ДОКЕР

ЧТО ЭТО И С ЧЕМ ЕГО ЕДЯТ

ЧТО ТАКОЕ ДОКЕР ?

- Docker — это инструмент, предоставляющий удобный интерфейс для работы с контейнерами.
- Контейнеры (в основном LXC – “Linux Containers”) - это система виртуализации на уровне операционной системы для запуска нескольких изолированных экземпляров операционной системы Linux на одном узле. LXC не использует виртуальные машины, а создаёт виртуальное окружение с собственным пространством процессов и сетевым стеком. Все экземпляры LXC используют один экземпляр ядра операционной системы.

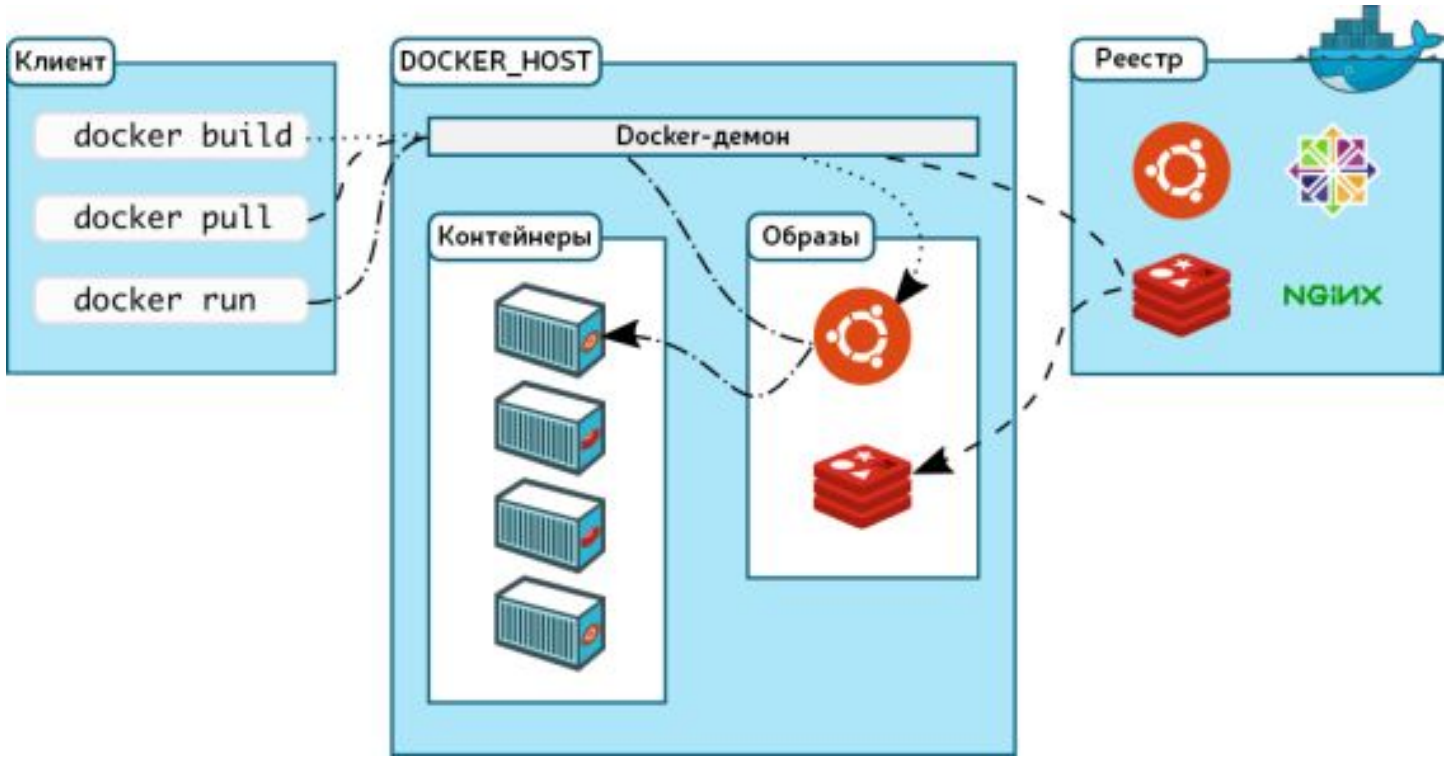
ГЛАВНЫЕ КОМПОНЕНТЫ DOCKER

Docker состоит из двух главных компонент:

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

АРХИТЕКТУРА DOCKER

- Docker использует архитектуру клиент-сервер. Docker клиент общается с демоном Docker, который берет на себя создания, запуска, распределения ваших контейнеров. Оба, клиент и сервер могут работать на одной системе, вы можете подключить клиент к удаленному демону docker. Клиент и сервер общаются через сокет или через RESTful API.



