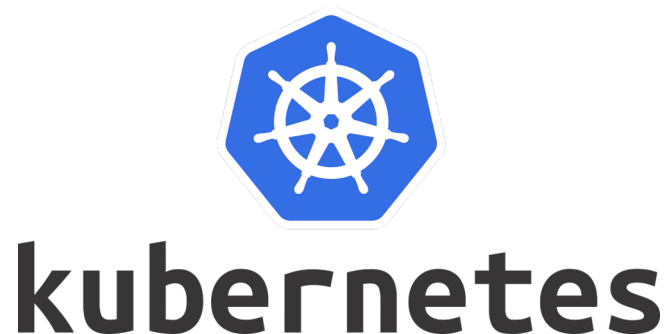
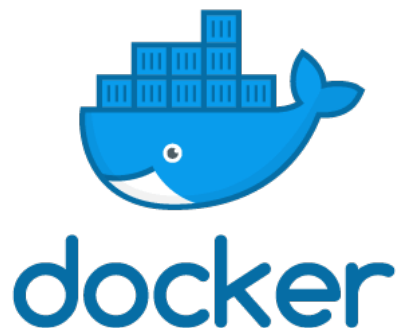


Seminarios OfiLibre

Docker y Kubernetes



Micael Gallego
@micael_gallego

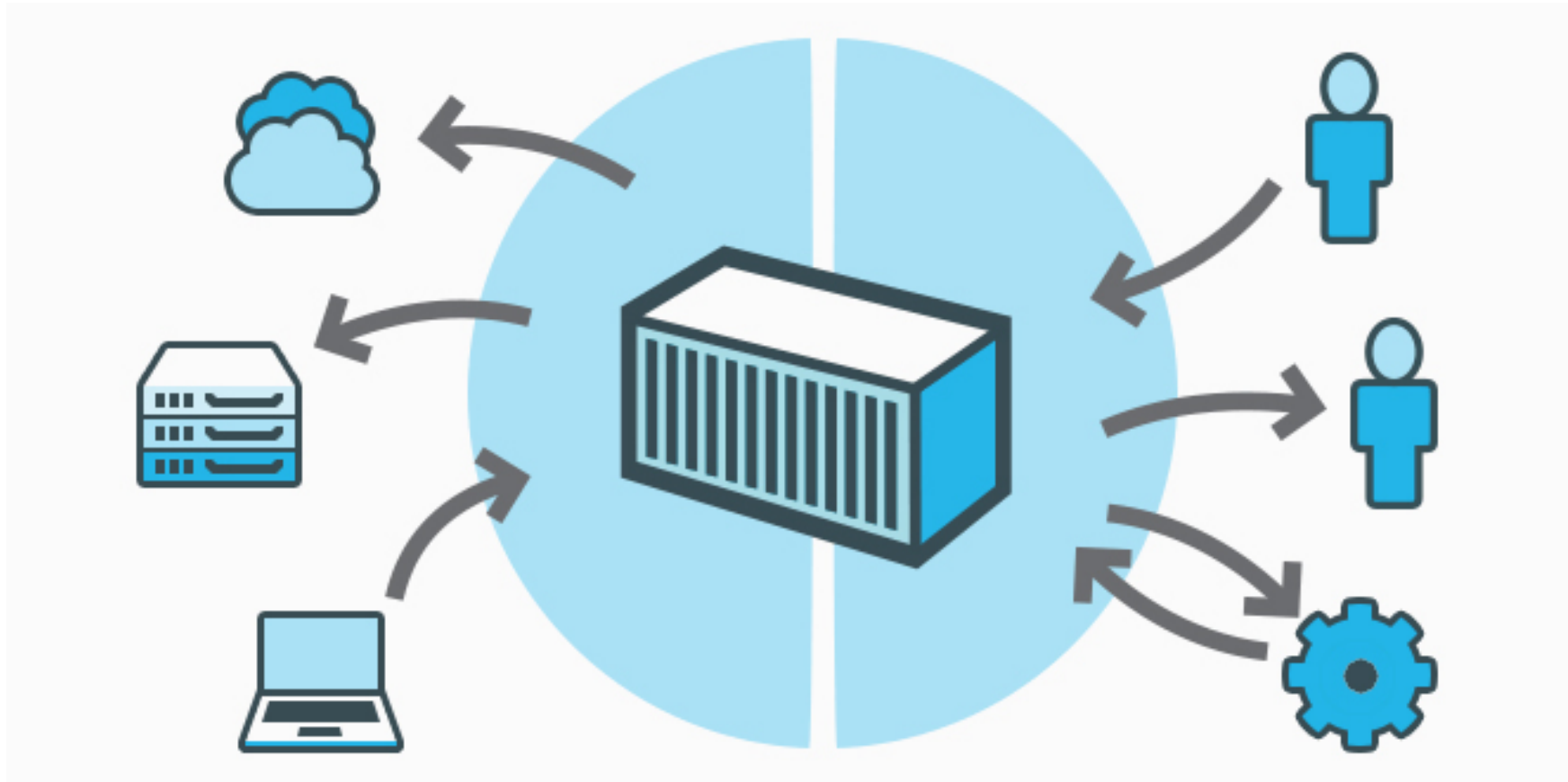
Pablo Chico de Guzmán
@pchico83

Docker



Los **contenedores Docker** permiten empaquetar, distribuir y ejecutar servicios de red, con un formato **estándar**

