

System Vs OS Virtualization

Environment Setup

Host System Configuration:

CPU: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz

RAM: 16 GB

Storage: 1 TB

System Type: 64-bit operating system, x64-based processor

Primary OS: Windows 11

QEMU VM Configuration:

CPU Cores: 2

RAM: 2 GB

Storage: 10 GB

OS: Ubuntu 22.04.1

Docker Container Configuration:

CPU Cores: 2

RAM: 2 GB

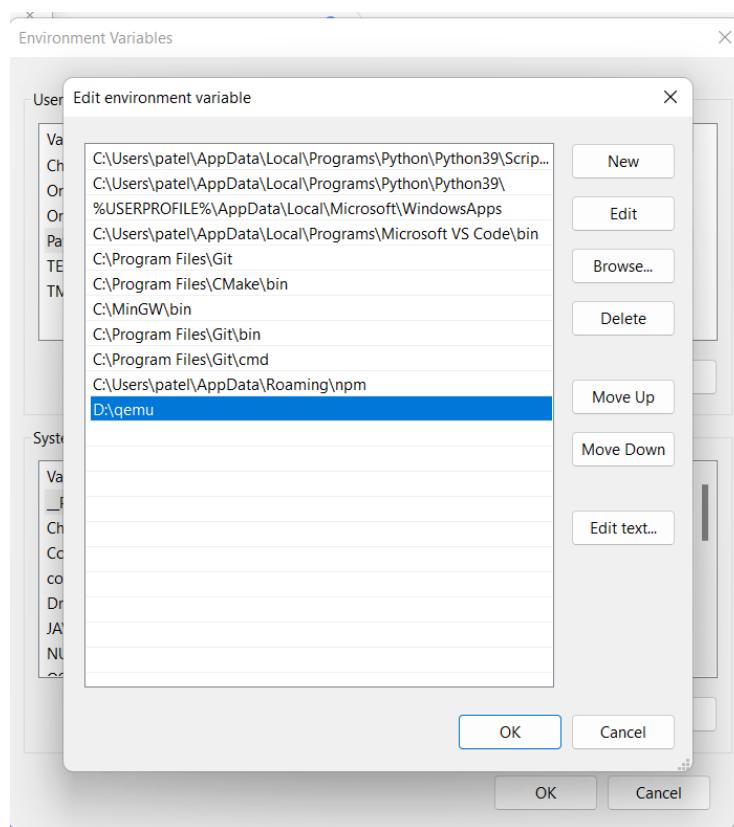
OS: Ubuntu 22.04.1

System Virtualization Setup

QEMU Setup:

Below are the steps followed to install QEMU on Windows

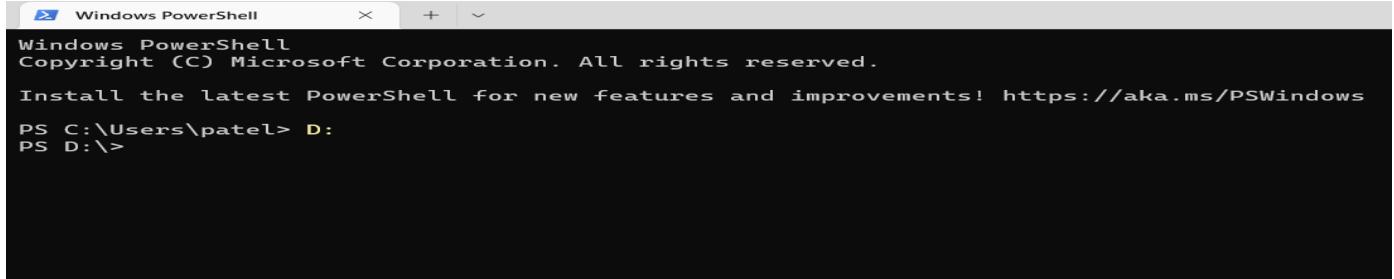
1. Downloaded ubuntu server image from the following link:
<https://releases.ubuntu.com/22.04/>
2. Downloaded QEMU for Windows(x64) from the following link:
<https://www.qemu.org/download/#windows>
After the download is complete, install it.
3. Add QEMU path into Environment Variables as shown in the image below



4. Open the command Prompt/Powershell and execute the following commands

a. Go to the directory where QEMU is installed(In this case: D:)

D:



