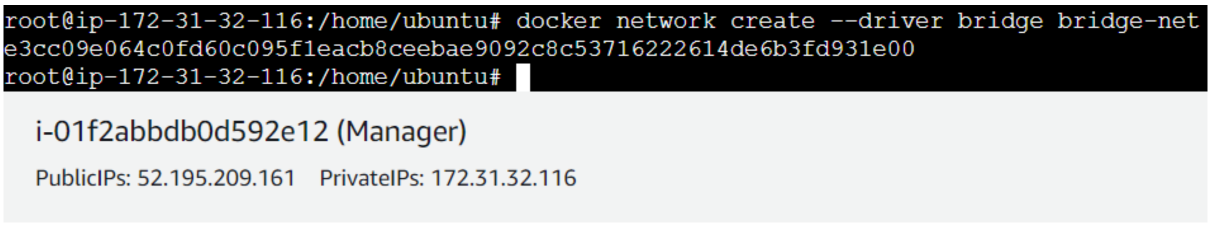
Checking the connection between two containers in a Bridge Network

1. Create a new bridge network

docker network create –driver bridge bridge-net



```
root@ip-172-31-32-116:/home/ubuntu# docker network create --driver bridge bridge-net
e3cc09e064c0fd60c095f1eacb8ceebae9092c8c53716222614de6b3fd931e00
root@ip-172-31-32-116:/home/ubuntu#
```

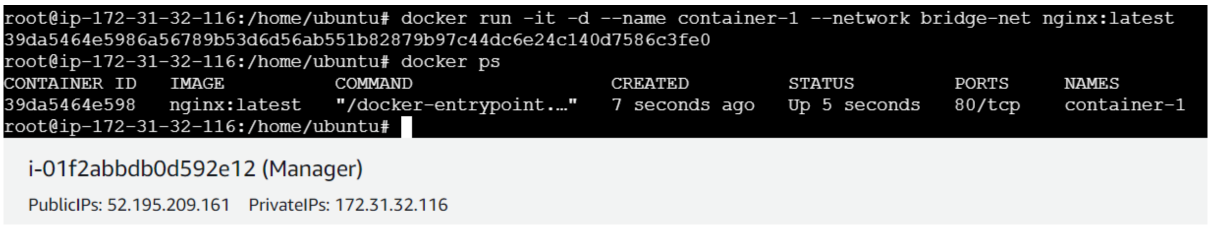
i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

2. Then create a new container attached to the user defined bridge network

docker run -it -d --name <container_name> –network <network_name> <image-name>

docker ps



```
root@ip-172-31-32-116:/home/ubuntu# docker run -it -d --name container-1 --network bridge-net nginx:latest
39da5464e5986a56789b53d6d56ab551b82879b97c44dc6e24c140d7586c3fe0
root@ip-172-31-32-116:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS     NAMES
39da5464e598   nginx:latest   "/docker-entrypoint.…"   7 seconds ago   Up 5 seconds   80/tcp    container-1
root@ip-172-31-32-116:/home/ubuntu#
```

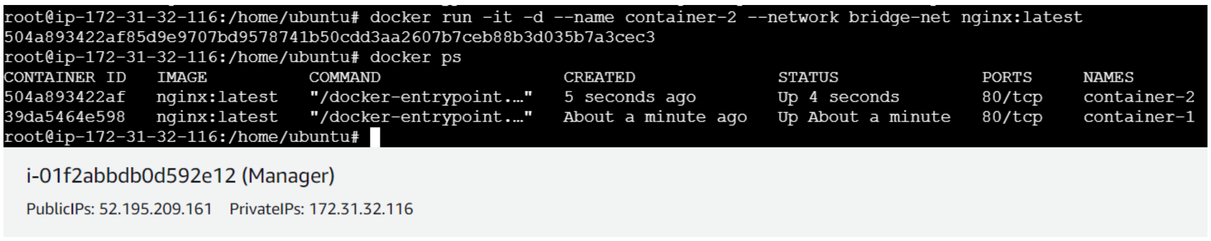
i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

