



docker

Что такое докер

Платформа, которая поможет:

- 1) Упаковать приложение и его зависимости в контейнер
- 2) Перенести его на любой сервер, на котором установлен докер
- 3) Запустить контейнер с приложением за доли секунды

Это виртуальная машина?

Нет. Оно выглядит и работает как виртуальная машина, но:

- ❖ Докер намного быстрее
- ❖ Сам контейнер меньше в размерах

Зачем это нужно

- ❖ Простой способ доставки продукта по серверам
- ❖ Контейнер защищает продукт от нюансов конфигурации каждого принимающего сервера
- ❖ Контейнер защищает сервер от багов и дыр в безопасности продукта

Создатели видят это так

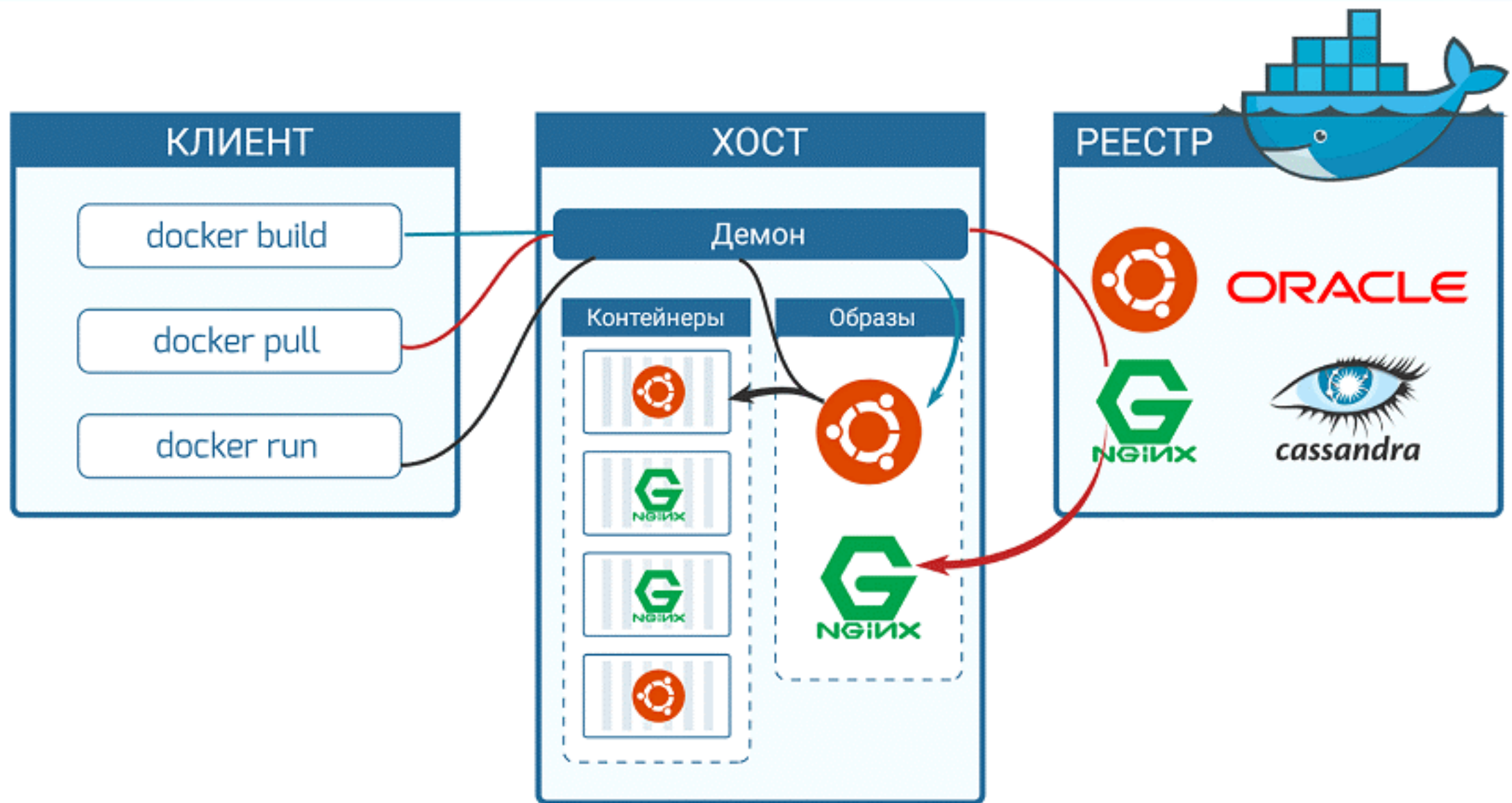


Схема работы

- ❖ Пользователь отдает команду с помощью клиентского интерфейса Docker-демону, развернутому на [Docker-хосте](#). Например, скачать готовый образ из реестра (хранилища Docker-образов) с помощью команды **docker pull**. Взаимодействие между клиентом и демоном обеспечивает REST API. Демон может использовать публичный ([Docker Hub](#)) или частный реестры.
- ❖ Исходя из команды, заданной клиентом, демон выполняет различные операции с образами на основе инструкций, прописанных в файле [Dockerfile](#). Например, производит их автоматическую сборку с помощью команды **docker build**.
- ❖ Работа образа в контейнере. Например, запуск [docker-image](#), посредством команды **docker run** или удаление контейнера через команду **docker kill**.

И докер очень простой

❖ Запустить Ubuntu и выполнить в ней echo

