

Experiment 3: Docker Network

Aim

To understand networking in docker and create a network allowing containers to communicate.

Step 1: Listing all networks

```
root@redhat:~  
File Edit View Search Terminal Help  
[root@redhat ~]# docker network ls  
NETWORK ID          NAME                DRIVER              SCOPE  
1a1b1d30ae7e        bridge             bridge              local  
75539a863aad        host               host                local  
7b99ee49f373        none              null                local  
[root@redhat ~]#
```

Step2: Creating our network

```
[root@redhat ~]# docker network create mynet  
85bd068d6d2e000002e703d8ab7a1c74db28063de07d4a8664d83359d1f76cd0  
[root@redhat ~]#
```

Step 3: Launching container in the created mynet network

```
[root@redhat ~]# docker run -it --net=mynet --name C1 centos  
[root@8b6ccaf6f50e /]#
```

Step 4: Launching another container in the same network

```
[root@redhat ~]# docker run -it --net=mynet --name C2 centos
[root@eca363b5389f /]#
[root@eca363b5389f /]#
```

Step 5: Now from C2 we can ping to C1

```
[root@eca363b5389f /]# ping C1
PING C1 (172.18.0.3) 56(84) bytes of data.
64 bytes from C1.mynet (172.18.0.3): icmp_seq=1 ttl=64 time=0.227 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=2 ttl=64 time=0.284 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=3 ttl=64 time=0.176 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=4 ttl=64 time=0.176 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=5 ttl=64 time=0.184 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=6 ttl=64 time=0.199 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=7 ttl=64 time=0.177 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=8 ttl=64 time=0.288 ms
^X64 bytes from C1.mynet (172.18.0.3): icmp_seq=9 ttl=64 time=0.174 ms
64 bytes from C1.mynet (172.18.0.3): icmp_seq=10 ttl=64 time=0.180 ms
^C
--- C1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 216ms
rtt min/avg/max/mdev = 0.174/0.206/0.288/0.044 ms
[root@eca363b5389f /]#
```

Inspecting network mynet :-

```
[root@redhat ~]# docker network inspect mynet
[
  {
    "Name": "mynet",
    "Id": "85bd068d6d2e000002e703d8ab7a1c74db28063de07d4a8664d83359d1f76cd0",
    "Created": "2021-02-13T16:39:27.579720819+05:30",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    }
  }
]
```

```

      "8b6ccaf6f50e0f44143bdbba32ec77d75a855a991a22b4f4118654b25a1c3335":
    {
      "Name": "C1",
      "EndpointID": "097e55e1a4b9b4f9743fe9a8c9ceb28d5622b27a525d50f37
b3604ca784f8c86",
      "MacAddress": "02:42:ac:12:00:03",
      "IPv4Address": "172.18.0.3/16",
      "IPv6Address": ""
    },
    "eca363b5389f6f8f7415f89c2c4b9fde4cb461b3227eaa21cc7f22c0cfc64db6":
    {
      "Name": "C2",
      "EndpointID": "e85eb5fa6ed7c0bc57fe38ef02d2c433ca79c831b8b3a7fbe
9033d5d00f6aaee",
      "MacAddress": "02:42:ac:12:00:02",
      "IPv4Address": "172.18.0.2/16",
      "IPv6Address": ""
    }
  },
  "Options": {},
  "Labels": {}
}
]
[root@redhat ~]#

```

Disconnecting the container from the network :-

```

[root@redhat ~]# docker network disconnect mynet C1
[root@redhat ~]#

```