



链滴

# docker 容器之间互联

作者: [yuanhenglizhen](#)

原文链接: <https://ld246.com/article/1546703605353>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

起源：国庆断电之后，公司的环境挂掉了一部分，基本是因为容器之间不能访问导致的，还有就是服务器性能一般，多个容器向一个容器请求配置文件，导致堵塞卡住。还有一小部分原因，在于刚开始起器的時候没有加自起。

启动一个doc容器

```
docker run -ti -p 4000:4000 docs/docker.github.io:latest
```

关于docker容器的网络驱动，官网介绍有以下几个：

\* **bridge**：默认网络驱动程序。如果未指定驱动程序，则这是您要创建的网络类型。当您的应用程序在需要通信的独立容器中运行时，通常会使用桥接网络。见 [桥接网络](https://docs.docker.com/network/bridge/)。

\* **host**：对于独立容器，删除容器和Docker主机之间的网络隔离，并直接使用主机的网络。host 仅用于Docker 17.06及更高版本上的群集服务。请参阅 [使用主机网络](https://docs.docker.com/network/host/)。

\* **overlay**：覆盖网络将多个Docker守护程序连接在一起，并使群集服务能够相互通信。您还可以使覆盖网络来促进群集服务和独立容器之间的通信，或者在不同Docker守护程序上的两个独立容器之间进行通信。此策略无需在这些容器之间执行OS级别的路由。请参阅[覆盖网络](https://docs.docker.com/network/overlay/)。

\* **macvlan**：Macvlan网络允许您为容器分配MAC地址，使其显示为网络上的物理设备。Docker守护程序通过其MAC地址将流量路由到容器。macvlan 在处理期望直接连接到物理网络的传统应用程序，使用驱动程序有时是最佳选择，而不是通过Docker主机的网络堆栈进行路由。见 [Macvlan网络](https://docs.docker.com/network/macvlan/)。

\* **none**：对于此容器，禁用所有网络。通常与自定义网络驱动程序一起使用。none不适用于群组服务。请参阅 [禁用容器网络](https://docs.docker.com/network/none/)。

\* [网络插件](https://docs.docker.com/engine/extend/plugins\_services/)：您可以使用Docker装和使用第三方网络插件。这些插件可从 [Docker Hub](https://hub.docker.com/search?category=network&q=&type=plugin) 或第三方供应商处获得。有关安装和使用给定网络插件的信息，请参阅供应商的文档。

如何选择合适的驱动

\* 当您需要多个容器在同一个Docker主机上进行通信时，用户定义的桥接网络是最佳选择。

\* 当网络堆栈不应与Docker主机隔离时，主机网络最佳，但您希望隔离容器的其他方面。

\* 当您需要不同Docker主机上运行的容器进行通信时，或者当多个应用程序使用swarm服务协同工作时，覆盖网络是最佳选择。

\* 当您从VM设置迁移或需要您的容器看起来像网络上的物理主机（每个都具有唯一的MAC地址）时Macvlan网络是最佳的。

\* 第三方网络插件允许您将Docker与专用网络堆栈集成。

环境需要 容器访问以主机ip+port访问其他的容器的服务，这边撇开合不合理说，因为一般都会用link连接容器等，这边有这个需求

一般来说我们使用桥接的方式，应该没有这些问题，可能是因为重启之后没有执行一下命令导致的，在没想明白为啥，包转发打开的

1. 配置Linux内核以允许IP转发。

```
$ sysctl net.ipv4.conf.all.forwarding=1
```

2. 将策略的iptablesFORWARD策略更改DROP为 ACCEPT。

```
$ sudo iptables -P FORWARD ACCEPT
```

这些设置在重新启动时不会持续存在，因此您可能需要将它们添加到启动脚本中。

忽然想到一点,应该以下方式也可以

db

```
docker run -d -p 3306:3306 --name=db mysql
```

Web

```
docker run -d -p 80:80 --link db:db web
```

总结下，关于网络这边还是需要好好研究下，不然每次遇到容器互联，要么关掉防火墙，要么重启docker的方式太原始以及low，还是应该合理的去操作，既然有些知识没有掌握，还是应该看官方文档，人写的博客都是别人理解的总结，不一定适用，就算适用了，下次也不一定记得住。

环境的问题应该是，iptables的FORWARD策略没修改导致的

没文化多读书的这个道理，没有骗人啊 哈哈

参考文档：

<https://docs.docker.com/network/#network-drivers>

<https://docs.docker.com/network/iptables/#add-iptables-policies-before-dockers-rules>

<https://success.docker.com/article/networking>

<https://docs.docker.com/network/host/>

<https://docs.docker.com/network/bridge/#manage-a-user-defined-bridge>