
Imágenes Docker

Gabriela Alejandra Caicedo
Chaves

Introducción

Docker utiliza **imágenes** como plantillas inmutables para lanzar contenedores.

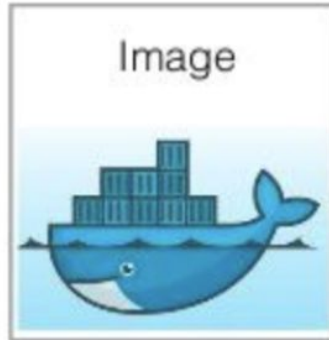
Las imágenes están compuestas por **capas**, optimizando reutilización y almacenamiento.

Son la base de la portabilidad y escalabilidad en arquitecturas de microservicios.

```
FROM ubuntu:16.04
MAINTAINER docker@docker.com
RUN apt-get update && apt-get install -y python3
RUN apt-get install -y python3-pip
RUN pip install Flask==0.12.2
EXPOSE 5000
CMD ["python3", "app.py"]
```

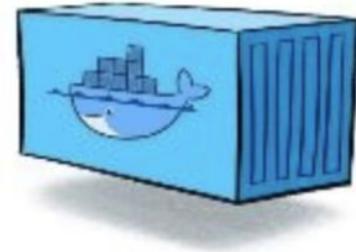
Dockerfile

build



Docker Image

run



Docker Container

Listar imagenes

docker images

```
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    74cc54e27dc4   3 months ago  10.1kB
```



hello-world

latest

74cc54e27dc4

Repositorio Imagenes

Filter by (2) [Clear All](#)

Products

- ☒ Images
- ☐ Extensions
- ☐ Plugins

Trusted content

- ☒  Docker Official Image ⓘ
- ☐  Verified Publisher ⓘ
- ☐  Sponsored OSS ⓘ

Categories


- ☐ API Management
- ☐ Content Management System
- ☐ Data Science

1 - 25 of 178 available images.

Docker Official Image x Images x

Suggested



memcached 

Updated 2 days ago

Free & open source, high-performance, distributed memory object caching system.

DATABASES & STORAGE

↓ 1B+ • ☆ 2.4K

Pulls: 5,971,302
Last week

[Learn more](#)



nginx 

Updated a day ago

Official build of Nginx.


WEB SERVERS

↓ 1B+ • ☆ 10K+

Pulls: 10,841,871
Last week

[Learn more](#)



busybox 

Updated 5 days ago

Busybox base image.

↓ 1B+ • ☆ 3.4K

Pulls: 10,760,552
Last week

<https://hub.docker.com/search?badges=official&type=image>

Buscar en repositorio

```
$ docker search memcached
```

NAME	DESCRIPTION	STARS	OFFICIAL
memcached	Free & open source, high-performance, distri...	2362	[OK]
bitnami/memcached	Bitnami container image for Memcached	35	
bitnamicharts/memcached	Bitnami Helm chart for Memcached	0	
ubuntu/memcached	Memcached, in-memory keyvalue store for smal...	5	
manageiq/memcached	Container with memcached and built on CentOS...	0	
corpusops/memcached	https://github.com/corpusops/docker-images/	0	
openeuler/memcached		0	
jahacdropboxa/memcached	memcached	0	
jelastic/memcached	An image of the Memcached cache server maint...	0	
atsystem/memcached		2	
sameersbn/memcached		7	
frodenas/memcached	A Docker Image for Memcached	6	
modularitycontainers/memcached	Container for memcached - high-performance,...	0	
motilevy/memcached		0	
sylvainlasnier/memcached	Memcached docker images based on Ubuntu 14.1...	28	
eeacms/memcached	DEPRECATED. Please use the official memcache...	4	
xrowgmbh/memcached	memcached	0	
wodby/memcached		1	
amd64/memcached	Free & open source, high-performance, distri...	0	
arm32v7/memcached	Free & open source, high-performance, distri...	1	
tutum/memcached	Base docker image to run a Memcached server	4	
83092cb2/memcached		0	
devbeta/memcached		0	
arm64v8/memcached	Free & open source, high-performance, distri...	4	
ppc64le/memcached	Free & open source, high-performance, distri...	0	

