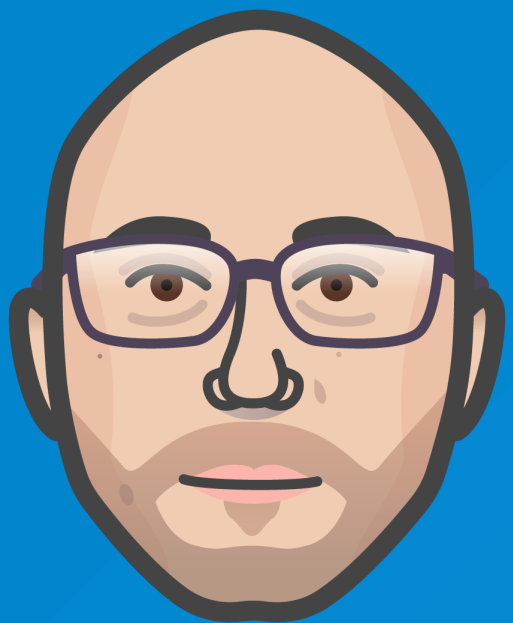


Taller Docker





by David Carvajal

<https://github.com/davcd/coders-docker>

Índice

- Introducción
- Imágenes
- Contenedores
- Avanzado

Índice

- **Introducción**

- Maquinas Virtuales vs. Contenedores
- Contenedores y microservicios
- ¿Qué es Docker?
- Características
- Componentes
- Docker-Hub
- Comandos básicos

Índice

- **Imágenes**
 - Definición
 - Comandos
 - Dockerfile
 - Build caché
 - Buenas Prácticas

Índice

- **Contenedores**
 - ¿Qué es docker-compose?
 - docker-compose.yml
 - Comandos
 - Volúmenes
 - Redes

Índice

- **Avanzado**
 - Gestión de clusters

Índice

- **Introducción**

- Maquinas Virtuales vs. Contenedores
- Contenedores y microservicios
- ¿Qué es Docker?
- Características
- Componentes
- Docker-Hub
- Comandos básicos

Maquinas Virtuales vs. Contenedores

“ En informática, una máquina virtual es un software que simula un sistema de computación y puede ejecutar programas como si fuese una computadora real.

-- [Máquina Virtual - Wikipedia](#)

”

