

SQL Server и Docker

Цель занятия

Цель занятия — как docker может быть полезен для разработчиков (в том числе как можно использовать для выполнения ДЗ на курсе):

- Понять, что такое docker, для чего нужен, научиться управлять контейнерами (запускать, останавливать).
- Запустить SQL Server в docker.
- Развернуть БД (из бэкапа или SQL-скриптами).

Аудитория — SQL Server-разработчики, которые работали с SQL Server в Windows, но мало знакомы с Docker.

Содержание

- Как быстро "установить" SQL Server через Docker
- Что такое контейнеры
- Архитектура Docker
- Разворачивание БД в Docker
- Dockerfile

Пример

ДЕМО: "Установка" и удаление SQL Server

"Устанавливаем" SQL Server

```
docker run \  
  --name sql_demo \  
  -d \  
  -p 1433:1433 \  
  -e SA_PASSWORD=P@ssw0rd \  
  -e ACCEPT_EULA=Y mcr.microsoft.com/mssql/server:2019-latest
```

Проверяем (+ SSMS / ADS)

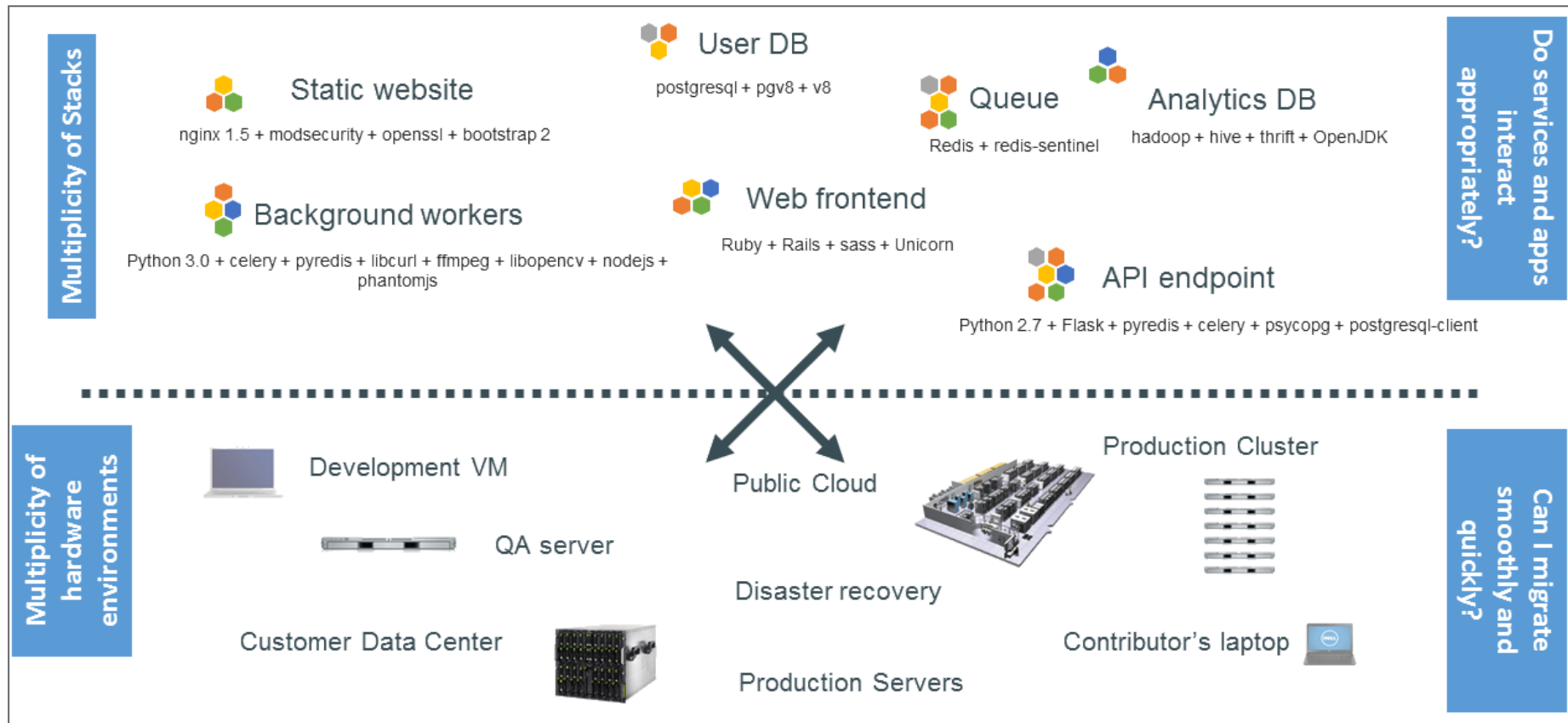
```
docker ps
```

Удаляем SQL Server

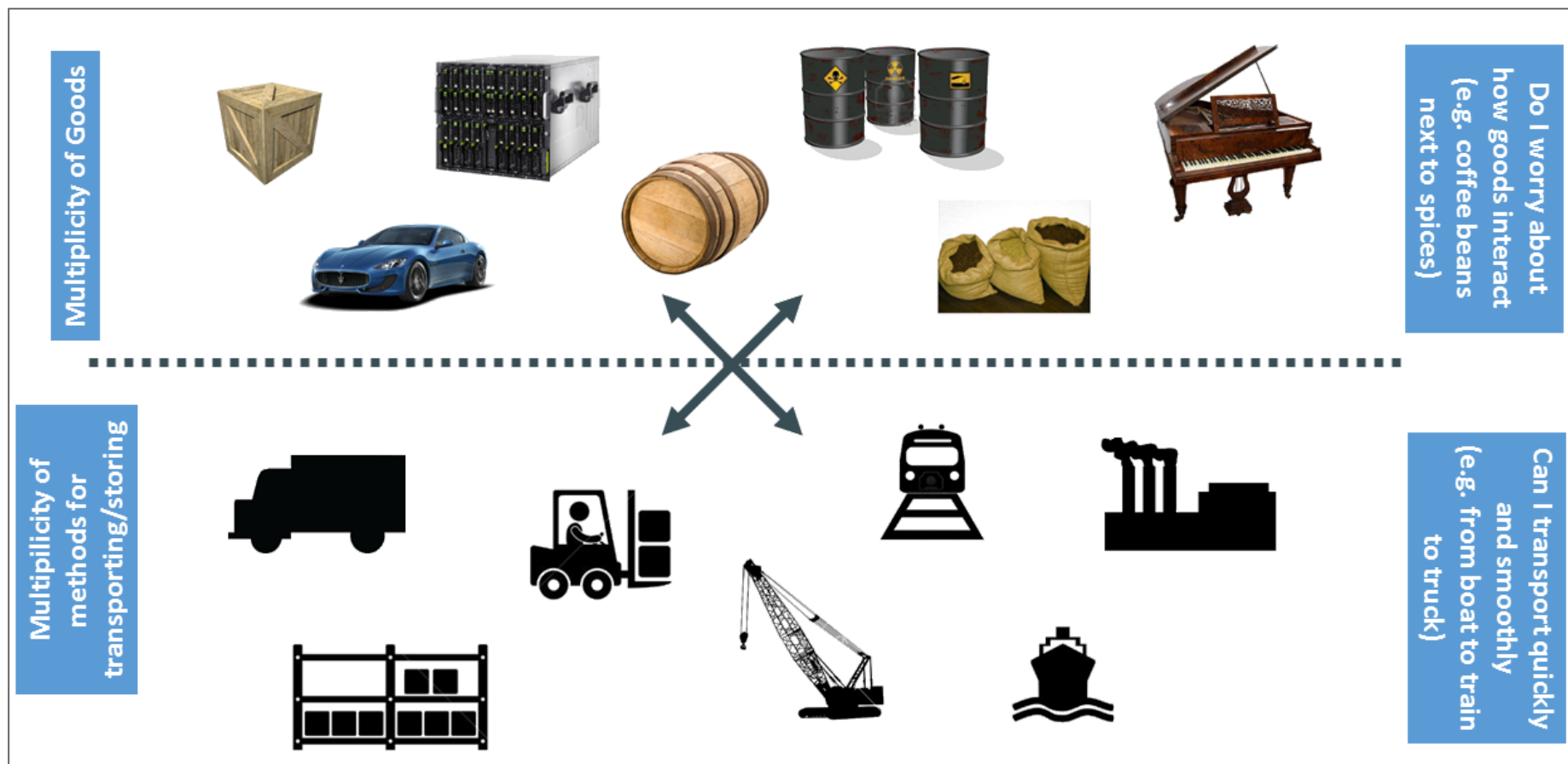
```
docker stop sql_demo  
  
docker ps  
docker ps -a  
  
docker rm sql_demo
```

