

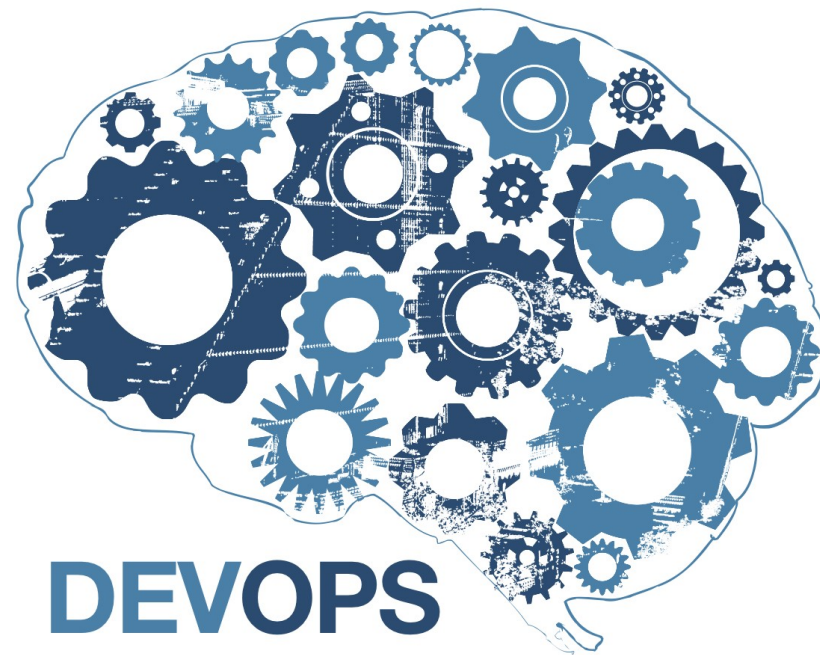
INTERMEDIO

## Unidad 04 Docker

→ CURSOS DE  
DESARROLLO

# Objetivos

¿Qué voy a a aprender?

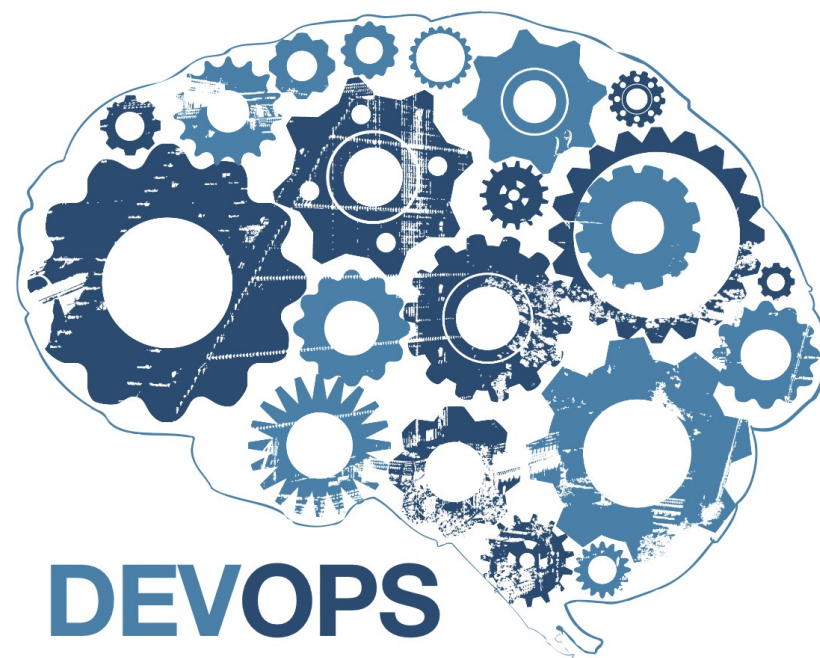


- Conocer los fundamentos de Docker
- Saber crear imágenes
- Saber montar contenedores



# Contenidos

¿Cómo voy a aprenderlo?



- 1.Introducción
- 2.Infraestructura de hoy
- 3.Sistemas de contenedores
- 4.Arquitectura Docker
- 5.Herramientas Docker
- 6.Docker Hub
- 7.Docker Compose
- 8.Kubernetes
- 9.Conclusiones

# Introducción

¿Por dónde empezamos?



¿Qué es Docker?  
<https://www.docker.com/>

# Introducción

¿Por dónde empezamos?



Docker es una plataforma de software libre que permite la ejecución de servicios en forma de contenedores de una manera automatizable

# Introducción

¿Por dónde empezamos?



Ayuda a la puesta en marcha de una  
infraestructura compleja desde la perspectiva del  
desarrollador



# Introducción

¿Por dónde empezamos?



Al manejar las mismas versiones que en  
producción reduce los fallos de  
incompatibilidades entre desarrollo y producción



# Introducción

¿Por dónde empezamos?



Es muy ligero a la hora de ejecutar stacks completos

# Introducción

¿Por dónde empezamos?



Está enfocado a la creación de entornos seguros de producción

# Introducción

¿Por dónde empezamos?



Básicamente es una evolución de chroot



# Introducción

¿Por dónde empezamos?



Es muy fácil desplegar entornos de múltiples servicios

# Introducción

¿Por dónde empezamos?



Es muy flexible a nivel de puesta en marcha de nuevos servicios y muy adaptable

# Infraestructura a día de hoy

¿Por dónde empezamos?



A día de hoy existen distintos tipos de infraestructura que están desplegados en las empresas



# Infraestructura a día de hoy

¿Qué modelos tenemos?



Modelo tradicional, tu compras los equipos,  
instalas los sistemas operativos y las soluciones  
de los fabricantes

# Infraestructura a día de hoy

¿Qué modelos tenemos?



# Infraestructura a día de hoy

¿Qué modelos tenemos?

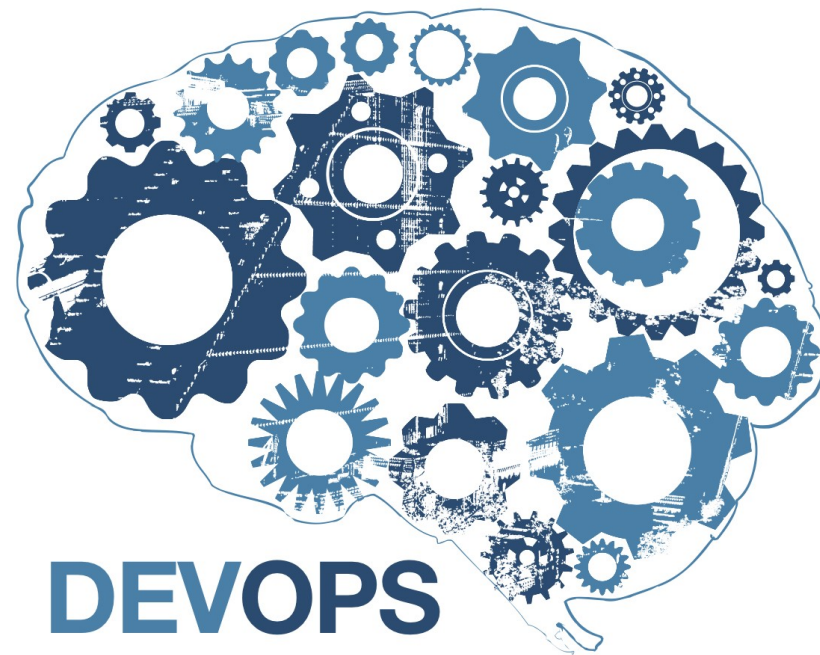


Modelo subcontratado, antiguos proveedores de servicios de internet, ellos instalan los sistemas y las máquinas y tú instalas lo demás a mano



# Infraestructura a día de hoy

¿Qué modelos tenemos?



**ISPConfig**  
hosting control panel

LOGOUT ADMIN

Search

Home System Client DNS Help Email Monitor Sites Tools VServer

### Welcome admin

#### Latest news

- 2014-04-25  
ISPConfig 3.0.5.4 Patch1 released
- 2014-04-23  
ISPConfig 3.0.5.4 final released
- 2014-04-15  
ISPConfig 3.0.5.4 RC2 released
- 2014-04-08  
How to find out if your server is affected from OpenSSL Heartbleed vulnerability (CVE-2014-0160) and how to fix that
- 2014-04-03  
ISPConfig 3.0.5.4 RC1 released
- 2014-03-19

#### Available Modules

System Client Email Monitor

Sites DNS VServer Tools

Help

#### Account limits

Number of email domains	0 of Unlimited
Number of mailing lists	0 of Unlimited
Number of mailboxes	0 of Unlimited
Number of email aliases	0 of Unlimited
Number of domain aliases	0 of Unlimited
Number of email forwarders	0 of Unlimited
Number of email catchall accounts	0 of Unlimited
Number of email routes	0 of Unlimited
Number of email filters	0 of Unlimited
Number of fetchmail accounts	0 of Unlimited

