UD5: INTRODUCCIÓN A DOCKER

2º DAW – DESPLIEGUE DE APLICACIONES WEB

IES Macia Abela

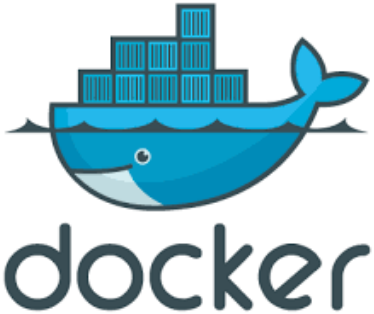
Índice

- Introducción
  - Contenedores
  - Características
  - Arquitectura
  - Imágenes vs Contenedores
  - DockerHub
  - DockerFile
    - Directivas
  - Volúmenes
  - Redes
  - Terminología
- Instalación

Introducción

I.E.S. MACIA ABELA

- Docker (estibador en inglés) → Es un **Sistema de Virtualización de Aplicaciones mediante contenedores**, creado por Solomon Hykes y su equipo de ingenieros.
- En **2013 se convirtió en un proyecto de software libre (licencia Apache)** en el que participan cada vez más empresas.
- Versión 1.0 publicada en junio de 2014.



Introducción

I.E.S. MACIA ABELA

- Objetivos
  - **Solucionar problemas y errores de dependencias entre los diferentes sistemas operativos** de los desarrolladores y los entornos de pruebas y producción.
  - **Evitar** la **elevada carga y consumo** de recursos de las **máquinas virtuales**.
