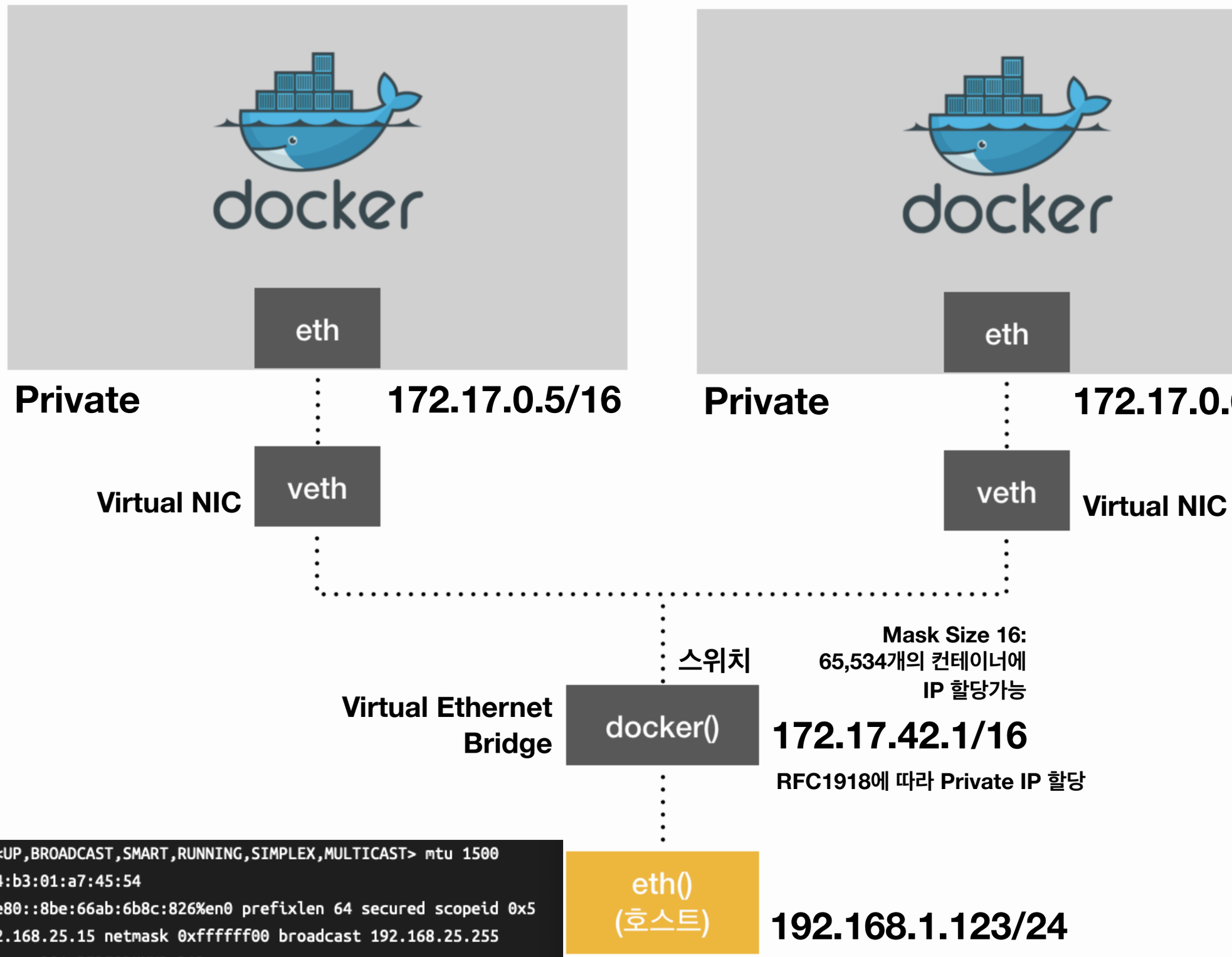
Team bbansrun

Docker 네트워크 구조

NIC

- Network Interface Controller
- 네트워크 신호를 주고 받을 때 사용하는 하드웨어 (LAN Card)
- 리눅스에서는 이 장치를 /dev/eth0 (Basic), /dev/eth1로 인식




```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether c4:b3:01:a7:45:54
        inet6 fe80::8be:66ab:6b8c:826%en0 prefixlen 64 secured scopeid 0x5
        inet 192.168.25.15 netmask 0xffffff00 broadcast 192.168.25.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

Bridge (브릿지)

호스트와 게스트 네트워크를 하나로 연결하여 두 네트워크를 하나의 네트워크처럼 사용하는 방식

