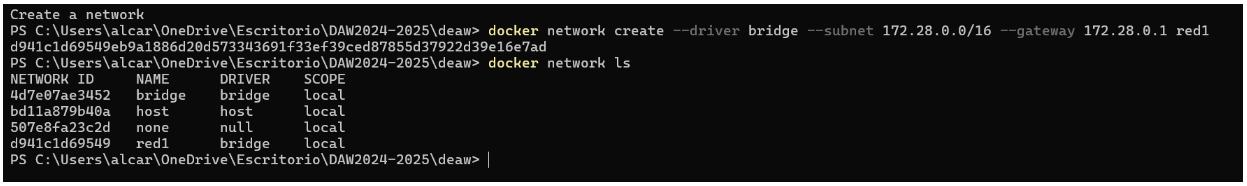
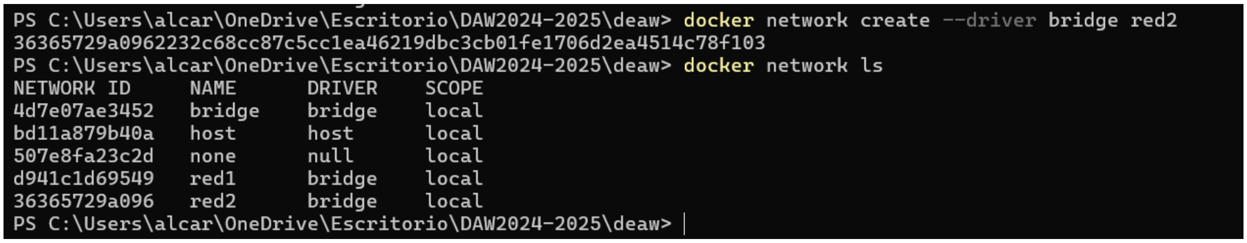
Configuración de redes y conectividad en Contenedores

1. Crear redes de tipo bridge:
   - Crea las siguientes redes:
     - red1:
       - Dirección de red: `172.28.0.0/16`
       - Puerta de enlace: `172.28.0.1`



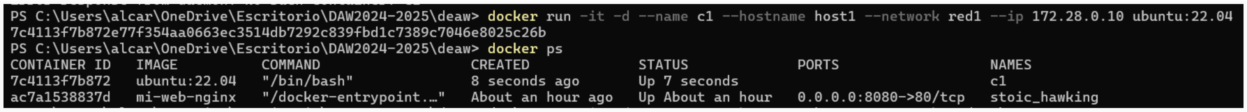
```
Create a network
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker network create --driver bridge --subnet 172.28.0.0/16 --gateway 172.28.0.1 red1
d941c1d69549eb9a1886d20d573343691f33ef39ced87855d37922d39e16e7ad
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
4d7e07ae3452   bridge    bridge    local
bd11a879b40a   host      host      local
507e8fa23c2d   none      null      local
d941c1d69549   red1      bridge    local
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw>
```

     - red2: (Configuración automática de dirección y puerta de enlace)



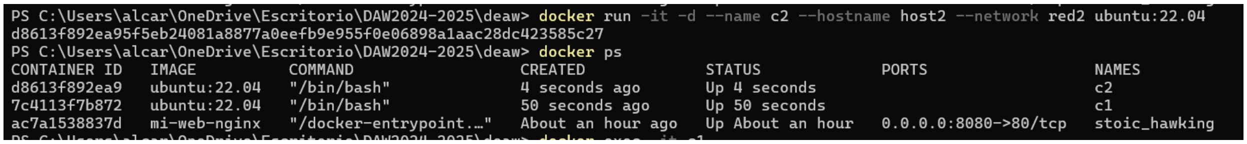
```
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker network create --driver bridge red2
36365729a0962232c68cc87c5cc1ea46219dbc3cb01fe1706d2ea4514c78f103
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
4d7e07ae3452   bridge    bridge    local
bd11a879b40a   host      host      local
507e8fa23c2d   none      null      local
d941c1d69549   red1      bridge    local
36365729a096   red2      bridge    local
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw>
```

2. Configurar el contenedor 'c1':
   - Arranca un contenedor llamado `c1` basado en la imagen `ubuntu:22.04`.
   - Configura los siguientes parámetros:
     - Hostname: `host1`
     - IP: `172.28.0.10` (conectado a `red1`)
   - Una vez arrancado el contenedor, instala la aplicación `ping` ejecutando el comando:
   - `apt update && apt install -y inetutils-ping`



