Lab Exercise 3- Working with Docker Networking

Step 1: Understanding Docker Default Networks

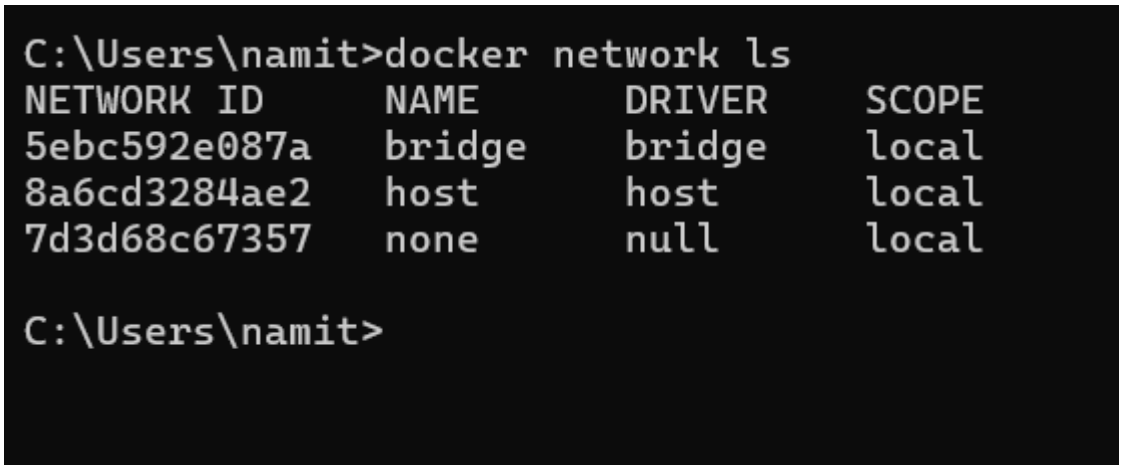
Docker provides three default networks:

- 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

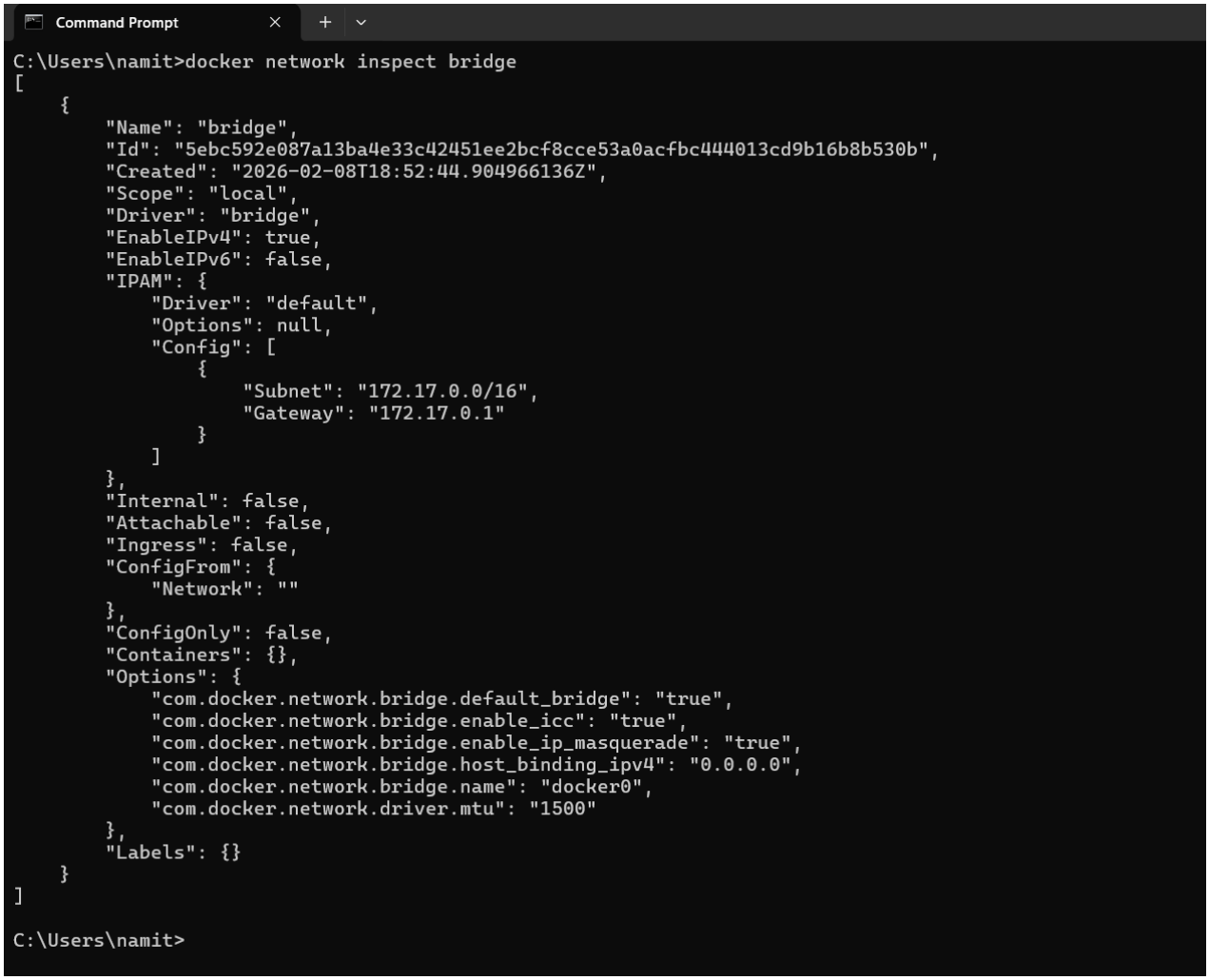
Check Docker's default networks using:

```
docker network ls
```



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.

