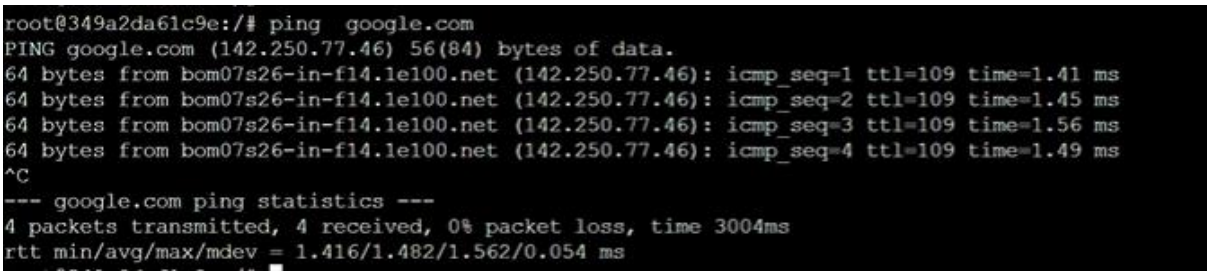
Summary

Session No – 5

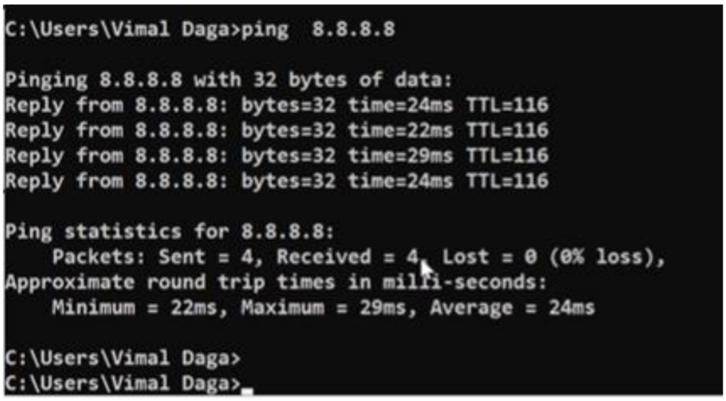
- Whenever we launch a container docker assigns the new IP address to the container
- By default, a container can ping the outside world on the internet



```
root@349a2da61c9e:/# ping  google.com
PING google.com (142.250.77.46) 56(84) bytes of data.
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=1 ttl=109 time=1.41 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=2 ttl=109 time=1.45 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=3 ttl=109 time=1.56 ms
64 bytes from bom07s26-in-f14.1e100.net (142.250.77.46): icmp_seq=4 ttl=109 time=1.49 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.416/1.482/1.562/0.054 ms
```

- If someone from the internet tries to connect the container it does not have connectivity but the container can connect to the outside world
- If two operating systems want to connect to each other the basic requirement is, both operating systems should have a network card, IP address & network connectivity
- Every IP address belongs to some network
- In the IP address 192.168.1.2 & 192.168.1.3 the initial three octets i.e. 192.168.1 is the network name
- If the network name is same, then both OS can connect to each other with the help of a device called a switch or wireless Hub, or network table
- If the network name is different, then both OS can connect to each other with the help of a device called a router

