

# Docker

# Índice

- 01 [Docker](#)
- 02 [Virtualización](#)
- 03 [Imágenes](#)
- 04 [Contenedores](#)
- 05 [Ejemplo en vivo](#)



01

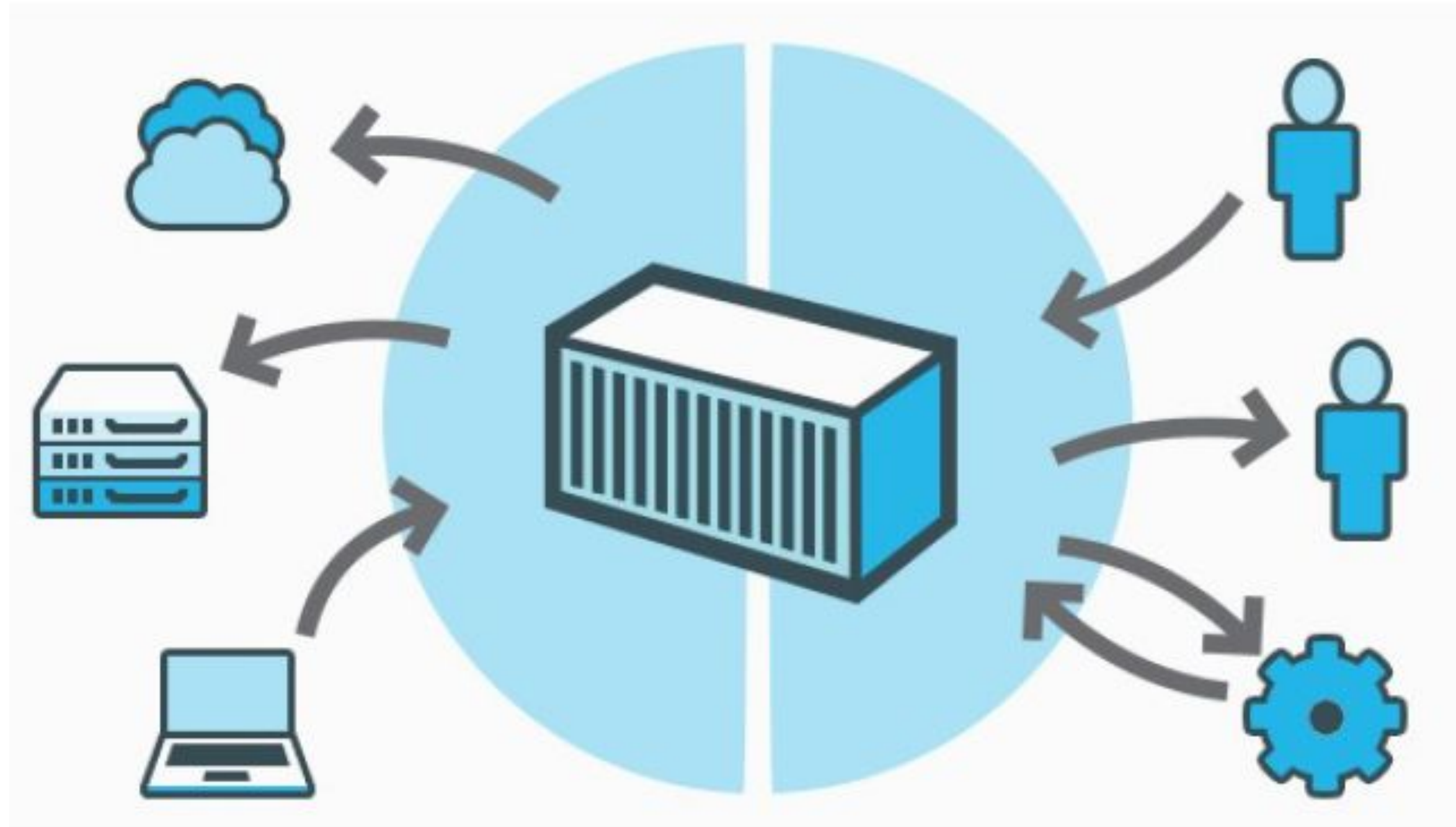
# Docker

# Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de **contenedores** que permiten empaquetar, distribuir y ejecutar servicios de red con un **formato estándar** con todas las **dependencias incluidas**.



