

# Docker

Урок #2 части «Микросервисная архитектура и контейнеризация»

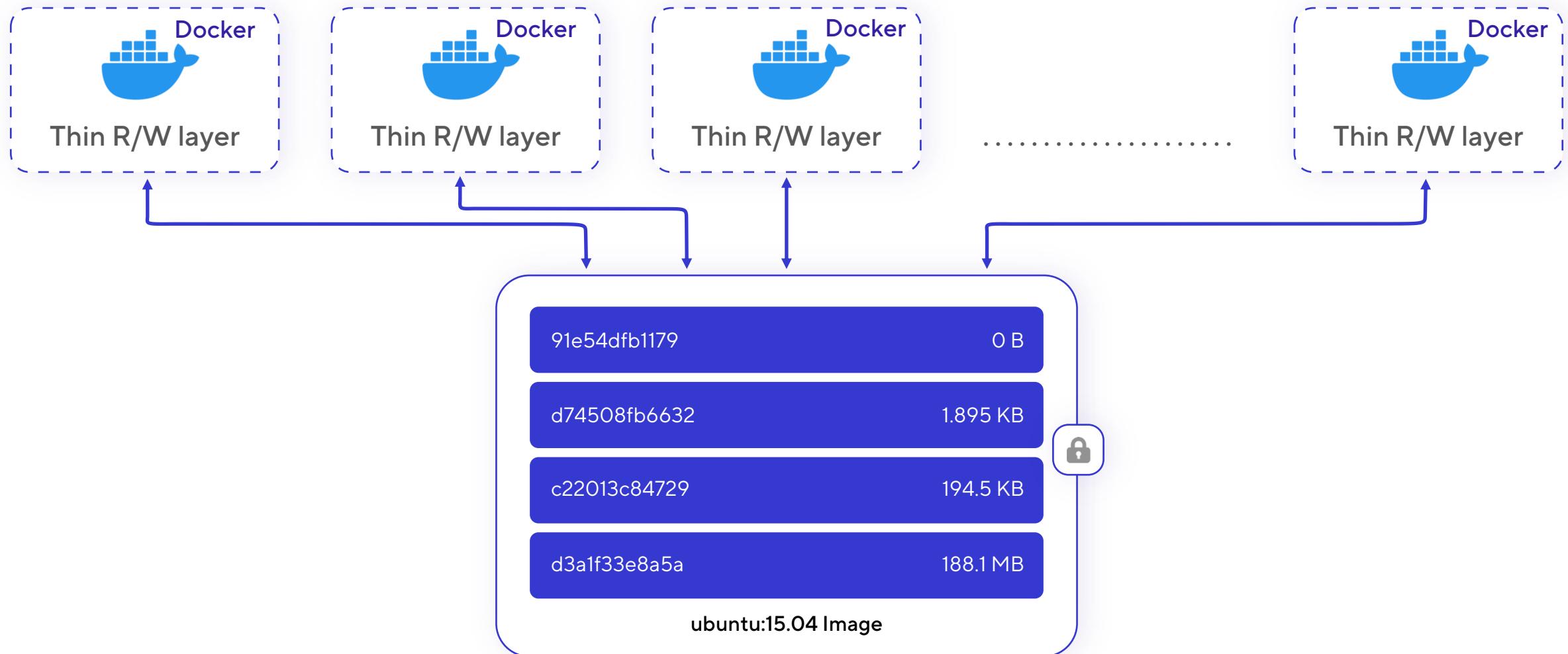


# План урока



1. Запуск контейнеров
2. Создание образов
3. Dockerfile
4. Запуск мультисервисных окружений  
(docker-compose)
5. Оркестратор контейнеров

# Образ и контейнер



# Установка Docker



<https://docs.docker.com/get-docker/>



# Запускаем первый контейнер

Имя контейнера

```
$ docker run --name nginx -d -p 8080:80 nginx:1.15
```

Имя контейнера

Образ и тег, из которого  
создается контейнер

```
$ docker run --name nginx -d -p 8080:80 nginx:1.15
```

