

Contenedores de red

Tiempo estimado de lectura: 6 minutos

Si está trabajando a través de la guía del usuario, acaba de construir y ejecutar una aplicación sencilla. Usted también ha construido en sus propias imágenes. Esta sección le enseña cómo conectar en red sus contenedores.

Iniciar un contenedor en la red predeterminada

Docker incluye soporte para los contenedores de red mediante el uso de **controladores de red**. De forma predeterminada, Docker proporciona dos controladores de red para usted, el **bridge** y los **overlay** controladores. También puede escribir un controlador de red plugin para que pueda crear sus propios controladores, pero que es una tarea avanzada.

Cada instalación del Docker Engine incluye automáticamente tres redes predeterminadas. Puede listarlos:

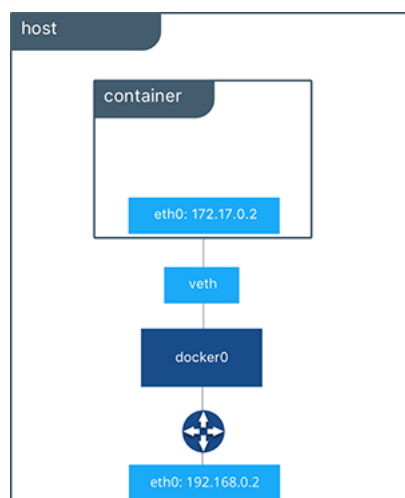
```
$ docker network ls
```

NETWORK ID	NAME	DRIVER
18a2866682b8	none	null
c288470c46f6	host	host
7b369448dccb	bridge	bridge

La red denominada **bridge** es una red especial. A menos que le diga lo contrario, Docker siempre lanza sus contenedores en esta red. Prueba esto ahora:

```
$ docker run -itd --name=networktest ubuntu
```

```
74695c9cea6d9810718fddadc01a727a5dd3ce6a69d09752239736c030599741
```



La inspección de la red es una manera fácil de averiguar la dirección IP del contenedor.

```
$ docker network inspect bridge

[
  {
    "Name": "bridge",
    "Id": "f7ab26d71dbd6f557852c7156ae0574bbf62c42f539b50c8ebde0f728a253b6f",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.1/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Containers": {
      "3386a527aa08b37ea9232cbcace2d2458d49f44bb05a6b775fba7ddd40d8f92c": {
        "Name": "networktest",
        "EndpointID": "647c12443e91faf0fd508b6edfe59c30b642abb60dfab890b4bdccee38750bc1",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "9001"
    },
    "Labels": {}
  }
]
```

Puede quitar un contenedor de una red desconectando el contenedor. Para ello, suministra el nombre de red y el nombre de contenedor. También puede utilizar el ID de contenedor. En este ejemplo, sin embargo, el nombre es más rápido.

```
$ docker network disconnect bridge networktest
```

Mientras puede desconectar un contenedor de una red, no puede quitar la `bridge` red integrada denominada `bridge`. Las redes son maneras naturales de aislar contenedores de otros contenedores u otras redes. Por lo tanto, a medida que obtenga más experiencia con Docker, querrá crear sus propias redes.

Crea tu propia red de puentes

Docker Engine soporta nativamente tanto las redes de puente como las redes de superposición. Una red de puente está limitada a un solo host que ejecute Docker Engine. Una red de superposición puede incluir varios hosts y es un tema más avanzado. En este ejemplo, creará una red de puente:

```
$ docker network create -d bridge my_bridge
```

El `-d` indicador le indica a Docker que utilice el `bridge` controlador para la nueva red. Usted podría haber dejado esta bandera apagado como `bridge` es el valor por defecto para este indicador. Sigue adelante y lista las redes en tu máquina:

```
$ docker network ls
```

NETWORK ID	NAME	DRIVER
7b369448dccb	bridge	bridge
615d565d498c	my_bridge	bridge
18a2866682b8	none	null
c288470c46f6	host	host

