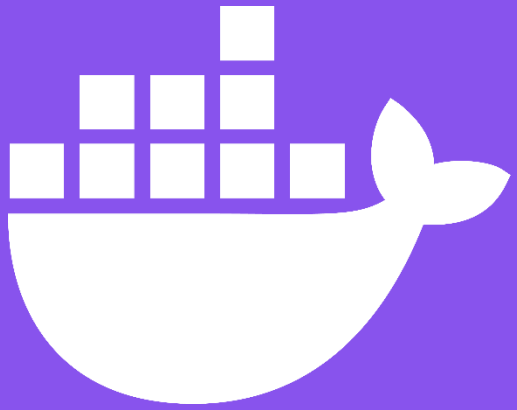


docker®

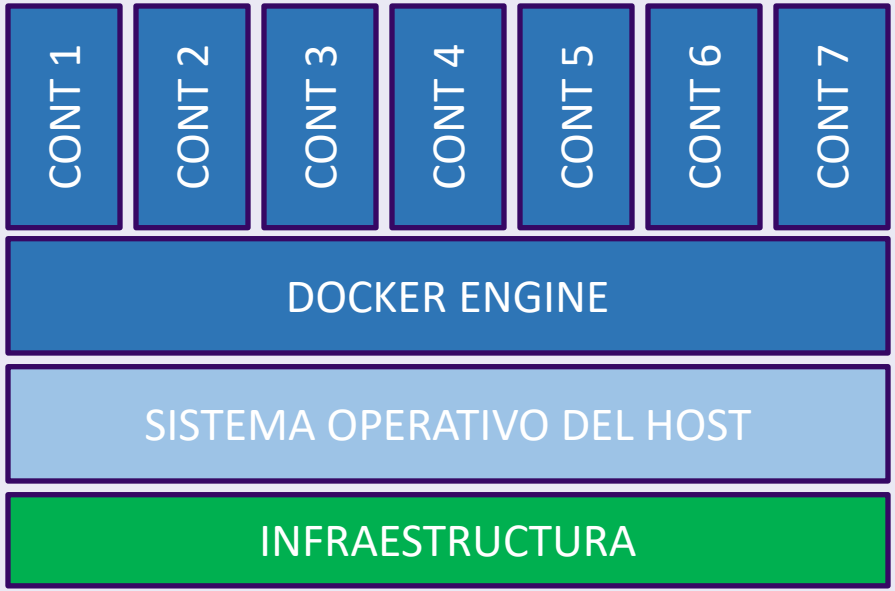
Fundamentos de Docker

Sección 1: Introducción

¿Qué es Docker?

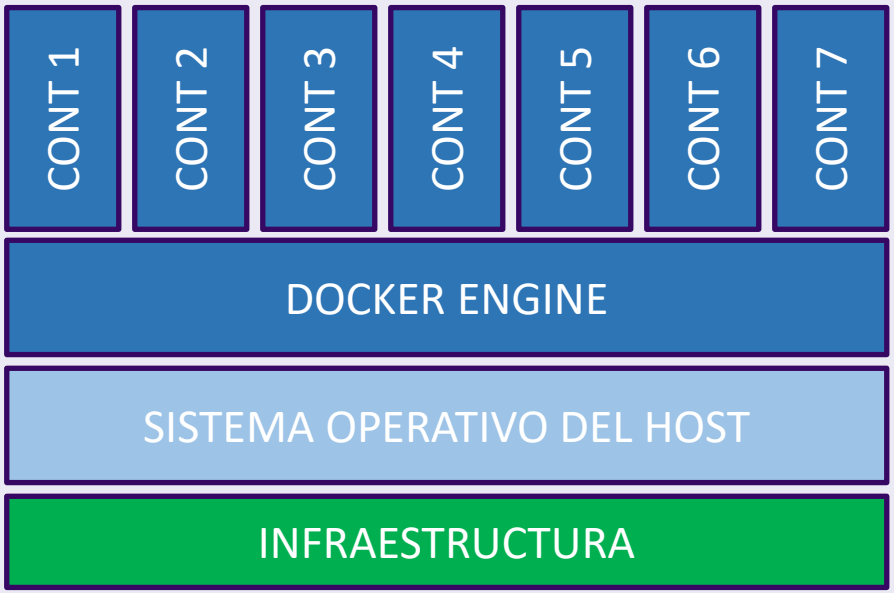
- Plataforma open-source utilizada para desarrollar, distribuir y ejecutar aplicaciones.
- Empaqueta y ejecuta las aplicaciones en un entorno llamado contenedores.
- Permite ejecutar varios contenedores en simultaneo en un mismo host.
- Cada contenedor contiene todo lo necesario para ejecutar la aplicación.
- Docker es ideal para trabajar en flujos de trabajo de CI/CD.

