DOCKER NETWORK IPVLAN L3 AND L2

Docker IPvlan L3 Mode (Layer 3)

In L3 mode, containers are placed on a Layer 3 network, which means they need to be routed to communicate with the host or other containers. In this mode, containers communicate by routing traffic through a router rather than simply switching traffic at Layer 2.

Features:

Containers are placed in different Layer 3 networks (subnets), so communication between them requires routing.

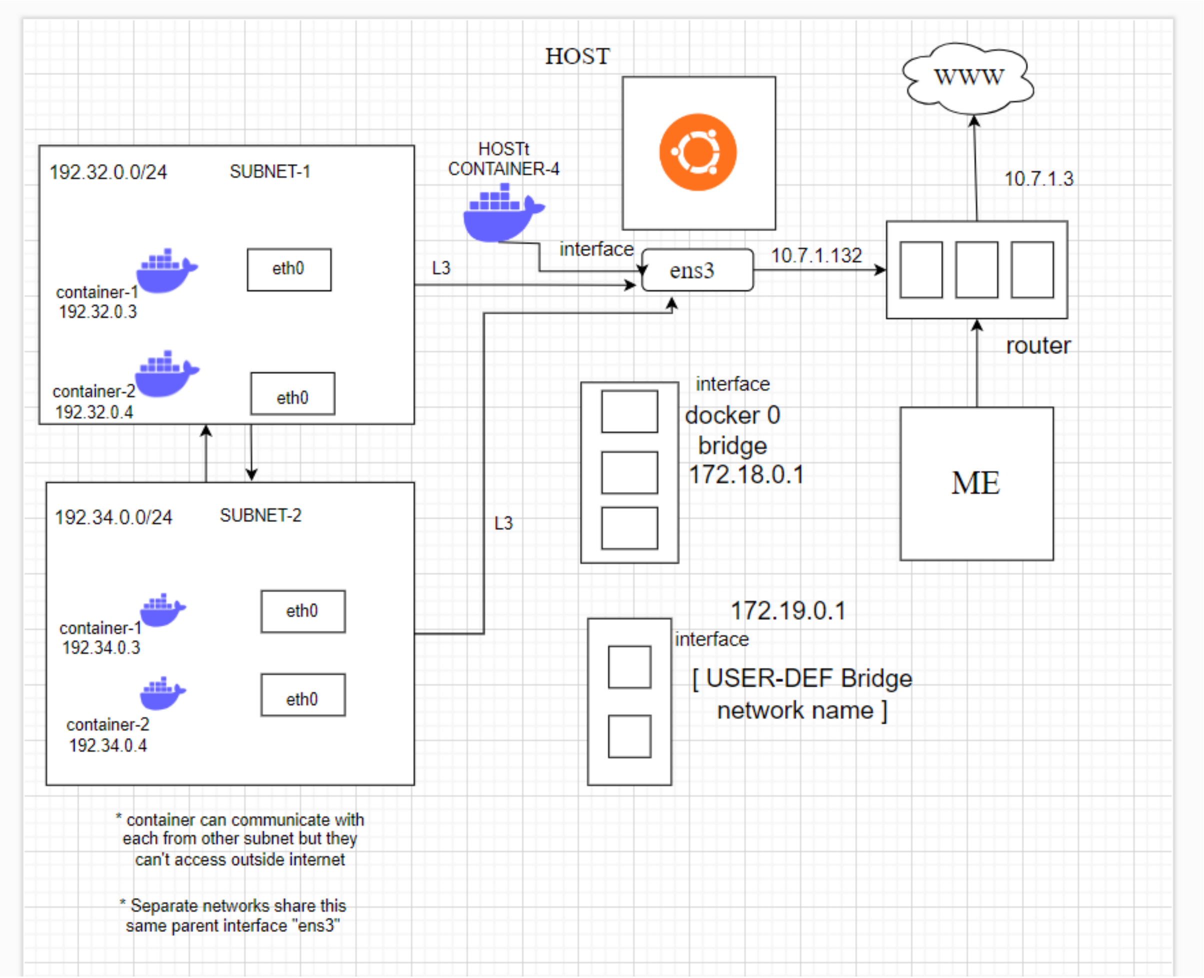
There is no Layer 2 broadcast domain between containers; they are isolated at Layer 3 and need routing for inter-container communication.

Suitable when you want containers to operate in different IP subnets or when you need more granular control over routing between containers.

Use Case:

Ideal when you need to implement routing between containers and host interfaces.

Useful in larger-scale, routed network environments, where multiple subnets or complex routing policies are needed.

