COEN 241 : Cloud Computing

Homework 1
System vs. OS Virtualization

Submitted by – Shubham Murkute
(W1648631)

Environment Setup

Host System Configuration:

Operating System	Mac OS Monterey
Memory	16 GB
CPU	8 cores (4 performance and 4 efficiency)
Chip	Apple M1

System Virtualization Setup

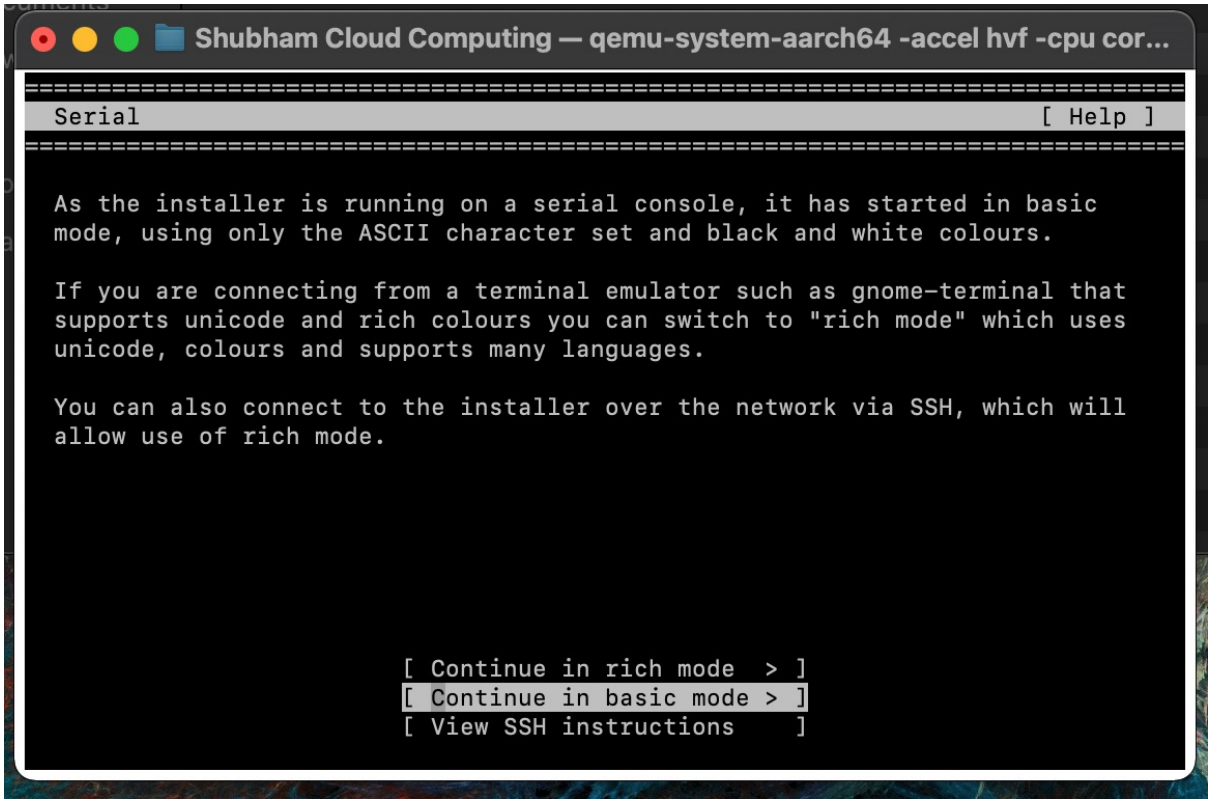
QEMU SETUP -

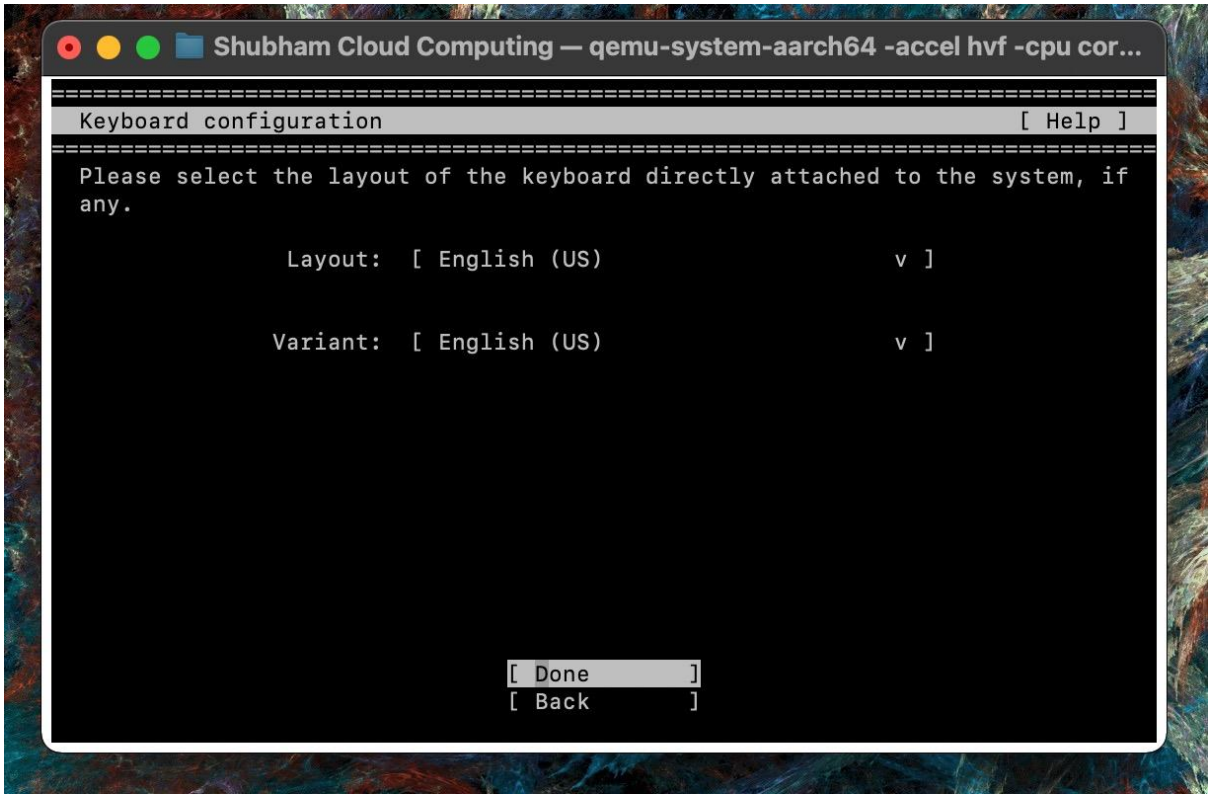
1. Install Homebrew using Rosetta 2 and add it to the PATH variable
    a. $ arch -x86_64 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
    b. export PATH="/opt/homebrew/bin:/usr/local/bin:$PATH"
2. Once Homebrew is installed, install required packages for building QEMU
    a. brew install libffi gettext pkg-config autoconf automake pixman
3. Run the following commands to build QEMU in the QEMU folder
    a. mkdir build
    b. cd build
    c. ../configure --target-list=aarch64-softmmu --disable-gnutls
    d. make -j8
    e. sudo make install
4. Downloading the Ubuntu Guest Virtual Machine :
    a. We will download the ISO Image based on our system configuration. The system's architecture is arm64v8. We download the image from the link given below –
        ARM (Apple Silicon) - Ubuntu 20.04 Server for ARM
    b. Create one folder to store the ubuntu image and the ISO file (ex. Shubham Cloud Computing)
5. Install QEMU using the following command – **brew install qemu**
6. Create QEMU image – Create the image in the same folder where the Ubuntu guest virtual machine ISO file was copied using the following command :
    qemu-img create ubuntu.img 40G -f qcow2
7. Install Ubuntu VM – Use the below command

