



Desarrollo de Aplicaciones Web
Desarrollo de Aplicaciones Multiplataforma
CURSO 24-25



Actividad práctica 6

Ejercicio Práctico - Subnetting con Docker

Implanta la solución de la [práctica anterior](#) en un entorno de desarrollo real.

Una vez implantado prueba la comunicación entre contenedores mediante el comando ping. Recuerda incluir tu nombre en el hostname de cada contenedor que crees.



Paso a paso:

1. Creamos las subredes con:

- `docker network create --subnet=172.20.100.0/27 frontend_net`
- `docker network create --subnet=172.20.100.32/27 backend_net`
- `docker network create --subnet=172.20.100.64/27 db_net`
- `docker network create --subnet=172.20.100.96/27 cache_net`
- `docker network create --subnet=172.20.100.128/27 monitoring_net`

```
PS C:\Users\guill> docker network create --subnet=172.20.100.0/27 frontend_net
0adb184d4f9a85b351d29f3ac4d0ac3981bc0a78decac36eec11802664e09ef9
PS C:\Users\guill> docker network create --subnet=172.20.100.32/27 backend_net
10c2b7aad35457776244100b7c7ffdeaf32c4fc2d7b6f102b1a541907610c956
PS C:\Users\guill> docker network create --subnet=172.20.100.64/27 db_net
6c1382de5e059a3f69da43a1aa12c683b98c5b2bb1971429ef854669a769af35
PS C:\Users\guill> docker network create --subnet=172.20.100.96/27 cache_net
7e490f3789bb49b2c690c8ce036abdb61a7af76eb3620c04546f05407c02af06
PS C:\Users\guill> docker network create --subnet=172.20.100.128/27 monitoring_net
8fc996b9622ffbef06d9f28131c3553e650c47023e1122e4fe0af45412aa17e9
```

2. Comprobamos que están creadas y con sus ip correctamente asignadas:

- `docker network inspect $(docker network ls -q) | jq -r '.[] | "\(.Name): \(.IPAM.Config[]?.Subnet) "'`

```
C:\Users\guill>for /f %i in ('docker network ls -q') do @docker network inspect %i --format "{{.Name}}: {{range .IPAM.Config}}{{.Subnet}}{{end}}"
backend_net: 172.20.100.32/27
bridge: 172.17.0.0/16
cache_net: 172.20.100.96/27
db_net: 172.20.100.64/27
frontend_net: 172.20.100.0/27
host:
monitoring_net: 172.20.100.128/27
none:
```

3. Crearemos 1 contenedor de prueba asociado a cada una de las subredes:

- `docker run -d --name frontend_container_donGuillermo --network frontend_net --ip 172.20.100.10 nginx`

```
C:\Users\guill>docker run -d --name frontend_container_don_Guillermo --network frontend_net --ip 172.20.100.10 nginx
ed541269a0dfb8c771eb6d7b33e82511099b208759e95ac7684675fc44b71135
```

- ★ Podemos acceder a este contenedor con: `docker exec -it frontend_container_don_Guillermo bash`

```
C:\Users\guill>docker exec -it frontend_container_don_Guillermo bash
root@ed541269a0df:/#
```



- `docker run -it --name backend_container_don_Guillermo --network backend_net --ip 172.20.100.40 node bash`

```
C:\Users\guill>docker run -it --name backend_container_don_Guillermo --network backend_net --ip 172.20.100.40 node bash
```

- `docker run -it --name db_container_don_Guillermo --network db_net --ip 172.20.100.70 postgres bash`

```
C:\Users\guill>docker run -it --name db_container_don_Guillermo --network db_net --ip 172.20.100.70 postgres bash
```

- `docker run -it --name cache_container_don_Guillermo --network cache_net --ip 172.20.100.100 redis bash`

```
C:\Users\guill>docker run -it --name cache_container_don_Guillermo --network cache_net --ip 172.20.100.100 redis bash
```

- `docker run -d --name monitoring_container_don_Guillermo --network monitoring_net --ip 172.20.100.130 -p 9090:9090 prom/prometheus`

```
C:\Users\guill>docker run -d --name monitoring_container_don_Guillermo --network monitoring_net --ip 172.20.100.130 -p 9090:9090 prom/prometheus
```

4. Comprobamos que no existe comunicaciones entre ninguno de los contenedores. Vemos que Docker por defecto aísla las redes.

- Para realizar la comprobación tendremos que instalar el paquete que permite usar el comando ping:

★ `apt update && apt install iputils-ping -y`

(PRIMERO INSTALAMOS EL PING EN EL CONTENEDOR BASH DE LA RED FRONTEND)

```
C:\Users\guill>docker exec -it frontend_container_don_Guillermo bash
root@ed541269a0df:/# apt update && apt install iputils-ping -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8792 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [512 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [252 kB]
```

SI INTENTAMOS HACERLE PING DESDE OTRA RED (EN ESTE CASO HA SIDO DB_NET) VEREMOS QUE NO HAY COMUNICACIONES, LOS PAQUETES NO LLEGAN.



```
--- 172.20.100.50 ping statistics ---
6 packets transmitted, 0 received, 100% packet loss, time 5236ms

root@198f778e85ba:/# ping 172.20.100.10
PING 172.20.100.10 (172.20.100.10) 56(84) bytes of data.

^C
--- 172.20.100.10 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9405ms

root@198f778e85ba:/#
```

5. Creamos un segundo contenedor dentro de alguna de las redes y comprobamos que si que existe comunicación entre ellos. (Recuerda instalar el paquete `iputils-ping` también en este nuevo contenedor)

A CONTINUACIÓN, INSTALAMOS OTRA IMAGEN EN LA RED DE BACKEND, DESCARGAMOS LA UTILIDAD DE PING Y PROBAMOS UNA COMUNICACIÓN ENTRE AMBAS:

```
C:\Users\guill>docker run -it --name test_backend --network backend_net --ip 172.20.100.35 debian bash
root@be593f4df689:/# apt update && apt install iputils-ping -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8792 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [512 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [252 kB]
```

```
root@be593f4df689:/# ping 172.20.100.35
PING 172.20.100.35 (172.20.100.35) 56(84) bytes of data.
64 bytes from 172.20.100.35: icmp_seq=1 ttl=64 time=1.67 ms
64 bytes from 172.20.100.35: icmp_seq=2 ttl=64 time=0.085 ms
```

```
^C
--- 172.20.100.35 ping statistics ---
211 packets transmitted, 211 received, 0% packet loss, time 218367ms
rtt min/avg/max/mdev = 0.036/0.097/1.670/0.123 ms
```