

LINUX DevOps

DOCKER

```
...or object to mirror...  
mirror_mod.mirror_object =
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
...selection at the end -ad...  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier...  
mirror_ob.select = 0  
= bpy.context.selected_obj...  
data.objects[one.name].select
```

```
print("please select exactly
```

```
-- OPERATOR CLASSES --
```

```
...types.Operator):  
    X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

Virtualización frente al uso de contenedores

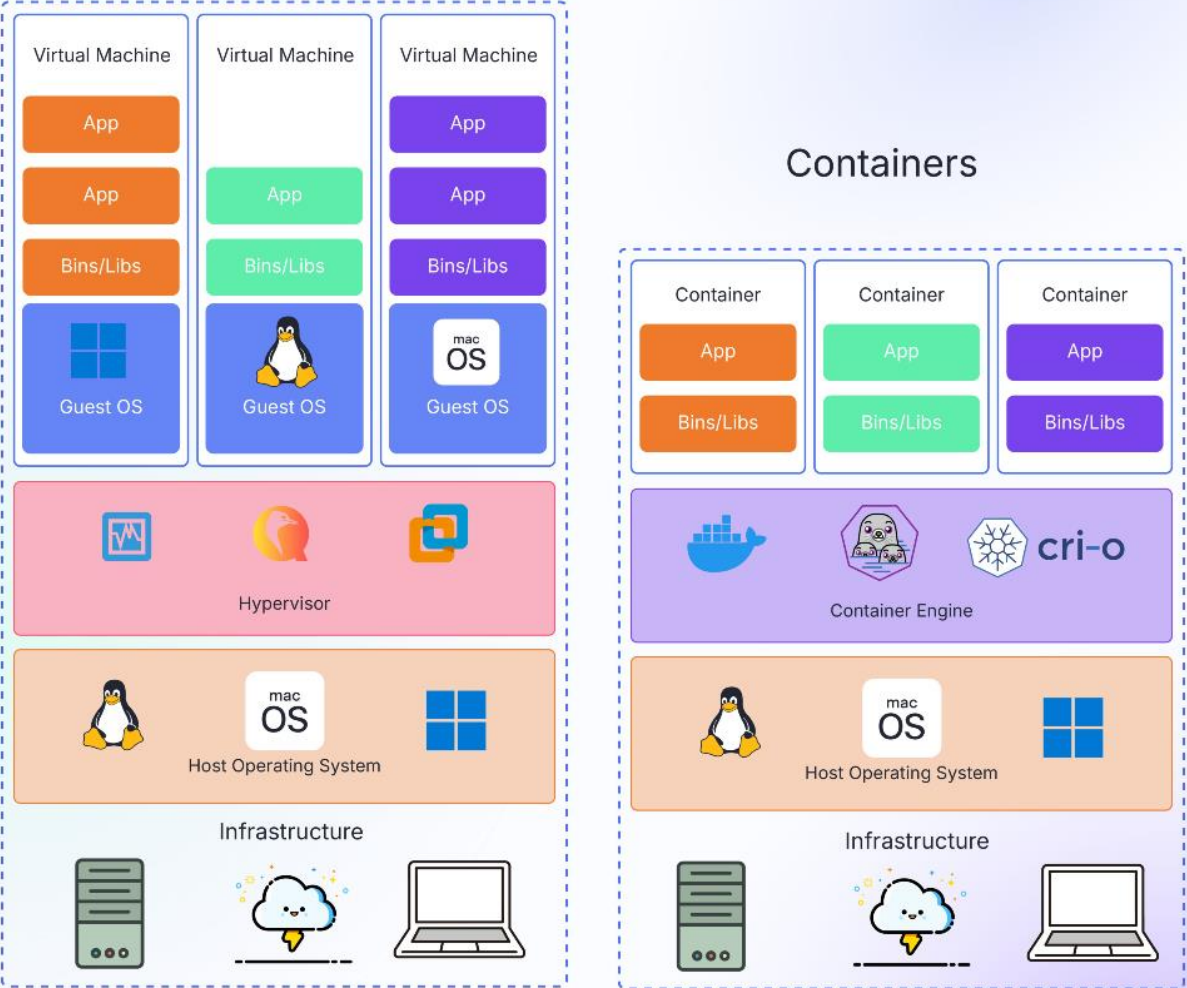
- La virtualización es una técnica que consiste en ejecutar un ordenador simulado completo dentro de otro ordenador.
 - **Completo** significa que refleja todo lo que tendría un ordenador físico: placa base, BIOS, procesador, discos duros, puertos USB, etcétera.
 - **Simulado** significa que es totalmente un producto de software. Este ordenador no existe físicamente, por eso se llama virtual.
- Para existir, la máquina virtual (VM), como se suele llamar a los ordenadores simulados, necesita una real, física, que la emule.
- La máquina física se llama **host o hipervisor**.

- Tengo un ordenador físico. Es potente....

¿Por qué querría ejecutar una máquina virtual en él?

- La VM será **menos potente que el host**: después de todo, el host necesita RAM, CPU y espacio en el disco duro para sí mismo.
- También **hay una pequeña caída en el rendimiento** (ya que en realidad estamos ejecutando un programa que emula el hardware completo) en comparación con la máquina física.

VMs vs CONTAINERS



Virtual Machines

Ventajas de la virtualización

- **Aislamiento:** Las máquinas virtuales se presentan, como ordenadores totalmente funcionales. Para el sistema operativo en ejecución, parecen ser máquinas físicas separadas.
- Cualquier cosa que ejecutemos en ellas no debería poder acceder al ordenador anfitrión (a menos que se permita específicamente) y, de hecho, salvo unos pocos incidentes en los que tales cosas fueron posibles (como resultado de un error de programación), las máquinas virtuales han proporcionado entornos seguros.
- **Puesta a punto:** Con un host suficientemente potente, es posible particionar sus recursos de forma que cada máquina virtual en ejecución tenga garantizada la RAM, el espacio en el disco duro y la CPU.

