



Docker Network

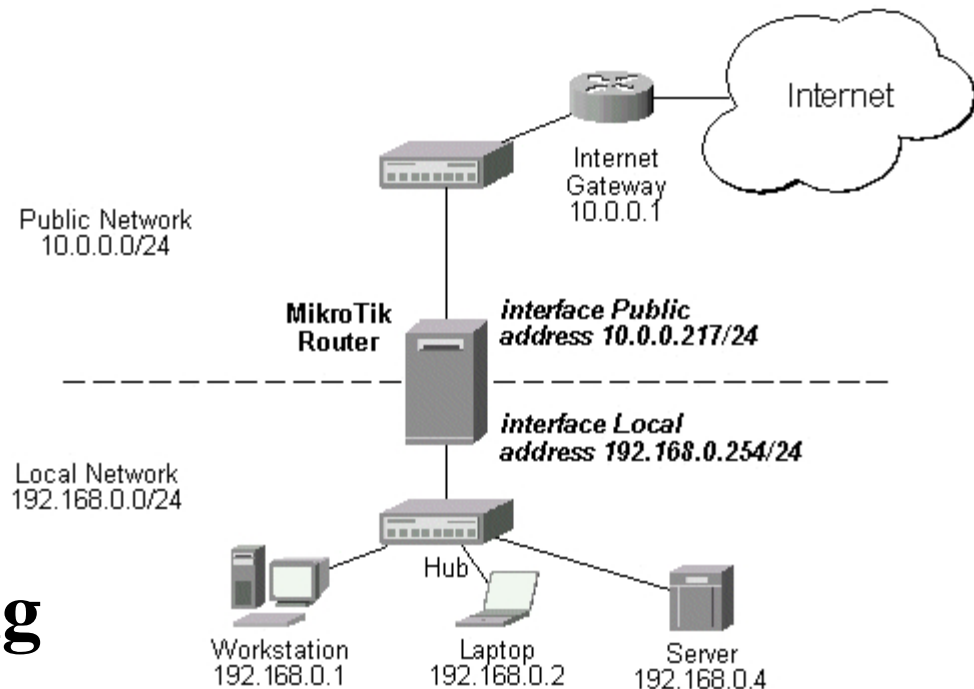


Agenda

- **Khái niệm về mạng Private và Public trên Container**
- **Quản lý các Virtual Network**
- **DNS trên Docker**

Một số thuật ngữ Network

- NIC / Interface
- IP / Subnet
- Protocol / Port
- DHCP
- NAT / Port Forwarding
- DNS



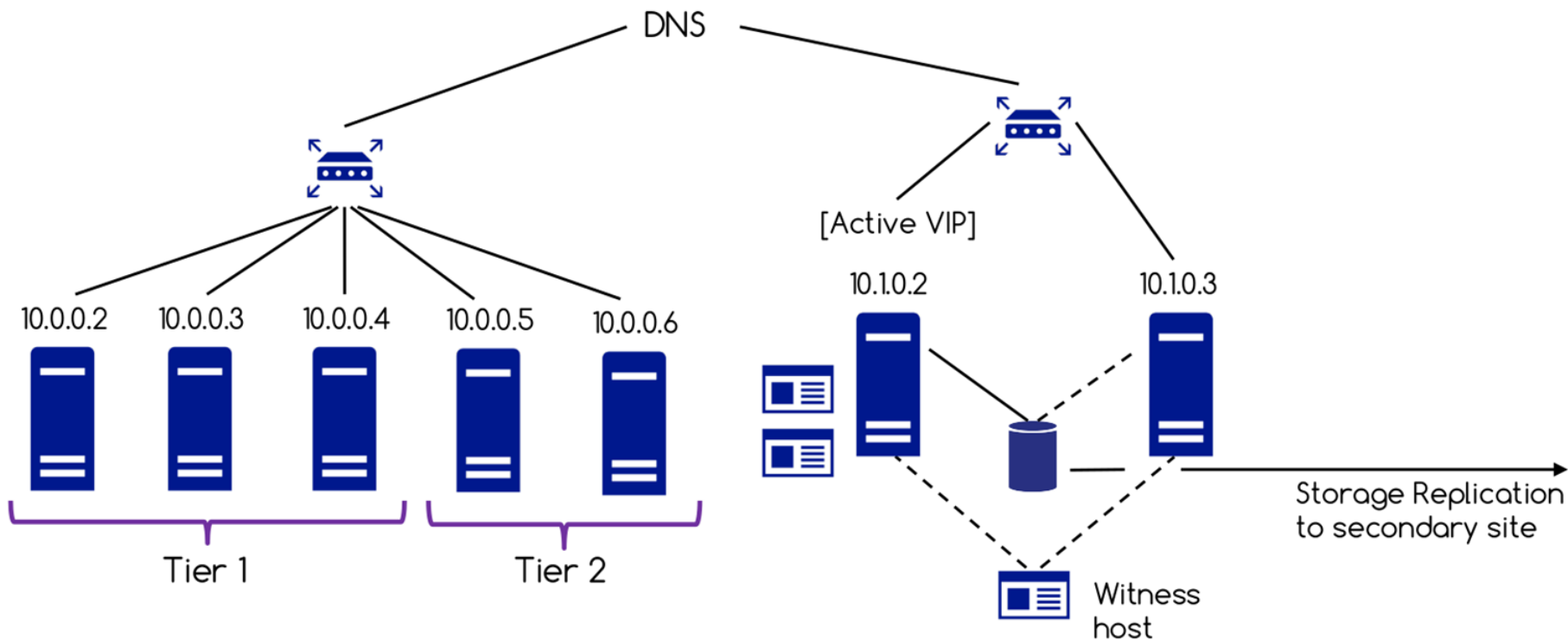


Tại sao chúng ta cần Container Networking ?

