

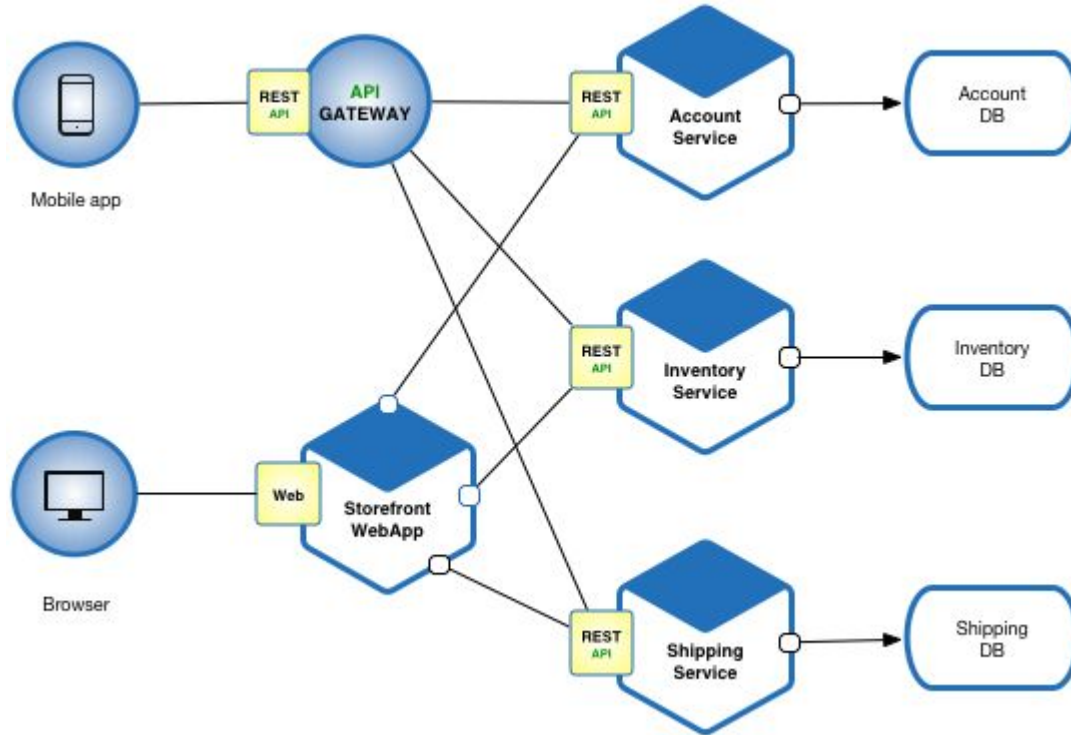
# Microservicios

Docker - Orquestadores

Lucas Bullian

David Wolovelsky

# Microservicios - Introducción

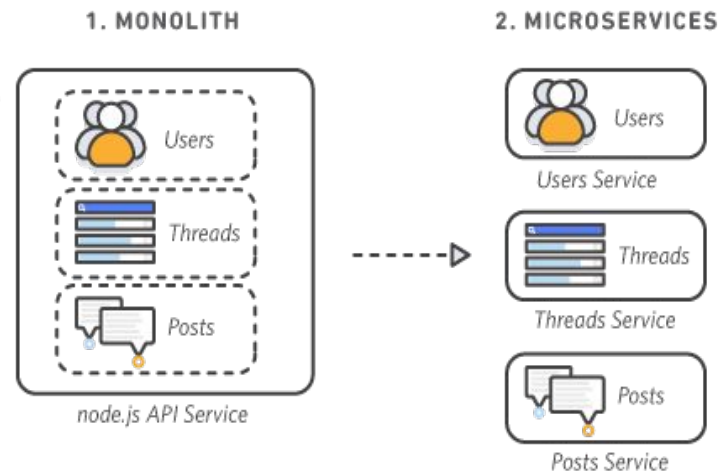


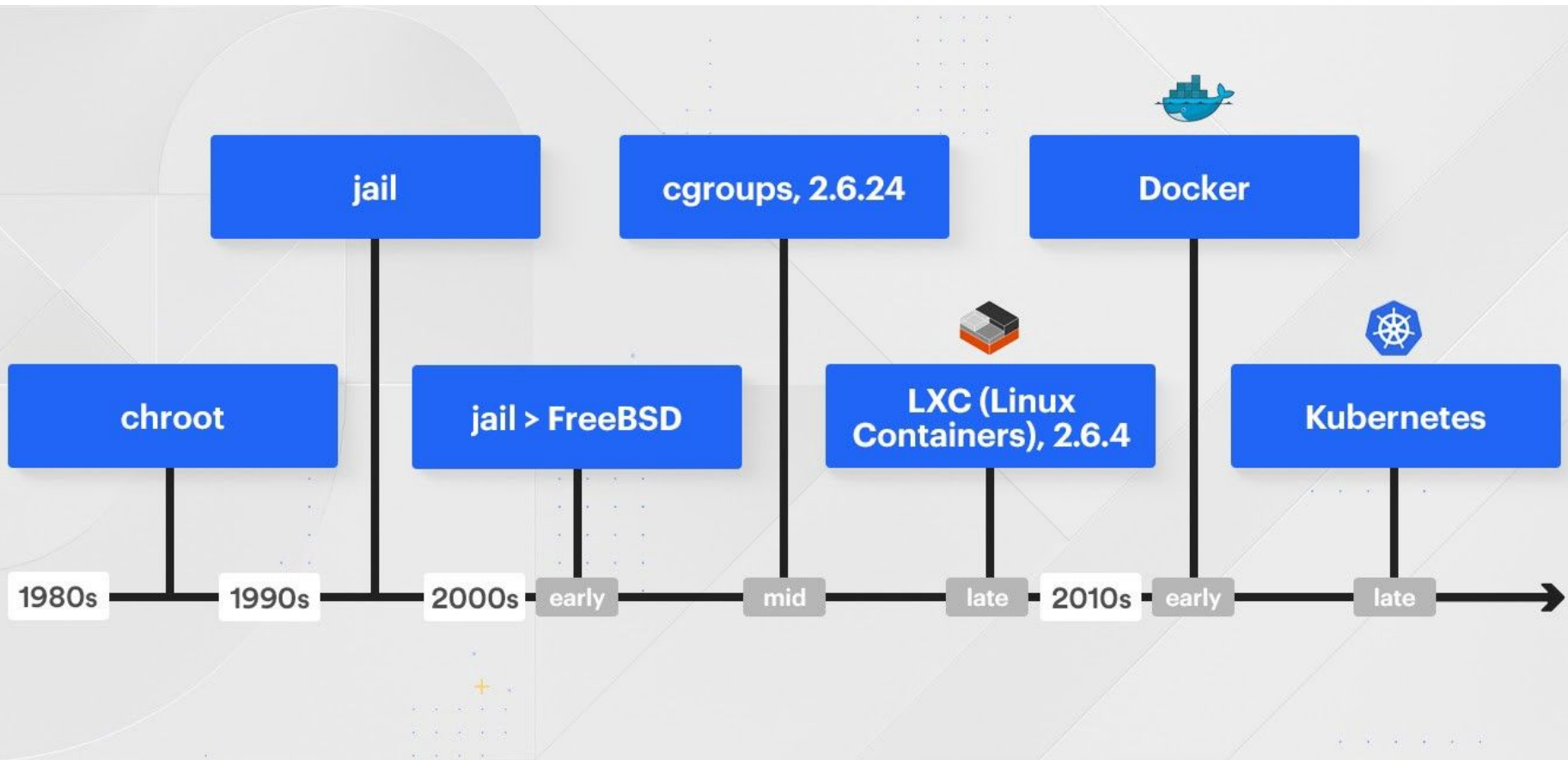
# Microservicios - Introducción

- Estilo para arquitectar aplicaciones de software
- Sin integración compleja entre servicios
- Se aplican independientemente
- Hacen una “sola” función específica
- Son independientes del OS host
- Son reutilizables

