



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Tecnología de Contenedores

CIDIA



OBJETIVO

Identificar conceptos básicos de Docker

Tecnología de contenedores

Sirve para

- Aislar
- Empaquetar
- Distribuir
- Implementar

SOFTWARE

Existen desde 1979 en UNIX
(chroot)

2000 En FreeBSD Jails

2001 En Linux VServer

2013 Google Let Me Contain That For
You (LMCTFY) -> Docker

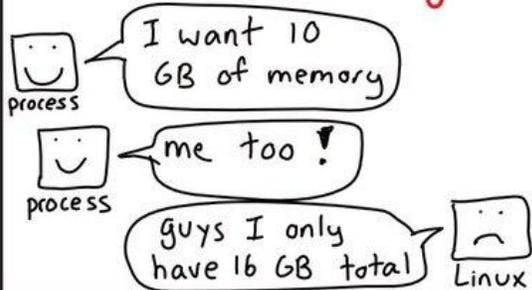
Yo lo uso desde 2014 !!!!!!!

cgroups

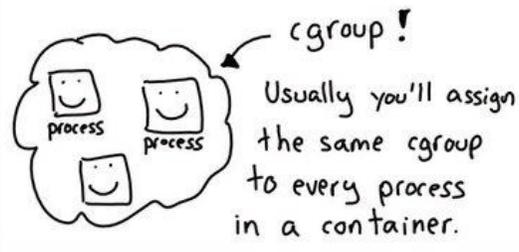
JULIA EVANS
@b0rk

cgroups

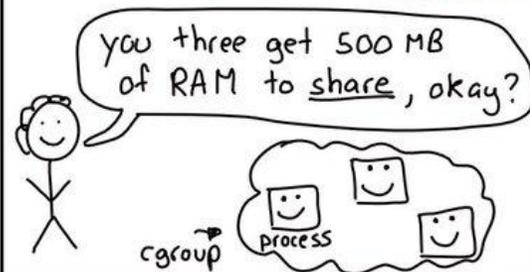
processes can use
a lot of memory



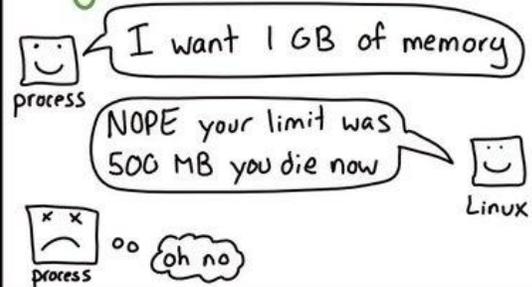
a cgroup is a
group of processes



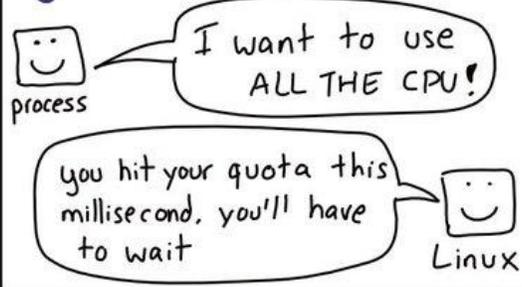
cgroups have
memory/CPU limits



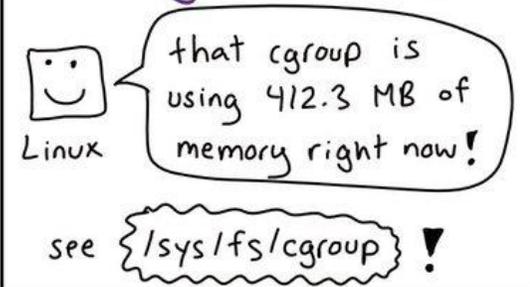
use too much memory:
get OOM killed



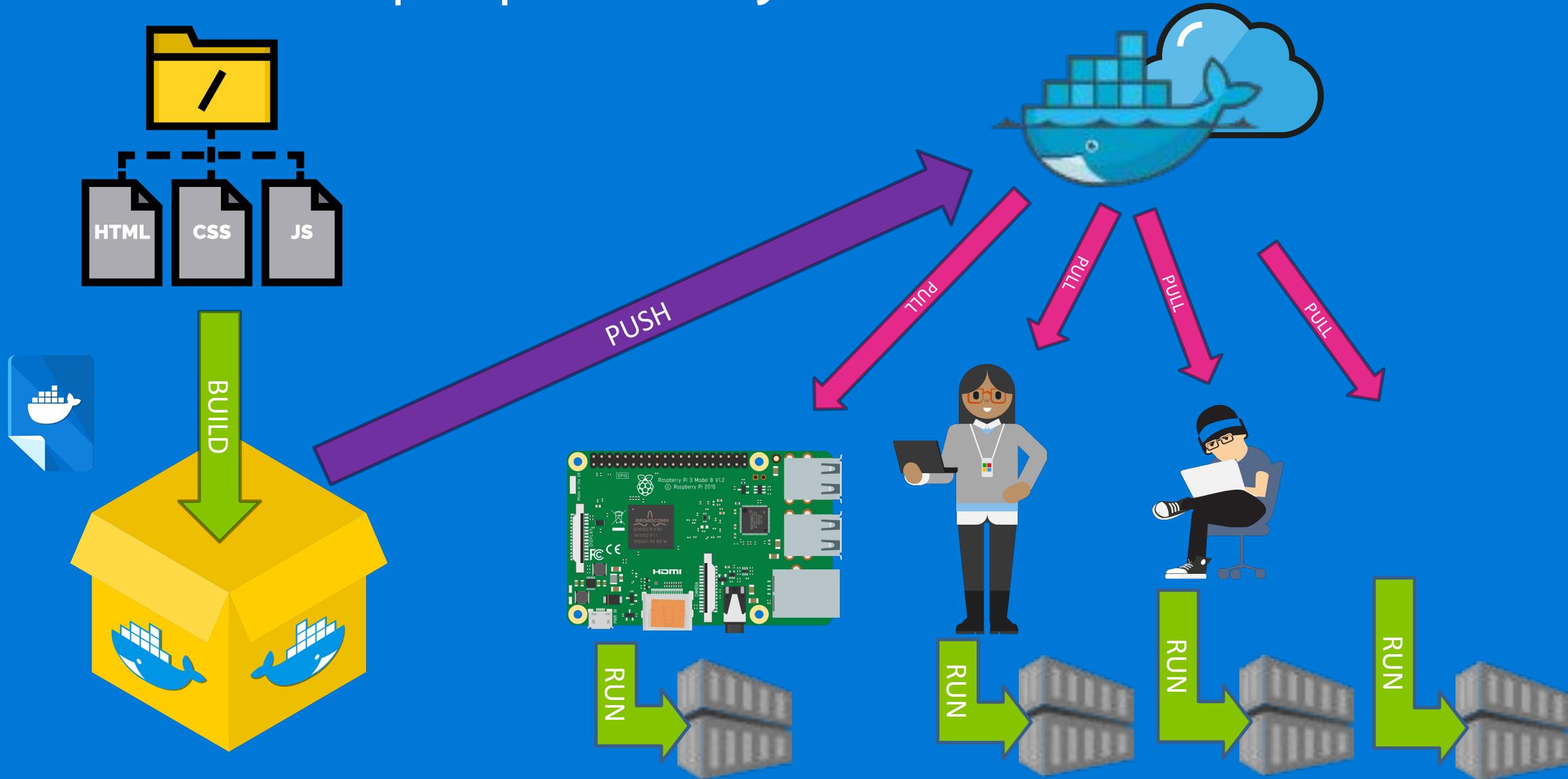
use too much CPU:
get slowed down



cgroups track
memory & CPU usage



