Shree Rahul Education Society's (Regd.)
SHREE L. R. TIWARI COLLEGE OF ENGINEERING
Kanakia Park, Near Commissioner's Bungalow, Mira Road (East), Thane 401107, Maharashtra
(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)
NAAC Accredited | ISO 9001:2015 Certified
Tel. No.: 022-28120144 / 022-28120145 | Email: slrtce@rahuleducation.com | Website: www.slrtce.in

DEPARTMENT OF COMPUTER ENGINEERING

CSL502Computer Network Laboratory

Fifth Semester, 2020-2021 (Odd Semester)

Name of Student : Neha Mahendra Yadav
Roll No. : 67
Division : TE-CMPN
Batch : B3
Day / Session : Tuesday/Afternoon
Venue : Google Meet
Experiment No. : 05
Title of Experiment : To implement Multiple IP on single LAN.
Date of Conduction :
Date of Submission :

Particulars	Max. Marks	Marks Obtained
Preparedness and Efforts(PE)	3	
Knowledge of tools(KT)	3	
Debugging and results(DR)	3	
Documentation(DN)	3	
Punctuality & Lab Ethics(PL)	3	
Total	15	

Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)

Checked and Verified by

Name of Faculty : Prof. Rajesh Gaikwad
Signature :
Date : 4/08/2020

EXPERIMENT 5

Title: To implement Multiple IP on single LAN.

Objectives:

- In order to host multiple SSL sites as already mentioned
- Because you may be consolidating services from multiple hosts and you need to preserve the addresses
- In order to use an IP address that can later be transferred to another host · To compensate for a host that's down at that moment by adding its IP address to another one
- If you have multiple IP networks on the same physical/logical network/vlan it will prevent traffic from being exchanged via the gateway, speeding things up and reducing the load
- In order to setup a device that has a default IP address and thus you need to add an address on the same network
- In order to use different public IP addresses to avoid firewalls or to avoid being blacklisted in SPAM filters
- In order to make things less obvious to external people. E.g. you may be running apache on IP address 1.2.3.4 and only allow SSH on 1.2.3.5. That way if someone attempts to attack the IP address behind a site they won't find SSH running.
- In order to run the same service multiple times
- In order to use different hostnames in reverse DNS lookups. E.g. if you're connecting from this host to something external and you want to be presented as two different domain/hostnames
- In order not to expose commonality between services. E.g. if you host site1.example.com and site2.example.org and you map them on different IPs instead of using CNAMEs there won't be an obvious link between them

Pre Concepts:

1) IP Address: The IP address is the core component on which the networking architecture is built; no network exists without it. An IP address is a logical address that is used to uniquely identify every node in the network. Because IP addresses are logical, they can change. They are similar to addresses in a town or city because the IP address gives the network node an address so that it can communicate with other nodes or networks, just like mail is sent to friends and relatives.

2) LAN: A local area network (LAN) is a group of computers and associated devices that share a common communications line or wireless link to a server. Typically, a LAN encompasses computers and peripherals connected to a server within a distinct geographic area such as an

office or a commercial establishment. Computers and other mobile devices use a LAN connection to share resources such as a printer or network storage.

3) Linux Commands: There are many more commands than listed in this chapter. For information about other commands or more detailed information, we recommend the O'Reilly publication Linux in a Nutshell. In the following overview, the individual command elements are written in different typefaces. The actual command is always printed as **command**. Without this, nothing can function. Options without which the respective program cannot function are printed in italics. Further details, like file names, which must be passed to a command for correct functioning, are written in the Courier font. Specifications or parameters that are not required are placed in [brackets].

New Concepts:

1) The main advantage of using this IP aliasing is, you don't need to have a physical adapter attached to each IP, but instead you can create multiple or many virtual interfaces (aliases) to a single physical card.

2) The instructions given here are applies to all major Linux distributions like Red Hat, Fedora, and CentOS.

3) Creating multiple interfaces and assign IP address to it manually is a daunting task. Here we'll see how we can assign IP address to it defining a set of IP range.

4) Also understand how we are going to create a virtual interface and assign different range of IP Address to an interface in one go. In this article we used LAN IP's, so replace those with ones you will be using.

