# Промышленное машинное обучение на Spark

## Лекция 1: Docker

7 октября 2023 г.

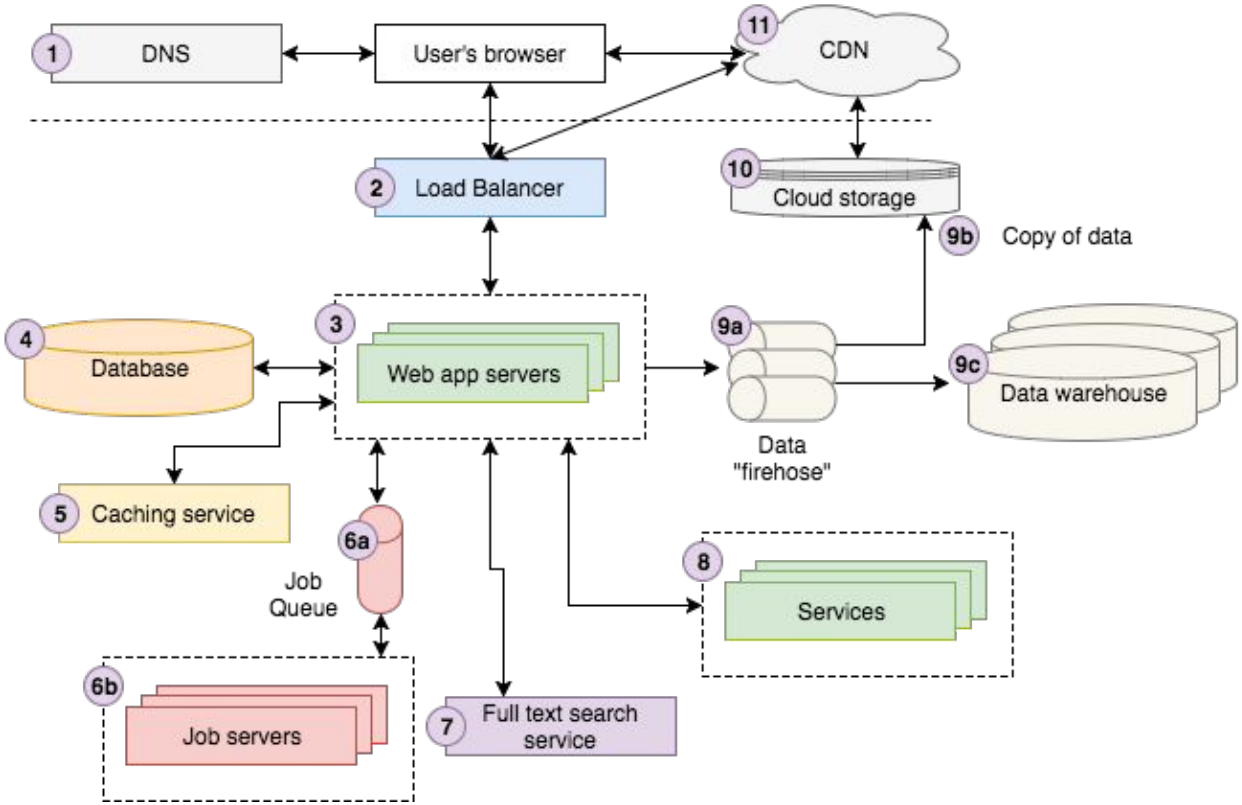
# Содержание курса

1. Оборачиваем модель в сервис: Docker. Flask.
2. Оборачиваем модель в сервис: Requests. REST API.
3. Погружение в среду Spark. Распределённые вычисления. MapReduce.
4. Распределённое хранение данных. Форматы хранения данных. Погружение в среду Spark.
5. Генерация признаков. Spark feature engneering.
6. Распределённое обучение моделей. Spark ML
7. Обработка и хранение текстовых данных и картинок. Spark image processing. Spark NLP.
8. Обработка потоковых данных. Spark Streaming.

