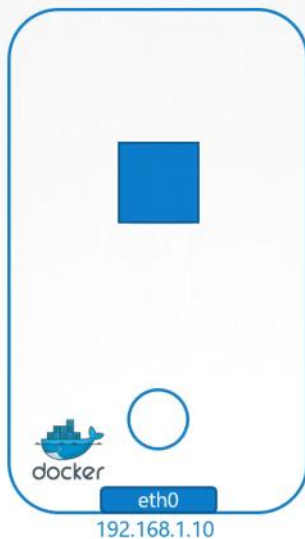
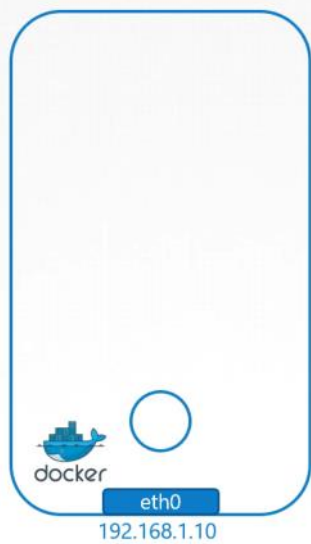
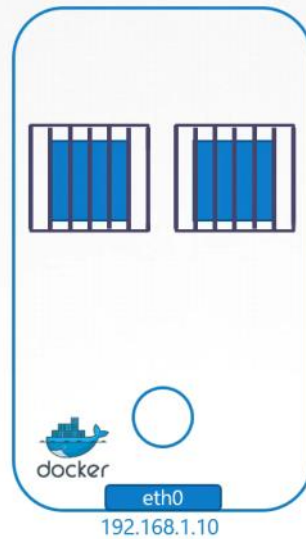


# PRE-REQUISITE DOCKER NETWORKING



```
▶ docker run --network none nginx
```

# NONE

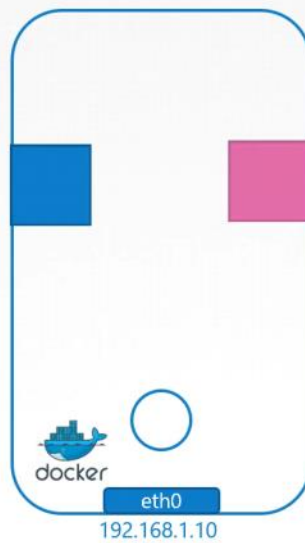


```
▶ docker run --network none nginx
```

```
▶ docker run --network none nginx
```

# HOST

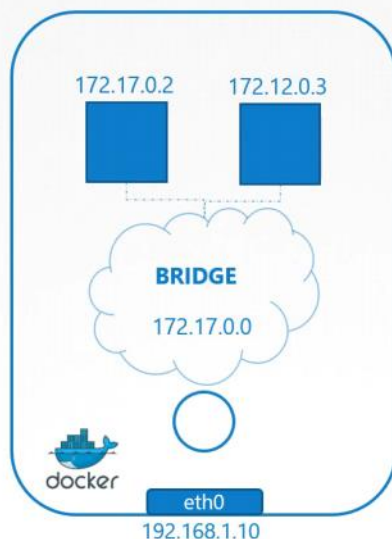
```
http://192.168.1.10:80
```



```
▶ docker run --network host nginx
```

```
▶ docker run --network host nginx
```

# BRIDGE



```
▶ docker run nginx
```

```
▶ docker run nginx
```

# BRIDGE

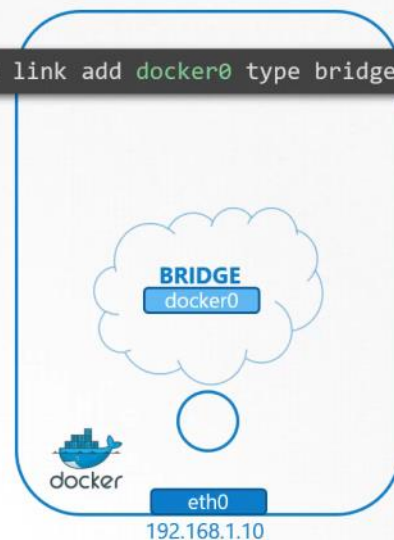
```
▶ docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
2b60087261b2	bridge	bridge	local
0beb4870b093	host	host	local
99035e02694f	none	null	local