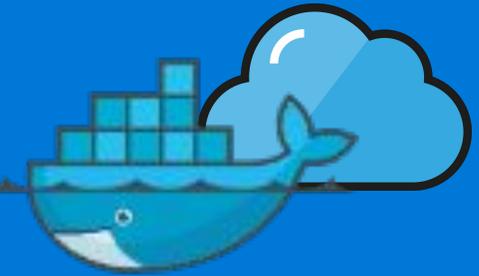
Docker : Empaquetado y Distribución



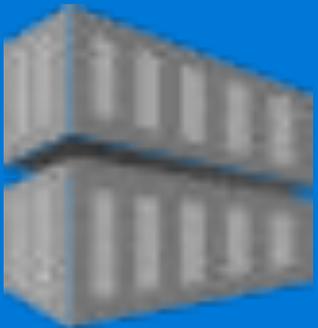
Conceptos de Docker



IMAGEN: Set de archivos de solo lectura que sirve de plantilla para crear un CONTENEDOR. A menudo, una imagen se basa en otra imagen, con alguna personalización adicional.



REGISTRY: REPOSITORIO DE IMÁGENES. Puede ser público o privado



CONTENEDOR: La definición más simple es un conjunto de PROCESOS que corre aislado. Pero también es una instancia de una imagen. Configurable mediante el uso de parámetros determinados por la imagen se puede configurar

Mnemotecnia

Algo **dinámico** es una instancia de algo **estático**

- Objeto instancia de clase
- Proceso instancia de un programa
- Contenedor instancia de una imagen

Contenedor vs imagen

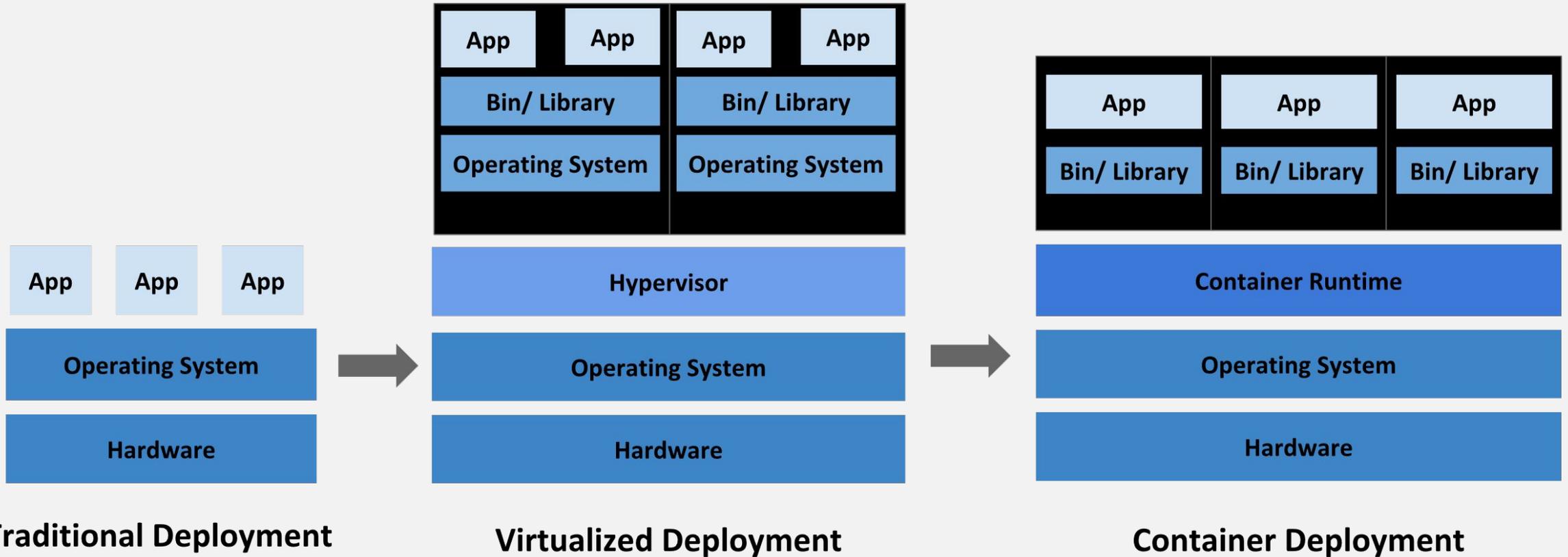
Imagen

- Solo lectura
- Se almacena localmente (cache) o en el registry
- Pueden existir sin un contenedor