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether c4:b3:01:a7:45:54
        inet6 fe80::8be:66ab:6b8c:826%en0 prefixlen 64 secured scopeid 0x5
        inet 192.168.25.15 netmask 0xffffff00 broadcast 192.168.25.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

eth0
(호스트)

192.168.1.123/24

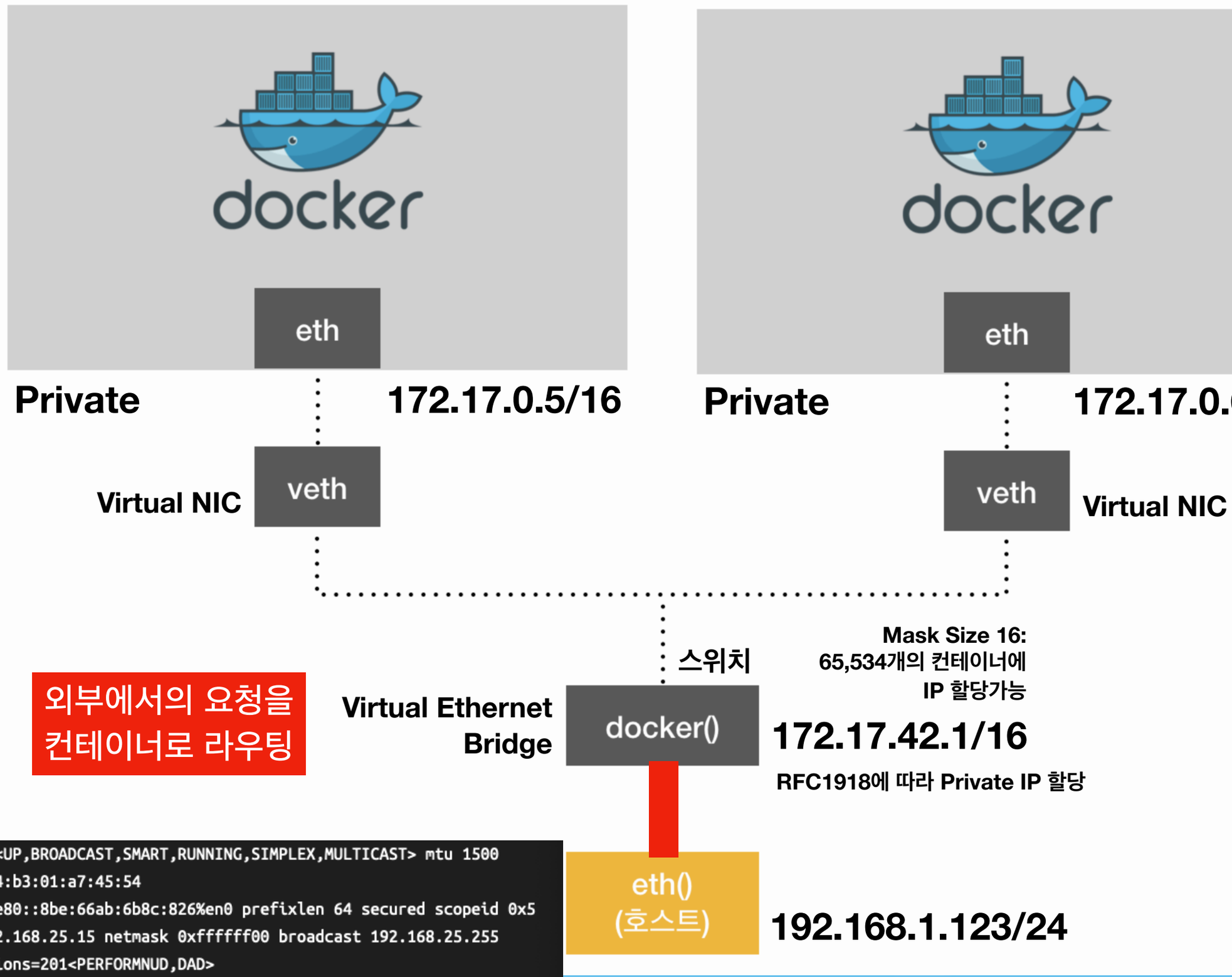
Bridge (브릿지)

호스트와 게스트 네트워크를 하나로 연결하여 두 네트워크를 하나의 네트워크처럼 사용하는 방식

호스트와 게스트가 서로 동등한 자격을 갖추어, **게스트 OS에도 호스트 OS와 같이 Public IP 할당 가능**
(호스트 OS가 사설망에 연결된 경우 호스트와 동일한 IP 대역 할당 가능)

