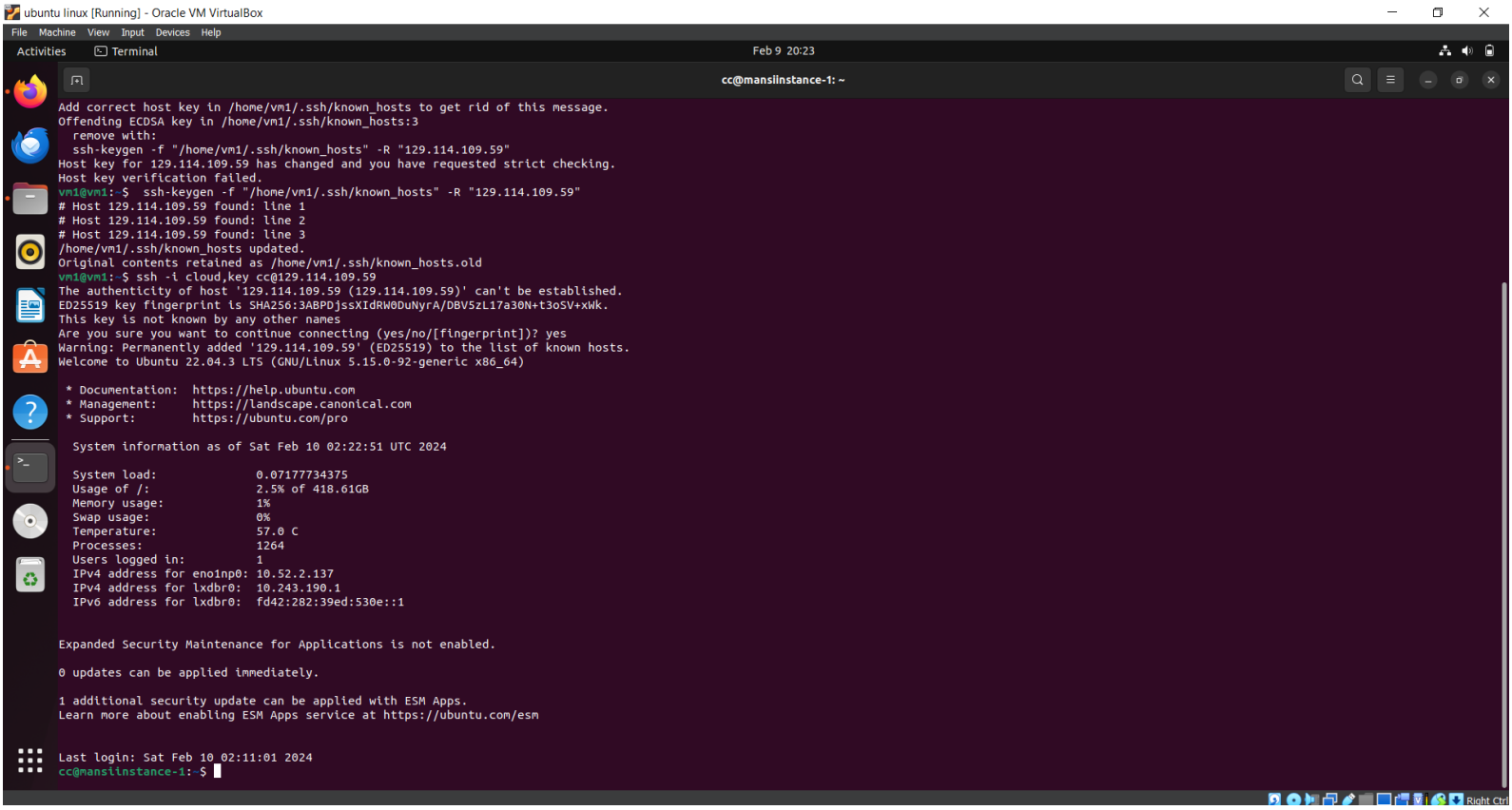