# La arquitectura de Docker



# Tipos de aplicaciones



## Servicios de red

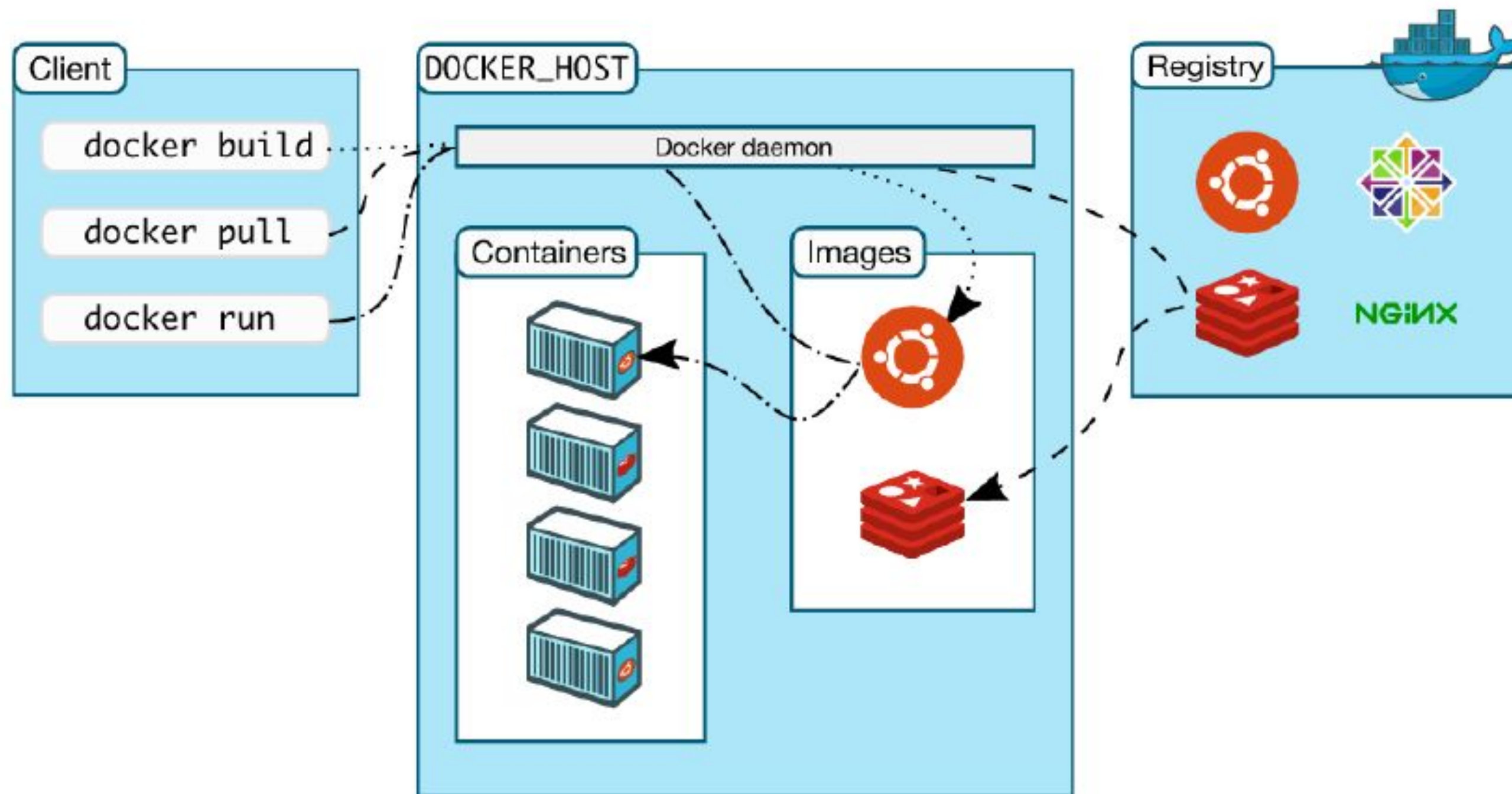
- Web
- Bases de datos
- Colas de mensajes
- Cache
- Etc.



