version: "3"

services:

  proxy:

    image: vfarcic/docker-flow-proxy

    ports:

      - 80:80

      - 443:443

    networks:

      - proxy

    environment:

      - LISTENER\_ADDRESS=swarm-listener

      - MODE=swarm

    deploy:

      replicas: 2

  swarm-listener:

    image: vfarcic/docker-flow-swarm-listener

    networks:

      - proxy

    volumes:

      - /var/run/docker.sock:/var/run/docker.sock

    environment:

      - DF\_NOTIFY\_CREATE\_SERVICE\_URL=<http://proxy:8080/v1/docker-flow-proxy/reconfigure>

      - DF\_NOTIFY\_REMOVE\_SERVICE\_URL=<http://proxy:8080/v1/docker-flow-proxy/remove>

    deploy:

      placement:

        constraints: [node.role == manager]

networks:

  proxy:

    external: true

|  |
| --- |
| root@dockermaster:~# history |
|  | 1 apt-get update |
|  | 2 apt-get install apt-transport-https ca-certificates curl gnupg lsb-release |
|  | 3 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg |
|  | 4 echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \ |
|  | $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null |
|  | 5 apt-get update |
|  | 6 apt-get install docker-ce docker-ce-cli containerd.io |
|  | 7 docker ps |
|  | 8 history |
|  | 9 ifconfig |
|  | 10 docker swarm init --advertise-addr=10.128.0.2 |
|  | 11 docker node ls |
|  | 12 docker service create --replicas 4 -p 80:80 --name web nginx |
|  | 13 docker service ls |
|  | 14 docker service ps web |
|  | 15 docker service scale web=8 |
|  | 16 docker service ps web |
|  | 17 docker service scale web=4 |
|  | 18 docker service ps web |
|  | 19 docker service scale web=8 |
|  | 20 docker service ps web |
|  | 21 docker ps |
|  | 22 docker service ls |
|  | 23 docker service ps web |
|  | 24 docker node ls |
|  | 25 docker service ps web |
|  | 26 docker service ls |
|  | 27 docker service update --image nginx:perl web |
|  | 28 docker service ps web |
|  | 29 docker service update --image nginx:latest web |
|  | 30 docker service ps web |
|  | 31 docker network ls |
|  | 32 docker service ps web |
|  | 33 docker service create --help |
|  | 34 history |

Docker orchesteration:

With the help of services we can deploy containers in docker swerm\

Docker networks:

Bridge==default network

root@mani-VirtualBox:~# docker network ls

NETWORK ID NAME DRIVER SCOPE

dc853409e146 bridge bridge local

64474e8196e9 host host local

e064b5bc753a none null local

17920a7e9dfd root\_default bridge local

root@mani-VirtualBox:~#

root@mani-VirtualBox:~# docker network inspect dc853409e146

[

{

"Name": "bridge",

"Id": "dc853409e146689bf16650545e9cc6389c3da48fd01bfe018ed91520636e8da3",

"Created": "2022-07-24T20:31:43.420017321+05:30",

"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,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

root@mani-VirtualBox:~# docker network inspect bridge

[

{

"Name": "bridge",

"Id": "dc853409e146689bf16650545e9cc6389c3da48fd01bfe018ed91520636e8da3",

"Created": "2022-07-24T20:31:43.420017321+05:30",

"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,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

root@mani-VirtualBox:~# docker network create shiva

b187c2bd6db1f57b9d6ba6a6e60ed64dedffbf3ea20beef4b2669c83b22b4105

root@mani-VirtualBox:~# docker network ls

NETWORK ID NAME DRIVER SCOPE

dc853409e146 bridge bridge local

64474e8196e9 host host local

e064b5bc753a none null local

17920a7e9dfd root\_default bridge local

b187c2bd6db1 shiva bridge local

docker network types:

bridge

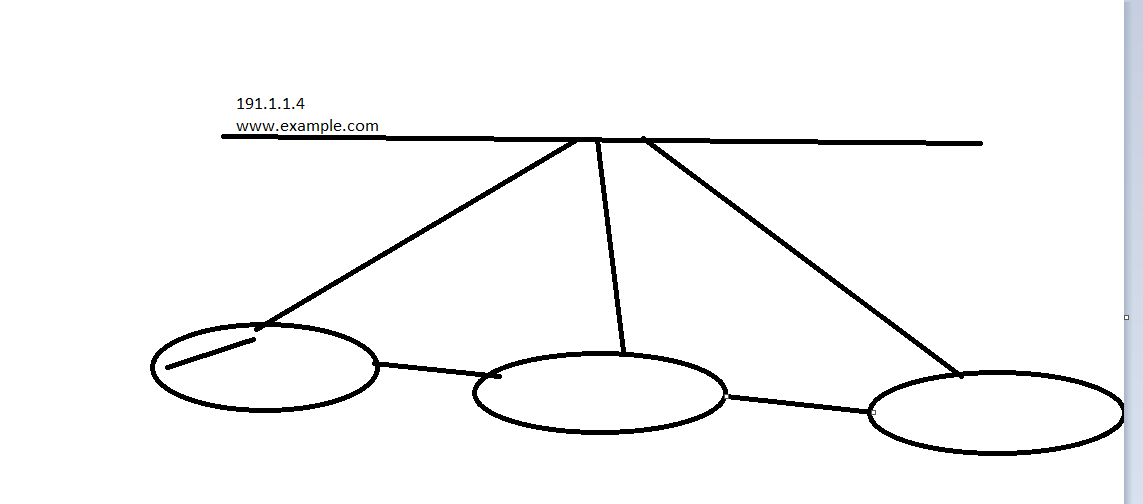
host

overlay

macvlan

none

overlay: it will create by default when docker swerm is initating its cluster because it is always used to communicate data between the worker nodes(machines).which are in different regions.



root@mani-VirtualBox:~# docker run -it --network=shiva ubuntu /bin/bash

root@mani-VirtualBox:~# docker run -it --network=shiva ubuntu /bin/bash

root@3a0a15598dab:/# cat /etc/hosts

127.0.0.1 localhost

::1 localhost ip6-localhost ip6-loopback

fe00::0 ip6-localnet

ff00::0 ip6-mcastprefix

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

172.21.0.2 3a0a15598dab

root@3a0a15598dab:/# exit

exit

root@mani-VirtualBox:~# docker network inspect shiva

[

{

"Name": "shiva",

"Id": "b187c2bd6db1f57b9d6ba6a6e60ed64dedffbf3ea20beef4b2669c83b22b4105",

"Created": "2022-07-24T23:34:46.441007309+05:30",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "172.21.0.0/16",

"Gateway": "172.21.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {},

"Options": {},

"Labels": {}

}

]

root@mani-VirtualBox:~# docker network --help

Usage: docker network COMMAND

Manage networks

Commands:

connect Connect a container to a network

create Create a network

disconnect Disconnect a container from a network

inspect Display detailed information on one or more networks

ls List networks

prune Remove all unused networks

rm Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.

root@mani-VirtualBox:~# docker network COMMAND --help

Usage: docker network COMMAND

Manage networks

Commands:

connect Connect a container to a network

create Create a network

disconnect Disconnect a container from a network

inspect Display detailed information on one or more networks

ls List networks

prune Remove all unused networks

rm Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.

|  |
| --- |
|  |
|  | apt-get update  apt-get install apt-transport-https ca-certificates curl gnupg lsb-release |
|  | curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg |
|  | echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \ |
|  | $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null |
|  | apt-get update |
|  | apt-get install docker-ce docker-ce-cli containerd.io |

