

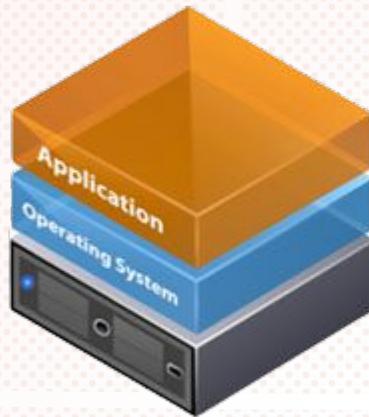


# CONTENEDORES

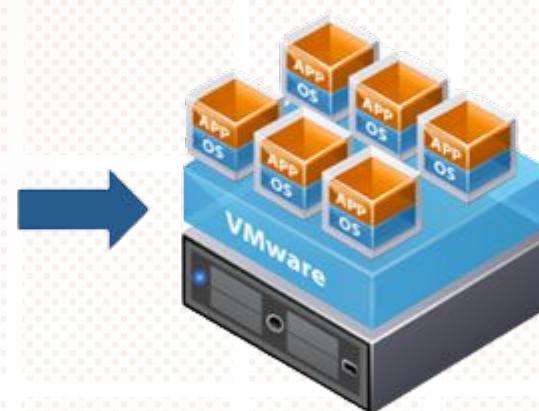
Los contenedores son una tecnología de virtualización.



