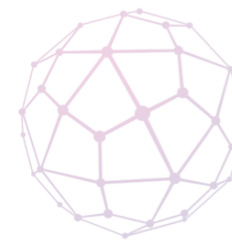
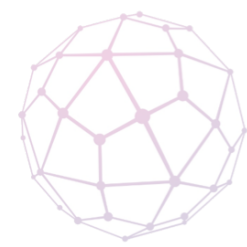


# Контейнерная Виртуализация

Антоненко Виталий

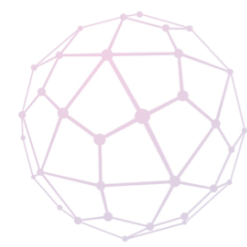


# docker



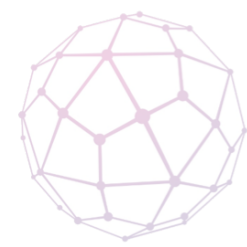
# История DOCKER .....

- A dotCloud (PAAS provider) project
- Первый коммит January 18, 2013
- Docker 0.1.0 March 25, 2013
- 18,600+ github stars, 3800+ forks, 740 Contributors....  
and continues



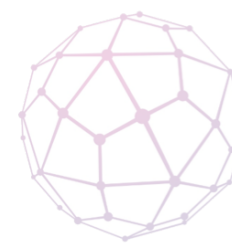
# Что такое Docker?

- Docker — это открытая платформа для разработки, доставки и эксплуатации приложений.
- отделить ваше приложение от вашей инфраструктуры
- позволяет запускать практически любое приложение, безопасно изолированное в контейнере



# Когда Docker полезен?

- упаковывание приложения (и так же используемых компонент) в docker контейнеры;
- раздача и доставка этих контейнеров вашим командам для разработки и тестирования;
- размещение контейнеров на серверах, как в дата центры так и в облака.



Multiplicity of Stacks

**Static website**  
nginx 1.5 + modsecurity + openssl + bootstrap 2

**User DB**  
postgresql + pgv8 + v8

**Queue**  
Redis + redis-sentinel

**Analytics DB**  
hadoop + hive + thrift + OpenJDK

**Background workers**  
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

**Web frontend**  
Ruby + Rails + sass + Unicorn

**API endpoint**  
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

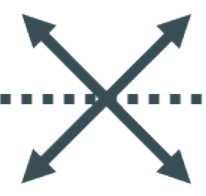
Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

QA server

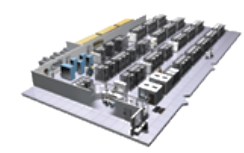
Customer Data Center



Public Cloud

Disaster recovery

Production Servers



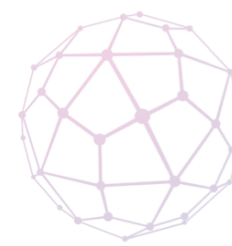
Production Cluster



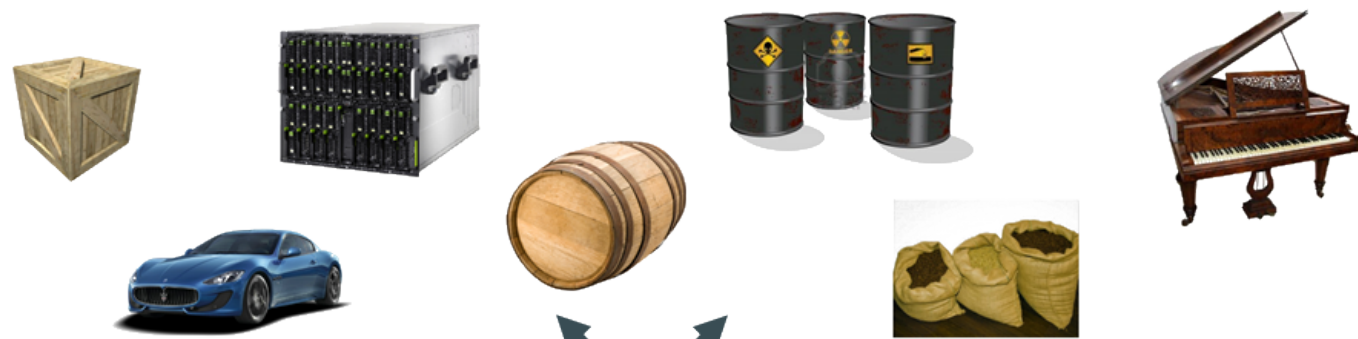
Contributor's laptop

Can I migrate smoothly and quickly?