Ifconfig

docker swarm init --advertise-addr=10.128.0.2

root@mani-VirtualBox:~# docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION

dnhnoj0u877v1q8t7h7qt0exk \* mani-VirtualBox Ready Active Leader 20.10.17

root@mani-VirtualBox:~#

root@mani-VirtualBox:~# docker swarm init --advertise-addr=192.168.1.108

Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm and join another one.

root@mani-VirtualBox:~# docker swarm leave

Error response from daemon: You are attempting to leave the swarm on a node that is participating as a manager. Removing the last manager erases all current state of the swarm. Use `--force` to ignore this message.

root@mani-VirtualBox:~# docker swarm leave --force

Node left the swarm.

root@mani-VirtualBox:~# docker swarm init --advertise-addr=192.168.1.108

Swarm initialized: current node (0q293l2tvx7fpm02x8ktb4r7b) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-2kfjlr82u10ct7jzi5hjnn8kqlmbrpytqh5lptksjpv6qlcjj3-dycygvp4kqkbi0rdw29ra1da5 192.168.1.108:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

root@mani-VirtualBox:~#

slave:

root@mani-VirtualBox:~# docker swarm join --token SWMTKN-1-2kfjlr82u10ct7jzi5hjnn8kqlmbrpytqh5lptksjpv6qlcjj3-dycygvp4kqkbi0rdw29ra1da5 192.168.1.108:2377

This node joined a swarm as a worker.

root@mani-VirtualBox:~#

root@mani-VirtualBox:~# docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION

6y19d15wnyr04t2eh708jo6wy docker-slave2 Ready Active 20.10.17

0q293l2tvx7fpm02x8ktb4r7b \* mani-VirtualBox Ready Active Leader 20.10.17

kg8hn5aepoqqanc652r1v961g mani-VirtualBox Ready Active 20.10.17

root@mani-VirtualBox:~# docker service create --replicas 4 -p 80:80 --name web nginx

gu5lww4tl93i9jfgmq8qqkdn5

overall progress: 0 out of 4 tasks

1/4: preparing [=================================> ]

2/4: preparing [=================================> ]

3/4: preparing [=================================> ]

4/4: preparing [=================================> ]

root@mani-VirtualBox:~# docker service ls

ID NAME MODE REPLICAS IMAGE PORTS

gu5lww4tl93i web replicated 4/4 nginx:latest \*:80->80/tcp

root@mani-VirtualBox:~# docker service ps web

ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS

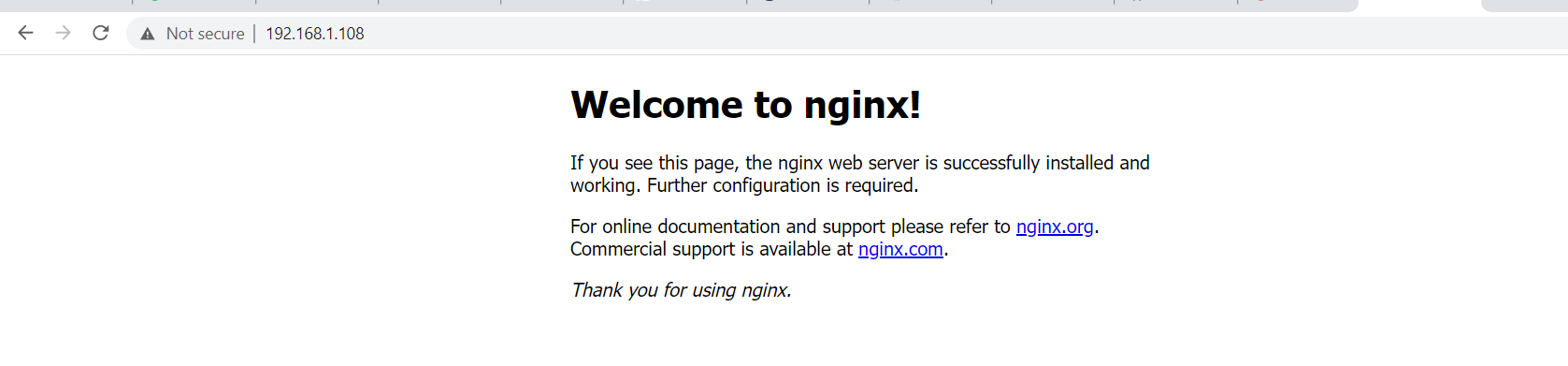
op53y088od3y web.1 nginx:latest mani-VirtualBox Running Running 41 seconds ago

