

# **DOCKER**

Docker ejecuta los procesos de forma aislada, pero en tu propia maquina, de forma "mágica" con los Cgroups y namespaces

# Dockerfiles y docker-compose

#### **Sentencias**

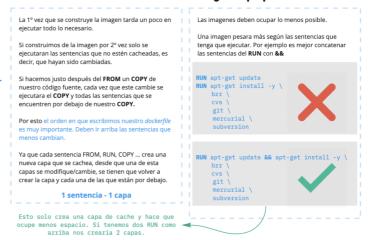
FROM	Crear la imagen desde otra imagen oficial		
RUN	Va seguido de los comanados a ejecutar dentro del contenedor		
CMD	Comando que se ejecutara al hacer docker run	Si no se indica, se ejecuta el de la imagen del FROM. Se ejecuta como argumento de lo que se indica en el ENTRYPOINT	
LABEL	Etiquetar la imagen		
EXPOSE	Exponer puertos		
ADD	Copia en el contenedor, permite poner URL's		
COPY	Copia en el contenedor, solo rutas locales		
ENV	Variables de entorno a incluir en el conteredor		
ENTRYPOINT	Se ejecuta siempre que se arranca un contenedor, lo que se indique aquí	default: "/bin/sh", "c" Si no se indica, se ejecuta el de la imagen del <b>FROM</b>	

# Construir imagen (Repositorio de imagenes <u>hub.docker.com</u>)



#### Docker cache

#### Imagenes pequeñas

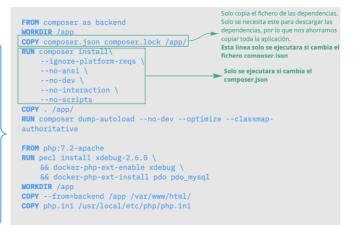


## Multistage build

Es un *dockerfile* con varias sentencias **FROM**. Nos permiten separar la imagen de *build*, donde generamos nuestra aplicación, de la imagen que finalmente ejecutaremos, haciendo que ocupe menos y sea más segura.



### **Good practice**



#### docker-compose

Es un fichero de configuración .**yml** en el que se especifica como se levantan los contenedores evitando hacerlo de forma manual (*docker run* ...).

	build	Construye las imágenes definidas en el fichero docker-compose
	up	Levanta (y construye, si no lo está) los contenedores especificados en el
	up -d	Igual modo <b>detach</b>
	upbuild	Fuerza a construir la imagen
dl	ps	Lista los contenedores levantados
docker-compose	logs -f nameService	Muestra los logs del servicio indicando el nombre del servicio
	top	Muestra los procesos en ejecución
	stop	Para los contenedores especificados en el fichero
	down	Elimina los contenedores parados

#### Volumen

Sirve para persistir los datos de por ejemplo una base de datos, de forma que si el contenedor se para o se elimina no perdemos la información ya que la tendremos en local.

También vale para publicar aplicaciones de lenguajes interpretados, ya que simplemente tenemos que tener en el volumen los ficheros generados para desplegar la aplicación.

### docker-compose.yml

