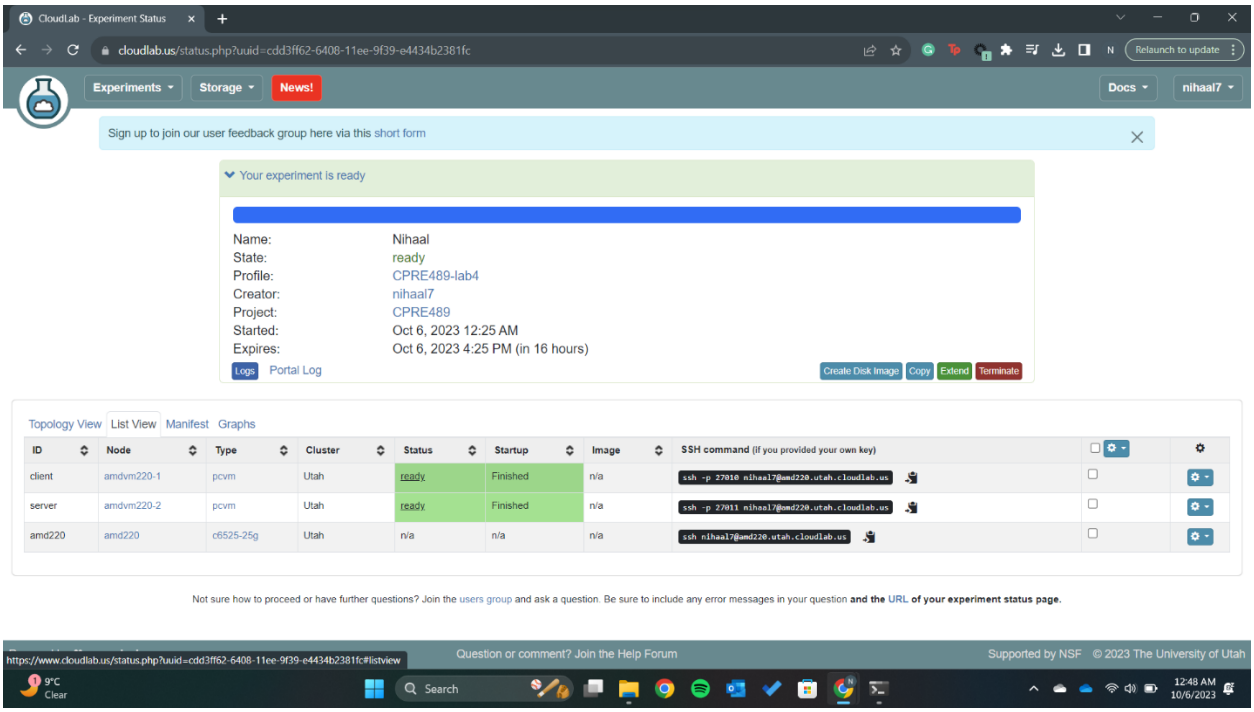
Lab4 – Cloudlab

Summary

In this lab, we learn about using Cloud lab, setting up, and running experiments. The lab teaches us to remotely log in to a cloud server and interact with the nodes. In exercise 3, we experiment with using multiple TCP connections parallelly and assessing their performance. We go on further to assess if each of the ports on the servers(eth0, eth1, and lo) are accessible by the client, as well as systems on the world wide web/internet.

Note: Throughout the lab, I ran different experiments since the old ones terminated, so the IP addresses are not fixed throughout my report.

