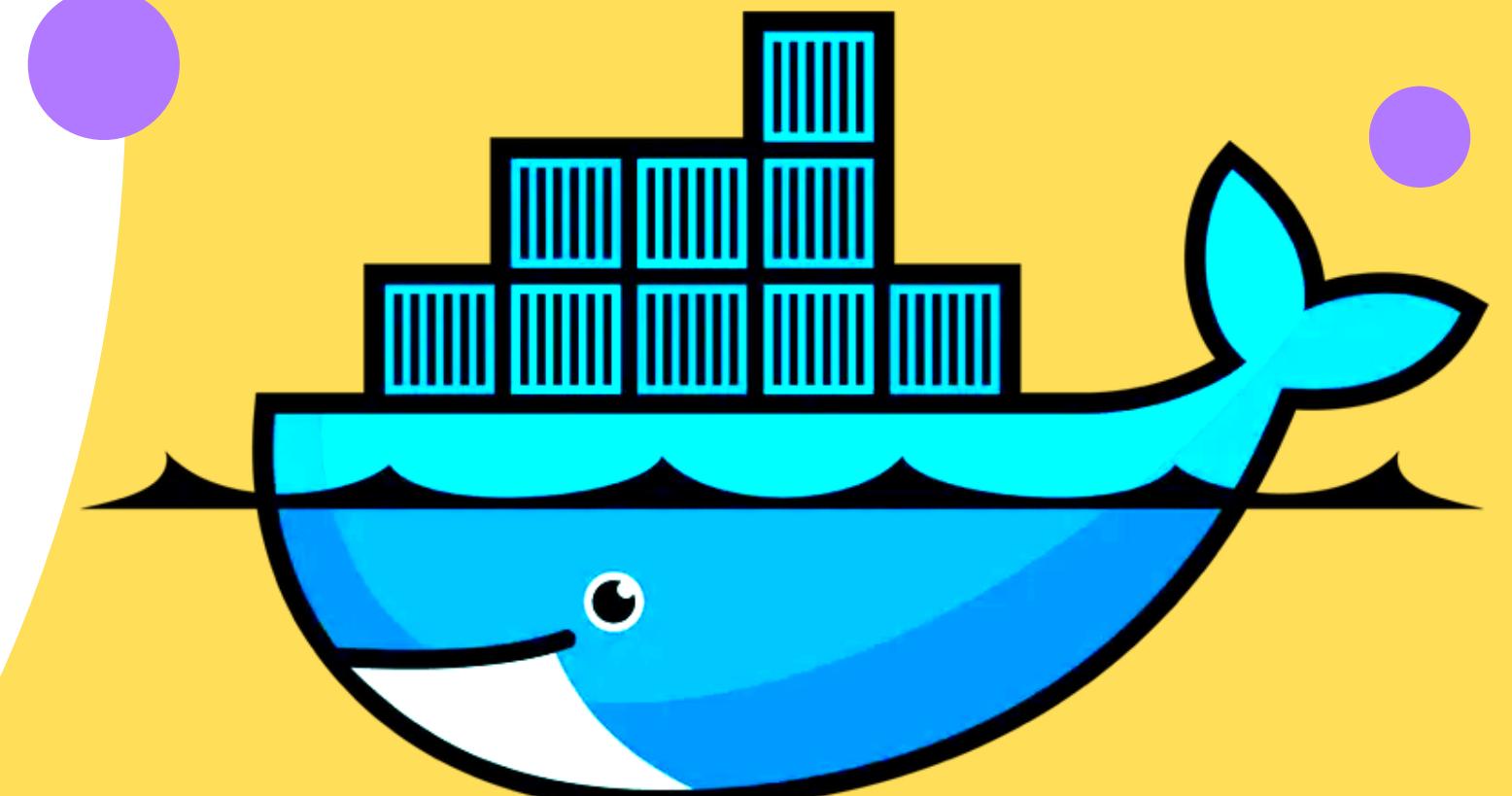


Docker

SW empaquetado



docker

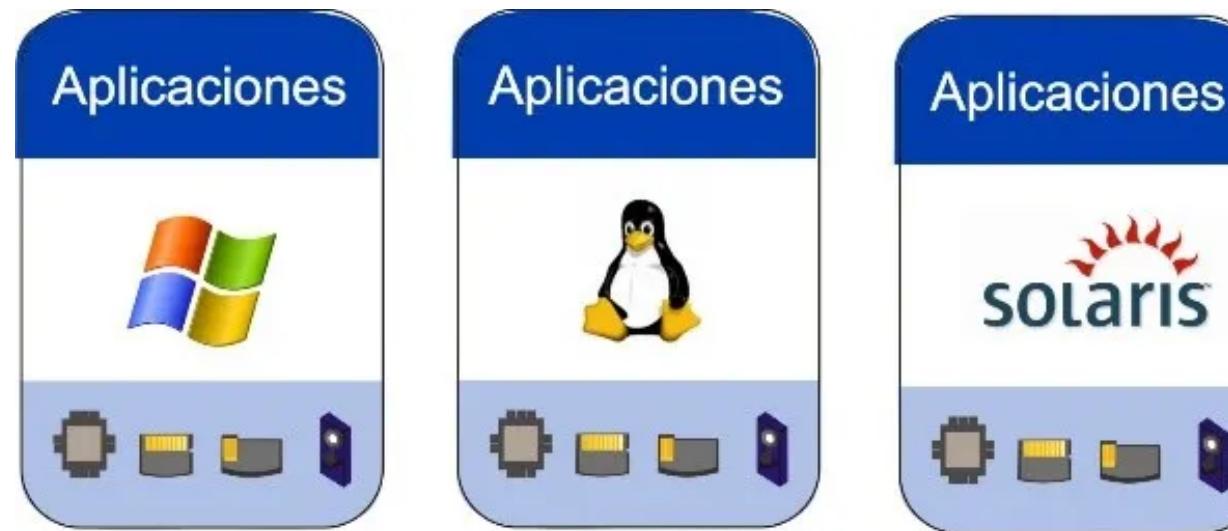
Carranza Ochoa José David
Ríos Lira Gamaliel

Máquinas virtuales

Implementación de arquitecturas de **hardware real** a través de **software**.

Requiere de dos partes:

- **Host.** Computadora donde se ejecuta el **hypervisor**.
- **Guest.** Computadora emulada.



