Установка PostgreSQL

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо
“-”, если есть проблемы

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в учебной группе ТГ

Задаем вопрос
в чат или голосом

Вопросы вижу в чате,
могу ответить не сразу

Маршрут вебинара

1.Установка из пакетов

2.Обновление кластера на новую версию

3. Yandex Cloud

4. Docker & Docker-compose

Цели вебинара

К концу занятия вы сможете

1. Разворачивать кластер PostgreSQL различными способами;
2. Знать возможности и ограничения каждого способа;
3. Выбирать наиболее оптимальный способ в зависимости от задачи.

Смысл

Зачем вам это уметь

1. Чтобы самим развернуть кластер в любой среде;
2. При наличии выбора сделать наиболее оптимальный;

Что мы хотим сделать

- создаем кластер;
- там где можно меняем имя кластера на test;
- делаем все необходимое для удаленного подключения;
- подключаемся удаленно.

Установка PostgreSQL из пакетов

PostgreSQL из пакетов

- sudo apt install postgres

кастомный PostgreSQL из пакетов

- https://www.postgresql.org/download/linux/ubuntu/
- после установки останавливаем и удаляем кластер
- создаем заново с именем test
- настраиваем для удаленного подключения
- netstat -na|grep 5432
- /etc/postgres/13/test/postgresql.conf
- pg_hba.conf
- пароль postgres
- рестарт кластера

psql -h 104.197.151.20 -U postgres

Установка PostgreSQL из исходников

PostgreSQL из исходников

для гурманов и применения кастомных патчей - патч заменяет код C на свой

самый большой минус - апгрейд версии постгреса

https://github.com/AbdulYadi/postgresql-private

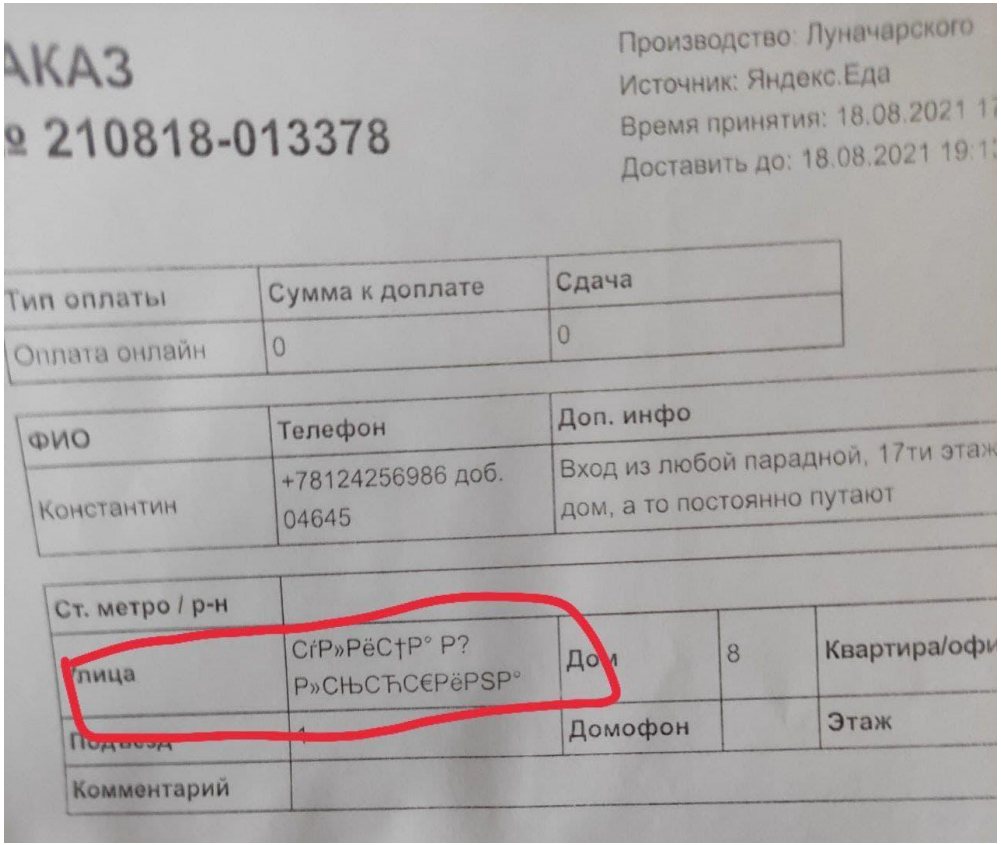
INSTALL Short Version

```
./configure
make
su
make install
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data -l logfile start
/usr/local/pgsql/bin/createdb test
/usr/local/pgsql/bin/psql test
```

Кодировка при установке

Кодировка

Рекомендую только UTF!!!