# Definir virtualización

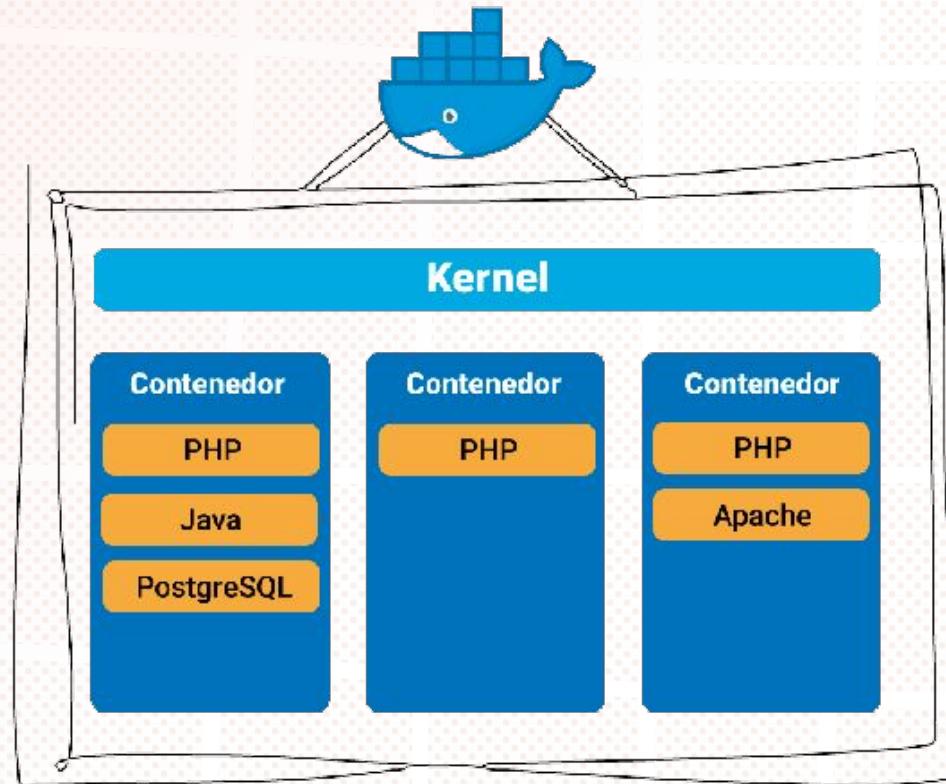


Arquitectura tradicional

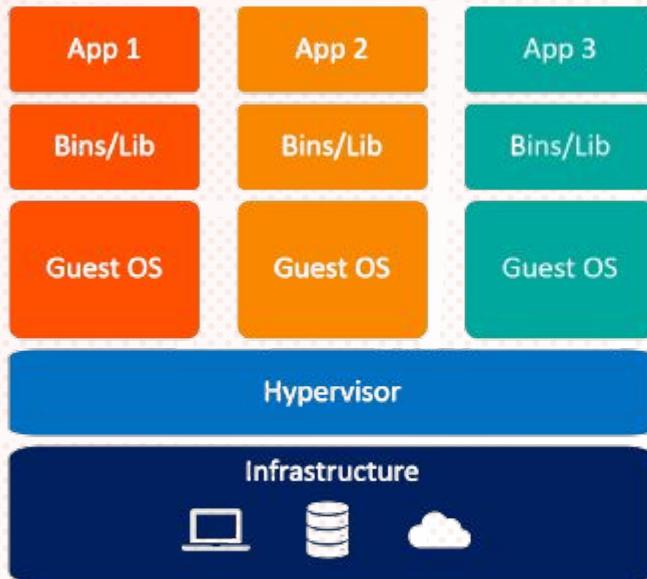


Arquitectura Virtual

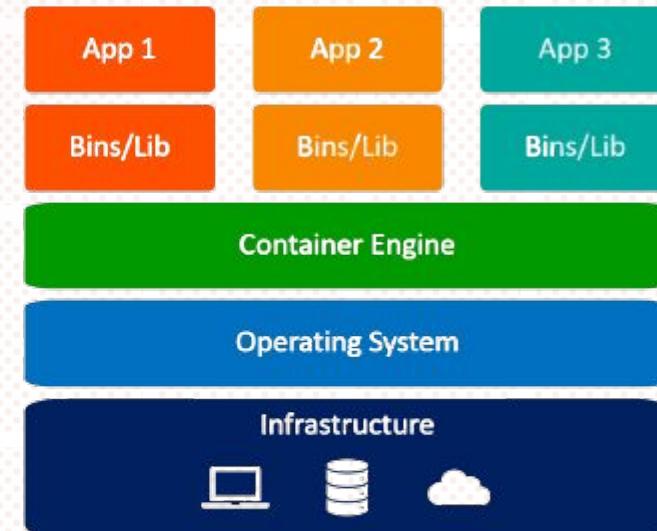
# Definir contenedores



# Máquinas virtuales vs Contenedores

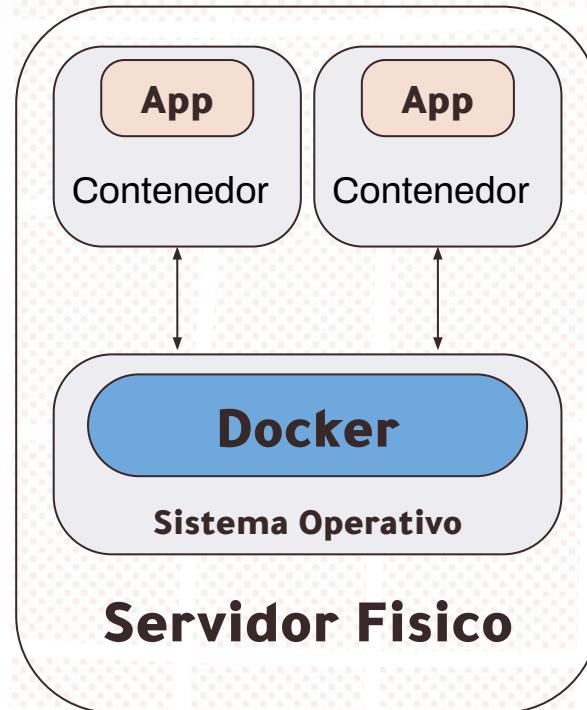
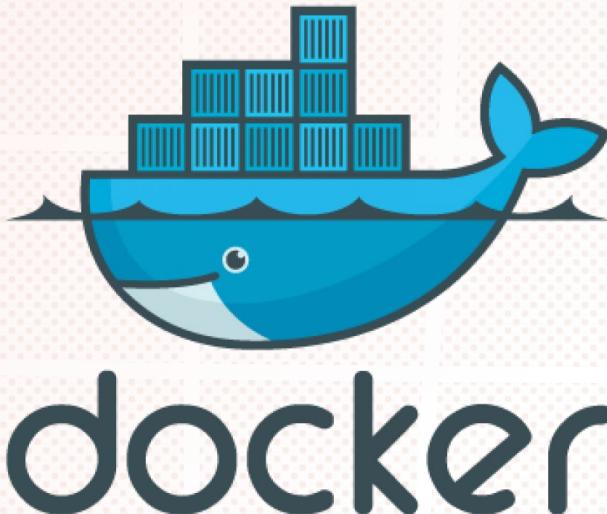


