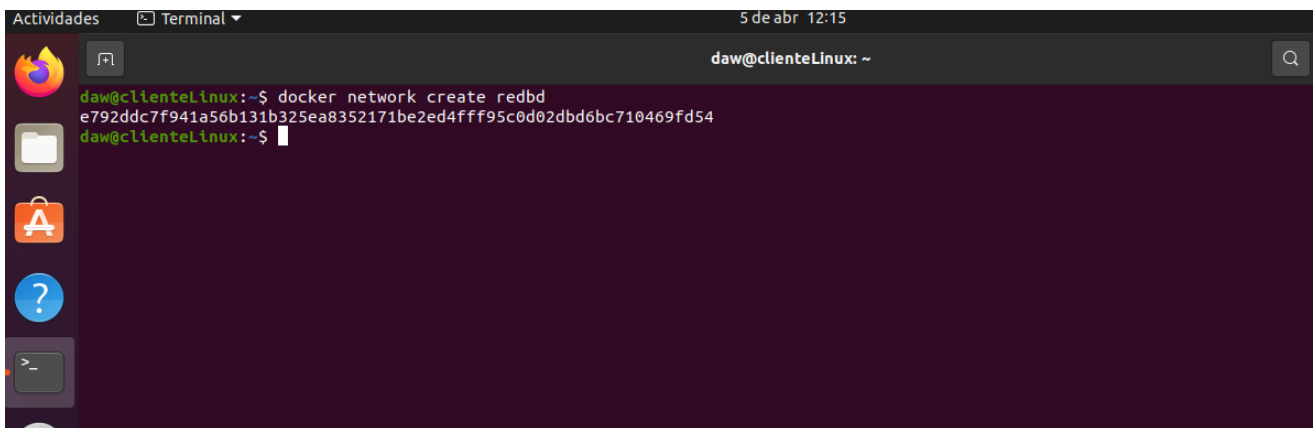


# Ejercicio 3 - redes Despliegue de contenedores en red: Adminer y MariaDB

## 1. Crea una red bridge redbd

```
docker network create redbd
```



---

## 2. Crea un contenedor con una imagen de mariaDB que estará en la red redbd .

Este contenedor se ejecutará en segundo plano, y será accesible a través del puerto 3306. (Es necesario definir la contraseña del usuario root y un volumen de datos persistente)

```
docker run --name contenedor1 -v /home/daw/datadir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root -e MARIADB_DATABASE=prueba -d --network redbd -p 3306:3306 mariadb
```

```
daw@clienteLinux: ~  
daw@clienteLinux:~$ docker run --name contenedor1 -v /home/daw/datadir:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root -e MARIADB_DATABASE=prueba -  
d --network redbd -p 3306:3306 mariadb  
cde8aa41e49c776862925d298d2accdbd6a90c6952312421be07aa657359a687  
daw@clienteLinux:~$
```

### 3. Crea un contenedor con Adminer que se pueda conectar al contenedor de la BD

```
docker pull adminer
```

```
daw@clienteLinux:~$ docker pull adminer  
Using default tag: latest  
latest: Pulling from library/adminer  
40e059520d19: Already exists  
17005d1e8de6: Pull complete  
a7b4c7024b81: Pull complete  
c42f86642e80: Pull complete  
43dc8104beab: Pull complete  
bf5953a063bc: Pull complete  
ac6308d67965: Pull complete  
c9ca1d648f5f: Pull complete  
6ff4f902cbae: Pull complete  
83e78a45af34: Pull complete  
15279d93718d: Pull complete  
24599d250f0a: Pull complete  
8cbfa55dcdee: Pull complete  
ce956e8c750e: Pull complete  
15cec3c253a9: Pull complete  
Digest: sha256:31bd1517fd89ee609150b28bd4ea8553eeabe026e2acc0d4076748744fb0f588  
Status: Downloaded newer image for adminer:latest  
docker.io/library/adminer:latest  
daw@clienteLinux:~$
```