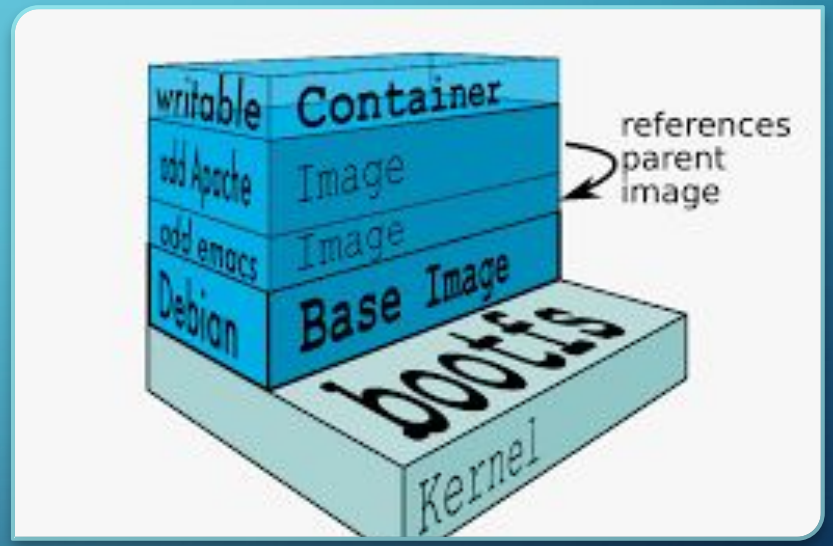
ВНУТРИ ДОКЕР-А

- Образы (images) - это read-only шаблон. Например, образ может содержать операционку Ubuntu с Apache и приложением на ней. Образы используются для создания контейнеров. Docker позволяет легко создавать новые образы, обновлять существующие, или вы можете скачать образы созданные другими людьми. Образы — это компонента сборки docker-а.
- Реестр (registries) - хранит образы. Есть публичные и приватные реестры, из которых можно скачать либо загрузить образы. Публичный Docker-реестр — это Docker Hub. Реестры — это компонента распространения.
- Контейнеры - Контейнеры похожи на директории. В контейнерах содержится все, что нужно для работы приложения. Каждый контейнер создается из образа. Контейнеры могут быть созданы, запущены, остановлены, перенесены или удалены. Каждый контейнер изолирован и является безопасной платформой для приложения. Контейнеры — это компонента работы.

КАК РАБОТАЕТ ОБРАЗ?

- Каждый образ состоит из набора уровней. Docker использует union file system для сочетания этих уровней в один образ. Union file system позволяет файлам и директориями из разных файловых систем (разным ветвям) прозрачно накладываться, создавая когерентную файловую систему.
- Docker образы могут создаться из этих базовых образов, шаги описания для создания этих образов мы называем инструкциями. Каждая инструкция создает новый образ или уровень. Инструкциями будут следующие действия:
- запуск команды
- добавление файла или директории
- создание переменной окружения
- указания что запускать когда запускается контейнер этого образа
- Эти инструкции хранятся в файле Dockerfile. Docker считывает это

- Одна из причин, по которой docker легковесен — это использование таких уровней. Когда вы изменяете образ, например, обновляете приложение, создается новый уровень. Так, без замены всего образа или его пересборки, как вам возможно придётся сделать с виртуальной машиной, только уровень добавляется или обновляется. И вам не нужно раздавать весь новый образ, раздается только обновление, что позволяет распространять образы проще и быстрее.



КАК РАБОТАЕТ DOCKER РЕЕСТР?