Máquinas virtuales

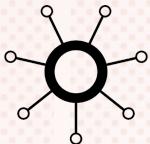


Contenedores

# ¿Qué es Docker?



# Componentes de Docker



El daemon de Docker

Administrar todos los contenedores y las imágenes de Docker.



La API de Docker

Interfaz de programación de aplicaciones permite interactuar con el daemon de Docker a través de aplicaciones y scripts

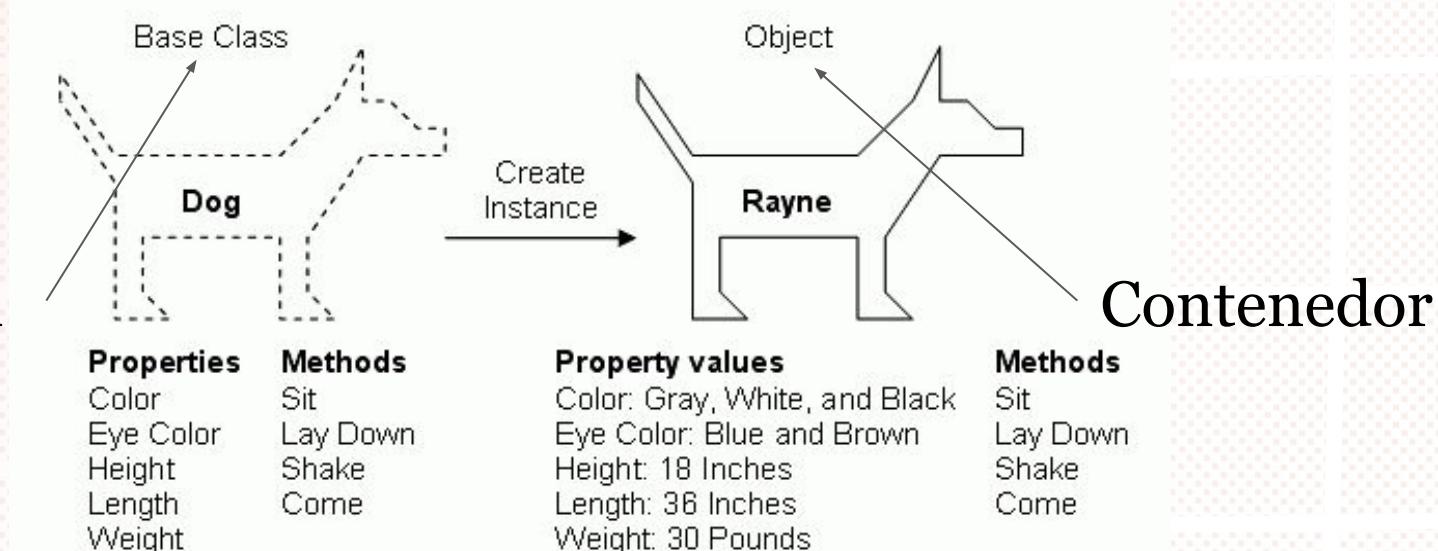


El cliente de Docker