Now will create our second container

docker run -it -d --name <container_name> –network <network_name> <image-name>

docker ps



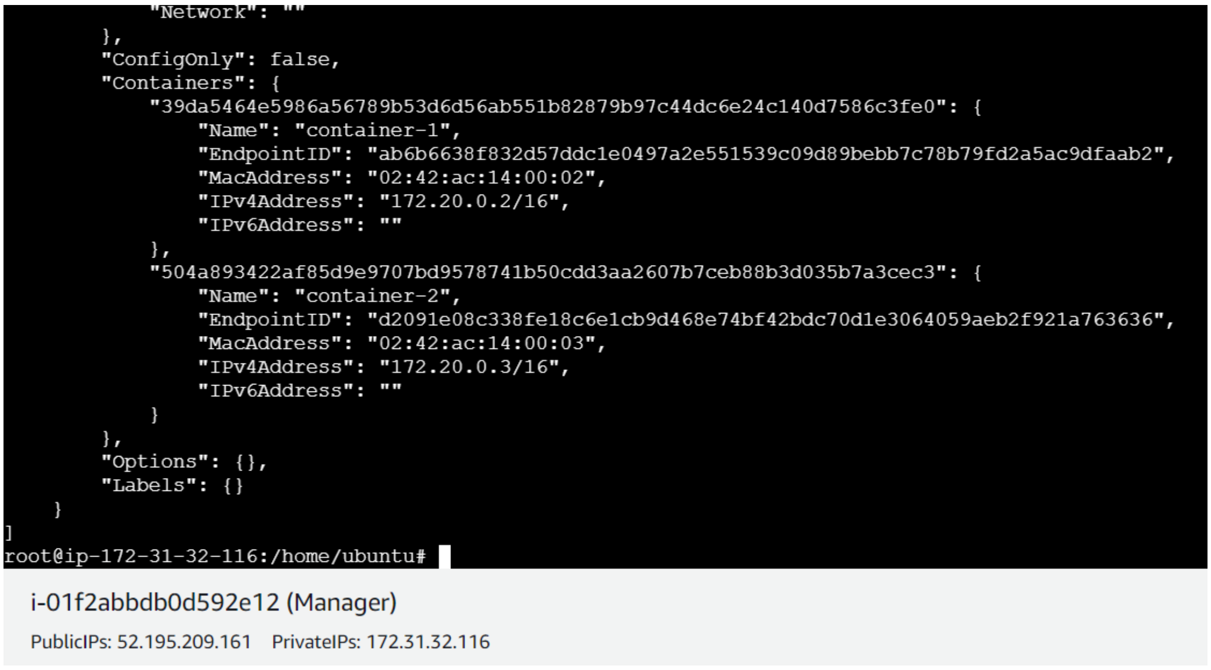
```
root@ip-172-31-32-116:/home/ubuntu# docker run -it -d --name container-2 --network bridge-net nginx:latest
504a893422af85d9e9707bd9578741b50cdd3aa2607b7ceb88b3d035b7a3cec3
root@ip-172-31-32-116:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS     NAMES
504a893422af   nginx:latest   "/docker-entrypoint.…"   5 seconds ago        Up 4 seconds        80/tcp    container-2
39da5464e598   nginx:latest   "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp    container-1
root@ip-172-31-32-116:/home/ubuntu#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

Inspect the user defined bridge network and check under the “Containers” to verify if the containers were attached successfully or not.

docker network inspect <network_name>



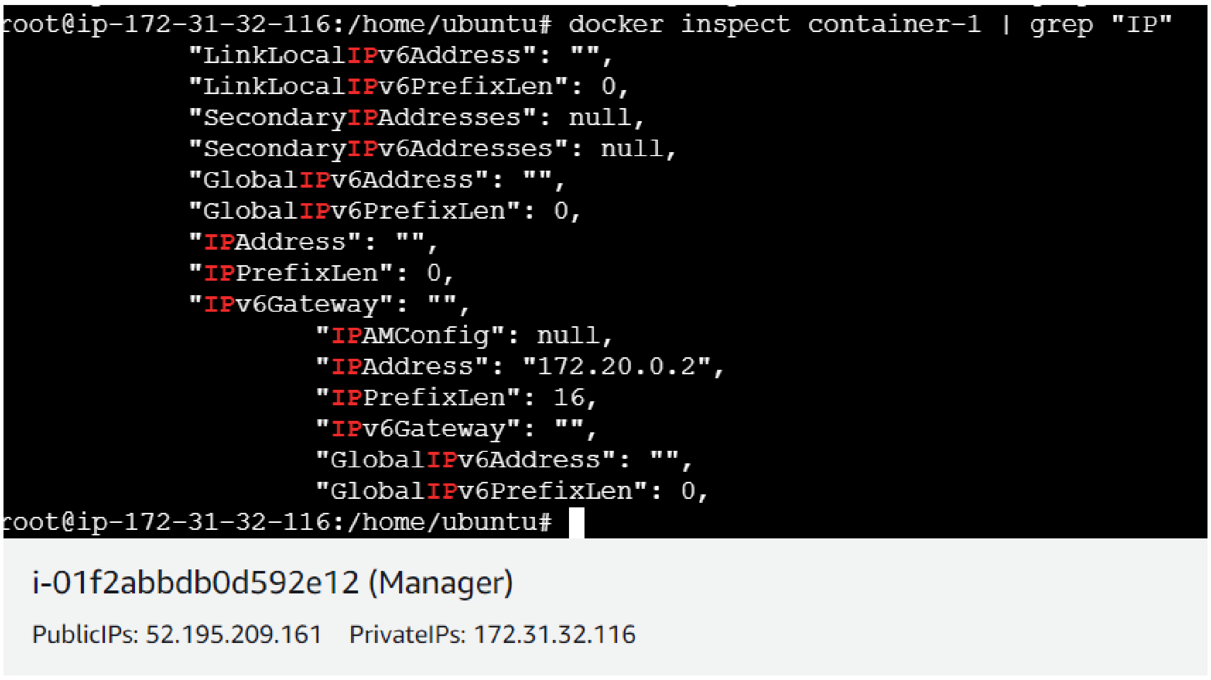
```
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "39da5464e5986a56789b53d6d56ab551b82879b97c44dc6e24c140d7586c3fe0": {
                "Name": "container-1",
                "EndpointID": "ab6b6638f832d57ddc1e0497a2e551539c09d89bebb7c78b79fd2a5ac9dfaab2",
                "MacAddress": "02:42:ac:14:00:02",
                "IPv4Address": "172.20.0.2/16",
                "IPv6Address": ""
            },
            "504a893422af85d9e9707bd9578741b50cdd3aa2607b7ceb88b3d035b7a3cec3": {
                "Name": "container-2",
                "EndpointID": "d2091e08c338fe18c6e1cb9d468e74bf42bdc70d1e3064059aeb2f921a763636",
                "MacAddress": "02:42:ac:14:00:03",
                "IPv4Address": "172.20.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
root@ip-172-31-32-116:/home/ubuntu#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

