**Investigar los siguientes comandos de Docker CLI:**

1. **docker build**

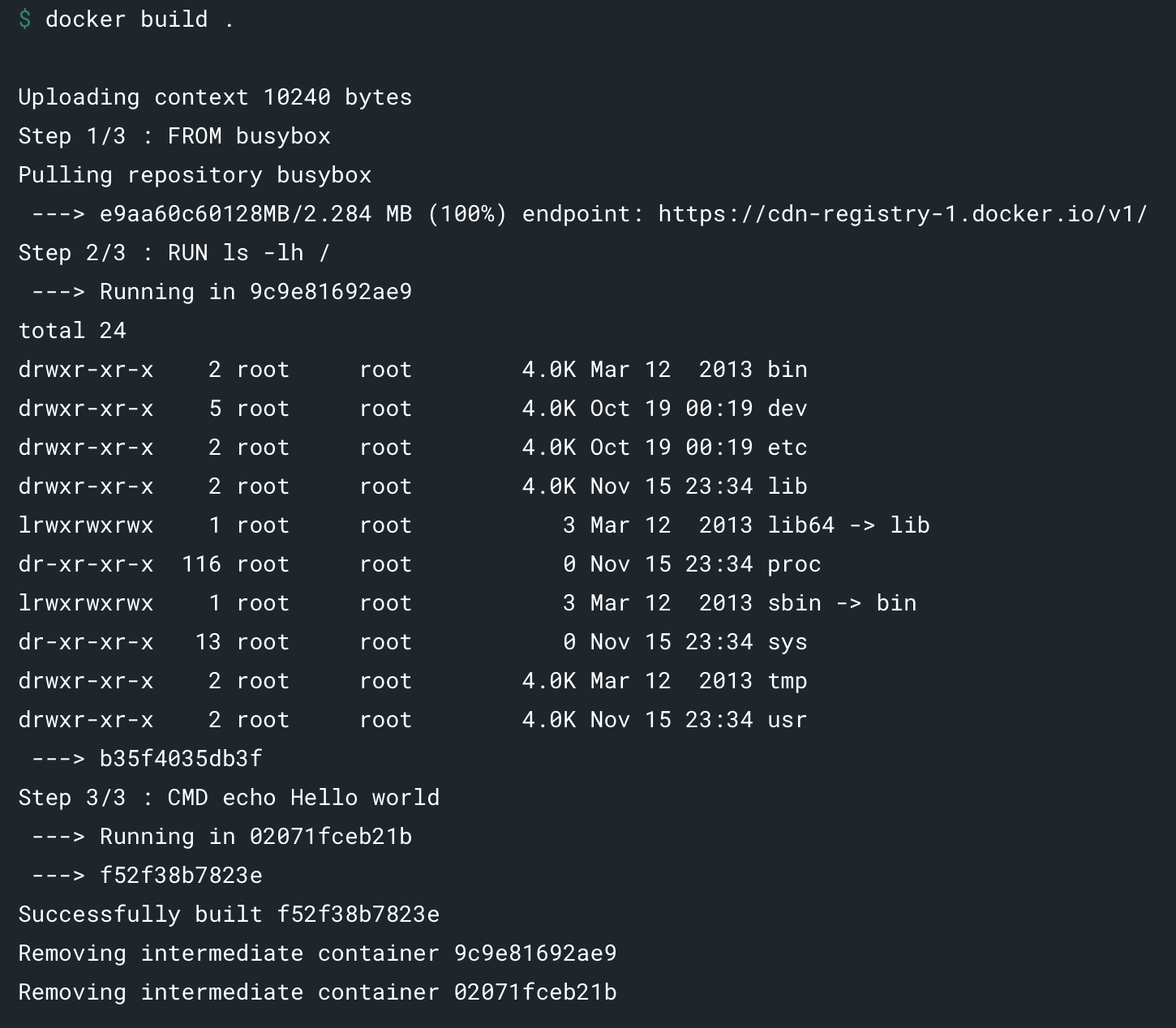
Función:

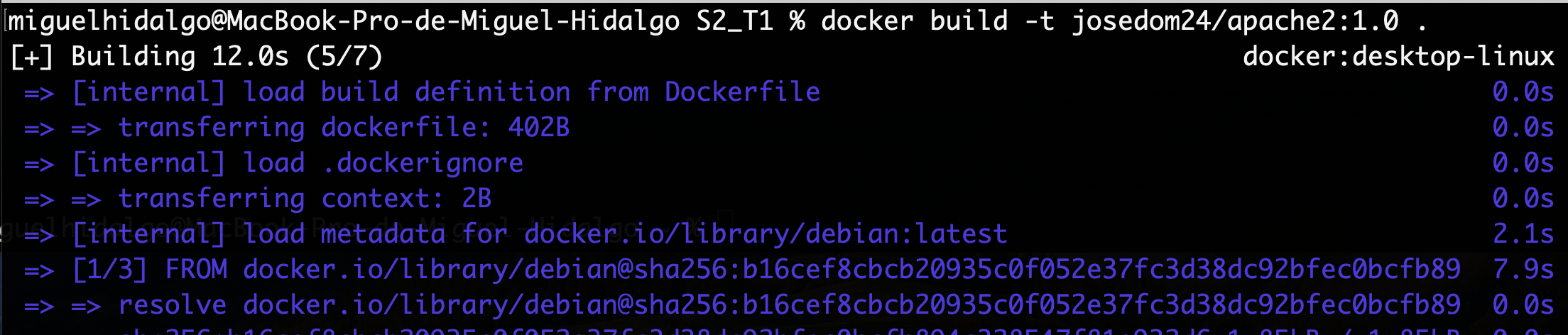
Crear una imagen a partir de un Dockerfile.

Sintaxis:

docker image build [OPTIONS] PATH | URL | -

Ejemplos:





1. **docker tag**

Función:

Crear una etiqueta TARGET\_IMAGE que se refiera a SOURCE\_IMAGE.

Se utiliza este comando para añadir metadatos a una imagen (como la version). Las etiquetas suelen crearse cuando se construye la imagen, pero el comando **docker tag** permite añadir una etiqueta más tarde, creando esencialmente un alias para la imagen de destino.

Sintaxis:

docker image tag SOURCE\_IMAGE[:TAG] TARGET\_IMAGE[:TAG]

Ejemplos:



1. **docker rmi**

Función:

Eliminar una o más imágenes.

Alias:

docker image rm

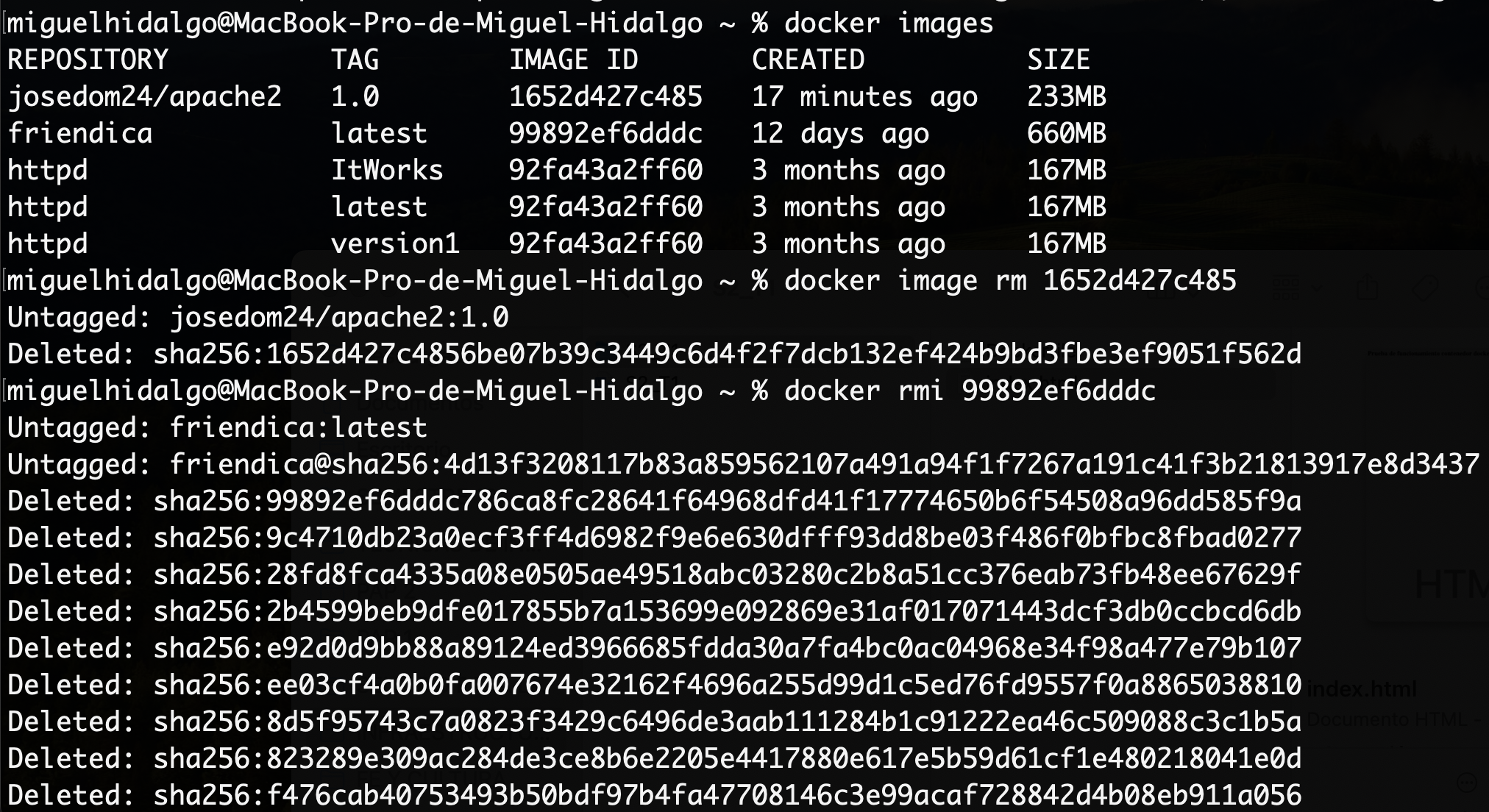
docker image remove

docker rmi

Sintaxis:

docker image rm [OPTIONS] IMAGE [IMAGE...]

Ejemplos:



1. **docker history**

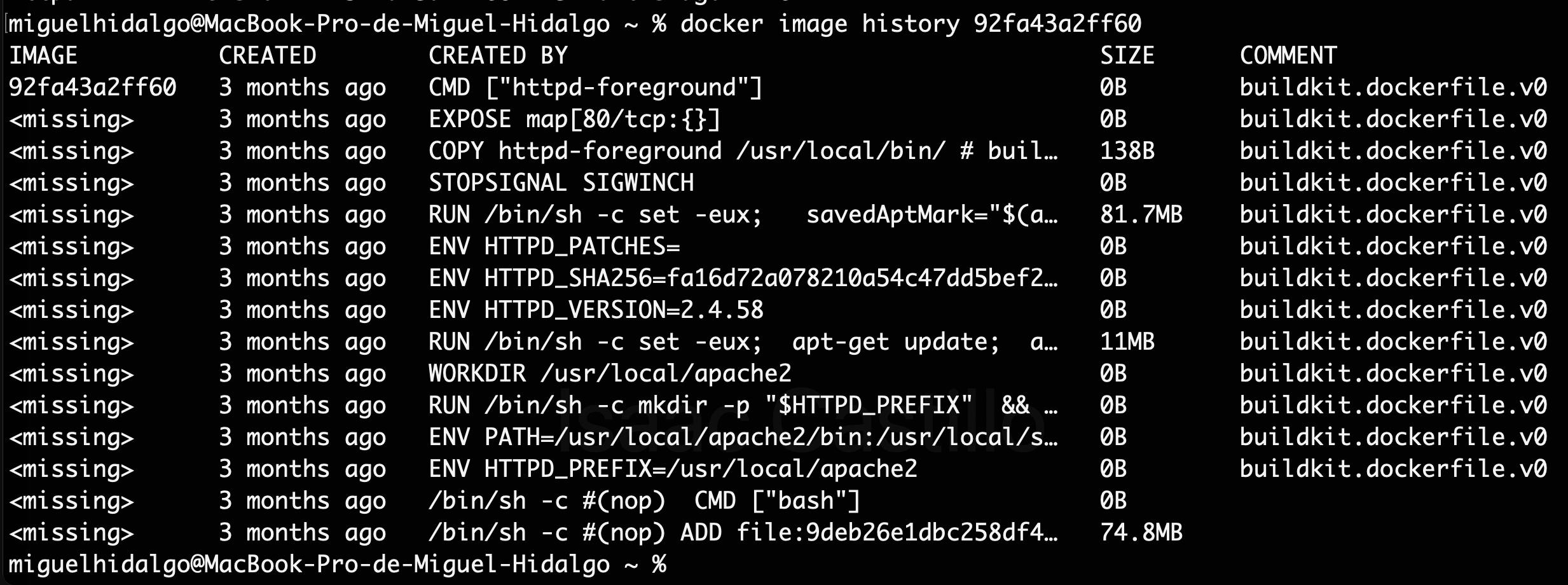
Función:

Mostrar la historia de una imagen.

Sintaxis:

docker image history [OPTIONS] IMAGE

Ejemplos:





1. **docker commit**

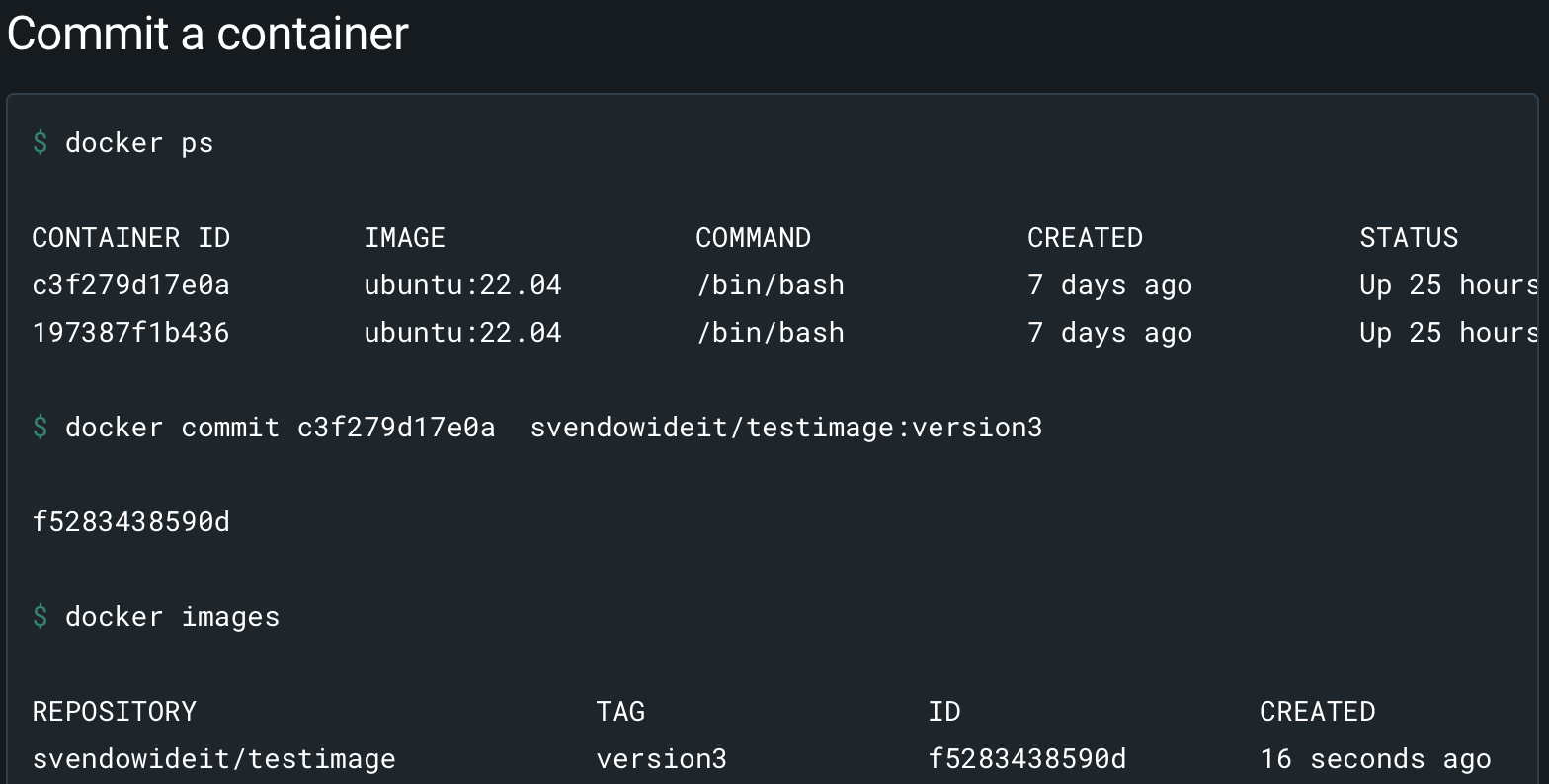
Función:

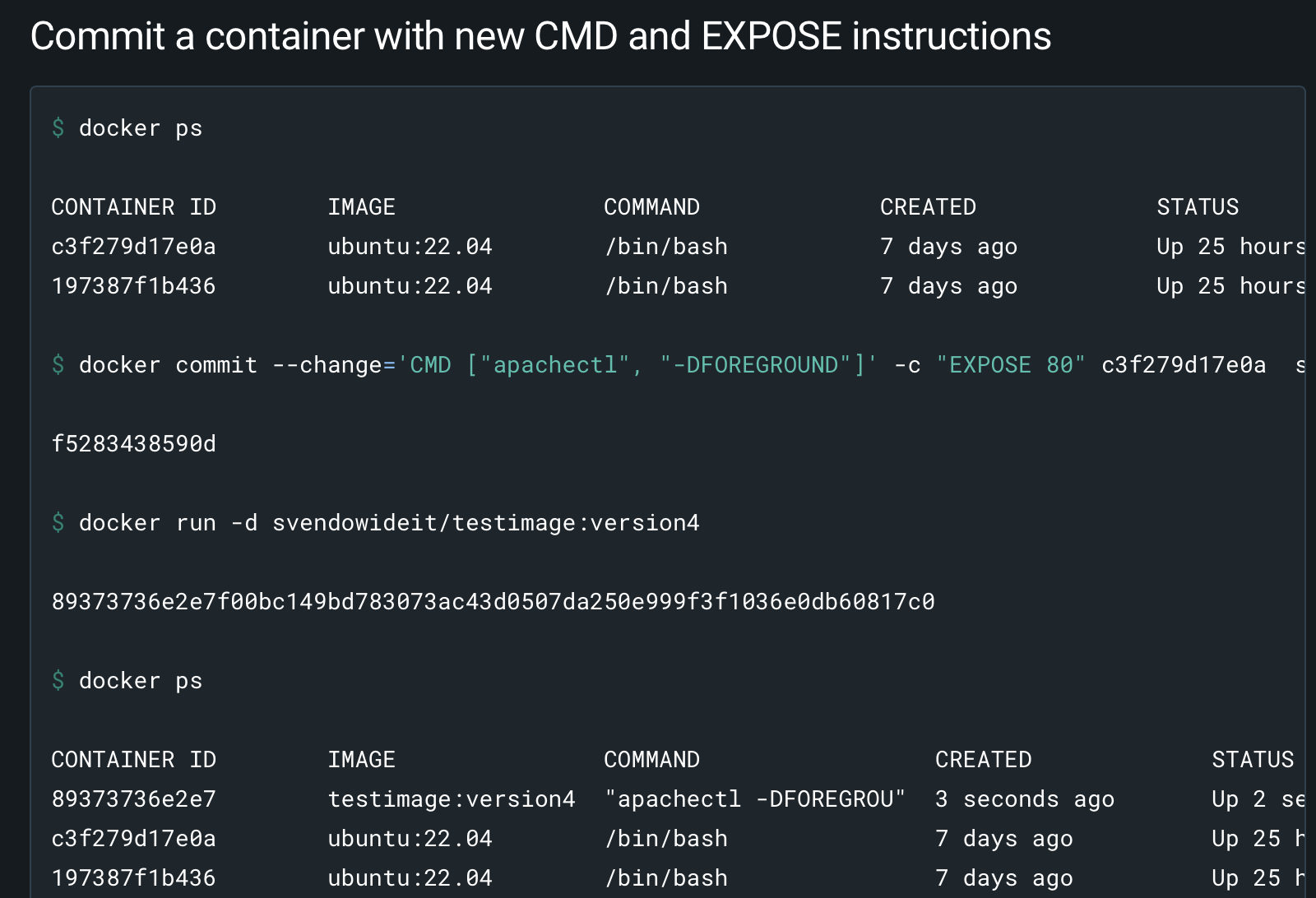
Crear una nueva imagen a partir de los cambios de un contenedor.