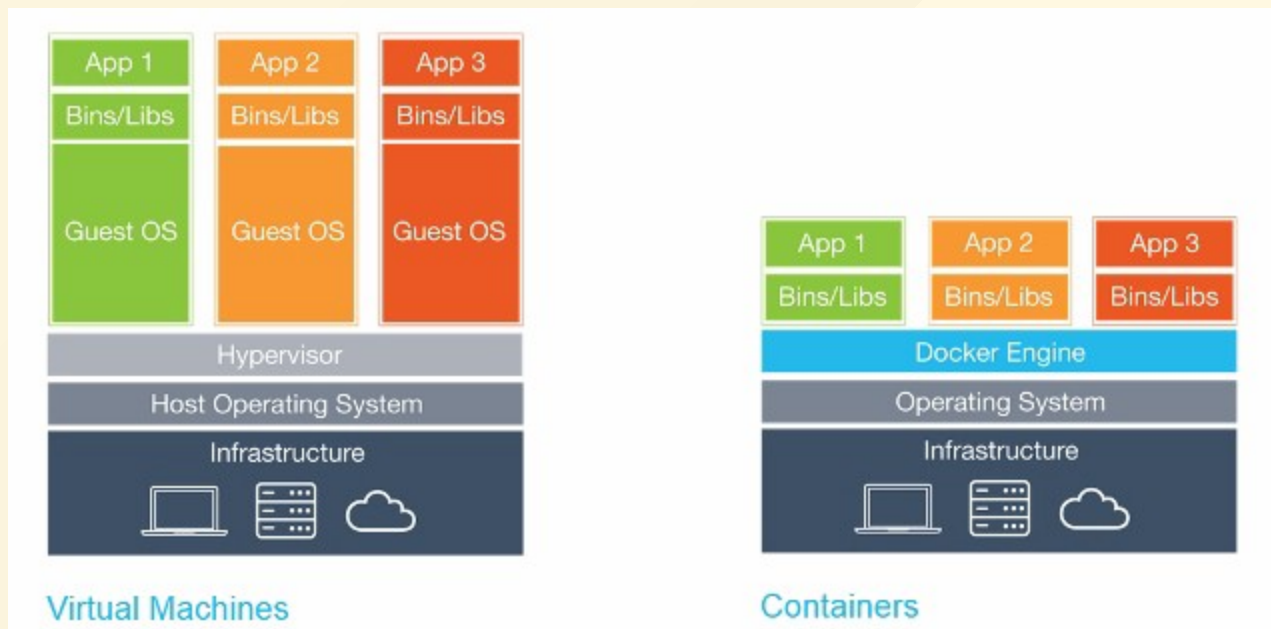
Maquinas Virtuales vs. Contenedores

“ Un contenedor es una unidad estándar de software que empaqueta todas sus dependencias para que la aplicación se ejecute de forma rápida y confiable de un entorno informático a otro.

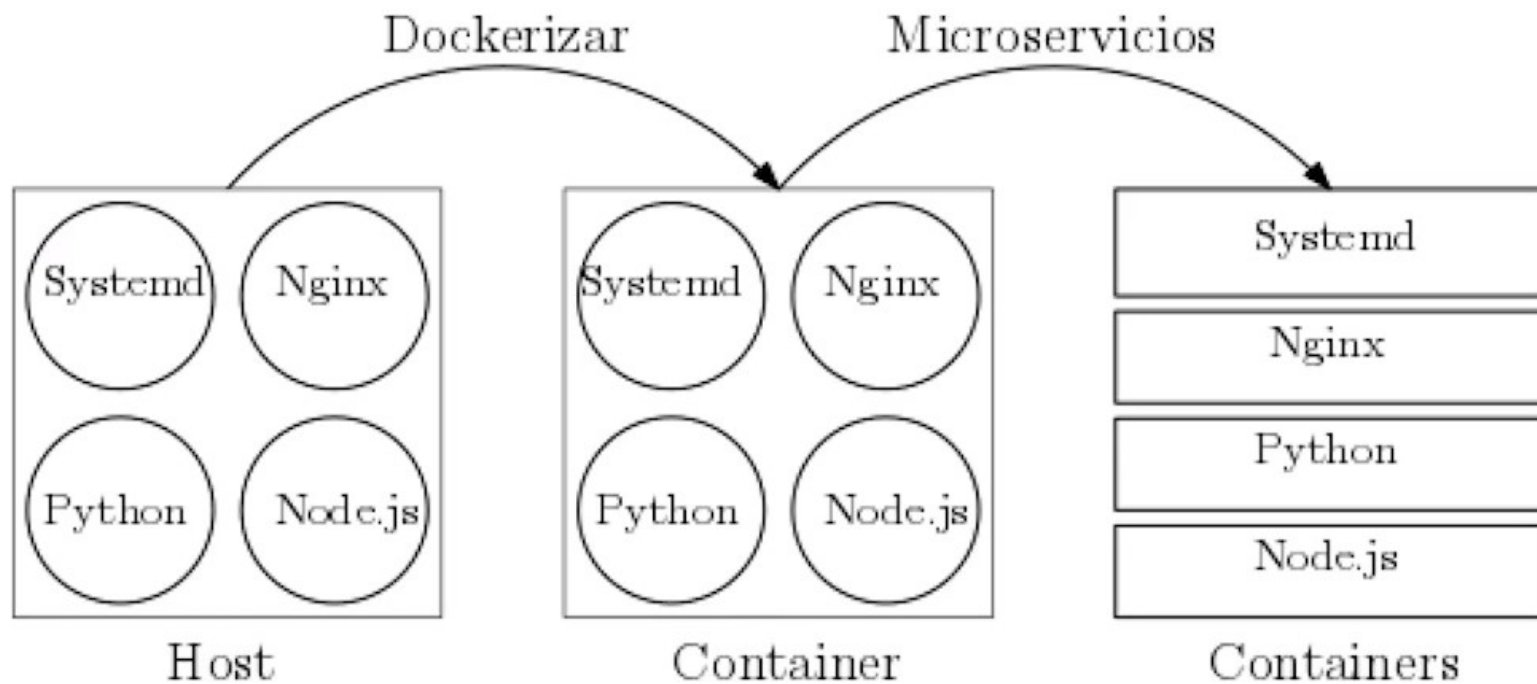
-- [Docker Resources](#)

