Autor: Ismael Fernando Brito Monar

Actividad 3 - DOCKER

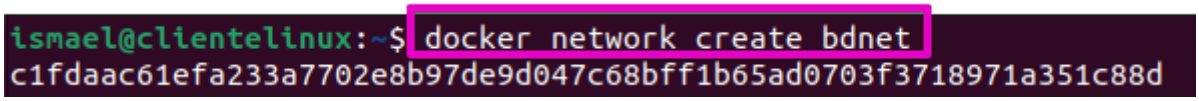
Índice

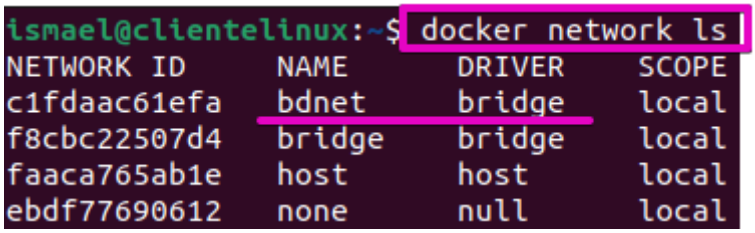
Ejercicio 2 - Redes y Almacenamiento

Creación de contenedores

1. Creamos una red bridge `bdnet` y comprobamos.

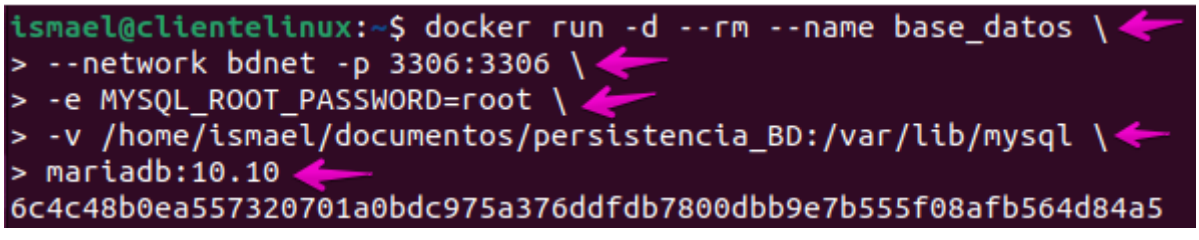
```
docker network create bdnet
docker network ls
```





2. Creamos un contenedor con la imagen de `mariaDB` conectado a la red `bdnet`, que se ejecute en segundo plano y sea accesible por el puerto 3306. (Definimos contraseña del usuario root y volumen de datos persistentes).

```
docker run -d --rm --name base_datos \
--network bdnet -p 3306:3306 \
-e MYSQL_ROOT_PASSWORD=root \
-v /home/ismael/documentos/persistencia_BD:/var/lib/mysql \
mariadb:10.10
```



3. Creamos un contenedor con el programa `Adminer` que se conecte al contenedor de la BD.

```
docker run --name adminer_01 --network bdnet -p 8080:8080 \
-e ADMINER_DEFAULT_SERVER=base_datos adminer:4
```

```
ismael@clientelinux:~$ docker run --name adminer_01 --network bdnet -p 8080:8080 \
> -e ADMINER_DEFAULT_SERVER=base_datos adminer:4
Unable to find image 'adminer:4' locally
4: Pulling from library/adminer
3e440a704568: Pull complete
8d6386bc062c: Pull complete
5dbc633dab93: Pull complete
39317196bba2: Pull complete
31ea8e1da1f9: Pull complete
d3ce0ac05636: Pull complete
ac3ee3b23021: Pull complete
Digest: sha256:4203fd6bcd82fe25dceaf8acb4826129cf7a6e93b22a5ab2fc0ec5a7cdaca3f4
Status: Downloaded newer image for adminer:4
[Sat Apr  1 16:55:28 2023] PHP 7.4.33 Development Server (http://[::]:8080) started
```

