

CSEN 241: Cloud Computing HW 1

Title: System VS OS Virtualization



By:

Name	Student ID	Email ID
Kartiki Rajendra Dindorkar	1651519	kdindorkar@scu.edu

Table of Contents

I.	System Configurations.....	3
II.	GitHub Repository	3
III.	QEMU Installation and Ubuntu 20.04	3
IV.	Ubuntu 20.04 Configuration on QEMU.....	9
V.	Install SysBench on QEMU Ubuntu VM.....	10
VI.	SysBench Manual	11
VII.	Docker Installation	13
VIII.	Ubuntu 20.04 with SysBench on Docker.....	13
IX.	Experiments.....	14
A.	QEMU (System Virtualization)	14
1.	Image Format: raw	14
2.	Image Format: qcow2.....	18
B.	Docker (OS Virtualization)	21
X.	Automation.....	22
A.	QEMU	23
B.	Docker	28
XI.	Analysis.....	32
XII.	Vagrant	33
XIII.	Encrypted Image.....	36
XIV.	References.....	38

I. System Configurations

Operating System	MAC OS Sonoma 14.2.1
Chip	Apple M2
System Architecture	arm64
Memory	8 GB
Number of Cores	8 (4 performance and 4 efficiency)

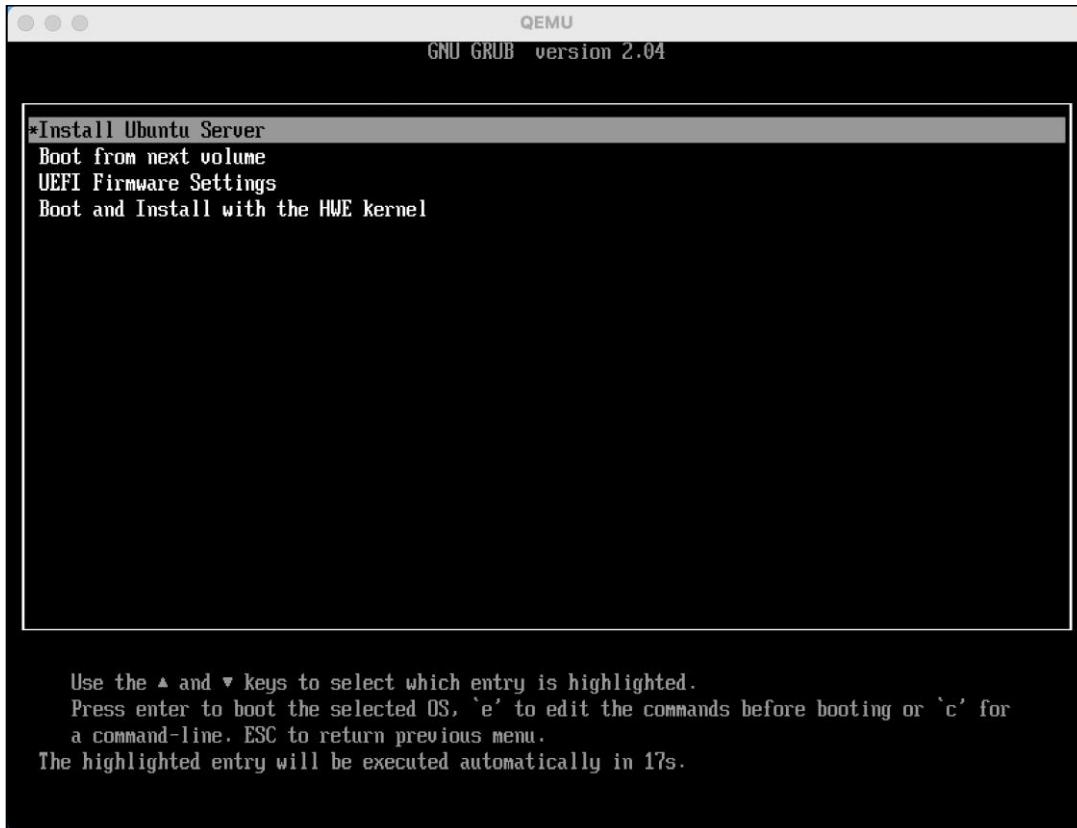
II. GitHub Repository

[GitHub - CloudComputing](#)

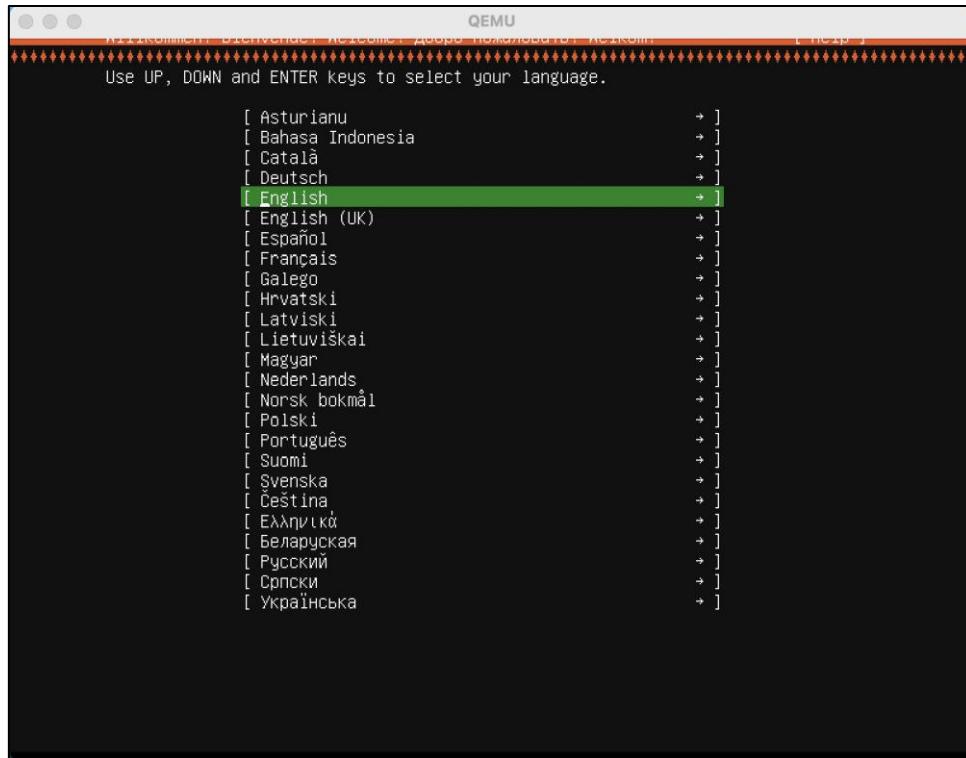
III. QEMU Installation and Ubuntu 20.04

Refer to [QEMU and Ubuntu 20.04 Setup](#) (QEMUUbuntuSetup.pdf attached in the HW1 folder) for Installing QEMU on Apple MAC M2 chip and setting Ubuntu 20.04 OS. Following are the steps for Ubuntu Setup:

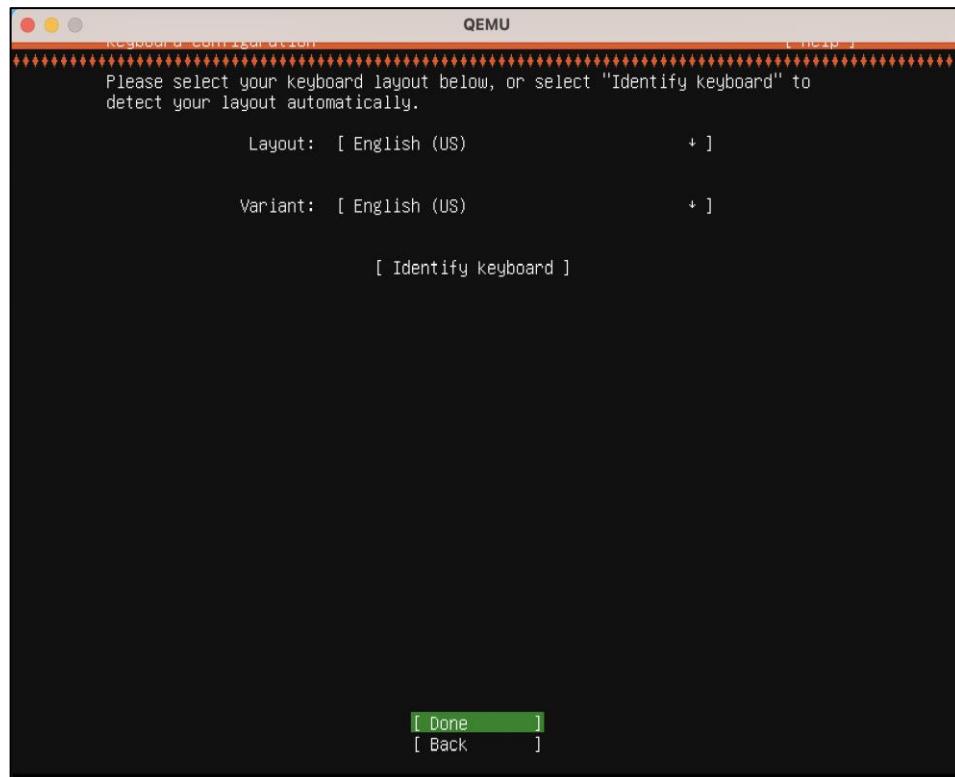
1. Click on Install Ubuntu Server



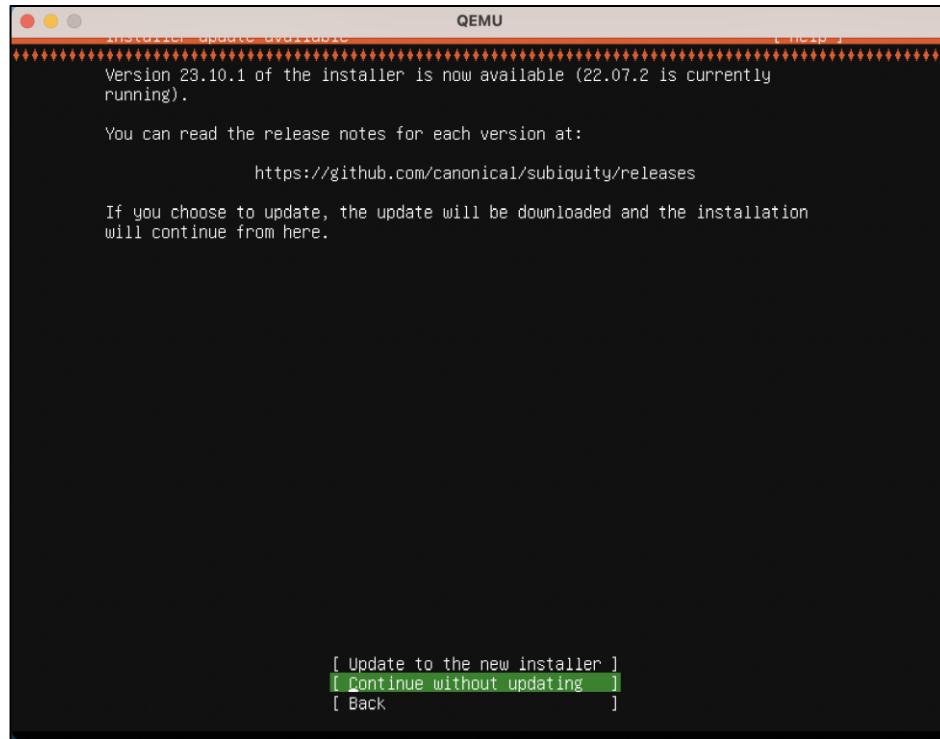
2. Select the language



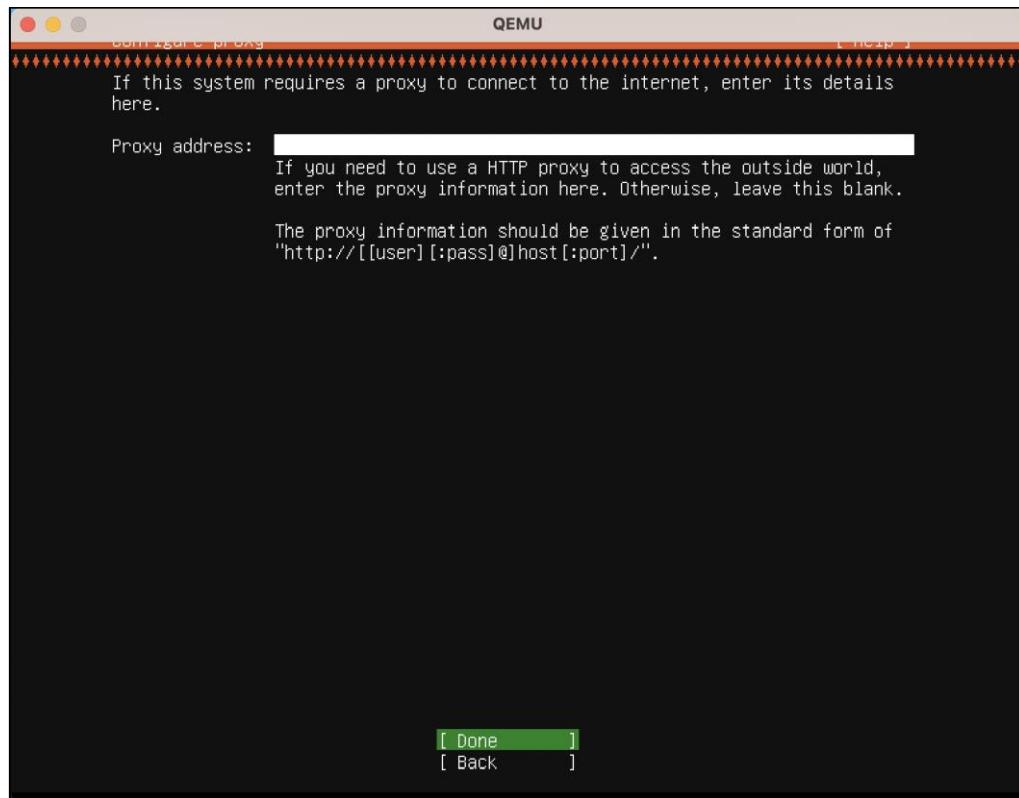
3. Select Keyboard Layout



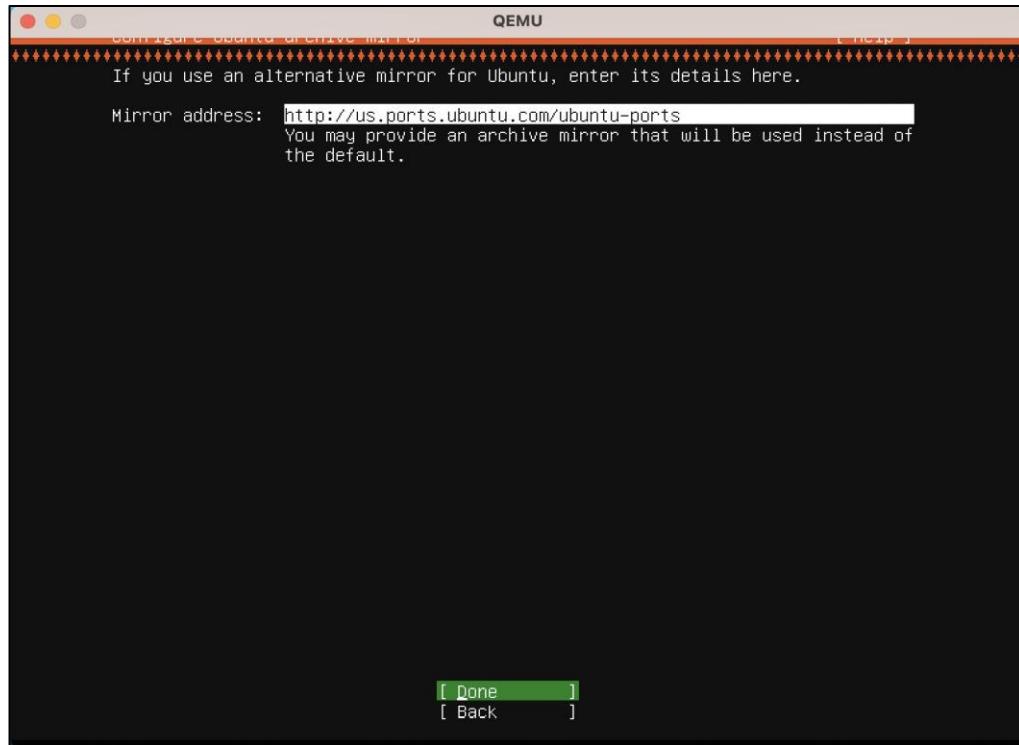
4. Select the Update option



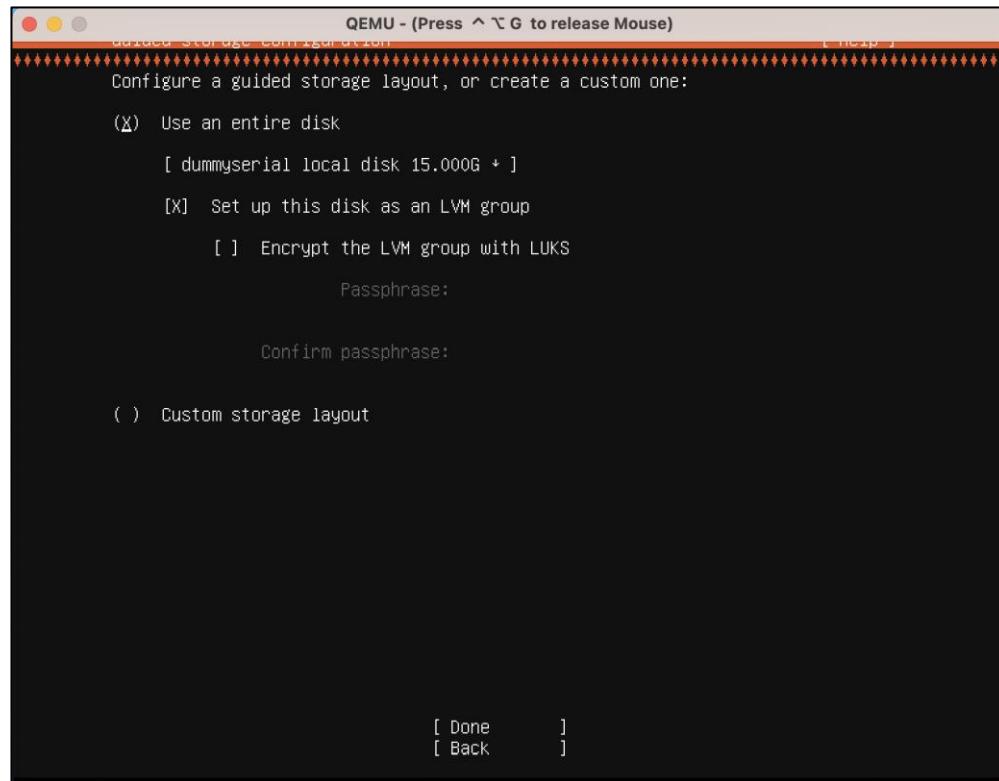
5. Internet Connection



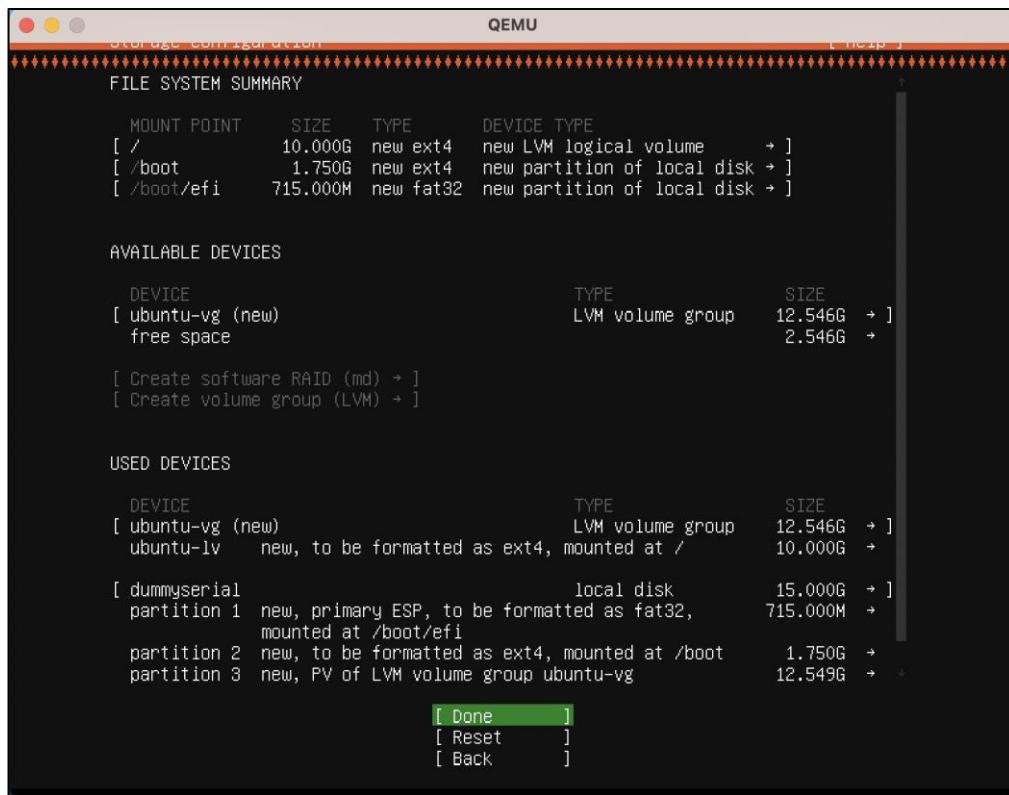
6. Mirror address



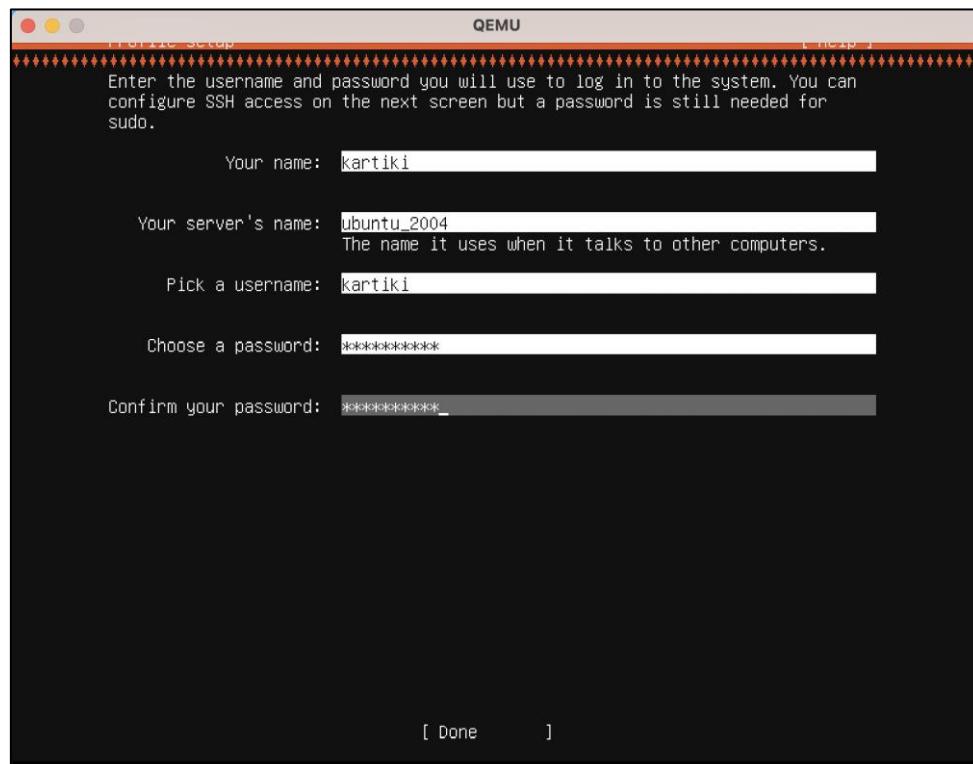
7. Configure Storage



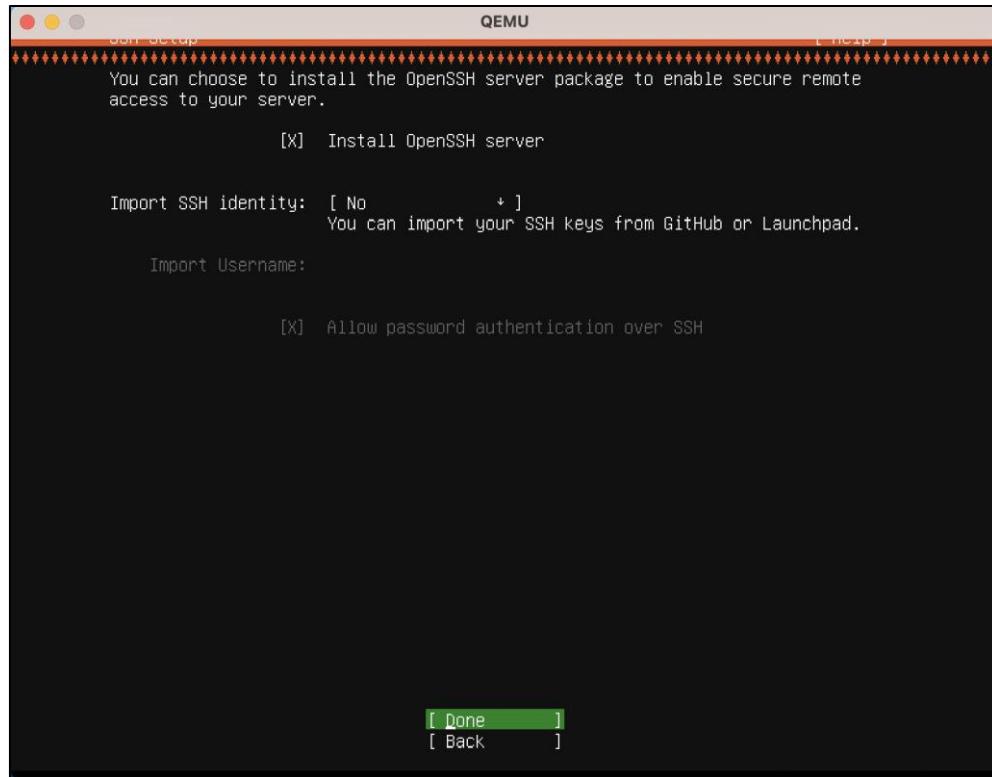
8. File System Summary



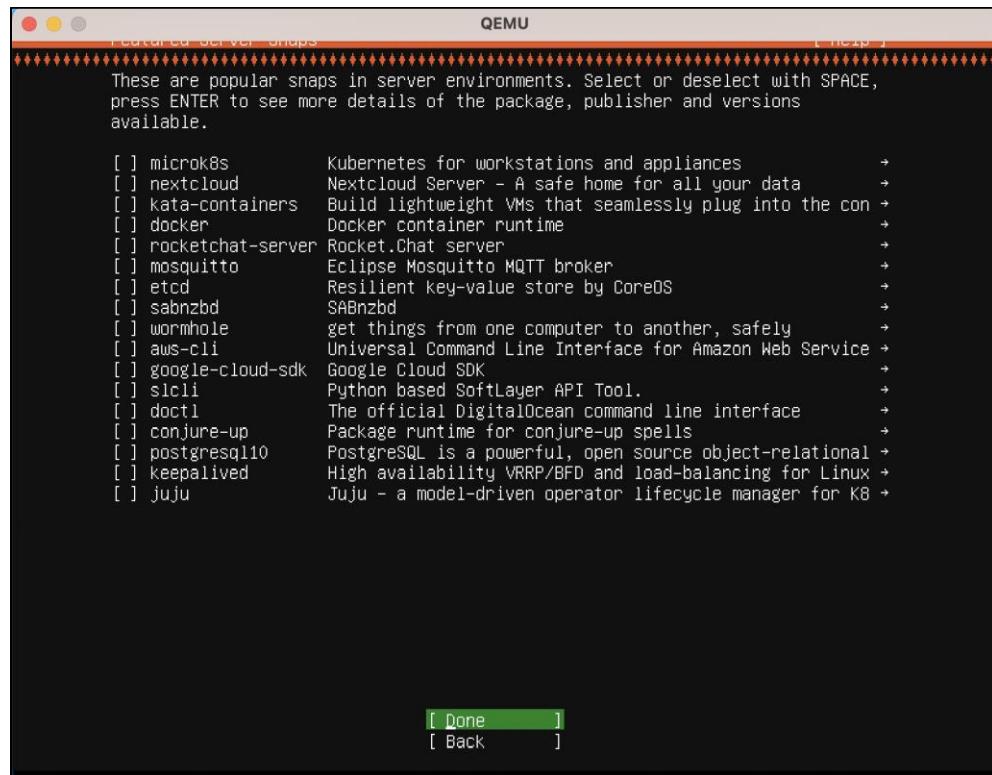
9. Username and Password Configuration



10. OpenSSH Package Installation



11. Package Installation



IV. Ubuntu 20.04 Configuration on QEMU

After [creating raw image](#) to [install it using installation command](#), let us look what all the used flags means:

```
# Simulate 64-bit ARM machine with created QEMU image and Ubuntu 20.04 ISO file
sudo qemu-system-aarch64 \
--nodefaults \
--accel hvf \
--cpu cortex-a57 \
-M virt,highmem=off \
-m 2048 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=none,file=ubuntulatest.img,format=raw,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-device usb-ehci -device usb-kbd -device usb-mouse \
-usb
```

qemu-system-aarch64	QEMU package to simulate 64-bit ARM machine.
-nodefaults	Don't create default devices like serial port, parallel port, VGA adapter, etc.
-accel	Enable an accelerator depending on target device. (kvm, xen, hvf, nvmm, whpx or tcg).
-m	Guest startup RAM size, default is 128 M, here we are allocating 2 Gb RAM.
-smp	Number of CPU allocating to the VM, here we are allocating 2 CPUs.
-drive file=file	Defines a new drive and which disk image to use with this drive.
-device	Adds device driver, here we have added, keyboard, mouse, ehci, network device and block device.
-netdev	Configures user mode host network backend.
-vga	Selects type of VGA card.
-cdrom file	Use the file as CD-ROM image to boot the ISO file.
-usb	Enables USB emulation on machines with on-board USB host controller.

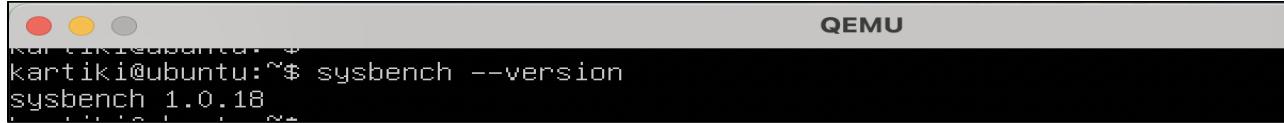
V. Install SysBench on QEMU Ubuntu VM

- Once you sign in with your username and password on Ubuntu VM, install “SysBench” using following commands:

```
$ sudo apt update  
$ sudo apt install sysbench
```

- Check the SysBench version using:

```
$ sysbench --version
```



The screenshot shows a terminal window titled "QEMU". The command "sysbench --version" is run, and the output "sysbench 1.0.18" is displayed.

```
kartiki@ubuntu:~$ sysbench --version  
sysbench 1.0.18