5) Linux is increasingly popular in the computer networking/telecommunications industry. Acquiring the Linux operating system is a relatively simple and inexpensive task since virtually all of the source code can be downloaded from several different FTP or HTTP sites on the Internet.

6) A *LAN* is a communications network that interconnects a variety of devices and provides a means for exchanging information among those devices.

7) A single broadcast packet sends out information that reaches every device connected to that network because each device has an entry point into the network. A large number of entry points, however, can negatively impact internetwork switching device performance, as well as your network's overall performance.

8) However, using a router to move traffic between subnets results in no broadcast traffic or any information that doesn't need to be routed being moved to other subnets.

Program Code:

Sudo ifconfig

Sudo ip addr add 192.168.1.104124 dev enpos3

Sudo ip address show enp0s3

To Remove:

Sudo ipd addrdel 192.168.1.104124 dev enpos3

Check:

Sudo ip address show enp0s3

Add Permanently:

Sudo cat/etc/network/interfaces

Now we will assign addition address

192.168.104124 Sudo nano/etc/network/interfaces

Source

Iface enpos3 inet static

Address 192.168.1.104124 (save and close)

Take effect without ebooting

Sudo ifdwon enpos3 & sudo ifup enp0s3

Check:

Sudo ip address show enp0s3

Outputs: Below the outputs for commands:

```
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.901 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.571 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.521 ms
64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=0.524 ms
```

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether 08:00:27:2a:03:4e brd ff:ff:ff:ff:ff:ff
inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3
valid_lft forever preferred_lft forever
inet 192.168.1.104/24 scope global secondary enp0s3
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe2a:34e/64 scope link
valid_lft forever preferred_lft forever
```