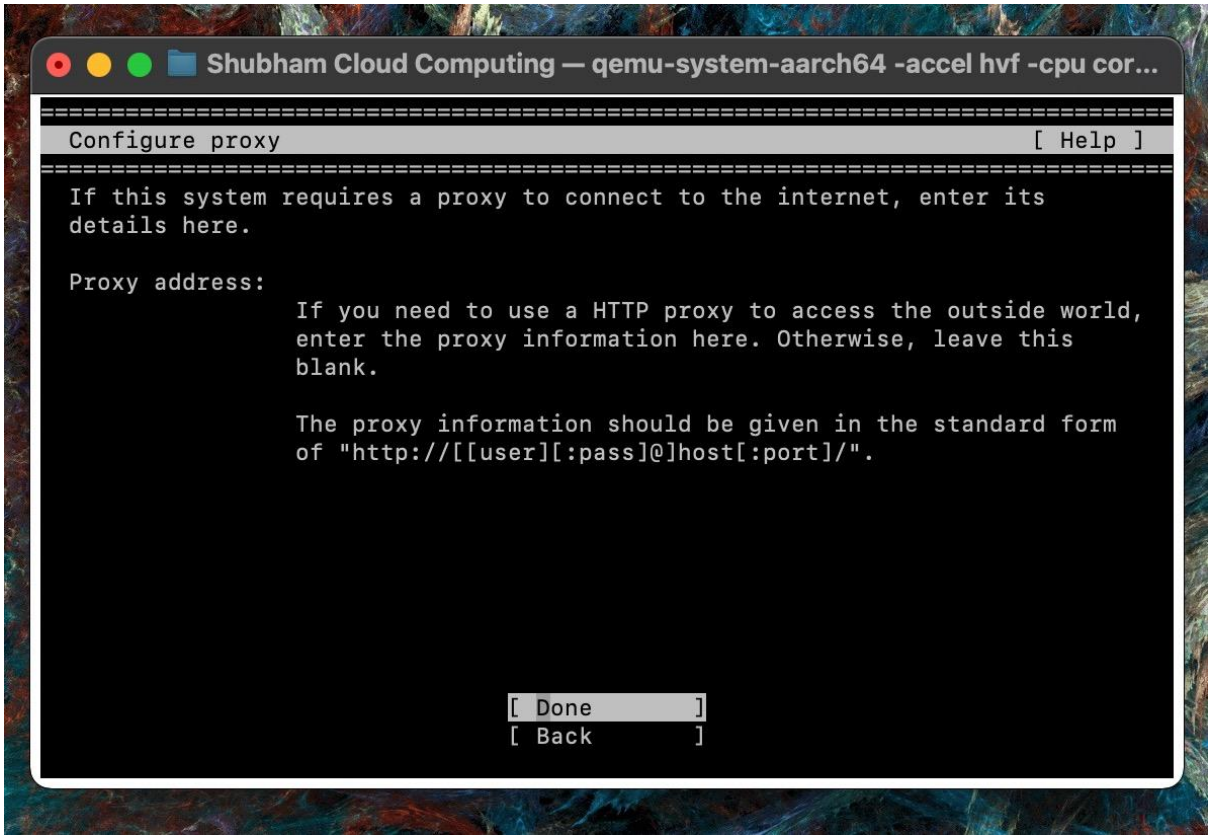
```
/opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/7.2.0/share/qemu/edk2-aarch64-
code.fd,if=pflash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-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 -nographic
```

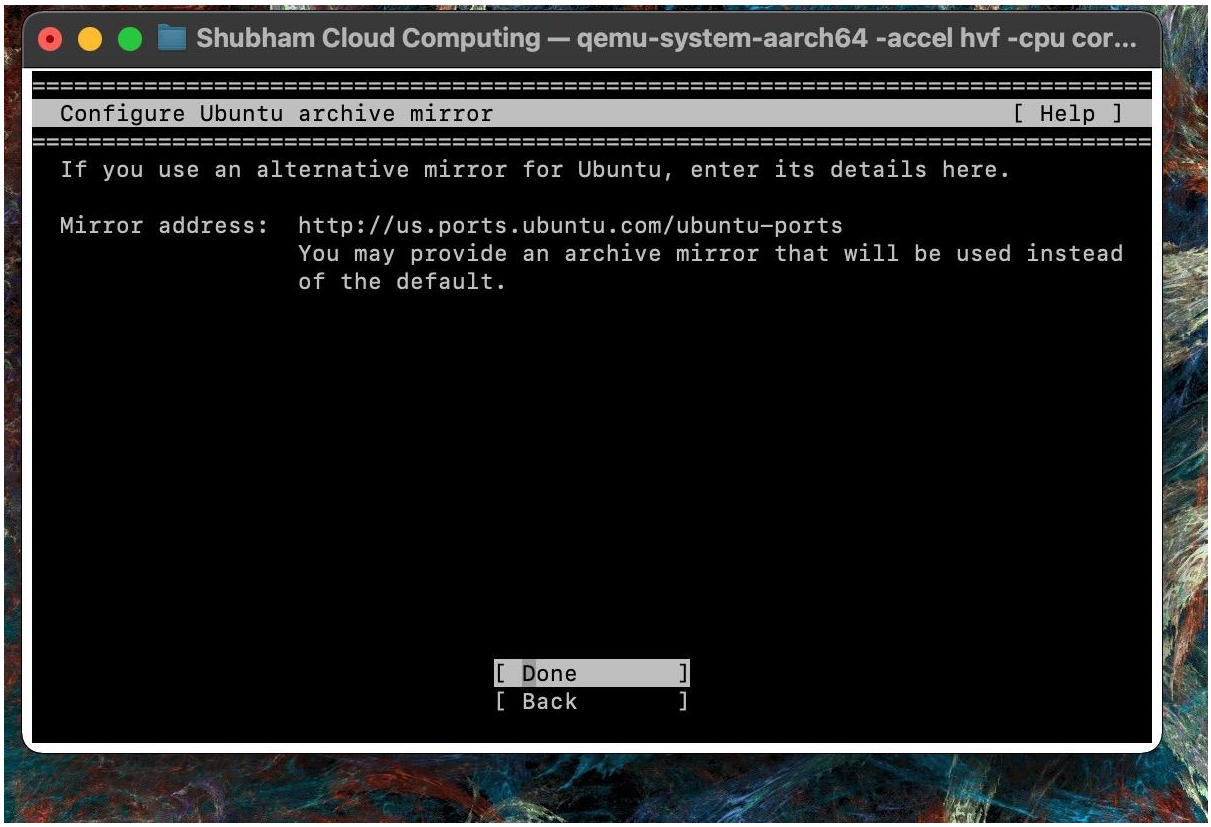
After the above command is executed, we need to setup and install QEMU.

