

Docker Networking

DigitalHouse >
Coding School



**Certified Tech
Developer**
The Ultimate Degree

Índice

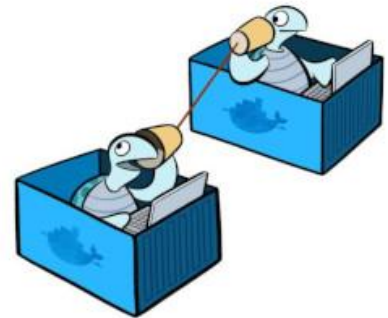
1. [Introducción a las redes de Docker](#)
2. [Bridge Driver en acción](#)
3. [Resolución de nombres](#)

1

Introducción a las redes de Docker

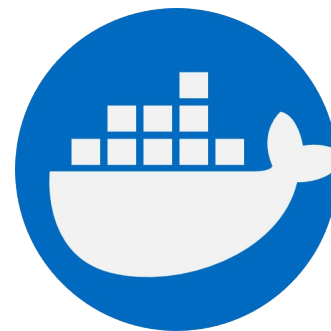
Introducción a las redes de Docker

Nos vamos a enfocar en el driver llamado bridge, ya que con el nuevo enfoque de Docker orientado a habilitar a los desarrolladores es el que mayoritariamente le darán uso en sus computadoras. Todos los otros drivers están orientados a ambientes productivos, pero ese ámbito hoy está dominado por Kubernetes, que tienen otro modelo de Networking completamente diferente.



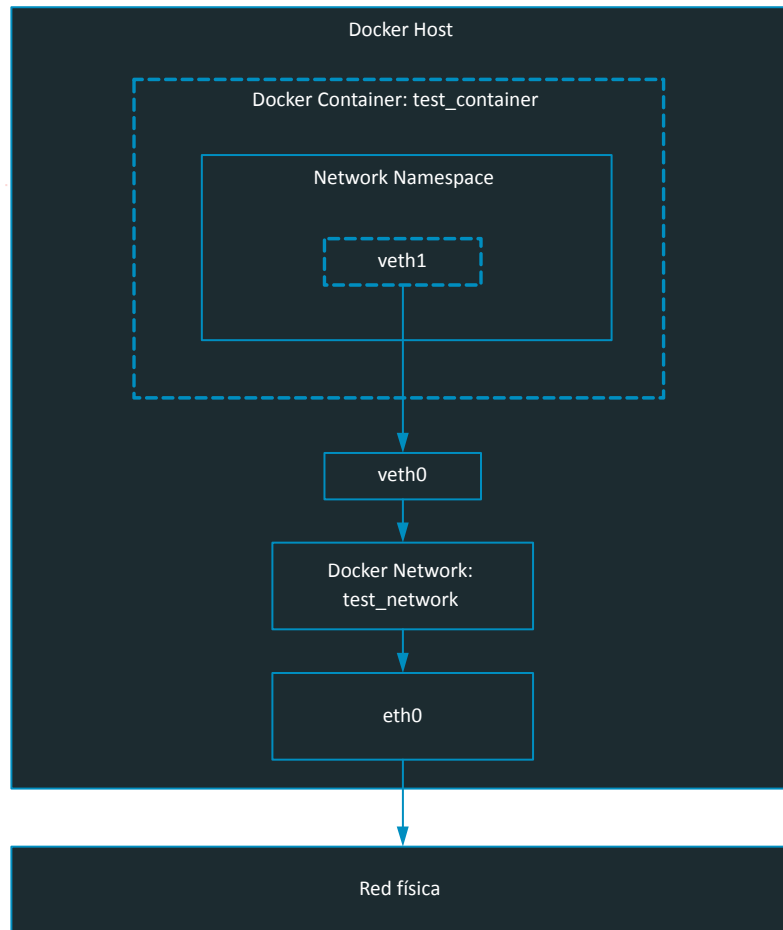
Introducción a las redes de Docker

- Las redes de tipo “bridge” son locales y exclusivas del host en donde fueron creadas.
- Son el tipo de red por defecto en Docker.
- Simula la creación de switches o hubs, de nivel 2 (en el modelo OSI). Podemos utilizar herramientas como ‘brctl’ en Linux para ver el funcionamiento interno.
- Luego de la instalación de Docker se crea por defecto una red de tipo bridge llamada “bridge”, las buenas prácticas indican que esta no debe ser utilizada y en su lugar se deben crear nuevas redes para usos específicos.



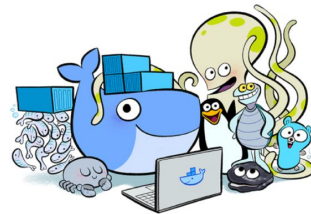
2 | Bridge Driver en acción.

Bridge Driver en acción



3 | Resolución de nombres

Resolución de nombres



Las IPs son efímeras y más aún en el mundo de los containers —estos también deberían serlo—. De modo que necesitamos una forma que le permita a un container hablar con otro sin conocer necesariamente la IP que le fue asignada.

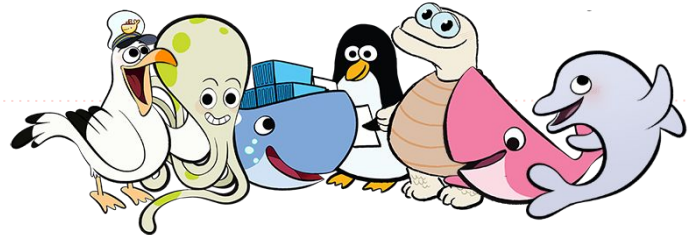
¿Qué hace Docker por nosotros?

Para todas las redes que creemos —las redes creadas por defecto no nos proveen esta ventaja—, y para todos los containers a los que se le asigne un nombre de manera deliberada —usando `--name` en `docker create`— conectados a estas, Docker nos provee con un servicio de resolución de nombres dentro del ámbito de la red misma utilizando el nombre del container para identificarlo por medio del servicio DNS.

Resolución de nombres

docker network

- **create:** nos permite crear una red.
 - **--internal:** agrando esta opción la red creada será del tipo 'privada'.
- **connect:** nos permite conectar un container existente a una red.



docker run

- **--network <networkName>:** nombre de la red a la cual conectaremos el container.
- **--name:** nombre del container, necesario para que funcione la resolución de nombres.
- **-p <hostPort>:<containerPort>:** nos permite publicar ciertos puertos de un container en el host para poder acceder al servicio dentro del container.
- **-P:** nos permite publicar todos los puertos definidos en la especificación del container (Dockerfile) en puertos aleatorios del host.

¿Dónde encontrar más información?



Documentación de Docker:

[docker network create](#)

[Networking Overview](#)

[Networking with standalone containers](#)



Blogs:

[How Docker Container Networking Works - Mimic It Using Linux Network Namespaces](#)



Libro:

[Poulton, N. \(2017\). Docker Deep Dive.](#)

DigitalHouse>
Coding School