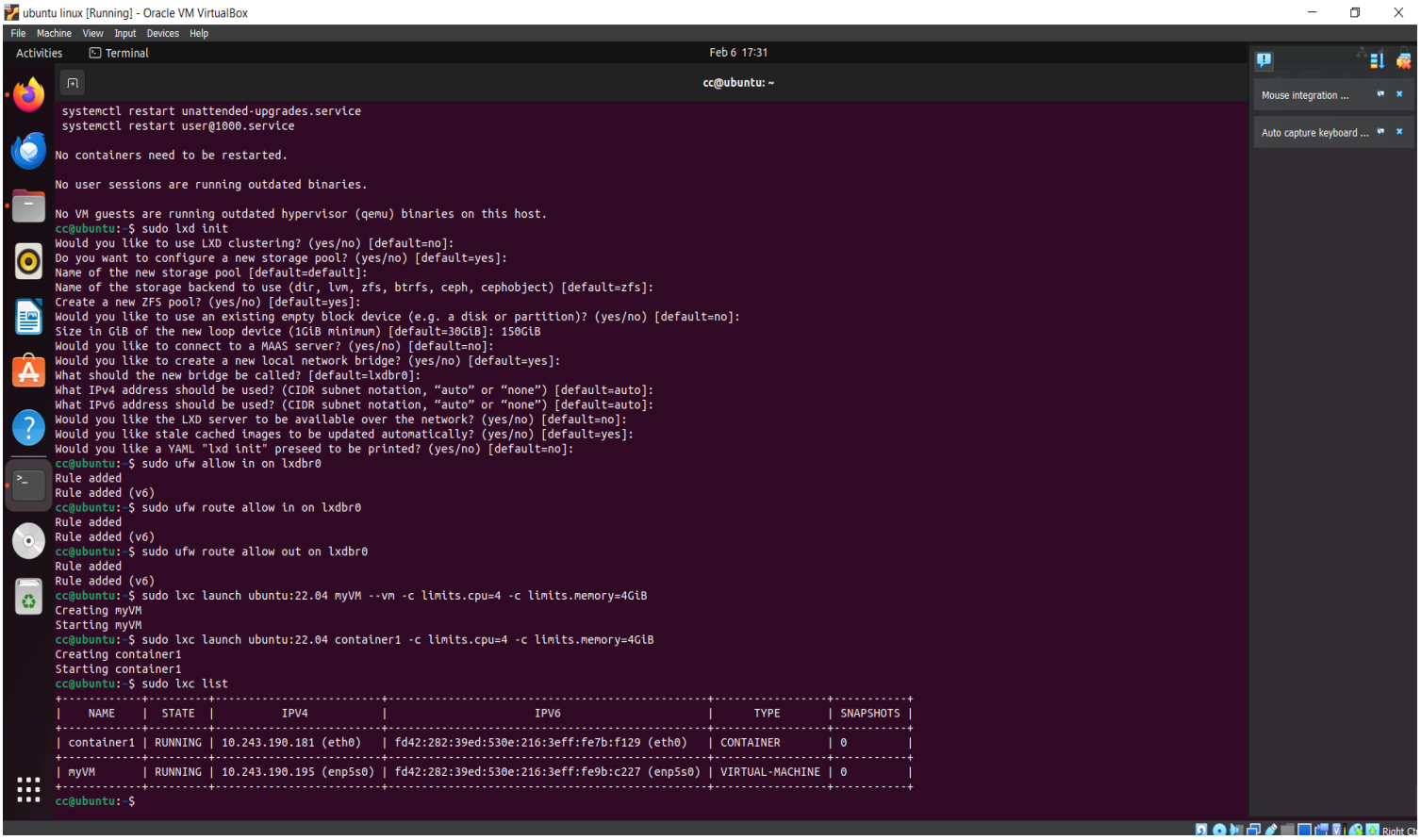
Assignment 2