АКАЗ
№ 210818-013378

Производство: Луначарского
Источник: Яндекс.Еда
Время принятия: 18.08.2021 1
Доставить до: 18.08.2021 19:1

Тип оплаты	Сумма к доплате	Сдача
Оплата онлайн	0	0

ФИО	Телефон	Доп. инфо
Константин	+78124256986 доб. 04645	Вход из любой парадной, 17ти этаж дом, а то постоянно путают

Ст. метро / р-н				
Улица	СґР»РёС†Р° Р? Р»СЊСЋС€РёРSР°	Дом	8	Квартира/офи
Подъезд	1	Домофон		Этаж
Комментарий				

Вопросы?

Yandex cloud

минитест

https://forms.gle/xmRfpitGit8DCUB57

5 минут

Docker & Docker-compose

Виртуализация

Виртуализация

- Виртуализация - программная имитация аппаратного обеспечения

Оптимизация

- Аппаратных ресурсов(больше на одном сервере)
- Стоимости(сокращение кол-ва серверов)

Изоляция

- От чужих зависимостей
  - Поддержка окружений с множеством приложений и сервисов со временем становится головной болью администраторов
- От других приложений
  - Тонкая конфигурация и оптимизация операционной системы требует человеческих ресурсов
- От сторонних пользователей

Эмуляция

- Другая ОС (Windows, Linux, Solaris…)
- Другая платформа (ARM, MIPS…)

Эмуляция – попытка зеркально сымитировать внутреннее устройство эмулируемой системы таким образом, чтобы программа, отвечающая за эмуляцию какой либо из систем, в точности повторяла все ее процессы и работу компонентов эмулируемой системы.

Типы виртуализации

- Программная виртуализация:
  - Динамическая трансляция(VirtualBox)
  - Паравиртуализация(Xen)
- Аппаратная виртуализация (KVM, Xen, VMware, Hyper-V)

Контейнеризация

- Контейнеризация (LXC, OpenVZ, Jail, Zones) — виртуализация на уровне операционной системы.

Как работает Docker

Как работает Docker

- Namespaces
- Cgroups
- UnionFS
- RunC

Namespaces

- Изолирование окружения
- Каждый контейнер работает со своими namespace’ами
  - pid : Изоляция процессов (PID: Process ID)
  - net : Изоляция сетей (NET: Networking)
  - ipc : Изоляция IPC (IPC: InterProcess Communication)
  - mnt : Изоляция файловой системы (MNT: Mount)
  - uts : Изоляция UTS (UTS: Unix Timesharing System)
  - user: Изоляция пользователей
- **Namespace закрывается, если PID 1 умер**

Control groups

- Позволяет контейнерам использовать общие ресурсы
- Ограничивает набор доступных ресурсов
- ЦПУ, память, IO …

