



# DOCKER

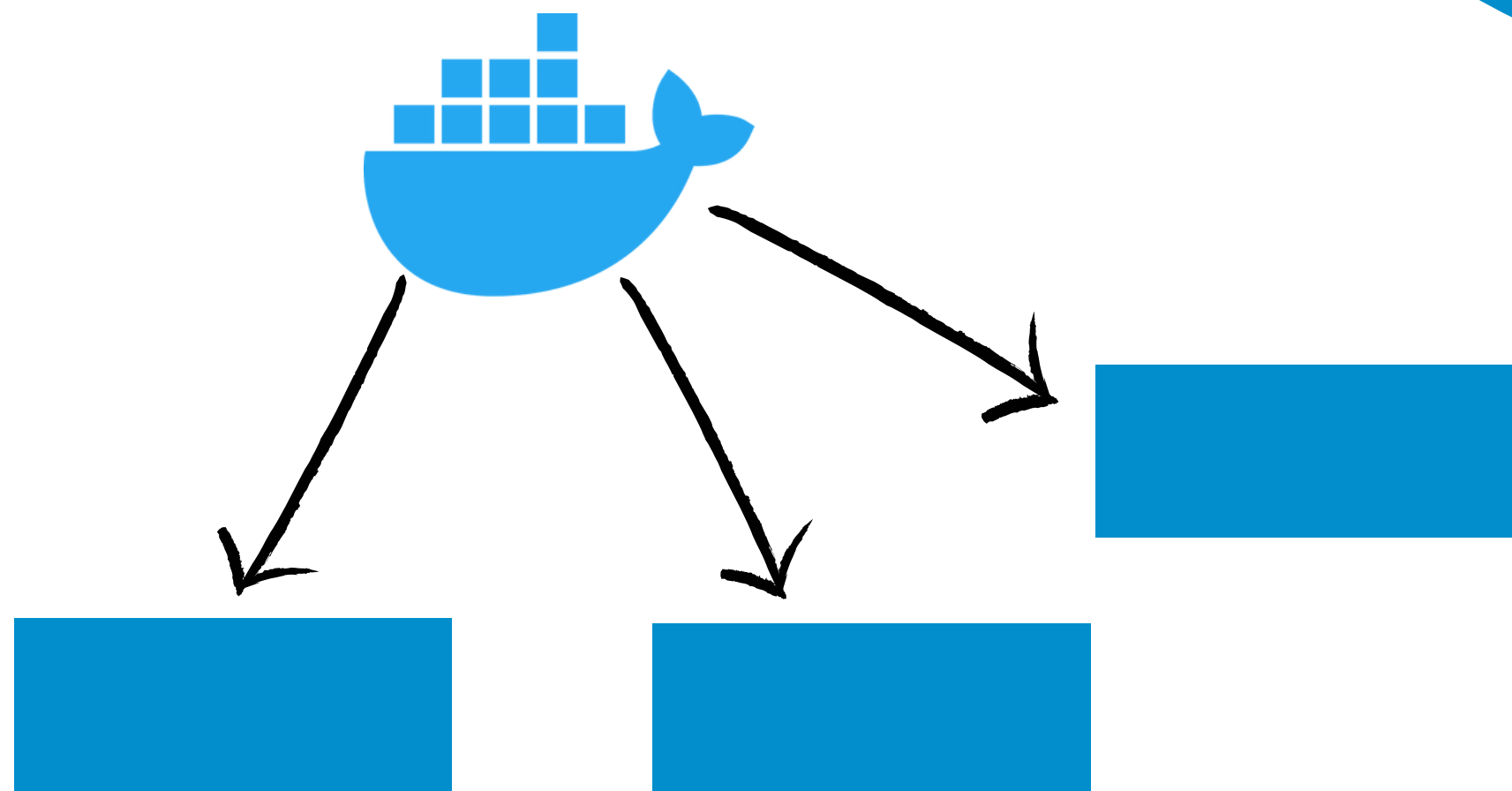


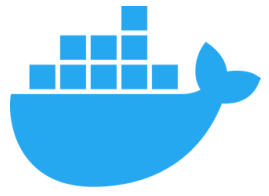


# ¿QUÉ ES DOCKER?

Es una tecnología que nos permite desarrollar, desplegar y ejecutar aplicaciones en un entorno aislado que llamamos contenedores.

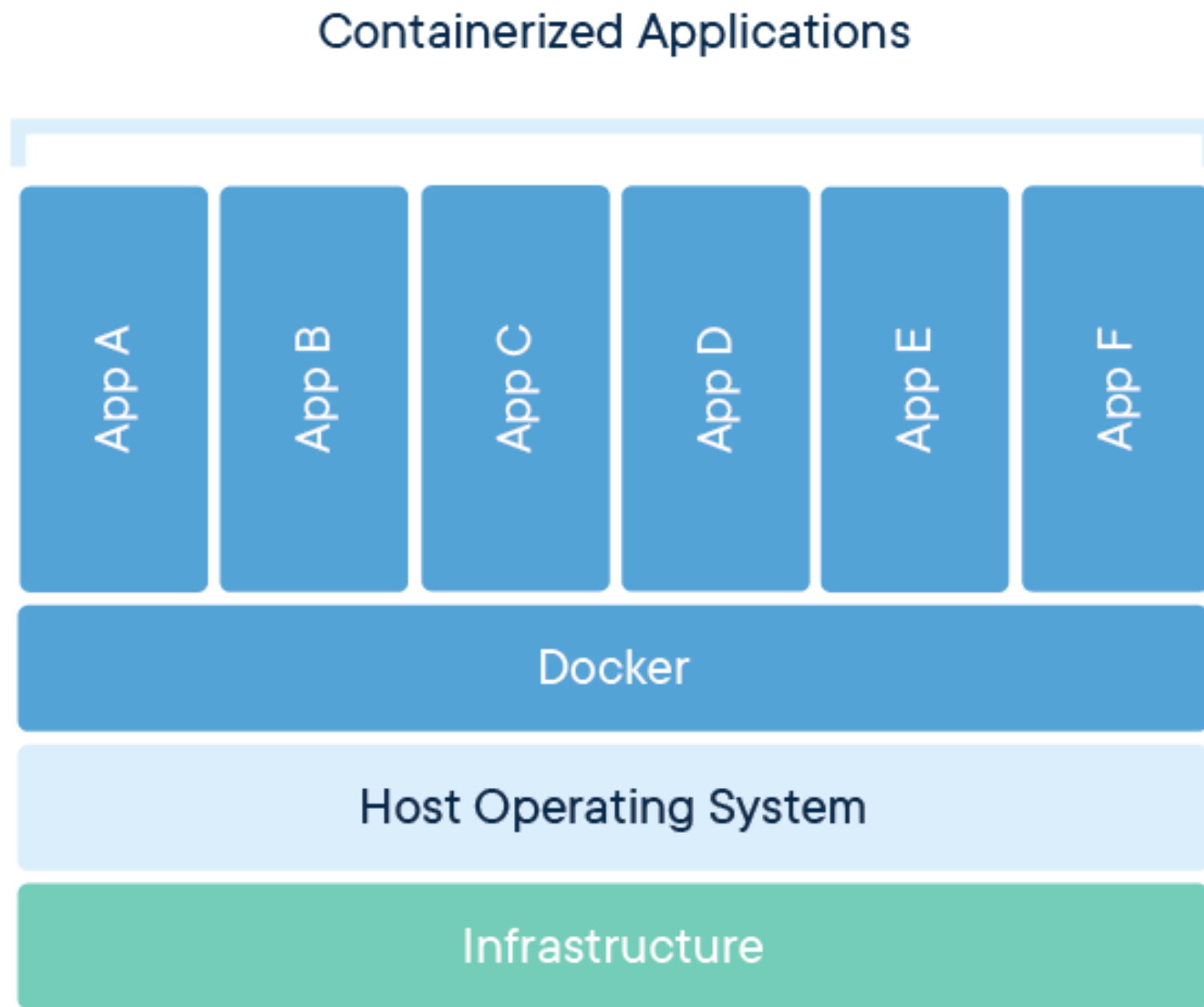
Permite empaquetar todos los recursos que necesita una aplicación para que la ejecución de la aplicación no dependa de la máquina en la que se va a ejecutar o del software que tenga instalado.



