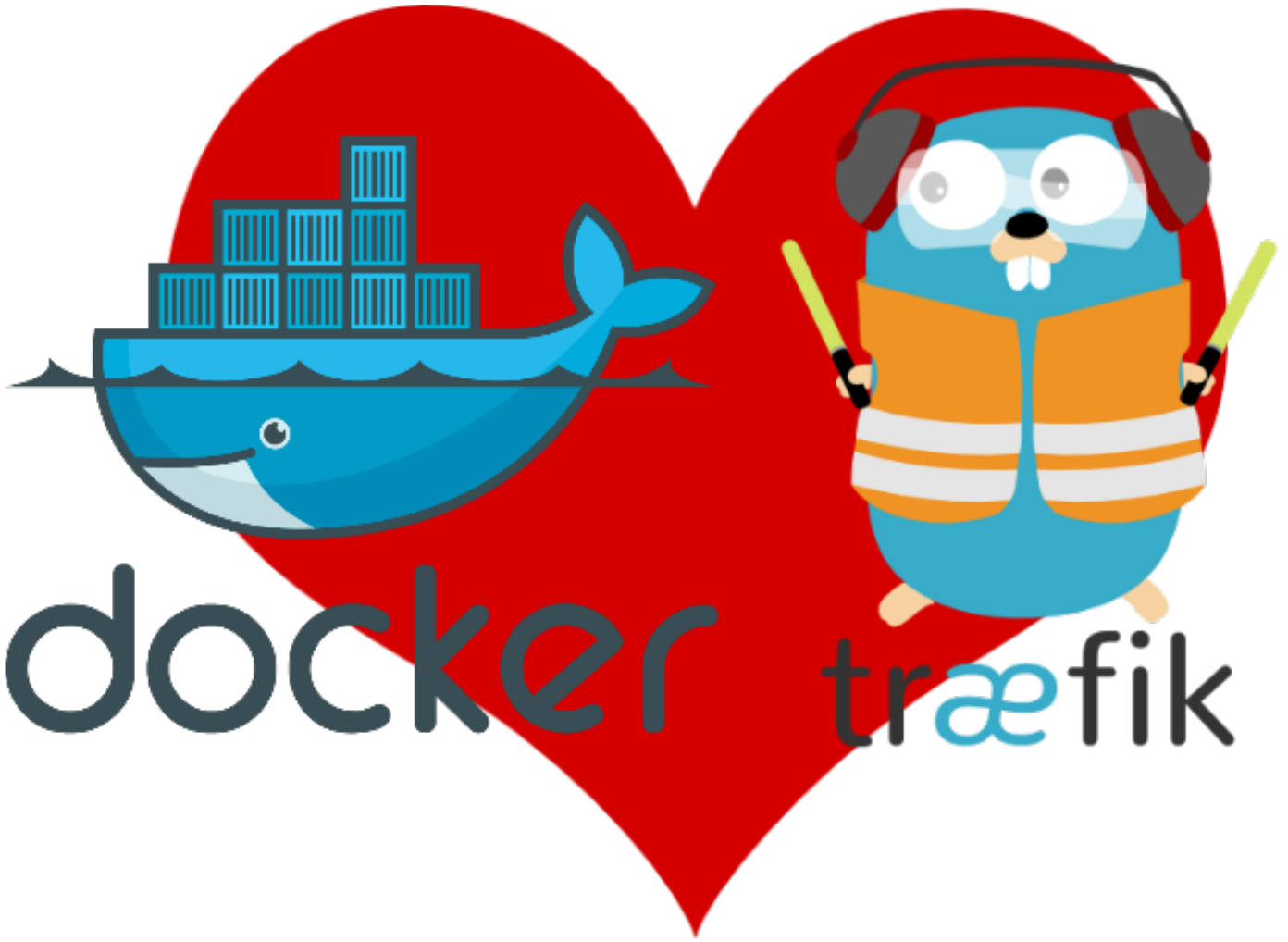
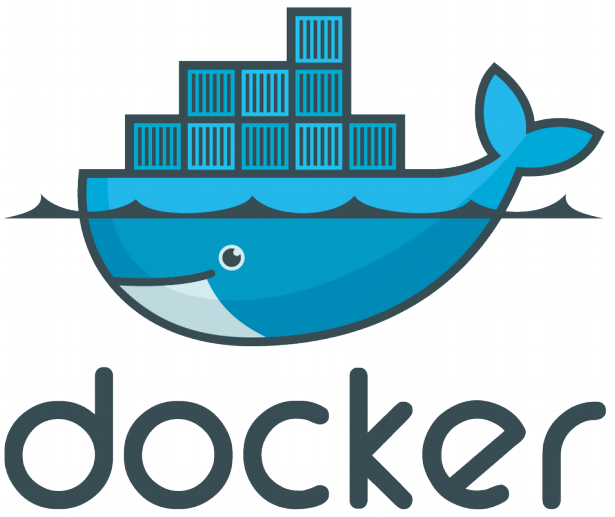
Docker <3 Træfik