Exercise 1



List view

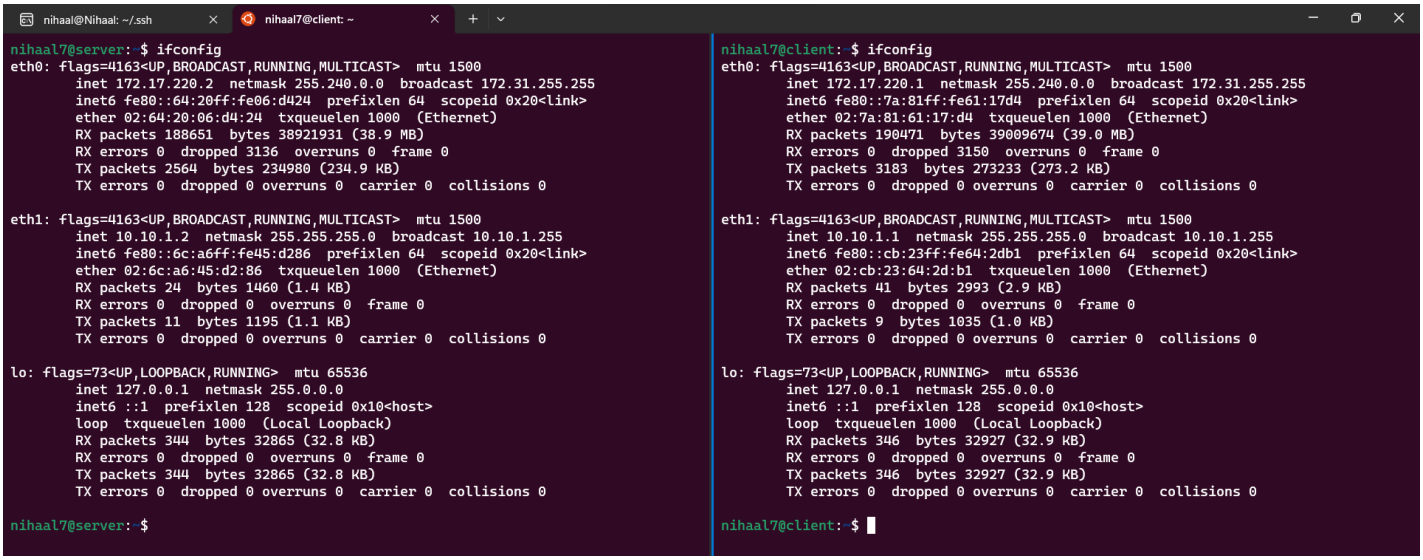
Exercise 2

```
[nzaheer@cpre587-f23-07 ~]$ ssh -i ~/.ssh/id_cloudlab_rsa nihaal7@amd220.utah.cloudlab.us -p 27011
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

nihaal7@server:~$
```

```
nihaal@Nihaal:~$ ssh nzaheer@cpre587-f23-07.ece.iastate.edu
nzaheer@cpre587-f23-07.ece.iastate.edu's password:
Activate the web console with: systemctl enable --now cockpit.socket

Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Fri Oct  6 00:53:01 2023 from 10.48.95.234
[nzaheer@cpre587-f23-07 ~]$ ssh -i ~/.ssh/id_cloudlab_rsa nihaal7@amd220.utah.cloudlab.us -p 27010
The authenticity of host '[amd220.utah.cloudlab.us]:27010 ([128.110.219.131]:27010)' can't be established.
ECDSA key fingerprint is SHA256:bB4TBZMIEhrtJN2mb/Mzn8fvSRUBUC+UamxqkhCU7HA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[amd220.utah.cloudlab.us]:27010,[128.110.219.131]:27010' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

nihaal7@client:~$
```

