****

Types Of Network

****

Bridge Network:

1) It is a default network driver for containers
2) Docker Engine supports default and user-defined bridge network
3) It is a software bridge created by Docker
4) Containers in same bridge can communicate with each other, whereas other bridge network is blocked
5) Docker bridge network creates some rules (Iptables) on the host machine

Host Network:

1) Host network driver will assign a container with host network as your Docker host machine uses
2) We need to ensure that applications running on host and containers with host network driver do not use same port number
3) Host networks only work on Linux, not on Docker Desktop for Mac, Windows, or Docker EE on Windows Server
4) It is not recommended to use host network driver for applications, only if it Dynamic ports

None Network

1) If containers should not be attached to any network stack in Docker, we must use none
2) With this driver, container will not be able to connect to outside world as well
3) Applications inside container, will be using loopback interface

Overlay Network:

Overly network will ensure containers sitting on two different Docker Host machines can communicate with each other
This feature is available by default on Docker Swarm
To implement overlay network without using Docker Swarm, we have to use Consul or eted key-value storage

IPvlan (optional topic)

1) It provides complete control of layer2 VLAN tagging and even IPvlan L3 routing for users
2) With this approach, there is no need of bridge network as Docker supports
3) IP address will be assigned to container from network how a Docker host uses

Macvlan - (optional topic) - (advanced)

1) Containers will be assigned MAC addresses, to ensure they appear as physical interface on Docker host machine
2) Docker daemon will route the traffic to appropriate container using MAC addresses
3) It will avoid latency as well for legacy applications to have direct connect

Network Plugins - (optional topic) - (advanced)

1) Docker Engine supports to integrate third-party plugins for Network setup
2) We can pull this info from Docker Hub

****

Type 1: Bridge network

****

check the container

root@ubuntu:~# docker container ps

```
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS       PORTS     NAMES
4ba7c1ec1d11   nginx:latest   "/docker-entrypoint.…"   3 hours ago   Up 2 hours   80/tcp    nginx_practice1
63b049d7ae64   nginx          "/docker-entrypoint.…"   3 hours ago   Up 3 hours   80/tcp    nginx_practice
```

Login to container to identify the IP address

root@ubuntu:~# docker container exec -t -i nginx_practice1 bash

Identify the IP address

root@nginx2:/# ip a

```
bash: ip: command not found
```

Package not available so we need to install the relevant Package

root@nginx2:/# apt update

```
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
:
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

root@nginx2:/# apt install iproute2

```
Reading package lists... Done
Building dependency tree... Done
Processing triggers for libc-bin (2.36-9+deb12u6) ..
```

root@nginx2:/# apt install iputils-ping

```
Reading package lists... Done
Building dependency tree... Done
:
:
Need to get 47.1 kB of archives.
```

Containers Created of the same network are able to communicate with each other

root@nginx2:/# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

root@nginx:/# ping 172.17.0.3

```
PING 172.17.0.3 (172.17.0.3) 56(84) bytes of data.
64 bytes from 172.17.0.3: icmp_seq=1 ttl=64 time=0.947 ms
64 bytes from 172.17.0.3: icmp_seq=2 ttl=64 time=0.094 ms
64 bytes from 172.17.0.3: icmp_seq=3 ttl=64 time=0.083 ms
```

Note : Container within the network can communicate to each other

user defined bridge network creation

root@ubuntu:~# docker network create --subnet 10.100.0.0/16 --gateway 10.100.10.1 Dhileep_network → **Network Creation**

```
235079ae3c9401d78b3f8648a0b6793778c55d485c4af3b64326ea1c8c834fa9
```

root@ubuntu:~# docker network ls

NETWORK ID	NAME	DRIVER	SCOPE
235079ae3c94	Dhileep_network	bridge	local
274daf1d488d	bridge	bridge	local
edf92f2985e3	host	host	local
af1449d68ae4	none	null	local

root@ubuntu:~# #docker container run -d --name ingnx_network --network Dhileep_network nginx:latest

root@ubuntu:~# docker container ps

```
CONTAINER ID   IMAGE          COMMAND                  CREATED       STATUS       PORTS     NAMES
4ba7c1ec1d11   nginx:latest   "/docker-entrypoint.…"   7 hours ago   Up 7 hours   80/tcp    nginx_practice1
63b049d7ae64   nginx          "/docker-entrypoint.…"   7 hours ago   Up 7 hours   80/tcp    nginx_practice
```

root@ubuntu:~# docker container run -d --name ingnx_network --network Dhileep_network nginx:latest

```
2162c68a7c8ff1c721e3474e8b62b53c38ae483671efcd8de7b1be5fccda9b61
```

root@ubuntu:~# docker container ps

```
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
2162c68a7c8f   nginx:latest   "/docker-entrypoint.…"   2 seconds ago    Up 2 seconds    80/tcp    ingnx_network
4ba7c1ec1d11   nginx:latest   "/docker-entrypoint.…"   7 hours ago      Up 7 hours      80/tcp    nginx_practice1
63b049d7ae64   nginx          "/docker-entrypoint.…"   7 hours ago      Up 7 hours      80/tcp    nginx_practice
```

root@ubuntu:~# docker container inspect ingnx_network


```
[
:
        "Networks": {
            "Dhileep_network": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": null,
                "MacAddress": "02:42:0a:64:00:01",
                "NetworkID":
"235079ae3c9401d78b3f8648a0b6793778c55d485c4af3b64326ea1c8c834fa9",
                "EndpointID":
"ae9224852b934080968682a46ca1f170e5f48d3aa50b72ebe4b33ac1ec9b0037",
```