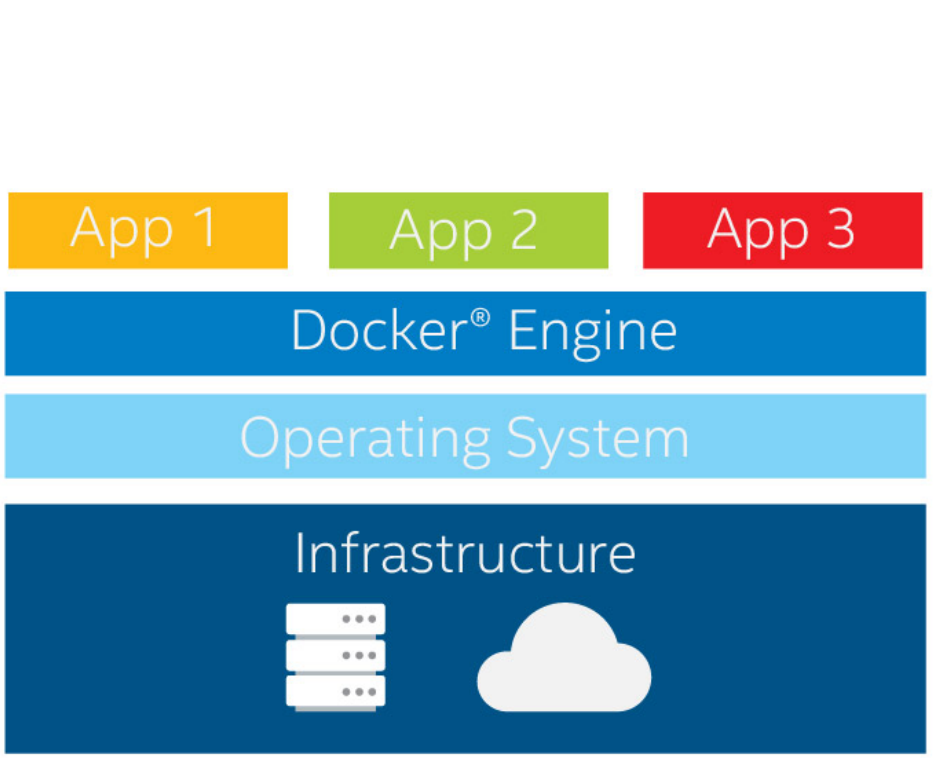


Docker

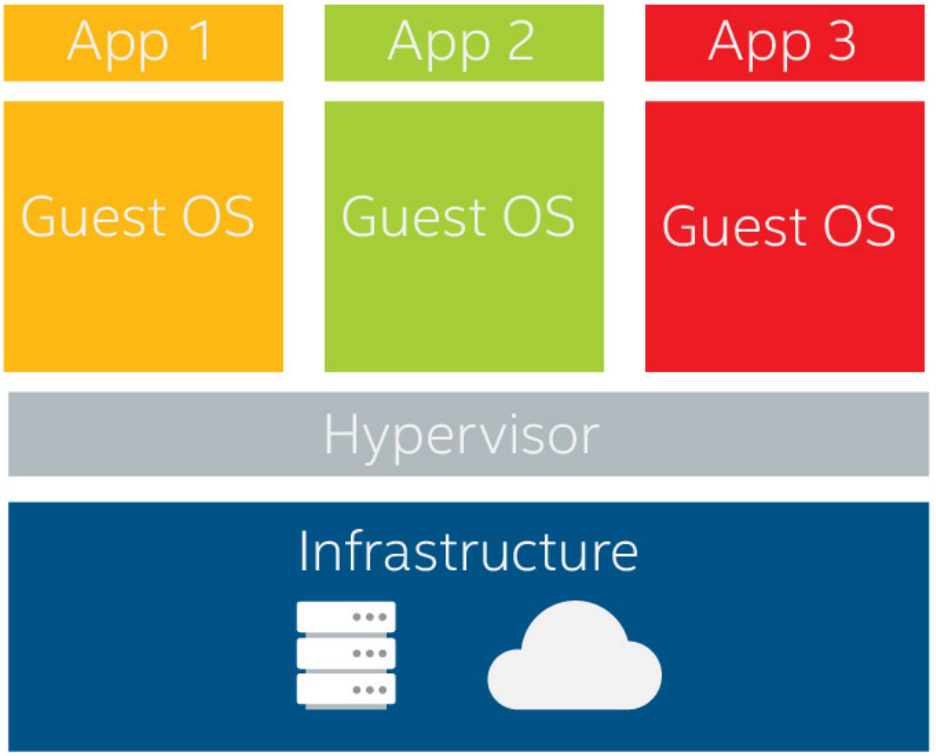
- Open Source
- Multiplataforma
- MicroServicios
- Stateless
- Escalable/Clustering
- Escrito en Go
- Versiones
  - Docker CE (Community Edition)
  - Docker EE (Enterprise Edition)



Docker vs VM



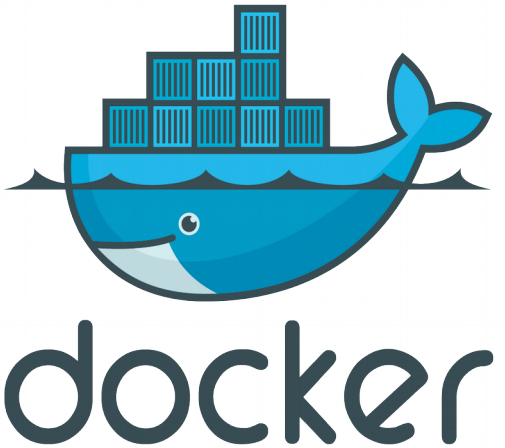
Docker Container Approach



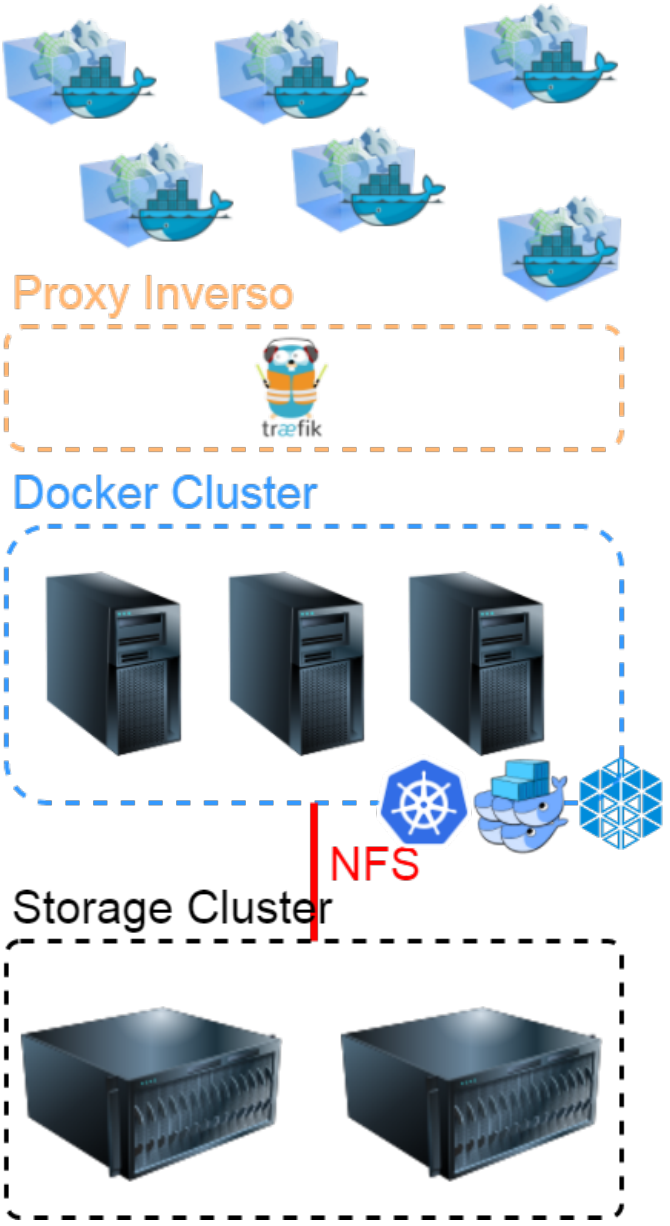
Virtualization Approach

Características Docker

- Namespaces
  - Capa de aislamiento para separar procesos de ejecución de los contenedores con el Host
- Cgroups
  - Limitación del consumo de recursos de los CT
- UnionFS
  - Montar sistemas de archivos

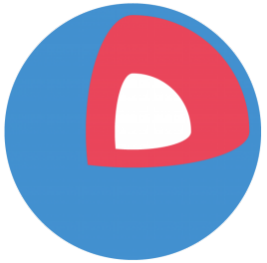


Infraestructura



Sistemas Operativos “Especiales”