```
inet 192.168.25.15 netmask 0xffffff00 broadcast 192.168.25.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
```

(호스트) **192.168.1.123/24**

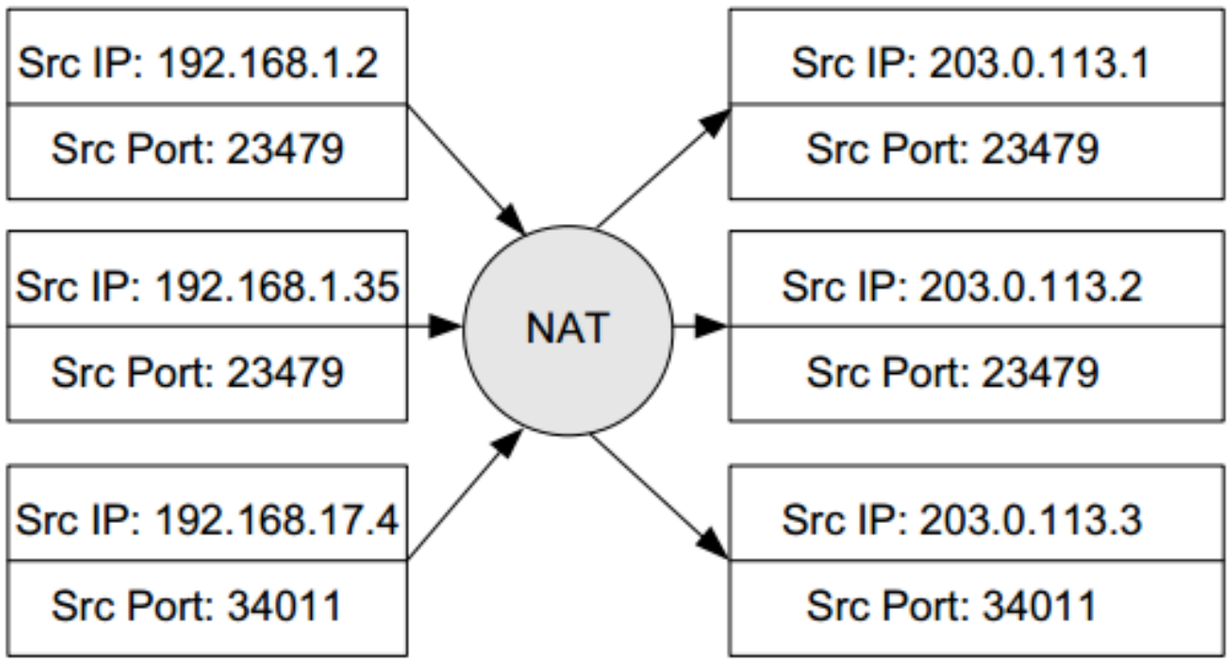


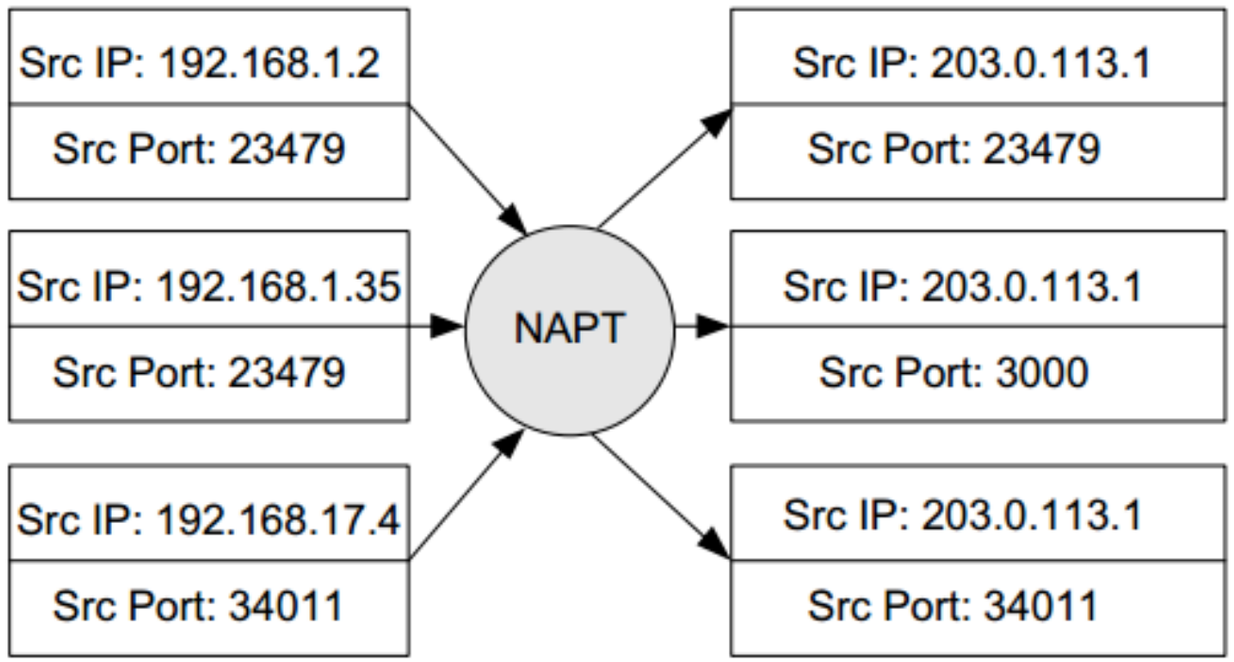

