

Docker Instalación

Windows



Docker en Windows tiene algunos requisitos previos que deben cumplirse antes de puede instalarlo. Éstas incluyen:

- Compatibilidad con Hyper-V
- Compatibilidad con virtualización de hardware: suele ser habilitado desde el BIOS de su sistema
- Solo las ediciones de 64 bits de Windows 10 (Pro/Education/Ediciones empresariales con la actualización de aniversario v1607) son compatibles en este momento

<https://docs.docker.com/desktop/install/windows-install/>

WSL-2

El subsistema para Linux (WSL) es una forma para que los desarrolladores ejecuten aplicaciones GNU/Linux desde Windows sin la configuración de una máquina virtual de terceros o sin tener que realizar un arranque dual en Linux.

WSL es compatible con la mayoría de las aplicaciones de línea de comandos y la compatibilidad con aplicaciones GUI aún se encuentra en modo de vista previa temprana

<https://learn.microsoft.com/en-us/windows/wsl/install>

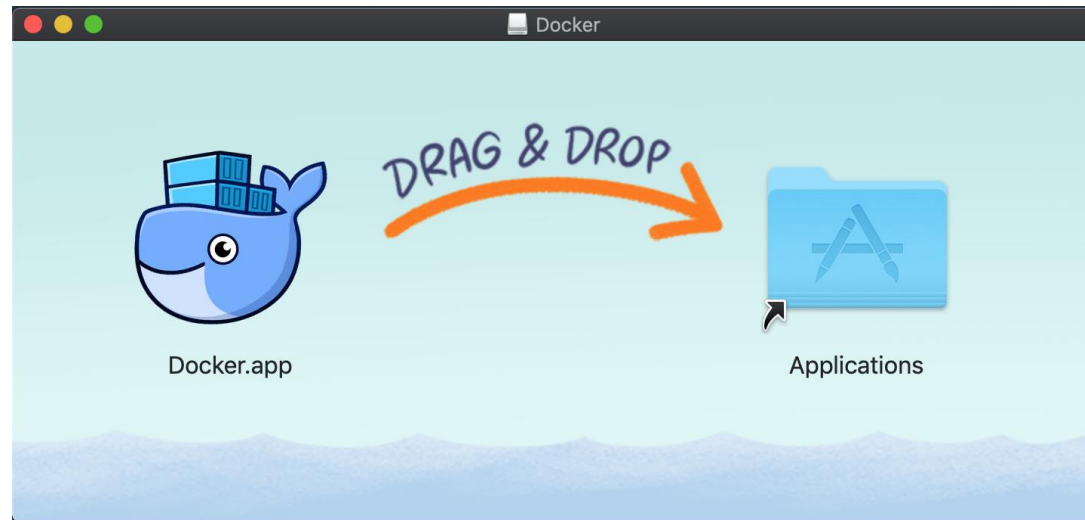
MacOS

Instalar Docker para Mac es muy parecido a instalar cualquier otra aplicación. Ves a:

<https://docs.docker.com/desktop/install/mac-install/>

Haz click en el enlace Obtener Docker para CE Mac (estable)

Haz click en el archivo para ejecutar el instalador que se descargó.



Linux

1. `sudo apt-get update`
 2. `sudo apt-get install apt-transport-https ca-certificates curl software-properties-common`
 3. `curl -fsSL https://download.docker.com/linux/ubuntu/gpg`
`| sudo apt-key add -`
 4. `sudo add-apt-repository "deb [arch=amd64 https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"`
 5. `sudo apt-get update`
 6. `sudo apt-get install docker-ce`
- # Algunas veces es necesario
7. `sudo groupadd docker`
 8. `sudo usermod -aG docker $USER`

Docker “Jerga”

Layers

Una capa (layer) es una modificación aplicada a una imagen de Docker representada por una instrucción en un Dockerfile.

FROM ubuntu

Run mkdir /tmp/logs

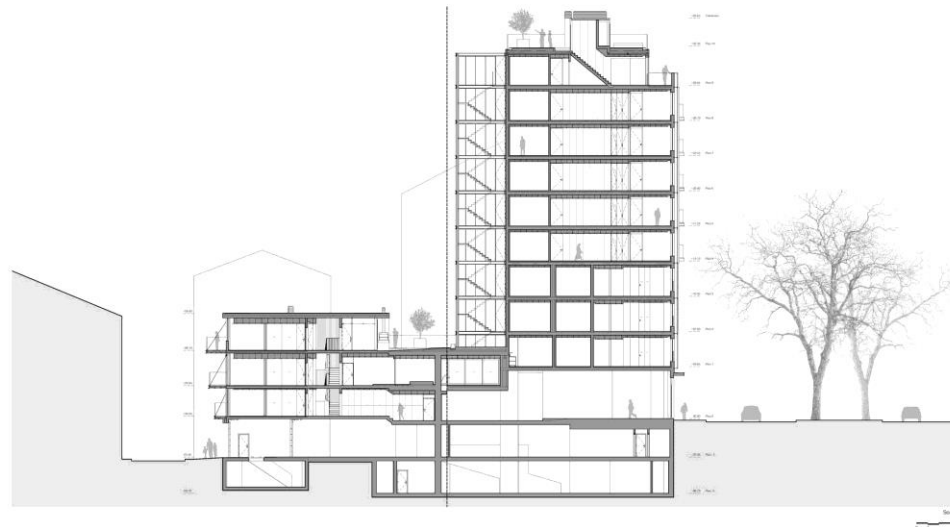
RUN apt-get install vim

RUN apt-get install htop

Docker image

La imagen de Docker es una plantilla de solo lectura que forma la base de su aplicación. Es muy parecido a un script de shell que prepara un sistema con el estado deseado.