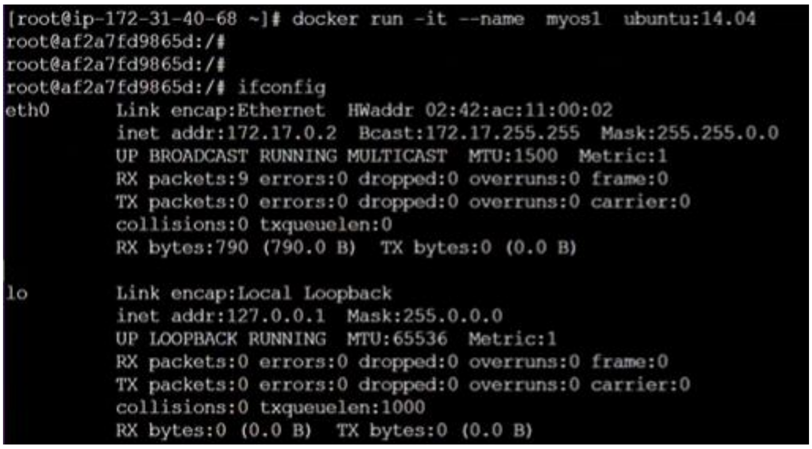
- Two IPs can ping each other because there is a device called a router in between



```
C:\Users\Vimal Daga>ping  8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=24ms TTL=116
Reply from 8.8.8.8: bytes=32 time=22ms TTL=116
Reply from 8.8.8.8: bytes=32 time=29ms TTL=116
Reply from 8.8.8.8: bytes=32 time=24ms TTL=116

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 22ms, Maximum = 29ms, Average = 24ms

C:\Users\Vimal Daga>
C:\Users\Vimal Daga>
```

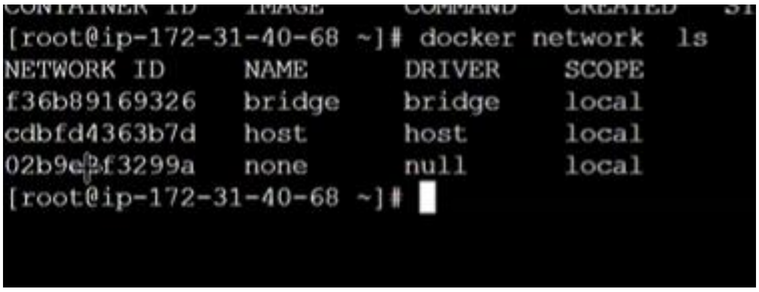
- There are two types of IP addresses public & private IP address
- The rule is private IP can only connect to Private IP & public IP can only connect to public IP.
- Private IP can not connect to public IP
- If we need public IP to connect anywhere on the internet
- If we want to connect private & public addresses then we have to do nating
- A router works as a Gateway to connect between public and private IP address
- A router has two IP addresses public & private
- Network packet contains the source address, destination address & data
- Network packet from the operating system goes to the router and it changes the source IP to the router’s public IP this changing of IP address is called nating or masquerade
- A software define network (SDN) is a tool or technology through which we can create a switch or router which works as a real physical hardware device
- Docker has created the entire network infrastructure with the help of software-defined networking

- As soon as we launch the container internally docker provides one network card to it



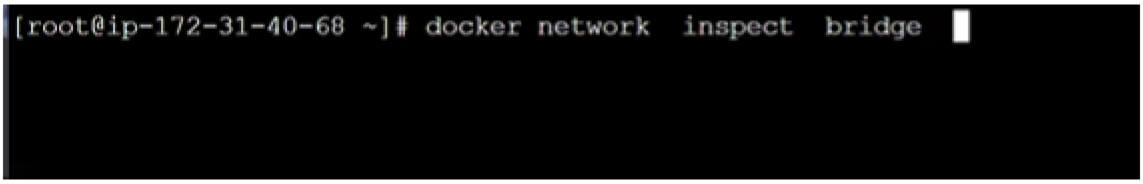
```
[root@ip-172-31-40-68 ~]# docker run -it --name  myos1  ubuntu:14.04
root@af2a7fd9865d:/#
root@af2a7fd9865d:/#
root@af2a7fd9865d:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 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:790 (790.0 B)  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)
```

- **docker network ls** is a command in docker to list all the network docker created by default



