TIVIT

FUNDAMENTOS DE DOCKER

TALLER

INTRODUCCIÓN

FUNDAMENTOS

TIVIT

VENTAJAS

EXPLORANDO DOCKER

TIVIT

DOCKERFILE

DOCKER COMPOSE

TIVIT

¿QUÉ ES KUBERNETES?

PREGUNTAS Y RESPUESTAS

TIVIT

INTRODUCCIÓN Y FUNDAMENTOS

¿QUÉ ES DOCKER?

Docker es una herramienta open-source que nos permite realizar una ‘virtualización ligera’, con la que poder empaquetar entornos y aplicaciones que posteriormente podremos desplegar en cualquier sistema que disponga de esta tecnología

TIVIT

INTRODUCCION Y FUNDAMENTOS

EN PALABRAS MAS SIMPLES

Docker es una herramienta que permite a los desarrolladores, administradores de sistemas, etc. **implementar fácilmente sus aplicaciones** en un entorno limitado (llamados **contenedores**) para que se ejecuten en el sistema operativo **host**, es decir, Linux. El beneficio clave de Docker es que **permite a los usuarios empaquetar una aplicación con todas sus dependencias en una unidad estandarizada para el desarrollo de software**. A diferencia de las máquinas virtuales, los contenedores no tienen una sobrecarga elevada y, por lo tanto, permiten un uso más eficiente del sistema y los recursos subyacentes.