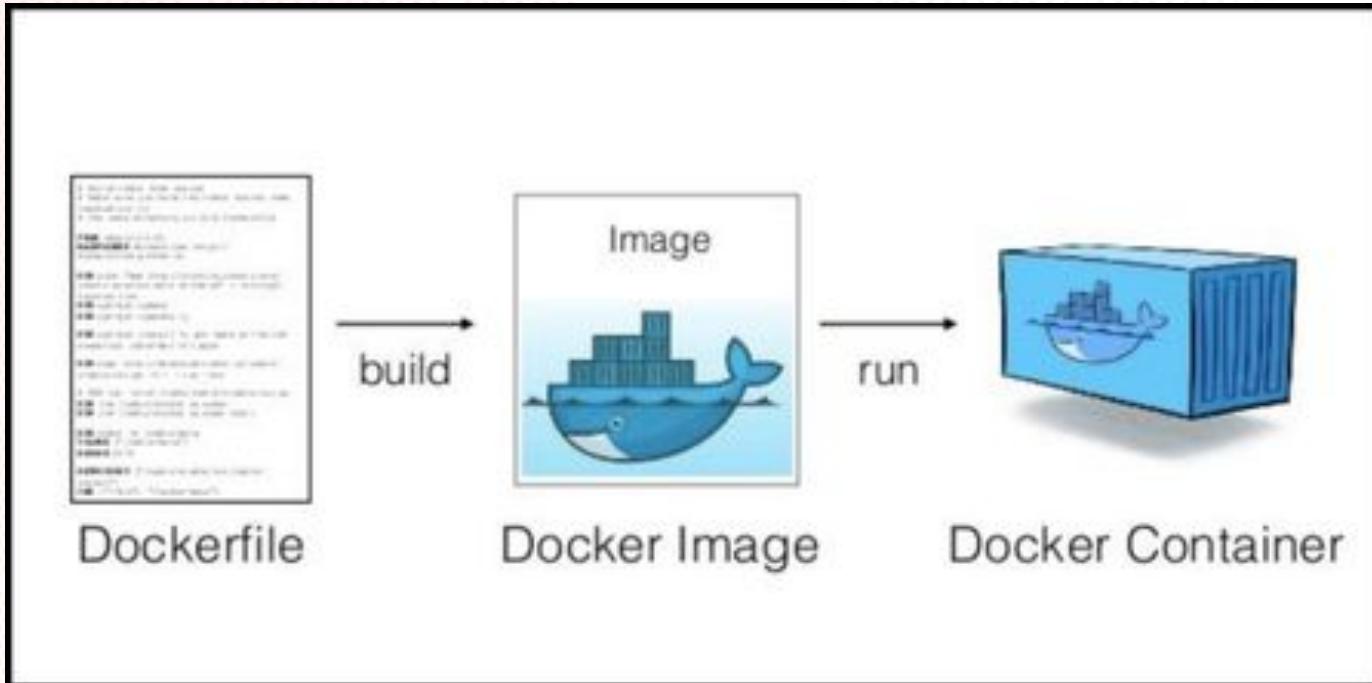
El cliente de Docker envía comandos a la API de Docker y recibe respuestas del daemon de Docker

# Entendiendo el proceso de creación de contenedores

Imagen



# Creación de un contenedor Docker



# Dockerfile

Creamos nuestro build context

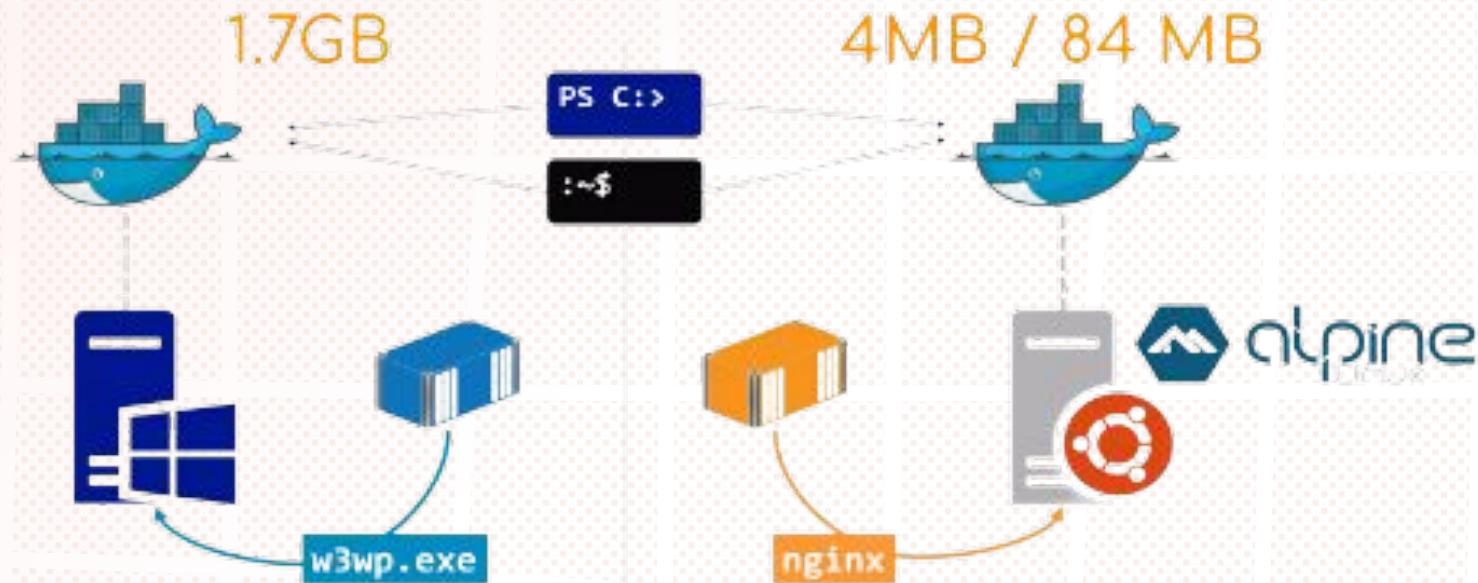
```
mkdir -p ~/Sites/hello-world  
cd ~/Sites/hello-world  
echo "hello" > hello
```