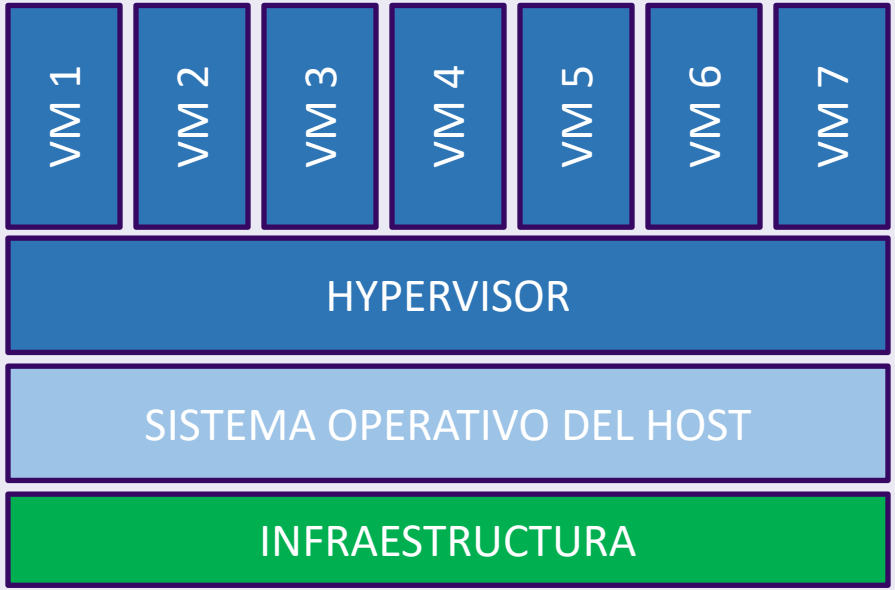
Arquitectura



DOCKER

Arquitectura: Comparación Docker vs. VM





DOCKER

MAQUINAS VIRTUALES

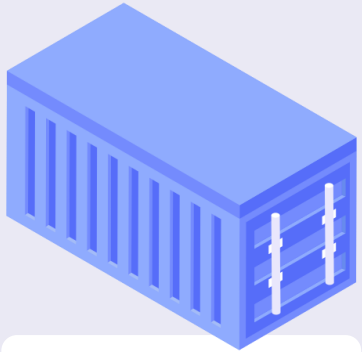
¿Por qué Docker?: Ventajas

DOCKER
+ LIVIANO
ESCALABILIDAD
PORTABILIDAD
AISLAMIENTO

Componentes Básicos

Dockerfile

Imagen

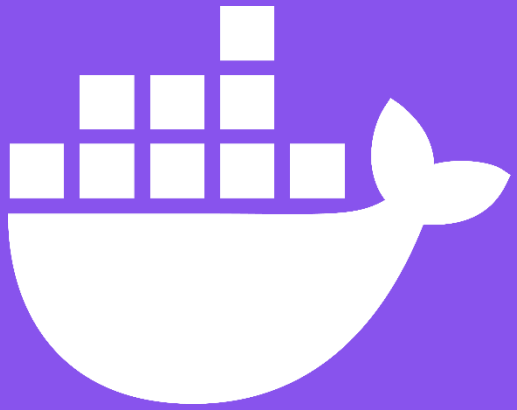


Contenedor

Instalación de Docker

Desde la documentación oficial:

https://docs.docker.com/get-started/get-docker/



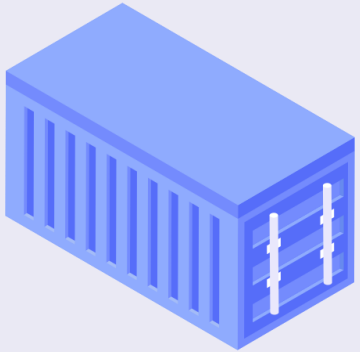
docker®

Sección 2: Contenedores

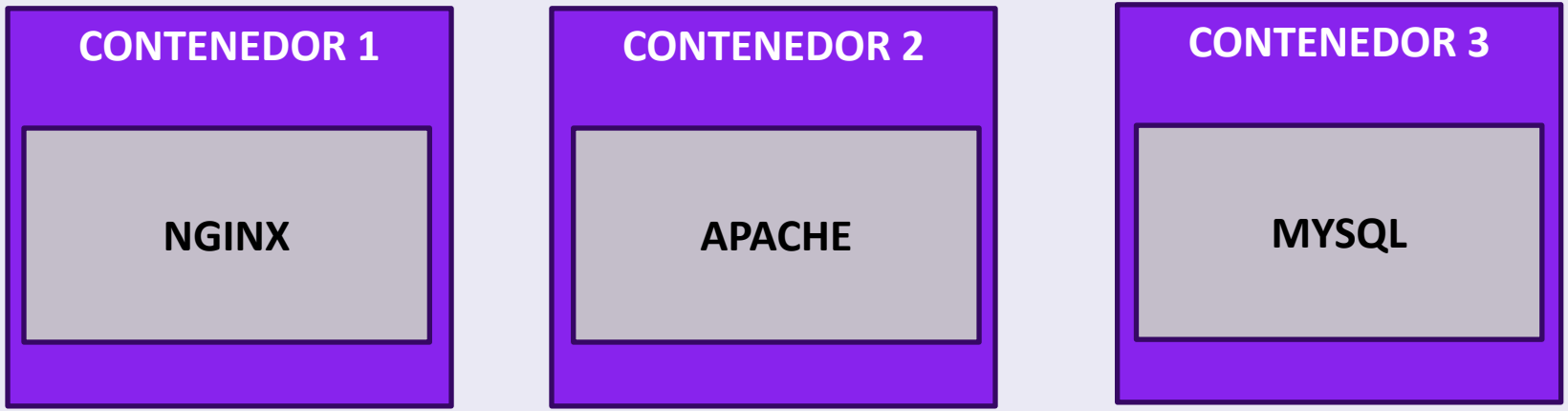
Analogía con contenedores de mercancías

¿Qué es un Contenedor?

- Es una estandarización que empaqueta Código y todas las dependencias que hacen funcionar a ese Código, de forma ligera.
- Multiples contenedores pueden correr en un mismo servidor, y compartir el kernel del Sistema operativo.
- Cada ejecución se ejecuta como un proceso aislado.
- Se crea a partir de una imágen



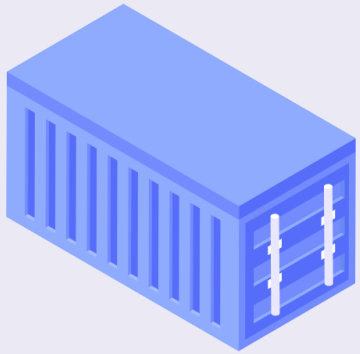
¿Qué es un Contenedor?



Cada contenedor puede correr cualquier imagen

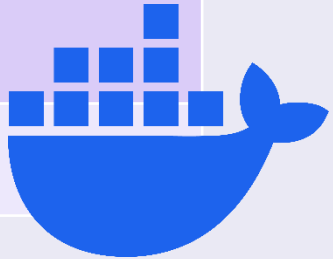
Estados de los contenedores

- **EN CREACION**
- **CREADO**
- **EN EJECUCIÓN**
- **PARADO**



Comandos para Contenedores

COMANDO	DESCRIPCION
docker run	Crea un nuevo contenedor a partir de una imanen
docker start	Inicia un contenedor que estaba en un estado de “Stopped” o parado
docker stop	Para un contenedor que se esta ejecutando
docker rm	Elimina un contenedor
docker ps	Lista los contenedores que se encuentran ejecutandose
docker ps -a	Lista TODOS los contenedores

