



DigitalHouse>

Docker Hub



**Certified Tech
Developer**

The Ultimate Degree

“

Docker Hub es un **servicio de registro de repositorios** proporcionado por Docker Inc. Compartir y colaborar son sus premisas.



”

Índice

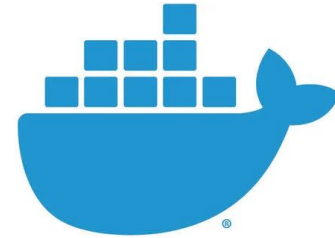
1. [¿Para qué sirve Docker Hub?](#)
2. [Características de Docker Hub](#)
3. [Creación del primer repositorio](#)
4. [Explorar las imágenes](#)
5. [Descargar una imagen](#)
6. [Crear una imagen](#)
7. [Empujar una imagen](#)
8. [¿Qué son las imágenes certificadas por Docker?](#)
9. [Imágenes populares en Docker Hub](#)

1 | ¿Para qué sirve Docker Hub?

¿Para qué sirve Docker Hub?

Nos permite extraer y enviar imágenes de la ventana acoplable hacia y desde Docker Hub. Podemos tratar esto como un GitHub, donde obtenemos y enviamos nuestro código fuente, pero en el caso de Docker Hub descargamos o publicamos nuestras imágenes de contenedor.

Es un repositorio en línea basado en la nube que almacena ambos tipos de repositorios, es decir, el repositorio público y el privado. Los repositorios públicos son accesibles para todos, pero el privado es accesible para el propietario interesado de los repositorios. También hay un costo asociado si almacenamos más de un cierto número de repositorios como privado.



[Why Docker?](#) [Products](#) [Use Cases](#) [Developers](#) [Company](#)

[Sign In](#) [Get Started](#)

Docker Hub

The world's leading service for finding and sharing container images with your team and the Docker community.

[Go to Docker Hub](#)

[Secure, Private Repo Pricing](#)

Share and Collaborate with Docker Hub

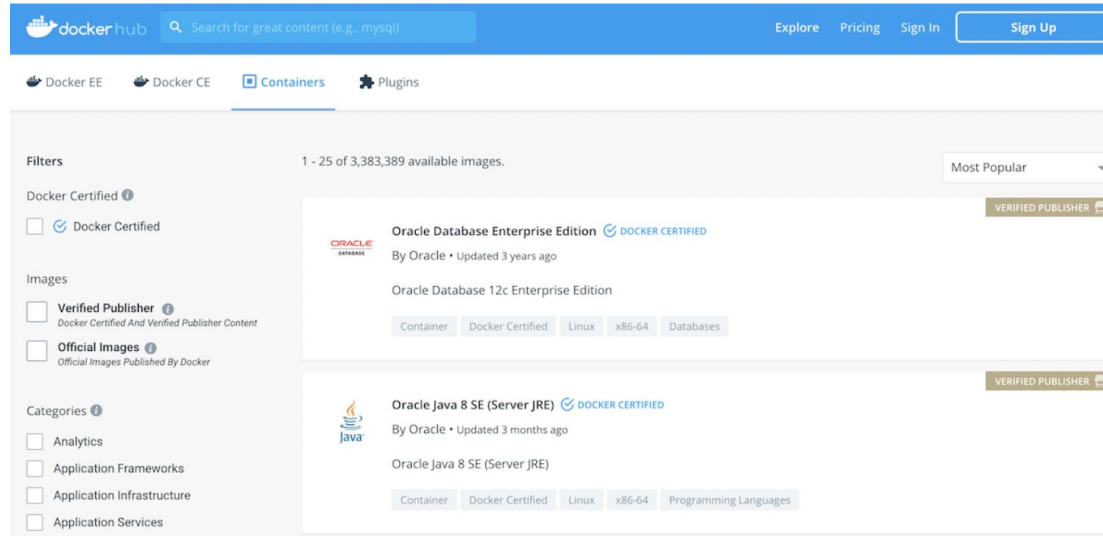
Docker Hub is the world's largest repository of [container images](#) with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers. Users get access to free public repositories for storing and sharing images or can choose subscription plan for private repos.

2 | **Características de Docker Hub**

Características de Docker Hub

1. Repositorios de imágenes:

Nos ayuda a encontrar y extraer imágenes de contenedores de Docker Hub.

