# Docker network

1. First we launch the one instance in docker.
2. **Sudo su -**
3. Update the linux **sudo update –y**
4. Next, we will install docker follow the below commands
5. **yum install docker –y**
6. **Systemctl start docker**
7. **Systemctl status docker**
8. **sudo docker run -it ubuntu**
9. **Apt update**
10. Install the docker we will check the how many networks you have ,use the below command ,see the network list
11. **docker network ls**

```
[root@ip-172-31-7-88 ~]# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a9ddc4d1da38   bridge    bridge    local
deef60eb9a31   host      host      local
2f894a6801be   none      null      local
```

12.
13. **docker run –it ubuntu**
14. Back to root user **exit**
15. See the container ls **docker ps ,docker ps –a**
16. Inspect the container **docker inspect 73c0e1ed6d9d**


17.

18. Create the network **docker network create devops**
19. Go to check the network it will add or not **docker network ls**

```
[root@ip-172-31-7-88 ~]# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
a9ddc4d1da38   bridge    bridge    local
194f1638c849   devops    bridge    local
deef60eb9a31   host      host      local
2f894a6801be   none      null      local
[root@ip-172-31-7-88 ~]#
```

20.

21. **Docker run –it –-name cont1 –-network devops ubuntu**
22. Back to root user use **exit** command
23. Next see the continer list **docker ps –a**
24. Inspect the container **docker inspect cont1**
25. **Docker attach cont1**
26. First update **apt update –y ,**
27. Next install ping **apt install iputils-ping**
28. See the networking status **ping goole.com**

```
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): i
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): i
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): i
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): i
C
-- goole.com ping statistics ---
7 packets transmitted, 57 received, 0% packet loss, time 56084ms
tt min/avg/max/mdev = 131.467/131.871/132.384/0.265 ms
oot@3c94e027d498:/#
```

29.

30. Back to root user **Exit, docker ps –a**
31. Next we will attach the exsiting **container docker attach**
32. First we need to update **apt update**
33. **apt install iputils-ping**
34. **Ping goole.com, back to root user exit**
35. **Docker ps –a**
36. **Docker inspect 73c0e1ed6d9d**
37. **Docker attach cont1**
38. ping ip addres cont 1

```
4 bytes from 3c94e027d498 (172.18.0.2): icmp_seq=11 ttl=127 time=0.032
4 bytes from 3c94e027d498 (172.18.0.2): icmp_seq=12 ttl=127 time=0.031
4 bytes from 3c94e027d498 (172.18.0.2): icmp_seq=13 ttl=127 time=0.037
4 bytes from 3c94e027d498 (172.18.0.2): icmp_seq=14 ttl=127 time=0.035
C
-- cont1 ping statistics ---
4 packets transmitted, 14 received, 0% packet loss, time 13493ms
tt min/avg/max/mdev = 0.011/0.059/0.416/0.099 ms
oot@3c94e027d498:/#
```

39.

40. **Exit, docker ps –a**
41. Copy the defalut container

42. **docker network connect devops naughty_chebyshev**
43. **Docker inspect 73c0e1ed6d9d**
44. **Docker attach cont1**
45. **Ping ip adress**
46. **Exit** back to root user
47. We can remove un used networks **docker network prune** this will remove all custom network.

```
[root@ip-172-31-7-88 ~]# docker network prune
WARNING! This will remove all custom networks not used by at least
Are you sure you want to continue? [y/N] y
Deleted Networks:
devops
```

48.
49. **Docker rm container id**

# Docker swarm

Docker Swarm is a native clustering and orchestration tool for Docker containers. It allows you to manage a cluster of Docker hosts, which can be used to deploy and scale applications across multiple machines.

1. initialize swarm **docker swarm init**
2.

```
root@ip-172-31-7-88 ~]# docker swarm init
warm initialized: current node (2z6gv7pg49vempevtdeap622l) is now a manager.

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

   docker swarm join --token SWMTKN-1-3qd3tps3dfehr06bscsg5zd7gqn2mun6lb25chyq0bg6

o add a manager to this swarm, run 'docker swarm join-token manager' and follow th

root@ip-172-31-7-88 ~]#
```

3.next we edit inbound rule alltrafic,anywhere

4. next we will launch another worker ec2 instance , edit inbound rule allow alltrafic.