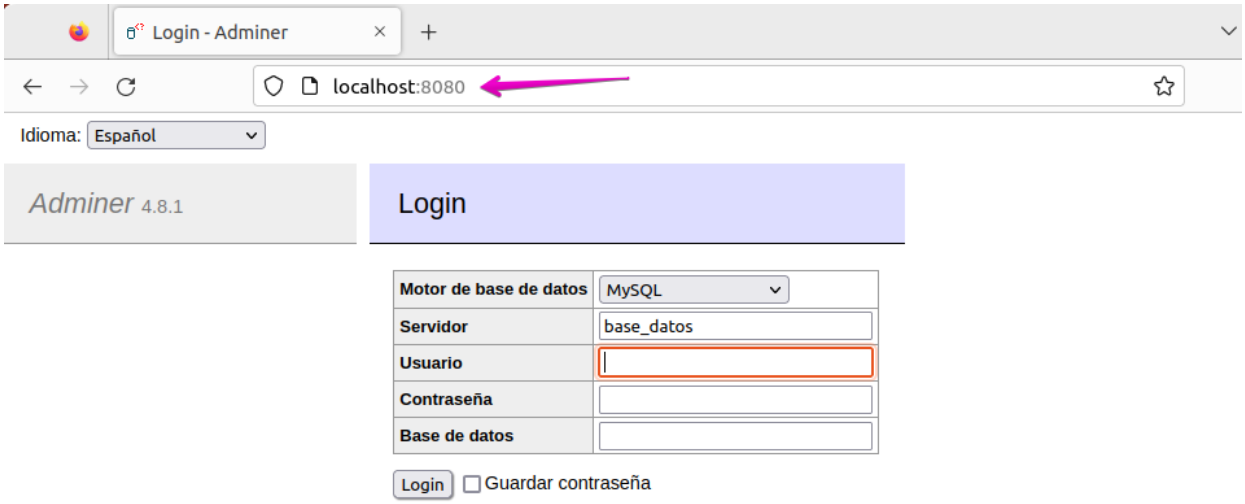
4. Comprobamos que los contenedores estén en ejecución.

```
docker ps -a
```

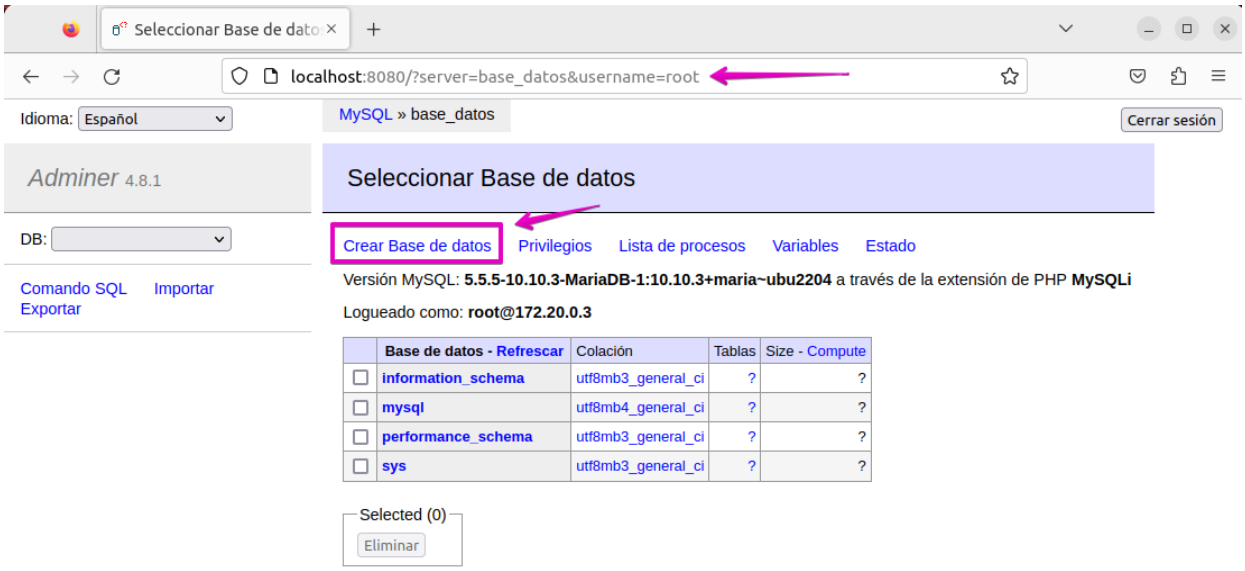
```
ismael@clientelinux:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
                          NAMES
d21f37baae82   adminer:4      "entrypoint.sh php -…"   17 minutes ago   Up 20 seconds   0.0.0.0:8080->
8080/tcp, :::8080->8080/tcp   adminer_01
6c4c48b0ea55   mariadb:10.10  "docker-entrypoint.s…"   21 minutes ago   Up 21 minutes   0.0.0.0:3306->
3306/tcp, :::3306->3306/tcp   base_datos
```