  - **Cubrir diferentes tipos de despliegue**:
    - Monolítico: Todos los servicios en la misma máquina.
    - SOA (Service Oriented Architecture): Diferentes máquinas, una con cada servicio, conectadas.
    - Microservicios: División más pequeña de los servicios.

Introducción

I.E.S. MACIA ABELA

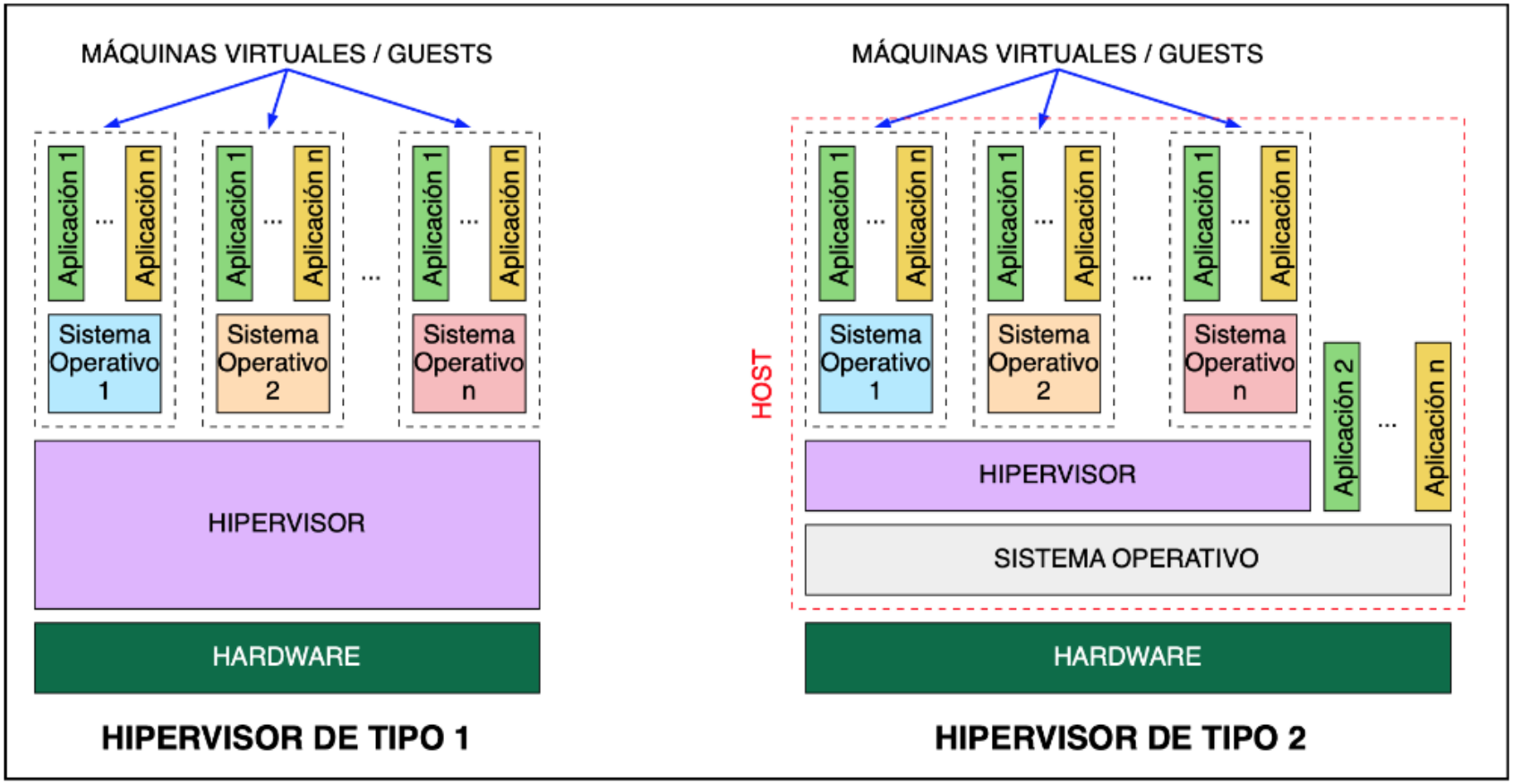
- **Virtualización**
  - Objetivo: **Utilizar simultáneamente en un mismo dispositivo dos o más sistemas operativos.**
  - Requiere de un **hipervisor**.
  - **Ventajas**:
    - **Ejecutar aplicaciones** que **no** están **disponibles para el sistema operativo host** (anfitrión).
    - Permite la **conservación del software**. Sistemas operativos antiguos pueden instalarse y seguir utilizándose.
    - Permite **mejorar el aprovechamiento del hardware**, ya que un mismo ordenador puede contener muchos sistemas operativos, utilizados por usuarios diferentes, aislados unos de otros.