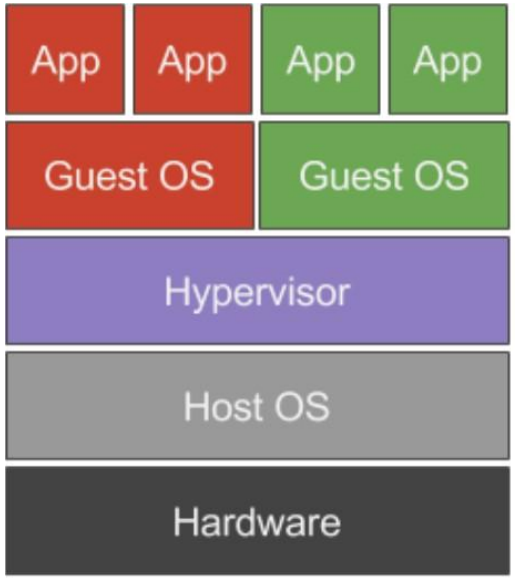
Union File Systems

- Разделение по слоям
- Переиспользование слоев

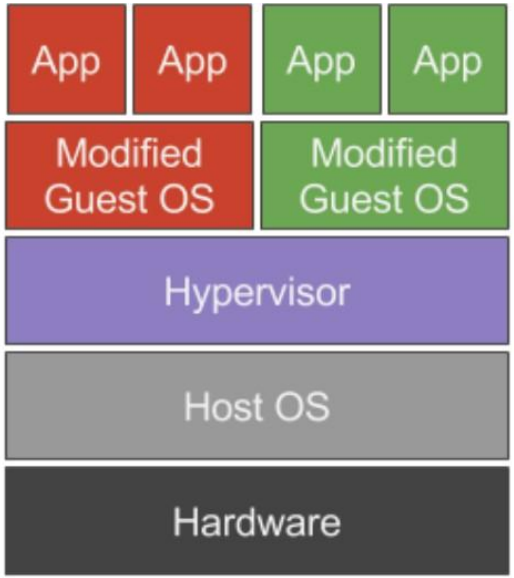
RunC

- Библиотека-обертка над Namespaces, cgroups, UnionFS

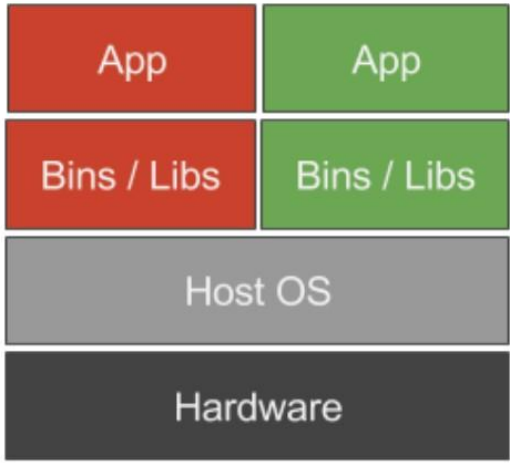
Типы виртуализации



Full Virtualization



Paravirtualization



OS Level virtualization

Контейнеризация

- Существовала достаточно давно
- Не получила широкого распространения
- В определенных случаях заменила аппаратную виртуализацию
- **Почему выстрелил именно Docker?**

Docker

- Не столько про контейнеры (как технологию)
- Хотя использует контейнеризацию как основу

Docker

- Абстракция от host-системы
- Легковесное изолированное окружение
- Общие слои файловой системы
- **Компоновка и предсказуемость**
- **Простое управление зависимостями**
- **Дистрибуция и тиражируемость**

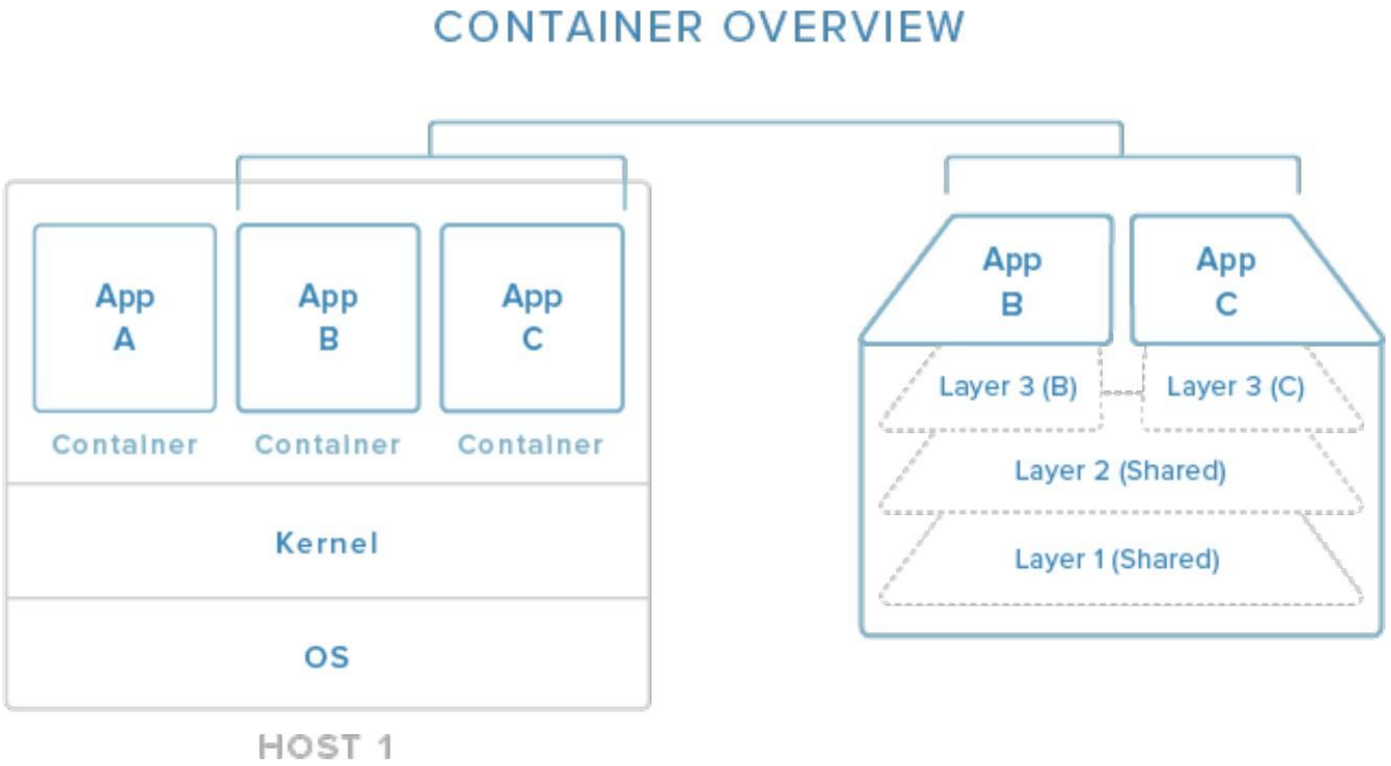
Docker это про стандарты

- Стандартизация описания окружения, сборки, деплоймента
- Стандартизированная дистрибуция
- 100% консистентная(иммутабельная) среда приложения
- Воспроизводимость (DEV->QA->Production)
- Zero time deployment

Docker

- Контейнер — это **НЕ** виртуальная машина, а приложение и его зависимости упакованные в стандартизированное, изолированное, легковесное окружение.