- Container cần giao tiếp với bên ngoài
- Kết nối Container từ bên ngoài để sử dụng các dịch vụ mà Container cung cấp
- Cho phép Container giao tiếp với host
- Giao tiếp giữa các Container trong cùng hoặc khác host
- Tự động khám phá các dịch vụ được cung cấp bởi container
- Load balance giữa các containers trong một dịch vụ

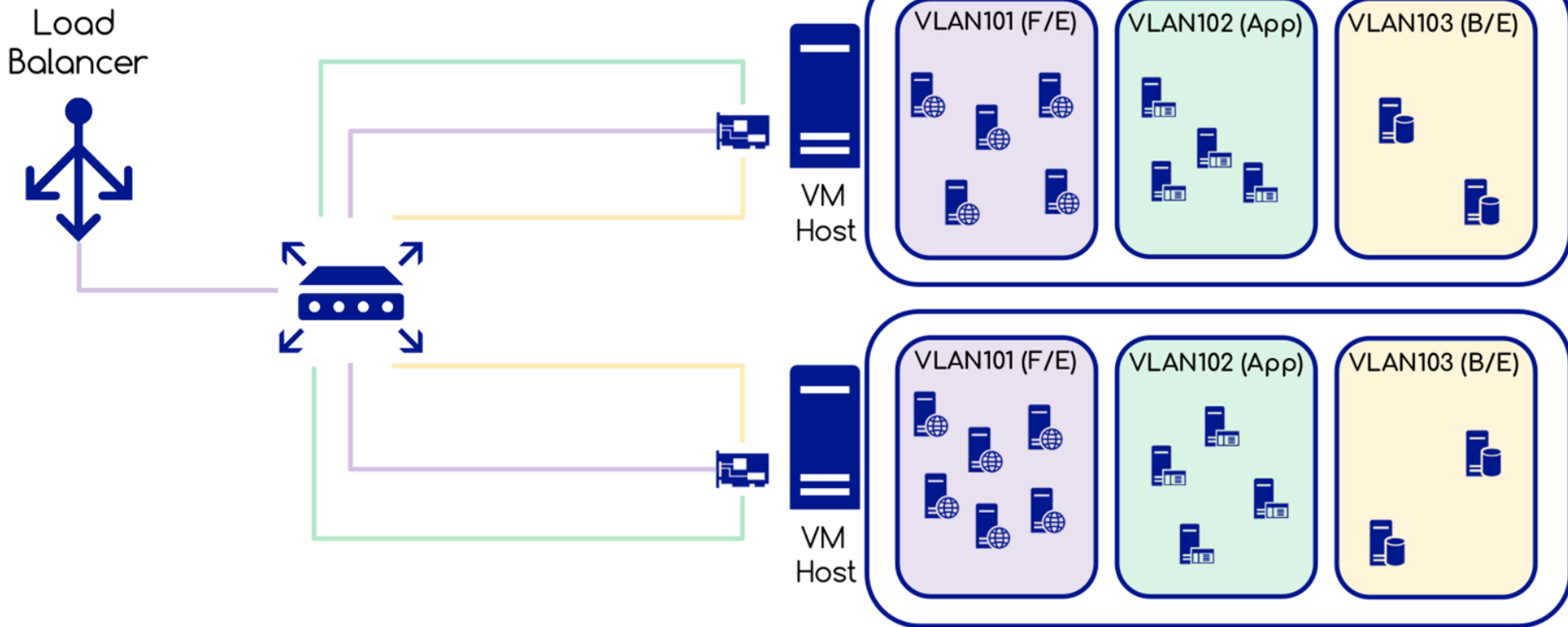


Physically hosted network





Virtual Machine network

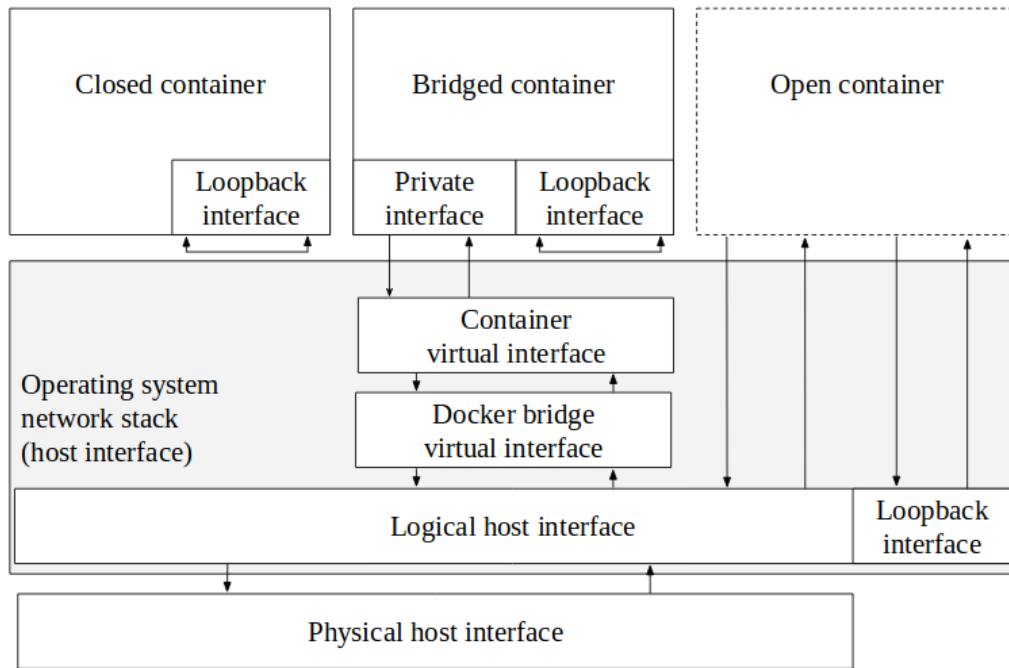




Docker network trên cùng host

Có 3 loại:

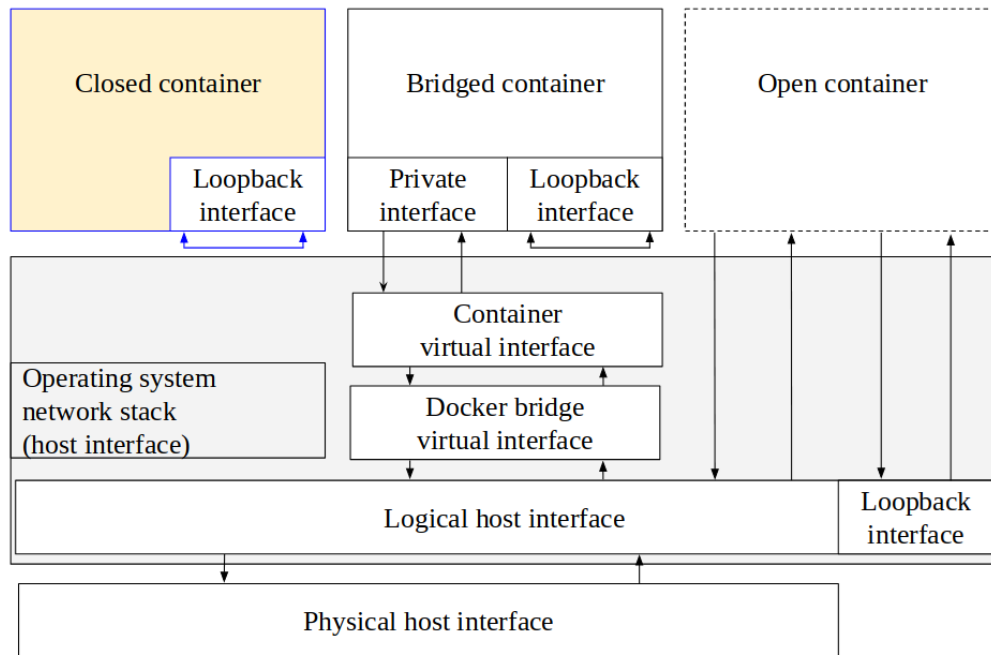
- Closed Container
- Bridged Container
- Open Container





Closed Container

- Không cho phép giao tiếp ra bên ngoài container
- Chỉ có thể giao tiếp với chính nó

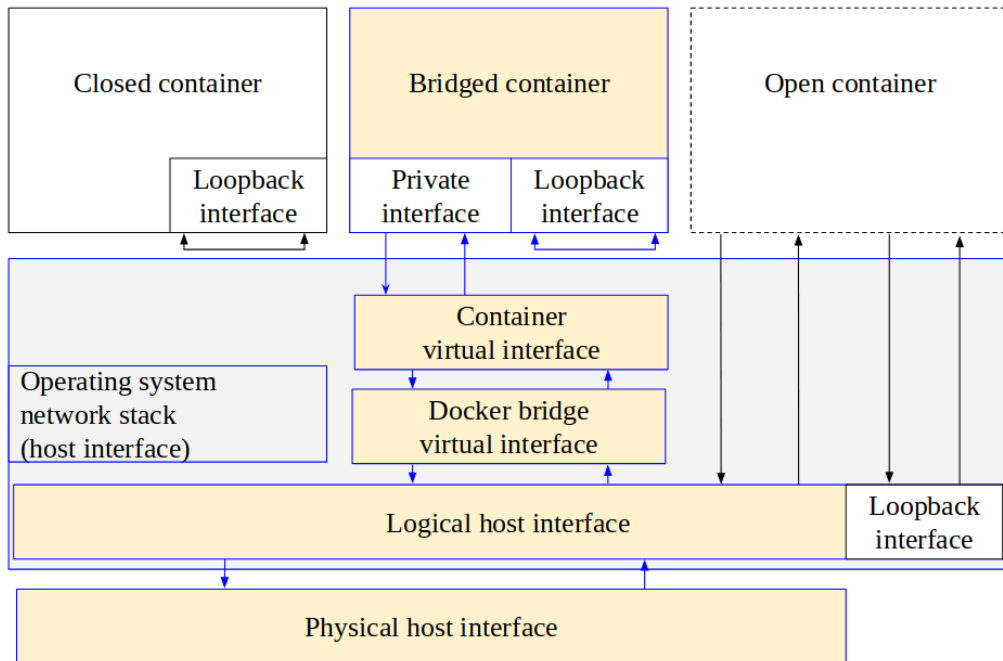


```
docker container run --net none alpine:latest ip addr
```



