

Tarea 4 - Reder en docker

Realizado por Pablo R.

1. Vamos a crear dos redes de tipo bridge con los siguientes datos:

- Red 1
 - Nombre: red1
 - Dirección de red: 172.28.0.0
 - Máscara de red: 255.255.0.0
 - Gateway: 172.28.0.1
- Red 2
 - Nombre: red2
 - El resto de los datos serán proporcionados automáticamente por Docker.

```
docker network create red1 --subnet 172.28.0.0/16 --gateway 172.28.0.1
docker network create red2
```

```
root@daw-docker:/home/daw# docker network create red1 --subnet 172.28.0.0/16 --gateway 172.28.0.1
d5c47074bdfdd5716953bc9ae58b7b399bfd9adc344a218002037357a381c3bd
root@daw-docker:/home/daw# docker network create red2
0d1ad09a96578f13e71f898436b1bae817241041c352576064a341056b2fb4e6
root@daw-docker:/home/daw#
```

Creamos las redes que se nos pide.

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.

```
docker run -it --name u1 --network red1 --hostname host1 --ip 172.28.0.10 ubuntu:20.04
docker start u1
```

```
root@daw-docker:/home/daw# docker run --name u1 --network red1 --hostname host1 --ip 172.28.0.10 ubuntu:20.04
root@daw-docker:/home/daw# docker start u1
u1
root@daw-docker:/home/daw#
```

Creamos el contenedor con la red1 y la ip proporcionada en el ejercicio.

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

```
docker exec -it u1 bash
apt update
apt install inetutils-ping
```

```
root@daw-docker:/home/daw# docker start u1
u1
root@daw-docker:/home/daw# docker exec -it u1 bash
root@host1:/# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 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://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [986 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 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://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2439 kB]
Get:11 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.7 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [31.2 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1290 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2920 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2009 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Fetched 25.1 MB in 18s (1412 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
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 4 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
```

Iniciamos el contenedor, entramos por bash e instalamos la herramienta ping.

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.

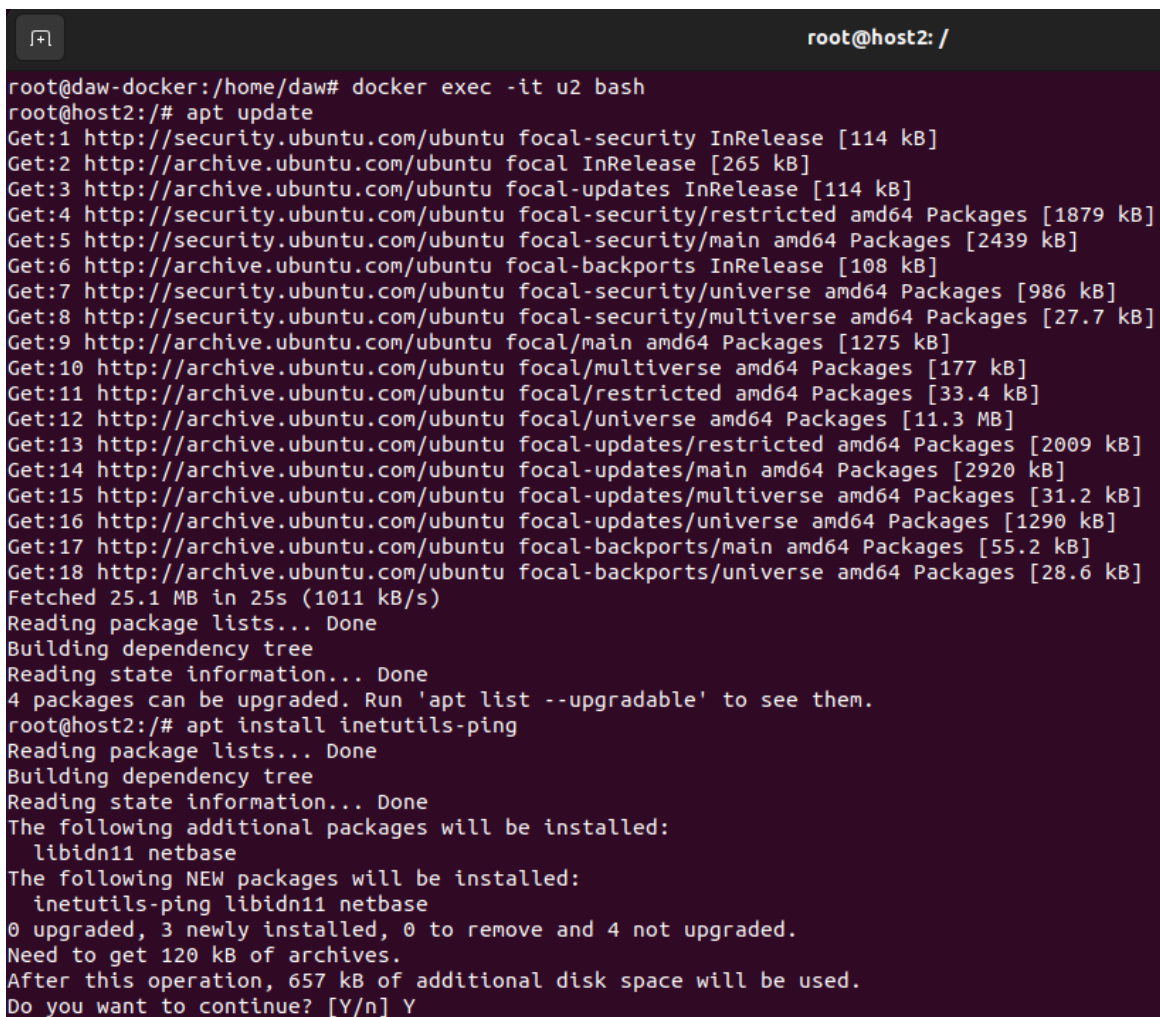
```
docker run -it --name u2 --network red2 --hostname host2 ubuntu:20.04
docker start u2
```

```
root@daw-docker:/home/daw# docker run -it --name u2 --network red2 --hostname host2 ubuntu:20.04
root@host2:/# exit
exit
root@daw-docker:/home/daw# docker start u2
u2
root@daw-docker:/home/daw# █
```