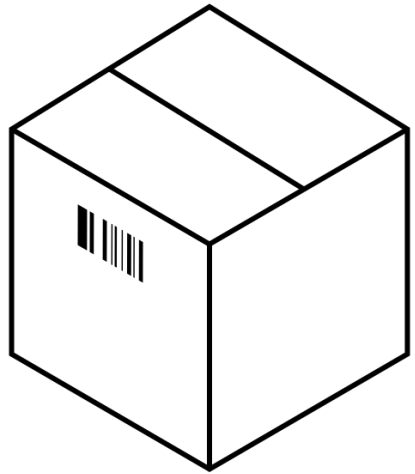
Docker


CONTAINER OVERVIEW
App A
App B
App C
Container
Container
Container
Kernel
OS
HOST 1
App B
App C
Layer 3 (B)
Layer 3 (C)
Layer 2 (Shared)
Layer 1 (Shared)

Что внутри?

Docker:

- App (your Java/Ruby/Go/... app)
- Libraries (libxml, wkhtmltopdf, ..)
- Services (postgresql, redis, ...)
- Tooling (sbt, ant, gems, eggs, ...)
- Frameworks&runtime (jre, ruby, ...)
- OS packages (libc6, tar, ps, bash, ...)



Created by Grant Fisher
from Noun Project

Из чего состоит Docker

Из чего состоит Docker?

- Daemon
- Client
- Registry

Docker daemon

- Предоставляет API
- Управляет Docker-объектами
- Общается с другими docker daemon'ами

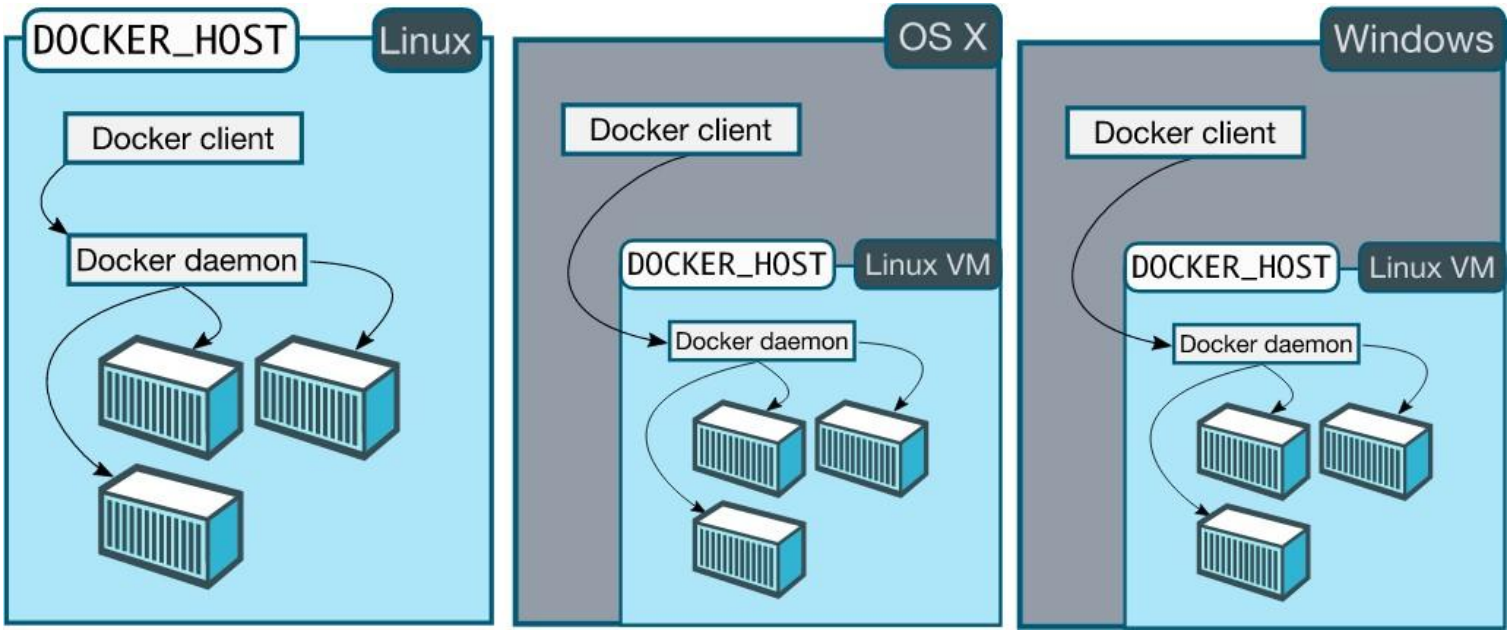
Docker daemon

- Запускается на хост машине, где планируется запускать контейнеры
- Хост машина – vm, физический сервер(x86, arm64), aws ec2, ваш ноутбук, raspberry pi …

Docker client

- Принимает команды пользователя
- Общается по API с docker daemon’ом
- Может общаться с несколькими daemon’ами

