



Summary

Session No – 16

- When we have multi-tier architecture or 3-tier architecture & one single container will not work i.e. if one container goes down the entire architecture goes down example MySQL database & word press
- In multi-tier architecture, if one container goes down an entire architecture or website goes down
- In Swarm, we have to give a task. It can be a single container or multiple containers
- If we use docker in it we have an image from it we launch the container so docker can see & manage the container directly
- The minimum thing Swarm can manage is the task
- If due to any reason, the base system goes down & container is deleted swarm will automatically launch the container in another node it is called fault tolerance
- Swarm cluster gives the capability to make the exact copy of the task in another node this concept is called replication
- Let's say we are running the same website in different nodes we can place a program in between the client and the server which has the capability to take the request from the client and connect to the website or node. This program is called a load balancer
- Swarm gives us a pre-created load balancer known as a service
- Whenever we launch a task swarm will always launch one service which is known as a load balancer

- **docker node ls** command is used to list the nodes in the cluster

```
[root@ip-172-31-2-3 ~]# docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS
ENGINE VERSION uvykc1f39gywckh2ycliwc2zs *	ip-172-31-2-3.ap-south-1.compute.internal	Ready	Active	Leader
20.10.17 idxdku7aor668iuol8q8yq2tq	ip-172-31-3-99.ap-south-1.compute.internal	Ready	Active	
20.10.17 laxtnyozznf4g0157bpa2tldy	ip-172-31-4-217.ap-south-1.compute.internal	Ready	Active	

```
[root@ip-172-31-2-3 ~]#
```

- Creating service

- Command:- **docker service create --name (Service name) (image name)**

```
[root@ip-172-31-2-3 ~]# docker service create --name myweb httpd
9sf910celait4ctqbflha3imb
overall progress: 0 out of 1 tasks
1/1: preparing [=====]>
```

- **docker service ls** command is used to list all services

```
[root@ip-172-31-2-3 ~]# docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
9sf910celait	myweb	replicated	1/1	httpd:latest	

```
[root@ip-172-31-2-3 ~]#
```

- **docker service ps** command is used to know more detail about the service

```
[root@ip-172-31-2-3 ~]# docker service ps myweb
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE
lhqrhh8ybix8	myweb.1	httpd:latest	ip-172-31-4-217.ap-south-1.compute.internal	Running	Running 3 minute

```
s ago
```

- On one of the nodes, we can see the container is created

```
[root@ip-172-31-4-217 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
47b16ae078a9	httpd:latest	"httpd-foreground"	4 minutes ago	Up 4 minutes	80/tcp	myweb.1.lhqrhh8ybix8mqlk9d

```
nz2so5
[root@ip-172-31-4-217 ~]#
```

- Now the duty of the manager node is to monitor the container & if it does down or removed recreate it
- Deleting the container manually

```
[root@ip-172-31-4-217 ~]# docker rm -f 47b16ae078a9
47b16ae078a9
```

- If we go to the manager node & see the details of the service we can see it is automatically detected & relaunched in another node

```
[root@ip-172-31-2-3 ~]# docker service ps myweb
ID                NAME          IMAGE          NODE                                     DESIRED STATE  CURRENT STATE
a54rz2ho5uyq      myweb.1       httpd:latest   ip-172-31-2-3.ap-south-1.compute.internal  Running        Running 37 s
econds ago
lhrqhh8ybix8      myweb.1       httpd:latest   ip-172-31-4-217.ap-south-1.compute.internal  Shutdown       Failed 43 se
conds ago
"task: non-zero exit (137)"
[root@ip-172-31-2-3 ~]#
```

- By default nature of the service is that it is isolated or private which means anyone from the internet who tries to connect to the service will not connect to the container
- **docker service inspect (service name)** command is used to know more detail about the service

```
[root@ip-172-31-2-3 ~]# docker service inspect myweb
[
  {
    "ID": "9sf910celait4ctqbf1ha3imb",
    "Version": {
      "Index": 210964
    },
    "CreatedAt": "2022-11-22T16:17:50.613750116Z",
    "UpdatedAt": "2022-11-22T16:17:50.613750116Z",
    "Spec": {
      "Name": "myweb",
      "Labels": {},
      "TaskTemplate": {
        "ContainerSpec": {
          "Image": "httpd:latest@sha256:f2e89def4c032b02c83e162c1819ccfcbd4ea6bdbc"
```

