Tarea 2. Instalación de servidor Web (Apache) y Base de Datos en Docker

Contenedores de app web

- ***Listar las imágenes disponibles:***

  docker images

```
root@ibrahimserve:/home/ibrahim# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
ubuntu        latest    6d79abd4c962   2 weeks ago   78.1MB
hello-world   latest    1b44b5a3e06a   7 weeks ago   10.1kB
root@ibrahimserve:/home/ibrahim#
```

***Ejecutar un servidor web Apache en segundo plano y mapear puertos**:*

- ***Crear un Dockerfile en vuestro proyecto***
  FROM httpd:2.4
  COPY ./public-html/ /usr/local/apache2/htdocs/

Pon el nombre que quieras a tu contenedor.

Lanzas los comandos para construir y lanzar una imagen Docker

- $ docker build -t my-apache2 .

```
ibrahim@ibrahimserve:~/apache-docker$ sudo docker build -t my-apache2 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future rel
ease.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.584kB
Step 1/3 : FROM httpd:2.4
 ---> 2416cb32cb59
Step 2/3 : COPY ./public-html/ /usr/local/apache2/htdocs/
 ---> a47cf6e2b141
Step 3/3 : EXPOSE 80
 ---> Running in 148b1d272f69
 ---> Removed intermediate container 148b1d272f69
 ---> 45117fcfdd6e
Successfully built 45117fcfdd6e
Successfully tagged my-apache2:latest
ibrahim@ibrahimserve:~/apache-docker$
```

- $ docker run -dit --name my-running-app -p 8080:80 my-apache2

```
ibrahim@ibrahimserve:~/apache-docker$ sudo docker run -dit --name my-running-app
 -p 8080:80 my-apache2
52b2760954b9f1351d87ebfdd0db91c4e89ec1ea32ab5766de200a35968d5a84
ibrahim@ibrahimserve:~/apache-docker$
```

Verificar que el contenedor está en ejecución:

- docker ps

```
ibrahim@ibrahimserve:~/apache-docker$ sudo docker ps
CONTAINER ID   IMAGE        COMMAND              CREATED          STATUS
  PORTS                                   NAMES
52b2760954b9   my-apache2   "httpd-foreground"   40 seconds ago   Up 39 seconds
  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   my-running-app
ibrahim@ibrahimserve:~/apache-docker$
```

Visita http://localhost:8080 y ve si funciona.

Ver los logs del contenedor:

- docker logs mi-running-app

```
ame, using 172.17.0.2. Set the 'ServerName' directive globally to suppres
 message
H00558: httpd: Could not reliably determine the server's fully qualified d
ame, using 172.17.0.2. Set the 'ServerName' directive globally to suppres
 message
Thu Oct 02 17:13:19.364508 2025] [mpm_event:notice] [pid 1:tid 1] AH00489:
e/2.4.65 (Unix) configured -- resuming normal operations
Thu Oct 02 17:13:19.382063 2025] [core:notice] [pid 1:tid 1] AH00094: Comm
e: 'httpd -D FOREGROUND'
72.17.0.1 - - [02/Oct/2025:17:14:37 +0000] "GET / HTTP/1.1" 200 36
72.17.0.1 - - [02/Oct/2025:17:14:37 +0000] "GET /favicon.ico HTTP/1.1" 404
Thu Oct 02 17:16:42.304112 2025] [mpm_event:notice] [pid 1:tid 1] AH00492:
 SIGWINCH, shutting down gracefully
H00558: httpd: Could not reliably determine the server's fully qualified d
ame, using 172.17.0.2. Set the 'ServerName' directive globally to suppres
 message
H00558: httpd: Could not reliably determine the server's fully qualified d
ame, using 172.17.0.2. Set the 'ServerName' directive globally to suppres
 message
Fri Oct 03 10:43:56.136519 2025] [mpm_event:notice] [pid 1:tid 1] AH00489:
e/2.4.65 (Unix) configured -- resuming normal operations
Fri Oct 03 10:43:56.138923 2025] [core:notice] [pid 1:tid 1] AH00094: Comm
e: 'httpd -D FOREGROUND'
```

Ejecutar un comando en el contenedor en ejecución:

- docker exec -it mi-running-app **bash**

```
root@ibrahimserve:/home/ibrahim# docker exec -it my-running-app bash
root@52b2760954b9:/usr/local/apache2#
```