Contenedor

- Es un conjunto de procesos
- Efímero / stateless : Cuando se re/crea se pierden los cambios del sistema de archivos
- Se almacena localmente
- No pueden existir sin una imagen

“TRADICIONAL” | VM | CONTENEDORES



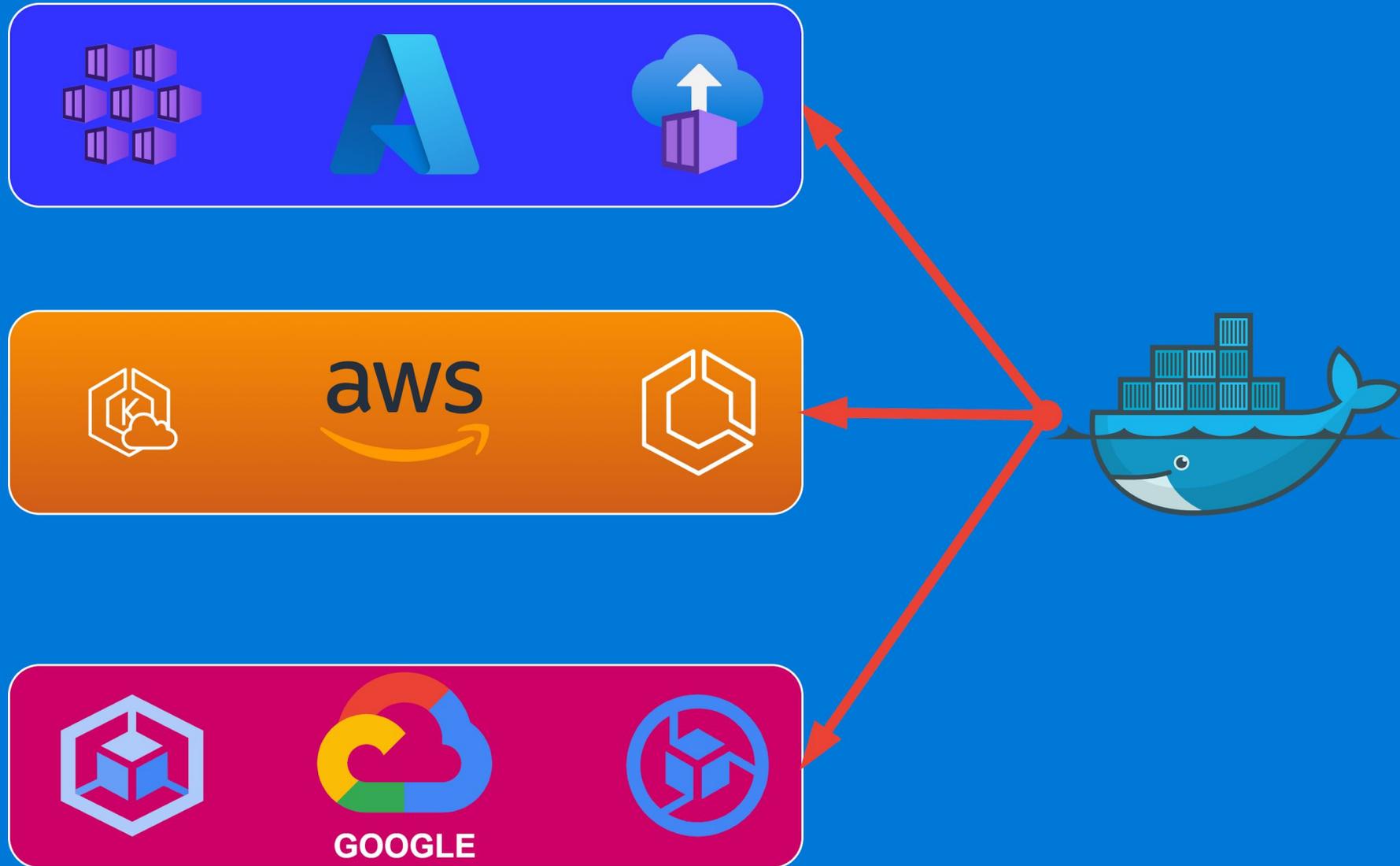
Ventajas

- Entorno homogéneo
 - Facilidad en la distribución
 - Facilidad en la administración
 - Multi-nube
- Empaquetado / Versionado / Etiquetado
 - Despliegue mas rápido
 - Despliegue continuo (CD)
- Aislamiento entre otros componentes en el mismo entorno
 - Mas seguro
 - Mas estable
 - Mas “liviano” que una maquina virtual
 - Mas fácil de instalar que un sistema “suelto”

Estandarización por medio de la CNCF

- 2014: Google lanza Kubernetes (aka Borg)
- 2015: CNCF Cloud Native Computing Foundation: Google + Linux foundation + socios (Red Hat, IBM, Microsoft, VMware)
- 2015- Open Container Initiative (OCI) image specification
- 2016 CRI: Container Runtime Interface

Nube, on premise, en cualquier lado



WORKSHOP – EJECUTANDO CONSOLA UBUNTU

```
$ docker run -i -t ubuntu /bin/bash
```

```
Unable to find image 'ubuntu:latest' locally
```