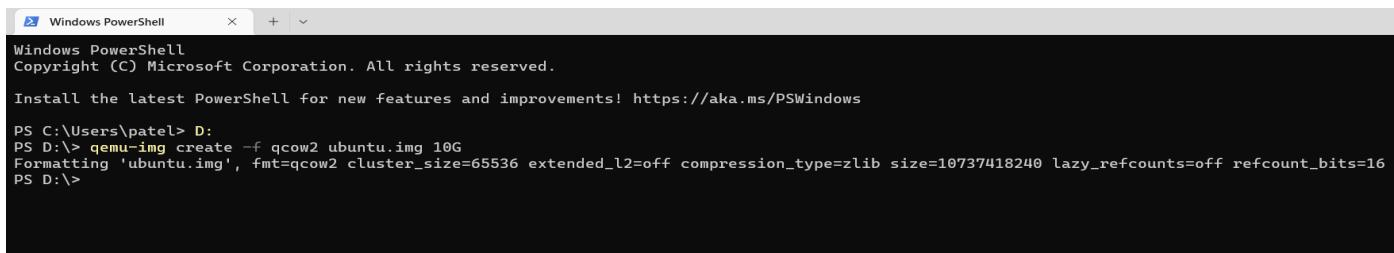
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patel> D:
PS D:>
```

b. Create QEMU image of ubuntu in qcow2 file format using following command

qemu-img create -f qcow2 ubuntu.img 10G



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

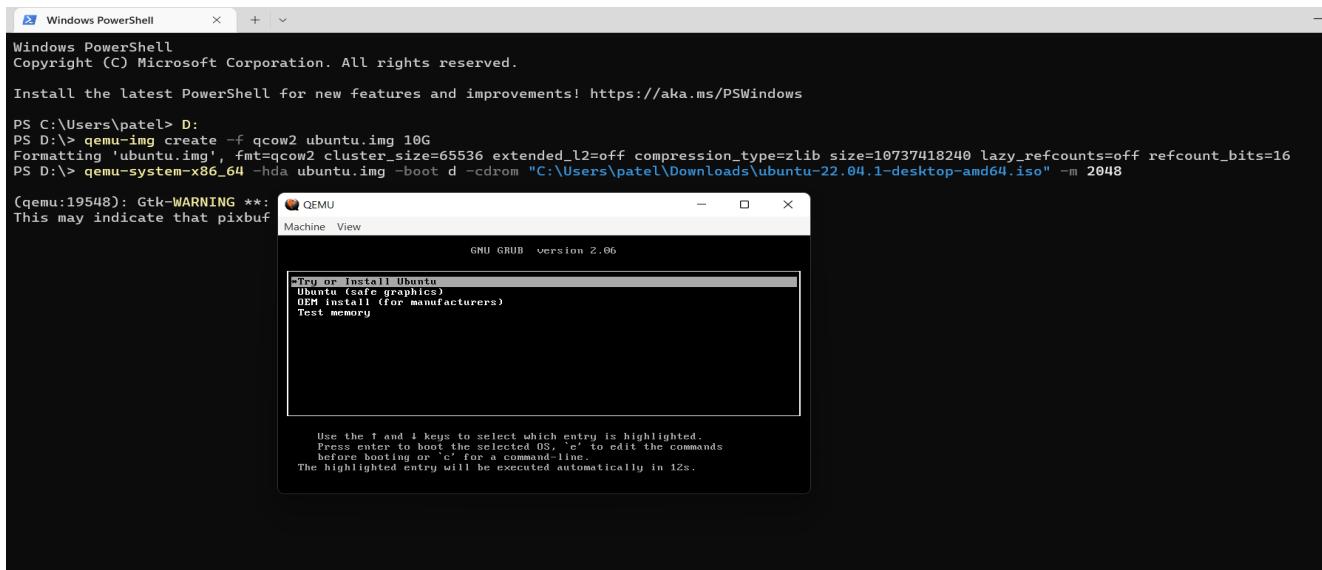
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patel> D:
PS D:> qemu-img create -f qcow2 ubuntu.img 10G
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=10737418240 lazy_refcounts=off refcount_bits=16
PS D:>
```

c. Boot iso file on QEMU with the following command

qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom

"C:\Users\patel\Downloads\ubuntu-22.04.1-desktop-amd64.iso" -m 2048

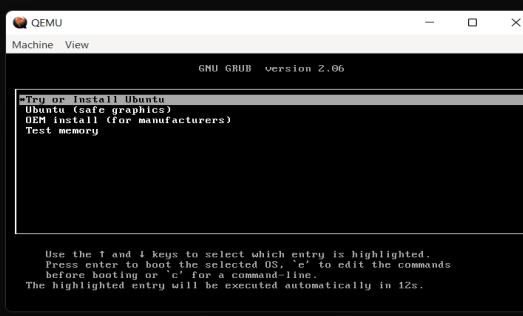


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patel> D:
PS D:> qemu-img create -f qcow2 ubuntu.img 10G
Formatting 'ubuntu.img', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib size=10737418240 lazy_refcounts=off refcount_bits=16
PS D:> qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom "C:\Users\patel\Downloads\ubuntu-22.04.1-desktop-amd64.iso" -m 2048