Multiplicity of Goods

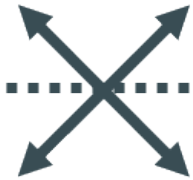


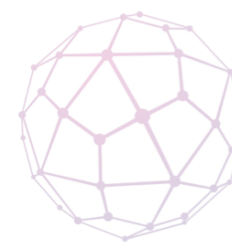
Do I worry about how goods interact (e.g. coffee beans next to spices)

Multiplicity of methods for transporting/storing



Can I transport quickly and smoothly (e.g. from boat to train to truck)





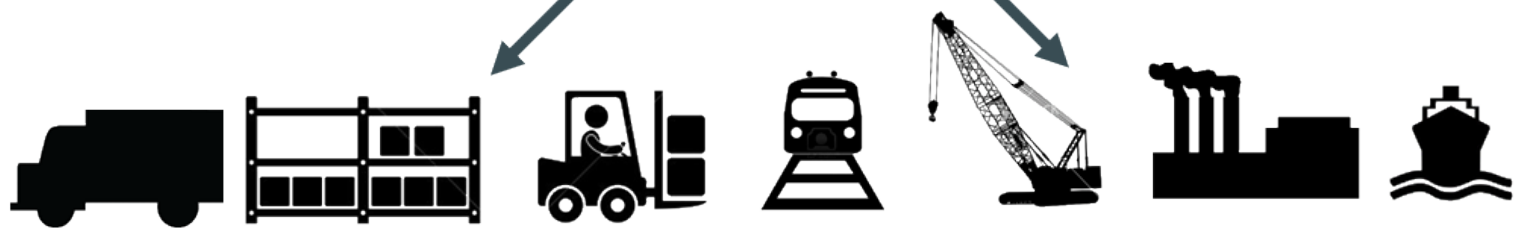
Multiplicity of Goods



A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

Do I worry about how goods interact (e.g. coffee beans next to spices)

