

Онлайн образование

otus.ru



Проверить, идет ли запись

Меня хорошо видно && слышно?



Тема вебинара

Docker



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

<https://vk.com/nick.lavlinsky>



Преподаватель



Лавлинский Николай

Более 15 лет в веб-разработке

Преподавал в ВУЗе более 10 лет
Более 3 лет в онлайн-образовании

Специализация: оптимизация производительности,
ускорение сайтов и веб-приложений

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Slack #general



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом

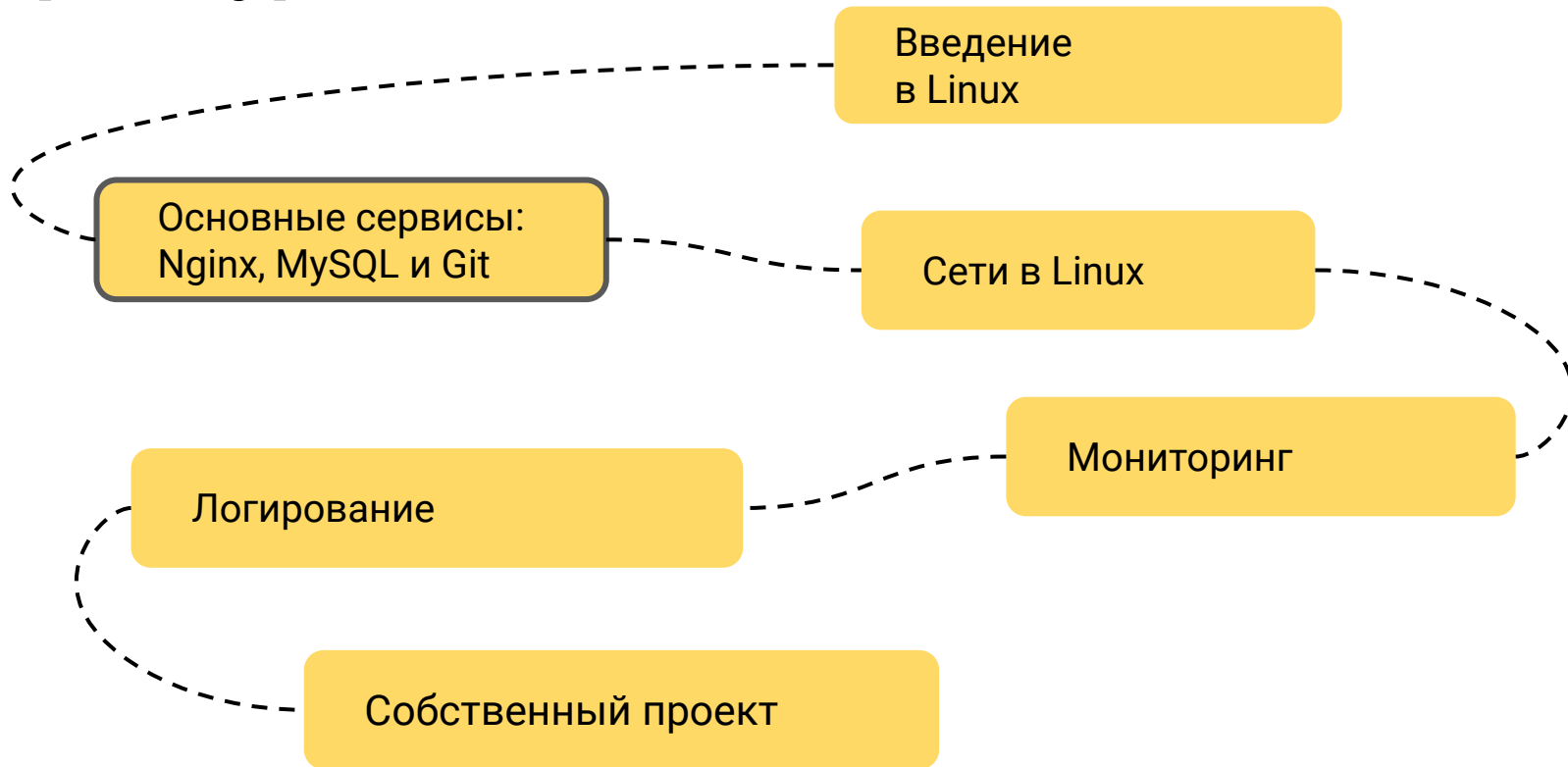


Документ



Ответьте себе или
задайте вопрос

Карта курса



Маршрут вебинара



Контейнеризация

Архитектура Docker

Управление контейнерами

Сборка контейнеров

Цели вебинара

После занятия вы сможете

1. Ориентироваться в терминах контейнеризации
2. Управлять образами контейнеров
3. Запускать контейнеры
4. Создавать собственные образы

Смысл

Зачем вам это уметь

1. Запускать системы из docker-образов
2. Снимать ограничения дистрибутива ОС
3. Упаковывать собственные приложения в образ
4. Автоматизировать публикацию микросервисов



Контейнеризация



Что такое, зачем?

