Docker 网络模式总结

Bridge container　桥接式网络模式
Host(open) container　开放式网络模式
Container(join) container　联合挂载式网络模式，是 host 网络模式的延伸
None(Close) container　封闭式网络模式
Overlay 网络模式

Bridge container　桥接式网络模式

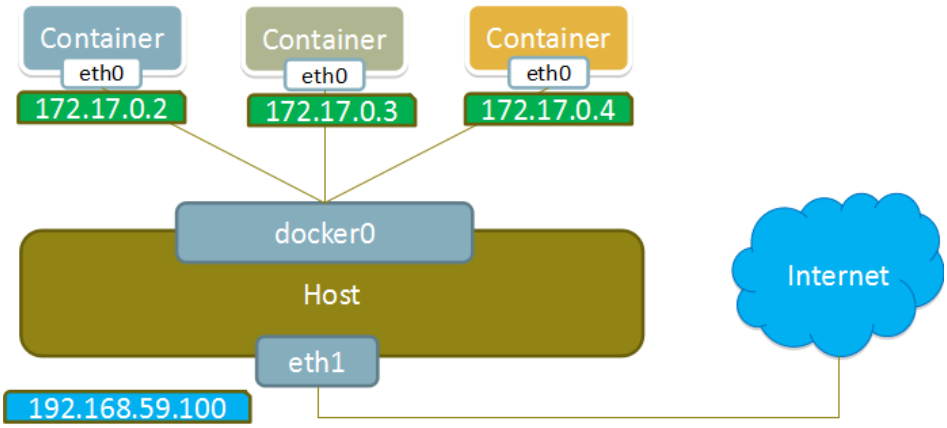
当 Docker 进程启动时，会在主机上创建一个名为 docker0 的虚拟网桥，此主机上启动的 Docker 容器会连接到这个虚拟网桥上，所以有默认地址 172.17.0.0/16 的地址。虚拟网桥的工作方式和物理交换机类似，这样主机上的所有容器就通过交换机连在了一个二层网络中。

从 docker0 子网中分配一个 IP 给容器使用，并设置 docker0 的 IP 地址为容器的默认网关。在主机上创建一对虚拟网卡 veth pair 设备，Docker 将 veth pair 设备的一端放在新创建的容器中，并命名为 eth0（容器的网卡），另一端放在主机中，以 vethxxx 这样类似的名字命名，并将这个网络设备加入到 docker0 网桥中。可以通过 brctl show 命令查看。

```
[root@along ~]# brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.024241c45d6e       no
```

bridge 模式是 docker 的默认网络模式，不写--net 参数，就是 bridge 模式。使用 docker run -p 时，docker 实际是在 iptables 做了 DNAT 规则，实现端口转发功能。可以使用 iptables -t nat -vnL 查看。

```
[root@along ~]# iptables -t nat -vnL
Chain POSTROUTING (policy ACCEPT 20 packets, 1238 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      !docker0  172.17.0.0/16        0.0.0.0/0
```



```
8c166c98adbf
root@ubuntu:~# brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.024214260fc4       no
virbr0          8000.52540096f4fa       yes             virbr0-nic
root@ubuntu:~#
```

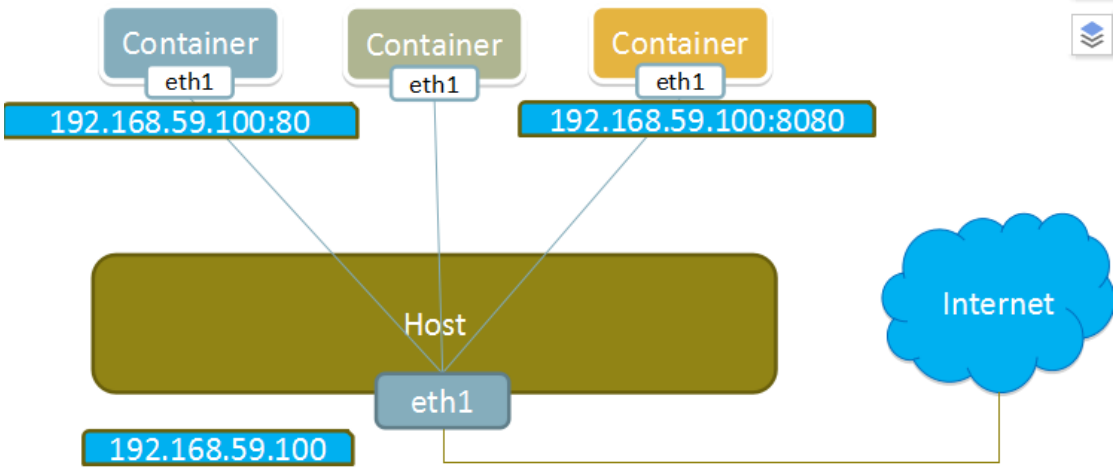
开始 docker0 上没有任何网络设备，

```
root@ubuntu:~#
root@ubuntu:~# docker run -d httpd
2b668e52480e493beffc35e8bef8ca6dac0241afc8a332217511995f0c383f38
root@ubuntu:~#
root@ubuntu:~# brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.024214260fc4       no              veth28c57df
virbr0          8000.52540096f4fa       yes             virbr0-nic
root@ubuntu:~#
```