Контейнеризация

Проблема — деплой



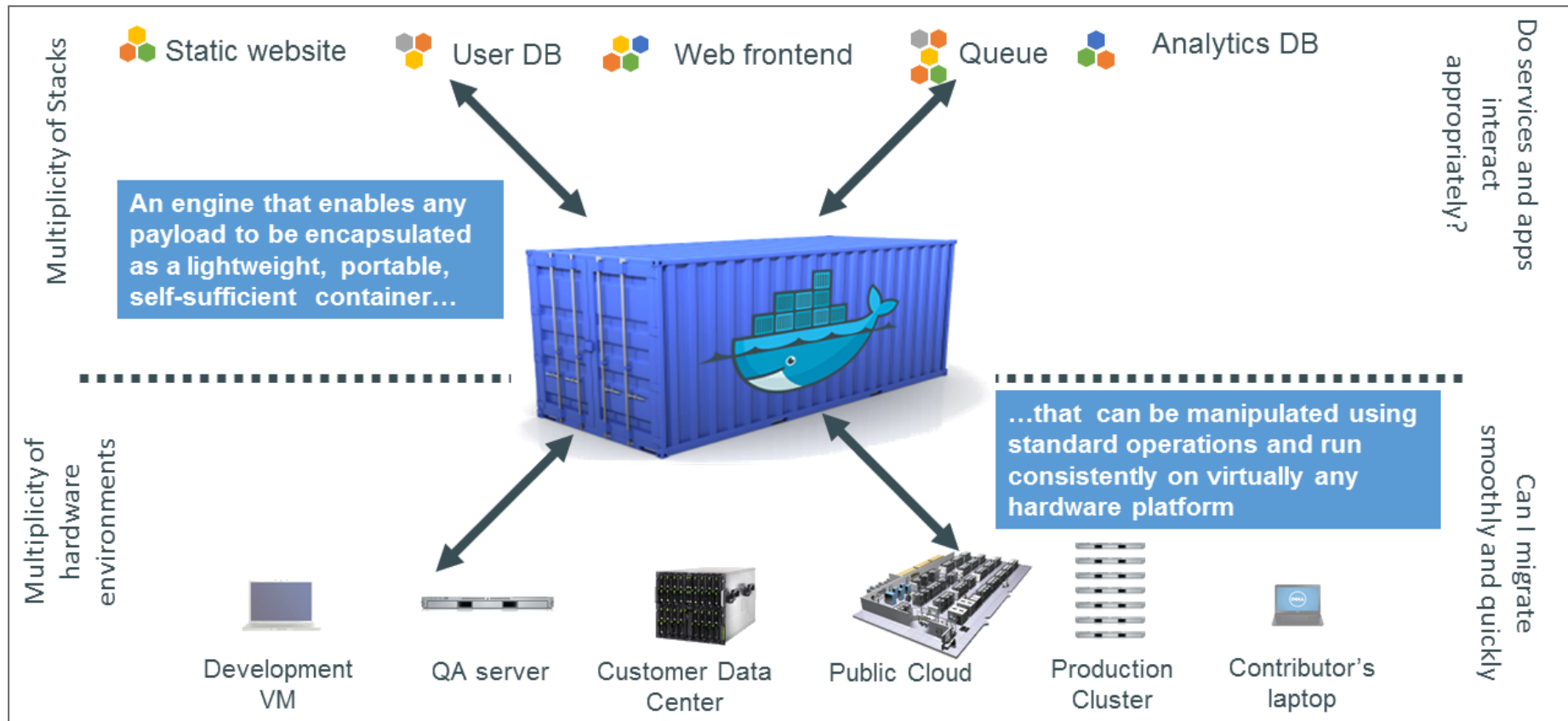
Перевозка грузов