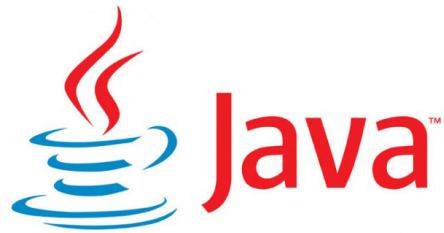
Docker



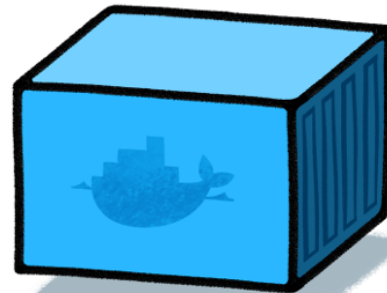
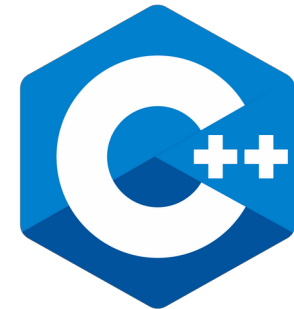
Docker



Docker



Go



Docker



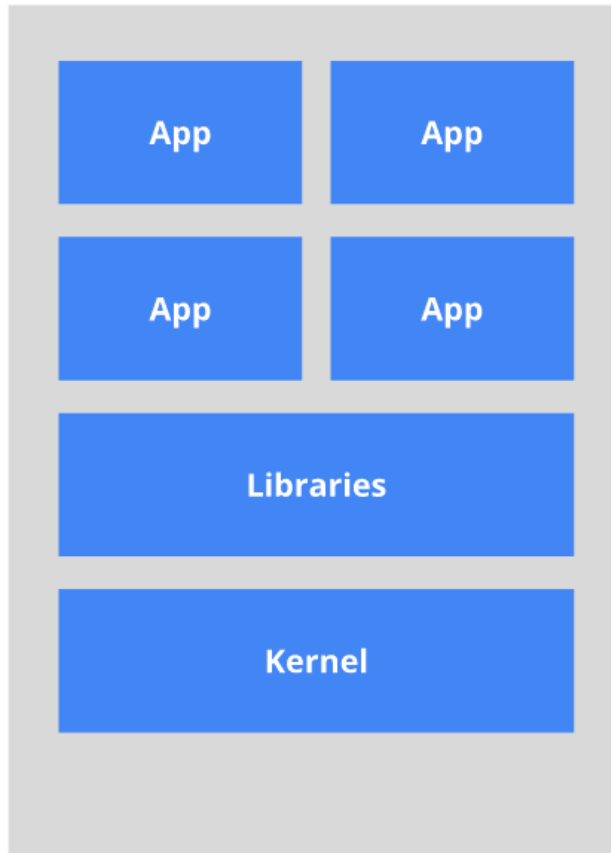
- Es la tecnología de contenedores **más popular** (aunque existen otras tecnologías de contenedores)
- Muy utilizada en sistemas **linux**, aunque dispone de herramientas para **desarrolladores en windows y mac**
- Con **repositorio de imágenes** (DockerHub) con imágenes públicas de contenedores
- Creada en **2013**

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

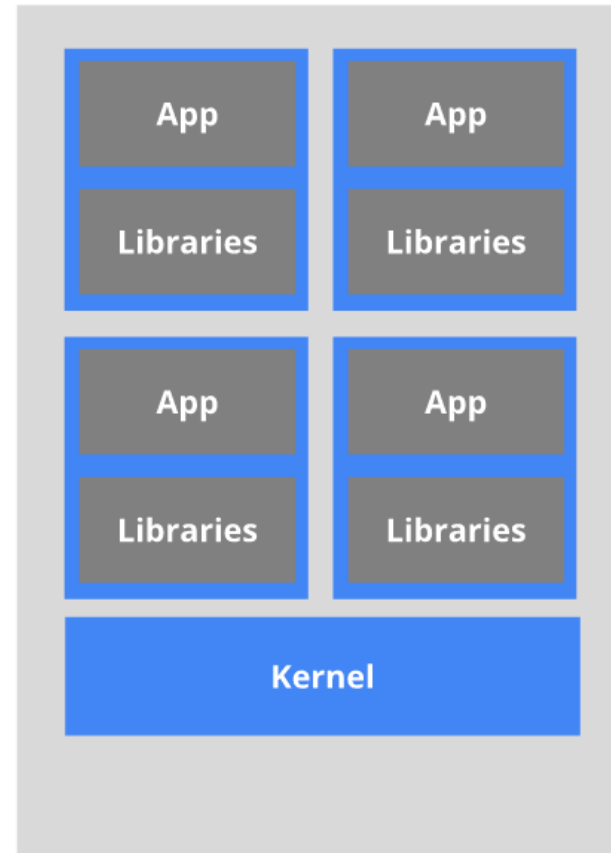
- ¿Qué son los contenedores Docker?
 - Son aplicaciones empaquetadas con **todas sus dependencias**
 - Se pueden ejecutar en **cualquier sistema operativo**
 - En linux de forma **óptima**
 - En windows y mac con **virtualización ligera**
 - Se **descargan de forma automática** si no están disponibles en el sistema
 - Por defecto están aisladas del host (mayor seguridad)