TIVIT

INTRODUCCION Y FUNDAMENTOS

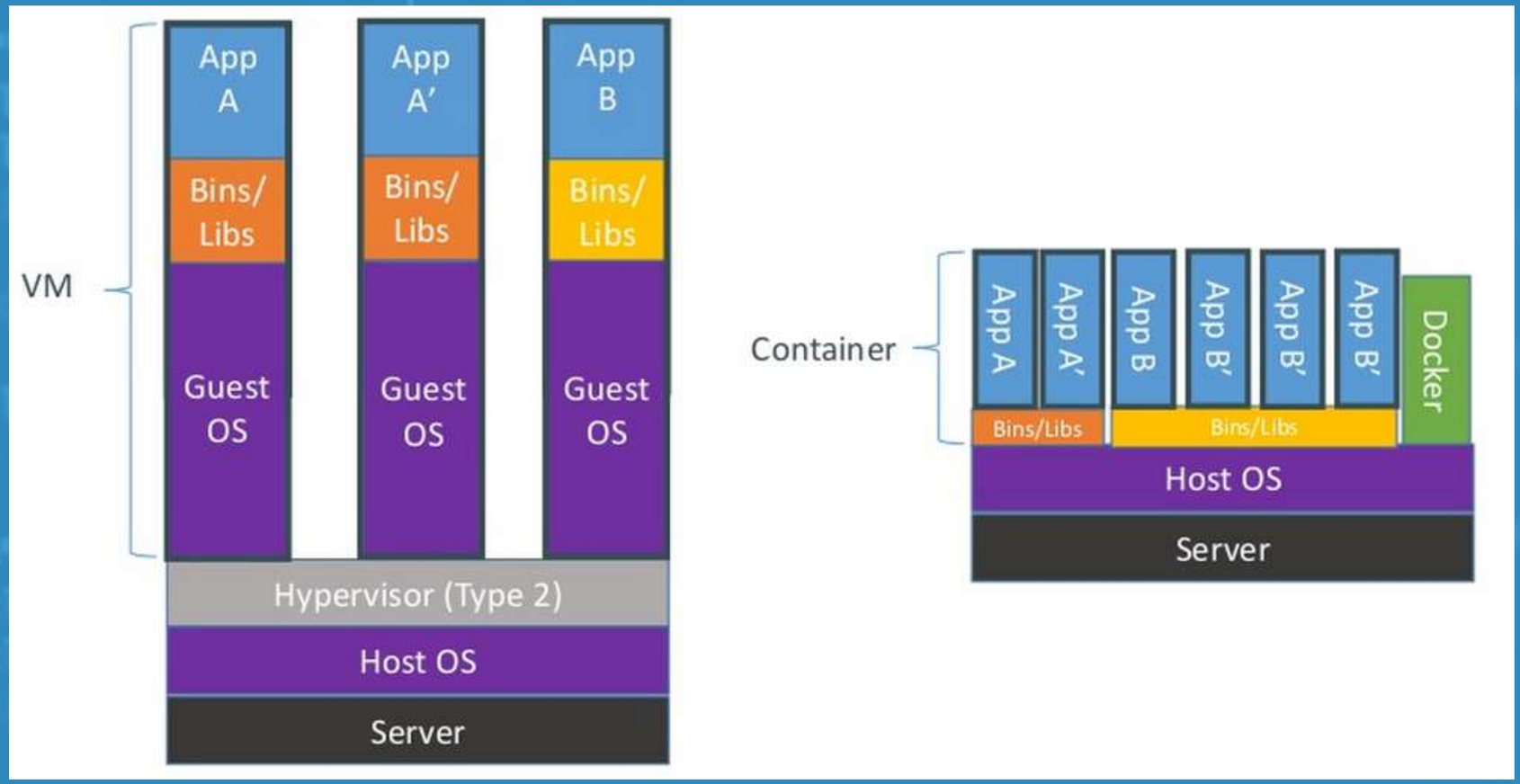
DOCKER VS MÁQUINAS VIRTUALES

La gran diferencia es que una máquina virtual necesita contener todo el sistema operativo mientras que un contenedor Docker aprovecha el sistema operativo sobre el cual se ejecuta, comparte el kernel del sistema operativo anfitrión e incluso parte de sus bibliotecas.

TIVIT

INTRODUCCION Y FUNDAMENTOS

DOCKER VS MÁQUINAS VIRTUALES



TIVIT

INTRODUCCION Y FUNDAMENTOS

Respecto al almacenamiento en disco, una máquina virtual puede ocupar varios gigas ya que tiene que contener el sistema operativo completo, sin embargo los contenedores Docker sólo contienen aquello que las diferencia del sistema operativo en las que se ejecutan, por ejemplo un Ubuntu con Apache ocuparía unos 180 Mb.

Desde el punto de vista del consumo de procesador y de memoria RAM, los contenedores Docker hacen un uso mucho más eficiente del sistema anfitrión, pues comparten con él, el núcleo del sistema operativo y parte de sus bibliotecas, con lo que únicamente usarán la memoria y la capacidad de cómputo que estrictamente necesiten.

TIVIT

VENTAJAS

¿ALGUNA VEZ TE HAN HECHO ESTA PREGUNTA?

- Alguien tiene un script para instalar automaticamente Apache?
- Alguna vez has podido llevar una aplicación rápidamente de un servidor a otro?
- Alguna ves has escalado un servicio rapidamente?

