Инструментальные средства информационных систем

Лекция 2 Инструменты виртуализации и контейнеризации

Содержание

1 Введение

В незапамятные времена на заре компьютерной эры можно было написать программу, запустить у себя в IDE, проверить, что код работает, сохранить код в файл. И потом вы могли отнести этот код на дискете другу и он у него скорее всего тоже будет работать точно так же и, вы уже вместе могли бы работать над кодом дальше.

Что происходит сейчас? Вы написали программу, использовали Java+Postgre, пришли к другу с кодом, а у него Java то есть, а вот PostgreSQL нет. Совсем не хочется устанавливать и настраивать PostgreSQL, тем более что это не так просто и вы не помните, какие конкретно параметры БД указывали (вам другой друг помогал).

Вот с этого момента на сцену выходят инструменты:

– облачные сервисы,

– виртуальные машины,

– контейнеризация.

Опустим облачные сервисы, хотя это очень актуально на сегодняшний день.

Виртуальные машины и контейнеризация помогут упаковать код, среду выполнения, системные библиотеки и настройки в образ (в случае виртуальных

машин) или в контейнер (с лучае контейнеризации) и затем этот образ или контейнер можно будет развернуть на другом компьютере.

2 Определения

Контейнеризация и виртуализация похожи, но работают они по-разному:

– **Виртуализация** создает виртуальный отдельный компьютер внутри компьютера, имеет свою ОС (гостевую), заимствует ресурсы: ЦП, память и хранилище у host-машины, реального компьютера или сервера (такое заимствование называется виртуализацией оборудования).

– **Контейнеризация** подразумевает также наличие своей ОС, но процессы запущенные в контейнере делят ресурсы host-машины с процессами запущенными в host-системе, т.е. контейнер не виртуализирует оборудование, поэтому потребляет меньше ресурсов.

Сами определения двух этих подходов звучат следующим образом:

Виртуальная машина — это компьютерный файл (обычно его называют образом), который действует как обычный компьютер.

Пример ПО для виртуализации VirtualBox, VMWare и пр, логотипы:

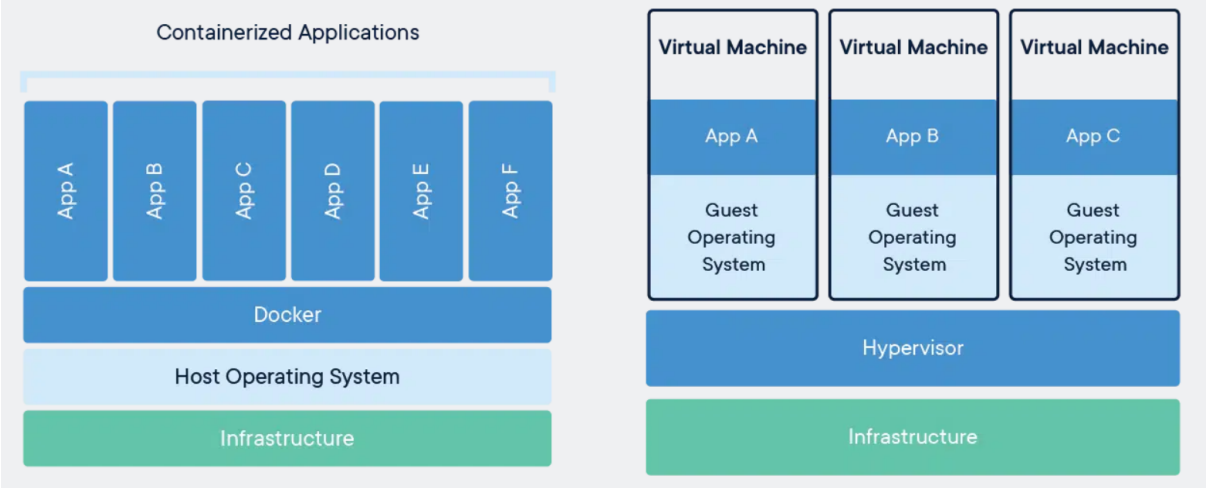
Контейнер — это стандартная единица программного обеспечения, которая заключает в себе код и все его зависимости, чтобы приложение быстро и надежно запускалось из одной среды внутри другой.

Пример ПО для создания, доставки и запуска контейнеров Docker, логотип:

3 Схемы

Контейнеры — это абстракция на уровне приложения.

Виртуальные машины (ВМ) — это абстракция физического оборудования, превращающая один сервер во множество серверов.



Infrastructure – инфраструктура, другими словами окружение или среда.

Гипервизор или диспетчер виртуальных машин, позволяет одновременно запускать разные операционные системы на нескольких виртуальных машинах на одной машине.

Основные компоненты Docker:

Контейнеры – про них мы уже говорили достаточно.

Демон (Docker-daemon) — сервер контейнеров. Демон управляет Docker-объектами (сети, хранилища, образы и контейнеры). Демон также может связываться с другими демонами для управления сервисами Docker.

