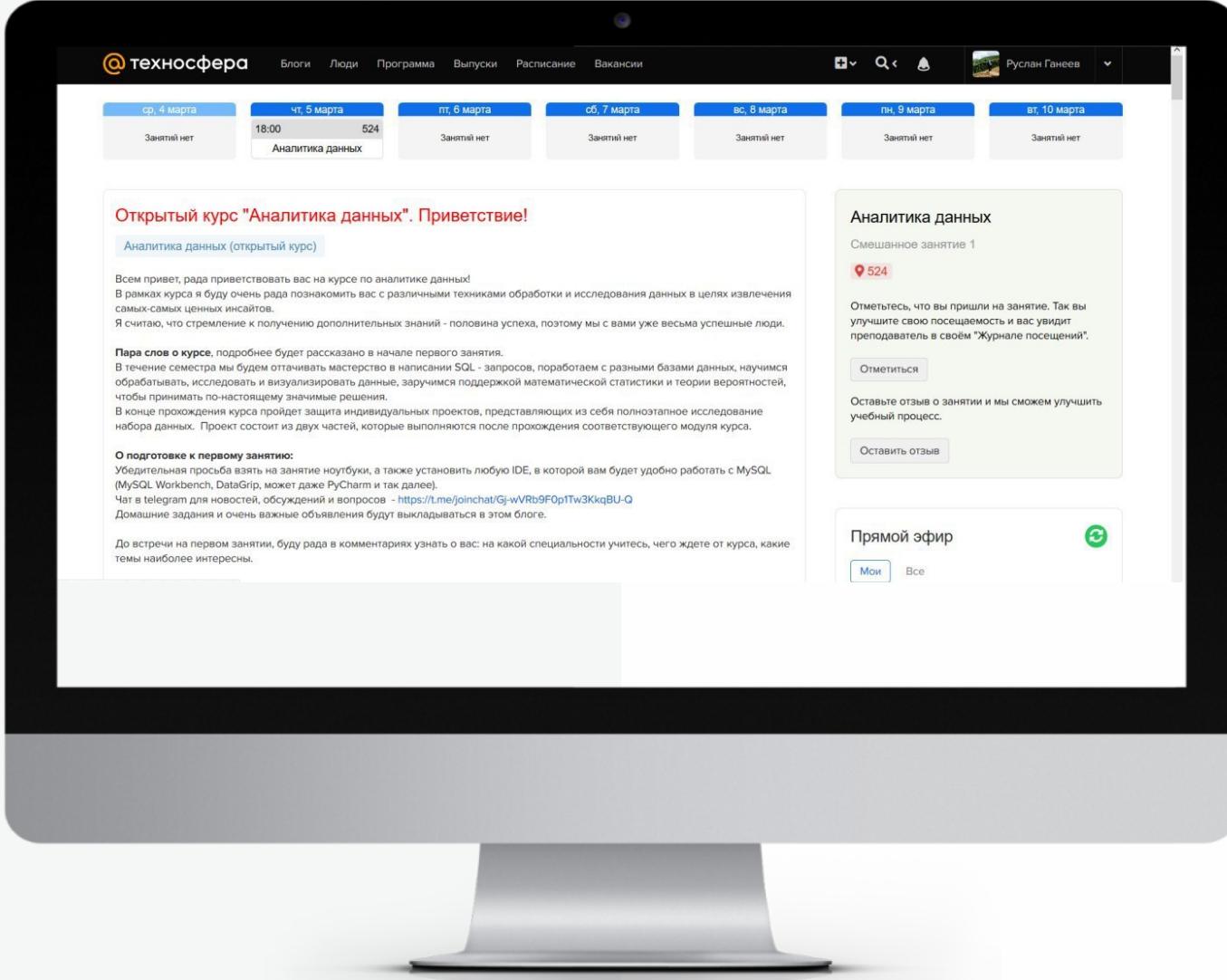


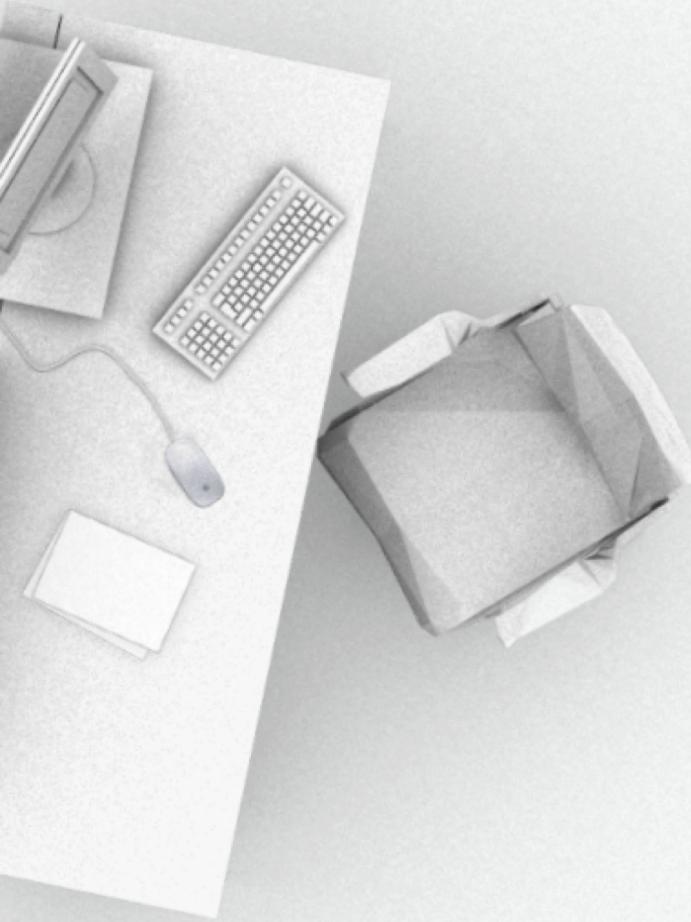
Автоматизированное тестирование на Python

