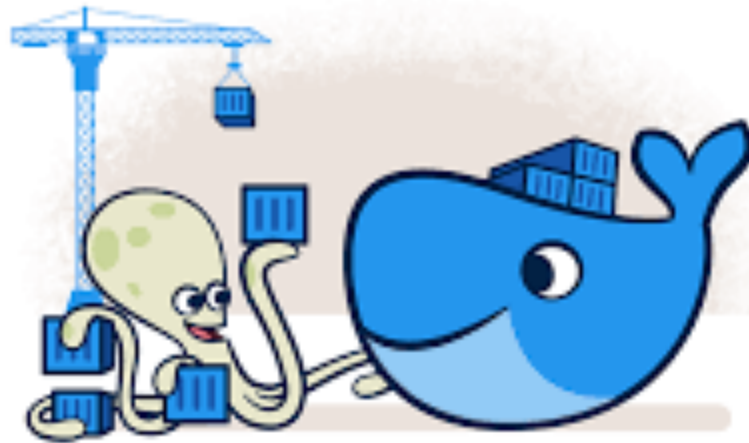


Arquitectura de Microservicios Usando Contenedores



Docker

- Plataforma para crear **containers de software**
- Estos containers empaquetan software en un sistema de archivos completo que contiene todo lo que necesita para ejecutarse: código, runtime, librerías. En fin cualquier componente que pueda ser instalado en un servidor.
- Esto garantiza el que software correrá siempre de la misma manera, independientemente de su ambiente.

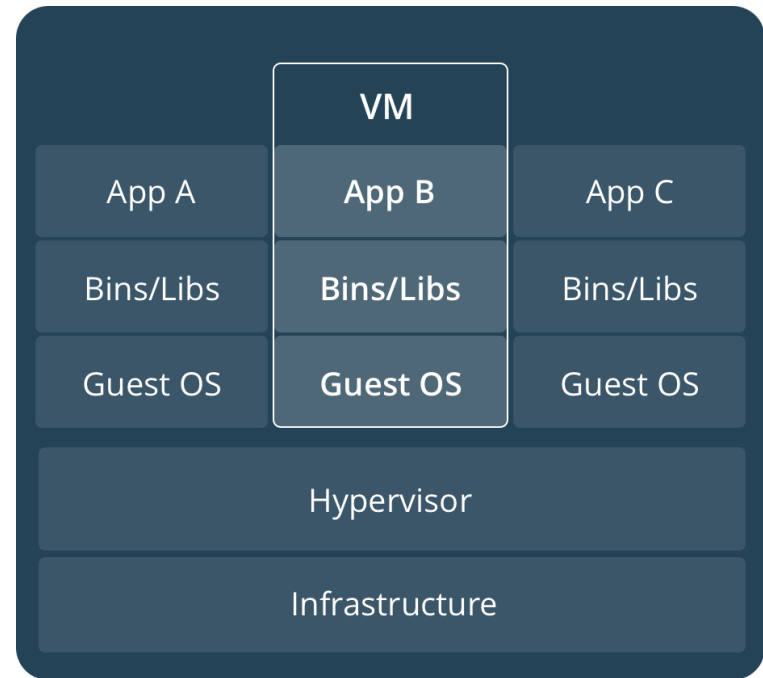
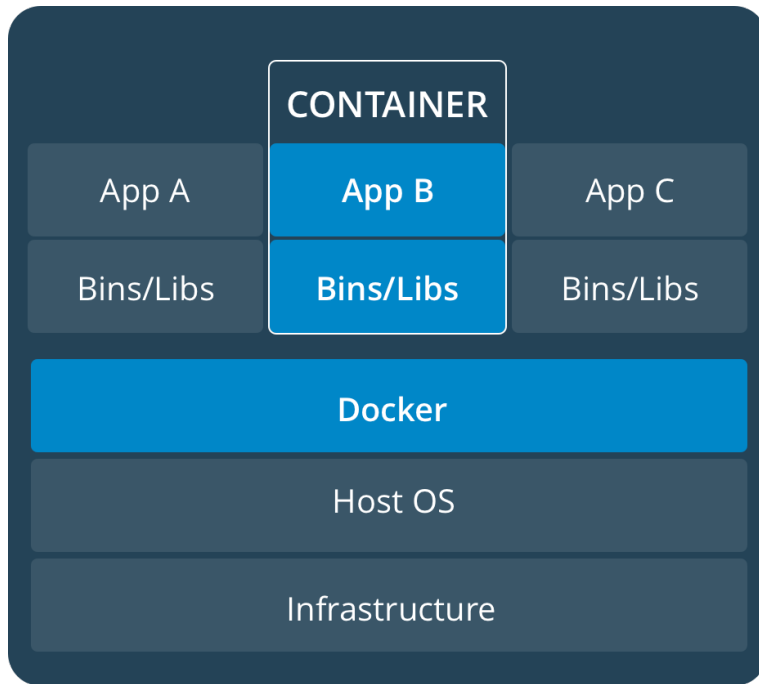
Por qué Docker?