```
▶ ip link
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 02:42:ac:11:00:08 brd ff:ff:ff:ff:ff:ff
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:88:56:50:83 brd ff:ff:ff:ff:ff:ff
```

```
▶ ip link add docker0 type bridge
```



```
▶ ip addr
```

```
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc
noqueue state DOWN group default
    link/ether 02:42:88:56:50:83 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/24 brd 172.17.0.255 scope global docker0
        valid_lft forever preferred_lft forever
```

# BRIDGE

```
ip addr
```

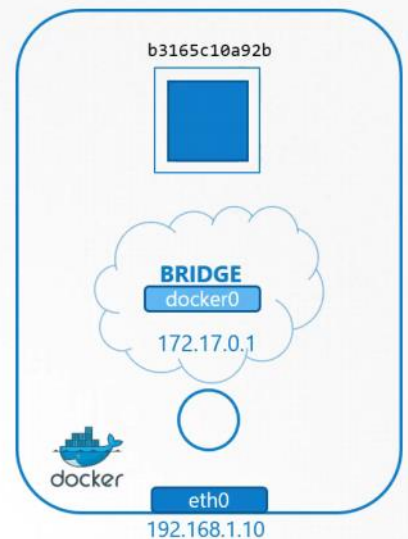
```
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:88:56:50:83 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/24 brd 172.17.0.255 scope global docker0
        valid_lft forever preferred_lft forever
```

```
ip netns
```

```
b3165c10a92b
```

```
docker inspect 942d70e585b2
```

```
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "b3165c10a92b50edce4c8aa5f37273e180907ded31",
  "SandboxKey": "/var/run/docker/netns/b3165c10a92b",
```



```
docker run nginx
```

```
2e41deb9ef1b8b3d141c7bb55d883541b4
```

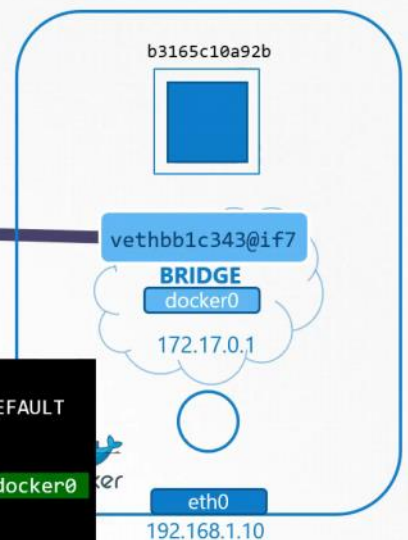
# BRIDGE

```
ip netns
```

```
b3165c10a92b
```

```
ip link
```

```
...
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
   group default
   link/ether 02:42:9b:5f:d6:21 brd ff:ff:ff:ff:ff:ff
8: vethbb1c343@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0
   state UP mode DEFAULT group default
   link/ether 9e:71:37:83:9f:50 brd ff:ff:ff:ff:ff:ff link-netnsid 1
```