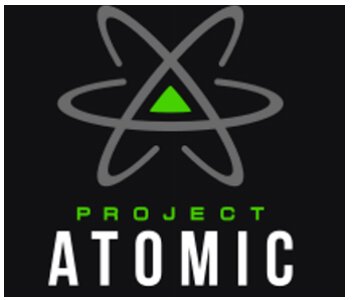
- CoreOS



- Rancher

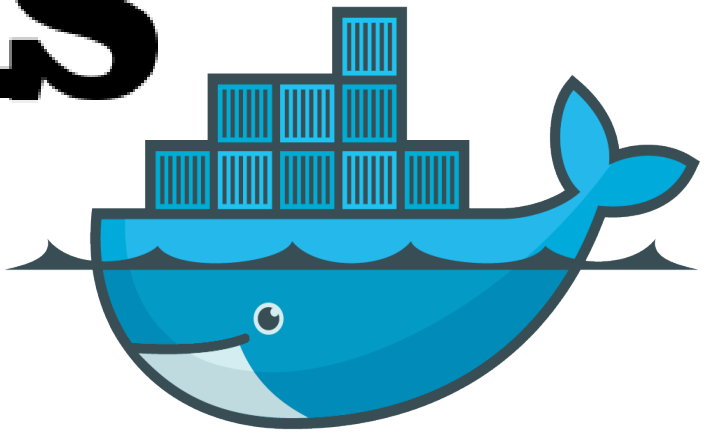
RANCHER

- Atomic



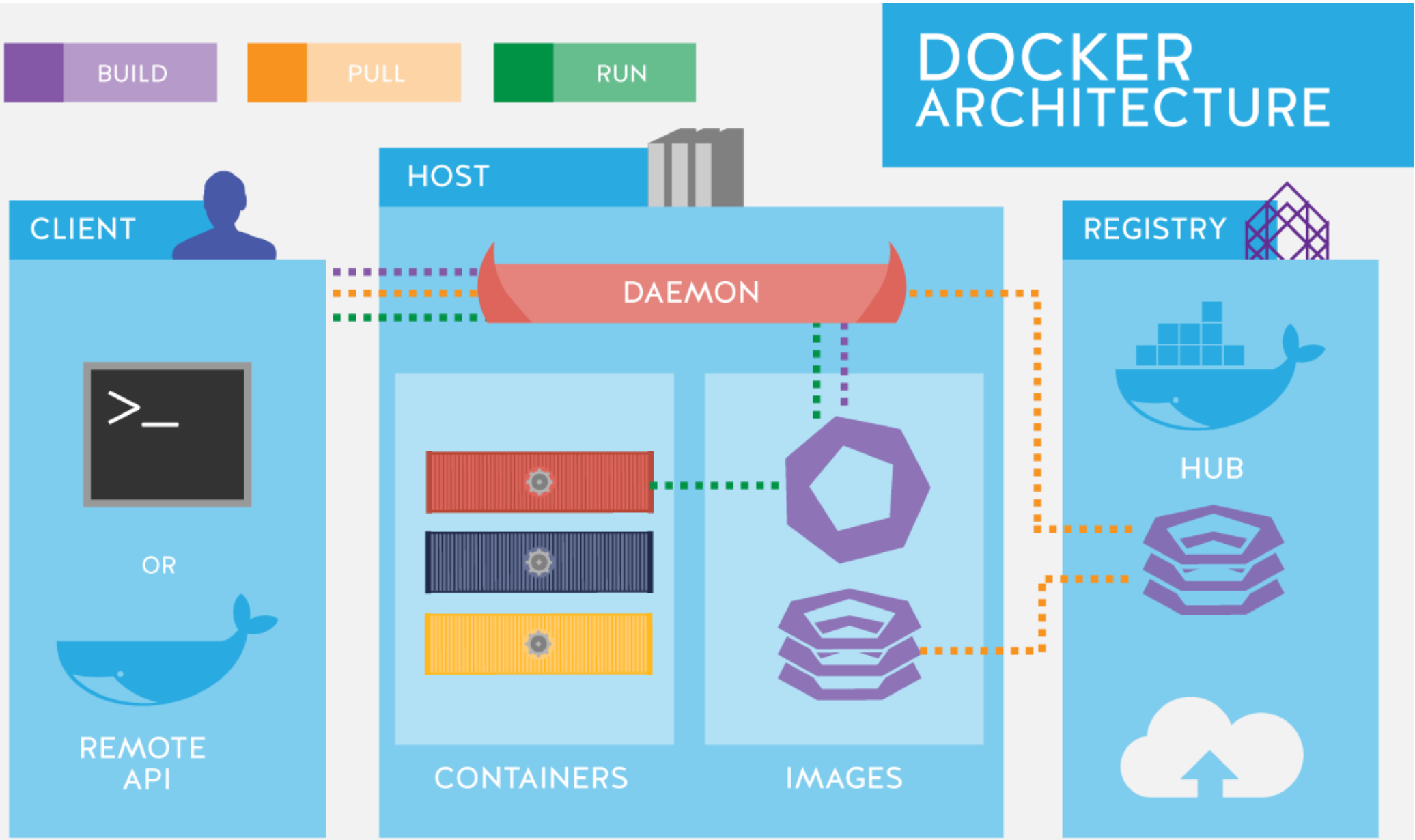
En mi ordenador funciona

How It Works



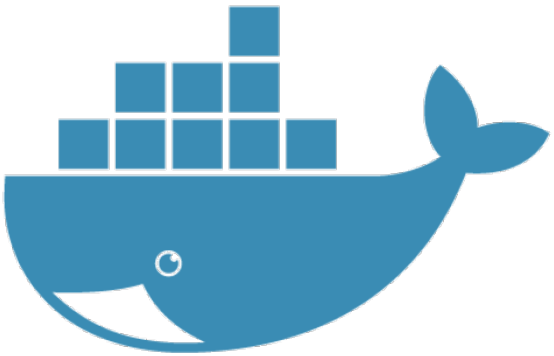
docker

Flow


BUILD
PULL
RUN
DOCKER ARCHITECTURE
HOST
CLIENT
REGISTRY
DAEMON
>_
OR
HUB
REMOTE API
CONTAINERS
IMAGES

Docker Hub

- Registro de imágenes de docker
  - https://hub.docker.com/
- Repositorios oficiales de **imágenes**



Docker Registry

- Privado
- OpenSource
- Versionado

Instalación Docker

- Ubuntu

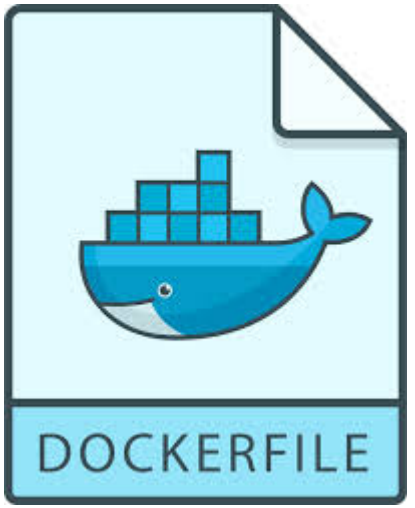
```
# apt -y install apt-transport-https ca-certificates curl gnupg2 software-properties-common

# curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $
(lsb_release -cs) stable"

# apt update && apt -y install docker-ce
```

- CentOS

```
# yum install -y yum-utils device-mapper-persistent-data lvm2

# yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

# yum-config-manager --enable docker-ce-edge

# yum -y install docker-ce
```

Imágenes

- Imagen ≈ Plantilla
- Dockerfile
- Versiones



PoC: Crear una imagen

```
FROM debian

MAINTAINER ichasco version 1.0

RUN apt-get update && apt-get install -y php5 libapache2-mod-php5 php5-mysql php5-cli && apt-get clean && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid

EXPOSE 80

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

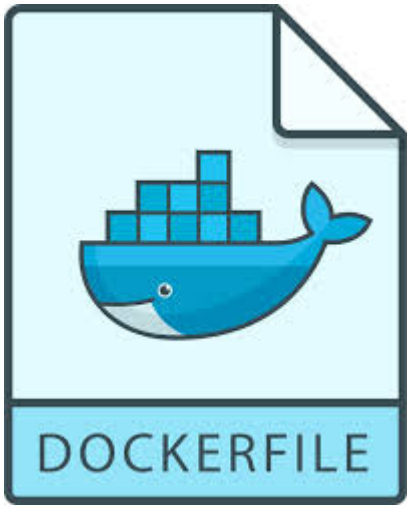
docker build -t <owner>/<aplic> /path

```
Sending build context to Docker daemon 18.81 MB
Step 1/4 : FROM debian
latest: Pulling from library/debian
ef0380f84d05: Pull complete
Digest: sha256:e283dc7bdfe4df3672ba561cf50022528c493cc5800e80670ca47315aad6a5de
Status: Downloaded newer image for debian:latest
 ---> a25c1eed1c6f
Step 2/4 : MAINTAINER ichasco version 1.0
 ---> Running in 6d69ee74c56d
 ---> 479ae9efb311
Removing intermediate container 6d69ee74c56d
Step 3/4 : RUN apt-get update && apt-get install -y php5 libapache2-mod-php5 php5-mysql php5-cli && apt-get clean && rm -rf /var/lib/apt/lists/*
 ---> Running in d7778b5f867f
```