```
lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
 valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
 valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
 link/ether 08:00:27:2a:03:4b brd ff:ff:ff:ff:ff:ff
 inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3
 valid_lft forever preferred_lft forever
 inet6 fe80::a00:27ff:fe2a:34e/64 scope link
 valid_lft forever preferred_lft forever'
```

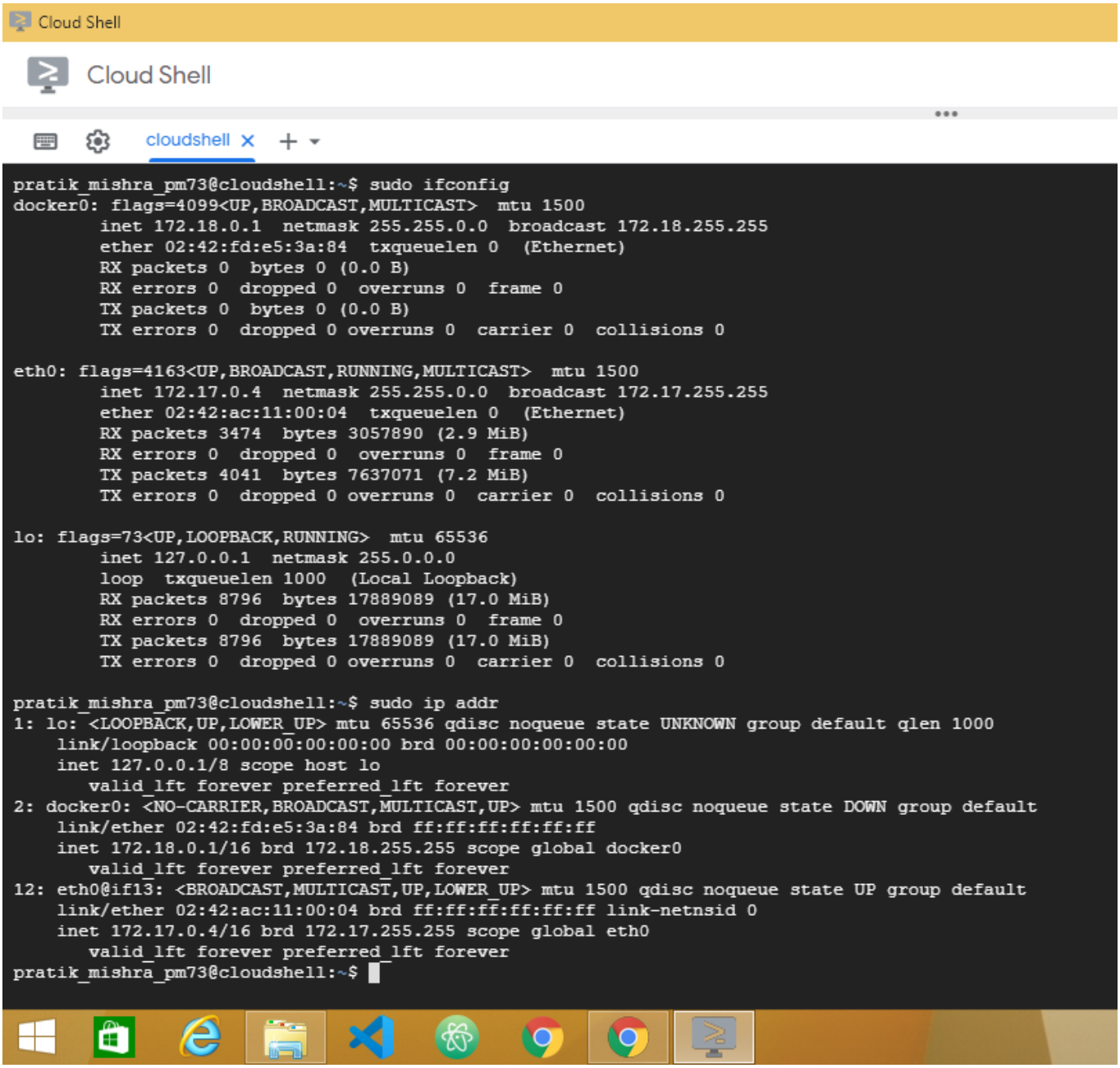
```
Listening on LPF/enp0s3/08:00:27:2a:03:4e
Sending on LPF/enp0s3/08:00:27:2a:03:4e
Sending on Socket/fallback
DHCPRELEASE on enp0s3 to 192.168.1.1 port 67 (xid=0x225f35)
Internet Systems Consortium DHCP Client 4.3.1
Copyright 2004-2014 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/enp0s3/08:00:27:2a:03:4e
Sending on LPF/enp0s3/08:00:27:2a:03:4e
Sending on Socket/fallback
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0xdfb94764)
DHCPREQUEST of 192.168.1.103 on enp0s3 to 255.255.255.255 port 67 (xid=0x6447b9df)
DHCPOFFER of 192.168.1.103 from 192.168.1.1
DHCPACK of 192.168.1.103 from 192.168.1.1
bound to 192.168.1.103 -- renewal in 35146 seconds.
```

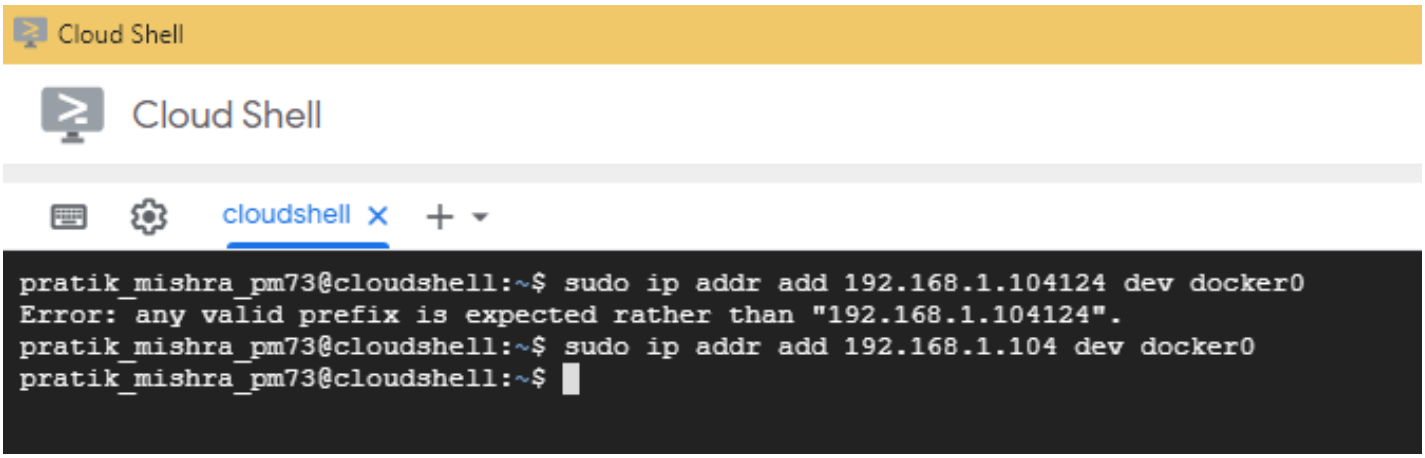
```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether 08:00:27:2a:03:4e brd ff:ff:ff:ff:ff:ff
inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe2a:34e/64 scope link
valid_lft forever preferred_lft forever
```

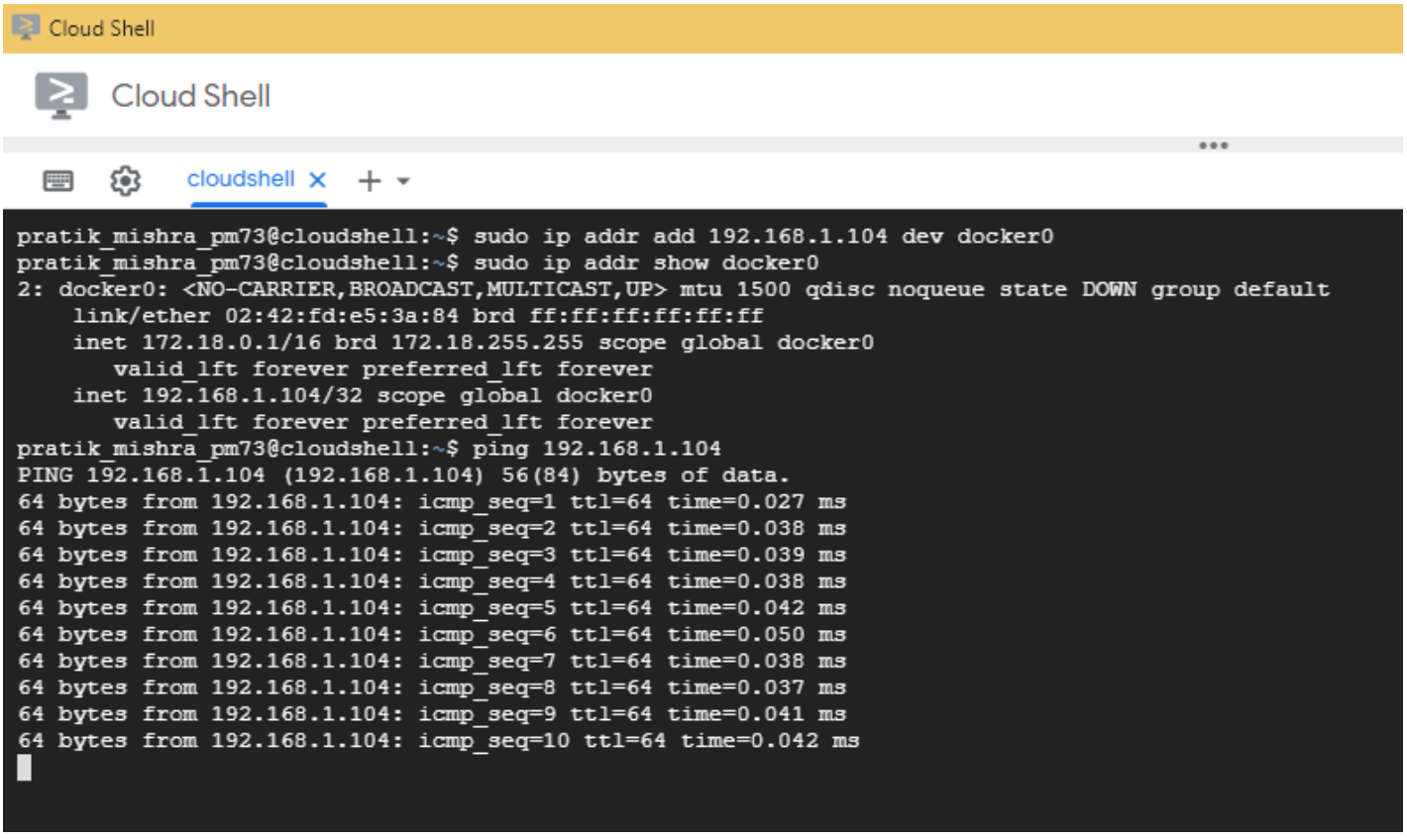
```
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.137 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=0.067 ms
```

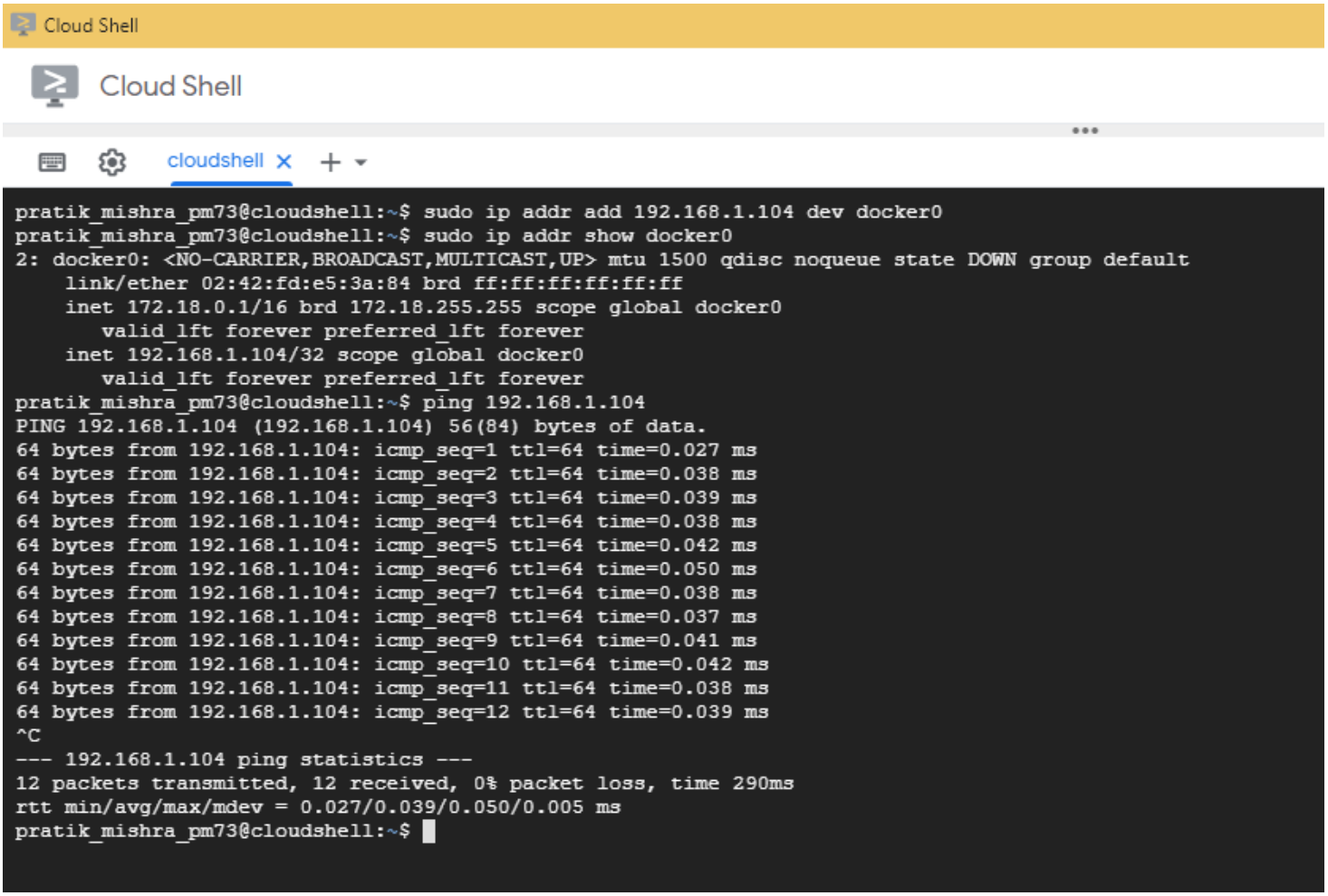
```
 enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether 08:00:27:2a:03:4e brd ff:ff:ff:ff:ff:ff
inet 192.168.1.103/24 brd 192.168.1.255 scope global enp0s3
valid_lft forever preferred_lft forever
inet 192.168.1.104/24 brd 192.168.1.255 scope global secondary enp0s3
valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe2a:34e/64 scope link
valid_lft forever preferred_lft forever
```

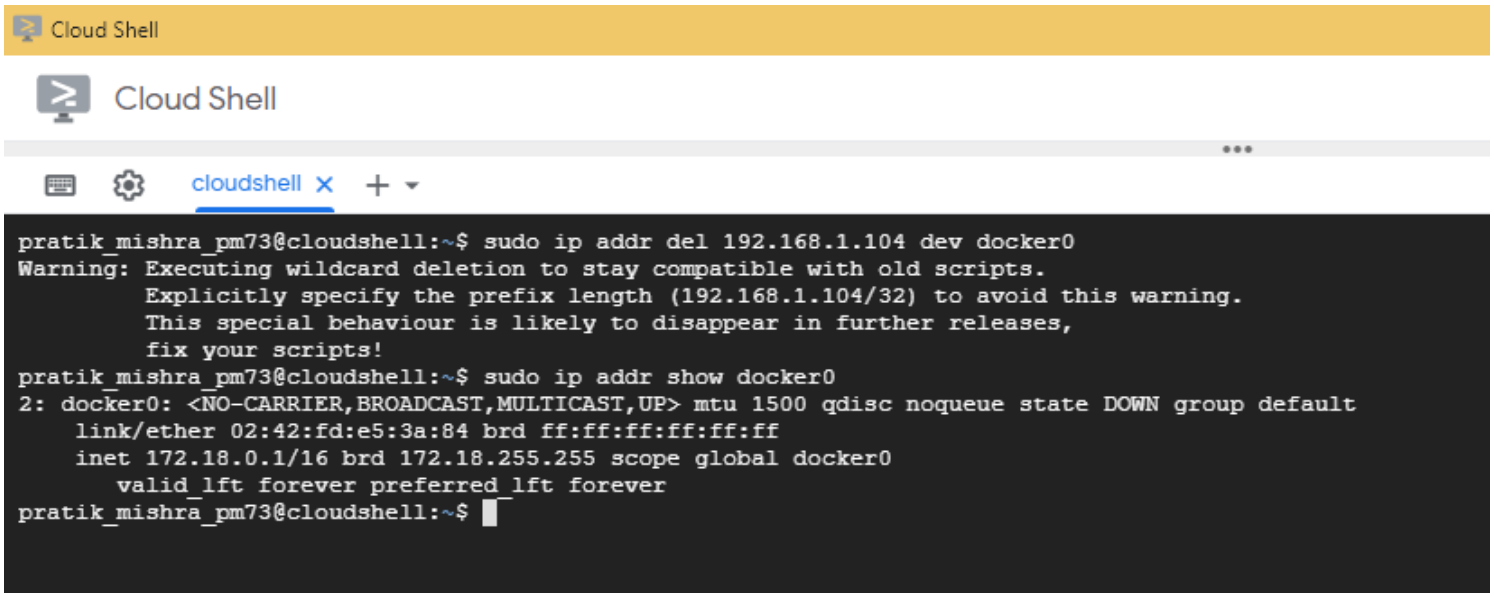
My Output:


Cloud Shell
Cloud Shell
cloudshell
pratik_mishra_pm73@cloudshell:~$ sudo ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
inet 172.18.0.1 netmask 255.255.0.0 broadcast 172.18.255.255
ether 02:42:fd:e5:3a:84 txqueuelen 0 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.17.0.4 netmask 255.255.0.0 broadcast 172.17.255.255
ether 02:42:ac:11:00:04 txqueuelen 0 (Ethernet)
RX packets 3474 bytes 3057890 (2.9 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4041 bytes 7637071 (7.2 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