Mansi Dinesh
A20556560

Cloud connection



Container and VM created

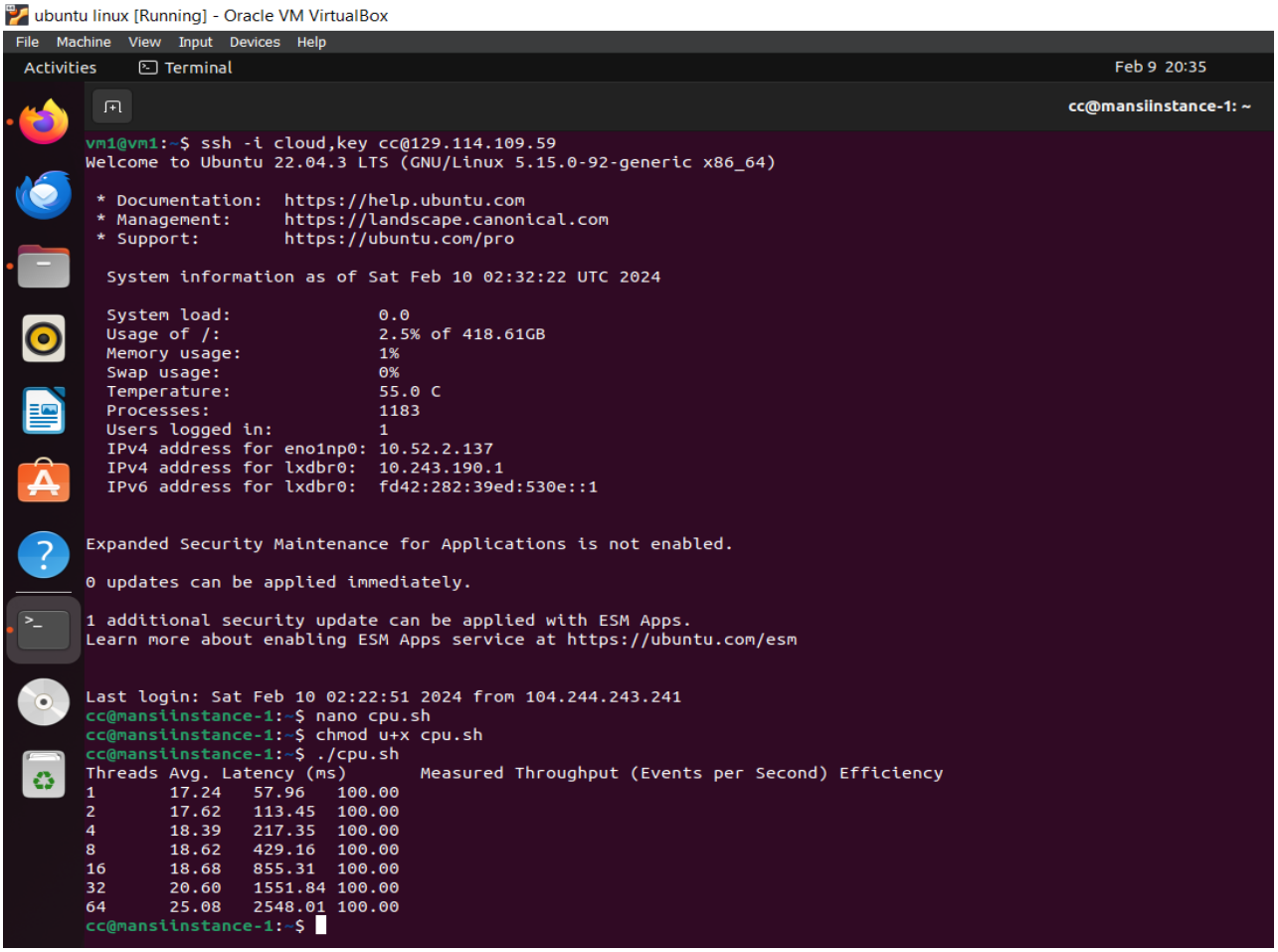


CPU:

- Strong scaling studies: Fixed prime numbers limit at 100,000. Then, measure the performance of each virtualization technologies when varying the number of threads.
- Sample command (you might need to use additional command line arguments):
  $ sysbench cpu --cpu-max-prime=100000 --threads=1 run
- Fill in the below using benchmark results of each scale regarding the processor performance:
  Note that the efficiency denotes a relative performance of a virtualization type vs. baremetal. EX:
  - Baremetal: 10 events per second
  - Container: 9 events per second
  - VM: 8 events per second

  This translates to the efficiency of:
  - Baremetal: 100%
  - Container: 90% (Container is 10% slower than Baremetal)
  - VM: 80% (VM is 20% slower than Baremetal)

Virtualization Type	Threads	Avg. Latency (ms)	Measured Throughput (Events per Second)	Efficiency
Baremetal	1	17.24	57.96	100
Container	1	28.26	35.37	99.47
Virtual Machine	1	28.37	35.23	99.07
Baremetal	2	17.62	113.45	100
Container	2	28.33	70.55	99.28
Virtual Machine	2	28.37	70.46	99.04
Baremetal	4	18.39	217.35	100
Container	4	30.38	131.19	92.33
Virtual Machine	4	28.30	141.28	99.34
Baremetal	8	18.62	429.16	100
Container	8	56.50	140.74	49.51
Virtual Machine	8	72.14	141.36	49.93
Baremetal	16	18.68	855.31	100
Container	16	112.09	141.44	25.03
Virtual Machine	16	112.29	141.84	25.13
Baremetal	32	20.60	1551.84	100
Container	32	222.57	141.43	14.14
Virtual Machine	32	223.62	141.77	14.20
Baremetal	64	25.08	2548.01	100
Container	64	439.57	141.66	10.76
Virtual Machine	64	442.07	142.09	10.70




```
root@container1:~# nano networkcontainer.sh
root@container1:~# ./cputextcontainer.sh
Threads Avg. Latency (ms)       Measured Throughput (Events per Second)
1       28.26   35.37
2       28.33   70.55
4       30.38   131.19
8       56.50   140.74
16      112.09  141.44
32      222.57  141.43
64      439.57  141.66
root@container1:~#
```

```
root@myVM:~# nano cputestvm.sh
root@myVM:~# ./cputestvm.sh
Threads Avg. Latency (ms)       Measured Throughput (Events per Second)
1       28.37   35.23
2       28.37   70.46
4       28.30   141.28
16      112.29  141.84
32      223.62  141.77
64      442.07  142.09
root@myVM:~#
```

Efficiency = (vm or container measured Throughput/baremetal measured Throughput)*100

Memory:

- Strong scaling studies: Fixed total data size in memory at 120GB. Then, measure the performance of each virtualization technologies with the following specifications:
  a. Block size: 1KB i.e., 2^{10} to 2^{20} bytes
  b. Operations: Read
  c. Access pattern: Random
- Sample command:
  $ sysbench memory --memory-block-size=1K --memory-total-size=120G --threads=1 run
- Fill in the below using benchmark results of each scale/type regarding the memory performance: Similar to efficiency example in CPU benchmark, the efficiency denotes a relative performance of a virtualization type vs. baremetal.