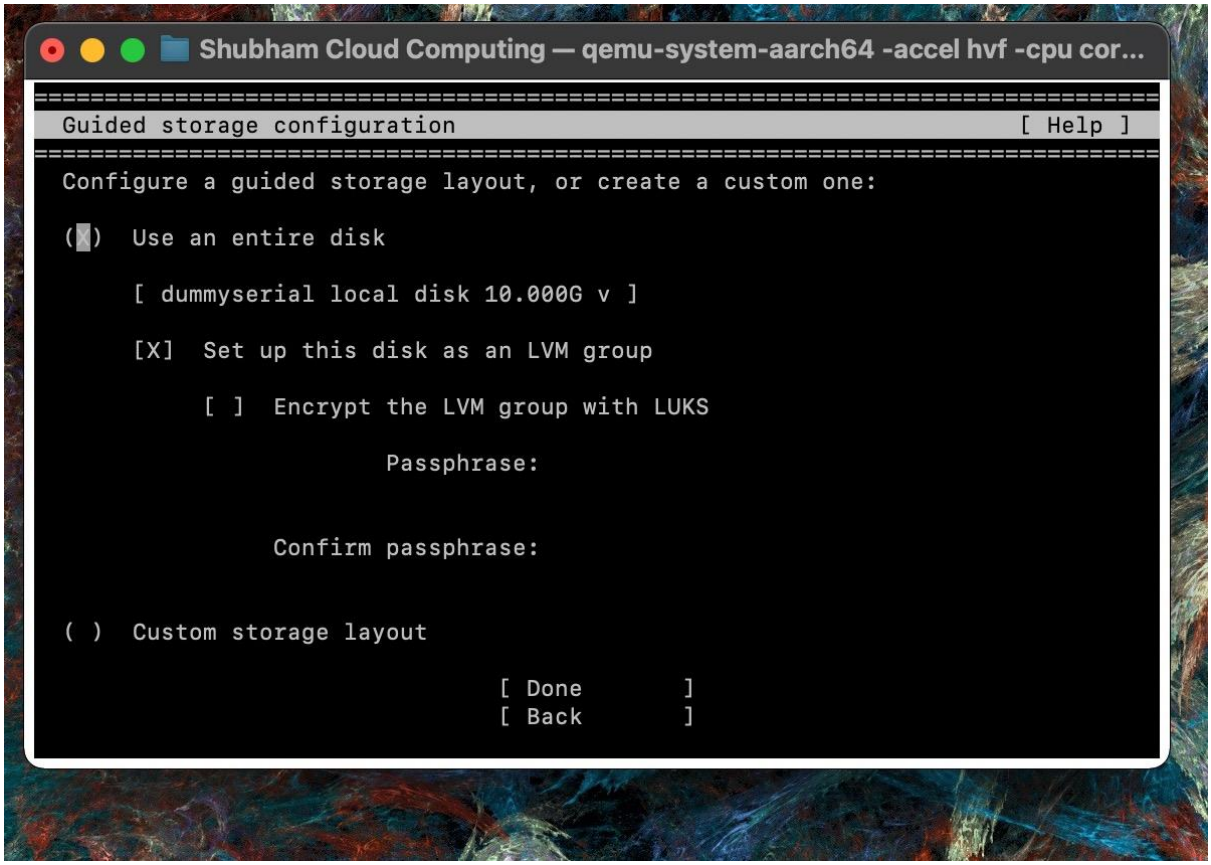
We continue in basic mode.