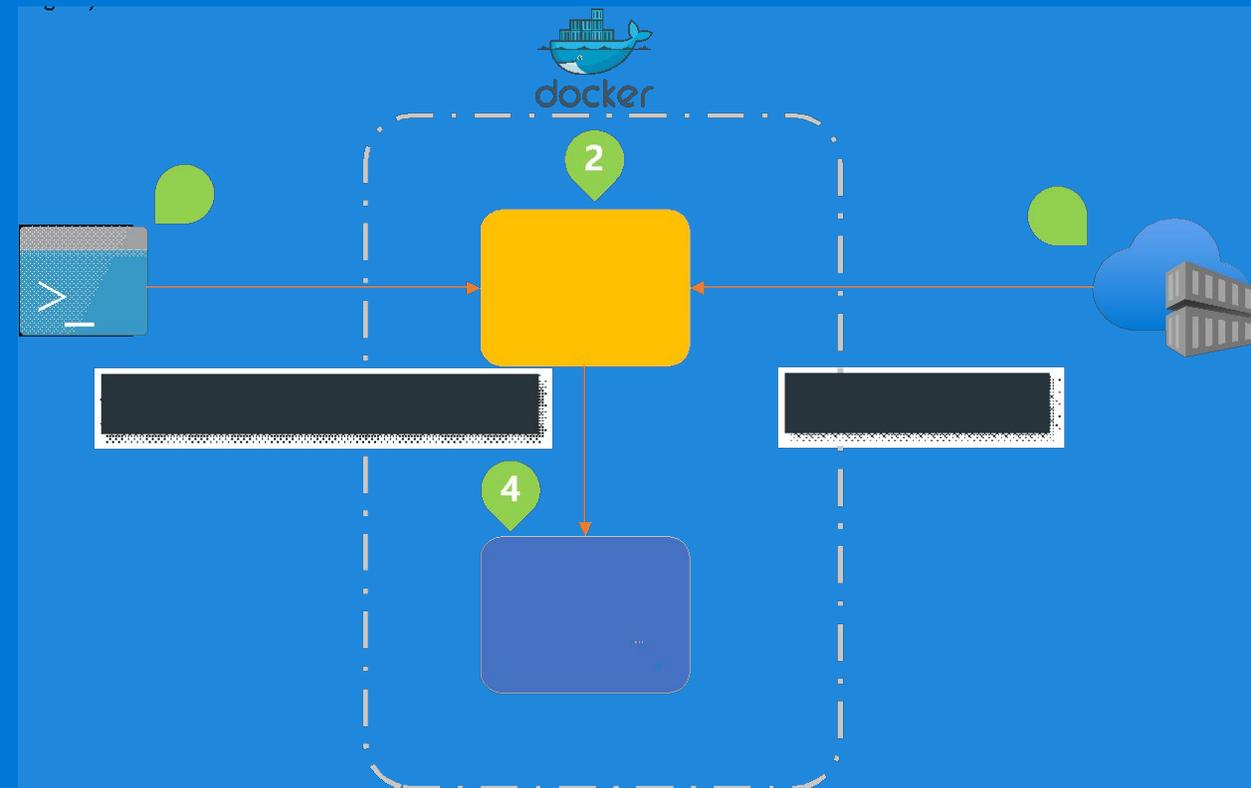
```
f3ef4ff62e0d: Pull complete
```

```
Status: Downloaded newer image for ubuntu:latest
```

