

Docker网络模式与命令汇总

文档说明：

- OS与Docker版本：**CentOS 7.2、Docker 1.13.1**

Docker网络模式汇总：

- Docker网络模式包括5种：man docker-run

1. **bridge**模式：

- a. 容器可使用默认docker0网桥或创建自定义的Docker网桥
- b. 容器使用veth对与Docker网桥相连
- c. 在同一Docker宿主机上的容器可通过bridge模式相互通信

2. **none**模式：

- 容器不使用任何网络，实现网络栈的完全隔离。

3. **container**模式：

- a. 容器与其他容器共享相同的网络栈，使用相同的虚拟网络接口与IP地

址。

- b. **Kubernetes**使用**pause-amd64**容器连接各容器的网络栈，使同一Pod中

的所有

容器共享相同的网络栈。

4. **host**模式：

- a. 容器与Docker宿主机共享网络栈，容器使用Docker宿主机的网络。
- b. 该方法将使用Docker宿主机的 D-bus 系统服务存在安全风险。

5. **user-defined**模式：

- 使用用户自定义网络模式来运行容器。

*** 注意：**man docker-run中关于Docker网络模式的说明

```
--network="bridge"          # man docker-run: 设置Docker容器的网络模式，默认为桥接模式。
    Set the Network mode for the container
    'bridge': create a network stack on the default Docker bridge
    'none': no networking
    'container:<name|id>': reuse another container's network stack
    'host': use the Docker host network stack. Note: the host mode gives the container full access to local
system services such as D-bus and is therefore considered insecure.
    '<network-name>|<network-id>': connect to a user-defined network

# container模式：与另一个Docker容器共用同一个网络栈。
# Kubernetes使用container模式将1个或多个Docker容器共
用同一个网络栈来实现Pod。
```

- Docker网络**bridge**模式配置汇总：

1. 创建自定义Docker网桥：

- a. 配置/etc/sysconfig/docker-network文件：

DOCKER_NETWORK_OPTIONS="--bridge=<user_definied_bridge>"

- b. 重启docker守护进程后将重新生成默认docker0网桥，但该网桥不属于任何

Docker网络。

```
[root@kubernetes ~]# ip address show dev docker0
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:53:60:b9:ed brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0 # 查看Docker默认docker0网桥IP地址
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# systemctl stop docker.service # 停止docker守护进程并删除docker0网桥
[root@kubernetes ~]# ip link set docker0 down
[root@kubernetes ~]# brctl delbr docker0
[root@kubernetes ~]# brctl addbr docker-custom # 创建自定义的Docker网桥，激活并指定IP地址。
[root@kubernetes ~]# ip link set docker-custom up
[root@kubernetes ~]# ip address add 172.42.0.1/16 dev docker-custom
[root@kubernetes ~]# vim /etc/sysconfig/docker-network
[root@kubernetes ~]# systemctl restart docker.service
[root@kubernetes ~]# cat /etc/sysconfig/docker-network
# /etc/sysconfig/docker-network
DOCKER_NETWORK_OPTIONS="--bridge=docker-custom" # --bridge选项：指定Docker的自定义网桥
[root@kubernetes ~]# ip address show dev docker-custom
5: docker-custom: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN qlen 1000
    link/ether 86:d6:49:36:1d:8e brd ff:ff:ff:ff:ff:ff
    inet 172.42.0.1/16 scope global docker-custom # 查看Docker的自定义网桥的IP地址
        valid_lft forever preferred_lft forever
    inet6 fe80::84d6:49ff:fe36:1d8e/64 scope link
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# ip address show dev docker0 # 重启docker守护进程后将会再次创建docker0网桥，但该网桥不存在与任何Docker网络。
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:28:84:7d:f5 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# docker network ls

```

NETWORK ID	NAME	DRIVER	SCOPE
45ee14c7533b	bridge	bridge	local
ea82408339aa	host	host	local
e022b8476f4c	none	null	local

```
[root@kubernetes ~]#
[root@kubernetes ~]# docker network inspect 45ee14c7533b
[
  {
    "Name": "bridge", # 自定义Docker网桥网络的名称
    "Id": "45ee14c7533bc969570fe4298b2a1763754d529a7758866185d1771008e208ae",
    "Created": "2018-09-15T23:57:46.969791928+08:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.42.0.0/16",
          "Gateway": "172.42.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Containers": {},
    "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": "docker-custom", # 自定义Docker网桥的名称
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
[root@kubernetes ~]#
```