Dentro de este directorio crearemos un archivo llamado Dockerfile con este contenido:

```
FROM busybox  
COPY /hello /  
RUN cat /hello
```

Para conocer todas las directivas visita la [documentación oficial de Dockerfile](#).

# Imágenes de Docker



# Trabajando con Docker

# Instalación de Docker



# Comandos básicos de Docker

Aquí te presento algunos de los comandos básicos de Docker que te permitirán empezar a trabajar con contenedores:

`docker run`: Crea y ejecuta un contenedor a partir de una imagen Docker.

Ejemplo: `docker run hello-world`

`docker pull`: Descarga una imagen Docker desde un registro de imágenes.

Ejemplo: `docker pull nginx`

`docker ps`: Lista los contenedores en ejecución.

Ejemplo: `docker ps`

`docker images`: Lista las imágenes de Docker que se encuentran en el sistema.

Ejemplo: `docker images`

# Comandos básicos de Docker

docker stop: Detiene un contenedor en ejecución.

Ejemplo: docker stop [ID del contenedor]

docker rm: Elimina un contenedor.

Ejemplo: docker rm [ID del contenedor]

docker rmi: Elimina una imagen de Docker.

Ejemplo: docker rmi [ID de la imagen]

docker exec: Ejecuta un comando dentro de un contenedor en ejecución.

Ejemplo: docker exec [ID del contenedor] ls -l

docker logs: Muestra los registros de un contenedor en ejecución.

Ejemplo: docker logs [ID del contenedor]

docker build: Construye una imagen Docker a partir de un archivo Dockerfile.

Ejemplo: docker build -t nombre\_de\_la\_imagen .

# Pasos para crear y ejecutar un contenedor en Docker

1.



Obtén una imagen de Docker

2.



Crea un contenedor a partir de la imagen

3.



Verifica que el contenedor esté en ejecución

4.



Interactúa con el contenedor

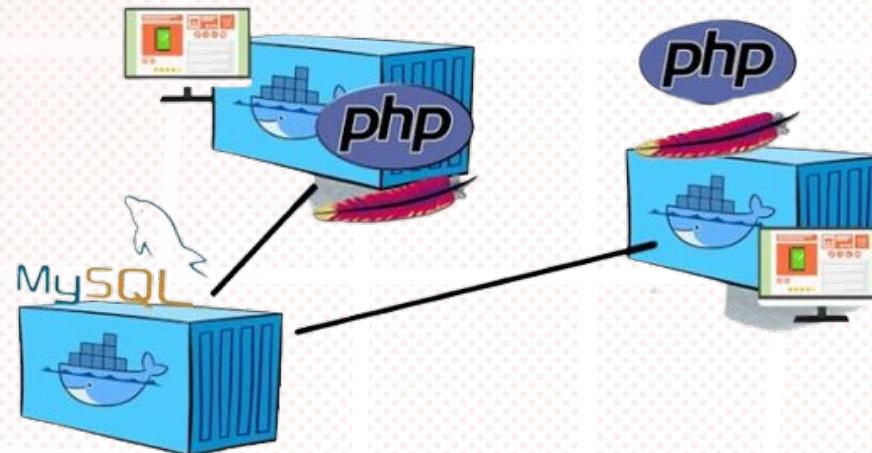
5.