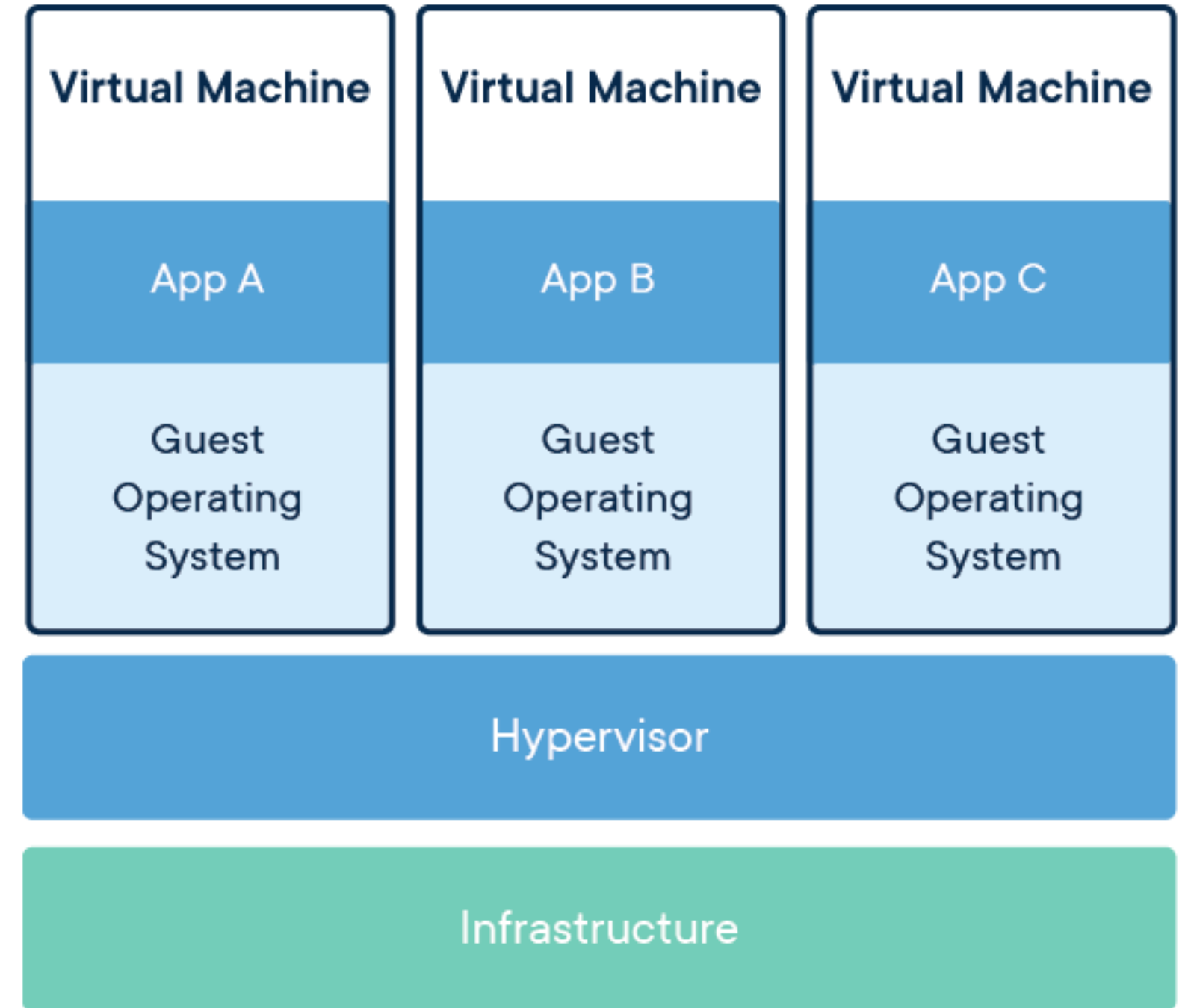


# DOCKER VS MÁQUINA VIRTUAL

## DOCKER



## MÁQUINAS VIRTUALES



# PRINCIPALES VENTAJAS

1

## **ESTANDARIZACIÓN Y PRODUCTIVIDAD**

Contenedores livianos que facilitan su almacenamiento, transporte y despliegue

2

## **COMPATIBILIDAD Y MANTENIBILIDAD**

Aplicaciones libres de dependencias instaladas en el sistema anfitrión

3

## **SIMPLICIDAD Y AHORRO DE COSTES**

Capacidad de desplegar múltiples contenedores en un mismo equipo físico

# COMPONENTES DE DOCKER

## Cliente del docker

- Docker CLI
- Docker Compose

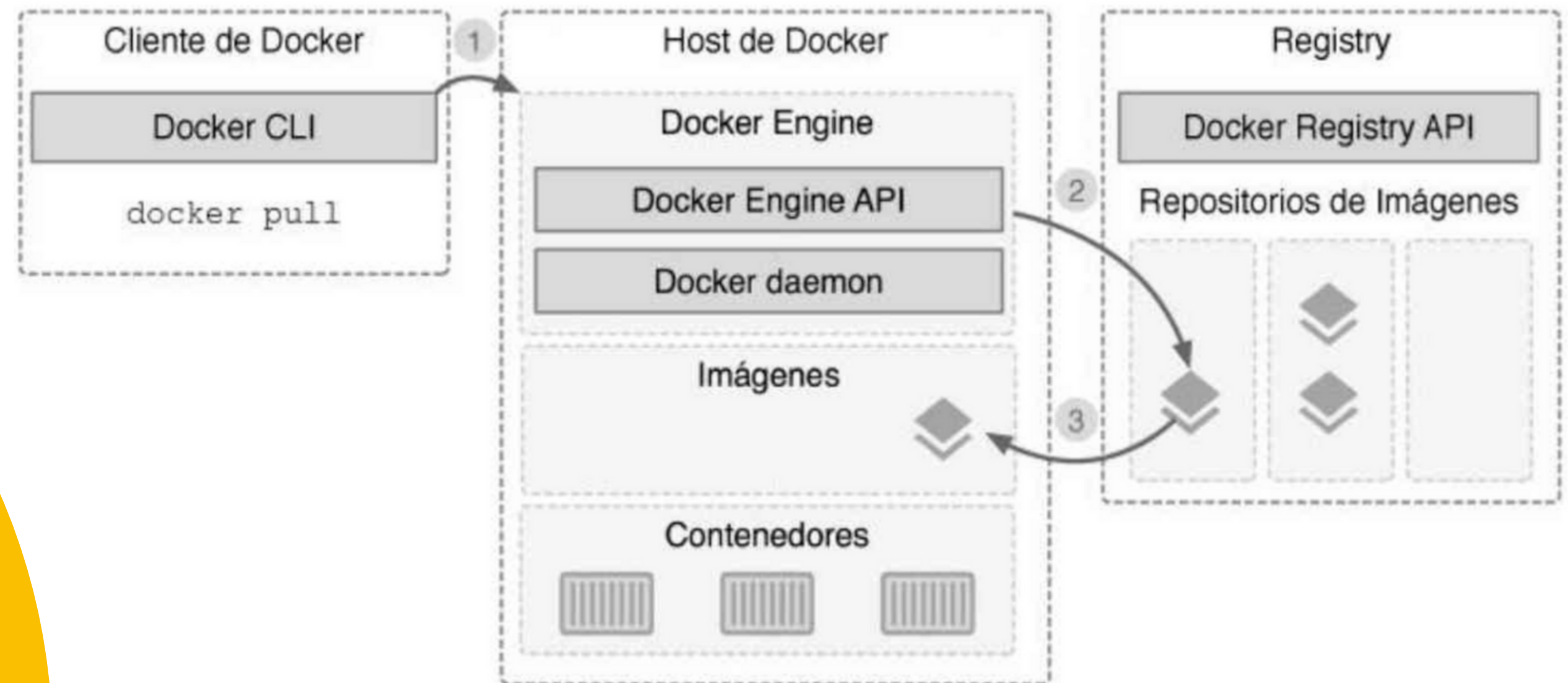
## Docker engine

- Docker CLI