# Infraestructura a día de hoy

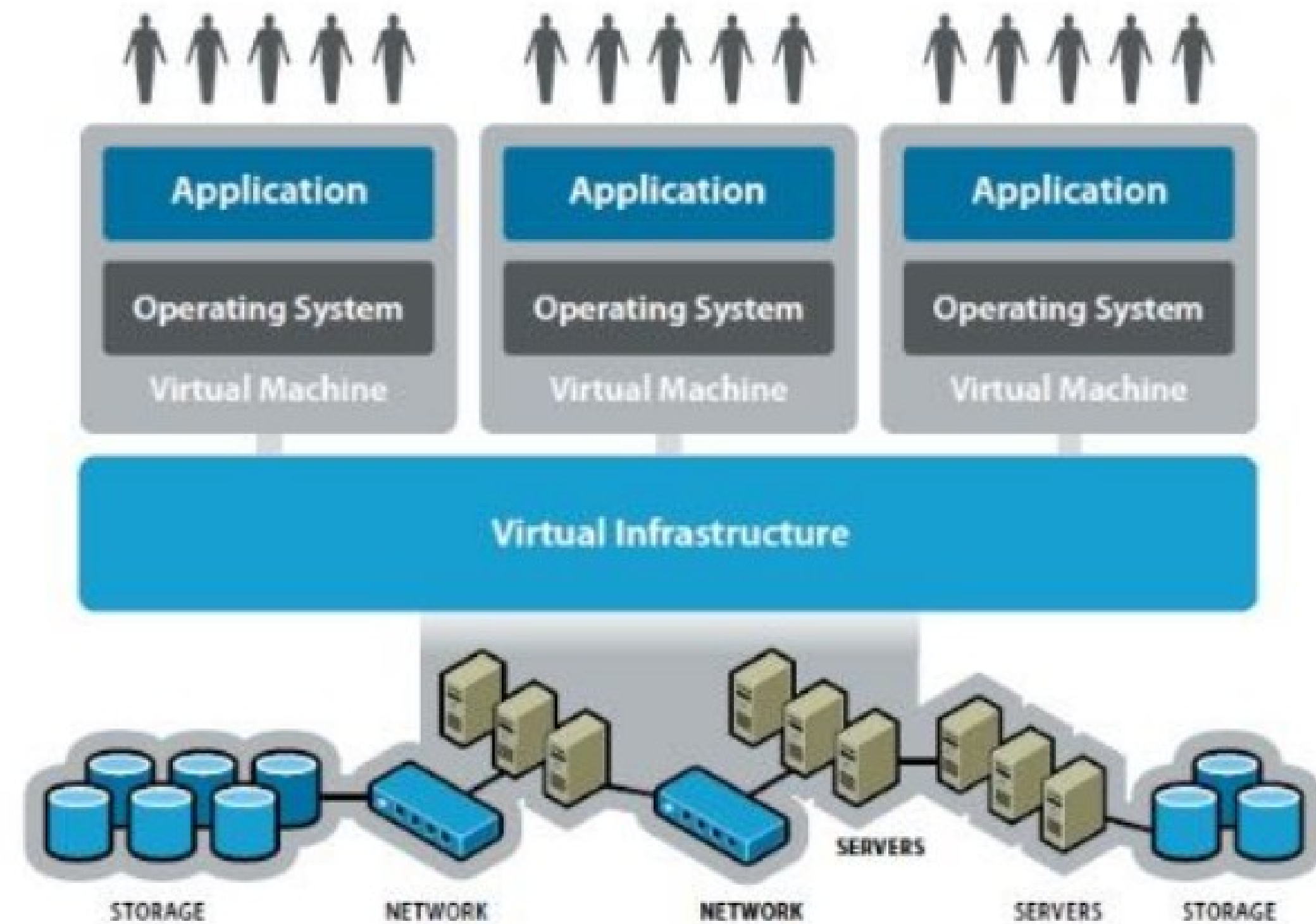
¿Qué modelos tenemos?



Modelo Virtualizado, mantienes el control de la infraestructura, tienes una serie de host que permiten la ejecución de máquinas virtuales y sobre ellas tú despliegas las soluciones

# Infraestructura a día de hoy

¿Qué modelos tenemos?





# Infraestructura a día de hoy

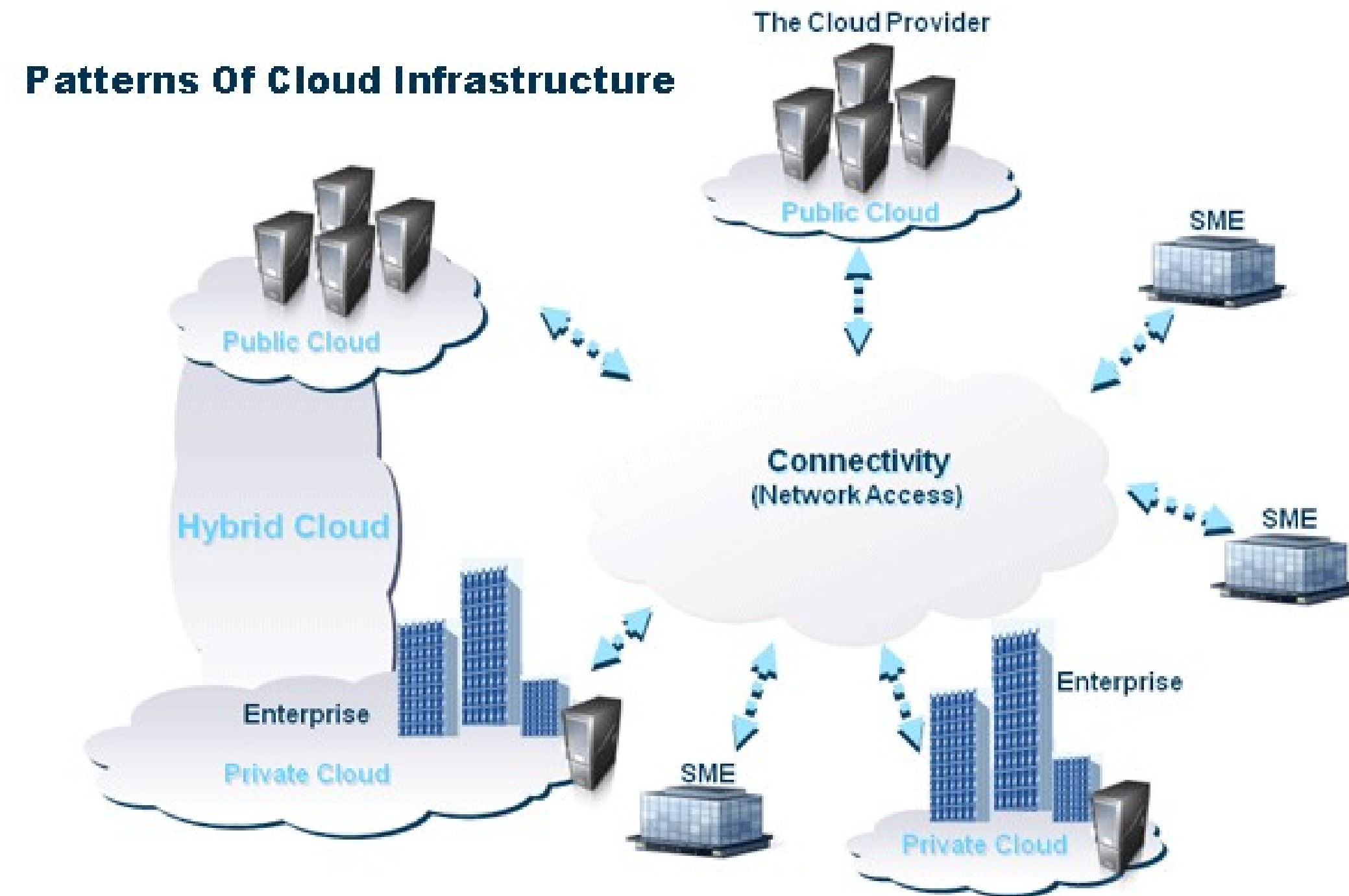
¿Qué modelos tenemos?



Modelo Cloud, evolución del modelo de virtualización donde delegamos la gestión de la infraestructura de virtualización en un tercero

# Infraestructura a día de hoy

¿Qué modelos tenemos?