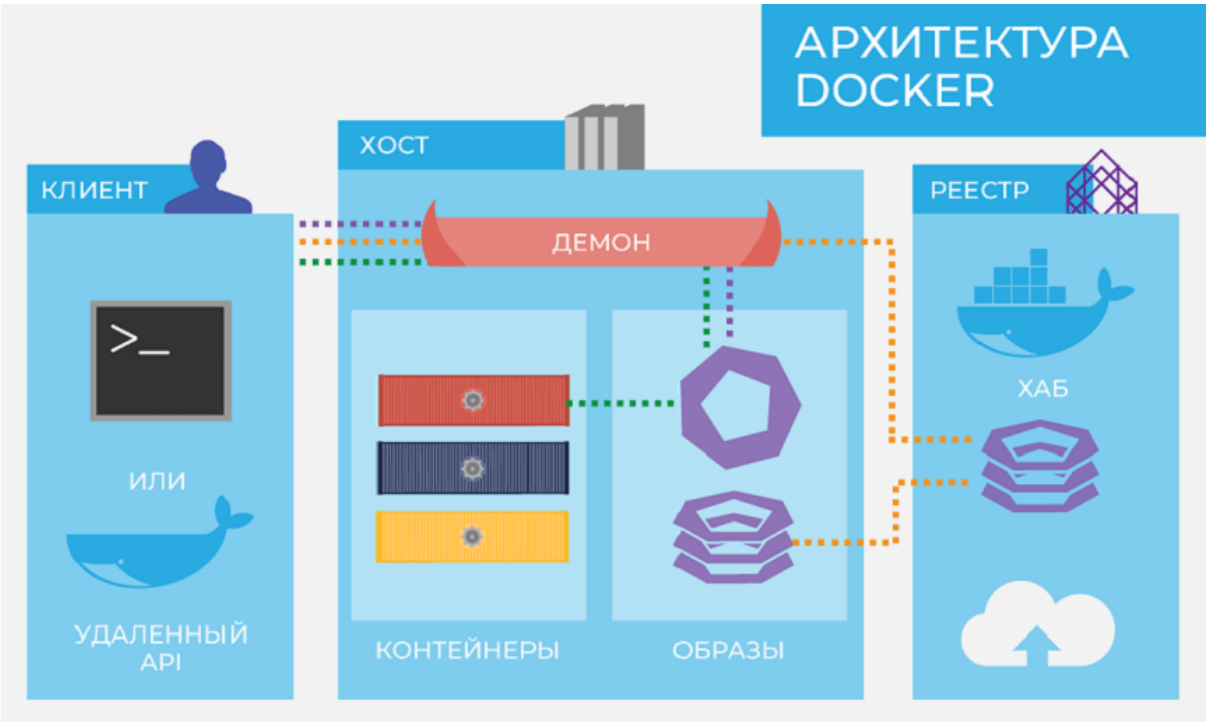
Клиент (Docker-client / CLI) — интерфейс взаимодействия пользователя с Docker-демоном. Клиент и Демон — важнейшие компоненты «движка» – Docker Engine. Клиент Docker может взаимодействовать с несколькими демонами.

Образ (Docker-image) — файл, включающий зависимости, сведения, конфигурацию для дальнейшего развертывания и инициализации контейнера. Только для чтения.

Хаб (Docker-hub) или хранилище данных — репозиторий, предназначенный для хранения образов с различным программным обеспечением.

Адрес Docker-hub: https://hub.docker.com/ – самая большая библиотека образов контейнеров.

Хост (Docker-host) — машинная среда для запуска контейнеров с программным обеспечением.



4 Плюсы

Преимущество	Виртуализация	Контейнеризация
экономия ресурсов	консолидирует множество небольших нагрузок на одном компьютере, что позволяет обеспечить высокую степень использования ресурсов;	распределение ресурсов между разными приложениями, более эффективное использование ресурсов;
безопасность	виртуальные машины работают в нескольких операционных системах, использование гостевой операционной системы на виртуальной машине позволяет запускать приложения с недостаточным уровнем безопасности и защитить операционную систему узла;	позволяет запустить приложение отдельно от всей системы без конфликтов с другими программами, если код контейнерного приложения окажется небезопасным, это не навредит серверу-хосту, при правильной настройке контейнера деятельность кода не затронет основную систему;
портативность	можно перемещать между физическими компьютерами в сети и даже между локальной и облачной средами;	позволяет перенести приложение со всеми зависимостями на другую систему с помощью пары команд в терминале;
ускорение разработки	запустить виртуальную машину легче, быстрее и	На настройку среды, разворачивание приложений

	намного проще, чем выполнять подготовку новой среды для разработчиков;	под разными платформами тратится меньше времени;

Плюс самого Docker, который и сделал контейнеры очень популярными в 2014 году, это наличие стандартного API.

5 Минусы

Недостатки виртуальных машин:

– размер образа, он может быть достаточно большим;

– сложное управление ресурсами;

– vendor lock in, т.е. привязка к поставщику.

Недостатки самого Docker в том, что его бывает достаточно сложно установить и запустить.