En términos más simples, es equivalente a una receta de cocina que tiene instrucciones paso a paso para crear el plato final.



Docker tags

Una etiqueta es un nombre que identifica de forma única una versión específica de una imagen de Docker. Las etiquetas son etiquetas de texto sin formato, que a menudo se usan para identificar detalles específicos, como la versión, el sistema operativo base de la imagen o la arquitectura de la imagen de Docker.

```
docker pull ubuntu:latest
```

```
docker pull ubuntu:20.04
```

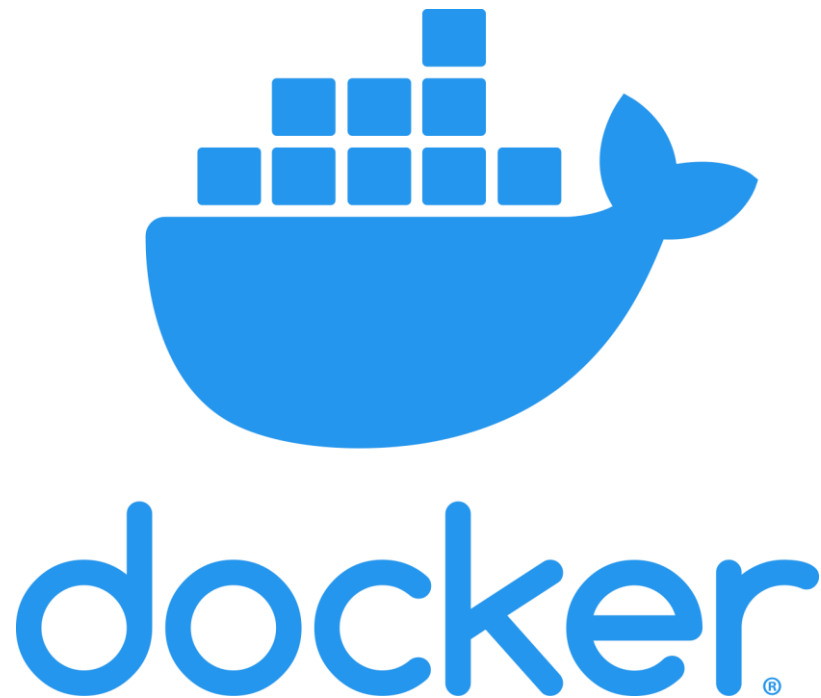
Docker containers

La imagen de Docker, cuando se ejecuta en una computadora host, genera un proceso con su propio espacio de nombres y se conoce como contenedor de Docker.



Docker repository

Un repositorio de Docker es un lugar donde puedes cargar y almacenar imágenes de Docker. Estos repositorios permiten una fácil distribución de las imágenes de Docker dentro de su empresa o con el público.



Docker registry

Un Docker Registry es una colección de varios repositorios de Docker. Los registros de Docker están alojados por empresas de terceros, o puede alojarlos usted mismo si necesita cumplir con requisitos de cumplimiento más estrictos. Ejemplos

- Google Container Registry
- Amazon Elastic Container Registry
- JFrog Artifactory

Dockerfile

Un Dockerfile es un conjunto de instrucciones que le dice a Docker cómo crear una imagen.

- FROM
- ENV
- RUN
- CMD o ENTRYPOINT

Dockerfile

```
FROM python:3.9-slim
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 5000
CMD [ "python", "app.py" ]
```

Dockerengine

El motor Docker es una parte central de Docker. Docker Engine es una aplicación cliente-servidor que proporciona la plataforma, el tiempo de ejecución y las herramientas para crear y administrar imágenes de Docker, contenedores de Docker y mucho más. Cuenta con el:

- Docker Deamon
- Docker CLI
- Docker API

Dockerengine

- **Docker Deamon:** ejecuta en segundo plano en la computadora host y maneja el trabajo pesado de la mayoría de los comandos de Docker.
- **Docker CLI:** La CLI de Docker es la principal forma de interactuar con Docker. (docker pull, docker build, docker run, docker exec)
- **Docker API:** es una forma de interactuar con Docker engine

Docker Compose

Docker Compose es una herramienta para definir y ejecutar aplicaciones de varios contenedores.

Al igual que Docker le permite crear una imagen para su aplicación y ejecutarla en su contenedor, Compose usa las mismas imágenes en combinación con un archivo de definición (conocido como el archivo de redacción usualmente `.yaml`) para crear, iniciar y ejecutar aplicaciones de varios contenedores. incluidos los contenedores dependientes y vinculados.