Treat as cattle, not as pets

Tratarlos como ganado, no como mascotas



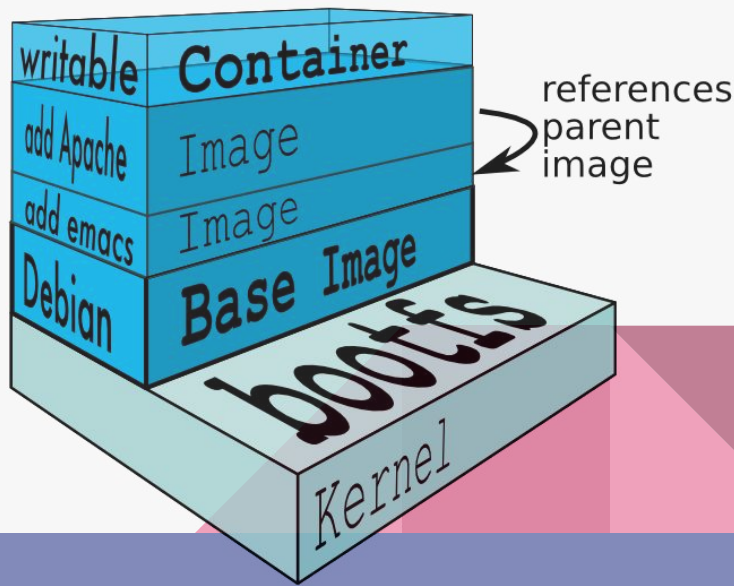


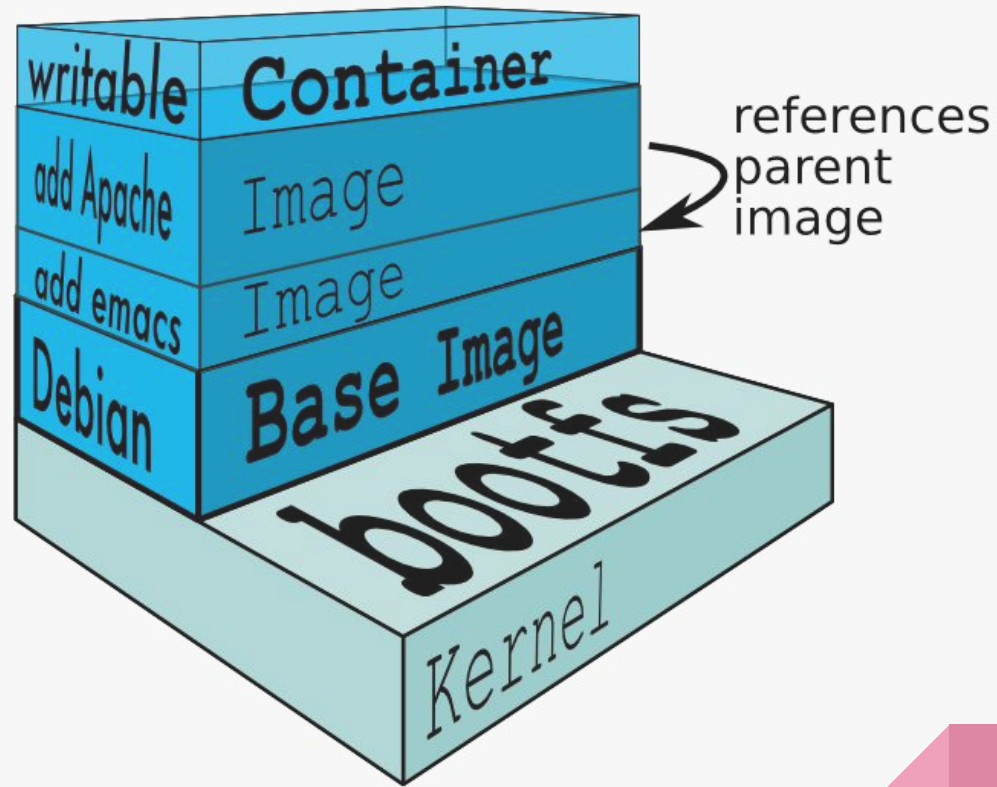
# Intro a Docker

Docker es una plataforma que permite armar, probar y poner en producción software de manera rápida.

