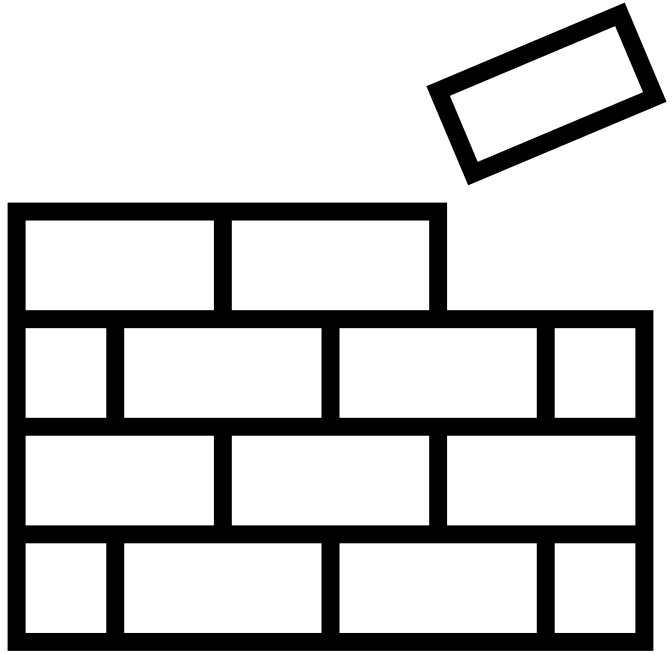
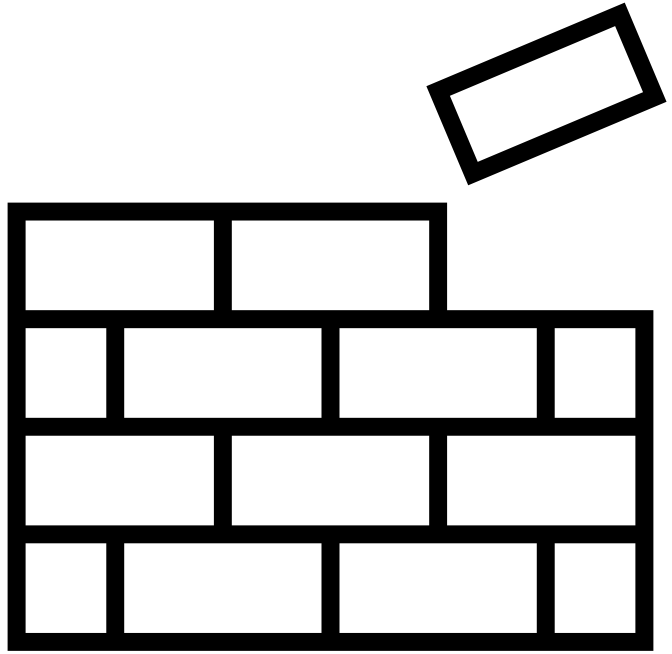