- **Simplificación del sistema operativo**: Cuando se ejecutan varias cargas de trabajo, como bases de datos, servidores WWW y servidores de correo, la complejidad de mantener un único servidor que las ejecute todas crece bastante rápido. Cada programa instalado requiere la instalación de software adicional (como bibliotecas y programas de ayuda). El software de hipervisor moderno soluciona muchos obstáculos de administración del sistema mediante clones, instantáneas, imágenes doradas, etc.
- **Aceleración**: Configurar correctamente un flujo de trabajo utilizando VMs nos permite iniciar un nuevo sistema operativo completo con su propio hardware de servidor o escritorio en cuestión de minutos en lugar de horas. Esto abre nuevas posibilidades para entornos de pruebas, soluciones de escritorio remoto, etc.

- **Automatización**: El software de virtualización moderno ofrece muchas funciones que favorecen la automatización de la gestión del sistema a muchos niveles.
- Las **instantáneas** -una captura puntual de todo el sistema- permiten volver en cualquier momento a un estado anterior del sistema. Esto permite retroceder fácilmente de cambios no deseados al último estado bueno conocido.
- Los **clones** nos permiten aprovisionar nuevas máquinas virtuales basándonos en otra ya en funcionamiento y configurada.
- Las **imágenes archivadas** de máquinas virtuales que podemos importar y arrancar fácil y rápidamente, omitiendo por completo la instalación y limitando la configuración al mínimo absoluto. Esto también permite recrear entornos fiables.

Contenedorización... (uso contenedores)

- Los contenedores son entornos virtuales ligeros que nos permiten ejecutar un **único proceso de forma aislada**.
- Un contenedor idealmente preparado **consta únicamente del software y las bibliotecas** necesarias para ejecutar la aplicación.
- El sistema operativo anfitrión se encarga de operar el hardware, gestionar la memoria y todas las demás tareas periféricas.
- El principal supuesto del contenedor es que no emula **ser un sistema operativo independiente** o un servidor independiente.

- Los procesos o usuarios dentro del contenedor **pueden saber fácilmente que están encerrados allí.**
- El inconveniente es que los contenedores **no emulan hardware.** No se pueden utilizar para probar nuevos controladores, por ejemplo.