```
# docker images
```

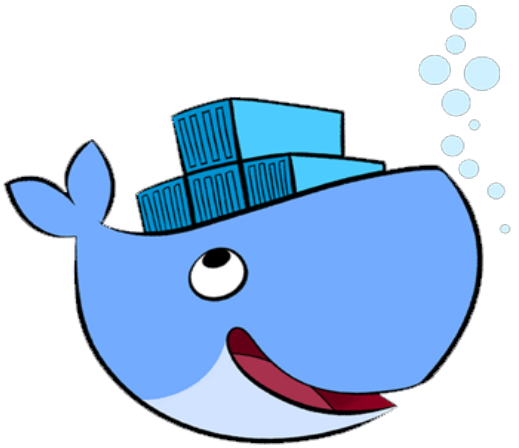
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ismael/apache	latest	b70f15f7c06b	19 seconds ago	214 MB

```
# docker commit docker_ct  docker_img
```



Desplegar Docker

- Modos de desplegar docker
  - Docker run
  - Docker-compose



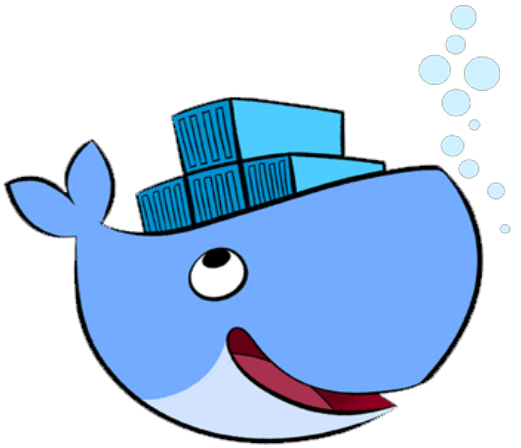
Docker Run

```
# docker run
```

-d: modo background

-p: mapeo de puertos

-v: mapeo de volúmenes

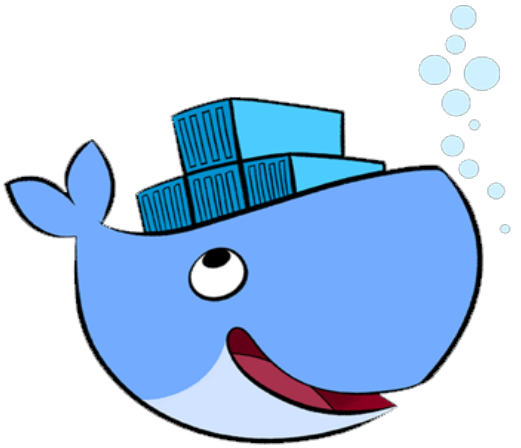


PoC: Desplegar docker con docker run

```
# docker run -d --name apache -p 80:80 -v apache:/var/www/html ichasco/apache
```

```
# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e5dabf8a8506	ismael/apache	"/usr/sbin/apache2..."	3 seconds ago	Up 2 seconds	0.0.0.0:80->80/tcp	apache

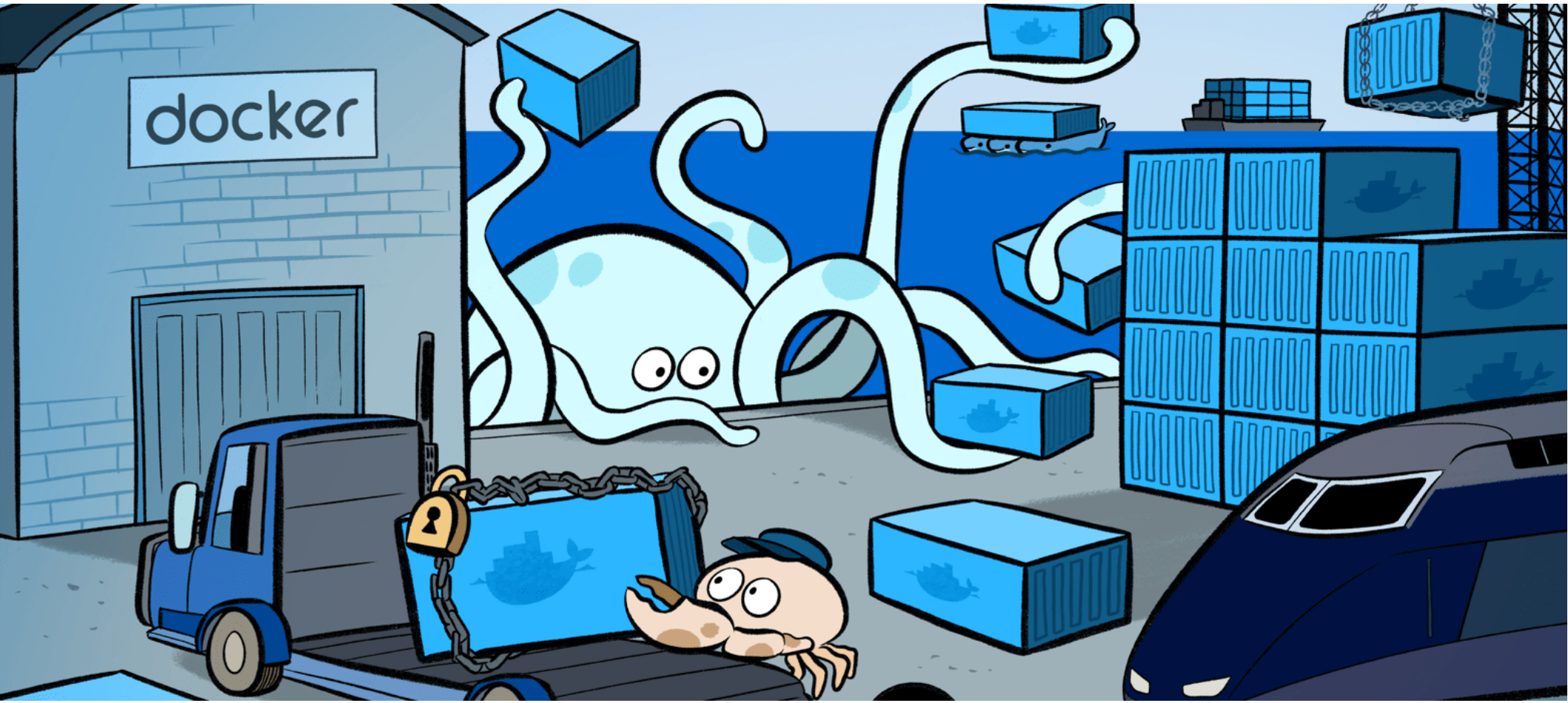


Nota: directorio de volúmenes de docker:
/var/lib/docker/

Comandos interesantes

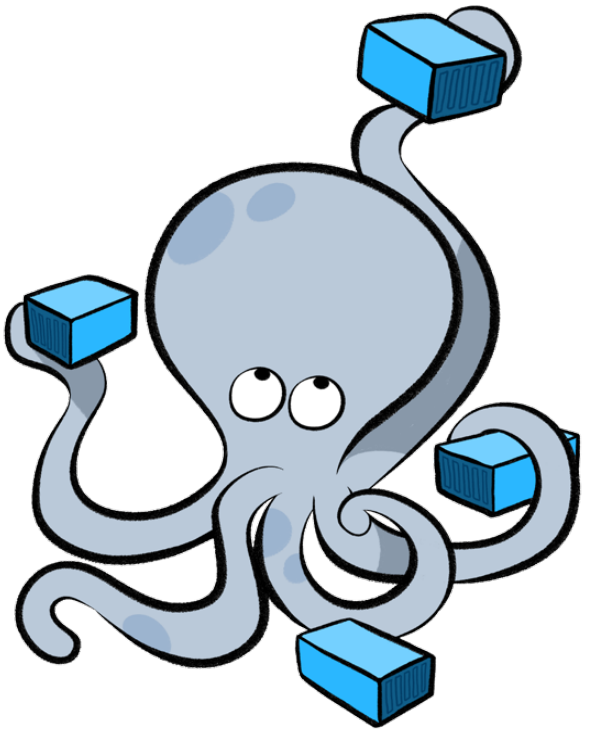
- **Apagar**: # docker stop docker_name
- **Iniciar**: # docker start docker_name
- **Reiniciar**: # docker restart docker_name
- **Eliminar**: # docker rm docker_name
- **Logs**: # docker logs docker_name
- **Shell**: # docker exec -it docker_name /bin/bash