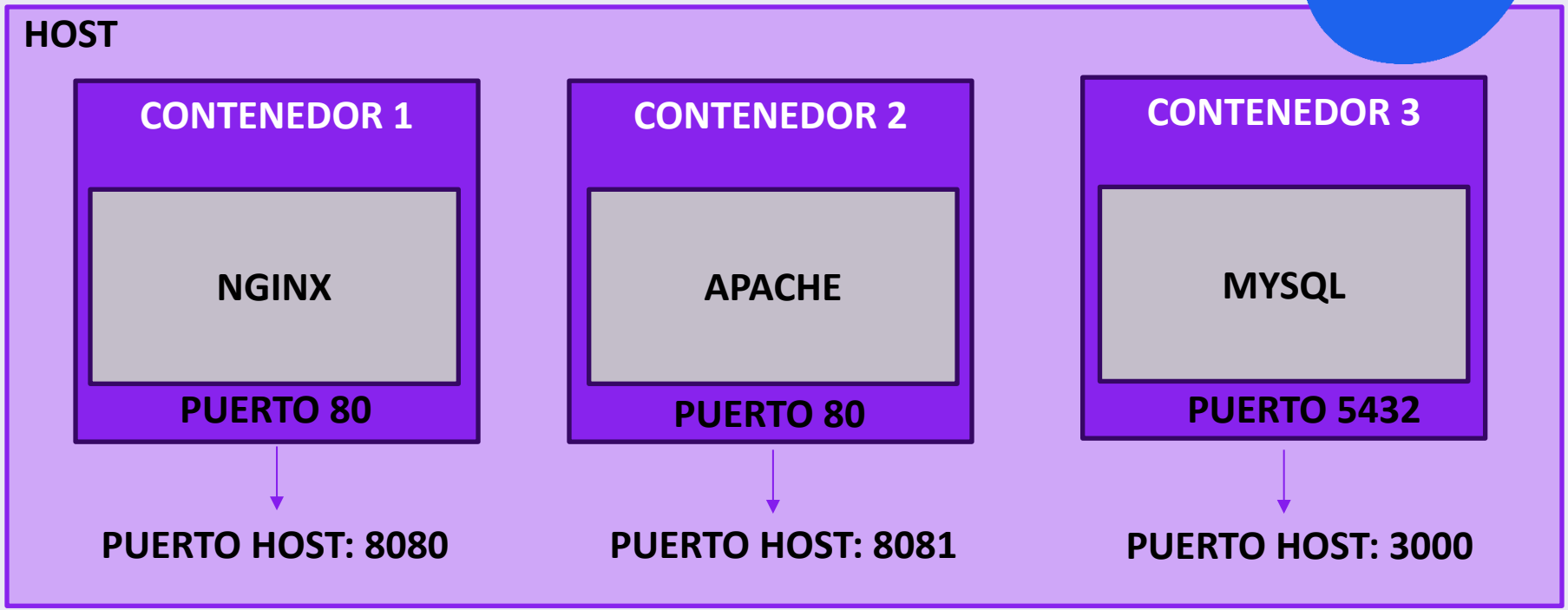


Comandos para Contenedores

COMANDO	DESCRIPCION
docker inspect	Información detallada acerca del contenedor
docker logs	Muestra los logs de un contenedor
docker restart	Reinicia uno o mas contenedores
docker prune	Elimina todos los contenedores parados
docker exec	Ejecuta un comando en un contenedor en ejecución
docker stats	Muesta los recursos consumidos por los contenedores

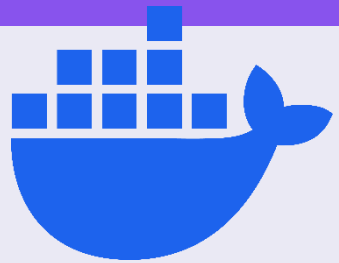
Puertos en contenedores

Mapeo de puertos



docker run –d –p PUERTO_HOST:PUERTO_CONTENEDOR imagen

Puertos en contenedores



Mapeo de puertos

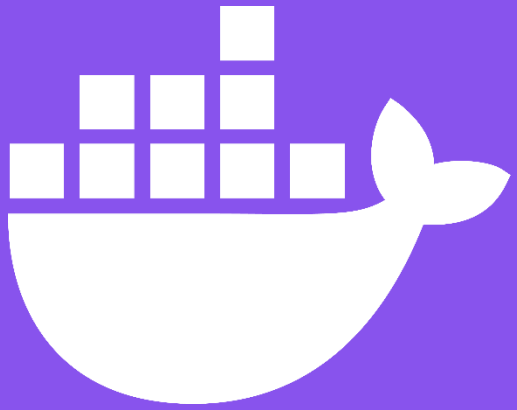
docker run –d –p PUERTO_HOST:PUERTO_CONTENEDOR imagen

Puerto del Host (HOST PORT)

Es el puerto en la máquina host que se asigna al puerto del contenedor.

Puerto del Contenedor (CONTAINER PORT)

Es el puerto dentro de un contenedor donde una aplicación escucha las conexiones entrantes.



docker®

Sección 3: Imágenes

¿Qué es una imagen de Docker?

- Es una plantilla de lectura que define cómo se construye un contenedor Docker. Contiene el código que se ejecuta, así como las dependencias o bibliotecas que el código necesita.
- Son portables, por lo que se pueden usar en diferentes entornos sin modificaciones.
- Se puede compartir y transportar; es decir, se puede implementar la misma imagen en varias ubicaciones a la vez.

¿Donde se obtienen las imágenes?

Localmente

Las imágenes que hemos utilizado previamente quedan almacenadas localmente, a menos que las eliminemos.

Registros de Imágenes

Registros de imágenes de Docker en la Nube.

Por ejemplo: Docker Hub, AWS ECR, etc.

Dockerfile

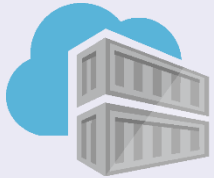
Podemos crear una imagen propia a partir de un archivo llamado Dockerfile.