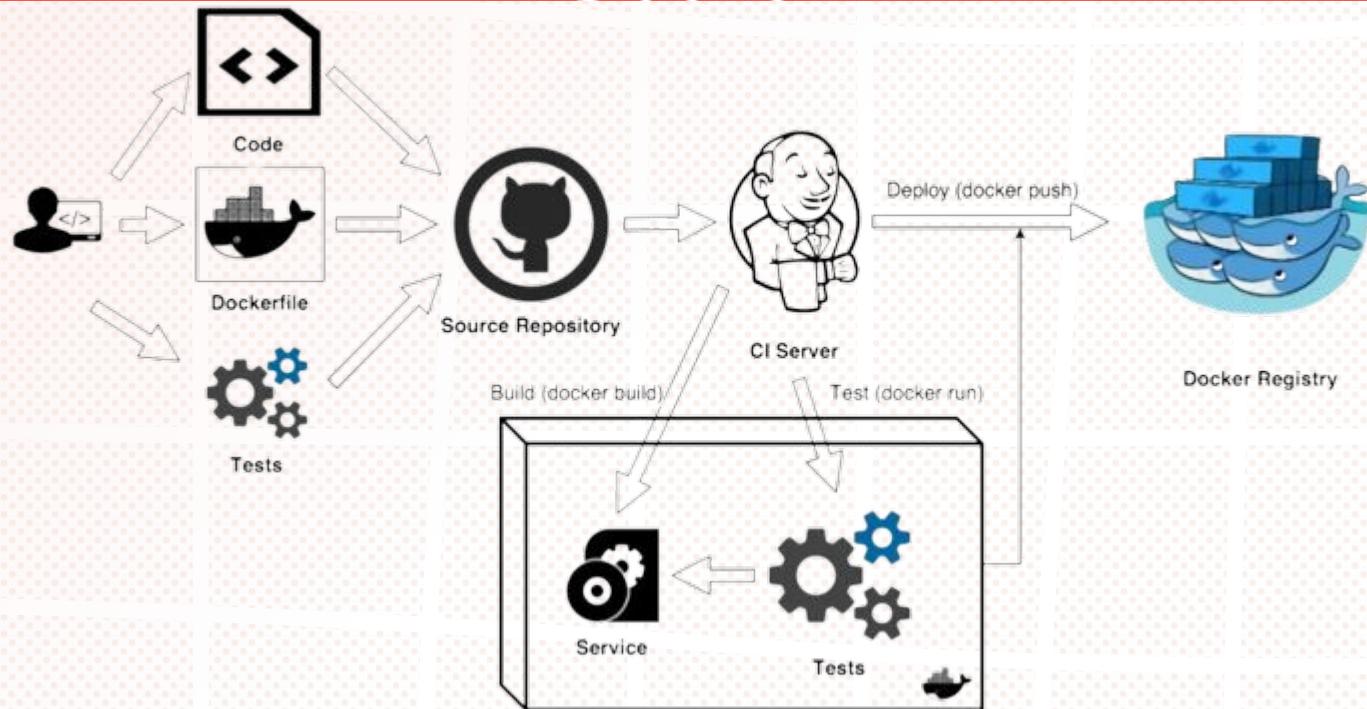
Detén y elimina el contenedor

# Utilizando Docker para Desarrollo

# Integración de Docker con entornos de desarrollo



# Construcción y prueba de aplicaciones con Docker



# Escalabilidad y orquestación de contenedores



## Kubernetes Orquestación de Contenedores



Planificación Inteligente



Self-healing



Escalamiento Horizontal



Descubrimiento & Balanceo de Carga



Rollouts & Rollbacks Automatizados



Gestión de Secretos & Configuración

# Despliegue de Aplicaciones con Docker

# Empaquetamiento de aplicaciones



# **Despliegue de aplicaciones con Docker en servidores locales y en la nube**

# Flujo de trabajo de DevOps en Docker