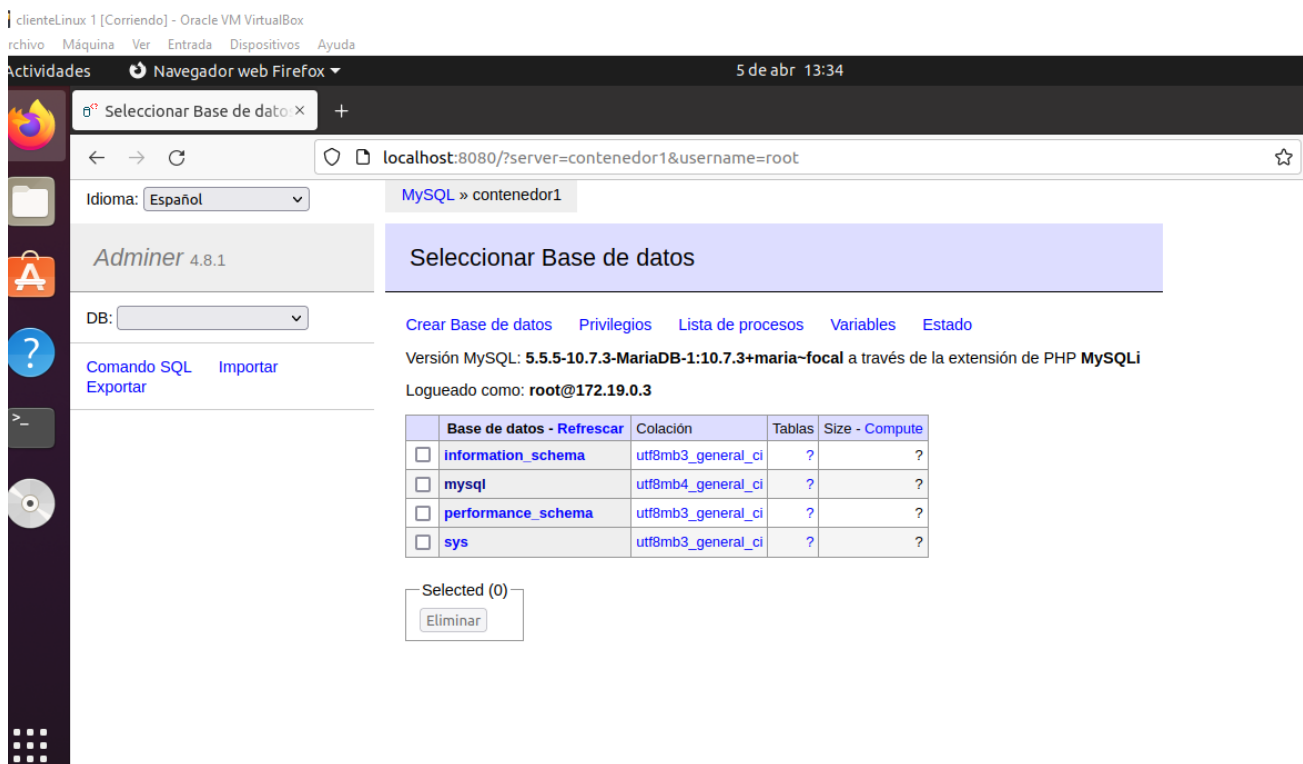
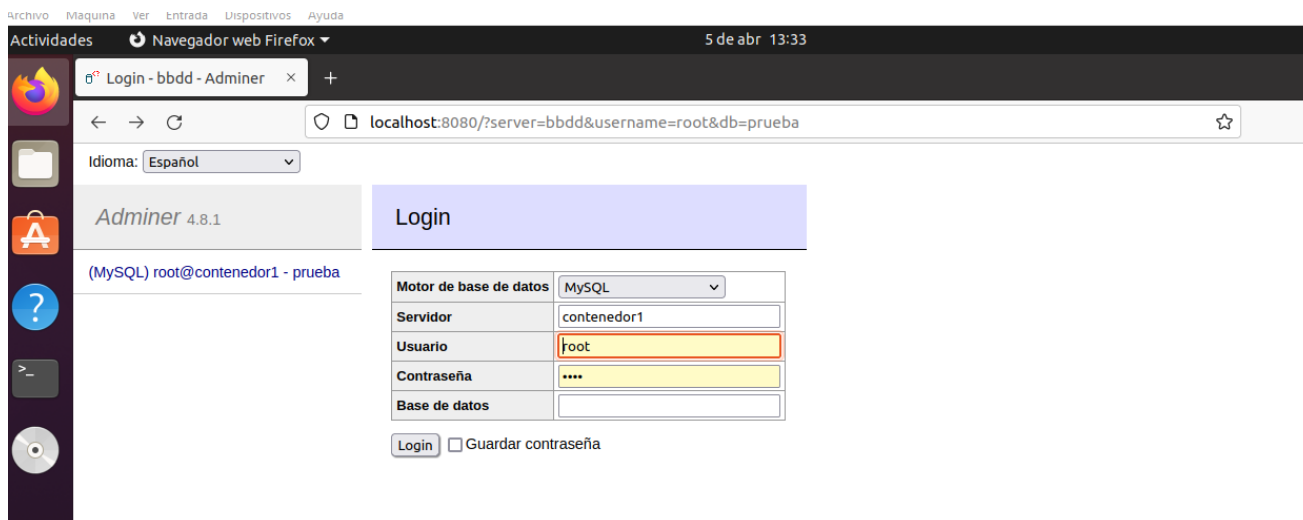
```
docker run -d --name contenedor2 --link contenedor1:mariadb --network redbd -p 8080:8080 adminer
```

```
daw@clienteLinux:~$ docker run -d --name contenedor2 --link contenedor1:mariadb --network redbd -p 8080:8080 adminer  
cc9ee8a0710ad1743ad7c65dee83991bda391220784f11c2015b1f527e91da28  
daw@clienteLinux:~$
```