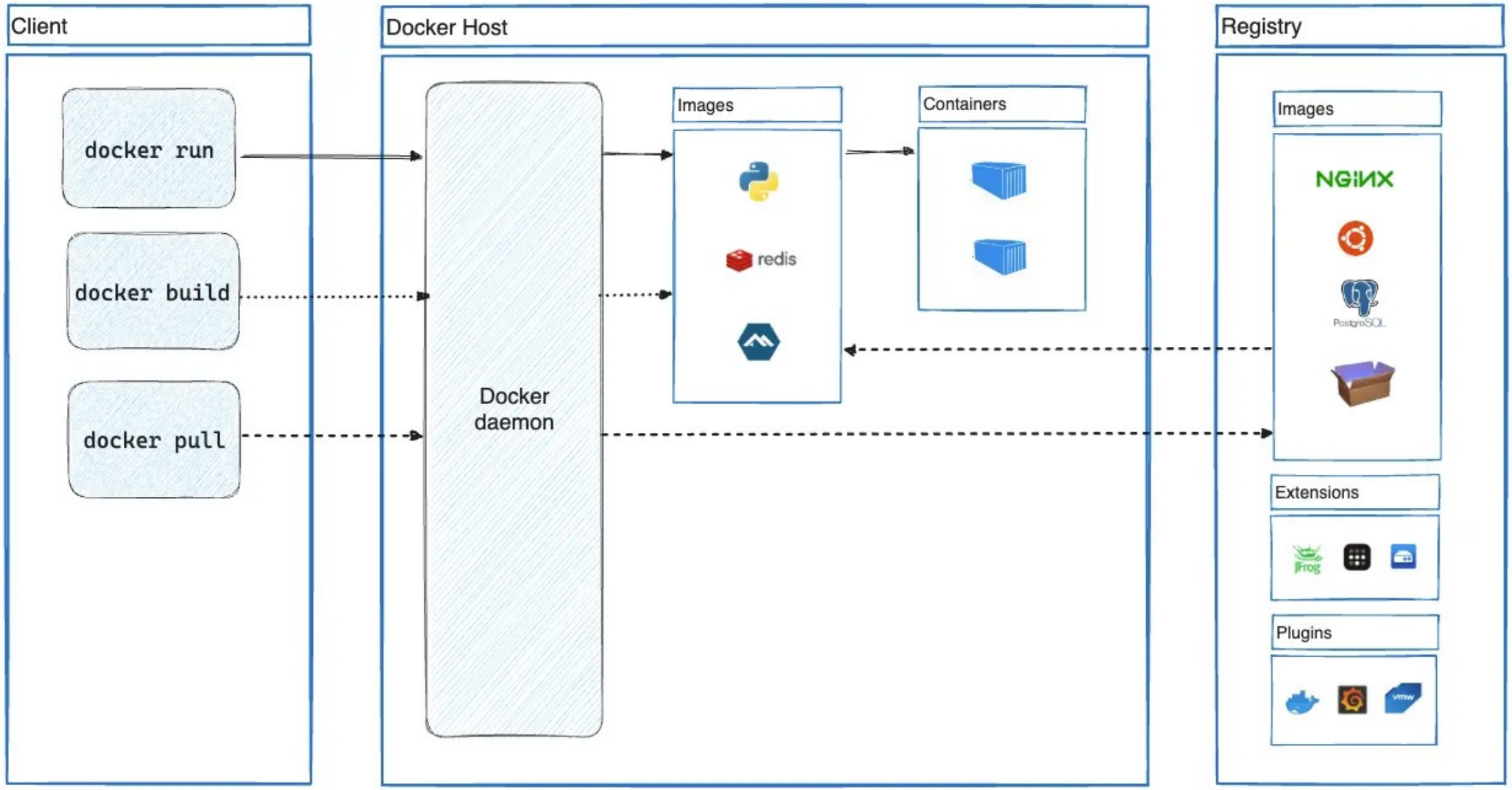
Arquitectura Docker

Índice

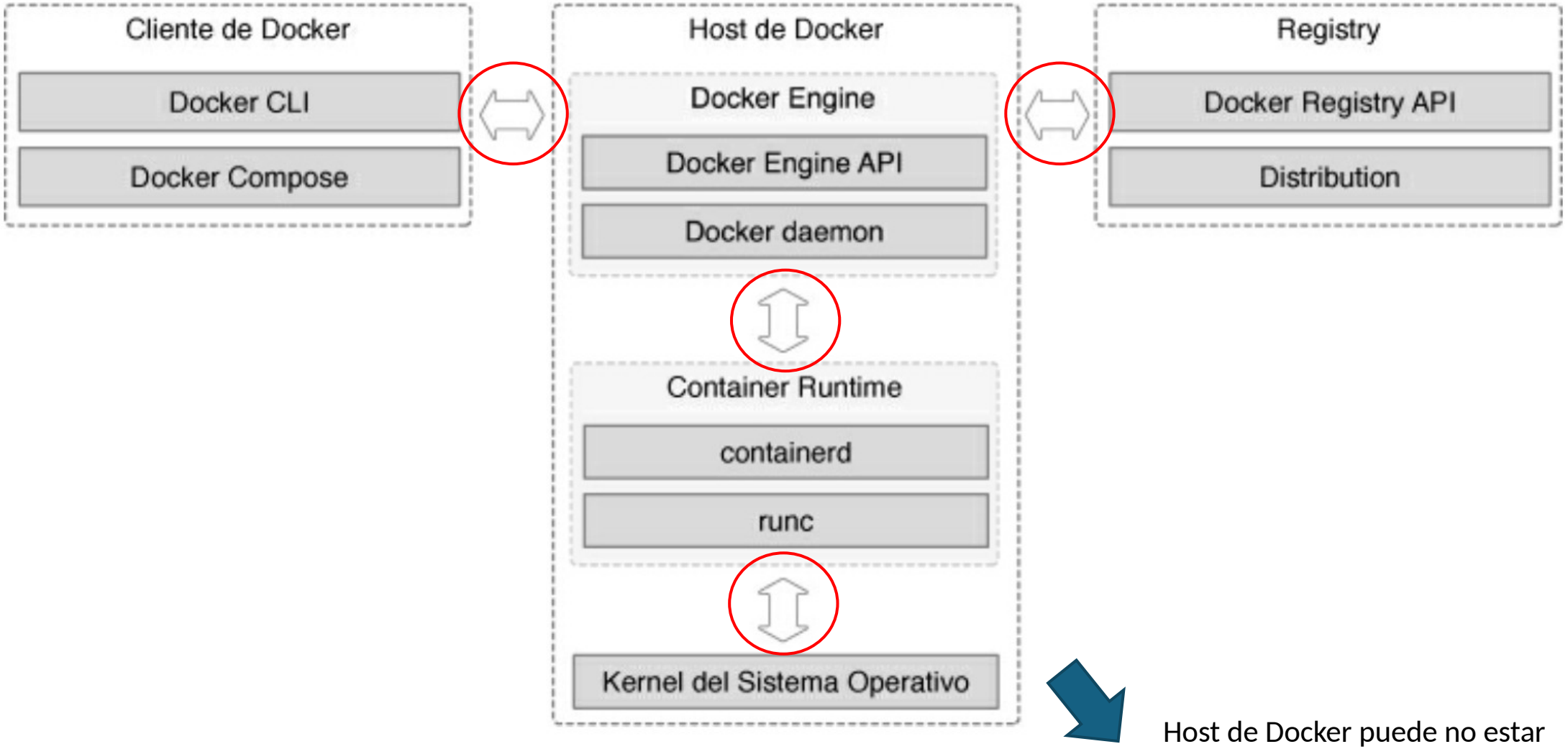


1. Introducción