# Поговорим о …

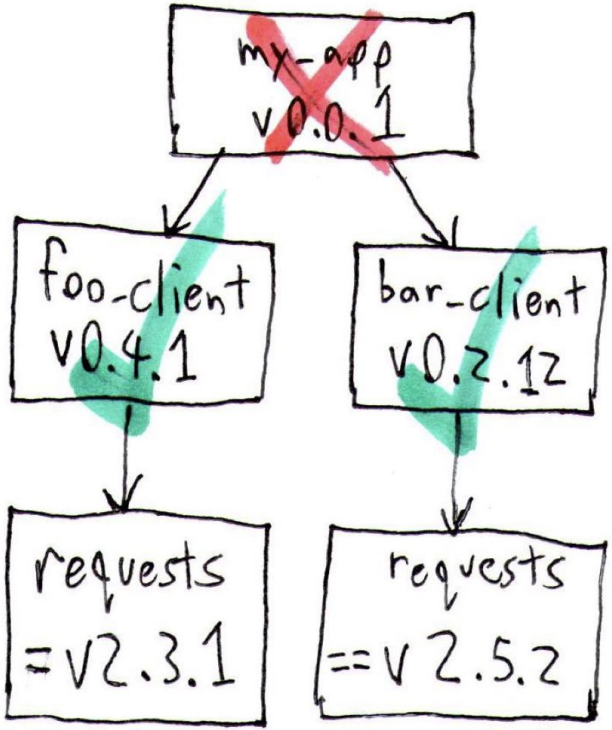
1. Какие задачи возникают при разработке сервисов
2. Зачем нужна контейнеризация приложений и какие задачи решает
3. Что такое Docker и из каких сущностей он состоит
4. Какое ещё существует ПО помимо Docker

# Типичный портрет современного сервиса



# Проблема 1. Dependency hell

Разные компоненты разрабатываемой системы могут требовать разные версии сторонних библиотек



# Проблема 2. Non-safety execution

Поломка или ошибки
одного компонента системы
может привести к отказу
всего сервиса

# Проблема 3. Сложный деплой

Не всегда коллеги из команды DevOps понимают как запустить ваш сервис в продакшен среде

```
> Firstly download libraries lib1 & lib2
> Prepare data file in csv format
> Rename prepared filed to dataset.csv
> Next copy main.py to src folder
> Set params at config file
....
> Now run you can run service!!!
```

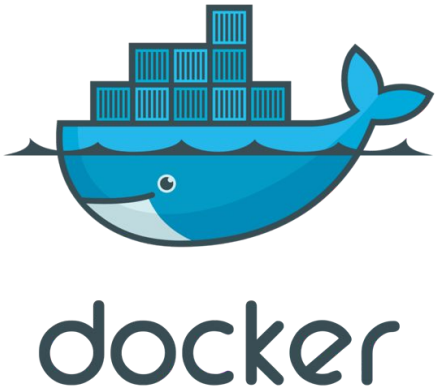
# Контейнеризация

# Принципы контейнеризации

1. Изоляция объектов системы друг от друга
2. Доставка объекта со всеми необходимыми компонентами
3. Универсальный интерфейс работы с контейнером
4. Независимость от способа доставки

# Технология Docker

Docker - набор программных инструментов  с открытым исходным кодом, решающих задачи контейнеризации, автоматического внедрения, запуска и эксплуатации приложений.