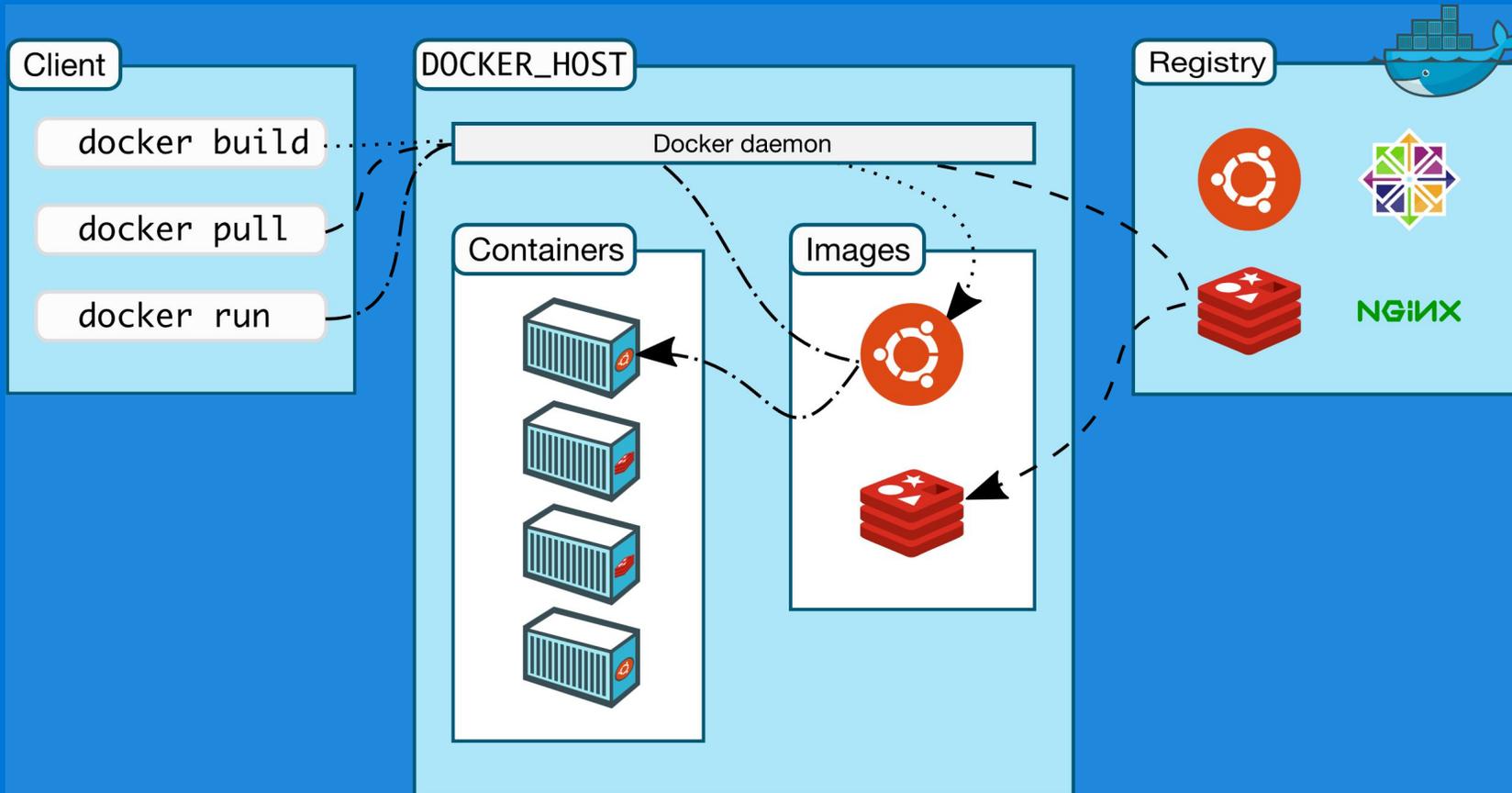
```
root@2a5d294575c1:/# ls
```

¿Que paso? Estas corriendo una consola linux

1. El Docker CLI (docker.exe) envia un comando al servicio de Docker via su API
 2. Servicio de Docker: busca o la imagen ubuntu localmente y no la encontró
 3. Servicio de Docker: Bajo la imagen desde el <https://hub.docker.com/> repositorio por defecto (docker pull ubuntu)
 4. Servicio de Docker: instancia un nuevo contenedor.
 - Crea el sistema de archivos del contenedor
 - Ejecuto el programa /bin/bash (el interprete de COMANDOS) que se ejecuta de forma interactiva (por los argumentos -i y -t)
- **DOCKER CLI: Queda una consola dentro del contenedor, ingresar el comando exit para salir.**



Arquitectura de Docker



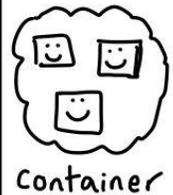
- **Client:** Envía comandos a un Docker local o remoto.
 - `build`: Genera una imagen
 - `run`: crea un contenedor a partir de una imagen
 - `pull`: Baja de un registry una imagen y la graba localmente
 - `push`: sube una imagen local a un registry
- **DOCKER_HOST:** Equipo donde está instalado Docker.
- **Registry:** Repositorio de imágenes. Análogo al “Play Store” de Android o Microsoft Store de Windows.

Contenedores

JULIA EVANS
@bork

what's a container?

a Linux container
is a group of processes



Container

We have our own
filesystem but
we're still just
regular processes!

Linux containers are
isolated from other processes

they can have their own:

- users
- network namespace
- filesystem
- process IDs
- memory / CPU limits

Kernel features that
isolate Linux containers

cgroups

namespaces

capabilities

seccomp-bpf

there are many ways
to run Linux containers

runc

systemd-nspawn

LXC

Docker

Docker
uses runc
under the
hood

your own homegrown
bash script

and containers can be
set up in different ways



container 1

I have my own
filesystem!

I don't! I
have my system
calls restricted!



container 2

extra confusion:

"container" sometimes means
"lightweight VM"

Fargate and kata Containers
are actually VMs and not
Linux containers (they don't
share a kernel with other
containers)

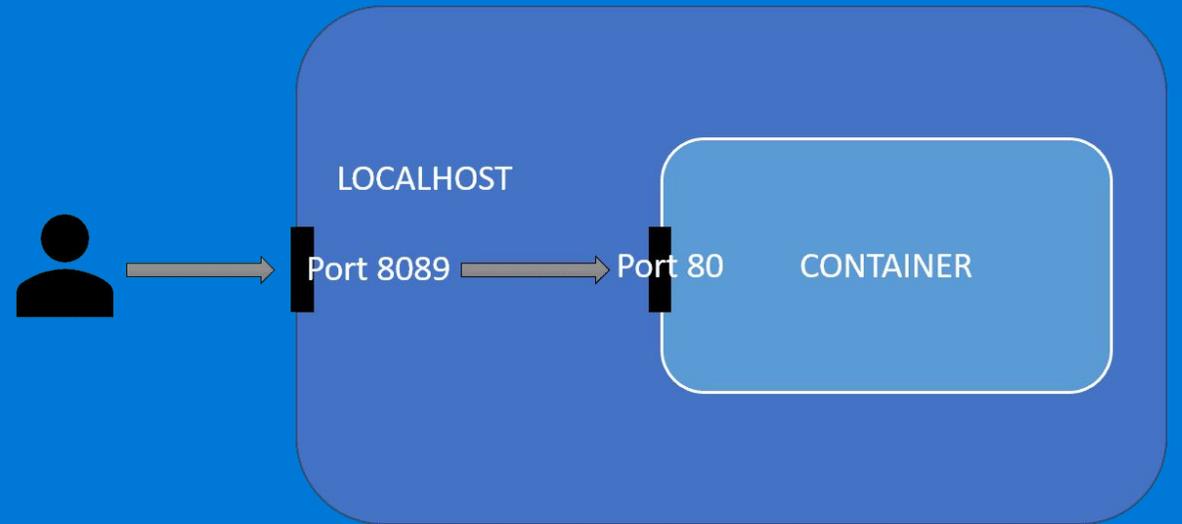
Conectando con el mundo exterior

- **Habilitar la comunicación de red.**
 - Mapeo de puertos TCP/UDP
- **Mantener el ESTADO / Persistir información:**
 - Archivos de datos
 - logs
 - * no es buena idea meter en contenedores sistemas PRODUCTIVOS que tengan un uso intensivo de archivos, como una base de datos. Es mejor tenerlos por fuera
- **Mapeo de archivos**
 - Archivos de configuración
 - Archivos estáticos

