Summary

Session No - 6

- Docker is a tool that not only helps us in running the container it is used to provide the entire network infrastructure
- If we run the **Docker info** command we can see docker has the network plugin & volume plugin which help us to provide the network infrastructure & storage to a container

```
Cgroup Driver: cgroupfs
Cgroup Version: 1
Plugins:
 Volume: local
 Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
Default Runtime: runc
```

- The run time for the docker is runc
- The primary responsibility of the switch is to connect the operating systems with the same network name
- The primary responsibility of the router is to connect the operating systems with the different network name
- Combo devices work as a switch & router both, example is broadband or WIFI router.
- In the docker network bridge means with the help of the software we have created a device that works as L3 switch which has switch & router capabilities

```
[root@ip-172-31-40-68 ~]# docker network  ls
NETWORK ID     NAME      DRIVER    SCOPE
f36b89169326   bridge    bridge    local
cdbfd4363b7d   host      host      local
02b9e2f3299a   none      null      local
[root@ip-172-31-40-68 ~]#
```

- DHCP is a program that will provide the network settings (IP, Gateway) to a device when it is connected
- DNS is a program that gives the name with IP & this name can be used to ping with the name also
- DNS & DHCP both together is also known as IPAM (IP Address Management)
- Two containers can ping each other with names because the docker network has given an IPAM facility

```
[root@ip-172-31-40-68 ~]# docker network  inspect bridge
[
    {
        "Name": "bridge",
        "Id": "f36b891693265ccb50187a0283a9dcd63d3682a6fdb880c165efd0079fd29157",
        "Created": "2022-10-04T16:41:06.520827173Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
```

- A router has two network cards for public & private IP address
- The main network card in Linux is **eth0** through which we can go to the internet

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.40.68  netmask 255.255.240.0  broadcast 172.31.47.255
        inet6 fe80::9e:5bff:fef4:338a  prefixlen 64  scopeid 0x20<link>
        ether 02:9e:5b:f4:33:8a  txqueuelen 1000  (Ethernet)
        RX packets 2002161  bytes 1806846652 (1.6 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 797361  bytes 140604169 (134.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
```

- **docker0** is the private side network card of the router because of this network card we can ping the container from the base system

```
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:afff:fe45:2b83  prefixlen 64  scopeid 0x20<link>
        ether 02:42:af:45:2b:83  txqueuelen 0  (Ethernet)
        RX packets 61927  bytes 61031808 (58.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 66063  bytes 419847849 (400.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- By default, docker launches the container in the bridge network
- Creating a network in the docker
    - Command:- docker network create --driver bridge --subnet 10.1.2.0./24 (name of N/W)

```
[root@ip-172-31-40-68 ~]# docker network  create   --driver bridge  --subnet 10.1.2.0/24  lwnet
0bc48daf8a7f2075b1b52088d4e7c8a5a31e8f8d7b38fd73fb96e0de86844a31
[root@ip-172-31-40-68 ~]# docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
f36b89169326   bridge    bridge    local
cdbfd4363b7d   host      host      local
0bc48daf8a7f   lwnet     bridge    local
02b9e2f3299a   none      null      local
[root@ip-172-31-40-68 ~]#
```

- Launching the container in a custom network
    - -- network (Name of N/W) keyword is used in the run command to launch the container

```
[root@ip-172-31-40-68 ~]# docker run -dit --name os4  --network lwnet  ubuntu:14.04
5b1855f2b90774bd267fc154634d2c29be9b021645379f329fca8c788fc1e4a6
[root@ip-172-31-40-68 ~]# docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED              STATUS              PORTS     NAMES
5b1855f2b907   ubuntu:14.04   "/bin/bash"   3 seconds ago        Up 2 seconds                  os4
bdcb1a399245   ubuntu:14.04   "/bin/bash"   About a minute ago   Up About a minute             os3
692ac398a6d6   centos:7       "/bin/bash"   4 days ago           Up 4 days                     newos1
8ed5826a64f2   ubuntu:14.04   "/bin/bash"   4 days ago           Up 4 days                     myos2
[root@ip-172-31-40-68 ~]#
```

- The container has an IP 10.1.2.2 in this range because we have given the range to the IPAM while creating it

```
root@5b1855f2b907:/#
root@5b1855f2b907:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:0a:01:02:02
          inet addr:10.1.2.2  Bcast:10.1.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1080 (1.0 KB)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

- The IP address of the container started from 10.1.2.2 because 10.1.2.0 IP is reserved for the network name & 10.1.2.1 is reserved for the router's private IP address
- On the base system, a new network card is created which works as a router's private IP address

```
[root@ip-172-31-40-68 ~]# ifconfig
br-0bc48daf8a7f: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.2.1  netmask 255.255.255.0  broadcast 10.1.2.255
        inet6 fe80::42:ebff:fed6:d5d7  prefixlen 64  scopeid 0x20<link>
        ether 02:42:eb:d6:d5:d7  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 111  bytes 7766 (7.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- The two networks are isolated they do not have connectivity

```
root@2a0c4325f22c:/# ping  172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
^C
--- 172.17.0.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4075ms
```

- Whenever we create our own custom network it gives the facility to ping another container in the same network with the name

```
root@5b1855f2b907:/# ping vimalos1
PING vimalos1 (10.1.2.3) 56(84) bytes of data.
64 bytes from vimalos1.lwnet (10.1.2.3): icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from vimalos1.lwnet (10.1.2.3): icmp_seq=2 ttl=64 time=0.061 ms
^C
--- vimalos1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1004ms
```

- Launching word press application in custom network
  - **Command** :- docker run -dit --name mywp1 -p 8080:80 --network lwnet wordpress:latest

```
[root@ip-172-31-40-68 ~]#
[root@ip-172-31-40-68 ~]# docker run -dit  --name mywp1  -p  8080:80   --network  lwnet   wordpress:latest
74502456238c1769e3980cabf2280ee28d8b7bbde578e70e4f808ca42aea2eff
```

- Creating MY-SQL database in a container
  - **Command** :- docker run -dit --name mydb1 –network lwnet -e MYSQL_ROOT_PASSWORD=redhat -e MYSQL_DATABASE=mydb -e MYSQL_USER=jibbran -e MYSQL_PASSWORD=redhat -v /mydata:/var/lib/mysql mysql:latest

```
[root@ip-172-31-40-68 ~]# docker run -dit --name mydb1  --network lwnet   -e MYSQL_ROOT_PASSWORD=redhat -e   MYSQL_DAT
mal  -e MYSQL_PASSWORD=redhat  mysql:latest
aa0043315db6012ba31d4ff7029c4da49525805ad5a51d8f33a077831d265fb9
[root@ip-172-31-40-68 ~]# docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
              NAMES
aa0043315db6   mysql:latest       "docker-entrypoint.s…"   4 seconds ago    Up 3 seconds    3306/tcp, 33060/tcp
              mydb1
74502456238c   wordpress:latest   "docker-entrypoint.s…"   2 minutes ago    Up 2 minutes    0.0.0.0:8080->80/tcp, :::8
080->80/tcp   mywp1
```

- Connecting MySQL database to word press application