```
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether c4:b3:01:a7:45:54
        inet6 fe80::8be:66ab:6b8c:826%en0 prefixlen 64 secured scopeid 0x5
        inet 192.168.25.15 netmask 0xffffff00 broadcast 192.168.25.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

IP Masquerade

- = NAPT(Network Address Port Translation)
- Private IP와 Public IP를 N:1로 연결
- LAN에서 인터넷으로 접근할 때 Source IP 주소와 Source Port를 같이 변환

IP Masquerade





NAT vs. NAPT

NAT	NAPT
Private IP와 Public IP를 **1:1로 연결**	Private IP와 Public IP를 **N:1로 연결**
LAN에서 인터넷으로 연결시 Source IP를 변환, 인터넷에서 LAN으로 연결시 Dest IP를 변환	LAN에서 인터넷으로 접근시 Source IP, Port를 함께 변환

NAT vs. NAPT

도커는 NAPT 방식 채택

NAT 방식과 다르게 포트 정보까지 활용하기 때문에
하나의 Public IP로 여러 대의 Machine을 동시 연결 가능

Docker Default Network

None

Host

Bridge

Non-Removable

User-defined Network

- Bridge
- Overlay
- Macvlan

User-defined Network

Bridge

IP Subnet과 Gateway를 독립적으로 보유한다는 점에서
Default로 자동 생성되는 Bridge 네트워크와 비슷하나,
User-defined Bridge 네트워크 컨테이너들은 **포트포워딩 없이 서로 통신이 가능**하다.
또한 **DNS를 활용한 자동 서비스 검색 기능이 완벽 호환되기도 함**.
직접 정의한 Bridge 네트워크 내의 Asset과 다른 네트워크에 연결된 기기와 커뮤니케이션하려면
간단히 TCP/UDP 포트를 필요에 따라 열면된다. 도커가 해당 네트워크 주소와 포트를 외부 네트워크에 공개할 것이다.