# Sistemas de Contenedores

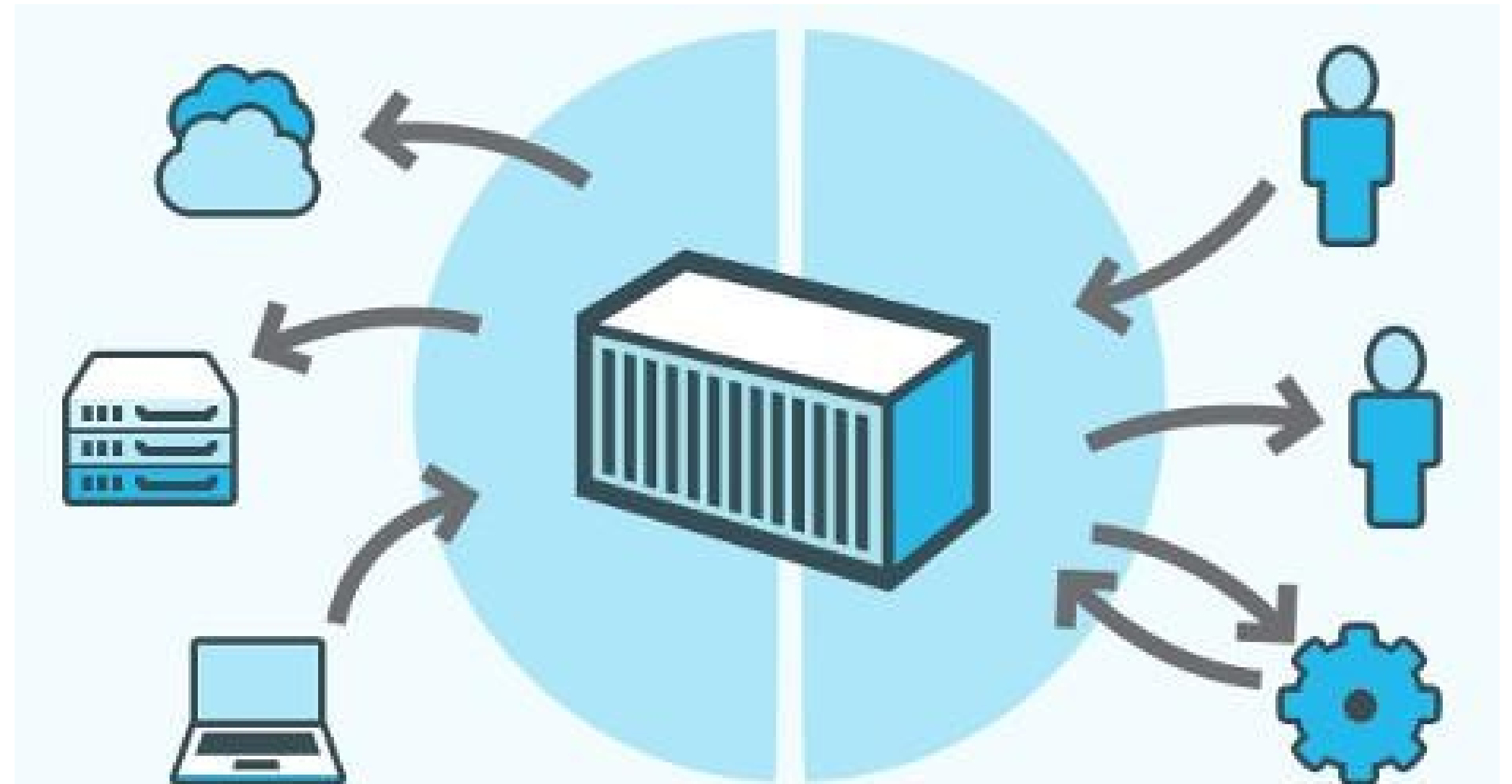
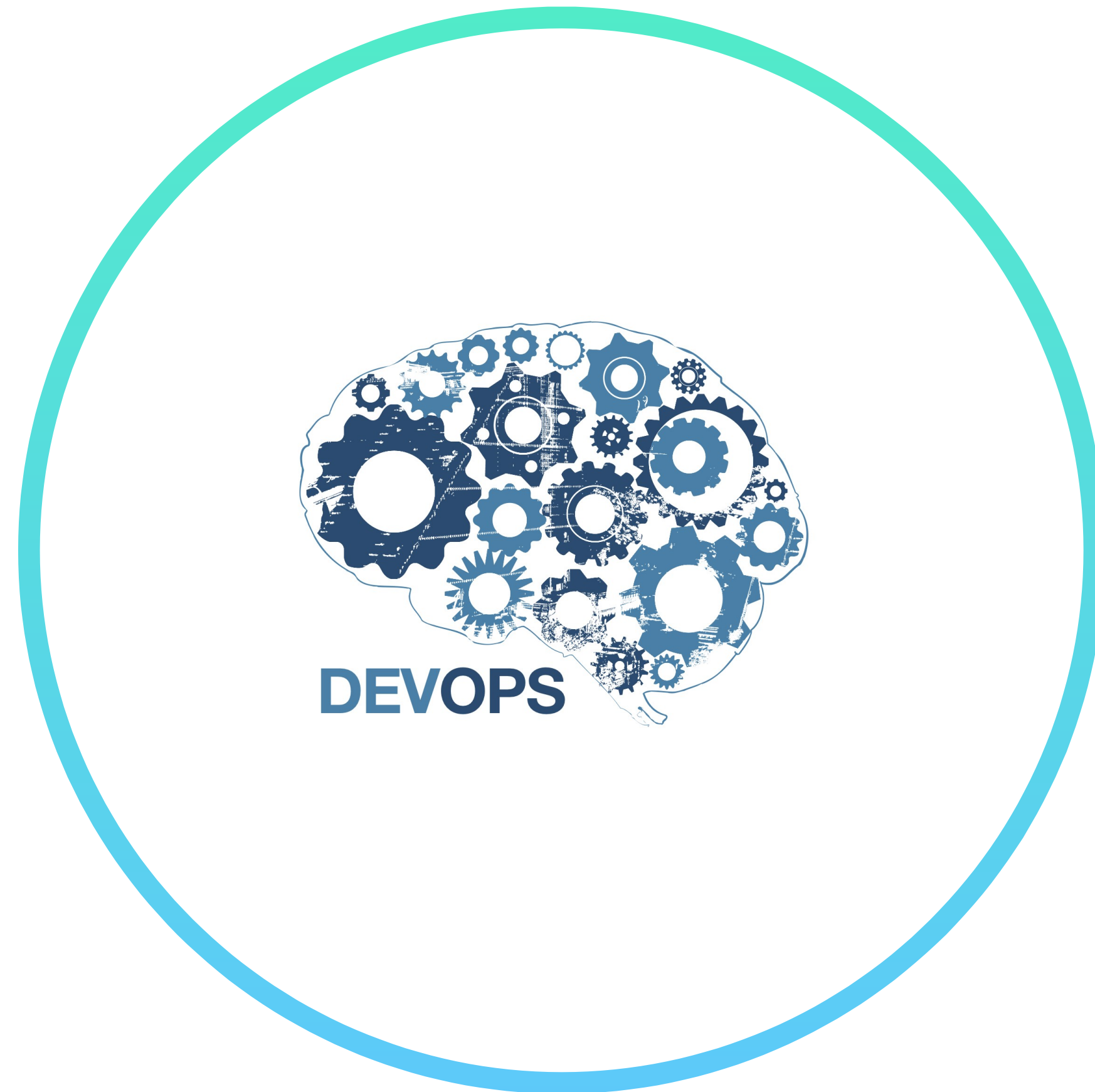
¿Qué es diferente?



Un contenedor nos va a permitir introducir un determinado servicio de una manera replicable

# Sistemas de Contenedores

¿Qué es diferente?





# Sistemas de Contenedores

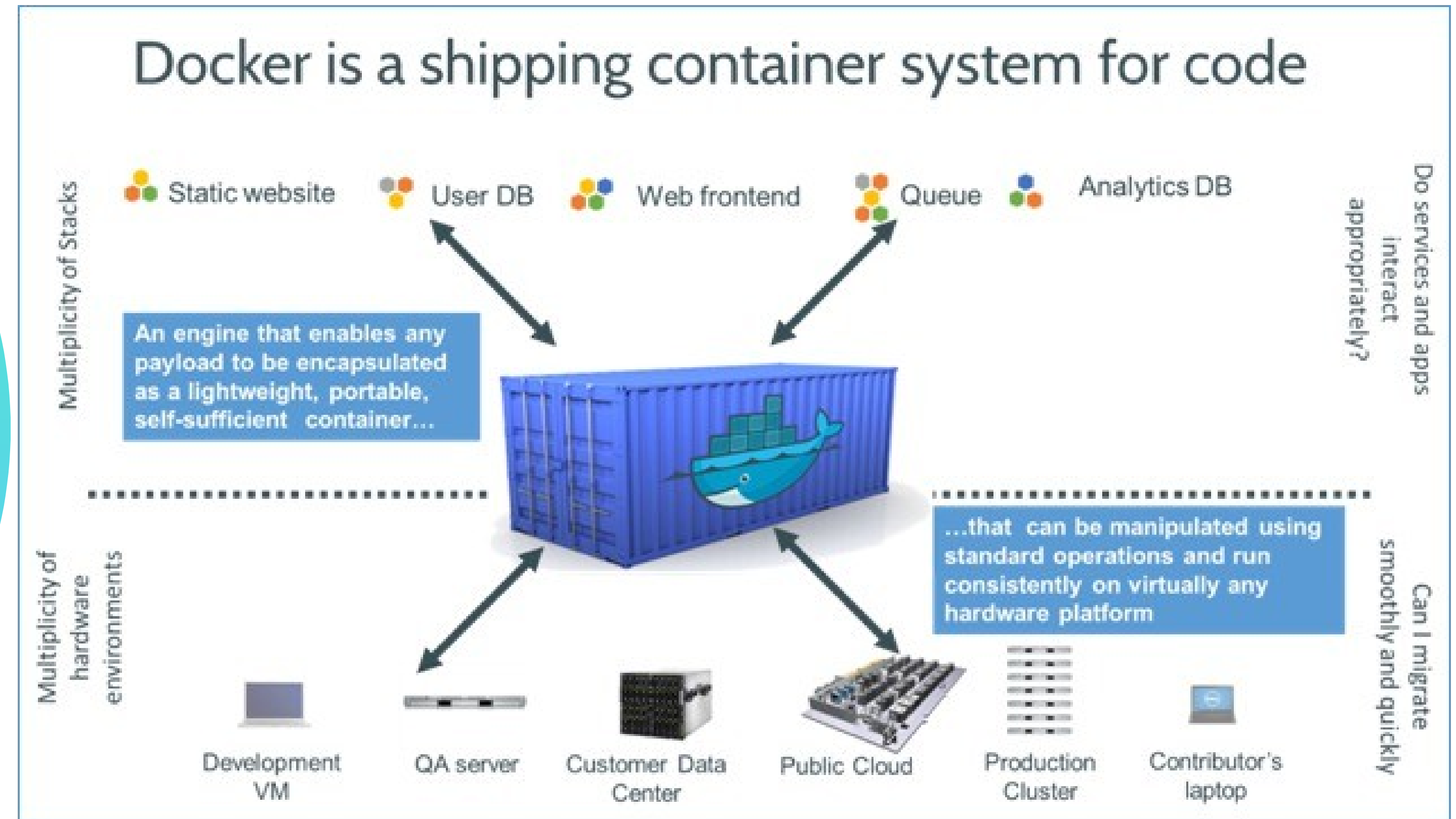
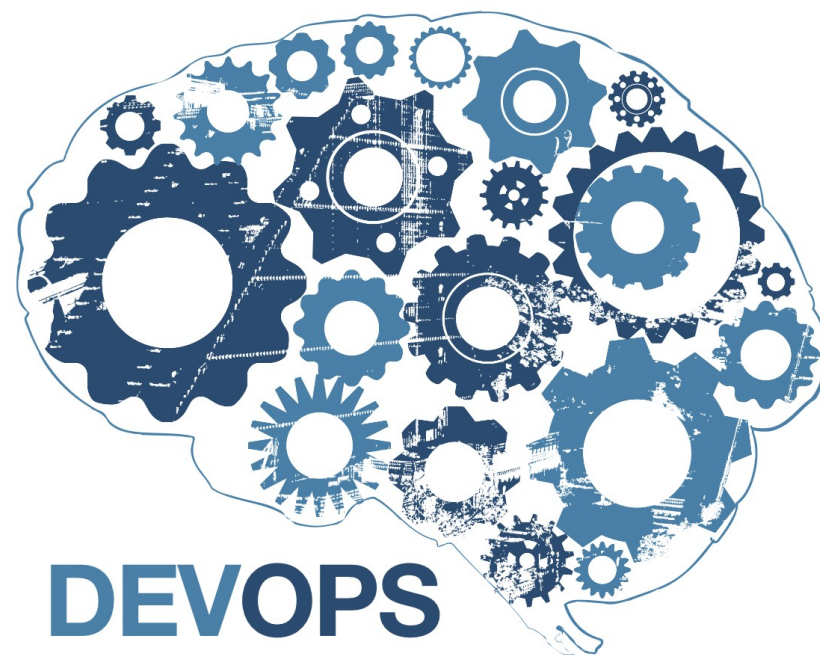
¿Qué es diferente?



Prácticamente cualquier servicio puede llegar a componentizarse a nivel de docker

# Sistemas de Contenedores

¿Qué es diferente?



# Arquitectura Docker

¿Que partes tiene esto?



La infraestructura se ejecuta directamente sobre el sistema operativo y el motor de docker

# Arquitectura Docker

¿Que partes tiene esto?

