

## Contenedores con Docker. Instalación y primeros pasos

1. **Instalación y acceso al Hub.** Se pide la instalación del Docker engine según los procedimientos indicados en la documentación en línea <https://docs.docker.com/engine/install/debian/>. De entre las opciones de instalación posibles, vamos a elegir la instalación desde repositorios de Docker disponibles para Debian

Una vez completada la instalación debe crearse una cuenta en el DockerHub <https://hub.docker.com/> y posteriormente desde la consola del sistema debe emplearse el comando:

```
# docker login
```

para acceder al Hub desde la consola  
A continuación, debe ejecutarse el comando:

```
# docker run hello-world
```

Para verificar que es la imagen "hello-world" es accesible y puede ejecutarse en la máquina

Verificar también con `# docker images` que ya disponemos de una imagen descargada en nuestro repositorio local

2. **Gestión de un contenedor.** A modo de ejemplo ilustrativo puedes seguir los siguientes pasos:

- a. Ejecuta el comando:

```
# docker run -it -p 80:80 --name ubuntu_apache ubuntu
```

El parámetro `-p 80:80` lo redirige el puerto 80 de la máquina real al puerto 80 del contenedor.

El parámetro `--name ubuntu_apache` pone ese nombre al contenedor que se va a crear

- b. Ahora que estas dentro del contenedor actualiza la lista de repositorios de Ubuntu y descarga el servicio Apache2. Una vez descargado, inícialo.
- c. Abandona el contenedor con `<Control+P>`, `<Control+Q>` y regresarás a la consola del sistema. Esta salida asegura que el contenedor sigue en ejecución. Verifícalo con `docker ps`
- d. Accede al servidor web del contenedor usando el navegador del sistema Windows con `http://localhost`

- e. Volvemos a acceder al contenedor con `docker attach ubuntu_apache` y de nuevo salimos de él pero esta vez con el comando `exit`. Esto provoca que el contenedor se pare. Compruébalo con `docker ps`
  - f. Verifica de nuevo desde el navegador que ahora la pagina web anterior ya no es accesible como en el apartado d
  - g. Ahora lo ponemos de nuevo en marcha con `docker start ubuntu_apache`
  - h. Accedemos de nuevo al navegador y la página web del contenedor y vemos que sigue sin estar accesible → cuando se para un contenedor y se arranca de nuevo, los procesos que estaban funcionando se mantienen detenidos. Conéctate al contenedor, arranca el servicio `apache2` y vuelve a probar el acceso a la web del contenedor que en este caso debe ser accesible (importante salir del contenedor sin usar `exit`)
3. **Guardar el contenedor en una imagen nueva.** Ahora se piden realizar los siguientes pasos para salvar el estado del contenedor en una imagen nueva:
- a. Ejecuta el comando:  
  

```
docker commit -m "ubuntu con apache2" -a "nombre_autor" ID_contenedor repositorio/nueva_imagen
```

  
donde:
    - `ID_contenedor` es el ID del contenedor
    - `repositorio` es normalmente tu nombre de usuario en el Hub de Docker
    - `nueva_imagen` es el nombre que le queremos dar a la nueva imagen, por ejemplo, `ubuntu_apache2`
  - b. Verifica con `docker images` que la nueva imagen ya está guardada en nuestro repositorio local de imágenes. Comprueba que el tamaño de la nueva imagen es mayor que el tamaño de la imagen original "ubuntu"
4. **Crear nuevos contenedores.** Se pide ahora crear un nuevo contenedor a partir de la imagen anterior. Para ello:
- a. Inicia un nuevo contenedor de nombre "ubuntu\_apache\_2" a partir de la imagen creada anteriormente, pero esta vez con el parámetro `-p 81:80` para que no se solapen los puertos con el contenedor anterior
  - b. Una vez dentro del contenedor inicia el servicio `Apache2` y sal del contenedor sin detenerlo

- c. Verifica desde un navegador que la web del nuevo contenedor es accesible con <http://localhost:81>

5. **Publicar la imagen en el DockerHub.** Se debe:

- a. Realizar previamente un `docker login` en el caso de que no estuviésemos con el login activo
- b. Subimos la imagen al DockerHub con:  
`docker push usuario_docker/nombre_imagen`
- c. Accedemos al DockerHub desde el navegador y comprobamos que la imagen nueva ya figura en nuestro repositorio

Ahora ya se podría hacer un `docker pull imagen` para descargar la imagen a una maquina nueva e iniciar nuevos contenedores a partir de ella.