Docker engloba todo el software en paquetes denominados containers los cuales contienen todas las librerías, funciones de sistema, binarios etc.. que el software necesite.

Los containers son una forma de virtualizar a nivel de sistema operativo en un ambiente aislado y limitado de recursos.





# INSTALACIÓN

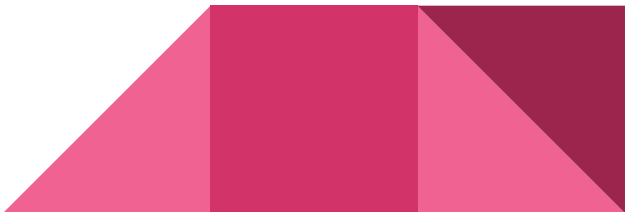
# Correr imagenes

Trae la imagen de docker hub

```
docker pull fedora:34
```

Correr la imagen de modo interactivo

```
docker run -it fedora:34
```





# Docker - Dockerfile

Imagen de base

```
# Imagen base
```

Ejecuta comandos  
en el container

```
FROM ubuntu:12.04
```

```
RUN apt-get update && apt-get install -y apache2 && apt-get  
clean && rm -rf /var/lib/apt/lists/*
```

Variables de entorno  
del container

```
ENV APACHE_RUN_USER www-data
```

Expone puertos del  
container a otros  
containers en la  
misma red

```
EXPOSE 80
```

Comando de inicio

```
CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

# Imagenes - Docker HUB

[Explore](#)[Repositories](#)[Organizations](#)[Get Help](#)[pbullian](#)[Docker EE](#)[Docker CE](#)[Containers](#)[Plugins](#)

## Filters

Docker Certified

☐ ☒ Docker Certified

## Images

☐ Verified Publisher

Docker Certified And Verified Publisher Content

☐ Official Images

Official Images Published By Docker

1 - 25 of 63,712 results for **ubuntu**. [Clear search](#)

Most Popular



**ubuntu**


Updated 17 minutes ago

Ubuntu is a Debian-based Linux operating system based on free software.

OFFICIAL IMAGE

**10M+** **9.9K**  
Downloads Stars

[Container](#)[Linux](#)[PowerPC 64 LE](#)[386](#)[ARM 64](#)[ARM](#)[x86-64](#)[IBM Z](#)[Base Images](#)[Operating Systems](#)

 docker-library-bot Update to 20190813 for amd64 (amd64)

9db8c72 17 days ago

1 contributor

44 lines (40 sloc) 2.58 KB

Raw

Blame

History