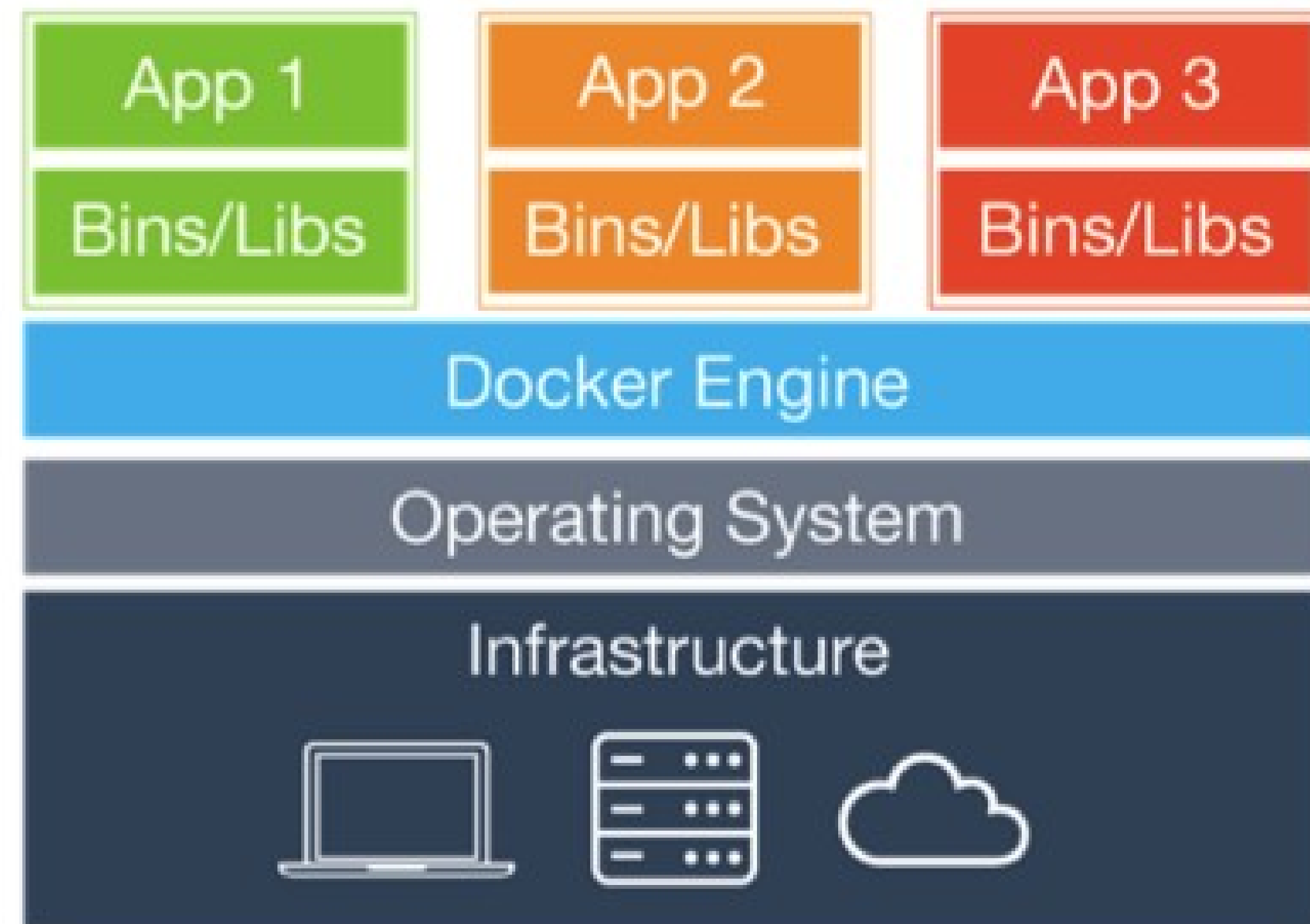
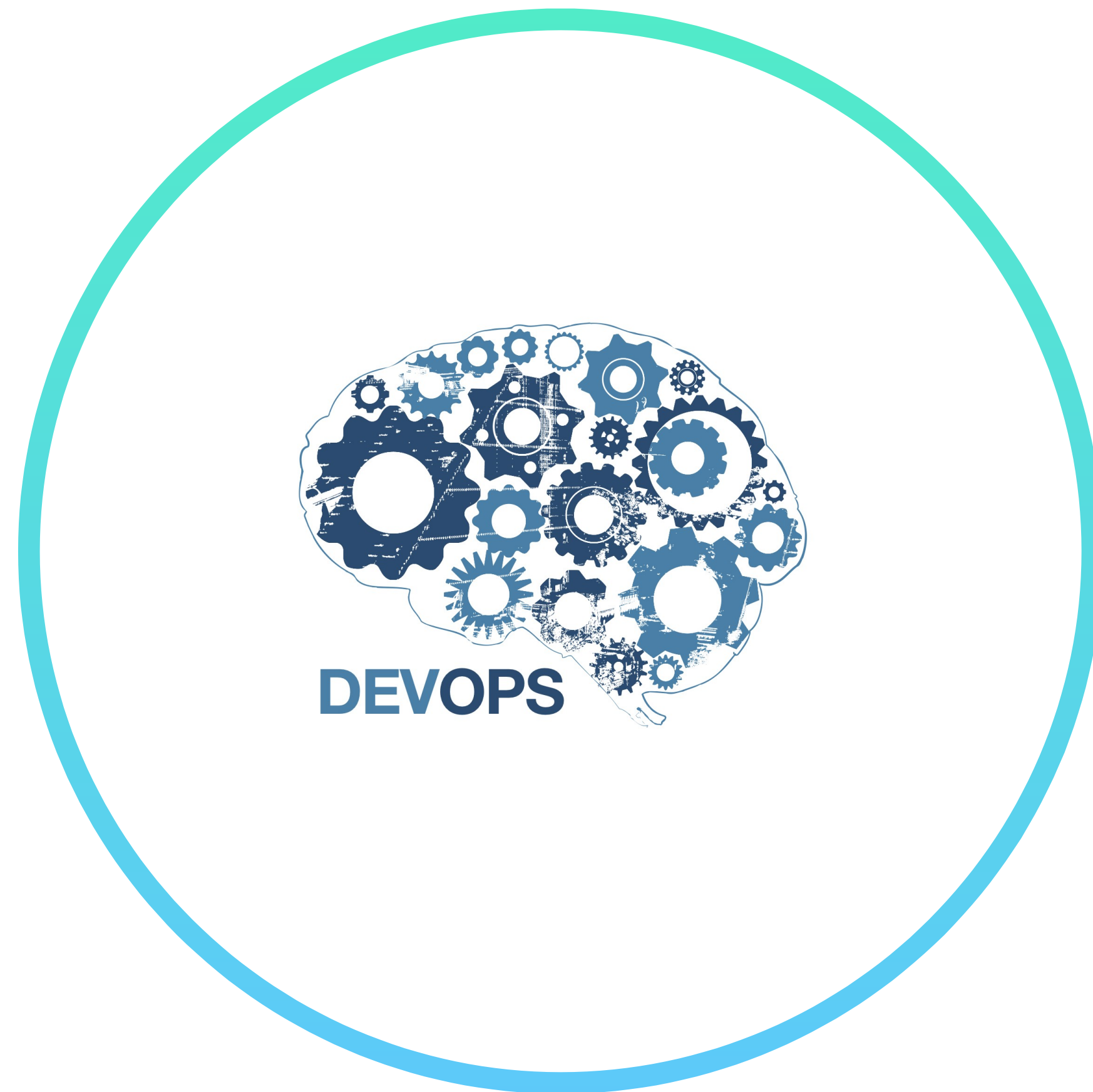


La infraestructura se ejecuta directamente sobre el sistema operativo y el motor de docker



# Arquitectura Docker

¿Que partes tiene esto?



# Arquitectura Docker

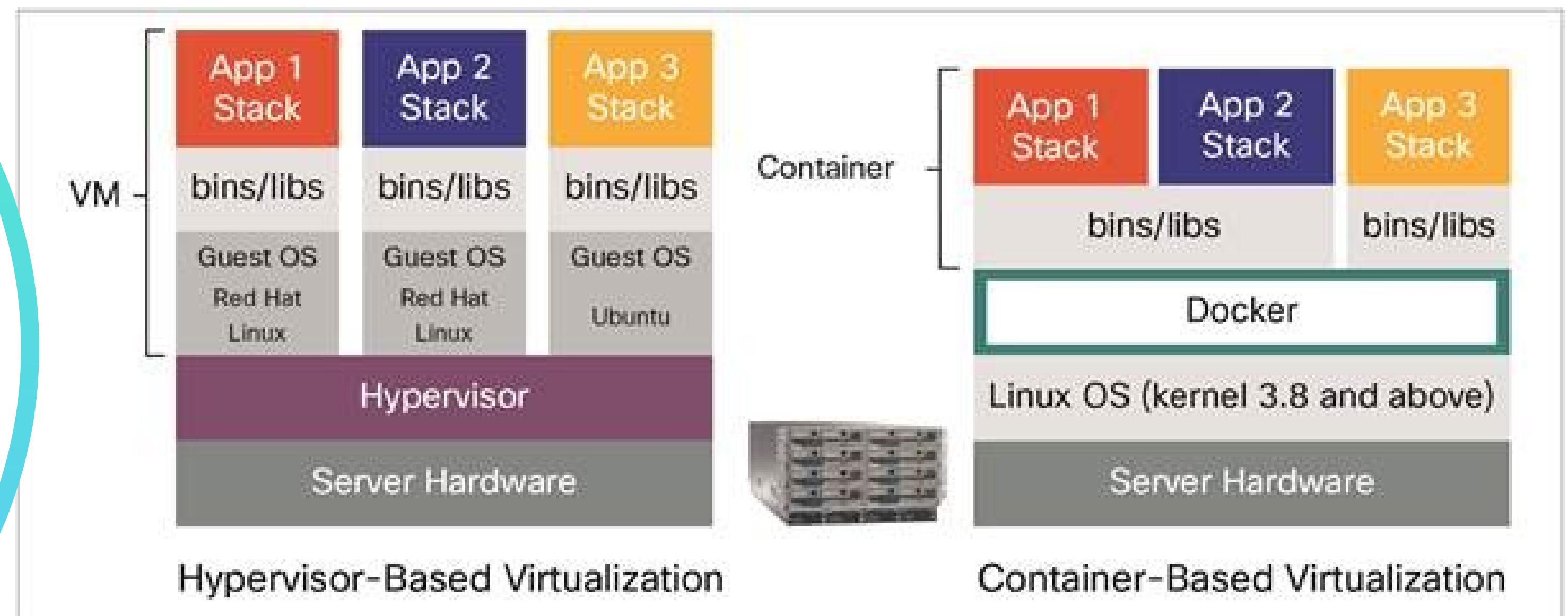
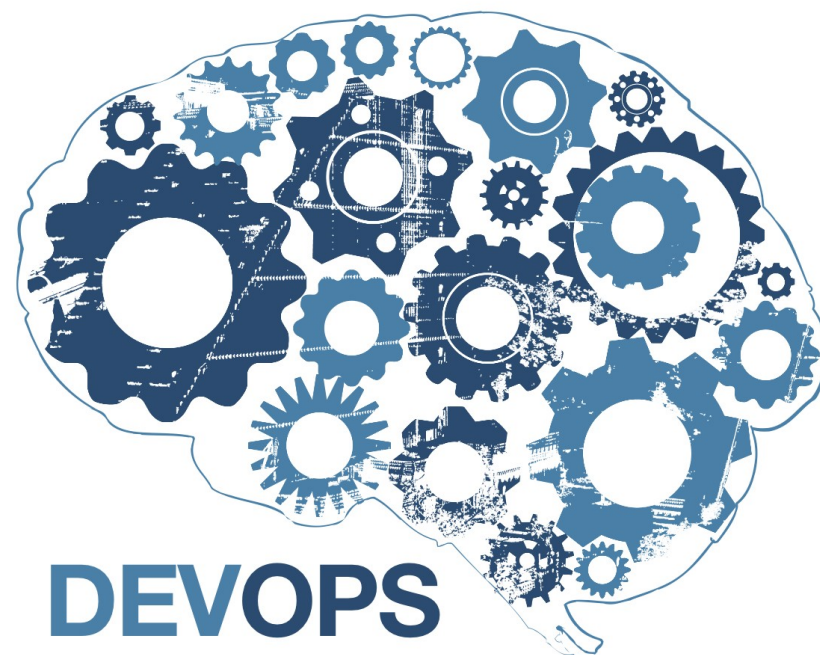
¿Que partes tiene esto?