”

Maquinas Virtuales vs. Contenedores



Contenedores y microservicios



¿Qué es Docker?

“ Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de **contenedores de software**, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

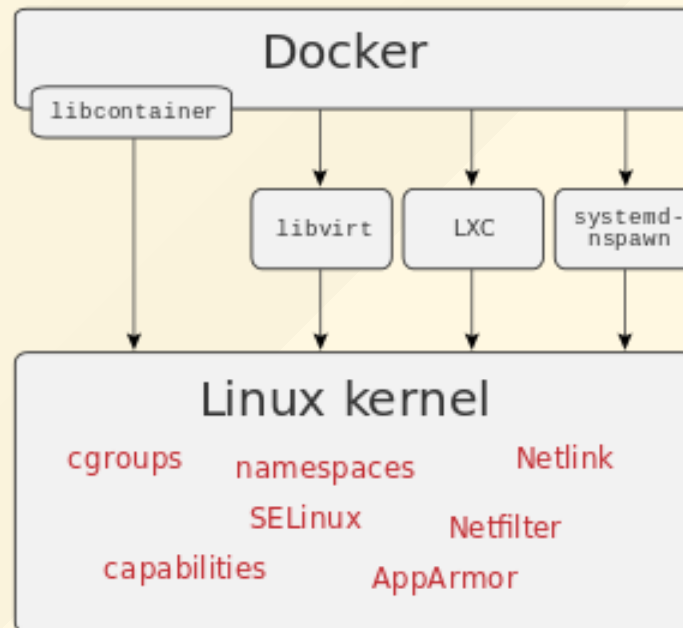
-- [Docker \(software\) - Wikipedia](#)

”