3. Get the container 1’s IP address using the command below and copy it so that you can use it later to ping from inside container 2.

docker inspect container-1 | grep “IP”



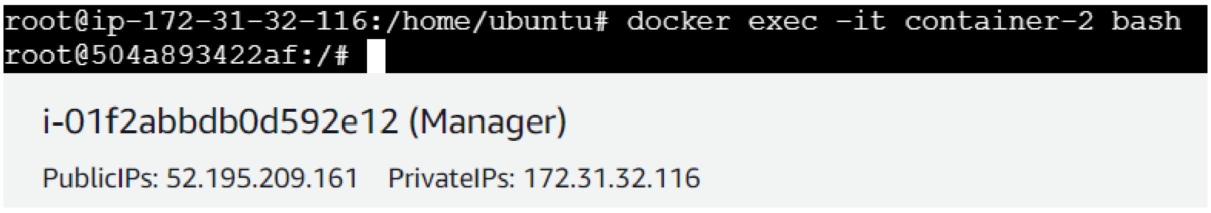
```
root@ip-172-31-32-116:/home/ubuntu# docker inspect container-1 | grep "IP"
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "",
            "IPPrefixLen": 0,
            "IPv6Gateway": "",
                    "IPAMConfig": null,
                    "IPAddress": "172.20.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
root@ip-172-31-32-116:/home/ubuntu#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

4. Go inside Container 2 using the below command and install the ping utility.

docker exec -it container-2 bash

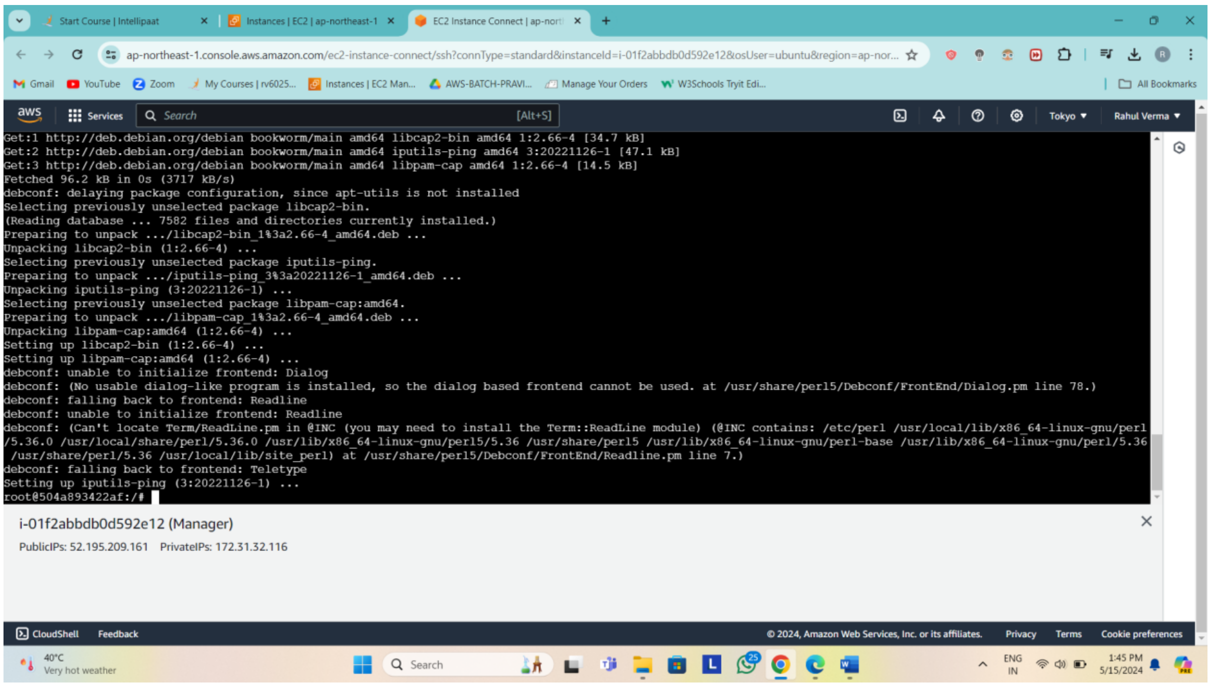