Si inspecciona la red, encontrará que no tiene nada en ella.

```
$ docker network inspect my_bridge

[
  {
    "Name": "my_bridge",
    "Id": "5a8afc6364bccb199540e133e63adb76a557906dd9ff82b94183fc48c40857ac",
    "Scope": "local",
    "Driver": "bridge",
    "IPAM": {
      "Driver": "default",
      "Config": [
        {
          "Subnet": "10.0.0.0/24",
          "Gateway": "10.0.0.1"
        }
      ]
    },
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
```

Agregar contenedores a una red

Para crear aplicaciones web que actúen en concierto, pero que lo hagan de forma segura, cree una red. Las redes, por definición, proporcionan un aislamiento completo de los contenedores. Puede agregar contenedores a una red cuando ejecuta por primera vez un contenedor.

Inicie un contenedor que ejecute una base de datos PostgreSQL y pase el `--net=my_bridge` indicador para conectarlo a su nueva red:

```
$ docker run -d --net=my_bridge --name db training/postgres
```

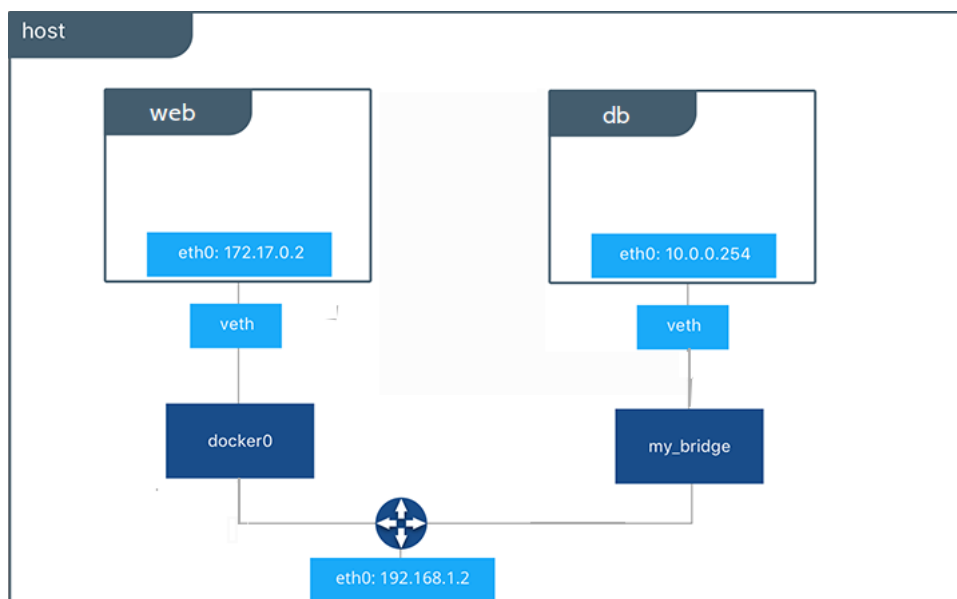
Si inspecciona su `my_bridge` verá que tiene un contenedor adjunto. También puede inspeccionar su contenedor para ver dónde está conectado:

```
$ docker inspect --format='{{json .NetworkSettings.Networks}}' db

{"my_bridge":{"NetworkID":"7d86d31b1478e7cca9ebd7e73aa0fdeec46c5ca29497431d3007d2d9e15ed99",
"EndpointID":"508b170d56b2ac9e4ef86694b0a76a22dd3df1983404f7321da5649645bf7043", "Gateway":"10.0.0.1", "IPAddress":"10.0.0.254",'
```

Ahora, vaya adelante y comience su por ahora familiar aplicación web. Esta vez no especifica una red.

```
$ docker run -d --name web training/webapp python app.py
```