Introducción

I.E.S. MACIA ABELA

- Hipervisor: **Elemento fundamental en la virtualización**.
  - Se distinguen dos tipos:
    - **Tipo 1 o "bare metal"**, son hipervisores que están en contacto directo con el hardware de la máquina, **sin necesidad de ningún sistema operativo** previo.
    - **Tipo 2 o "alojados"**, son una aplicación más del sistema operativo instalado en la máquina. **Accede al hardware a través de ese sistema operativo**.

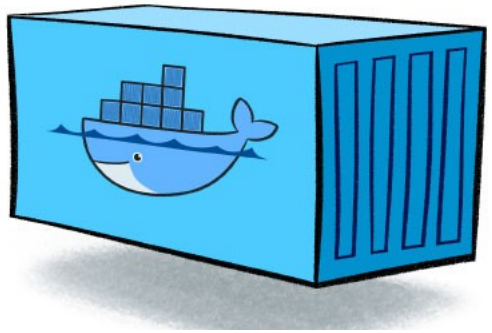
Introducción



Contenedores

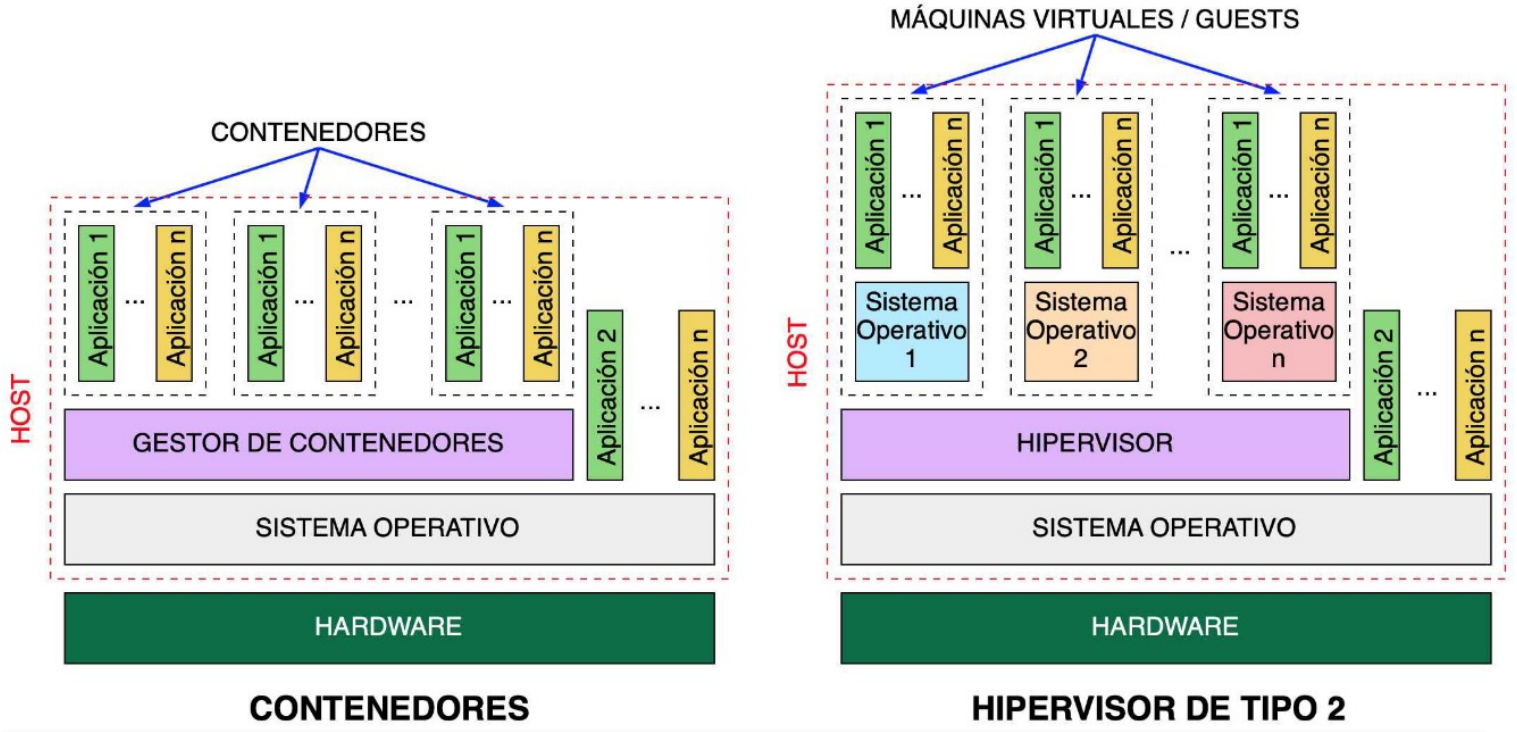
I.E.S. MACIA ABELA

- Consiste en **Agrupar y Aislar Aplicaciones** o grupos de aplicaciones que se **ejecutan sobre un mismo núcleo** de sistema operativo.
- Característica principal: Tiene su **propio sistema de archivos,** pero se puede ejecutar en cualquier sistema operativo.
- **No es necesario emular el hardware y software completo** de las máquinas virtuales → **LIGEROS**



Contenedores

- **Máquinas Virtuales** → ocupan mucho espacio en memoria, disco y mayor tiempo de ejecución. Emulan el hardware y el software.
- **Contenedores** → Ligero y Portable. Soluciona problemas de espacio y compatibilidades en desarrollos de software.



Características

- Docker es **Open Source**
- Docker **genera un proceso aislado** del resto de los procesos de la máquina gracias a ser ejecutado sobre su propio sistema de ficheros, su propio espacio de usuarios y procesos, sus propias interfaces de red…
- Docker es **modular**, esta dividido en varios componentes.
- Docker es **portable e inmutable**, utilizando la plataforma DockerHub.
- Su lema es “Build, Ship and Run, any app”.

