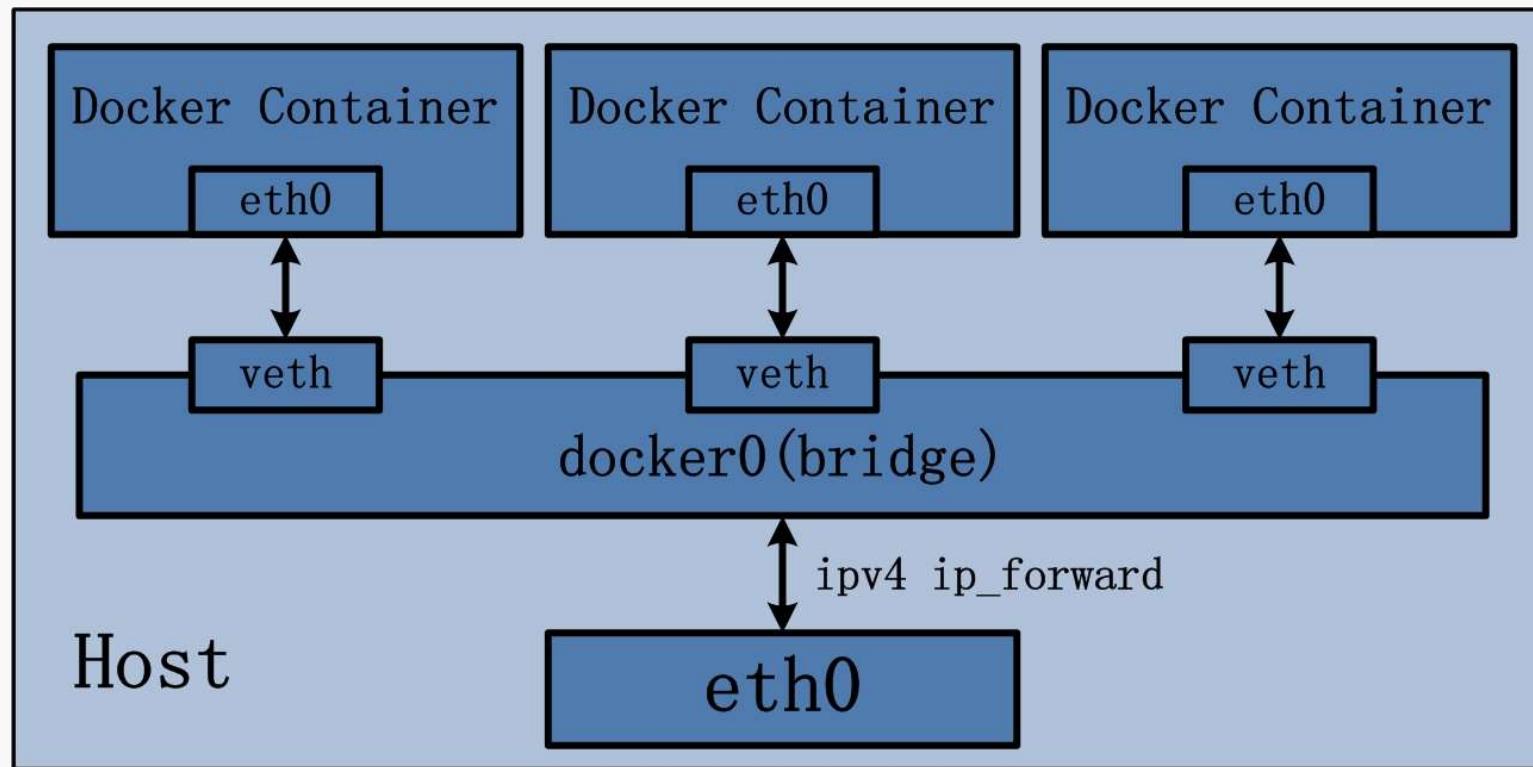


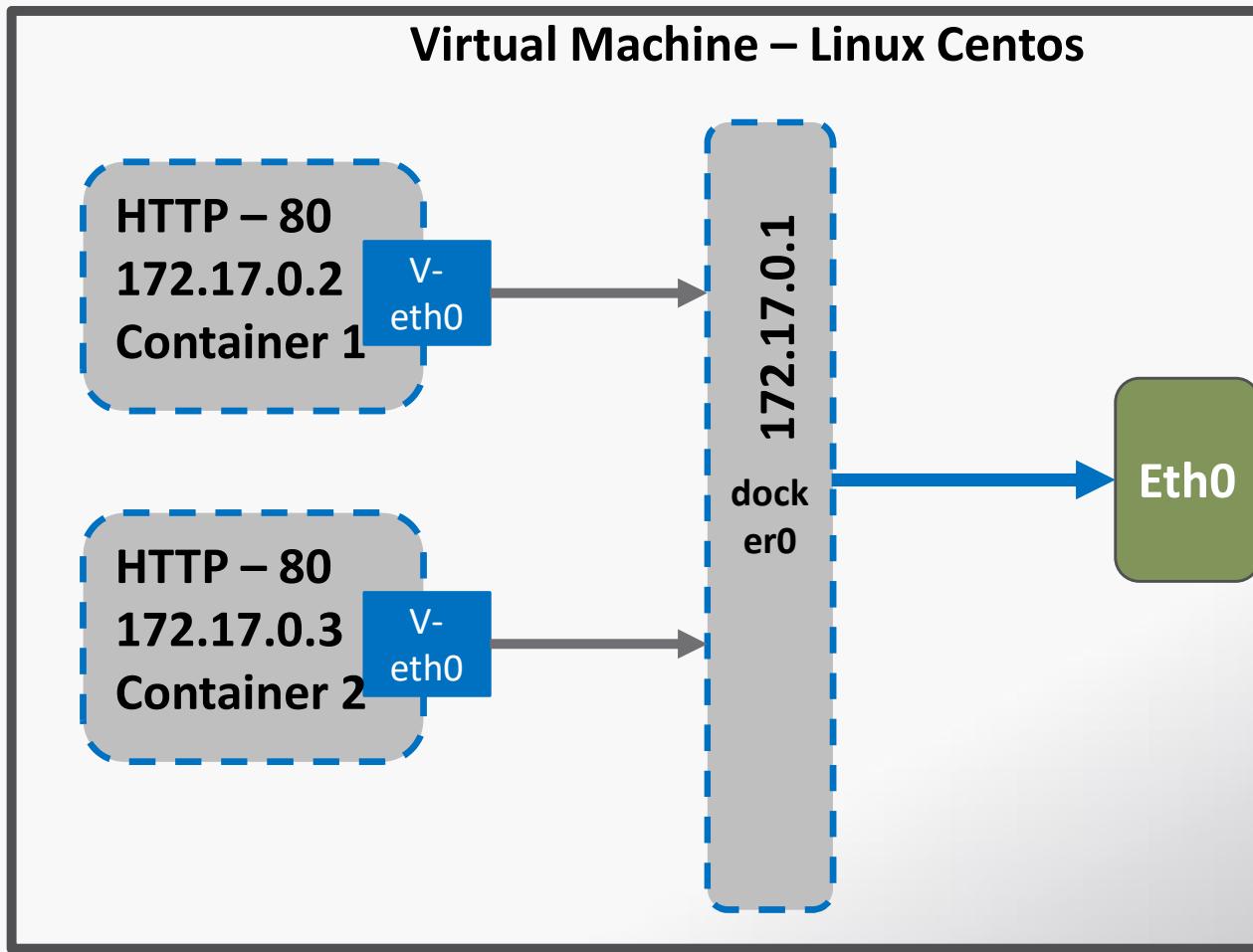
Docker Networking

VISHWANATH M S
VISHWACLOUDLAB.COM



Docker Default Networking





Network Interface on VM
192.168.118.148

**8000 – container1:80
8001 – container2:80**





Basic Commands

- \$ docker network ls

NETWORK ID	NAME	DRIVER
7fca4eb8c647	bridge	bridge
9f904ee27bf5	none	null
cf03ee007fb4	host	host

- The bridge network represents the docker0 network present in all Docker installations.

```
[root@docker-mas01 ~]# ifconfig | more
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 172.17.0.1 netmask 255.255.0.0 broadcast 0.0.0.0
              ether 02:42:c7:98:c2:f4 txqueuelen 0 (Ethernet)
                    RX packets 0 bytes 0 (0.0 B)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 0 bytes 0 (0.0 B)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



\$ docker network inspect bridge

