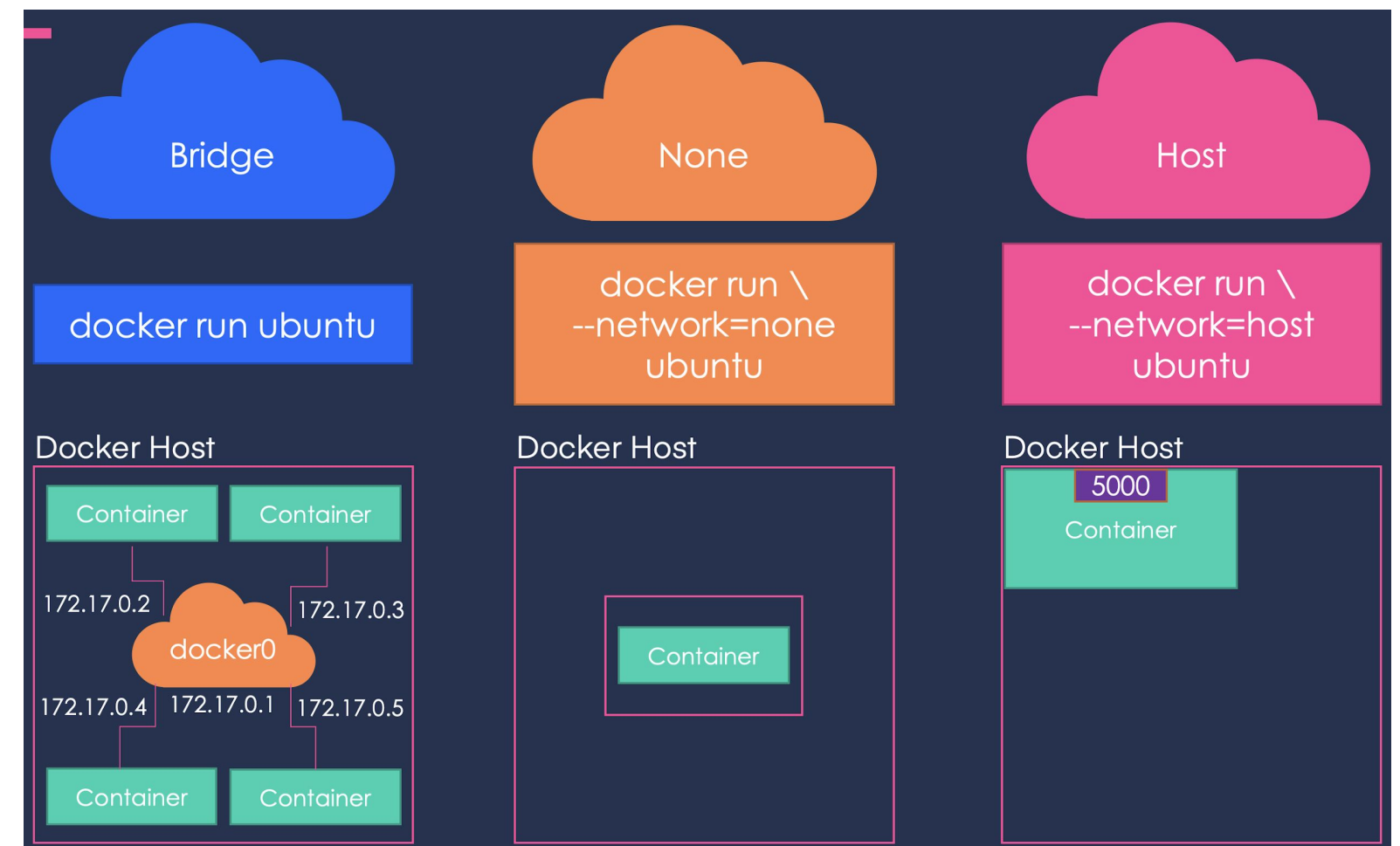


Работа с сетью в Docker

Обзор сетей Docker

Виды сетей Docker:

- bridge
- host
- overlay
- macvlan
- none



Управление пользовательскими сетями bridge

Создание определяемой пользователем сети bridge — **\$ docker network create my-net**

Вы можете указать подсеть, диапазон IP-адресов, шлюз и другие параметры

Подсказку можно посмотреть через команду — **\$ docker network create --help**

Чтобы удалить определённую пользователем мостовую сеть — **\$ docker network rm my-net**

(Если контейнеры в настоящее время подключены к сети, сначала отключите их)

```
$ docker create --name my-nginx \  
  --network my-net \  
  --publish 8080:80 \  
  nginx:latest
```

Подключить контейнер к сети — **\$ docker network connect my-net my-nginx**

Отключить контейнер от сети — **\$ docker network disconnect my-net my-nginx**

Overlay-сеть

- Называется оверлейной сетью ingress, которая обрабатывает трафик управления и данных
- Называется мостовая сеть, которая соединяет отдельный демон Docker с другими демонами, участвующими в swarm

Создать оверлейную сеть для использования со службами Swarm:

```
$ docker network create -d overlay my-overlay
```

Создать оверлейную сеть для использования со службами или автономными контейнерами:

```
$ docker network create -d overlay --attachable my-attachable-overlay
```

Создать оверлейную сеть и подключить к ней неуправляемые контейнеры:

```
$ docker network create --opt encrypted --driver overlay --attachable my-attachable-multi-host-network
```

Создать новую оверлейную сеть с помощью настраиваемых параметров:

```
$ docker network create --driver overlay --ingress --subnet=10.11.0.0/16 --gateway=10.11.0.2 --opt  
com.docker.network.driver.mtu=1200 my-ingress
```