```
root@ip-172-31-32-116:/home/ubuntu# docker exec -it container-2 bash
root@504a893422af:/#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

apt-get update && apt-get install -y iputils-ping



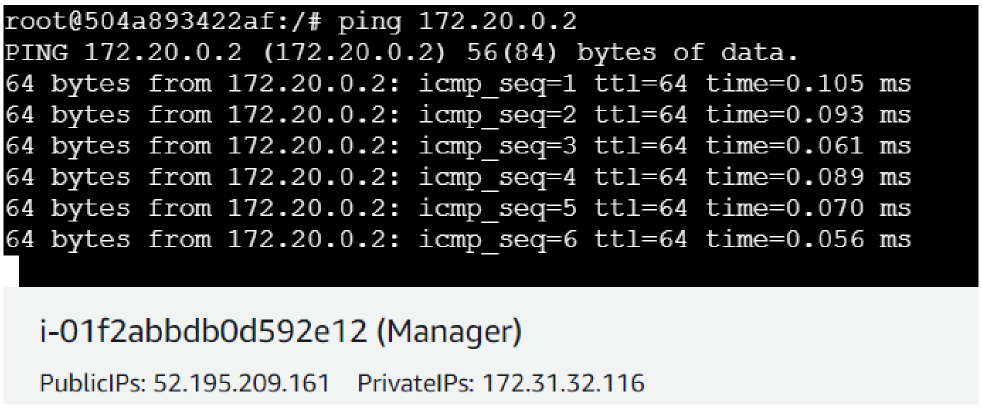
```
Get:1 http://deb.debian.org/debian bookworm/main amd64 libcap2-bin amd64 1:2.66-4 [34.7 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 iputils-ping amd64 3:20221126-1 [47.1 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 libpam-cap amd64 1:2.66-4 [14.5 kB]
Fetched 96.2 kB in 0s (3717 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libcap2-bin.
(Reading database ... 7582 files and directories currently installed.)
Preparing to unpack .../libcap2-bin_1%3a2.66-4_amd64.deb ...
Unpacking libcap2-bin (1:2.66-4) ...
Selecting previously unselected package iputils-ping.
Preparing to unpack .../iputils-ping_3%3a20221126-1_amd64.deb ...
Unpacking iputils-ping (3:20221126-1) ...
Selecting previously unselected package libpam-cap:amd64.
Preparing to unpack .../libpam-cap_1%3a2.66-4_amd64.deb ...
Unpacking libpam-cap:amd64 (1:2.66-4) ...
Setting up libcap2-bin (1:2.66-4) ...
Setting up libpam-cap:amd64 (1:2.66-4) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl
/5.36.0 /usr/local/share/perl/5.36.0 /usr/lib/x86_64-linux-gnu/perl5/5.36 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl-base /usr/lib/x86_64-linux-gnu/perl/5.36
 /usr/share/perl/5.36 /usr/local/lib/site_perl) at /usr/share/perl5/Debconf/FrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
Setting up iputils-ping (3:20221126-1) ...
root@504a893422af:/#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

5. now ping container 1

ping <container-ip>

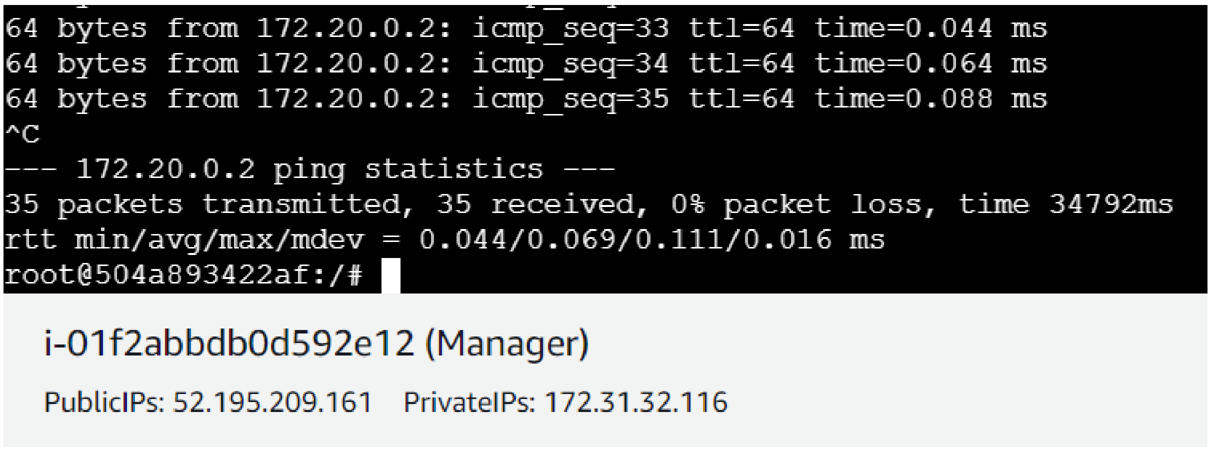