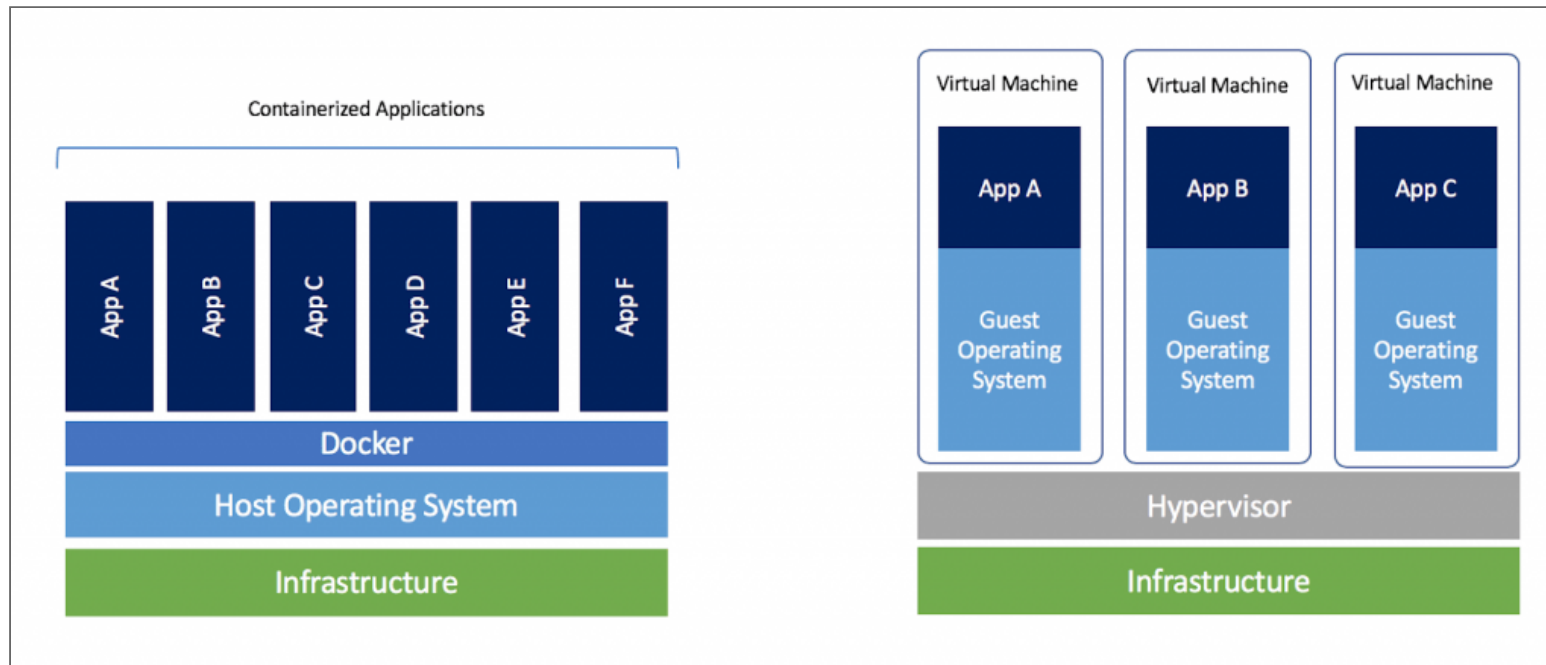
Контейнеры для грузов



Контейнеры docker

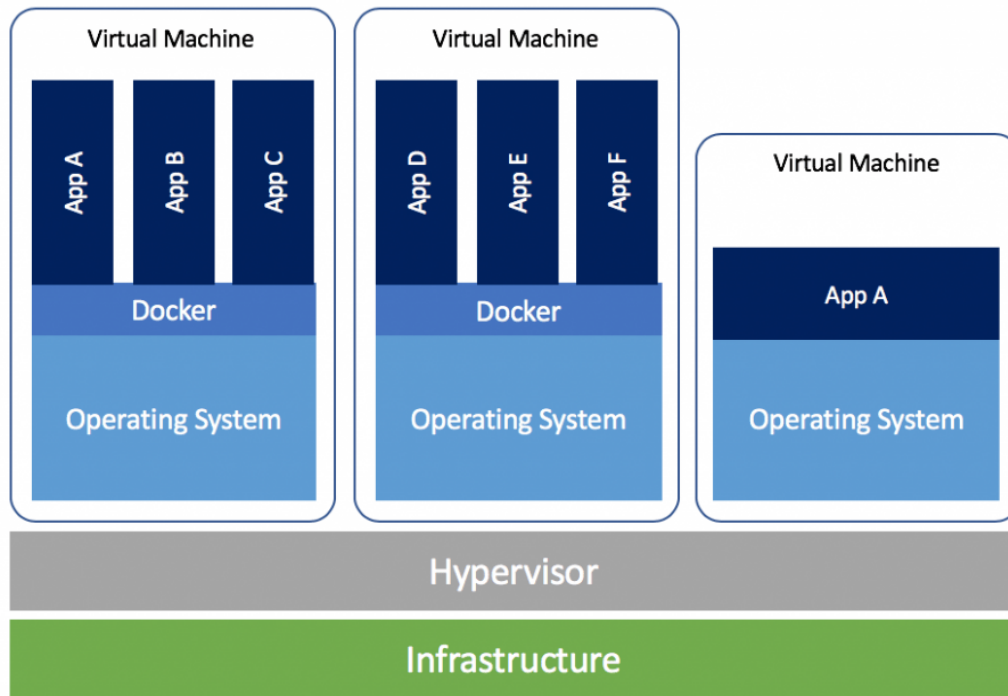


Виртуализация и контейнеризация



- Виртуализация — абстракция физического железа.
- Контейнеризация — абстракция ОС.