Client and Server Login page

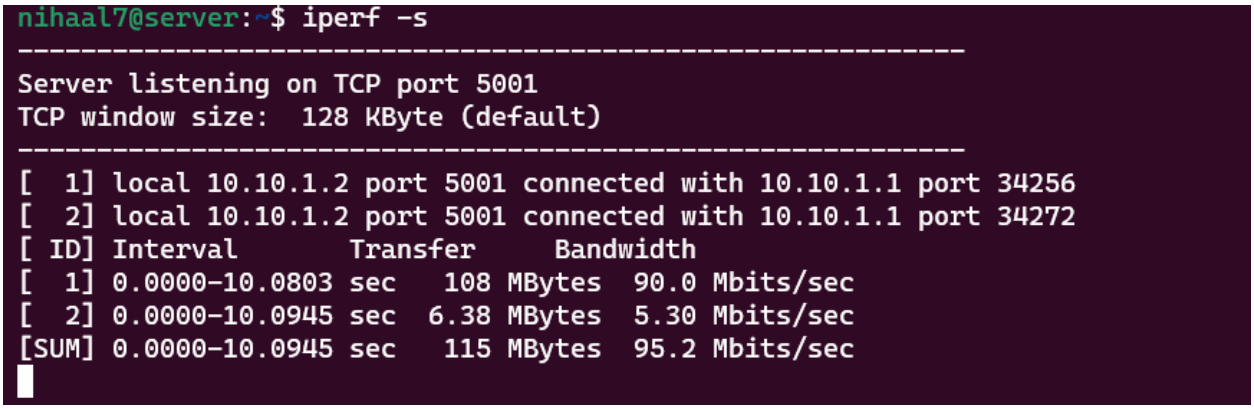
Exercise 3



Ifconfig

Server IP: 172.17.220.2

Client IP: 172.17.220.1



man page of iperf

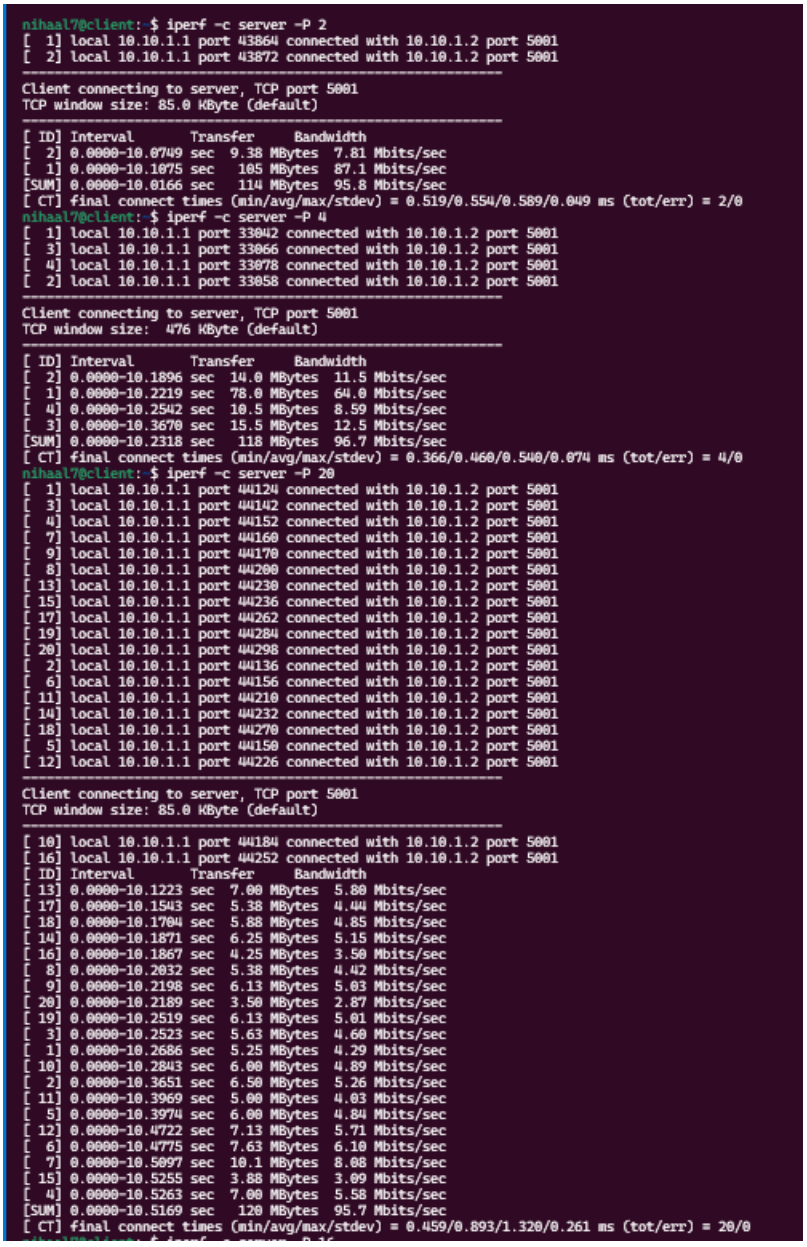
```
nihaal7@client:~$ iperf -c server -P 2
[  1] local 10.10.1.1 port 34256 connected with 10.10.1.2 port 5001
[  2] local 10.10.1.1 port 34272 connected with 10.10.1.2 port 5001
------------------------------------------------------------
Client connecting to server, TCP port 5001
TCP window size:  136 KByte (default)
------------------------------------------------------------
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0893 sec   108 MBytes  89.9 Mbits/sec
[  2] 0.0000-10.1059 sec  6.38 MBytes  5.29 Mbits/sec
[SUM] 0.0000-10.0843 sec   115 MBytes  95.2 Mbits/sec
[ CT] final connect times (min/avg/max/stdev) = 0.678/0.713/0.748/0.049 ms (tot/err
) = 2/0
```