Comparativamente es mejor solución que la virtualización por cpu, memoria y tiempo

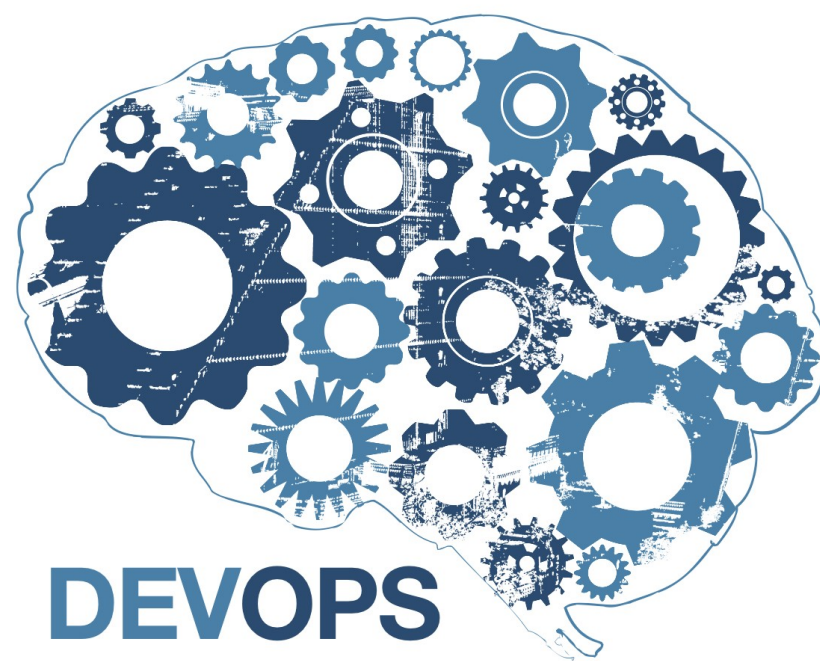
# Arquitectura Docker

¿Que partes tiene esto?



# Arquitectura Docker

¿Que partes tiene esto?

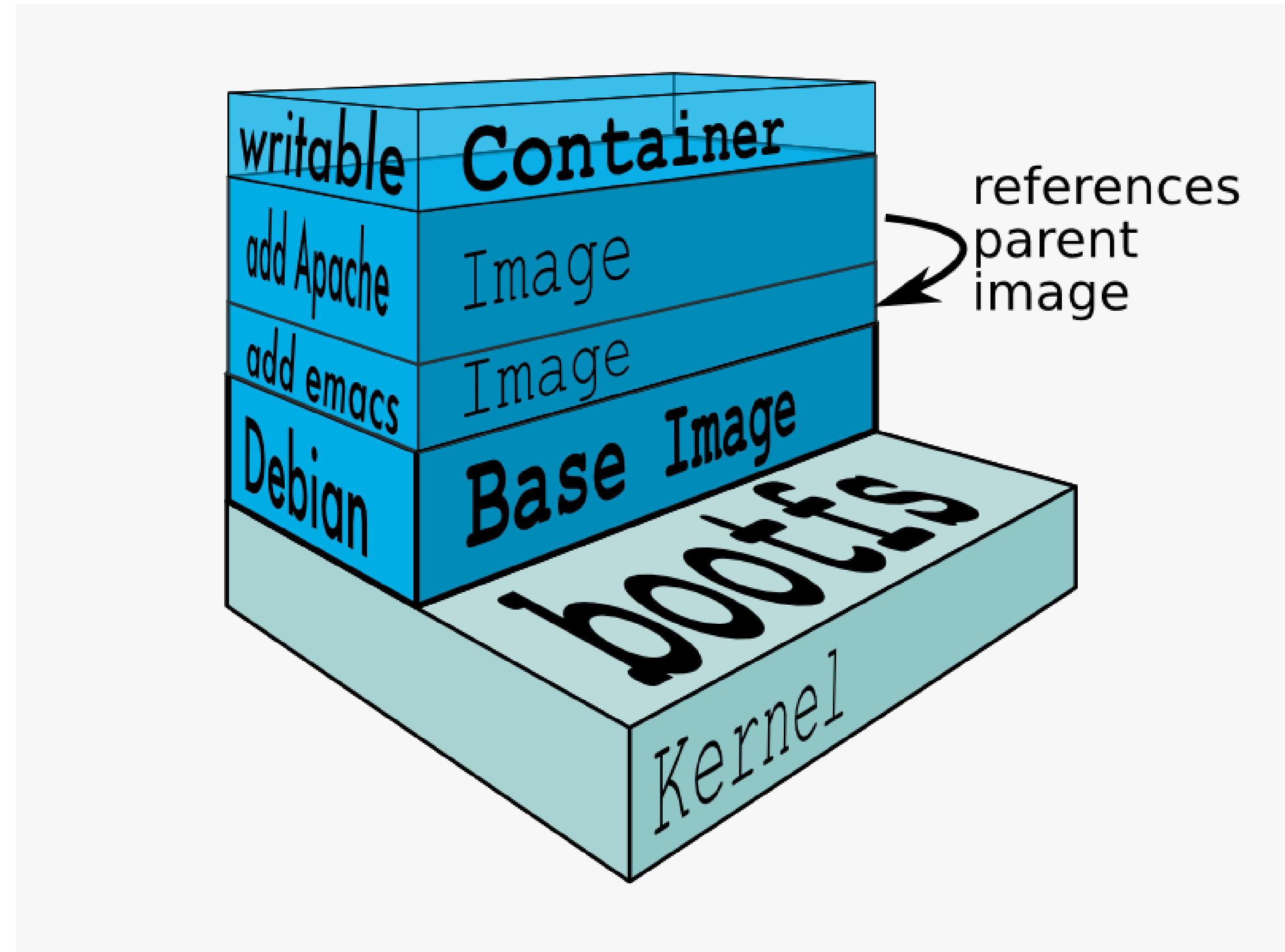


Dentro de cada uno de los contenedores se pueden gestionar distintas capas en el sistema de ficheros que permiten consolidar los cambios hechos en el sistema



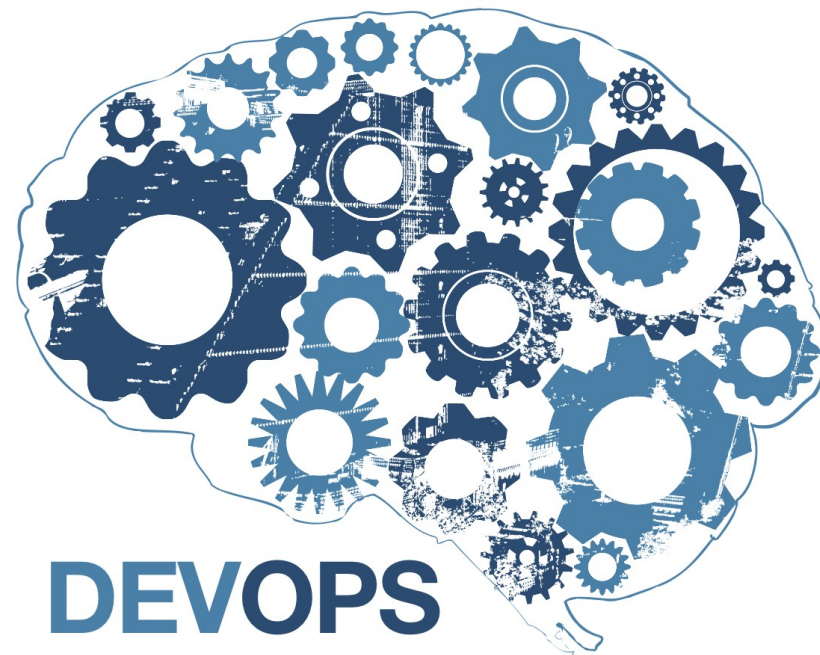
# Arquitectura Docker

¿Que partes tiene esto?



# Herramientas Docker

¿Cómo manejamos esto?