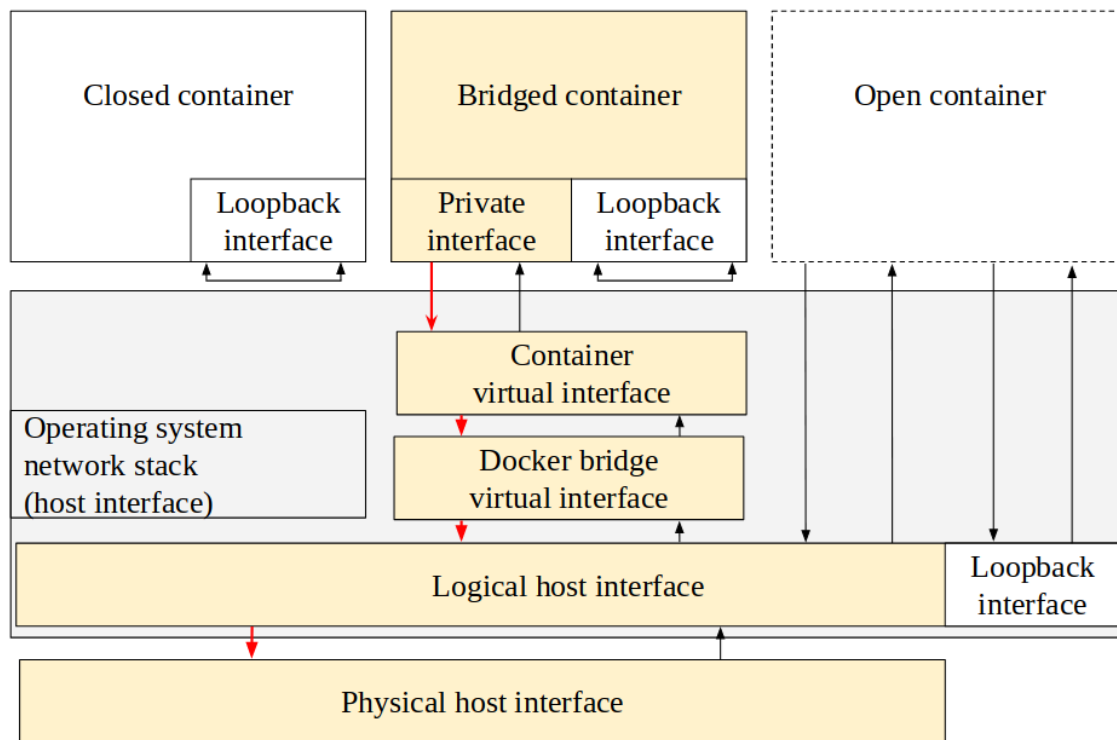

Bridged Container

- Một loopback interface
- Một private interface kết nối với Docker bridge (Docker0) interface để kết nối ra bên ngoài





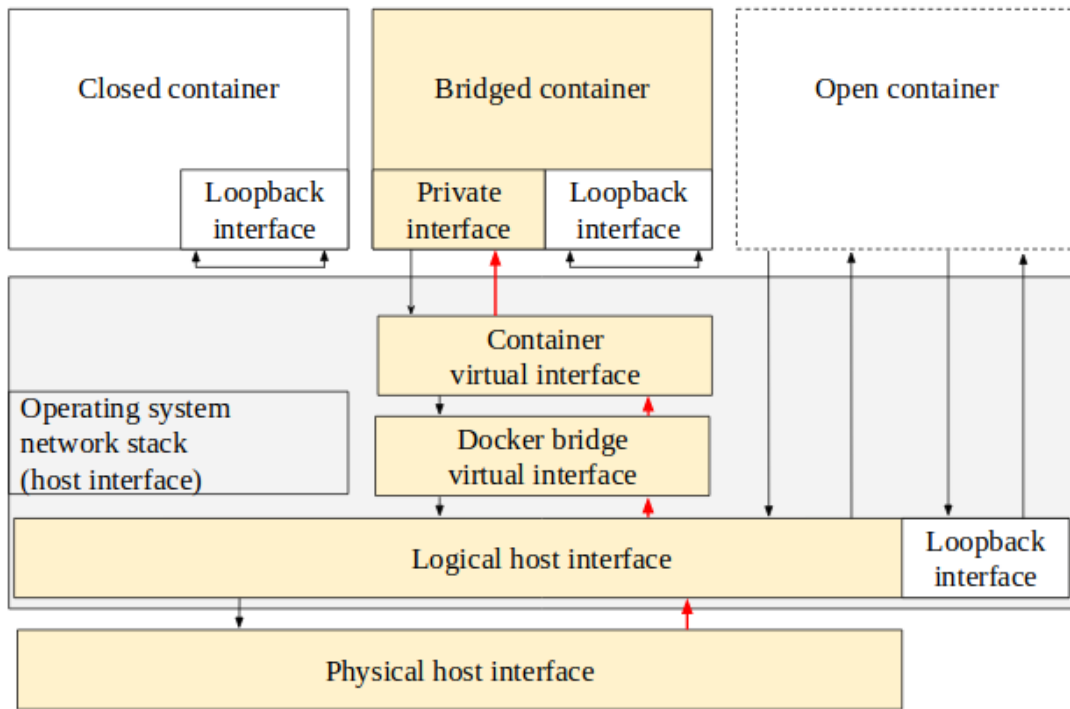
Bridged Container



```
docker run --rm alpine:latest ping -w 2 8.8.8.8
```