Mapeo de Puertos

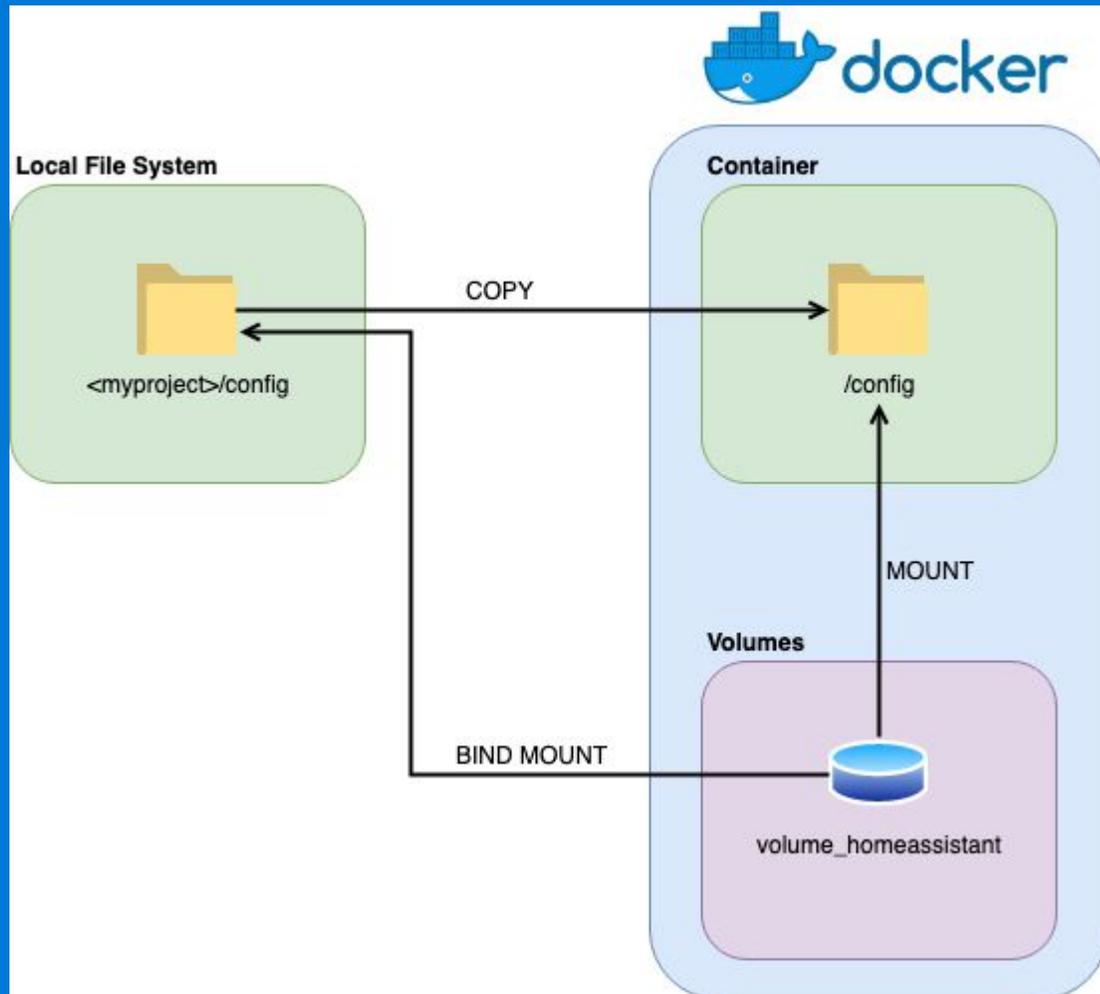
Podemos conectar nuestro contenedor a la red mapeando puertos utilizando el argumento

`-p puerto_host :
puerto_contenedor`



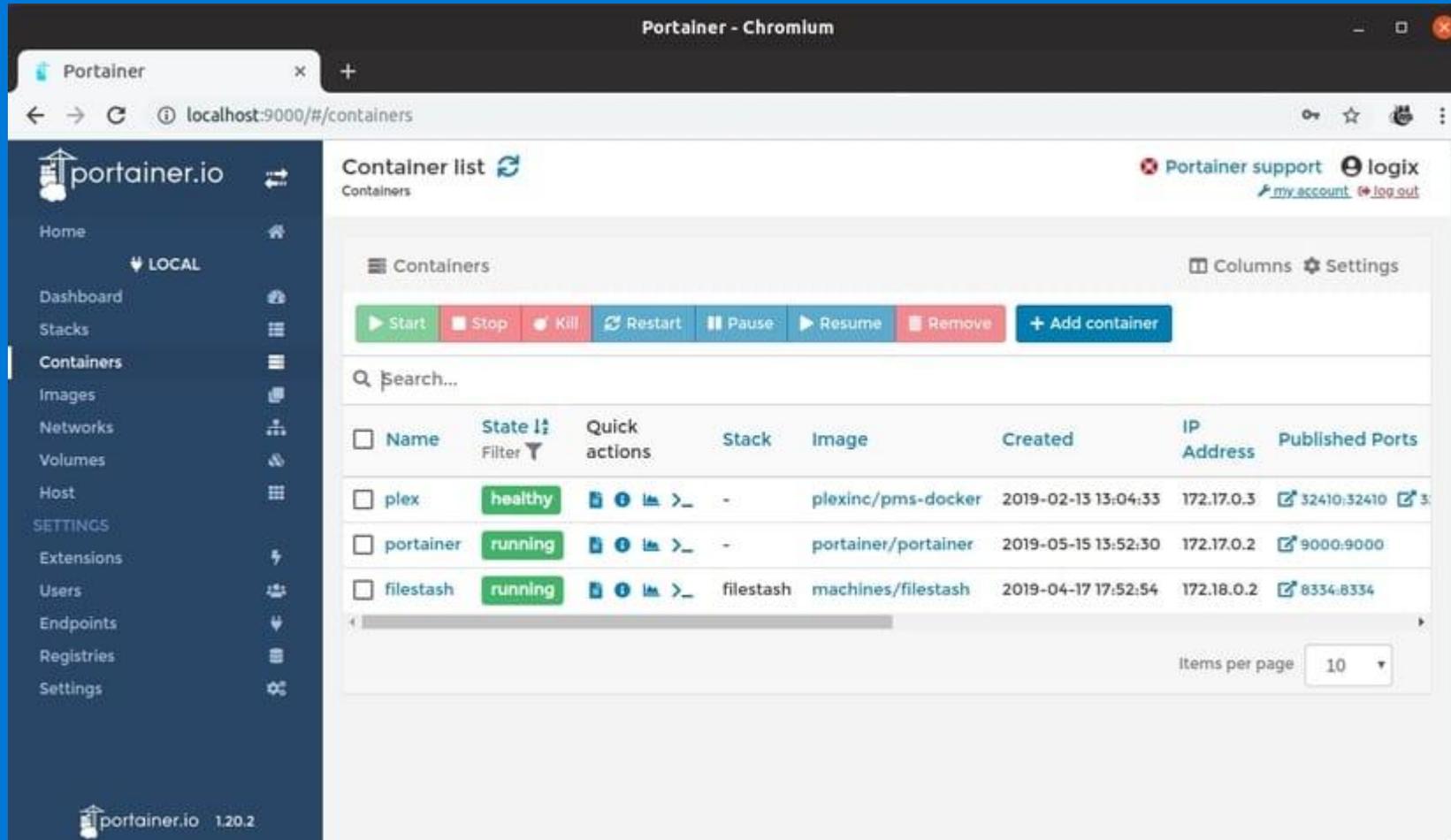
```
docker run -d -p 8089:80 docker/getting-started
```

