Я хочу рассказать про Докер. Что это такое и зачем он нужен.

1. Согласно википедии определение докера – это

Для того чтобы понять это определение надо разобраться с такими понятиями как виртуализация и контейнеризация.

Окунемся чуток в историю. Большое количество времени стандартом для запуска серверных приложений являлись виртуальные машины. Виртуальная машина – это по сути программа, задачей которой является имитация оборудования, т.е. железа, которое необходимо для запуска приложений.

Выглядит это так.

У нас есть реальный сервер с настоящим процессором, оперативной памятью, дисками и т.д. Этот сервер называется хостом. На сервере стоит операционная система. Затем идет программа отвечающая за иммитацию оборудования и управления ВМ. Называют ее Hypervisor . Гипервизор позволяет из 1 сервера сделать несколько, разделив его ресурсы, причем в любой пропорции. Например на рисунке у нас есть 3 вирт. Машины. На каждой из них надо установить операционную систему. Дальше туда устанавливается само наше приложение с его зависимостями. Эти виртуальные машины полностью изолированны друг от друга. Их можно переносить на другие сервера без необходимости остановки работы самого приложения.

Контейнеризация решает фактически ту же самую задачу, что и виртуализация. Также создается виртуальная изолированная среда внутри которой запускается приложение. Контейнеризация реализуется за счет ядра операционной системы. Затем идет устанавливаемая на хост систему платформа контейнеризации (докер). Каждый контейнер содержит необходимые приложению зависимости, а так же само приложение. Внутри контейнера нет своей полноценной операционной системы

Подытожим преимущества.

Преимущества контейнера перед виртуальным сервером:

- Для запуска и исполнения контейнеров используется одно и тоже ядро операционной системы. Это позволяет избежать накладных расходов связанных с симулированием оборудования а так же необходимостью запуска отдельной ОС для каждого контейнера, как это происходит в случае виртуального сервера.

- более быстрый запуск приложения (т.к. не необходимости ждать запуска гостевой операционки)

- так же образ контейнера весит меньше чем образ ВМ

- контейнер легче перенести и развернуть в новом окружении или масштабировать

Ну и чуть подробнее о самой механике создания контейнеров.

В докер есть два основных понятия. Docker-image и докер-контейнер

Докер-имэйдж это сборка (набор приложений, библиотек со всеми зависимостями и настройками. Т.е. уже готовое к запуску приложение.

Докер имэйдж создается на основе докер файла. В котором указывается весь этот софт со всеми зависимостями и настройками.

Докер контейнер – это уже работающее приложение, созданное на базе Docker-image

 это хранилище образов. Самым известным является DockerHub. Он напоминает GitHub, только содержит образы, а не исходный код.

**Изоляция процессов**

**На основании одного файла**

Докер позволяет изолировать любые процессы, в том числе и процессы запуска тестов. Значит, мы можем сколько угодно раз запустить наши тесты на нужной системе, так еще и сделать это быстрее, чем с ВМ (виртуальной машиной). И для этого нам потребуется компьютер меньшей мощности

**Быстрое разворачивание окружений**

Докер умеет быстро разворачивать систему. Можно запустить приложение на настройку и установку которого тратится много времени.

*Если нужно запустить тест, теперь не надо каждый раз разворачивать на свою машину testNG или selenium, достаточно скачать и запустить нужный вам докер-контейнер.*

И при этом потребление памяти будет куда ниже, чем у виртуалки. А, значит, контейнеров, запущенных одновременно, может быть сколько угодно.

**Чистота на рабочей машине**

Вы можете поддерживать чистоту на своей рабочей машине. Если нужно попробовать другую версию какой-то программы, не нужно будет ставить ее в параллели или удалять старую. Можно просто запустить ее в другом контейнере.

Докер состоит из нескольких компонентов. 1. Doker-deamon (ядро докера) это программа умеющая создавать образы или скачивать готовые из сети

Для их хранения используют репозитории. Докер представляет свой публичный репозиторий. Там содержатся огромной количество образов, которые можно использовать для контейнеризации.

Преимущества контейнера перед виртуальным сервером:

- Для запуска и исполнения контейнеров используется одно и тоже ядро операционной системы. Это позволяет избежать накладных расходов связанных с симулированием оборудования а так же необходимостью запуска отдельной ОС для каждого контейнера, как это происходит в случае виртуального сервера.

- более быстрый запуск приложения (т.к. не необходимости ждать запуска гостевой операционки)

- так же образ контейнера весит меньше чем образ ВМ

- контейнер легче перенести и развернуть в новом окружении или масштабировать

Минусы:

Чуть менее секьюрны. Т.к. нет дополнительного слоя изоляции в виде виртуального оборудования.

Докер состоит из трех основных компонентов.

1. Docker deamon - ядро докера, представляющее из себя программу умеющую создавать образы или скачивать готовые из сети. Так же демон запускает из образов контейнеры. Организует между ними сетевое взаимодействие и управляет томами, представляя контейнеру доступ на чтение или запись в файловой системе.

2. Следующий компонент это REST API— это, по сути, http интерфейс для передачи команд докеру.

3. Третий компонент — это консольный клиент CLI. Вызываемый с помощью команды докер Он взаимодействует с демоном докера через REST API