

# Redes Docker

Carlota Menéndez Álvarez

## Redes Docker

Trabajar con redes Docker

Ejercicio 1

Ejercicio 2

Ejercicio 3

Ejercicio 4

Ejercicio 5

Despliegue de Nextcloud + mariadb

Ejercicio 1

Ejercicio 2

Ejercicio 3

Ejercicio 4

## Trabajar con redes Docker

### Ejercicio 1

Vamos a crear dos redes de ese tipo (BRIDGE) con los siguientes datos:

#### Red1

Nombre: red1

Dirección de red: 172.28.0.0

Máscara de red: 255.255.0.0

Gateway: 172.28.0.1

#### Red2

Nombre: red2

Es resto de los datos será proporcionados automáticamente por Docker.

*Creación de red1*

```
daw@daw-docker:~$ docker network create -d bridge --subnet 172.28.0.0/16 --gateway 172.28.0.1 red1
91e2b962f3882d4d2b2d8feaed0db6fd90b1f7b2c7363b82e294361ecae9390c
daw@daw-docker:~$
```

*Creación de red2*

```
91e2b962f3882d4d2b2d8feaed0db6fd90b1f7b2c7363b82e294361ecae9390c
daw@daw-docker:~$ docker network create red2
4eec54978e7edc856bcb5238d45a64aeb29aa548f66ddddd813d54c5b49eea6ef
daw@daw-docker:~$
```

## Ejercicio 2

Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host1 , como IP 172.28.0.10 y que esté conectado a la red1. Lo llamaremos u1 .

Comando:

```
docker run -it --name contenedor1 --network red1 --ip 172.28.0.10 --hostname host1 ubuntu:20.04
```

```
See 'docker run --help'.
daw@daw-docker:~$ docker run -it --name contenedor1 --network red1 --ip 172.28.0.10 --hostname host1 ubuntu:20.04
Unable to find image 'ubuntu:20.04' locally
20.04: Pulling from library/ubuntu
846c0b181fff: Pull complete
Digest: sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb
Status: Downloaded newer image for ubuntu:20.04
root@host1:/#
```

Comprobaciones

```
root@host1:/# cat /etc/hostname
host1
```

## Ejercicio 3

Entrar en ese contenedor e instalar la aplicación ping ( apt update && apt install inetutils-ping ).

Instalación

```
root@host1:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1879 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2916 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1288 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2003 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [31.2 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Get:16 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2437 kB]
Get:17 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [984 kB]
```

```
root@host1:/# apt install inetutils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libidn11 netbase
The following NEW packages will be installed:
  inetutils-ping libidn11 netbase
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 120 kB of archives.
After this operation, 657 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 netbase all 6.1 [13.1 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2ubuntu2 [46.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 inetutils-ping amd64 2:1.9.4-11ubuntu0.1 [60.7 kB]
Fetched 120 kB in 1s (140 kB/s)
```

## Ejercicio 4

Poner en ejecución un contenedor de la imagen ubuntu:20.04 que tenga como hostname host2 y que esté conectado a la red2. En este caso será docker el que le de una IP correspondiente a esa red. Lo llamaremos u2 .

```
docker run -it --name u2 --network red2 --hostname host2 ubuntu:20.04
```

```
u2
daw@daw-docker:~$ docker run -it --name u2 --network red2 --hostname host2 ubuntu:20.04
root@host2:/#
```

Comprobación

```
root@host2:/# cat /etc/hostname
host2
root@host2:/#
```

## Ejercicio 5

Entrar en ese contenedor e instalar la aplicación ping ( apt update && apt install inetutils-ping ).

Instalación:

```
root@host2:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1879 kB]
Get:9 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2437 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [984 kB]
Get:12 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [59.5 kB]
Fetched 101 MB in 10s (10.1 MB/s)
root@host2:/# apt install inetutils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libidn11 netbase
The following NEW packages will be installed:
  inetutils-ping libidn11 netbase
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 120 kB of archives.
After this operation, 657 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

## Comprobaciones

Contenedor 1:

```
    "MacAddress": " ",
    "Networks": {
      "red1": {
        "IPAMConfig": {
          "IPv4Address": "172.28.0.10"
        },
        "Links": null,
        "Aliases": [
          "686cb3b7abf8",
          "host1"
        ],
      },
    },
  },
}
```

Contenedor 2:

```
root@host2:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.23.0.2 netmask 255.255.0.0 broadcast 172.23.255.255
    ether 02:42:ac:17:00:02 txqueuelen 0 (Ethernet)
    RX packets 78 bytes 202548 (202.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 56 bytes 3201 (3.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Ping entre contenedores:

```
root@host2:/# ping 172.28.0.10
PING 172.28.0.10 (172.28.0.10): 56 data bytes
^Z
[1]+  Stopped                  ping 172.28.0.10
root@host2:/#
```

```
root@host2:/# ping host1
ping: unknown host
root@host2:/#
```

Cambio la red del contenedor1:

```
daw@daw-docker:~$ docker network connect red2 contenedor1
daw@daw-docker:~$
```

Ahora mismo estan en la misma red los dos contenedores, y hago ping :

```
ctrl
daw@daw-docker:~$ docker start -i contenedor1
root@host1:/# ping 172.23.0.2
PING 172.23.0.2 (172.23.0.2): 56 data bytes
64 bytes from 172.23.0.2: icmp_seq=0 ttl=64 time=0.074 ms
64 bytes from 172.23.0.2: icmp_seq=1 ttl=64 time=0.213 ms
64 bytes from 172.23.0.2: icmp_seq=2 ttl=64 time=0.231 ms
64 bytes from 172.23.0.2: icmp_seq=3 ttl=64 time=0.049 ms
```

Se ven, pruebo haciendo Ping con el nombre del host

```

root@host1:/# ping host1
PING host1 (172.28.0.10): 56 data bytes
64 bytes from 172.28.0.10: icmp_seq=0 ttl=64 time=0.064 ms
64 bytes from 172.28.0.10: icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from 172.28.0.10: icmp_seq=2 ttl=64 time=0.322 ms
64 bytes from 172.28.0.10: icmp_seq=3 ttl=64 time=0.117 ms
64 bytes from 172.28.0.10: icmp_seq=4 ttl=64 time=0.072 ms
64 bytes from 172.28.0.10: icmp_seq=5 ttl=64 time=0.123 ms
64 bytes from 172.28.0.10: icmp_seq=6 ttl=64 time=0.266 ms
64 bytes from 172.28.0.10: icmp_seq=7 ttl=64 time=0.091 ms
64 bytes from 172.28.0.10: icmp_seq=8 ttl=64 time=0.263 ms

```

## Despliegue de Nextcloud + mariadb

### Ejercicio 1

Crea una red de tipo bridge.

```
docker network create redNueva
```

```

daw@daw-docker:~$ docker network create redNueva
59e27ceec94363e3c6e80520c79b0fd2a653ccea1543a34d11b312e9da6d7a6e
daw@daw-docker:~$

```

### Ejercicio 2

Crea el contenedor de la base de datos conectado a la ed que has creado. La base de datos se debe

configurar para crear una base de dato y un usuario. Además el contenedor debe utilizar almacenamiento (volúmenes o bind mount) para guardar la información. Puedes seguir la documentación de mariadb o la de PostgreSQL .

```

docker run -d --name contenedorMaria --network redNueva -v /opt/mysql_base1:/var/lib/mysql -e MYSQL_DATABASE=b1 -e MYSQL_USER=usu1 -e MYSQL_PASSWORD=123456 -e MYSQL_ROOT_PASSWORD=123456 mariadb:10.5

```

```

$ sudo apt install docker
$ docker run -d --name contenedorMaria --network redNueva -v /opt/mysql_base1:/var/lib/mysql -e MYSQL_DATABASE=b1 -e MYSQL_USER=usu1 -e MYSQL_PASSWORD=123456 -e MYSQL_ROOT_PASSWORD=123456 mariadb:10.5
Unable to find image 'mariadb:10.5' locally
10.5: Pulling from library/mariadb
846c0b181fff: Already exists
2279a7485340: Pull complete
30c7fe7ba3fd: Pull complete
db43169afb5c: Pull complete
179c81612958: Pull complete
770ca0d9f1d8: Pull complete
fb77e1d666e4: Pull complete
13c598045ade: Pull complete
Digest: sha256:b16e4cbe3d507bff1e40542196854a8da16bd1831e2cc413d7166f9b698ec30f
Status: Downloaded newer image for mariadb:10.5
d7682b3e24f240e793273bc0c726ec0140edd38f7638c7ed07d275940d4d4fdb