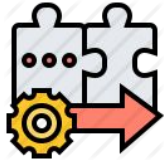


Características de Docker Hub



2. Equipo y organizaciones:

Nos permite crear grupos de trabajo e impulsar los repositorios privados, que están disponibles para su uso únicamente dentro de nuestra organización. De esta forma, hemos gestionado el acceso a nuestros repositorios privados de imágenes de contenedores.



3. Integración de GitHub y Bitbucket:

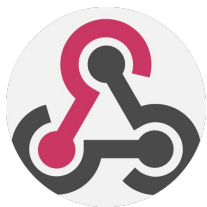
Permite la integración con repositorios de código fuente como GitHub y Bitbucket.



4. Construcciones automatizadas:

Si se ha enviado algún cambio en el código fuente a los repositorios, automáticamente detecta y crea imágenes de contenedor desde GitHub o Bitbucket y las envía a Docker Hub.

Características de Docker Hub



5. Webhooks:

Una vez que hemos enviado nuestras imágenes con éxito, con la ayuda de un webhook, desencadena una acción para integrar Docker Hub con otros servicios.



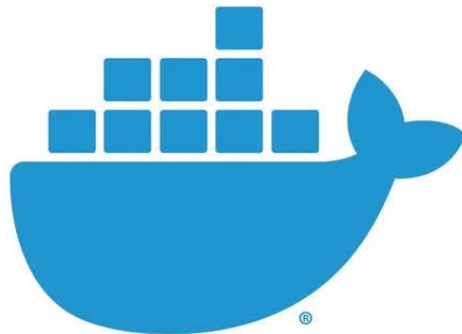
6. Imágenes oficiales y del editor:

Las imágenes de alta calidad proporcionadas por los dockers se consideran imágenes oficiales y se pueden extraer y utilizar. Del mismo modo, las imágenes de alta calidad proporcionadas por proveedores externos son imágenes del editor –también llamadas imágenes certificadas– que brindan soporte y garantía de compatibilidad con Docker Enterprise.

3 | Creación del primer repositorio

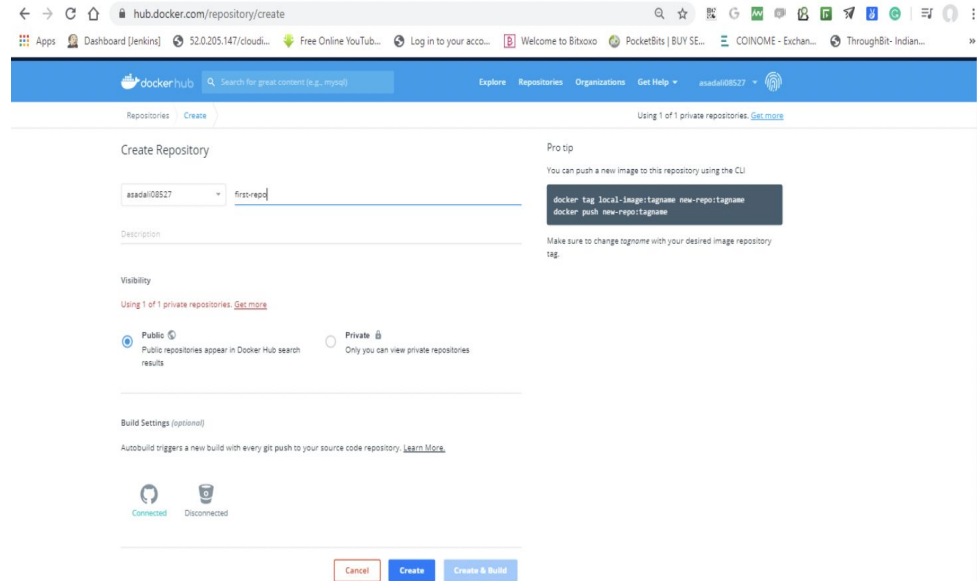
Creación del primer repositorio

Este paso requiere iniciar sesión en Docker Hub utilizando tus credenciales de inicio de sesión. Si no tenés una cuenta, podés crearla haciendo clic en el botón **"Registrarse"**, disponible en la página web.



Una vez que hayas iniciado sesión, podés crear el repositorio haciendo clic en **“Crear repositorio”** en la página de bienvenida.