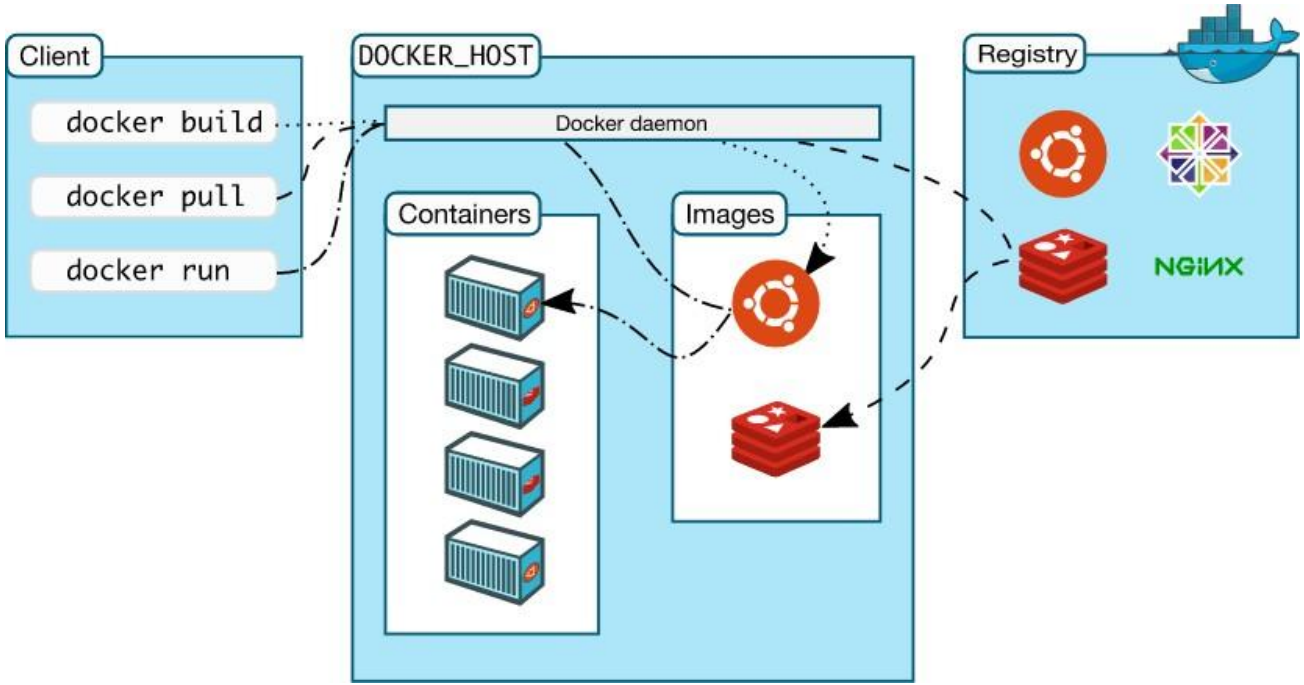
Docker engine



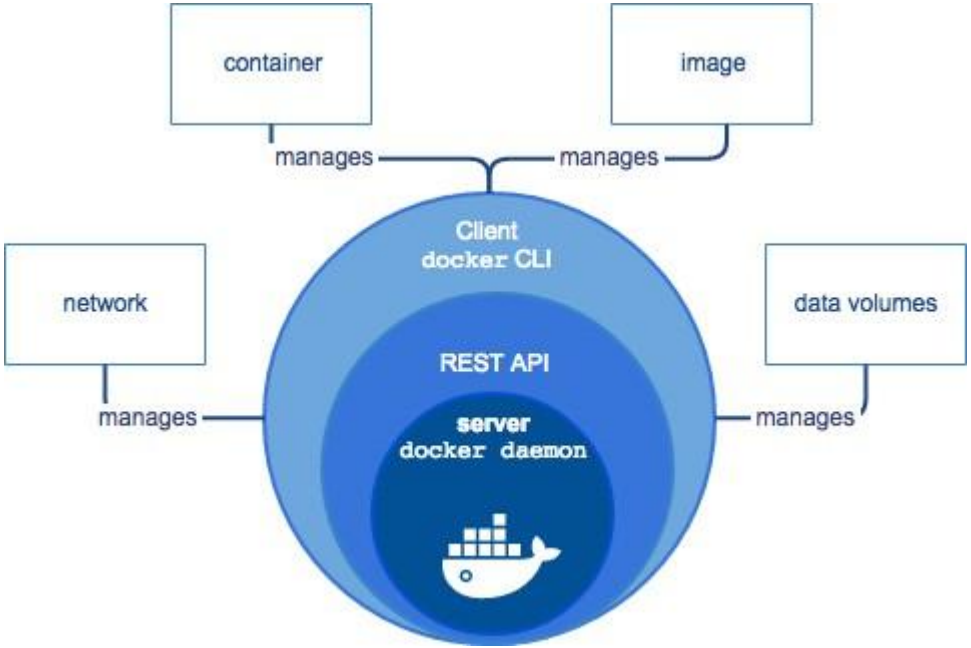
Docker registry

- Docker Hub
- Private Registry
- Docker Trusted Registry
- Docker store

Docker engine



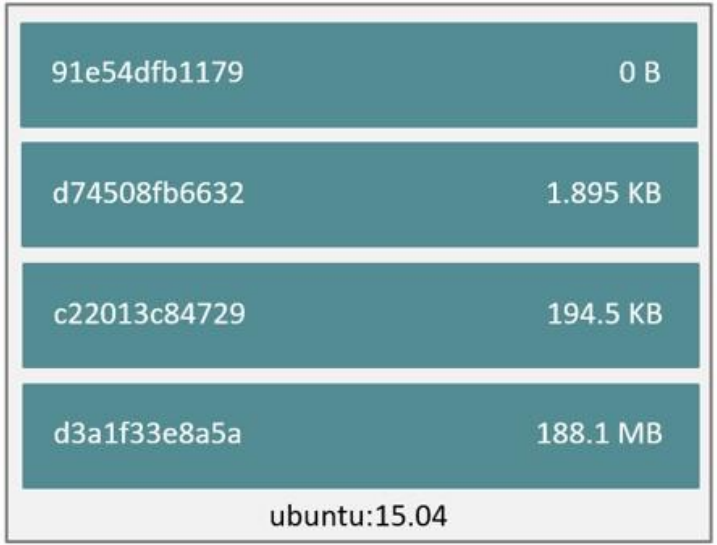