Bridged Container

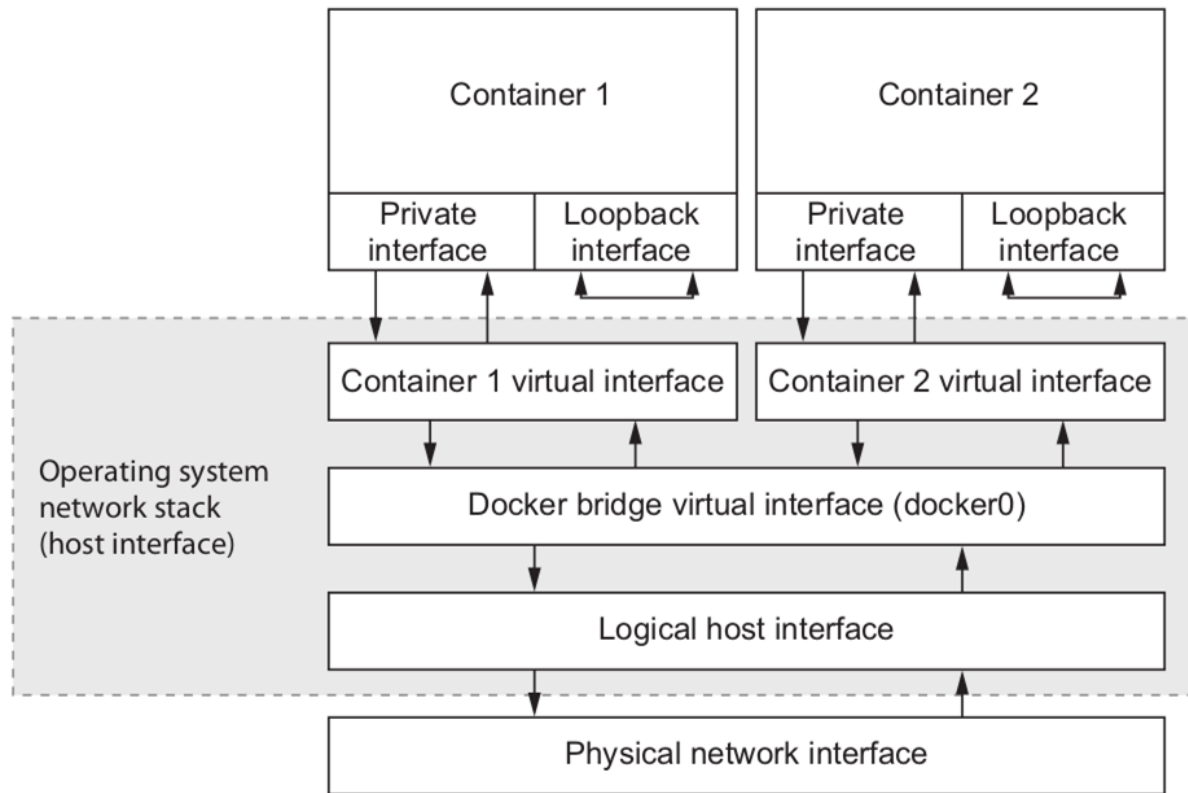


```
docker container run -p 3333:3333 ...
```



Liên kết giữa các container

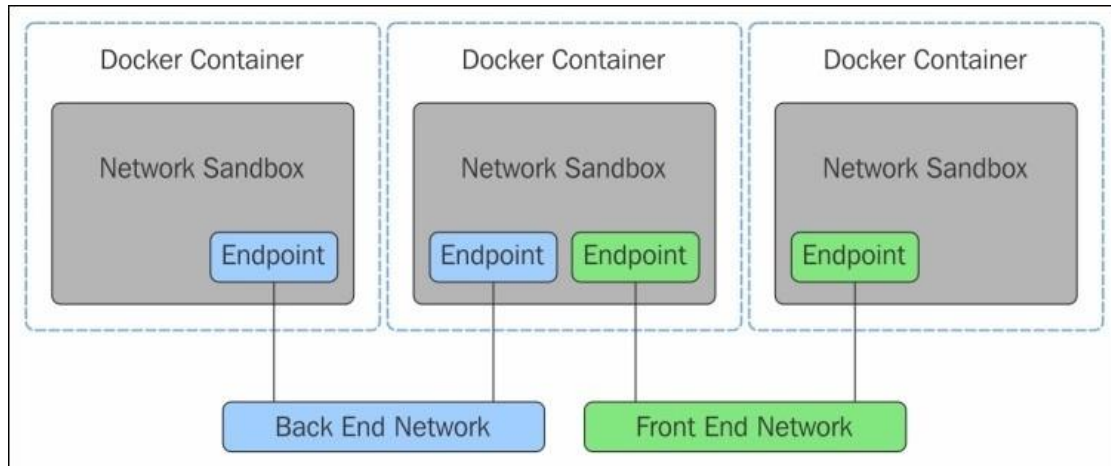
Tất cả các container trong cùng một mạng có thể giao tiếp được với nhau





Liên kết giữa các container

- Container có thể được gán vào một hay nhiều lớp mạng
- Best practice: tạo một mạng riêng cho từng app
 - + Network "my_web_app" cho mysql và php/apache containers
 - + Network "my_api" cho mongo và nodejs containers