User-defined Network

Macvlan

Bridge 네트워크와 Overlay 네트워크를 사용할 때 컨테이너와 호스트 사이의 Bridge를 제거함으로써 컨테이너 통신을 간단하고 보다 원활하게 만드는 네트워크.
네트워크 바깥 면을 향해야 하는 컨테이너 자산을 외부 네트워크에 포트포워딩 없이 연결시킬 수 있다.
‘Layer 3 IP 할당’ 대신 ‘Layer 2 MAC 주소’를 사용하기에 가능한 일.

팀 빤스런

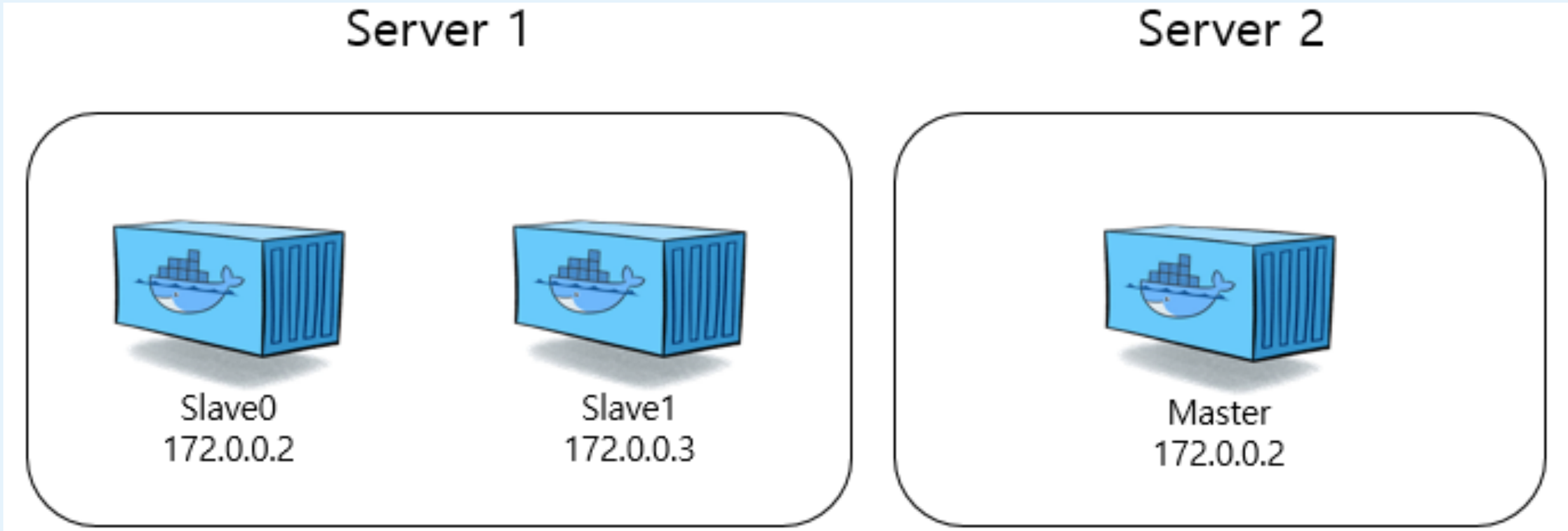
User-defined Network

Overlay

분산된 네트워크를 VXLAN Encapsulation을 사용해 모든 컨테이너가 같은 노드 안에 있는 것처럼 서로간의 Directly한 통신이 가능해짐. (Network Tunneling)