Docker-Compose



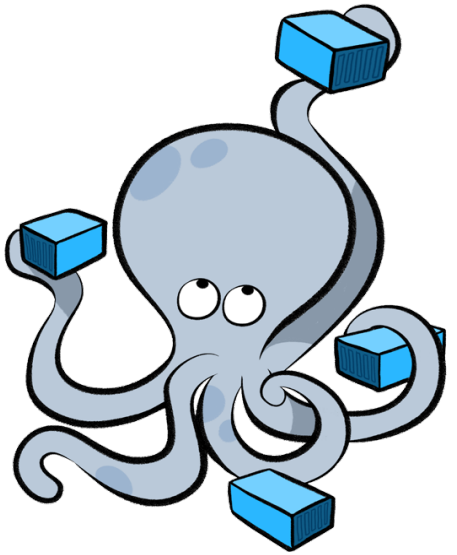
¿Que es?

- Despliegue de Stacks
- Relaciones mas fáciles
- Escalado básico
- Control sobre el stack
- Escrito en YAML



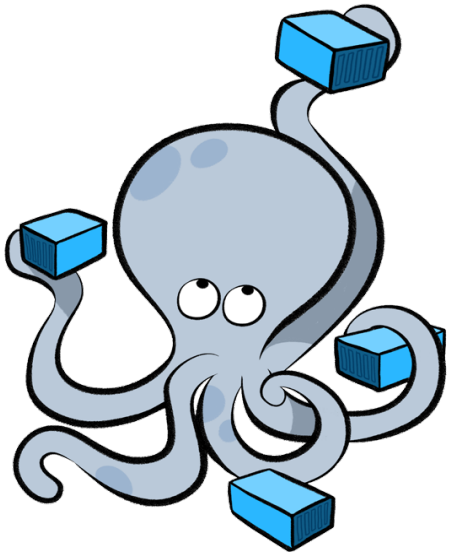
Versiones

- Versión 3.0/3.1
  - Docker 1.13.0+
- Versión 2.1
  - Docker 1.12.0
- Versión 2.0
  - Docker 1.10
- Versión 1.0
  - Docker 1.9.1
- Información de cambios



Recomendaciones

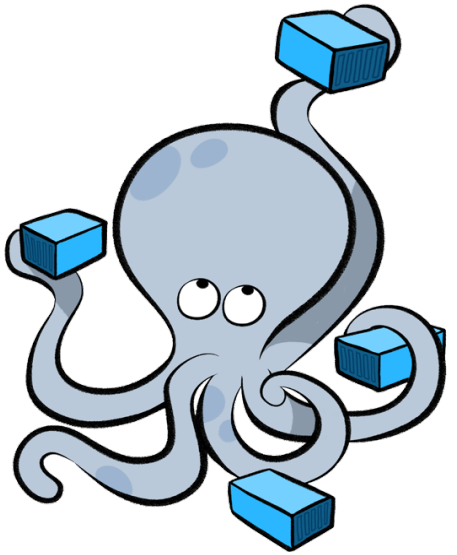
- Utilizar carpetas para los proyectos
- No poner nombres a los contenedores
- Si son redes generales, no crearlas con docker-compose



Instalación

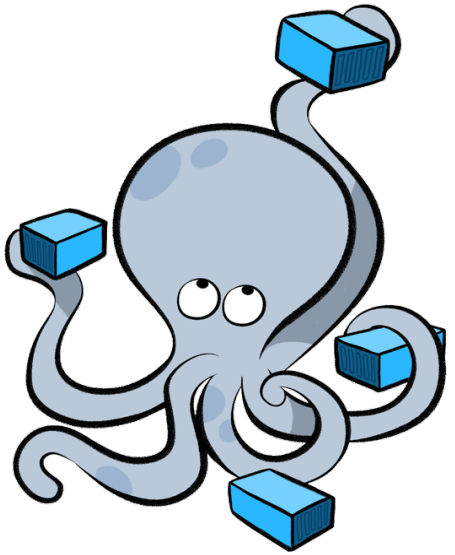
```
# curl -L https://github.com/docker/compose/releases/download/1.13.0/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```

```
# chmod +x /usr/local/bin/docker-compose
```



Estructura docker-compose

```
version:

services:
  servicio1:
    image:
    restart:
    volumes:
      - mapeo
    environment:
      - variables
    ports:
      - mapeo
    depends_on:
      - docker_depend
    networks:
      - network

volumes:
  volume1:
    driver:

networks:
  network1:
    driver:
```

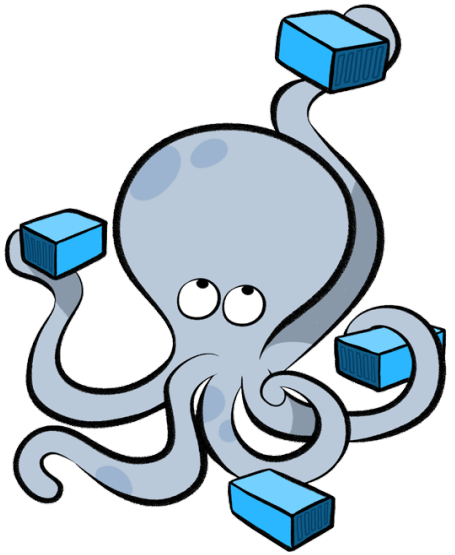


Comandos interesantes

- Desplegar stack
  - # docker-compose -f <file> up -d
- Parar stack
  - # docker-compose -f <file> stop
- Borrar stack
  - # docker-compose -f <file> rm -f
- Escalar stack
  - # docker-compose -f <file> scale docker_name=<nº instancias>

PoC: Despliegue stack