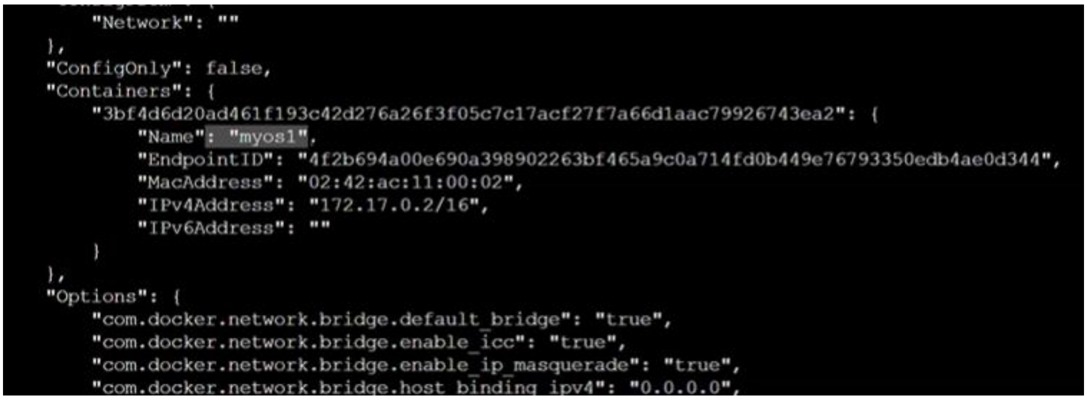
```
[root@ip-172-31-40-68 ~]# docker network  ls
NETWORK ID     NAME      DRIVER    SCOPE
f36b89169326   bridge    bridge    local
cdbfd4363b7d   host      host      local
02b9e8f3299a   none      null      local
[root@ip-172-31-40-68 ~]#
```

- **docker network inspect (network name)** command is used to see more detail about the network



```
[root@ip-172-31-40-68 ~]# docker network  inspect  bridge
```

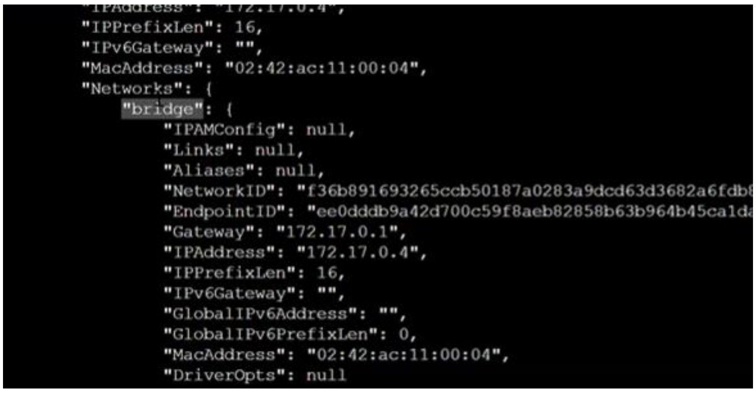
- We can see which container is attached to the network with the **docker network inspect (network name)** command



```
        "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
        "3bf4d6d20ad461f193c42d276a26f3f05c7c17acf27f7a66d1aac79926743ea2": {
            "Name": "myos1",
            "EndpointID": "4f2b694a00e690a398902263bf465a9c0a714fd0b449e76793350edb4ae0d344",
            "MacAddress": "02:42:ac:11:00:02",
            "IPv4Address": "172.17.0.2/16",
            "IPv6Address": ""
        }
    },
    "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",
```

- IPAM is a concept that is used for IP address management that helps in network assigning addresses which means any container docker launch it will be assigned with the IP address
- Docker by default launches containers in a bridge network

- --network keyword in the run command is used for launching containers in a particular network
- **docker inspect (container name)** command will tell us the IP address of the container & network name



```
            "IPAddress": "172.17.0.4",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "MacAddress": "02:42:ac:11:00:04",
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID": "f36b891693265ccb50187a0283a9dcd63d3682a6fdb
                    "EndpointID": "ee0dddb9a42d700c59f8aeb82858b63b964b45ca1d
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.4",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:04",
                    "DriverOpts": null
```