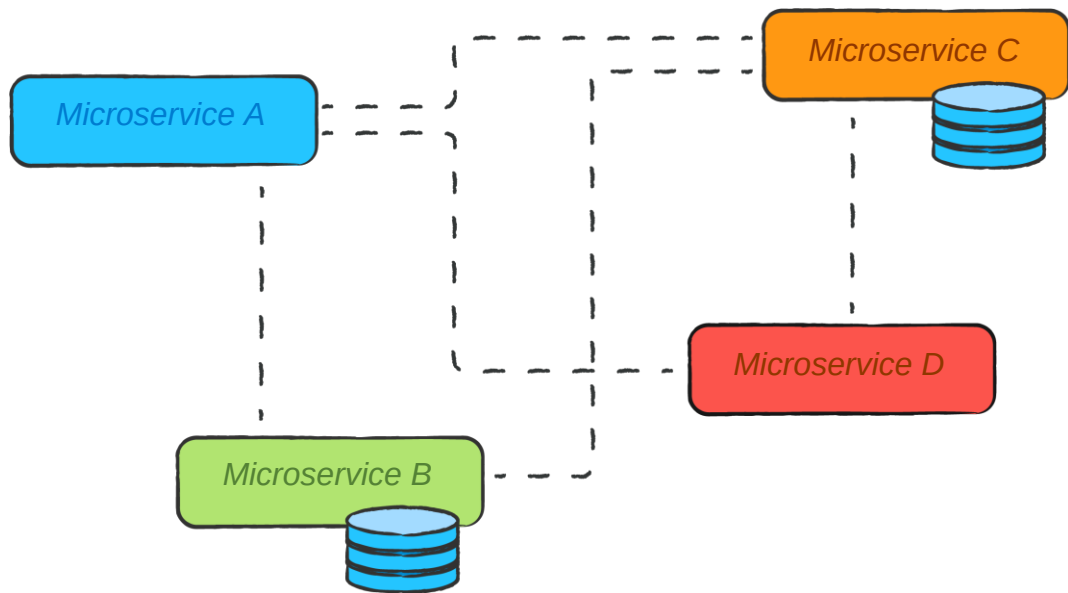


# DEMO



# Docker-compose

<https://docs.docker.com/compose/>





# Docker-compose

- Позволяет запустить и настроить многоконтейнерное приложение
- Все описываем в `docker-compose.yml`
- Создает свою сеть проекту
- Дает возможность обращаться контейнерам друг к другу по именам



# Docker-compose

```
version: '2'
services:
  nginx:
    image: nginx:1.2.3
    ports:
      - "8000:80"
    volumes:
      - ./hosts:/etc/nginx/conf.d
      - ./www:/var/www
      - ./logs:/var/log/nginx
  php:
    build: ./images/php
    volumes:
      - ./www:/var/www
```



# Docker-compose

- **docker-compose build** – собрать проект
- **docker-compose up -d** – запустить проект
- **docker-compose down** – остановить проект
- **docker-compose logs -f [service name]** – посмотреть логи сервиса
- **docker-compose ps** – вывести список контейнеров
- **docker-compose exec [service name] [command]** – выполнить команду
- **docker-compose images** – список образов





# DEMO



# Проблематика использования Docker (compose) в продакте:

- Как откатить?
- Как бесшовно обновить?
- ВЫХОД: использовать оркестратор



## Потому что там:

- «Качественный» деплой
- Service discovery
- Балансировка
- Отказоустойчивость
- Удобство управления
- Варианты сетевых реализаций и файловых хранилищ



## Полезные ссылки:

- <https://clck.ru/MBtKt> - про CI/CD в целом
- <https://docs.docker.com/compose/>
- <https://docs.docker.com/compose/gettingstarted/>
- [https://docs.gitlab.com/ee/ci/docker/using\\_docker\\_build.html](https://docs.gitlab.com/ee/ci/docker/using_docker_build.html)
- <https://docs.docker.com/develop/develop-images/baseimages/>
-