Características

I.E.S. MACIA ABELA

- Docker es **un contenedor de aplicaciones.**
- Puede **ejecutarse en una máquina física, virtual, ….**
- Las aplicaciones que mejor se ajustan al despliegue en contenedores son las desarrolladas con **microservicios:**
  - **Cada componente** de la aplicación “microservicio” se puede desplegar **en un contenedor**.
  - **Comunicación vía HTTP REST** y colas de mensajes.
  - **Facilita** enormemente las **actualizaciones de versiones** de cada componente.
- Un **contenedor ejecuta un proceso y cuando termina la ejecución** el contenedor **se para** (no es como un sistema operativo al uso normal).

Arquitectura Docker

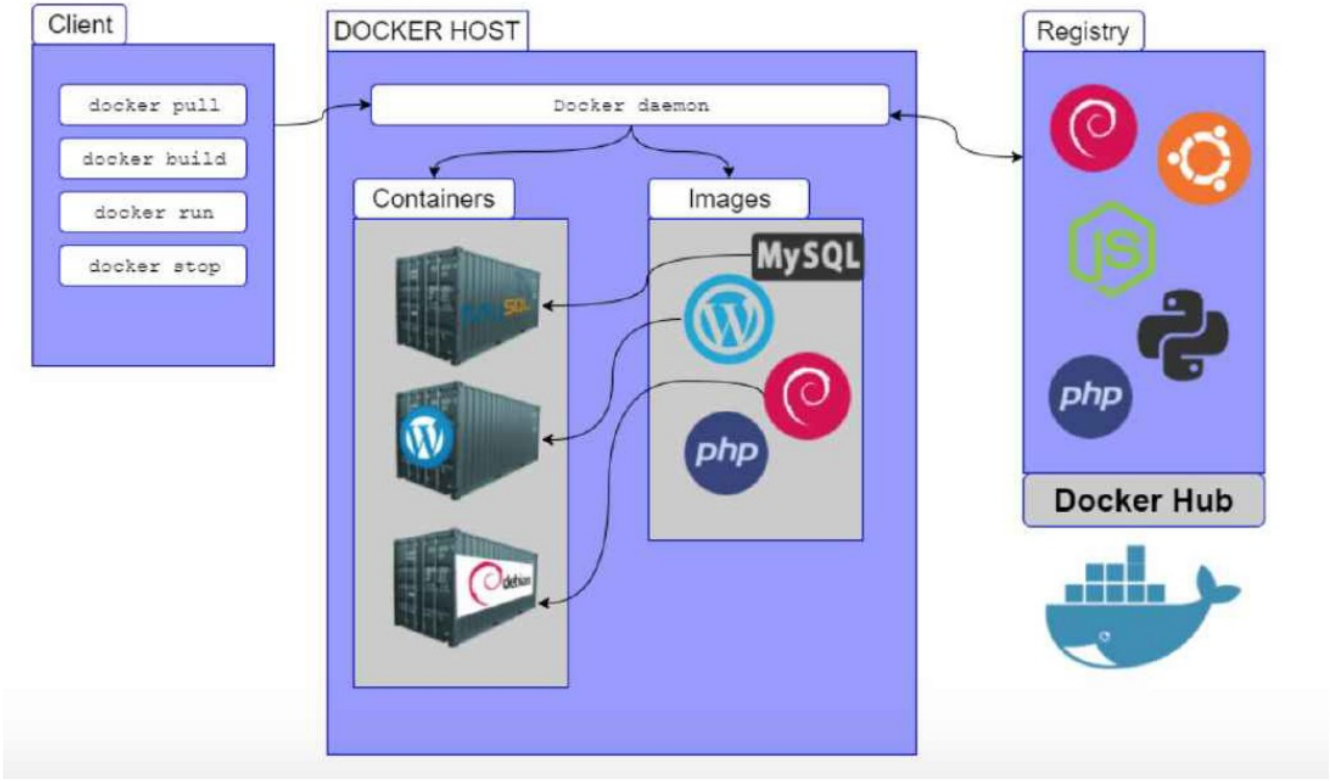
I.E.S. MACIA ABELA

- **Docker Engine**: Motor del gestor. Basado en la arquitectura cliente-servidor (pueden estar en la misma máquina o en distintas), y se realiza mediante un API REST (usa HTTP).
- **Daemon**: Servicio. Lleva a cabo la gestión y enlace de los componentes del gestor.
- **Imágenes**: Especie de plantillas que contienen como mínimo todo el software que necesita la aplicación para ponerse en marcha. Formadas por una colección ordenada de: sistemas de archivos, repositorios, comandos, parámetros y aplicaciones.
- **Contenedores**: Conjunto de procesos que encapsulan e identifican una imagen. Se pueden crear, inicializar, parar, volver a ejecutar y destruir.