1. Introducción



1. Introducción



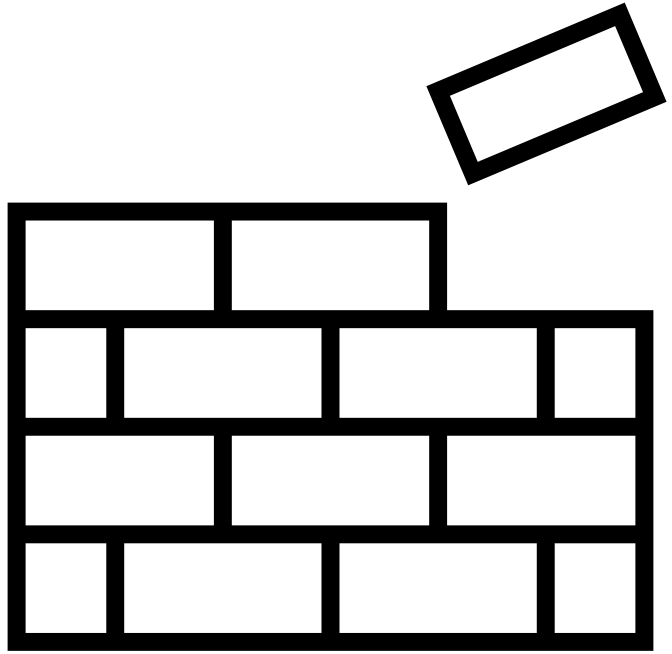
Host de Docker puede no estar en nuestra red local