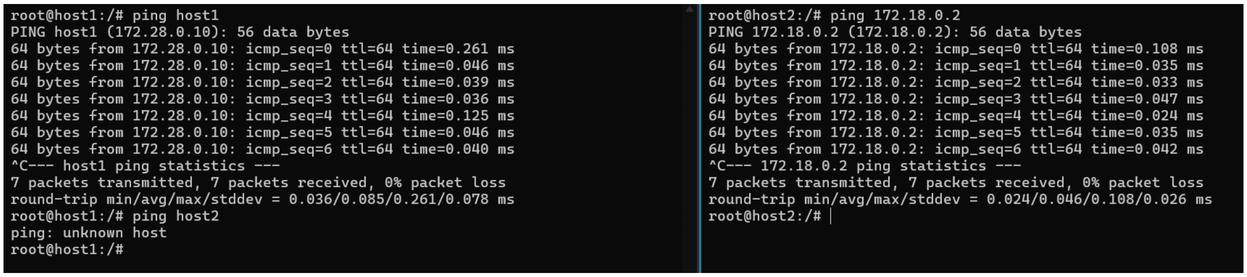
```
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker run -it -d --name c1 --hostname host1 --network red1 --ip 172.28.0.10 ubuntu:22.04
7c4113f7b872e77f354aa0663ec3514db7292c839fbd1c7389c7046e8025c26b
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS                  NAMES
7c4113f7b872   ubuntu:22.04   "/bin/bash"              8 seconds ago       Up 7 seconds                               c1
ac7a1538837d   mi-web-nginx   "/docker-entrypoint.…"   About an hour ago   Up About an hour    0.0.0.0:8080->80/tcp   stoic_hawking
```

3. Configurar el contenedor 'c2':
   - Arranca otro contenedor llamado `c2`, basado en la misma imagen `ubuntu:22.04`.
   - Configura los siguientes parámetros:
     - Hostname: `host2`
     - Conéctalo a la red `red2`.
   - Instala también la aplicación `ping` con el mismo comando usado para `c1`.



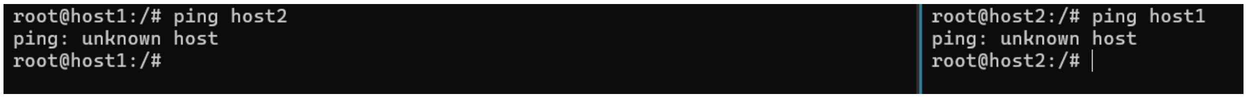
```
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker run -it -d --name c2 --hostname host2 --network red2 ubuntu:22.04
d8613f892ea95f5eb24081a8877a0eefb9e955f0e06898a1aac28dc423585c27
PS C:\Users\alcar\OneDrive\Escritorio\DAW2024-2025\deaw> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS            PORTS                  NAMES
d8613f892ea9   ubuntu:22.04   "/bin/bash"              4 seconds ago    Up 4 seconds                             c2
7c4113f7b872   ubuntu:22.04   "/bin/bash"              50 seconds ago   Up 50 seconds                            c1
ac7a1538837d   mi-web-nginx   "/docker-entrypoint.…"   About an hour ago   Up About an hour   0.0.0.0:8080->80/tcp   stoic_hawking
```

4. Pruebas de conectividad iniciales:
   - Realiza pruebas de conectividad entre `c1` y `c2`. Verifica que inicialmente no tienen visibilidad entre sí, ni por IP ni por nombre DNS.



```
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=0.261 ms
64 bytes from 172.28.0.10: icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from 172.28.0.10: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from 172.28.0.10: icmp_seq=3 ttl=64 time=0.036 ms
64 bytes from 172.28.0.10: icmp_seq=4 ttl=64 time=0.125 ms
64 bytes from 172.28.0.10: icmp_seq=5 ttl=64 time=0.046 ms
64 bytes from 172.28.0.10: icmp_seq=6 ttl=64 time=0.040 ms
^C--- host1 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.036/0.085/0.261/0.078 ms
root@host1:/# ping host2
ping: unknown host
root@host1:/#
```

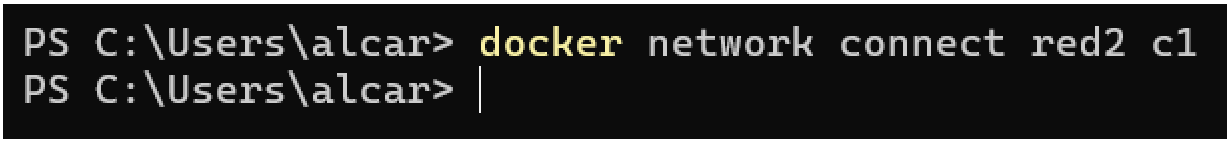
```
root@host2:/# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: icmp_seq=0 ttl=64 time=0.108 ms
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.035 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 172.18.0.2: icmp_seq=4 ttl=64 time=0.024 ms
64 bytes from 172.18.0.2: icmp_seq=5 ttl=64 time=0.035 ms
64 bytes from 172.18.0.2: icmp_seq=6 ttl=64 time=0.042 ms
^C--- 172.18.0.2 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.024/0.046/0.108/0.026 ms
root@host2:/#
```



