

# Docker Básico

---

**Arturo Silvelo**

Try New Roads

# ¿Qué es una IP?

- Una **IP** (Protocolo de Internet) es como la dirección de una casa, pero en una red.
- Permite identificar de forma única a cada dispositivo conectado a una red (ordenador, móvil, servidor, etc).
- Sin una IP, los dispositivos no podrían comunicarse entre sí.

# Ejemplo de dirección IP

- Una dirección IP típica tiene este aspecto: `192.168.1.10`
- Está formada por cuatro números separados por puntos, cada uno entre 0 y 255.
- Ejemplo visual:

Dispositivo	Dirección IP
Ordenador	192.168.1.10
Impresora	192.168.1.20
Móvil	192.168.1.30

# ¿Cómo funciona una IP?

- Cuando un dispositivo quiere comunicarse con otro, utiliza la IP de destino.
- Es como enviar una carta: necesitas la dirección del destinatario.
- En redes locales (como en casa o en una oficina), las IP suelen empezar por `192.168.x.x` o `10.x.x.x`.
- En Internet, las IP pueden ser públicas y únicas en todo el mundo.

# ¿Por qué es importante la IP en Docker?

- Docker asigna una IP a cada contenedor para que puedan comunicarse entre sí.
- Puedes ver y configurar estas IPs al crear redes personalizadas.
- Entender las IPs ayuda a diagnosticar problemas de conexión entre contenedores.

# ¿Qué es una máscara de red?

- Una **máscara de red** es un número que define qué parte de una dirección IP corresponde a la red y qué parte a los dispositivos (hosts) dentro de esa red.
- Ayuda a los dispositivos a saber si otro está en su misma red local o si debe comunicarse a través de un router.

# Ejemplo de máscara de red

- La máscara más común en redes domésticas es `255.255.255.0` (también se puede ver como `/24`).
- Ejemplo:
  - IP: `192.168.1.10`
  - Máscara: `255.255.255.0`
  - Esto significa que todos los dispositivos con IP `192.168.1.x` están en la misma red local.
- Si la máscara fuera `255.255.0.0` (`/16`), entonces todos los dispositivos con IP `192.168.x.x` estarían en la misma red.

# ¿Por qué es importante la máscara de red?

- Permite dividir redes grandes en redes más pequeñas (subredes).
- Ayuda a organizar y aislar dispositivos dentro de una red.
- En Docker, al crear redes personalizadas, puedes definir la máscara de red para controlar cuántos contenedores pueden estar en la misma red.



# Redes

## Introducción a Redes en Docker

---

- Las redes en Docker permiten la comunicación entre contenedores.
- Proporcionan aislamiento y control sobre cómo se comunican los contenedores.
- Las redes pueden persistir más allá de la vida de los contenedores.

## Tipos de Redes:

---

### bridge (predeterminada):

- Red privada para los contenedores que se ejecutan en el mismo host.
- Ideal para entornos de desarrollo donde los contenedores necesitan comunicarse entre sí.
- Los contenedores pueden acceder al exterior a través del gateway, pero están aislados de otros contenedores.

- El rango de IPs predeterminado para la red `bridge` es `172.17.0.0/16`.
- Docker automáticamente asigna una IP dentro de este rango cuando un contenedor se ejecuta en esta red.
- Modificar el rango por defecto del bridge:  
<https://docs.docker.com/engine/network/drivers/bridge>

## Comandos de Ejemplo:

```
docker network create --subnet 172.20.0.0/16 my_network
docker network create --driver bridge --subnet 172.19.0.0/16 my_network_2
docker network create my_network_3
docker network create --subnet 172.20.0.0/16 my_network_4
docker network inspect NETWORK_NAME|NETWORK_ID
```

## host:

- El contenedor comparte la red del host, eliminando la capa de aislamiento.
- Utiliza la red del sistema directamente, mejorando el rendimiento en aplicaciones que requieren baja latencia.
- Sin embargo, esto sacrifica el aislamiento entre contenedores.

## Comando de Ejemplo:

```
docker run --network host -d nginx
```

## none:

- No se asigna ninguna red al contenedor, dejándolo completamente aislado.
- Útil para pruebas de seguridad o para aplicaciones que no necesitan comunicación de red.

## Comando de Ejemplo:

```
docker run --network none -d busybox top docker exec -it container_id ping google.com
```

## Comparación de Tipos de Redes en Docker

Características	Aislamiento	Acceso a la Red Externa	Usos Comunes
<b>bridge</b>	Aislado entre contenedores y host.	A través de NAT y gateway.	Desarrollo local, múltiples contenedores en un mismo host.
<b>host</b>	No hay aislamiento, comparte la red del host.	Directo, sin NAT.	Contenedores de alto rendimiento, aplicaciones que requieren baja latencia.
<b>none</b>	Totalmente aislado, sin acceso a la red.	No tiene acceso a la red.	Pruebas de seguridad, contenedores que no necesitan conectividad.