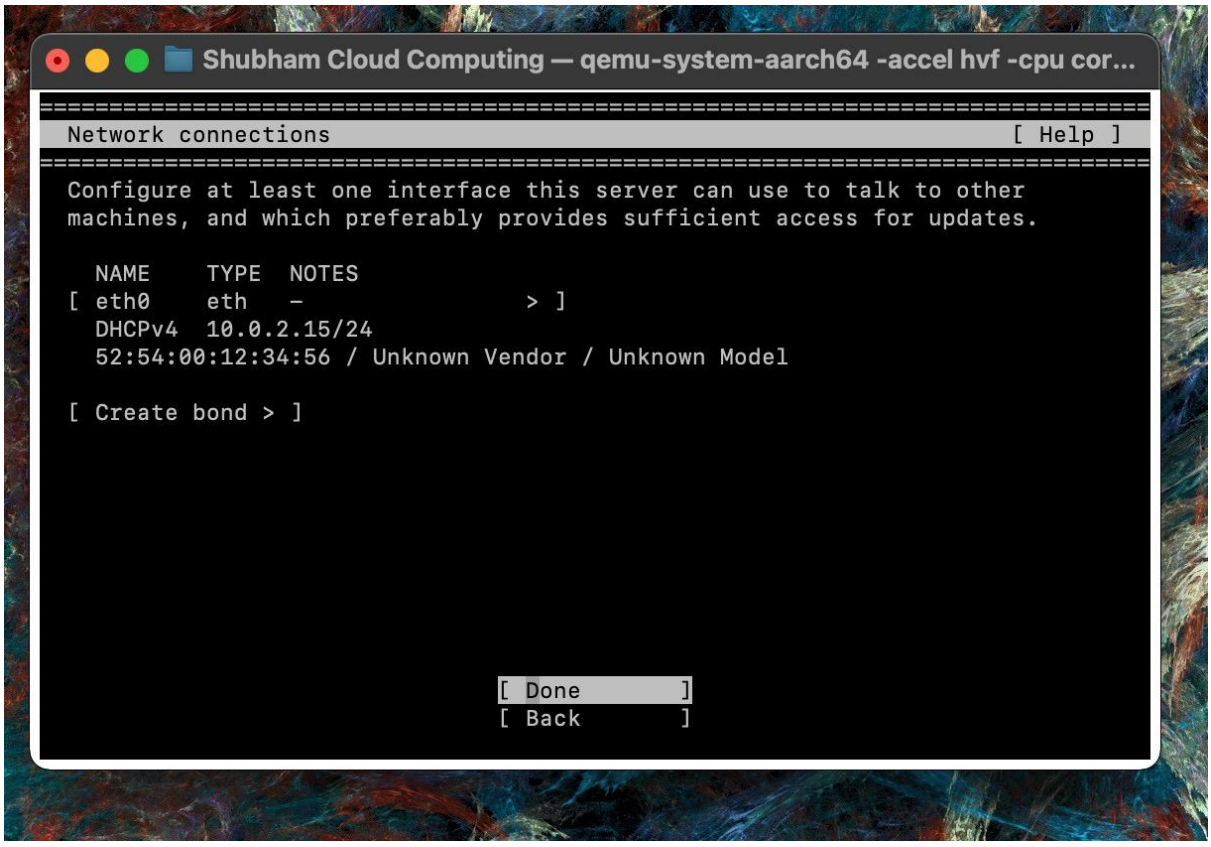


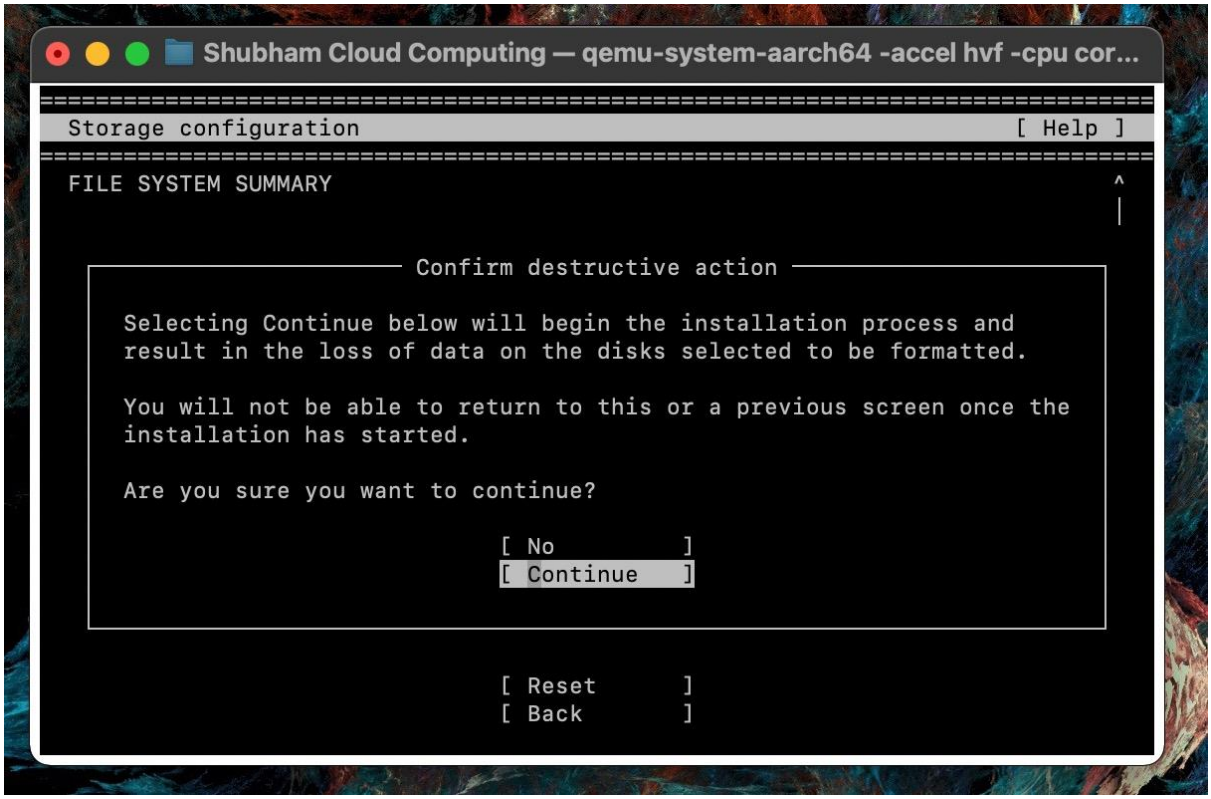

Shubham Cloud Computing — qemu-system-aarch64 -accel hvf -cpu cor...
Keyboard configuration
[Help]
Please select the layout of the keyboard directly attached to the system, if any.
Layout: [English (US) v]
Variant: [English (US) v]
[Done]
[Back]

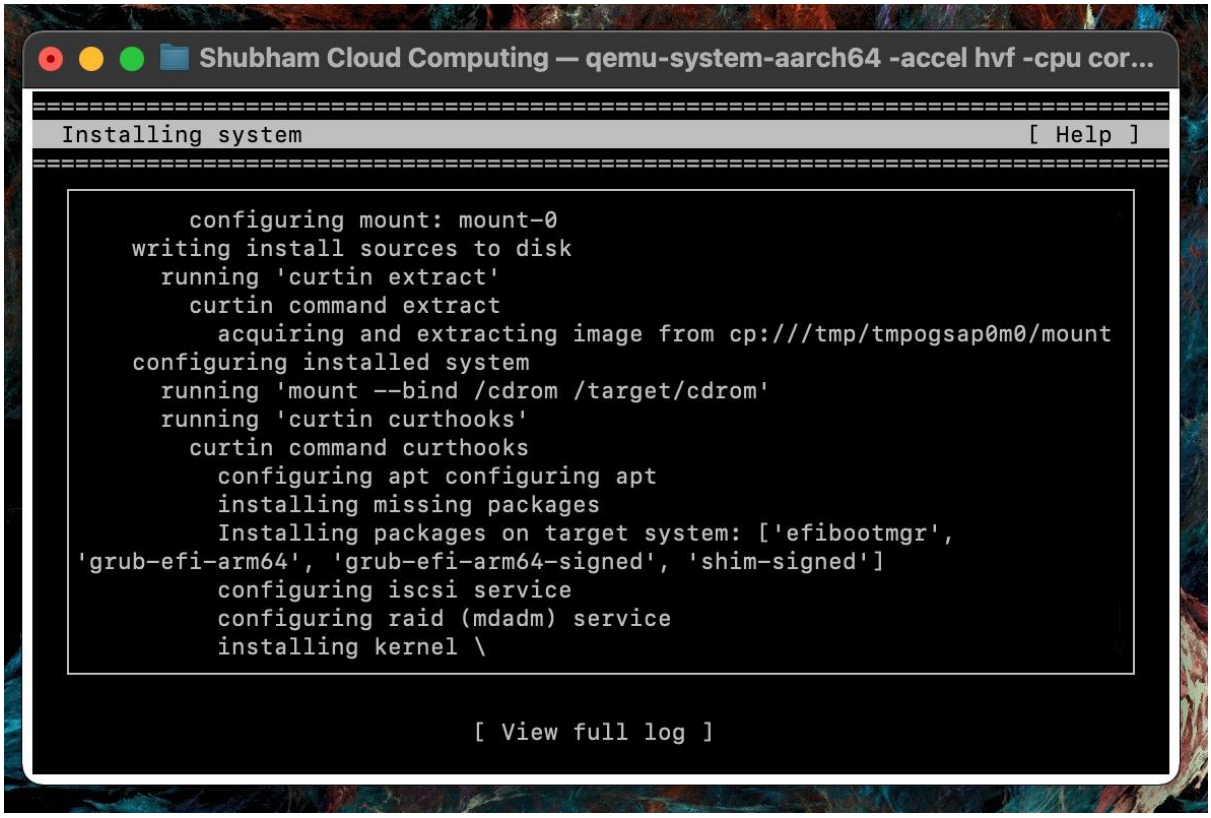

Shubham Cloud Computing — qemu-system-aarch64 -accel hvf -cpu cor...
Configure proxy
[Help]
If this system requires a proxy to connect to the internet, enter its details here.
Proxy address:
If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.
The proxy information should be given in the standard form of "http://[[user][:pass]@]host[:port]/".
[Done]
[Back]


Shubham Cloud Computing — qemu-system-aarch64 -accel hvf -cpu cor...
Configure Ubuntu archive mirror
[Help]
If you use an alternative mirror for Ubuntu, enter its details here.
Mirror address: http://us.ports.ubuntu.com/ubuntu-ports
You may provide an archive mirror that will be used instead of the default.
[Done]
[Back]


Shubham Cloud Computing — qemu-system-aarch64 -accel hvf -cpu cor...
Guided storage configuration [Help]
Configure a guided storage layout, or create a custom one:
(X) Use an entire disk
[dummyserial local disk 10.000G v]
[X] Set up this disk as an LVM group
[ ] Encrypt the LVM group with LUKS
Passphrase:
Confirm passphrase:
( ) Custom storage layout
[Done]
[Back]


Shubham Cloud Computing — qemu-system-aarch64 -accel hvf -cpu cor...
Network connections [Help]
Configure at least one interface this server can use to talk to other
machines, and which preferably provides sufficient access for updates.
NAME TYPE NOTES
[eth0 eth - >]
DHCPv4 10.0.2.15/24
52:54:00:12:34:56 / Unknown Vendor / Unknown Model
[Create bond >]
[Done]
[Back]