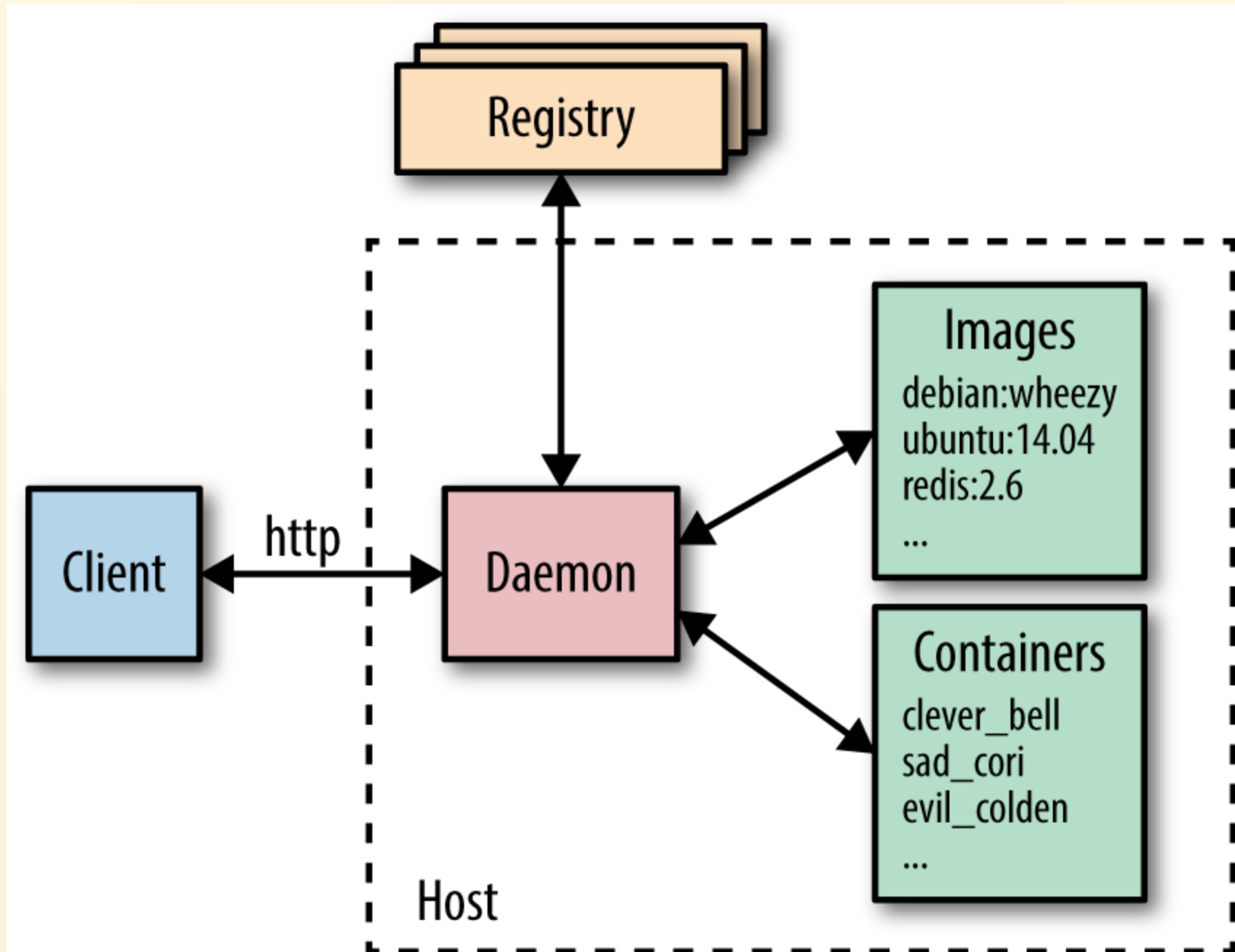
Características

- Docker Inc.
- Open Source
- Primera release 2013
- Community Edition y Enterprise Edition
- "Build, Ship and Run. Any application, Anywhere"
- Portabilidad, Inmutabilidad y Ligereza

Características



Componentes



Docker Hub

El registro oficial de Docker

Comandos Comunes

Índice

- **Imágenes**
 - Definición
 - Comandos
 - Dockerfile
 - Build caché
 - Buenas Prácticas

Definición

Una Imagen es la información necesaria para arrancar un contenedor. Se compone de un sistema de ficheros y otros metadatos:

Comandos

- *docker build*
- *docker login*
- *docker pull*
- *docker image ls*
- *docker inspect*
(ó *docker inspect -f '{{.Size}}' imagen*)
- *docker image rm*
(ó *docker rmi*)

Dockerfile

```
FROM ubuntu:latest
RUN apt-get update -y
RUN apt-get install -y python-pip python-dev
WORKDIR /app
ENV DEBUG=True
EXPOSE 80
VOLUME /data
COPY . /app
RUN pip install -r requirements.txt
ENTRYPOINT ["python"]
CMD ["app.py"]
```

Build caché

- Proceso costoso y repetitivo
- Imagen base (*docker build --pull*)
- Capas
- Invalidaciones de cache (*docker build --no-cache*)
- Instrucciones ADD y COPY

Buenas Prácticas

1. Usa `.dockerignore`
2. Reduce el tamaño de tus imágenes al mínimo (alpine)
3. Ejecuta sólo un proceso por contenedor
4. Minimiza el número de capas de tu imagen.
5. Optimiza el uso de la cache .
6. Parametriza tus Dockerfiles usando argumentos

Índice

- **Contenedores**
 - ¿Qué es docker-compose?
 - docker-compose.yml
 - Comandos
 - Volumenes
 - Redes

docker-compose

Proyecto open source que permite definir aplicaciones multi-contenedor de una manera sencilla y declarativa. Es una herramienta ideal para gestionar entornos de desarrollo.