Multiplicity of methods for transporting/storing

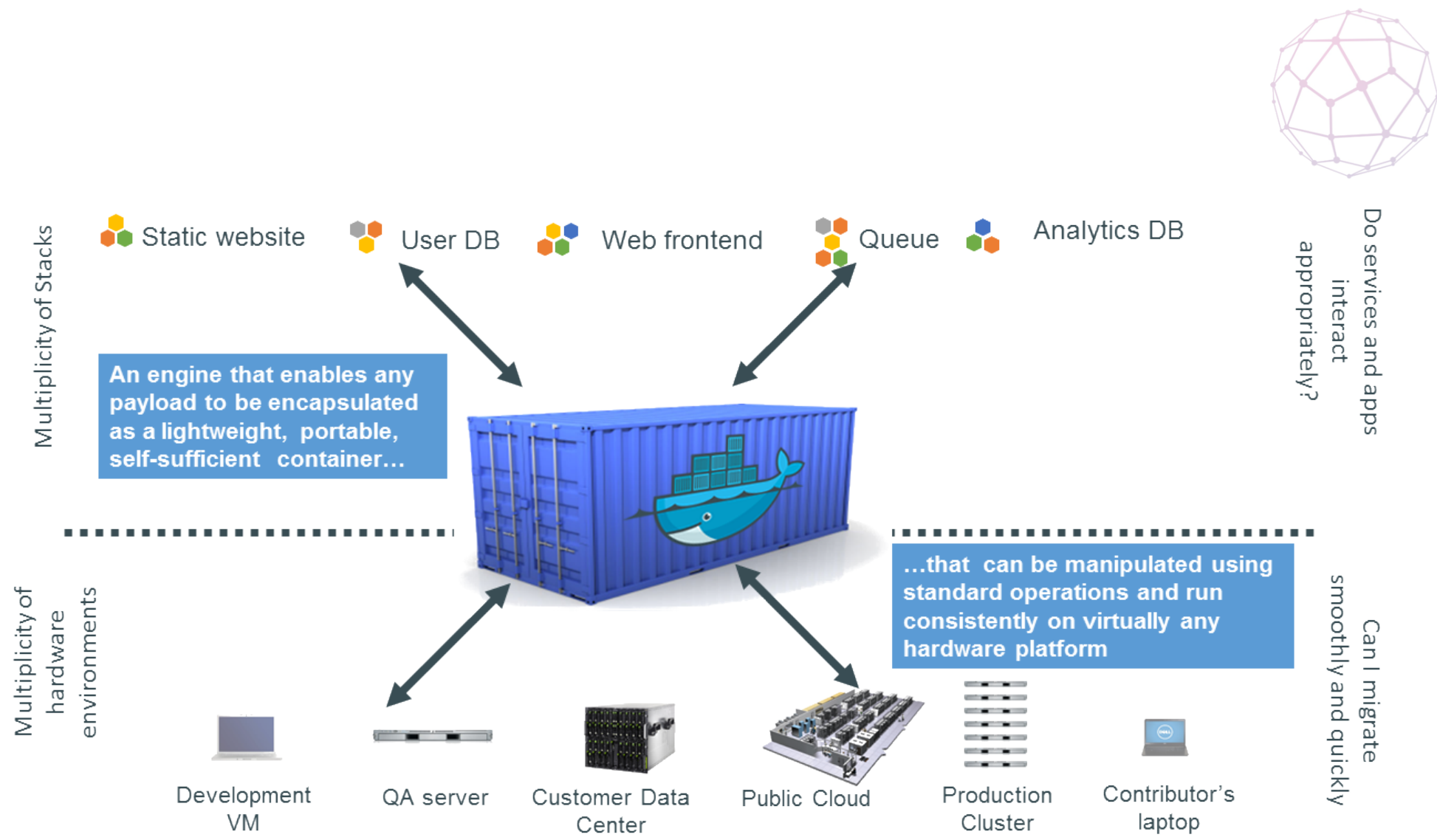


...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

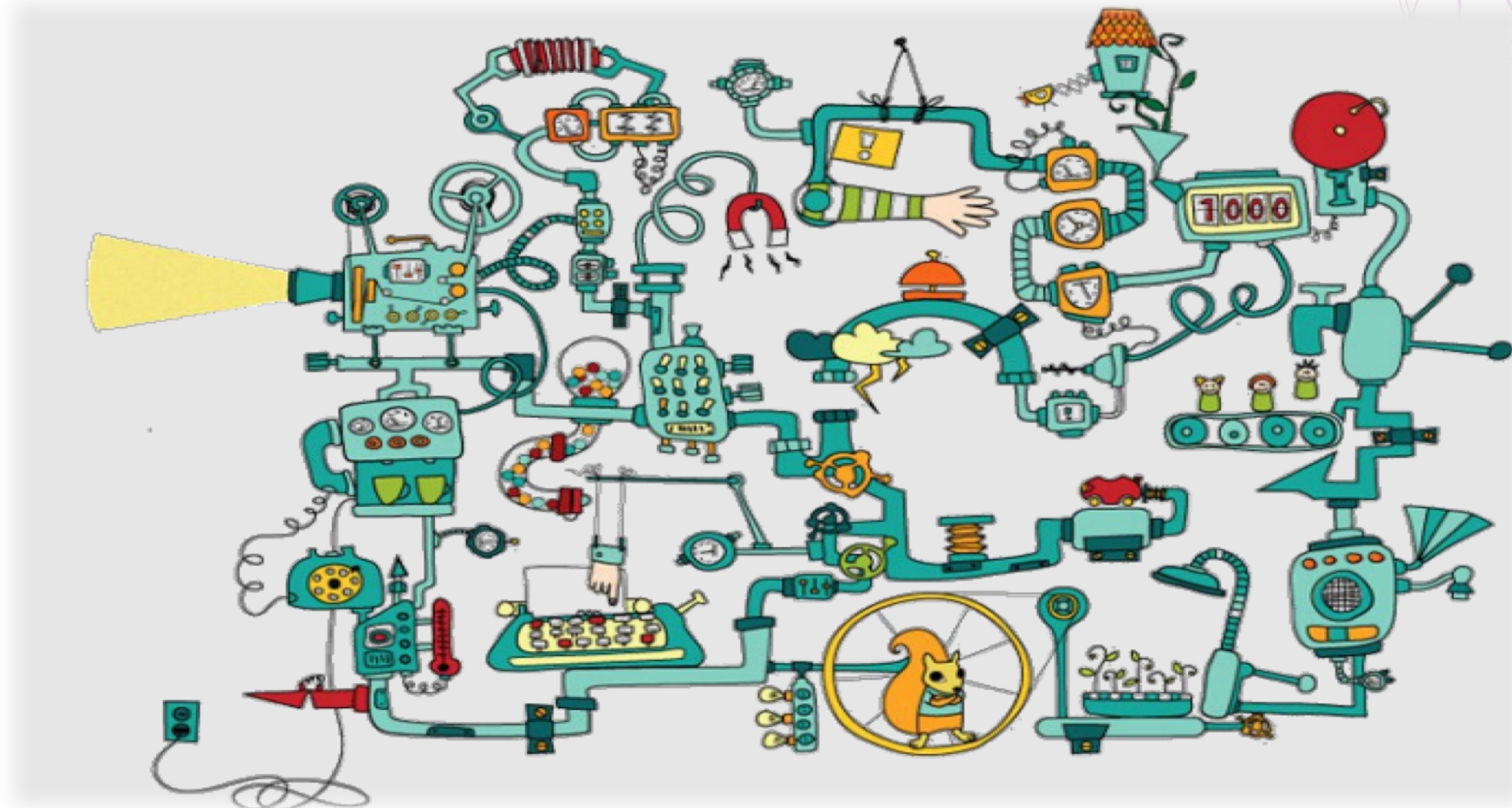
Can I transport quickly and smoothly (e.g. from boat to train to truck)