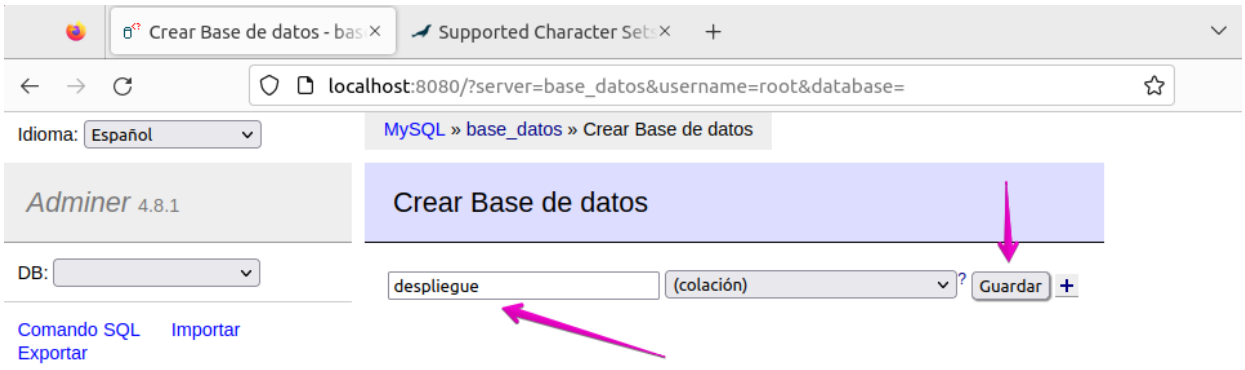
Acceso a la interfaz Web Adminer

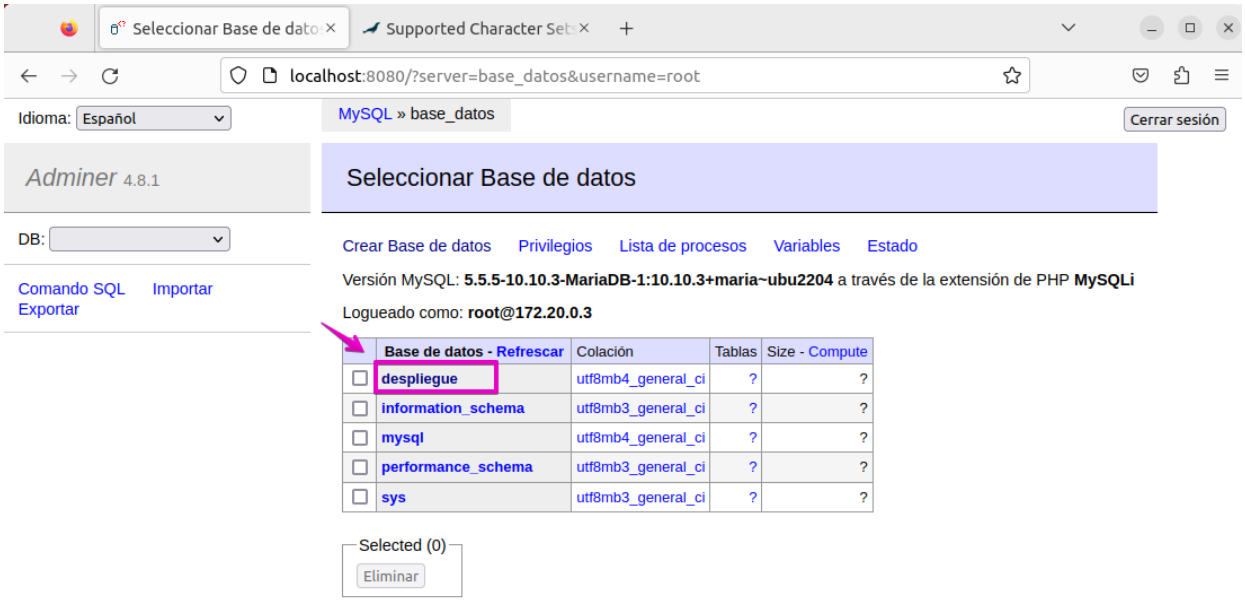
5. Accedemos a `Adminer` desde el navegador a través del puerto 8080 para conectarnos a la base de datos del contenedor `mariaDB` e iniciamos sesión con el usuario `root`.