```
root@504a893422af:/# ping 172.20.0.2
PING 172.20.0.2 (172.20.0.2) 56(84) bytes of data.
64 bytes from 172.20.0.2: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 172.20.0.2: icmp_seq=2 ttl=64 time=0.093 ms
64 bytes from 172.20.0.2: icmp_seq=3 ttl=64 time=0.061 ms
64 bytes from 172.20.0.2: icmp_seq=4 ttl=64 time=0.089 ms
64 bytes from 172.20.0.2: icmp_seq=5 ttl=64 time=0.070 ms
64 bytes from 172.20.0.2: icmp_seq=6 ttl=64 time=0.056 ms
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

You Can see that the pings are successful which means that the network is working properly.



```
64 bytes from 172.20.0.2: icmp_seq=33 ttl=64 time=0.044 ms
64 bytes from 172.20.0.2: icmp_seq=34 ttl=64 time=0.064 ms
64 bytes from 172.20.0.2: icmp_seq=35 ttl=64 time=0.088 ms
^C
--- 172.20.0.2 ping statistics ---
35 packets transmitted, 35 received, 0% packet loss, time 34792ms
rtt min/avg/max/mdev = 0.044/0.069/0.111/0.016 ms
root@504a893422af:/#
```

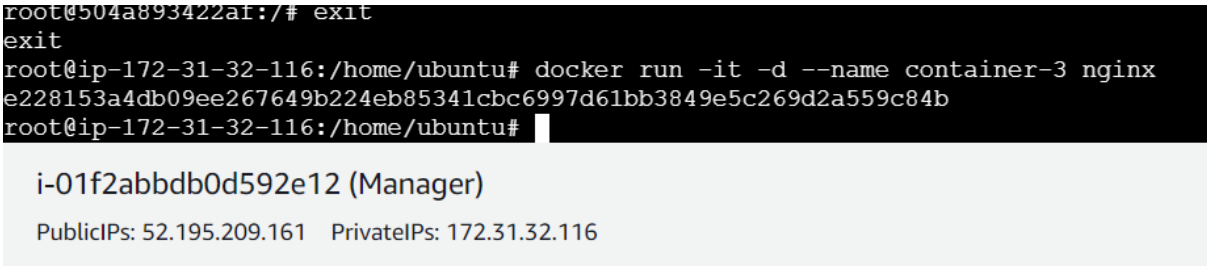
i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