```
docker run nginx
```

```
2e41deb9ef1b8b3d141c7bb55d883541b4
```

# BRIDGE

```
ip netns
```

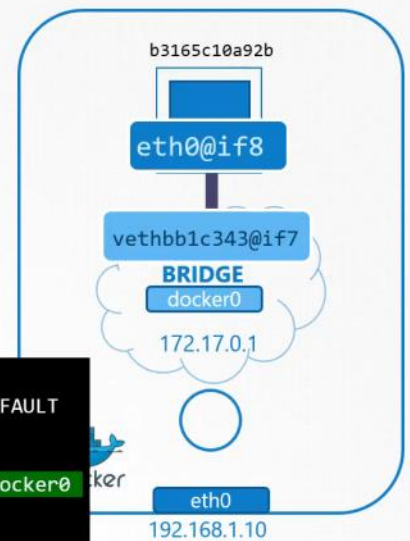
```
b3165c10a92b
```

```
ip link
```

```
...
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
   group default
   link/ether 02:42:9b:5f:d6:21 brd ff:ff:ff:ff:ff:ff
8: vethbb1c343@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0
   state UP mode DEFAULT group default
   link/ether 9e:71:37:83:9f:50 brd ff:ff:ff:ff:ff:ff link-netnsid 1
```

```
ip -n b3165c10a92b link
```

```
...
7: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
   group default
   link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