Имя контейнера

Образ и тег, из которого создается контейнер

Запуск в **background**

```
$ docker run --name nginx -d -p 8080:80 nginx:1.15
```



# List containers



Показать все контейнеры, в  
том числе остановленные

```
$ docker ps -a
```



Получить tty  
Держать STDIN



Команда для запуска в  
контейнере

```
$ docker exec -t -i nginx bash
```



Имя или ID контейнера

Получать новые логи  
по мере появления

Получить последние 10  
записей

```
$ docker logs -f --tail 10 nginx
```



Имя или ID контейнера

 Остановить  
запущенный контейнер

```
$ docker rm -f nginx
```



Имя или ID контейнера

# Docker не про контейнеры



## 1. Стандартизация

- сборка
- поставка
- конфигурация

## 2. Воспроизводимость

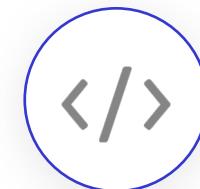
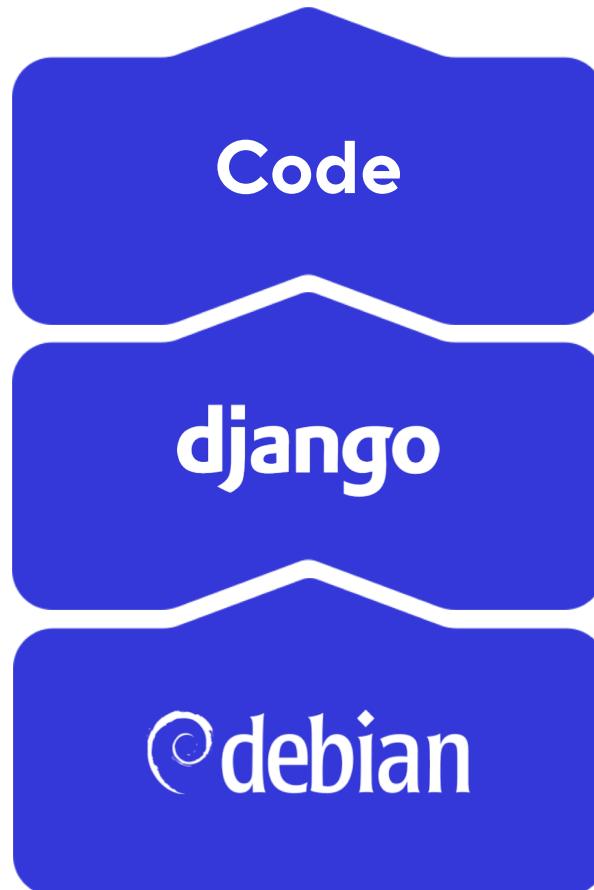
## 3. Консистентность



# Dockerfile



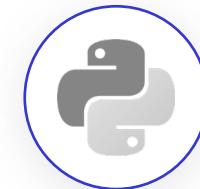
<https://docs.docker.com/engine/reference/builder/>