TIVIT

VENTAJAS

DESARROLLADORES DE APLICACIONES

1. Lanzamientos de calidad superior
2. Mejor escalabilidad de aplicaciones
3. Mayor aislamiento de aplicaciones

ARQUITECTOS DE TI

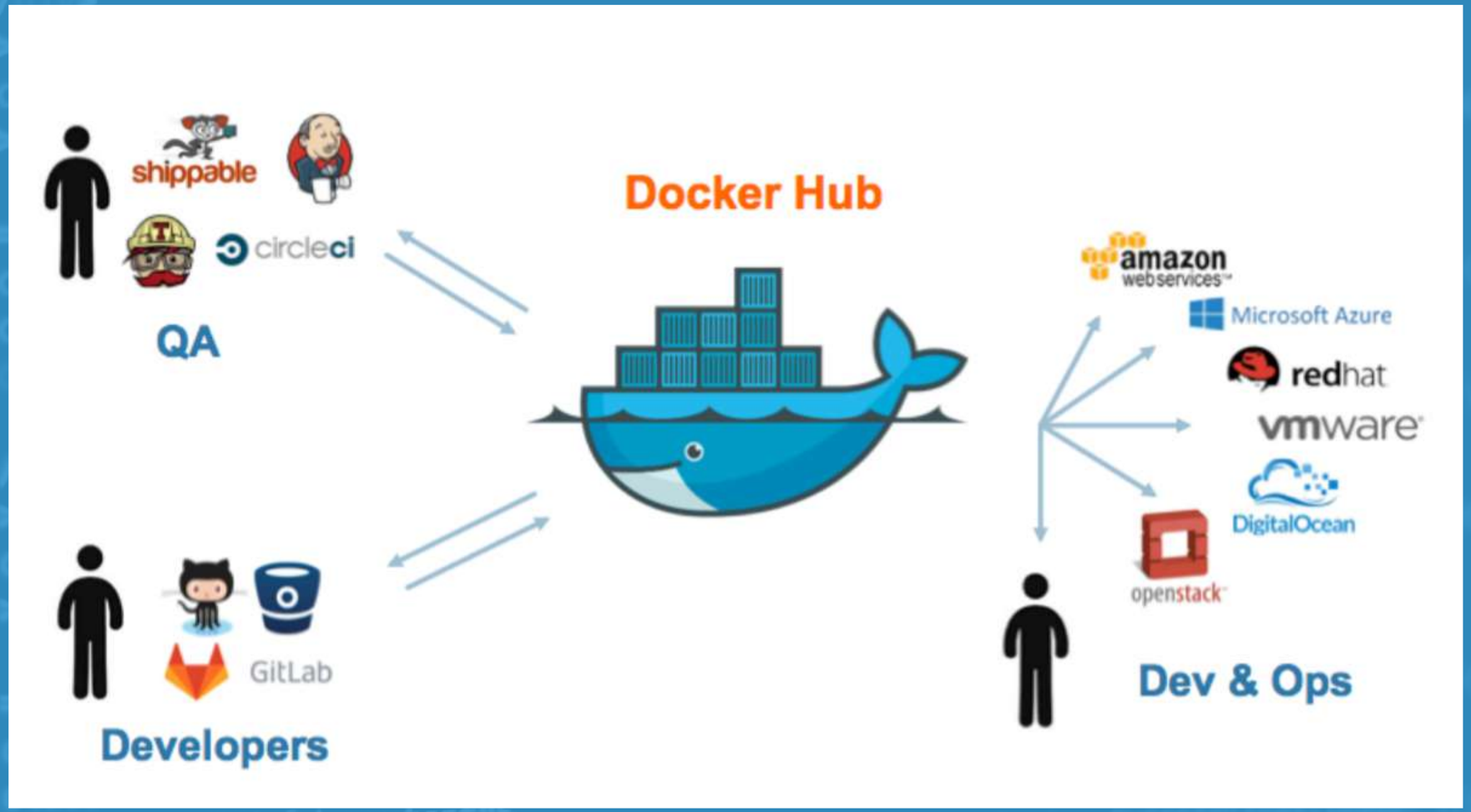
1. Escalabilidad horizontal más rápida
2. Ciclos de pruebas más cortos
3. Menos errores de implementación

OPERACIONES DE TI

1. Lanzamientos de calidad superior
2. Sustitución eficiente de todas las máquinas virtuales en producción
3. Gestión de aplicaciones más fácil

