

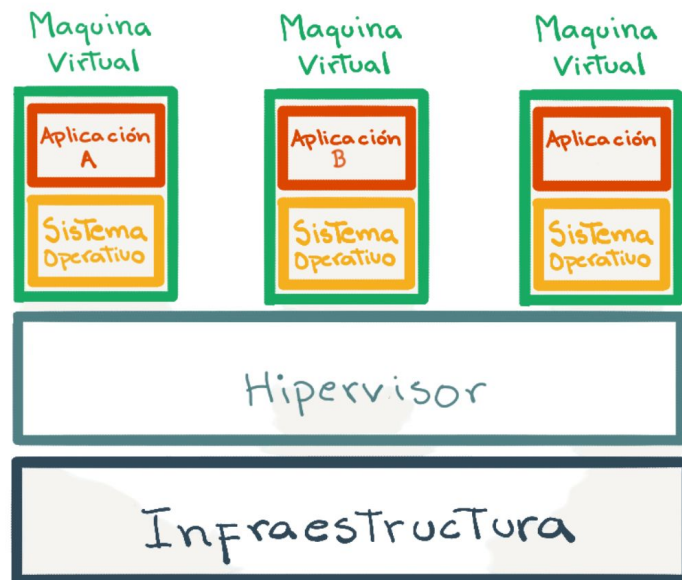
Contenedores

¿Qué son los contenedores?

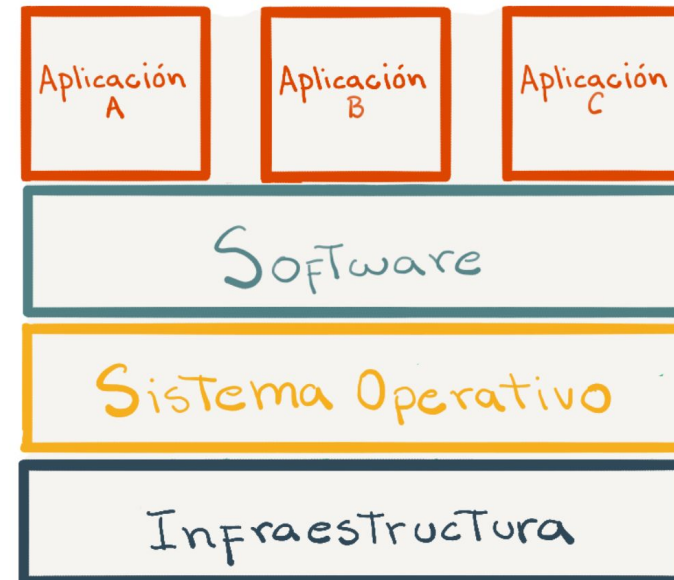
Son una tecnología que permite **empaquetar** y **aislar** las aplicaciones junto con todo el entorno que necesitan durante su tiempo de ejecución.



Virtualización completa



Virtualización a nivel de sistema operativo



Ventajas

Desventajas

Optimización de
recursos

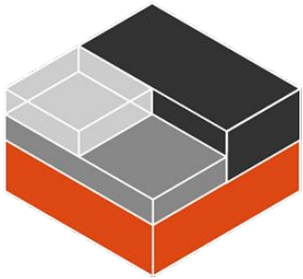
Estandarización

Portabilidad

Compatibilidad



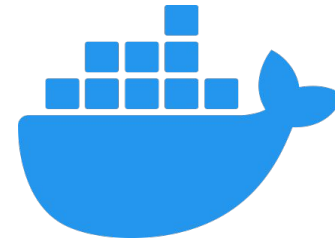
Herramientas que administran contenedores