- Docker Compose

## Container Runtime

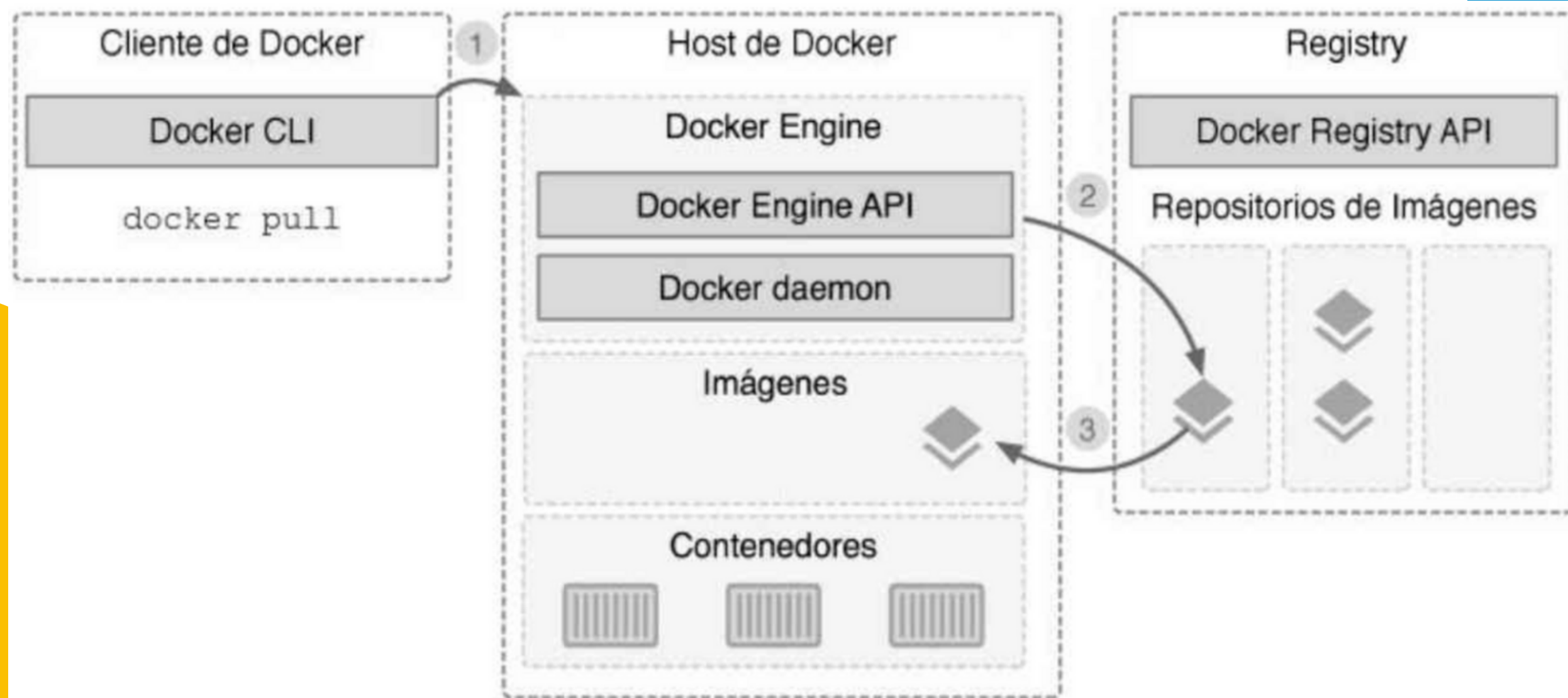
- Containerd
- Run

## Docker registry



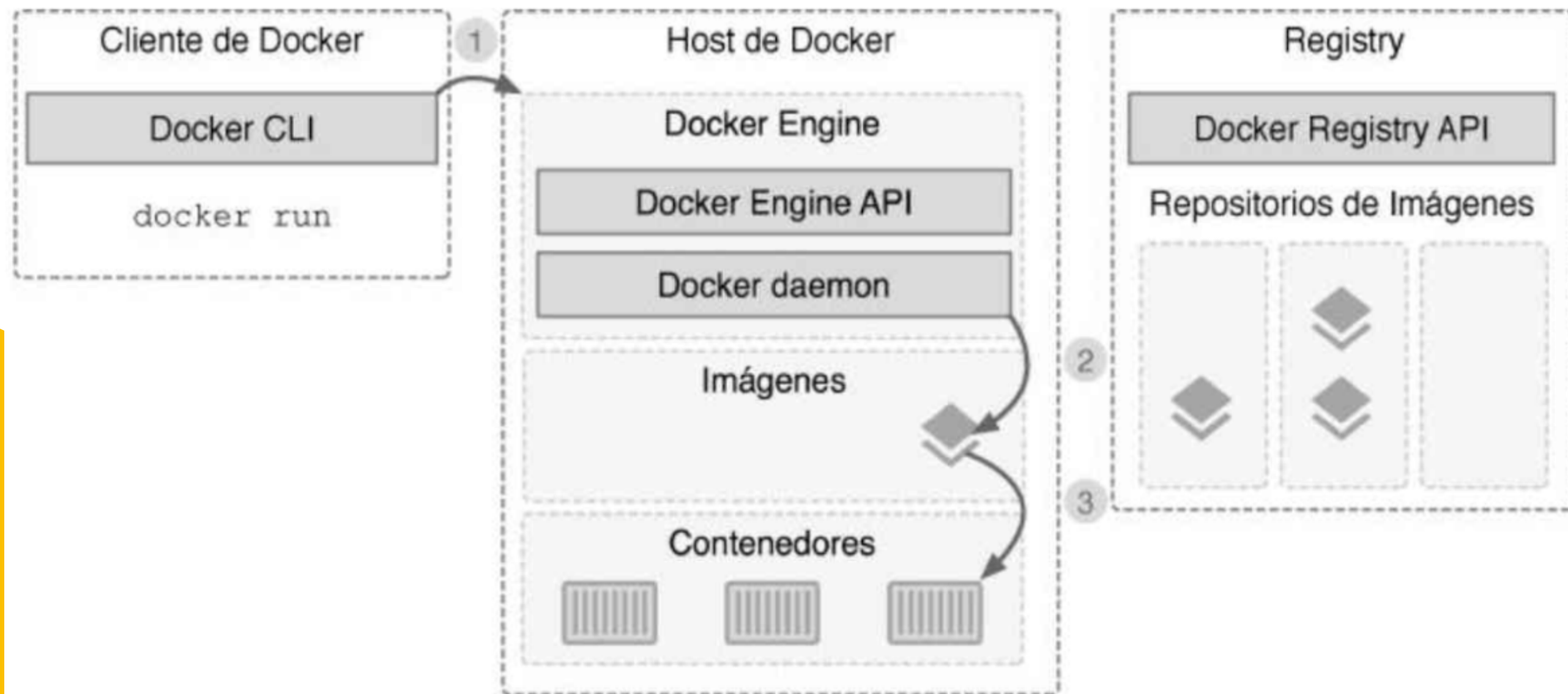
# DOCKER PULL

Ejemplo funcionamiento del comando docker pull



# DOCKER RUN

Ejemplo funcionamiento del comando docker run



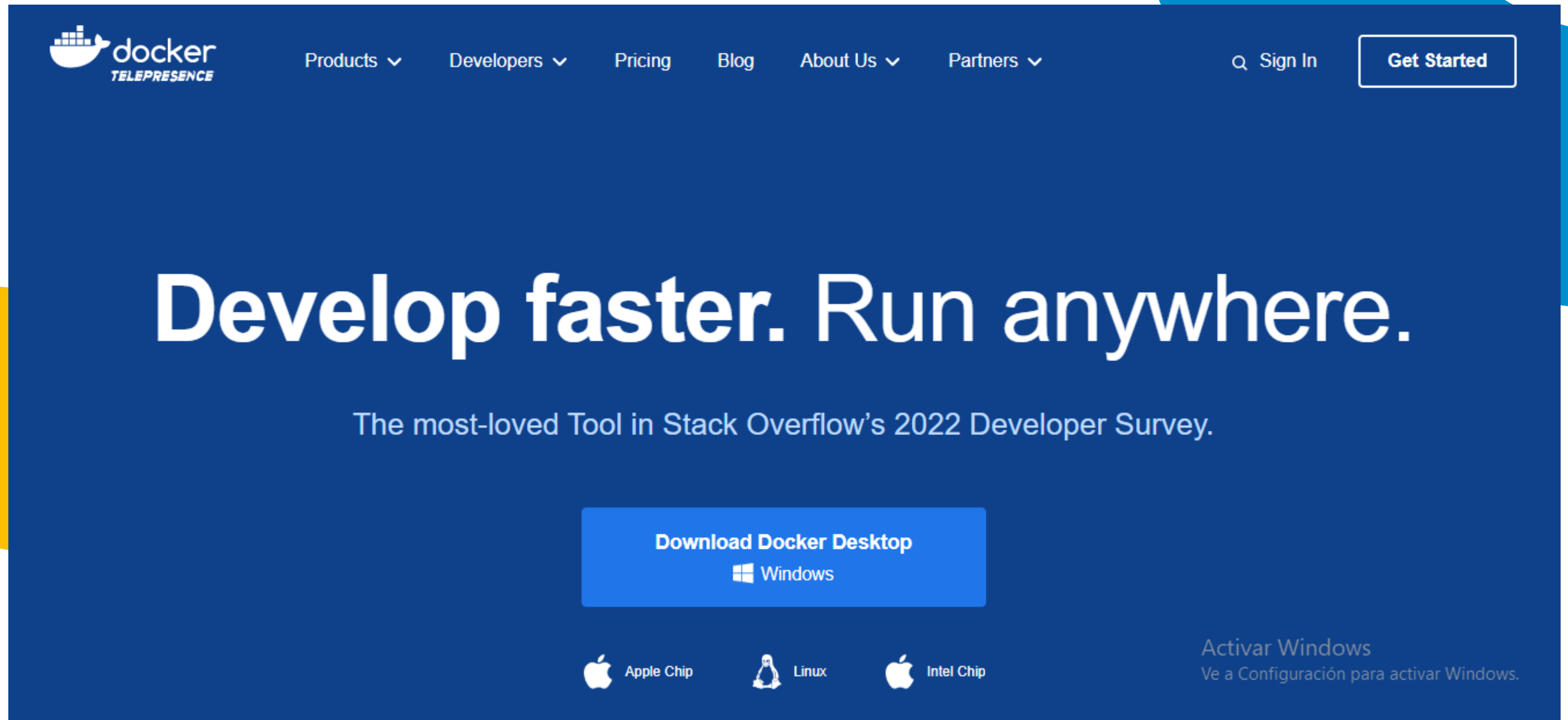


# **INSTALACIÓN WINDOWS**



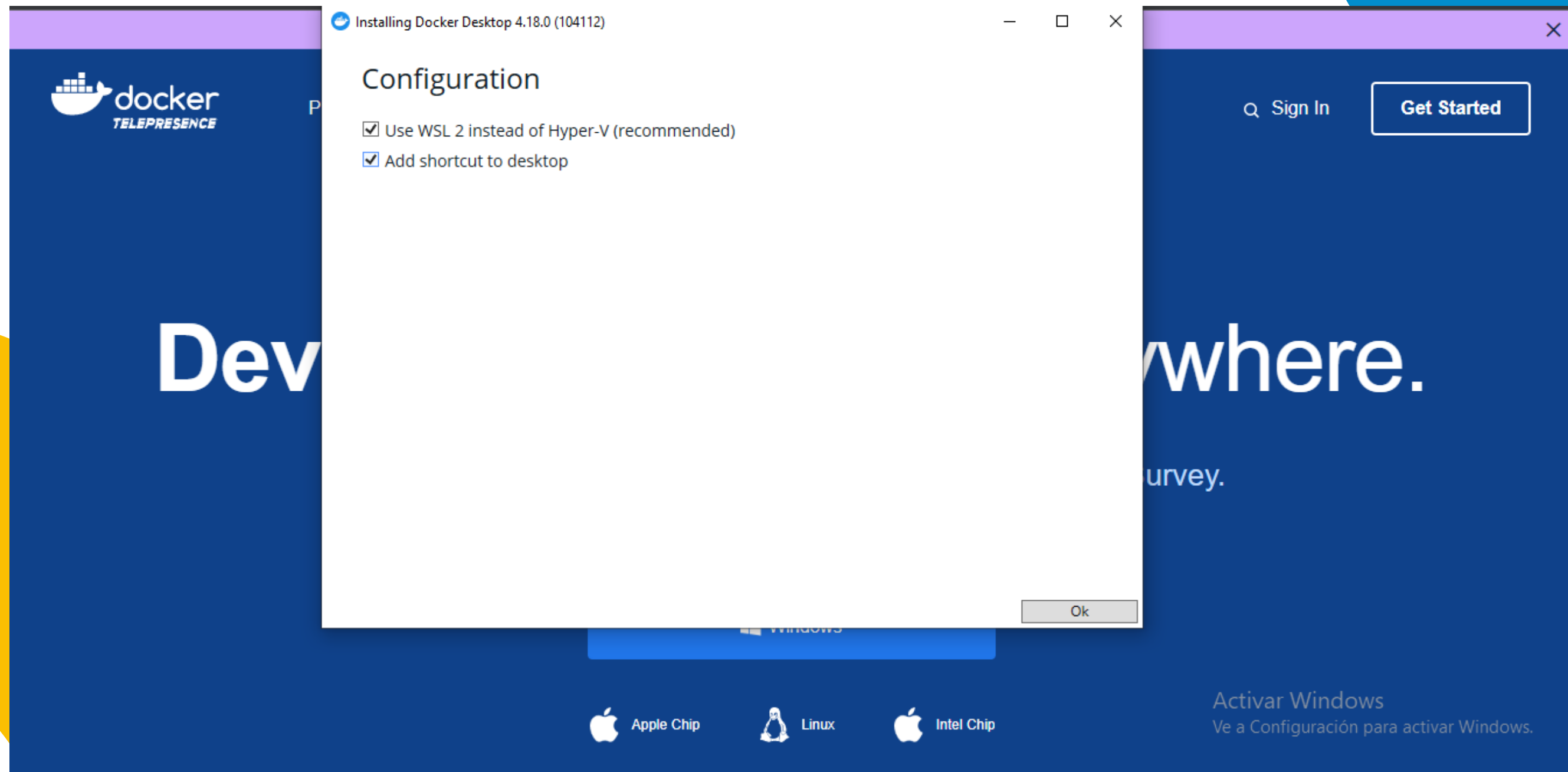
# DESCARGAR DOCKER

<https://www.docker.com>

The image is a screenshot of the Docker website's homepage. The background is a solid dark blue. At the top left is the Docker logo, which consists of a white ship icon and the word "docker" in white, with "TELEPRESENCE" in smaller white letters below it. To the right of the logo is a navigation menu with the following items: "Products" with a dropdown arrow, "Developers" with a dropdown arrow, "Pricing", "Blog", "About Us" with a dropdown arrow, and "Partners" with a dropdown arrow. Further right is a search icon followed by the text "Sign In". On the far right of the top navigation bar is a white rectangular button with the text "Get Started" in blue. Below the navigation bar, the main heading "Develop faster. Run anywhere." is written in large, white, sans-serif font. Underneath this heading, a smaller line of text reads "The most-loved Tool in Stack Overflow's 2022 Developer Survey." in a lighter blue font. In the center of the page is a large blue rectangular button with the text "Download Docker Desktop" in white, and below it, the Windows logo followed by the word "Windows" in white. At the bottom of the page, there are three logos: the Apple logo followed by "Apple Chip", the Linux penguin logo followed by "Linux", and the Apple logo followed by "Intel Chip". In the bottom right corner, there is a small white text area that says "Activar Windows" and "Ve a Configuración para activar Windows."