TIVIT

VENTAJAS



TIVIT

VENTAJAS

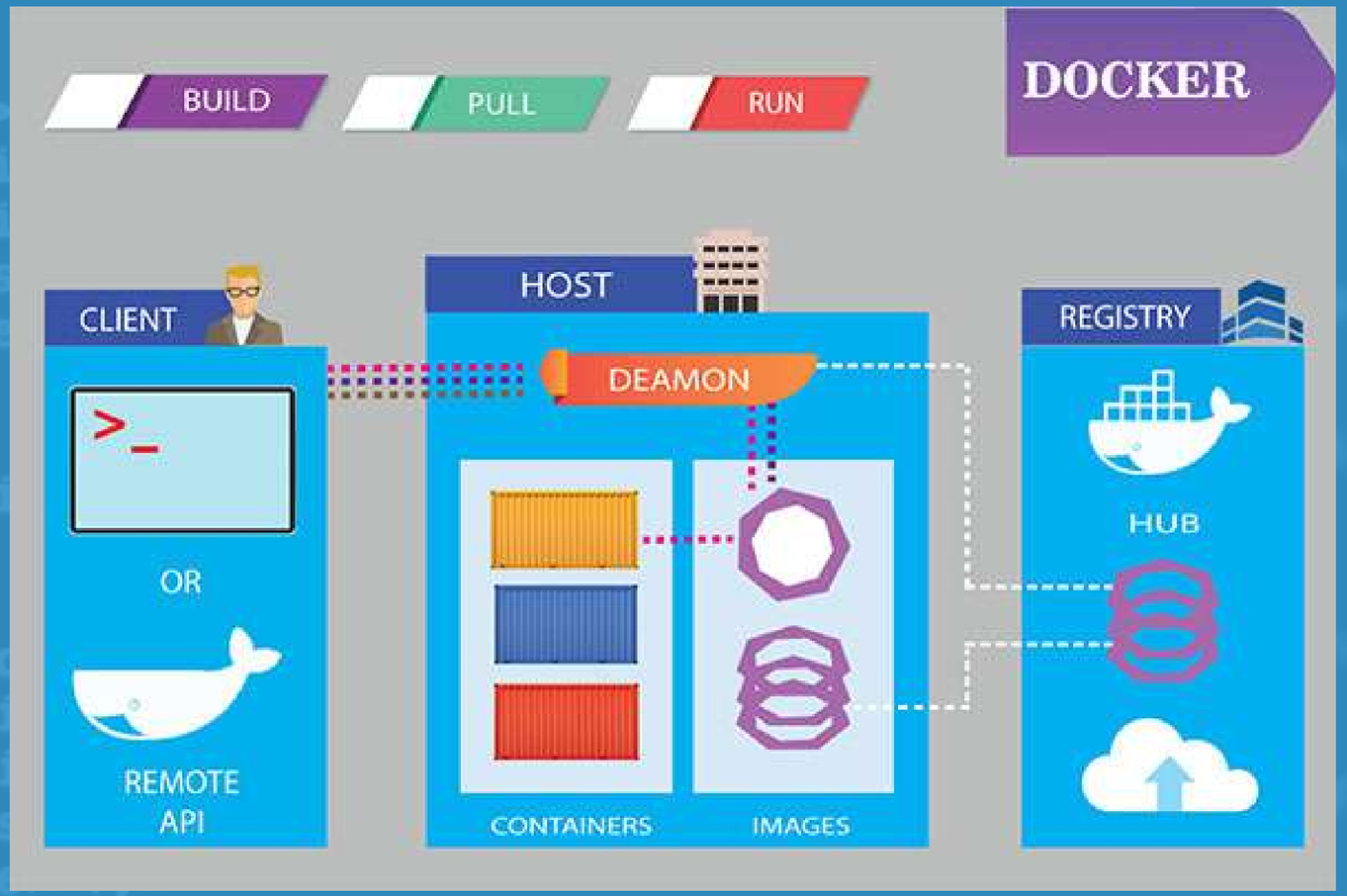
OTRAS VENTAJAS DESTACABLES DEL USO DE DOCKER SERÍAN:

- Las instancias se arrancan en pocos segundos.
- Es fácil de automatizar e implantar en entornos de integración continua.
- Existen multitud de imágenes que pueden descargarse y modificarse libremente.

TIVIT

EXPLORANDO DOCKER

ARQUITECTURA



TIVIT

EXPLORANDO DOCKER

COMPONENTES DE DOCKER

- Las imágenes de Docker: Describen los contenido que tienen el contenedor: (configuración, proceso, aplicación).
- Los registros: Las imágenes que se crean, se guardan en registros, y existen dos tipos: públicos y privados.
- Los contenedores: Es el resultado de ejecutar una imágen.

TIVIT

EXPLORANDO DOCKER

ALGO QUE ANOTAR

- El comportamiento de las imagenes y los contenedores es parecido a la programación orientada a objetos:

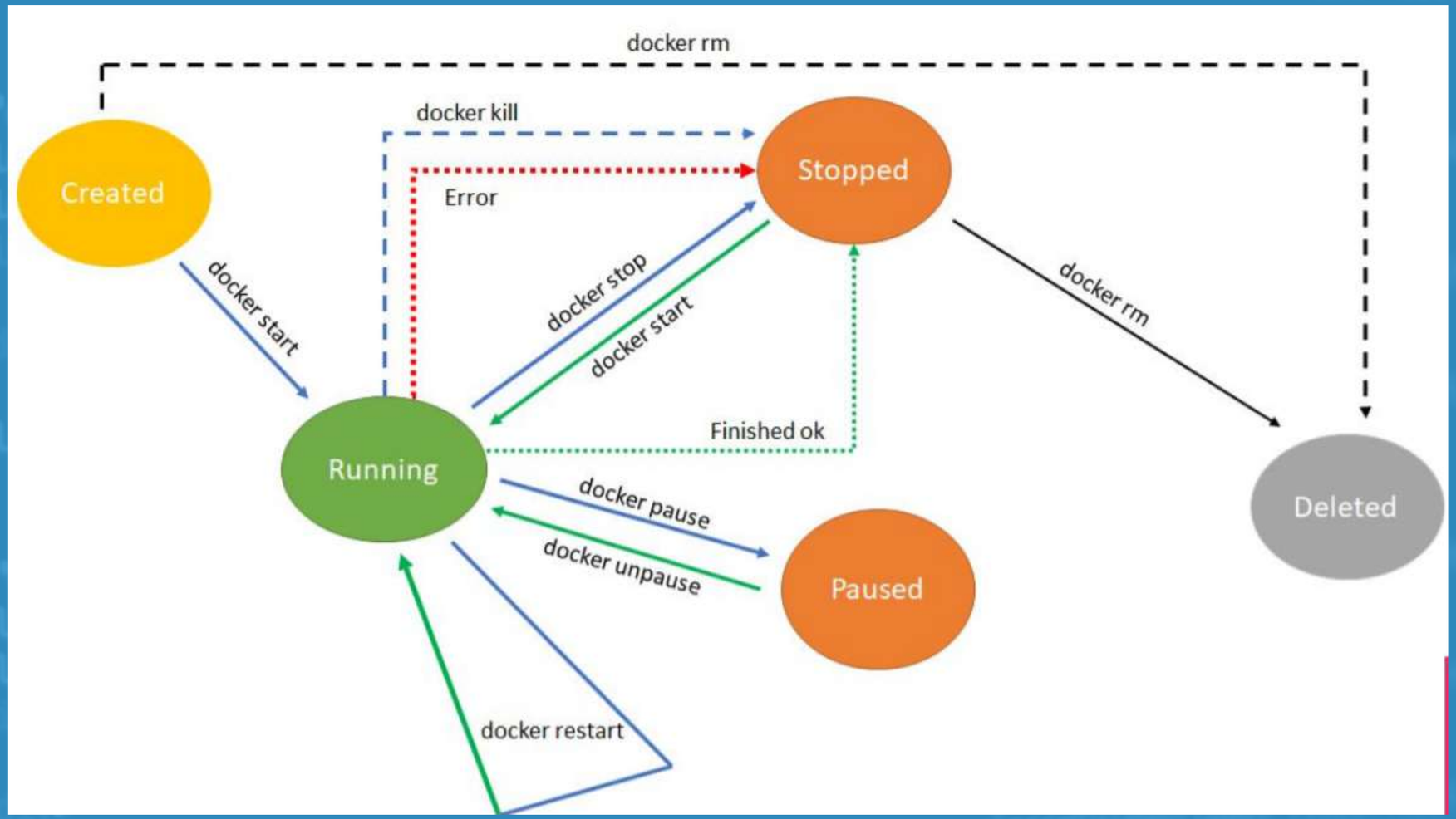
1. Clase es la imagen
2. Objeto es el contenedor.

- Las imágenes no se ejecutan.
- Los contenedores tienen un ciclo de vida.
- Las imágenes se pueden copiar entre hosts, el contenedor no.
- Solo se puede crear contenedores de imágenes descargadas en el sistema