Arquitectura Docker

I.E.S. MACIA ABELA

- **Registros**: las imágenes se guardan en registros para almacenarlas y distribuirlas. Registro público se utiliza **Docker Hub** o Registro privado (instalado en un servidor local).



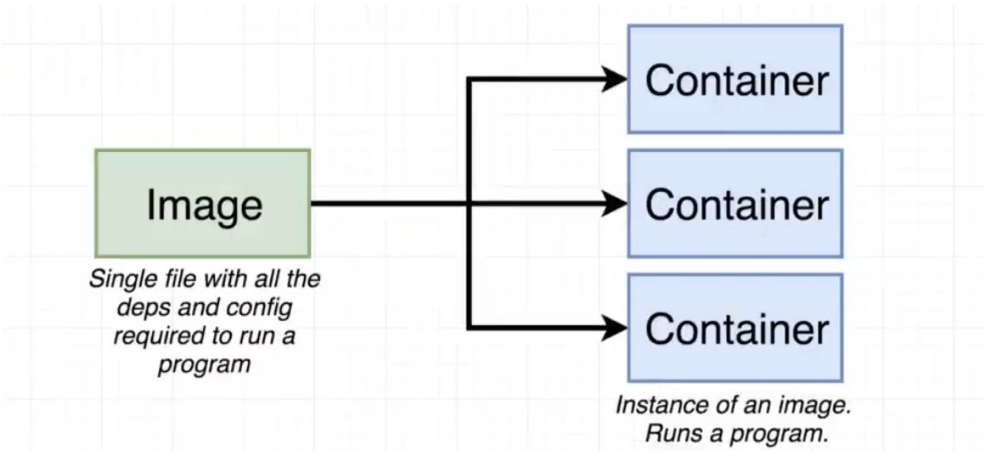
Imágenes vs Contenedores

I.E.S. MACIA ABELA

- Una **imagen** consiste en **definir que aplicación queremos ejecutar.**
- Una **imagen** contiene el **conjunto de ficheros binarios, con sus dependencias que forman una aplicación o pieza software.**
- Una **imagen** tiene asociados **metadatos e instrucciones sobre cómo ejecutar contenedores que se creen a partir de ella**.
- Una **imagen no es un sistema operativo completo**, no contiene un kernel o módulos del kernel (drivers, etc..), ya que es el host (anfitrión) el que provee todo eso.