- La **ventaja** es que un solo contenedor puede **ocupar tan sólo unos pocos megabytes** de espacio en el disco duro y **sólo la memoria necesaria** para que el proceso se ejecute.

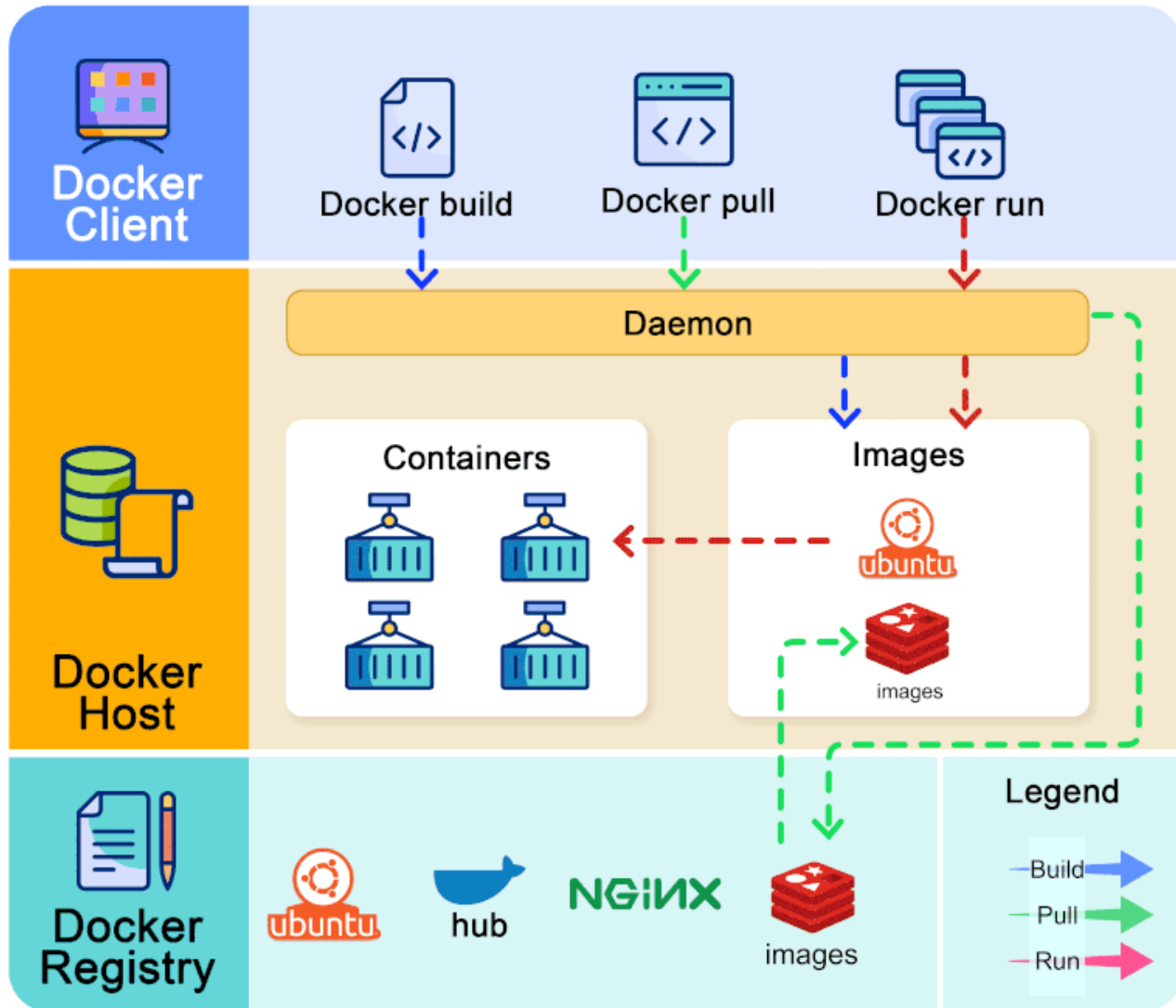
- Como consecuencia, **arrancar un contenedor sólo lleva el tiempo que la aplicación necesita para iniciarse.**
- El **tiempo de arranque** -la BIOS, las pruebas de hardware y el tiempo de arranque del sistema operativo- **se reduce al mínimo.**
- **Todo el software no requerido por la aplicación** puede, y debe, **omitirse.**
- Dado el pequeño tamaño de las imágenes contenedoras, sus tiempos de redistribución son casi insignificantes, sus tiempos de arranque casi instantáneos, y el tiempo y el proceso de construcción se reducen bastante
- Esto ha llevado a una recreación mucho más fácil del entorno. Esto, a su vez, ha llevado a una configuración más fácil del entorno de prueba y, muy a menudo, a **despliegues de nuevas versiones del software, hasta varios miles de veces al día.**
- El escalado de las aplicaciones se ha hecho mucho más fácil y rápido

- Supuso un cambio en el enfoque de la ejecución de aplicaciones.
- La consecuencia lógica es que los contenedores no se mantienen del mismo modo que el sistema operativo.
- **No se actualiza el software dentro del contenedor**, sino que se despliega un contenedor con una nueva versión de software en lugar de la obsoleta.
- Esto lleva a suponer **que no se guardan los datos dentro de un contenedor**, sino en un **sistema de archivos** que se **adjunta al contenedor durante la ejecución**.
- Docker dispone de un **ecosistema para compartir fácilmente imágenes de contenedores**

- Una **imagen de contenedor** es, en el caso de Docker, un simple archivo que **consta de todos los binarios y bibliotecas necesarias** para la aplicación y unos pocos archivos con información de configuración.
- Puesto que el tamaño suele ser bastante pequeño y **la imagen nunca contiene datos**, es lógico permitir que la gente comparta la imagen que ha creado.
- **Docker tiene un centro de imágenes** (llamado **Docker Hub**) con una interfaz de usuario WWW y **herramientas de línea de comandos para buscar, descargar y cargar imágenes**.
- El **hub** permite **puntuar las imágenes y hacer comentarios** a los autores.

How does Docker Work ?

 blog.bytebytego.com



DOCKER HUB