```

VI. SysBench Manual

1. The general format of SysBench is:

```
$ sysbench [common-options] --test=name [test-options] command
```

2. Use “cleanup” command to remove temporary data after the test run:

```
$ sysbench --test=fileio --file-total-size=3G cleanup
```

3. Multiple Common options are:

Option	Description	Default Value
--num-threads	Total no of threads to create	1
--max-requests	Total number of requests (0=Infinite)	10000
--max-time	Limit total execution time (sec)	0
--forced-shutdown	Amount of time to wait after --max-time to force shutdown. (sec / %)	off
--test	Name of the test mode to run	REQUIRED
--debug	Display more debug info	off
--validate	Validate the test results	off
--batch	Periodic dump of current status of run	off
--batch-delay	Delay between batch dump (sec)	300

4. Batch option can be utilized as:

```
$ sysbench -batch -batch-delay=5 --test(cpu) -cpu-max-prime=500 run
```

5. Test Modes:

- a. **CPU:** Using `-cpu-max-primes` option, we will request to calculate prime numbers up to specified value. Each thread executed concurrently until calculation is completed or `-max-time` is completed

```
$ sysbench --test(cpu) -cpu-max-prime=50000 run
```

- b. **Memory:** It is used to benchmark sequential memory reads or writes. Using `-memory-block-size` we can provide the size of memory block to use, its default value is 1 KiB.

```
$ sysbench --test(memory) -memory-block-size=1000K run
```

- c. **Fileio:** This test mode can help to produce multiple types of file I/O workloads
 - i. Prepare stage – SysBench creates specified number files with specified total size.
 - ii. run stage – each thread performs a specified operation.
 - iii. Supported I/O Operations:

seqwr	Sequential Write
-------	------------------

seqrewr	Sequential Rewrite
seqrd	Sequential Read
rndrd	Random Read
rndwr	Random Write
rndrw	Random Read + Random Write

iv. Test specific options:

Option	Description	Default Value
--file-num	No. Of files to create	128
--file-block-size	Block size to use in all I/O operations	16 K
--file-total-size	Total Size of files	2 G
--file-test-mode	Type of workload (supported I/O operations)	Required

```
$ sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw \
    prepare

$ sysbench --test=fileio --file-total-size=1G --file-test-mode=rndrw \
    run

$ sysbench --test=fileio --file-total-size=3G --file-test-mode=rndrw \
    cleanup
```

VII. Docker Installation

Refer [Docker Installation with Apple Mac M2](#) (DockerInstallation.pdf is attached in the folder) chip and setting Ubuntu 20.04 OS.