## Línea de comandos

- Compiladores
- Conversores de video
- Generadores de informes

# Flujo de Docker



**Docker Engine** es el servicio local utilizado para gestionar **Docker**



Gestiona las imágenes.  
Descarga, creación, subida, etc.



Gestiona los contenedores:  
Inicio, freno, reinicio.



Se utiliza desde el cliente Docker por línea de comandos o a través de una API REST.



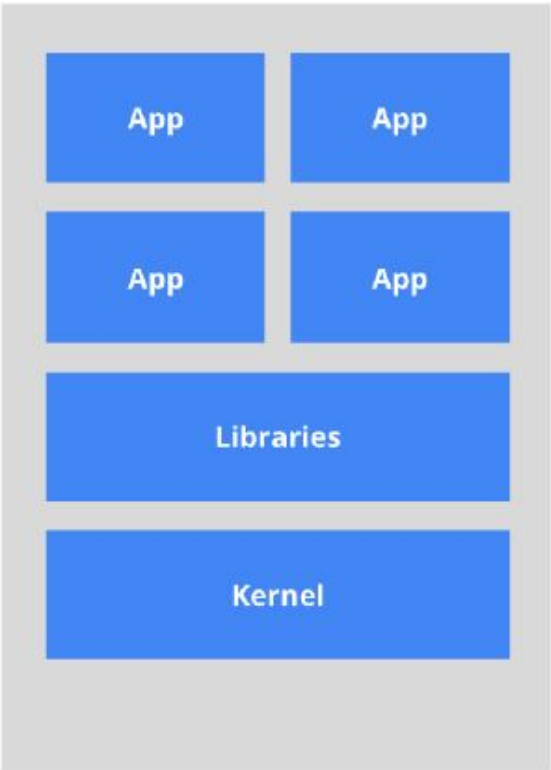
02

# Virtualización

# Virtualización

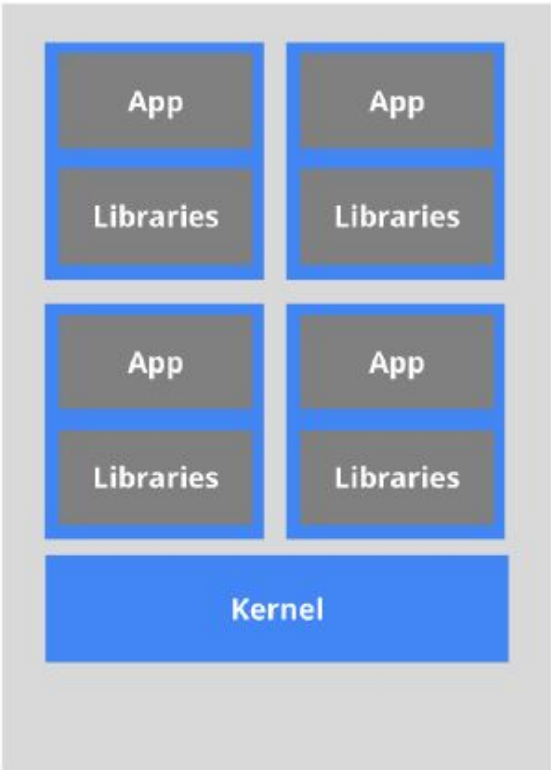
## Docker

The old way: Applications on host

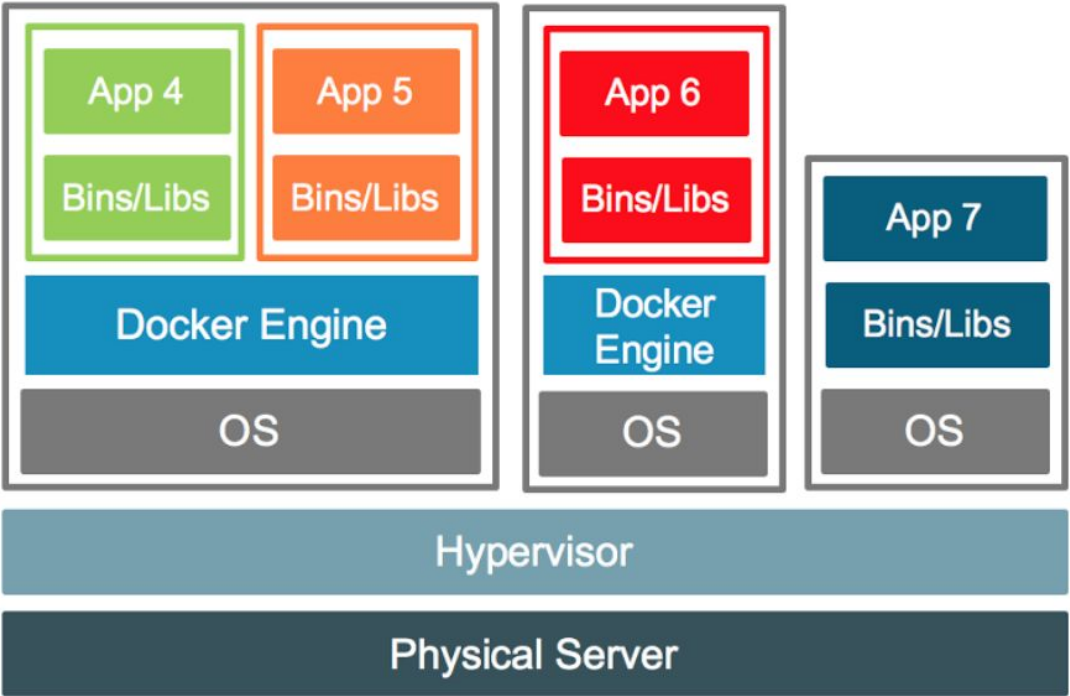
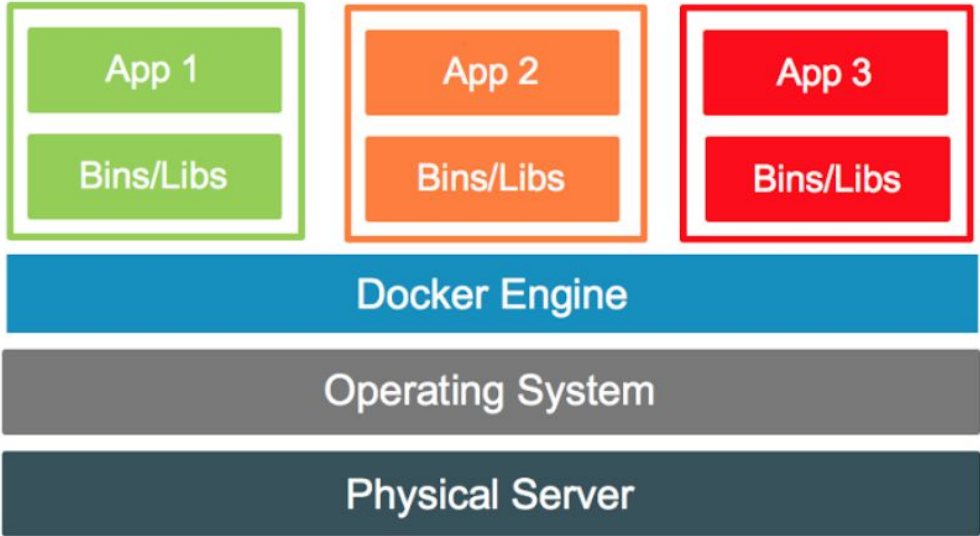


Heavyweight, non-portable  
Relies on OS package manager

The new way: Deploy containers



Small and fast, portable  
Uses OS-level virtualization



03

# Imágenes

# Docker images

Una imagen es una **representación estática** de la aplicación o servicio y de su configuración y las dependencias.

# Descarga una imagen

Podemos descargar una imagen de **Docker Hub** usando **pull**.

Si ya tenemos imagen en nuestra máquina, el comando actualizará automáticamente la imagen a la última versión.

En el ejemplo de la derecha estamos descargando la última versión de la base de datos MySQL.

```
# docker pull mysql
```

# Imágenes más populares

Hay muchas imágenes seleccionadas y optimizadas disponibles en Docker Hub. La popularidad de estas imágenes depende de varios factores: presencia en el mercado, calificaciones, puntajes de satisfacción, entre otros. Para obtener una lista detallada de los repositorios más populares, se puede navegar hasta el Docker Hub sitio web. El uso de una imagen también depende del sistema operativo y su arquitectura.