6. To confirm that bridge networks isolate any containers attached to it, create a third container and DO NOT attach it to the user defined bridge network.

exit

docker run -it -d --name <container_name> <image>

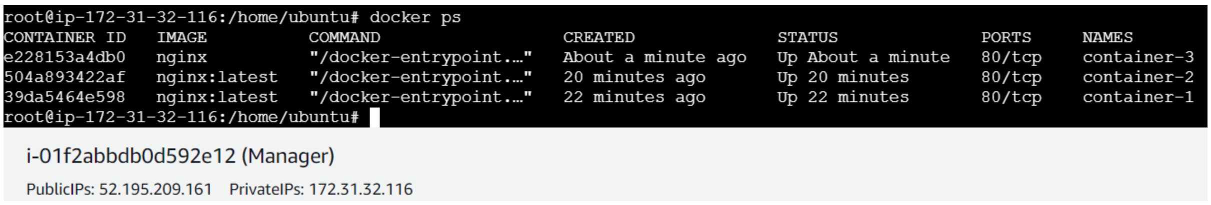


```
root@504a893422af:/# exit
exit
root@ip-172-31-32-116:/home/ubuntu# docker run -it -d --name container-3 nginx
e228153a4db09ee267649b224eb85341cbc6997d61bb3849e5c269d2a559c84b
root@ip-172-31-32-116:/home/ubuntu#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

docker ps



```
root@ip-172-31-32-116:/home/ubuntu# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS     NAMES
e228153a4db0   nginx          "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp    container-3
504a893422af   nginx:latest   "/docker-entrypoint.…"   20 minutes ago       Up 20 minutes       80/tcp    container-2
39da5464e598   nginx:latest   "/docker-entrypoint.…"   22 minutes ago       Up 22 minutes       80/tcp    container-1
root@ip-172-31-32-116:/home/ubuntu#
```

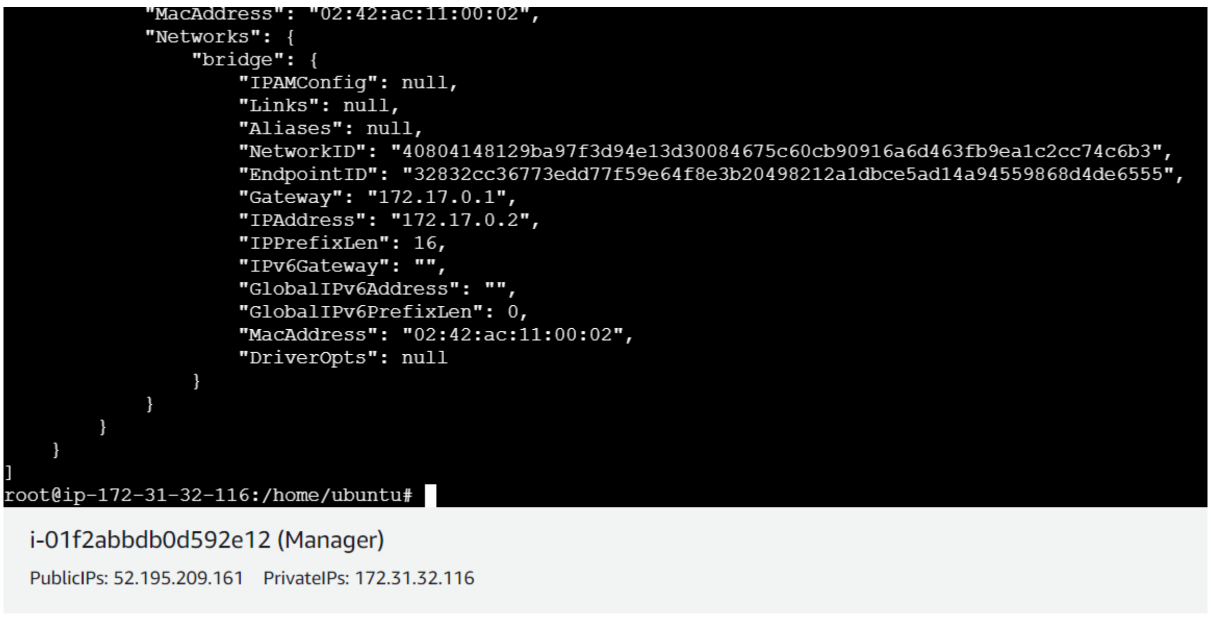
i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

Inspect container-3

docker inspect <container_name>

so here we have default bridge network



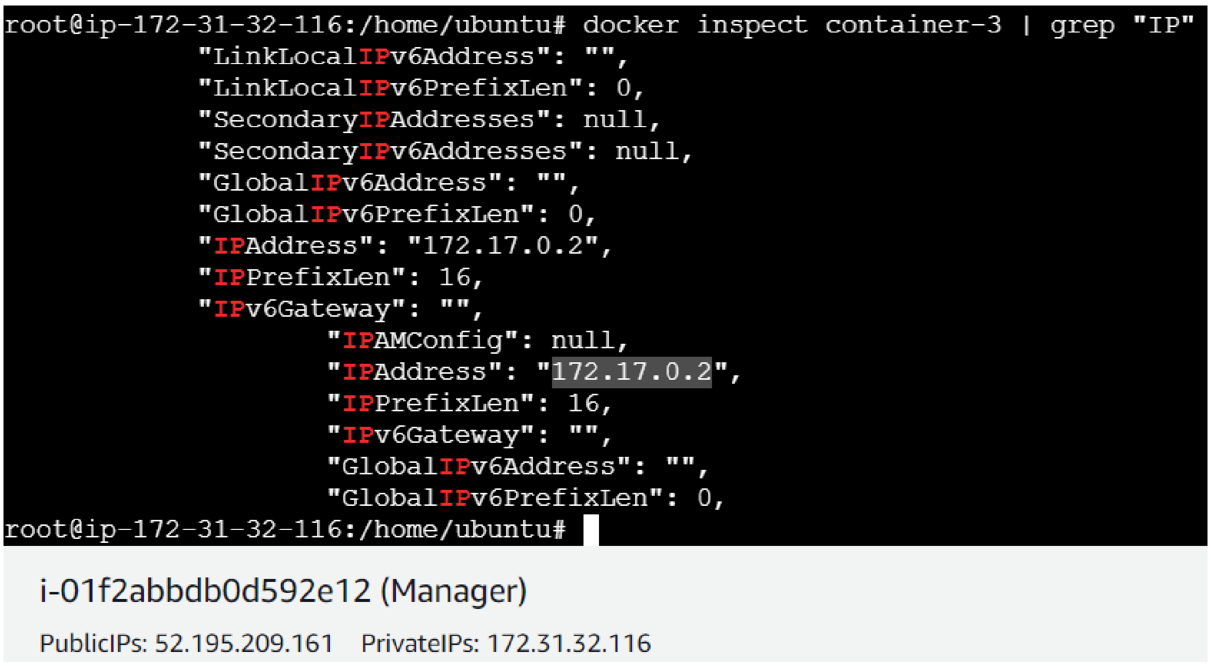
```
            "MacAddress": "02:42:ac:11:00:02",
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID": "40804148129ba97f3d94e13d30084675c60cb90916a6d463fb9ea1c2cc74c6b3",
                    "EndpointID": "32832cc36773edd77f59e64f8e3b20498212a1dbce5ad14a94559868d4de6555",
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:02",
                    "DriverOpts": null
                }
            }
        }
    }
]
root@ip-172-31-32-116:/home/ubuntu#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