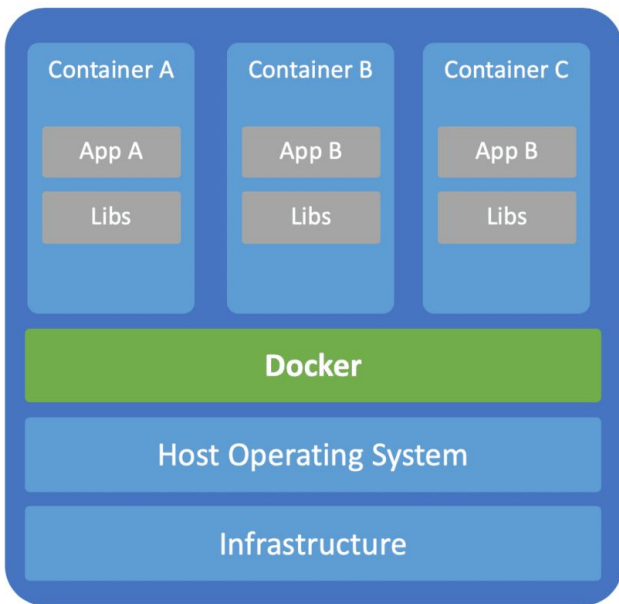


Контейнеризация

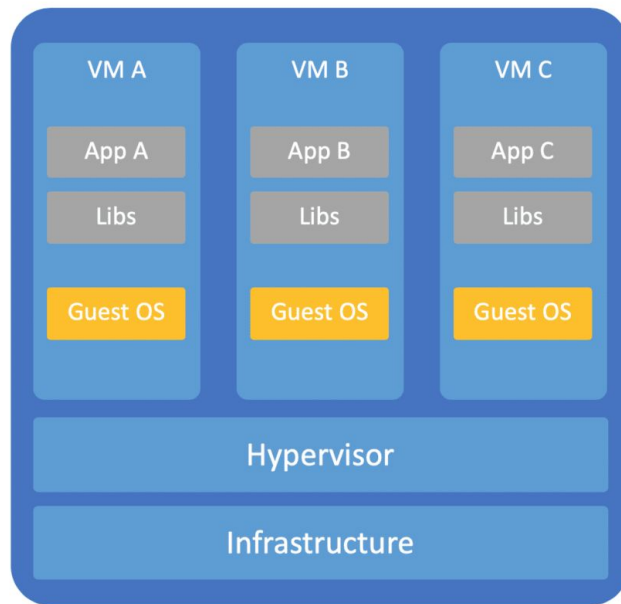
- Метод легковесной виртуализации процессов
- В основе - изоляция процессов через cgroups, namespaces и другие механизмы
- Используется только одно ядро ОС
- Возможны ограничения ресурсов по CPU, RAM, сети и т.д.
- Реализации:
 - Docker
 - LXC
 - OpenVZ (Virtuozzo)