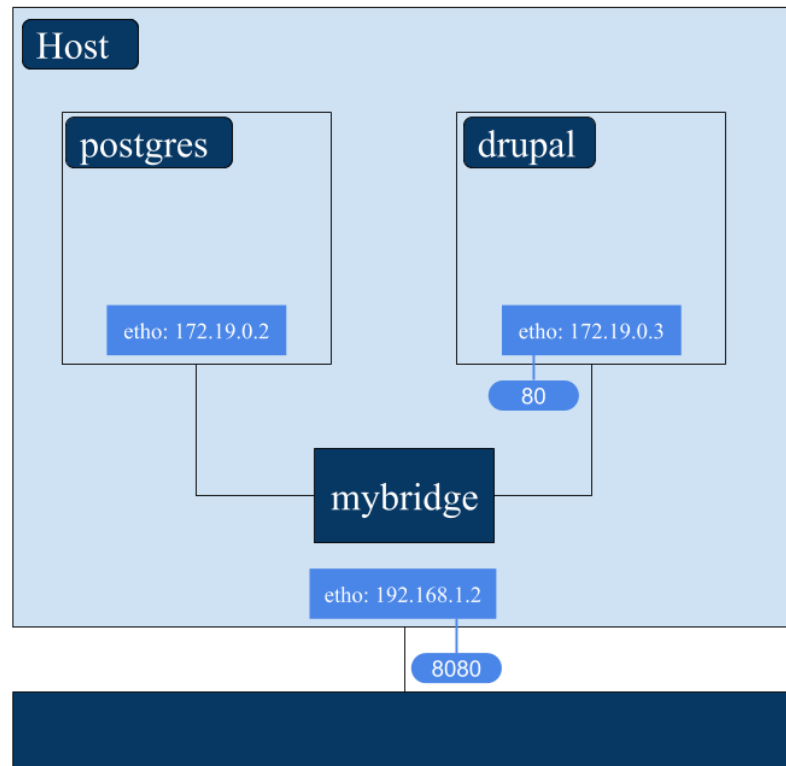


Thực hành: cách truyền thông tin trong Docker Network

```
docker network create -d bridge mybridge
```

```
docker container run -d \  
  --net mybridge \  
  --name postgres \  
  -e POSTGRES_PASSWORD=123qwe \  
  -e POSTGRES_USER=postgres postgres
```

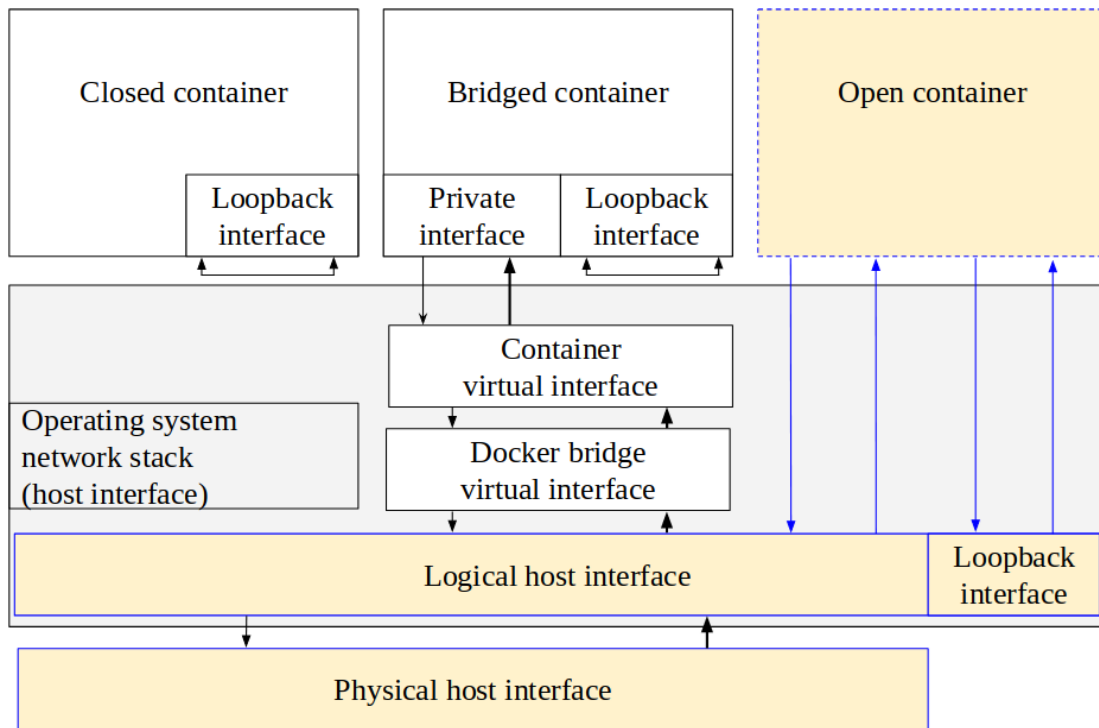
```
docker container run -d \  
  --net mybridge \  
  --name drupal \  
  --publish 8080:80 drupal
```