VOLUMENES



- Por defecto los contenedores son efímeros / stateles: vuelven a su estado original (el de la imagen) cuando se recrean
- Los volúmenes permiten persistir archivos independientemente del ciclo de vida del contenedor
- En Linux todo se maneja con el sistema de archivo, incluso dispositivos.

DOCKER - PORTAINER



The screenshot displays the Portainer web interface in a Chromium browser window. The browser's address bar shows the URL `localhost:9000/#/containers`. The interface features a dark sidebar on the left with navigation options: Home, LOCAL, Dashboard, Stacks, Containers, Images, Networks, Volumes, Host, SETTINGS, Extensions, Users, Endpoints, Registries, and Settings. The main content area is titled "Container list" and includes a toolbar with buttons for Start, Stop, Kill, Restart, Pause, Resume, Remove, and Add container. Below the toolbar is a search bar and a table of containers.

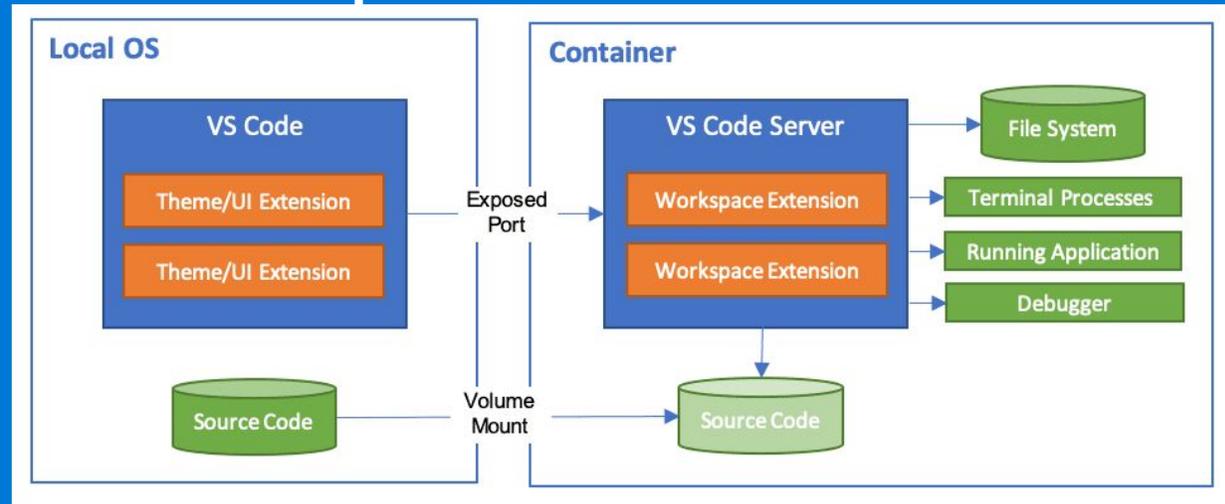
<input type="checkbox"/>	Name	State Filter	Quick actions	Stack	Image	Created	IP Address	Published Ports
<input type="checkbox"/>	plex	healthy	   	-	plexinc/pms-docker	2019-02-13 13:04:33	172.17.0.3	 32410:32410  32410
<input type="checkbox"/>	portainer	running	   	-	portainer/portainer	2019-05-15 13:52:30	172.17.0.2	 9000:9000
<input type="checkbox"/>	filestash	running	   	filestash	machines/filestash	2019-04-17 17:52:54	172.18.0.2	 8334:8334

At the bottom right of the table, there is a dropdown menu for "Items per page" set to 10. The footer of the interface shows the Portainer logo and version number 1.20.2.