¿En qué red se está `web` ejecutando su aplicación? Inspeccione la aplicación y verá que se está ejecutando en la `bridge` red predeterminada .

```
$ docker inspect --format='{{json .NetworkSettings.Networks}}' web

{"bridge":{"NetworkID":"7ea29fc1412292a2d7bba362f9253545fecdfa8ce9a6e37dd10ba8bee7129812",
"EndpointID":"508b170d56b2ac9e4ef86694b0a76a22dd3df1983404f7321da5649645bf7043", "Gateway":"172.17.0.1", "IPAddress":"10.0.0.2",'
```

A continuación, obtenga la dirección IP de su `web`

```
$ docker inspect --format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' web

172.17.0.2
```

Ahora, abra un shell a su `db` contenedor en ejecución :

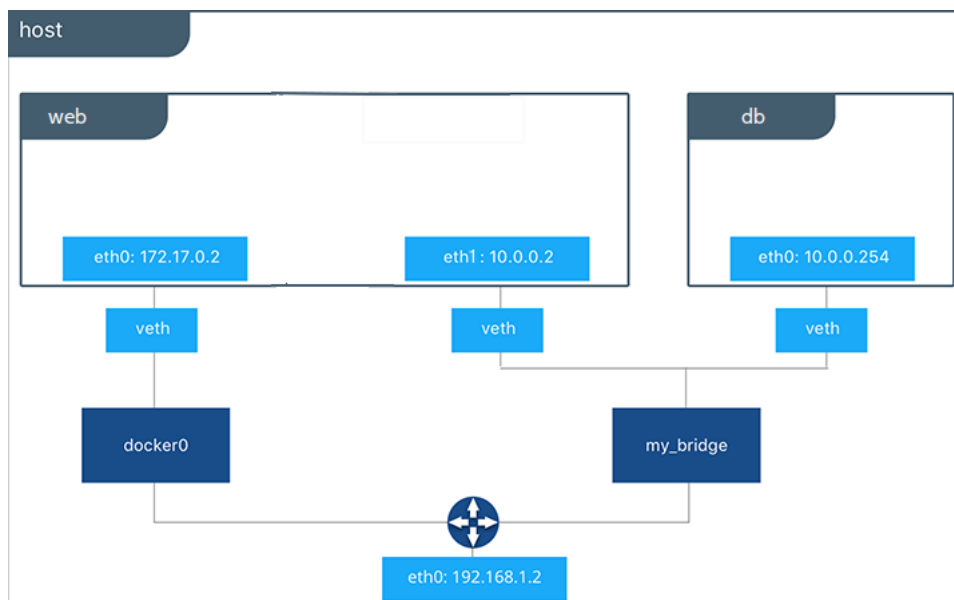
```
$ docker exec -it db bash

root@a205f0dd33b2:/# ping 172.17.0.2
ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
^C
--- 172.17.0.2 ping statistics ---
44 packets transmitted, 0 received, 100% packet loss, time 43185ms
```

Después de un poco, utilice `CTRL-C` para terminar el `ping` y usted encontrará que el ping falló. Esto se debe a que los dos contenedores se ejecutan en redes diferentes. Puedes arreglar eso. A continuación, utilice el `exit` comando para cerrar el contenedor.

La red Docker le permite conectar un contenedor a tantas redes como desee. También puede adjuntar un contenedor ya en ejecución. Adelante y adjunte su `web` aplicación en ejecución a la `my_bridge` .

```
$ docker network connect my_bridge web
```



Abra una shell en la `db` aplicación de nuevo y pruebe el comando ping. Esta vez simplemente use el nombre del contenedor en `web` lugar de la dirección IP.

```
$ docker exec -it db bash

root@a205f0dd33b2:/# ping web
PING web (10.0.0.2) 56(84) bytes of data.
64 bytes from web (10.0.0.2): icmp_seq=1 ttl=64 time=0.095 ms
64 bytes from web (10.0.0.2): icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from web (10.0.0.2): icmp_seq=3 ttl=64 time=0.066 ms
^C
--- web ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.060/0.073/0.095/0.018 ms
```

La `ping` muestra está poniéndose en contacto con una dirección IP diferente, la dirección en la `my_bridge` que es diferente de su dirección en la `bridge` red.