Кириллов Илья





Не забудьте
отметиться на
портале

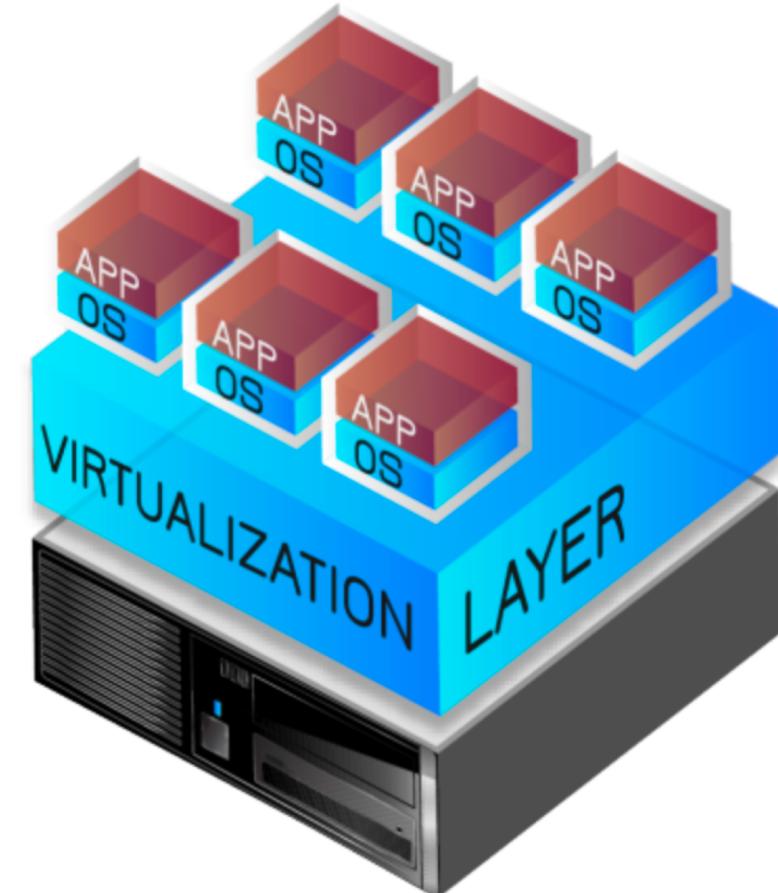


Docker

Виртуализация

Виртуализация — это создание изолированных окружений в рамках одного физического устройства

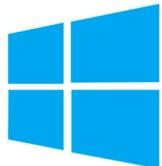
- хост-система (host)
- гостевая система (quest)



Hypervisor

Hypervisor — занимается созданием виртуальных машин и их управлением.