1. Introducción

$ docker version

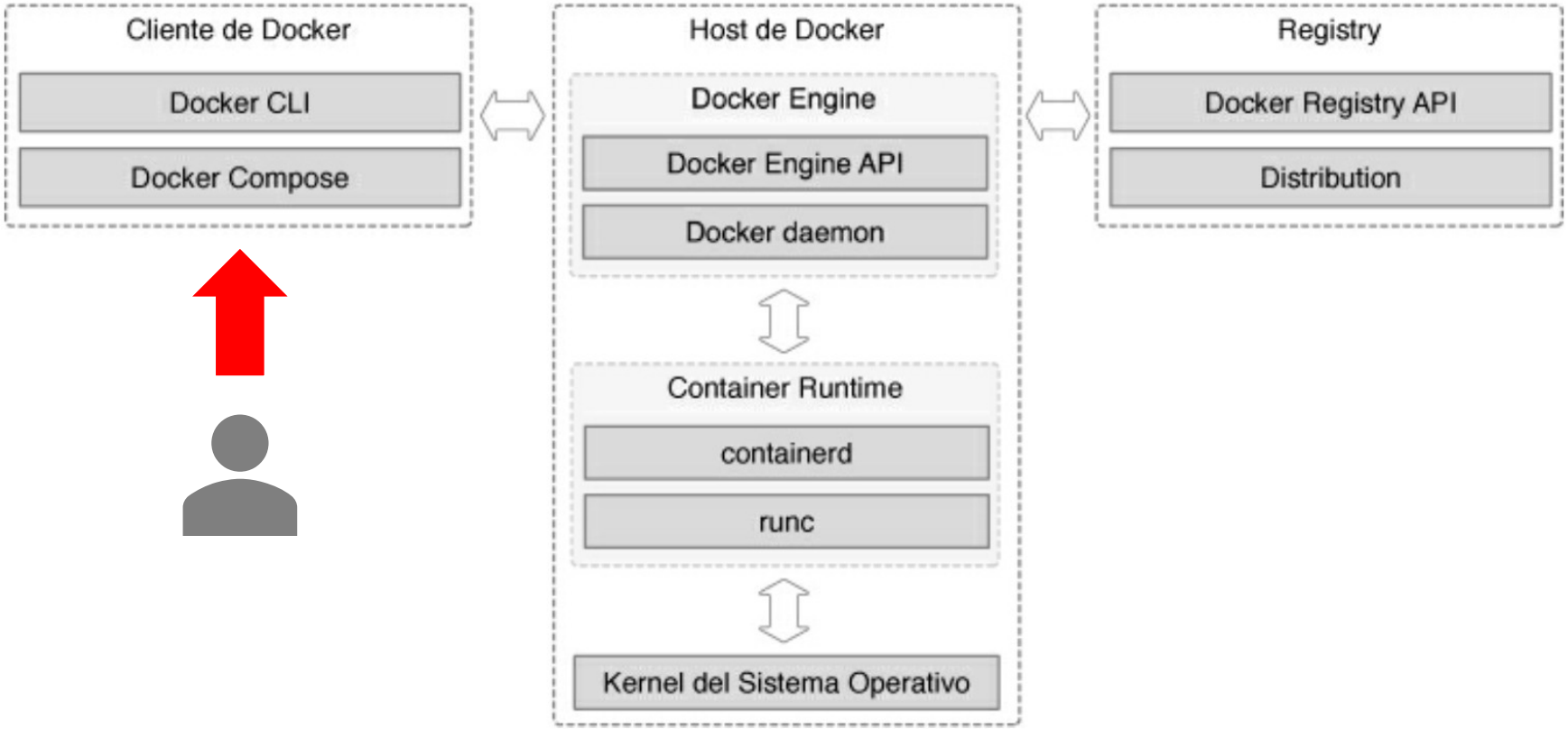
Client:
Cloud integration: 1.0.17
Version: 20.10.7
API version: 1.41
Go version: go1.16.4
Git commit: f0df350
Built: Wed Jun 2 11:56 OS/Arch: darwin/arm64

Server: Docker Engine
Engine:
Version: 20.10.7
API version: 1.41
Go version: go1.13.15
Git commit: b0f5bc3
Built: Wed Jun 2 11:55:36
OS/Arch: linux/arm64
containerd:
Version: 1.4.6
GitCommit: d71fcd7d
runc:
Version: 1.0.0-rc95
GitCommit: b9ee9c
docker-init:
Version: 0.19.0



2. Cliente Docker

2. Cliente Docker



2. Cliente Docker

- Es la capa más cercana al usuario.
- Permite interaccionar Host de Docker de una manera sencilla.
- Existen diferentes CLI:
  - Docker CLI: un único contenedor.
  - Docker Compose: varios contenedores.
  - SDK: automatizar procesos.
- Docker Cheatsheet

2. Cliente Docker

Ejemplo de código para SDK Python

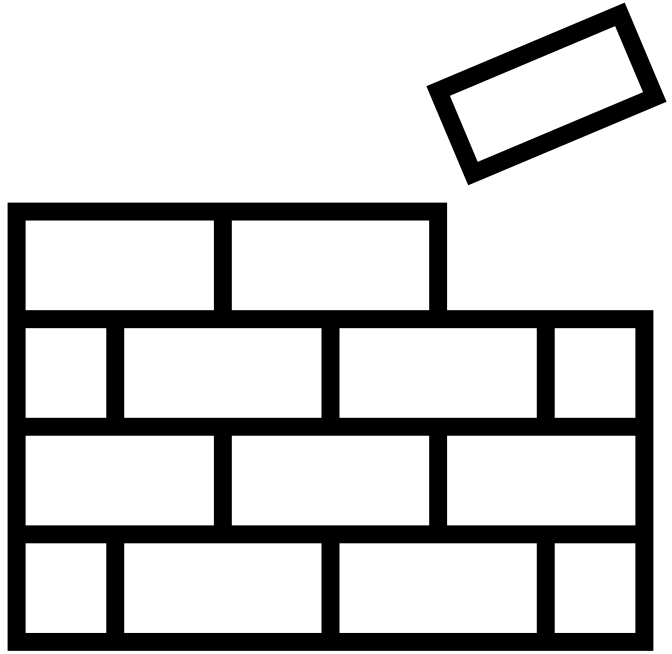
```
Import docker

Client = docker.from_env()

Print(client.containers.run("alpine", ["echo","hello","world"]))
```