(qemu:19548): Gtk-WARNING **:
This may indicate that pixbuf
```

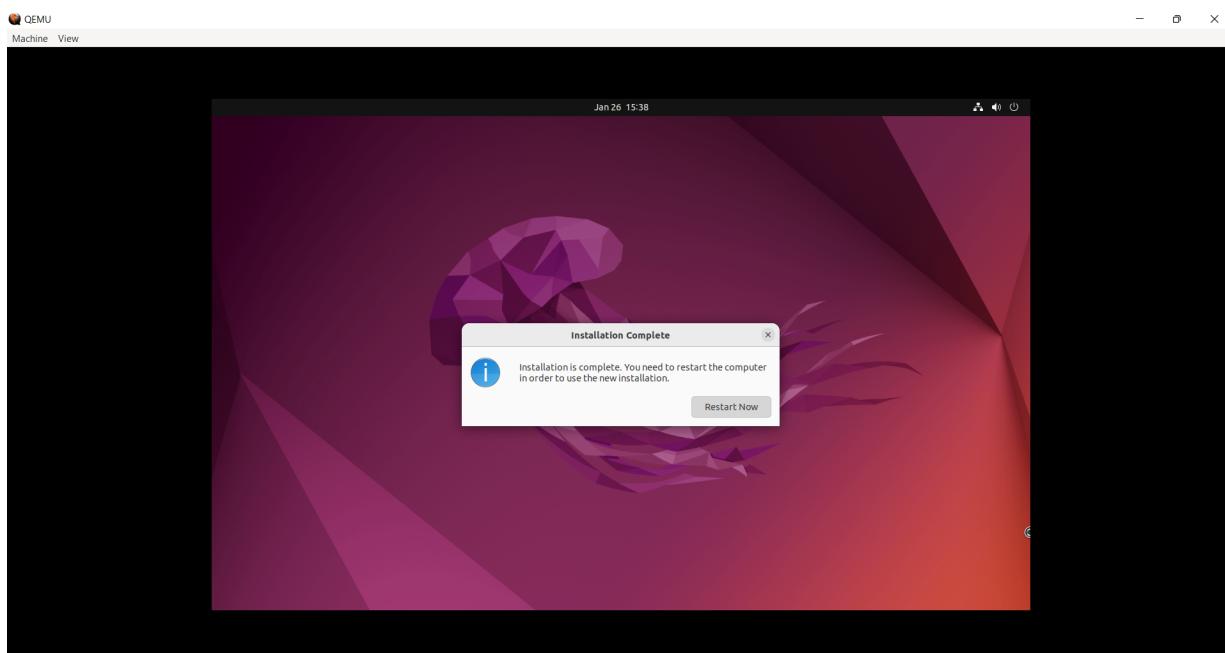
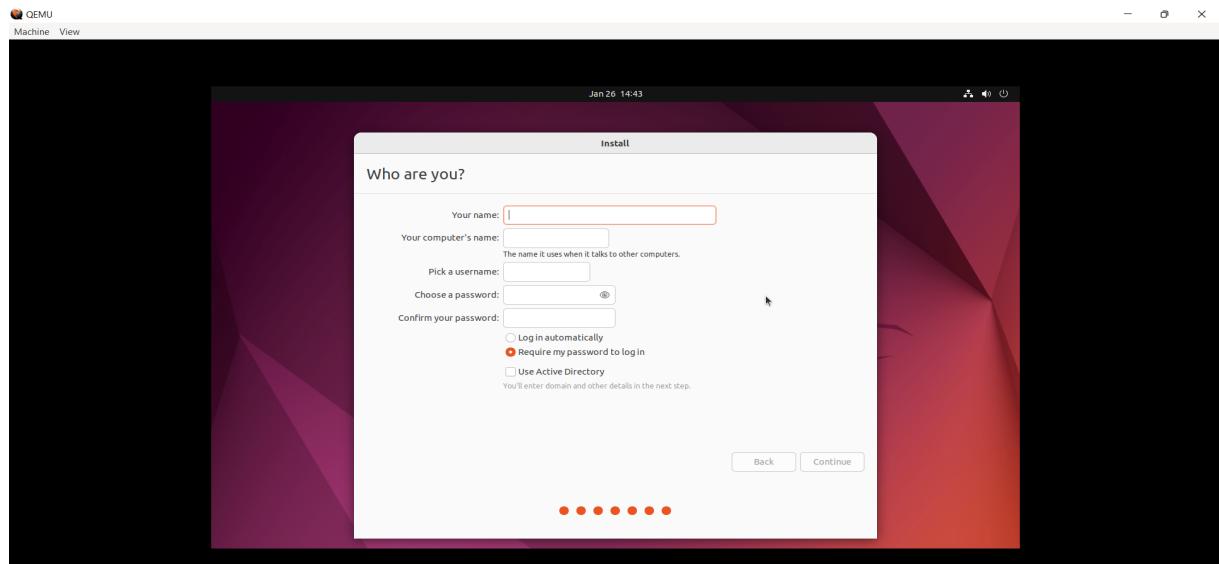


GRUB version 2.06 (Ubuntu)

Ubuntu (safe graphics)
DEB install (for manufacturers)
Test memory