VIII. Ubuntu 20.04 with SysBench on Docker

1. Create a docker file as instructed in [create a Dockerfile](#)

```
CloudComputing > HW1 > Dockerfile
You, 5 minutes ago | 1 author (You)
1 # Use Ubuntu 20.04 as base image
2 FROM ubuntu:20.04
3
4 # Install sysbench packages
5 RUN apt-get update && \
6     apt-get install -y sysbench && \
7     apt-get clean && \
8     rm -rf /var/lib/apt/lists/*
9
10 # CMD instruction
11 CMD ["/bin/bash"]
12
```

2. Build the image

```
$ docker build -t <image_name> .
```

3. Run the container

```
$ docker run --rm <image_name>
```

4. Run the container with different CPU and Memory allocation

```
$ docker run --cpus=<cpu_count> --memory=<memory_limit> \
<image_name>
```

5. Check the running containers

```
$ docker ps
```

6. List all the docker images

```
$ docker images
```

7. Getting the history of your image

```
$ docker history <image_id>
```

IX. Experiments

A. QEMU (System Virtualization)

The experiment involves multiple test cases where there are variations in CPU and RAM allocated to the VM. Also, 3 test modes of SysBench will involve 2 variations each. Thus, the variables and their variations are listed below:

Sr No.	Variable Category	Variable Name	Parameter Name	Values
1	VM	Image Format		raw, qcow2
2	VM	Number of CPU	-smp	2, 4
3	VM	Memory Allocated	-m	2G, 3G
4	SysBench Test Mode	CPU	--cpu-max-prime	30000, 60000
		Memory	--memory-block-size	1G, 2G
		FileIO	--file-test-mode	Sequential Write (seqwr), Random Write (rndwr), Random Read-Write (rndrw)

1. Image Format: raw

Instructions to create raw image format and running the VM is given in [QEMU and Ubuntu 20.04 Setup](#) [QEMU and Ubuntu 20.04 Setup.](#)

Following are the SysBench test cases running on QEMU VM:

a. Raw image variations

As per the above table, for raw image format, created 4 variations to run SysBench Test Cases.

- i. Raw Image with CPU = 2, Memory = 2G
- ii. Raw Image with CPU = 2, Memory = 3G
- iii. Raw Image with CPU = 4, Memory = 2G
- iv. Raw Image with CPU = 4, Memory = 3G

b. Running SysBench CPU test case:

```
$ sysbench -test=cpu -cpu-max-prime=30000 run
```

```
sysbench 1.0.18
kartiki@ubuntu:~$ sysbench --test=cpu --cpu-max-prime=30000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuAJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 30000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 2625.67

General statistics:
total time: 10.0005s
total number of events: 26262

Latency (ms):
min: 0.37
avg: 0.38
max: 0.98
95th percentile: 0.40
sum: 9996.05

Threads fairness:
events (avg/stddev): 26262.0000/0.00
execution time (avg/stddev): 9.9960/0.00

kartiki@ubuntu:~$ _
```

c. Running SysBench FileIO Testcase:

```
$ sysbench -test=fileio -file-total-size=2G -file-test-mode=seqwr \
prepare

$ sysbench -test=fileio -file-total-size=2G -file-test-mode=seqwr \
run

$ sysbench -test=fileio -file-total-size=2G -file-test-mode=seqwr \
cleanup
```

```
QEMU
Creating file test_file.0
Creating file test_file.95
Creating file test_file.96
Creating file test_file.97
Creating file test_file.98
Creating file test_file.99
Creating file test_file.100
Creating file test_file.101
Creating file test_file.102
Creating file test_file.103
Creating file test_file.104
Creating file test_file.105
Creating file test_file.106
Creating file test_file.107
Creating file test_file.108
Creating file test_file.109
Creating file test_file.110
Creating file test_file.111
Creating file test_file.112
Creating file test_file.113
Creating file test_file.114
Creating file test_file.115
Creating file test_file.116
Creating file test_file.117
Creating file test_file.118
Creating file test_file.119
Creating file test_file.120
Creating file test_file.121
Creating file test_file.122
Creating file test_file.123
Creating file test_file.124
Creating file test_file.125
Creating file test_file.126
Creating file test_file.127
2147483648 bytes written in 2.35 seconds (871.85 MiB/sec).
kartiki@ubuntu:~$
```

```
QEMU
lub total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        12328.29
  fsyncs/s:        15790.81

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   192.63

General statistics:
  total time:           10.0090s
  total number of events: 281330

Latency (ms):
  min:                 0.00
  avg:                 0.04
  max:                11.18
  95th percentile:     0.05
  sum:               9934.91

Threads fairness:
  events (avg/stddev): 281330.0000/0.00
  execution time (avg/stddev): 9.9349/0.00

kartiki@ubuntu:~$ sysbench --test=fileio --file-total-size=2G --file-test-mode=seqwr run.
```

```
kartiki@ubuntu:~$ sysbench --test=fileio --file-total-size=2G --file-test-mode=seqwr cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
kartiki@ubuntu:~$
```

d. Running SysBench Memory Testcase:

```
$ sysbench -test=memory -file-block-size=2G run
```

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Running memory speed test with the following options:
  block size: 2097152KiB
  total size: 102400MiB
  operation: write
  scope: global

Initializing worker threads...

Threads started!

Total operations: 2 (    0.15 per second)
4096.00 MiB transferred (303.64 MiB/sec)

General statistics:
  total time:          13.4889s
  total number of events: 2

Latency (ms):
  min:                 6741.47
  avg:                 6742.67
  max:                 6743.86
  95th percentile:    6718.97
  sum:                 13485.33

Threads fairness:
  events (avg/stddev):   2.0000/0.00
  execution time (avg/stddev): 13.4853/0.00
kartiki@ubuntu:~$ sysbench --test=memory --memory-block-size=2G run
```

2. Image Format: qcow2

a. Creating qcow2 image:

```
$ sudo qemu-img create ubuntu.img 15G -f qcow2
```

```
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-img create ubuntu.img 15G -f qcow2
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=16106127360
lazy_refcounts=off refcount_bits=16
```

b. Booting Ubuntu:

```
1 sudo qemu-system-aarch64 \
2 -nodefaults -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 2 \
3 -drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
4 -drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
5 -device virtio-blk-device,drive=hd0,serial="dummyserial" \
6 -device virtio-net-device,netdev=net0 \
7 -netdev user,id=net0 \
8 -vga none -device ramfb \
9 -boot d \
10 -cdrom ubuntu-20.04.5-live-server-arm64.iso \
11 -boot strict=on \
12 -device usb-ehci -device usb-kbd -device usb-mouse -usb
```

c. qcow2 Image Variation

Steps to run QEMU VM with different sets of CPU and Memory values:

i. CPU = 2, Memory = 2G

```
sudo qemu-img create ubuntu.img 15G -f qcow2
Password:
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=16106127360 lazy_refcounts=off refcount_bits=16
kartikidindorkar@Kartikis-MacBook-Pro QMEU % ls
ubuntu-20.04.5-live-server-arm64.iso  ubuntu.img
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-system-aarch64 \
--nodefaults -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=None,file=ubuntu.img,format=qcow2,id=hdb0 \
-device virtio-blk-device,drive=hdb0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-boot d \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
kartikidindorkar@Kartikis-MacBook-Pro QMEU %
```

ii. CPU = 4, Memory = 2G

```
sudo qemu-img create ubuntu.img 15G -f qcow2
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=16106127360 lazy_refcounts=off refcount_bits=16
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-system-aarch64 \
--nodefaults -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 4 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=None,file=ubuntu.img,format=qcow2,id=hdb0 \
-device virtio-blk-device,drive=hdb0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-boot d \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-boot strict=on \
-device usb-ehci -device usb-kbd -device usb-mouse -usb
kartikidindorkar@Kartikis-MacBook-Pro QMEU % ls
```

iii. CPU = 4, Memory = 3G

```
sudo qemu-img create ubuntu.img 15G -f qcow2
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=16106127360 lazy_refcounts=off refcount_bits=16
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-system-aarch64 \
--nodefaults -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 4 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=None,file=ubuntu.img,format=qcow2,id=hdb0 \
-device virtio-blk-device,drive=hdb0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-boot d \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-boot strict=on \
-device usb-ehci -device usb-kbd -device usb-mouse -usb
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-system-aarch64 \
ls
```

iv. CPU = 2, Memory = 3G

```
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-img create ubuntu.img 15G -f qcow2
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=16106127360 lazy_refcounts=off refcount_bits=16
kartikidindorkar@Kartikis-MacBook-Pro QMEU % ls
ubuntu-20.04.5-live-server-arm64.iso  ubuntu.img
kartikidindorkar@Kartikis-MacBook-Pro QMEU % sudo qemu-system-aarch64 \
--nodefaults -accel hvf -cpu cortex-a57 -M virt,highmem=off -m 3072 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/8.2.0/share/qemu/edk2-aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=None,file=ubuntu.img,format=qcow2,id=hdb0 \
-device virtio-blk-device,drive=hdb0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-boot d \
-cdrom ubuntu-20.04.5-live-server-arm64.iso \
-boot strict=on \
-device usb-ehci -device usb-kbd -device usb-mouse -usb
```

d. Running SysBench Testcases

I have used shell scripts to take sysbench readings for all the SysBench testcases. Refer section [Automation](#) and [HW1-SysBenchReadings.xlsx](#) (attached in the HW1 folder as well) file for all the readings.

B. Docker (OS Virtualization)

The experiment involves multiple test cases where there are variations in CPU and RAM allocated to the container. Also, 3 test modes of SysBench will involve 2 variations each. Thus, the variables and their variations are listed below:

Sr No.	Variable Category	Variable Name	Parameter Name	Values
1	Docker	Number of CPU	--cpus	2, 4
2	Docker	Memory Allocated	--memory	2G, 3G
3	SysBench Test Mode	CPU	--cpu-max-prime	30000, 60000
		Memory	--memory-block-size	1G, 2G
		FileIO	--file-test-mode	Sequential Write (seqwr), Random Write (rndwr), Random Read-Write (rndrw)

1. \$ docker run --cpus=2 --memory=2g ubuntu2004

```
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
<none>          <none>    f49152ed1600  5 days ago   85.7MB
ubuntu2004      latest    d80a51818899  5 days ago   85.7MB
ubuntu_sysbench  latest    d80a51818899  5 days ago   85.7MB
ubuntusysbench  1.0      ec2117dc3365  5 days ago   85.7MB
ubuntu          20.04    fde9c12d7d3f  6 weeks ago  65.7MB
hello-world     latest    ee301c921b8a  9 months ago  9.14kB
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker run -it --cpus=2 --memory=2g ubuntu2004
root@fea62e4f9dba:/# ls
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
```

2. \$ docker run --cpus=2 --memory=3g ubuntu2004

```
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
<none>          <none>    f49152ed1600  5 days ago   85.7MB
ubuntu2004      latest    d80a51818899  5 days ago   85.7MB
ubuntu_sysbench  latest    d80a51818899  5 days ago   85.7MB
ubuntusysbench  1.0      ec2117dc3365  5 days ago   85.7MB
ubuntu          20.04    fde9c12d7d3f  6 weeks ago  65.7MB
hello-world     latest    ee301c921b8a  9 months ago  9.14kB
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker run -it --cpus=2 --memory=3g ubuntu2004
root@4757e3748020:/# ls
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@4757e3748020:/#
```

3. \$ docker run --cpus=4 --memory=3g ubuntu2004

```
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu2004      latest   d80a51818899  5 days ago   85.7MB
ubuntu_sysbench latest   d80a51818899  5 days ago   85.7MB
ubuntusysbench  1.0     ec2117dc3365  5 days ago   85.7MB
<none>          <none>   f49152ed1600  5 days ago   85.7MB
ubuntu          20.04   fde9c12d7d3f   6 weeks ago  65.7MB
hello-world     latest   ee301c921b8a   9 months ago  9.14kB
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker run -it --cpus=4 --memory=3g ubuntu2004
root@b09771237f37:/# ls
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@b09771237f37:/#
```

4. \$ docker run --cpus=4 --memory=2g ubuntu2004