CPU Memoria Tarjeta
de Red

Contenedor

Abstracciones en la capa de aplicación que empaquetan el código y las dependencias juntas.

- Tiene todo lo necesario para ejecutar la aplicación en cualquier momento.
- Hace uso del **kernel**
- Sólo hay **un S.O. en ejecución**

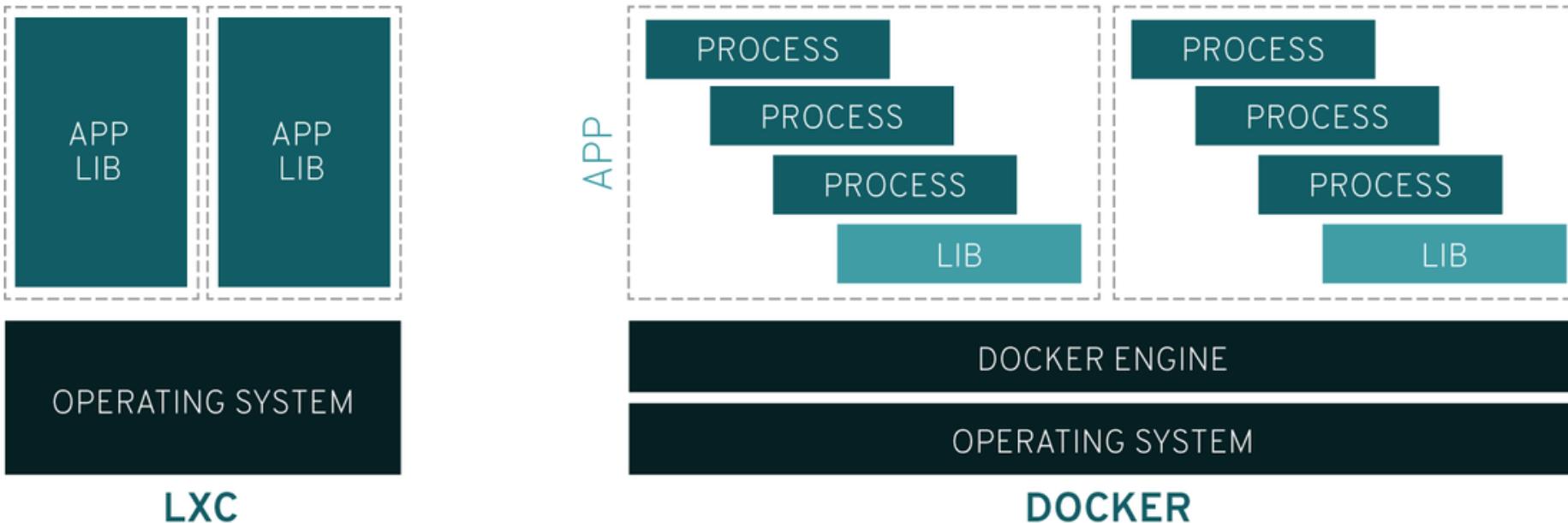


Docker

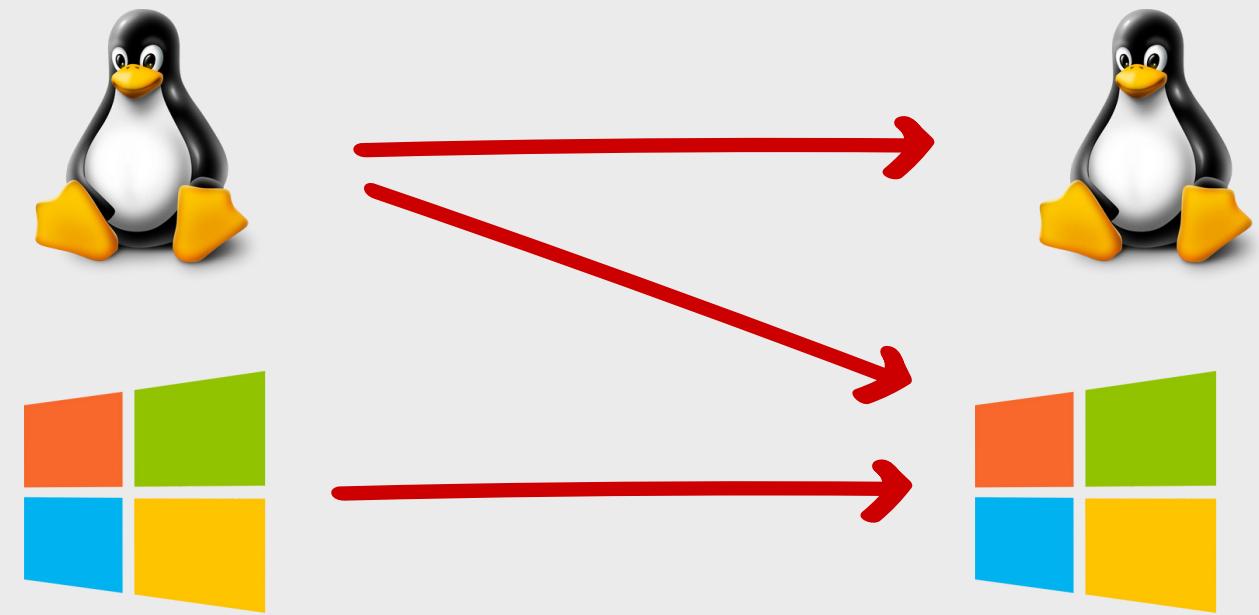
Plataforma Open Source que automatiza el despliegue de contenedores, combinando **código fuente** de la aplicación con **bibliotecas del S.O.**