- аппаратная виртуализация
- программная виртуализация



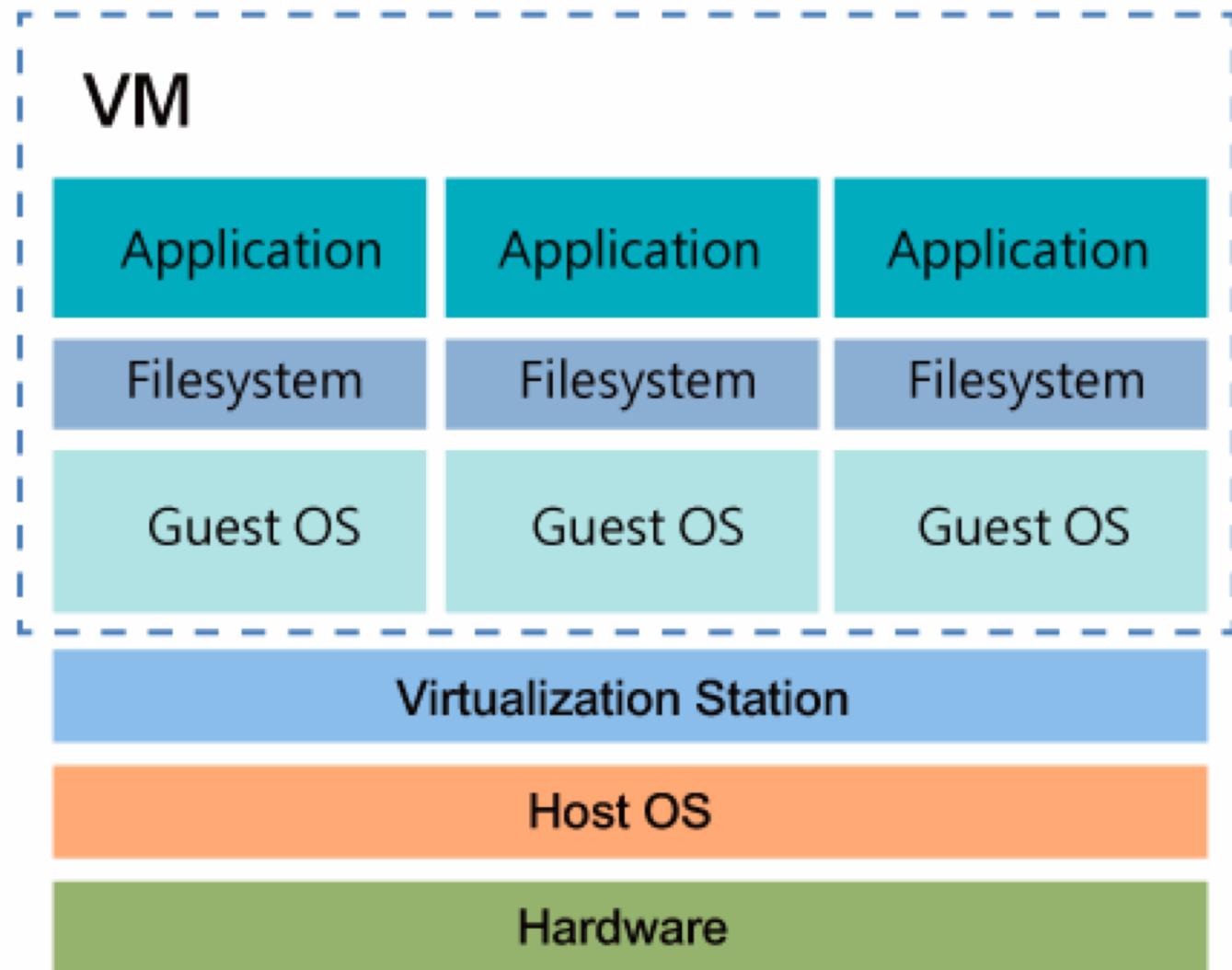
Microsoft
Hyper-V



|| Parallels®



Принцип работы

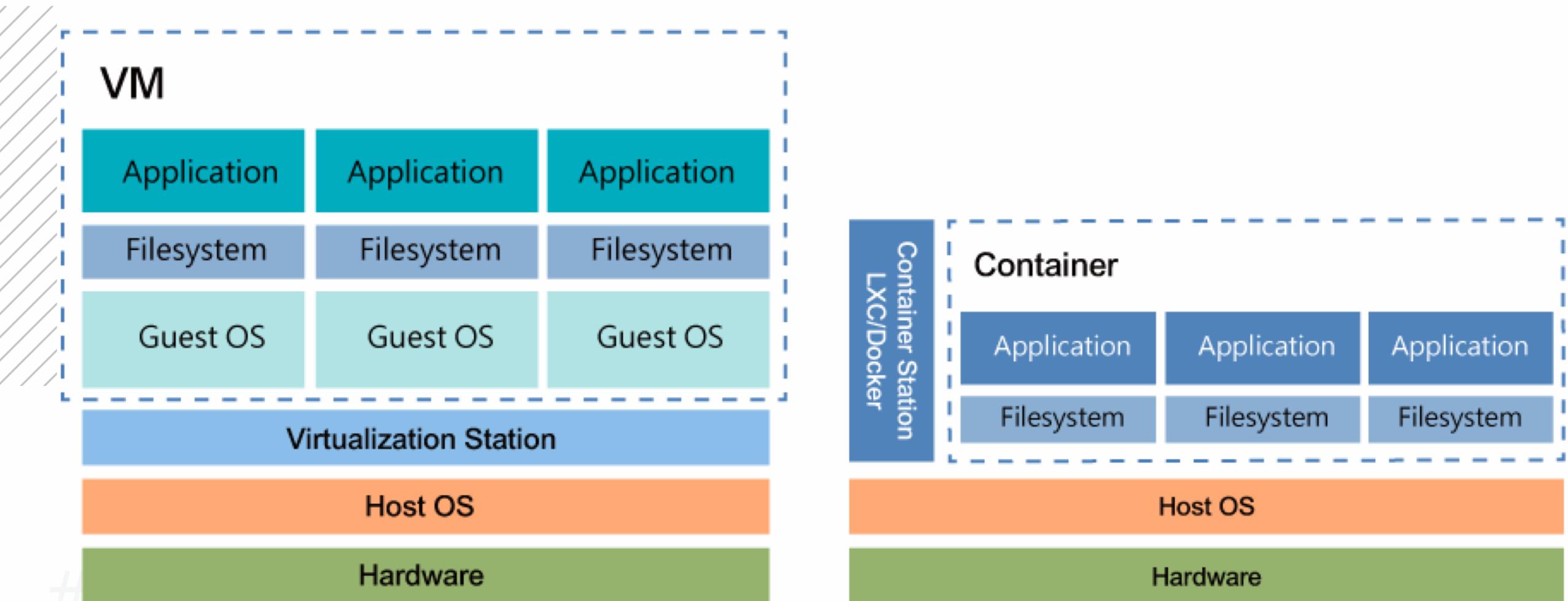


Контейнеризация

Контейнеризация – это легковесная виртуализация и изоляция ресурсов на уровне операционной системы linux