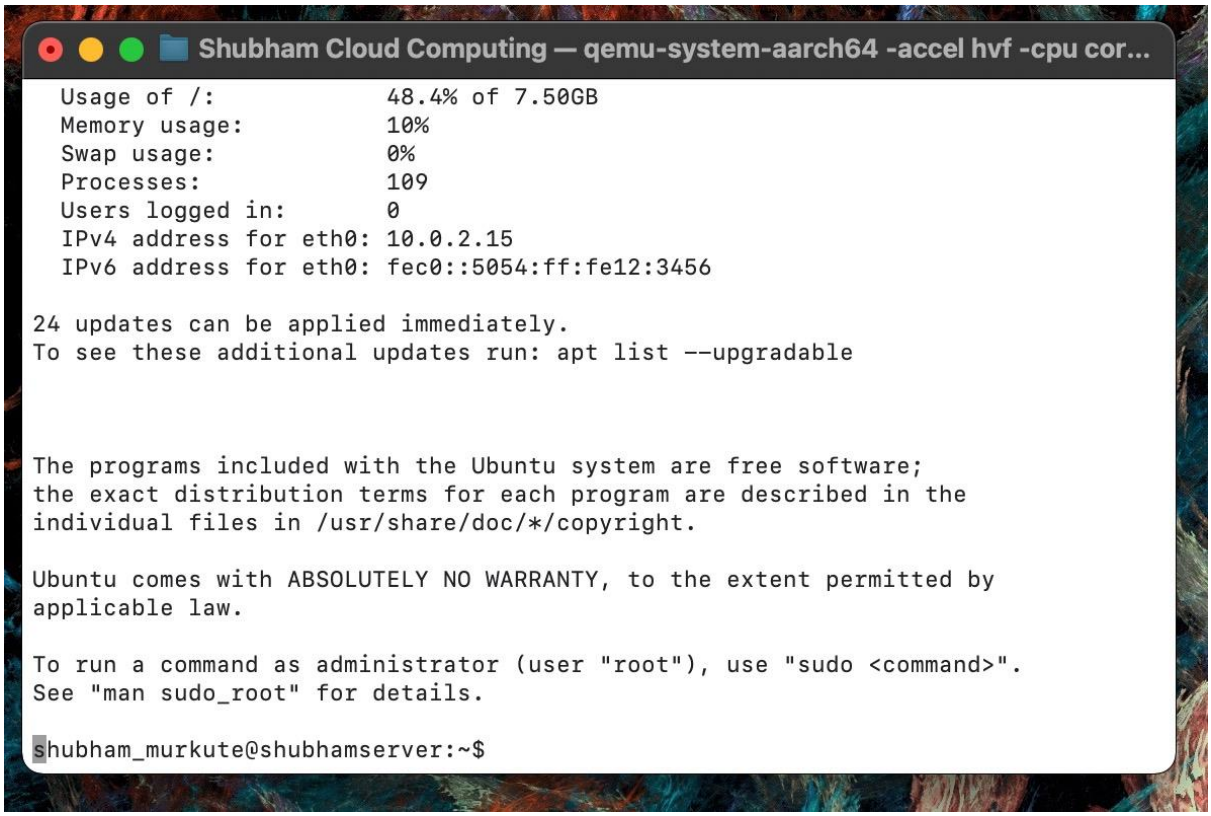




During the installation process we must provide a server name, username and a password which would be used to login and then we select the reboot option. After rebooting we can login using the above command but without the cdrom argument.

```
/opt/homebrew/bin/qemu-system-aarch64 \
-accel hvf -cpu cortex-a57 -M virt,highmem=off -m 2048 -smp 2 \
-drive file=/opt/homebrew/Cellar/qemu/7.2.0/share/qemu/edk2-
aarch64-code.fd,if=pflash,format=raw,readonly=on \
-drive if=none,file=ubuntu.img,format=qcow2,id=hd0 \
-device virtio-blk-device,drive=hd0,serial="dummyserial" \
-device virtio-net-device,netdev=net0 \
-netdev user,id=net0 \
-vga none -device ramfb \
-device usb-ehci -device usb-kbd -device usb-mouse -usb -nographic
```

After this we are prompted with username and password. Logging in with the correct credentials will log us in QEMU system.



Explanation of arguments used in the above command :

-accel	Activates an accelerator. Kvm, xen, hax, hvf, nvmm, whpx, or tcg could be available depending on the target architecture.
-m	This symbol represents memory. We are currently using 2G as the argument value, which means that we are allocating 2GB RAM memory to our VM.
-smp	Denotes the number of cores. We have currently set the argument value to 2, indicating that we have allocated two cores to our VM.
-cdrom	Use file as CD-ROM image (you cannot use -hdc and -cdrom at the same time). You can use the host CD-ROM by using /dev/cdrom as filename.
-device	Mainly to add a device driver where prop=value sets the driver properties.

DOCKER SETUP -

1. Installation of docker –
    a. We download Docker Desktop from the below link – https://docs.docker.com/desktop/mac/apple-silicon/
    b. Docker Desktop comes with Docker CLI and Docker Engine. Run the following commands in terminal to see if Docker is properly installed:

```
docker pull hello-world

docker images

docker run hello-world
```

2. Docker Image Creation and running the docker image –
   We create a docker image from the docker file using the following commands –

```
docker images

sudo docker run -it -entrypoint
“bin/bash/” arm64v8/ubuntu:20.04

apt update

apt install <package name>
```

3. Sysbench Installation –
   We need to make sure to use the same version of sysbench in QEMU as well as in Docker.

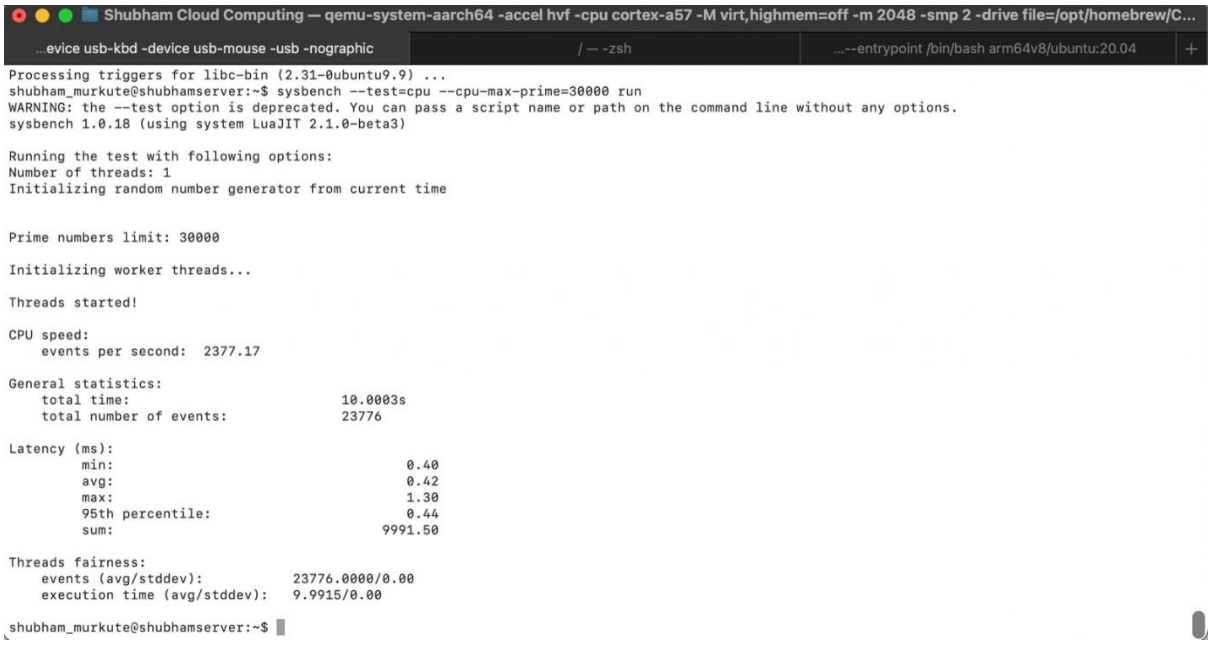
```
apt update

apt install sysbench
```

Proof of experiment

We will now run tests related to sysbench for CPU Performance and File I/O both on QEMU and Docker respectively.