2013 → en base a tecnología LXC

Traditional Linux containers vs. Docker



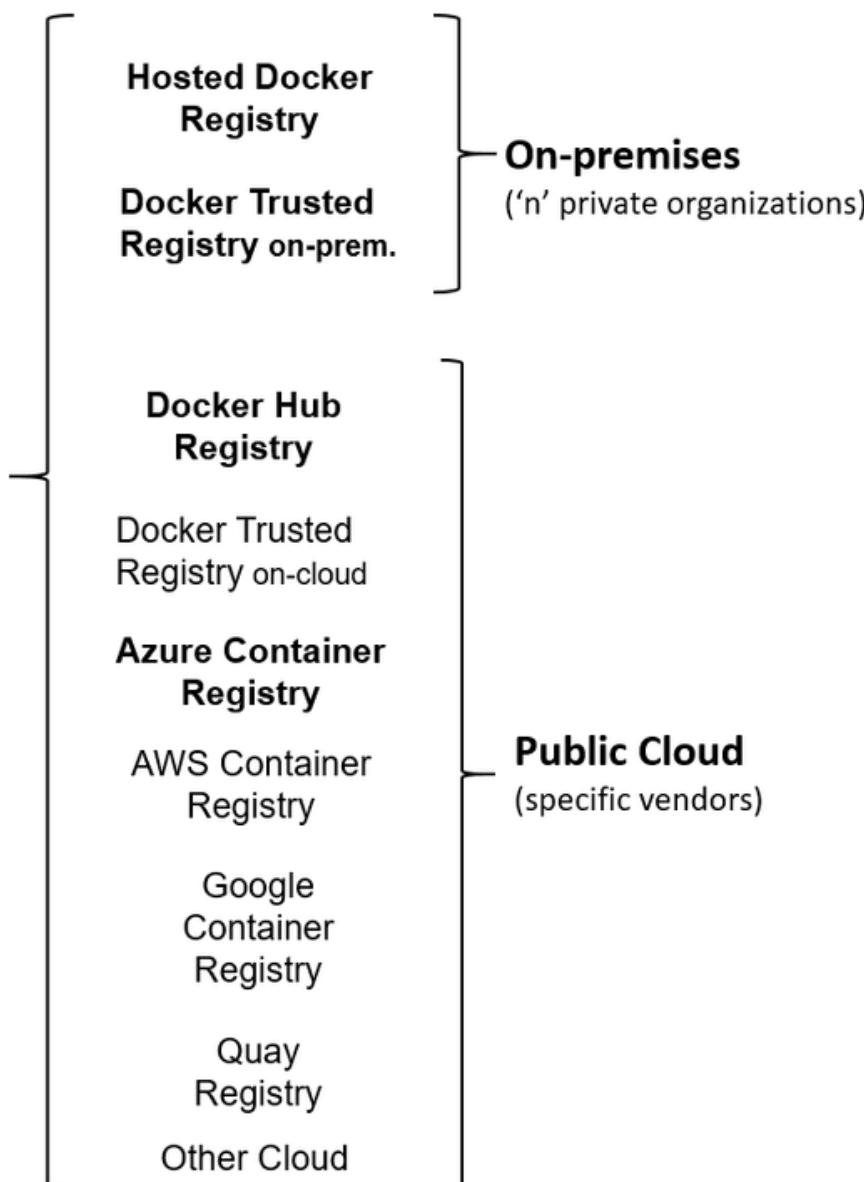
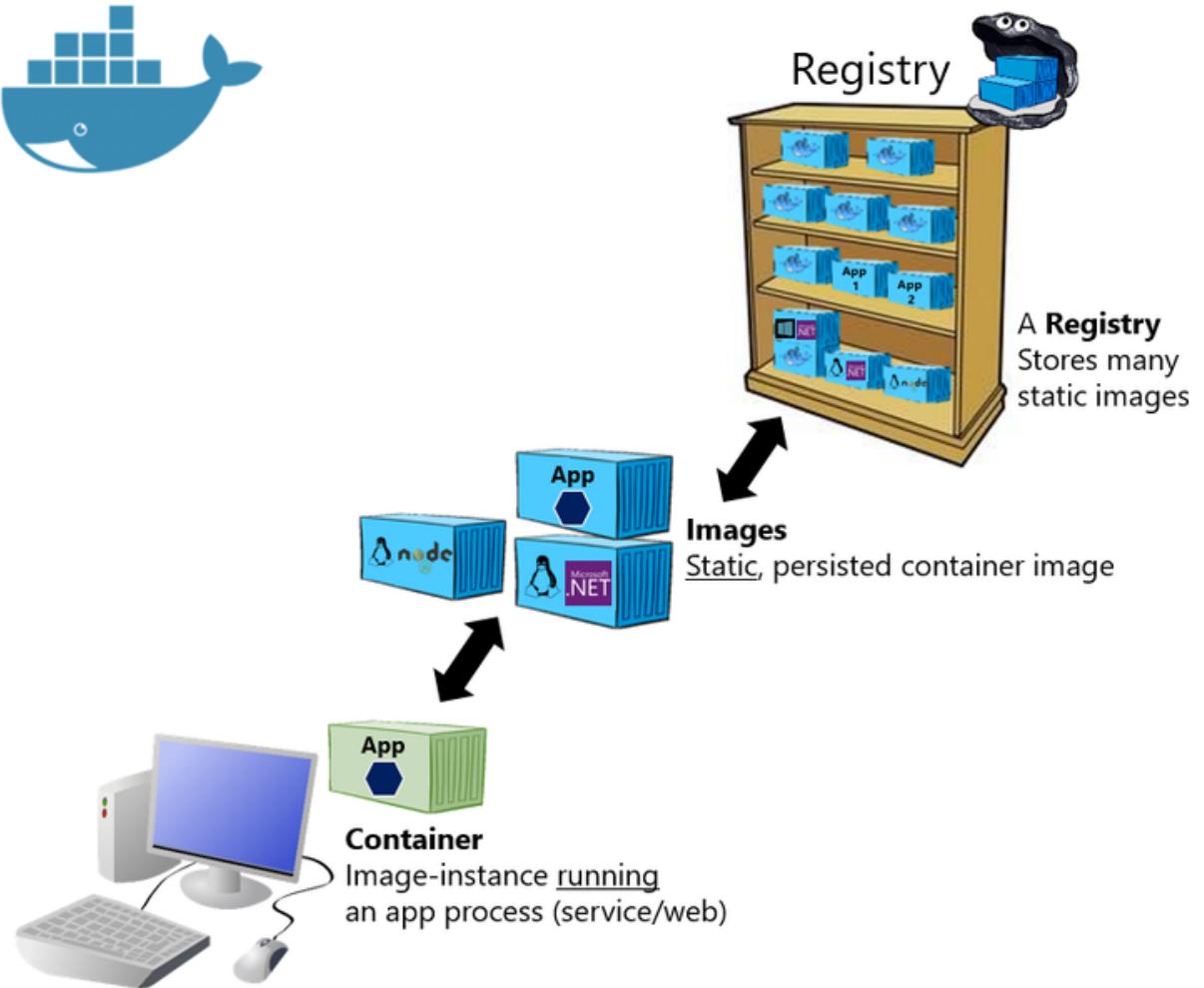
Ejecución de forma nativa en Linux y Windows aunque:



Host de Windows y de macOS requieren una VM con Hyper-V

Componentes

Basic taxonomy in Docker



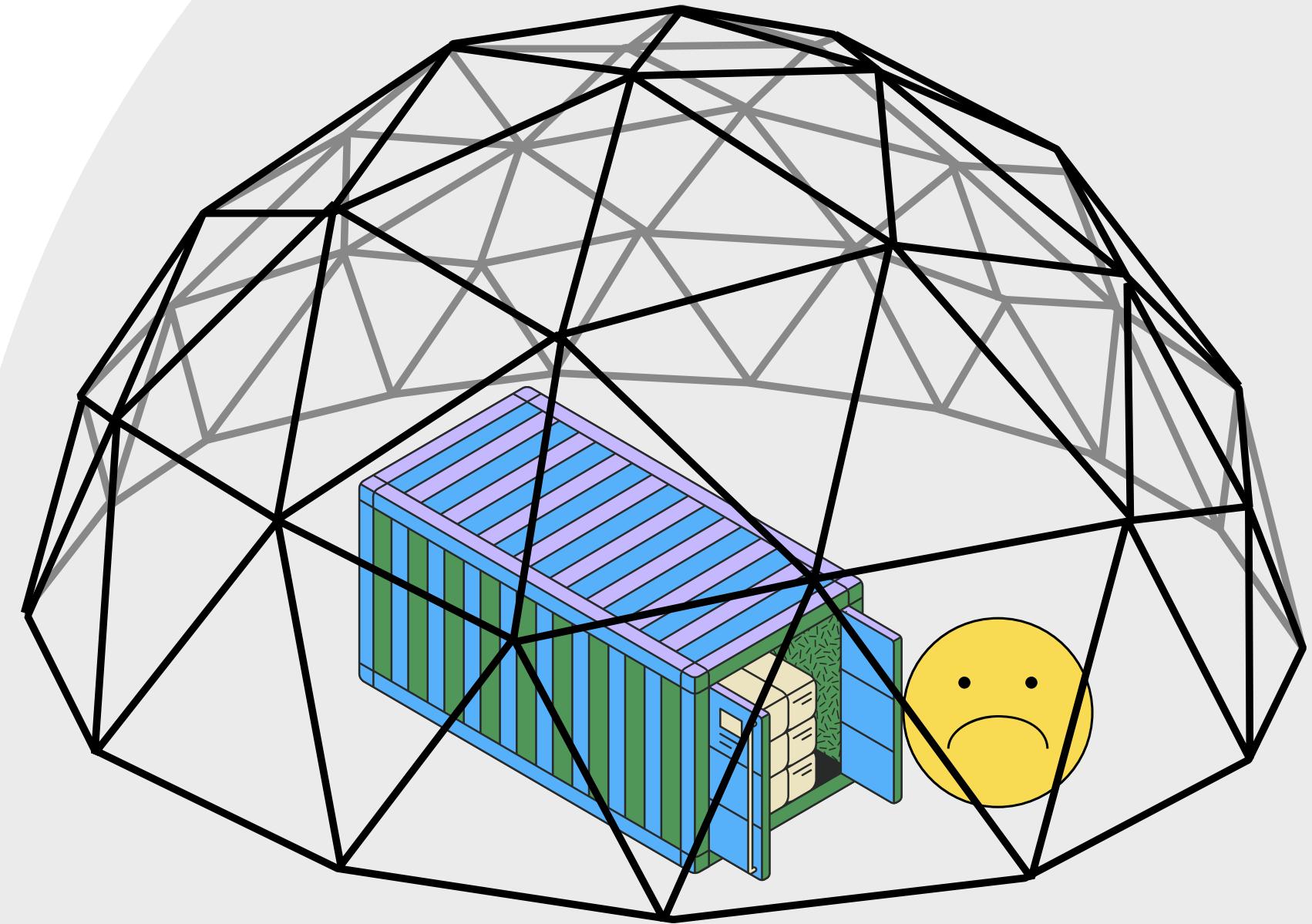
- 1 Contenedor
- 2 Docker
- 3 Motor de Docker
- 4 Imágenes de Docker
- 5 Docker Registry
- 6 Docker Compose
- 7 Kubernetes



Contenedores Docker

Sus características...

- Son **procesos en Linux**.
- **Aislado** del sistema operativo *host*.
- Cantidad de **recursos limitada**.
- Sistema de archivos **independiente**.



Namespaces

Aislamiento entre procesos generando
instancias del recurso global



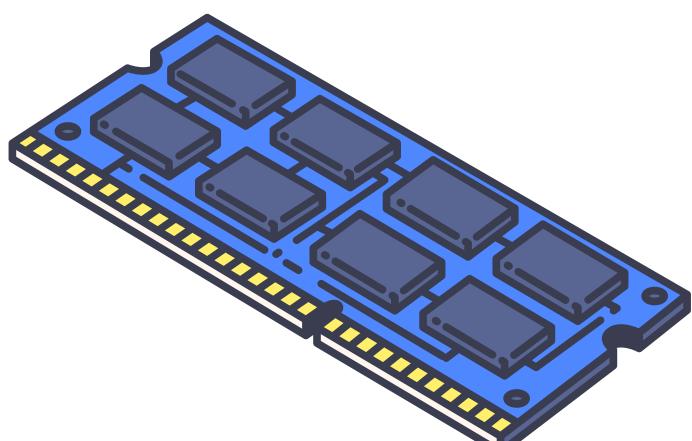
Contenedores como "subsistemas"

- **PID:** Identificador de proceso
- **NET:** Aislamiento de redes
- **MNT:** Aislamiento de unidades
- **UTS:** Identificadores del sistema
- **CHROOT:** Acceso al sistema
de archivos del host

