Pratik Agrawal

500123601

Devops B2

Lab Exercise 3- Working with Docker Networking

Step 1: Understanding Docker Default Networks

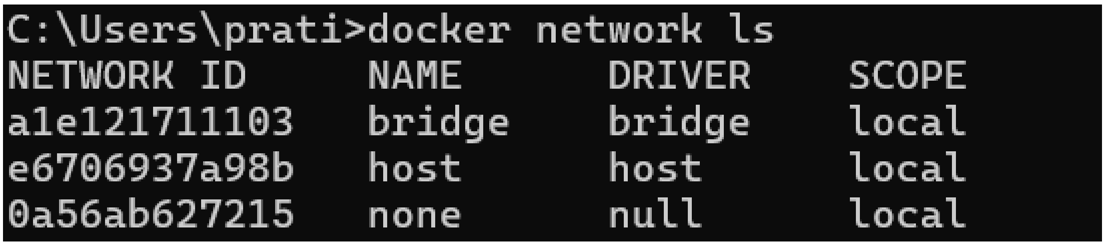
Docker provides three default networks:

- bridge: The default network when a container starts.
- host: Bypasses Docker's network isolation and attaches the container directly to the host network.
- none: No networking is available for the container.

1.1. Inspect Default Networks

Check Docker's default networks using:

```
docker network ls
```

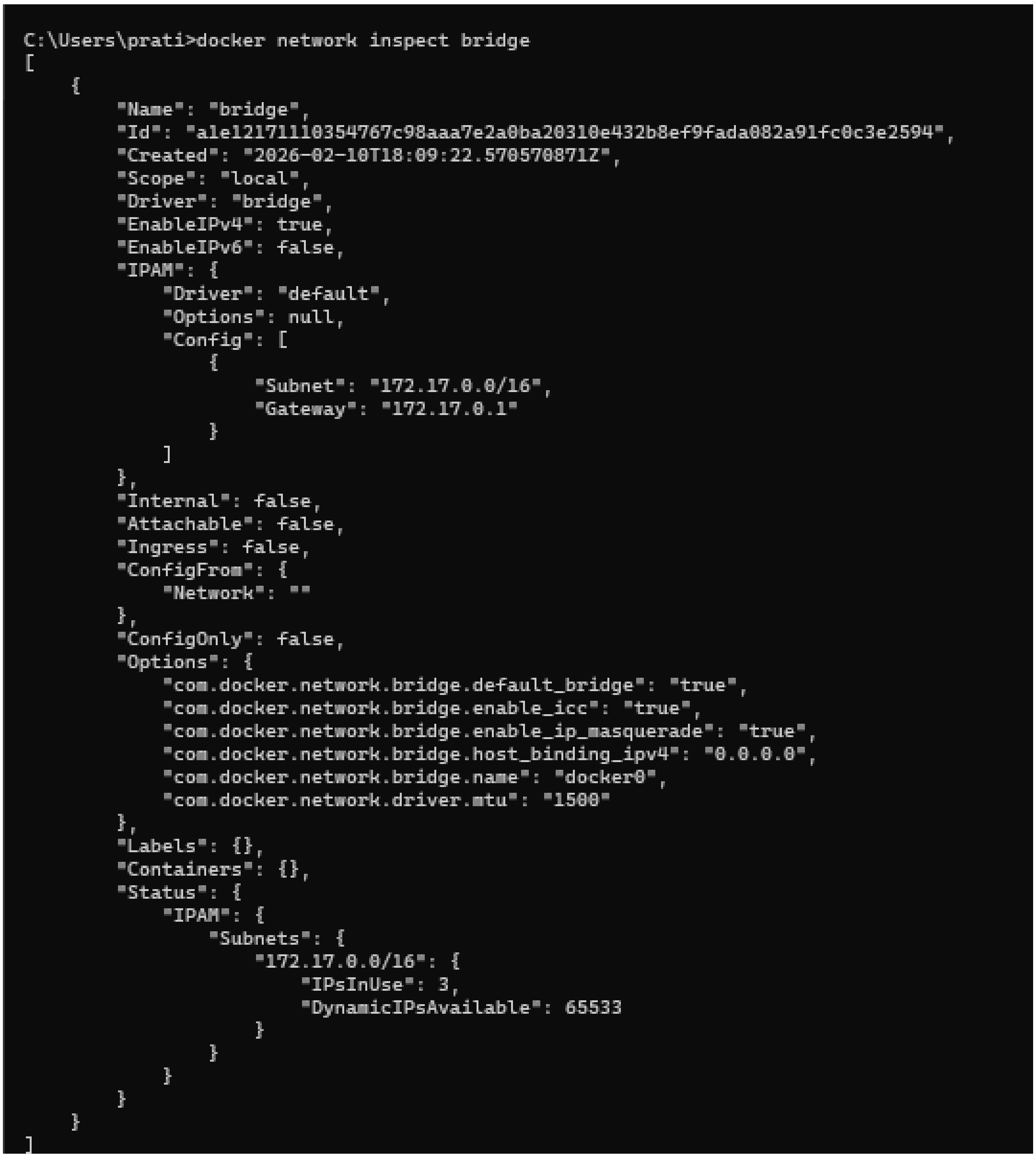


```
C:\Users\prati>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a1e121711103   bridge    bridge    local
e6706937a98b   host      host      local
0a56ab627215   none      null      local
```

1.2. Inspect the Bridge Network

```
docker network inspect bridge
```

This command will show detailed information about the bridge network, including the connected containers and IP address ranges.