QEMU Tests -

1. CPU TESTS –
    a. **Test 1 –** ***sysbench --test=cpu --cpu-max-prime=30000***



    b. **Test 2** – ***sysbench --test=cpu --cpu-max-prime=50000***

```
shubham_murkute@shubhamserver:~$ sysbench --test=cpu --cpu-max-prime=50000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wi
thout 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: 50000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1212.40

General statistics:
    total time:                          10.0004s
    total number of events:              12126

Latency (ms):
         min:                                    0.79
         avg:                                    0.82
         max:                                    1.24
         95th percentile:                        0.86
         sum:                                 9985.56

Threads fairness:
    events (avg/stddev):           12126.0000/0.00
    execution time (avg/stddev):   9.9856/0.00
```

c. **Test 3** – ***sysbench --test=cpu --cpu-max-prime=20000***

```
shubham_murkute@shubhamserver:~$ sysbench --test=cpu --cpu-max-prime=20000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line wi
thout 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: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  4245.61

General statistics:
    total time:                          10.0004s
    total number of events:              42464

Latency (ms):
         min:                                    0.23
         avg:                                    0.24
         max:                                    1.25
         95th percentile:                        0.25
         sum:                                 9985.97

Threads fairness:
    events (avg/stddev):           42464.0000/0.00
    execution time (avg/stddev):   9.9860/0.00
```

2. FileIO Tests –
   a. **Test 1 –**
      - ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr prepare***

```
[shubham_murkute@shubhals         sysbench --num-threads=8 --test=fileio --file-total-size=1
[G --file-test-mode=rndwr prepare
[WARNING: the --test option is deprecated. You can pass a script name or path on the command
[line without any options.
[WARNING: --num-threads is deprecated, use --threads instead
[sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)
[
 128 files, 81920Kb each, 10240Mb 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.10
 Creating file test_file.11
 Creating file test_file.12
 Creating file test_file.13
 Creating file test_file.14
 Creating file test_file.15
 Creating file test_file.16
 Creating file test_file.17
 Creating file test_file.18
 Creating file test_file.19
 Creating file test_file.20
 Creating file test_file.21
 Creating file test_file.22
 Creating file test_file.23
 Creating file test_file.24
 Creating file test_file.25
 Creating file test_file.26
 Creating file test_file.27
 Creating file test_file.28
 Creating file test_file.29
 Creating file test_file.30
 Creating file test_file.31
 Creating file test_file.32
 Creating file test_file.33
 Creating file test_file.34
 Creating file test_file.35
 Creating file test_file.36
 Creating file test_file.37
 Creating file test_file.38
 Creating file test_file.39
 Creating file test_file.40
```

- ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr run***

```
shubham_murkute@shubhamserver:~$ sysbench --num-threads=8 --test=fileio --file-total-size=10
G --file-test-mode=rndwr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 8
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 80MiB each
10GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     24242.49
    fsyncs/s:                     31129.38

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               378.79

General statistics:
    total time:                          10.0109s
    total number of events:              553323

Latency (ms):
         min:                                    0.00
         avg:                                    0.14
         max:                                    8.98
         95th percentile:                        0.44
         sum:                                79653.46

Threads fairness:
    events (avg/stddev):           69165.3750/807.96
    execution time (avg/stddev):   9.9567/0.00
```

- ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr cleanup***

```
shubham_murkute@shubhamserver:~$ sysbench --num-threads=8 --test=fileio --file-total-size=10
G --file-test-mode=rndwr cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

b. Test 2 –

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr prepare***

```
shubham_murkute@shubhamserver:~$
shubham_murkute@shubhamserver:~$ sysbench --num-threads=4 --test=fileio --file-total-size=3G
 --file-test-mode=rndwr prepare
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

128 files, 24576Kb each, 3072Mb 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.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
Creating file test_file.21
Creating file test_file.22
Creating file test_file.23
Creating file test_file.24
Creating file test_file.25
Creating file test_file.26
Creating file test_file.27
Creating file test_file.28
Creating file test_file.29
Creating file test_file.30
Creating file test_file.31
Creating file test_file.32
Creating file test_file.33
```

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr run***

```
shubham_murkute@shubhamserver:~$ sysbench --num-threads=4 --test=fileio --file-total-size=3G
 --file-test-mode=rndwr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     16331.79
    fsyncs/s:                     20951.09

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               255.18

General statistics:
    total time:                          10.0128s
    total number of events:              372817

Latency (ms):
         min:                                    0.00
         avg:                                    0.11
         max:                                   17.88
         95th percentile:                        0.28
         sum:                                39621.48

Threads fairness:
    events (avg/stddev):           93204.2500/521.29
    execution time (avg/stddev):   9.9054/0.00
```

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr cleanup***

```
shubham_murkute@shubhamserver:~$ sysbench --num-threads=4 --test=fileio --file-total-size=3G
 --file-test-mode=rndwr cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

c. Test 3 –

- ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr prepare***

```
[shubham_murkute@shubhamserver:~$ sysbench --num-threads=6 --test=fileio --file-total-size=8G]
 --file-test-mode=rndwr prepare
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

128 files, 65536Kb each, 8192Mb 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.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
Creating file test_file.21
Creating file test_file.22
Creating file test_file.23
Creating file test_file.24
Creating file test_file.25
Creating file test_file.26
Creating file test_file.27
Creating file test_file.28
Creating file test_file.29
Creating file test_file.30
Creating file test_file.31
Creating file test_file.32
Creating file test_file.33
Creating file test_file.34
Creating file test_file.35
Creating file test_file.36
Creating file test_file.37
Creating file test_file.38
```

- ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr run***

```
[shubham_murkute@shubhamserver:~$ sysbench --num-threads=6 --test=fileio --file-total-size=8G
 --file-test-mode=rndwr run
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 6
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 64MiB each
8GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     20633.13
    fsyncs/s:                     26481.59

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               322.39

General statistics:
    total time:                          10.0145s
    total number of events:              471091

Latency (ms):
         min:                                    0.00
         avg:                                    0.13
         max:                                   34.31
         95th percentile:                        0.35
         sum:                                59653.37

Threads fairness:
    events (avg/stddev):           78515.1667/280.68
    execution time (avg/stddev):   9.9422/0.00
```

- ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr cleanup***

```
[shubham_murkute@shubhamserver:~$ sysbench --num-threads=6 --test=fileio --file-total-size=8G]
 --file-test-mode=rndwr cleanup
WARNING: the --test option is deprecated. You can pass a script name or path on the command
line without any options.
WARNING: --num-threads is deprecated, use --threads instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

DOCKER TESTS –

1. CPU TESTS –
    a. **Test 1 –** ***sysbench cpu --threads=1 --cpu-max-prime=30000 --time=10 run***

```
root@c40a2dd86f9e:/# sysbench cpu --threads=1 --cpu-max-prime=30000 --time=10 run
sysbench 1.0.20 (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:   322.92

General statistics:
    total time:                          10.0021s
    total number of events:              3231

Latency (ms):
         min:                                    2.96
         avg:                                    3.09
         max:                                    6.48
         95th percentile:                        3.36
         sum:                                 9991.02

Threads fairness:
    events (avg/stddev):           3231.0000/0.00
    execution time (avg/stddev):   9.9910/0.00
```

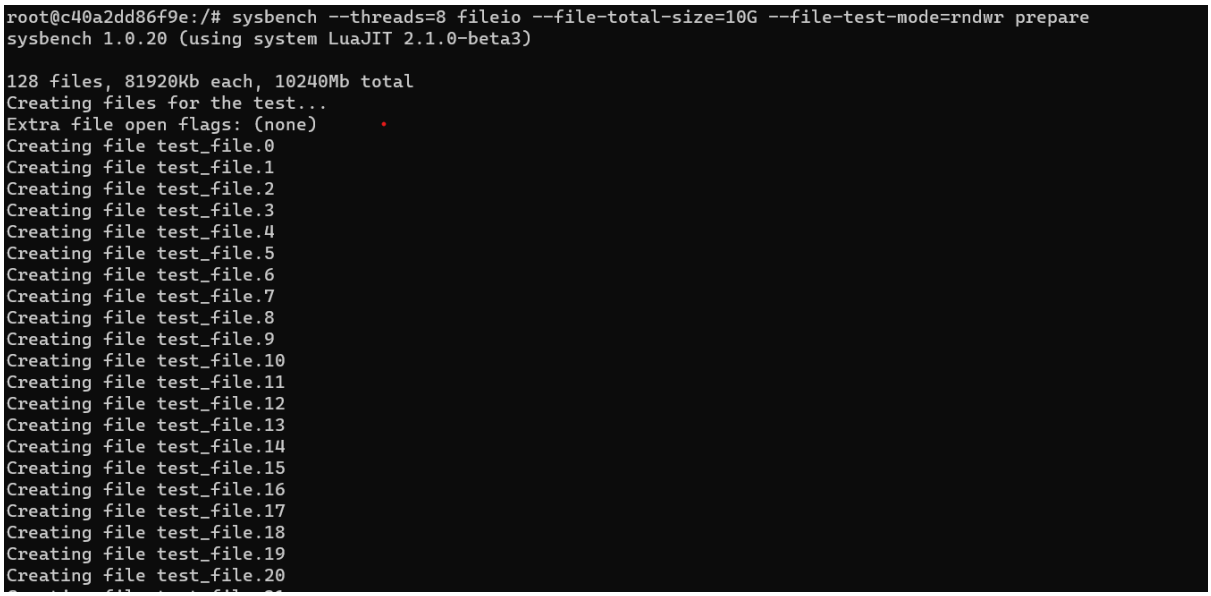
    b. **Test 2 –** ***sysbench cpu --threads=4 --cpu-max-prime=50000 --time=30 run***

```
root@c40a2dd86f9e:/# sysbench cpu --threads=4 --cpu-max-prime=50000 --time=30 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 50000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:   554.58

General statistics:
    total time:                          30.0053s
    total number of events:              16641

Latency (ms):
         min:                                    5.99
         avg:                                    7.21
         max:                                   14.20
         95th percentile:                        8.58
         sum:                               120000.44

Threads fairness:
    events (avg/stddev):           4160.2500/57.79
    execution time (avg/stddev):   30.0001/0.00
```

c. **Test 3** – ***sysbench cpu --threads=4 --cpu-max-prime=20000 --time=20 run***

```
root@c40a2dd86f9e:/# sysbench cpu --threads=4 --cpu-max-prime=20000 --time=20 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
    events per second:  1950.34

General statistics:
    total time:                          20.0015s
    total number of events:              39019

Latency (ms):
         min:                                    1.70
         avg:                                    2.05
         max:                                   10.05
         95th percentile:                        2.48
         sum:                               79986.05

Threads fairness:
    events (avg/stddev):           9754.7500/36.93
    execution time (avg/stddev):   19.9965/0.00
```

2. FILEIO TESTS –
    a. **Test 1 –**
        - ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr prepare***



        - ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr run***

```
root@c40a2dd86f9e:/# sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 8
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 80MiB each
10GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     6428.90
    fsyncs/s:                     8318.94

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               100.45

General statistics:
    total time:                          10.0937s
    total number of events:              147856

Latency (ms):
         min:                                    0.00
         avg:                                    0.54
         max:                                   53.84
         95th percentile:                        1.42
         sum:                                79877.42

Threads fairness:
    events (avg/stddev):           18482.0000/133.21
    execution time (avg/stddev):   9.9847/0.01
```

- ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr cleanup***

```
root@c40a2dd86f9e:/# sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

b. Test 2 –

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr prepare***

```
root@c40a2dd86f9e:/# sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 24576Kb each, 3072Mb 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.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
```

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr run***

```
root@c40a2dd86f9e:/# sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 4
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     6235.62
    fsyncs/s:                     8022.37

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               97.43

General statistics:
    total time:                          10.0528s
    total number of events:              142854

Latency (ms):
         min:                                    0.00
         avg:                                    0.28
         max:                                   53.44
         95th percentile:                        0.50
         sum:                                39910.68

Threads fairness:
    events (avg/stddev):           35713.5000/194.57
    execution time (avg/stddev):   9.9777/0.00
```

- ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr cleanup***

```
root@c40a2dd86f9e:/# sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

c. **Test 3 –**
   - ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr prepare***

```
root@c40a2dd86f9e:/# sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

128 files, 65536Kb each, 8192Mb 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.10
Creating file test_file.11
Creating file test_file.12
Creating file test_file.13
Creating file test_file.14
Creating file test_file.15
Creating file test_file.16
Creating file test_file.17
Creating file test_file.18
Creating file test_file.19
Creating file test_file.20
```

   - ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr run***

```
root@c40a2dd86f9e:/# sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 6
Initializing random number generator from current time


Extra file open flags: (none)
128 files, 64MiB each
8GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random write test
Initializing worker threads...

Threads started!