loop txqueuelen 1000 (Local Loopback)
RX packets 8796 bytes 17889089 (17.0 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8796 bytes 17889089 (17.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pratik_mishra_pm73@cloudshell:~$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
link/ether 02:42:fd:e5:3a:84 brd ff:ff:ff:ff:ff:ff
inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
valid_lft forever preferred_lft forever
12: eth0@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
link/ether 02:42:ac:11:00:04 brd ff:ff:ff:ff:ff:ff link-netnsid 0
inet 172.17.0.4/16 brd 172.17.255.255 scope global eth0
valid_lft forever preferred_lft forever
pratik_mishra_pm73@cloudshell:~$


Cloud Shell
Cloud Shell
cloudshell
pratik_mishra_pm73@cloudshell:~$ sudo ip addr add 192.168.1.104124 dev docker0
Error: any valid prefix is expected rather than "192.168.1.104124".
pratik_mishra_pm73@cloudshell:~$ sudo ip addr add 192.168.1.104 dev docker0
pratik_mishra_pm73@cloudshell:~$


Cloud Shell
Cloud Shell
cloudshell
pratik_mishra_pm73@cloudshell:~$ sudo ip addr add 192.168.1.104 dev docker0
pratik_mishra_pm73@cloudshell:~$ sudo ip addr show docker0
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
