





sde borra los servers el cliente(como su ejecucion termino se borro antes el solo) y la red prueba


**Objetivo**

Desplegar tres servidores UDP que escuchan mensajes de descubrimiento por **broadcast** dentro de una subred Docker aislada, y ejecutar un cliente que difunde a la dirección de broadcast, detecta a los servidores y se comunica con el primero que responde.

**Entorno y preparación**

* Carpeta de trabajo con los programas del ejercicio: servidor de broadcast y cliente.
* Imagen de Python basada en Debian (variante “bullseye”), para disponer de utilidades de sistema.
* Red Docker llamada **pruebas** (si ya existía, se reutilizó).

**Procedimiento realizado**

1. **Red de trabajo**  
   Se utilizó la red Docker pruebas para aislar el laboratorio y permitir la resolución de nombres entre contenedores.
2. **Lanzamiento de los servidores**  
   Se iniciaron **tres contenedores** (servidor1, servidor2 y servidor3), todos conectados a la red pruebas, montando la carpeta local como /app y ejecutando el script del servidor en primer plano del contenedor.
3. **Comprobación inicial**  
   Se verificó que los tres servidores estuvieran “Up” y asociados a la red pruebas.  
   (Captura recomendada: listado de contenedores mostrando los tres servidores activos y la red.)
4. **Obtención de la subred y cálculo de broadcast**  
   Se inspeccionó la red pruebas para conocer su subred. El sistema devolvió **172.18.0.0/16**.  
   A partir de esa máscara (/16), la dirección de broadcast correspondiente es **172.18.255.255**.  
   (Captura recomendada: inspección de la red mostrando la subred.)
5. **Ejecución del cliente**  
   Se lanzó un contenedor en la misma red pruebas con el script del cliente, indicándole como destino la **IP de broadcast 172.18.255.255** y el puerto del ejercicio. El cliente difundió su mensaje de descubrimiento y recogió las respuestas.
6. **Resultado de la detección**  
   El cliente informó en pantalla del **servidor encontrado** (mostrando su IP interna en la red pruebas y el puerto). Con ello quedó demostrado que los tres servidores recibieron el broadcast y que el cliente seleccionó al primero que respondió para enviar el mensaje “HOLA”.  
   (Captura recomendada: salida del cliente con el texto “Servidor encontrado: (IP, puerto) → SERVIDOR ACTIVO”.)
7. **Verificación final (opcional)**
   * Identificación de la IP de cada servidor dentro de la red pruebas.
   * Consulta de los registros de cada servidor para observar la recepción del mensaje de descubrimiento y, en el elegido, la petición “HOLA”.  
     (Capturas recomendadas: inspección de contenedores con sus IPs internas y, si se desea, extractos de logs.)

**Evidencias incluidas**

* Listado de contenedores con los tres servidores activos en pruebas.
* Inspección de la red mostrando la subred **172.18.0.0/16**.
* Salida del cliente indicando el servidor activo (IP interna y puerto).
* (Opcional) Inspecciones/logs de servidores donde se aprecia la recepción del broadcast.

**Conclusión**

Se ha puesto en marcha un escenario UDP con **descubrimiento por broadcast** en una red Docker aislada. El cliente difundió a la dirección de broadcast **172.18.255.255**, recibió respuesta de los servidores y estableció comunicación con el primero que contestó. El ejercicio confirma el funcionamiento del broadcast en una red Docker, la resolución por nombres dentro de la misma red y el uso de contenedores para aislar servicios y pruebas.

Principio del formulario

Pensando

Final del formulario