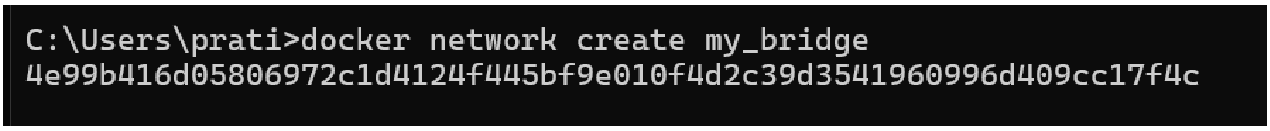
```
C:\Users\prati>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "a1e12171110354767c98aaa7e2a0ba20310e432b8ef9fada082a91fc0c3e2594",
        "Created": "2026-02-10T18:09:22.570570871Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "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": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {},
        "Containers": {},
        "Status": {
            "IPAM": {
                "Subnets": {
                    "172.17.0.0/16": {
                        "IPsInUse": 3,
                        "DynamicIPsAvailable": 65533
                    }
                }
            }
        }
    }
]
```

Step 2: Create and Use a Bridge Network

2.1. Create a User-Defined Bridge Network

A user-defined bridge network allows containers to communicate by name instead of IP.

```
docker network create my_bridge
```



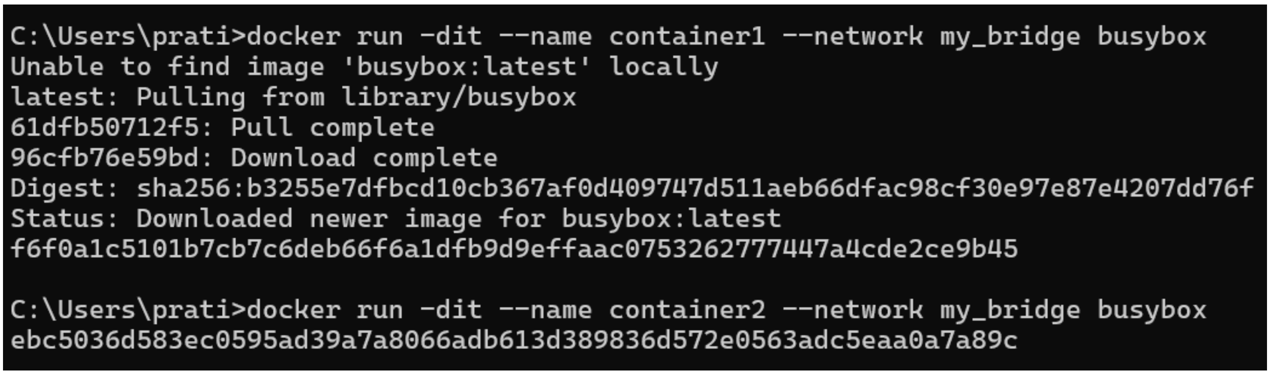
```
C:\Users\prati>docker network create my_bridge
4e99b416d05806972c1d4124f445bf9e010f4d2c39d3541960996d409cc17f4c
```

2.2. Run Containers on the User-Defined Network

Start two containers on the newly created my_bridge network:

```
docker run -dit --name container1 --network my_bridge busybox


docker run -dit --name container2 --network my_bridge busybox
```



```
C:\Users\prati>docker run -dit --name container1 --network my_bridge busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
61dfb50712f5: Pull complete
96cfb76e59bd: Download complete
Digest: sha256:b3255e7dfbcd10cb367af0d409747d511aeb66dfac98cf30e97e87e4207dd76f
Status: Downloaded newer image for busybox:latest
f6f0a1c5101b7cb7c6deb66f6a1dfb9d9effaac0753262777447a4cde2ce9b45

C:\Users\prati>docker run -dit --name container2 --network my_bridge busybox
ebc5036d583ec0595ad39a7a8066adb613d389836d572e0563adc5eaa0a7a89c
```

2.3. Test Container Communication

Execute a ping command from container1 to container2 using container names:

```
docker exec -it container1 ping container2
```

The containers should be able to communicate since they are on the same network.

Step 3: Disconnect and Remove Networks

3.1. Disconnect Containers from Networks

To disconnect container1 from my_bridge:

```
docker network disconnect my_bridge container1
```

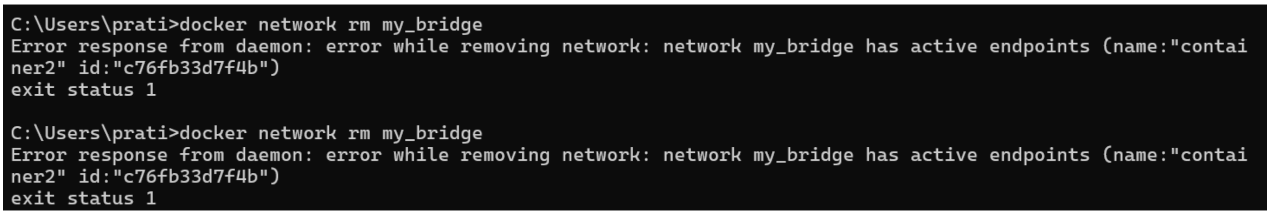
4.2. Remove Networks

To remove the user-defined network:

```
docker network rm my_bridge
```

Step 4: Clean Up

Stop and remove all containers created during this exercise:



```
C:\Users\prati>docker network rm my_bridge
Error response from daemon: error while removing network: network my_bridge has active endpoints (name:"contai
ner2" id:"c76fb33d7f4b")
exit status 1

C:\Users\prati>docker network rm my_bridge
Error response from daemon: error while removing network: network my_bridge has active endpoints (name:"contai
ner2" id:"c76fb33d7f4b")
exit status 1
```

```
docker network rm my_bridge
```