Open Container

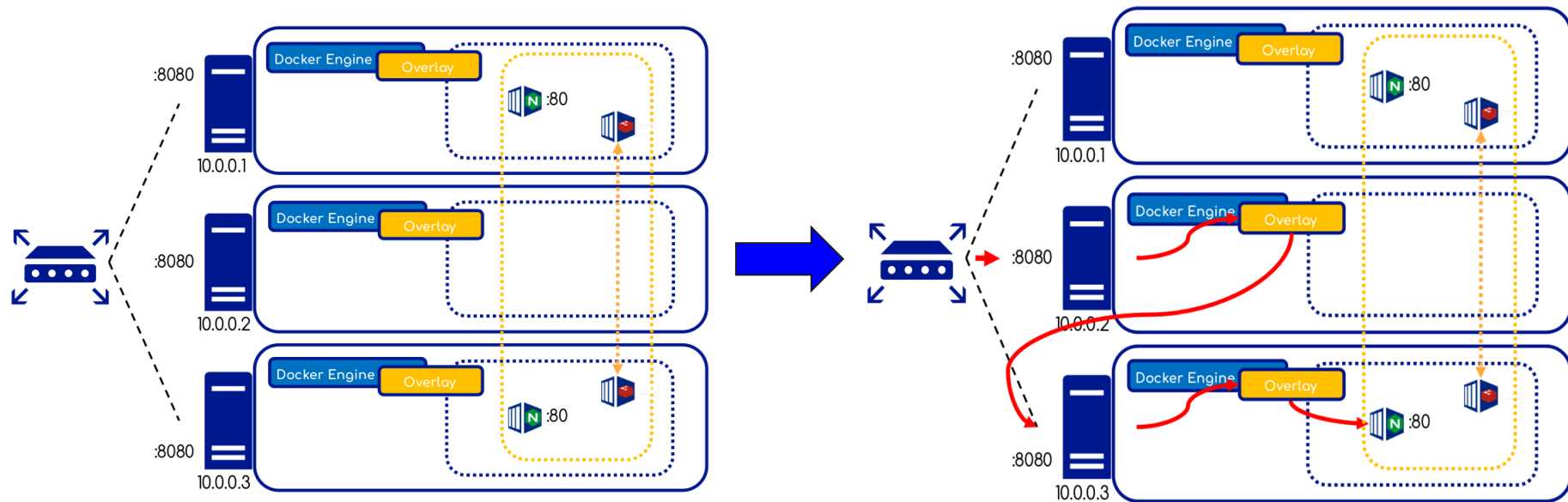
Không sử dụng
network container mà
sử dụng host network



```
docker run --net host alpine:latest ip addr
```