Grupos de control

Contabilizan y controlar recursos
del sistema

- Memory cgroup
- CPU cgroup
- Devices cgroup





Cualidades

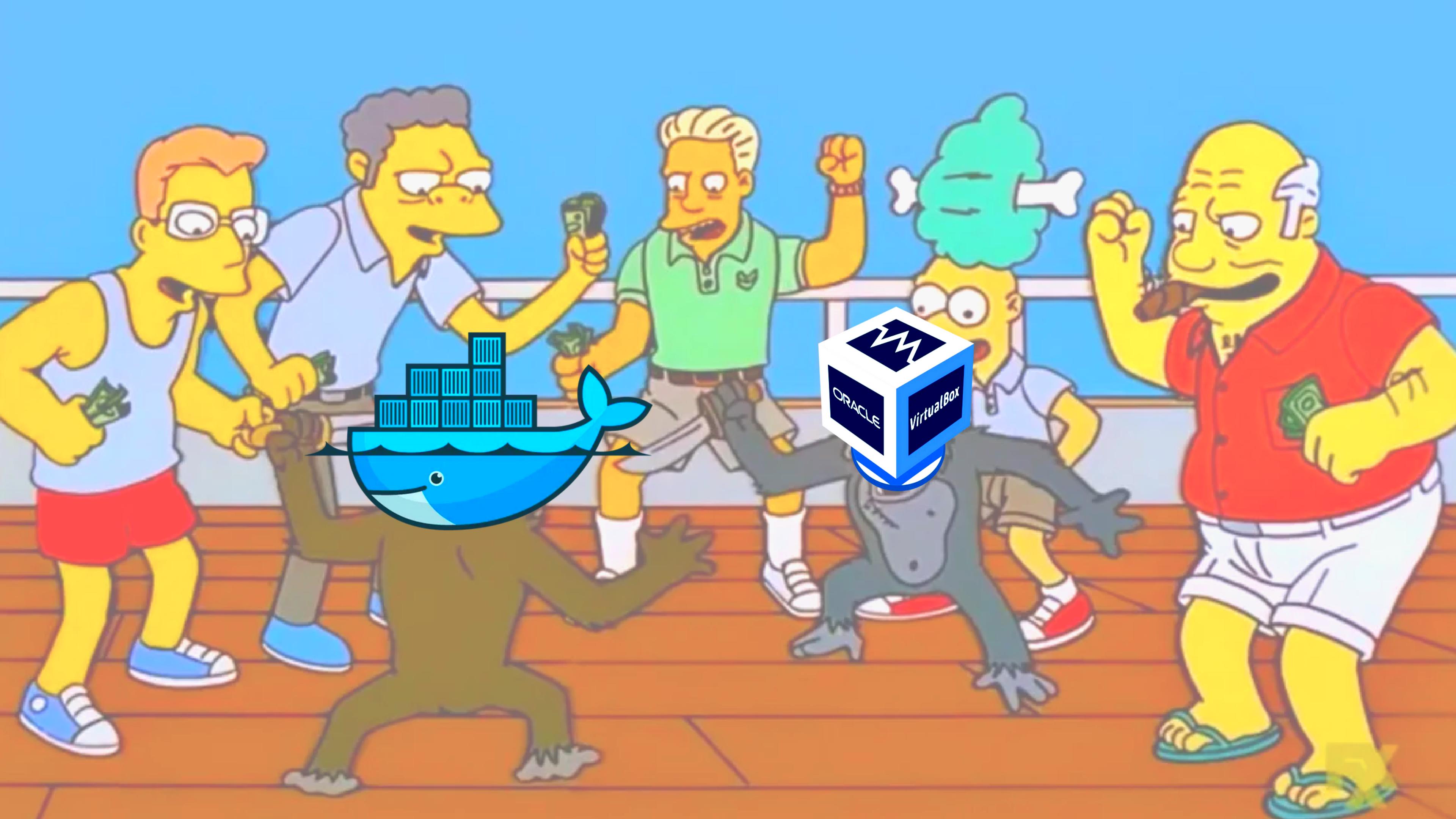
- **Menos recursos**
- **Aislamiento** de servicios
- **Estandarización** de SW
- **Portabilidad** de SW
- **Automatización** y **reutilización** de contenedores
- Control de versiones
- Comunidad abierta

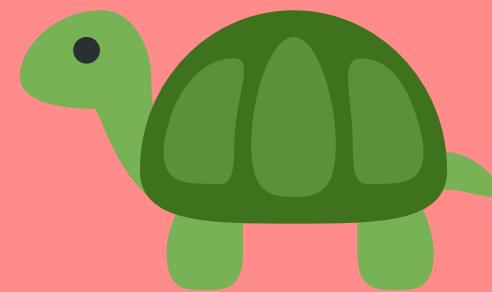
Limitantes

- Complejidad al desplegar muchos contenedores → Kubernetes
- Funciones del tipo UNIX ≠ contenedores tradicionales de Linux
- Riesgo de seguridad al **compartir el kernel del host**
- Imágenes de dudosa procedencia