Docker

Aplicaciones Linux tradicionales

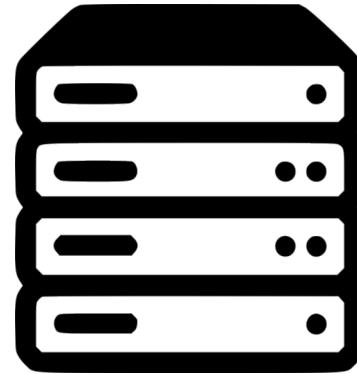


Aplicaciones Linux en contenedores Docker



Docker

- **Tipos de aplicaciones:**



- **Aplicaciones de red:**

- Web, bbdd, colas de mensajes, cachés, etc.

- **Aplicaciones de línea de comandos:**

- Compiladores, generadores de sitios web, conversores de vídeo, generadores de informes...

Docker

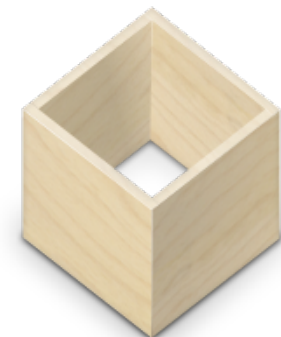
- Tipos de aplicaciones:

- Aplicaciones **gráficas**:

- Es posible pero no está diseñado para ello
- Alternativas en linux



<https://snapcraft.io/>



FLATPAK

<https://flatpak.org/>

¿Cómo funciona Docker?

- Ejecuta procesos linux de forma aislada del resto con **namespaces** y **cgroups**
- Permiten definir un sistema de ficheros “virtual” específico para el proceso
- Cuando se ejecuta el servicio tiene un entorno similar al que tendría si estuviera en su **propia máquina**
- Se tienen ventajas similares a las **máquinas virtuales** pero de forma mucho **más eficiente**

¿Cómo funciona Docker?

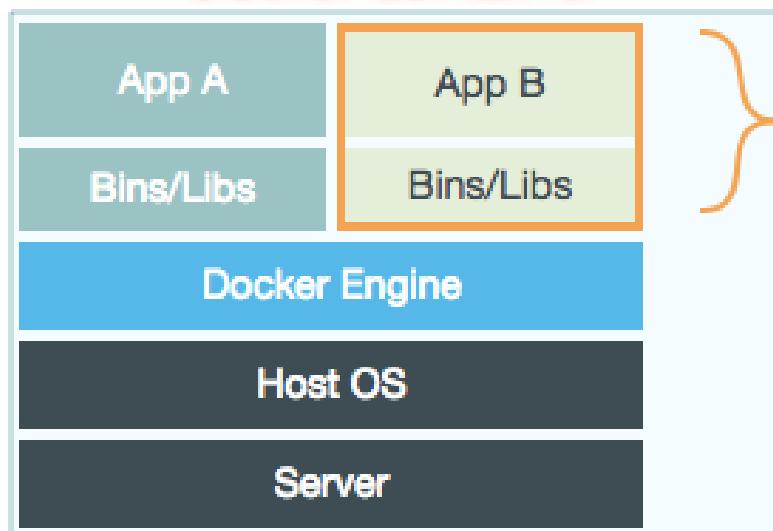
- Los **contenedores** son una tecnología que ofrece unas **ventajas similares** a las VMs pero **aprovechando** mejor los recursos:
 - Los contenedores tardan **milisegundos** en arrancar
 - Consumen únicamente la **memoria** que necesita la app ejecutada en el contenedor.
 - Una VMs **reserva la memoria completa** y es usada por el sistema operativo huésped (*guest*) y la aplicación

¿Cómo funciona Docker?