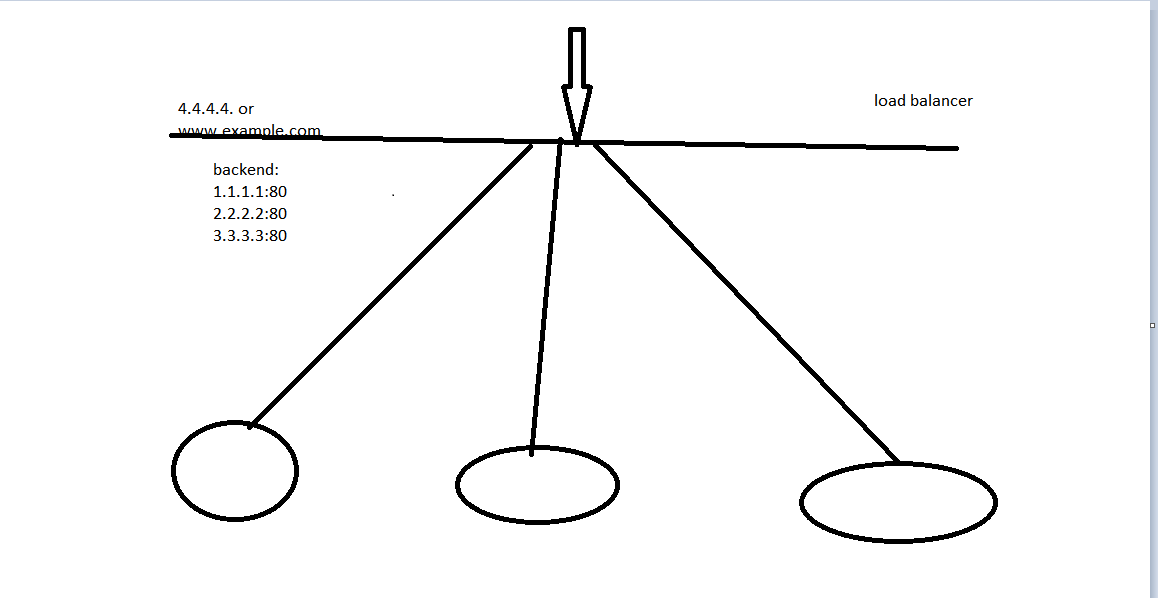
sxs78p6t7vn2 web.2 nginx:latest docker-slave2 Running Running 43 seconds ago

yfxi8s7bnfq9 web.3 nginx:latest mani-VirtualBox Running Running 44 seconds ago

y8sv09dy7i74 web.4 nginx:latest mani-VirtualBox Running Running 41 seconds ago

This distribution of the container will be taken care by master node.





root@mani-VirtualBox:~# docker service scale web=8

web scaled to 8

overall progress: 8 out of 8 tasks

1/8: running [==================================================>]

2/8: running [==================================================>]

3/8: running [==================================================>]

4/8: running [==================================================>]

5/8: running [==================================================>]

6/8: running [==================================================>]

7/8: running [==================================================>]

8/8: running [==================================================>]