Docker tiene un juego de herramientas que nos van a permitir hacer funcionar los contenedores de servicios

# Herramientas Docker

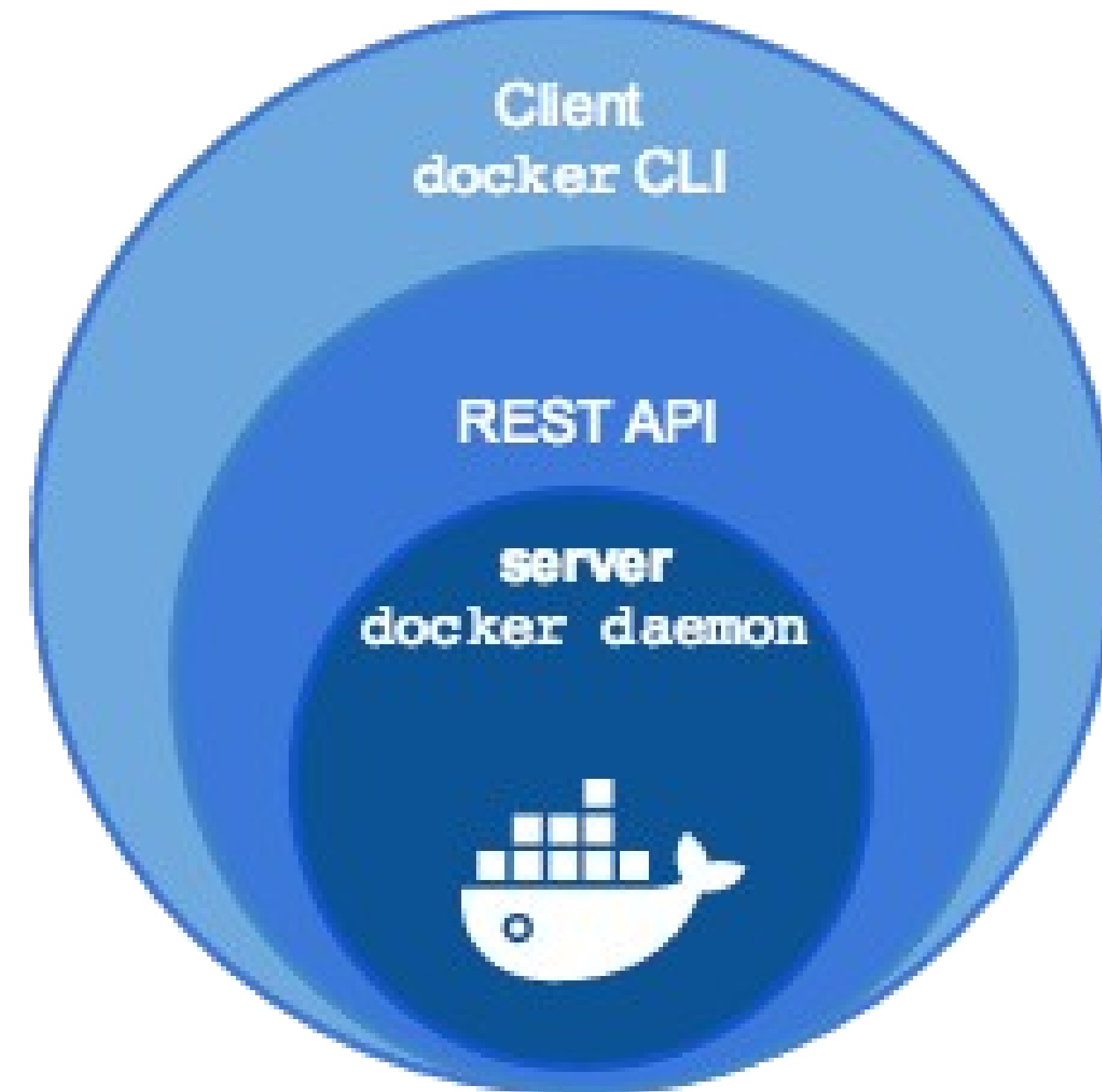
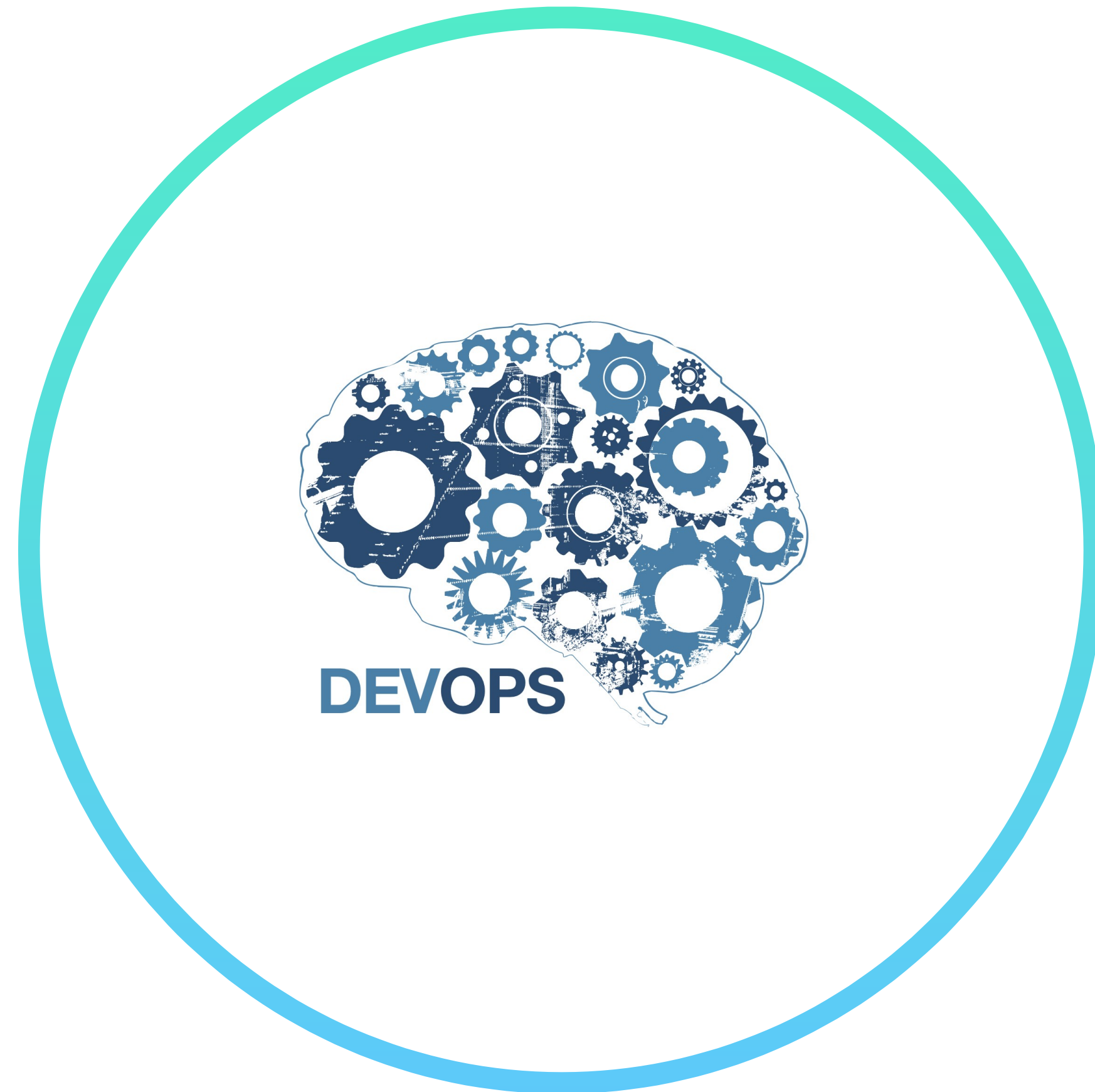
¿Cómo manejamos esto?



La pieza principal del juego es el servidor o engine de docker que es el que puede llegar a ejecutar los contenedores

# Herramientas Docker

¿Cómo manejamos esto?





# Herramientas Docker

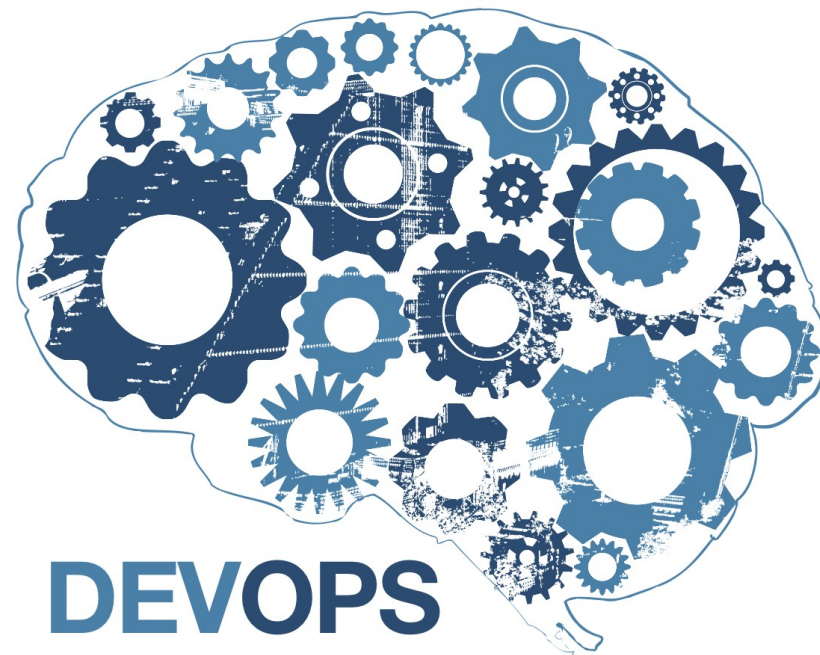
¿Cómo manejamos esto?



El engine nos ofrece un API Rest que nos permite comunicarnos con él a través de un cliente que entienda el API

# Herramientas Docker

¿Cómo manejamos esto?



Inicialmente el API debería podernos ofrecer todas las posibilidades de Docker

# Herramientas Docker

¿Cómo manejamos esto?



En windows o en Mac tenemos disponibles  
varios GUI's que nos sirven de clientes

# Herramientas Docker

¿Cómo manejamos esto?



Existe una nueva versión Enterprise que ofrece versiones mantenidas durante un año y soporte de pago

[https://blog.docker.com/2017/03/docker-enterpris  
eedition/](https://blog.docker.com/2017/03/docker-enterpris-eedition/)



# Herramientas Docker

¿Cómo manejamos esto?



La versión enterprise soporta oficialmente:

AWS y Azure

Windows 10 y Mac

Centos, Fedora, RHEL, Debian y Ubuntu

# Herramientas Docker

¿Cómo manejamos esto?



La Enterprise tendrá contenedores y plugins  
certificados

# Herramientas Docker

¿Cómo manejamos esto?



Docker Datacenter es la herramienta de gestión de contenedores y de seguridad. Permite escalar los servicios. Promete ser más seguro al asegurar el transporte de los contenedores.

<https://www.docker.com/pricing>

# Herramientas Docker

¿Cómo manejamos esto?