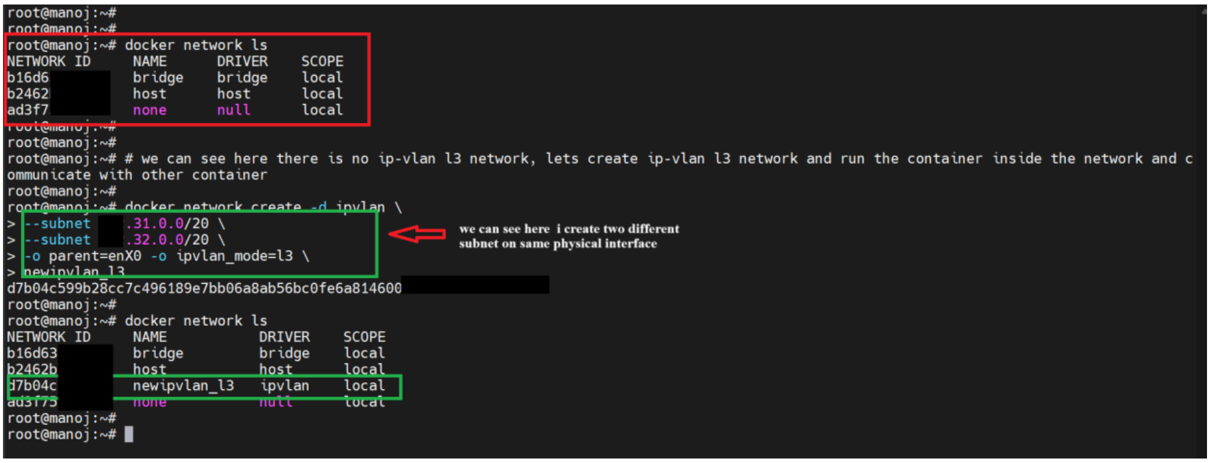


* container can communicate with each from other subnet but they can't access outside internet

* Separate networks share this same parent interface "ens3"

DOCKER NETWORK IPVLAN L3 AND L2

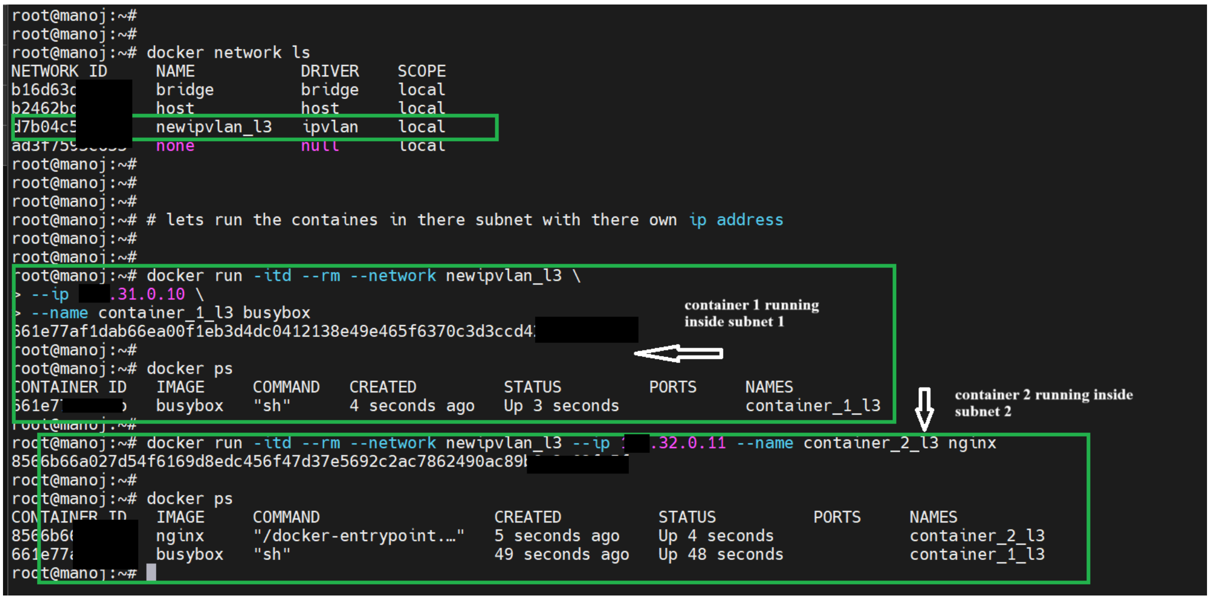
Create an IPvlan L3 network:



```
root@manoj:~#
root@manoj:~#
root@manoj:~# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
b16d6[redacted]     bridge    bridge    local
b2462[redacted]     host      host      local
ad3f7[redacted]     none      null      local
root@manoj:~#
root@manoj:~#
root@manoj:~# # we can see here there is no ip-vlan l3 network, lets create ip-vlan l3 network and run the container inside the network and c
ommunicate with other container
root@manoj:~#
root@manoj:~# docker network create -d ipvlan \
> --subnet [redacted].31.0.0/20 \
> --subnet [redacted].32.0.0/20 \
> -o parent=enX0 -o ipvlan_mode=l3 \
> newipvlan_l3
d7b04c599b28cc7c496189e7bb06a8ab56bc0fe6a814600[redacted]
root@manoj:~#
root@manoj:~# docker network ls
NETWORK ID     NAME           DRIVER    SCOPE
b16d63[redacted]     bridge         bridge    local
b2462b[redacted]     host           host      local
d7b04c[redacted]     newipvlan_l3   ipvlan    local
ad3f75[redacted]     none           null      local
root@manoj:~#
root@manoj:~# 
```

we can see here i create two different subnet on same physical interface

Run a container in the IPvlan L3 network:

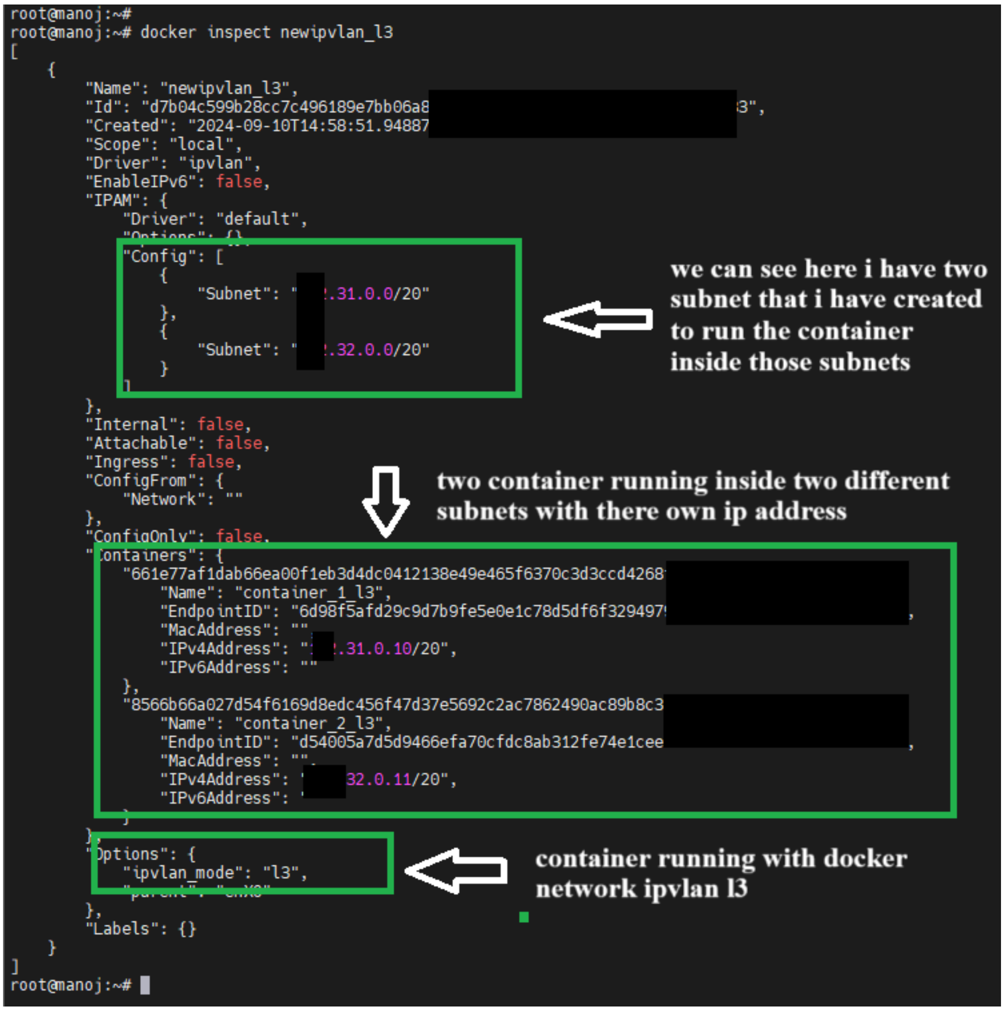


```
root@manoj:~#
root@manoj:~#
root@manoj:~# docker network ls
NETWORK ID     NAME           DRIVER    SCOPE
b16d63c[redacted]   bridge         bridge    local
b2462bc[redacted]   host           host      local
d7b04c5[redacted]   newipvlan_l3   ipvlan    local
ad3f75[redacted]    none           null      local
root@manoj:~#
root@manoj:~#
root@manoj:~#
root@manoj:~# # lets run the containes in there subnet with there own ip address
root@manoj:~#
root@manoj:~#
root@manoj:~# docker run -itd --rm --network newipvlan_l3 \
> --ip [redacted].31.0.10 \
> --name container_1_l3 busybox
661e77af1dab66ea00f1eb3d4dc0412138e49e465f6370c3d3ccd4[redacted]
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED          STATUS         PORTS     NAMES
661e7[redacted]   busybox   "sh"      4 seconds ago    Up 3 seconds             container_1_l3
root@manoj:~#
root@manoj:~# docker run -itd --rm --network newipvlan_l3 --ip [redacted].32.0.11 --name container_2_l3 nginx
8566b66a027d54f6169d8edc456f47d37e5692c2ac7862490ac89[redacted]
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
8566b6[redacted]   nginx     "/docker-entrypoint.…"   5 seconds ago    Up 4 seconds              container_2_l3
661e77[redacted]   busybox   "sh"                     49 seconds ago   Up 48 seconds             container_1_l3
root@manoj:~# 
```