6. **Descargar la imagen desde el DockerHub.** Se pide:

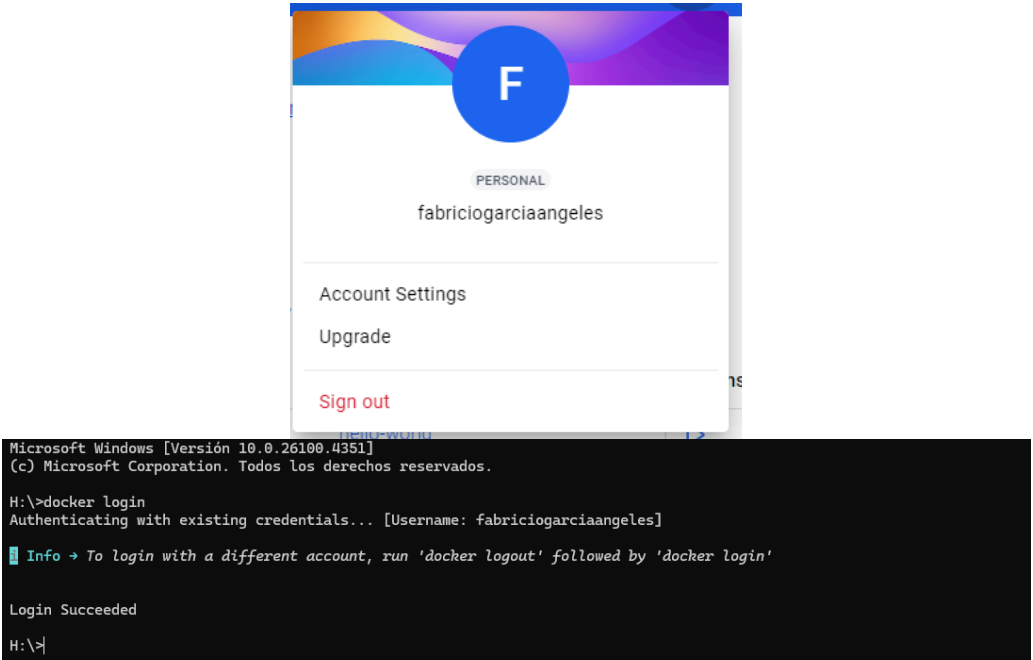
- a. En otra maquina con Docker instalado se debe hacer:

`docker pull usuario_docker/nombre_imagen`

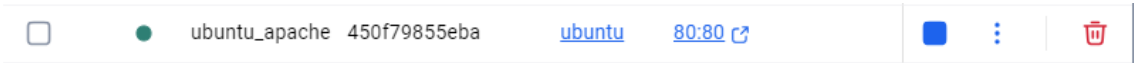
- b. crea un contenedor basado en la imagen que acabas de descargar