File operations:
    reads/s:                      0.00
    writes/s:                     6318.33
    fsyncs/s:                     8151.94

Throughput:
    read, MiB/s:                  0.00
    written, MiB/s:               98.72

General statistics:
    total time:                          10.0639s
    total number of events:              144889

Latency (ms):
         min:                                    0.00
         avg:                                    0.41
         max:                                  264.46
         95th percentile:                        0.78
         sum:                                59924.78

Threads fairness:
    events (avg/stddev):           24148.1667/310.01
    execution time (avg/stddev):   9.9875/0.00
```

- ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr cleanup***

```
root@c40a2dd86f9e:/# sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr cleanup
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Removing test files...
```

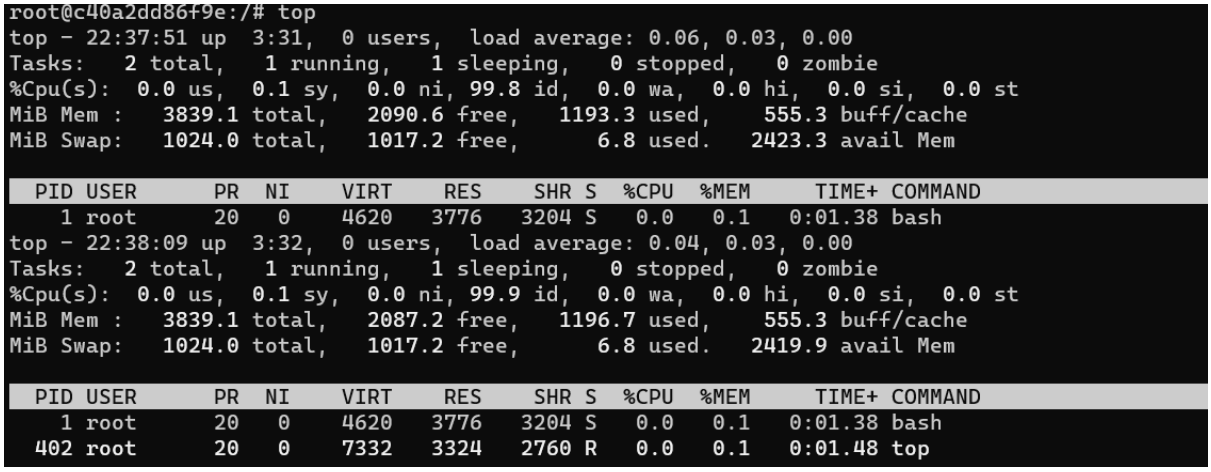
Analysis of Results

CPU Utlization and Performance –

We can view the CPU utilization using the top command.

1.QEMU – Encountered an issue with QEMU. Reinstalled and tried again but couldn't boot up QEMU again.

2.DOCKER –



<u>CPU TESTS</u> :

Three parameters are considered while performing the CPU test

- --cpu-max-prime : This sets the maximum prime number to check for in the CPU tests.
- --time : This sets the total time for the test in seconds.
- --threads : This sets the number of parallel threads to use in the test.

Test 1 - ***sysbench cpu --threads=1 --cpu-max-prime=30000 --time=10 run***

Platform	min	avg	max	Total no. of events
QEMU	0.40	0.42	1.30	23776
Docker	2.96	3.09	6.48	3231

Test 2 - ***sysbench cpu --threads=4 --cpu-max-prime=50000 --time=30 run***

Platform	min	avg	max	Total no. of events
QEMU	0.79	0.82	1.24	12126
Docker	5.99	7.21	14.2	16641

Test 3 - ***sysbench cpu --threads=4 --cpu-max-prime=20000 --time=20 run***

Platform	min	avg	max	Total no. of events
QEMU	0.23	0.24	1.23	42464
Docker	1.70	2.05	10.05	39019

FILE I/O TESTS :

Following parameters are considered to perform these tests, namely –

- --file-total-size : It sets the total size of the test file in megabytes.
- --file-test-mode : It sets the type of file I/O test to run.
- --threads : This sets the total number of threads used for the process.

Test 1 - ***sysbench --threads=8 fileio --file-total-size=10G --file-test-mode=rndwr run***

Platform	min	avg	max	Total no. of events
QEMU	0	0.14	8.98	553323
Docker	0	0.54	53.84	147856

Test 3 - ***sysbench --threads=4 fileio --file-total-size=3G --file-test-mode=rndwr run***

Platform	min	avg	max	Total no. of events
QEMU	0	0.11	17.88	372817
Docker	0	0.28	23.44	142854

Test 3 - ***sysbench --threads=6 fileio --file-total-size=8G --file-test-mode=rndwr run***

Platform	min	avg	max	Total no. of events
QEMU	0	0.13	34.31	471091
Docker	0	0.41	264.46	144889

CONCLUSION OF CPU & FILE IO TESTS

1. We can conclude that QEMU is faster than Docker as per the CPU test results.
2. As the cpu-max-prime argument value increases the latency increases but still QEMU is faster than Docker.
3. QEMU VM performed well in the file I/O tests than Docker although the total no. of events were more in case of QEMU VM.
4. Even in case when resources are increased and configuration is changed, QEMU performs better than Docker.

SHELL SCRIPTS FOR THE EXPERIMENT

1. Script for CPU TESTS –

```
#! /bin/bash

echo "CPU TESTS"

PRIMES_UPTO=("30000" "50000" "20000")
MAX_TIME=("10" "30" "20")
THREADS=("1" "4" "4")
TEST_RUNS=5
TEST_CASES=3

for ((i=0; i<$TEST_CASES;i++))
do
      echo "**************************Starting ${i+1} Test Case************************************"
      for (( j=1; j <=$TEST_RUNS; j++ ))
      do
            echo "Running ${j}th run of Test Case ${i+1}"
            sysbench cpu --threads=${THREADS[$i]} --cpu-max-prime=${PRIMES_UPTO[$i]} --time=${MAX_TIME[$i]} run
            echo "Completed ${j}th run of Test Case ${i+1}"
      done
      echo "**************************Completed ${i} Test Case************************************"
done
```

2. Script for FILE I/O TESTS –

```
#! /bin/bash

echo "FILE I/O TESTS"

THREADS=("8" "4" "6")
FILE_TOTAL_SIZES=("10G" "3G" "8G")
TEST_MODE=("rndwr" "seqrewr" "rndrw")
TEST_RUNS=5
TEST_CASES=3

for ((i=0; i<$TEST_CASES;i++))
do
      echo "**************************Starting ${i+1}st Test Case************************************"
      for (( j=1; j <=$TEST_RUNS; j++ ))
      do
            echo "Running ${j}st run of Test Case ${i+1}"
            sysbench --threads=${THREADS[$i]} fileio --file-total-size=${FILE_TOTAL_SIZES[$i]} --file-test-mode=${TEST_MODE[i]} prepare
            sysbench --threads=${THREADS[$i]} fileio --file-total-size=${FILE_TOTAL_SIZES[$i]} --file-test-mode=${TEST_MODE[i]} run
            sysbench --threads=${THREADS[$i]} fileio --file-total-size=${FILE_TOTAL_SIZES[$i]} --file-test-mode=${TEST_MODE[i]} cleanup
            echo "Completed ${j}st run of Test Case ${i+1}"
      done
      echo "**************************Completed ${i}st Test Case************************************"
done
```

Optional Part

DOCKER FILE AUTOMATION –

```
FROM shubham-murkute/sysbench-ubuntu:version1
COPY docker_script.sh /docker_script.sh
COPY cpu_test.sh /cpu_test.sh
COPY fileio_test.sh /fileio_test.sh
RUN chmod +x docker_script.sh
RUN chmod +x cpu_test.sh
RUN chmod +x fileio_test.sh
ENTRYPOINT bash docker_script.sh
```

VAGRANT FILE AUTOMATION –

```
Vagrant.configure("2") do |config|
   config.vm.box = "ubuntu/hirsute64"
   config.vm.provider "virtualbox" do |vb|
    vb.memory = "2048"
    vb.cpus = 2
   end
   config.vm.synced_folder "qemu", "/"
   config.vm.provision "shell", path: "vagrant_setup.sh"
end
```