Use the **f** and **j** keys to select which entry is highlighted.
Press enter to boot the selected OS, **e** to edit the commands before booting or **c** for a command-line.
The highlighted entry will be executed automatically in 10s.

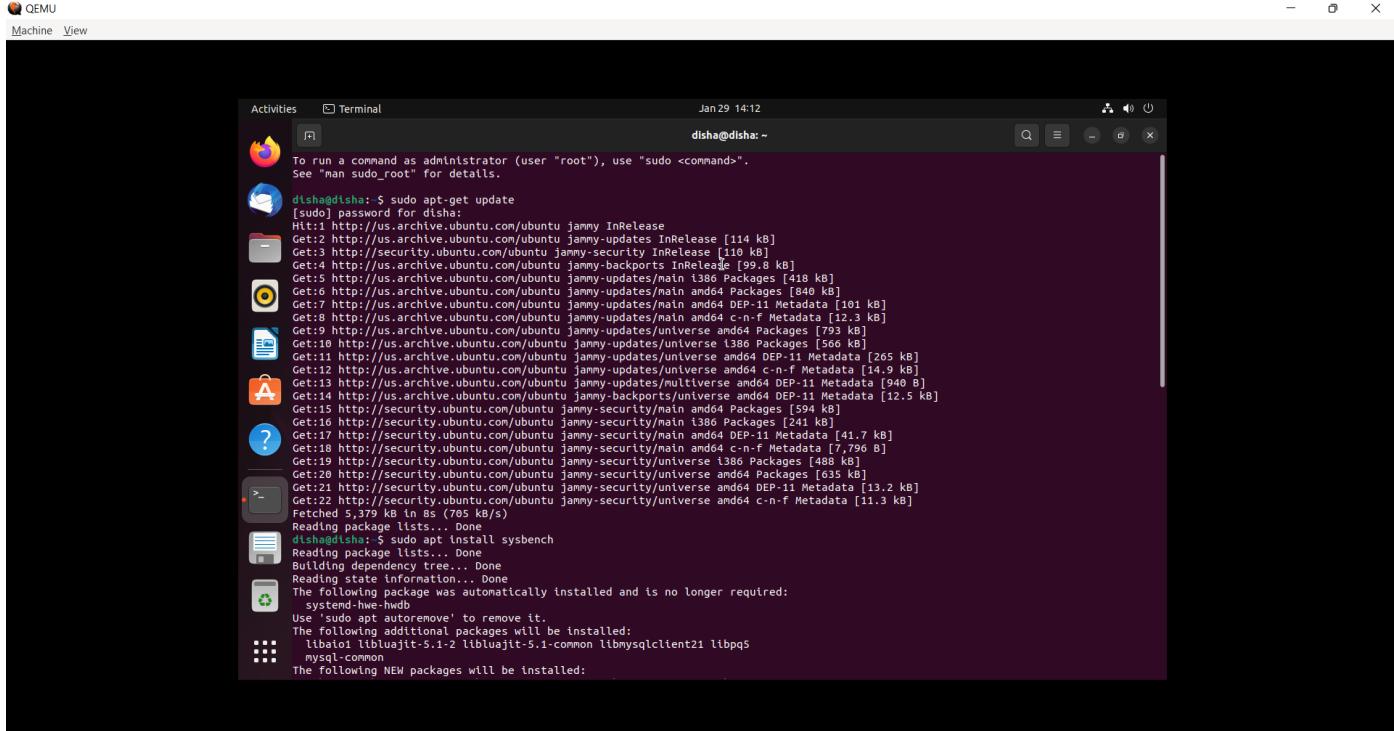