Descargar imagen

Descarga la imagen del repositorio en máquina local.

```
$ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
13b7e930469f: Pull complete
Digest: sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fafе15388214
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
```

<input checked="" type="checkbox"/>	● hello-world	latest	74cc54e27dc4	3 months ago	10.07 KB	▶	⋮	🗑️
<input type="checkbox"/>	○ ubuntu	20.04	b7bab04fd9aa	23 days ago	72.81 MB	▶	⋮	🗑️

Historial de la imagen

Muestra cada capa de la imagen y el comando que la creó.

```
$ docker history ubuntu:20.04
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
b7bab04fd9aa	3 weeks ago	/bin/sh -c #(nop) CMD ["/bin/bash"]	0B	
<missing>	3 weeks ago	/bin/sh -c #(nop) ADD file:f9ee450324e6ff2c9...	72.8MB	
<missing>	3 weeks ago	/bin/sh -c #(nop) LABEL org.opencontainers...	0B	
<missing>	3 weeks ago	/bin/sh -c #(nop) LABEL org.opencontainers...	0B	
<missing>	3 weeks ago	/bin/sh -c #(nop) ARG LAUNCHPAD_BUILD_ARCH	0B	
<missing>	3 weeks ago	/bin/sh -c #(nop) ARG RELEASE	0B	

Copia de seguridad, restaurar y eliminar

Se Crea un archivo `.tar` con la imagen.

```
$ docker save -o ubuntu.tar ubuntu:20.04
```

Recupera una imagen desde el archivo exportado.

```
$ docker load -i ubuntu.tar  
Loaded image: ubuntu:20.04
```





Elimina una imagen del sistema local.

```
$ docker rmi ubuntu:20.04
Untagged: ubuntu:20.04
Untagged: ubuntu@sha256:8feb4d8ca5354def3d8fce243717141ce31e2c428701f6682bd2fafef15388214
Deleted: sha256:b7bab04fd9aa0c771e5720bf0cc7cbf993fd6946645983d9096126e5af45d713
Deleted: sha256:470b66ea5123c93b0d5606e4213bf9e47d3d426b640d32472e4ac213186c4bb6
```

Convertir un contenedor en una imagen

El comando `docker ps -a` sirve para listar todos los contenedores en tu sistema Docker, tanto los que están en ejecución como los detenidos.










```
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
43010bc118e2   hello-world    "/hello"                 34 minutes ago  Exited (0) 32 minutes a
go             lucid_shamir
```

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> lucid_shamir	43010bc118e2	hello-world		N/A	33 minutes ago	  

El contenedor se llama lucid_shamir y su ID es 43010bc118e2.

Se creará una nueva imagen llamada my_hello_image con la etiqueta v1.

```
ACER@DESKTOP-15177DK MINGW64 ~  
$ docker commit 43010bc118e2 my_hello_image:v1  
sha256:8b1c2b226d91b6cbd47779cd25cfbf6df178ccf330c3cd7c87c17bd17
```


	name	tag	image id	created	size	actions
<input type="checkbox"/>	 hello-world	latest	74cc54e27dc4	3 months ago	10.07 KB	  
<input type="checkbox"/>	 my_hello_image	v1	8b1c2b226d91 	2 minutes ago	10.07 KB	  

Etiquetado de imagenes y publicar

```
$ docker tag my_hello_image:v1 gabriela12345/hello-world:v1
```

se inicia sesión

```
$ docker login
Authenticating with existing credentials... [Username: gabriela12345]
```

 Info → To login with a different account, run 'docker logout' followed by 'docker login'

```
Login Succeeded
```

Publicar imagen

```
$ docker push gabriela12345/hello-world:v1
The push refers to repository [docker.io/gabriela12345/hello-world]
63a41026379f: Pushed
v1: digest: sha256:978c68b7ed18c1bce3ec97f8beb184eef87c14c702dd219f2f01869d0eb537f0 size: 525
```

gabriela12345/hello-world

Last pushed 1 minute ago

Add a description  

Add a category  

General

Tags

Image Management BETA

Collaborators



Webhooks

Settings

Tags

 DOCKER SCOUT INACTIVE
[Activate](#)

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 v1		Image	less than 1 day	2 minutes

[See all](#)

<https://hub.docker.com/repository/docker/gabriela12345/hello-world/general>

Repositorio local

crear registro local