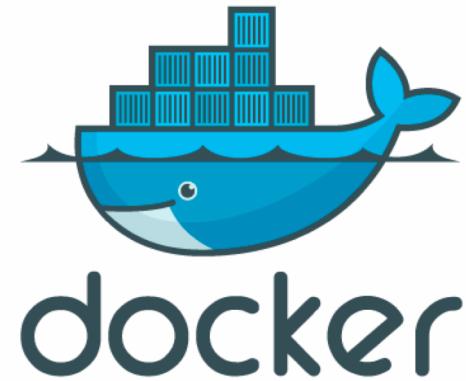
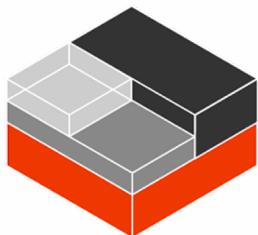
- namespace – механизм изоляции и группировки структур данных ядра
- control groups – механизм изоляции ресурсов ядра

Принцип работы



Немного выводов

- процессы используют ядро хост-машины
- большая производительность
- гибкость



Virtuozzo



WE HEARD

DOCKER

SAY

WERE

DOCK

IN THE FUTURE

EVERYTHING IS DOCKERIZED

GOT CONFUSED

DOCKER

#010

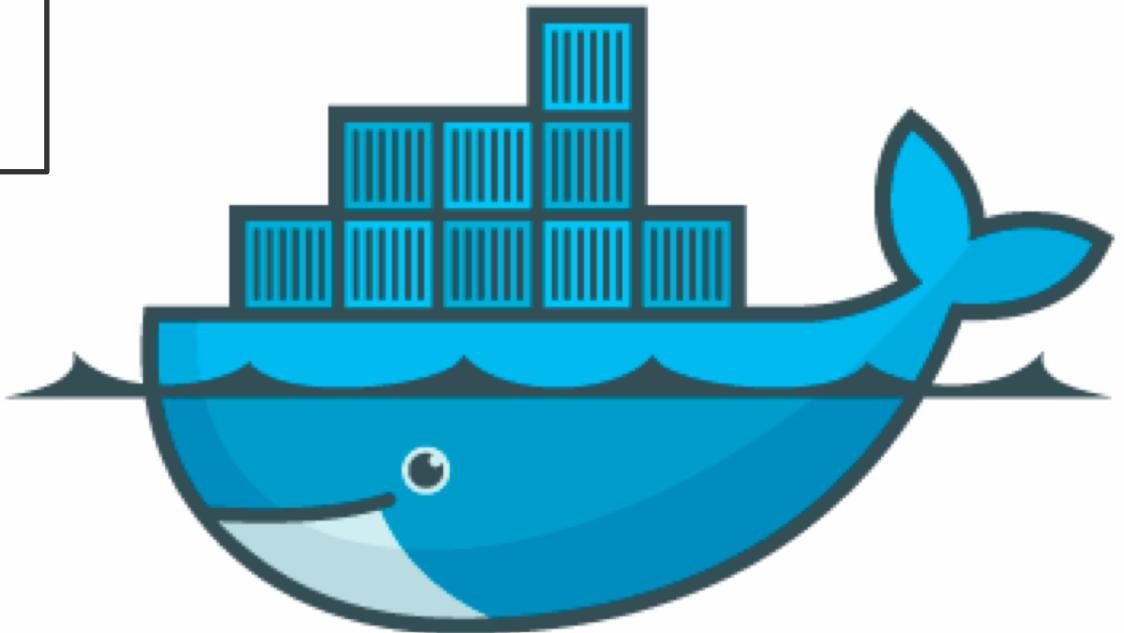
makeameme.org

memegenerator.net

ERE
akeameme.org

Docker

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.