```
docker run ubuntu echo "hello virtual world"
```

❖ Запустить CentOS с башем и подключиться к его терминалу

```
docker run -ti CentOS /bin/bash
```

Очень-очень простой

- ❖ Запустить nodejs-app контейнер в бэкграунде и разрешить доступ к нему по 80му порту

```
docker run -p 80:80 -d nodejs-app
```


Как приоткрыть контейнер

- ❖ Открыть конкретные порты (host:container)

```
docker run -p 8080:80 -p 4443:4443 ...
```

- ❖ Или сразу все

```
docker run -P ...
```

Подключить FS

- ❖ Подключить папку файловой системы хоста для чтения

```
docker run -v ~/www:/var/www:ro ...
```

- ❖ Сделать контейнер-хранилище, и раздавать его файловую систему остальным

```
docker create -v /temp/db --name dbhost
```

```
docker run --volumes-from dbhost ...
```

Как его раздобыть

❑ Нативно работает на ubuntu, debian, CentOS, etc.

apt-get install docker.io

❑ Для Windows и Mac OS X

boot2docker

boot2docker

VirtualBox, который притворяется консольной утилитой

```
[pav@pav-macbookpro ~]$ uname -a
Darwin pav-macbookpro.local 13.4.0 Darwin Kernel Version 13.4.0:
[pav@pav-macbookpro ~]$ boot2docker start
Waiting for VM and Docker daemon to start...
.....0000
Started.
[pav@pav-macbookpro ~]$ docker --version
Docker version 1.4.1, build 5bc2ff8
[pav@pav-macbookpro ~]$ boot2docker stop
[pav@pav-macbookpro ~]$
```

Важно!

- ❖ Если открыть любой порт контейнера, работая с boot2docker, он откроется в Virtual Box, не на хосте
=> заходим по IP виртуалки, не 127.0.0.1