- Liviano
 - Los containers que corren en una misma máquina comparten el mismo kernel de S.O. Y usan menos RAM
 - Las imágenes comparten archivos comunes en el sistema de archivos, por lo tanto el uso del disco es mas eficiente.
- Abierto
 - Basado en estándares abiertos
 - Por lo tanto los contenedores corren en la mayoría de distribuciones de Linux y Windows
- Seguridad por defecto
 - Los contenedores son componentes de software aislados entre ellos y aislados de la infraestructura que los soporta

Docker – Containers Vs. Maquinas Virtuales

- Maquinas Virtuales
 - Incluyen la aplicación, binarios y librerías
 - Incluyen el sistema operativo completo del invitado (guest)
 - Pueden ocupar decenas de GBs
- Containers
 - Incluyen la aplicación y todas sus dependencias
 - Comparten el kernel con otros contenedores
 - Cada container corre como un proceso aislado en “user space” en el sistema operativo host
 - Docker no esta ligado a una infraestructura especifica: Corren en cualquier computador, cualquier infraestructura y cualquier cloud

Docker – Containers Vs. Maquinas Virtuales



Ediciones de Docker / Imágenes

Personal

\$0

[Start Now](#)

Pro

\$5 /month ⓘ

[Select Pro](#)

Team

\$9 /user/month ⓘ

[Select Team](#)

Business

\$24 /user/month ⓘ

[Contact Sales](#)

Image Management

Public repositories

Unlimited

Unlimited

Unlimited

Unlimited

Private repositories

1

Unlimited

Unlimited

Unlimited

Advanced Image Management ⓘ

-



Ediciones de Docker / Componentes

	Personal	Pro	Team	Business
	Start Now	Select Pro	Select Team	Contact Sales
Docker Desktop				
Hardened Docker Desktop	-	-	-	✓
Docker Engine	✓	✓	✓	✓
Docker CLI	✓	✓	✓	✓
Docker Compose	✓	✓	✓	✓
Kubernetes	✓	✓	✓	✓
VDI support ⓘ	-	-	-	✓
Volume Management ⓘ	✓	✓	✓	✓
Docker Extensions ⓘ	✓	✓	✓	✓

Ediciones de Docker / Soporte

Personal

[Start Now](#)

Pro

[Select Pro](#)

Team

[Select Team](#)

Business

[Contact Sales](#)

