

Docker 网络使用指南

[前言](#)

[Docker网络类型](#)

[Docker 网络概念](#)

[使用 Docker 网络](#)

[运行容器并连接到网络](#)

[检查网络](#)

[清理网络资源](#)

[Docker Compose网络](#)

前言

Docker 是一种容器化平台，它允许你将应用程序和它们的依赖项打包成一个称为容器的独立单元。Docker 提供了一种轻量级、可移植和自包含的方式来部署应用程序。在 Docker 中，网络是一个关键的概念，因为容器可能需要与其他容器或外部网络进行通信。

Docker网络类型

bridge（桥接）： 这是默认的网络驱动程序。它用于在同一主机上运行的容器之间创建网络连接。每个容器都分配了一个唯一的 IP 地址，并且它们可以通过这个网络进行通信。

host（主机）： 使用主机网络驱动程序，容器与主机共享网络命名空间。这意味着容器使用主机的网络栈，与主机共享 IP 地址。这种模式对于需要最大化网络性能的情况可能很有用。

overlay（覆盖）： Overlay 驱动程序允许在多个 Docker 守护程序之间创建网络，以便容器可以跨主机进行通信。这对于构建分布式应用程序和服务很有用。

macvlan（MACVLAN）： MACVLAN 驱动程序允许为容器分配一个与物理网络接口卡（NIC）上的真实硬件设备相对应的 MAC 地址。这使得容器看起来像是物理设备上的一个真实节点。

none（无）： 使用 none 驱动程序将容器连接到一个没有网络接口的网络，即容器内部没有网络连接。这对于某些特殊用途的容器很有用。

Docker 网络概念

在 Docker 中，有多种网络驱动程序，包括 bridge、host、overlay 等。每个网络驱动程序都有其特定的用途。熟悉这些概念对于有效使用 Docker 网络是很重要的。你可以通过运行以下命令来查看可用的网络驱动程序：



Plain Text |

```
1 docker network ls
```

使用 Docker 网络

创建自定义网络



Plain Text |

```
1 docker network create one_network
```

network create，用于创建网络，one_network，新创建的网络指定的名称。

运行容器并连接到网络

通过 **docker run** 命令，我们可以启动一个容器并将其连接到指定的网络。



Plain Text |

```
1 docker run -p 8080:80 --name container --network one_network -d image
```

-p 端口映射。

--name container 容器指定了一个名称，即"container"。

--network my_network 容器连接到名为"one_network"的网络，能够与其他连接到同一网络的容器进行通信。

-d 这个选项表示在后台运行容器。

image 这是容器中运行的镜像的名。

检查网络

用于检查指定 Docker 网络的详细信息

```
1 docker network inspect my_network
```

清理网络资源

当不再需要网络时，可以使用 `docker network rm` 命令删除它

```
1 docker network rm my_network
```

Docker Compose 网络

当在 Docker Compose 文件中使用多个服务时，你通常希望这些服务能够相互通信，并且最好能够隔离它们，使得它们不直接暴露给主机或外部网络。这时候，你可以使用 Docker Compose 中的网络功能。

以下是一个 Docker Compose 示例，其中包含 MySQL 服务和一个简单的 Web 服务（使用 Nginx）。它还定义了一个名为 `my_network` 的自定义网络，用于连接这两个服务：

```
1  version: '3'
2  services:
3    mysql:
4      image: mysql:8.0
5      environment:
6        MYSQL_ROOT_PASSWORD: root_password
7        MYSQL_DATABASE: my_database
8        MYSQL_USER: my_user
9        MYSQL_PASSWORD: my_password
10     networks:
11       - my_network
12     volumes:
13       - ./data:/var/lib/mysql
14
15     web:
16       image: nginx
17       networks:
18         - my_network
19       ports:
20         - "8080:80"
21
22     networks:
23       my_network:
24         driver: bridge
```

mysql 和 web 服务中，都添加了 networks 配置项。它们都连接到一个名为 my_network 的自定义网络。

networks 定义：在文档底部，定义了 my_network 网络，指定了它使用桥接网络驱动。

运行以下命令启动服务：

```
1  docker-compose up
```