<https://hub.docker.com/search>

04

# Contenedores

# Contenedores

Al tener **Docker** instalado sobre el sistema operativo que estamos trabajando contaremos con:

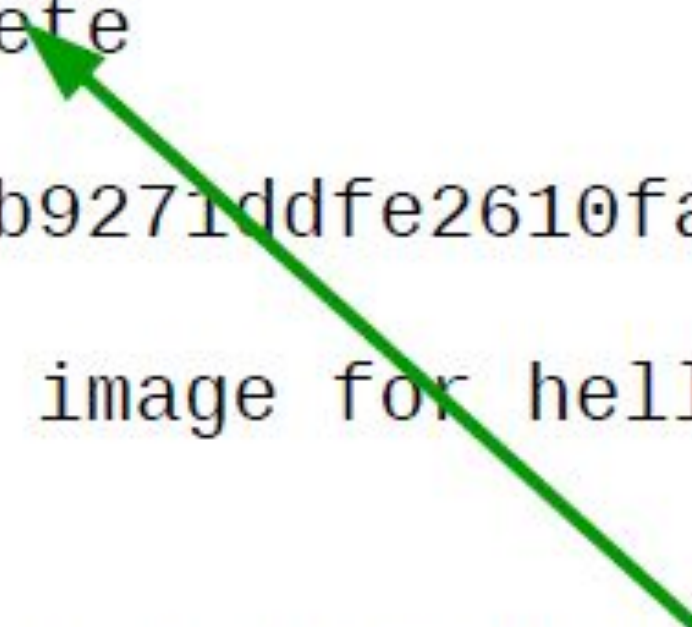
- Aplicaciones empaquetadas con todas sus dependencias.
- Ejecutar aplicaciones en cualquier entorno (Windows, Linux, Mac).
- Descarga de imágenes de forma automática si no se encuentran disponibles.



# Hello world con Docker

```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
03f4658f8b78: Pull complete
a3ed95caeb02: Pull complete
Digest:
sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc
1ca36966a7
Status: Downloaded newer image for hello-world:latest

Hello from Docker.
This message shows that your installation
working correctly.
...
```