7. Get the container 3's IP address using the command below and copy it so that you can use it later to ping from inside container 2.

docker inspect container-3 | grep "IP"



```
root@ip-172-31-32-116:/home/ubuntu# docker inspect container-3 | grep "IP"
            "LinkLocalIPv6Address": "",
            "LinkLocalIPv6PrefixLen": 0,
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
                    "IPAMConfig": null,
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
root@ip-172-31-32-116:/home/ubuntu#
```

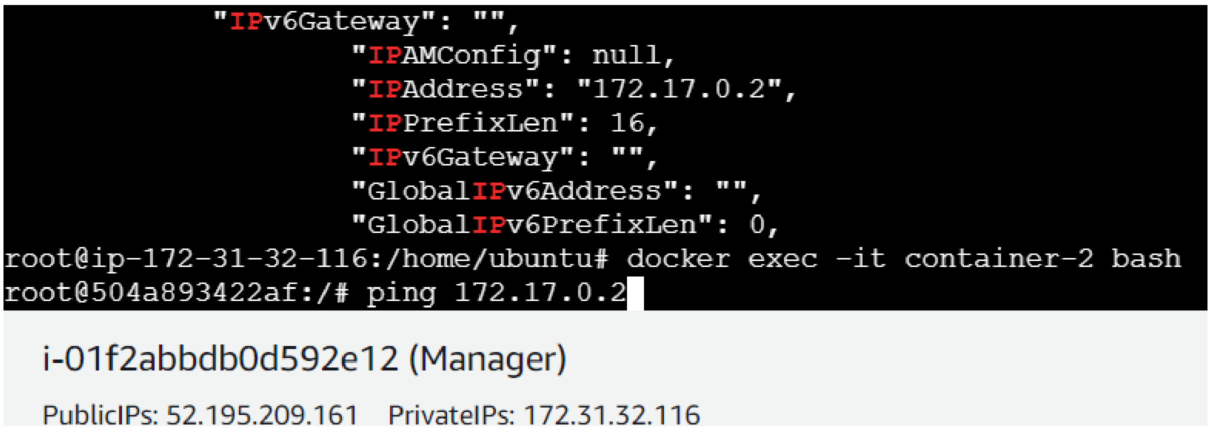
i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