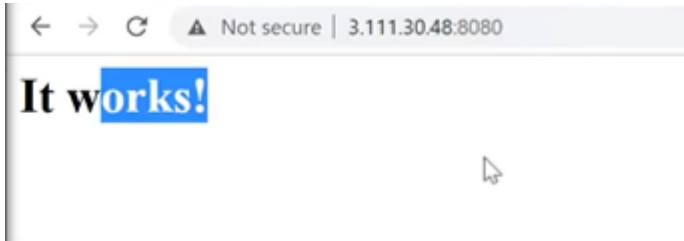
- **--publish** keyword in docker service command is used for exposing it to the outside world
 - Command:- **docker service create --name (service name) --publish 8080:80 (image name)**

```
[root@ip-172-31-2-3 ~]# docker service create --name myweb --publish 8080:80 httpd
jh461y5219315r2wupeapgv17
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Waiting 3 seconds to verify that tasks are stable...
```

- Connecting to the website with the curl command

```
[root@ip-172-31-2-3 ~]# curl http://172.31.2.3:8080
<html><body><h1>It works!</h1></body></html>
[root@ip-172-31-2-3 ~]#
```

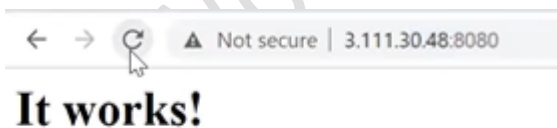
- Connecting to the website from the browser



- Between the client and the task, we have a load balancer running which is taking the request from the client and connecting to the task. The task gives the reply to the load balancer & load balancer gives to the client
- If we delete the container manually client won't see the dysconnectivity

```
[root@ip-172-31-4-217 ~]# docker rm -f 582f32f8f41c
582f32f8f41c
[root@ip-172-31-4-217 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
[root@ip-172-31-4-217 ~]#
```

- Accessing the webpage from the browser



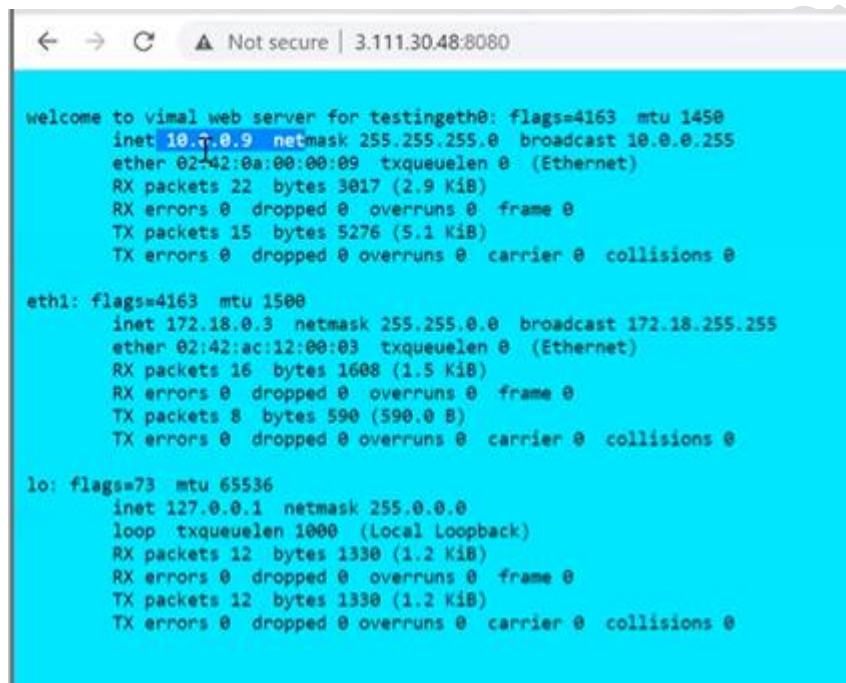
- As soon as the container is deleted swarm master automatically launches another container with the same service
- As soon as we remove the service client won't be able to see the website

```
[root@ip-172-31-2-3 ~]# docker service rm myweb
myweb
[root@ip-172-31-2-3 ~]#
```

- Creating service from **vimal13/apache-webserver-php** image
 - Command:- **docker service create --name(service name) --publish 8080:80 vimal13/apache-webserver-php**

```
[root@ip-172-31-2-3 ~]# docker service create --name myweb --publish 8080:80 vimal13/apache-webserver-php
td9yevstpebnt4rnh6fyzdu
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Service converged
[root@ip-172-31-2-3 ~]# docker service ls
ID NAME MODE REPLICAS IMAGE PORTS
td9yevstpebn myweb replicated 1/1 vimal13/apache-webserver-php:latest *:8080->80/tcp
[root@ip-172-31-2-3 ~]# docker service ps myweb
ID NAME IMAGE NODE DESIRED
0klv16iigy0i myweb.1 vimal13/apache-webserver-php:latest ip-172-31-4-217.ap-south-1.compute.internal Running
Running 21 seconds ago
[root@ip-172-31-2-3 ~]#
```