link/ether 02:42:fd:e5:3a:84 brd ff:ff:ff:ff:ff:ff
inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
valid_lft forever preferred_lft forever
inet 192.168.1.104/32 scope global docker0
valid_lft forever preferred_lft forever
pratik_mishra_pm73@cloudshell:~$ ping 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=5 ttl=64 time=0.042 ms
64 bytes from 192.168.1.104: icmp_seq=6 ttl=64 time=0.050 ms
64 bytes from 192.168.1.104: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=8 ttl=64 time=0.037 ms
64 bytes from 192.168.1.104: icmp_seq=9 ttl=64 time=0.041 ms
64 bytes from 192.168.1.104: icmp_seq=10 ttl=64 time=0.042 ms


Cloud Shell
Cloud Shell
cloudshell
pratik_mishra_pm73@cloudshell:~$ sudo ip addr add 192.168.1.104 dev docker0
pratik_mishra_pm73@cloudshell:~$ sudo ip addr show docker0
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
link/ether 02:42:fd:e5:3a:84 brd ff:ff:ff:ff:ff:ff
inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
valid_lft forever preferred_lft forever
inet 192.168.1.104/32 scope global docker0
valid_lft forever preferred_lft forever
pratik_mishra_pm73@cloudshell:~$ ping 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=0.039 ms
64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=5 ttl=64 time=0.042 ms