Dispone de una tienda donde se pone en marcha la compra de contenedores  
<https://store.docker.com/>



# Docker Hub

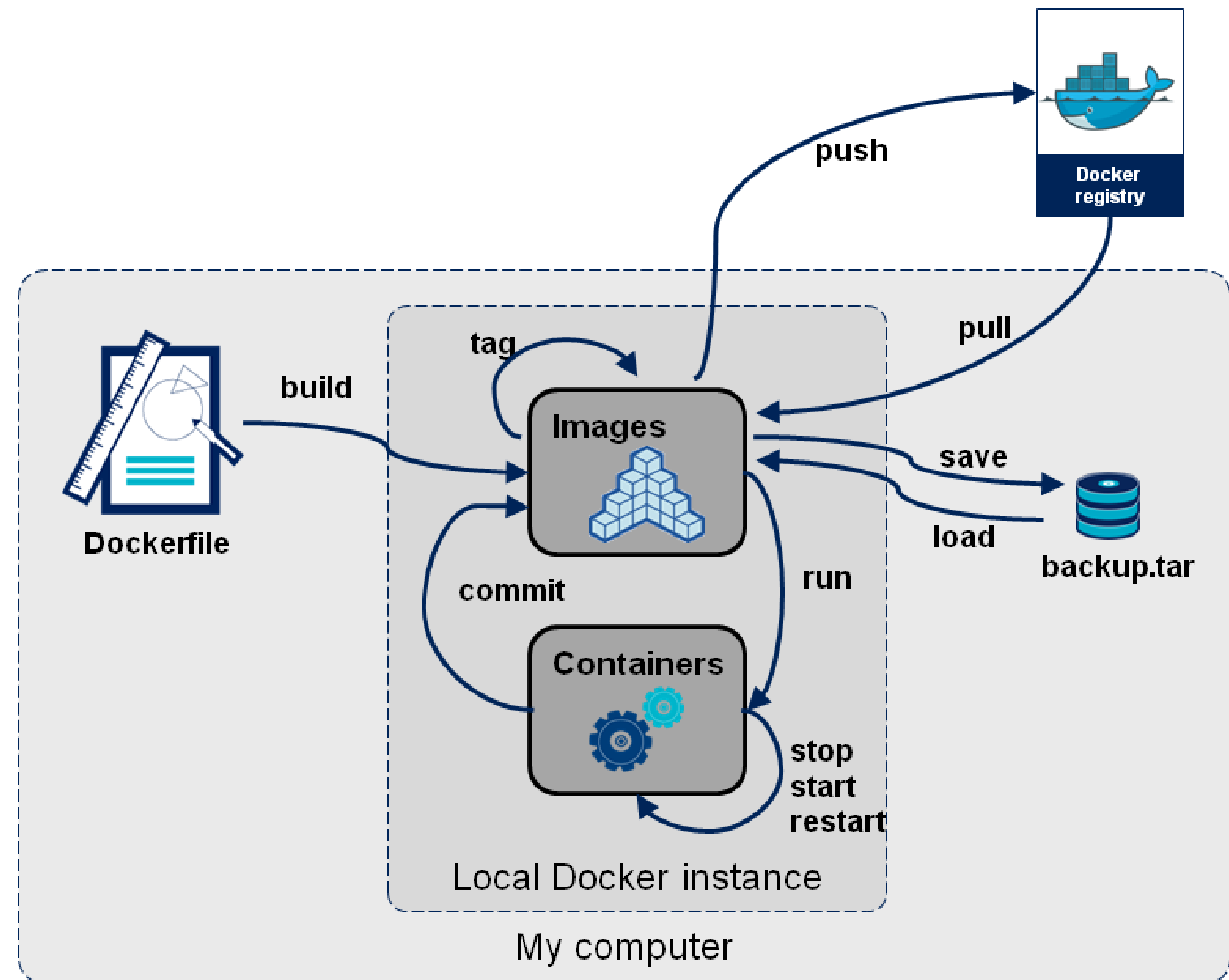
¿Donde estará mi carro?



Es el registro oficial de contenedores de Docker  
<https://hub.docker.com/>

# Docker Hub

¿Donde estará mi carro?



# Docker Compose

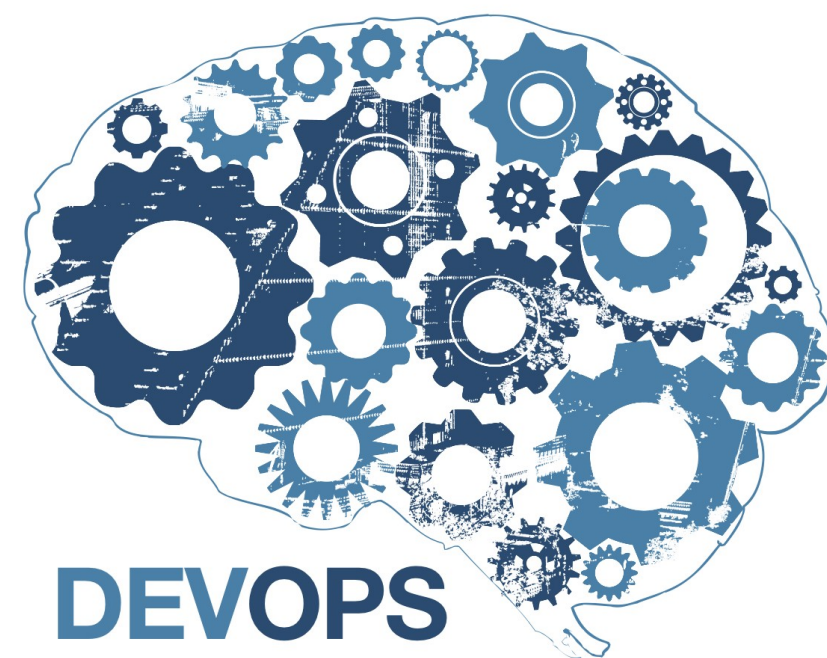
¿Cómo monto el lego?



Docker compose es ha herramienta que permite ejecutar varios contenedores de manera simultánea en un servidor o cluster Docker

# Docker Compose

¿Cómo monto el lego?

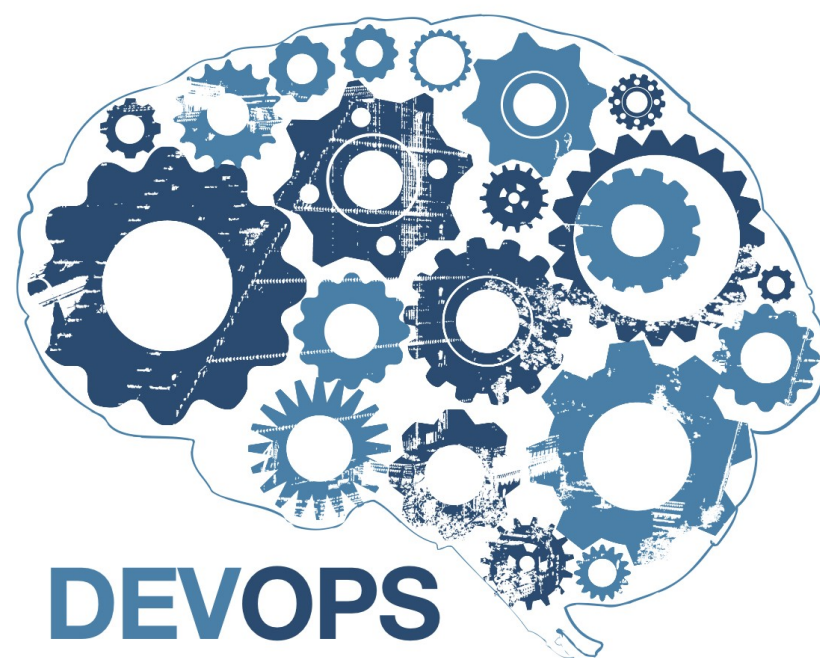


Se trata de un fichero YAML que define que servicio se debe levantar y los parámetros de configuración de cada contenedor



# Docker Compose

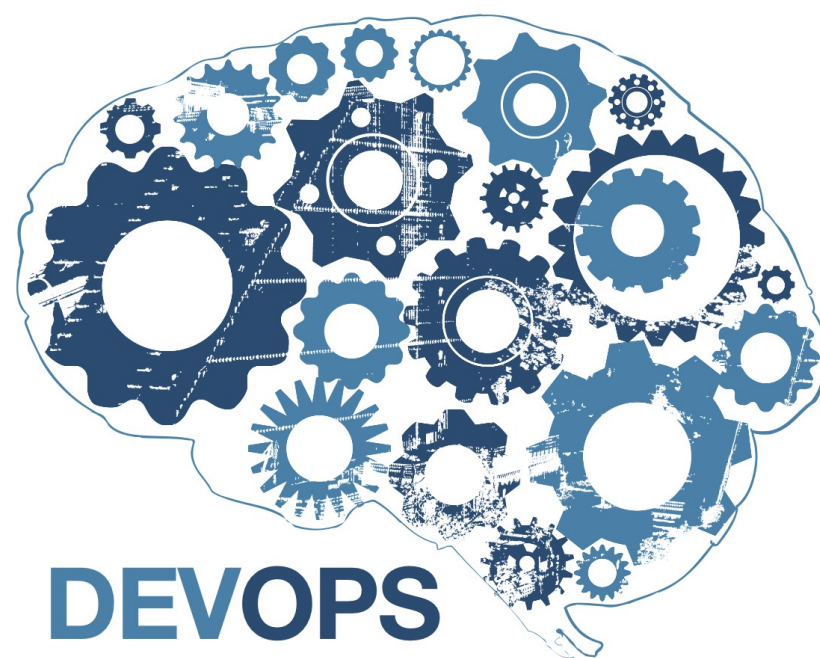
¿Cómo monto el lego?



Dentro de los parámetros estarán:  
Imagen, Volumenes, Puertos, Variables de  
Entorno

# Docker Compose

¿Cómo monto el lego?



Una vez creado el fichero docker-compose.yml  
se ejecutará el comando para levantar los  
contenedores:  
`docker-compose up -d`  
El -d es opcional

# Docker Compose

¿Cómo monto el lego?



Para parar los contenedores ejecutaremos  
docker-compose down  
O Control+C si hemos usado -d

# Kubernetes

¿Cómo monto el lego?