- d. The ubuntu now boots up and we do the normal installation as shown in the following images



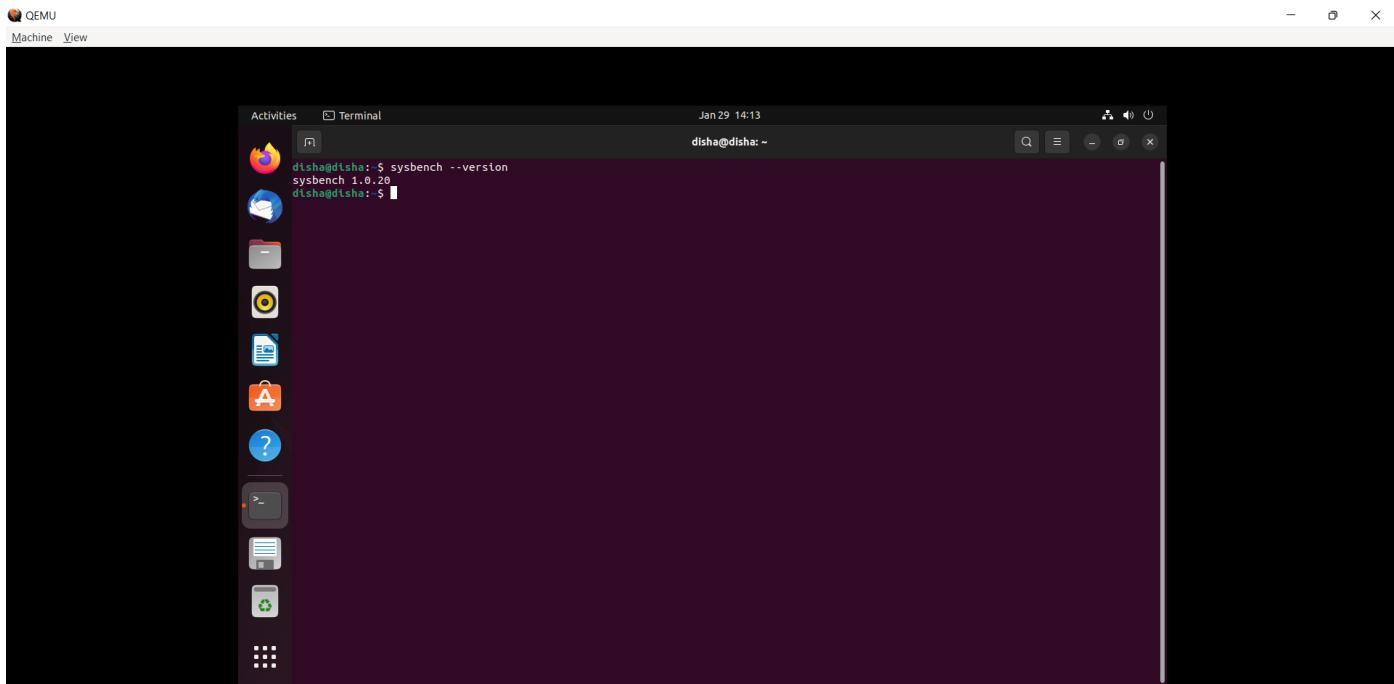
5. Following commands installs sysbench on QEMU

sudo apt-get update

sudo apt install sysbench