64 bytes from 192.168.1.104: icmp_seq=6 ttl=64 time=0.050 ms
64 bytes from 192.168.1.104: icmp_seq=7 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=8 ttl=64 time=0.037 ms
64 bytes from 192.168.1.104: icmp_seq=9 ttl=64 time=0.041 ms
64 bytes from 192.168.1.104: icmp_seq=10 ttl=64 time=0.042 ms
64 bytes from 192.168.1.104: icmp_seq=11 ttl=64 time=0.038 ms
64 bytes from 192.168.1.104: icmp_seq=12 ttl=64 time=0.039 ms
^C
--- 192.168.1.104 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 290ms
rtt min/avg/max/mdev = 0.027/0.039/0.050/0.005 ms
pratik_mishra_pm73@cloudshell:~$


Cloud Shell
Cloud Shell
cloudshell
pratik_mishra_pm73@cloudshell:~$ sudo ip addr del 192.168.1.104 dev docker0
Warning: Executing wildcard deletion to stay compatible with old scripts.
Explicitly specify the prefix length (192.168.1.104/32) to avoid this warning.
This special behaviour is likely to disappear in further releases,
fix your scripts!
pratik_mishra_pm73@cloudshell:~$ sudo ip addr show docker0
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
link/ether 02:42:fd:e5:3a:84 brd ff:ff:ff:ff:ff:ff
inet 172.18.0.1/16 brd 172.18.255.255 scope global docker0
valid_lft forever preferred_lft forever
pratik_mishra_pm73@cloudshell:~$

Conclusion: Thus we have implemented Multiple IP Address into Single LAN.