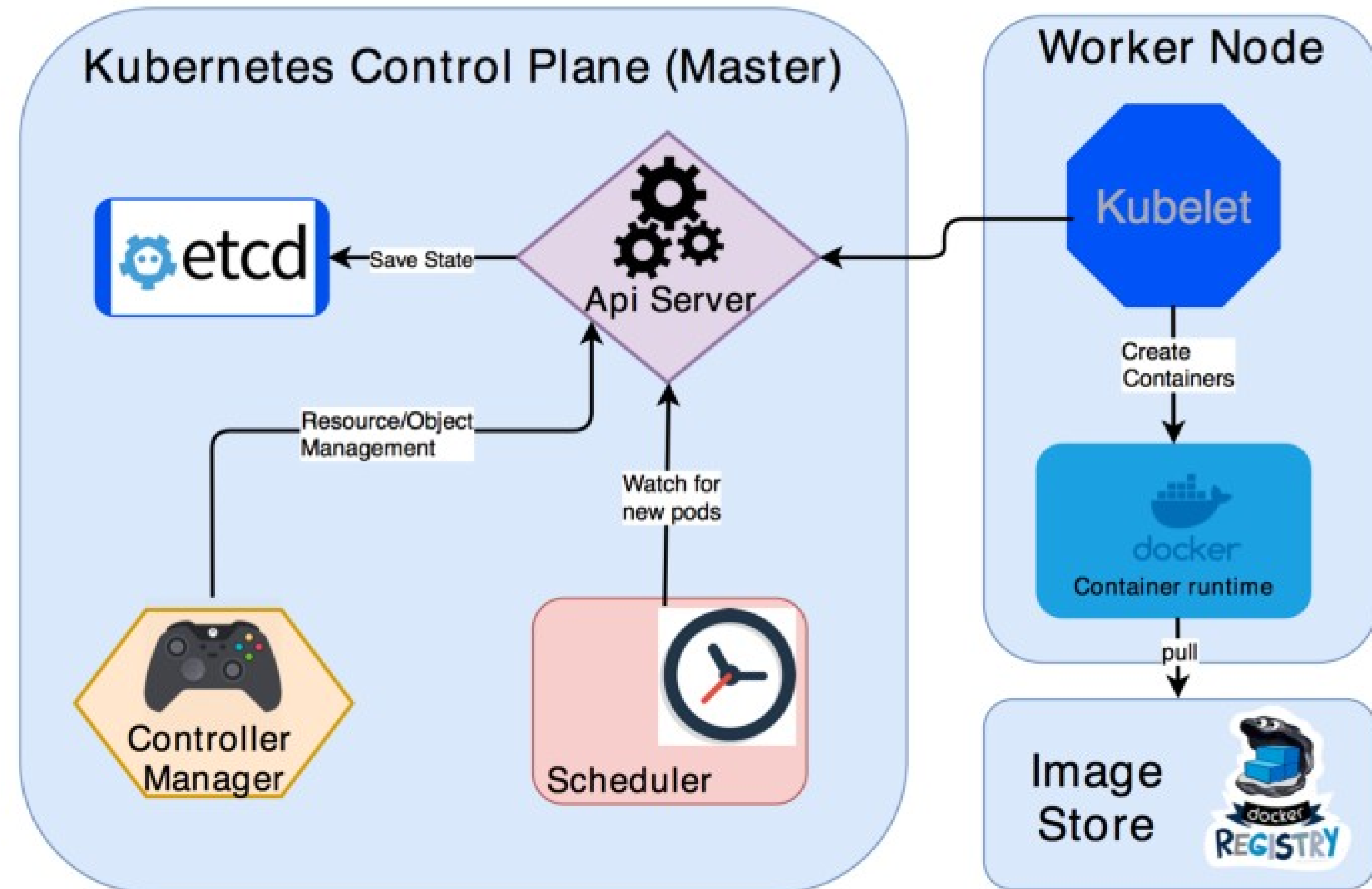
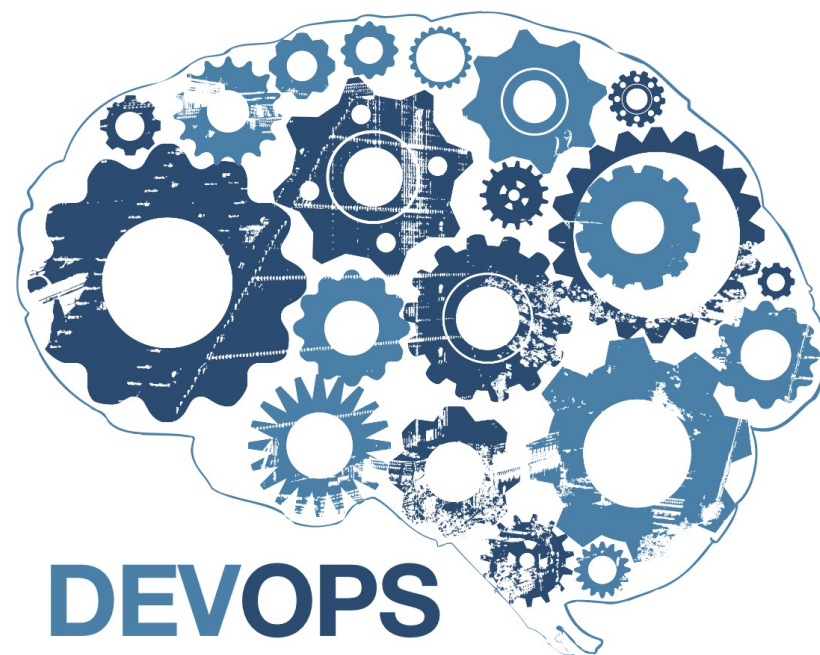


Kubernetes es la herramienta que permite orquestar los contenedores de Docker



# Kubernetes

¿Cómo monto el lego?



# Kubernetes

¿Cómo monto el lego?



Kubernetes define los pods como conjuntos de contenedores que deben funcionar juntos, muy parecido al docker-compose

# Kubernetes

¿Cómo monto el lego?



Un servicio es un conjunto de pods  
sincronizados (escalado)

# Kubernetes

¿Cómo monto el lego?

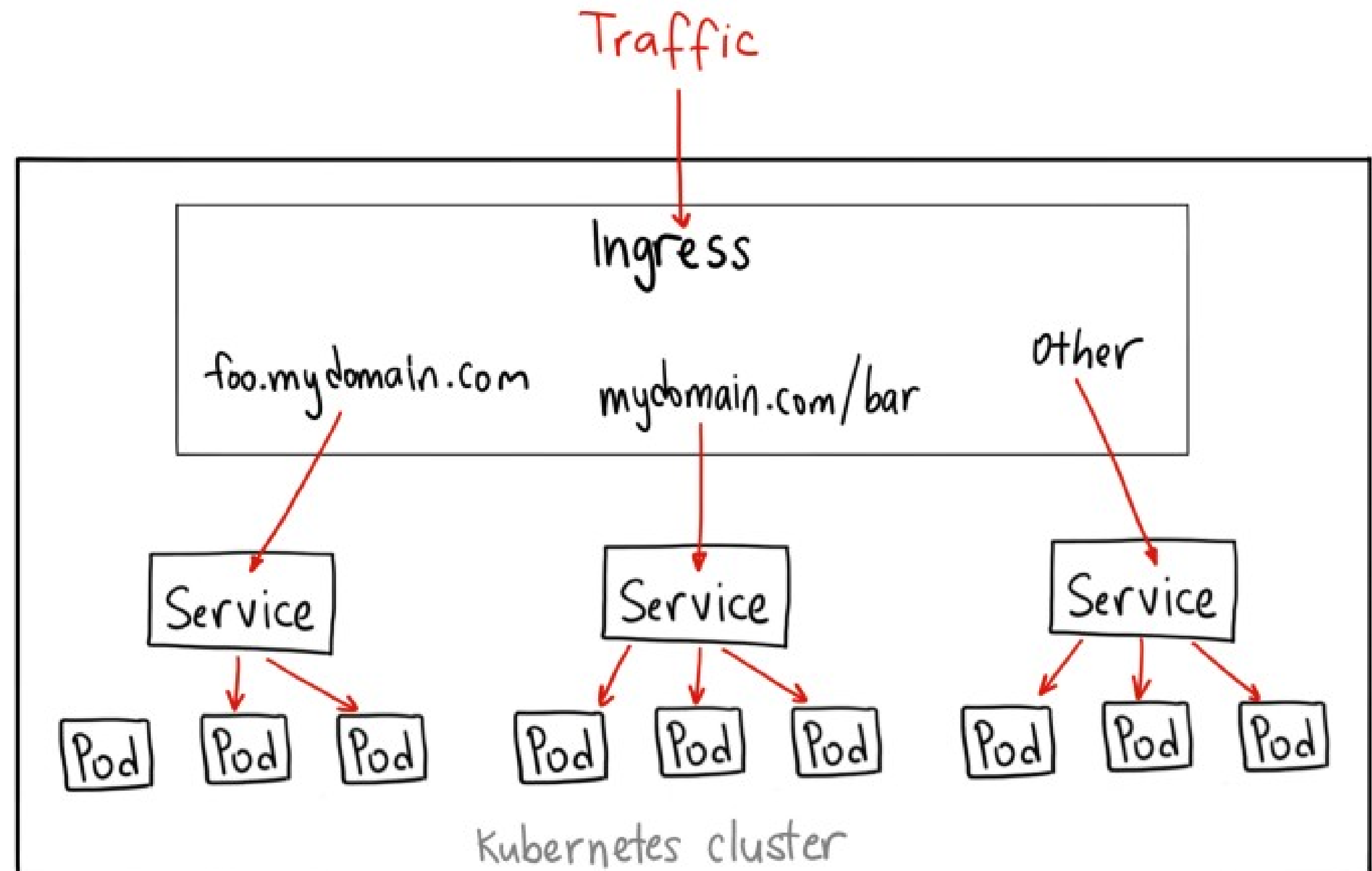


Ingress permite el acceso balanceado a los pods



# Kubernetes

¿Cómo monto el lego?



# Conclusiones

¿Qué podemos sacar en claro?



- Hemos visto los fundamentos de docker
- Hemos visto cómo se gestionar una imagen docker
- Hemos visto cómo levantar contenedores con docker-compose
- Hemos visto la orquestación con Kubernetes

# Referencias

¿Fuentes de información?



Cosecha propia :)

**CURSOS DE DESARROLLO**  
**DAVID VAQUERO**  
**LICENCIA CC-BY-SA-NC 4.0**

[info@cursosdedesarrollo.com](mailto:info@cursosdedesarrollo.com)

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

<http://cursosdedesarrollo.com>