```yaml
version: '3'

services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     networks:
       - wp_back
     environment:
       MYSQL_ROOT_PASSWORD: wordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     restart: always
     networks:
      - wp_back
     ports:
      - 80:80
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
     volumes:
       - wp_data:/var/www/html

volumes:
    db_data:
    wp_data:

networks:
  wp_back:
    driver: bridge
```

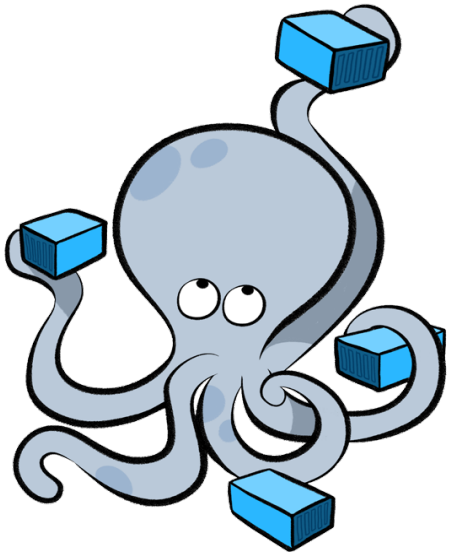


link

docker-compose -f <file> up -d

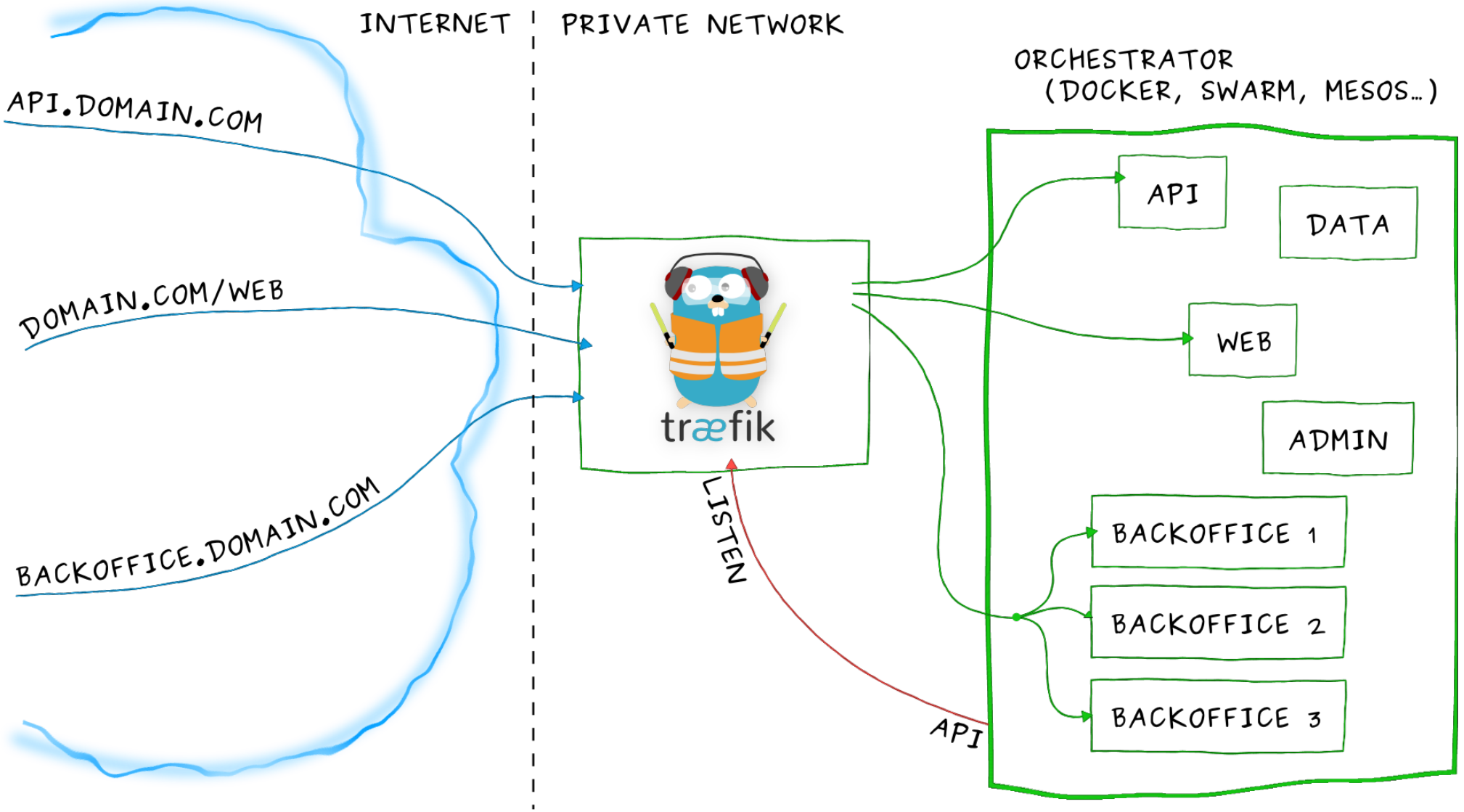
docker-compose -f <file> ps

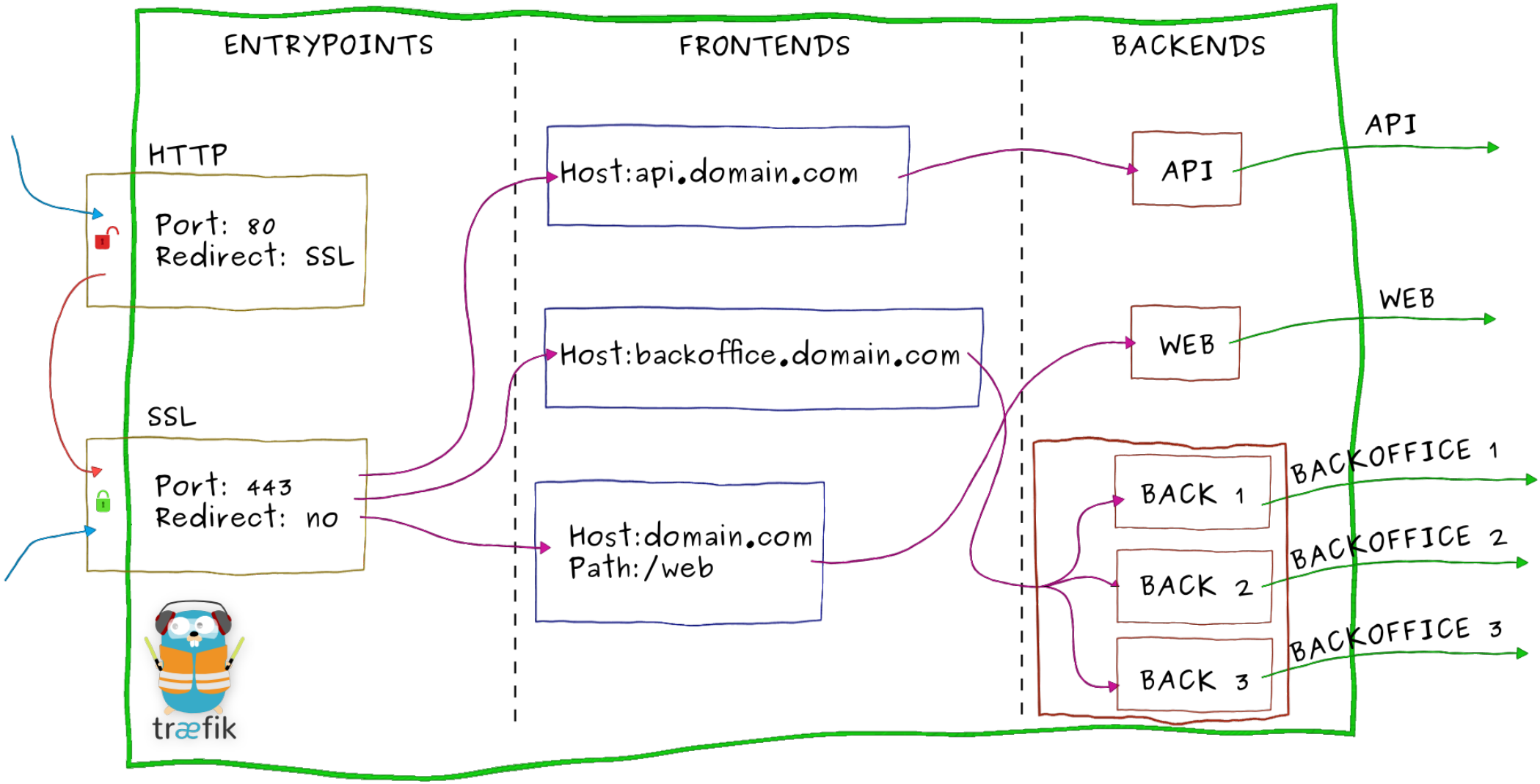
¿? # docker-compose -f <file> scale <name>=2



Træfik

- Proxy reverso
- Balanceo Round-Robin
- HTTP/2
- Let’s Encrypt
- SNI


INTERNET
PRIVATE NETWORK
ORCHESTRATOR
(DOCKER, SWARM, MESOS…)
API.DOMAIN.COM
DOMAIN.COM/WEB
BACKOFFICE.DOMAIN.COM
træfik
LISTEN
API
API
DATA
WEB
ADMIN
BACKOFFICE 1
BACKOFFICE 2
BACKOFFICE 3


ENTRYPOINTS
FRONTENDS
BACKENDS
HTTP
Port: 80
Redirect: SSL
SSL
Port: 443
Redirect: no
Host:api.domain.com
Host:backoffice.domain.com
Host:domain.com
Path:/web
API
WEB
BACK 1
BACK 2
BACK 3
API
WEB
BACKOFFICE 1
BACKOFFICE 2
BACKOFFICE 3
træfik

Instalación