Sintaxis:

docker container commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

Ejemplos:





**Investigar que es y para que sirve un Dockerfile**

Dockerfile es un archivo de texto que contiene las instrucciones secuenciales necesarias para crear una nueva imagen utilizando el comando *build*. Estas instrucciones incluyen la identificación de una imagen existente que se usará como base, los comandos que se ejecutarán durante el proceso de creación de la imagen y un comando que se ejecutará cuando se implementen instancias nuevas de la imagen del contenedor.



Sintaxis

# Sample Dockerfile

# Indicates that the windowsservercore image will be used as the base image.

FROM mcr.microsoft.com/windows/servercore:ltsc2019

# Metadata indicating an image maintainer.

LABEL maintainer="jshelton@contoso.com"

# Uses dism.exe to install the IIS role.

RUN dism.exe /online /enable-feature /all /featurename:iis-webserver /NoRestart

# Creates an HTML file and adds content to this file.

RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html

# Sets a command or process that will run each time a container is run from the new image.

CMD [ "cmd" ]

FROM: es una opción de Dockerfile que debe presentarse como la primera instrucción. Cumple con la función de establecer la imagen sobre la que los pasos e imágenes siguientes se desarrollan en el sistema. La imagen mínima que da origen al resto de imágenes en Docker es llamada scratch.

ENV: hace referencia a la opción que indica las variables de entorno que se necesitan en el proceso de construcción de una imagen en Docker y permite la ejecución de los contenedores y sus labores en el sistema.

USER: esta herramienta se utiliza en los archivos de instrucciones de dokcer file con el objetivo de cambiar el usuario y su pertenencia a un grupo determinado. Una vez se ejecute esta opción, se aplicará a la totalidad de instrucciones siguientes.

RUN: es una de las opciones de mayor importancia y popularidad en Dockerfile. Cumple la labor de ejecutar una instrucción incluida en la línea de comandos de la imagen durante su proceso de construcción. El run dockerfile puede escribirse en formato SHELL o bajo la opción de escritura EXEC.

ADD: este elemento se encarga de las tareas relacionados con la copia de ficheros, directorios y archivos de una imagen en Dockerfile. Se debe tener en cuenta que el uso de la instrucción ADD implica la creación de una nueva capa de imagen, por lo que debes ser cuidadoso al implementar esta opción.

EXPOSE: es la opción que tiene como labor la definición de las asignaciones referentes a los puertos para los contenedores de la plataforma que se encuentren en su etapa de ejecución.

Ejemplo

FROM debian

MAINTAINER José Domingo Muñoz "josedom24@gmail.com"

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/\*

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

EXPOSE 80

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

ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

**Fuentes:**

* <https://docs.docker.com/engine/reference/commandline/docker/>
* <https://kinsta.com/es/blog/comandos-docker/#:~:text=11.-,docker%20tag,para%20la%20imagen%20de%20destino>
* <https://refine.dev/blog/docker-run-command/#:~:text=Examples%20of%20using%20the%20Docker%20run%20command%E2%80%8B&text=The%20command%20%27docker%20run%20%2Dd,the%20%27%2Dp%27%20option>.
* <https://learn.microsoft.com/es-es/virtualization/windowscontainers/manage-docker/manage-windows-dockerfile>
* <https://keepcoding.io/blog/que-es-dockerfile/>
* <https://www.josedomingo.org/pledin/2016/02/ejemplos-de-ficheros-dockerfile-creando-imagenes-docker/>