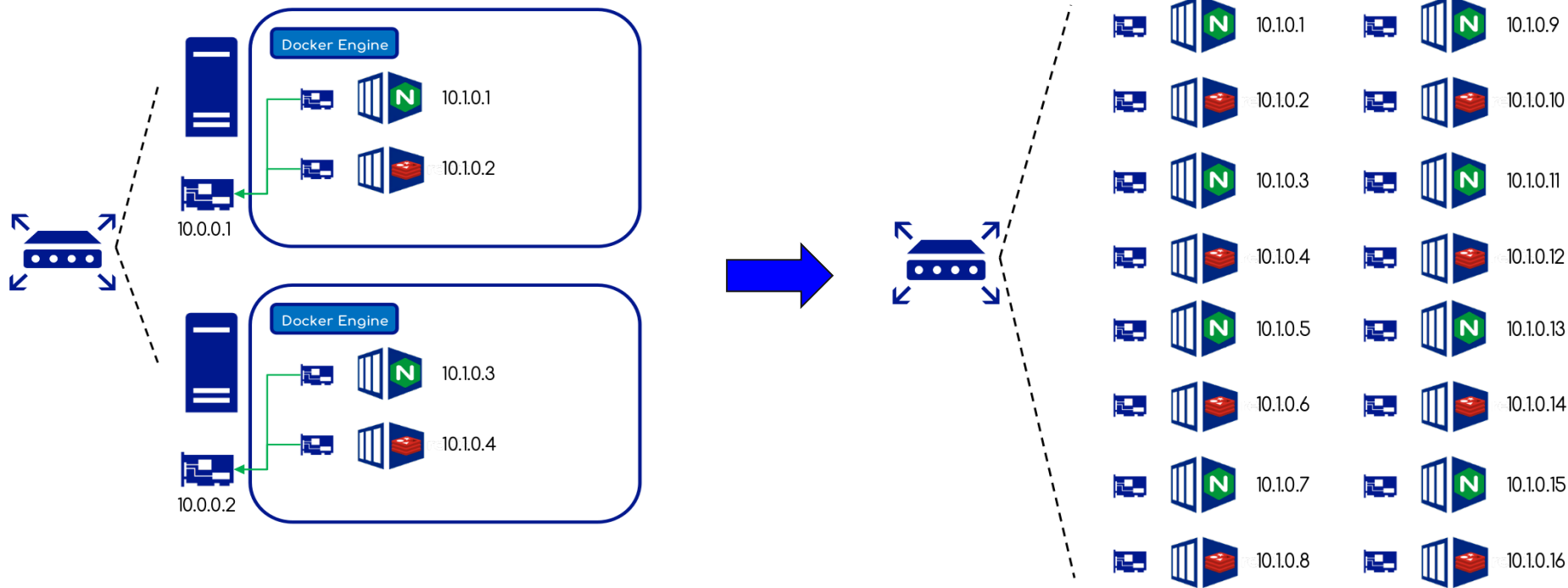


Multi host networking - Overlay



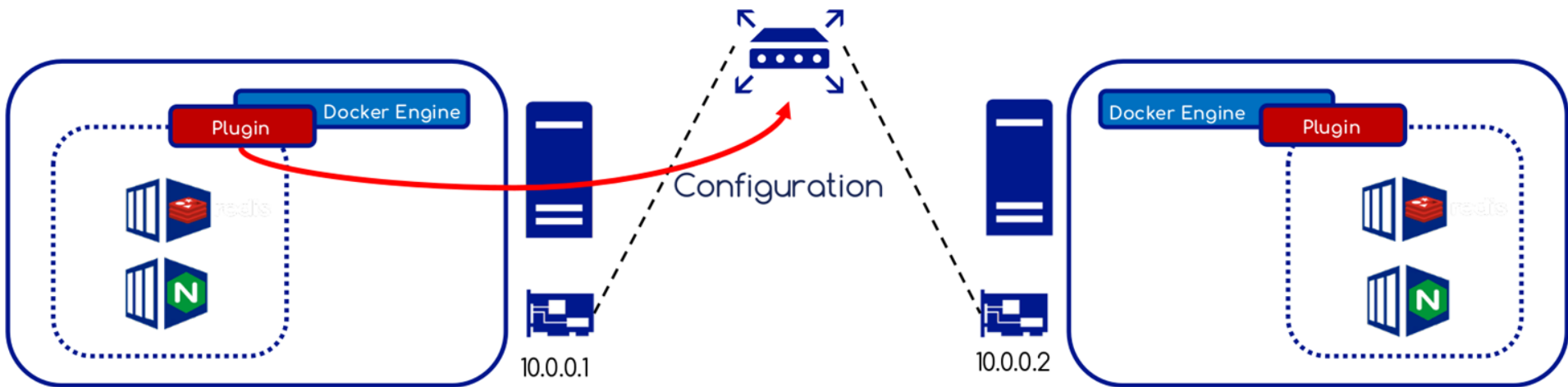


Multi host networking - MACVlan





Multi host networking - Network Plugin





Docker network drivers

| Driver/ Features | Bridge | User defined bridge | Host | Overlay | Macvlan/ipvlan |
|---------------------------------|---|--|--|--|--|
| Connectivity | Same host | Same host | Same host | Multi-host | Multi-host |
| Service Discovery and DNS | Using “links”. DNS using /etc/hosts | Done using DNS server in Docker engine | Done using DNS server in Docker engine | Done using DNS server in Docker engine | Done using DNS server in Docker engine |
| External connectivity | NAT | NAT | Use Host gateway | No external connectivity | Uses underlay gateway |
| Namespace | Separate | Separate | Same as host | Separate | Separate |



Docker Networks: Các câu lệnh quản lý

Show networks

`ls`

`docker network`

Inspect a network

`docker network inspect`

Create a network

`create --driver`

`docker network`

Attach a network to container

`docker network connect`

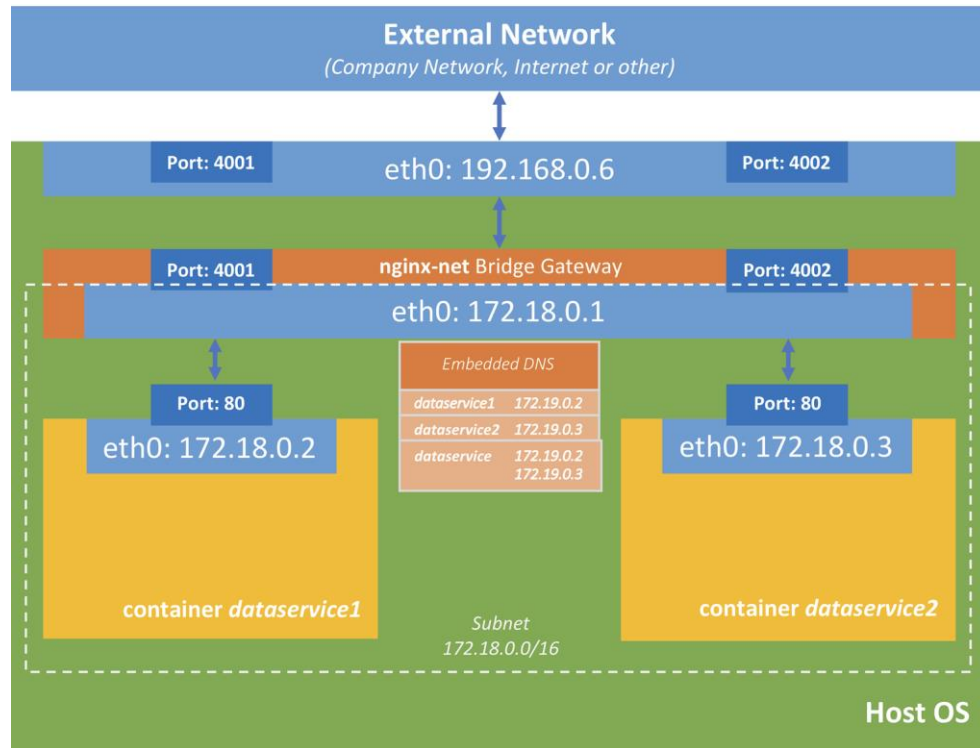
Detach a network from container

`docker network disconnect`



Docker Networks: DNS

- Docker daemon có một built-in DNS server được các container sử dụng
- Mặc định hostname của container chính là container name





Bài tập: DNS Round Robin Test

- Chúng ta có thể tạo nhiều container trên cùng một lớp mạng với trùng địa chỉ DNS
- Tạo một mạng mới với driver bridge
- Tạo 2 container với image `elasticsearch:2`
- Tìm hiểu và sử dụng `--network-alias search` để tạo DNS cho 2 container trên
- Chạy container với image `centos` cùng network và sử dụng `curl -s search:9200` để kiểm tra bằng trường "name"