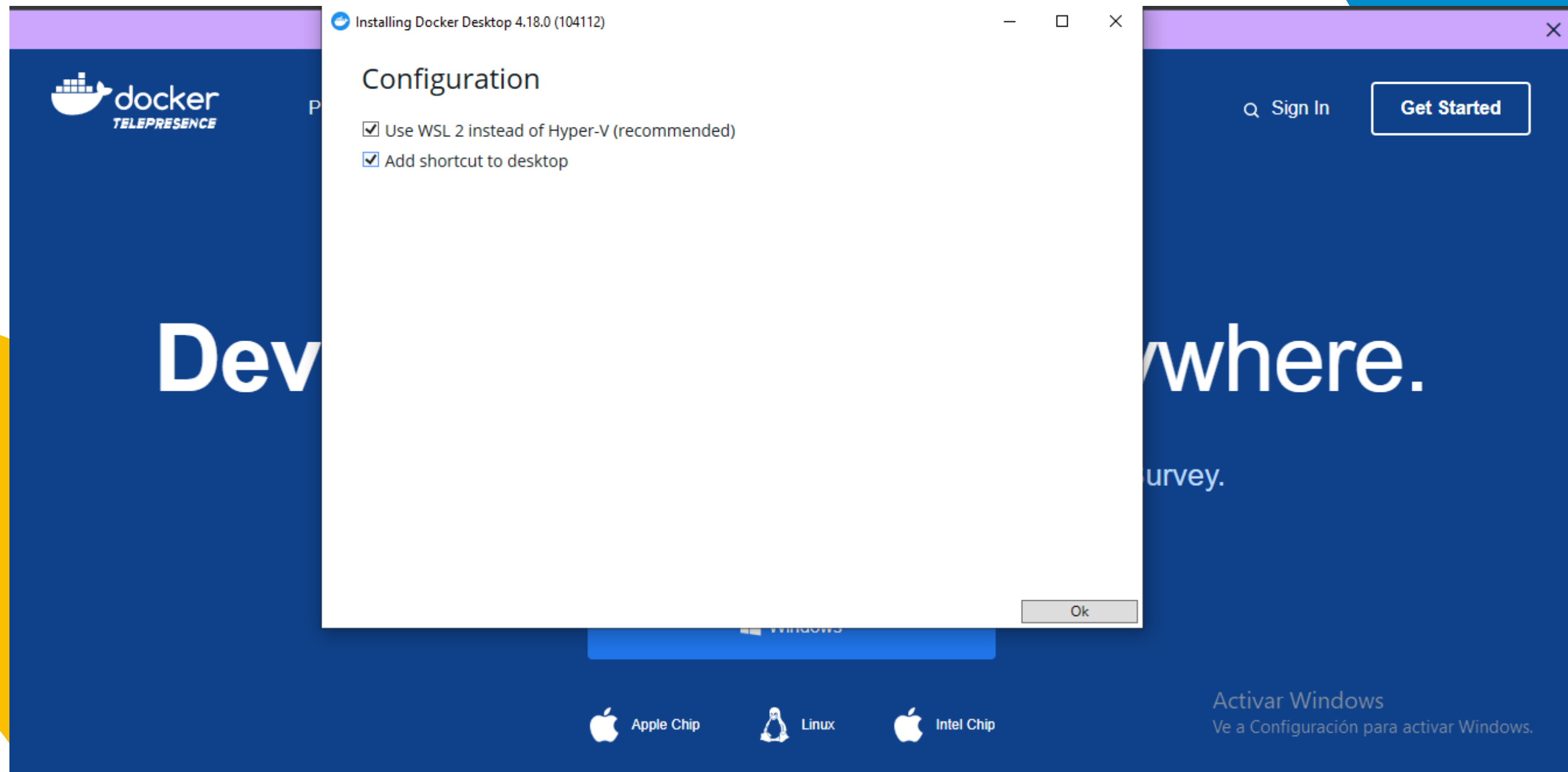
# INSTALAR DOCKER

Importante tener marcada la opción WSL 2



# INSTALAR DOCKER

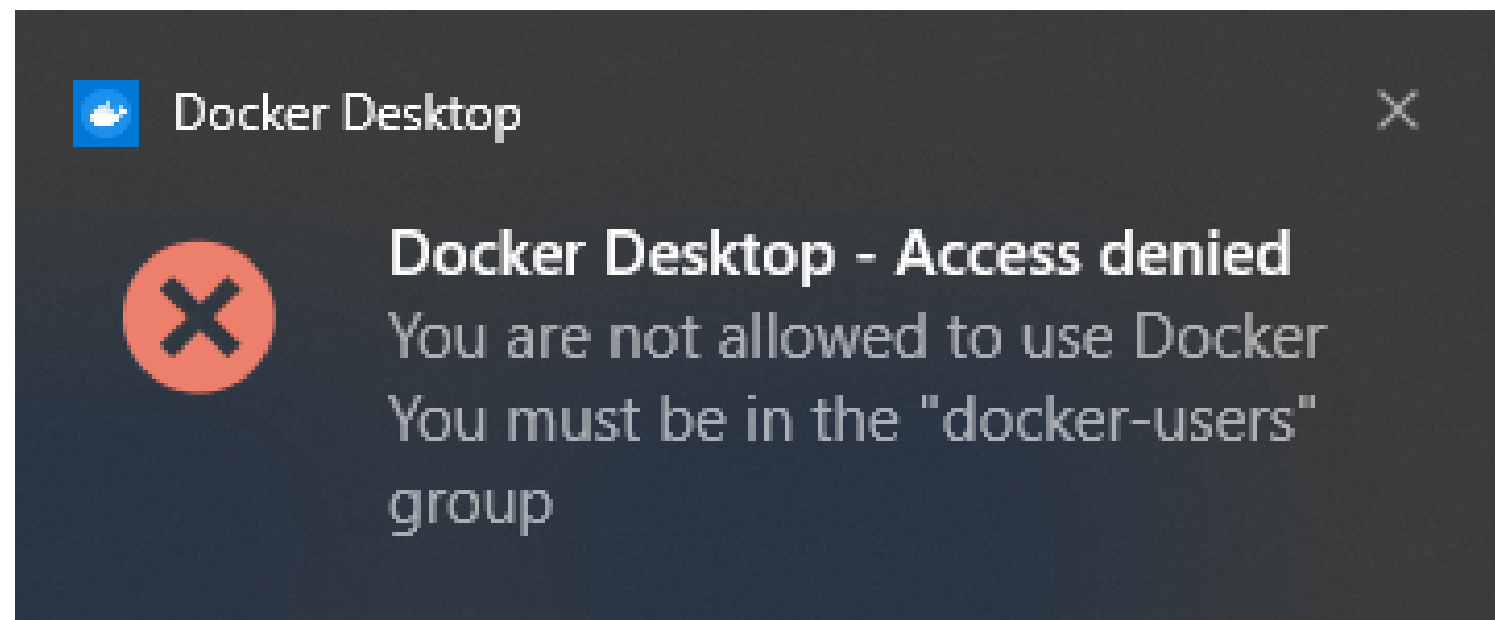
Importante tener marcada la opción WSL 2