Docker в виртуальных машинах

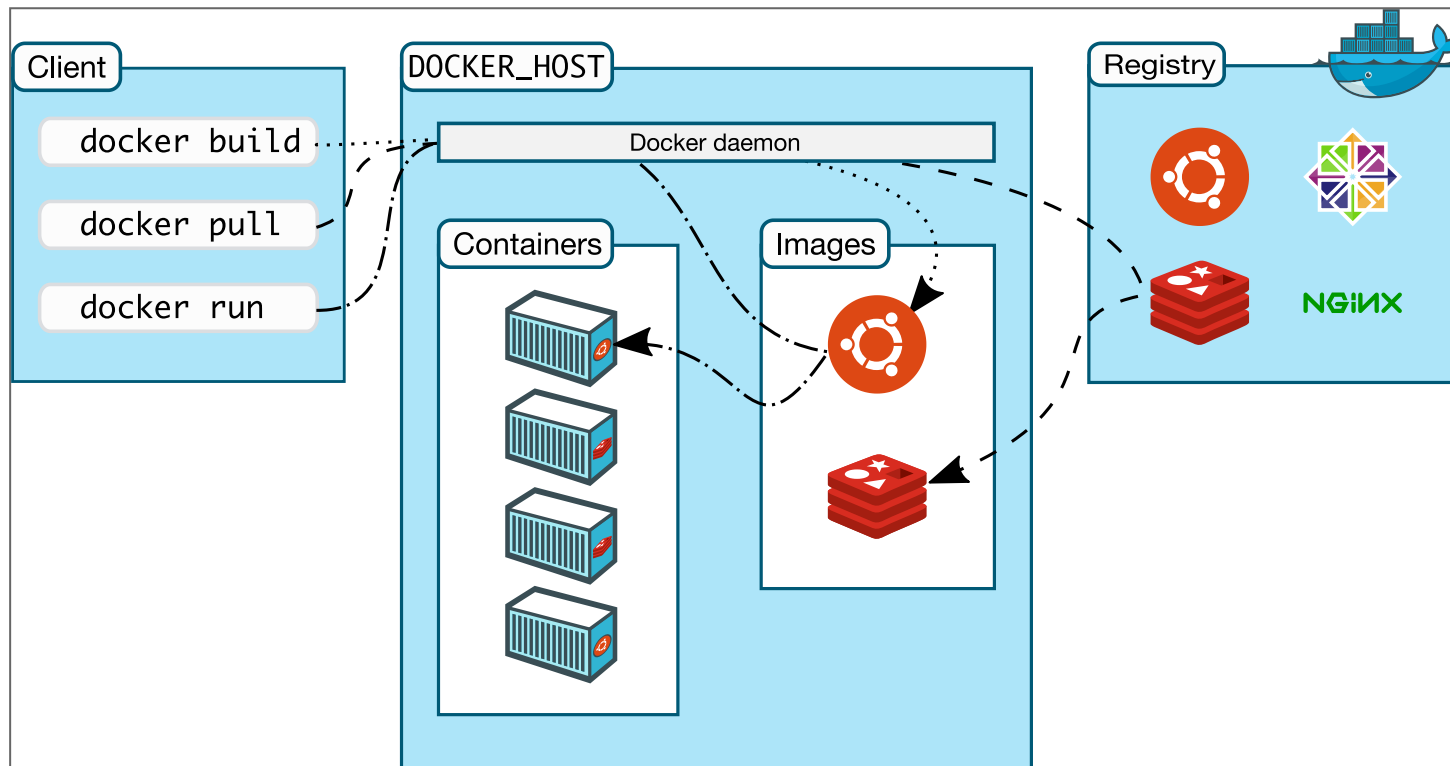


Резюме: контейнеры

- Контейнеры — это про доставку, деплой ПО, а не про замену виртуальных машин.
- С контейнерами получаем "повторяемость": одинаковая конфигурация на DEV-, TEST-, PROD-окружениях.