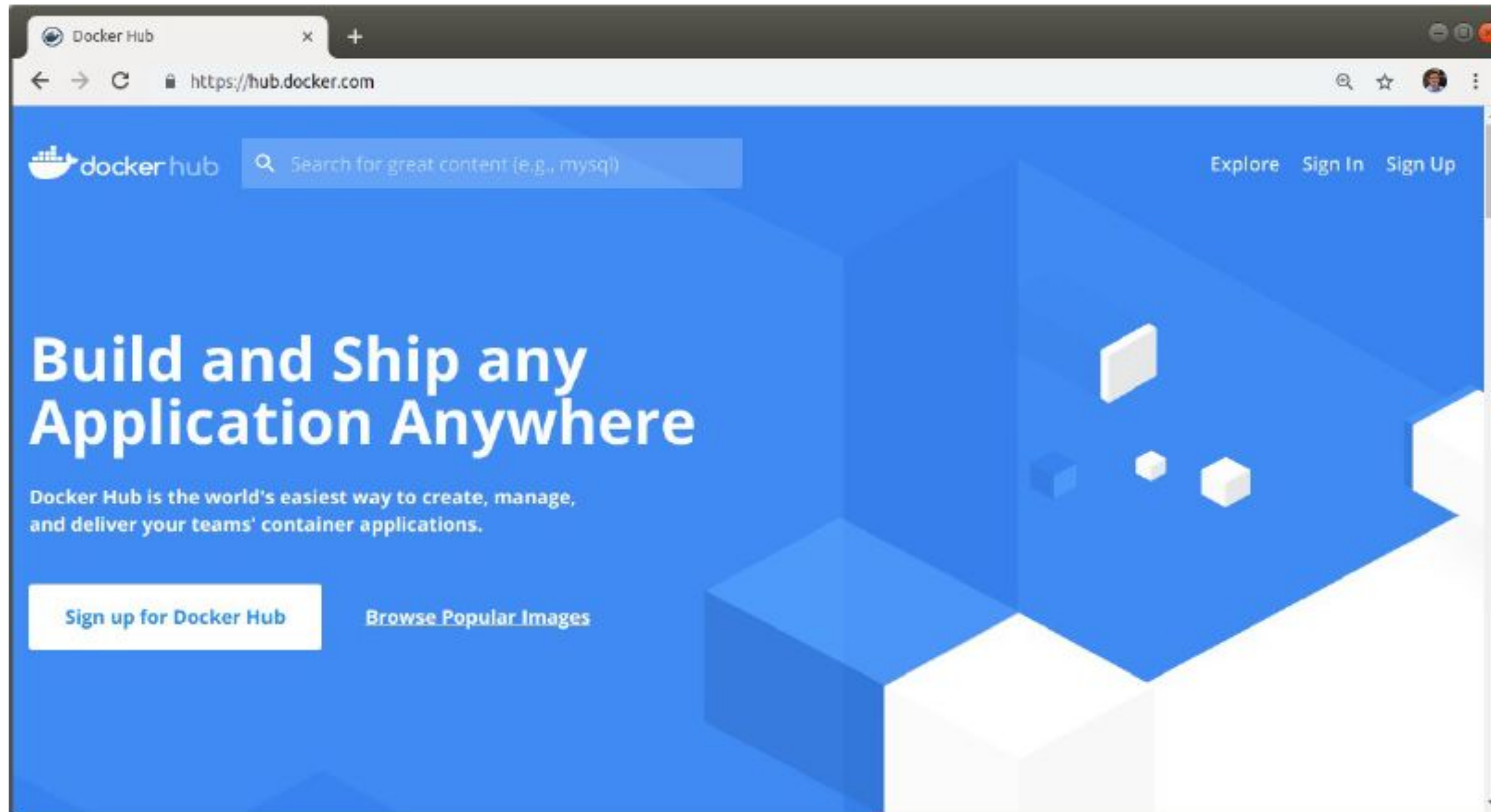


La primera vez la  
imagen se descarga

05

# Ejemplo en vivo

# Creamos una cuenta en Docker Hub



# Servidor web estático

En el siguiente ejemplo vamos a descargar una imagen de un servidor web estático y crear un contenedor.



<https://hub.docker.com/r/sequence/static-site>

# Creamos el contenedor

```
docker run --name static-site -e Author="Tu nombre" -d -p 9000:80 sequence/static-site
```

01

**--name**

El nombre que le damos al contenedor.

02

**-e**

Variables de entorno que enviamos al contenedor.

03

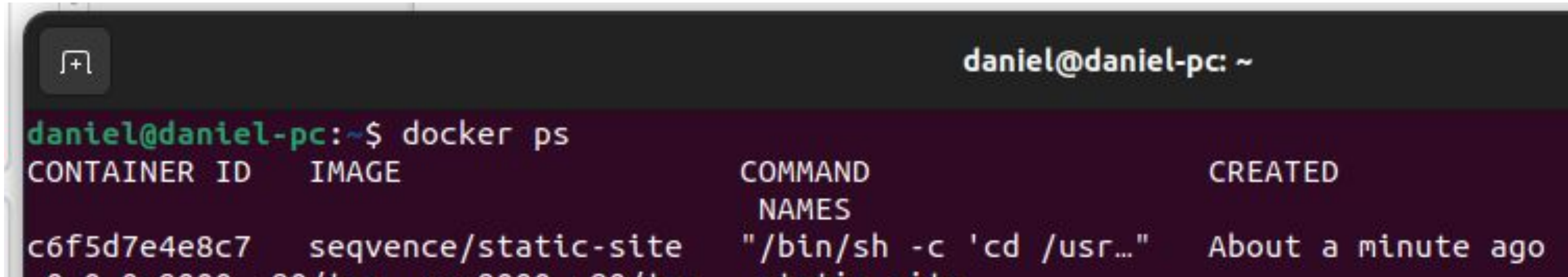
**-p**

Puerto desde el que exponemos la aplicación y el puerto de destino.



# Verificamos el contenedor

Con **docker ps** podemos ver todos los contenedores que se encuentran activos.