```
docker pull mysql
```

- MySQL

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
5645ac95474c: Download complete
d1014e296527: Download complete
c5d32318ef6d: Download complete
5dd6251984b1: Download complete
bcba864ffae7: Download complete
666d78c46a16: Download complete
2adf46058f45: Download complete
9dcae24b624f: Download complete
e176816781e5: Download complete
903087d703a7: Download complete
Digest: sha256:146682692a3aa409eae7b7dc6a30f637c6cb49b6ca901c2cd160becc81127d3b
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
damiansualdea@Mac-Studio-de-Damian-2 ~ %
```

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker image ls
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest      72297848456d  3 weeks ago   139MB
mysql           latest      146682692a3a  3 weeks ago   1.1GB
damiansualdea@Mac-Studio-de-Damian-2 ~ %
```

```
docker run -d -p 3306:3306 --name mysql -e MYSQL_ROOT_PASSWORD=1234 mysql:latest  
7b9554cfbab09038cf6db862
```

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6cd87a46515f	mysql:latest	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql

```
docker exec -it mysql mysql -u root -p
```

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker exec -it mysql mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 11  
Server version: 9.2.0 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2025, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> █
```



```

public class Principal {
    public static void main(String[] args) throws SQLException {
        Connection con
            = DriverManager.getConnection(
                url:"jdbc:mysql://localhost:3306/prueba", user:"root", password:"1234");

        System.out.println(x:"Conexión correcta...");
    }
}

```

t - BBDDXAMPP (run) x

run:

Conexión correcta...