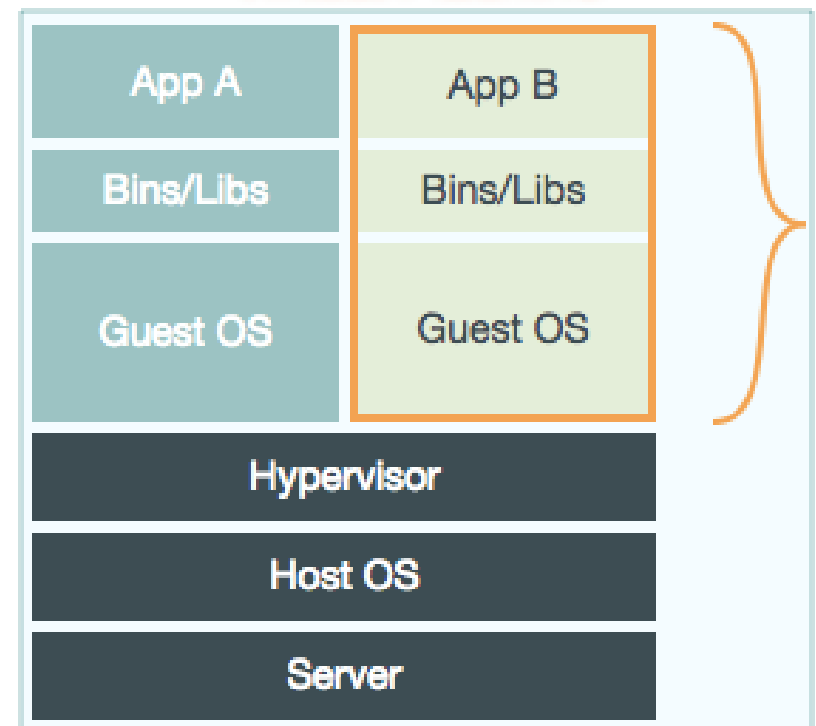
- ¿Por qué son tan eficientes los contenedores?
 - Para ejecutar un contenedor no se necesita **hypervisor** porque **no se ejecuta** un sistema operativo invitado y no hay que simular HW
 - Un contenedor es un **paquete** que contiene una **app** y todo el sw necesario para que se ejecute (python, Java, gcc, libs....)
 - El contenedor es ejecutado directamente por el **kernel del sistema operativo** como si fuera una aplicación normal pero de forma **aislada del resto**

¿Cómo funciona Docker?