```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
disha@disha:~$ sudo apt-get update  
[sudo] password for disha:  
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [418 kB]  
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [840 kB]  
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]  
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [12.3 kB]  
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [793 kB]  
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [566 kB]  
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [265 kB]  
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [14.9 kB]  
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [948 kB]  
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.5 kB]  
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [154 kB]  
Get:16 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [214 kB]  
Get:17 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [41.7 kB]  
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [7,796 kB]  
Get:19 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [488 kB]  
Get:20 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [635 kB]  
Get:21 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [13.2 kB]  
Get:22 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [11.3 kB]  
Fetched 5,379 kB in 8s (705 kB/s)  
Reading package lists... Done  
disha@disha:~$ sudo apt install sysbench  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
    sysbench  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
    libaio1 libbluetooth-5.1-2 libbluetooth-5.1-common libmysqldclient21 libpq5  
    mysql-common  
The following NEW packages will be installed:
```



```
disha@disha:~$ sysbench --version  
sysbench 1.0.20  
disha@disha:~$
```

The above screenshot shows the installation of sysbench

Operating System Virtualization Setup

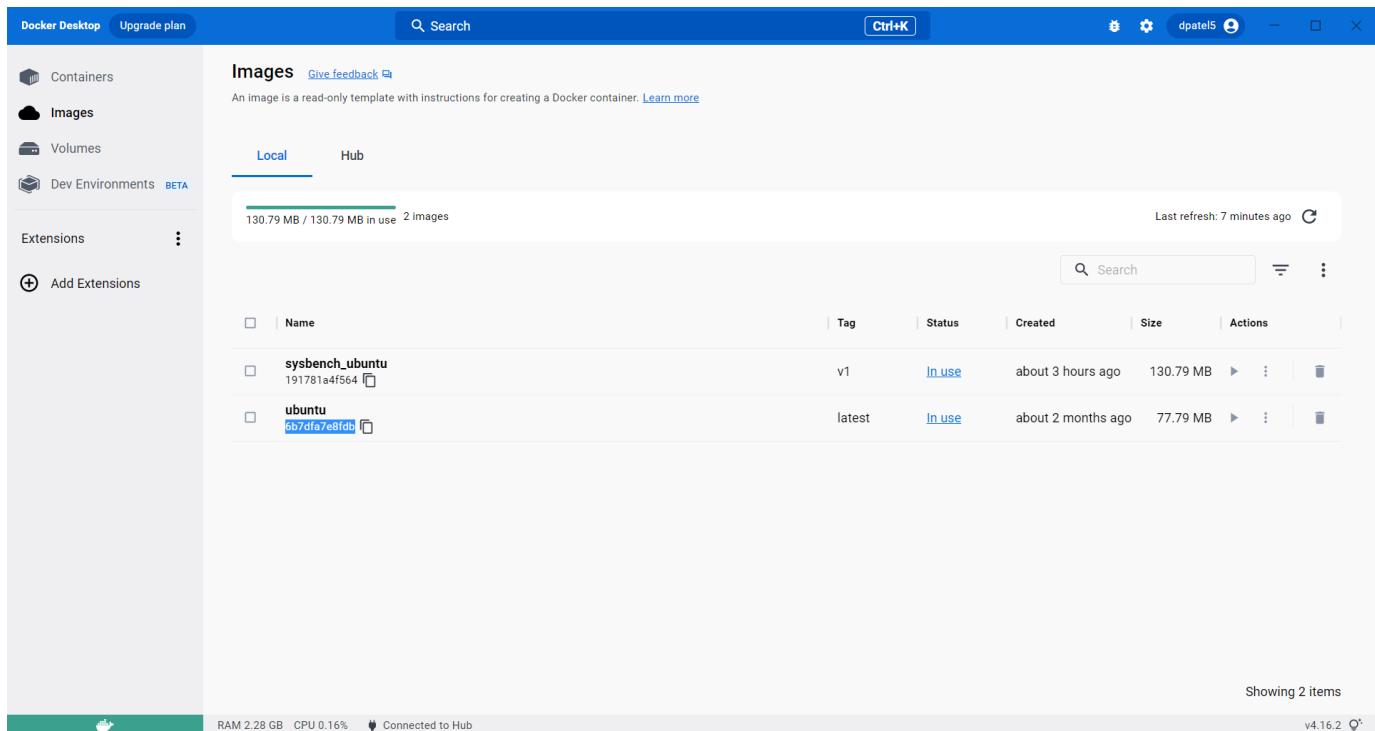
Docker Setup:

Create and start a new container based on Ubuntu image