# INSTALAR DOCKER

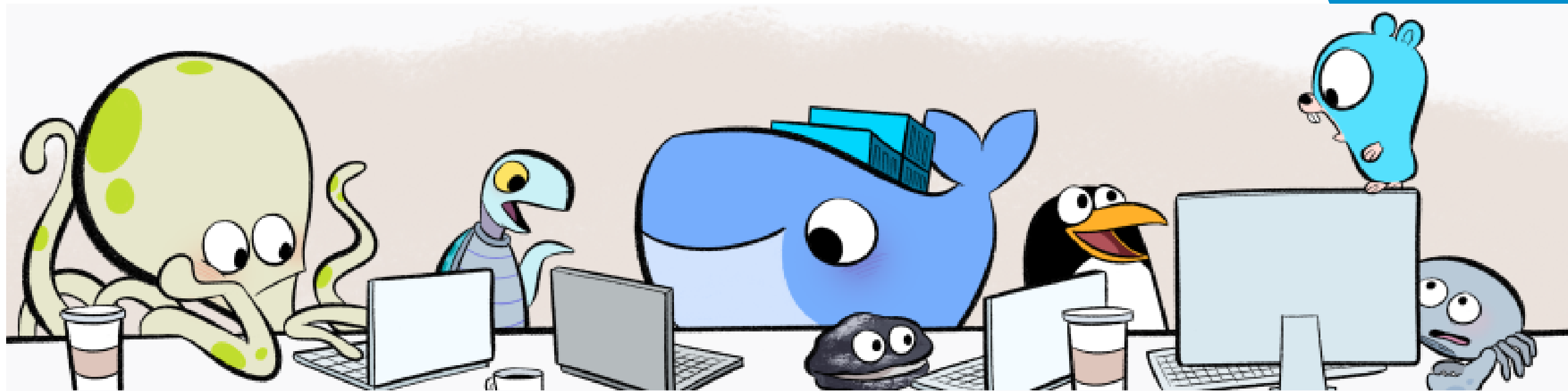
Si al intentar iniciar docker desktop aparece el mensaje de acceso denegado, deberemos darle el permiso con el siguiente comando:

**net localgroup docker-users "alumnoT" /ADD**



# INSTALAR DOCKER

Aceptar los terminos



## Docker Subscription Service Agreement

By selecting **accept**, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), and the [Data Privacy Policy](#).

**Note:** Docker Desktop is free for small businesses (fewer than 250 employees AND less than \$10 million in annual revenue), personal use, education, and non-commercial open source projects. Otherwise, it requires a paid subscription for professional use. Paid subscriptions are also required for government entities. [Read the FAQ to learn more.](#)

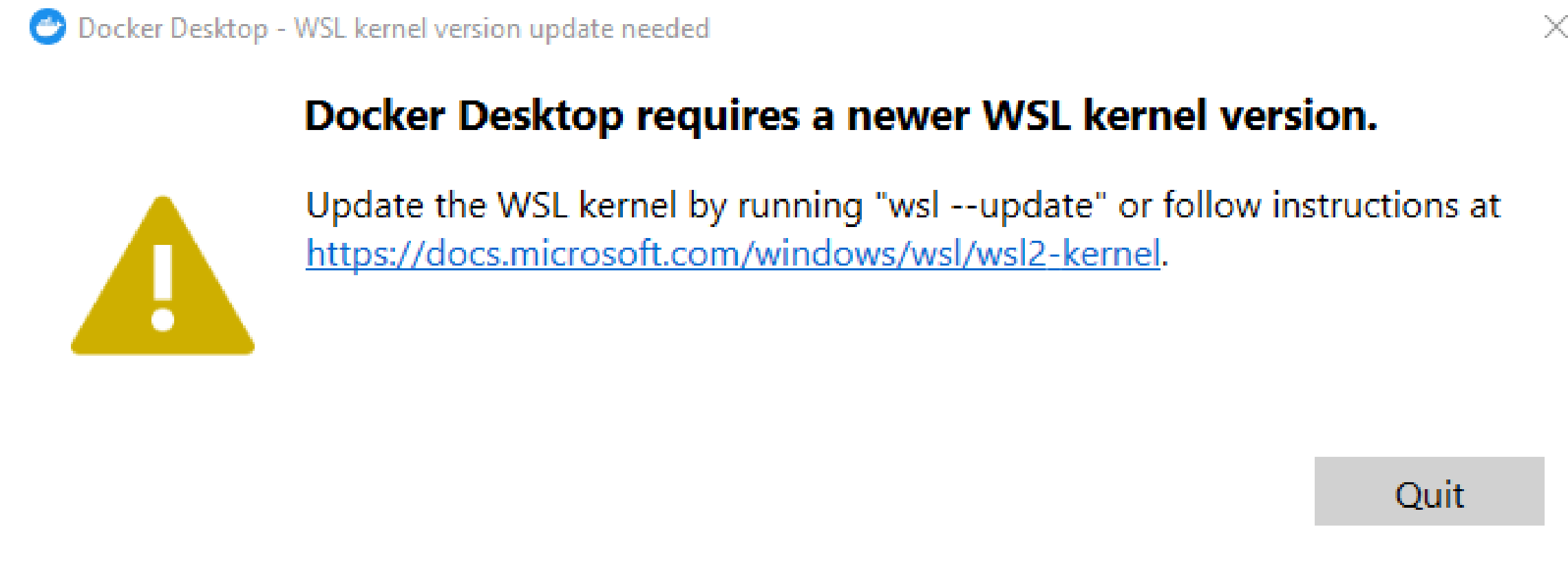
[View Full Terms](#) 

**Accept**

**Close**

# INSTALAR DOCKER

Si el sistema operativo no tiene instalado WSL al iniciar la aplicación nos aparecerá el siguiente mensaje, utiliza el enlace para descargarlo e instalarlo.



# IMAGEN DOCKER

## DESCARGAR IMAGEN DE DOCKER HUB

Podemos descargar cualquier imagen de <https://hub.docker.com/>

Comando:

```
docker pull <nombre imagen>
```

Ejemplo:

```
docker pull ubuntu:latest
```

## MANEJO DE IMAGENES

listar las imagenes disponibles:

```
docker images ls
```

Eliminar una imagen

```
docker image rm <id de la imagen o su tag>
```





# FICHERO DOCKER FILE

## DEFINICIÓN

Es un archivo que contiene las instrucciones para crear una imagen Docker. El nombre del fichero debe ser **Dockerfile**

Las instrucciones se ejecutan de manera secuencial ( por convención se escriben en mayúsculas, aunque las puede ejecutar también en minúsculas) (#) para hacer comentarios

## EJEMPLO

```
FROM ubuntu:bionic
```

```
WORKDIR /home
```

```
COPY hello.sh .
```

```
RUN chmod u+x hello.sh
```

```
CMD ./hello.sh
```

# FICHERO DOCKER FILE

## EXPONER UN PUERTO

EXPOSE 80 -> indica que se va a exponer el puerto 80

## CONSTRUIR UNA IMAGEN A PARTIR DEL FICHERO

`docker build -t pruebaweb .`

donde t indica el nombre que se le va a asignar a la imagen

El "." final indica que el fichero se encuentra en el mismo directorio donde se va a ejecutar el comando si no es así indicar la ruta

