Docker Networking

Docker Host Network -

Diagram

Description automatically generated

Cleanup

docker stop $(docker ps -q)

docker rm $(docker ps -qa)

NOTES:

**APARTMENT EXAMPLE**

Docker (Master- 172.31.38.37/20)

A Block (NetworkA) - 3\*10 = 30(VM-IP) (/16)

1 (Subnet-1) (/24)

A101(IP)… A110

2 (Subnet-2)

A201… A210

3

A310

B Block (NetworkB)

1

2

3

**DOCKER NETWORKING**

Docker (NodeA- 172.31.38.37/20)

Default (docker0) (172.17.0.0/16) – Gateway (172.17.0.1)

container1(172.17.0.2)

container2(172.17.0.3)

container3(172.17.0.4)

UserNet (net01) (172.18.0.0/16) - Gateway (172.18.0.1)

container4(172.18.0.2)

container5(172.18.0.3)

ip a

+> Find the IP address of master node – eth0

docker run -dt --name cont01 donch/net-tools

docker run -dt --name cont02 donch/net-tools

docker inspect cont01 | grep -i IPAddress

docker inspect cont02 | grep -i IPAddress

docker exec -it cont01 /bin/sh

# ping 172.17.0.3

docker network ls

# **User defined Network container**

VM># docker network create net-01

VM># docker network ls

# Check the Network IP Range of each Network - bridge and exnet

VM># docker inspect net-01

VM># docker inspect bridge

VM># docker run -dt --network net-01 --name usr-cont donch/net-tools

VM># docker ps

VM># docker exec -it usr-cont /bin/sh

CONTAINER># ifconfig

# IP will be created in net-01 network

CONTAINER># exit

# **Multi Network Container**

Timeline

Description automatically generated

Diagram

Description automatically generated

# **Lab:**

Diagram, Teams

Description automatically generated

## 2-Creating new Network

VM># docker network create my\_bridge

VM># docker network ls

# Check the Network IP Range of each Network - bridge and exnet

VM># docker inspect my\_bridge

VM># docker inspect bridge

## 3-Create db container in exnet network

VM># docker run -dt --network my\_bridge --name db donch/net-tools

VM># docker ps

VM># docker exec -it db /bin/sh

db># ifconfig

# IP will be created in my\_bridge network, note down the ip-addr

db># exit

## 4-Create web container in default/bridge network

VM># docker run -dt --name web donch/net-tools

VM># docker ps

VM># docker exec -it web /bin/sh

web># ifconfig

# IP will be created in default/bridge network

web># exit

## 5-Attaching my\_bridge network with web container

VM># ***docker network connect*** my\_bridge ***web***

VM># docker exec -it web /bin/sh

web># ifconfig

# 2 different IP will be created in default/bridge & my\_bridge networks

web># ping <ip-addr-of-db-ip>

web># exit

## Detach my\_bridge network from web container

VM># *docker network disconnect* my\_bridge *centos*

VM># docker exec -it web /bin/sh

web># ifconfig

# 1 IP will be shown in default/bridge network

web># exit

# **Defining CIDR and Gateway for a Docker Network**

docker network create --subnet=172.16.0.0/24 --gateway=172.16.0.1 custom-net

docker network ls

docker network inspect custom-net

docker run -dt --network custom-net --name custom-alpine donch/net-tools

docker container inspect custom-alpine

docker exec -it custom-alpine /bin/sh

# **Cleanup**

docker stop $(docker ps -qa)

docker rm $(docker ps -qa)

docker network rm net-01