etcd 이용

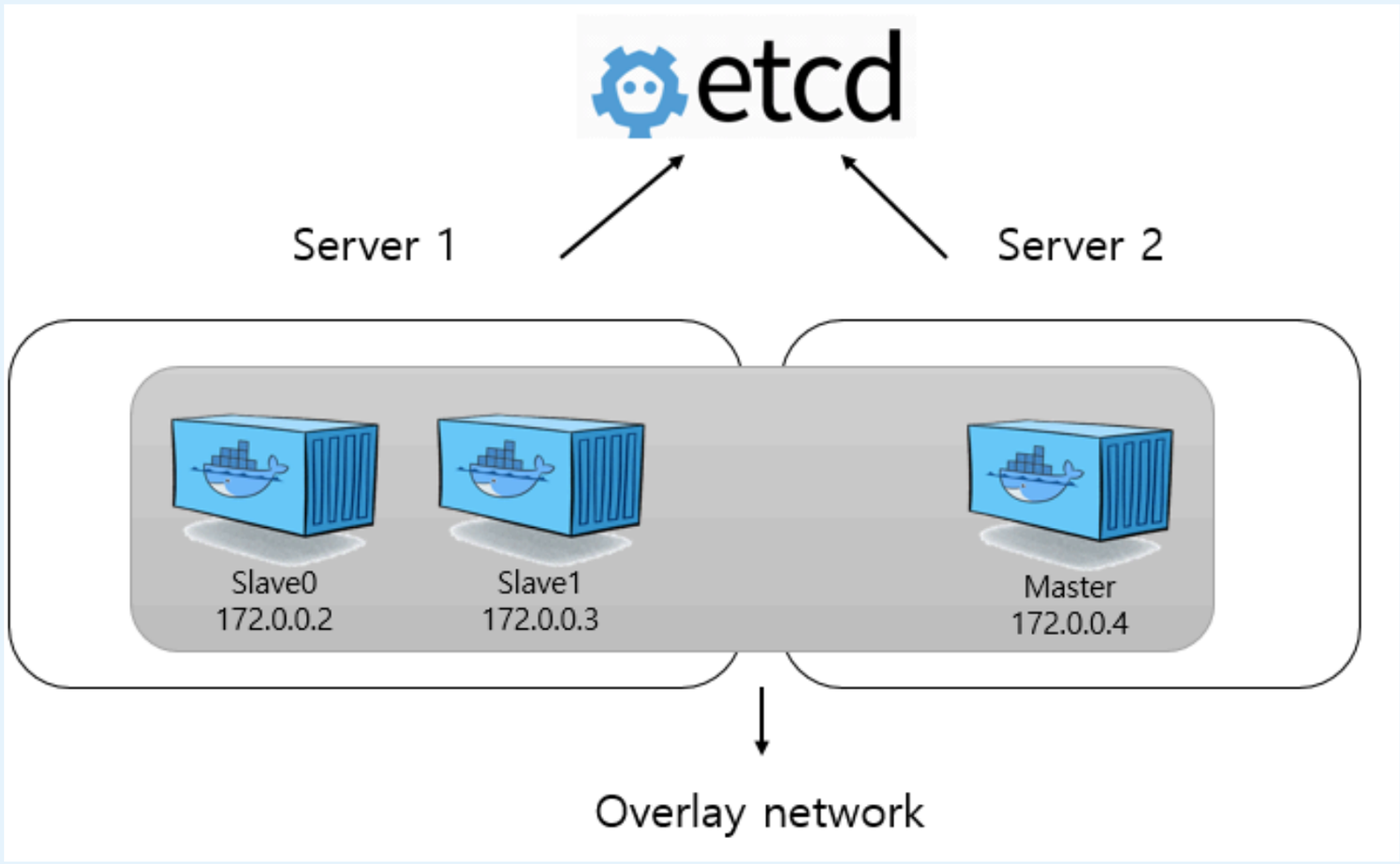
User-defined Network



etcd 이용

서로 다른 NAT로 구성이 된 경우.
호스트와의 포트포워딩으로 해결이 되나, 일부 애플리케이션에서는 런타임으로 Inbound-Outbound 포트가 랜덤하게 지정되는 경우가 있음.
이 경우는 포트포워딩으로도 불가.

User-defined Network



etcd 이용

(Single Network Pool 생성)

References

- https://futurecreator.github.io/2018/11/16/docker-container-basics/
- https://itmore.tistory.com/entry/NAT-%EC%99%80-Bridged-Networking-%EA%B0%9C%EB%85%90-%EC%A0%95%EB%A6%AC
- http://snowdeer.github.io/common-sense/2018/02/02/understanding-about-nat/
- https://www.joinc.co.kr/w/man/12/docker/AdvancedNetwork
- https://www.boannews.com/media/view.asp?idx=55957
- https://doitnow-man.tistory.com/m/183
- https://m.blog.naver.com/PostView.nhn?blogId=alice_k106&logNo=220772125819