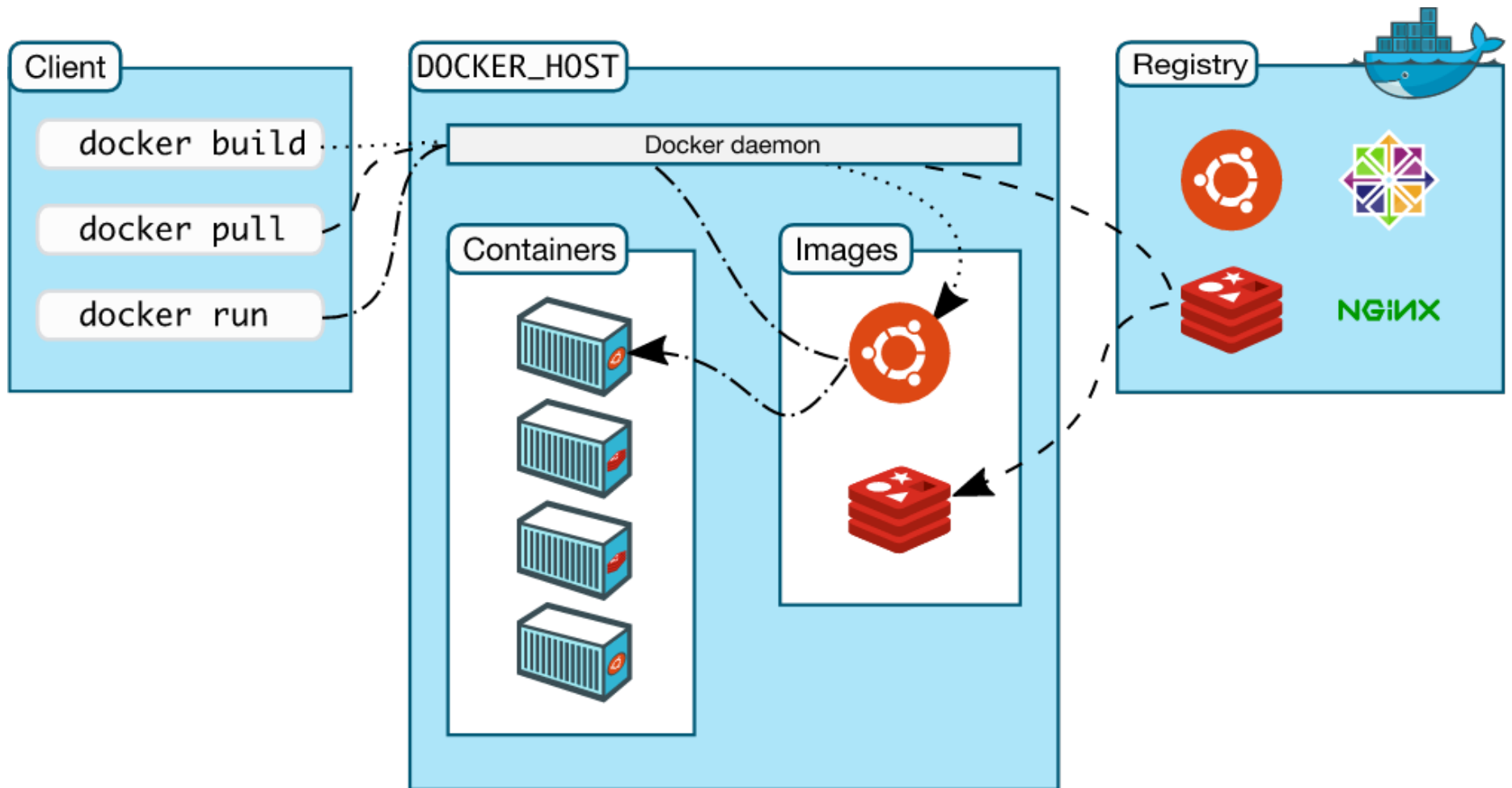
Docker Container



Virtual Machine



Conceptos Docker



Conceptos Docker

- **Imagen docker**
 - Ficheros a los que tendrá acceso el contenedor cuando se ejecute.
 - Herramientas/librerías de una distribución linux menos el kernel (**ubuntu, alpine**)
 - Runtime de ejecución (**Java**)
 - La aplicación en sí (**webapp.jar**)

Conceptos Docker

- **Imagen docker**
 - Un contenedor siempre se inicia desde una **imagen**
 - Si se quiere arrancar un contenedor partiendo de una imagen que no está disponible, se **descarga automáticamente** de Internet