创建一个容器之后一个新的网络接口被挂载到了 docker0 上，这个就是容器创建时创建的虚拟网卡。bridge 模式为容器创建独立的网络栈，保证容器内的进程使用独立的网络环境，使容器之间，容器和 docker host 之间实现网络隔离。

Host(open) container　开放式网络模式

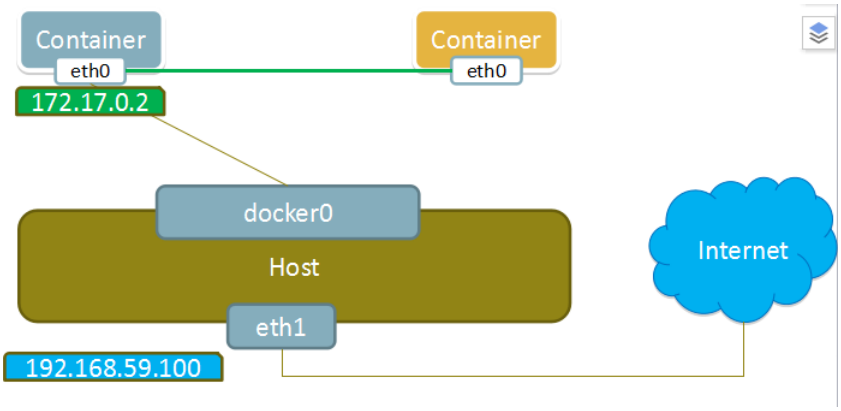
如果启动容器的时候使用 host 模式，那么这个容器将不会获得一个独立的 Network Namespace，而是**和宿主机共用一个 Network Namespace**。容器将不会虚拟出自己的网卡，配置自己的 IP 等，而是使用宿主机的 IP 和端口。但是，容器的其他方面，如文件系统、进程列表等还是和宿主机隔离的



```
root@ubuntu:~#
root@ubuntu:~# docker run -it --network=host busybox
/ #
/ # ip l
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:5f:79:3f brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:9c:21:5e brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
    link/ether 02:42:14:26:0f:c4 brd ff:ff:ff:ff:ff:ff
7: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue qlen 1000
    link/ether 52:54:00:96:f4:fa brd ff:ff:ff:ff:ff:ff
8: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 qlen 1000
    link/ether 52:54:00:96:f4:fa brd ff:ff:ff:ff:ff:ff
/ #
/ # hostname
ubuntu
/ #
```

Container(join) container　联合挂载式网络模式，是 host 网络模式的延伸

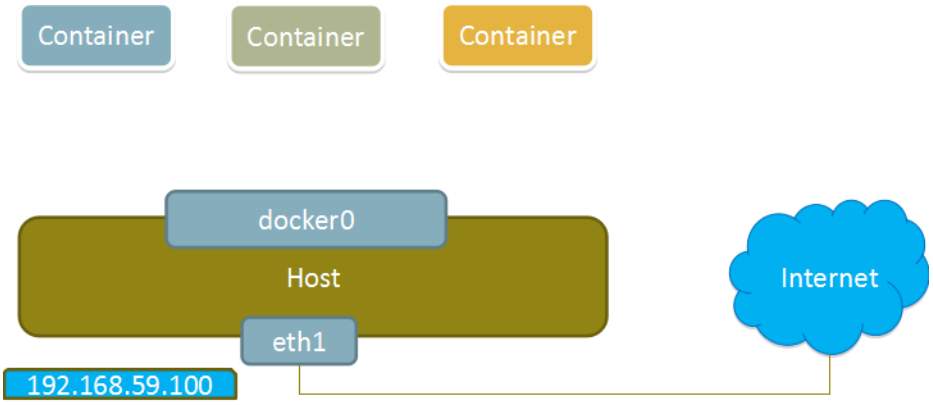
这个模式指定新创建的容器和已经存在的一个容器共享一个 Network Namespace，而不是和宿主机共享。新创建的容器不会创建自己的网卡，配置自己的 IP，而是和一个指定的容器共享 IP、端口范围等。同样，两个容器除了网络方面，其他的如文件系统、进程列表等还是隔离的。两个容器的进程可以通过 lo 网卡设备通信。



None(Close) container　封闭式网络模式

使用 none 模式，**Docker 容器拥有自己的 Network Namespace，但是，并不为 Docker 容器进行任何网络配置**。也就是说，这个 Docker 容器没有网卡、IP、路由等信息，只有 lo 网络接口。需要我们自己为 Docker 容器添加网卡、配置 IP 等。

不参与网络通信，运行于此类容器中的进程仅能访问本地回环接口；仅适用于进程无须网络通信的场景中，例如：备份、进程诊断及各种离线任务等。




```
root@ubuntu:~# docker run -it --network=none busybox
/ #
/ # ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

/ #
```


Overlay 网络模式

容器在**两个跨主机进行通信**的时候，是使用 overlay network 这个网络模式进行通信，如果使用 host 也可以实现跨主机进行通信，直接使用这个物理的 ip 地址就可以进行通信。overlay 它会虚拟出一个网络比如 10.0.9.3 这个 ip 地址，在这个 overlay 网络模式里面，有一个类似于服务网关的地址，然后把这个包转发到物理服务器这个地址，最终通过路由和交换，到达另一个服务器的 ip 地址。