```
$ docker run -d -p 5000:5000 --name registry registry:2
Unable to find image 'registry:2' locally
2: Pulling from library/registry
44cf07d57ee4: Pull complete
bbbdd6c6894b: Pull complete
8e82f80af0de: Pull complete
3493bf46cdec: Pull complete
6d464ea18732: Pull complete
Digest: sha256:a3d8aaa63ed8681a604f1dea0aa03f100d5895b6a58ace528858a7b332415373
Status: Downloaded newer image for registry:2
1b904bd6a06417552f13c3bd61e9490daf76c8b4b5a5e1cfef64212c15e07751
```

Etiquetar imagen

```
$ docker tag my_hello_image:v1 localhost:5000/hello-world
```

Subirla

```
$ docker push localhost:5000/hello-world
Using default tag: latest
The push refers to repository [localhost:5000/hello-world]
63a41026379f: Pushed
latest: digest: sha256:978c68b7ed18c1bce3ec97f8beb184eef87c14c702dd219f2f01869d0eb537f0 size: 525
```

Dockerfile

El Dockerfile es un archivo de texto que define paso a paso cómo construir una imagen Docker.

Contiene instrucciones declarativas que permiten automatizar y controlar el contenido y comportamiento de la imagen resultante.

Un Dockerfile se compone de **instrucciones clave** que definen:

- Qué imagen base usar
- Qué software instalar
- Qué archivos copiar
- Qué comandos ejecutar por defecto
- Qué configuraciones y metadata incluir

```
$ mkdir imagen1
```

```
$ cd imagen1/
```

```
$ nano Dockerfile
```

```
FROM alpine
```

```
RUN echo "Hola mundo"
```

```
$ docker build -t mi_imagen:latest .
```

```
UMARIANA@INGCIV69953 MINGW64 ~/imagen1/imagen1
```

```
$ docker history mi_imagen:latest
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
e976cd3e329c	23 minutes ago	RUN /bin/sh -c echo "Hola mundo" # buildkit	4.1kB	buildkit.dockerfile.v0
<missing>	2 months ago	CMD ["/bin/sh"]	0B	buildkit.dockerfile.v0
<missing>	2 months ago	ADD alpine-minirootfs-3.21.3-x86_64.tar.gz /...	8.5MB	buildkit.dockerfile.v0

```
UMARIANA@INGCIV69953 MINGW64 ~/imagen1/imagen1
```

```
$ docker history alpine
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
a8560b36e8b8	2 months ago	CMD ["/bin/sh"]	0B	buildkit.dockerfile.v0
<missing>	2 months ago	ADD alpine-minirootfs-3.21.3-x86_64.tar.gz /...	8.5MB	buildkit.dockerfile.v0

Dockerfile

Crea una carpeta para tu proyecto (opcional, pero recomendado):

