Name : Gourav Das

Sap : 500122586

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
```

```
gouravs-MacBook-Air:~ gouravdas$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a37f8ca49305   bridge    bridge    local
361b1e82596d   host      host      local
629872d4fc68   none      null      local
gouravs-MacBook-Air:~ gouravdas$
```

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.

```
a37f8ca49305   bridge    bridge    local
361b1e82596d   host      host      local
629872d4fc68   none      null      local
gouravs-MacBook-Air:~ gouravdas$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "a37f8ca49305985014869fe0bb4b6cb83156319e77887e269f7550e5c5007284",
        "Created": "2026-01-27T08:54:11.557120752",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "IPRange": "",
                    "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": "65535"
        },
        "Labels": {},
        "Containers": {
            "b1ac409c39c6c5fcb2cbcf55893ef2620aa631e93bd70b2b0d8db3f67d76f704": {
                "Name": "welcome-to-docker",
                "EndpointID": "f108885f979ec9d3439b7f5c7203124811ee543dfcdc8d38a7bc0d521fa890e7",
                "MacAddress": "1e:da:bd:f1:4b:61",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Status": {
            "IPAM": {
                "Subnets": {
                    "172.17.0.0/16": {
                        "IPsInUse": 4,
                        "DynamicIPsAvailable": 65532
                    }
                }
            }
        }
    }
]
gouravs-MacBook-Air:~ gouravdas$
```

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
```

```
gouravdas@gouravs-MacBook-Air ~ % docker network create my_bridge

Error response from daemon: network with name my_bridge already exists
gouravdas@gouravs-MacBook-Air ~ %
```

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
```

```
Run 'docker run --help' for more information
gouravs-MacBook-Air:~ gouravdas$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED              STATUS              PORTS                                       NAMES
22d12e526620   busybox                            "sh"                     44 seconds ago       Up 43 seconds                                           container2
138754a2a26d   busybox                            "sh"                     About a minute ago   Up About a minute                                       container1
b1ac409c39c6   docker/welcome-to-docker:latest    "/docker-entrypoint.…"   17 minutes ago       Up 17 minutes       0.0.0.0:8088->80/tcp, [::]:8088->80/tcp   welcome-to-docker
gouravs-MacBook-Air:~ gouravdas$
```

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

```
64 bytes from 172.18.0.3: seq=49 ttl=64 time=0.318 ms
64 bytes from 172.18.0.3: seq=50 ttl=64 time=0.284 ms
64 bytes from 172.18.0.3: seq=51 ttl=64 time=0.283 ms
64 bytes from 172.18.0.3: seq=52 ttl=64 time=0.259 ms
64 bytes from 172.18.0.3: seq=53 ttl=64 time=0.248 ms
64 bytes from 172.18.0.3: seq=54 ttl=64 time=0.273 ms
64 bytes from 172.18.0.3: seq=55 ttl=64 time=0.263 ms
64 bytes from 172.18.0.3: seq=56 ttl=64 time=0.153 ms
64 bytes from 172.18.0.3: seq=57 ttl=64 time=0.288 ms
64 bytes from 172.18.0.3: seq=58 ttl=64 time=0.314 ms
64 bytes from 172.18.0.3: seq=59 ttl=64 time=0.418 ms
64 bytes from 172.18.0.3: seq=60 ttl=64 time=0.291 ms
```

3.1. Disconnect Containers from Networks

To disconnect container1 from my_bridge:

```
docker network disconnect my_bridge container1
```

4.2. Remove Networks

```
Run 'docker run --help' for more information
gouravdas@gouravs-MacBook-Air ~ % docker network disconnect my_bridge container1

Error response from daemon: container 138754a2a26d34c62bec2a2035f4e54168a56c5b7b6d7fb774e7e53a6b0ef21e is not connected to network my_bridge
gouravdas@gouravs-MacBook-Air ~ %
```

To remove the user-defined network:

```
docker network rm my_bridge
```

```
Error response from daemon: error while removing network: network my_bridge has active endpoints (name:"container2" id:"c8a90204eb70")
exit status 1
gouravdas@gouravs-MacBook-Air ~ %
```

Step 4: Clean Up

Stop and remove all containers created during this exercise:

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

```
Error response from daemon: error while removing network: network my_bridge has active endpoints (name:"container2" id:"c8a90204eb70")
exit status 1
gouravdas@gouravs-MacBook-Air ~ % docker rm -f container1 container2

container1
container2
gouravdas@gouravs-MacBook-Air ~ %
```