## Код приложения

**Слой 6: ENTRYPOINT** команда для запуска приложения

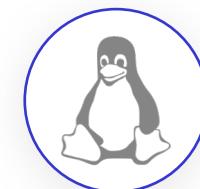
**Слой 5: ADD** файлы приложения



## Зависимости приложения

**Слой 4: RUN** команды установки модулей и библиотек

**Слой 3: RUN** команды для установки пакетов OS

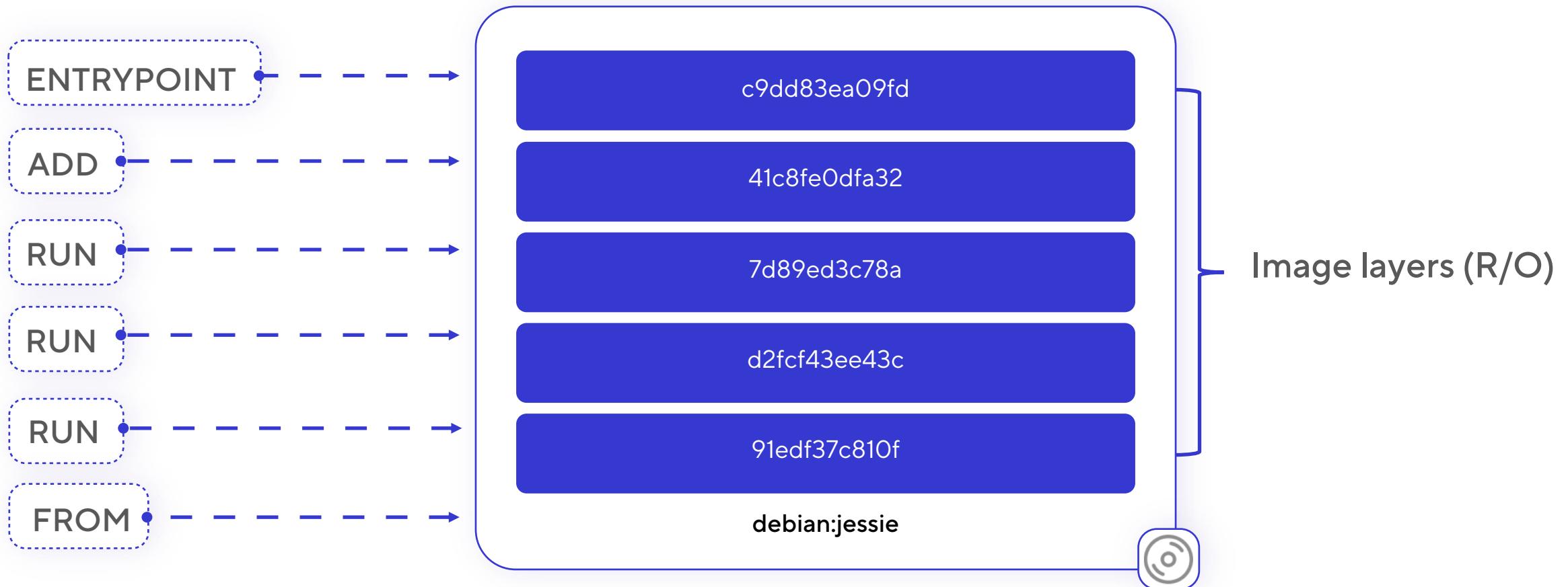


## Операционная система

**Слой 2: RUN** команды для настройки OS

**Слой 1: FROM** debian:jessie

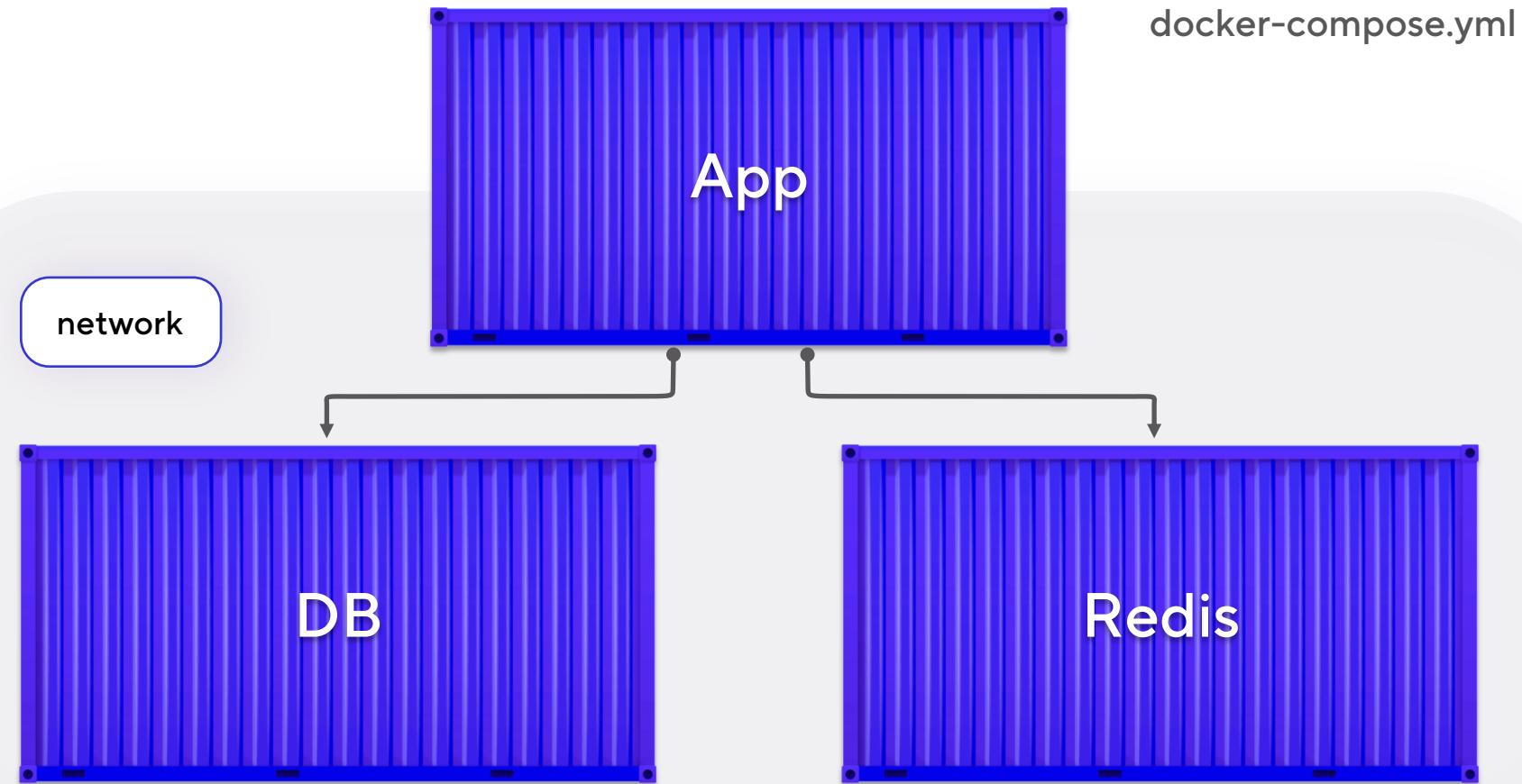
# Image



# Docker-compose



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



# Docker Cheat Sheet



## Build

Сборка образа из Dockerfile в текущей директории  
`docker build -t image_name:tag`.

Изменение имени образа или тэга  
`docker tag old_image_name new_image_name`

Сохранение образа в файл image.tar  
`docker save image_name -o image.tar`

Загрузка образа из файла image.tar  
`docker load -i image.tar`

## List

Список всех запущенных контейнеров  
`docker ps`

Все контейнеры на сервере  
`docker ps -a`

Все образы на сервере  
`docker images`

## Run

Чтобы создать и запустить контейнер  
`docker run image_name:tag`

Полезные параметры для команды run:

- d – Запуск контейнера в фоне
- rm – Удаление контейнера после остановки
- p – Проксирование локального порта в порт контейнера
- e – Переменные окружения, передаваемые приложению
- v – Подключение вольюмов в контейнер
- ti – Интерактивный режим (для запуска shella)

## Clean Up

Удаление всех образов, неиспользуемых контейнерами  
`docker image prune -a`

Удаление всех остановленных контейнеров  
`docker container prune -f`

Остановка запущенного контейнера  
`docker stop container_name`  
`docker kill container_name`

## Images

Скачать имадж  
`docker pull image_name:tag`

Загрузка имаджа в registry  
`docker push image_name:tag`

Просмотр команд в имадже  
`docker history image_name:tag --no-trunc`

## One-liners

Удаление всех контейнеров  
`docker rm -f $(docker ps -qa)`

Удаление неименованных образов  
`docker rmi $(docker images -q -f dangling=true)`

Просмотр файла в контейнере  
`docker exec container_name cat /etc/config.conf`

## Docker-compose

Запуск проекта из директории с docker-compose.yml файлом  
`docker-compose up -d`

Остановка и удаление проекта  
`docker-compose down`