Объекты Docker



Docker images

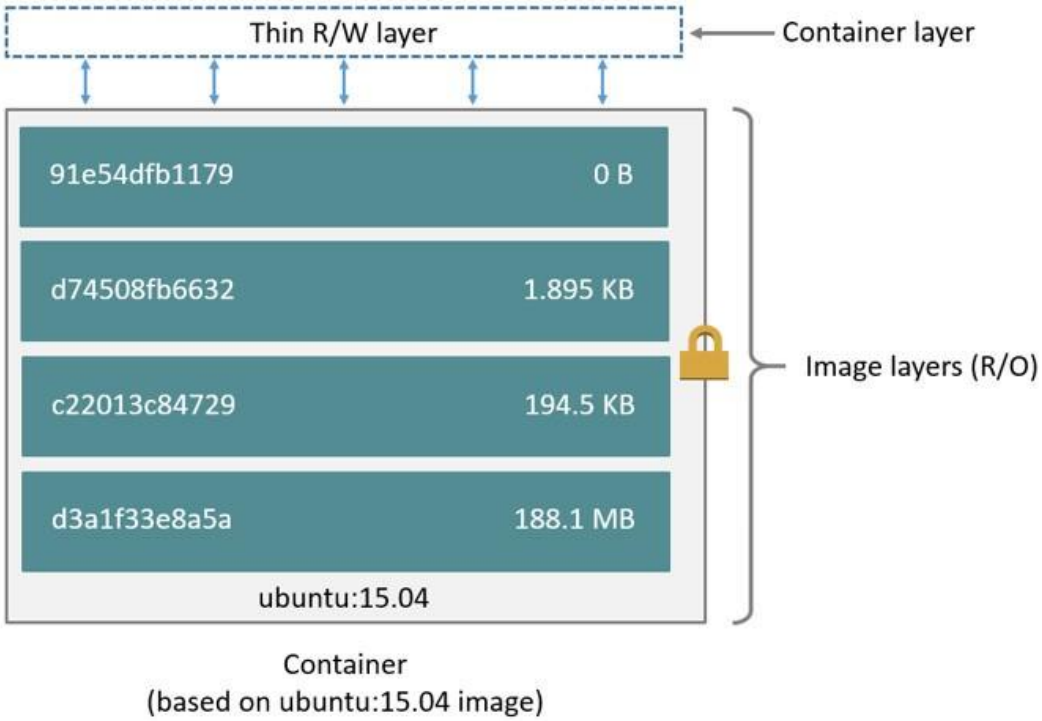
- image – неизменяемая сущность, snapshot контейнера
- image состоит из слоев(layers)
- layers – read-only diff изменений файловой системы