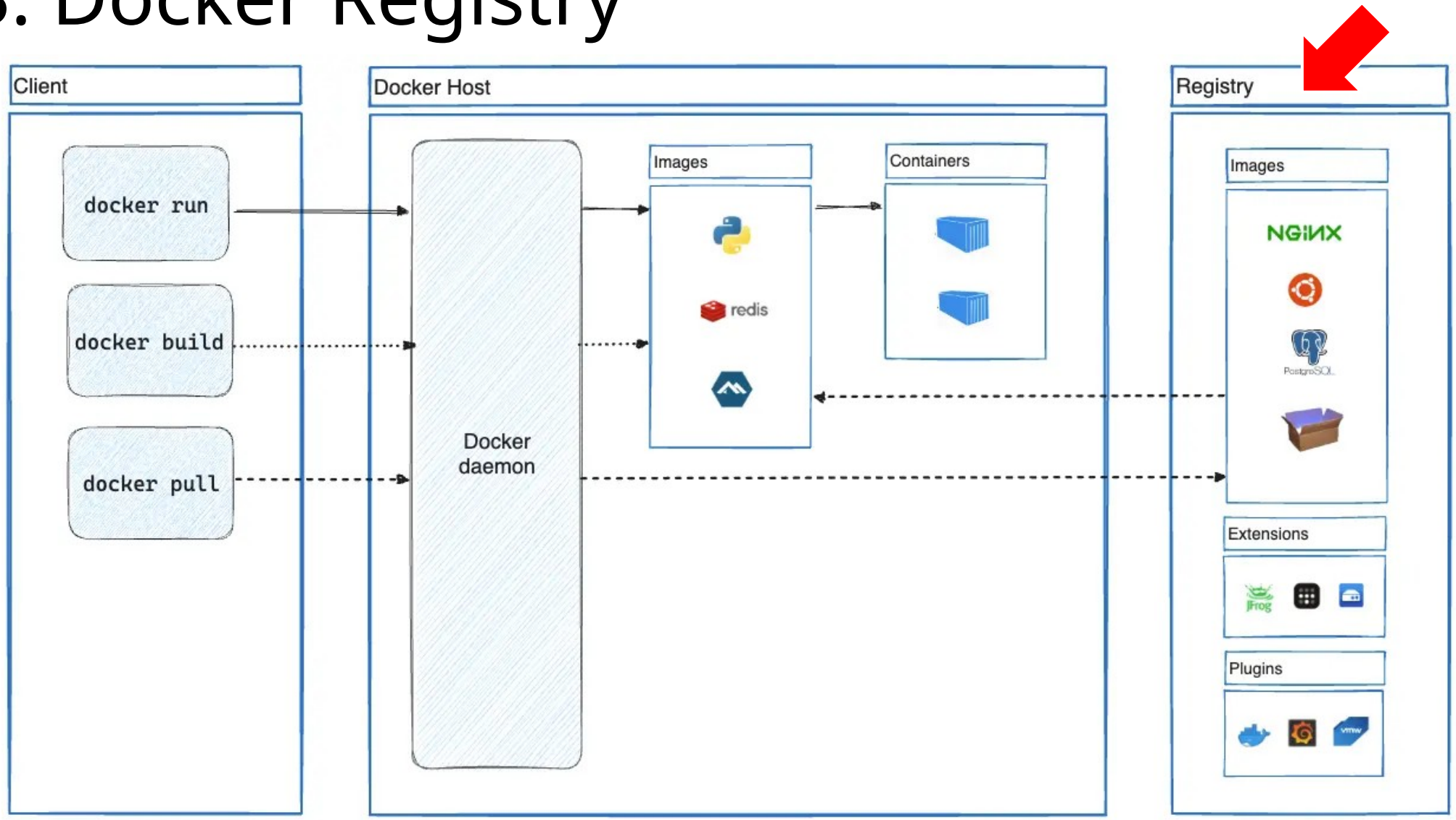
¿Cómo se traduce a Docker CLI?