Virtualization Type	Threads	Block Size (KB)	Operation	Access Pattern	Total Operations	Throughput (MiB/sec)	Efficiency
Baremetal	1	1	Read	Random	73048810	7130.19	100
Container	1	1	Read	Random	41161665	4017.73	
Virtual Machine	1	1	Read	Random	42375020	4136.19	
Baremetal	2	1	Read	Random	50368221	4916.36	100
Container	2	1	Read	Random	59958945	5854.04	
Virtual Machine	2	1	Read	Random	56073237	5473.29	
Baremetal	4	1	Read	Random	57983849	5659.73	100
Container	4	1	Read	Random	33080994	3229.84	
Virtual Machine	4	1	Read	Random	69210308	6757.30	
Baremetal	8	1	Read	Random	83132903	8114.48	100
Container	8	1	Read	Random	32991963	3221.17	
Virtual Machine	8	1	Read	Random	69360870	6772.17	
Baremetal	16	1	Read	Random	93228864	9099.94	100
Container	16	1	Read	Random	32520912	3175.17	
Virtual Machine	16	1	Read	Random	69620981	6797.54	
Baremetal	32	1	Read	Random	79167943	7727.39	100
Container	32	1	Read	Random	32018301	3185.09	
Virtual Machine	32	1	Read	Random	71327932	6964.11	
Baremetal	64	1	Read	Random	97536119	9523.24	100
Container	64	1	Read	Random	33146373	3236.09	
Virtual Machine	64	1	Read	Random	70046274	6838.87	

```
cc@mansiinstance-1:~$ nano networkbare.sh
cc@mansiinstance-1:~$ chmod u+x networkbare.sh
cc@mansiinstance-1:~$ ./networkbare.sh
Threads Total Operations        Throughput (MiB/sec)
1       73048810        7130.19
2       50368221        4916.36
4       57983849        5659.73
8       83132903        8114.48
16      93228864        9099.94
32      79167943        7727.39
64      97536119        9523.24
cc@mansiinstance-1:~$
```

```
root@myVM:~# nano networkvm.sh
root@myVM:~# ./networkvm.sh
-bash: ./networkvm.sh: Permission denied
root@myVM:~# chmod u+x networkvm.sh
root@myVM:~# ./networkvm.sh
Threads Total Operations        Throughput (MiB/sec)
1       42375020        4136.19
2       56073237        5473.29
4       69210308        6757.30
8       69360870        6772.17
16      69620981        6797.54
32      71327932        6964.11
64      70046274        6838.87
root@myVM:~# exit
logout
```

```
root@container1:~# nano networkcontainer.sh
root@container1:~# ./networkcontainer.sh
-bash: ./networkcontainer.sh: Permission denied
root@container1:~# chmod u+x networkcontainer.sh
root@container1:~# ./networkcontainer.sh
Threads Total Operations        Throughput (MiB/sec)
1       41161665        4017.73
2       59958945        5854.04
4       33080994        3229.84
8       32991963        3221.17
16      32520912        3175.17
32      32623170        3185.09
64      33146373        3236.09
root@container1:~# exit
logout
```

Disk:

- Strong scaling studies: Fixed total data size on disk at 120GB. Then, measure the performance of

each virtualization technologies with the following specifications:

a. Number of files: 128
b. File block size: 4,096 bytes
c. Total file size: 120GB
d. Test mode: Random Read
e. IO Mode: Synchronous
f. Extra IO flag: DirectIO

- Sample commands:

  $ sysbench fileio --file-num=128 --file-block-size=4096 --file-total-size=120G --file-test-mode=rndrd --file-io-mode=sync --file-extra-flags=direct --threads=1 <prepare/run/cleanup>
- Fill in the below using benchmark results of each scale/type regarding the I/O performance:
- Similar to efficiency example in CPU benchmark, the efficiency denotes a relative performance of a virtualization type vs. baremetal.

