

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. **exit**
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**

```
"IPv6Gateway": "",
"MacAddress": "",
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "MacAddress": "",
    "NetworkID": "a9ddc4d1da3810fdcbaa079c56bdec425337bac5de9110d2d0b7dcb5a8cb43ed",
    "EndpointID": "",
    "Gateway": "",
    "IPAddress": "",
    "IPPrefixLen": 0,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "DriverOpts": null,
    "DNSNames": null
  }
}
```

17. **exit**
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

20. [root@ip-172-31-7-88 ~]#

21. Docker run -it --name cont1 --network devops ubuntu

22. Back to root user use **exit** command

23. Next see the container 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 google.com**

```

4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): 3
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): 3
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): 3
4 bytes from 217-160-0-201.elastic-ssl.ui-r.com (217.160.0.201): 3
C
-- google.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 existing container **docker attach**

32. First we need to update **apt update**

33. **apt install iputils-ping**

34. Ping google.com, back to root user **exit**

35. Docker ps -a

36. Docker inspect 73c0e1ed6d9d

37. Docker attach cont1

38. ping ip address 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. oot@3c94e027d498:/#

40. Exit, **docker ps -a**

41. Copy the default 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 (2z6gv7pg49vempevtdeap6221) is now a manager.

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

    docker swarm join --token SWMTKN-1-3qd3tps3dfehr06bscsg5zd7gqn2mun61b25chyq0bg6

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 alltraffic, anywhere

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

5. **sudo su -**

6. install docker **yum install docker -y**

7. **systemctl start docker**

8. **systemctl status docker**

9. copy the docker swarm token

```
root@ip-172-31-7-88 ~]# docker swarm init
swarm initialized: current node (2z6gv7pg49vempevtdeap6221) is now a manager.

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

    docker swarm join --token SWMTKN-1-3qd3tps3dfehr06bscsg5zd7gqn2mun61b25chyg0bg6q8awcn-a0mt5dcm1lhjlp0ktpfxn41t 172.31.7.88:2377

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

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

i-00b9f72ec20d49b4f (server)
PublicIPs: 3.108.252.50 PrivateIPs: 172.31.7.88

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Type here to search 25°C Haze 10:02 PM 11/5/2024

show in above image.

10.

```
root@ip-172-31-13-85 ~]# docker swarm join --token SWMTKN-1-3qd3tps3dfehr06bscsg5zd7gqn2mun61b25chyg0bg6q8awcn-a0mt5dcm1lhjlp0ktpfxn41t 172.31.7.88:2377
This node joined a swarm as a worker.
root@ip-172-31-13-85 ~]#
```

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

Docker node ls

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

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
z6gv7pg49vempevtdeap6221 *	ip-172-31-7-88.ap-south-1.compute.internal	Ready	Active	Leader	25.0.6
v2cxrulepw879ckew4kt49ni	ip-172-31-13-85.ap-south-1.compute.internal	Ready	Active		25.0.6

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

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

```
root@ip-172-31-7-88 ~]# docker service create --name zomato --publish 8081:80 httpd
xka2segrnc3tasgovuh3jucp
verall progress: 1 out of 1 tasks
/1: running [=====>]
erify: Service converged
root@ip-172-31-7-88 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
dd0422f6130	httpd:latest	"httpd-foreground"	5 minutes ago	Up 5 minutes	80/tcp	paytm.1.o1lyvd62085fp1aisvzjuz6bh

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

see in worker ec2 service created or not

```
root@ip-172-31-13-85 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2dccc6228ee8	httpd:latest	"httpd-foreground"	2 minutes ago	Up 2 minutes	80/tcp	zomato.1.ig38398umzqqbo6k04hqv1f75

```
root@ip-172-31-13-85 ~]#
```

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
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e356debccdc	nginx:latest	"/docker-entrypoint..."	3 minutes ago	Up 3 minutes	80/tcp	hotstar.1.ucvndvut34s4krlomk920j817
2dccc6228ee8	httpd:latest	"httpd-foreground"	16 minutes ago	Up 16 minutes	80/tcp	zomato.1.ig38398umzqqbo6k04hvg1f75

16. We have only one service in server ec2

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
dd0422f6130	httpd:latest	"httpd-foreground"	23 minutes ago	Up 23 minutes	80/tcp	paytm.1.o1lyvd62085fplaisvzjuz6bh

17. Modify and update

Docker service update --image nginx hotstar

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e356debccdc	nginx:latest	"/docker-entrypoint..."	15 minutes ago	Up 15 minutes	80/tcp	hotstar.1.ucvndvut34s4krlomk920j817
2dccc6228ee8	httpd:latest	"httpd-foreground"	28 minutes ago	Up 28 minutes	80/tcp	zomato.1.ig38398umzqqbo6k04hvg1f75