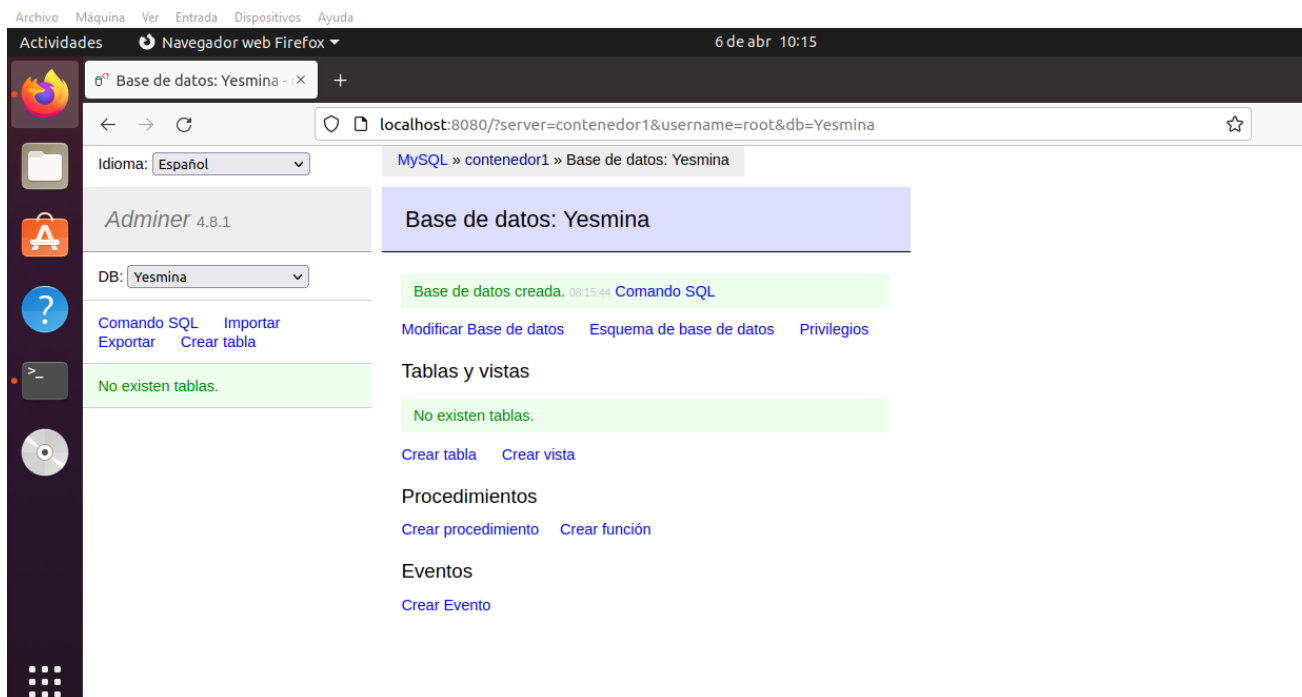
-compruebo que los dos contenedores estan ok

```
daw@clienteLinux:~$ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES  
cc9ee8a0710a   adminer   "entrypoint.sh docke..." 28 seconds ago Up 27 seconds 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp contenedor2  
beab88f84b43   mariadb   "docker-entrypoint.s..." 4 minutes ago Up 4 minutes   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp contenedor1  
daw@clienteLinux:~$
```