container 1 running inside subnet 1

container 2 running inside subnet 2

DOCKER NETWORK IPVLAN L3 AND L2



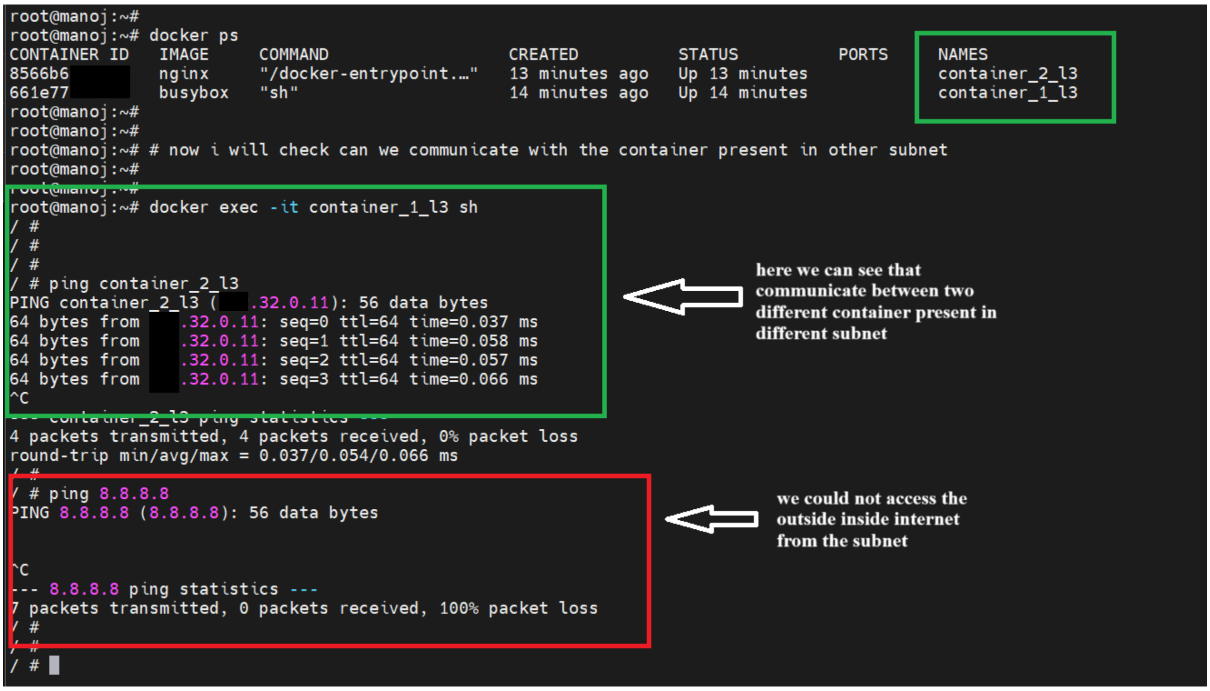
```
root@manoj:~#
root@manoj:~# docker inspect newipvlan_l3
[
    {
        "Name": "newipvlan_l3",
        "Id": "d7b04c599b28cc7c496189e7bb06a8[redacted]3",
        "Created": "2024-09-10T14:58:51.94887
        "Scope": "local",
        "Driver": "ipvlan",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "[redacted].31.0.0/20"
                },
                {
                    "Subnet": ".32.0.0/20"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "661e77af1dab66ea00f1eb3d4dc0412138e49e465f6370c3d3ccd4268[redacted]
                "Name": "container_1_l3",
                "EndpointID": "6d98f5afd29c9d7b9fe5e0e1c78d5df6f329497,
                "MacAddress": "",
                "IPv4Address": "[redacted].31.0.10/20",
                "IPv6Address": ""
            },
            "8566b66a027d54f6169d8edc456f47d37e5692c2ac7862490ac89b8c3[redacted]
                "Name": "container_2_l3",
                "EndpointID": "d54005a7d5d9466efa70cfdc8ab312fe74e1cee,
                "MacAddress": "",
                "IPv4Address": '[redacted]32.0.11/20",
                "IPv6Address": '
            }
        },
        "Options": {
            "ipvlan_mode": "l3",
            "parent": "enX0"
        },
        "Labels": {}
    }
]
root@manoj:~#
```

we can see here i have two subnet that i have created to run the container inside those subnets

two container running inside two different subnets with there own ip address

container running with docker network ipvlan l3

We can see container are isolated from the outside network. But we can communicate with in the same network.



```
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
8566b6[redacted]   nginx     "/docker-entrypoint.…"   13 minutes ago   Up 13 minutes             container_2_l3
661e77         busybox   "sh"                     14 minutes ago   Up 14 minutes             container_1_l3
root@manoj:~#
root@manoj:~#
root@manoj:~# # now i will check can we communicate with the container present in other subnet
root@manoj:~#
root@manoj:~#
root@manoj:~# docker exec -it container_1_l3 sh
/ #
/ #
/ #
/ # ping container_2_l3
PING container_2_l3 ([redacted].32.0.11): 56 data bytes
64 bytes from [redacted].32.0.11: seq=0 ttl=64 time=0.037 ms
64 bytes from .32.0.11: seq=1 ttl=64 time=0.058 ms
64 bytes from .32.0.11: seq=2 ttl=64 time=0.057 ms
64 bytes from .32.0.11: seq=3 ttl=64 time=0.066 ms
^C
--- container_2_l3 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.037/0.054/0.066 ms
/ #
/ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes


^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss
/ #
/ #
/ #
```

here we can see that communicate between two different container present in different subnet

we could not access the outside inside internet from the subnet

DOCKER NETWORK IPVLAN L3 AND L2

Docker IPvlan L2 Mode (Layer 2)

In L2 mode, the containers operate like traditional Ethernet devices on the same network as the parent interface. Each container gets its own IP address and behaves like it's directly connected to the same physical or virtual network as the host.