Ponemos en ejecución la segunda máquina.

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

```
docker exec -it u2 bash
apt update
apt install inetutils-ping
```



A terminal window titled 'root@host2: /' showing the execution of Docker commands. The user runs 'docker exec -it u2 bash' to enter the container. Inside the container, 'apt update' is run, showing a list of updates from various Ubuntu repositories. Then, 'apt install inetutils-ping' is run, showing the installation of 'inetutils-ping' along with 'libidn1' and 'netbase'. The terminal output is as follows:

```
root@daw-docker:/home/daw# docker exec -it u2 bash
root@host2:/# apt update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [1879 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [2439 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [986 kB]
Get:8 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.7 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2009 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2920 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [31.2 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1290 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [55.2 kB]
Get:18 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [28.6 kB]
Fetched 25.1 MB in 25s (1011 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@host2:/# apt install inetutils-ping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libidn1 netbase
The following NEW packages will be installed:
  inetutils-ping libidn1 netbase
0 upgraded, 3 newly installed, 0 to remove and 4 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
```

Instalamos la herramienta ping.

PANTALLAZOS. Pantallazos.

- Pantallazo donde se vea la configuración del contenedor u1.

```
docker inspect u1
```

- Pantallazo donde se vea la configuración del contenedor u2.



root@daw-docker: /home/daw

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "b83ebb0402e829de6a19157fba0c11b486e73687371060b29f9ba59086cf6ba8",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {},
  "SandboxKey": "/var/run/docker/netns/b83ebb0402e8",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "",
  "Gateway": "",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "",
  "IPPrefixLen": 0,
  "IPv6Gateway": "",
  "MacAddress": "",
  "Networks": {
    "red1": {
      "IPAMConfig": {
        "IPv4Address": "172.28.0.10"
      },
      "Links": null,
      "Aliases": [
        "02fd4630d442",
        "host1"
      ],
      "NetworkID": "d5c47074bdfdd5716953bc9ae58b7b399bfd9adc344a218002037357a381c3bd",
      "EndpointID": "92386b97fec21f885e58e21660b88d1a967359bbfeb1f777dc15442936bc14f7",
      "Gateway": "172.28.0.1",
      "IPAddress": "172.28.0.10",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:1c:00:0a",
      "DriverOpts": null
    }
  }
}
}
]
root@daw-docker:/home/daw# docker inspect u1
```

```
docker inspect u2
```

```

    "Labels": {}
  },
  "NetworkSettings": {
    "Bridge": "",
    "SandboxID": "45607cdf7e6d842e87571230d9782f758fe3363bf9dc218085f3d2d9bc3c38b8",
    "HairpinMode": false,
    "LinkLocalIPv6Address": "",
    "LinkLocalIPv6PrefixLen": 0,
    "Ports": {},
    "SandboxKey": "/var/run/docker/netns/45607cdf7e6d",
    "SecondaryIPAddresses": null,
    "SecondaryIPv6Addresses": null,
    "EndpointID": "",
    "Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "IPAddress": "",
    "IPPrefixLen": 0,
    "IPv6Gateway": "",
    "MacAddress": "",
    "Networks": {
      "red2": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
          "7cccdc269d43",
          "host2"
        ],
        "NetworkID": "0d1ad09a96578f13e71f898436b1bae817241041c352576064a341056b2fb4e6",
        "EndpointID": "cce70b83a39a0e2ba1167e4b166344a3918990e42119303e0b2e81a383393383",
        "Gateway": "172.19.0.1",
        "IPAddress": "172.19.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:13:00:02",
        "DriverOpts": null
      }
    }
  }
}
]
root@daw-docker:/home/daw# docker inspect u2

```

- Pantallazo donde desde cualquiera de los dos contenedores se pueda ver no hacemos ping al otro ni por ip ni por nombre.