```
docker run ubuntu
```

Start a bash shell in the Ubuntu container

```
docker run -it ubuntu bash
```

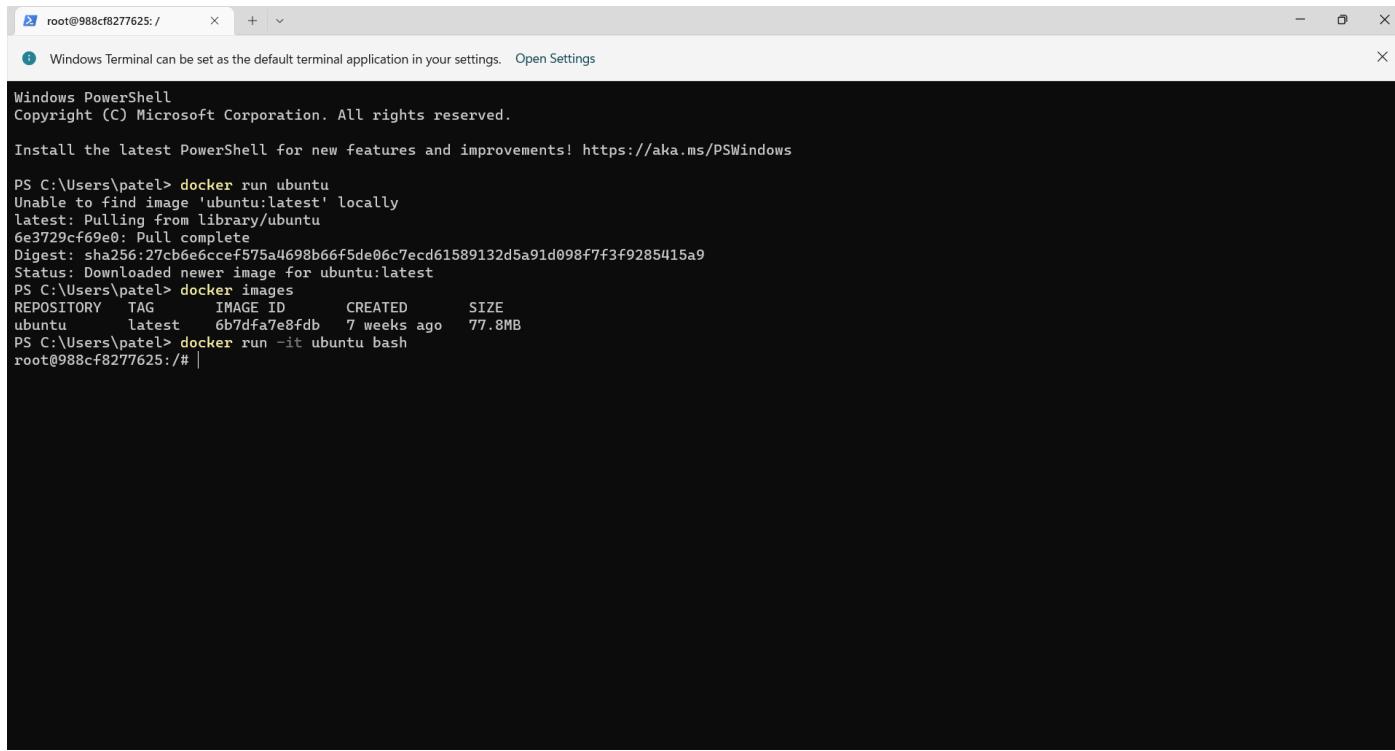


The screenshot shows the Docker Desktop interface. On the left, there's a sidebar with icons for Containers, Images (which is selected), Volumes, Dev Environments (Beta), and Extensions. The main area is titled 'Images' with a 'Local' tab selected. It displays two images: 'sysbench_ubuntu' (tag v1) and 'ubuntu' (tag latest). Both images are listed as 'In use'. The table has columns for Name, Tag, Status, Created, Size, and Actions. At the bottom, it says 'Showing 2 items'.

Name	Tag	Status	Created	Size	Actions
sysbench_ubuntu 191781a4f564	v1	In use	about 3 hours ago	130.79 MB	...
ubuntu 6b7dfa7e8fdb	latest	In use	about 2 months ago	77.79 MB	...

The above screenshot shows the image id highlighted of the Ubuntu image that we created and started the bash script.

Image ID: `6b7dfa7e8fdb`



The screenshot shows a Windows Terminal window with a single tab open. The title bar indicates it's a Windows Terminal window. The content area is a PowerShell session. The session starts with the standard PowerShell welcome message. Then, the user runs three commands: `docker run ubuntu`, `docker images`, and `docker run -it ubuntu bash`. The output of the first command shows that Docker is unable to find the 'ubuntu:latest' image locally and is pulling it from the library. The second command lists the available images, showing one named 'ubuntu' with a specific digest. The third command starts a new container and enters a bash shell. The session ends with the root prompt at the end of the command line.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patel> docker run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
6e3729cf69e0: Pull complete
Digest: sha256:27cb6e6cccef575a4698b66f5de06c7ecd61589132d5a91d098f7f3f9285415a9
Status: Downloaded newer image for ubuntu:latest
PS C:\Users\patel> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest 6b7dfa7e8fdb 7 weeks ago 77.8MB
PS C:\Users\patel> docker run -it ubuntu bash
root@988cf8277625:/# |
```