TIVIT

EXPLORANDO DOCKER

CICLO DE VIDA DE UN CONTENEDOR



TIVIT

EXPLORANDO DOCKER

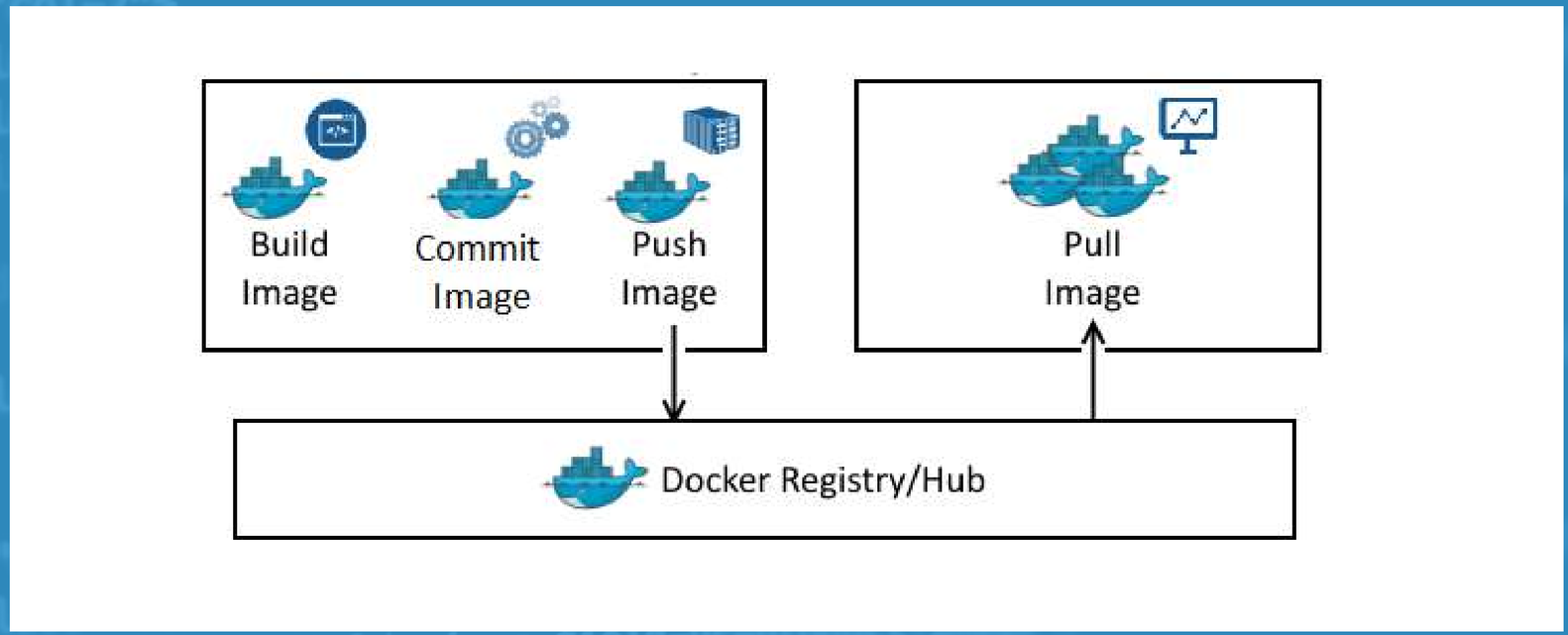
COMANDOS PRINCIPALES

- docker pull
- docker push
- docker images
- docker ps
- docker start
- docker stop
- docker restart
- docker exec
- docker attach