Виртуализация и контейнеры

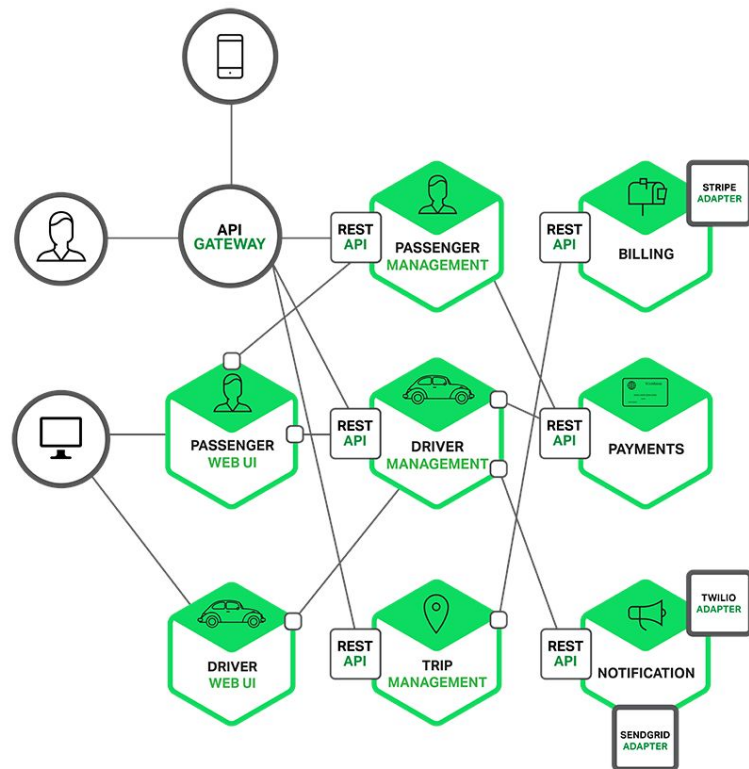
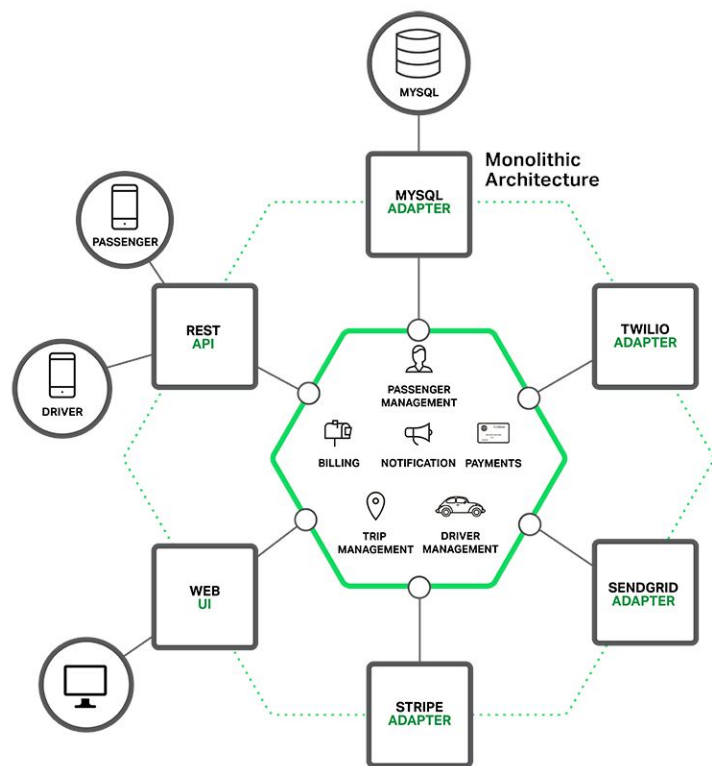
Container



