

Ejercicios Docker 4

Trabajar con redes docker

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

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

```
daw@daw-docker:~$ docker network create -d bridge --subnet 172.28.0.0/16 --gateway 172.28.0.1  
reed1  
46a55077f8caa37326a367036d2a87beaa073a3e9ee301db4081ef4aa066c4a1
```

Red2 Nombre: red2 Es resto de los datos será proporcionados automáticamente por Docker.

```
docker network create red2
```

```
daw@daw-docker:~$ docker network create red2  
12c635b173162c17c37ba1942af6ffbf8b17f030ee006bb6c052f8188a2342b8
```

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 reed1 --ip 172.28.0.10 --hostname host1 ubuntu:20.04
```

□

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

```
apt update
```

□

```
apt install inetutils-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
```

□

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

```
apt update
```

□

```
apt install inetutils-ping
```

□

El documento debe contener, además, los siguientes pantallazos:

- Pantallazo donde se vea la configuración de red del contenedor u1.

```
ifconfig en host1
```

□

- Pantallazo donde se vea la configuración de red del contenedor u2.

```
ifconfig en host2
```

□

- Pantallazo donde desde cualquiera de los dos contenedores se pueda ver que no podemos hacer ping al otro ni por ip ni por nombre.
- Pantallazo donde se pueda comprobar que si conectamos el contenedor u1 a la red2 (con docker network connect), desde el contenedor u1, tenemos acceso al contenedor u2 mediante ping, tanto por nombre como por ip.

```
docker network connect red2 u1
```

□

Despliegue de Nextcloud + mariadb/postgreSQL

1. Crea una red de tipo bridge.

```
docker network create rbridge
```

□

```
docker network inspect rbridge
```

□

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 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 mymaria --env MARIADB_USER=usuario --env MARIADB_PASSWORD=contraseña --env MARIADB_ROOT_PASSWORD=laboral1 -v volumensql:/var/run/mysqld --network rbridge -p 8080:80 mariadb:10.5
```

□

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 -v volumennextcloud:/var/www/html --network rbridge --env MYSQL_DATABASE=mymaria -p 8081:80 nextcloud
```

□

4. Accede a la aplicación usando un navegador web.

```
localhost:8081
```

□ □ □



created with the evaluation version of [Markdown Monster](#)