-p 2

```
^Cnihaal7@server:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  1] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 37890
[  2] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 37898
[  3] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 37914
[  4] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 37928
[ ID] Interval       Transfer     Bandwidth
[  2] 0.0000-10.0981 sec  21.9 MBytes  18.2 Mbits/sec
[  4] 0.0000-10.1446 sec  27.0 MBytes  22.3 Mbits/sec
[  3] 0.0000-10.2677 sec  12.1 MBytes  9.91 Mbits/sec
[  1] 0.0000-10.3709 sec  57.1 MBytes  46.2 Mbits/sec
[SUM] 0.0000-10.3709 sec   118 MBytes  95.5 Mbits/sec
```

```
nihaal7@client:~$ iperf -c server -P 4
[  1] local 10.10.1.1 port 37890 connected with 10.10.1.2 port 5001
[  3] local 10.10.1.1 port 37914 connected with 10.10.1.2 port 5001
[  4] local 10.10.1.1 port 37928 connected with 10.10.1.2 port 5001
[  2] local 10.10.1.1 port 37898 connected with 10.10.1.2 port 5001
------------------------------------------------------------
Client connecting to server, TCP port 5001
TCP window size:  170 KByte (default)
------------------------------------------------------------
[ ID] Interval       Transfer     Bandwidth
[  2] 0.0000-10.1109 sec  21.9 MBytes  18.1 Mbits/sec
[  4] 0.0000-10.1593 sec  27.0 MBytes  22.3 Mbits/sec
[  3] 0.0000-10.2716 sec  12.1 MBytes  9.90 Mbits/sec
[  1] 0.0000-10.3849 sec  57.1 MBytes  46.1 Mbits/sec
[SUM] 0.0000-10.1598 sec   118 MBytes  97.5 Mbits/sec
[ CT] final connect times (min/avg/max/stdev) = 0.417/0.525/0.625/0.089 ms (tot/err
) = 4/0
```

-p 4



-p 20

This is what we get by changing the number of parallel connections.

We notice that for 2 parallel connections, the throughput is 95.8Mbps and the connection speed is between 7.81 Mbps and 87.1Mbps. Increasing the number of parallel connections initially improves throughput. This is seen when the number of parallel connections is made to 4. Here we see that the throughput is 96.7Mbps and the connection speed ranges from 11.5Mbps to 64Mbps. As we considerably increase the number of parallel connections, the returns diminish. As seen, when the number of parallel connections is 20, throughput decreases to 95.7Mbps, and individual connections range from 2.87Mbps to 8.08Mbps.