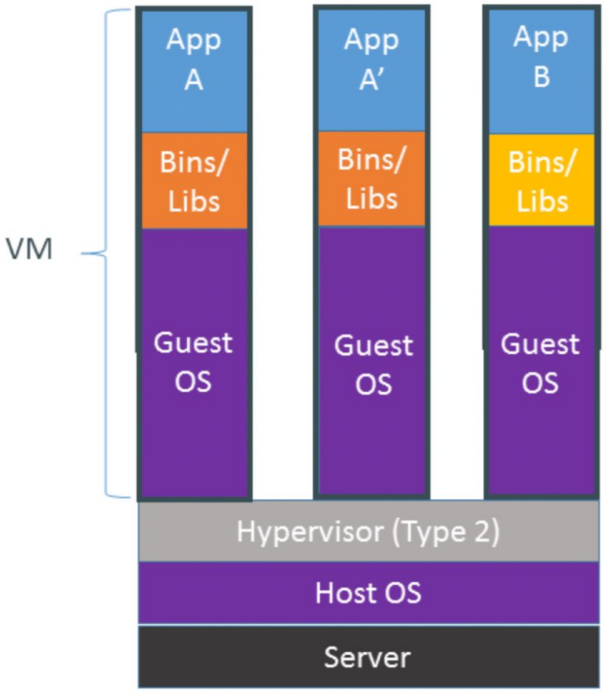


# Преимущества docker

- Платформонезависимость
- Абстрагирование частей приложения друг от друга
- Универсальный интерфейс работы с приложением
- Упрощения построения отказоустойчивых систем
- Упрощение масштабирования системы
- Автоматизация рутинных задач: тестирование, CI/CD

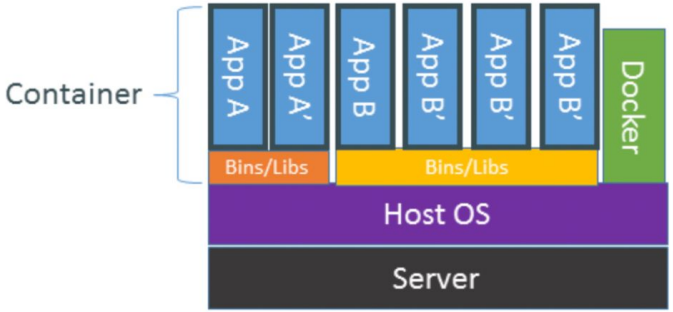
# Virtual machines & docker

Зачем нужно было изобретать docker, когда уже существовали виртуальные машины?




Containers are isolated, but share OS and, where appropriate, bins/libraries

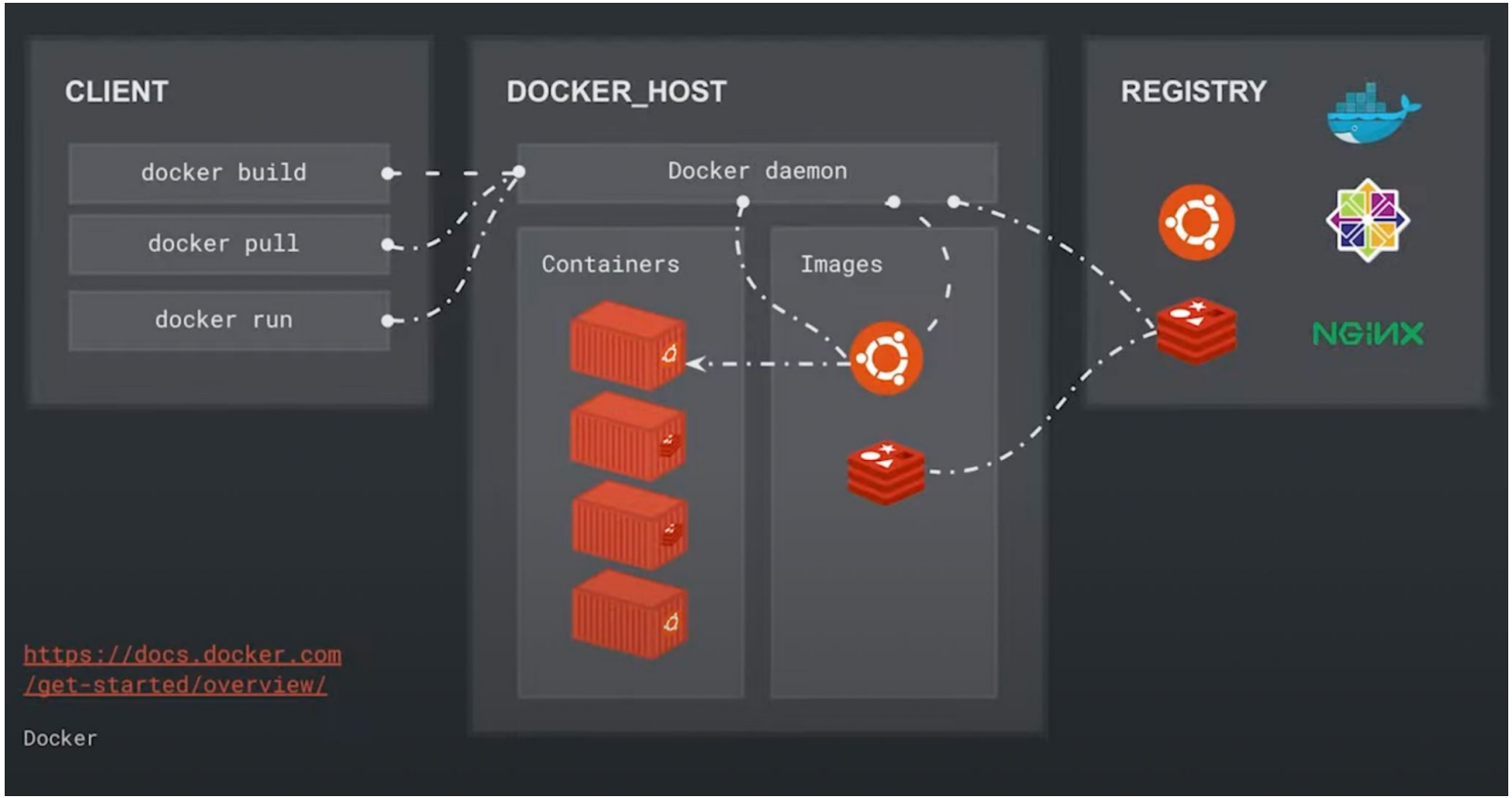
...result is significantly faster deployment, much less overhead, easier migration, faster restart