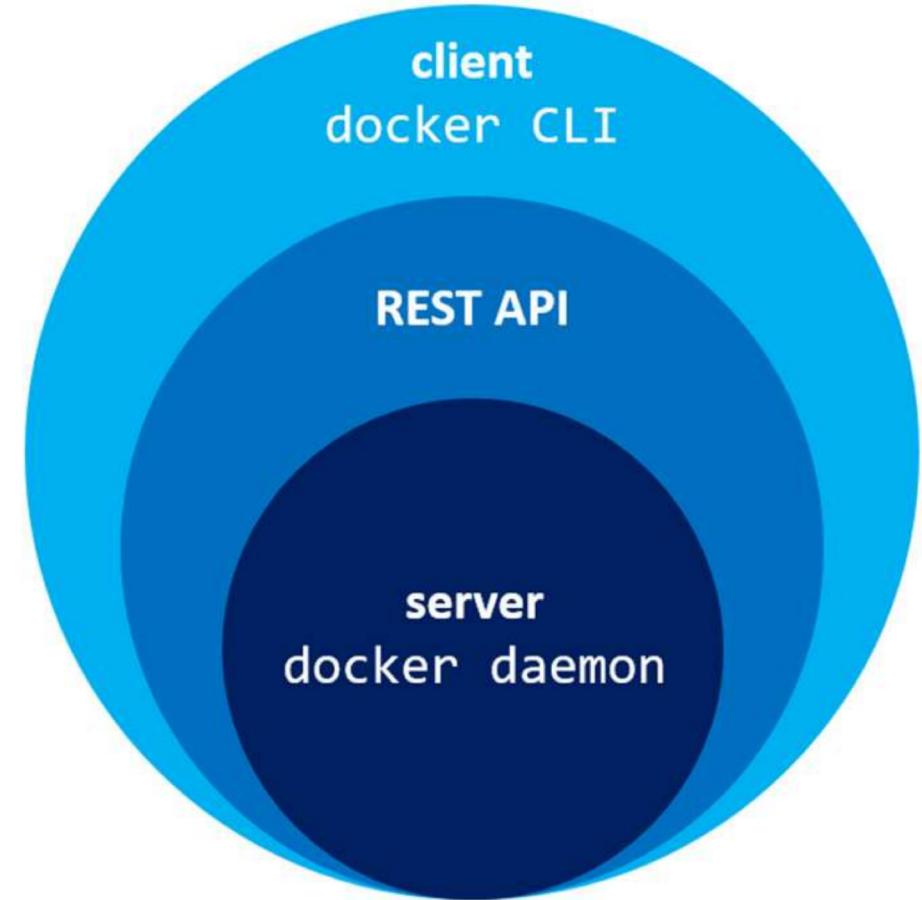
docker

Архитектура

- Docker-демон
- Docker-клиент

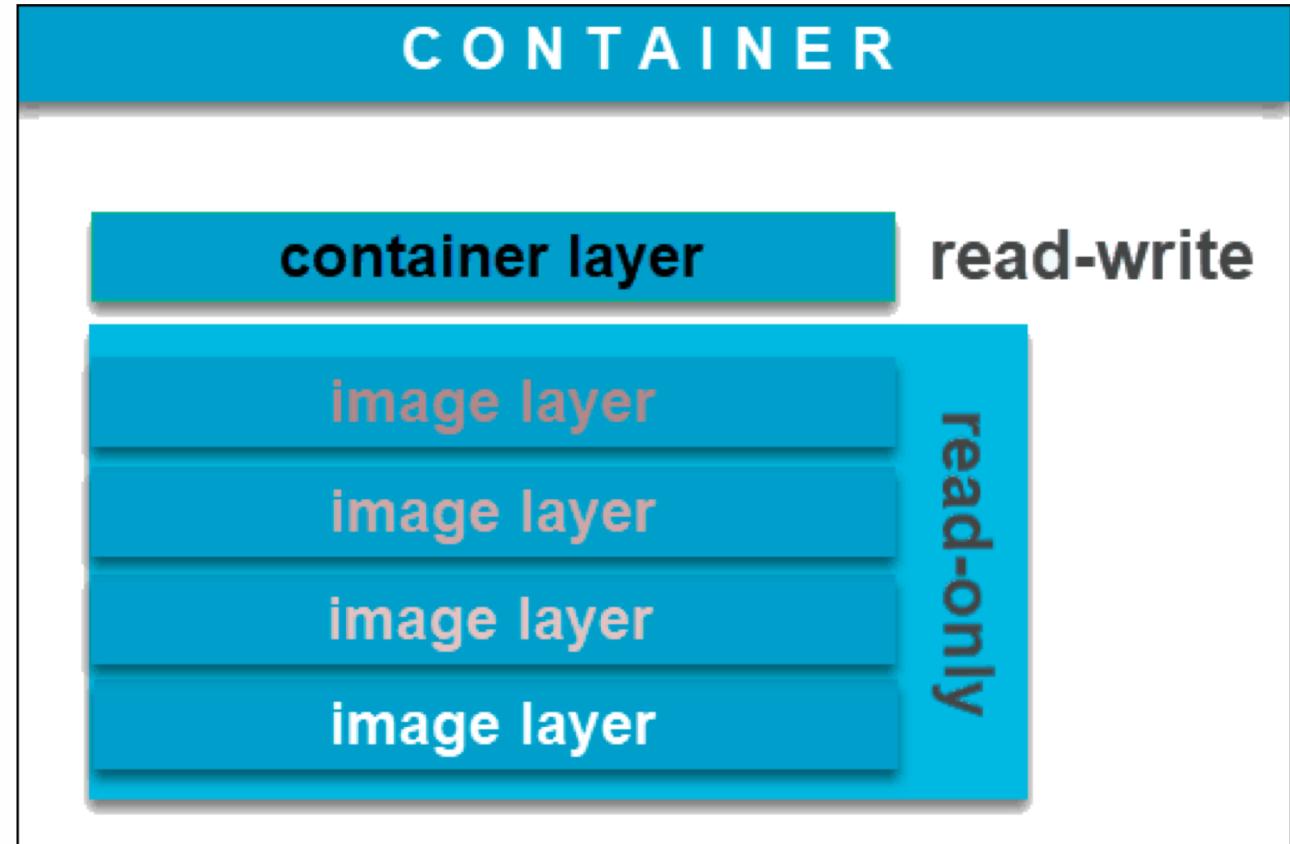
Внутри docker-а:

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



Образы

- запуск команды
- добавление файла или директории
- создание переменной окружения
- указания что запускать когда запускается контейнер этого образа



Реестр

Docker Hub

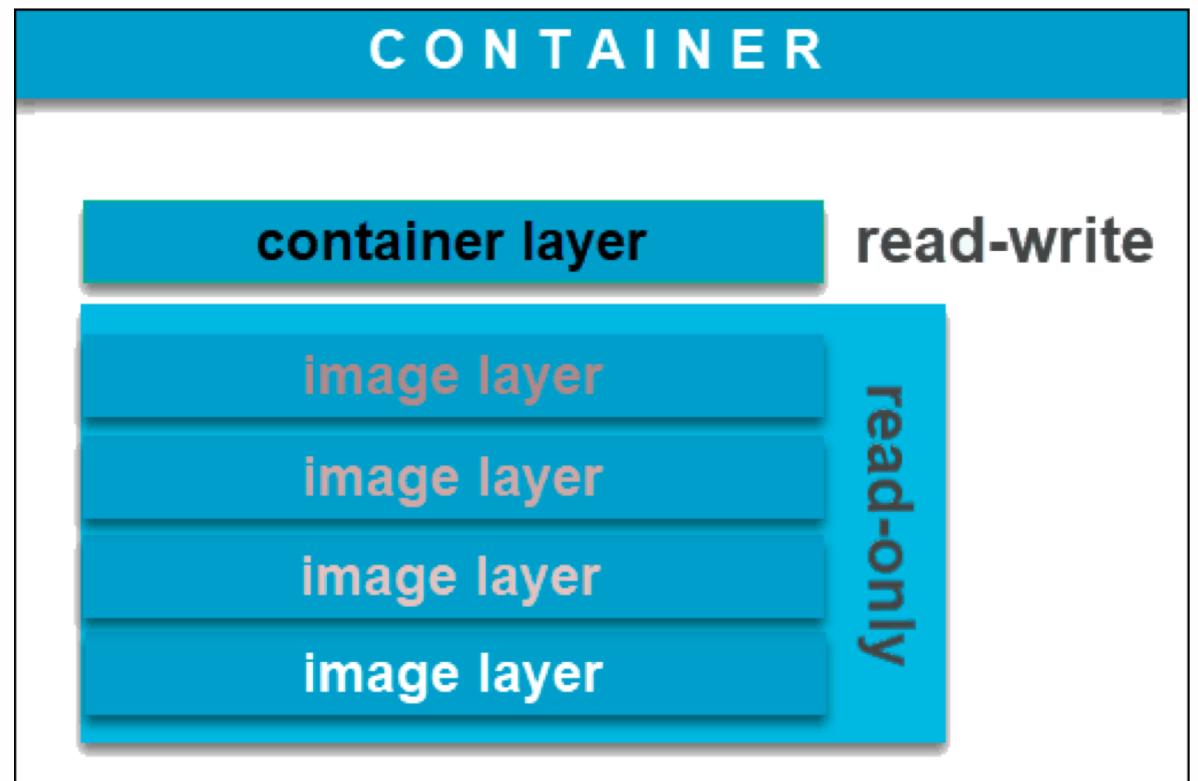


#014

Контейнер

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

- скачивает образ ubuntu
- создает контейнер
- инициализирует файловую систему
- инициализирует сеть/мост
- установка IP адреса
- запускает указанный процесс
- выдает вывод процесса