boot2docker ip

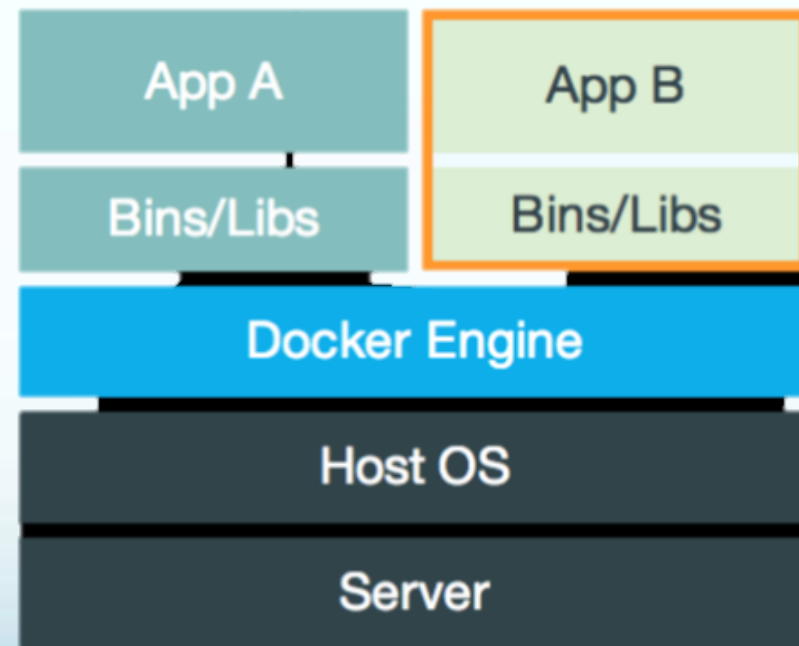
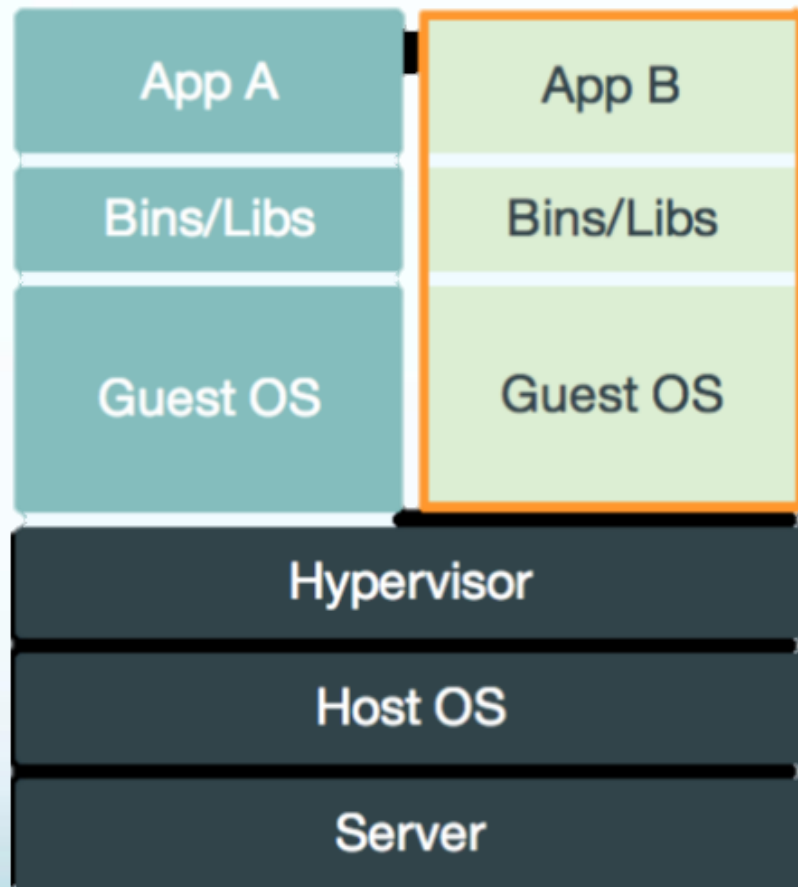
VM и Docker

- ❖ Контейнер **!=** виртуальная машина
- ❖ Docker **!=** гипервизор
- ❖ Виртуальная машина **эмулирует** всё, в т.ч. железо
- ❖ Контейнер **изолирует** процессы, пользователей и файловую систему

Поэтому

- ❖ Запуск контейнера занимает доли секунды
- ❖ У контейнеров общее Linux ядро хоста
=> нативно запускается только на Linux машинах
- ❖ Маленький размер образа
- ❖ На одной машине можно запустить раз в 6 больше контейнеров, чем виртуальных машин