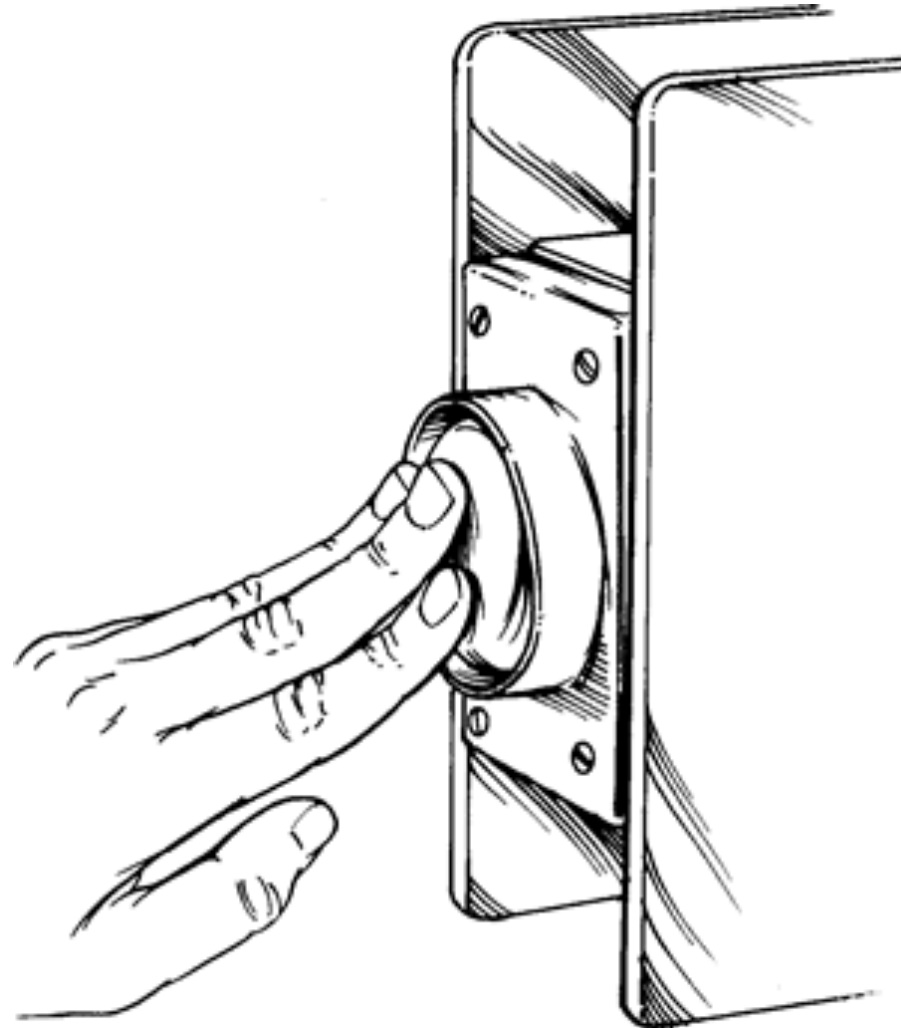
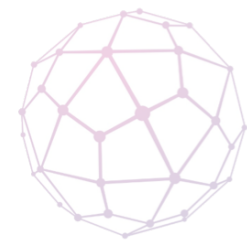




