Crear imágenes en Docker

Docker Build

docker build es el comando para generar imáganes en base a un fichero **Dockerfile** que describe la imágen y el contexto que suele ser el directorio de trabajo local donde van a estar los ficheros

Dockerfile

Dockerfile es un ficherto de texto que contienen una serie de instrucciones del tipo script, que se utiliza para construir automáticamente imágenes. Su función es automatizar la creación de imágenes, permitiendo facilmente repetir o modificar la creación de imágenes.

Instrucciones en Dockerfile

https://docs.docker.com/engine/reference/builder/

- **FROM image** indicamos que imagen base usar para nuestro contenedor.
- **MAINTAINER <nombre> <correo> <cualquier_info>** para describir al autor
- **RUN comando** para ejecutar un comando en el contexto de la imagen.
- **WORKDIR path** definimos el directorio de trabajo en el contenedor.
- **ENV var=value** definimos variables de entorno en el contenedor
- **EXPOSE puerto**: indicamos puertos donde el contenedor acepta conexiones.
- **VOLUME path** definimos volúmenes en el contenedor. Creamos un punto de montaje que nos permite compartir archivos con la máquina anfitriona o con otros contenedores.
- **COPY origen destino** para copiar ficheros dentro de la imagen. También se usa para multi-stage builds.
- **ADD <fuente>..<destino>** permite copiar ficheros dentro de una imagen, pero a diferencia de copy permite usar URLs remotas o archivos comprimidos con tar
- **USER** usuario a usar cuando se lanza un contenedor y para la ejecución de cualquier instrucción RUN, CMD
- **LABEL** añade metadatos a una imagen.

Ejemplo de Dockerfile

```
#Utilizamos la imagen base de ubuntu 16.04
FROM ubuntu: 16.04

# instalamos el servidor web apache2, para reducir el tamaño, borramos la caché de paquetes apt y la lista de paquetes descargada
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*

#  exponemos el puerto http TCP/80, copiamos el fichero index.html al DocumentRoot
EXPOSE 80
```

```
ADD ["index.html","/var/www/html/"]

#  indicamos el comando que se va a ejecutar al crear el contenedor, y
además, al usar el comando ENTRYPOINT, no permitimos ejecutar ningún otro
#comando durante la creación.
ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

Si nos fijamos en el ejemplo todas las instrucciones como FROM, MAINTAINER y FROM se han escrito con mayúsculas, esto no es un requisito sino mas bien una buena práctica al escribir nuestro archivo Dockerfile.

Una vez creado nuestro fichero dockerfile para generar la imagen suponiendo que estamos en el mismo directorio que nuestro fichero dockerfile **docker build <ruta>** en el caso de que estamos ejecutando el comando desde el mismo directorio donde está el dockerfile

```
docker build .
```

Para eliminar una imagen

```
docker image rm <id>
```

Etiquetas

Añadirle etiquetas a nuestras imágenes nos va a servir para añadir en los metadatos de la imagen información que nos permita identificar a nuestras imágenes. por ejemplo, para crear una imagen que llamaremos holamundo con una etiqueta 1.0 suponiendo que estamos en el directorio donde se encuentra el dockerfile sería:

```
docker build -t holamundo:1.0 .
```

Referencias

- https://docs.docker.com/engine/reference/builder/
- https://www.josedomingo.org/pledin/2016/02/ejemplos-de-ficheros-dockerfile-creando-imagenes-docker/
- https://www.josedomingo.org/pledin/2016/02/dockerfile-creacion-de-imagenes-docker/
- http://magmax.org/blog/docker/