Virtual Machines



Микросервисная архитектура



Средства изоляции

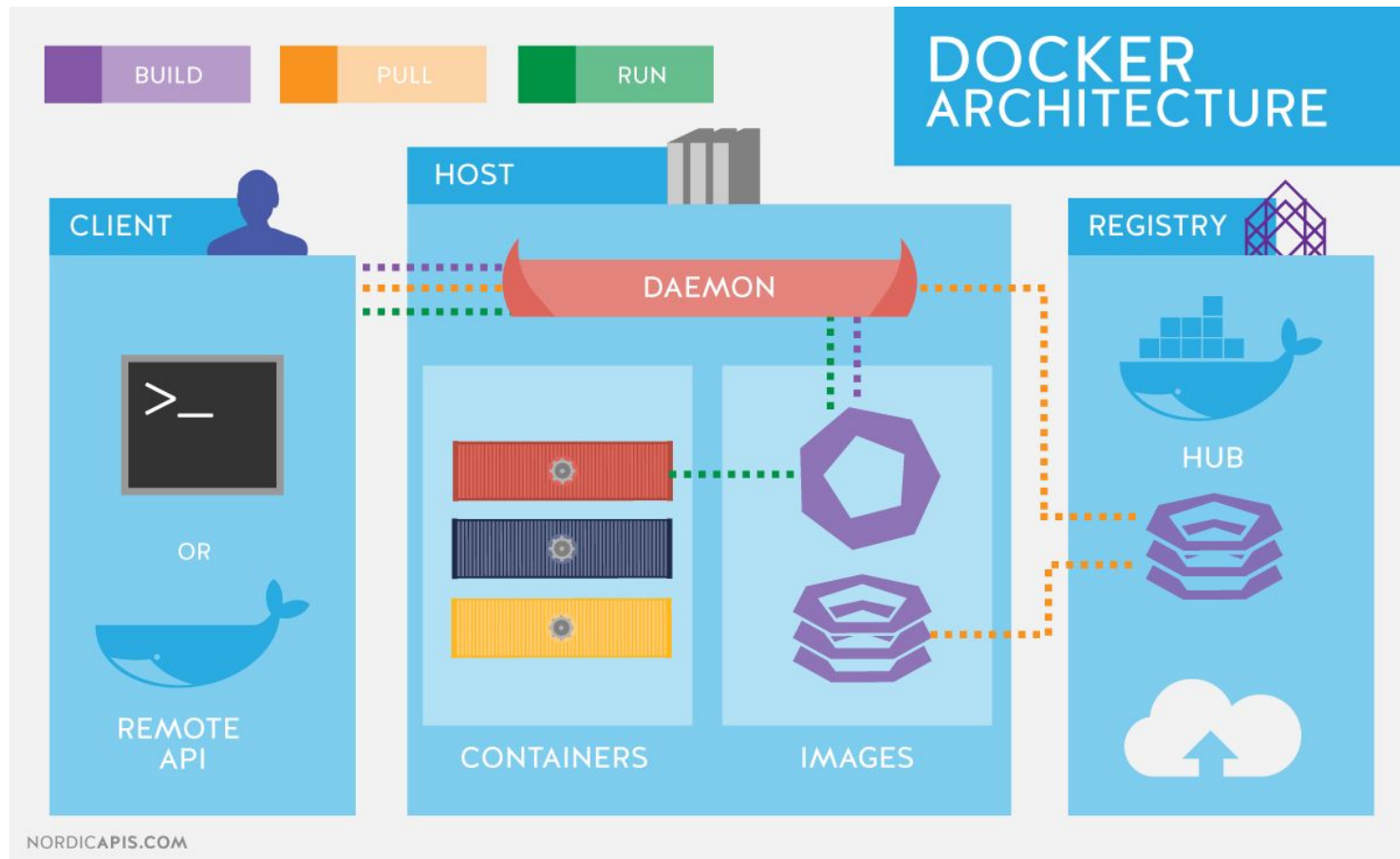
- Пространства имён (**namespaces**)
 - PID PID процессов
 - NETWORK Сетевые устройства, стеки, порты и т.п.
 - USER ID пользователей и групп
 - MOUNT Точки монтирования
 - IPC System V IPC, очереди сообщений POSIX
 - UTS Имя хоста и доменное имя NIS
- Контрольные группы (**cgroups**) ограничение ресурсов
 - использование памяти, в том числе виртуальной
 - приоритизацию: разным группам можно выделить разное количество процессорного ресурса
 - пропускной способности подсистемы ввода-вывода

Выбор: VM или контейнеры?