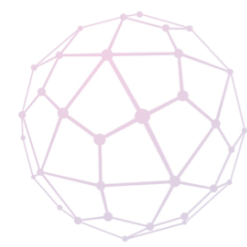
# Контейнеры до Docker



# Контейнеры после Docker ....

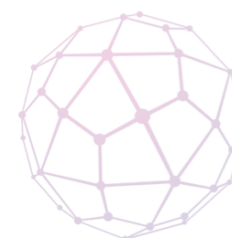


Виртуализация и Облачные Вычисления, Лекция по  
Контейнерам

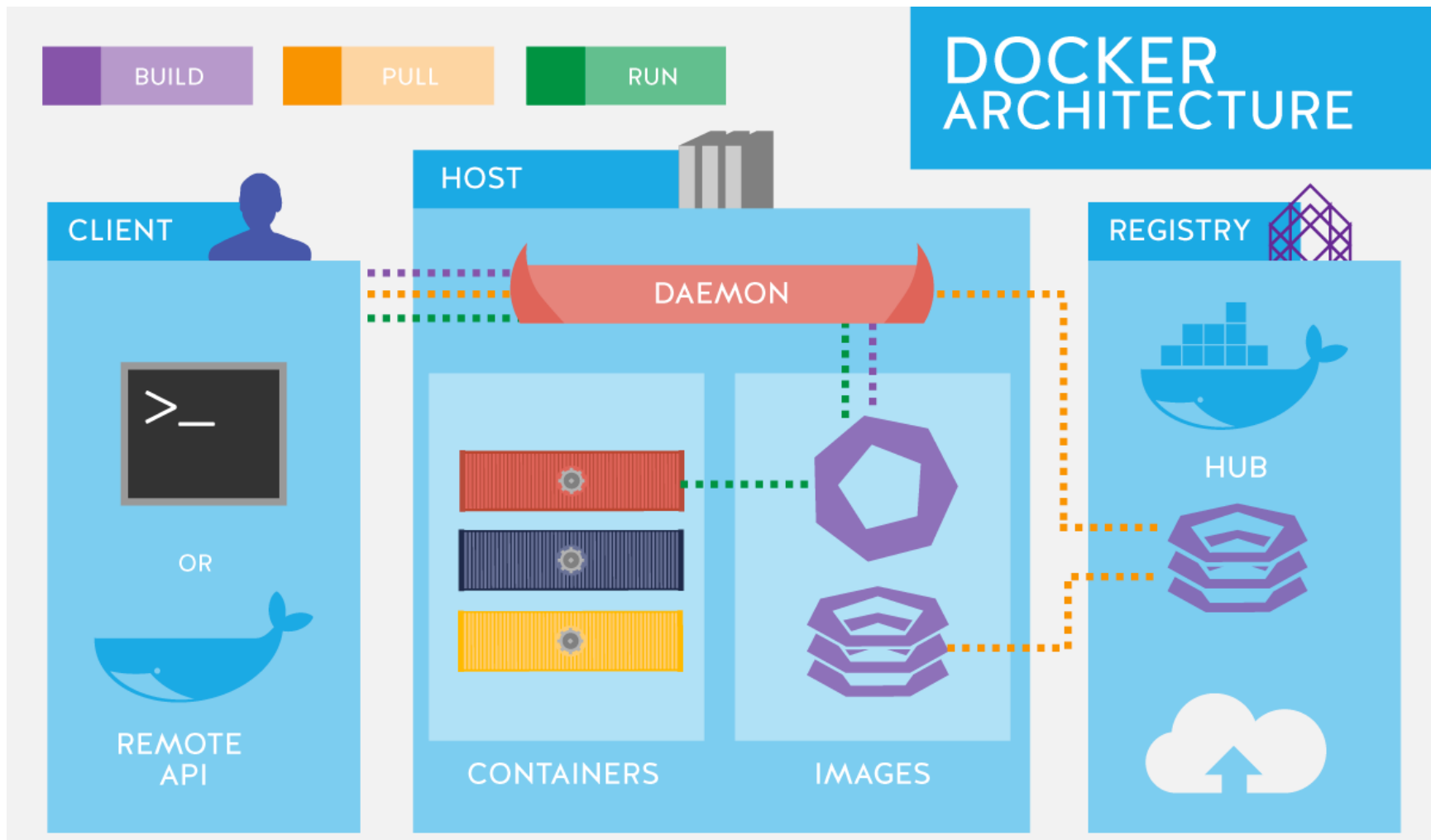


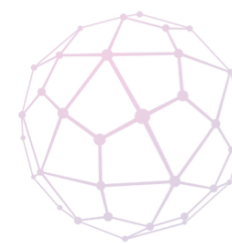
# Из чего состоит Docker?