Devcontainers: Contenedores para desarrollo

Encapsula y aísla el ambiente de desarrollo dentro de una imagen

- Utiliza **VS Code Server** para conectar el contenedor con el VS Code IDE
- Permite PRE-Instalar extensiones de VS Code DENTRO DEL CONTENEDOR
- Reduce el tiempo para inicializar un ambiente de desarrollo
- Entorno de desarrollo **HOMOGENEO**



REQUISITOS devcontainers

- **Docker Desktop**
 - WSL2 BACKEND
- **Visual Studio Code**
 - Microsoft's Remote Development Extension



Configuración *devcontainer*

Configuración en

`.devcontainer/devcontainer.json`

Especifica una imagen que debe contener el `vscode-server` instalado

Se pueden incluir extensiones de `vscode` que se instalarán en el entorno

Se puede incluir puertos que expondrá el entorno

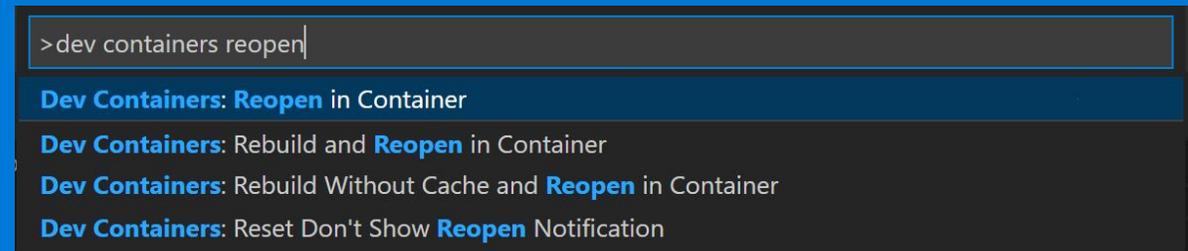
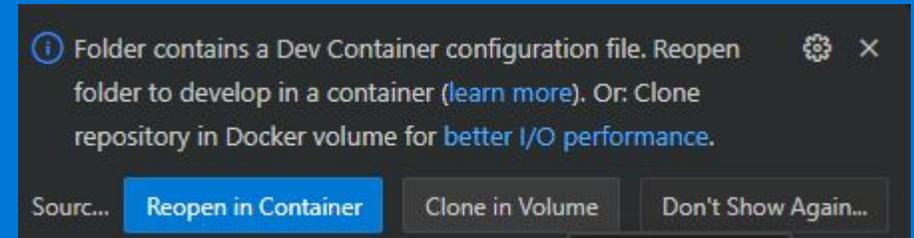
```
{
  "image": "mcr.microsoft.com/devcontainers/typescript-node",
  "customizations": {
    "vscode": {
      "extensions": ["streetsidesoftware.code-spell-checker"]
    }
  },
  "forwardPorts": [3000]
}
```

Abrir un devcontainer

Vscode detecta automáticamente que el entorno tiene configurado un dev container

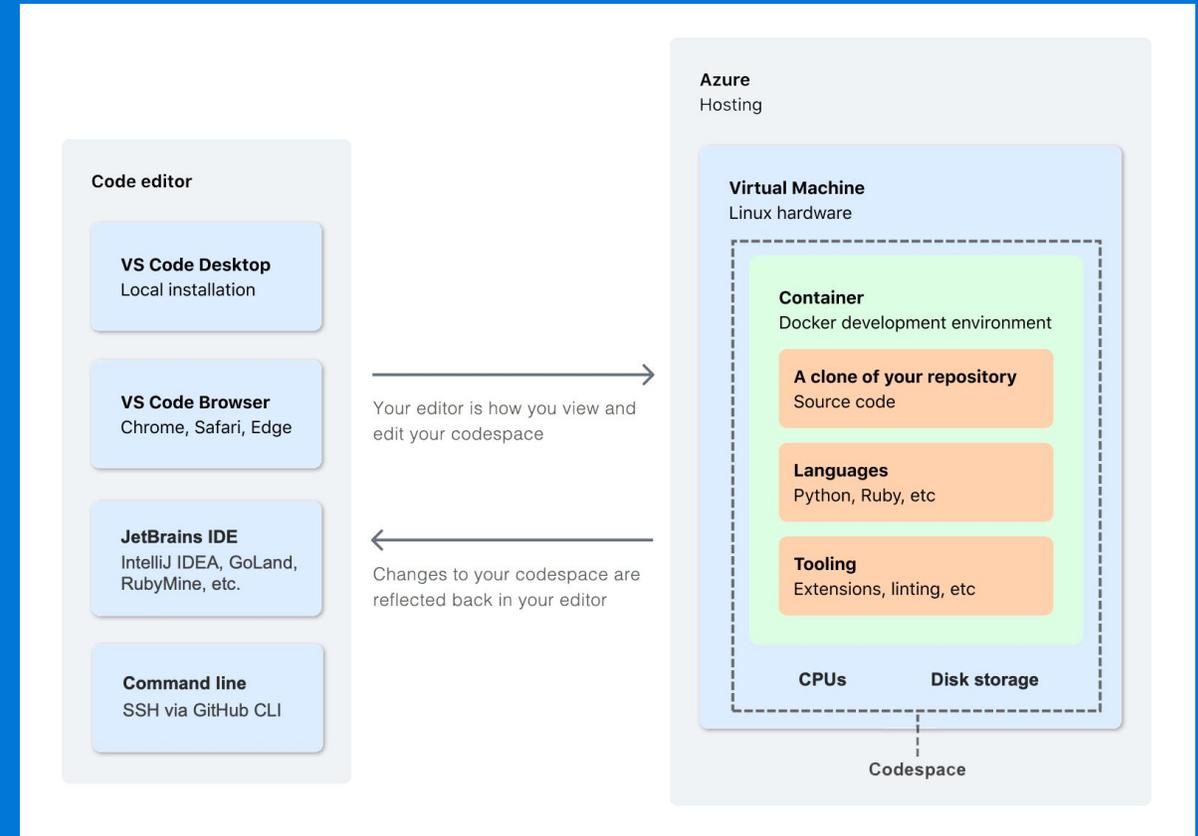
Se puede re abrir el entorno en un contenedor

Es recomendable la opción *Clone in Volume* para una mejor performance con archivos



GitHub codespaces

- Es un entorno de desarrollo remoto que se ejecuta en Azure (nube de Microsoft)
- Posee una interface web o también se puede conectar con varios IDEs o incluso via SSH
- La mayoría de entornos con devcontainers funcionan en codespaces





Educación pública, gratuita y de calidad