## Descripción

Desde Docker se pueden**construir imágenes automáticamente**, detallando las instrucciones en un fichero Dockerfile. Se trata de un documento de texto que contiene todas las órdenes a las que se puede llamar, desde la línea de comandos, para crear una imagen.

Para ello crearemos un directorio que contenga el fichero Dockerfile con el guion y otros ficheros que fuesen necesarios para crear la imagen. Seguidamente se construirá la imagen con el comando**dockerbuild**

Las opciones más comunes de dicho comando son son:

* **-t, nombre [:etiqueta]** Crea una imagen con el nombre y la etiqueta especificada a partir de las instrucciones indicadas en el fichero.
* **--file, -f.** Indica el nombre del Dockerfile, por defecto “Dockerfile”.
* **--no-cache.** Por defecto, Docker guarda en memoria caché las acciones realizadas recientemente. Si se diese el caso de que ejecutamos un docker build varias veces, Docker comprobará si el fichero contiene las mismas instrucciones y, en caso afirmativo, no generará una nueva imagen. Para generar una nueva imagen omitiendo la memoria caché utilizaremos siempre esta opción.
* **--pull.**Por defecto. Docker solo descargará la imagen especificada en la expresión FROM. Para forzar que descargue la nueva versión de la imagen utilizaremos esta opción.
* **--quiet.**Por defecto, se muestra todo el proceso de creación, los comandos ejecutados y su salida. Utilizando esta opción solo mostrará el identificador de la imagen creada.

Indicamos en el guión qué imagen base vamos a utilizar mediante **FROM**, después con **RUN** indicamos los comandos a ejecutar y con [**CMD**](https://colaboratorio.net/glosario/cmd/) decimos el comando por defecto si no se especifica al ejecutarlo.**ENTRYPOINT** establece que comando ejecuta el contenedor.

## Ejemplo 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Generamos el fichero

FROM ubuntu

ENTRYPOINT ["/bin/echo"]

### Creamos la imagen

docker build .

docker images

### Ejecutamos la imagen creada

docker run [ID] Hola Docker !!

### Resultado de la ejecución

Hola Docker !!

## Ejemplo-2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Con directiva CMD

FROM ubuntu

CMD ["/bin/echo" , "Hola Docker "]

docker build .

docker images

docker run [ID]

### Resultado de la ejecución

Hola Docker

Si le pasamos un nuevo argumento al comando dockerrun, se ejecutará en lugar de /bin/echo

docker run [ID] /bin/date

Podemos crear una nueva imagen con el nombre **prueba-1** y la etiqueta **con-dockerfile**, a través de la opción -t.

docker build -t prueba-1:con-dockerfile .

docker images

## Ejemplo-3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

mkdir ubuntutest

cd ubuntutest

vi Dockerfile

FROM ubuntu:latest

RUN apt-get -y update; \

apt-get -y upgrade; \

apt-get -y install apt-utils \

vim \

htop; monitoriza admin., visor procesos

RUN apt-get -y install dstat “ recursos sistema

CMD ["bash"]

### Construimos la imagen

docker build -t "prueba-2:dockerfile" .

### Construimos el contenedor

docker images

docker run -ti ID