2. 创建自定义Docker网桥网络：

- 创建的自定义Docker网桥网络与默认docker0网桥网络可同时存在。
- 自定义的Docker网桥网络将自动生成新的Docker网桥，该网桥名称将随机生成。

成。

```
[root@kubernetes ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
c80f66f4ccc9        bridge             bridge             local
ea82408339aa        host              host              local
e022b8476f4c        none              null              local
[root@kubernetes ~]# docker network create --driver=bridge docker-custom
6bf409d6fc2edd3d2328282e76f18c149a8902ae7048e2c75573968559873de
[root@kubernetes ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
c80f66f4ccc9        bridge             bridge             local
6bf409d6fc2e        docker-custom      bridge             local
ea82408339aa        host              host              local
e022b8476f4c        none              null              local
[root@kubernetes ~]# ip address show dev br-6bf409d6fc2e
15: br-6bf409d6fc2e: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:b4:1d:99:ef brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 scope global br-6bf409d6fc2e
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# docker run -d --network=docker-custom \
> kube-registry.domain12.example.com:5000/centos:6.8 \
> /bin/bash -c 'while true; do echo hello world; sleep 5; done'
5efa44dedbc2369a874a0b7db62213914c1f00be0c58b22b316a0cef5c465e
[root@kubernetes ~]# docker exec 5efa44dedbc ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:AC:12:00:02
          inet addr:172.18.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe12:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2682 (2.6 KiB)  TX bytes:648 (648.0 b)
```

docker守护进程创建的默认docker0网桥网络

创建以bridge网桥驱动自定义网桥网络，该网络可与默认docker0网桥网络共存。

该Linux虚拟网桥由自定义网桥网络创建

运行容器时指定使用自定义的网桥网络

容器的网络命名空间的虚拟网卡桥接于自定义的网桥上

3. 更改默认docker0网桥IP地址:

a. 配置/etc/sysconfig/docker-network文件:

```
DOCKER_NETWORK_OPTIONS="--bip=<new_ip_address>"
# /etc/sysconfig/docker-network
DOCKER_NETWORK_OPTIONS="--bip=172.42.0.1/16"
~
# 指定Docker默认网桥docker0的IP地址，默认为172.17.0.1/16。
```

```
[root@kubernetes ~]# ip address show dev docker0
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:65:ef:8f:17 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:65ff:feef:8f17/64 scope link
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# vim /etc/sysconfig/docker-network
[root@kubernetes ~]# systemctl restart docker.service
[root@kubernetes ~]# ip address show dev docker0
5: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:65:ef:8f:17 brd ff:ff:ff:ff:ff:ff
    inet 172.42.0.1/16 scope global docker0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:65ff:feef:8f17/64 scope link
        valid_lft forever preferred_lft forever
[root@kubernetes ~]# docker run -d kube-registry.domain12.example.com:5000/centos:6.8 \
> /bin/bash -c 'while true; do echo hello world; sleep 5; done'
7bf4dbcb6b26f043d8ce0344f701cd6a8aa8316d77ae2e4e0045afcd3eb89b10
[root@kubernetes ~]# docker exec 7bf4dbcb6b26 ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:AC:2A:00:02
          inet addr:172.42.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe2a:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:648 (648.0 b)  TX bytes:648 (648.0 b)
```

配置Docker默认网桥docker0的指定IP地址

重启docker守护进程后查看指定IP地址是否生效

运行的容器默认情况下将桥接于docker0网桥上，并使用新指定的IP地址。

Docker network管理命令:

- > \$ docker network --help
查看docker network命令的使用方法
- > \$ docker network create --driver=bridge <docker_network_name>
创建以bridge网桥驱动类型的的Docker自定义网络
- > \$ docker network rm <docker_network_name>|<docker_network_id>
删除Docker自定义网络
- > \$ docker network ls
查看创建的Docker网络，默认为bridge模式。
- > \$ docker network inspect <docker_network_name>|<docker_network_id>

查看指定Docker网络的详细信息