```
            "Gateway": "10.100.10.1",
            "IPAddress": "10.100.0.1",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "DriverOpts": null,
            "DNSNames": [
                "ingnx_network",
                "2162c68a7c8f"
            ]
```

root@ubuntu:~# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e6:4c:3f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 85920sec preferred_lft 85920sec
    inet6 fe80::a00:27ff:fee6:4c3f/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:15:a5:3b:21 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

```
    inet6 fe80::42:15ff:fea5:3b21/64 scope link
       valid_lft forever preferred_lft forever
5: veth29d5831@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether 86:b4:17:4e:05:3d brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::84b4:17ff:fe4e:53d/64 scope link
       valid_lft forever preferred_lft forever
13: veth577e627@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0 state UP group default
    link/ether f2:b6:8a:0a:f5:35 brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::f0b6:8aff:fe0a:f535/64 scope link
       valid_lft forever preferred_lft forever
14: br-235079ae3c94: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:41:b2:f8:a9 brd ff:ff:ff:ff:ff:ff
    inet 10.100.10.1/16 brd 10.100.255.255 scope global br-235079ae3c94
       valid_lft forever preferred_lft forever
    inet6 fe80::42:41ff:feb2:f8a9/64 scope link
       valid_lft forever preferred_lft forever
16: veth4ea64e7@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-235079ae3c94 state UP group default
    link/ether c6:58:a1:61:74:6f brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet6 fe80::c458:a1ff:fe61:746f/64 scope link
       valid_lft forever preferred_lft forever
```

root@ubuntu:~# #172.17.0.1/16 brd 172.17.255.255

root@ubuntu:~# docker container ps

```
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
2162c68a7c8f   nginx:latest   "/docker-entrypoint.…"   37 minutes ago   Up 37 minutes   80/tcp    ingnx_network
4ba7c1ec1d11   nginx:latest   "/docker-entrypoint.…"   8 hours ago      Up 7 hours      80/tcp    nginx_practice1
```

```
63b049d7ae64   nginx     "/docker-entrypoint.…"   8 hours ago   Up 8 hours   80/tcp
nginx_practice
```

Login to Container to check the network

root@ubuntu:~# docker container exec -t -i nginx_practice bash

root@nginx:/# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
```

root@nginx:/# #172.17.0.2/16 brd 172.17.255.255

root@nginx:/# exit

```
exit
```

Check whether the default network is able to communicate with manually created bridge network

root@ubuntu:~# docker container exec -t -i nginx_practice1 bash

root@nginx2:/# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
```

valid_lft forever preferred_lft forever

inet6 ::1/128 scope host

valid_lft forever preferred_lft forever

12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default

link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0

valid_lft forever preferred_lft forever

root@nginx2:/# #172.17.0.3/16 brd 172.17.255.255

root@nginx2:/# exit

exit

root@ubuntu:~# docker container exec -t -i ingnx_network bash

root@2162c68a7c8f:/# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

inet6 ::1/128 scope host

valid_lft forever preferred_lft forever

15: eth0@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default

link/ether 02:42:0a:64:00:01 brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 10.100.0.1/16 brd 10.100.255.255 scope global eth0

valid_lft forever preferred_lft forever

root@2162c68a7c8f:/# ping 170.100.0.0

PING 170.100.0.0 (170.100.0.0) 56(84) bytes of data.

error so need to connect with other network using connect command in host system

root@nginx:/# #172.17.0.2/16 brd 172.17.255.255

root@nginx:/# exit

exit

Container 2

root@ubuntu:~# docker container exec -t -i nginx_practice1 bash

root@nginx2:/# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

inet6 ::1/128 scope host

valid_lft forever preferred_lft forever

12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default

link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0

inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0

valid_lft forever preferred_lft forever

root@nginx2:/# #172.17.0.3/16 brd 172.17.255.255

root@nginx2:/# exit

exit

Container 3

root@ubuntu:~# docker container exec -t -i ingnx_network bash

root@2162c68a7c8f:/# ping 172.17.0.3

PING 172.17.0.3 (172.17.0.3) 56(84) bytes of data.

^C

--- 172.17.0.3 ping statistics ---

5 packets transmitted, 0 received, 100% packet loss, time 4103ms

Host System

root@ubuntu:~# docker network ls

```
NETWORK ID     NAME            DRIVER    SCOPE
235079ae3c94   Dhileep_network   bridge    local
274daf1d488d   bridge          bridge    local
edf92f2985e3   host            host      local
af1449d68ae4   none            null      local
```

root@ubuntu:~# docker container ps

```
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
2162c68a7c8f   nginx:latest   "/docker-entrypoint.…"   42 minutes ago   Up 42 minutes   80/tcp    ingnx_network
4ba7c1ec1d11   nginx:latest   "/docker-entrypoint.…"   8 hours ago      Up 7 hours      80/tcp    nginx_practice1
63b049d7ae64   nginx          "/docker-entrypoint.…"   8 hours ago      Up 8 hours      80/tcp    nginx_practice
```

Creating connection between two network:

root@ubuntu:~# docker network connect Dhileep_network nginx_practice

root@ubuntu:~# docker container exec -t -i nginx_practice bash

root@nginx:/# ip a

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
4: eth0@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
```

```
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
17: eth1@if18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default
    link/ether 02:42:0a:64:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.100.0.2/16 brd 10.100.255.255 scope global eth1
       valid_lft forever preferred_lft forever
```

root@nginx:/# ping 10.100.0.1

```
PING 10.100.0.1 (10.100.0.1) 56(84) bytes of data.
64 bytes from 10.100.0.1: icmp_seq=1 ttl=64 time=0.976 ms
64 bytes from 10.100.0.1: icmp_seq=2 ttl=64 time=0.061 ms
^Z
[1]+  Stopped                 ping 10.100.0.1
```

Successfully connected with the manually created bridge network

****

Type 2: Host network

****