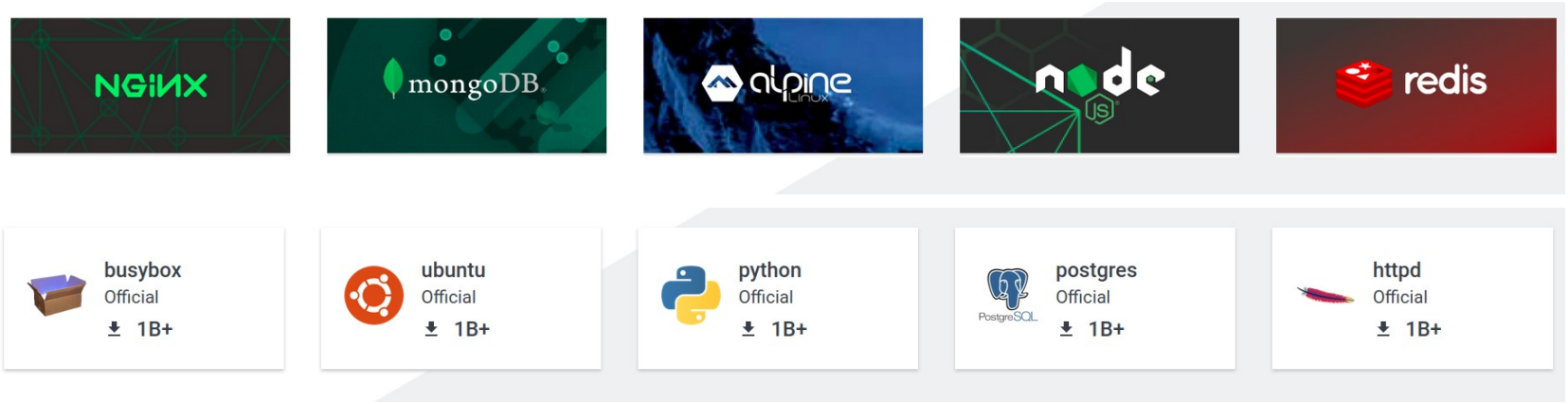
Imágenes vs Contenedores

- Un **contenedor** es una **instancia de una imagen**, **ejecutándose como un proceso** en el sistema operativo host (o anfitrión).
- Se pueden lanzar **uno o más contenedores basados en la misma imagen**.
- Si realizamos **cambios en el contenedor ya lanzado**, al detenerlo **esto no se verá reflejado en la imagen**.



Docker Hub

- Se trata de un **repositorio** donde están **alojadas las imágenes base** que podemos utilizar en nuestros contenedores.
- Puede alojar imágenes **públicas y privadas**.
- **Gratuito para imágenes públicas**.
- Web Oficial: https://hub.docker.com/



Dockerfile

I.E.S. MACIA ABELA

- **Archivo de texto plano** que **contiene** una serie de **instrucciones necesarias** que nos permiten **crear imágenes**.
- Ejemplo: https://github.com/sismics/docker-apache2/blob/master/Dockerfile

```
4
5  FROM ubuntu:focal
6  MAINTAINER Jean-Marc Tremeaux <jm.tremeaux@sismics.com>
7
8  ENV DEBIAN_FRONTEND noninteractive
9  RUN apt-get update && apt-get install -y apache2 curl unzip && apt-get clean && rm -rf /var/lib/apt/lists/*
10
11 ENV APACHE_RUN_USER www-data
12 ENV APACHE_RUN_GROUP www-data
13 ENV APACHE_LOG_DIR /var/log/apache2
14 ENV APACHE_PID_FILE /var/run/apache2.pid
15 ENV APACHE_RUN_DIR /var/run/apache2
16 ENV APACHE_LOCK_DIR /var/lock/apache2
17
18 RUN mkdir -p /var/lock/apache2
19 COPY opt /opt
20 COPY etc /etc
21
22 RUN ln -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/
23 RUN ln -s /etc/apache2/mods-available/remoteip.load /etc/apache2/mods-enabled/
24
25 EXPOSE 80 443
26
27 CMD ["/opt/bin/start-apache.sh"]
```

Dockerfile - Directivas

I.E.S. MACIA ABELA

- **FROM:** Requerido. Distribución mínima a partir de la cual crear la imagen. Primero buscará la imagen en local y si no la encuentra la descargará de internet. La más común es alpine. Otras distribuciones: Ubuntu, Debian, Fedora...
- **MAINTAINER**: Datos de la persona que mantiene el contenedor.
- **ENV:** Nos permite configurar las variables de entorno en el contenedor.
- **COPY/ADD**: Copia un archivo o directorio al contenedor.
- **RUN:** Utilizada para instalar paquetes adicionales en el contenedor, editar ficheros dentro del contenedor, etc. Puede lanzar scripts que han sido previamente copiados dentro del contenedor.