```

## Ejercicio 3

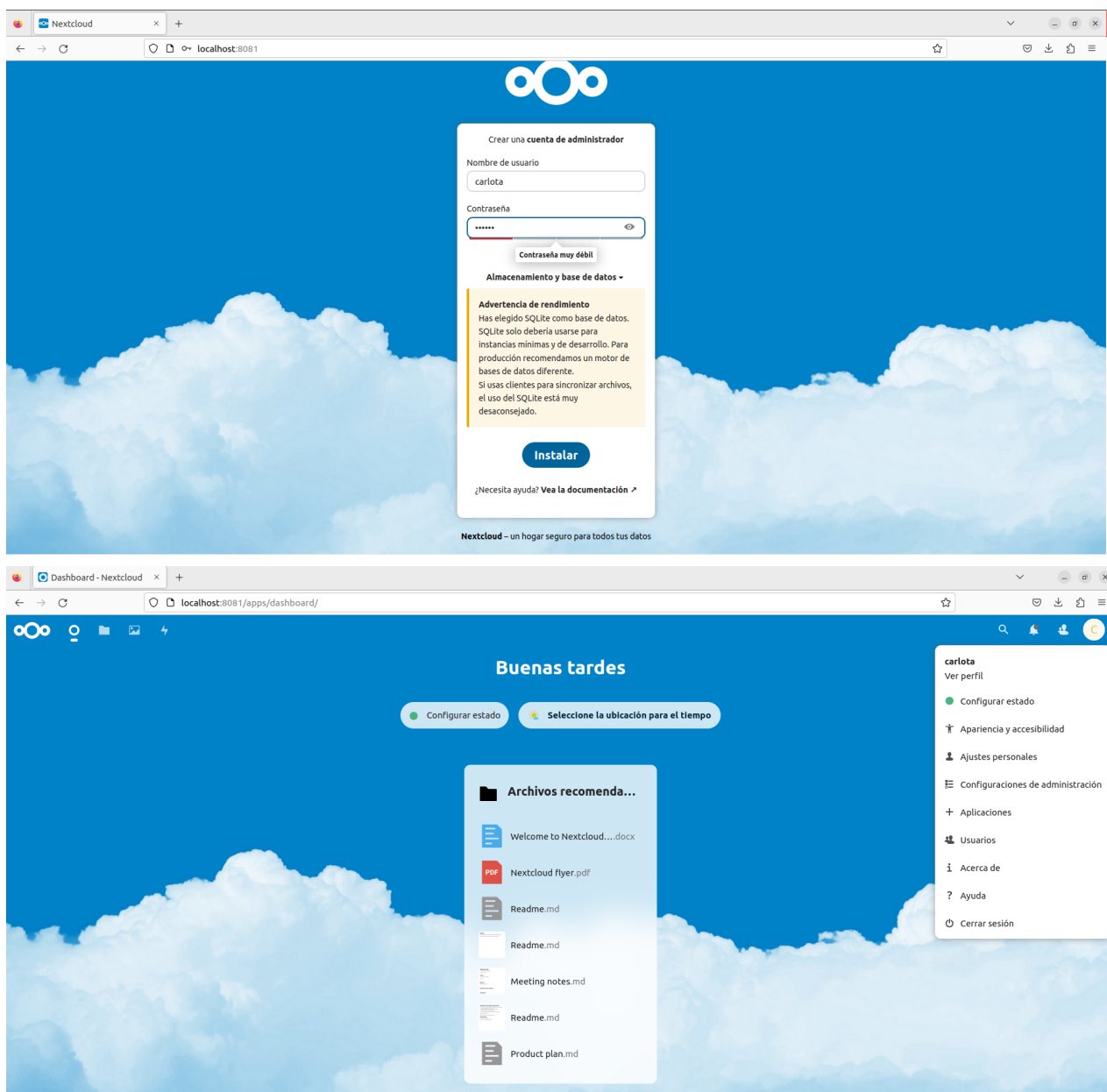
A continuación, siguiendo la documentación de la imagen nextcloud , crea un contenedor conectado a la misma red, e indica las variables adecuadas para que se configure de forma adecuada y realice la conexión a la base de datos. El contenedor también debe ser persistente usando almacenamiento.

```
docker run -d -p 8081:80 --name contenedorMaria2 --network redNueva --link contenedorMaria -v nextcloud:/var/www/html nextcloud
```

```
daw@daw-docker:~$ docker run -d -p 8080:80 --name contenedorMaria2 --network redNueva --link contenedorMaria -v nextcloud:/var/www/html nextcloud
e7e8ba01aa4ecd4f2d08970cc092b3490d1afd7c61010d1aa7e54b69be5d727
```

## Ejercicio 4

Accede a la aplicación usando un navegador web.



A tener en cuenta:

Id de todos los contenedores q tenga `docker ps -a -q`

`docker rm -f $(docker ps -a -q)` - Borra todos los contenedores existentes.

`docker start -i contenedor`