# Resolver

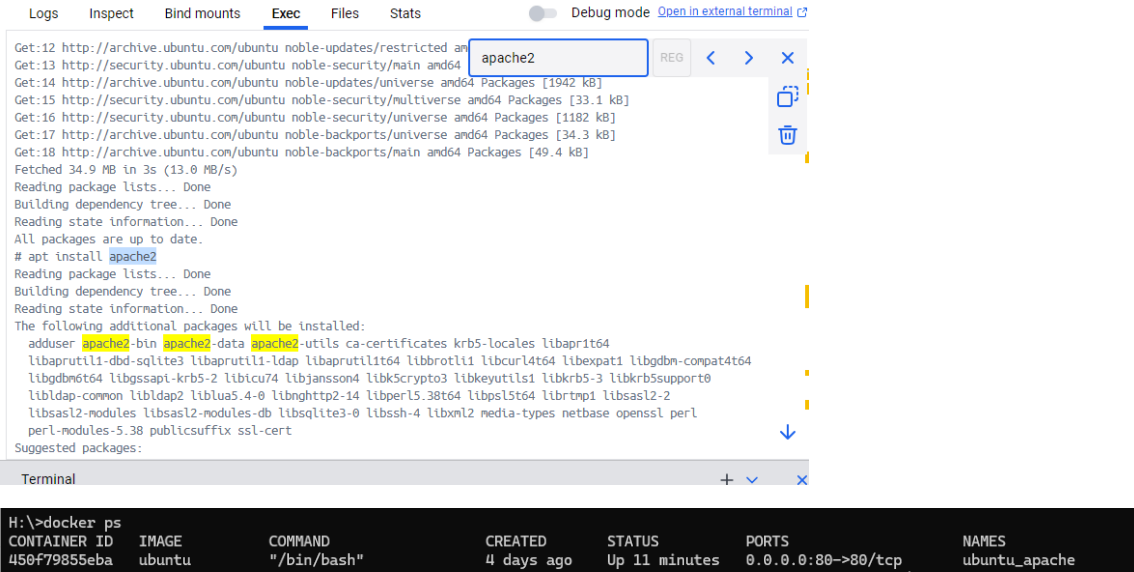
## Primer paso Crear la cuenta




## Creo la maquina de Ubuntu



## Entro y instalo el apache2 antes actualizando los paquetes



localhost
☆



English - Français  
Provided by NLTechno

## Configuración Servidor

**Versión de Apache:** 2.4.51 (Apache/2.4.51 (Win64) PHP/7.4.26)

**Versión de PHP:** 7.4.26

**Extensiones**

**Cargadas:**

Core	bcmath	calendar	ctype
date	filter	hash	iconv
json	SPL	pcre	readline
Reflection	session	standard	mysqlnd
tokenizer	zip	zlib	libxml
dom	PDO	bz2	SimpleXML
xml	xmlreader	xmlwriter	apache2handler
openssl	curl	gd	intl
imap	ldap	mbstring	exif
mysqli	Phar	pdo_mysql	soap
sockets			

**Versión de MySQL:** 10.6.5

**Nombre/IP Servidor** 218-14

Entro al contenedor con attach y esto hace que pare

```

H:\>docker attach ubuntu_apache
root@450f79855eba:/# docker ps
bash: docker: command not found
root@450f79855eba:/# exit
exit

H:\>docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
63bc585c442a   mariadb:11.2   "docker-entrypoint.s..." 2 weeks ago    Up 39 minutes  0.0.0.0:3310->3306/tcp             mariadb
H:\>

```

Logs
Inspect
Bind mounts
Exec
Files
Stats
Debug mode
[Open in external terminal](#)

```

# service apache2 restart
* Restarting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set
the 'ServerName' directive globally to suppress this message

#

```

[ OK ]

Creando nueva

```

H:\>docker commit -m "ubuntu con apache2" -a "fabricio" 450f79855eba704e755cdcb138b2bc9ef8afe269c03e9034dafbed2e84ab408 fabriciogarciaang
les/ubuntu_apache2

```

## Docker images

```
H:\>docker images
REPOSITORY              TAG               IMAGE ID          CREATED           SIZE
fabriciogarciaangeles/ubuntu_apache2  latest           66d046f59775     #9 seconds ago   365MB
mysql                    latest           569c4128dfa6     6 weeks ago      1.27GB
mysql                    8.0             f37951fc3753     6 weeks ago      1.07GB
ubuntu                   latest           c35e29c94501     6 weeks ago      117MB
hello-world              latest           f7931603f70e     3 months ago     20.3kB
mariadb                  11.2            ff87d49107a1     13 months ago    551MB
H:\>
```

## Crear una nueva imagen

```
H:\>docker run -it -p 81:80 --name ubuntu_apache_2 ubuntu
root@b1ff26914778:/# apt update
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
```

## Finalmente termino haciendo el Docker push

```
H:\>docker push fabriciogarciaangeles/ubuntu_apache2
Using default tag: latest
The push refers to repository [docker.io/fabriciogarciaangeles/ubuntu_apache2]
2a5f15ac0bc1: Pushed
20043066d3d5: Mounted from library/ubuntu
latest: digest: sha256:66d046f5977505c37bf8b9f06590cd76e33659fe1701b521d8358ee82281e97c size: 751
H:\>
```

## En el Docker Hub compruebo que la imagen esta subida