TIVIT

EXPLORANDO DOCKER

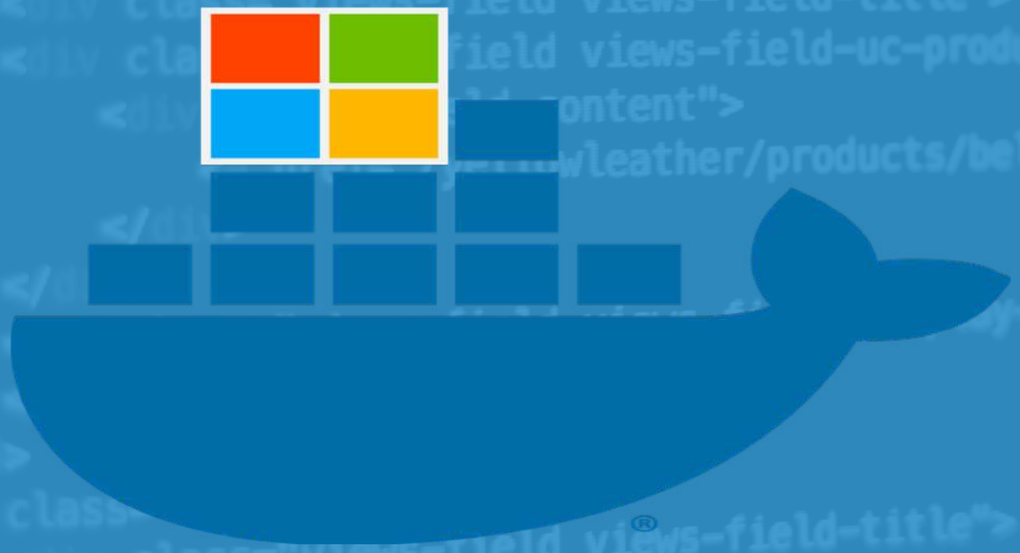
REGISTRY



TIVIT

EXPLORANDO DOCKER

IMAGES

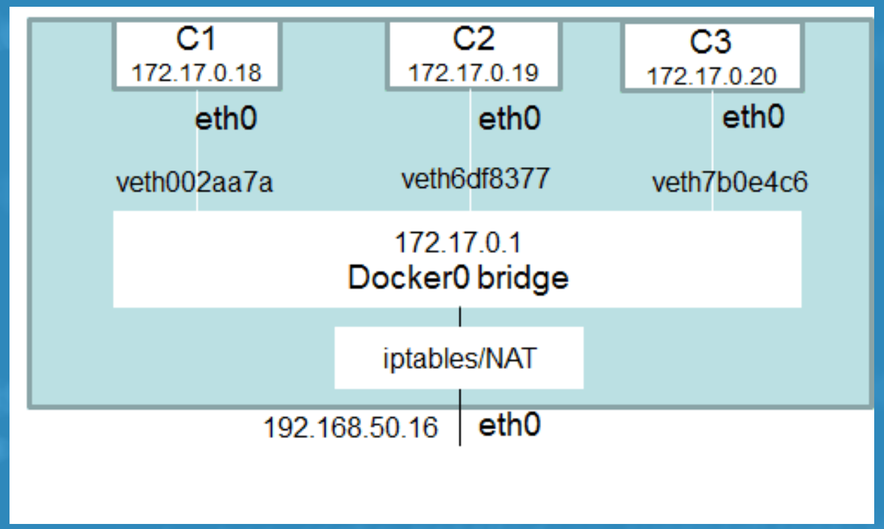


- Es de solo lectura
- Puede contener una aplicación, un sistemas operativo, etc.

TIVIT

EXPLORANDO DOCKER

REDES EN DOCKER



Existen 3 redes preconfiguradas en Docker,

Bridge. La red standard que usarán todos los contenedores.

Host. El contenedor usará el mismo IP del servidor real que tengamos.

None. Se utiliza para indicar que un contenedor no tiene asignada una red.