- Docker: платформа виртуализации с открытым кодом;
- Docker Hub: платформа-как-сервис для распространения и управления docker контейнерами.



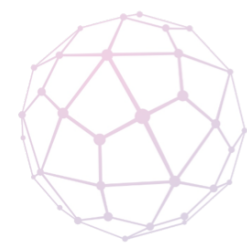
# Архитектура Docker





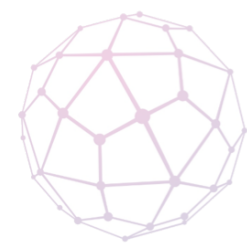
# Внутри docker-a

- образы (images)
- реестр (registries)
- контейнеры



# Как работает Docker?

- Каждый образ состоит из набора уровней.
- Docker использует [union file system](#) для сочетания этих уровней в один образ.
- В основе каждого образа находится базовый образ.
- использовать образы как базу для создания новых образов.

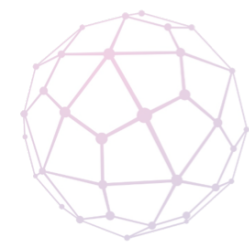


# Как создать Docker образ?

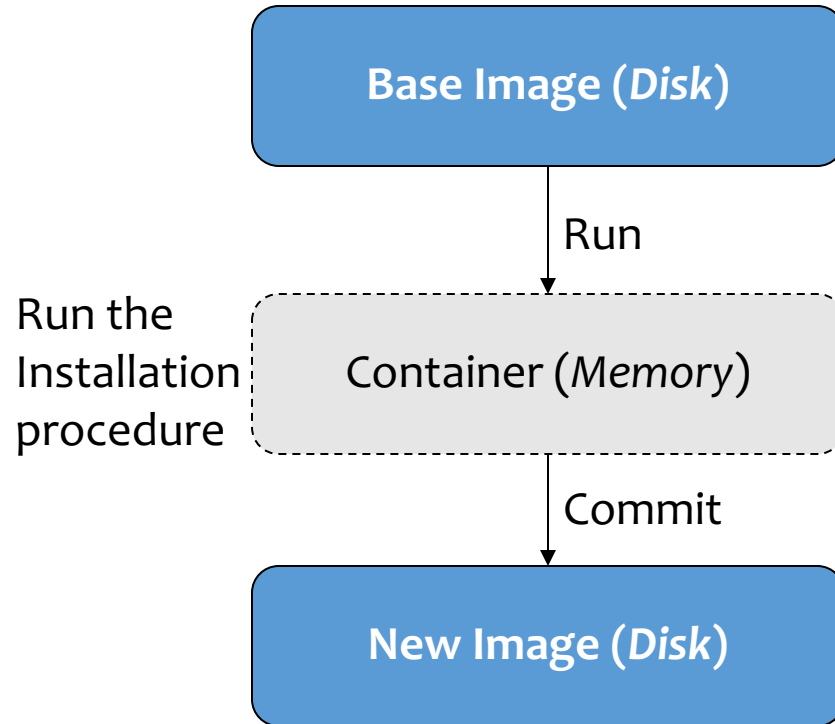
- запуск команды
- добавление файла или директории
- создание переменной окружения
- указания что запускать, когда запускается контейнер этого образа
  
- инструкции хранятся в файле Dockerfile. Docker считывает это Dockerfile выполняет эти инструкции, и возвращает конечный образ.