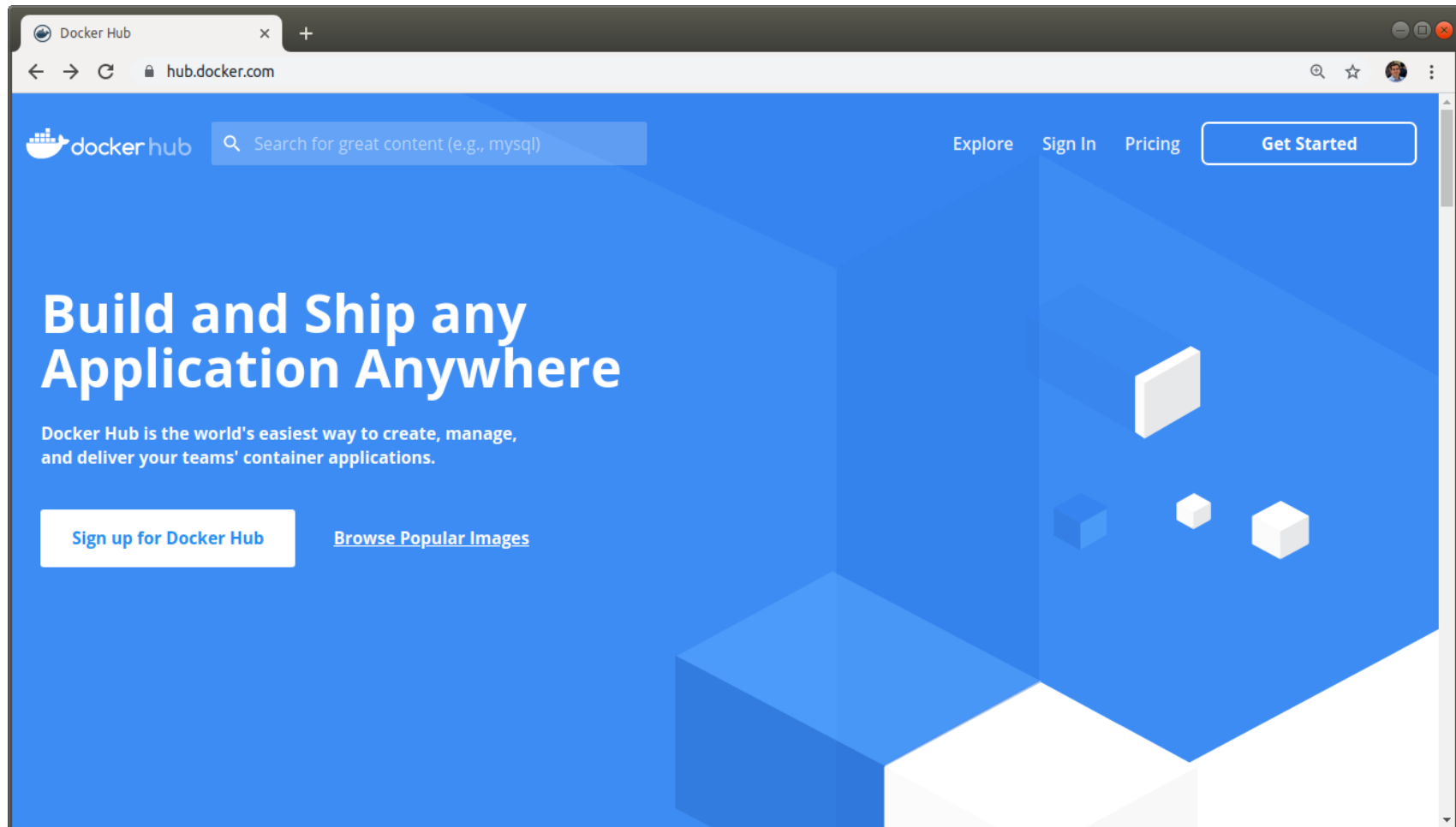
Conceptos Docker

• Docker Registry

- **Servicio remoto** para subir y descargar imágenes
- Puede guardar varias “versiones” (**tags**) de la misma imagen
- Las diferentes versiones de una misma imagen se almacenan en un **repositorio (mysql, drupal...)**
- **Docker Hub** es un registro público y gratuito gestionado por Docker Inc.
- Puedes instalar un **registro privado**

Conceptos Docker

- Docker Hub



Conceptos Docker

- Docker Hub: Algunos repositorios oficiales



ubuntu  The Official Ubuntu base image



WordPress is a free and open source blogging tool and a content management system



Popular open-source relational database management system



Document-oriented NoSQL database



Official CentOS base image



High performance reverse proxy server

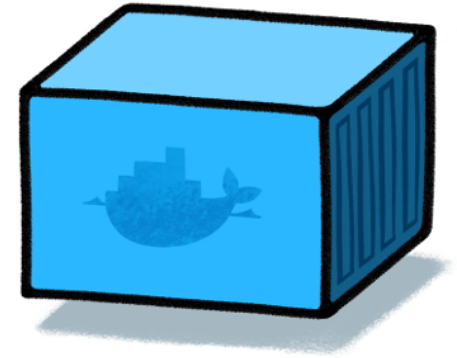


Relational database management system



Node.js is a platform for scalable server-side and networking applications

Conceptos Docker



- **Contenedor Docker**
 - Representa la **aplicación en ejecución**
 - Un contenedor se crea desde **una imagen**
 - Si la aplicación escribe un fichero, el fichero queda dentro del contenedor, no se **modifica la imagen**
 - Los contenedores se pueden **arrancar, pausar y parar**
 - Puede haber **varios contenedores ejecutandose** a la vez partiendo desde la **misma imagen**

Conceptos Docker

- **Docker Engine**

- **Proceso encargado** de gestionar docker
- Gestiona las **imágenes** (descarga, creación, subida, etc...)
- Gestiona los **contenedores** (arranque, parada, etc..)
- Habitualmente se controla desde el **cliente docker** por **línea de comandos** (aunque también se puede controlar por **API REST**)

Conceptos Docker

- Docker client
- Herramienta por línea de comandos (*Command line interface*, CLI) para controlar las imágenes y los contenedores

```
$ docker <params>
```

Ejecución de contenedores

Ejecutar “hello-world” en un contendor

```
$ 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:
sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc1ca369
66a7
Status: Downloaded newer image for hello-world:latest

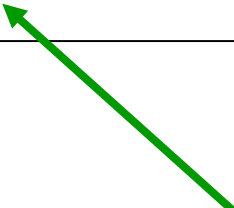
Hello from Docker.
This message shows that your installation a
correctly.
...
```

La primera vez la imagen se descarga

Ejecución de contenedores

Ejecutar “hello-world” por segunda vez

```
$ docker run hello-world  
Hello from Docker.  
This message shows that your installation appears to be working  
correctly.  
...
```



La segunda vez se usa
la vez la imagen se
descarga

Ejecución de contenedores

Inspeccionar los contenedores existentes

```
$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a6a9d46d0b2f	alpine	"echo 'hello'"	6 minutes ago	Exited (0) 6 minutes ago		lonely_kilby
ff0a5c3750b9	alpine	"ls -l"	8 minutes ago	Exited (0) 8 minutes ago		elated_ramanujan
c317d0a9e3d2	hello-world	"/hello"	34 seconds ago	Exited (0) 34 seconds ago		stupefied_mcclintock

Muestra los contenedores del sistema.
 Todos ellos tienen el estado STATUS Exited. Estos
 contenedores no se están ejecutando
 (pero consumen espacio en disco)

Ejecución de contenedores

Ejecutar “hello-world” en un contendor

```
$ 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:
sha256:8be990ef2aeb16dbcb9271ddfe2610fa6658d13f6dfb8bc72074cc1ca369
66a7
Status: Downloaded newer image for hello-world:latest

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

La primera vez la imagen se descarga

Servicios de red

● Servidor web en un contenedor

```
docker run --name static-site \  
  -e AUTHOR="Your Name" -d \  
  -p 9000:80 sequence/static-site
```