Dockerfile - Directivas

I.E.S. MACIA ABELA

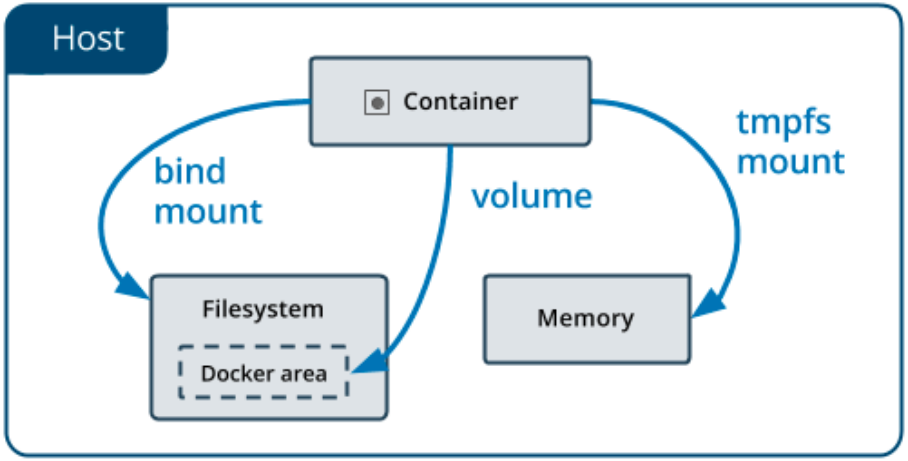
- **EXPOSE:** Indica que el contenedor escucha en los puertos especificados durante su ejecución.
- **CMD:** Comando final que se ejecutará cuando se cree un contenedor (o arranque uno existente) a partir de una imagen. Solo puede existir una instrucción CMD en un Dockerfile, si colocamos más de una solo la última tendrá efecto.
- **ENTRYPOINT**: Permite indicar el comando que queremos que se ejecute de forma indefinida en nuestro contenedor.
- **WORKDIR**: Define el directorio de trabajo.
- **VOLUME**: Crea un volumen.

- Herramienta para comprobar ficheros dockerfile:

  https://hadolint.github.io/hadolint/

Volúmenes

- Los **datos de los contenedores son efímeros**, es decir, los **ficheros, datos y configuraciones que creamos en los contenedores** sobreviven a las paradas de los mismos pero sin embargo, son **destruidos si el contenedor es destruido**.
- Los **volúmenes** son el **mecanismo** que utiliza Docker **para** hacer **persistentes los datos de un contenedor**.



Volúmenes

I.E.S. MACIA ABELA

- Docker nos proporciona **dos técnicas** para **persistir los datos** de los contenedores:
  - **Directorios enlazados (bind mount)**: la información se guarda fuera de Docker, en la máquina host. En este tipo, una ruta del contenedor enlaza con una ruta en el sistema de archivos del host.
    - Mapear una parte de nuestro sistema de ficheros, de esta forma se comparten ficheros entre el host y los contenedores.
    - Otras aplicaciones que no sean Docker tienen acceso a esos ficheros para realizan operaciones sobre ellos.

Volúmenes

- Docker nos proporciona **dos técnicas** para **persistir los datos** de los contenedores:
  - **Volúmenes**: la información se guarda mediante Docker, en unos elementos llamados volúmenes independientes de las imágenes y los contenedores.
    - Permiten compartir datos entre contenedores.
    - Pueden ser gestionados por Docker y el acceso a estos volúmenes solo se puede realizar a través de algún contenedor que lo utilice.
    - Permiten almacenarlos en hosts remotos o proveedores cloud.
    - Pueden administrarse mediante comandos o API de docker.

Redes

I.E.S. MACIA ABELA

- Cada vez que creamos un **contenedor**, este **se conecta a una red virtual** y **docker realiza una configuración del sistema** (usando interfaces puente e iptables) para que la **máquina tenga una ip interna**, tenga acceso al exterior, se puedan mapear puertos, etc.

- Tipos de **redes predefinidas** en Docker:
  - **Bridge**: red por defecto. Todos los contenedores están en la misma red, separada del host. Los contenedores conectados a esta red que quieren exponer algún puerto al exterior tienen que usar la opción -p para mapear puertos.
    - Aisla contenedores del acceso exterior.
    - Publica servicios contenidos en los contenedores mediante redirecciones de puertos y reglas iptables.