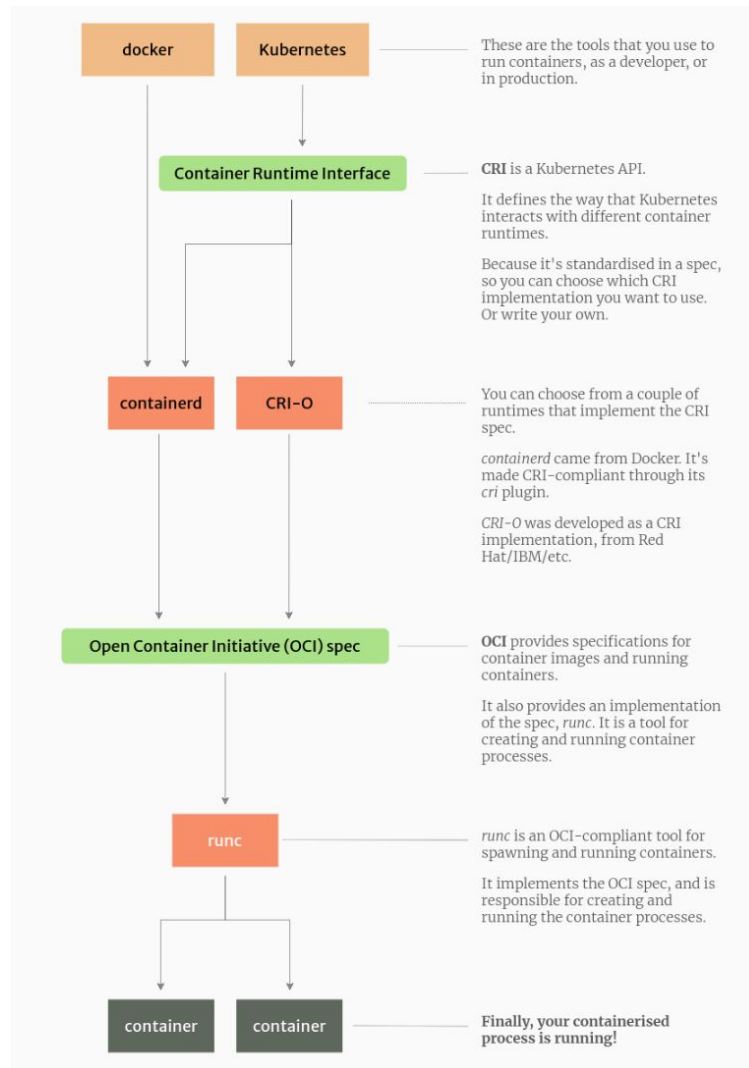
- Виртуализация
 - Полная свобода выбора ядер, модулей ядра
 - Эмуляция железа (сетевые устройства, контроллеры и т.д.)
 - Разнородные ОС (Windows, Linux, FreeBSD на одном хосте)
 - Полная изоляция гостевых систем
 - Высокие накладные расходы
- Контейнеры
 - Одно ядро (хостовое)
 - Изоляция на уровне окружения (библиотеки, утилиты, конфиги)
 - Быстрый запуск
 - Низкие накладные расходы

Архитектура Docker

Архитектура Docker



Docker



Компоненты Docker

- Сервер (демон `dockerd`)
- Клиент (команда `docker`). Команды по операциям с образами и контейнерами
- Клиент и сервер взаимодействуют по REST API

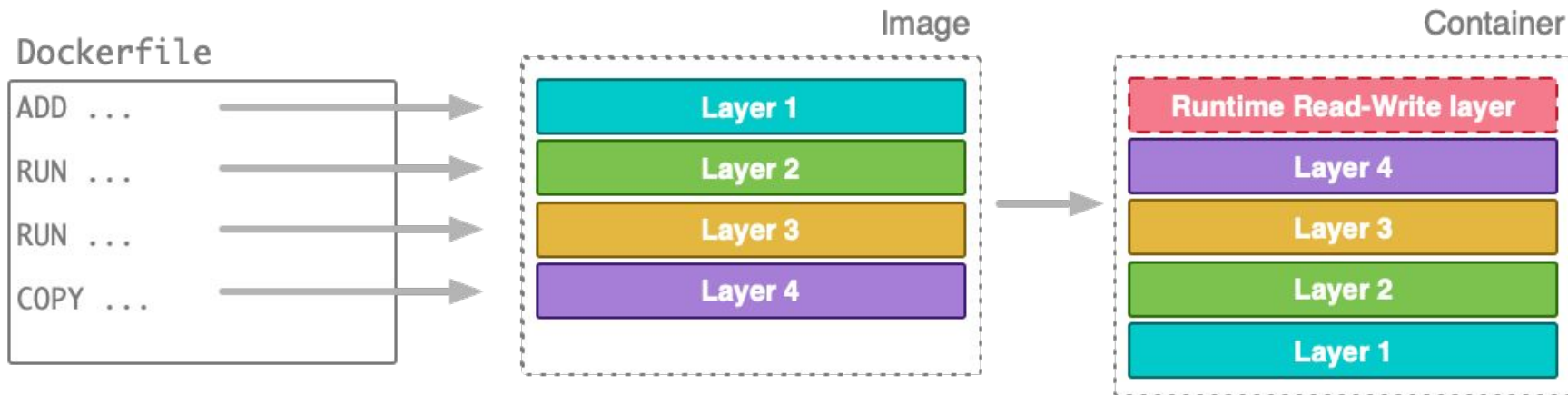
Объекты Docker

- Образы (images) – дистрибутивы для контейнеров
- Контейнеры (containers) – развёрнутые, рабочие образы
- Сети (networks) – передача данных
- Тома (volumes) – хранение данных

Образы Docker

- Готовы к использованию
- Есть открытый реестр (registry) Docker Hub <https://hub.docker.com/>
- Можно собирать собственные образы на базе существующих
- Состоят из слоёв (layers)