Docker images

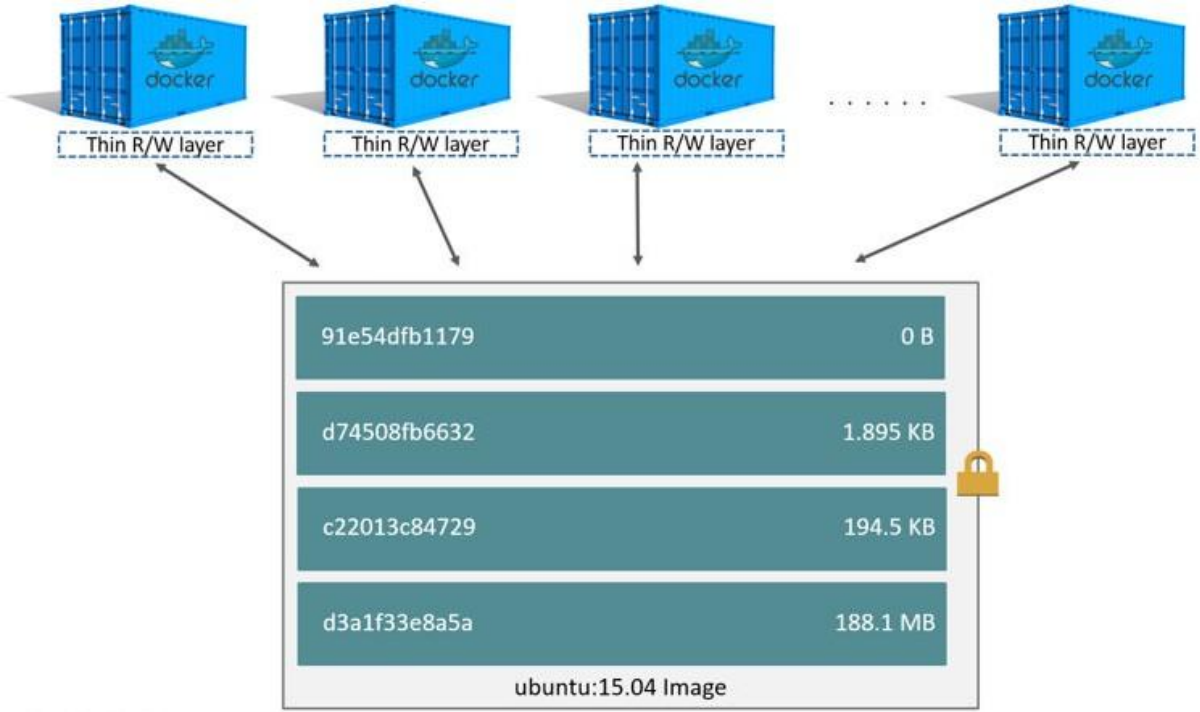


Image

Docker images



Docker containers



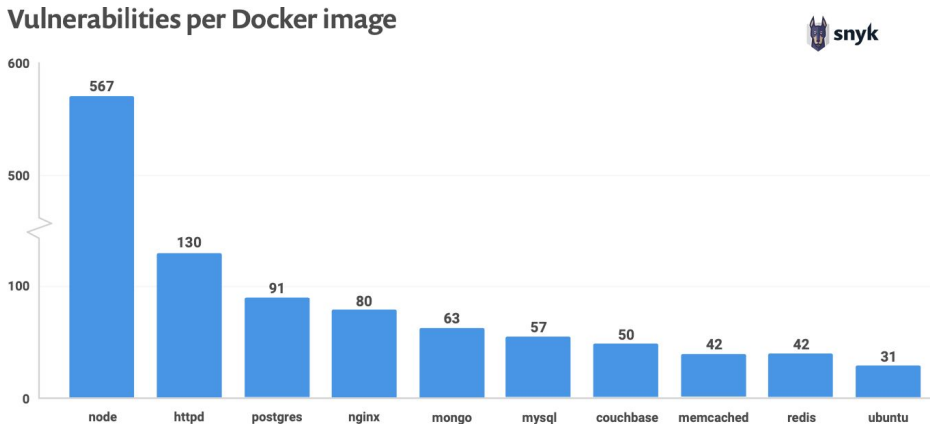
Безопасность

Безопасность

- Docker — это всего лишь тонкая прослойка
- Привилегии пользователей ограничены Whitelистом
- Можно включить user-namespace
- Не запускайте приложения от root
- Надо заниматься патч-менеджментом
- Не использовать передачу паролей и прочих секретов через ENV-переменные

State of Docker security (2019)

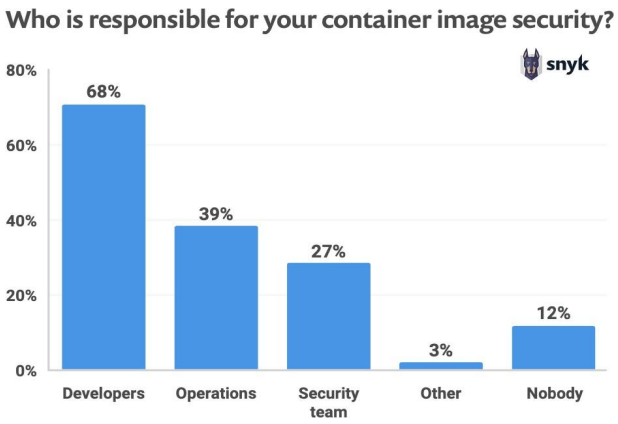
- В настоящее время Docker Hub содержит:
- 223 images от проверенных издателей 151 официальный image
- 40 сертифицированных docker images