Verify: Service converged

root@mani-VirtualBox:~# docker service ps web

ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS

op53y088od3y web.1 nginx:latest mani-VirtualBox Running Running 24 minutes ago

sxs78p6t7vn2 web.2 nginx:latest docker-slave2 Running Running 25 minutes ago

yfxi8s7bnfq9 web.3 nginx:latest mani-VirtualBox Running Running 25 minutes ago

y8sv09dy7i74 web.4 nginx:latest mani-VirtualBox Running Running 24 minutes ago

jmbgcj5s4uju web.5 nginx:latest mani-VirtualBox Running Running about a minute ago

rcw26enjm5i3 web.6 nginx:latest docker-slave2 Running Running about a minute ago

9kseali5df9q web.7 nginx:latest docker-slave2 Running Running about a minute ago

ydkmcd65ih4m web.8 nginx:latest mani-VirtualBox Running Running about a minute ago

root@mani-VirtualBox:~#

root@mani-VirtualBox:~# docker service scale web=4

web scaled to 4

overall progress: 4 out of 4 tasks

1/4: running [==================================================>]

2/4: running [==================================================>]

3/4: running [==================================================>]

4/4: running [==================================================>]

verify: Service converged

root@mani-VirtualBox:~# docker service ps web

ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS

op53y088od3y web.1 nginx:latest mani-VirtualBox Running Running 25 minutes ago

sxs78p6t7vn2 web.2 nginx:latest docker-slave2 Running Running 25 minutes ago

yfxi8s7bnfq9 web.3 nginx:latest mani-VirtualBox Running Running 25 minutes ago

y8sv09dy7i74 web.4 nginx:latest mani-VirtualBox Running Running 25 minutes ago

it is using overly network.

root@mani-VirtualBox:~# docker ps

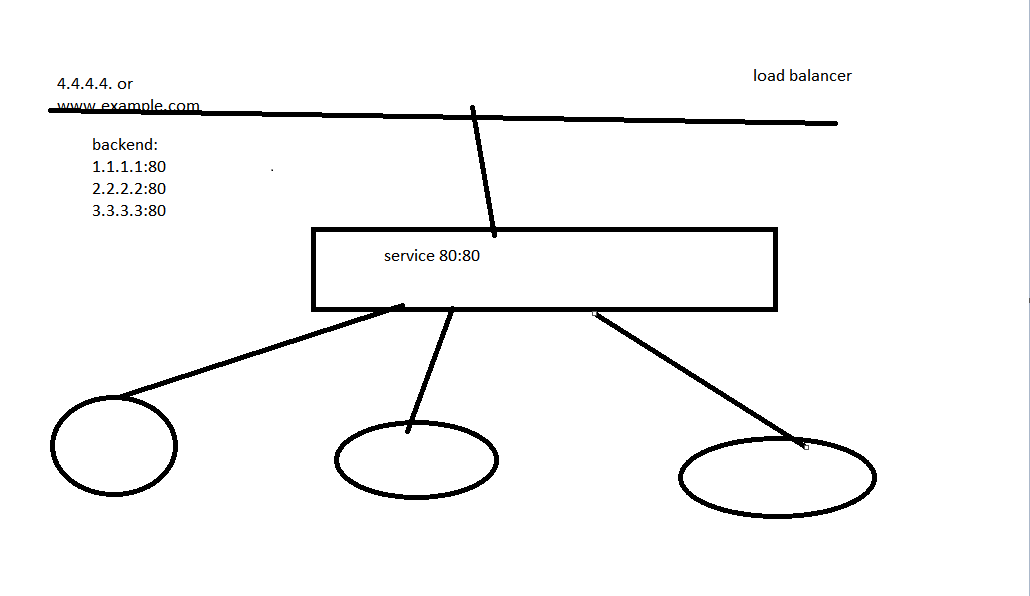
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

b5cf1a0e5153 nginx:latest "/docker-entrypoint.…" 28 minutes ago Up 28 minutes 80/tcp web.3.yfxi8s7bnfq9pgxw7um527scl