```
1 FROM scratch
2 ADD ubuntu-bionic-core-cloudimg-amd64-root.tar.gz /
3 # verify that the APT lists files do not exist
4 RUN [ -z "$(apt-get indextargets)" ]
5 # (see https://bugs.launchpad.net/cloud-images/+bug/1699913)
6
7 # a few minor docker-specific tweaks
8 # see https://github.com/docker/docker/blob/9a9fc01af8fb5d98b8eec0740716226fadb3735c/contrib/mkimage/debootstrap
9 RUN set -xe \
10 \
11 # https://github.com/docker/docker/blob/9a9fc01af8fb5d98b8eec0740716226fadb3735c/contrib/mkimage/debootstrap#L40-L48
12     && echo '#!/bin/sh' > /usr/sbin/policy-rc.d \
13     && echo 'exit 101' >> /usr/sbin/policy-rc.d \
14     && chmod +x /usr/sbin/policy-rc.d \
15     \
16 # https://github.com/docker/docker/blob/9a9fc01af8fb5d98b8eec0740716226fadb3735c/contrib/mkimage/debootstrap#L54-L56
17     && dpkg-divert --local --rename --add /sbin/initctl \
18     && cp -a /usr/sbin/policy-rc.d /sbin/initctl \
19     && sed -i 's/^exit.*/exit 0/' /sbin/initctl \
20     \
21 # https://github.com/docker/docker/blob/9a9fc01af8fb5d98b8eec0740716226fadb3735c/contrib/mkimage/debootstrap#L71-L78
22     && echo 'force-unsafe-io' > /etc/dpkg/dpkg.cfg.d/docker-apt-speedup \
23     \
24 # https://github.com/docker/docker/blob/9a9fc01af8fb5d98b8eec0740716226fadb3735c/contrib/mkimage/debootstrap#L85-L105
```

# Docker - Images & run

```
# docker images | grep mwe
mwe_ex1          latest          49a80191d5df    14 minutes ago    194MB
```

Local port      Container port

```
# docker run -p 4000:80 mwe_ex1:latest
```

Image name

tag



# Docker-compose

```
version: '3'
```

```
services:
```

```
  web:
```

```
    build: .
```

```
    ports:
```

```
      - "5000:80"
```

```
  redis:
```

```
    image: "redis:alpine"
```

```
# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	
STATUS	PORTS	NAMES		
978c8d7c1650	ex1_web	"python app.py"	4 seconds ago	Up 2
seconds	0.0.0.0:5000->80/tcp	ex1_web_1		
1580f61d9f71	redis:alpine	"docker-entrypoint..."	About an hour ago	Up
3 seconds	6379/tcp	ex1_redis_1		