También podemos utilizar otras imágenes para crear nuevas.

Registros de Imágenes

- Lugar donde se almacenan las imágenes.
- Pueden ser públicos o privados.

Amazon ECR



Azure Registry

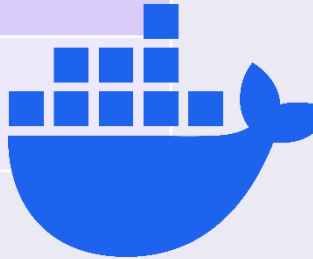
Google Container Registry

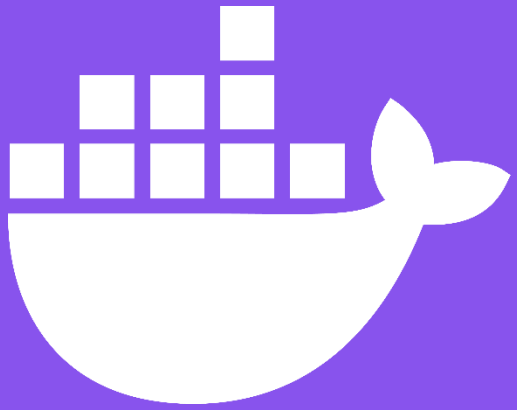
Dockerfile

- Es un archivo de texto que contiene una serie de instrucciones para permitir la creación de imágenes de forma personalizada por parte del usuario. En esta podemos hacer referencia al código fuente de la aplicación, la instalación de las dependencias, exponer un puerto, ejecutar un comando, copiar archivos, etc.

Comandos para Imágenes

COMANDO	DESCRIPCION
docker build	Crear una imagen a partir de un Dockerfile
docker image ls	Listar todas las imagenes
docker rmi o docker image rm	Eliminar una imagen
docker image prune	Elimina las imágenes que no se utilizan
docker image inspect	Obtener mas detalles de una imagen
docker image pull	Descargar una imagen desde el Docker Hub





docker®

Sección 4: Redes

¿Qué es una Red en Docker?

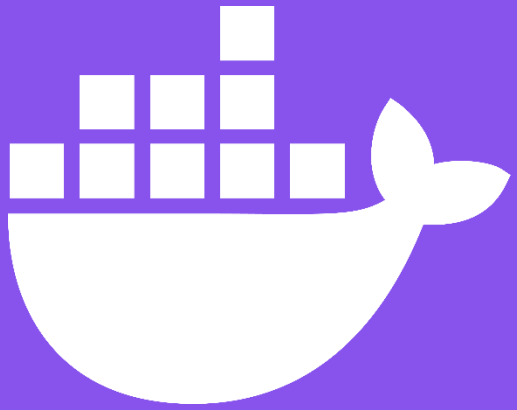
- Los contenedores se pueden conectar y comunicar entre sí, uno con otros, a partir de una red.
- Un contenedor se puede conectar a multiples redes.
- Por defecto, Docker crea una red propia “bridge”.
- Los contenedores dentro de una red, se pueden identificar por su nombre o su IP de contenedor.

Tipos de Redes

NOMBRE	DESCRIPCION
BRIDGE	Red por defecto. Los contenedores se pueden comunicar entre sí. Esta conectada a la red del host.
HOST	Los contenedores utilizan la red del Host. No están aislados.
NONE	Los contenedores están completamente aislados, por lo que no se comunican.

Comandos para Redes

COMANDO	DESCRIPCION
docker network ls	Listar todas las redes
docker network connect	Conectar un contenedor a una red
docker network disconnect	Desconectar un contenedor a una red
docker network create	Crear una nueva red
docker network rm	Eliminar una red
docker network prune	Eliminar redes que no se utilizan