- Реестр — это хранилище docker образов. После создания образа вы можете опубликовать его на публичном реестре Docker Hub или на вашем личном реестре. С помощью docker клиента вы можете искать уже опубликованные образы и скачивать их на вашу машину с docker для создания контейнеров.
  Docker Hub предоставляет публичные и приватные хранилища образов. Поиск и скачивание образов из публичных хранилищ доступно для всех. Содержимое приватных хранилищ не попадает в результат поиска. И только вы и ваши пользователи могут получать эти образы и создавать из них контейнеры.
- Команды docker для работы с реестром:
  - Docker pull <image-name:tag> – скачать образ с docker hub/приватного репозитория
  - Docker push <image-name:tag> - залить образ в docker hub/приватный репозитории

КАК РАБОТАЕТ КОНТЕЙНЕР?

- Контейнер состоит из операционной системы, пользовательских файлов и метаданных. Каждый контейнер создается из образа. Этот образ говорит docker-у, что находится в контейнере, какой процесс запустить, когда запускается контейнер и другие конфигурационные данные. Docker образ доступен только для чтения. Когда docker запускает контейнер, он создает уровень для чтения/записи сверху образа (используя union file system, как было указано раньше), в котором может быть запущено приложение.

ЧТО ПРОИСХОДИТ, КОГДА ЗАПУСКАЕТСЯ КОНТЕЙНЕР?

- Или с помощью программы docker, или с помощью RESTful API, docker клиент говорит docker демону запустить контейнер.
- $ sudo docker run -i -t ubuntu /bin/bash
  - Клиент запускается с помощью команды docker, с опцией run, которая говорит, что будет запущен новый контейнер. Минимальными требованиями для запуска контейнера являются следующие атрибуты:
    - какой образ использовать для создания контейнера. В нашем случае ubuntu
    - команду которую вы хотите запустить когда контейнер будет запущен. В нашем случае /bin/bash

ЧТО ЖЕ ПРОИСХОДИТ ПОД КАПОТОМ, КОГДА МЫ ЗАПУСКАЕМ ЭТУ КОМАНДУ?

- Docker, по порядку, делает следующее:
  - **скачивает образ ubuntu**: docker проверяет наличие образа ubuntu на локальной машине, и если его нет — то скачивает его с Docker Hub. Если же образ есть, то использует его для создания контейнера;
  - **создает контейнер**: когда образ получен, docker использует его для создания контейнера;
  - **инициализирует файловую систему и монтирует read-only уровень**: контейнер создан в файловой системе и read-only уровень добавлен образ;
  - **инициализирует сеть/мост**: создает сетевой интерфейс, который позволяет docker-у общаться хост машиной;
  - **Установка IP адреса**: находит и задает адрес;
  - **Запускает указанный процесс**: запускает ваше приложение;
  - **Обрабатывает и выдает вывод вашего приложения**: подключается и логирует стандартный вход, вывод и поток ошибок вашего приложения, что бы вы могли отслеживать как работает ваше приложение.
- Теперь у вас есть рабочий контейнер. Вы можете управлять своим контейнером, взаимодействовать с вашим приложением. Когда решите остановить приложение, удалите контейнер.

КАК РАБОТАЕТ КОНТЕЙНЕР?

- Контейнер состоит из операционной системы, пользовательских файлов и метаданных. Каждый контейнер создается из образа. Этот образ говорит docker-у, что находится в контейнере, какой процесс запустить, когда запускается контейнер и другие конфигурационные данные. Docker образ доступен только для чтения. Когда docker запускает контейнер, он создает уровень для чтения/записи сверху образа (используя union file system, как было указано раньше), в котором может быть запущено приложение.

DOCKER COMPOSE

- Docker Compose — это инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов

- Технология Docker Compose, если описывать её упрощённо, позволяет, с помощью одной команды, запускать множество сервисов.

РАЗНИЦА МЕЖДУ DOCKER И DOCKER COMPOSE

- Docker применяется для управления отдельными контейнерами (сервисами), из которых состоит приложение.

  Docker Compose используется для одновременного управления несколькими контейнерами, входящими в состав приложения. Этот инструмент предлагает те же возможности, что и Docker, но позволяет работать с более сложными приложениями.