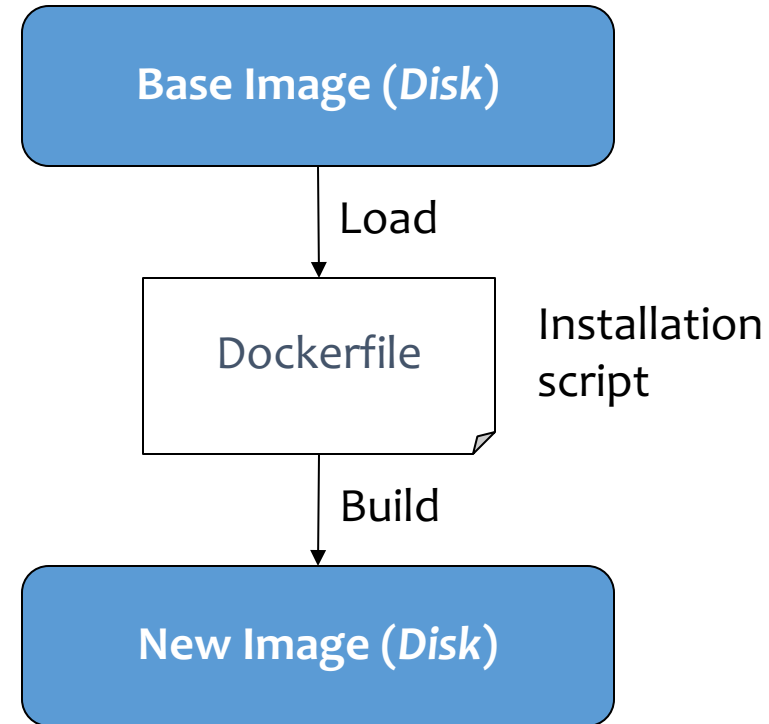
# Как создать Docker образ? (2)

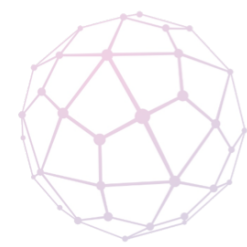


## Interactive building



## Building from a Docker File

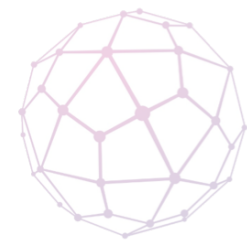




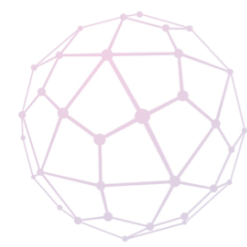
# Как работает контейнер?

- состоит из операционной системы, пользовательских файлов и метаданных.
- Docker образ доступен только для чтения.
- Когда docker запускает контейнер, он создает уровень для чтения/записи сверху образа

# Как запустить простой контейнер?

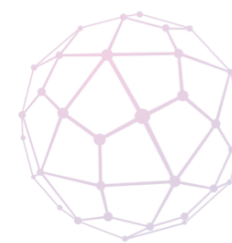


- `$ sudo docker run -i -t ubuntu /bin/bash`



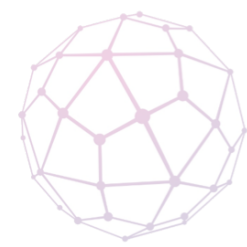
# Docker Workflow

- скачивает образ
- создает контейнер
- инициализирует файловую систему и монтирует read-only уровень
- инициализирует сеть/мост
- Установка IP адреса
- Запускает указанный процесс
- Обрабатывает и выдает вывод вашего приложения



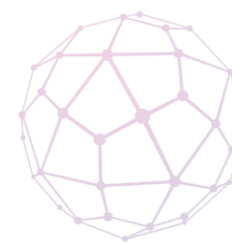
# Как он это делает?

- Пространство имен (namespaces)
- Control groups (контрольные группы)
- Union File System



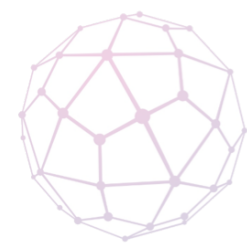
# Пространство имен(namespaces)

- **pid:** для изоляции процесса;
- **net:** для управления сетевыми интерфейсами;
- **ipc:** для управления IPC ресурсами. (ICP: InterProcess Communication);
- **mnt:** для управления точками монтирования;
- **utc:** для изолирования ядра и контроля генерации версий(UTC: Unix timesharing system).



# Выводы

- Стейтлес
- Чистый (pure)
- Ленивый
- Декларативный
- Функциональный
- Строгий



Спасибо за внимание!  
Вопросы?

[anvial@lvk.cs.msu.su](mailto:anvial@lvk.cs.msu.su)

Антоненко Виталий