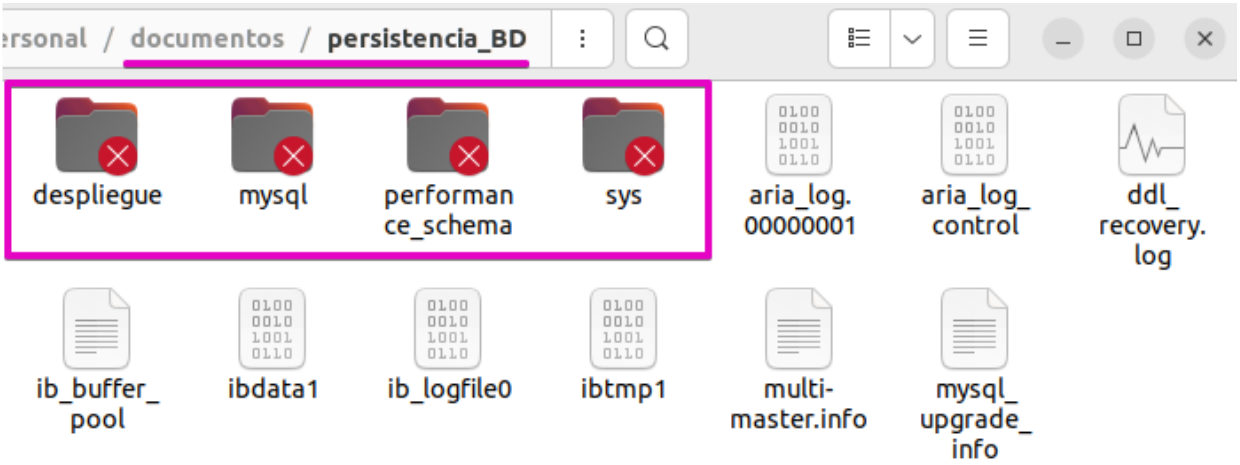
5. Creamos la BD `Despliegue` por medio de la interfaz web Adminer.







6. Nos dirigimos a la ruta empleada para datos persistentes en el que se almacenan los datos generados por el contenedor del servidor de base de datos.



Borrado de contenedores, red y volúmenes

7. Procedemos a borrar los contenedores, la red y los volúmenes utilizados

Empezamos borrando los contenedores y luego comprobamos.

```
docker rm -f base_datos adminer_01
docker ps -a
```

```
ismael@clientelinux:~$ docker rm -f base_datos adminer_01
base_datos
adminer_01
ismael@clientelinux:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ismael@clientelinux:~$
```

Borramos la red y comprobamos.

```
docker network rm bdnet
docker network ls
```

```
ismael@clientelinux:~$ docker network rm bdnet
bdnet
ismael@clientelinux:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
f8cbc22507d4   bridge    bridge    local
faaca765ab1e   host      host      local
ebdf77690612   none      null      local
```

Por ultimo comprobamos los volúmenes pendientes, es decir, los que no están enlazados con ningún contenedor y los borramos con `prune`.

```
docker volume ls -f dangling=true
docker volume prune
docker volume ls
```

```
ismael@clientelinux:~$ docker volume ls -f dangling=true
DRIVER    VOLUME NAME
local     1bade4d4ee7b1ec8ba5142f2b80af9cfb8db9f23805d7ea360b6d4253bba3cf0
local     e133d1195e9d638beb8a2f1d95348649eb476cac2914e7909659042cfe1df008
local     miweb
ismael@clientelinux:~$ docker volume prune
WARNING! This will remove all local volumes not used by at least one container.
Are you sure you want to continue? [y/N] y
Deleted Volumes:
1bade4d4ee7b1ec8ba5142f2b80af9cfb8db9f23805d7ea360b6d4253bba3cf0
e133d1195e9d638beb8a2f1d95348649eb476cac2914e7909659042cfe1df008
miweb

Total reclaimed space: 309.5MB
ismael@clientelinux:~$ docker volume ls
DRIVER    VOLUME NAME
ismael@clientelinux:~$
```