Nota: dependiendo de la configuración será necesario ejecutarlo con "sudo"

# INICIAR UN CONTENEDOR

## INICIAR UN CONTENEDOR

Podemos descargar cualquier imagen de <https://hub.docker.com/>

Comando:

```
docker run <nombre imagen>
```

Ejemplo:

```
docker run -ti ubuntu bash
```

## MANEJO DE CONTENEDORES

listar los contenedores activos:

```
docker ps -a
```

el argumento -a muestra todos los contenedores aunque no esten activos

Eliminar una imagen

```
docker container rm <id del container o su nombre>
```

# CONEXIÓN A UN CONTENEDOR

Ejemplo:

```
docker run -it --name ubuntu_c  
ubuntu:21.04
```

**CTRL+P+Q** para salir del contenedor

## MANEJO DE CONTENEDORES

Dos opciones para entrar en el contenedor

**docker attach ubuntu\_c** nos permite volver a entrar en el contenedor. Si pulsamos ctrl+c o exit detendremos el contenedor

**docker exec -it ubuntu\_c bash ->** esto sería como si creásemos una terminal dentro del contenedor por lo que si ponemos exit no nos salimos del contenedor

# PARÁMETROS DEL RUN

Los parámetros dependen mucho del contenedor que se esté ejecutando. A continuación se indican algunos más usuales y típicos:

**-d:** Este parámetro se utiliza para ejecutar el contenedor en segundo plano (en modo "detached"), lo que permite al usuario seguir utilizando la terminal mientras el contenedor se ejecuta en segundo plano.

**--name:** Con este parámetro se puede asignar un nombre personalizado al contenedor, lo que facilita su identificación posteriormente.

**-p:** Con este parámetro se puede publicar un puerto del contenedor en el host. Por ejemplo, si el contenedor expone el puerto 80 y se quiere acceder a él desde el host en el puerto 8080, se puede utilizar el siguiente comando: `-p 8080:80`.

# PARÁMETROS DEL RUN

**--restart:** Este parámetro se utiliza para indicar la política de reinicio del contenedor en caso de fallo o de que se detenga. Algunas de las políticas disponibles son no, always y on-failure. Por ejemplo, si se quiere que el contenedor se reinicie siempre que se detenga, se puede utilizar el siguiente comando: **--restart always**.

**-e:** Con este parámetro se pueden establecer variables de entorno dentro del contenedor. Por ejemplo, `-e MYSQL_ROOT_PASSWORD=mypassword`.

Algunos parametros del run se pueden combinar por ejemplo:

**-dp** indica que se ejecuta en segundo plano y el redireccionamiento de puerto. Por ejemplo `-dp 8081:80` haría el redireccionamiento desde el puerto 8081 del host al 80 del contenedor.

# DOCKER COMPOSE

## DEFINICIÓN

Hasta aquí hemos visto como crear un contenedor y las opciones más importantes que podemos utilizar, para facilitar la creación del contenedor se crea un fichero .yaml que nos ayuda a simplificar el proceso

Es un fichero indentado con tabulación

## EJEMPLO

```
version: "3.9"
```

```
services:
```

```
testCompose1:
```

```
  build: //donde está el dockerfile
```

```
  ports:
```

```
    - "81:80"
```

```
  links:
```

```
    - testCompose2
```

```
testMongo:
```

```
  image: mongo
```

```
  ports:
```

```
    - "27087:27087"
```

```
  environment:
```

```
    - MONGO
```

Comando de lanzamiento:

```
docker compose up
```



# DOCKER VOLUME

Permite almacenar el contenido de un directorio fuera del contenedor.

- El contenido del contenedor no se elimina con el contenedor
- Un volumen puede estar compartido entre varios contenedores.

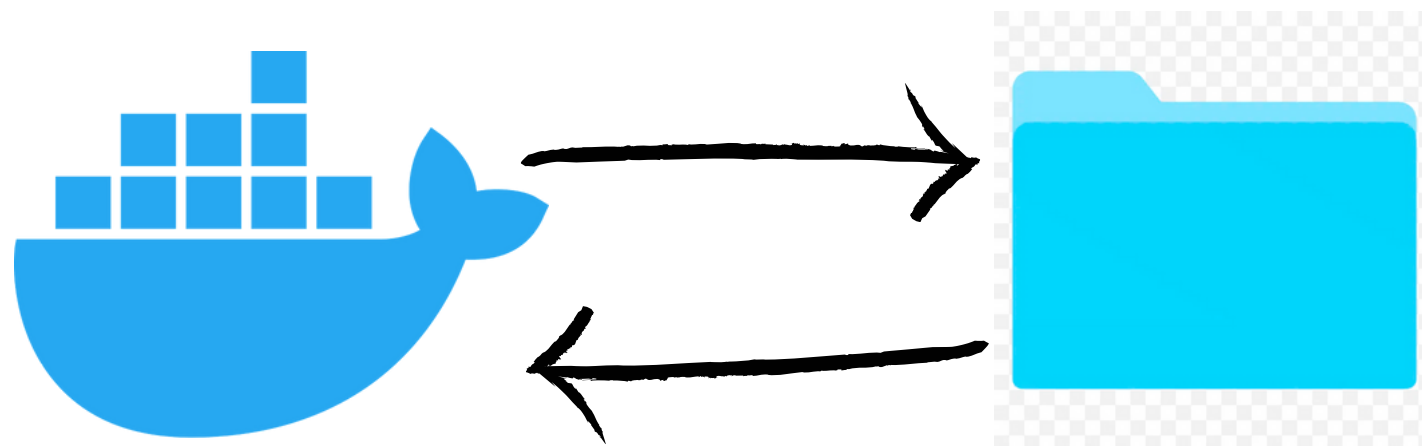
**docker create -v C:\ProgramData\MySQL:/home/app -p 8050:8000 --name ejemplo3\_1  
ejemplo3**

El parametro '-v' está dividido en dos partes por los ":" la parte izquierda es la carpeta del host y la parte derecha es del interior de docker

¿Cómo copiar archivos al contenedor?

**docker cp ejemploTest:/home/app/db.sqlite3 .**

El comando cp permite copiar archivos desde el sistema anfitrión hacia un contenedor y viceversa.



# REDES - REDES TIPO BRIDGE

```
docker network create my_network
```

En windows:

```
docker network create -d nat my_network
```

```
docker network ls -> para ver las redes disponibles
```

Nota: en lugar de localhost tenemos que poner el nombre del contenedor del docker para poder interactuar con él

En el docker create utilizar `--network my_network` para conectarlo