Linux Containers



Rocket



Docker

Docker

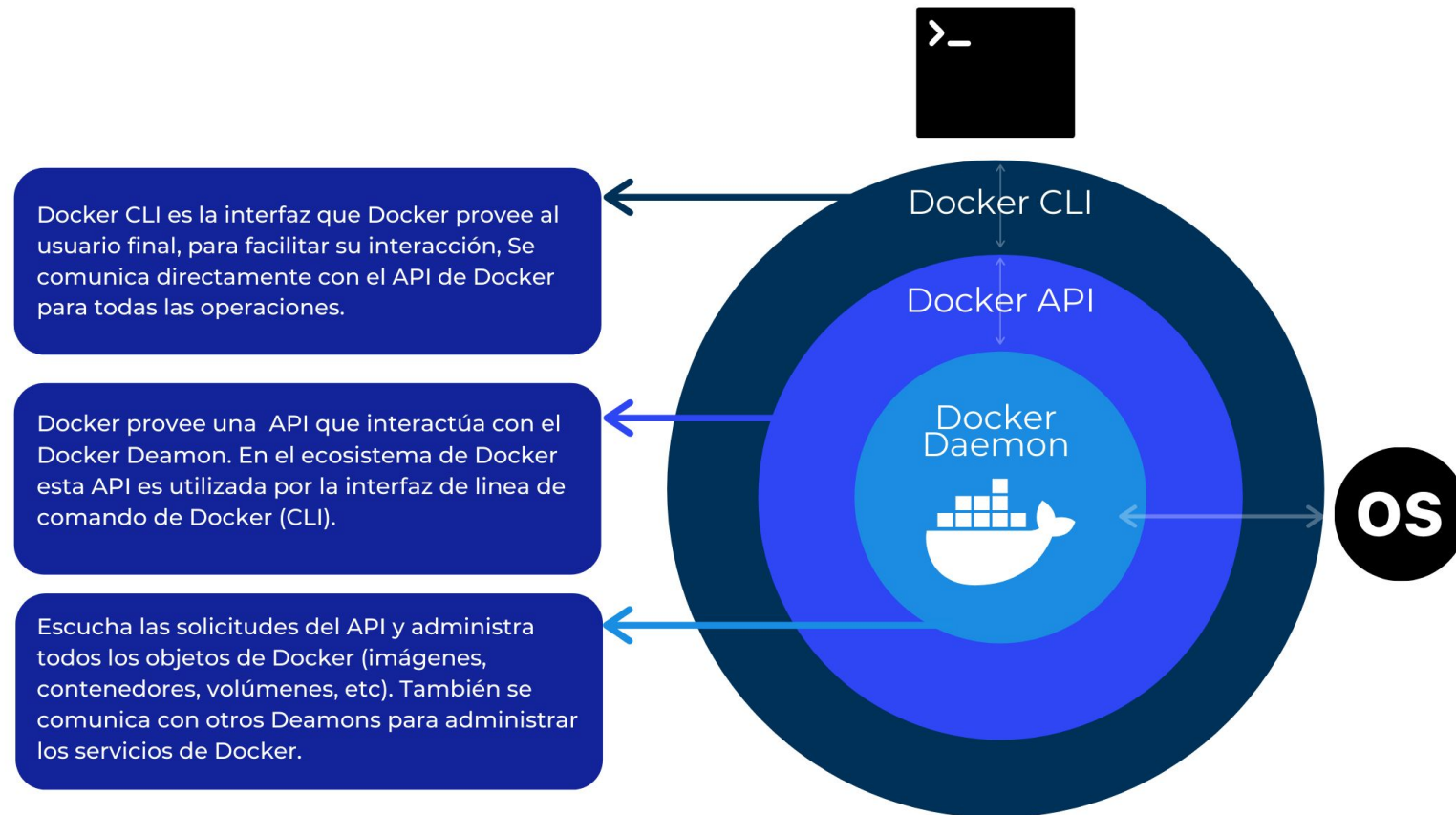
¿Qué es Docker?

- Es una herramienta para automatizar la implementación de aplicaciones a través de contenedores.
- Es un proyecto de código abierto.