### 4. Comprueba que el contenedor Adminer puede conectar con el contenedor mysql abriendo un navegador web y accediendo a la URL: <http://localhost:8080>

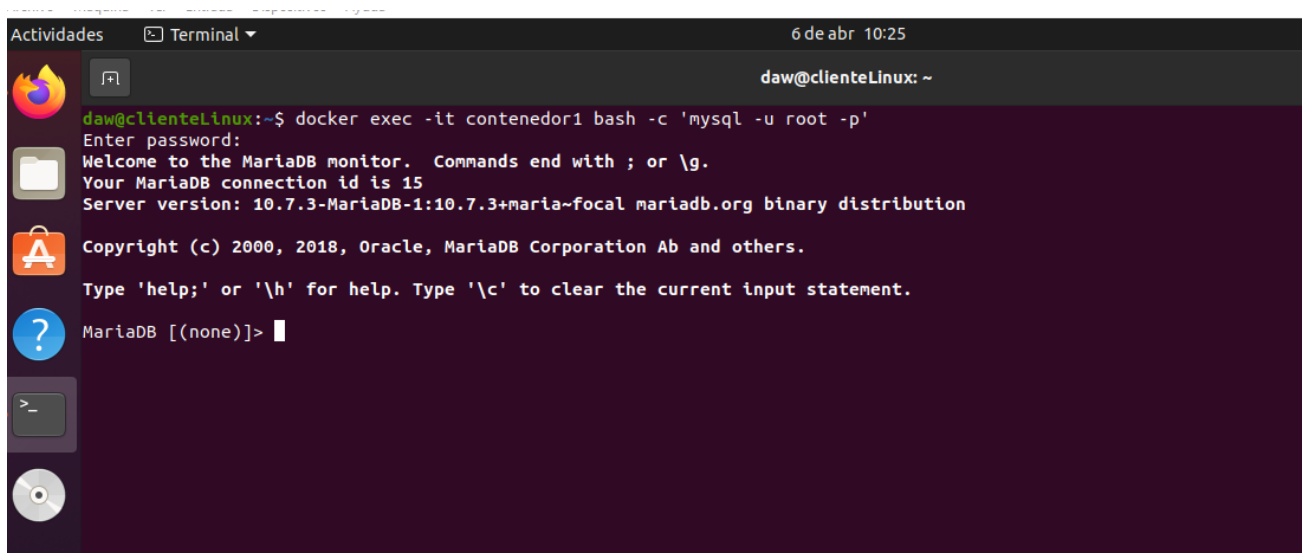


*Pantallazo donde se vea la creación de una BD con la interfaz web Adminer*



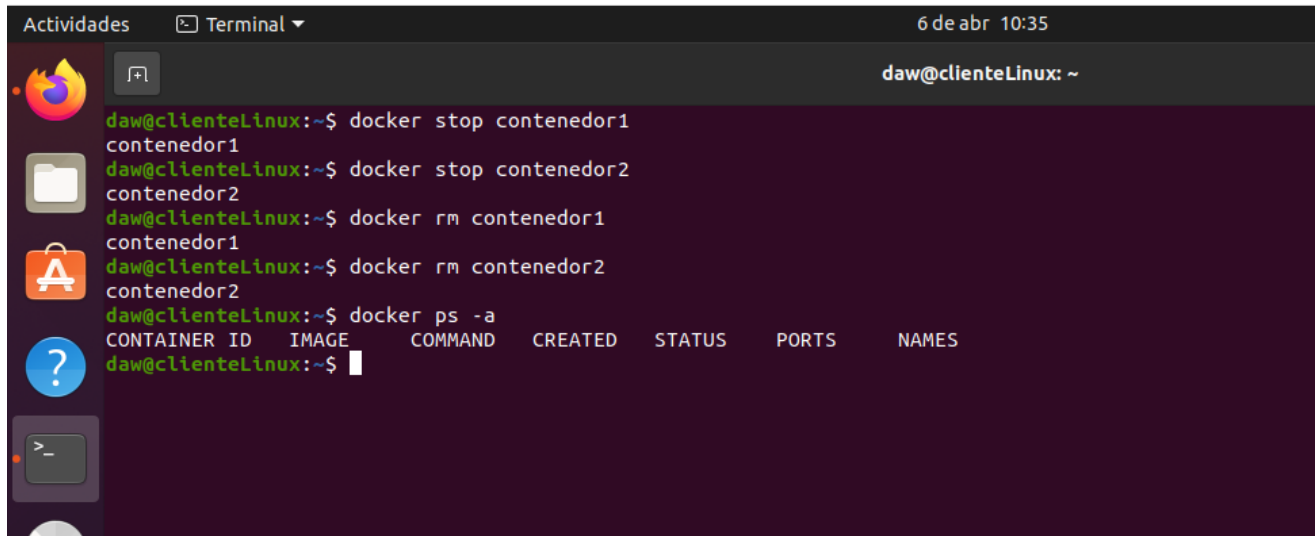
*Pantallazo donde se entre a la consola del servidor web en modo texto y se compruebe que se ha creado la BD*

```
docker exec -it contenedor1 bash -c 'mysql -u root -p'
```