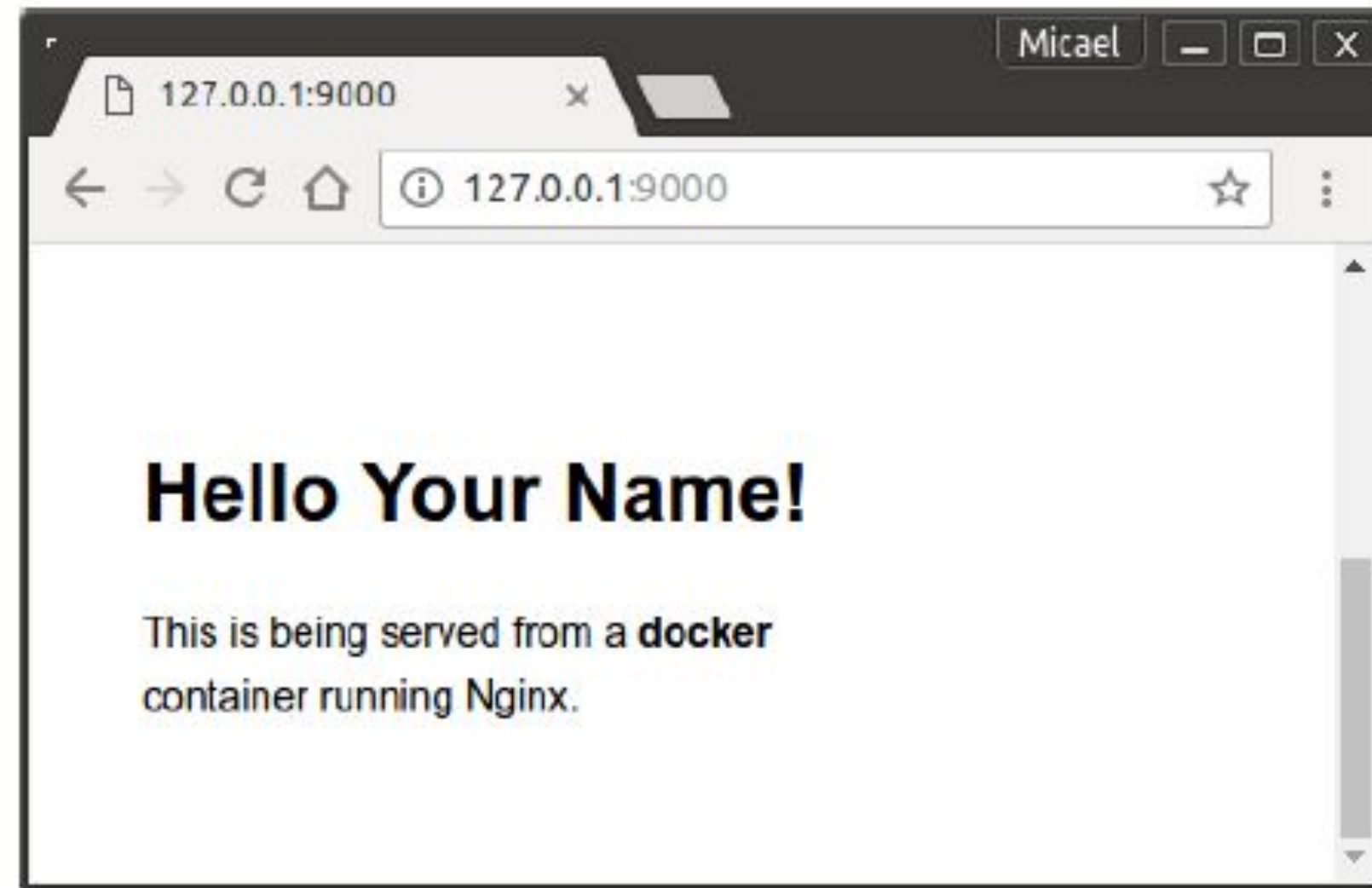


```
daniel@daniel-pc:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND NAMES	CREATED
c6f5d7e4e8c7	sequence/static-site	"/bin/sh -c 'cd /usr..."	About a minute ago

# Creamos una cuenta en Docker Hub

Probamos el servicio desde el navegador, ingresando a **http://127.0.0.1:9000**



# Comandos útiles

**docker logs [container\_id]** Muestra los logs que arroja la aplicación.

**docker stop [container\_id]** Detiene tu contenedor.

**docker restart [container\_id]** Reinicia tu contenedor.

**docker inspect [container\_id]** Observa toda la info en un contenedor.

**docker port [container\_id]** Muestra el puerto público de un contenedor.

**docker exec** Permite ejecutar comandos dentro del contenedor.



¡Muchas gracias!