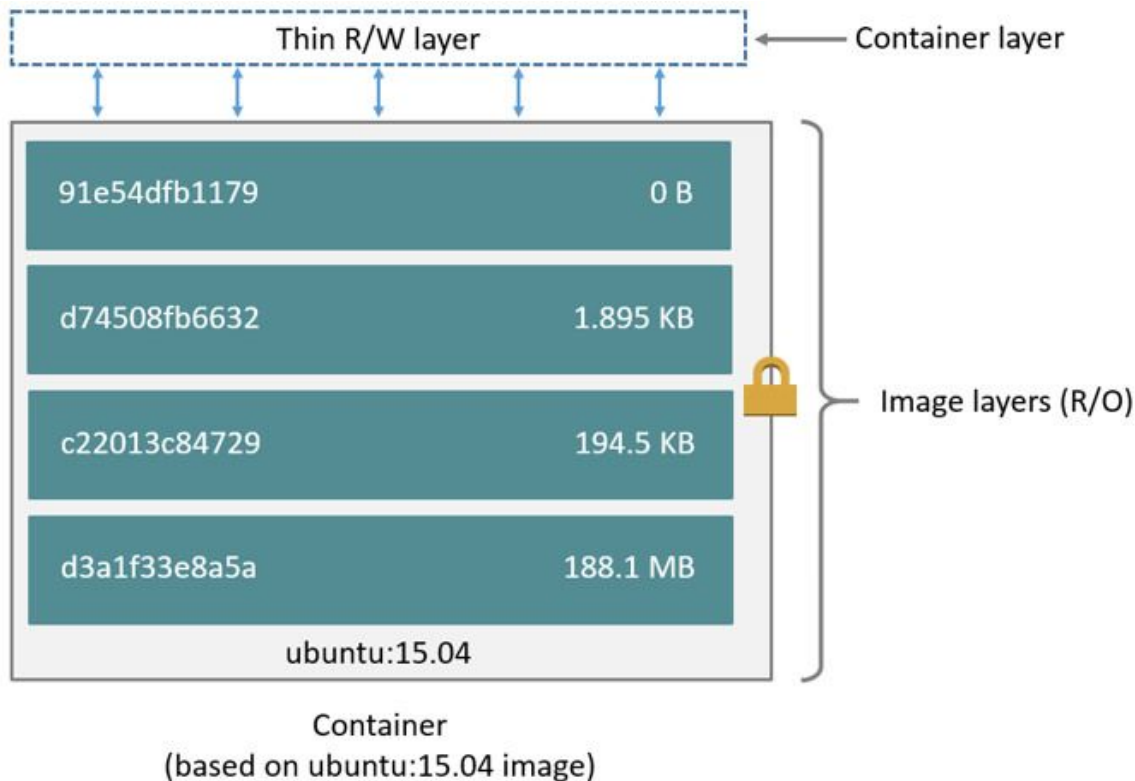
# Docker Volumes

- Docker volumes
- Bind mounts

```
$ docker volume create my-vol
```

```
$ docker run -d \
  -it \
  --name devtest \
  --mount source=my-vol,target=/app \
  nginx:latest
```

```
$ docker run -d \
  -it \
  --name devtest \
  --mount
type=bind,source="/var/www",target=/app \
  nginx:latest
```



# Docker Registry

```
$ docker run -d -p 5000:5000 --name registry registry:2
```

```
version: '3'
services:
  web:
    image: 127.0.0.1:5000/ex2
    build: .
    ports:
      - "5001:80"

  redis:
    image: "redis:alpine"
```

```
# docker push 127.0.0.1:5000/mwe:1
The push refers to a repository [127.0.0.1:5000/mwe]
4317d6310ed9: Pushed
57417954f538: Pushed
62a76f5bf2d0: Pushed
90c1efcf5e59: Pushed
271f16f8efa6: Pushed
15d2fe96bb43: Pushed
0d960f1d4fba: Pushed
1: digest:
sha256:2e5e32607cbb7ad2df96dd8063b124820446dc26c4bef41c292a8aa34d4
2038d size: 1787
```