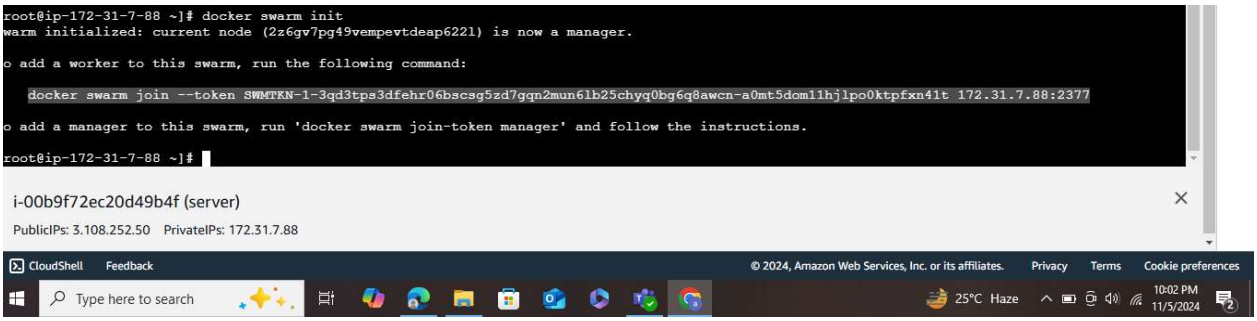
5. **sudo su -**

6. install docker **yum install docker –y**

7. **systemctl start docker**

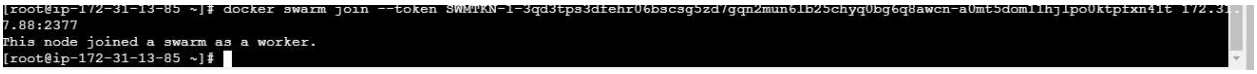
**8. systemctl status docker**

9. copy the docker swarm token




show in above image.

10.




11. Go to the server ec2 instance use this command see the how many nodes

**Docker node ls**

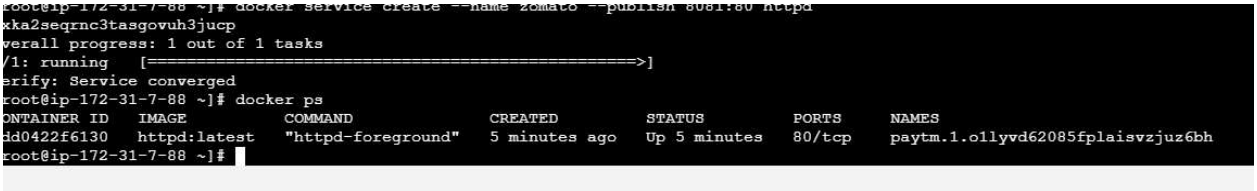

12. we need to create the service to the server ec2 ,this command

**Docker service create —-name paytm —publish 8080:80 httpd**

**Docker ps**

13. Create another change only prot and name

**docker service create --name zomato --publish 8081:80 httpd**




see in worker ec2 service created or not


14. **docker service ls (dockerec2)**

**docker service create --name hotstar --publish 8082:80 nginx**

To create service that publish service you require the image,image is nginx

15. See the worker ec2 create nginx or not

**Docker ps**

```
root@ip-172-31-13-85 ~]# docker ps
ONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
e356debccdc   nginx:latest   "/docker-entrypoint.…"   3 minutes ago    Up 3 minutes    80/tcp    hotstar.1.ucvndvut34s4krlomk920j817
2dcc6228ee8   httpd:latest   "httpd-foreground"       16 minutes ago   Up 16 minutes   80/tcp    zomato.1.ig38398umzqqbo6k04hvq1f75
```

16. We have only one service in server ec2

```
root@ip-172-31-7-88 ~]# docker ps
ONTAINER ID   IMAGE          COMMAND              CREATED          STATUS          PORTS     NAMES
dd0422f6130   httpd:latest   "httpd-foreground"   23 minutes ago   Up 23 minutes   80/tcp    paytm.1.o11yvd62085fplaisvzjuz6bh
```

17. Modify and update

**Docker service update --image nginx hotstar**

```
ee 'docker service update --help'.
root@ip-172-31-7-88 ~]# docker service update --image nginx hotstar
otstar
verall progress: 1 out of 1 tasks
/1: running   [==================================================>]
erify: Service converged
root@ip-172-31-7-88 ~]#
```

18. **Docker ps (worker)** its update

```
root@ip-172-31-13-85 ~]# docker ps
ONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS     NAMES
e356debccdc   nginx:latest   "/docker-entrypoint.…"   15 minutes ago   Up 15 minutes   80/tcp    hotstar.1.ucvndvut34s4krlomk920j817
2dcc6228ee8   httpd:latest   "httpd-foreground"       28 minutes ago   Up 28 minutes   80/tcp    zomato.1.ig38398umzqqbo6k04hvq1f75
root@ip-172-31-13-85 ~]#
```