docker®

Sección 5: Almacenamiento

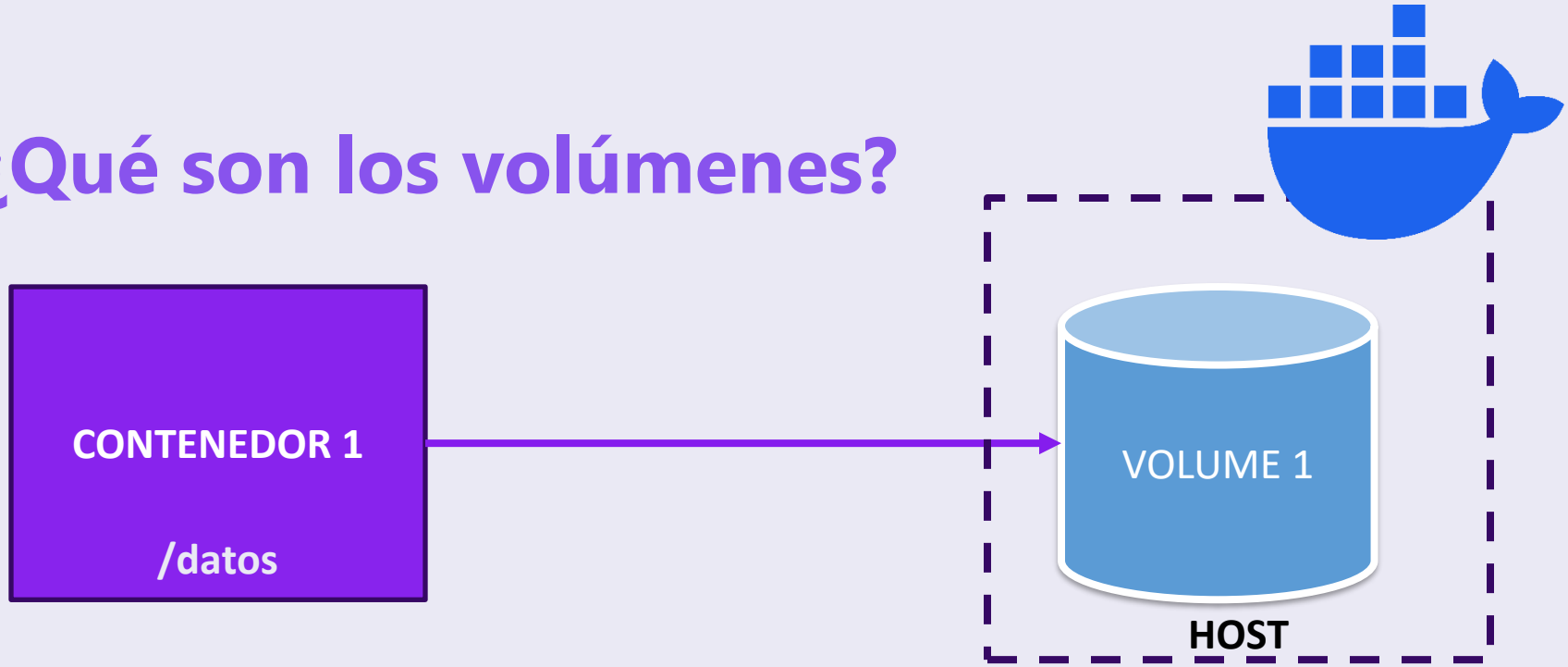
Almacenamiento persistente

- Por defecto, todos los archivos creados/modificados en los contendores no persisten cuando un contenedor es eliminado.
- Gracias al almacenamiento persistente, si el contenedor se para o elimina, los datos no se van a perder.
- Existen 3 tipos de almacenamiento en Docker: **Volúmenes**, **Bind Mounts** y **Tmpfs Mounts**

¿Qué son los volúmenes?

- Permiten garantizar el almacenamiento persistente, es decir si el contenedor se para o elimina, los datos no se van a perder.
- Se pueden asociar a varios contenedores.
- Se “monta” un Directorio dentro del contenedor con uno en la maquina host.
- Es gestionado por Docker y se almacenan en **/var/lib/docker/volumes** de la maquina host.
- Ningún proceso fuera de Docker tiene permisos para modificarlos.

¿Qué son los volúmenes?



- Dentro del contenedor existe un directorio que se asocia a ese volumen
- Si se elimina un contenedor, por defecto el volumen NO se elimina