```
[root@docker-mas01 ~]# docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "17eae3020524b5f5c1c9f799a9b6b7617bd60106b6e872ba9713d4f0120dd6c8",
    "Created": "2019-10-01T07:01:20.191932196-04:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Containers": {
      "097fc071af3af87e914628efa4f3a31bd28069cdd1b1e0d44abab8a0108852d": {
        "Name": "eloquent_mirzakhani",
        "EndpointID": "731db860766394957234df6c7ee0ed75c5a73d9ea5c5a1ca71",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": null
      }
    }
  }
]
```



To Disable the default Bridge port Network

- Add the below in the “daemon.json” file to disable the use of Default Bridge port Network on the LINUX HOST.
 - “*bridge*”: “*none*”,
 - “*iptables*”: “*false*”
- ***Note: -- Don't forget to restart the docker service for the changes to be effected.***



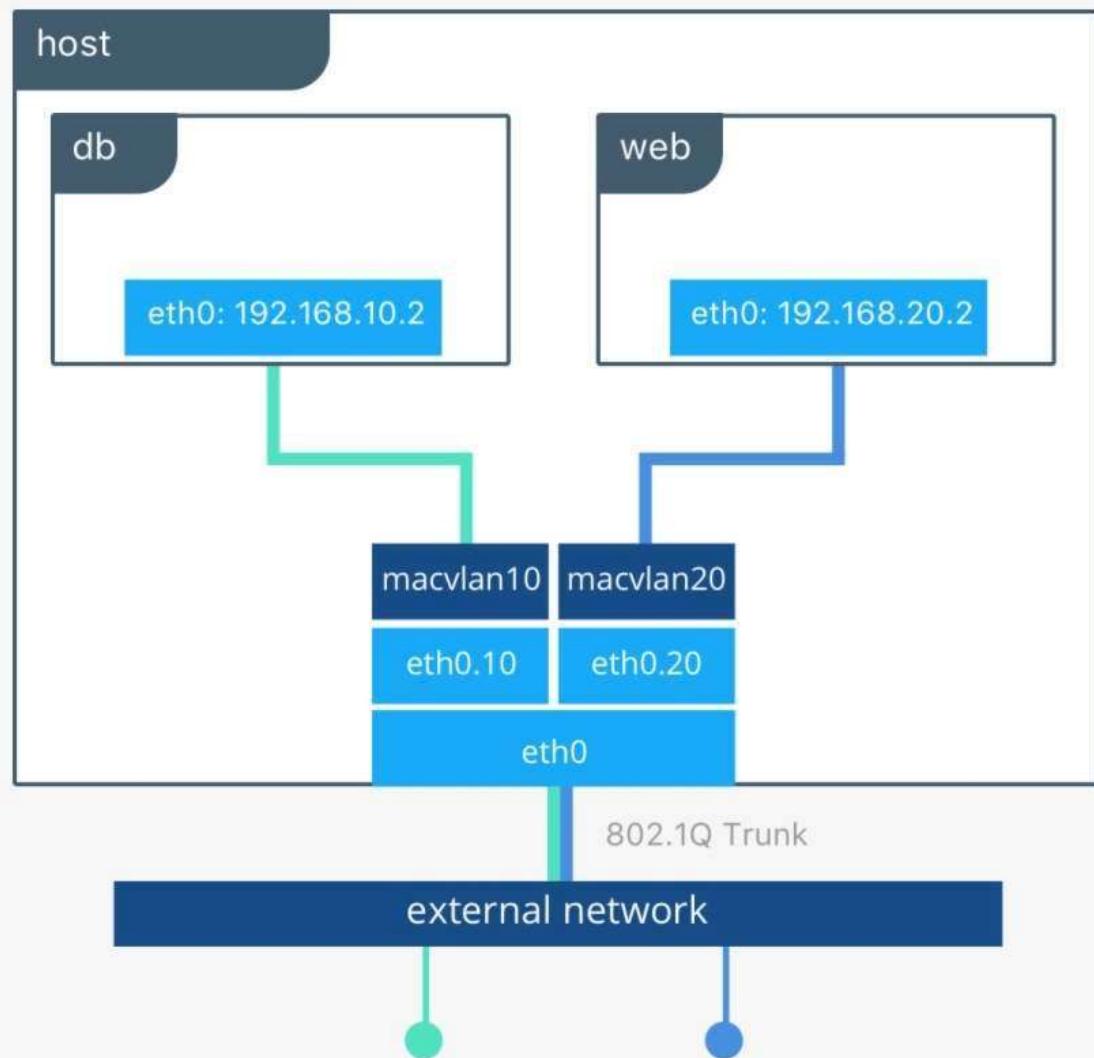
User-defined networks (UDN)

- User Defined networks are best suited in most cases,
- This gives better control over the network for controlling the communication between Containers.
- Docker provides default network drivers for creating the networks.
- There is NO limit to create the number of networks.
- Containers can be connected and disconnected to networks on the fly.
- **Note:** -- Since the Communication of Containers within the UDN is by default, using “linking” is not supported.



Types of User-Defined Network (UDN)

- Bridge Networks
- Overlay network
- MACVLAN Network





Bridge Networks - UDN

To create a new Bridge network