```

docker exec -it u1 bash
ping u2
ping 172.19.0.2

```

```

root@daw-docker:/home/daw# docker exec -it u1 bash
root@host1:/# ping u2
ping: unknown host
root@host1:/# ping 172.19.0.2
PING 172.19.0.2 (172.19.0.2): 56 data bytes
^C--- 172.19.0.2 ping statistics ---
9 packets transmitted, 0 packets received, 100% packet loss
root@host1:/#

```

Intentamos hacer ping desde u1 a u2.

- Pantallazo donde se pueda comprobar que si conectamos el contenedor u1 a la red2 (con docker network connect), desde el contenedor u1, tenemos al contenedor u2 mediante ping, tanto por nombre como por ip.

```
docker network connect red2 u1
```

```
root@daw-docker:/home/daw# docker network connect red2 u1
root@daw-docker:/home/daw# docker exec -it u1 bash
root@host1:/# ping u2
PING u2 (172.19.0.2): 56 data bytes
64 bytes from 172.19.0.2: icmp_seq=0 ttl=64 time=0.177 ms
64 bytes from 172.19.0.2: icmp_seq=1 ttl=64 time=0.239 ms
^C--- u2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.177/0.208/0.239/0.031 ms
root@host1:/# ping 172.19.0.2
PING 172.19.0.2 (172.19.0.2): 56 data bytes
64 bytes from 172.19.0.2: icmp_seq=0 ttl=64 time=0.247 ms
64 bytes from 172.19.0.2: icmp_seq=1 ttl=64 time=0.234 ms
^C--- 172.19.0.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.234/0.240/0.247/0.000 ms
root@host1:/#
```

Despliegue de Nextcloud + mariadb/postgreSQL

Vamos a desplegar la aplicación nextcloud con una base de datos (puedes elegir mariadb o PostgreSQL) (NOTA: Para que no te de errores utiliza la imagen mariadb:10.5). Te puede servir el ejercicio que hemos realizado para desplegar Wordpress . Para ello sigue los siguientes pasos:

1. Crea una red de tipo bridge.

```
docker network create mired
```

```
root@pablo-vm:/home/pablo# docker network create mired
4c7d8d1e62a2caca8f18d232c9f9e45235447315e95818b37a55087fd495cbe3
root@pablo-vm:/home/pablo#
```

Creamos la red bridge.

2. Crea el contenedor de la base de datos conectado a la red que has creado. La base de datos se debe configurar para crear una base de datos 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 .

```
mkdir /home/mariadb
docker run -it --name midb --network mired --env
MARIADB_DATABASE=nextcloud --e MARIADB_USER=nextcloud --env
MARIADB_PASSWORD=nextcloud MARIADB_ROOT_PASSWORD=mipasswordsecreta -
v /home/mariadb:/var/lib/mysql mariadb:latest
```



```

root@pablo-vm:/home/pablo# mkdir /home/mariadb
root@pablo-vm:/home/pablo# docker run -it --name midb --network mired --env MARIADB_DATABASE=nextcloud --env MARIADB_USER=nextcloud --env MARIADB_PASSWORD=nextcloud --env MARIADB_ROOT_PASSWORD=mipasswordsecreta -v /home/mariadb:/var/lib/mysql mariadb:latest
2023-01-28 10:53:49+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.10.2+maria-ubu2204 started.
2023-01-28 10:53:49+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-01-28 10:53:49+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.10.2+maria-ubu2204 started.
2023-01-28 10:53:49+00:00 [Note] [Entrypoint]: Initializing database files

PLEASE REMEMBER TO SET A PASSWORD FOR THE MariaDB root USER !
To do so, start the server, then issue the following command:

'/usr/bin/mysql_secure_installation'

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the MariaDB Knowledgebase at https://mariadb.com/kb

Please report any problems at https://mariadb.org/jira

The latest information about MariaDB is available at https://mariadb.org/.

Consider joining MariaDB's strong and vibrant community:
https://mariadb.org/get-involved/

2023-01-28 10:53:50+00:00 [Note] [Entrypoint]: Database files initialized
2023-01-28 10:53:50+00:00 [Note] [Entrypoint]: Starting temporary server
2023-01-28 10:53:50+00:00 [Note] [Entrypoint]: Waiting for server startup
2023-01-28 10:53:50 0 [Note] mariadbd (server 10.10.2-MariaDB-1:10.10.2+maria-ubu2204) starting as process 93 ...

```

Inicializamos la bd.

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.

```

mkdir /home/nextcloud
docker run -d -v /home/nextcloud:/var/www/html --network mired --env MYSQL_HOST=midb --env MYSQL_DATABASE=nextcloud --env MYSQL_USER=nextcloud --env MYSQL_PASSWORD=mipassword nextcloud

```

```

root@pablo-vm:/home/pablo# mkdir /home/nextcloud
root@pablo-vm:/home/pablo# docker run -d -v /home/nextcloud:/var/www/html --network mired --env MYSQL_HOST=midb --env MYSQL_DATABASE=nextcloud --env MYSQL_USER=nextcloud --env MYSQL_PASSWORD=nextcloud nextcloud
5df882cfac579c9baca33b064f344b5ba72dfb6aecbd5a9b1518fda5685090f
root@pablo-vm:/home/pablo# ^C

```

Inicializamos nextcloud.

4. Accede a la aplicación usando un navegador web.