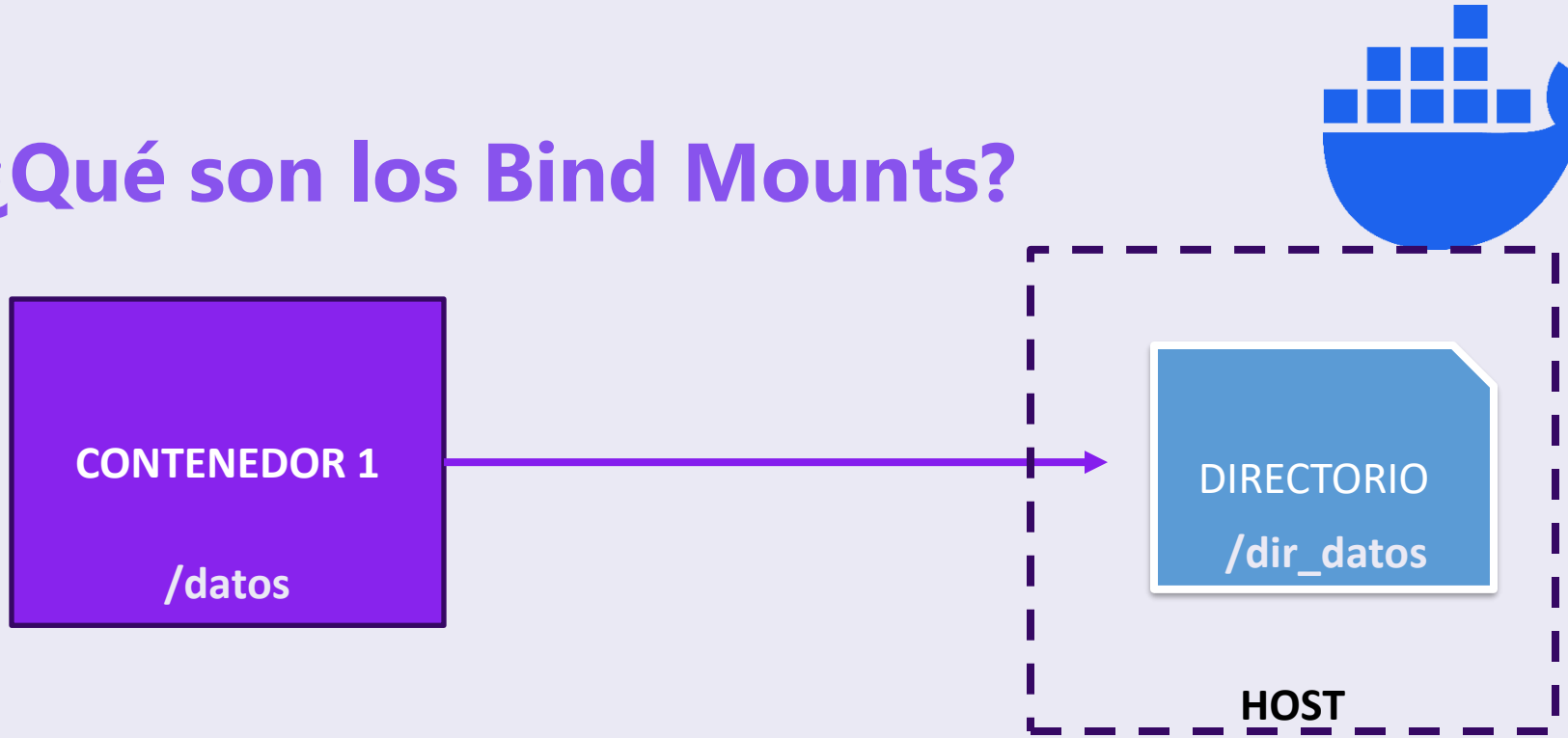
Comandos para Volúmenes

COMANDO	DESCRIPCION
docker volume ls	Listar todos los volúmenes
docker volumen create	Conectar un volumen
docker volumen inspect	Ver más detalles de un volumen
docker volumen rm	Eliminar un volumen
docker volumen prune	Eliminar todos los volúmenes que no son utilizados por contenedores

¿Qué son los Bind Mounts?

- Se “monta” un Directorio dentro del contenedor con otro directorio en la maquina host.
- Todo lo que se encuentre en un directorio se va a replica en el otro.
- Cualquier persona con permisos va a poder acceder a los datos y manipularlos.
- No son gestionados por Docker.
- No son portables como los volúmenes.

¿Qué son los Bind Mounts?



