1. Docker网络通信
2. 概述

Docker网络通信主要解决三个问题，容器与容器之间的通信，容器访问外部网络，外部网络访问容器。

1. 容器之间的通信

首先宿主机本身有一个网卡，在安装完Docker之后，会出现一个docker0，可以将它理解为一个网桥交换机，用于交换不同容器之间的数据，同时创建的还有不同的网络命名空间（Network Namespace），不同的命名空间是隔离的，且这个命名空间有两端，一端固定在容器的内部成为虚拟网卡， 另一端在宿主机中生成一个vethxxxxx，每多一个容器就会多一个vethxxxx。该交换机实现了容器之间的通信与隔离以及容器与宿主机之间的通信问题。

1. 容器与外部网络的通信
2. 容器访问外部网络

使用SNAT规则访问外部网络，如下所示：

iptables -t nat -A POSTROUTING -s 172.17.0.0/16 -o docker0 -j MASQUERADE

1. 外部网络访问容器

当我们启动如下的docker命令时，映射端口给宿主机，就会出现两条DNAT规则

docker run -d -p 80:80 apache

iptables -t nat -A PREROUTING -m addrtype -dst-type LOCAL -j DOCKER

iptables -t nat -A DOCKER ! -I docker0 -p tcp -m tcp -dport 80 -j DNAT -to-destination 172.17.0.2:80

1. namespace类型

前面使用Network Namespace实现了容器之间网络的隔离，其实namespace还有很多类型，可以实现多种资源的隔离：

1. UTS

隔离主机名和域名。

1. IPC

隔离信号量，消息队列和共享内存。

1. PID

隔离进程编号。

1. NetWork

隔离网络设备，网络栈，端口。

1. Mount

隔离挂载点（文件系统）。

1. User

隔离用户和用户组（内核版本要求3.8）。

1. Docker网络模式的修改
2. Docker进程网络修改

Docker进程网络修改相当于Docker主程序的修改，后面的容器都会相应地进行修改。

1. -b,--bridge = “”

指定Docker使用的网桥设备，由于默认Docker会自动创建和使用docker0网桥设备，通过此参数可以使用已经存在的网桥设备。

1. --bip

指定Docker0的IP和掩码，使用标准的CIDR形式，如10.10.10.10/24，这个用于防止容器的IP冲突问题。

1. --dns

配置容器的DNS服务，在启动Docker进程时添加，所有容器都会生效。

1. 容器网络修改
2. --dns

指定启动的容器的DNS

1. --net

指定容器的网络通讯方式，有以下四个值：

bridge: Docker 默认方式，网桥模式，隔离化的容器间数据通信；

none: 容器没有网络栈，只有回环网卡，即容器没有网络，用于特殊容器，如专门做计算的容器不需要网络，提高容器安全性；

container: 使用其他容器的网络栈，Docker容器会加入其他容器的network namespace中，即直接与其他容器共用网络；

host: 容器使用Host的网络，没有自己独立的网络栈，容器可以完全访问Host的网络，即容器将自己完全暴露给Host，Host可以直接访问容器中的服务，非常危险；

1. 网桥模式下暴露端口
2. -p :容器端口

将指定的容器端口映射到主机所有地址的一个动态端口；

1. -p 主机端口:容器端口

将指定容器端口映射到指定的主机端口；

1. -p 主机IP::容器端口

将指定容器端口映射到指定主机的动态端口；

1. -p 主机IP:主机端口:容器端口

将指定容器端口映射到指定主机的指定端口；

1. -P（大写）

暴露所需要的所有端口给主机；

1. 查看容器当前映射关系

使用如下命令：

docker port ContainerName

1. 修改docker0网桥的配置

在/etc/docker/daemon.json文件中存储的就是docker0网桥的配置，我们可以对它进行修改。

1. 常见隔离方式
2. 基础命令

查看当前可用的网络空间

docker network ls

docker network create -d 类型 网络空间名称

类型有

overlay network: 使不同主机的容器实现通信；

bridge network: 网桥通信

如：

docker network create -d bridge lamp

创建了一个名为lamp的网桥命名空间

1. 通过网络命名空间隔离容器

要使用该网络命名空间，可在运行容器时设置 --network=lamp即可，表示使用lamp这个网络命名空间。如果我们要让两个容器之间相互隔离，就让它们运行在不同的网络命名空间中即可。