docker run alpine echo hello world



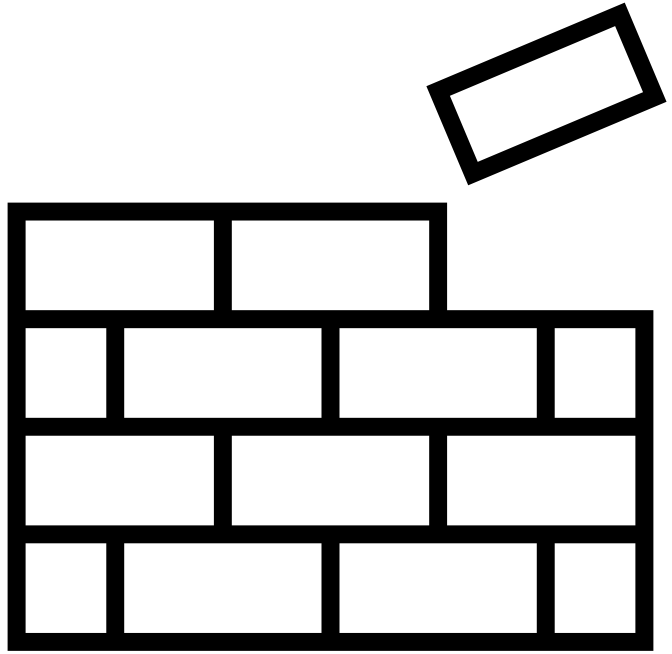
3. Docker Registry

3. Docker Registry



3. Docker Registry

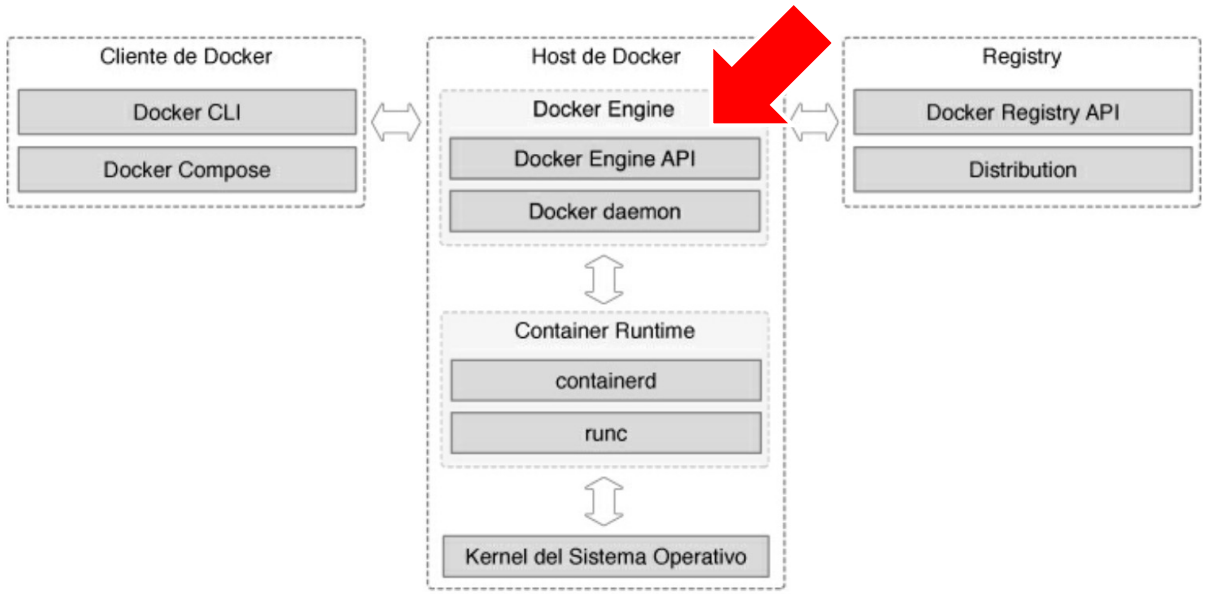
- Servicio encargado de almacenar y distribuir imágenes Docker.
- Servicios Docker Registry:
  - https://hub.docker.com/
  - GitHub, GitLab, Amazon Elastic Container Registry

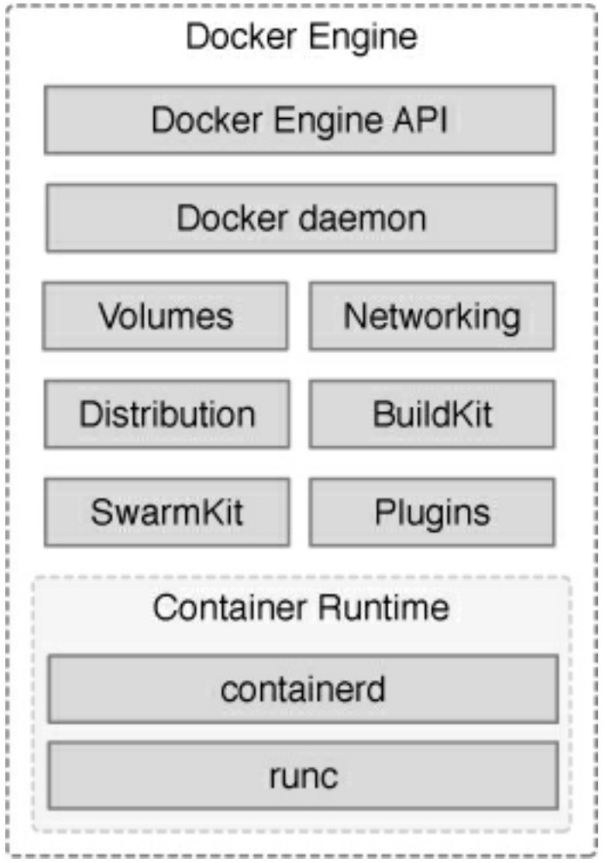


4. Host de Docker

4. Host de Docker

Docker Engine





4. Host de Docker

Docker Engine

- Es el componente principal de Docker.
- Tiene un diseño modular formado por varios componentes.
- Se ejecuta de forma nativa en los sistemas GNU/Linux y Windows Server*.
- En el resto de los sistemas operativos se ejecuta sobre una máquina virtual GNU/Linux.