So we can infer that an initial increase in parallel connections improves performance, after which it plateaus and doesn't increase further.

5)

The 3 interfaces for the server machine

```
nihaal7@server:~$ ifconfig
eth0: flags=4163<UP,BROADCA
        inet 172.17.234.2
```

```
eth1: flags=4163<UP,BR
        inet 10.10.1.2
```

```
lo: flags=73<UP,LOOPBACK
        inet 127.0.0.1
```

For eth0:

I. *Interface with the client*

```
nihaal7@client:~$ ping 172.17.234.2
PING 172.17.234.2 (172.17.234.2) 56(84) bytes of data.
64 bytes from 172.17.234.2: icmp_seq=1 ttl=64 time=0.734 ms
64 bytes from 172.17.234.2: icmp_seq=2 ttl=64 time=0.277 ms
```

```
nihaal7@server:~$ sudo tcpdump -i eth0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol dec
ode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144
 bytes
10:29:33.351545 IP amdvm234-1.utah.cloudlab.us > amdvm234-2.utah.cloud
lab.us: ICMP echo request, id 4, seq 27, length 64
10:29:33.351561 IP amdvm234-2.utah.cloudlab.us > amdvm234-1.utah.cloud
lab.us: ICMP echo reply, id 4, seq 27, length 64
```

II. *Interface with the World wide Web*

```
nihaal7@server:~$ ping google.com
PING google.com (142.250.105.102) 56(84) bytes of data.
64 bytes from yt-in-f102.1e100.net (142.250.105.102): icmp_seq=1 ttl=5
0 time=63.8 ms
64 bytes from yt-in-f102.1e100.net (142.250.105.102): icmp_seq=2 ttl=5
0 time=63.8 ms
```

```
nihaal@Nihaal:~$ ping 172.17.234.2
PING 172.17.234.2 (172.17.234.2) 56(84) bytes of data.
^C
--- 172.17.234.2 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4183ms
```

For the eth1:

I. *Interface with the client*

```
nihaal7@client:~$ ping 10.10.1.2
PING 10.10.1.2 (10.10.1.2) 56(84) bytes of data.
64 bytes from 10.10.1.2: icmp_seq=1 ttl=64 time=0.672 ms
64 bytes from 10.10.1.2: icmp_seq=2 ttl=64 time=0.224 ms
```

```
nihaal7@server:~$ sudo tcpdump -i eth1 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol dec
ode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144
 bytes
10:31:12.007445 IP client-link-0 > server-link-0: ICMP echo request, i
d 5, seq 24, length 64
10:31:12.007463 IP server-link-0 > client-link-0: ICMP echo reply, id
5, seq 24, length 64
```

II. *Interface with the World wide Web*

```
nihaal@Nihaal:~$ ping 10.10.1.2
PING 10.10.1.2 (10.10.1.2) 56(84) bytes of data.
^C
--- 10.10.1.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1025ms
```

For lo:

I. *Interface with the client*

```
nihaal7@client:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.010 ms
```

```
nihaal7@server:~$ sudo tcpdump -i lo icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

The client is not able to ping the server on port 'lo'.

II. Interface with the World wide Web

Since 127.0.0.1 is a generic local IP address, it will work on every computer. This does not mean that we are receiving the ping sent via the 'lo' port. This can be justified by running traceroute on the 'World wide Web PC', in this case, my personal computer.

```
nihaal@Nihaal:~$ traceroute 127.0.0.1
traceroute to 127.0.0.1 (127.0.0.1), 30 hops max, 60 byte packets
 1  localhost (127.0.0.1)  0.047 ms  0.005 ms  0.004 ms
```

Explanation for 'Interface with World wide Web'

In the past, for cloudlab, eth0 was given a public IP address, but after the update, all ports only have a local IP. When we ping google.com, this works, but when we try to ping the cloudlab server from any PC on the internet, this fails. The reason as mentioned before, is since the server does not have a public IP address.

Exercise 4