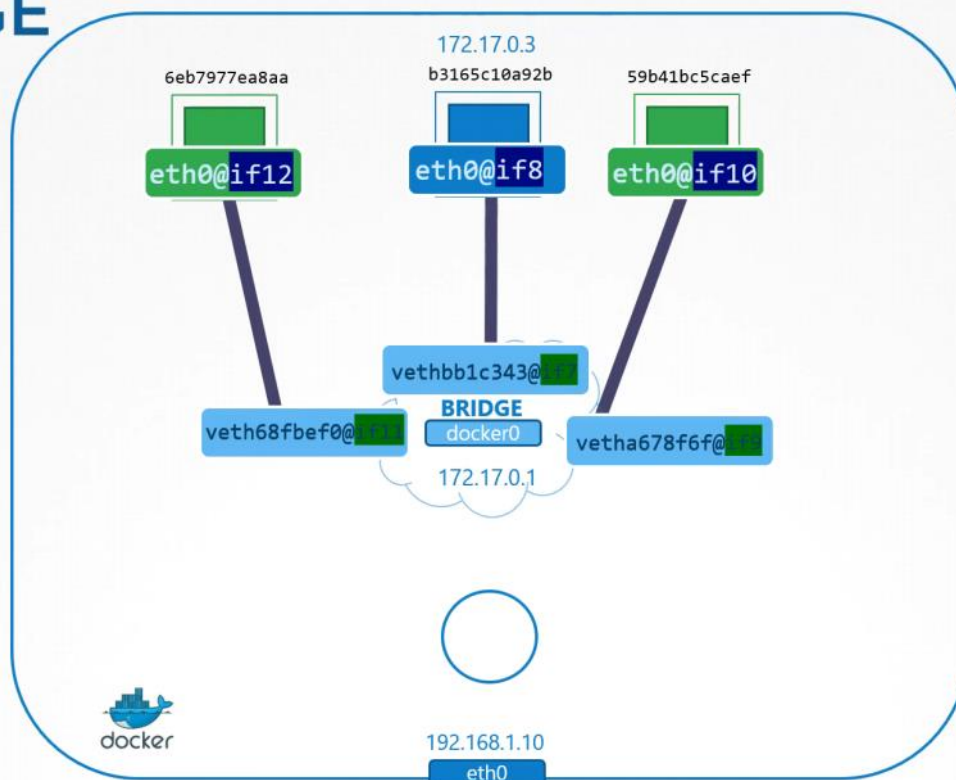


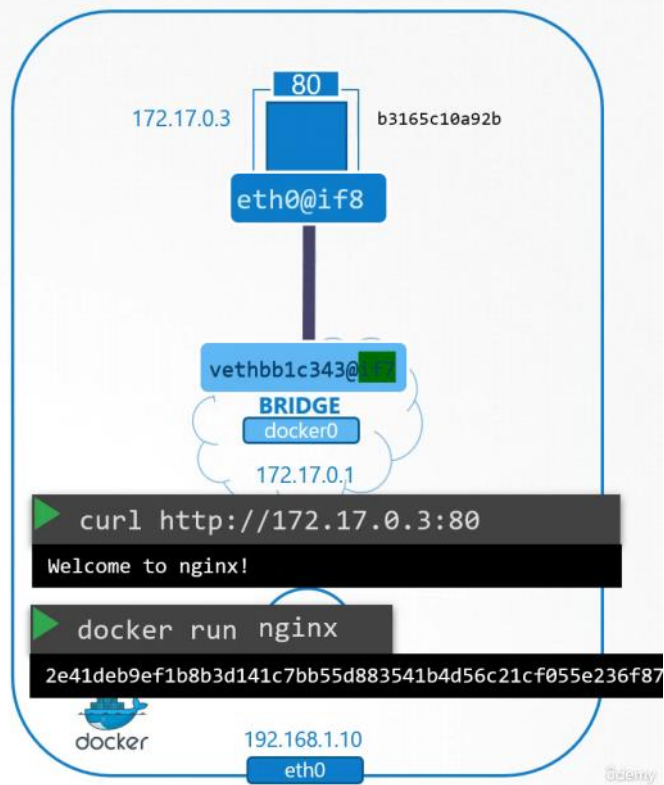
```
docker run nginx
```

```
2e41deb9ef1b8b3d141c7bb59
```

@denny

# BRIDGE





```

curl http://172.17.0.3:80
curl: (7) Failed to connect... No route to host

```

```

docker run -p 8080:80 nginx
2e41deb9ef1b8b3d141c7bb55d883541b4d56c21cf055e236f870bd0f274e52b

```



# BRIDGE

```

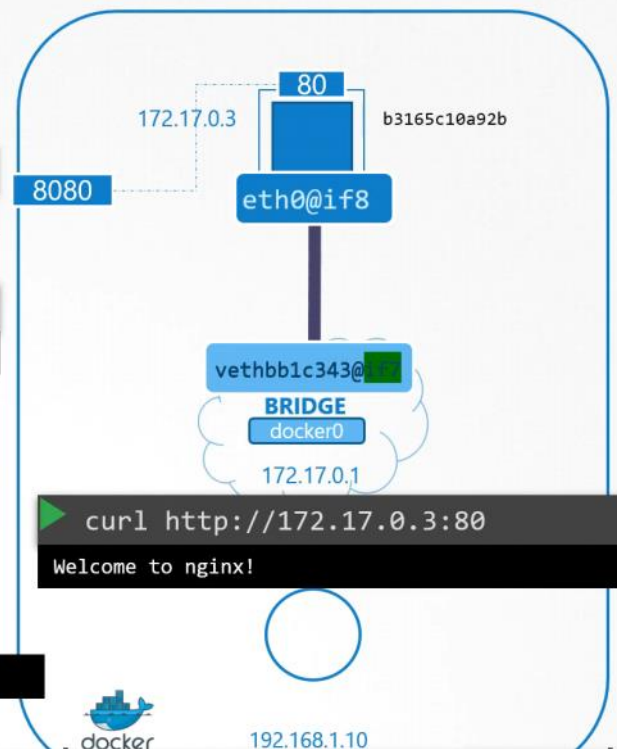
> curl http://172.17.0.3:80
curl: (7) Failed to connect... No route to host
    
```

```

> curl http://192.168.1.10:8080
Welcome to nginx!
    
```

```

> docker run -p 8080:80 nginx
2e41deb9ef1b8b3d141c7bb55d883541b4d56c21cf055e236f870bd0f274e52b
    
```