```
# mkdir traefik && cd traefik

# wget
https://raw.githubusercontent.com/ichasco/traefik/master/traefik.toml

# docker network create traefik

# docker run -d -p 8080:8080 -p 80:80 --network traefik --name traefik -v
$PWD/traefik.toml:/etc/traefik/traefik.toml -v
/var/run/docker.sock:/var/run/docker.sock traefik
```

Link compose

Desplegar los stacks bajo træfik

```yaml
version: '3'

services:
   db:
     image: mysql:5.7
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     labels:
       - "traefik.enable=false"
     networks:
       - wp_back
     environment:
       MYSQL_ROOT_PASSWORD: wordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:latest
     labels:
      - "traefik.backend=wp"
      - "traefik.frontend.rule=Host:wp.ichasco.com"
      - "traefik.docker.network=traefik"
     restart: always
     networks:
      - traefik
      - wp_back
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
     volumes:
       - wp_data:/var/www/html

volumes:
    db_data:
    wp_data:

networks:
  wp_back:
    driver: bridge
  traefik:
    external:
      name: traefik
```

Link

Voilà

http://dockerserver:8080

frontend-Host-wp-ichasco-com

Route	*Rule*
route-frontend-Host-wp-ichasco-com	`Host:wp.ichasco.com`

http Backend:backend-wp PassHostHeader

backend-wp

Server	*URL*	*Weight*
server-wordpress_wordpress_1	`http://172.19.0.3:80`	0

Load Balancer: wrr

Y ahora con HTTPS

traefik.toml

```
defaultEntryPoints = ["http", "https"]

[acme]
  email = "info@ichasco.com"
  storageFile = "/etc/traefik/acme/acme.json"
  entryPoint = "https"
  acmeLogging = true
  onDemand = true
  OnHostRule = true

 [entryPoints]
  [entryPoints.http]
    address = ":80"
      [entryPoints.http.redirect]
    entryPoint = "https"
  [entryPoints.https]
    address = ":443"
     [entryPoints.https.tls]
```

docker-compose.yml

```
ports:
  - "443:443"
volumes:
  - $PWD/acme:/etc/traefik/acme
```

frontend-Host-wp-ichasco-com

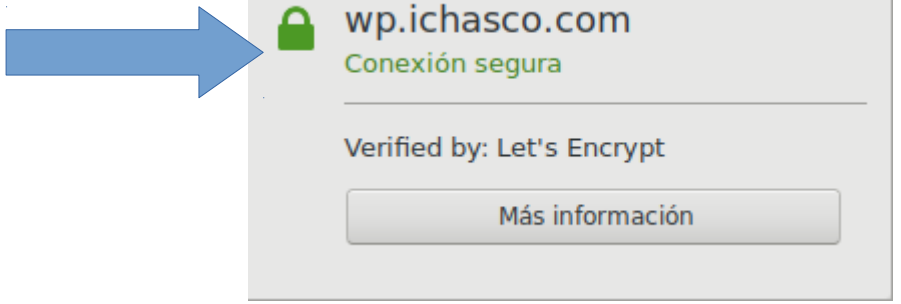
Route	*Rule*
route-frontend-Host-wp-ichasco-com	`Host:wp.ichasco.com`

http https Backend:backend-wp PassHostHeader

backend-wp

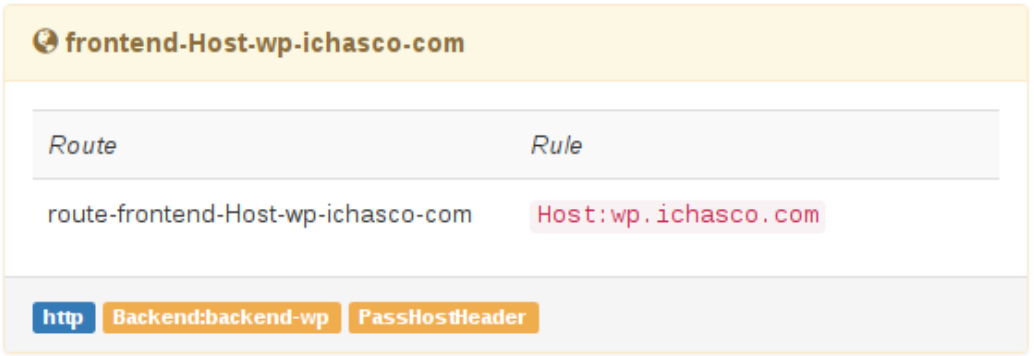
Server	*URL*	*Weight*
server-wordpress_wordpress_1	`http://172.18.0.3:80`	0

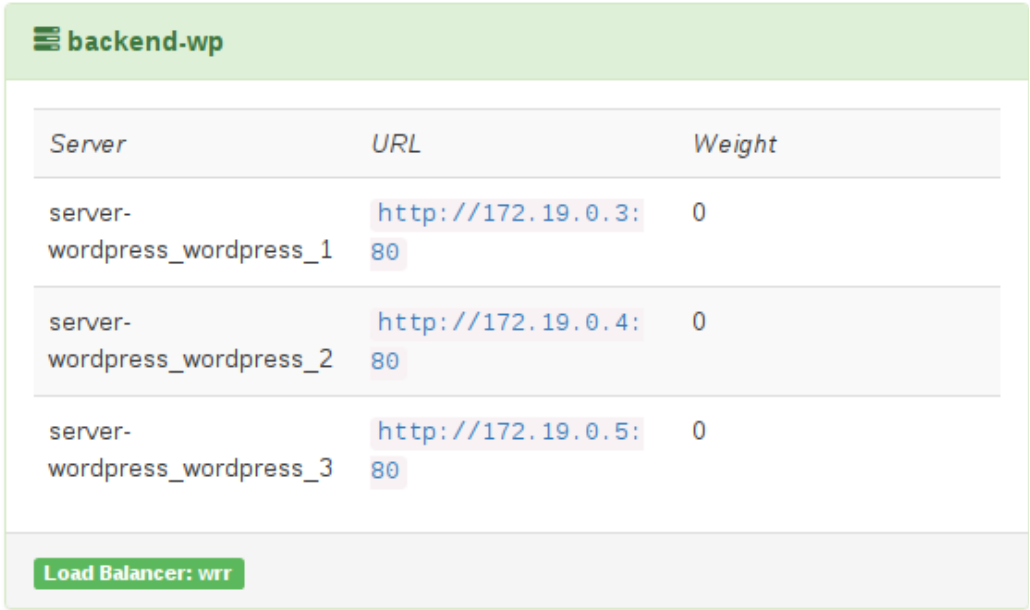
Load Balancer: wrr



Escalado