Features:

Containers appear as devices on the same Layer 2 broadcast domain (same VLAN or subnet).

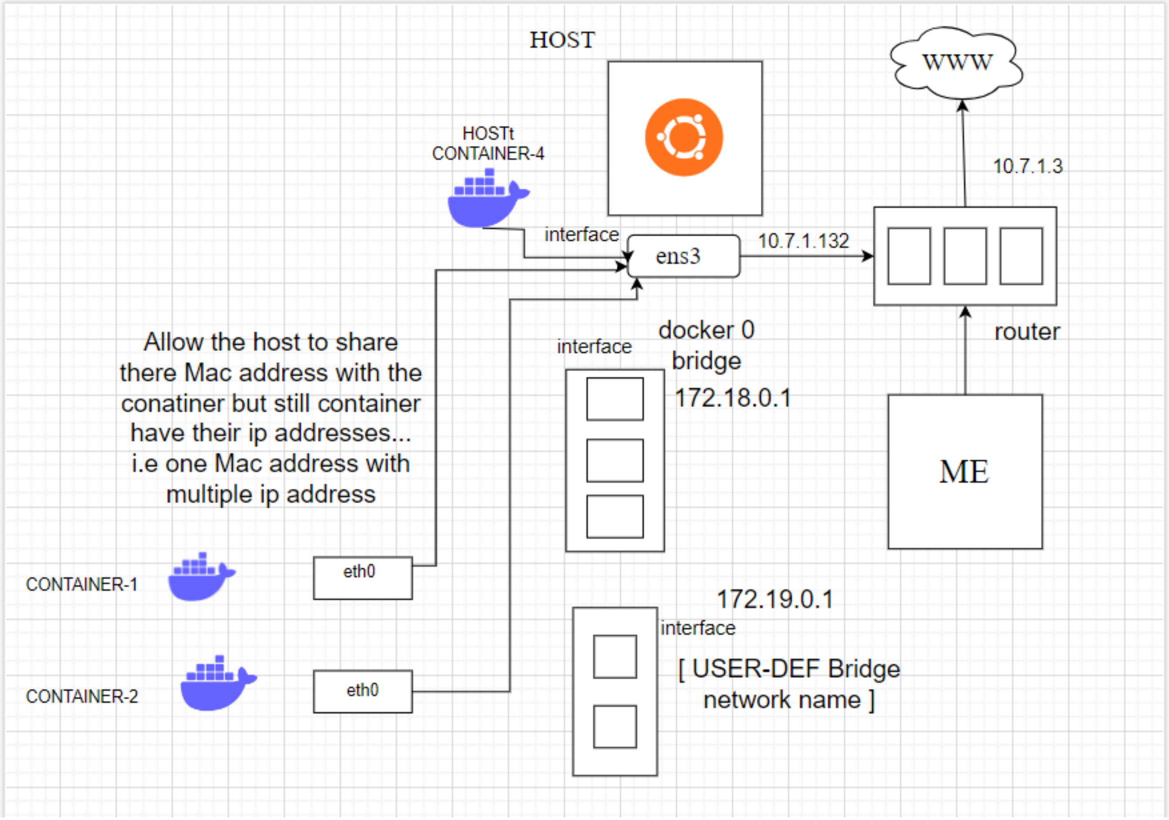
Containers share the MAC address of the host's parent network interface.

Suitable when you want containers to operate on the same LAN or VLAN as the host, without needing to deal with routing.

Use Case:

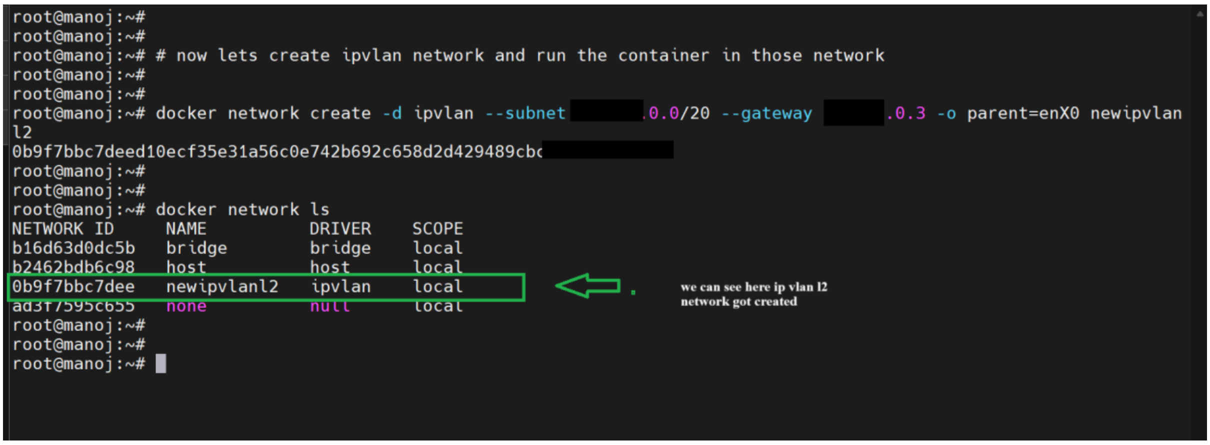
Ideal when you want containers to be part of the same local network (same subnet) as the host.