Support

Community Forums

✓

✓

✓

✓

Community Slack

✓

✓

✓

✓

Commercial support

-

✓

✓

✓

Support for the latest version of Docker Desktop

-

✓

✓

✓

Support for Docker Desktop versions up to 6 months older than the latest ⓘ

-

-

-

✓

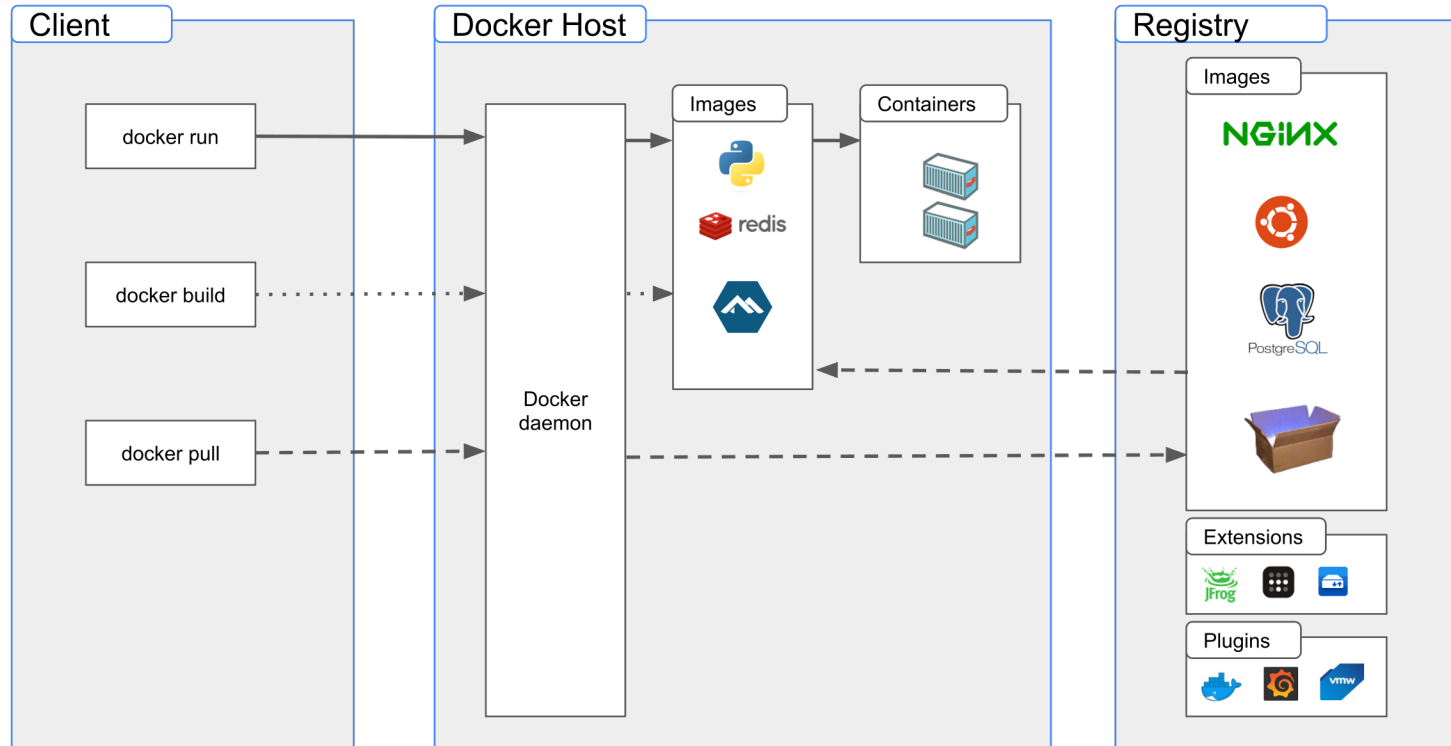
Plataforma Docker

- Docker Engine
- Docker Hub

Docker Engine

- Docker Daemon
 - Construcción de imágenes
 - Administración de contenedores
- Docker CLI
 - Cliente de línea de comandos

Arquitectura Docker



Fuente: <https://docs.docker.com/get-started/overview/>

Docker CLI

<code>docker build</code>	<code># Build an image from a Dockerfile</code>
<code>docker images</code>	<code># List all images on a Docker host</code>
<code>docker run</code>	<code># Run an image</code>
<code>docker ps</code>	<code># List all running and stopped instances</code>
<code>docker stop</code>	<code># Stop a running instances</code>
<code>docker rm</code>	<code># Remove an instance</code>
<code>docker rmi</code>	<code># Remove an image</code>

Docker Hub

- Provee servicios Docker
- Librería de imágenes publicas
- Almacenamiento de imágenes para desarrolladores: privadas o publicas

Docker Hub

The screenshot shows the Docker Hub website interface. At the top, there's a navigation bar with the Docker logo, a search bar, and links for Explore, Repositories, Organizations, and Help. A user profile for 'omondragon' is visible in the top right corner. Below the navigation bar, there's a search bar with 'omondragon' entered. A dropdown menu shows 'All Content'. A 'Create repository' button is present. The main content area displays three repositories for the user 'omondragon':

- omondragon / centosweb**
Contains: Image | Last pushed: 8 days ago
Inactive | 0 stars | 2 downloads | Public
- omondragon / kubermatic-dl**
Contains: Image | Last pushed: 10 months ago
Inactive | 0 stars | 3 downloads | Public
- omondragon / test-gha**
Contains: Image | Last pushed: a year ago
Inactive | 0 stars | 22 downloads | Public

On the right side, there's a promotional banner for creating an organization and managing repositories with a team. Below that, there's a 'community ALL-HANDS' banner featuring Docker characters.

Instalación Docker

```
$ sudo apt-get install docker-ce docker-ce-cli  
containerd.io
```

Workflow Básico

- Encontrar una imagen en Docker Hub
- Traer la imagen desde Docker Hub
- Correr la imagen en el host
- Parar la instancia
- Eliminar la instancia
- Eliminar la imagen

Workflow Básico

`docker version`

`docker info`

`docker search ubuntu`

`docker search --filter=stars=10 ubuntu`

`docker pull ubuntu`

`docker images`

`docker history ubuntu`

`cid=$(docker run -itd ubuntu)`

`echo $cid`

`docker ps`

`docker exec $cid pwd`

`docker exec $cid ls`

`docker stop $cid`

`docker rm $cid`

`docker rmi ubuntu`

`docker images`

Docker Compose

- Herramienta para definir y ejecutar aplicaciones Docker de contenedores múltiples
- Utiliza un archivo YAML para configurar los servicios de su aplicación
- Luego, con un solo comando, crea e inicia todos los servicios desde su configuración

Pasos para Usar Docker Compose

- Defina el entorno de su aplicación con un [Dockerfile](#)
- Defina los servicios que componen su aplicación en [docker-compose.yml](#) para que puedan ejecutarse juntos en un entorno aislado.
- Ejecute [docker-compose up](#) para que compose inicie y ejecute toda su aplicación.

Ejemplo docker-compose.yml

```
version: '2.0'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    links:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Referencias

- Docker: <https://www.docker.com/>
- Docker Compose: <https://docs.docker.com/compose/>
- Docker CLI:
<https://docs.docker.com/engine/reference/commandline>