Ecosistema de Docker

- Docker engine
 - Docker CLI
 - Docker API
 - Docker Daemon

Interacciones en el ecosistema de Docker

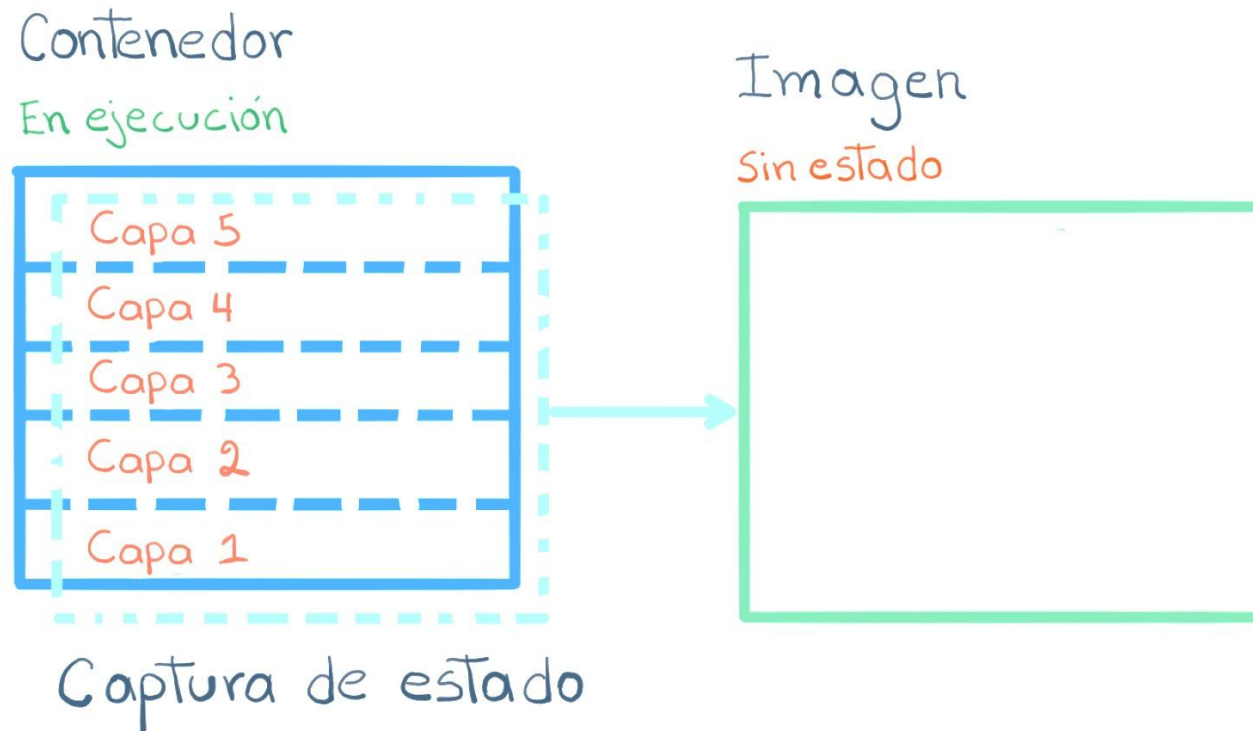


Imágenes de Docker

¿Qué es imagen de Docker?

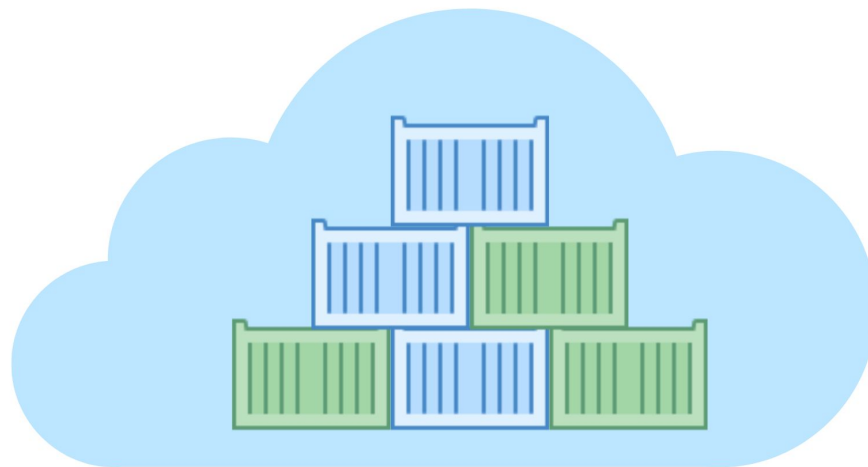
Es una captura del estado de un contenedor, la cual puede ser instanciada n veces dentro de un mismo host, o en diferentes hosts.

Origen de una imagen de Docker



Docker Hub

Docker Registry



Dockerfile

¿Qué es un Dockerfile?

Es un archivo de definición desde el cual se crea una imagen de Docker.

Anatomía de un Dockerfile

```
FROM node:12-alpine

RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app

WORKDIR /home/node/app

COPY app/package.json ./

USER node

RUN npm install

COPY --chown=node:node app/ .

EXPOSE 8080

CMD [ "node", "app.js" ]
```

Comandos de Docker

Construcción

```
# Construye una imagen con el nombre app.  
$ docker build .  
# Construye una imagen desde un Dockerfile remoto.  
$ docker build github.com/creack/docker-firefox  
# Para etiquetar la imagen se puede utilizar el argumento  
# --tag o -t. Si el Dockerfile se encuentra en otra  
# ubicación, enviar la ruta con el argumento -f
```

Ejecución

Ejecuta un contenedor con el o los comandos definidos en
el *entrypoint*, le asigna el nombre app.

```
$ docker run --name app nginx
```

Ejecuta el comando "echo hola" dentro del contenedor con
el nombre app, el contenedor debe estar en ejecución.

```
$ docker exec app echo hola
```

Ingresar al contenedor, los argumentos *-it* proveen una
terminal interactiva dentro del contenedor.