Good for environments where you want minimal network overhead and direct connectivity.


HOST
WWW
HOSTt CONTAINER-4
10.7.1.3
interface
ens3
10.7.1.132
router
docker 0 bridge
interface
172.18.0.1
Allow the host to share there Mac address with the conatiner but still container have their ip addresses... i.e one Mac address with multiple ip address
ME
CONTAINER-1
eth0
172.19.0.1
interface
[USER-DEF Bridge network name]
CONTAINER-2
eth0

DOCKER NETWORK IPVLAN L3 AND L2

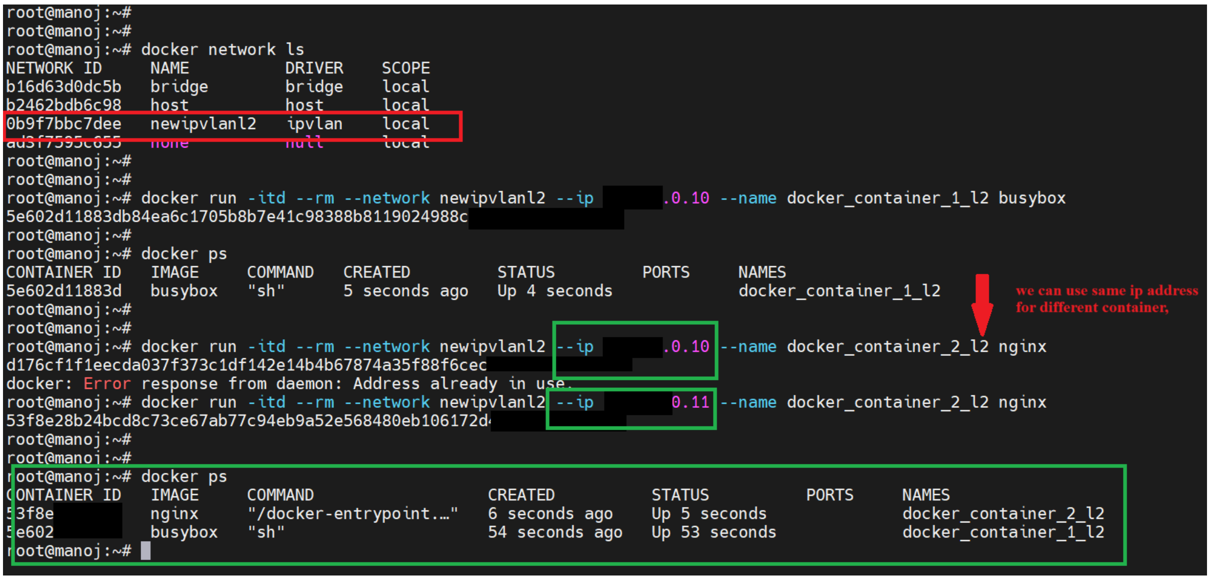
Create an IPvlan L2 network:



```
root@manoj:~#
root@manoj:~#
root@manoj:~# # now lets create ipvlan network and run the container in those network
root@manoj:~#
root@manoj:~#
root@manoj:~# docker network create -d ipvlan --subnet [redacted].0.0/20 --gateway [redacted].0.3 -o parent=enX0 newipvlan
l2
0b9f7bbc7deed10ecf35e31a56c0e742b692c658d2d429489cbc[redacted]
root@manoj:~#
root@manoj:~#
root@manoj:~# docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
b16d63d0dc5b   bridge        bridge    local
b2462bdb6c98   host          host      local
0b9f7bbc7dee   newipvlanl2   ipvlan    local
ad3f7595c655   none          null      local
root@manoj:~#
root@manoj:~#
root@manoj:~# ▌
```