Luego, te pedirá un nombre y le dará un nombre a tu repositorio.



The screenshot shows the Docker Hub 'Create Repository' page. The browser address bar displays 'hub.docker.com/repository/create'. The page header includes the Docker Hub logo, a search bar, and navigation links for 'Explore', 'Repositories', 'Organizations', and 'Get Help'. The user's profile 'asad@08527' is visible in the top right. The main content area is titled 'Create Repository' and features a dropdown menu with 'asad@08527' and a text input field containing 'first-repo'. Below this is a 'Description' field. The 'Visibility' section shows 'Using 1 of 1 private repositories' and two radio buttons: 'Public' (selected) and 'Private'. The 'Build Settings (optional)' section includes a link to 'Learn More' about autobuild. At the bottom, there are 'Connected' and 'Disconnected' status indicators for Docker Desktop. On the right side, a 'Pro tip' box provides CLI commands: 'docker tag local-image:tagname new-repo:tagname' and 'docker push new-repo:tagname'. At the bottom right, there are three buttons: 'Cancel', 'Create', and 'Create & Build'.

Podés seleccionar una **visibilidad pública o privada**. También podés integrar tus repositorios de código fuente como **GitHub** y **Bitbucket** a través de la configuración de compilación, pero es opcional (y también se puede hacer en una etapa posterior). Una vez que todo esté hecho, se debe hacer clic en **"Crear"**. Con estos pasos podés crear tus primeros repositorios. En la próxima diapositiva podés ver cómo se verán.



Docker Hub nos ofrece **solo un repositorio privado de forma gratuita.**

Si necesitamos más repositorios privados, podemos actualizar nuestra cuenta a un plan pago.

Con la herramienta terminal de **Docker Desktop**, descargada e instalada, podremos iniciar sesión en Docker Hub mediante un comando.

A screenshot of a terminal window with a dark background. The text 'docker login' is displayed in a light pink or red monospace font. The terminal has a vertical bar on the left side, suggesting a command prompt or cursor position.

```
docker login
```

4 | Explorar las imágenes

Explorar las imágenes

Hay dos formas de buscar imágenes y repositorios públicos desde Docker Hub. Podés buscarlo en el **sitio web de Docker Hub** o usar la **herramienta de línea de comandos** y ejecutar el siguiente comando, suponiendo que queremos buscar en la imagen del repositorio de MySQL.

```
docker search mysql
```

```
root@ip-xxx-xx-xx-xx:~# docker search mysql
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
mysql	MySQL is a widely used, open-source relation...	9417	[OK]	
mariadb	MariaDB is a community-developed fork of MyS...	3392	[OK]	
mysql/mysql-server	Optimized MySQL Server Docker images. Create...	688		[OK]
centos/mysql-57-centos7	MySQL 5.7 SQL database server	74		
mysql/mysql-cluster	Experimental MySQL Cluster Docker images. Cr...	66		
centurylink/mysql	Image containing mysql. Optimized to be link...	61		[OK]
deitch/mysql-backup	REPLACED! Please use http://hub.docker.com/r...	41		[OK]
bitnami/mysql	Bitnami MySQL Docker Image	39		[OK]
tutum/mysql	Base docker image to run a MySQL database se...	34		
schickling/mysql-backup-s3	Backup MySQL to S3 (supports periodic backup...	29		[OK]
prom/mysqld-exporter		27		[OK]
linuxserver/mysql	A Mysql container, brought to you by LinuxSe...	25		
centos/mysql-56-centos7	MySQL 5.6 SQL database server	19		
circleci/mysql	MySQL is a widely used, open-source relation...	19		
databack/mysql-backup	Back up mysql databases to... anywhere!	15		
mysql/mysql-router	MySQL Router provides transparent routing be...	15		
arey/mysql-client	Run a MySQL client from a docker container	13		[OK]
openshift/mysql-55-centos7	DEPRECATED: A Centos7 based MySQL v5.5 image...	6		
fradelg/mysql-cron-backup	MySQL/MariaDB database backup using cron tas...	6		[OK]
genschsa/mysql-employees	MySQL Employee Sample Database	5		[OK]
devilbox/mysql	Retagged MySQL, MariaDB and PerconaDB offici...	3		
ansibleplaybookbundle/mysql-apb	An APB which deploys RHSCl MySQL	2		[OK]
jelastic/mysql	An image of the MySQL database server mainta...	1		
widdpim/mysql-client	Dockerized MySQL Client (5.7) including Curl...	0		[OK]
monasca/mysql-init	A minimal decoupled init container for mysql	0		

5 | Descargar una imagen

Descargar una imagen

Podemos descargar una imagen de Docker Hub usando el comando **pull** de la siguiente manera:

```
# docker pull mysql
```

Si ya tenemos **mysql image** en nuestra máquina, el comando anterior actualizará automáticamente la imagen a la última versión. Una cosa a tener en cuenta aquí es que si notamos la salida del comando de búsqueda de la ventana acoplable, hay muchas imágenes de MySQL en Docker Hub, y eso se debe a que cualquiera puede enviar una imagen. Pero depende de nosotros saber cuál usar en función de nuestro caso de uso y asegurarnos de que sea el apropiado.

Digamos que queremos extraer una imagen **bitnami/mysql**:

```
# docker pull bitnami/mysql
```

6 | Crear una imagen

Crear una imagen

El proceso de crear una imagen requiere un Dockerfile. Podemos pensar un Dockerfile como un **manual de instrucciones** que le dice a Docker qué ensamblar. En resumen, es un archivo de configuración que sigue ensamblando instrucciones.

¿Cómo funciona?

Docker lee las instrucciones de un Dockerfile y crea imágenes automáticamente. La imagen de Docker es un sistema de archivos en capas y consta de varias capas de solo lectura. Cada capa de una imagen de Docker representa las instrucciones de un Dockerfile. A continuación, sigamos los pasos para crear una imagen usando un Dockerfile que especifica la configuración de nuestra aplicación.

```
# sudo vim Dockerfile
```

Nota: el nombre del archivo debe ser Dockerfile con “D” mayúscula.

```
FROM ubuntu:16.04
ENV DEBIAN_FRONTEND noninteractive
MAINTAINER someuser@somedomain.com
RUN apt-get update
RUN apt-get install mysql-server -y
CMD echo "My first image created."
```

Echemos un vistazo a algunas de las palabras clave importantes que se utilizan en Dockerfile. Podemos utilizar **#** para agregar un comentario en un Dockerfile.

FROM (de): define la imagen base que se utilizará.

MANTEINER (mantenedores): persona que va a mantener esa imagen.

RUN (correr): se utiliza para ejecutar la instrucción dada para la imagen. En nuestro caso, primero actualiza el sistema y luego instala MySQL.

CMD: se utiliza para ejecutar un comando una vez que se ha lanzado el contenedor.

COPY (copiar): se utiliza para copiar un archivo de nuestro sistema operativo host al contenedor de la ventana acoplable.

EXPOSE (exponer): se utiliza para especificar el número de puerto en el que el contenedor ejecutará su proceso.

Una vez que nuestro Dockerfile se ha creado correctamente, debemos ejecutar **docker build** para “armar” nuestra imagen localmente, para luego enviarla a Docker Hub. Este comando debemos ejecutarlo dentro de la carpeta donde se encuentra el Dockerfile.

```
docker build ./ -t asadali08527/first-repo
```

Podemos verificar que la imagen está creada con la siguiente línea de código:

```
docker image ls
```

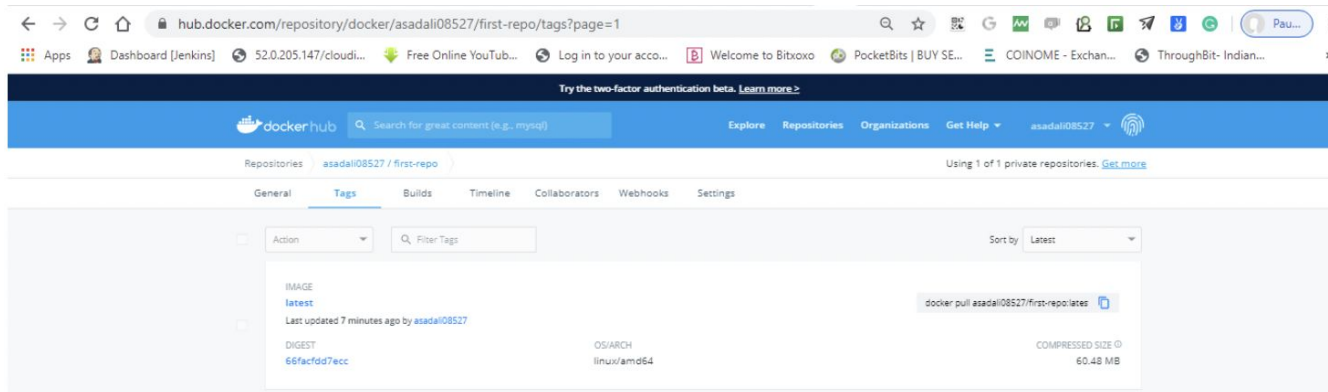
7 | Empujar una imagen

Empujar una imagen

Una vez que nuestra imagen se ha creado correctamente y se está ejecutando, podemos enviarla a Docker Hub mediante el comando push.