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\namit>docker network create my_bridge
4eacbcc25eab6a18356cb1053b555137f6672dd056131595c8fdf2bdf3e4c237

C:\Users\namit>
```

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
```

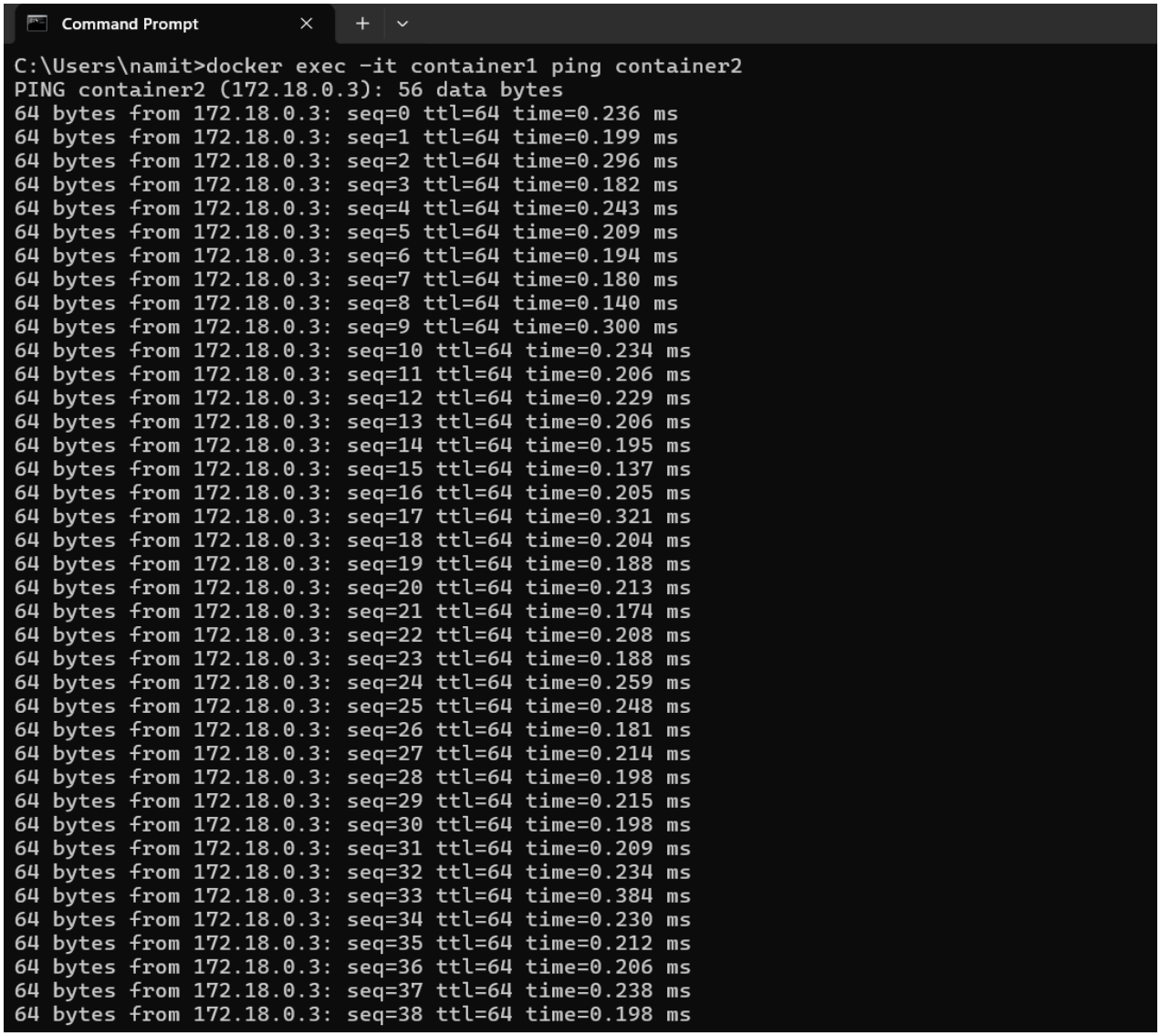
```
C:\Users\namit>docker run -dit --name container1 --network my_bridge busybox
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
61dfb50712f5: Pull complete
Digest: sha256:b3255e7dfbcd10cb367af0d409747d511aeb66dfac98cf30e97e87e4207dd76f
Status: Downloaded newer image for busybox:latest
b78f0c0137384c2b2b6c5ec109a0ac834ca0ca602e1960230f1f8f24c9ed3170

C:\Users\namit>docker run -dit --name container2 --network my_bridge busybox
f21777137710eb8c66e2d7b5620ff8de6894acf00db7cbd51b6e975be907c7cf

C:\Users\namit>
```

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
```

```
--- container2 ping statistics ---
91 packets transmitted, 91 packets received, 0% packet loss
round-trip min/avg/max = 0.131/0.217/0.968 ms

C:\Users\namit>docker network disconnect my_bridge container1

C:\Users\namit>
```

```
C:\Users\namit>docker network disconnect my_bridge container2

C:\Users\namit>
```

4.2. Remove Networks

To remove the user-defined network:

```
docker network rm my_bridge
```

```
C:\Users\namit>docker network rm my_bridge
my_bridge

C:\Users\namit>
```

Step 4: Clean Up

Stop and remove all containers created during this exercise:

```
docker rm -f container1 container2
```

```
C:\Users\namit>docker rm -f container1 container2
container1
container2

C:\Users\namit>
```