we can see here ip vlan l2 network got created

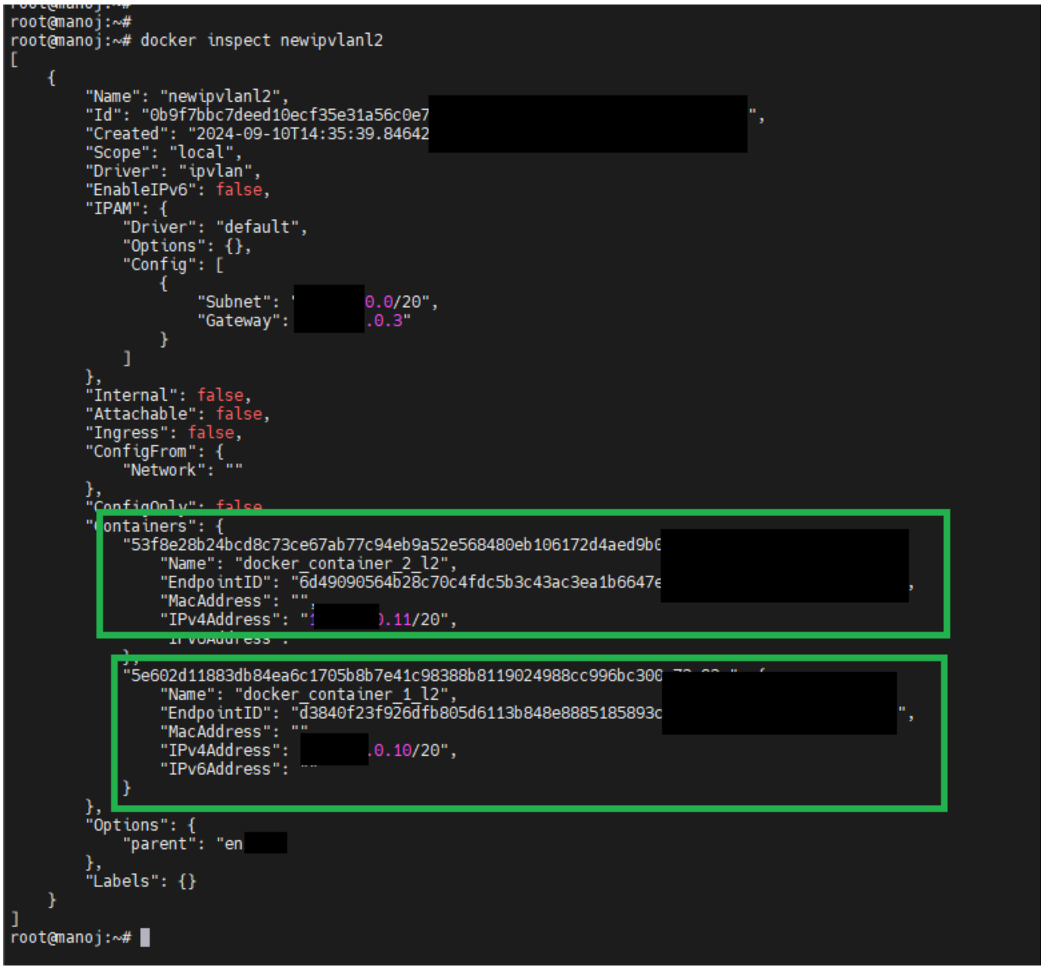
Run a container in the IPvlan L2 network:



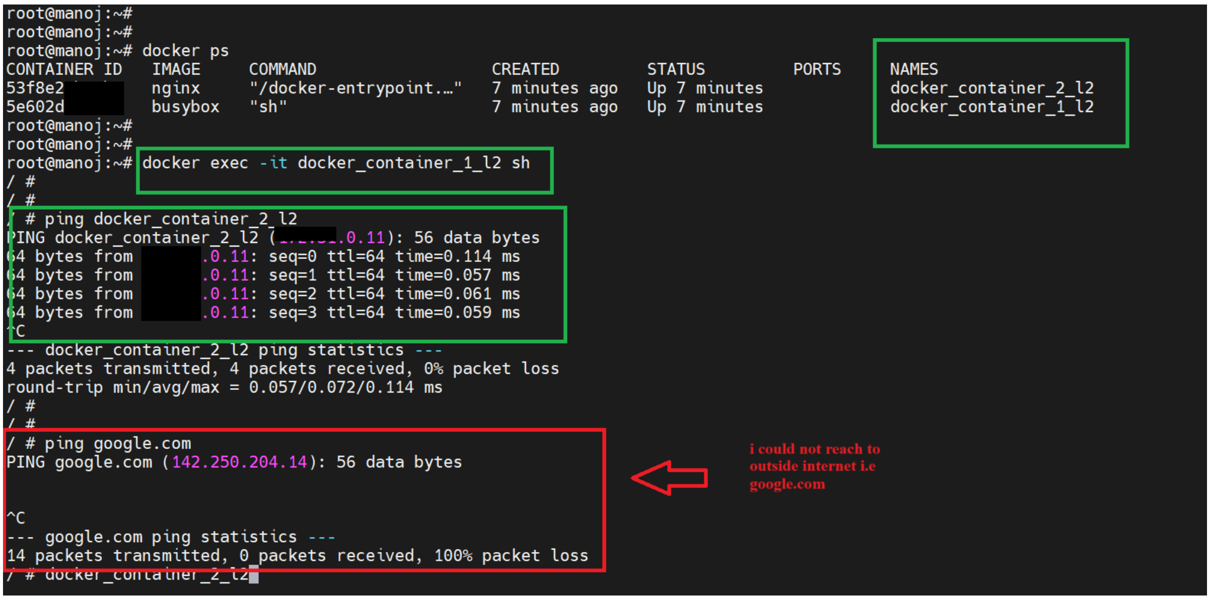
```
root@manoj:~#
root@manoj:~#
root@manoj:~# docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
b16d63d0dc5b   bridge        bridge    local
b2462bdb6c98   host          host      local
0b9f7bbc7dee   newipvlanl2   ipvlan    local
ad3f7595c655   none          null      local
root@manoj:~#
root@manoj:~#
root@manoj:~# docker run -itd --rm --network newipvlanl2 --ip [redacted].0.10 --name docker_container_1_l2 busybox
5e602d11883db84ea6c1705b8b7e41c98388b8119024988c[redacted]
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS         PORTS     NAMES
5e602d11883d   busybox   "sh"      5 seconds ago   Up 4 seconds             docker_container_1_l2
root@manoj:~#
root@manoj:~#
root@manoj:~# docker run -itd --rm --network newipvlanl2 --ip [redacted].0.10 --name docker_container_2_l2 nginx
d176cf1f1eecda037f373c1df142e14b4b67874a35f88f6cec[redacted]
docker: Error response from daemon: Address already in use.
root@manoj:~# docker run -itd --rm --network newipvlanl2 --ip [redacted]0.11 --name docker_container_2_l2 nginx
53f8e28b24bcd8c73ce67ab77c94eb9a52e568480eb106172d[redacted]
root@manoj:~#
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
53f8e[redacted]   nginx     "/docker-entrypoint.…"   6 seconds ago    Up 5 seconds              docker_container_2_l2
5e602[redacted]   busybox   "sh"                     54 seconds ago   Up 53 seconds             docker_container_1_l2
root@manoj:~# ▌
```