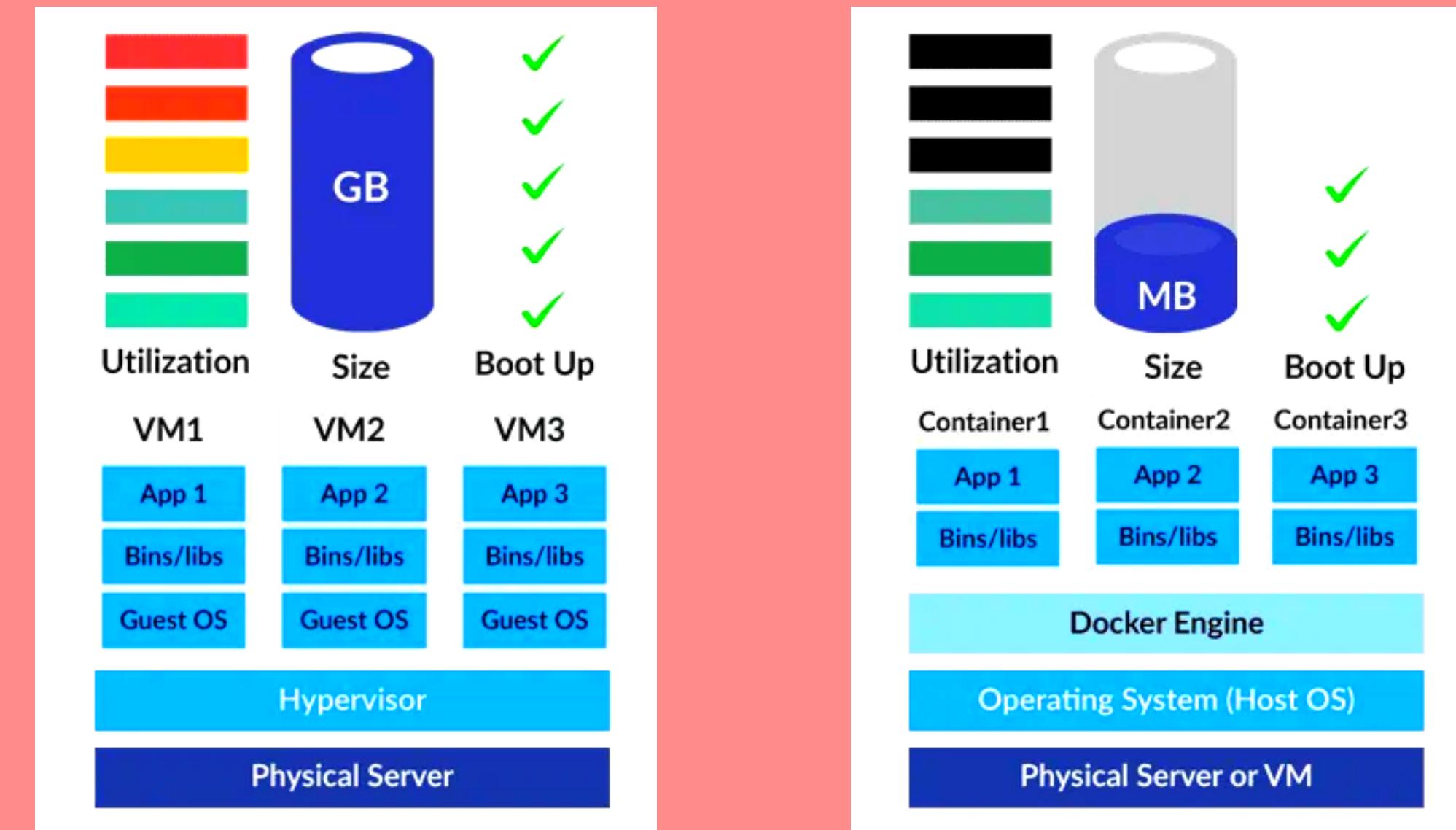


** Docker no se lleva bien con GUI





- 1. Recursos**
- 2. Virtualización**
- 3. Dependencia**
- 4. Aislamiento**
- 5. Flexibilidad**



Manuales GB
Hardware
Hypervisor
Sistema completo
Migración rápida



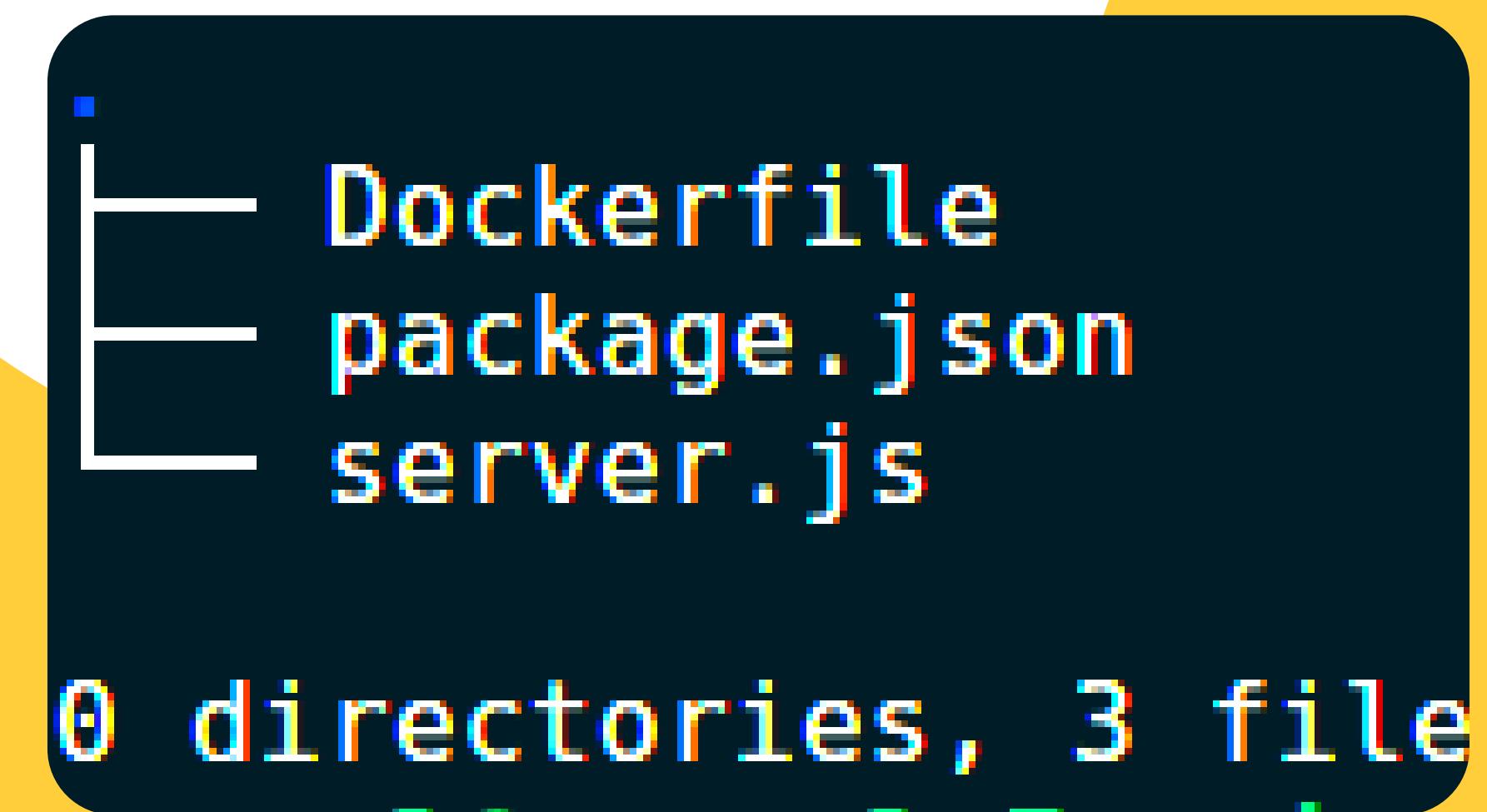
Automáticos KB, MB
S.O.
Portables
Individual de Apps
Destrucción y recreación



Hello World!

