1.1 容器网络

1.1.1 Bridge 网络

步骤 1　创建一个名为 net1 的 Bridge 网络。

```
[root@localhost ~]# docker network create --driver bridge net1
```

```
[root@localhost ~]# docker network create --driver bridge net1
93fb485fc2471ed14cc5292ddf3650f5eb277fe31a2c468b87052ad639a2135c
```

步骤 2　查看 net1 网桥，subnet 已自动配置为 172.18.0.0/16。

```
[root@localhost ~]# docker network inspect net1
```

```
[root@localhost ~]# docker network inspect net1
[
    {
        "Name": "net1",
        "Id": "93fb485fc2471ed14cc5292ddf3650f5eb277fe31a2c468b87052ad639a2135c",
        "Created": "2019-08-13T23:23:55.111499596-04:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
```

步骤 3　创建一个名为 net2 的 bridge 网桥，并指定 subnet=172.10.10.0/24。

```
[root@localhost ~]# docker network create --driver bridge --subnet
172.10.10.0/24 --gateway 172.10.10.1 net2
```

```
[root@localhost ~]# docker network create --driver bridge --subnet 172.10.10.0/24 --gateway 172.10.10.1 net2
471e136c28316c9e232f503fdb113908fdf23954d76410e535e57ee69510ccc2
```

步骤 4　启动名为 centos1 的容器，并加入 net1 网络。

```
[root@localhost ~]# docker run --name centos1 -dit --network=net1 centos
```

```
[root@localhost ~]# docker run --name centos1 -dit --network=net1 centos
be7ab6026d73ca4417eb9ef325e2a7d388e8881314cb3bd4e8e562dfef9ec48c
```

步骤 5　启动名为 centos2 的容器，并加入 net2 网络。

```
[root@localhost ~]# docker run --name centos2 -dit --network=net2 centos
```

```
[root@localhost ~]# docker run --name centos2 -dit --network=net2 centos
88056a697e0856129d67a64251dabc66388e6e09453a1303b37e7d6a3d2ac580
```

步骤 6　启动名为 centos3 的容器，并加入 net2 网络，同时指定该容器 IP=172.10.10.10。

```
[root@localhost ~]# docker run --name centos3 -dit --network=net2 --ip
172.10.10.10 centos
```

```
[root@localhost ~]# docker run --name centos3 -dit --network=net2 --ip 172.10.10.10 centos
7d66df3d3acbfe3be565c5421f2e86bf68c29889298caba35c9dbe07a02b6135
```

步骤 7　分别查看 3 个 centos 容器的 IP 地址信息。其中 centos2 和 centos3 位于同一网段，centos1 与前两者位于不同网段。

```
docker inspect centos1
```

```
"Networks": {
    "net1": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
            "be7ab6026d73"
        ],
        "NetworkID": "93fb485fc2471ed14cc5292ddf3650f5eb277fe31a2c468b87052ad639a2135c",
        "EndpointID": "7943c6017295cae634bb5ede204802f3eb356741250f6be81bd35dc4215e9d37",
        "Gateway": "172.18.0.1",
        "IPAddress": "172.18.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:12:00:02",
        "DriverOpts": null
    }
```

```
docker inspect centos2
```

```
"Networks": {
    "net2": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
            "88056a697e08"
        ],
        "NetworkID": "471e136c28316c9e232f503fdb113908fdf23954d76410e535e57ee69510ccc2",
        "EndpointID": "92d12ded719afdc0bc28d4dd0c0192ca3d12328a011e446a0945f1fb1501012f",
        "Gateway": "172.10.10.1",
        "IPAddress": "172.10.10.2",
        "IPPrefixLen": 24,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:0a:0a:02",
        "DriverOpts": null
    }
```

```
docker inspect centos3
```

```
"Networks": {
    "net2": {
        "IPAMConfig": {
            "IPv4Address": "172.10.10.10"
        },
        "Links": null,
        "Aliases": [
            "7d66df3d3acb"
        ],
        "NetworkID": "471e136c28316c9e232f503fdb113908fdf23954d76410e535e57ee69510ccc2",
        "EndpointID": "f47a1c7a32a018d27ab63b0c0fd64c65dc03731843931f992b56cf97ef43200e",
        "Gateway": "172.10.10.1",
        "IPAddress": "172.10.10.10",
        "IPPrefixLen": 24,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:0a:0a:0a",
        "DriverOpts": null
    }
```

步骤 8　进入容器 centos3，验证网络连通性。centos3 与 centos2 可以通信，但与 centos1 不能通信。

```
[root@localhost ~]# docker exec -it centos3 bash
```

```
[root@localhost ~]# docker exec -it centos3 bash
[root@7d66df3d3acb /]# ping 172.10.10.2
PING 172.10.10.2 (172.10.10.2) 56(84) bytes of data.
64 bytes from 172.10.10.2: icmp_seq=1 ttl=64 time=0.172 ms
64 bytes from 172.10.10.2: icmp_seq=2 ttl=64 time=0.105 ms
64 bytes from 172.10.10.2: icmp_seq=3 ttl=64 time=0.105 ms
^C
--- 172.10.10.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.105/0.127/0.172/0.032 ms
[root@7d66df3d3acb /]# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2) 56(84) bytes of data.
^C
--- 172.18.0.2 ping statistics ---
168 packets transmitted, 0 received, 100% packet loss, time 167000ms

[root@7d66df3d3acb /]# exit
```

步骤 9　为 centos1 添加一块网卡，并加入 net2 网络。

```
[root@localhost ~]# docker network connect net2 centos1
```

```
[root@localhost ~]# docker network connect net2 centos1
```

步骤 10　进入 centos1 容器测试连通性。

```
[root@localhost ~]# docker exec -it centos1 bash
[root@be7ab6026d73 /]# ping 172.10.10.2
PING 172.10.10.2 (172.10.10.2) 56(84) bytes of data.
64 bytes from 172.10.10.2: icmp_seq=1 ttl=64 time=0.126 ms
64 bytes from 172.10.10.2: icmp_seq=2 ttl=64 time=0.100 ms
64 bytes from 172.10.10.2: icmp_seq=3 ttl=64 time=0.109 ms
64 bytes from 172.10.10.2: icmp_seq=4 ttl=64 time=0.107 ms
^C
--- 172.10.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.100/0.110/0.126/0.014 ms
[root@be7ab6026d73 /]# ping 172.10.10.10
PING 172.10.10.10 (172.10.10.10) 56(84) bytes of data.
64 bytes from 172.10.10.10: icmp_seq=1 ttl=64 time=0.152 ms
64 bytes from 172.10.10.10: icmp_seq=2 ttl=64 time=0.123 ms
64 bytes from 172.10.10.10: icmp_seq=3 ttl=64 time=0.138 ms
64 bytes from 172.10.10.10: icmp_seq=4 ttl=64 time=0.132 ms
^C
--- 172.10.10.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.123/0.136/0.152/0.013 ms
```

步骤 11　为方便后续实验，删除本小节中的容器。

```
docker kill
docker rm
```