we can use same ip address for different container,

DOCKER NETWORK IPVLAN L3 AND L2



```
root@manoj:~#
root@manoj:~# docker inspect newipvlanl2
[
    {
        "Name": "newipvlanl2",
        "Id": "0b9f7bbc7deed10ecf35e31a56c0e7[redacted]",
        "Created": "2024-09-10T14:35:39.84642
        "Scope": "local",
        "Driver": "ipvlan",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "[redacted]0.0/20",
                    "Gateway": .0.3"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "53f8e28b24bcd8c73ce67ab77c94eb9a52e568480eb106172d4aed9b[redacted]
                "Name": "docker_container_2_l2",
                "EndpointID": "6d49090564b28c70c4fdc5b3c43ac3ea1b6647e
                "MacAddress": "",
                "IPv4Address": "[redacted].11/20",
                "IPv6Address": ""
            },
            "5e602d11883db84ea6c1705b8b7e41c98388b8119024988cc996bc300
                "Name": "docker_container_1_l2",
                "EndpointID": "d3840f23f926dfb805d6113b848e8885185893c[redacted]",
                "MacAddress": "",
                "IPv4Address": "[redacted].0.10/20",
                "IPv6Address": ""
            }
        },
        "Options": {
            "parent": "en[redacted]
        },
        "Labels": {}
    }
]
root@manoj:~#
```



```
root@manoj:~#
root@manoj:~#
root@manoj:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
53f8e2[redacted]   nginx     "/docker-entrypoint.…"   7 minutes ago   Up 7 minutes             docker_container_2_l2
5e602d[redacted]   busybox   "sh"                     7 minutes ago   Up 7 minutes             docker_container_1_l2
root@manoj:~#
root@manoj:~#
root@manoj:~# docker exec -it docker_container_1_l2 sh
/ #
/ #
/ # ping docker_container_2_l2
PING docker_container_2_l2 ([redacted].0.11): 56 data bytes
64 bytes from [redacted].0.11: seq=0 ttl=64 time=0.114 ms
64 bytes from [redacted].0.11: seq=1 ttl=64 time=0.057 ms
64 bytes from [redacted].0.11: seq=2 ttl=64 time=0.061 ms
64 bytes from [redacted].0.11: seq=3 ttl=64 time=0.059 ms
^C
--- docker_container_2_l2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.057/0.072/0.114 ms
/ #
/ #
/ # ping google.com
PING google.com (142.250.204.14): 56 data bytes


^C
--- google.com ping statistics ---
14 packets transmitted, 0 packets received, 100% packet loss
/ # docker_container_2_l2
```

i could not reach to outside internet i.e google.com

DOCKER NETWORK IPVLAN L3 AND L2

Key Differences Between IPvlan L2 and L3:

Feature	IPvlan L2 (Layer 2)	IPvlan L3 (Layer 3)
Communication	Works on Layer 2, same broadcast domain as the host	Works on Layer 3, requires routing between containers
MAC Address	Uses the host's MAC address	Does not require MAC addresses for containers
Network	Containers operate in the same subnet	Containers can be in different subnets
Use Case	Simple, direct access to the same LAN	More control over routing between containers
Complexity	Easier to set up, fewer routing concerns	Requires routing configuration

When to Use Each Mode:

- Use IPvlan L2 mode when you want containers to be on the same Layer 2 network (like a local LAN or VLAN) and you need them to have minimal routing overhead.
- Use IPvlan L3 mode when you want to isolate containers at Layer 3 and implement routing between subnets or across different network zones.