- **ls -la** /usr/local/apache2/htdocs/

```
root@52b2760954b9:/usr/local/apache2# ls -la /usr/local/apache2/htdocs/
total 16
drwxr-xr-x 1 root     root     4096 Oct  2 17:11 .
drwxr-xr-x 1 www-data www-data 4096 Sep 29 23:55 ..
-rw-rw-r-- 1 root     root       36 Oct  2 17:06 index.html
root@52b2760954b9:/usr/local/apache2#
```

exit

Meter contenido desde fuera al contenedor:

- docker cp index.html mi-running-app:/usr/local/apache2/htdocs/

```
root@ibrahimserve:/home/ibrahim# docker cp index.html my-running-app:/usr/local
apache2/htdocs/
Successfully copied 2.05kB to my-running-app:/usr/local/apache2/htdocs/
root@ibrahimserve:/home/ibrahim#
```

Abrir http://localhost:8080

Detener, iniciar y reiniciar el contenedor:

- docker stop mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker stop my-running-app
my-running-app
root@ibrahimserve:/home/ibrahim#
```

- docker start mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker start my-running-app
my-running-app
root@ibrahimserve:/home/ibrahim#
```

- docker restart mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker restart my-running-app
my-running-app
root@ibrahimserve:/home/ibrahim#
```

Inspeccionar la configuración del contenedor:

- docker inspect mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker inspect my-running-app
[
    {
        "Id": "52b2760954b9f1351d87ebfdd0db91c4e89ec1ea32ab5766de200
,
        "Created": "2025-10-02T17:13:16.715503762Z",
```

Eliminar el contenedor (primero hay que detenerlo):

- docker stop mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker stop my-running-app
my-running-app
root@ibrahimserve:/home/ibrahim#
```

- docker rm mi-running-app

```
root@ibrahimserve:/home/ibrahim# docker rm my-running-app
my-running-app
root@ibrahimserve:/home/ibrahim#
```

Forzar la eliminación de un contenedor en ejecución:

Crear un nuevo contenedor

- docker run --name otro-apache -d -p 8081:80 my-apache2

```
root@ibrahimserve:/home/ibrahim# docker run --name otro-apache -d -p 8081:80 my-
apache2
c458b8b819fbe35f72c4b54451bb0af176cb1b6ff284c8bbd500c46de9af3a0f
root@ibrahimserve:/home/ibrahim#
```

Eliminar forzosamente

- docker rm -f otro-apache

```
root@ibrahimserve:/home/ibrahim# docker rm -f otro-apache
otro-apache
root@ibrahimserve:/home/ibrahim#
```

Eliminar una imagen:

Solo si no hay contenedores que la utilicen

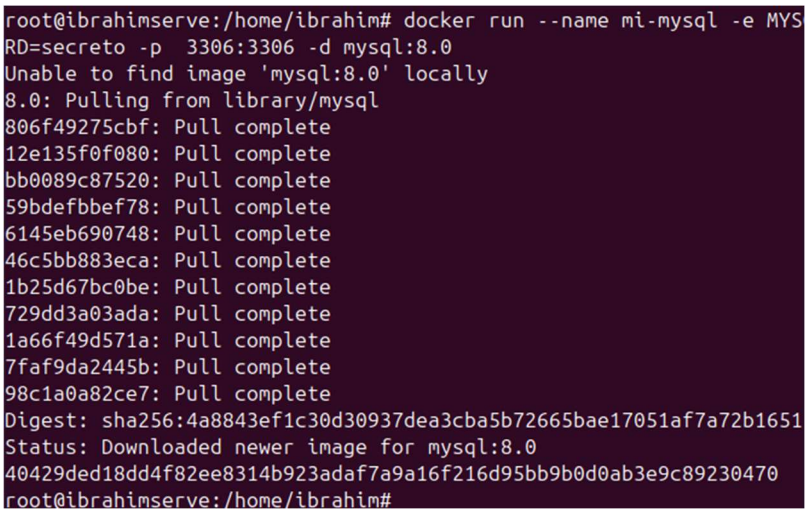
- docker rmi hello-world