Servicios de red

● Servidor web en un contenedor

```
docker run --name static-site \  
  -e AUTHOR="Your Name" -d \  
  -p 9000:80 sequence/static-site
```

--name static-site

Nombre del contenedor

Servicios de red

● Servidor web en un contenedor

```
docker run --name static-site \  
  -e AUTHOR="Your Name" -d \  
  -p 9000:80 sequence/static-site
```

`-e AUTHOR="Your Name"`

Pasar variables de entorno a la aplicación
que se ejecuta en el contenedor

Servicios de red

● Servidor web en un contenedor

```
docker run --name static-site \  
  -e AUTHOR="Your Name" -d \  
  -p 9000:80 sequence/static-site
```

-d

Ejecuta el contenedor en segundo plano
(no bloquea la shell durante la ejecución)

Servicios de red

● Servidor web en un contenedor

```
docker run --name static-site \  
  -e AUTHOR="Your Name" -d \  
  -p 9000:80 sequence/static-site
```

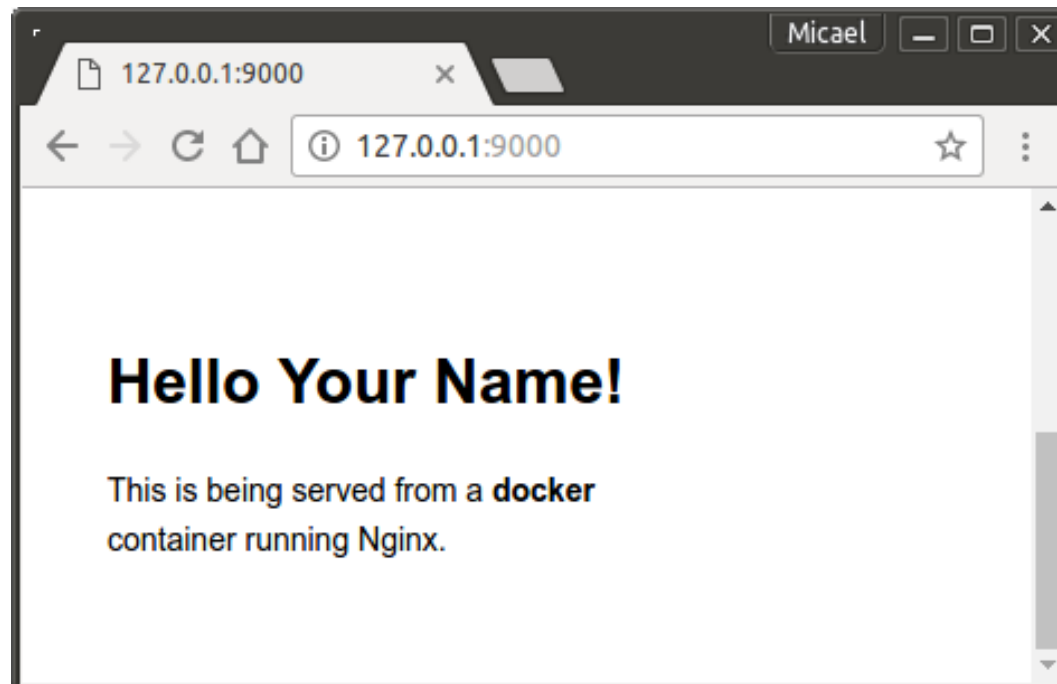
-p 9000:80

Conecta el puerto 9000 del host
al puerto 80 del contenedor

Servicios de red

- Usar el servicio

- Abre la URL <http://127.0.0.1:9000> en un browser accede al puerto 80 de la aplicación en el contenedor





“Dockerizar” una aplicación

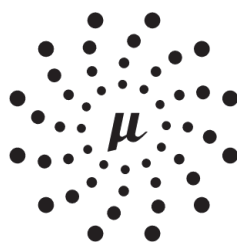
- Integración con tecnologías de desarrollo



QUARKUS



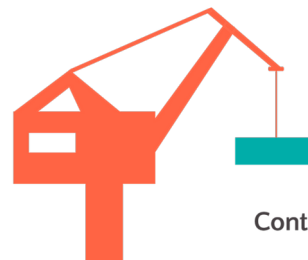
Spring **Boot**



MICRONAUT®



Buildpacks.io



Jib

Containerize your Java application.