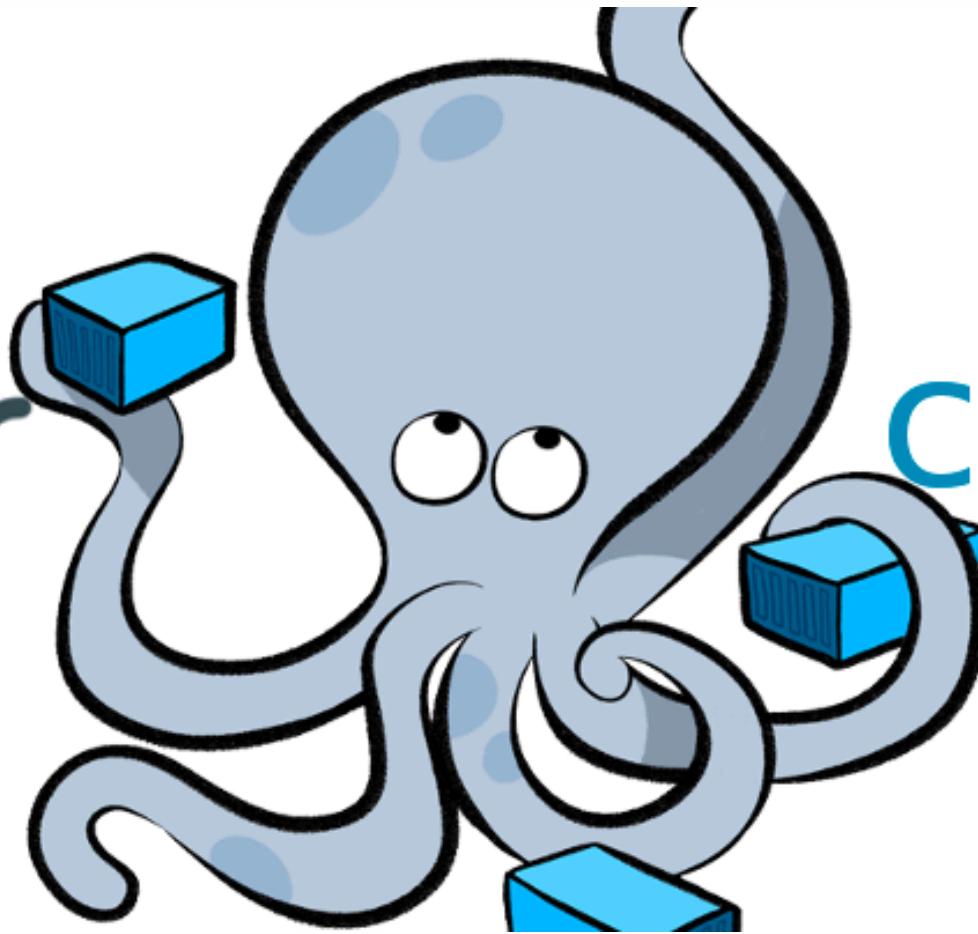
Клиент

- docker images (-a) (-q)
- docker ps (-a) (-q)
- docker pull <image_name>:[<tag>]
- docker build -t <name> <path_to_dockerfile_dir>
- docker push <image_name>:[<tag>]
- doker rmi <image_name|image_id>
- docker run <image_name|image_id>:[<tag>] <command>
- docker stop <container_id|container_name>
- docker kill <container_id|container_name>
- docker rm (-f) <container_id|container_name>
- docker logs <container_id|container_name>