```
root@ibrahimserve:/home/ibrahim# docker rmi hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:54e66cc1dd1fcb1c3c58bd8017914d
6bd9cc1642d31
Deleted: sha256:1b44b5a3e06a9aae883e7bf25e45c100be0bb81a0e0
Deleted: sha256:53d204b3dc5ddbc129df4ce71996b8168711e211274
root@ibrahimserve:/home/ibrahim#
```

Contenedores de BD

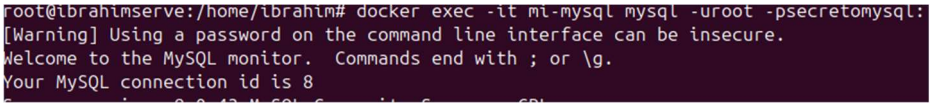
Crear un contenedor de base de datos MySQL:

- docker run --name mi-mysql -e MYSQL_ROOT_PASSWORD=secreto -p 3306:3306 -d mysql:8.0



Conectarse a la base de datos y crear una tabla:

- docker exec -it mi-mysql mysql -uroot -psecreto



Dentro del cliente MySQL:

CREATE DATABASE test;

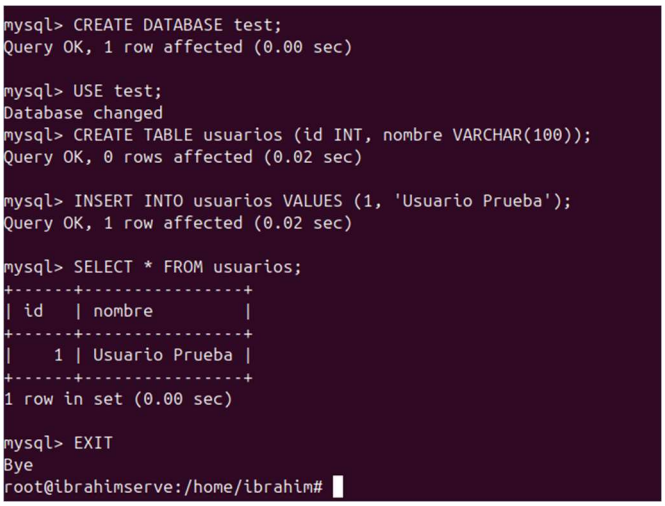
USE test;

CREATE TABLE usuarios (id INT, nombre VARCHAR(100));

INSERT INTO usuarios VALUES (1, 'Usuario Prueba');

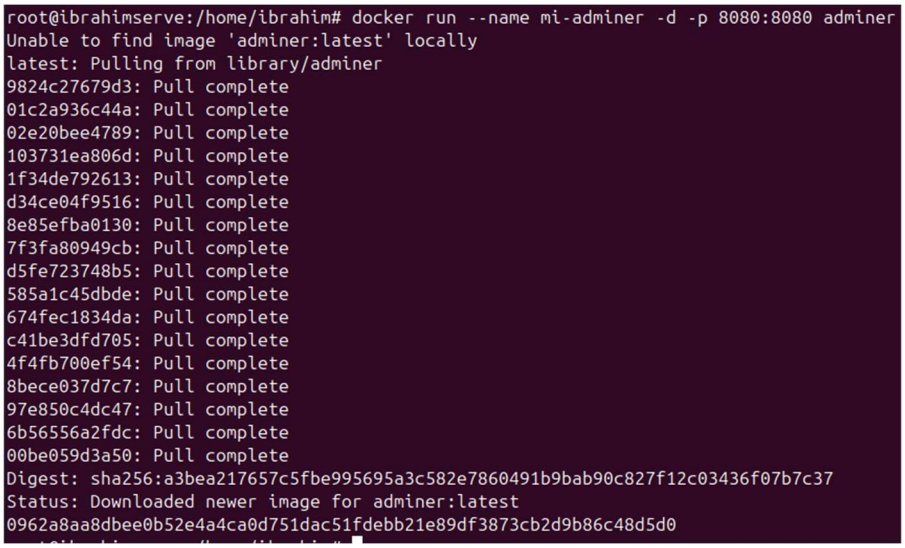
SELECT * FROM usuarios;

EXIT;



Crear un contenedor de un cliente de MySQL:

- docker run --name mi-adminer -d -p 8080:8080 adminer



o con php my admin

- docker run --name phpmyadmin -d -e PMA_HOST=172.17.0.7 -p 8080:80 phpmyadmin