Слои образа docker



Управление контейнерами

Установка Docker в CentOS 7

Docker install CentOS



```
yum install -y yum-utils
yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo

yum install docker-ce docker-ce-cli containerd.io
```

<https://docs.docker.com/engine/install/centos/>



Базовые команды

docker commands



```
systemctl status docker
```

```
docker info
```

```
docker run hello-world
```

```
docker pull ubuntu:20.04
```

```
docker images
```

```
docker ps
```

```
docker ps -a
```



Запуск контейнера с Nginx

Detached
(демон)

Проброс портов
хост:контейнер

Имя образа
образ:тэг

```
docker run -d --name nginx1 -p 8090:80 -v /var/www/html:/usr/share/nginx/html nginx
```

Создать и
запустить
контейнер

Подключить том
(volume)
хост:контейнер

Сети для Docker

Docker networks

— □ ×

```
docker network create mynet
```

```
docker run -d --name nginx2 --network=mynet -p 80:80 nginx
```

```
docker run -d --name nginx2 --network=host -p 80:80 nginx
```

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



Сборка образа Dockerfile



Сборка образа из Dockerfile

Dockerfile

— □ ×

```
FROM nginx:latest  
COPY ./index.html /usr/share/nginx/html/index.html
```

Docker build

— □ ×

```
docker build -t webserver .  
  
docker run -it --rm -d -p 8080:80 --name web webserver
```

Пример Dockerfile

```
Dockerfile

FROM ubuntu:20.04

RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends nano webp brotli
&& which cron && rm -rf /etc/cron.*/

COPY entrypoint.sh /entrypoint.sh

COPY mozjpeg.deb /root/mozjpeg.deb

RUN cd /root && dpkg -i mozjpeg.deb

COPY --chown=www-data:www-data ./multisite /home/httpd/multisite

COPY ect /usr/bin/

COPY avifenc /usr/bin/

ENTRYPOINT ["/entrypoint.sh"]
CMD ["cron", "-f", "-l", "2"]
```

Пример запуска (общая сеть)

Запуск двух контейнеров в одной сети

Docker multiple

— □ ×

```
docker network create some-network
```

```
docker run --rm --network some-network --name some-mariadb -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mariadb:10.5
```

```
docker run -it --network some-network --rm mariadb:10.5 mysql -hsome-mariadb -uroot -p
```

```
docker run --name myadmin -d --network some-network --link some-mariadb:db -p 8080:80 phpmyadmin
```



Практика

Домашнее задание

1. Установить Docker
2. Запустить контейнер Nginx
3. Настроить схему с балансировкой (FE - BE), где Front-end это контейнер с Nginx
4. Прислать отчет по командам создания и запуска контейнеров с конфигами



Сроки выполнения: указаны в личном кабинете



Что мы изучили?

Подведем итоги

1. Принципы контейнеризации
2. Отличия от виртуализации
3. Базовые команды по управлению контейнерами в Docker
4. Создание образов в Docker

Список материалов для изучения

1. <https://docs.docker.com/engine/reference/run/>
2. <https://selectel.ru/blog/mexanizmy-kontejnerizacii-namespaces/>
3. <https://www.katacoda.com/courses/container-runtimes/what-is-a-container-image>
4. <https://www.katacoda.com/courses/container-runtimes/what-is-a-container>
5. <https://habr.com/ru/post/310460/>
6. <https://habr.com/ru/company/southbridge/blog/528206/>
7. <https://docs.docker.com/engine/faq/>
8. <https://training.play-with-docker.com/>
9. <https://labs.play-with-docker.com/>

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Рефлексия

Цели вебинара

Проверка достижения целей

1. Разобрали, что такое контейнеризация.
2. Установили Docker.
3. Научились управлять контейнерами.



Вопросы для проверки

1. Чем контейнеризация отличается от виртуализации?
2. Какие системы контейнеризации вы знаете?
3. Какая команда создаёт контейнер?

Рефлексия



Что было самым полезным на занятии?



Как будете применять на практике то, что узнали на вебинаре?

Следующий вебинар



Git



Ссылка на вебинар
будет в ЛК за 15 минут



Материалы
к занятию в ЛК —
можно изучать



Обязательный материал
обозначен красной
лентой

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Лавлинский Николай

Технический директор “Метод Лаб”

<https://www.methodlab.ru/>

<https://www.youtube.com/c/NickLavlinsky>

<https://vk.com/nick.lavlinsky>