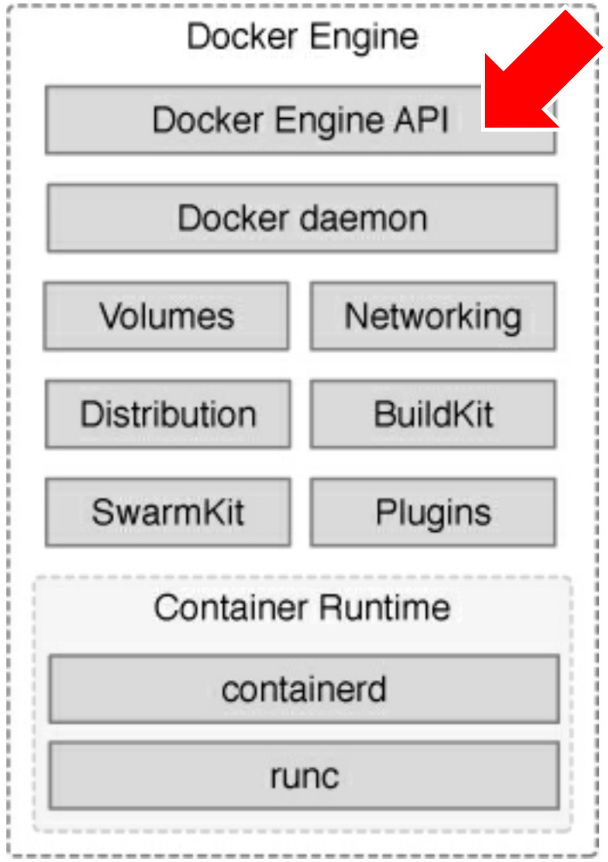
4. Host de Docker

Docker Engine API

- Expone dockerd a través de una API
- Esta API implementa todas las operaciones que un usuario puede realizar desde Docker CLI.

$ docker ps

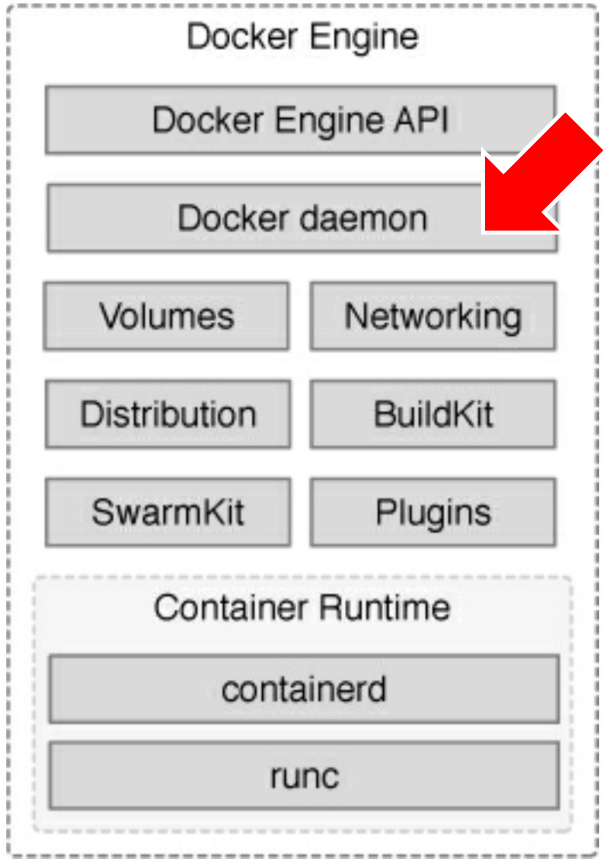
GET /v1.41/containers/json



4. Host de Docker

Docker daemon

- Crea y gestiona los objetos Docker.
- Servicio llamado dockerd.
- dockerd se expone a través de:
  - Sockets
    /var/run/docker.sock
  - TCP
    dockerd -H tcp://0.0.0.0:2375
  - File Descriptor (fd)
    dockerd -H fd://



4. Host de Docker

Docker daemon

¿QUÉ IMPLICA EXPONER DOCKERD?

Otros procesos pueden ejecutar y gestionar contenedores, imágenes y redes.