Virtualization Type	Threads	Block Size (KB)	Operation	Access Pattern	I/O Mode	I/O Flag	Total Operations	Measured Throughput (MiB/s)	Efficiency
Baremetal	1	4	Read	Random	SYNC	DirectIO			
Container	1	4	Read	Random	SYNC	DirectIO	381570		
Virtual Machine	1	4	Read	Random	SYNC	DirectIO			
Baremetal	2	4	Read	Random	SYNC	DirectIO			
Container	2	4	Read	Random	SYNC	DirectIO	737237		
Virtual Machine	2	4	Read	Random	SYNC	DirectIO			
Baremetal	4	4	Read	Random	SYNC	DirectIO			
Container	4	4	Read	Random	SYNC	DirectIO	1315837		
Virtual Machine	4	4	Read	Random	SYNC	DirectIO			
Baremetal	8	4	Read	Random	SYNC	DirectIO			
Container	8	4	Read	Random	SYNC	DirectIO	1306997		
Virtual Machine	8	4	Read	Random	SYNC	DirectIO			
Baremetal	16	4	Read	Random	SYNC	DirectIO			
Container	16	4	Read	Random	SYNC	DirectIO	1300893		
Virtual Machine	16	4	Read	Random	SYNC	DirectIO			
Baremetal	32	4	Read	Random	SYNC	DirectIO			
Container	32	4	Read	Random	SYNC	DirectIO	1295967		
Virtual Machine	32	4	Read	Random	SYNC	DirectIO			
Baremetal	64	4	Read	Random	SYNC	DirectIO			
Container	64	4	Read	Random	SYNC	DirectIO	1304732		
Virtual Machine	64	4	Read	Random	SYNC	DirectIO			

```
Threads: 1  | Total Operations: 381570  | Measured Throughput:
Threads: 2  | Total Operations: 737237  | Measured Throughput:
Threads: 4  | Total Operations: 1315837 | Measured Throughput:
Threads: 8  | Total Operations: 1306997 | Measured Throughput:
Threads: 16 | Total Operations: 1300893 | Measured Throughput:
Threads: 32 | Total Operations: 1295967 | Measured Throughput:
Threads: 64 | Total Operations: 1304732 | Measured Throughput:
```

Network:

- Strong scaling studies using one server vs. *N* number of clients. Measure the performance of each virtualization technologies with the following specifications:
  a. Server TCP window size: 1MB
  b. Client TCP write buffer size: 8,192KB
  c. Client TCP window size: 2.5MB
  d. Naggle algorithm: Off
- The configuration of client/server should communicate using TCP over local loopback.
- Sample commands:
  $ iperf -s -w 1M
  $ iperf -c 127.0.0.1 -e -i 1 --nodelay -l 8192K --trip-times --parallel 1
- Fill in the below using benchmark results of each scale/type regarding the I/O performance:
- Similar to efficiency example in CPU benchmark, the efficiency denotes a relative performance of a virtualization type vs. baremetal.

Virtualization Type	Server	Client Threads	Latency (ms)	Measured Throughput (Gbits/s)	Efficiency
Baremetal	1	1	1.920	44.6	
Container	1	1	2.046	42.6	
Virtual Machine	1	1	2.147	41.4	
Baremetal	1	2	2.120	41.1	
Container	1	2	2.023	40.6	
Virtual Machine	1	2	2.033	42.6	
Baremetal	1	4	2.133	40.6	
Container	1	4	3.247	23.4	
Virtual Machine	1	4	1.805	43.2	
Baremetal	1	8	2.565	32.6	
Container	1	8	4.549	14.7	
Virtual Machine	1	8	3.652	19.6	
Baremetal	1	16	3.156	26.7	
Container	1	16	10.286	6.51	
Virtual Machine	1	16	7.763	9.03	
Baremetal	1	32	5.471	15.2	
Container	1	32	17.695	4.52	
Virtual Machine	1	32	15.704	4.23	
Baremetal	1	64	6.521	13.5	
Container	1	64	22.653	3.01	
Virtual Machine	1	64	37.021	1.85	

System used

Chameleon Instance: compute_haswell_ib at CHI@NACC
o CPU: 2x Intel® Xeon® E5-2670 v3 @2.30GHz
o Memory: 8x 16GB (128GB) of DDR4-2,133 ECC Registered RAM
o Disk: 1x Seagate ST9250610NS SATA 7,200 RPM HDD
o Network: Broadcom NetXtreme II BCM57800 1/10 Gigabit Ethernet