```
# docker-compose scale wordpress=3
```





+ Servicios

frontend-Host-wp-ichasco-com

Route	Rule
route-frontend-Host-wp-ichasco-com	`Host:wp.ichasco.com`

http Backend:backend-wp PassHostHeader

frontend-Host-wp2-ichasco-com

Route	Rule
route-frontend-Host-wp2-ichasco-com	`Host:wp2.ichasco.com`

http Backend:backend-wp2 PassHostHeader

backend-wp

Server	URL	Weight
server-wordpress_wordpress_1	`http://172.19.0.3:80`	0

Load Balancer: wrr

backend-wp2

Server	URL	Weight
server-wordpress2_wordpress2_1	`http://172.19.0.4:80`	0

Load Balancer: wrr

Bonus: Portainer

- Interfaz UI
- Gestión de docker
  - Network
  - Volúmenes
  - CLI & Logs
  - Images

Instalación

Docker-compose

```yaml
version: '3'

services:
  web:
    image: portainer/portainer
    restart: always
    volumes:
      - $PWD/data:/data
      - /var/run/docker.sock:/var/run/docker.sock
    labels:
      - 'traefik.backend=portainer'
      - 'traefik.frontend.rule=Host:portainer.ichasco.com'
      - 'traefik.docker.network=traefik'
    networks:
      - traefik

networks:
  traefik:
    external:
      name: traefik
```

Link

Y mientras tanto en Træfik...

frontend-Host-portainer-ichasco-com

Route	*Rule*
route-frontend-Host-portainer-ichasco-com	`Host:portainer.ichasco.com`

http | Backend:backend-portainer | PassHostHeader

backend-portainer

Server	*URL*	*Weight*
server-portainer_web_1	`http://172.19.0.4:9000`	0

Load Balancer: wrr

Containers

Items per page: 10

▶ Start | ■ Stop | Kill | Restart | ❚❚ Pause | ▶ Resume | Remove | + Add container

Show all containers

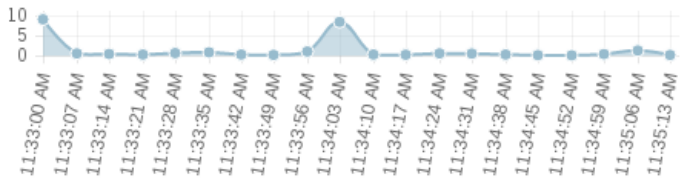
Filter...

State	Name	Image	IP Address	Published Ports	Ownership
stopped	wordpress2_wordpress2_1	wordpress:latest	-	-	public
stopped	wordpress2_db2_1	mysql:5.7	-	-	public
running	portainer_web_1	portainer/portainer	172.19.0.4	-	public
running	traefik_proxy_1	traefik	172.19.0.2	8080:8080 80:80	public
running	wordpress_wordpress_1	wordpress:latest	172.19.0.3	-	public
running	wordpress_db_1	mysql:5.7	172.18.0.2	-	public

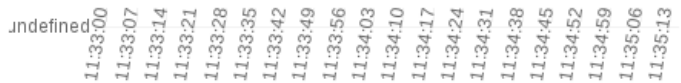
zabbix_zabbix-server_1

Name

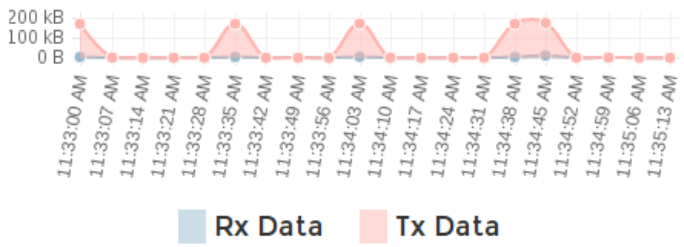
CPU usage



Memory usage



Network usage

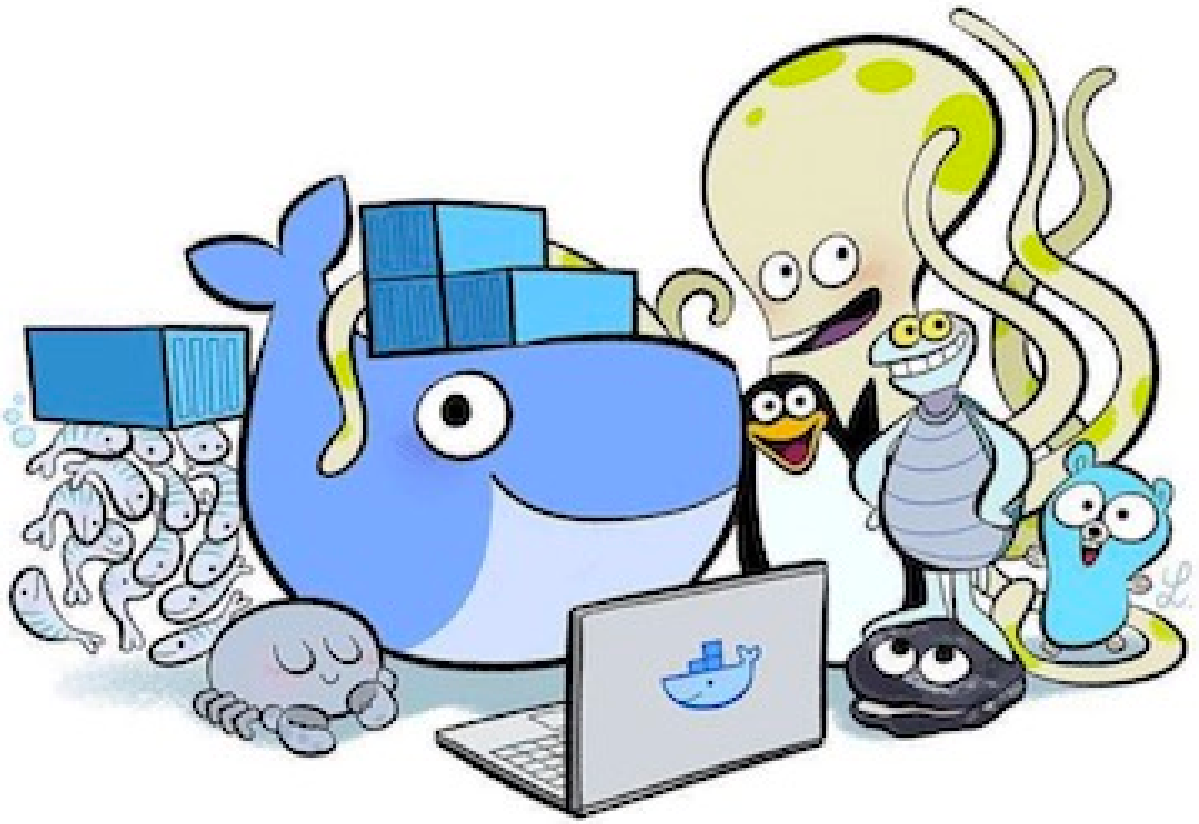


Processes

Items per page: 10

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1447	1403	0	11:06	?	00:00:00	/bin/bash /config/b
root	2349	1447	0	11:08	?	00:00:00	/usr/bin/python /usr/bin/supervisorc /etc/supervisord.cor
root	2368	2349	0	11:08	?	00:00:00	/usr/bin/bash -c wh sleep 3600; /usr/bin /config/init/12-xxl-p

To be continued...



Bibliografía

https://blog.ichasco.com

https://docs.docker.com

http://www.chris-kelly.net