```
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu2004      latest   d80a51818899  5 days ago   85.7MB
ubuntu_sysbench latest   d80a51818899  5 days ago   85.7MB
ubuntusysbench  1.0     ec2117dc3365  5 days ago   85.7MB
<none>          <none>   f49152ed1600  5 days ago   85.7MB
ubuntu          20.04   fde9c12d7d3f   6 weeks ago  65.7MB
hello-world     latest   ee301c921b8a   9 months ago  9.14kB
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker run -it --cpus=4 --memory=2g ubuntu2004
root@440f0522e6a0:/# ls
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@440f0522e6a0:/#
```

5. Running SysBench Testcases

I have used shell scripts to take sysbench readings for all the SysBench testcases. Refer section [Automation](#) and [HW1-SysBenchReadings.xlsx](#) (attached in the HW1 folder as well) file for all the readings.

6. Docker Image history:

```
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu_sysbench latest   d80a51818899  8 days ago   85.7MB
ubuntu2004      latest   d80a51818899  8 days ago   85.7MB
ubuntusysbench  1.0     ec2117dc3365  8 days ago   85.7MB
<none>          <none>   f49152ed1600  8 days ago   85.7MB
ubuntu          20.04   fde9c12d7d3f   6 weeks ago  65.7MB
hello-world     latest   ee301c921b8a   9 months ago  9.14kB
kartikidindorkar@Kartikis-MacBook-Pro HW1 % docker history d80a51818899
IMAGE      CREATED      CREATED BY      SIZE      COMMENT
d80a51818899  8 days ago   CMD ["/bin/bash"]  0B      buildkit.dockerfile.v0
<missing>    8 days ago   RUN /bin/sh -c apt-get update && apt-get...  20MB    buildkit.dockerfile.v0
<missing>    6 weeks ago  /bin/sh -c #(nop)  CMD ["/bin/bash"]  0B
<missing>    6 weeks ago  /bin/sh -c #(nop) ADD file:9ec8b7bbb2fbc8c90...  65.7MB
<missing>    6 weeks ago  /bin/sh -c #(nop) LABEL org.opencontainers....  0B
<missing>    6 weeks ago  /bin/sh -c #(nop) LABEL org.opencontainers....  0B
<missing>    6 weeks ago  /bin/sh -c #(nop) ARG LAUNCHPAD_BUILD_ARCH  0B
<missing>    6 weeks ago  /bin/sh -c #(nop) ARG RELEASE           0B
```

X. Automation

I have created 2 shell scripts which:

- Creates .csv file to store the results as:

“Test Mode | Parameter Value | Total Time (sec) | Number of Events | Events/Sec”

- Once the testcase runs, it gives the output as:

General statistics:	
total time:	10.0062s
total number of events:	305853

- I used pattern matching to get the exact values of total time and total number of events:

\s+	Matches one or more white spaces to detect space between “total time” and its actual value
\K	Once the whitespaces have been detected, we should start reading the actual value, thus ‘\K’ is used to reset the start of the reported match
\d+	Matched one or more digits
\.	Matches dot/decimal point

- For **fileio** test case, script will **prepare** the files, **run** the testcase and **remove** all the files after run.

A. QEMU

- Copy shell scripts from host OS to QEMU VM:

```
$ scp <host_username>@<ip_addr>:<path/of/files> <destination>
```

```
kartiki@ubuntu2004:~$ scp Kartikidindorkar@192.168.4.75:/Users/kartikidindorkar/Documents/Q5/run_cpu_memory.sh .
Password:
run_cpu_memory.sh                                         100% 1453      1.9MB/s   00:00
kartiki@ubuntu2004:~$ ls
run_cpu_memory.sh  run_fileio.sh
kartiki@ubuntu2004:~$ chmod +x run_cpu_memory.sh run_fileio.sh
kartiki@ubuntu2004:~$
```

- run_cpu_memory.sh

- Run the test mode as **cpu** with values **-cpu-max-prime** = 30000 and 60000
- Run the test mode as **memory** with values **-memory-block-size** = 1G and 2G

```

CloudComputing > HW1 > Shell Script > $ run_cpu_memory.sh
1 #!/bin/bash
2 # Calculate Events/Sec
3 cal_events_per_sec() {
4     local events=$1
5     local total_time=$2
6     local events_per_sec=$(bc <<< "scale=4; $events / $total_time")
7     echo $events_per_sec
8 }
9
10 # Run Sysbench test and save results in .csv file
11 run_sysbench_test(){
12     local test_mode=$1
13     local parameter_val=$2
14     local readings
15
16     for i in {1..5}; do
17         echo "Run : $i"
18
19         result=$(sysbench --test=$test_mode $parameter_val run | grep -E "total time:|total number of events:")
20         total_time=$(echo "$result" | grep -oP "total time:\s+\K(\d+\.\d+)")
21         total_events=$(echo "$result" | grep -oP "total number of events:\s+\K(\d+)")
22         events_per_sec=$(cal_events_per_sec $total_events $total_time)
23
24         echo "Total Time = $total_time, Total Events = $total_events, Events/Sec = $events_per_sec"
25
26         # Save in .csv
27         readings+=$(echo ",$total_time,$total_events,$events_per_sec")
28     done
29
30     param_val=$(echo $parameter_val | grep -oP '\d+G|\d+')
31     echo "$test_mode,$param_val,$readings" >> sysbench_readings.csv
32 }
33
34 # Main Function
35 # Creating .csv file with headers
36 echo "Test Mode,Parameter Value,Total Time(sec),Number of Events,Events/Sec" > sysbench_readings.csv
37
38 # SysBench Test Mode = cpu
39 run_sysbench_test cpu "--cpu-max-prime=30000"
40 run_sysbench_test cpu "--cpu-max-prime=60000"
41
42 # SysBench Test Mode = memory
43 run_sysbench_test memory "--memory-block-size=1G"
44 run_sysbench_test memory "--memory-block-size=2G"

```

c. Gives Output as:

```

kartiki@ubuntu:~$ ./run_cpu_memory.sh
Test Mode = cpu, Parameter = --cpu-max-prime=30000
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0004, Total Events = 25863, Events/Sec = 2586.1965
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0004, Total Events = 25976, Events/Sec = 2597.4961
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0003, Total Events = 26261, Events/Sec = 2626.0212
Run : 4
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0005, Total Events = 26093, Events/Sec = 2609.1695
Run : 5
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0005, Total Events = 25506, Events/Sec = 2550.4724
Test Mode = cpu, Parameter = --cpu-max-prime=60000
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0002, Total Events = 10154, Events/Sec = 1015.3796
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0009, Total Events = 10175, Events/Sec = 1017.4084
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.

```

d. Another Output is sysbench_readings.csv file:

CloudComputing > HW1 > Shell Script > qcows2_cpu_2_mem_2 > sysbench_readings.csv	
You, 11 seconds ago 1 author (You)	
1	Test Mode,Parameter Value,Total Time(sec),Number of Events,Events/Sec
2	cpu,30000,10.0006,25568,2556.6466,10.0003,26025,2602.4219,10.0004,26007,2600.5959,10.0005,25902,2590.0704,10.0001,25717,2571.6742
3	cpu,60000,10.0009,10047,1004.6095,10.0008,10060,1005.9195,10.0011,10055,1005.3894,10.0005,10109,1010.8494,10.0003,10076,1007.5697
4	memory,1G,3.3622,100,29.7424,3.1744,100,31.5020,3.1739,100,31.5069,3.3895,100,29.5028,3.4275,100,29.1757
5	memory,2G,12.6770,2,.1577,11.9641,2,.1671,11.9098,2,.1679,12.9530,2,.1544,12.5521,2,.1593
6	You, 11 seconds ago • Uncommitted changes

3. [run_fileio.sh](#)

a. Run the test mode as `fileio` with values `--file-test-mode`:

i. `seqwr` = Sequential Write

ii. rndwr = Random Write

iii. rndrw = Random Read and Write

```

CloudComputing > HW1 > Shell Script > $ run_fileio.sh
You, 2 days ago | 1 author (You)
1 #!/bin/bash
2 # Calculate Events/Sec
3 cal_events_per_sec() {
4     local events=$1
5     local total_time=$2
6     local events_per_sec=$(bc <<< "scale=4; $events / $total_time")
7     echo $events_per_sec
8 }
9 run_sysbench_fileio() {
10     local parameter_val=$1
11     local readings
12
13     echo "Parameter Value = $parameter_val"
14
15     sysbench --test=fileio --file-total-size=2G --file-test-mode=$parameter_val prepare
16     result=$(sysbench --test=fileio --file-total-size=2G --file-test-mode=$parameter_val run)
17     echo "result : $result"
18     sysbench --test=fileio --file-total-size=2G --file-test-mode=$parameter_val cleanup
19
20     total_time=$(echo "$result" | grep -oP "total time:\s+\K(\d+\.\d+)")
21     total_events=$(echo "$result" | grep -oP "total number of events:\s+\K(\d+)")
22     events_per_sec=$(cal_events_per_sec $total_events $total_time)
23
24     echo "Total Time = $total_time , Total Events = $total_events, Event/Sec = $events_per_sec"
25     You, 2 days ago • Created Shell script to run testcases and save th...
26     # Save in .csv
27     readings+=$(echo ",$total_time,$total_events,$events_per_sec")
28     echo "fileio,$parameter_val,$readings" >> sysbench_readings_fileio.csv
29 }
30
31 # Main Function
32 # Creating a .csv file with header
33 echo "Test Mode,Parameter Value,Total Time(sec),Number of Events,Events/Sec" > sysbench_readings_fileio.csv
34
35 # SysBench Test Mode = FileIo
36 run_sysbench_fileio "$1"
37

```

- b. For fileio test mode, we need to drop the cache of host os after every testcase, thus there is no for loop used in this script, after every single run of this file, dropped the cache on host machine using:

```
$ sync && sudo purge
```

```

|kartikidindorkar@Kartikis-MacBook-Pro ~ % sync && sudo purge
|Password:
|kartikidindorkar@Kartikis-MacBook-Pro ~ % sync && sudo purge
|^[[A^
|kartikidindorkar@Kartikis-MacBook-Pro ~ % sync && sudo purge
|kartikidindorkar@Kartikis-MacBook-Pro ~ %

```

- c. Gives Output as:

```

read/s:          0.00
writes/s:        13410.85
fsyncs/s:       17166.39

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   209.54

General statistics:
  total time:           10.0062s
  total number of events: 305853

Latency (ms):
  min:                  0.00
  avg:                  0.03
  max:                 10.64
  95th percentile:     0.05
  sum:                9923.50

Threads fairness:
  events (avg/stddev): 305853.0000/0.00
  execution time (avg/stddev): 9.9235/0.00
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
Total Time = 10.0062 , Total Events = 305853, Event/Sec = 30566.3488
kartiki@ubuntu:~$ cat sysbench_readings_fileio.csv
Test Mode,Parameter Value,Total Time(sec),Number of Events,Events/Sec
fileio,seqwr,,10.0115,296318,29597.7625
fileio,seqwr,,10.0089,285873,28561.8799
fileio,seqwr,,10.0080,291711,29147.7817
fileio,seqwr,,10.0090,293662,29389.7941
fileio,seqwr,,10.0062,305853,30566.3488
kartiki@ubuntu:~$
```

b. Another output is sysbench_readings_fileio.csv file:

CloudComputing > HW1 > Shell Script > qcow2_cpu_2_mem_2 > sysbench_readings_fileio.csv

You, 2 days ago | 1 author (You)