# Основные отличия docker от VM

- Docker не запускает отдельное ядро для каждого процесса по отдельности
- Контейнеры значительно быстрее запускаются
- VM требует больше ресурсов для запуска
- В VM сложно контролировать изменения, версионирование
- Контейнеры могут иметь разделяемые ресурсы между собой (бинарники, библиотеки)
- При изменении контейнера будет сохраняться только разница между исходником и новой версией
- VM имеют более надёжную изоляцию между собой
- VM чуть проще в использовании

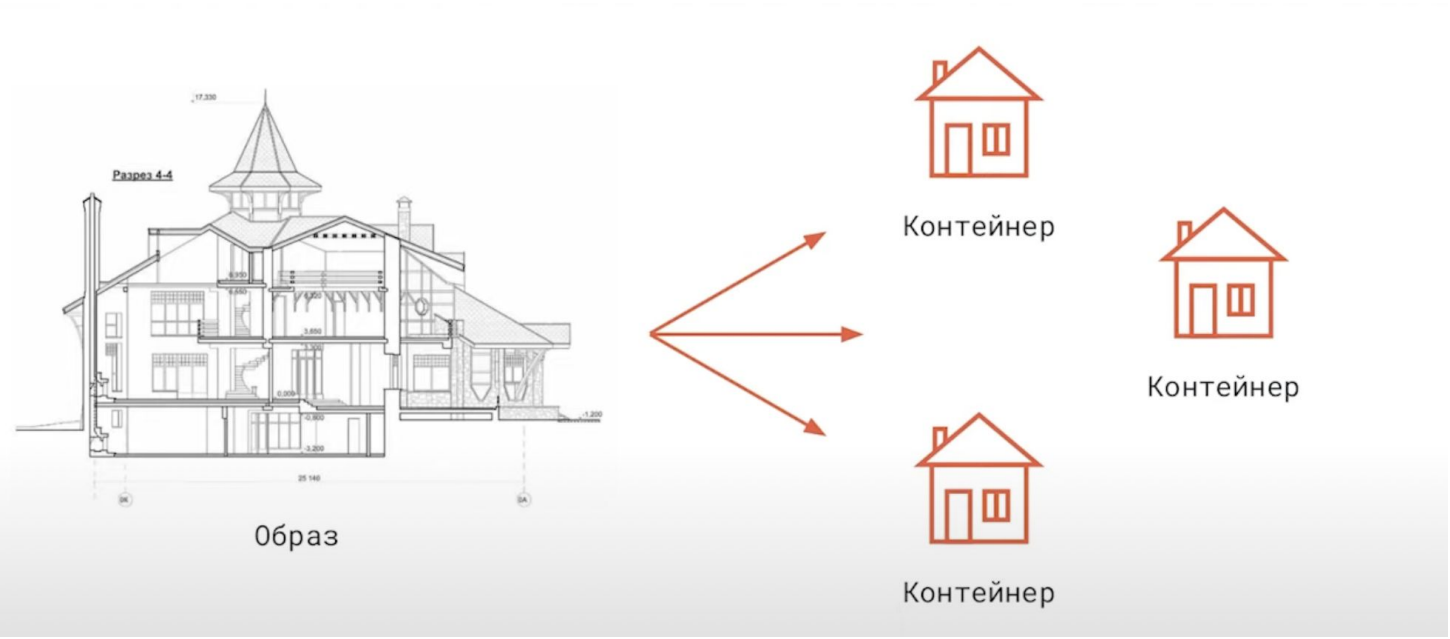
# Основные компоненты технологии



# Функции каждого компонента

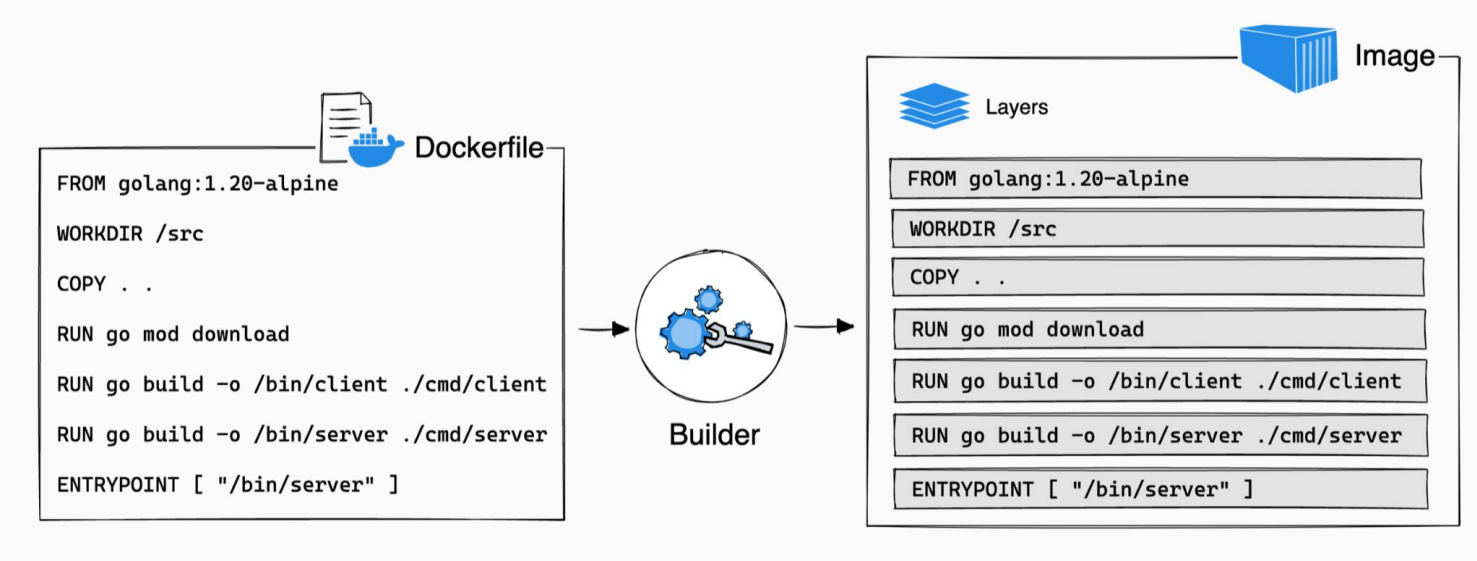
- Client - программа, из которой поступают команды
- Host - физическая машина, на которой располагаются контейнеры
- Docker daemon - компонент, который считывает команды клиента и управляет всеми объектами
- Images - инструкция того, каким должен быть контейнер
- Containers - сущность, создаваемая после запуска образа
- Registry - сервер, на котором хранятся готовые образы