Архитектура, установка docker

Компоненты docker

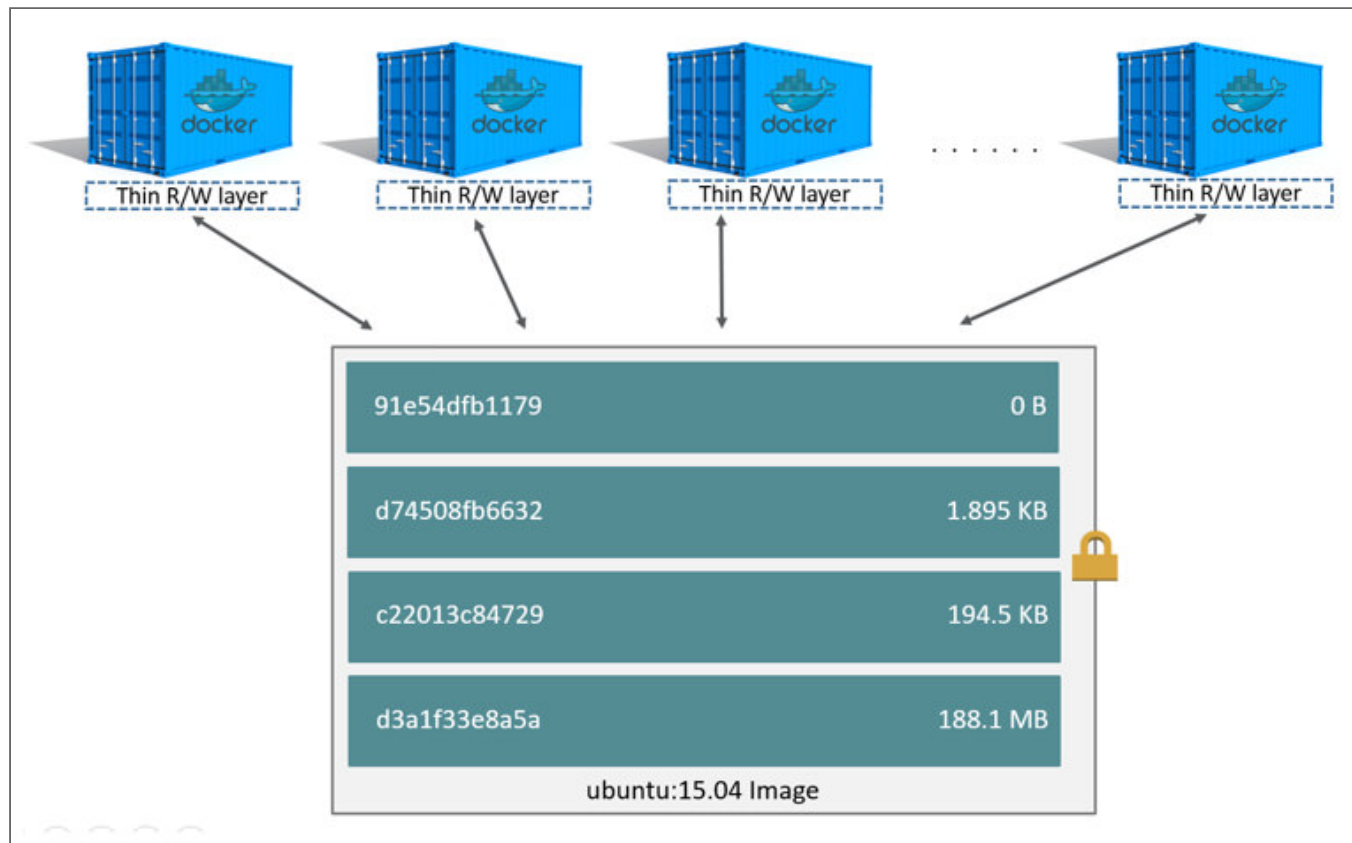


- <https://docs.docker.com/get-started/overview/>
- Registry — <https://hub.docker.com>

Образы, контейнеры

- Image (образ)
 - read-only шаблон для создания контейнера
 - можно сравнить с exe-файлом на диске
 - хранятся в Registry (репозиторий образов)
- Container (контейнер)
 - контейнер — runtime-экземпляр образа
 - можно сравнить с запущенным exe-файлом в ОЗУ
 - изменения после удаления контейнера теряются

Слоеная файловая система



Docker в Linux, Windows, Mac

- Windows
 - <https://docs.docker.com/docker-for-windows/install/>
- Linux
 - <https://docs.docker.com/engine/install/>
- Mac
 - <https://docs.docker.com/docker-for-mac/install/>

Linux container vs Windows container

- Образы Windows-контейнеров работают в Windows
 - <https://hub.docker.com/r/microsoft/mssql-server-windows-developer>
- Образы Linux-контейнеров работают в Linux (и в Windows)
 - <https://hub.docker.com/r/microsoft/mssql-server-linux/>