```

1 Test Mode,Parameter Value,Total Time(sec),Number of Events,Events/Sec
2 fileio,seqwr,10.0073,281784,28157.8447
3 fileio,seqwr,10.0054,302323,30215.9833
4 fileio,seqwr,10.0111,279015,27870.5636
5 fileio,seqwr,10.0036,299318,29921.0284
6 fileio,seqwr,10.0081,290150,28991.5168
7 fileio,rndwr,10.0093,197118,19693.4850
8 fileio,rndwr,10.0094,188927,18874.9575
9 fileio,rndwr,10.0098,185725,18554.3167
10 fileio,rndwr,,10.0082,190532,19037.5891
11 fileio,rndwr,10.0074,198262,19811.5394
12 fileio,rndrw,10.0085,202532,20235.9994
13 fileio,rndrw,10.0067,197711,19757.8622
14 fileio,rndrw,10.0049,189379,18928.6249
15 fileio,rndrw,10.0055,206675,20656.1391
16 fileio,rndrw,10.0064,194126,19400.1838
```

4. Copy Output files (SysBench Readings Output .csv) to host machine:

```
Kartiki@ubuntu:~$ scp sysbench_readings.csv sysbench_readings_fileio.csv kartikidindorkar@192.168.4.75:/Users/kartikidindorkar/Documents/Q5/2_2_
Password:
sysbench_readings.csv                                         100%   529    231.5KB/s  00:00
sysbench_readings_fileio.csv                                100%   670     1.4MB/s  00:00
kartiki@ubuntu:~$ _
```

B. Docker

1. Copy shell script to running container:

```
$ docker cp <file_path_host_mch> <image_id>:<path/docker>
```

```
kartikidindorkar@Kartikis-MacBook-Pro Q5 % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2bd6ac651c0 ubuntusysbench:1.0 "sysbench" 4 minutes ago Up 4 minutes dreamy_faraday
kartikidindorkar@Kartikis-MacBook-Pro Q5 % docker cp /Users/kartikidindorkar/Documents/Q5/run_cpu_memory.sh 2bd6ac651c0:/usr/src/
Successfully copied 3.07kB to 2bd6ac651c0:/usr/src/
kartikidindorkar@Kartikis-MacBook-Pro Q5 % docker exec -it 2bd6ac651c0 /bin/bash
root@2bd6ac651c0:/# cd /
sbin/  sys/
root@2bd6ac651c0:/# cd /usr/src/
root@2bd6ac651c0:/usr/src# ls
run_cpu_memory.sh run_fileio.sh
root@2bd6ac651c0:/usr/src# chmod +x run_cpu_memory.sh run_fileio.sh
root@2bd6ac651c0:/usr/src#
```

2. Running cpu and memory SysBench testcase same as VM:

```
$ ./run_cpu_memory.sh
```

```
root@0440f0522e6a0:/home# ls
run_cpu_memory.sh run_fileio.sh
root@0440f0522e6a0:/home# ./run_cpu_memory.sh
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0004, Total Events = 25435, Events/Sec = 2543.3982
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0006, Total Events = 25845, Events/Sec = 2584.3449
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0002, Total Events = 26168, Events/Sec = 2616.7476
Run : 4
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0002, Total Events = 25923, Events/Sec = 2592.2481
Run : 5
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0002, Total Events = 26182, Events/Sec = 2618.1476
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0011, Total Events = 10086, Events/Sec = 1008.4890
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0010, Total Events = 10109, Events/Sec = 1010.7989
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0006, Total Events = 10094, Events/Sec = 1009.3394
Run : 4
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0004, Total Events = 10121, Events/Sec = 1012.0595
Run : 5
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.0003, Total Events = 9998, Events/Sec = 999.7700
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 2.9178, Total Events = 100, Events/Sec = 34.2723
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 2.9197, Total Events = 100, Events/Sec = 34.2500
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 2.9145, Total Events = 100, Events/Sec = 34.3122
Run : 4
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 2.8946, Total Events = 100, Events/Sec = 34.5470
Run : 5
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 2.9151, Total Events = 100, Events/Sec = 34.3041
Run : 1
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.5199, Total Events = 3, Events/Sec = .2851
Run : 2
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 13.0443, Total Events = 4, Events/Sec = .3066
Run : 3
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 11.0680, Total Events = 4, Events/Sec = .3614
Run : 4
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.7459, Total Events = 4, Events/Sec = .3722
Run : 5
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
Total Time = 10.5651, Total Events = 3, Events/Sec = .2839
root@0440f0522e6a0:/home#
```

3. Running fileio test case of SysBench same as VM:

```
$ ./run_fileio.sh seqwr
```

```
root@440f0522e6a0:/home# ls
run_cpu_memory.sh run_fileio.sh sysbench_readings.csv
root@440f0522e6a0:/home# ./run_fileio.sh seqwr
Parameter Value = seqwr
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

128 files, 16384Kb each, 2048Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
Creating file test_file.8
Creating file test_file.9

Creating file test_file.126
Creating file test_file.127
2147483648 bytes written in 2.98 seconds (687.84 MiB/sec).
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
result : sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 16MiB each
2GiB total file size
Block size 16KiB
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing sequential write (creation) test
Initializing worker threads...

Threads started!

File operations:
  reads/s:          0.00
  writes/s:        16220.39
  fsyncs/s:        20764.10

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   253.44

General statistics:
  total time:       10.0048s
  total number of events: 369936

Latency (ms):
  min:              0.00
  avg:              0.03
  max:             20.25
  95th percentile:  0.03
  sum:            9925.23

Threads fairness:
  events (avg/stddev):    369936.0000/0.00
  execution time (avg/stddev):  9.9252/0.00
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
Total Time = 10.0048 , Total Events = 369936, Event/Sec = 36975.8515
root@440f0522e6a0:/home#
```

4. For fileio testcase, after running each testcase we need to drop cache from host VM, using:

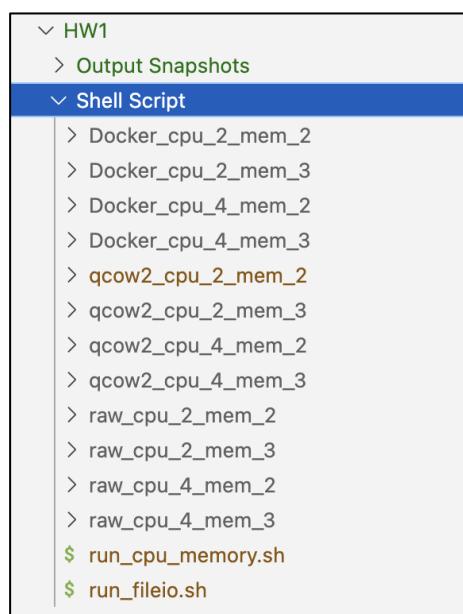
```
$ sync && sudo purge
```

5. Run for each test case 5 times and for every test mode (seqwr, rndwr, rndrw)