root@mani-VirtualBox:~# docker service ls

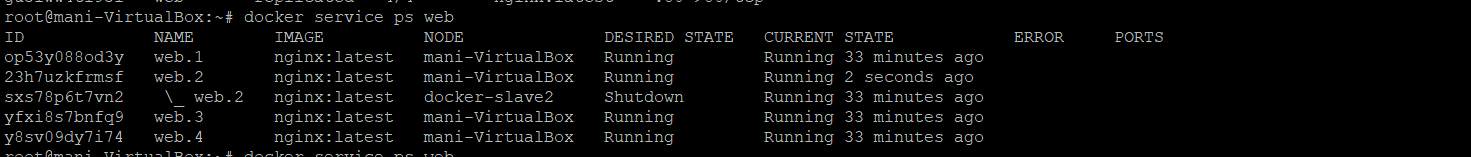
ID NAME MODE REPLICAS IMAGE PORTS

gu5lww4tl93i web replicated 4/4 nginx:latest \*:80->80/tcp



Service will take care that how to reach the respective containers. this services will run in a round robin fashion only

Service by default use round robin fashion



We have stopped one of the containers so master is distributing that load to other nodes.

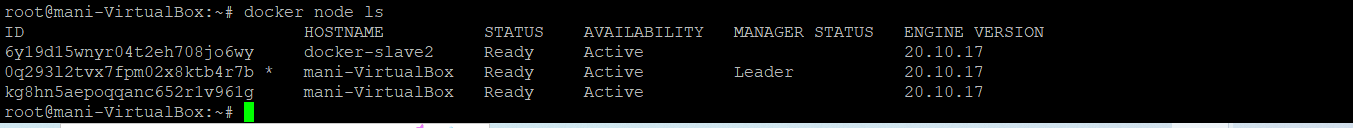
root@mani-VirtualBox:~# docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS ENGINE VERSION

6y19d15wnyr04t2eh708jo6wy docker-slave2 Down Active 20.10.17

0q293l2tvx7fpm02x8ktb4r7b \* mani-VirtualBox Ready Active Leader 20.10.17

kg8hn5aepoqqanc652r1v961g mani-VirtualBox Ready Active 20.10.17



We have started the third node so this node is up. And automatically it was joined to cluster master will always be checking the status of their nodes and once they are up they will join back to cluster.

When a new application got released for nginx it will update the things in a rolling update fashon.with out the down time.

root@mani-VirtualBox:~# docker service update --image nginx:perl web

web

overall progress: 0 out of 4 tasks

overall progress: 4 out of 4 tasks

1/4: running [==================================================>]

2/4: running [==================================================>]

3/4: running [==================================================>]

4/4: running [==================================================>]

verify: Service converged

root@mani-VirtualBox:~#

root@mani-VirtualBox:~# docker service ps web

ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS

ldmqsjyzh8wk web.1 nginx:perl mani-VirtualBox Running Running 56 seconds ago

op53y088od3y \\_ web.1 nginx:latest mani-VirtualBox Shutdown Shutdown about a minute ago

2e5w492bznio web.2 nginx:perl docker-slave2 Running Running about a minute ago

23h7uzkfrmsf \\_ web.2 nginx:latest mani-VirtualBox Shutdown Shutdown about a minute ago

sxs78p6t7vn2 \\_ web.2 nginx:latest docker-slave2 Shutdown Failed 27 minutes ago "task: non-zero exit (255)"

13qhcgqjo6la web.3 nginx:perl mani-VirtualBox Running Running 38 seconds ago

yfxi8s7bnfq9 \\_ web.3 nginx:latest mani-VirtualBox Shutdown Shutdown 38 seconds ago

h6ohy8781uu9 web.4 nginx:perl mani-VirtualBox Running Running 42 seconds ago

y8sv09dy7i74 \\_ web.4 nginx:latest mani-VirtualBox Shutdown Shutdown 53 seconds ago

root@mani-VirtualBox:~# docker service update --image nginx:latest web

web

overall progress: 4 out of 4 tasks

1/4: running [==================================================>]