BUILD SUCCESSFUL (total time: 0 seconds)

`docker start mysql`

```

damiansualdea@Mac-Studio-de-Damian-2 ~ % docker stop mysql
mysql
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
damiansualdea@Mac-Studio-de-Damian-2 ~ %

```

```
docker pull phpmyadmin
```

- MySQL+phpMyAdmin

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS        NAMES
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker image ls
REPOSITORY    TAG           IMAGE ID              CREATED        SIZE
ubuntu        latest       72297848456d         3 weeks ago   139MB
phpmyadmin ←  latest     b8e9de0186fe         3 weeks ago   819MB
mysql         latest      146682692a3a         3 weeks ago   1.1GB
damiansualdea@Mac-Studio-de-Damian-2 ~ %
```

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker network create mysql-net
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker run -d --name phpmyadmin-mysql --network mysql-net
-p 8080:80 -e PMA_HOST=mysql phpmyadmin:latest
0fd241654234e8c7b456eb67eb86c6180f53dfa583e9c61cde2b2f67a978b39e
damiansualdea@Mac-Studio-de-Damian-2 ~ %
```



Bienvenido a phpMyAdmin

❗ El servidor MySQL no autorizó su acceso

Idioma (*Language*)

Español - Spanish

Iniciar sesión ?

Usuario:

root

Contraseña:



Contraseñas...

Sugerir contraseña nueva

Iniciar sesión

❗ mysqli::real_connect(): php_network_getaddresses: getaddrinfo for mysql failed: Name or service not known

❗ mysqli::real_connect(): (HY000/2002): php_network_getaddresses: getaddrinfo for mysql failed: Name or service not known

```
% docker network connect mysql-net mysql  
% █
```

- **Postgresql**

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
bcfa6399358c: Download complete
90fff65a4d40: Download complete
99830d4c1f31: Download complete
595f8139f452: Download complete
1745f39c82f5: Download complete
697baaa59bd7: Download complete
3573814a3a24: Download complete
48bf67e6ea6c: Download complete
13d9eea0329b: Download complete
bb21cec0e8d9: Download complete
70b54767ab63: Download complete
f041c2df10cb: Download complete
368ee7c4ca8d: Download complete
Digest: sha256:6e3358e46e34dae6c184f48fd06fe1b3dbf958ad5b83480031907e52b9ec2a7d
Status: Downloaded newer image for postgres:latest
```

```
docker run -d --name postgres-server -e POSTGRES_PASSWORD=1234 -p 5432:5432 postgres:latest
49d1b8c6fe99adfba65f3eee
```

```
docker exec -it postgres-server psql -U postgres
```

```
postgres=# CREATE DATABASE hola;
CREATE DATABASE
```

```
docker pull mongo
```

- **Mongo**

```
damiansualdea ~ - 2311 - 145832  
docker run -d -p 27017:27017 --name mongo-server mongo:latest
```

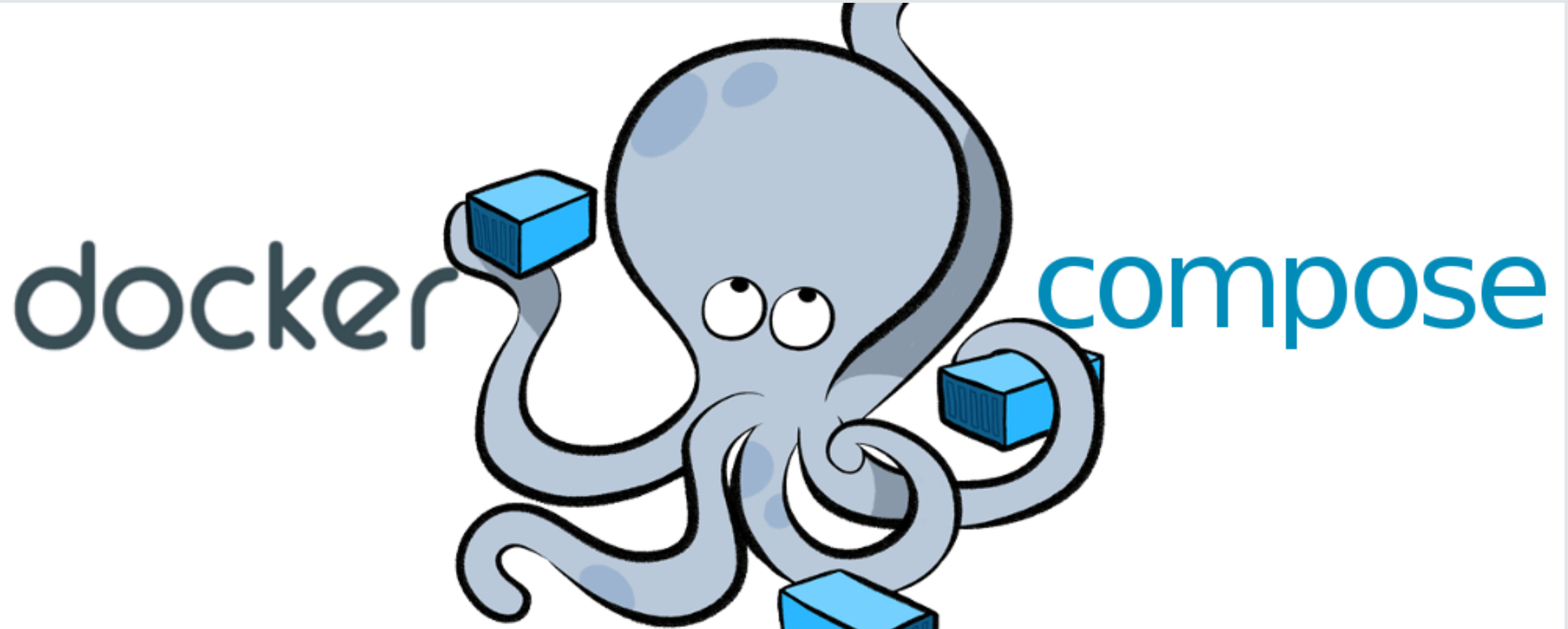
```
docker exec -it mongo-server mongosh
```