```
$ ./run_fileio.sh rndwr
```

```
$ ./run_fileio.sh rndrw
```

6. For Docker, output files, i.e., all the .csv files are stored in “[HW1/Shell Script/Docker_<testcase_mode>/](#)” folder
7. For VM, output files, i.e., all the .csv files are stored in “[HW1/Shell Script/<image_format>_<testcase_mode>/](#)” folder



8. All the readings are accumulated and stored in "[HW1-SysBenchReadings.xlsx](#)" (attached in HW1 folder) file to perform further calculations:

a. Docker readings:

		1		2		3		4		5		Min	Max	Avg	Std
		Total Time	No of Events/sec												
Docker	CPU	10.0004	26135. 2613.395	10.0003	26257. 2625.621	10.0002	26179. 2617.848	10.0004	26231. 2622.995	10.0004	25940. 2593.896	2593.896	2625.621	2614.751	12.57452
		10.0005	9946. 994.5502	10.0006	10111. 1011.039	10.0004	9909. 990.8603	10.0003	9996. 999.57	10.0009	9650. 964.9131	964.9131	1011.039	992.1866	17.03767
	Memory	3.4438	100. 29.0376	3.3926	100. 29.4759	3.4595	100. 28.9059	3.4553	100. 28.941	4.5701	100. 21.8813	21.8813	29.4759	27.64834	3.231919
		11.79	3. 0.2544	11.2558	4. 0.3553	11.1923	4. 0.3573	10.6232	4. 0.3765	10.0525	4. 0.3979	0.2544	0.3979	0.34828	0.055233
	FileIO	10.0041	364159. 36400.98	10.0044	373825. 37366.06	10.0039	385987. 38583.65	10.0036	368855. 36872.23	10.0036	349955. 34982.91	34982.91	38583.65	36841.16	1319.056
		10.0067	247058. 24689.26	10.0078	253644. 25344.63	10.0072	255704. 25552	10.0057	246621. 24648.05	10.0054	251424. 25128.83	24648.05	25552	25072.55	398.178
	Disk	10.0053	340258. 34007.78	10.0046	330076. 32992.42	10.0045	314831. 31468.94	10.0052	325039. 32487.01	10.0048	299063. 29891.95	29891.95	34007.78	32169.62	1568.063
		10.0005	25860. 2585.871	10.0005	26046. 2604.47	10.0002	25940. 2593.948	10.0005	26181. 2617.969	10.0003	25951. 2595.022	2585.871	2617.969	2599.456	12.27221
	Network	10.0002	10137. 1013.68	10.0004	10129. 1012.859	10.0011	10131. 1012.989	10.0009	10057. 1005.609	10.0012	10078. 1007.679	1005.609	1013.68	1010.563	3.664899
		3.006	100. 33.2667	3.1768	100. 31.4782	3.1619	100. 31.6265	3.1171	100. 32.0811	3.2005	100. 31.2451	31.2451	33.2667	31.93952	0.802256
	Process	11.9257	3. 0.2515	10.9503	3. 0.2739	10.2481	4. 0.3903	10.1458	3. 0.2956	11.1418	3. 0.2692	0.2515	0.3903	0.2961	0.054953
		10.0042	411649. 41147.62	10.0037	411030. 41078.8	10.0032	434320. 43418.11	10.0145	399558. 39897.95	10.0046	410092. 40990.34	39897.95	43418.11	41308.36	126.071
	IPC	10.0008	255233. 25509.2	10.0071	251831. 25165.23	10.0068	251842. 25167.09	10.0034	245555. 24547.15	10.0071	261185. 26099.97	24547.15	26099.97	25296.47	566.4938
		10.0056	343272. 34307.99	10.0038	333320. 33319.34	10.0042	340640. 34049.7	10.0006	329334. 32913.65	10.0054	310988. 31083.02	313083.02	34307.99	31314.7	127.5362
	Performance	10.0001	26138. 2613.774	10.0009	26029. 2602.666	10.0003	25355. 2535.424	10.0003	25285. 2528.424	10.0002	25675. 2567.449	2528.424	2613.774	2569.547	38.44717
		10.0014	9821. 981.9625	10.0013	9820. 981.8723	10.0007	9717. 971.6319	10.0009	9841. 984.0114	10.0004	10004. 1000.36	971.6319	1000.36	983.9676	10.35965
	Memory	3.606	100. 27.7315	4.1046	100. 24.3629	3.904	100. 25.6147	3.9411	100. 25.3736	3.7279	100. 26.8247	24.3629	27.7315	25.98148	1.313149
		10.7819	12. 1.1129	10.5785	12. 1.1343	10.1578	11. 1.0829	10.1019	12. 1.1878	10.2674	14. 1.3635	1.0829	1.3635	1.17628	0.111449
	FileIO	10.0051	382492. 38229.7	10.0038	397319. 39716.81	10.0036	383451. 38331.3	10.0036	408101. 40795.41	10.0045	367438. 36727.27	36727.27	40795.41	38760.1	1553.654
		10.0059	269661. 26950.2	10.0074	262539. 26234.49	10.0064	258459. 25829.37	10.0108	243377. 24311.44	10.0074	254788. 25459.96	24311.44	26950.2	25757.09	979.135
	Disk	10.0042	352361. 35221.31	10.0037	369786. 36964.92	10.0049	366904. 36672.43	10.0045	355022. 35486.23	10.0048	360407. 36023.41	35221.31	36964.92	36073.66	746.1533
		10.0003	26151. 2615.022	10.0003	26200. 2619.921	10.0004	26239. 2623.795	10.0001	26245. 2624.474	10.0002	25996. 2599.548	2599.548	2624.474	2616.552	10.22308
	Network	10.0007	10156. 1015.529	10.0008	10146. 1014.519	10.0004	10135. 1013.459	10.0005	10155. 1015.449	10.001	10150. 1014.899	1013.459	1015.529	1014.771	0.841802
		2.866	100. 34.8918	2.9885	100. 33.4616	2.9133	100. 34.3253	2.8797	100. 34.7258	2.8845	100. 34.668	34.8918	34.4415	35.71174	
	Process	3.6435	50. 13.723	3.2516	50. 15.377	3.2605	50. 15.335	3.4321	50. 15.4683	3.2785	50. 15.2508	13.723	14.85082	0.711174	
		10.0043	414654. 41447.58	10.0046	433354. 43315.47	10.0054	420627. 42040	10.0037	420425. 42026.95	10.0043	383861. 38369.6	43315.47	41439.92	1847.249	
	FileIO	10.0052	294319. 29416.6	10.0035	281803. 28170.44	10.007	289213. 28901.07	10.0048	262849. 26272.29	10.0086	273473. 27323.8	26272.29	29416.6	28016.84	1253.849
		10.0051	387262. 38706.46	10.0073	331179. 33093.74	10.0036	354538. 35441.04	10.0053	377900. 37769.98	10.0048	379266. 37908.4	33093.74	38706.46	36583.93	2299.657

b. QEMU VM Readings:

Sr. No.	Image Type	CPU	RAM	SysBenchmark Test Mode	Parameter	Value	1		2		3		4		5					
							Total Time	Total No. of Events	Events/sec	Total Time	Total No. of Events	Events/sec	Total Time	Total No. of Events	Events/sec	Total Time	Total No. of Events	Events/sec		
1	raw	CPU	40960	-	<cpu>	10.0004	26135. 2613.395	10.0004	26257. 2625.621	10.0002	26179. 2617.848	10.0004	26231. 2622.995	10.0004	25940. 2593.896	2593.896	2625.621	2614.751	12.57452	
						10.0005	9946. 994.5502	10.0006	10111. 1011.039	10.0004	10145. 1014.8603	10.0004	10117. 1011.8603	10.0008	10153. 1015.1617	1017.45	1013.369	0.07793		
2	raw	Memory	2 GB	<memory>	3.5379	100. 29.0376	3.2653	5.5857	100. 27.8985	3.4771	100. 28.7595	3.3917	100. 29.4537	3.4628	100. 28.875	28.8895	29.4537	28.8506	0.09969	
					14.1547	2. 0.1412	13.6529	2. 0.1464	13.5743	2. 0.1473	13.3663	2. 0.1496	13.6751	2. 0.1462	13.4132	0.1496	0.14610	0.03070		
3	raw	FileIO	<file-test-mode>	Sequential Write	10.0115	296318. 2967.7625	10.0089	285873. 28561.8799	10.0088	291711. 2914.7817	10.009	293662. 29359.7401	10.0062	293939. 29387.7401	10.0062	293397. 29332.7401	2933.74	2933.74	2933.74	
					10.0082	184412. 18426.9906	10.0089	183831. 18378.7625	10.0114	196939. 19672.1577	10.0108	178647. 1782.4577	10.0108	191855. 1917.9238	1917.9238	1917.9238	1917.9238			
4	raw	FileIO	<file-test-mode>	Random Write	10.0078	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781	10.0081	201508. 2013.4781
					10.0079	19998. 1998.4403	10.0081	19998. 1998.4403	10.0081	19998. 1998.4403	10.0081	19998. 1998.4403	10.0081	19998. 1998.4403	10.0081	19998. 1998.4403	10.0081			
5	qcow2	CPU	40960	<cpu>	3.2442	100. 29.0376	3.5171	100. 28.4325	3.8724	100. 28.2337	3.3504	100. 28.1916	3.1918	100. 28.0878	3.3461	100. 28.669	28.8373	30.8242	28.7104	0.87517
					10.9747	15. 1.0667	10.0721	13.2906	10.0712	13.2906	10.0712	13.2906	10.0712	13.2316	12.2336	14.096	13.0344	0.02133		
6	qcow2	Memory	2 GB	<memory>	10.0101	264061. 26397.6202	10.0089	249966. 24973.6163	10.0086	248661. 24852.8005	10.0085	248661. 24852.8005	10.0085	248661. 24852.						

XI. Analysis

Following is the result of our experiments:

Sr. No.	CPU	RAM	SysBench Test Mode	Parameter	Value	Avg - raw	Avg - qcow2	Avg - docker	Max	Fast VM
1	2	2	CPU	--cpu-max-prime"	30000	2593.871	2584.282	2294.762	2593.871	raw
			CPU	--cpu-max-prime"	60000	1012.359	1006.868	992.1866	1012.359	raw
			Memory	--memory-block-size"	1 GB	28.65506	30.28596	27.64839	30.28596	qcow2
	4	2	Memory	--memory-block-size"	2 GB	0.14614	0.16128	0.146857	0.16128	qcow2
			FileIO	--file-test-mode"	Sequential Write	29442.71	29031.39	25645.57	29442.71	raw
			FileIO	--file-test-mode"	Random Write	18420.14	19194.38	17677.44	19194.38	qcow2
			FileIO	--file-test-mode"	Random Read and Write	19716.22	19795.76	17776.65	19795.76	qcow2
			CPU	--cpu-max-prime"	30000	2592.076	2621.551	2599.456	2621.551	qcow2
			CPU	--cpu-max-prime"	60000	1004.87	1011.121	1010.563	1011.121	qcow2
			Memory	--memory-block-size"	1 GB	27.20228	27.80744	26.84655	27.80744	qcow2
2	4	2	Memory	--memory-block-size"	2 GB	0.13818	0.1493	0.1285	0.1493	qcow2
			FileIO	--file-test-mode"	Sequential Write	27630.26	24802.74	26716.53	27630.26	raw
			FileIO	--file-test-mode"	Random Write	18582.84	17484.97	17501.65	18582.84	raw
			FileIO	--file-test-mode"	Random Read and Write	18726.18	18312.25	17142.71	18726.18	raw
			CPU	--cpu-max-prime"	30000	2603.492	2616.486	2569.547	2616.486	qcow2
			CPU	--cpu-max-prime"	60000	1001.272	996.0145	983.9677	1001.272	raw
	4	3	Memory	--memory-block-size"	1 GB	28.7104	27.1094	25.98152	28.7104	raw
			Memory	--memory-block-size"	2 GB	1.3044	1.64482	1.17634	1.64482	qcow2
			FileIO	--file-test-mode"	Sequential Write	26084.91	25513.44	25765.45	26084.91	raw
			FileIO	--file-test-mode"	Random Write	17962.93	17848.25	17363.35	17962.93	raw
			FileIO	--file-test-mode"	Random Read and Write	23322.67	24602.74	24478.84	24602.74	qcow2
			CPU	--cpu-max-prime"	30000	2624.496	2567.652	2616.552	2624.496	raw
4	2	3	CPU	--cpu-max-prime"	60000	1014.153	1003.764	1012.951	1014.153	raw
			Memory	--memory-block-size"	1 GB	31.12834	25.89434	29.59206	31.12834	raw
			Memory	--memory-block-size"	2 GB	13.90158	1.23988	11.8807	13.90158	raw
			FileIO	--file-test-mode"	Sequential Write	28398.31	28594.86	23447.98	28594.86	qcow2
			FileIO	--file-test-mode"	Random Write	25610.72	19598.35	24418.86	25610.72	raw
			FileIO	--file-test-mode"	Random Read and Write	18673.12	26175.67	24590.32	26175.67	qcow2

- Based on our experiment results, we can conclude that the System Virtualization gives better performance i.e., a greater number of events per sec than OS virtualization.
- In the case of CPU test cases, as we increase our parameter's value, the events/seconds reduced.
- We have seen similar results in the case of memory test cases, where increasing memory block size gives reduced events/sec.
 - Also, when the memory allocated to the VM is increased, the events/sec has increased.
- For FileIO testcases, the sequential mode has performed well and given the maximum events/sec.

XII. Vagrant

1. Download Vagrant: <https://developer.hashicorp.com/vagrant/install> OR
2. Install Vagrant using brew:

```
$ brew install vagrant
```

3. Download and Install VirtualBox: <https://www.virtualbox.org/wiki/Downloads>
4. Search for the ubuntu box from: <https://app.vagrantup.com/boxes/search>
5. Create [Vagrantfile](#):

```
CloudComputing > HW1 > 📄 Vagrantfile
1   Vagrant.configure("2") do |config|
2     config.vm.box = "ubuntu/trusty64"
3     config.vm.provider "virtualbox" do |vb|
4       vb.memory = "2048"
5       vb.cpus = 4
6     end
7   end
```

6. Run the Vagrant:

```
$ vagrant up
```

```
vishwajeetkharote@Vishwajeets-MacBook-Pro vagrant_test % vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'ubuntu/trusty64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/trusty64' version '20190514.0.0' is up to date...
==> default: Setting the name of the VM: vagrant_test_default_1706671286713_37797
==> default: Clearing any previously set forwarded ports...
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:

  https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:

  VAGRANT_DISABLE_VBOXSYMLINKCREATE=1

or on a per folder basis within the Vagrantfile:

  config.vm.synced_folder '/host/path', '/guest/path', SharedFoldersEnableSymlinksCreate: false
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
  default: Adapter 1: nat
==> default: Forwarding ports...
  default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
  default: SSH address: 127.0.0.1:2222
  default: SSH username: vagrant
  default: SSH auth method: private key
  default: Warning: Connection reset. Retrying...
  default:
  default: Vagrant insecure key detected. Vagrant will automatically replace
  default: this with a newly generated keypair for better security.
  default:
  default: Inserting generated public key within guest...
  default: Removing insecure key from the guest if it's present...
  default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
  default: The guest additions on this VM do not match the installed version of
  default: VirtualBox! In most cases this is fine, but in rare cases it can
  default: prevent things such as shared folders from working properly. If you see
  default: shared folder errors, please make sure the guest additions within the
  default: virtual machine match the version of VirtualBox you have installed on
  default: your host and reload your VM.
  default:
  default: Guest Additions Version: 4.3.40
  default: VirtualBox Version: 7.0
==> default: Mounting shared folders...
  default: /vagrant => /Users/vishwajeetkharote/vagrant_test
vishwajeetkharote@Vishwajeets-MacBook-Pro vagrant_test % vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)
```

7. Once it is done, ssh into the VM:

```
$ vagrant ssh
```

```
[vishwajeetkharote@Vishwajeets-MacBook-Pro vagrant_test % vagrant ssh
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 3.13.0-170-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
 
 System information as of Wed Jan 31 03:21:49 UTC 2024

 System load:  0.61           Processes:          96
 Usage of /:   3.6% of 39.34GB  Users logged in:    0
 Memory usage: 7%            IP address for eth0: 10.0.2.15
 Swap usage:   0%

 Graph this data and manage this system at:
   https://landscape.canonical.com/
 
UA Infrastructure Extended Security Maintenance (ESM) is not enabled.