Сеть macvlan

Режим bridge

```
$ docker network create -d macvlan --subnet=172.16.86.0/24 --gateway=172.16.86.1 -o parent=eth0  
pub_net
```

Режим bridge с исключением IP-адресов

```
$ docker network create -d macvlan --subnet=192.168.32.0/24 --ip-range=192.168.32.128/25  
--gateway=192.168.32.254 --aux-address="my-router=192.168.32.129" -o parent=eth0 macnet32
```

Режим bridge 802.1q

```
$ docker network create -d macvlan --subnet=192.168.50.0/24 --gateway=192.168.50.1 -o parent=eth0.50  
macvlan50
```

Использование ipvlan вместо macvlan

```
$ docker network create -d ipvlan --subnet=192.168.210.0/24 --subnet=192.168.212.0/24  
--gateway=192.168.210.254 --gateway=192.168.212.254 -o ipvlan_mode=l2 -o parent=eth0 ipvlan210
```

Отключение сети для контейнера

Создать контейнер без сети командой:

```
$ docker run --rm -dit --network none --name no-net-alpine alpine:latest ash
```

```
$ docker exec no-net-alpine ip link show — покажет вам, что у контейнера нет сети
```

```
$ docker exec no-net-alpine ip route — команда вернёт пустое значение, потому что таблицы маршрутизации нет
```

Если вы остановите контейнер, он автоматически удалится, поскольку был создан с `--rm` флагом

```
$ docker stop no-net-alpine
```

Пример использования сети Docker

docker network ls

ifconfig

```
root@hell:/home/kryl# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
931fac9e474d    bridge   bridge      local
8d9728edb964    host     host        local
f74e37d1c8dc    none     null        local
root@hell:/home/kryl# ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:59ff:fe07:9d20 prefixlen 64 scopeid 0x20<link>
    ether 02:42:59:07:9d:20 txqueuelen 0 (Ethernet)
    RX packets 37181 bytes 1573246 (1.5 MB)
    RX errors 0 dropped 0 overruns 0 cb frame 0
    TX packets 60587 bytes 192256185 (192.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::93ef:909b:e10f:c6cb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1c:5d:0e txqueuelen 1000 (Ethernet)
    RX packets 169387 bytes 229398910 (229.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
```


Пример использования сети Docker

```
docker network create -d macvlan --subnet=10.0.2.15/24 --gateway=10.0.2.1 -o parent=enp0s3 pub_net
```

```
docker network ls
```

```
root@hell:/home/kryl# docker network create -d macvlan --subnet=10.0.2.15/24 --gateway=10.0.2.1 -o parent=enp0s3 pub_net
a33995fd96ae4b3c67d9a45db5ed1103b7470c3a88393265739f338bc2b5bf
root@hell:/home/kryl# docker network ls
NETWORK ID   NAME      DRIVER    SCOPE
931fac9e474d  bridge   bridge    local
8d9728edb964  host     host      local
f74e37d1c8dc  none     null      local
a33995fd96ae  pub_net  macvlan   local
root@hell:/home/kryl# docker run --net=pub_net --ip=10.0.2.16 -itd alpine /bin/sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4c0d98bf9879: Pull complete
Digest: sha256:08d6ca16c60fe7490c03d10dc339d9fd8ea67c6466dea8d558526b1330a85930
Status: Downloaded newer image for alpine:latest
1c6e1461315aadfc8efa69e1cd21ac3af4fa9d8f329c29f80bb78aba73eb98c6
```


Пример использования сети Docker

```
docker run --net=pub_net --ip=10.0.2.16 -itd alpine /bin/sh
```

```
root@hell:/home/kryl# docker run --net=pub_net --ip=10.0.2.16 -itd alpine /bin/sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
4c0d98bf9879: Pull complete
Digest: sha256:08d6ca16c60fe7490c03d10dc339d9fd8ea67c6466dea8d558526b1330a85930
Status: Downloaded newer image for alpine:latest
1c6e1461315aadfc8efa69e1cd21ac3af4fa9d8f329c29f80bb78aba73eb98c6
```

Пример использования сети Docker

```
root@hell:/home/kryl# ip link add mac0 link enp0s3 type macvlan mode bridge
root@hell:/home/kryl# ip addr add 10.0.2.17/24 dev mac0
root@hell:/home/kryl# ifconfig mac0 up
root@hell:/home/kryl# docker run --net=pub_net -d --ip=10.0.2.18 -p 80:80 nginx
```

```
root@hell:/home/kryl# docker run --net=pub_net -d --ip=10.0.2.18 -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
45b42c59be33: Pull complete
d0d9e9ea897e: Pull complete
66e650438339: Pull complete
76a3dfe4406b: Pull complete
410ff9d97480: Pull complete
Digest: sha256:8e10956422503824ebb599f37c26a90fe70541942687f70bbdb744530fc9eba4
Status: Downloaded newer image for nginx:latest
415c44a1ba6acd9b63ec901fd2a188dc2d1b7efdbd37c8c8a04c71f5a288bfed
root@hell:/home/kryl# ping 10.0.2.18
PING 10.0.2.18 (10.0.2.18) 56(84) bytes of data.
64 bytes from 10.0.2.18: icmp_seq=1 ttl=64 time=0.278 ms
64 bytes from 10.0.2.18: icmp_seq=2 ttl=64 time=0.064 ms
```

Итоги

- Узнали, что такое сети Docker
- Узнали, какие сети бывают и чем они отличаются
- Рассмотрели пример с использованием одной из сетей

**Спасибо
за внимание!**