Dockerfile

```
db:
  image: mysql:5.7
  restart:always
  enviroment:
    MYSQL_ROOT_PASSWORD: password

wordpress:
  image: wordpress:latest
  depends_on:
    - db
  ports:
    - "8000:80"
```

Comandos

- *docker-compose up -d*
- *docker-compose pull*
- *docker-compose build*
- *docker-compose push*
- *docker-compose run*
- *docker-compose rm*

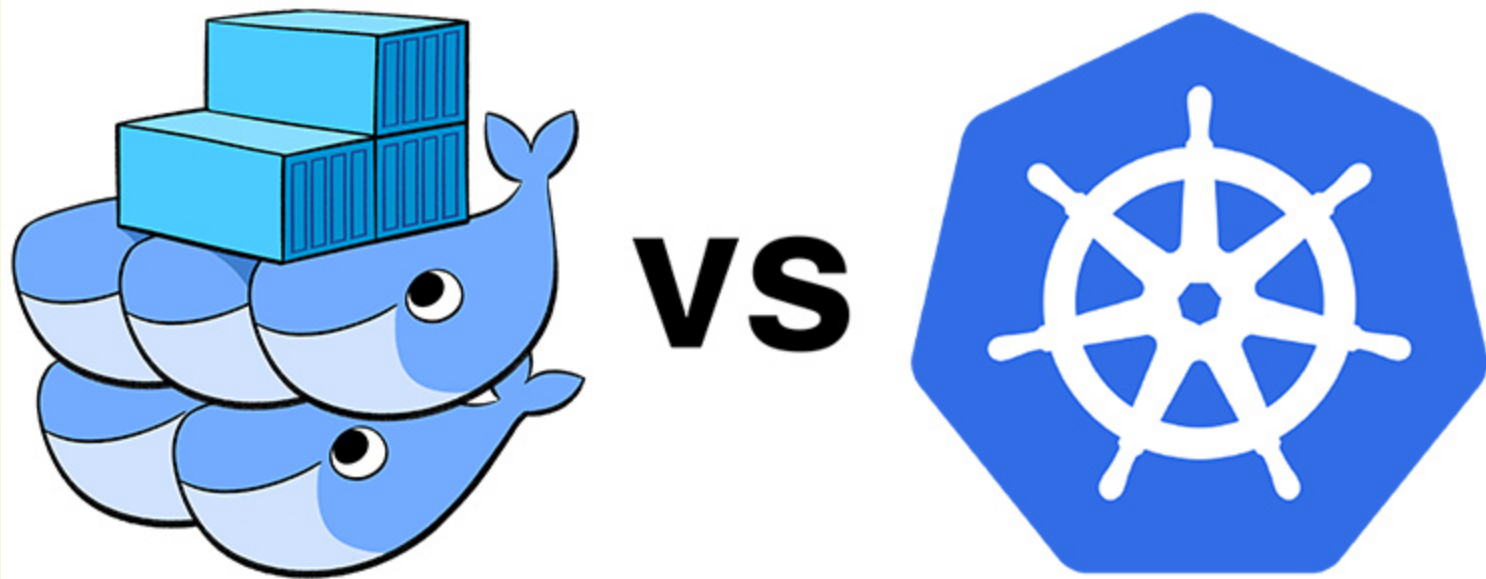
Redes

- Bridge
- Host
- None
- docker network (create | ls | inspect | rm)

Índice

- **Avanzado**
 - Gestión de clusters

Gestión de clusters



FIN