Redes LAN en Docker mil usos a tu alcance

Desde hace un tiempo a la mayoría de las personas que sean administradores de sistemas o formen parte de equipos de desarrollo habrá una tecnología que les sonara y en muchos casos usaran a diario facilitando mucho su trabajo, Docker, si el logo de la ballena… en otros muchos casos lo que pasará al oír este nombre es que tendrán falta de sueño, mareos y sarpullidos debido al cambio de paradigma que ha supuesto Docker en la gestión ciertos servicios.

Se trata de una tecnología bastante novedosa ya que fue liberada en 2013 pero que se ha generalizado apoyada en muchos casos en la facilidad de su uso dentro de las cloud más conocidas como Amazon Web Services, Google Cloud o Azure.

Su éxito radica en la facilidad de automatización de tareas permitiendo el despliegue de entornos varios entornos idénticos en una misma máquina que son independientes del hardware de la máquina de manera que si nos llevamos un mismo contenedor a diferentes equipos tendremos un servicio idéntico sobre el que, por ejemplo, desarrollar una aplicación, porque ha sido en este ámbito en el que mayor éxito ha tenido en el de desarrollo.

No voy a entrar en detalles sobre el funcionamiento de Docker ya que existen miles de artículos y documentos que lo detallen a la perfección, pero sí que quiero compartir un "problema" al que me he enfrentado en un proyecto que estoy llevando a cabo y que os desvelare un poco más adelante cuando esté más probado.

Ese problema era que quería ejecutar varios servicios que funcionaran de manera independiente en una misma máquina por lo que … ¡Perfecto! Docker es tu opción, pero (Siempre hay un *"pero"*) quería que cada servicio se ejecutara en una red y tuviera una IP propia asignada y que además fuera compartida por la red de mi adaptador de red. Lo vamos a ver en un esquema para verlo mejor.

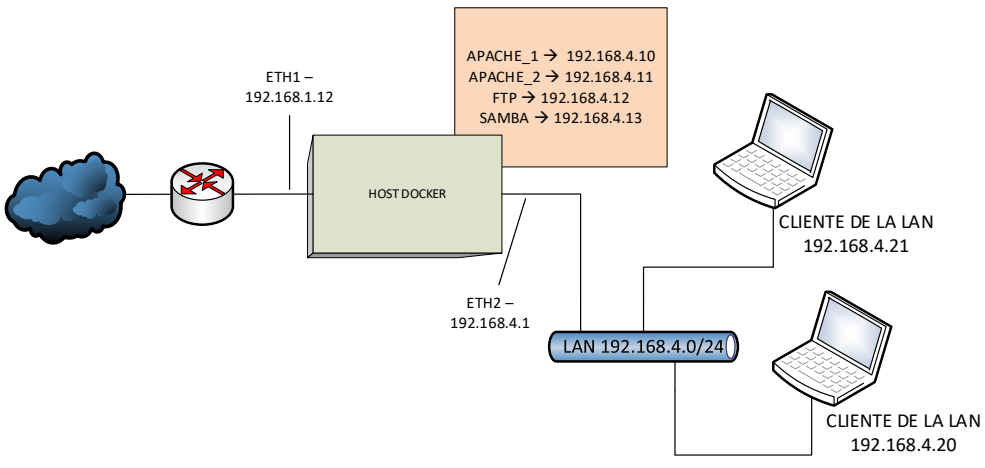

Ilustración 1 - Esquema de red que queremos

La idea ahora está más clara y es que quiero que los clientes accedan a los servicios publicados en una red, pero nos topamos con un pequeño problema en Docker y es que no podemos asignar las IP que queramos a los contenedores si no creamos una red propia además de las redes que crea Docker por defecto. En la siguiente imagen vemos ya la red creada y el listado de redes con el comando `docker network ls`

NETWORK ID	NAME	DRIVER	SCOPE
0863899058ec	bridge	bridge	local
56f1a904c619	host	host	local
0e182f56c0c0	none	null	local
2c84a355a18a	test	macvlan	local

Ilustración 2 - Listado de redes creadas

Y otro problema es que si creamos una red de tipo Bridge y le asignamos el mismo direccionamiento que el que tiene nuestra tarjeta de red perderemos la conexión con ese interface de red por lo que la solución pasa por crear una nueva red usando el driver macvlan (https://docs.docker.com/network/macvlan/).

Este tipo de driver nos permite asignar una MAC y por consiguiente una IP válida que pueda gestionar nuestro switch o router y de esta manera hacer accesibles todos los contenedores a los clientes de nuestra red con IP's independientes cada uno de ellos, lo cual nos brinda muchas posibilidades en algunos escenarios.

Para ello haremos uso del siguiente comando:

```
docker network create -d macvlan --subnet 192.168.4.0/24 --gateway 192.168.4.1 -o parent=eth1 test
```

Los parámetros son:

`-d` → El driver que queremos usar en este caso macvlan

`--subnet` → El direccionamiento de red que daremos a esta nueva red

`-- Gateway` → Puerta de enlace de nuestra red, en este caso usaremos la IP del interface de red de nuestro host que actuará como router para dar salida a internet a servicios y clientes de esta nueva red.

`-o parent=` → Interface de red a la que asociamos esta nueva red y por la que saldrán los paquetes de nuestros clientes y servicios.

Al final del todo asignamos el nombre que queramos para esta red que en este caso es "`test`".

Una vez creada la red solo nos queda lanzar nuestros contenedores y asignarles una IP propia mediante el comando:

```
docker run -dit --net test --ip 192.168.4.201 --name prueba -p 80:80 apache_basico /bin/bash
```

`--net` → Asociamos el contenedor a nuestra nueva red

`--ip` → Ip que damos a nuestro contenedor

Y ahora podemos ver como en la siguiente captura accedemos al servidor Apache desplegado desde un móvil conectado a esa misma red.

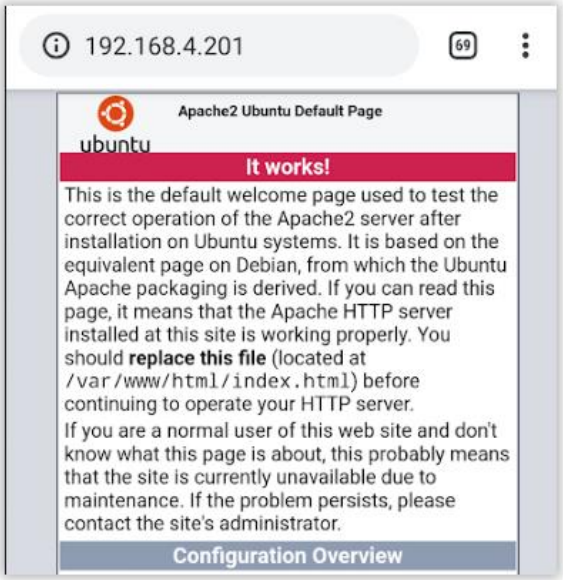

Ilustración 3 - Servicio publicado en la red

En las próximas entregas podréis ver un uso más concreto de esta característica que nos ofrece Docker y como puede dar mucho juego en determinadas situaciones.

Sobre todo, quería compartirlo con vosotros ya que habrá mucha gente como yo que no sea experta en Docker y este tipo de artículos pueda ahorrarle algún que otro quebradero de cabeza como los que he tenido.

¡Gracias una vez más por leerme!