Ecosistema de desarrollo

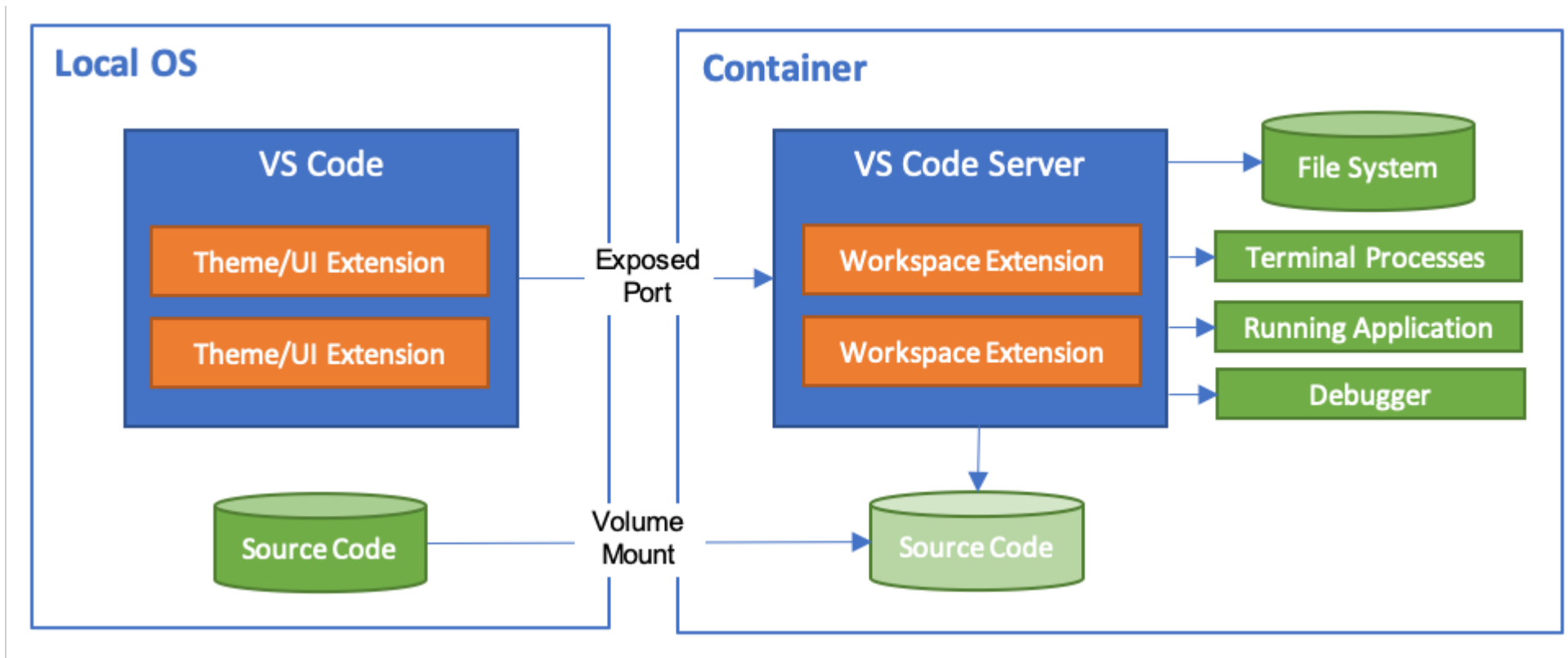


Visual Studio Code Remote - Containers

The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Explorer sidebar on the left shows the project structure for a Rust development container. The Dockerfile in the editor contains the following content:

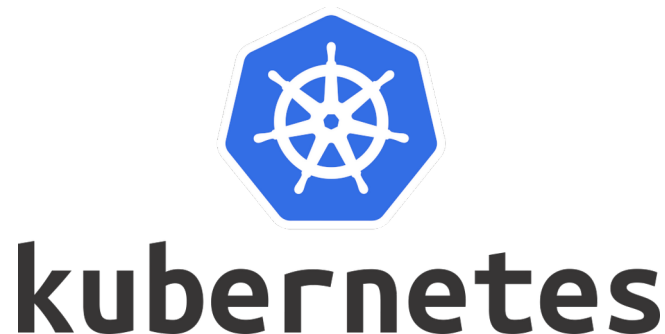
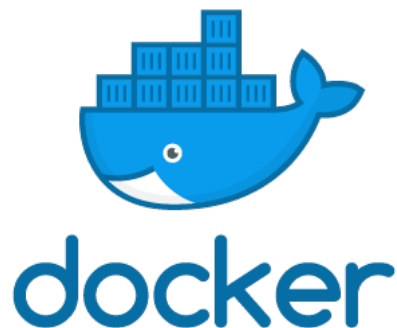
```
Dockerfile — rust [Dev Container: Rust]
1  #-----
2  # Copyright (c) Microsoft Corporation. All rights reserved.
3  # Licensed under the MIT License. See https://go.microsoft.com/fwlink/?linkid=2198365
4  #-----
5
6  FROM rust:1
7
8  # Copy endpoint specific user setting overrides into container
9  COPY settings.vscode.json /root/.vscode-remote/data/Machine
10
11 RUN rustup update
12 RUN rustup component add rls rust-analysis rust-src
13
14 # Install rust-analyzer tools
```

Ecosistema de desarrollo



Seminarios OfiLibre

Docker y Kubernetes



Micael Gallego
@micael_gallego

Pablo Chico de Guzmán
@pchico83