## Borrar los contenedores, la red, y los volúmenes utilizados

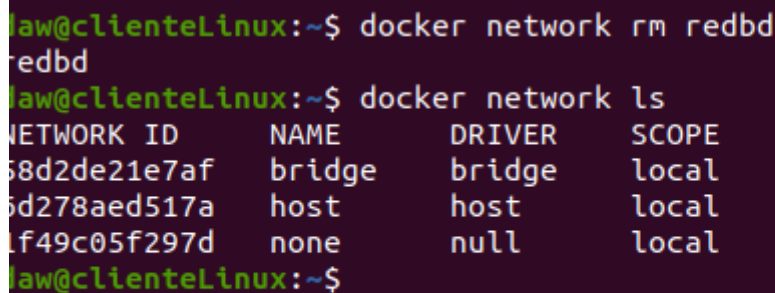
-Paramos y borramos los contenedores



```
Actividades Terminal 6 de abr 10:35 daw@clienteLinux: ~
daw@clienteLinux:~$ docker stop contenedor1
contenedor1
daw@clienteLinux:~$ docker stop contenedor2
contenedor2
daw@clienteLinux:~$ docker rm contenedor1
contenedor1
daw@clienteLinux:~$ docker rm contenedor2
contenedor2
daw@clienteLinux:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
daw@clienteLinux:~$
```