- Connecting to the website from the browser



The screenshot shows a web browser window with the address bar displaying "3.111.30.48:8080". The page content is a terminal output showing network interface details for eth0, eth1, and lo. The output includes IP addresses, netmasks, broadcast addresses, and various network statistics.

```
welcome to vimal web server for testing eth0: flags=4163 mtu 1450
inet 10.0.0.9 netmask 255.255.255.0 broadcast 10.0.0.255
ether 02:42:0a:00:00:09 txqueuelen 0 (Ethernet)
RX packets 22 bytes 3017 (2.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 15 bytes 5276 (5.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

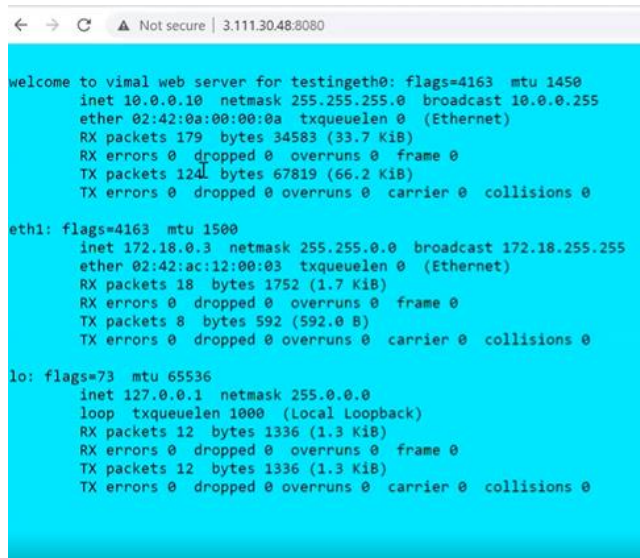
eth1: flags=4163 mtu 1500
inet 172.18.0.3 netmask 255.255.0.0 broadcast 172.18.255.255
ether 02:42:ac:12:00:03 txqueuelen 0 (Ethernet)
RX packets 16 bytes 1608 (1.5 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 590 (590.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 12 bytes 1330 (1.2 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12 bytes 1330 (1.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Now if we delete the container manually service will launch the container automatically

```
[root@ip-172-31-4-217 ~]# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
01a64f235a69 vimal13/apache-webserver-php:latest "/usr/sbin/httpd -DF..." About a minute ago Up About a minute
80/tcp myweb.1.0klv16iigy0ib2ma0t7bjg6j6
[root@ip-172-31-4-217 ~]# docker rm -f 01a6
01a6
[root@ip-172-31-4-217 ~]#
```


- Now this time we connect to our website we can see the IP address has changed



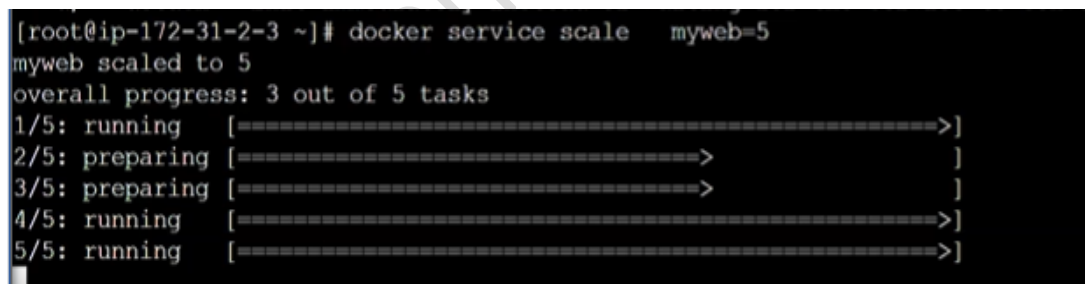
```

welcome to vimal web server for testing
eth0: flags=4163 mtu 1450
inet 10.0.0.10 netmask 255.255.255.0 broadcast 10.0.0.255
ether 02:42:0a:00:00:0a txqueuelen 0 (Ethernet)
RX packets 179 bytes 34583 (33.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 124 bytes 67819 (66.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163 mtu 1500
inet 172.18.0.3 netmask 255.255.0.0 broadcast 172.18.255.255
ether 02:42:ac:12:00:03 txqueuelen 0 (Ethernet)
RX packets 18 bytes 1752 (1.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 592 (592.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 12 bytes 1336 (1.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 12 bytes 1336 (1.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    
```

- **scale** keyword in the docker service is used for horizontally scaling the service
 - Command:- **docker service scale (Service name)=5**



