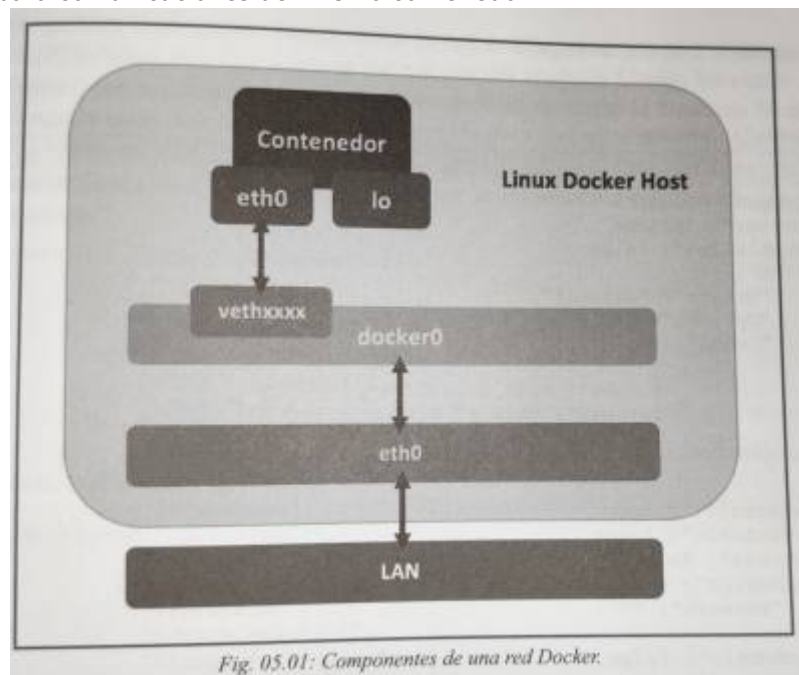


# [Docker SecDevOps] Capítulo 5 : Redes

## conceptos básicos

- net namespace: componente kernel Linux, encargado de crear cada instancia correspondiente a cada contenedor, aislado, evitando que acceda a internet u otro contenedor.
- Open vSwitch: switch programable virtual que permite multicapas, tunneling, y automatización
- IPables: firewall integrado de Linux
- Apparmor: aplicación que permite asignar unos perfiles de seguridad según la aplicación que se ejecute
- NAT
- instalación herramientas: iproute2, bridge-utils
- El demonio docker crea una interfaz de red virtual (bridge) con el nombre **docker0** y la IP **172.17.0.1/24**
- También se crean por defecto 3 redes distintas → `docker network ls`:
  - **bridge**: asociada a la interfaz **docker0**. Aquí se conectan por defecto todos los contenedores, actuando como una especie de proxy
  - **none**: se usa cuando queramos asignar el interface loopback (lo). Sin red asociada
  - **host**: asociada a la interfaz eth0 y es la que utiliza el host de Docker
- Cuando se crea un contenedor, se crean 2 interfaces asociadas:
  - **eth0**, asociada a **docker0** (y el rango de esta), para comunicación entre contenedores. Estas comunicaciones se realizan a través de una interfaz de Linux llamada *Virtual Ethernet (veth)*, que se crea con cada contenedor con el nombre **vethxxxx**
  - **lo**, 127.0.0.1. para comunicaciones del mismo contenedor



- En el proceso de creación del contenedor, se le puede especificar a `docker run` el parámetro **--net**:
  - `--net default`: interfaz **brige** por defecto usada
  - `--net=none`: sin configuración de red
  - `--netcontainer1:container2`: compartición entre los dos contenedores del *namespace*
  - `--net=host`: compartición del *namespace* con el host
  - para saber en que red está:
    - `docker network inspect bridge` (sección Containers)
      - además ofrece información del rango y el gateway (sección IPAM → Config)
    - `docker container instect <CONTAINER_ID>`
      - `-f o --format`: buscar información concreta en el resultado del comando
      - `docker inspect -f`

``range.networksettings.networksipseaddressend` <CONTAINER_ID>`

## tipos de redes

### bridge

- conecta por defecto todos los contenedores en la misma red privada.
- el contenedor está aislado del host
- solo se puede usar con un solo host
- Para que los contenedores tengan acceso a internet, hace falta usar NAT y mapeo de puertos
- Permite usar el mismo puerto para diferentes contenedores (con diferente IP dentro de la red bridge)
- Permite el acceso a través del mapeo de puertos

### host

- desaparece el aislamiento del contenedor
- comparte red con el host, igual de expuesto
- dos contenedores no pueden compartir puerto (como hacíamos en bridge)
- ejecutar aplicaciones o entornos de desarrollo con necesidad el mayor rendimiento posible de red.
- por contra, los servicios a ejecutar son más limitados.

### overlay

- entornos distribuidos con más de un host

### MacVLAN

From:  
<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:  
<https://miguelangel.torresegea.es/wiki/info:libros:docker-sec-dev-ops:cap5?rev=1638732357>

Last update: **05/12/2021 11:25**