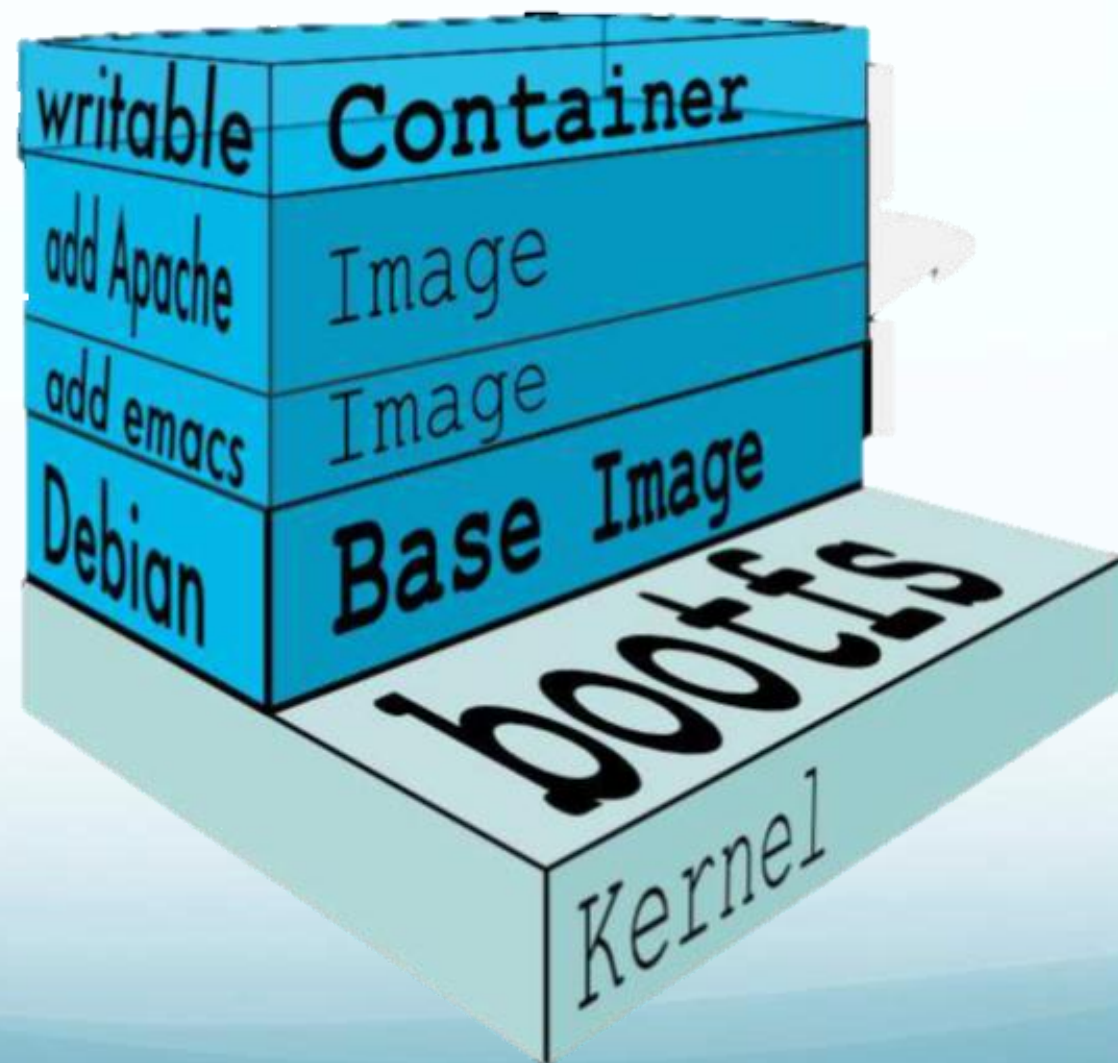
Bot так



Терминология

- ❖ Слой (layer) – слепок файловой системы
- ❖ Образ (image) – readonly слой + ссылка на родительский image
- ❖ Базовый образ – образ без родителя
- ❖ Контейнер – запущенный образ + writeable слой файловой системы + метаданные

Всё вместе



Docker и Vagrant

- ❖ Немного разные фокусы:
 - Docker – контейнерная доставка софта
 - Vagrant – конфигурация и развертывание рабочей станции, используя VM либо контейнер провайдеров
- ❖ Vagrant может использовать Docker в качестве провайдера

Как создать/сохранить образ

- ❖ Взять готовый с репозитория и изменить
`docker pull, docker commit`
- ❖ Собрать из Dockerfile на основе существующего
`docker build`
- ❖ Создать новый с нуля
`docker create, docker import`

Docker Hub

Почти как github, только для образов

- Бесплатные публич репозитории + 1 приватный
- `docker pull ubuntu`
- `docker push betterUbuntu`

Изменить image

- ❖ Образы не меняются, но после завершения контейнера можно найти его r/w слой и сохранить

```
docker run ubuntu mkdir ~/mydir
```

```
docker ps -a
```

```
docker commit <ID> image_name
```

Сохранить/загрузить

```
docker save image > /fdd/image.tar
```

```
docker load < /fdd/image.tar
```

Спасибо за внимание