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

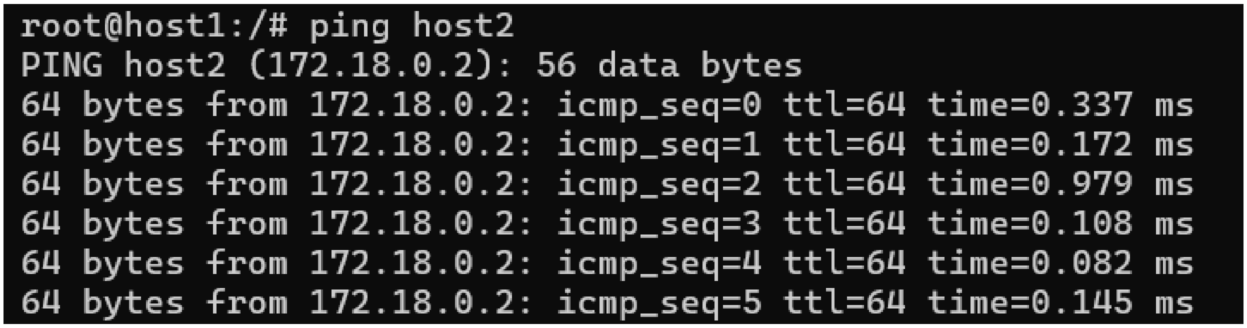
```
root@host2:/# ping host1
ping: unknown host
root@host2:/#
```

5. Conectar redes y verificar conectividad:
   - Conecta el contenedor `c1` a la red `red2` para que ambas redes tengan visibilidad. Para hacerlo sin pararlo y recrearlo usa: docker network connect red2 c1

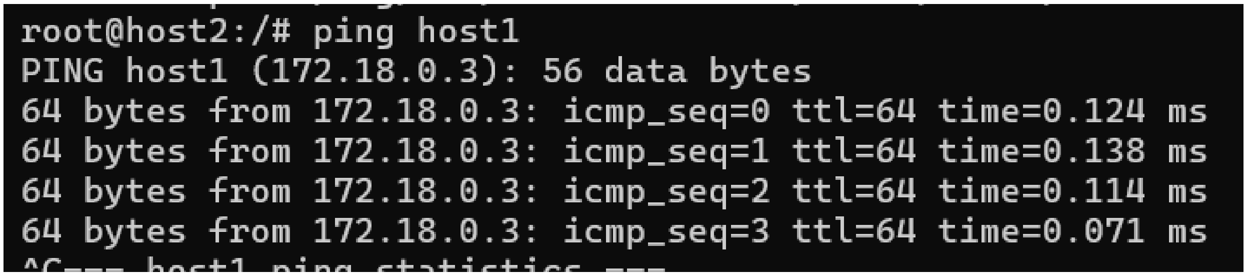


```
PS C:\Users\alcar> docker network connect red2 c1
PS C:\Users\alcar>
```

   - Comprueba nuevamente que ahora se puede hacer ping tanto por IP como por DNS entre `c1` y `c2`.



```
root@host1:/# ping host2
PING host2 (172.18.0.2): 56 data bytes
64 bytes from 172.18.0.2: icmp_seq=0 ttl=64 time=0.337 ms
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.172 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=0.979 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 172.18.0.2: icmp_seq=4 ttl=64 time=0.082 ms
64 bytes from 172.18.0.2: icmp_seq=5 ttl=64 time=0.145 ms
```



```
root@host2:/# ping host1
PING host1 (172.18.0.3): 56 data bytes
64 bytes from 172.18.0.3: icmp_seq=0 ttl=64 time=0.124 ms
64 bytes from 172.18.0.3: icmp_seq=1 ttl=64 time=0.138 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.114 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.071 ms
^C--- host1 ping statistics ---
```

6. Entrega en PDF:
    - Captura y entrega pantallazos de los siguientes pasos:
        - Configuración de red del contenedor `c1`.
        - Configuración de red del contenedor `c2`.
        - Comprobación de que no hay visibilidad entre `c1` y `c2` al inicio.
        - Verificación de la visibilidad tras conectar `c1` a `red2`.