-Ahora ya podriamos borrar la red

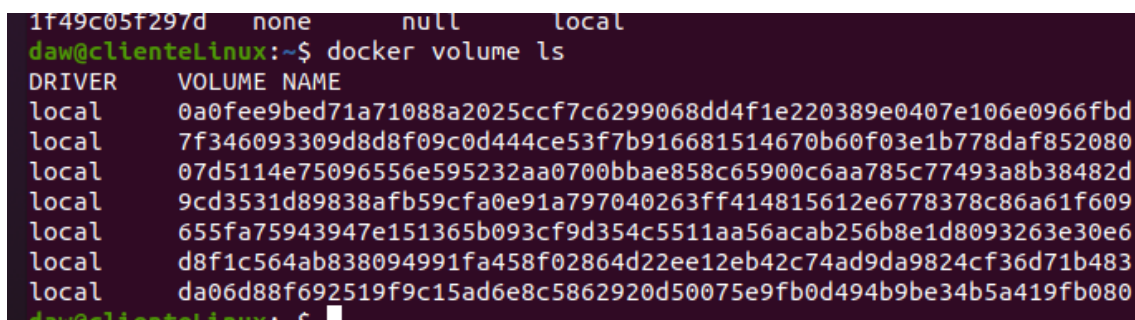
```
docker network rm redbd
```



```
daw@clienteLinux:~$ docker network rm redbd
redbd
daw@clienteLinux:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
58d2de21e7af   bridge    bridge    local
6d278aed517a   host      host      local
1f49c05f297d   none      null      local
daw@clienteLinux:~$
```

-Y por ultimo busco y limpio todos los volúmenes creados ( con lo del almacenamiento persistente )y porque me confundi varias veces al crear los contenedores .

```
docker volume ls
```



```
1f49c05f297d   none      null      local
daw@clienteLinux:~$ docker volume ls
DRIVER    VOLUME NAME
local     0a0fee9bed71a71088a2025ccf7c6299068dd4f1e220389e0407e106e0966fbd
local     7f346093309d8d8f09c0d444ce53f7b916681514670b60f03e1b778daf852080
local     07d5114e75096556e595232aa0700bbae858c65900c6aa785c77493a8b38482d
local     9cd3531d89838afb59cfa0e91a797040263ff414815612e6778378c86a61f609
local     655fa75943947e151365b093cf9d354c5511aa56acab256b8e1d8093263e30e6
local     d8f1c564ab838094991fa458f02864d22ee12eb42c74ad9da9824cf36d71b483
local     da06d88f692519f9c15ad6e8c5862920d50075e9fb0d494b9be34b5a419fb080
daw@clienteLinux:~$
```

```
docker volume prune
```

```
daw@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:
0a0fee9bed71a71088a2025ccf7c6299068dd4f1e220389e0407e106e0966fbd
655fa75943947e151365b093cf9d354c5511aa56acab256b8e1d8093263e30e6
7f346093309d8d8f09c0d444ce53f7b916681514670b60f03e1b778daf852080
9cd3531d89838afb59cfa0e91a797040263ff414815612e6778378c86a61f609
d8f1c564ab838094991fa458f02864d22ee12eb42c74ad9da9824cf36d71b483
da06d88f692519f9c15ad6e8c5862920d50075e9fb0d494b9be34b5a419fb080
07d5114e75096556e595232aa0700bbae858c65900c6aa785c77493a8b38482d

Total reclaimed space: 1.015GB
```

-compruebo que se hayan borrado

```
Total reclaimed space: 1.015GB
daw@clienteLinux:~$ docker volume ls
DRIVER      VOLUME NAME
daw@clienteLinux:~$
```