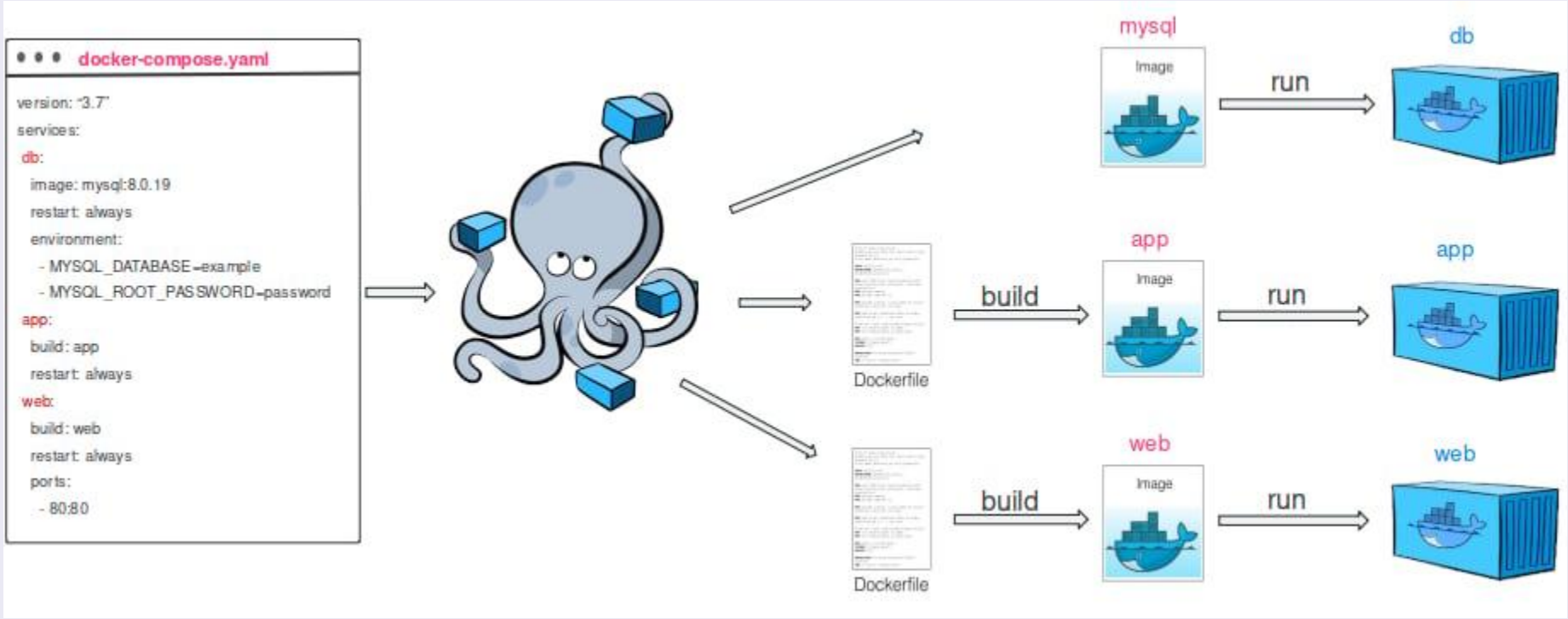
- Si se elimina el contenedor, el directorio en HOST seguirá conteniendo los datos

Sección 6: Docker Compose

¿Qué es Docker Compose?

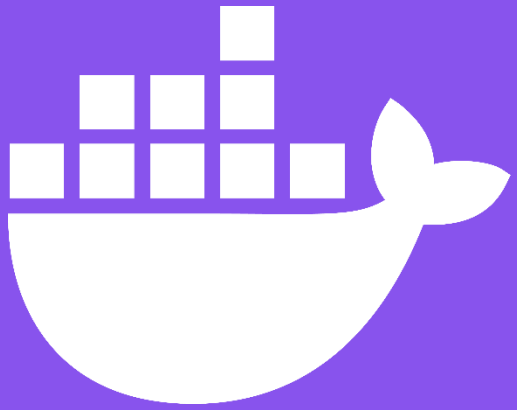
- Permite definir y ejecutar multiples aplicaciones en contenedores de forma simple y rapida.
- Simplifica el control del stack, hacienda mas facil la gestion de las redes, contenedores, volumenes, etc en un solo archivo de configuracion (.yaml).
- Posee una sintaxis simple.
- Ejemplo: en lugar de crear varios contenedores comando por comando, se define en un archivo **docker-compose.yaml** de una vez.

Flujo de Docker Compose



Comandos para Docker compose

COMANDO	DESCRIPCION
docker compose up	Crear servicios a partir del archivo .yaml
docker compose down	Para y remover lo creado
docker compose ls	Listar los proyectos utilizando compose
docker compose ps	Listar los contenedores
docker compose build	Construir o volver a construir servicios
docker compose restart	Reiniciar un servicio de compose



docker®

Sección 7: Orquestación de Contenedores

Orquestación de Contenedores

- A medida que aumenta la cantidad de contenedores, se vuelve muy complejo administrarlos.
- Realizarlos solamente con Docker puede ser ineficiente y poco práctico.
- Para solucionar este problema, existe la “Orquestación de Contenedores”.
- Permite automatizar y simplificar la administración de aplicaciones en Contenedores.
- Algunas herramientas son: **Docker Swarm**, **Kubernetes**, etc.

Herramientas de Orquestación

Kubernetes

Es la plataforma mas utilizada en la actualidad.

Aporta funcionalidades completas y sofisticadas para la gestión.

Docker Swarm

Es la solución oficial de Docker para la Orquestación de Contenedores.

Menos utilizada y menos compleja que Kubernetes.

Otras

Existen otras soluciones y alternativas como:

- **OPENSHIFT**
- **HASHICORP NOMAD**
- **PLATAFORMAS CLOUD**


docker®