8. Go inside container 2 again and use the ping command to ping the container 3.

docker exec -it container-2 bash

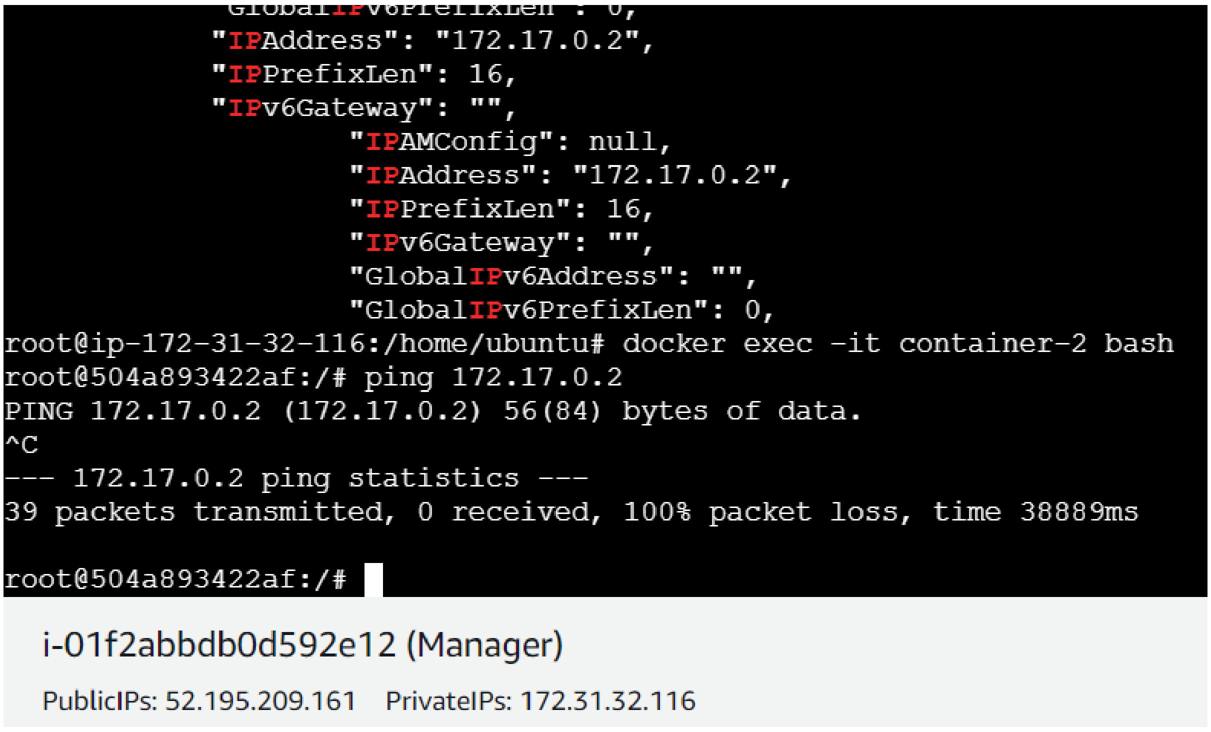
ping <container-3_ip>



```
            "IPv6Gateway": "",
                    "IPAMConfig": null,
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
root@ip-172-31-32-116:/home/ubuntu# docker exec -it container-2 bash
root@504a893422af:/# ping 172.17.0.2
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116



```
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
                    "IPAMConfig": null,
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
root@ip-172-31-32-116:/home/ubuntu# docker exec -it container-2 bash
root@504a893422af:/# 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 ---
39 packets transmitted, 0 received, 100% packet loss, time 38889ms

root@504a893422af:/#
```

i-01f2abbdb0d592e12 (Manager)

PublicIPs: 52.195.209.161   PrivateIPs: 172.31.32.116

As you can see here, Only those containers that are part of the bridge network are allowed to contact each other, any container from outside cannot do so.