```
damiansualdea@Mac-Studio-de-Damian-2 ~ % docker ex  
Current Mongosh Log ID: 67b4a4a1330601d431a00aa0  
Connecting to: mongodb://127.0.0.1:27017/  
Using MongoDB: 8.0.4  
Using Mongosh: 2.3.8
```

damiansualdea@Mac-Studio-de-Damian-2 ~ % docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
09d4fade3112	mongo:latest	"docker-entrypoint.s..."	3 minutes ago	Up 3 minutes	0.0.0.0:27017->27017/tcp	mongo-server
2e1d5cc85aa3	postgres:latest	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes	0.0.0.0:5432->5432/tcp	postgres-server
6cd87a46515f	mysql:latest	"docker-entrypoint.s..."	46 minutes ago	Up 21 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql

Docker compose




```
damiansualdea@Mac-Studio-de-Damian-2 proyecto001 % docker-compose -p lamp up -d
[WARN][0000] /Users/damiansualdea/proyecto001/docker-compose.yml: the attribute `version` is not supported
[+] Running 4/4
  ✓ Network lamp_dev-env      Created
  ✓ Container lamp-mysql-1    Started
  ✓ Container lamp-web-1      Started
  ✓ Container lamp-apache-1   Started
damiansualdea@Mac-Studio-de-Damian-2 proyecto001 %
```

```
version: '3'
services:
  web:
    image: php:8.2-apache
    ports:
      - '8080:80'
    depends_on:
      - mysql
    volumes:
      - /var/www/html
    networks:
      - dev-env
```

```
mysql:
  image: mysql:latest
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: 'secret'
    MYSQL_USER: 'vincentb'
    MYSQL_PASSWORD: 'secret'
    MYSQL_DATABASE: 'vincentb'
  volumes:
    - mysqldata:/var/lib/mysql
  ports:
    - 3306:3306
  networks:
    - dev-env
```

```
apache:
  image: httpd:latest
  ports:
    - '80:80'
  depends_on:
    - web
  networks:
    - dev-env
  volumes:
    mysqldata: {}
  networks:
    dev-env: {}
```

```
damiansualdea@Mac-Studio-de-Damian-2 proyecto001 % docker-compose -p lamp down
[+] Running 4/4
  ✓ Container lamp-apache-1    Removed
  ✓ Container lamp-web-1      Removed
  ✓ Container lamp-mysql-1    Removed
  ✓ Network lamp_dev-env      Removed
```