```
ACER@DESKTOP-19T7TDK MINGW64 ~  
$ mkdir mi_app  
  
ACER@DESKTOP-19T7TDK MINGW64 ~  
$ cd mi_app  
  
ACER@DESKTOP-19T7TDK MINGW64 ~/mi_app  
$ nano dockerfile
```

```
$ cp "C:/Users/UMARIANA/Documents/imagen.png" ./imagen.png
```

```
$ cat << 'EOF' > start.sh
```

```
> echo
```

```
> cd /output
```

```
> python3 -m http.server 5000
```

```
> EOF
```

```
$ chmod +x start.sh
```

```
$ nano Dockerfile
```

```
FROM debian:bullseye-slim
```

```
LABEL maintainer="Gabriela Caicedo <gabrielaac@unicauca.edu.co>"
```

```
RUN apt-get update && apt-get install -y python3
```

```
COPY start.sh .
```

```
COPY imagen.png /output/imagen.png
```

```
EXPOSE 5000
```





```
CMD ["python3", "-m", "http.server", "5000", "--directory", "/output"]
```

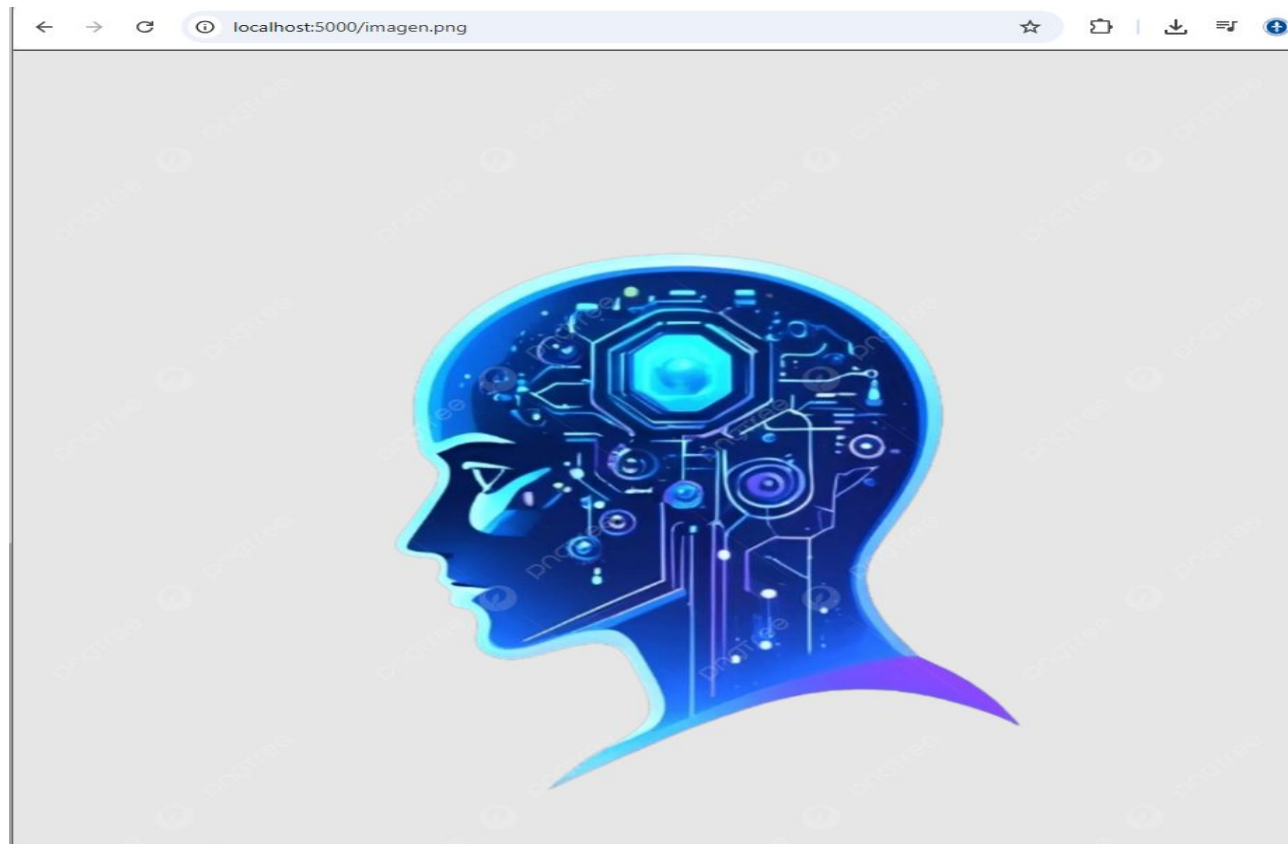
```
$ ls
```

```
dockerfile imagen.png start.sh
```

```
$ docker run -d -p 5000:5000 --name hello mi_app:v1
```

```
58db0a6a4b7bc5cca649146e1f2615765c379083b1e5c41879f7823b0f5e5816
```

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	● hello	834325a752a3	mi_app:v1	5000:5000 	0.01%	1 minute ago	  



<http://localhost:5000/imagen.png>