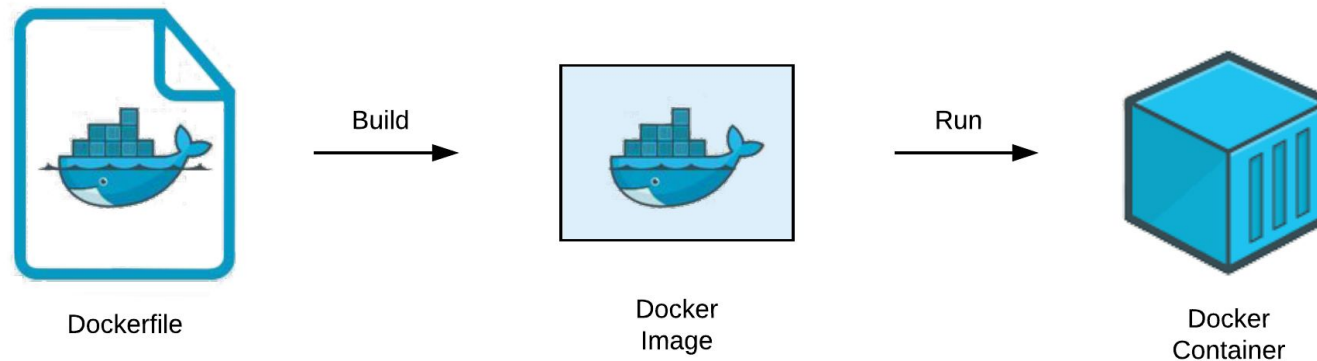
```
$ docker exec -it app bash
```

* Agregando el argumento *-d* docker deja el el proceso en
background.

Utileria

```
# Elimina la imagen nginx, deteniendo los contenedores
# relacionados.
$ docker rmi nginx -f
# Lista todos los contenedores, incluyendo los detenidos.
$ docker ps -a
# Inicia o detiene un contenedor previamente ejecutado.
$ docker stop | start app
# Muestra los logs del contenedor app en ejecución.
$ docker log app -f
```

Proceso de Construcción y ejecución



Docker multi-stage

La funcionalidad multi-stage de Docker permite definir múltiples pasos en un mismo Dockerfile para construir una imagen, utilizando múltiples imágenes base (una en cada paso).

```
FROM golang:1.7.3
WORKDIR /go/src/github.com/alexellis/href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .
```

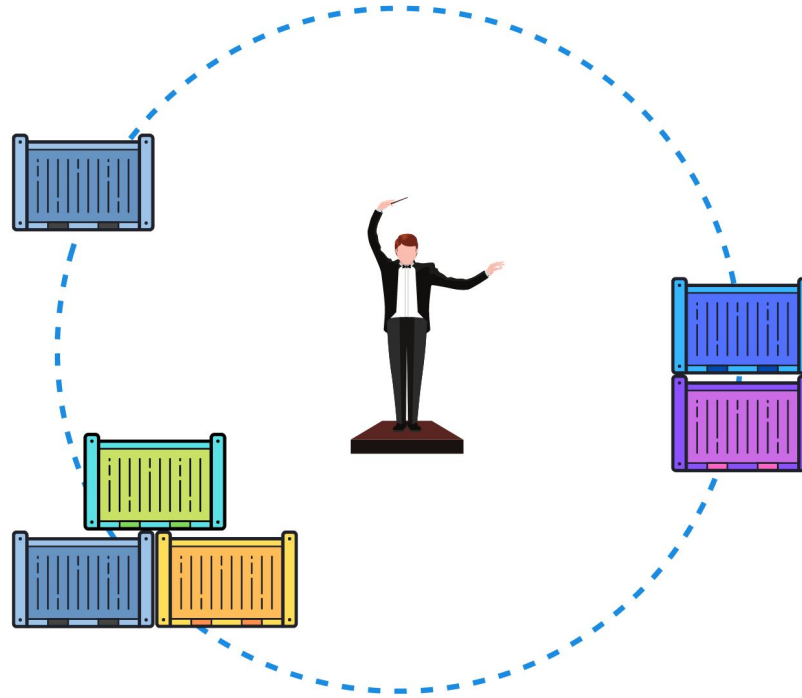
```
FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=0 /go/src/github.com/alexellis/href-counter/app .
CMD ["/app"]
```

Publicar una imagen a Docker Hub

Orquestación de contenedores

¿Que es un orquestador de contenedores?

Es una herramienta que se encarga de administrar correctamente un entorno de contenedores.



Docker Compose

Docker Compose es una herramienta para definir y ejecutar aplicaciones en múltiples contenedores de Docker.

Proceso de implementación