Redes

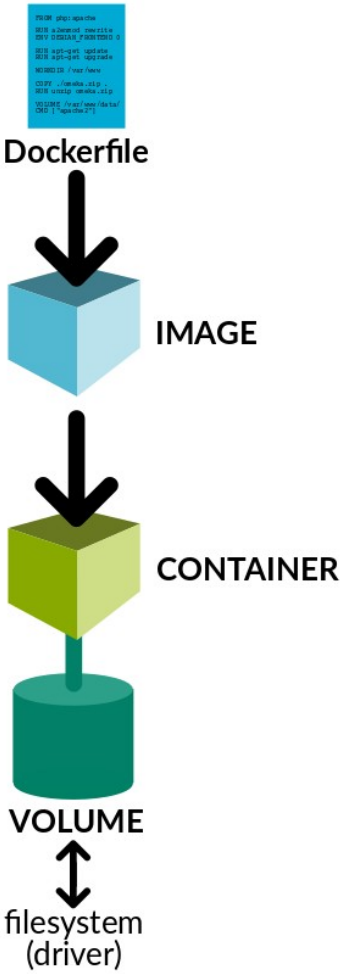
I.E.S. MACIA ABELA

- Tipos de **redes predefinidas** en Docker:

  - **Host**: El contenedor ofrece el servicio que tiene configurado en el puerto de la red del anfitrión. No tiene ip propia, sino que es cómo si tuviera la ip del anfitrión. Los puertos son accesibles directamente desde el host.

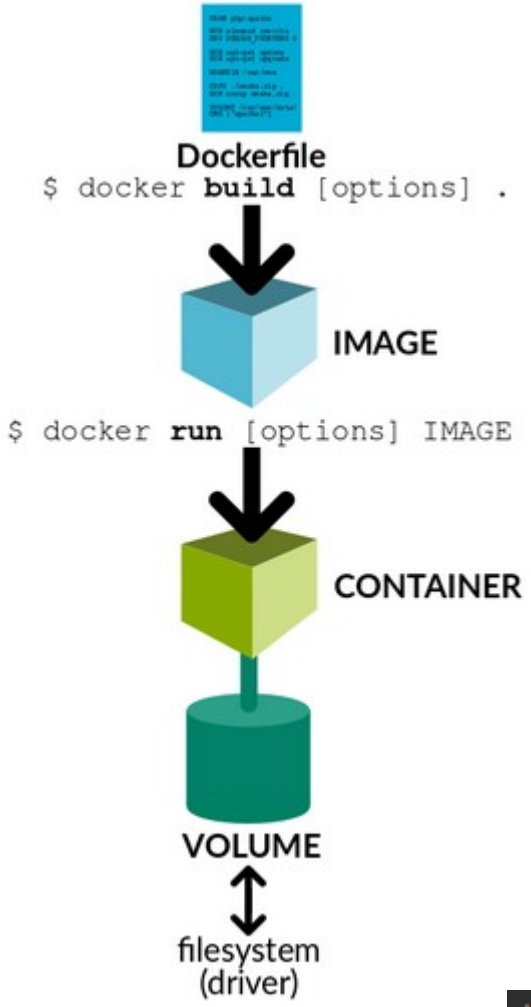
  - **None**: No configurará ninguna IP para el contenedor y no tiene acceso a la red externa ni a otros contenedores (contenedores aislados). Tiene la dirección de loopback y se puede usar para ejecutar trabajos por lotes.

Terminología



- **Dockerfile**: Instrucciones para construir la imagen.
- **Imagen**: El modelo abstracto de contenedor.
- **Contenedor**: La concreción de una imagen abstracta.
- **Volumen**: donde almacenar los datos persistentes del contenedor.

Terminología



Dockerfile: Este fichero se construye "build" para crear una "imagen".

```
$ docker build [options]
```

Imagen: La "imagen" se ejecuta "run" para crear un "contenedor"

```
$ docker run [options] image-name
```

Contenedor: En su definición (dockerfile) o al iniciarlos, estos contenedores incluyen o se les asocian "volúmenes"

```
$ docker run [options] -v ./web:/var/www/html img-name
```

Instalación

I.E.S. MACIA ABELA

- **Docker para Mac:**
  - https://docs.docker.com/desktop/install/mac-install/
- **Docker para Windows**:
  - https://docs.docker.com/desktop/install/windows-install/
- **Docker para Ubuntu:**
  - https://docs.docker.com/engine/install/ubuntu/
- **Probar docker en el navegador:**
  - https://labs.play-with-docker.com/

¿Preguntas?