Dockerfile

- ADD
- COPY
- RUN
- ENV
- EXPOSE
- FROM
- VOLUME
- CMD
- ENTRYPOINT

 docker compose

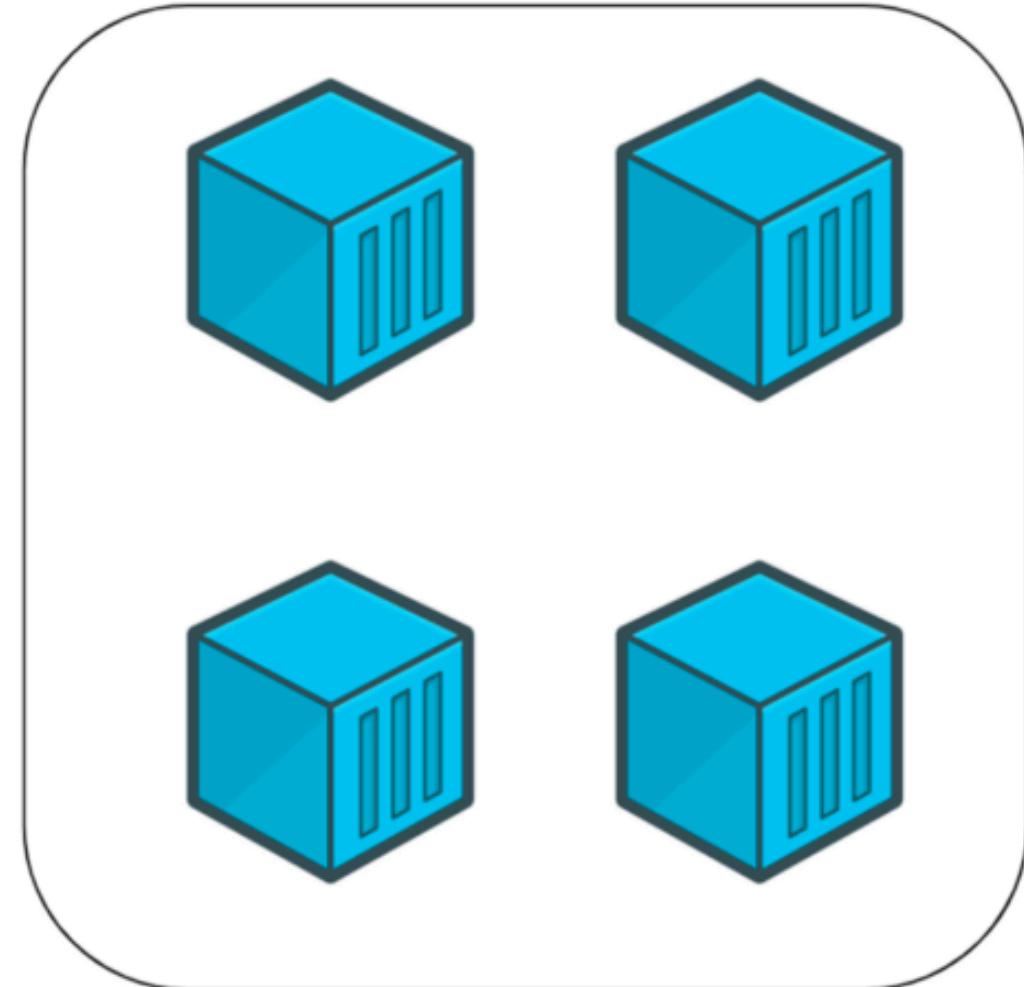


#018

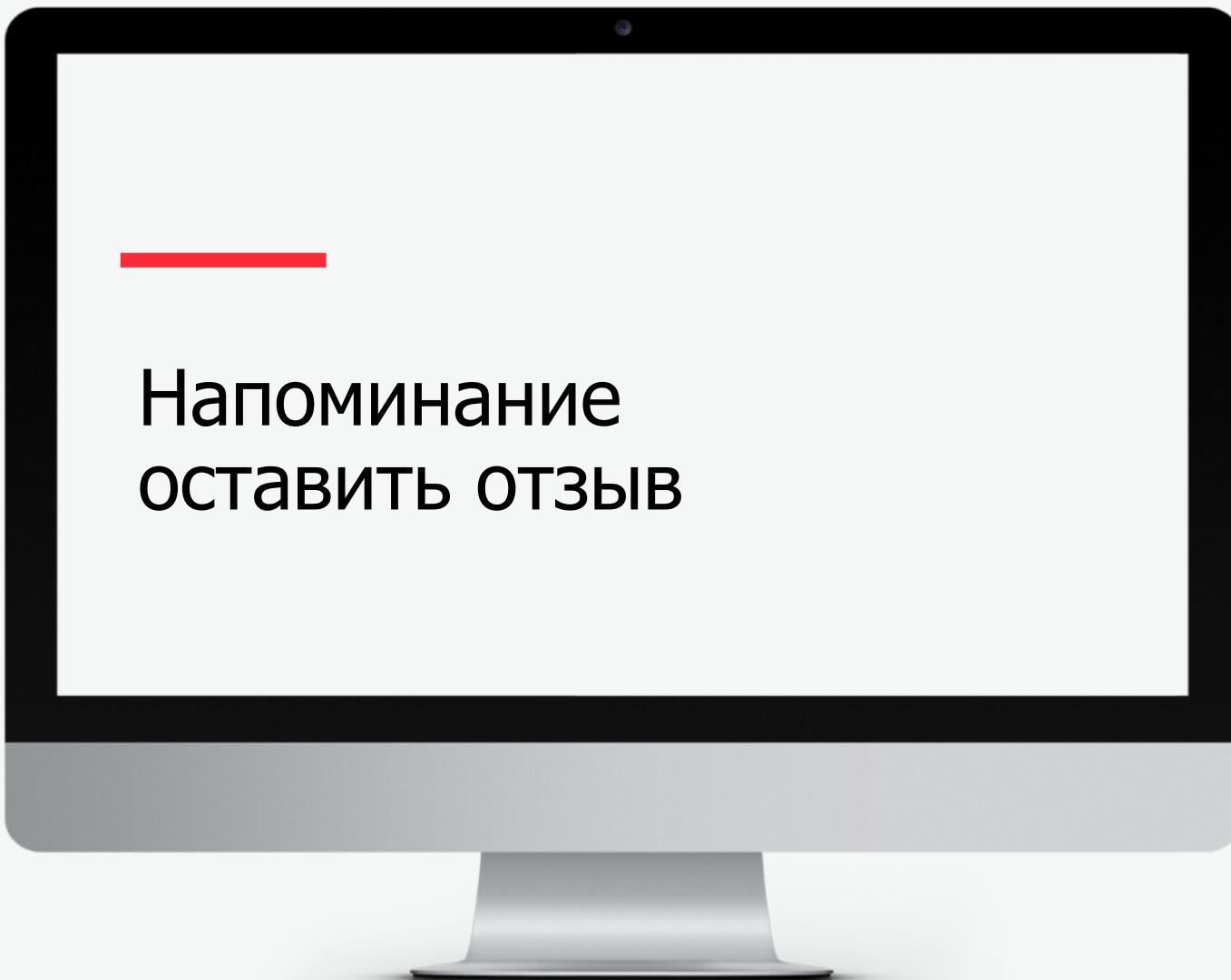
#019



Docker



Docker-Compose



Напоминание
оставить отзыв



**СПАСИБО
ЗА ВНИМАНИЕ**