4. Host de Docker

Docker daemon

¿POR QUÉ LO HACEMOS?

- Monitorización

docker run -d -v /var/run/docker.sock:/var/run/docker.sock google/cadvisor

- CI/CD - Docker in Docker (dind):

docker run -it -v /var/run/docker.sock:/var/run/docker.sock docker sh

- Orquestación

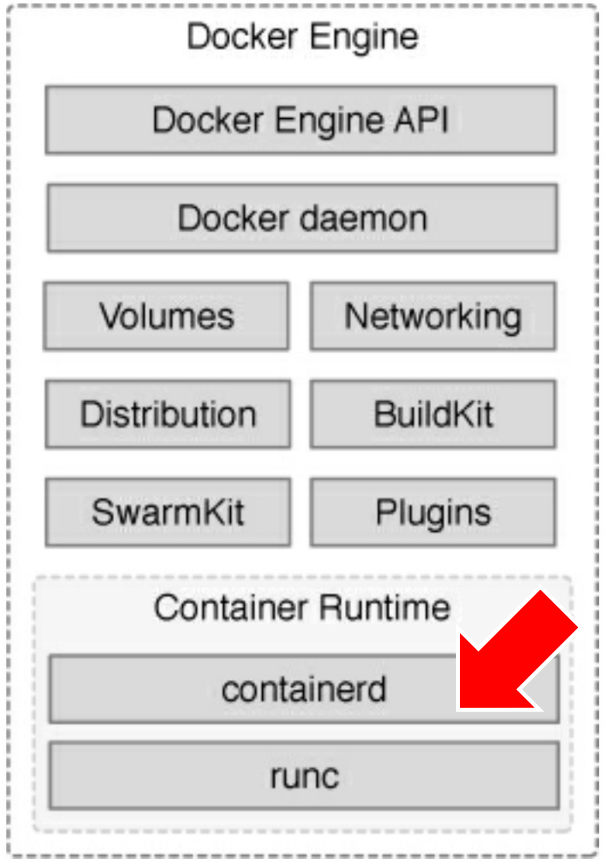
docker run -d -v /var/run/docker.sock:/var/run/docker.sock portainer/portainer

- Tener Docker Host en diferente máquina que Docker CLI

4. Host de Docker

containerd

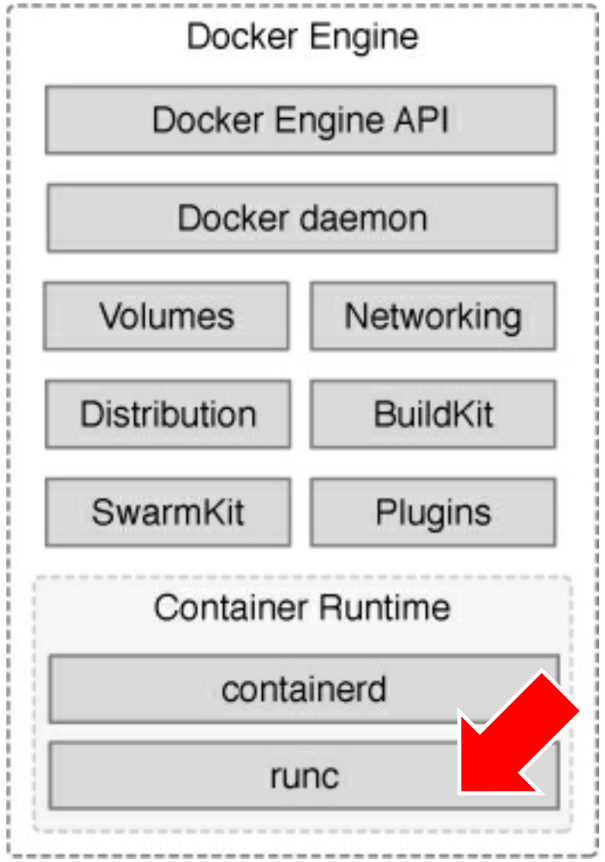
- Proporciona una plataforma para gestionar contenedores a nivel de sistema operativo.
- Se trata de un proyecto open source, que fue creado por Docker, Inc. en 2016, junto a Google e IBM.
- En 2017, fue donado a la Cloud Native Computing Foundation (CNCF) y, en 2019, se convirtió en un proyecto graduado de la CNCF.



4. Host de Docker

runc

- Encargado de interaccionar con el kernel del host de la máquina anfitriona donde se ejecutan los contenedores. Low level Runtime.
- Utiliza el componente libcontainer para interaccionar con el sistema operativo del host.
- Se trata de una implementación de código abierto de la especificación OCI Runtime Specification, que describe cómo tiene que ser la configuración, el entorno de ejecución y el ciclo de vida de un contenedor.



¿Dudas?