Desde Docker

(Sólo con fines ilustrativos)



```
tree
├── Dockerfile
├── package.json
└── server.js

0 directories, 3 files
```

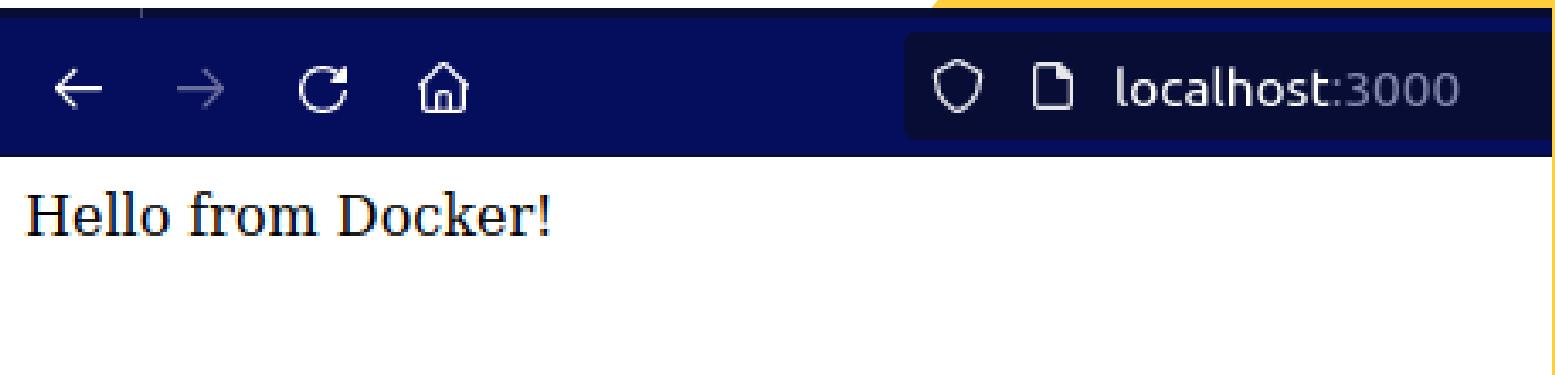
server.js

```
1 const express = require('express');
2
3 const PORT = 8000;
4 const HOST = '0.0.0.0';
5
6 const app = express();
7
8 app.get('/', (req, res) => {
9   res.send('Hello from Docker!');
10 });
11
12 app.listen(PORT, HOST, () => {
13   console.log(`Running on http://${HOST}:${PORT}`);
14 });
15
16
```

Dockerfile

```
1 FROM node:alpine
2
3 WORKDIR /usr/src/app
4
5 # Configuración de dependencias
6 COPY package*.json ./
7 RUN npm install
8
9 # Configuración del código
10 COPY . .
11
12 EXPOSE 8000
13
14 # Comando de ejecución del programa
15 CMD ["node", "server.js"]
16
```

```
piron:DockerExample$ docker build . -t gamarl/hello-world
  (10/10) FINISHED
    bad .dockerignore
    Ign context: 67B
    bad build definition from Dockerfile
    Ign dockerfile: 268B
    bad metadata for docker.io/library/node:alpine
    docker.io/library/node:alpine@sha256:d3a3d691797cef0b70e361788a
    bad build context
    Ign context: 125B
    WORKDIR /usr/src/app
    COPY package*.json .
    RUN npm install
    COPY .
    image
    layers
    image sha256:a6aca95b1b1723c2dd958c237fa0c787b94d95598cc06372f2
    docker.io/gamarl/hello-world
```

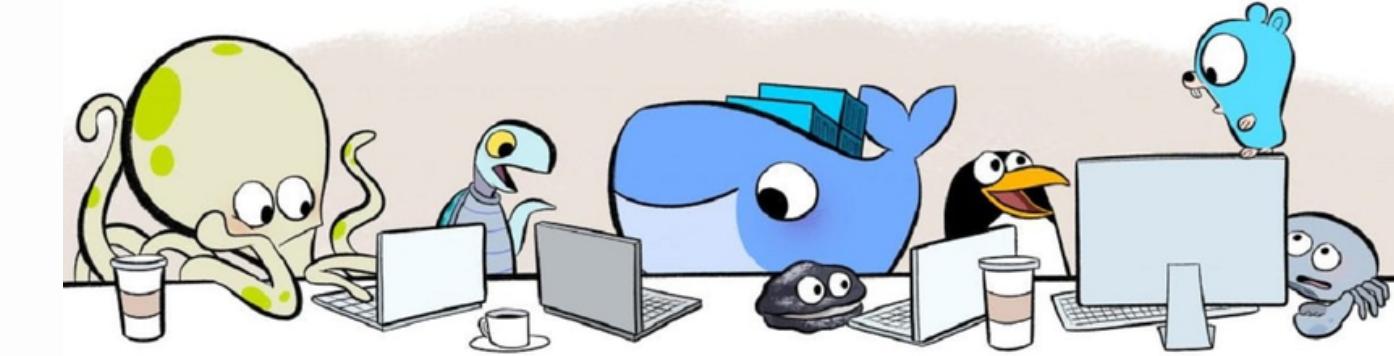
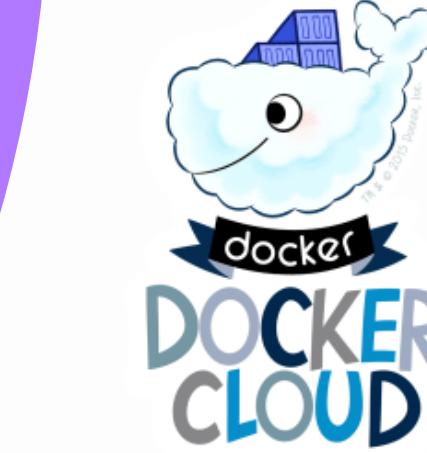
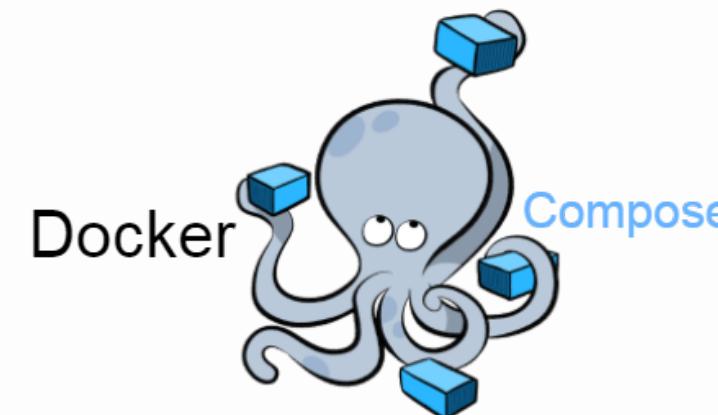


```
gamarl@gamarl-Inspiron:DockerExample$ docker run -p 3000:8000 gamarl/hello-world
Running on http://0.0.0.0:8000
```