```
$ docker network create --driver bridge <<N/W Name>>
```

To Check the network

```
$ docker network inspect <<N/W Name>>
```

To launch the container with the new bridge network

```
$ docker run --network=<<N/w Name>> -itd --name=cont3 <<imageName>>
```



More with network Commands

- \$ docker network create
- \$ docker network connect
- \$ docker network ls
- \$ docker network rm
- \$ docker network disconnect
- \$ docker network inspect

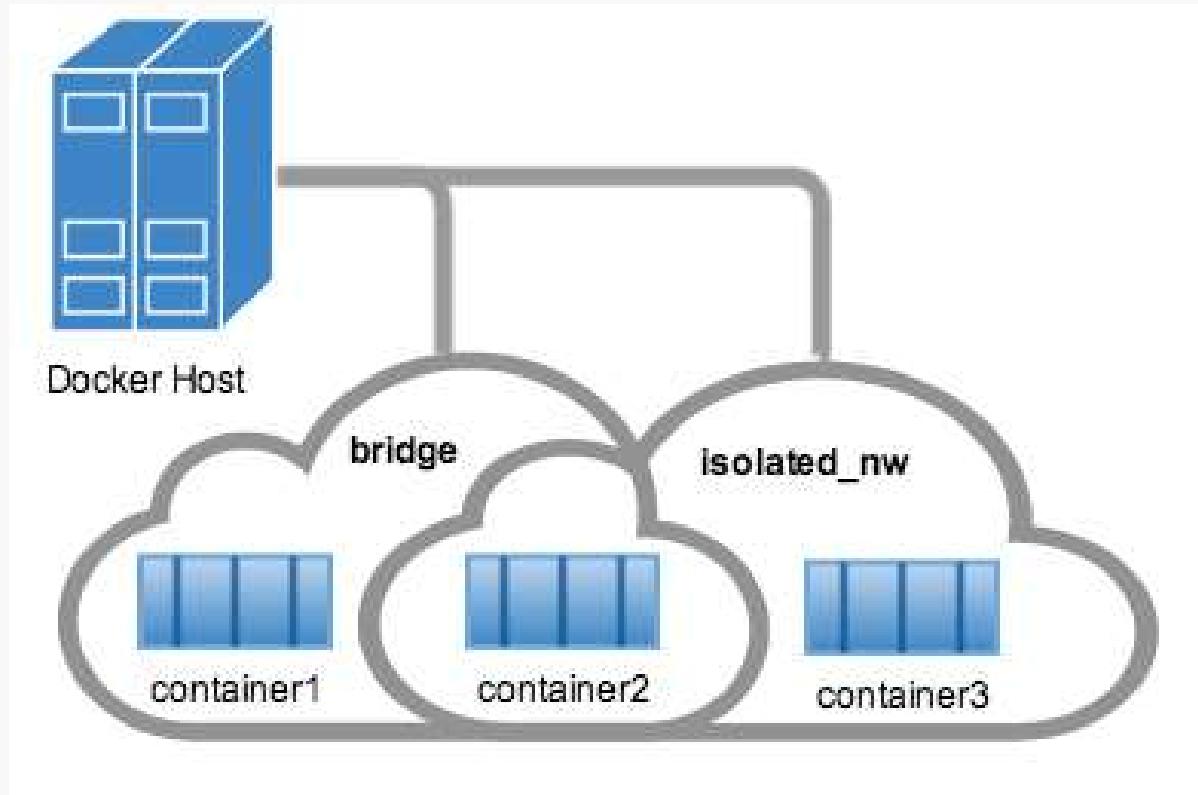


Network connect and disconnect

- \$ docker network connect --ip <host ip to be reserved> <<n/w name>> <<containername>>
- \$ docker network disconnect <<n/w name>> <<containername>>



Some Container Network Diagram...





Must reads links

Networking examples

<https://docs.docker.com/v17.09/engine/userguide/networking/work-with-networks/#basic-container-networking-example>



DEVOPS - DOCKER

VISHWACLOUDLAB.COM