```

[root@ip-172-31-2-3 ~]# docker service scale myweb=5
myweb scaled to 5
overall progress: 3 out of 5 tasks
1/5: running [=====>]
2/5: preparing [=====]
3/5: preparing [=====]
4/5: running [=====>]
5/5: running [=====>]
    
```

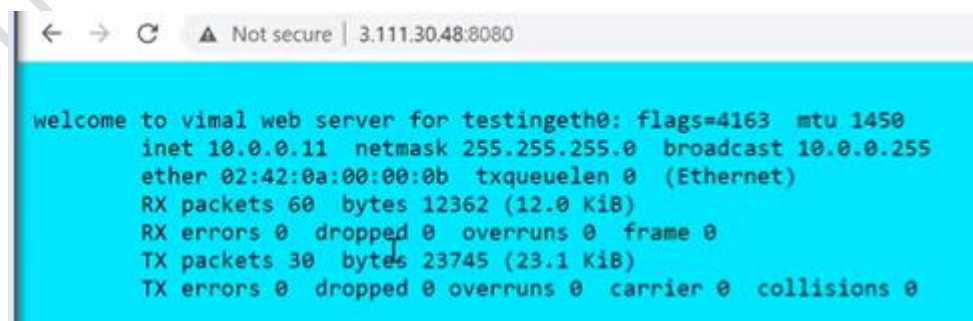
- Listing the service



```

[root@ip-172-31-2-3 ~]# docker service ls
ID            NAME      MODE      REPLICAS  IMAGE                                  PORTS
td9yevstpebn myweb     replicated 5/5        vimal13/apache-webserver-php:latest  *:8080->80/tcp
    
```

- Now if we refresh the web page we can see the IP address is changing



```

welcome to vimal web server for testing
eth0: flags=4163 mtu 1450
inet 10.0.0.11 netmask 255.255.255.0 broadcast 10.0.0.255
ether 02:42:0a:00:00:0b txqueuelen 0 (Ethernet)
RX packets 60 bytes 12362 (12.0 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 30 bytes 23745 (23.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    
```

```

welcome to vimal web server for testing
eth0: flags=4163 mtu 1450
inet 10.0.0.10 netmask 255.255.255.0 broadcast 10.0.0.255
ether 02:42:0a:00:00:0a txqueuelen 0 (Ethernet)
RX packets 352 bytes 68952 (67.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 239 bytes 139323 (136.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

- The good thing about scaling is in just one click we can scale the task or container

```

[root@ip-172-31-2-3 ~]# docker service scale myweb=10
myweb scaled to 10
overall progress: 6 out of 10 tasks
1/10: running [=====>]
2/10: running [=====>]
3/10: running [=====>]
4/10: starting [=====>]
5/10: running [=====>]
6/10: running [=====>]
7/10: starting [=====>]
8/10: starting [=====>]
9/10: starting [=====>]
10/10: running [=====>]

```

- Now if we see the service we can see 10 containers are running

```

[root@ip-172-31-2-3 ~]# docker service ps myweb

```

ID	NAME	IMAGE	ERROR	NODE	DESIR
lsgxstb626k4	myweb.1	vimal13/apache-webserver-php:latest		ip-172-31-3-99.ap-south-1.compute.internal	Runni
ng			Running 11 minutes ago		
0klv16iigy0i	myweb.1	vimal13/apache-webserver-php:latest		ip-172-31-4-217.ap-south-1.compute.internal	Shutd
own			Failed 11 minutes ago "task: non-zero exit (137)"		
miz8aumej06l	myweb.2	vimal13/apache-webserver-php:latest		ip-172-31-2-3.ap-south-1.compute.internal	Runni
ng			Running 4 minutes ago		
0rpbw7nbdfxk	myweb.3	vimal13/apache-webserver-php:latest		ip-172-31-3-99.ap-south-1.compute.internal	Runni
ng			Running 5 minutes ago		
m518v8x3wxgy	myweb.4	vimal13/apache-webserver-php:latest		ip-172-31-4-217.ap-south-1.compute.internal	Runni
ng			Running 5 minutes ago		
yuckvygauxzp	myweb.5	vimal13/apache-webserver-php:latest		ip-172-31-2-3.ap-south-1.compute.internal	Runni
ng			Running 4 minutes ago		
lhklmmp67gyu	myweb.6	vimal13/apache-webserver-php:latest		ip-172-31-4-217.ap-south-1.compute.internal	Runni
ng			Running 16 seconds ago		
1jq6w1mikew	myweb.7	vimal13/apache-webserver-php:latest		ip-172-31-3-99.ap-south-1.compute.internal	Runni

- Adding more containers or instances are called scale-out & removing the containers or instances is called scale in
- Scaling-in the container

```

[root@ip-172-31-2-3 ~]# docker service scale myweb=4
myweb 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

```