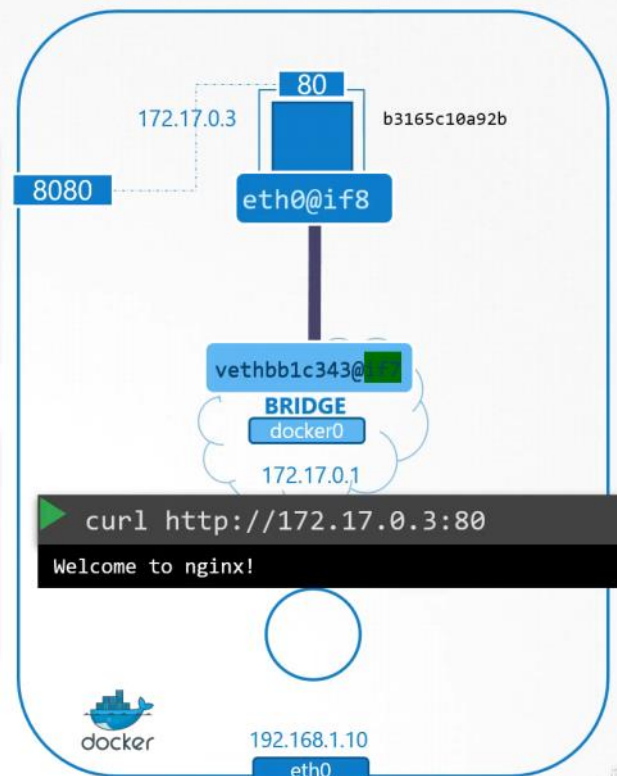
# BRIDGE

```

> iptables \
  -t nat \
  -A PREROUTING \
  -j DNAT \
  --dport 8080 \
  --to-destination 80
    
```

```

> iptables \
  -t nat \
  -A DOCKER \
  -j DNAT \
  --dport 8080 \
  --to-destination 172.17.0.3:80
    
```



```

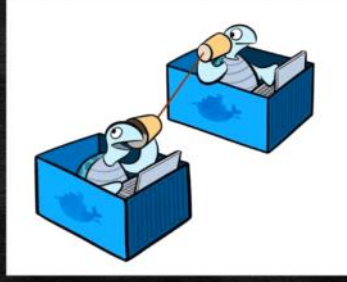
> iptables -nvL -t nat
    
```

```

Chain DOCKER (2 references)
target    prot opt source                destination
RETURN    all  --  anywhere              anywhere
DNAT      tcp  --  anywhere              anywhere    tcp dpt:8080 to:172.17.0.2:80
    
```

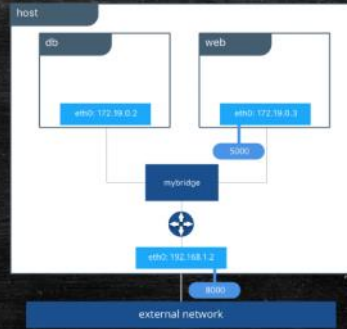
## Docker Network Driver

- Bridge
- Host
- Macvlan
- None
- Overlay



## Bridge

- Varsayılan olan driverdir. Network objesi yaratılırken özellikle başka bir driver belirtilmezse bridge driver ile network yaratılır.
- Her Docker kurulu host üstünde bridge driver ile yaratılmış "Bridge" adında bir network bulunur ve container yaratıldığında aksi belirtmediği sürece varsayılan olarak buna bağlanır.



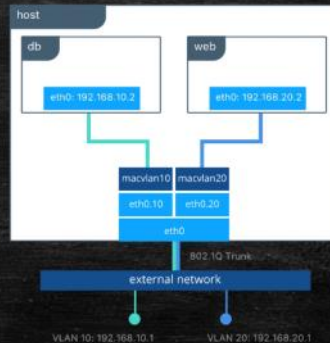
## Host

- Her sistemde host driver ile yaratılmış "Host" adından bir network bulunur.
- Bu network'e bağlı container'da network izolasyonu olmaz. Sanki o host üstünde çalışan bir process gibi host'un ağ kaynaklarını kullanır.



## MacVlan

- Bu driver ile oluşturulan network objeleri sayesinde containerlar fiziksel ağlara kendi mac adreslerine sahip birer fiziksel ağ adaptörüne sahipmişcesine bağlanabilirler.





## None

- Container hiçbir şekilde ağ bağlantısına sahip olmasın istenirse bu driver ile yaratılan network'e bağlanır.



## Overlay

- Ayrı hostlar üstündeki containerların aynı ağda çalışıyormuş gibi çalışması istendiği zaman Overlay networkler devreye girer.

