

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\OM VATS>docker network ls
NETWORK ID          NAME       DRIVER  SCOPE
2af526d2ce6c        bridge    bridge  local
9083563b1749        host      host    local
32b1da208656        none      null    local
```

1.2. Inspect the Bridge Network

```
docker network inspect bridge
```

```
C:\Users\OM VATS>docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "2af526d2ce6c301ef1f8d37e2edfd677fa89e615b165ca1879ab6a8cd208d
ade",
    "Created": "2024-09-13T06:12:35.64234999Z",
    "Scope": "local",
    "Driver": "bridge",
    "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,
    "Containers": {},
    "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",

```

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\OM VATS>docker network create my_bridge
d64d17de3ee8bfa5bc3f9b3af7e3ae5158b834263f8a743b1d1926f8e5d9b96f
```

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\OM VATS>docker run -dit --name container2 --network my_bridge busybox
6557b775208cbb34200677d6218957945a4b7ceae190f3a4255f946acb0c4f
```

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.

```
C:\Users\OM VATS>docker exec -it container1 ping container2
PING container2 (172.18.0.3): 56 data bytes
64 bytes from 172.18.0.3: seq=0 ttl=64 time=0.089 ms
64 bytes from 172.18.0.3: seq=1 ttl=64 time=0.069 ms
64 bytes from 172.18.0.3: seq=2 ttl=64 time=0.067 ms
64 bytes from 172.18.0.3: seq=3 ttl=64 time=0.058 ms
64 bytes from 172.18.0.3: seq=4 ttl=64 time=0.067 ms
64 bytes from 172.18.0.3: seq=5 ttl=64 time=0.087 ms
64 bytes from 172.18.0.3: seq=6 ttl=64 time=0.072 ms
64 bytes from 172.18.0.3: seq=7 ttl=64 time=0.070 ms
64 bytes from 172.18.0.3: seq=8 ttl=64 time=0.089 ms
64 bytes from 172.18.0.3: seq=9 ttl=64 time=0.063 ms
64 bytes from 172.18.0.3: seq=10 ttl=64 time=0.093 ms
64 bytes from 172.18.0.3: seq=11 ttl=64 time=0.069 ms
64 bytes from 172.18.0.3: seq=12 ttl=64 time=0.095 ms
64 bytes from 172.18.0.3: seq=13 ttl=64 time=0.066 ms
64 bytes from 172.18.0.3: seq=14 ttl=64 time=0.087 ms
64 bytes from 172.18.0.3: seq=15 ttl=64 time=0.066 ms
64 bytes from 172.18.0.3: seq=16 ttl=64 time=0.086 ms
64 bytes from 172.18.0.3: seq=17 ttl=64 time=0.071 ms
64 bytes from 172.18.0.3: seq=18 ttl=64 time=0.070 ms
64 bytes from 172.18.0.3: seq=19 ttl=64 time=0.074 ms
64 bytes from 172.18.0.3: seq=20 ttl=64 time=0.071 ms
64 bytes from 172.18.0.3: seq=21 ttl=64 time=0.064 ms
64 bytes from 172.18.0.3: seq=22 ttl=64 time=0.070 ms
64 bytes from 172.18.0.3: seq=23 ttl=64 time=0.068 ms
64 bytes from 172.18.0.3: seq=24 ttl=64 time=0.067 ms
```

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
```

```
C:\Users\OM VATS>docker network rm my_bridge
```

Step 4: Clean Up

Stop and remove all containers created during this exercise:

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

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