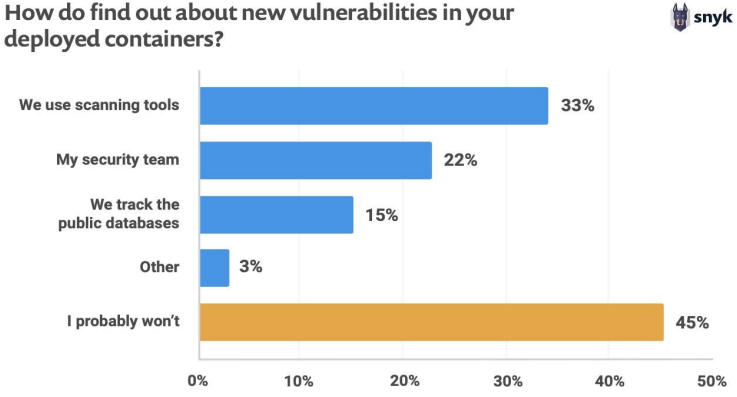


Отчет Snyk

Docker security

- **68%** разработчиков сами отвечают за безопасность контейнеров
- **50%** не сканируют *operating system (OS) layer* в docker image
- **45%** никогда не находят новые уязвимости в их контейнерах
- **80%** не тестируют свои Docker images во время разработки





Как улучшить Docker security

- Правильный выбор base image
  - используйте самое минимальный image: *не упаковывайте то, что вам не нужно!*
- Использование multi-stage сборок
  Пересборка images
  - при пересборке образов
    используйте --no-cache
- Сканирование images на стадии development
  Сканирование контейнеров в production

10 Docker Image Security Best Practices

Ссылки

- IBM Research Report
- “Проникновение в Docker”
- "Контейнеры в Linux"

Практика

- PostgreSQL docker images
  - https://github.com/bitnami/bitnami-docker-postgresql
  - https://github.com/GoogleCloudPlatform/postgresql-docker
- PostgreSQL K8s yaml files
  - https://github.com/GoogleCloudPlatform/click-to-deploy/tree/master/k8s/postgresql
- PostgreSQL Helm charts
  - https://github.com/bitnami/charts/tree/master/bitnami/postgresql-ha
- PostgreSQL K8s operator
  - https://github.com/CrunchyData/postgres-operator
  - https://postgres-operator.readthedocs.io/en/latest/

Практика

Docker https://docs.docker.com/engine/install/ubuntu/

1. Создаем docker-сеть:

$ sudo docker network create pg-net

2. подключаем созданную сеть к контейнеру сервера PostgreSQL:

$ sudo docker run --name pg-docker --network pg-net -e POSTGRES_PASSWORD=postgres -d -p 5432:5432 -v /var/lib/postgres:/var/lib/postgresql/data postgres:14

3. Запускаем отдельный контейнер с клиентом в общей сети с БД:

$ sudo docker run -it --rm --network pg-net --name pg-client postgres:14 psql -h pg-docker -U postgres

4. Проверяем, что подключились через отдельный контейнер:

$ sudo docker ps -a

5. Радуемся жизни

Вопросы?

Копирование файлов БД

Копирование файлов БД

- Будет в ДЗ
- Также можно между разными дистрибутивами линукса на остановленном кластере.
- Единственное файлы конфигурации могут быть в другом месте и должна быть **та же основная версия** Постгреса

Рефлексия

https://docs.google.com/forms/d/1XBIFMcSns3l35Qy00EbxVG9qS4pD0reiX-Rpdfz3iC0/edit#responses

Вопросы?

- Какие варианты установки PostgreSQL вы запомнили?
- Что узнали нового?

Домашнее задание

ДЗ

- создать ВМ с **Ubuntu** 20.04/22.04 или развернуть **докер** любым удобным способом
- поставить на нем Docker Engine
- сделать каталог /var/lib/postgres
- развернуть контейнер с PostgreSQL 15 смонтировав в него /var/lib/postgresql
- развернуть контейнер с клиентом postgres
- подключится из контейнера с клиентом к контейнеру с сервером и сделать таблицу с парой строк
- подключится к контейнеру с сервером с ноутбука/компьютера извне инстансов GCP/ЯО/места установки докера
- удалить контейнер с сервером
- создать его заново
- подключится снова из контейнера с клиентом к контейнеру с сервером
- проверить, что данные остались на месте
- оставляйте в ЛК ДЗ комментарии что и как вы делали и как боролись с проблемами

ДЗ

1. Выполнение ДЗ: 10 баллов
   - + 2 балла за красивое решение
   - - 2 балла за рабочее решение, и недостатки указанные преподавателем не устранены

2. Рекомендуемый путь задавать вопросы:
   - в чат дз в ЛК Отуса
   - или общий чат ТГ без тега преподавателя

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары

Ведущий разработчик PostgreSQL/Greenplum в Сбере

Специалист в области разработки и проектировании витрин данных в PostgreSQL/Greenplum, а также в области разработки хранимых процедур в таких СУБД как PostgreSQL/Greenplum, Oracle, MS SQL Server.