Próximos pasos

Ahora que ya sabes cómo conectar los contenedores, consulta cómo administrar los datos en contenedores (<https://docs.docker.com/engine/tutorials/dockervolumes/>) .

🔗 Ejemplos (<https://docs.docker.com/glossary/?term=Examples>) , Uso (<https://docs.docker.com/glossary/?term=Usage>) , volumen (<https://docs.docker.com/glossary/?term=volume>) , docker (<https://docs.docker.com/glossary/?term=docker>) , documentación (<https://docs.docker.com/glossary/?term=documentation>) , guía del usuario (<https://docs.docker.com/glossary/?term=user%20guide>) , datos (<https://docs.docker.com/glossary/?term=data>) , volúmenes (<https://docs.docker.com/glossary/?term=volumes>)

Califica esta página:

89 13

¿Qué es Docker? (<https://www.docker.com/what-docker>)

¿Qué es un contenedor? (<https://www.docker.com/what-container>)

Casos de Uso (<https://www.docker.com/use-cases>)

Clientes (<https://www.docker.com/customers>)

Fogonadura (<https://www.docker.com/partners/partner-program>)

Para el gobierno (<https://www.docker.com/industry-government>)

Acerca de Docker (<https://www.docker.com/company>)

administración (<https://www.docker.com/company/management>)

Prensa y Noticias (<https://www.docker.com/company/news-and-press>)

Empleo (<https://www.docker.com/careers>)

Producto (<https://www.docker.com/products/overview>)

Precio (<https://www.docker.com/pricing>)

Edición de comunidad (<https://www.docker.com/docker-community>)

Edición de Empresa (<https://www.docker.com/enterprise>)

Docker Datacenter (<https://www.docker.com/products/docker-datacenter>)

Docker Cloud (<https://cloud.docker.com/>)

Tienda Docker (<https://store.docker.com/>)

Docker para Mac (<https://www.docker.com/docker-mac>)

Docker para Windows (PC) (<https://www.docker.com/docker-windows>)

Docker para AWS (<https://www.docker.com/docker-aws>)

Docker para Azure (<https://www.docker.com/docker-microsoft-azure>)

Docker para Windows Server (<https://www.docker.com/docker-windows-server>)

Docker para distribución CentOS (<https://www.docker.com/docker-centos>)

Docker para Debian (<https://www.docker.com/docker-debian>)

Docker para Fedora® (<https://www.docker.com/docker-fedora>)

Docker para Oracle Enterprise Linux (<https://www.docker.com/docker-oracle-linux>)

Docker para RHEL (<https://www.docker.com/docker-rhel>)

Docker para SLES (<https://www.docker.com/docker-sles>)

Docker para Ubuntu (<https://www.docker.com//docker-ubuntu>)

Documentación (/)

Aprender (<https://www.docker.com/docker>)

Blog (<https://blog.docker.com>)

Formación (<https://training.docker.com/>)

Apoyo (<https://www.docker.com/docker-support-services>)

Base de conocimientos (<https://success.docker.com/kbase>)

Recursos (<https://www.docker.com/products/resources>)

Comunidad (<https://www.docker.com/docker-community>)

Fuente abierta (<https://www.docker.com/technologies/overview>)

Eventos (<https://www.docker.com/community/events>)

Foros (<https://forums.docker.com/>)

Capitanes del muelle (<https://www.docker.com/community/docker-captains>)

Becas (<https://www.docker.com/docker-community/scholarships>)

Noticias de la comunidad (<https://blog.docker.com/curated/>)

Estado (<http://status.docker.com/>) Seguridad (<https://www.docker.com/docker-security>) Legal (<https://www.docker.com/legal>)

Contacto (<https://www.docker.com/company/contact>)

Copyright © 2017 Docker Inc. Todos los derechos reservados.