Por investigar...



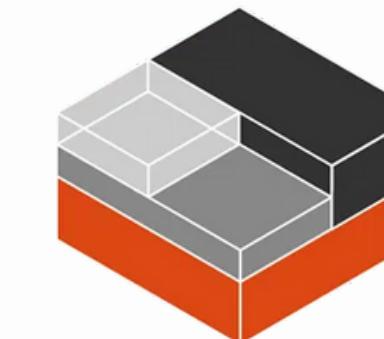
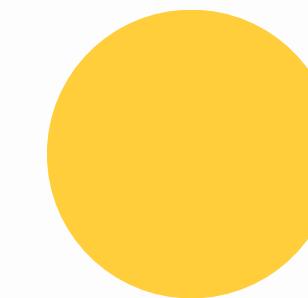
podman



kubernetes

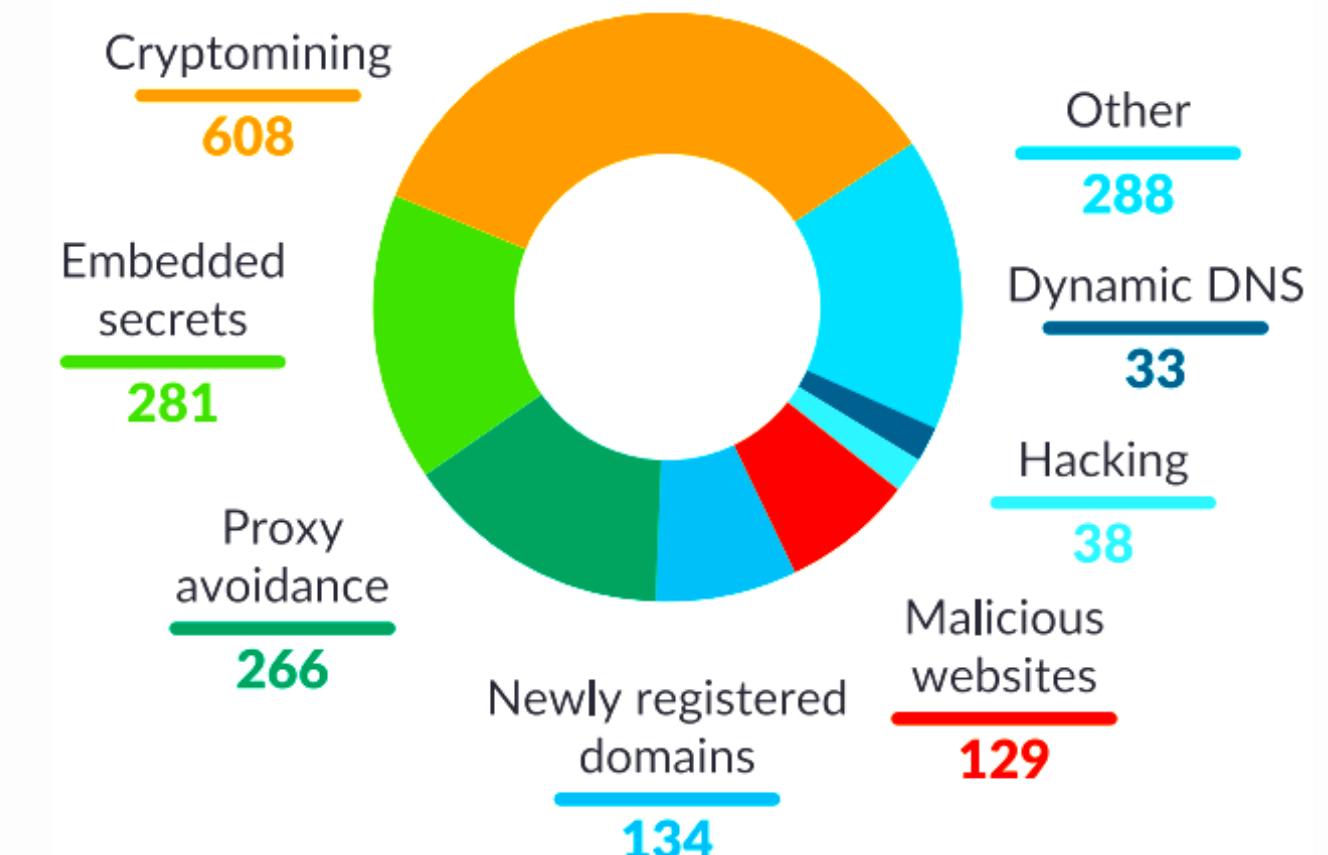


docker hub



LXC

Malicious Image Categories



Fuentes

<https://www.docker.com/resources/what-container/>

<https://aws.amazon.com/es/docker/>

<https://www.redhat.com/es/topics/containers/what-is-docker>

<https://www.netapp.com/es/devops-solutions/what-are-containers/>

<https://www.ibm.com/mx-es/topics/docker>

<https://learn.microsoft.com/es-es/dotnet/architecture/microservices/container-docker-introduction/docker-defined>

<https://www.oracle.com/mx/cloud/cloud-native/container-registry/what-is-docker/>

Nombres (nuevamente):

- Carranza Ochoa José David
- Ríos Lira Gamaliel