Основные команды docker

```
docker ps
docker ps -a

docker run -d -p port_host:port_docker container_name

docker stop container_name

docker exec -it container_name bash

docker inspect container_name
```

DEMO: SQL Server

Запуск SQL Server

Теперь подробнее про SQL Server в Docker

DEMO: Запуск SQL Server

```
docker run \  
  --name sql_demo \  
  -d \  
  -p 1433:1433 \  
  -e MSSQL_SA_PASSWORD=P@ssw0rd \  
  -e ACCEPT_EULA=Y \  
  mcr.microsoft.com/mssql/server:2019-latest
```

Подробнее про параметры контейнера

- <https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-environment-variables?view=sql-server-ver15>
- https://hub.docker.com/_/microsoft-mssql-server

ДЕМО: Разворачиваем БД из бэкапа

- В курсе используется демо БД WideWorldImporters ([ссылка на backup](#))
- Копирование файла в контейнер

```
docker cp ./WideWorldImporters-Full.bak sql_demo:/tmp/WideWorldImporters-Full.bak  
docker exec -it sql_demo ls /tmp
```

```
docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH  
docker cp [OPTIONS] SRC_PATH CONTAINER:DEST_PATH
```

- RESTORE DATABASE ...
- Помните — все пути в SQL Server будут внутри контейнера

DEMO: stop/start, remove

- Сейчас все данные находятся внутри контейнера
- Контейнеры stateless — без сохранения состояния
- Если контейнер остановить и перезапустить, то данные останутся

```
docker ps
```

```
docker stop sql_demo
```

```
docker ps
```

```
docker ps -a
```

```
docker start sql_demo
```

```
docker ps
```

- Если контейнер удалить и запустить заново, то данные теряются

```
docker stop sql_demo
```

```
docker rm sql_demo
```

```
docker ps -a
```

```
docker run --name sql_demo -d -p 1433:1433 \  
-e MSSQL_SA_PASSWORD=P@ssw0rd -e ACCEPT_EULA=Y \  
mcr.microsoft.com/mssql/server:2019-latest
```


DEMO: Сохраняем данные вне контейнера

- bind mount — монтирование каталогов с хоста.
- <https://docs.docker.com/storage/bind-mounts/>

```
sudo chown 10001:0 ./mssql_data
```

```
docker run --name sql_demo2 -d -p 1435:1433 \  
-v /home/spv/otus_sql_docker/mssql_data:/var/opt/mssql/ \  
-e MSSQL_SA_PASSWORD=P@ssw0rd -e ACCEPT_EULA=Y \  
mcr.microsoft.com/mssql/server
```

```
sudo ls ./mssql_data/
```

DEMO: попробуем создать БД, удалить, запустить контейнер.

Volumes (тома хранения данных)

- Тома — рекомендуемый разработчиками Docker способ хранения данных.
- Создаются через `docker volume create`
- Один том может быть примонтирован одновременно в несколько контейнеров.
- Когда никто не использует том, он не удаляется, а продолжает существовать.

Материалы:

- <https://habr.com/ru/company/southbridge/blog/534334/>
- <https://docs.docker.com/storage/volumes/>

Создание образов, Dockerfile

DEMO: Dockerfile

Dockerfile — текстовый файл, который содержит команды для создания образа.
Сборка образа с именем `mssql2019-wwi`:

```
docker build -t mssql2019-wwi .
```

Какие образы есть

```
docker images
```

Запуск контейнера из собранного образа `mssql2019-wwi`:

```
docker run --name mssql2019 -d -p 1436:1433 mssql2019-wwi
```

Это были только самые основы

Сегодня обсудили

- Что такое docker, для чего используется
- Как использовать docker для запуска SQL Server
- Как развернуть БД внутри контейнера

Заполните, пожалуйста опрос <https://forms.gle/zqepLyKqKMDw2Rq9A>