2/4: running [==================================================>]

3/4: running [==================================================>]

4/4: running [==================================================>]

verify: Service converged

root@mani-VirtualBox:~# docker service ps web

ID NAME IMAGE NODE DESIRED STATE CURRENT STATE ERROR PORTS

7pt4e5gtmiob web.1 nginx:latest mani-VirtualBox Running Running 9 seconds ago

ldmqsjyzh8wk \\_ web.1 nginx:perl mani-VirtualBox Shutdown Shutdown 10 seconds ago

op53y088od3y \\_ web.1 nginx:latest mani-VirtualBox Shutdown Shutdown 2 minutes ago

bbqbvfm8seee web.2 nginx:latest docker-slave2 Running Running 18 seconds ago

2e5w492bznio \\_ web.2 nginx:perl docker-slave2 Shutdown Shutdown 18 seconds ago

23h7uzkfrmsf \\_ web.2 nginx:latest mani-VirtualBox Shutdown Shutdown 3 minutes ago

sxs78p6t7vn2 \\_ web.2 nginx:latest docker-slave2 Shutdown Failed 29 minutes ago "task: non-zero exit (255)"

gtf1bigbemcl web.3 nginx:latest mani-VirtualBox Running Running 22 seconds ago

13qhcgqjo6la \\_ web.3 nginx:perl mani-VirtualBox Shutdown Shutdown 23 seconds ago

yfxi8s7bnfq9 \\_ web.3 nginx:latest mani-VirtualBox Shutdown Shutdown 2 minutes ago

j6apl2bxoqbf web.4 nginx:latest mani-VirtualBox Running Running 14 seconds ago

h6ohy8781uu9 \\_ web.4 nginx:perl mani-VirtualBox Shutdown Shutdown 14 seconds ago

y8sv09dy7i74 \\_ web.4 nginx:latest mani-VirtualBox Shutdown Shutdown 2 minutes ago

root@mani-VirtualBox:~# docker network ls

NETWORK ID NAME DRIVER SCOPE

2b3558c906dc bridge bridge local

a14a667d26d0 docker\_gwbridge bridge local

833d02faa84a host host local

b4y08s9o14h3 ingress overlay swarm

f85d6f000fc8 none null local

Because of this overlay we are seeing this is launching the containers and talking with in different vm and containers.

root@mani-VirtualBox:~# netstat -tunlp

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 2488/sshd: /usr/sbi

tcp 0 0 127.0.0.53:53 0.0.0.0:\* LISTEN 393/systemd-resolve

tcp 0 0 127.0.0.1:631 0.0.0.0:\* LISTEN 661/cupsd

tcp6 0 0 :::2377 :::\* LISTEN 3833/dockerd

tcp6 0 0 :::22 :::\* LISTEN 2488/sshd: /usr/sbi

tcp6 0 0 :::80 :::\* LISTEN 3833/dockerd

demon d is exposed on 80

tcp6 0 0 ::1:631 :::\* LISTEN 661/cupsd

tcp6 0 0 :::7946 :::\* LISTEN 3833/dockerd

udp 0 0 0.0.0.0:5353 0.0.0.0:\* 557/avahi-daemon: r

udp 0 0 0.0.0.0:45520 0.0.0.0:\* 557/avahi-daemon: r

udp 0 0 0.0.0.0:631 0.0.0.0:\* 808/cups-browsed

udp 0 0 0.0.0.0:4789 0.0.0.0:\* -

udp 0 0 127.0.0.53:53 0.0.0.0:\* 393/systemd-resolve

udp6 0 0 :::5353 :::\* 557/avahi-daemon: r

udp6 0 0 :::45790 :::\* 557/avahi-daemon: r

udp6 0 0 :::7946 :::\* 3833/dockerd

service to demon 3833/dockerd---🡪 so service can be communicated through it.