# Docker Orchestration

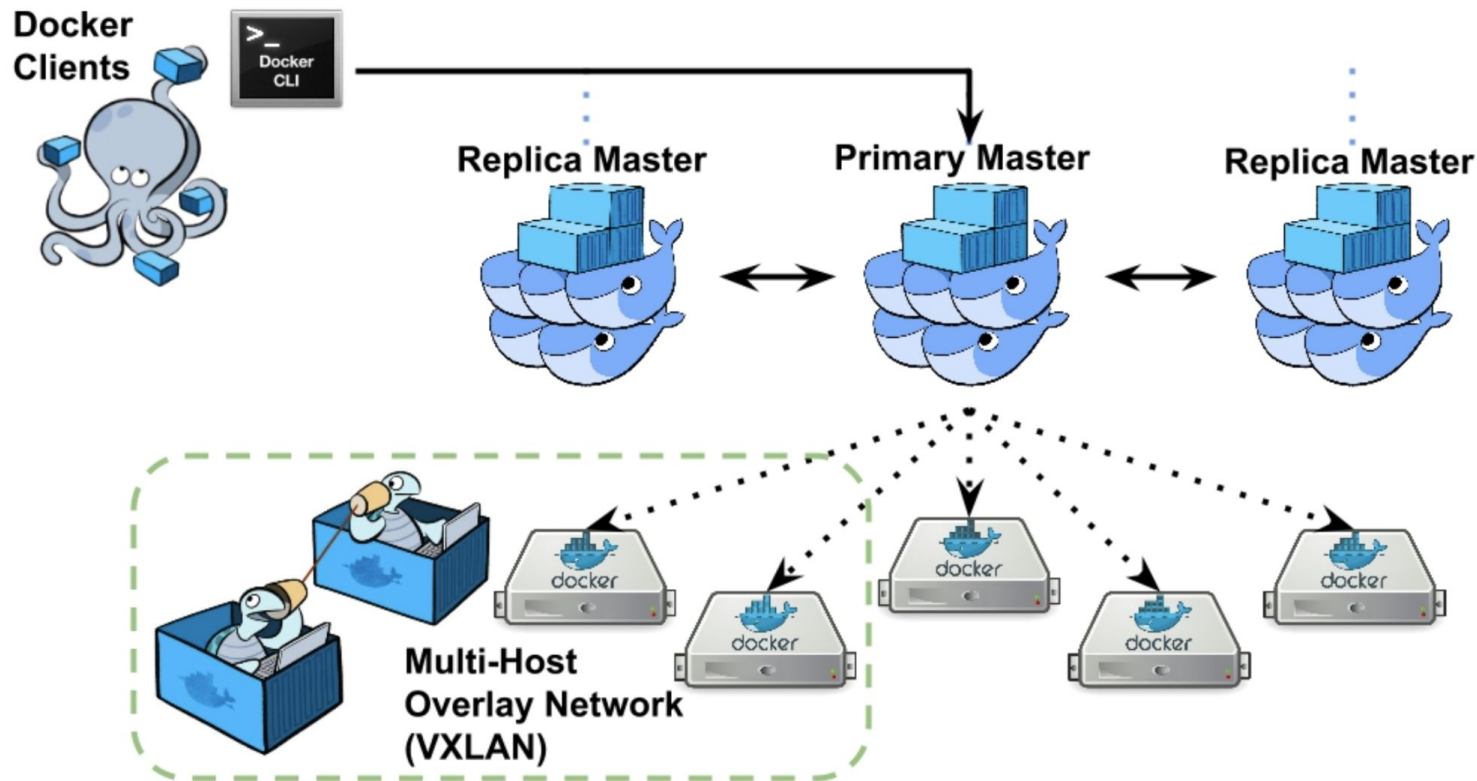
Manejar muchos contenedores de manera distribuida es un proceso complejo y por eso se necesita de un orquestador.

Los más usados son kubernetes y docker swarm





# Docker Swarm



# Docker Swarm

```
# docker stack deploy --compose-file docker-compose.yml ex2
```

```
Ignoring unsupported options: build  
Creating network ex2_default  
Creating service ex2_redis  
Creating service ex2_web
```

```
# docker stack services ex2
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
i6ef27pt03th	ex2_web	replicated	1/1	127.0.0.1:5000/ex2:latest	
*:5001->80/tcp					
xaq1dk0xjk7x	ex2_redis	replicated	1/1	redis:alpine	

# Seguridad de los contenedores

- Las imágenes que usamos, ¿de dónde vienen?
- ¿Tenemos nuestro software actualizado?
  - ¿Por qué es tan fácil olvidarse de actualizar?
- ¿Quién tiene acceso a los contenedores y definiciones?
- ¿Cómo guardamos los secretos?



# Trabajos Prácticos

- Creación de un servidor WEB Apache el cual deberá servir una página estática personalizada tomando como base la imagen oficial de Ubuntu, Debian o Fedora. (Solo para grupos de hasta 3 personas)
- Creación de un servidor WEB NGINX el cual deberá servir una página estática personalizada tomando como base la imagen oficial de Ubuntu, Debian o Fedora. (Solo para grupos de hasta 3 personas)
- Creación de una base de datos MySQL la cual debe servir una base de datos personalizada con el usuario "admlinux". (Solo para grupos de hasta 3 personas)
- Creación de un ambiente Cloud utilizando **owncloud**.
- Creación de un ambiente linux Ubuntu, Debian o Fedora que cuente con usuario personalizado, alias de bash, prompt personalizado, y al menos dos scripts personalizados instalados en el sistema (que estén incluidos en PATH). (Solo para grupos de hasta 3 personas)
- Crear una página web con **wordpress** (base de datos y sitio en docker)
- Algún otro proyecto que les interese armar en docker (previo consultarme por mail)