# Отношение между образом и контейнером



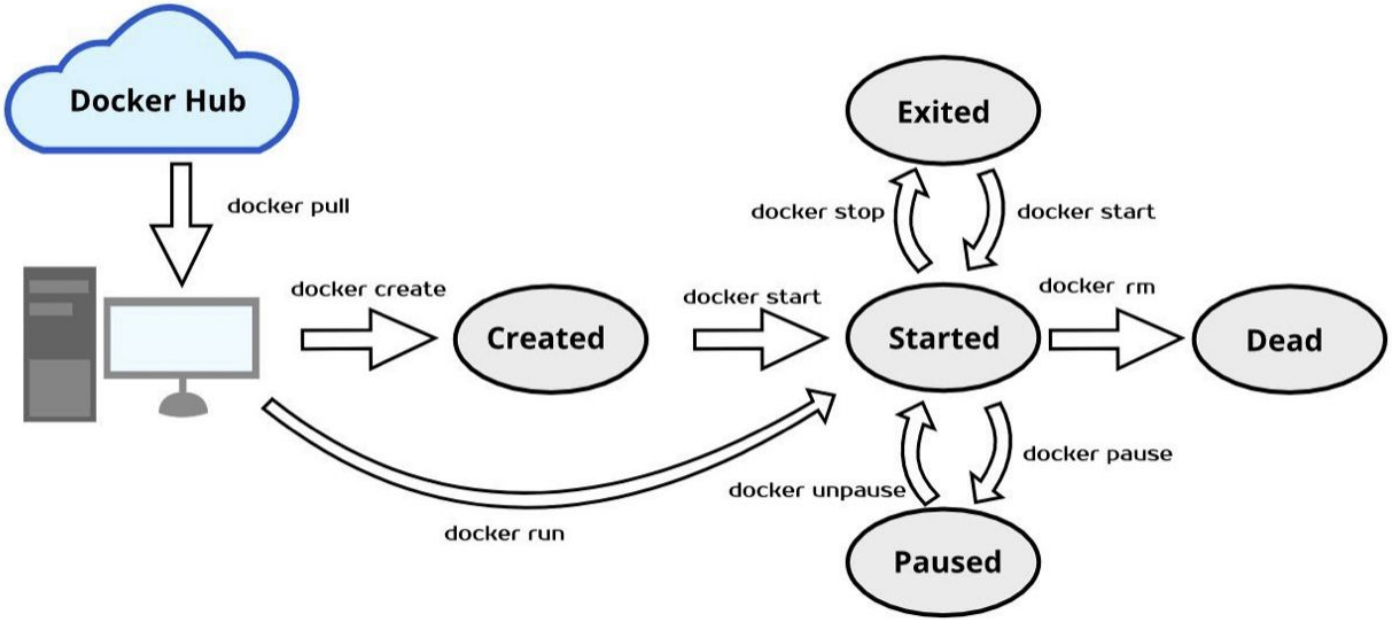

- Образ - план того, каким должен быть контейнер
- Контейнер -  конкретная реализация, построенная на основе образа

# Слои образа




- **Dockerfile -** файл с описанием процедуры создания файловой системы контейнера
- **Layer** - атомарный набор изменений файловой системы, как правило, описывается командой в Dockerfile

# Жизненный цикл контейнера



# Какие есть альтернативы?



# Что ещё почитать

1. https://docs.docker.com/get-started/
2. https://karpov.courses/docker
3. https://cloudacademy.com/blog/docker-vs-virtual-machines-differences-you-should-know/