```
docker push asadali08527/first-repo
```

Podemos verificar las etiquetas de la imagen y el estado en Docker Hub, que se verá así.

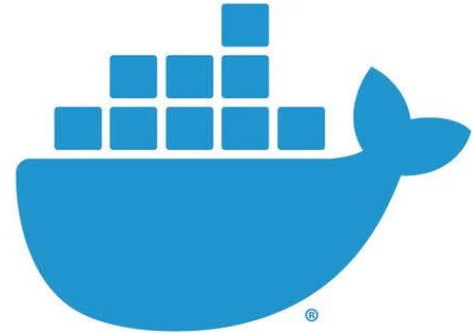


8 | ¿Qué son las imágenes certificadas por Docker?

¿Qué son las imágenes certificadas por Docker?

Estas son las **imágenes oficiales** impulsadas por proveedores o contribuyentes. Una imagen solo puede ser certificada por Docker Hub si su contenido cumple con las **reglas, estándares y leyes proporcionadas por Docker Hub**. En resumen, esa imagen debe pasar ciertas pruebas de referencia.

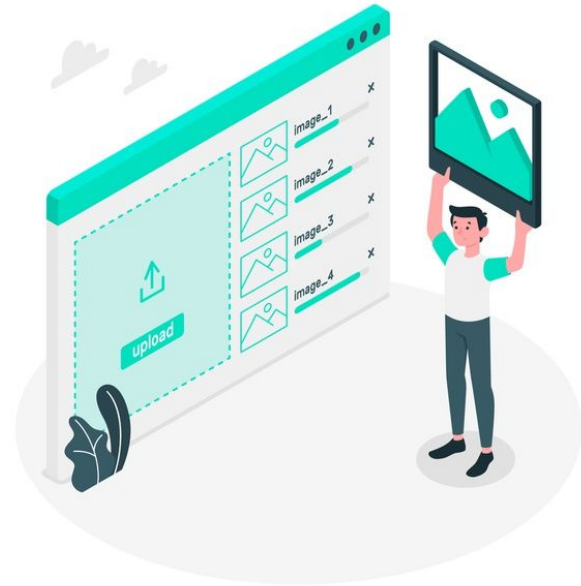
Docker Hub proporciona **inspectDockerImage**, herramienta a través de la cual un proveedor puede autocertificar las imágenes y los complementos (por lo general, el proveedor o contribuyente publica sus complementos para registrar volúmenes y redes).



9 | Imágenes populares en Docker Hub

Imágenes populares en Docker Hub

Hay muchas imágenes seleccionadas y optimizadas disponibles en Docker Hub. La popularidad de estas imágenes depende de varios factores, **presencia en el mercado, calificaciones, puntajes de satisfacción, entre otros**. Para obtener una lista detallada de los repositorios más populares, podés visitar el sitio web de Docker Hub. El uso de una imagen también depende del sistema operativo y su arquitectura.



Si conocemos para qué sistema operativo y arquitectura se utilizarán las imágenes, debemos considerar los siguientes factores clave antes de extraer una imagen:

- Buscar una versión específica utilizando etiquetas (principalmente la última).
- Optar por el que tenga máxima cantidad descargas y estrellas.
- Buscar la fecha de su última actualización.
- Si es posible, verificar su tipo, ya sea del editor verificado u oficial (Docker Certified).



DigitalHouse>