TIVIT

EXPLORANDO DOCKER

REDES

--net=my-network

container1 --net=my_network

container2 --net=my_network

TIVIT

EXPLORANDO DOCKER

ALGUNAS CARACTERISTICAS DE UN CONTENEDOR

- Puertos
- Volumen
- Variables

TIVIT

EXPLORANDO DOCKER

PUERTOS

-p 8081:5050

Puerto Local
8081

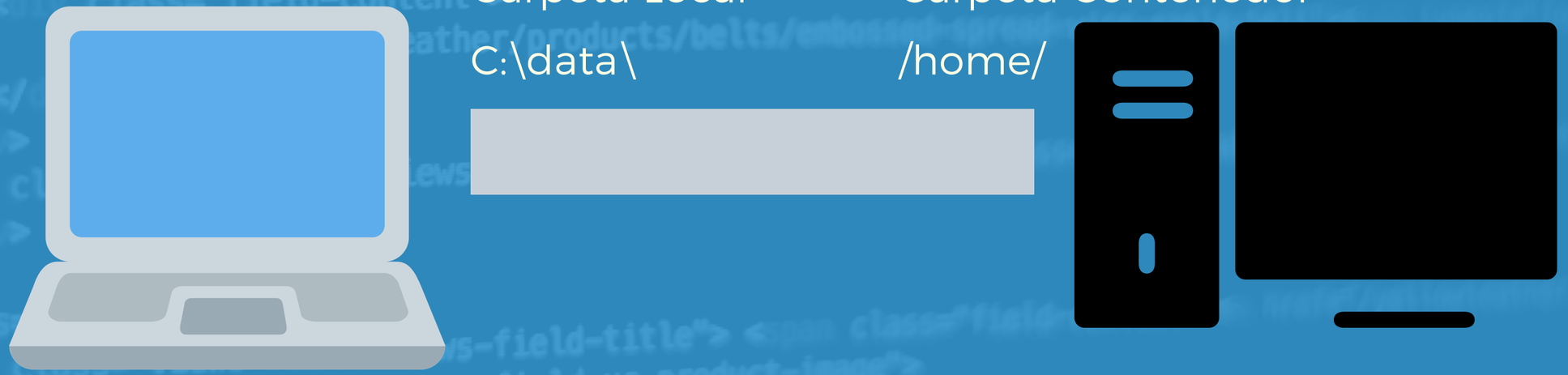
Puerto Contenedor
5050

TIVIT

EXPLORANDO DOCKER

VOLUMEN

-v C:\data\:/home/

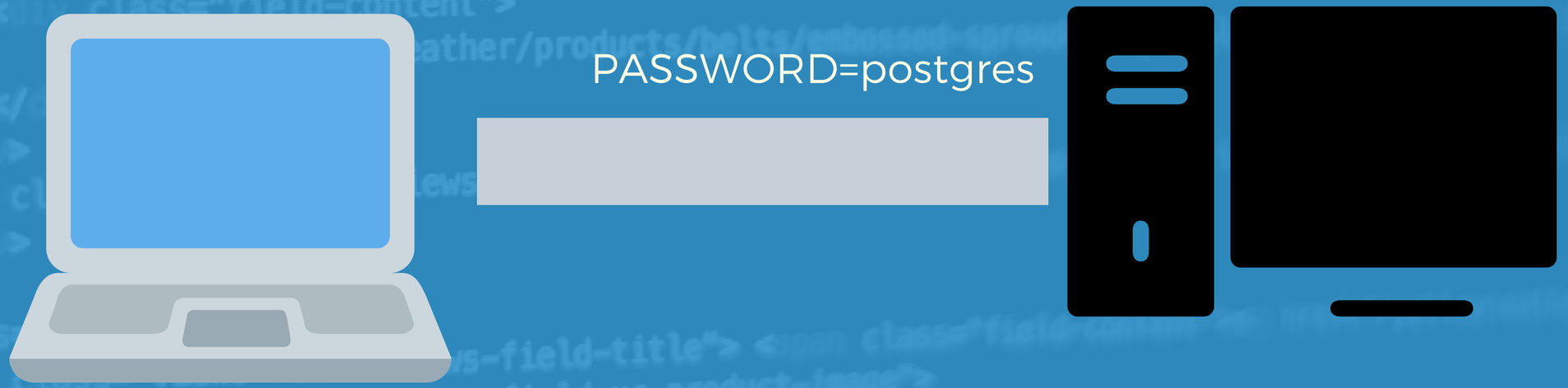


TIVIT

EXPLORANDO DOCKER

VARIABLES

```
-e POSTGRES_PASSWORD=postgres
```



TIVIT

DOCKERFILE

¿QUÉ ES DOCKERFILE?

Dockerfile es un documento de texto que contiene todos los comandos que un usuario podría llamar en la línea de comandos para ensamblar una imagen.

USO

El docker build comando crea una imagen a partir de Dockerfile.

TIVIT

DOCKERFILE

Dockerfile

```
FROM microsoft/dotnet:sdk AS build-env
WORKDIR /Docker

# Copy csproj and restore as distinct layers

RUN dotnet restore

# Copy everything else and build
COPY . ./
RUN dotnet publish -c Release -o out


# Build runtime image
FROM microsoft/dotnet:aspnetcore-runtime
WORKDIR /app
COPY --from=build-env /app/out .
ENTRYPOINT ["dotnet", "aspnetapp.dll"]
```

TIVIT

DOCKER COMPOSE

¿QUÉ ES DOCKER COMPOSE?

Compose es una herramienta para definir y ejecutar aplicaciones Docker de contenedores múltiples.
Con Compose, utiliza un archivo YAML para configurar los servicios de su aplicación. Luego, con un solo comando, crea e inicia todos los servicios desde su configuración.

USO

Compose funciona en todos los entornos: producción, puesta en escena, desarrollo, pruebas, así como flujos de trabajo de CI.

TIVIT

DOCKER COMPOSE

```
version: '2'
services:
  database:
    build: ./Docker/Database/
    ports:
    - "5432:5432"
  web1:
    build: .
    ports:
    - "5000:5000"
    links:
    - database
  web2:
    build: .
    ports:
    - "5001:5000"
    links:
    - database
  loadbalancer:
    build: ./Docker/loadbalancer/
    ports:
    - "80:80"
    links:
    - web1
    - web2
    command: haproxy -f /usr/local/etc/haproxy/haproxy.cfg
```

TIVIT

RECURSOS

- https://docs.docker.com/
- https://docker-curriculum.com
- https://riptutorial.com/Download/docker-es.pdf.

TIVIT