0 updates can be installed immediately.
0 of these updates are security updates.

Enable UA Infrastructure ESM to receive 64 additional security updates.
See https://ubuntu.com/advantage or run: sudo ua status

New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@vagrant-ubuntu-trusty-64:~$ ]
```

8. You can install SysBench in the virtual box and run the testcases:

```
[vagrant@vagrant-ubuntu-trusty-64:~$ sysbench --test=cpu --cpu-max-prime=30000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

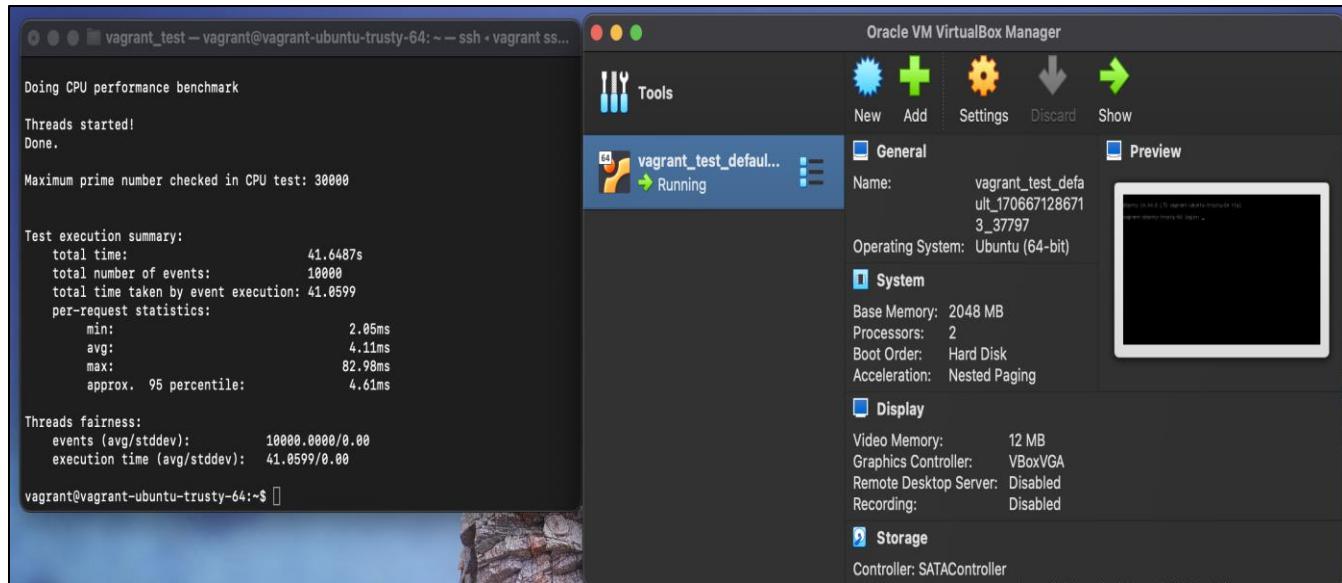
Maximum prime number checked in CPU test: 30000

Test execution summary:
  total time:                      41.6487s
  total number of events:           10000
  total time taken by event execution: 41.0599
  per-request statistics:
    min:                            2.05ms
    avg:                            4.11ms
    max:                            82.98ms
    approx. 95 percentile:          4.61ms

Threads fairness:
  events (avg/stddev):           10000.0000/0.00
  execution time (avg/stddev):    41.0599/0.00

vagrant@vagrant-ubuntu-trusty-64:~$ ]
```

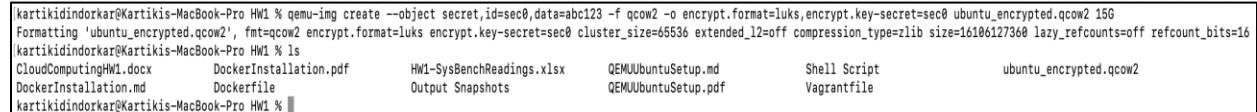
9. When you start the VM from terminal, you can see it is running in the virtual box application:



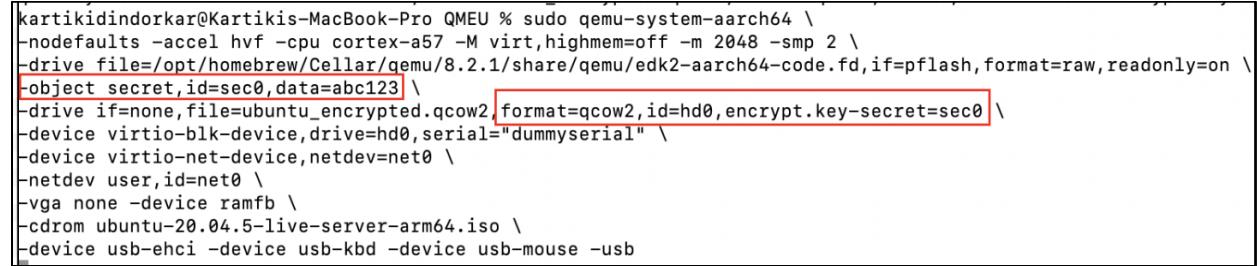
XIII. Encrypted Image

1. Create a qcow2 encrypted image using:

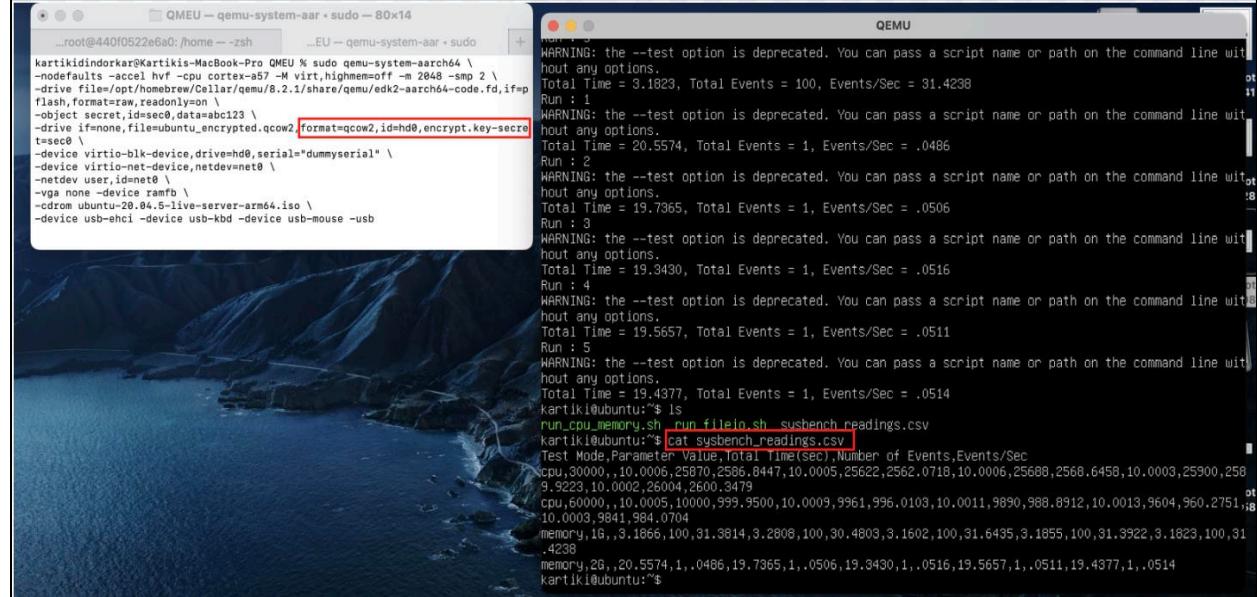
```
$ qemu-img create --object secret,id=sec0,data=abc123 -f qcow2 -o \
    encrypt.format=luks,encrypt.key-secret=sec0 \
    ubuntu_encrypted.qcow2 15
```



2. Install the VM using following command, where pass the secret-key to “encrypt.key-secret”



3. Run CPU SysBench test case:

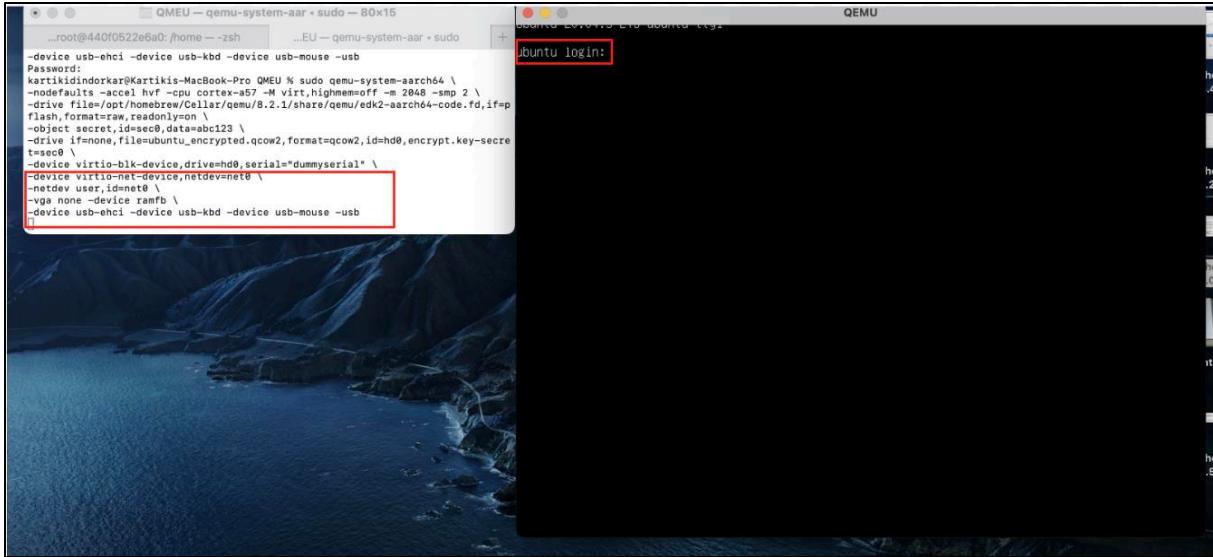


4. Run FileIo SysBench Testcases

5. Results for encrypted image

Sr. No.	Image Type	CPU	RAM	SysBench Test Mode	Parameter	Value	Min	Max	Avg	Std
1	Encrypted qcow2	2	2	CPU	--cpu-max-prime	30000	2562.072	2600.348	2581.567	15.79088
				CPU	--cpu-max-prime	60000	960.2751	999.95	985.8394	15.56063
				Memory	--memory-size	1 GB	30.4803	31.6435	31.26424	0.451099
				Memory	--memory-size	2 GB	0.0486	0.0516	0.05066	0.001212
				FileIO	--file-test-type	Sequential	14672.28	15709.03	15089.19	447.545
				FileIO	--file-test-type	Random Write	12097.39	12794.78	12460.02	316.435
				FileIO	--file-test-type	Random Read	14159.46	15040.86	14757.23	357.4191

6. Rebooting VM without “-cdrom” option



XIV. References

- [1] QEMU Manual: <https://www.qemu.org/docs/master/system/qemu-manpage.html>
- [2] SysBench Manual: <https://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf>
- [3] Vagrant in MAC OS: <https://github.com/ppggff/vagrant-qemu?tab=readme-ov-file>