The above screenshot shows the execution of the commands mentioned above

Now, to get the sudo command working we run the following commands:

apt-get update

apt-get -y install sudo

Following command installs sysbench:

sudo apt install -y sysbench

```
root@988cf8277625:/# whoami
root
root@988cf8277625:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [681 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [745 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [795 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [730 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1062 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1002 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8978 B]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7286 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
Fetched 25.3 MB in 9s (2803 kB/s)
Reading package lists... Done
root@988cf8277625:/# apt-get -y install sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 820 kB of archives.
After this operation, 2564 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 sudo amd64 1.9.9-1lubuntu2.2 [820 kB]
Fetched 820 kB in 2s (527 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Preparing to unpack .../05-libluajit-5.1-common_2.1.0~beta3+dfsg-6_all.deb ...
Unpacking libluajit-5.1-common (2.1.0~beta3+dfsg-6) ...
```

```
Preparing to unpack .../05-libluajit-5.1-common_2.1.0~beta3+dfsg-6_all.deb ...
Unpacking libluajit-5.1-common (2.1.0~beta3+dfsg-6) ...
Selecting previously unselected package libluajit-5.1-2:amd64.
Preparing to unpack .../06-libluajit-5.1-2_2.1.0~beta3+dfsg-6_amd64.deb ...
Unpacking libluajit-5.1-2:amd64 (2.1.0~beta3+dfsg-6) ...
Selecting previously unselected package mysql-common.
Preparing to unpack .../07-mysql-common_5.8+1.0.8_all.deb ...
Unpacking mysql-common (5.8+1.0.8) ...
Selecting previously unselected package libmysqldclient21:amd64.
Preparing to unpack .../08-libmysqldclient21_8.0.32-0ubuntu0.22.04.1_amd64.deb ...
Unpacking libmysqldclient21:amd64 (8.0.32-0ubuntu0.22.04.1) ...
Selecting previously unselected package libpq5:amd64.
Preparing to unpack .../09-libpq5_14.6-0ubuntu0.22.04.1_amd64.deb ...
Unpacking libpq5:amd64 (14.6-0ubuntu0.22.04.1) ...
Selecting previously unselected package libsasl2-modules:amd64.
Preparing to unpack .../10-libsasl2-modules_2.1.27+dfsg2-3ubuntu1.1_amd64.deb ...
Unpacking libsasl2-modules:amd64 (2.1.27+dfsg2-3ubuntu1.1) ...
Selecting previously unselected package sysbench.
Preparing to unpack .../11-sysbench_1.0.20+ds-2_amd64.deb ...
Unpacking sysbench (1.0.20+ds-2) ...
Setting up mysql-common (5.8+1.0.8) ...
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up libmysqldclient21:amd64 (8.0.32-0ubuntu0.22.04.1) ...
Setting up libsasl2-modules:amd64 (2.1.27+dfsg2-3ubuntu1.1) ...
Setting up libldap-common (2.5.13+dfsg-0ubuntu0.22.04.1) ...
Setting up libsasl2-modules-db:amd64 (2.1.27+dfsg2-3ubuntu1.1) ...
Setting up libluajit-5.1-common (2.1.0~beta3+dfsg-6) ...
Setting up libsasl2-2:amd64 (2.1.27+dfsg2-3ubuntu1.1) ...
Setting up libai01:amd64 (0.3.112-13build1) ...
Setting up libluajit-5.1-2:amd64 (2.1.0~beta3+dfsg-6) ...
Setting up libldap-2.5-0:amd64 (2.5.13+dfsg-0ubuntu0.22.04.1) ...
Setting up libpq5:amd64 (14.6-0ubuntu0.22.04.1) ...
Setting up sysbench (1.0.20+ds-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
root@988cf8277625:/# sysbench --version
sysbench 1.0.20
root@988cf8277625:/# |
```

The above screenshot shows the installation of sysbench

Now, we save this image in order to use it in future test cases. To save the image out of this container we use the following command:

To get the container ID of last run container:

```
docker ps -a
```

```
exit
PS C:\Users\patel> docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED      STATUS          PORTS     NAMES
988cf8277625  ubuntu    "bash"    24 minutes ago  Exited (127) 6 seconds ago
7a0d2d467d19   ubuntu    "bash"    30 minutes ago  Exited (0) 30 minutes ago
PS C:\Users\patel> docker images
```

To create a new image out of this container with container ID = 988cf8277625:

```
docker commit 988cf8277625 sysbench_ubuntu:v1
```

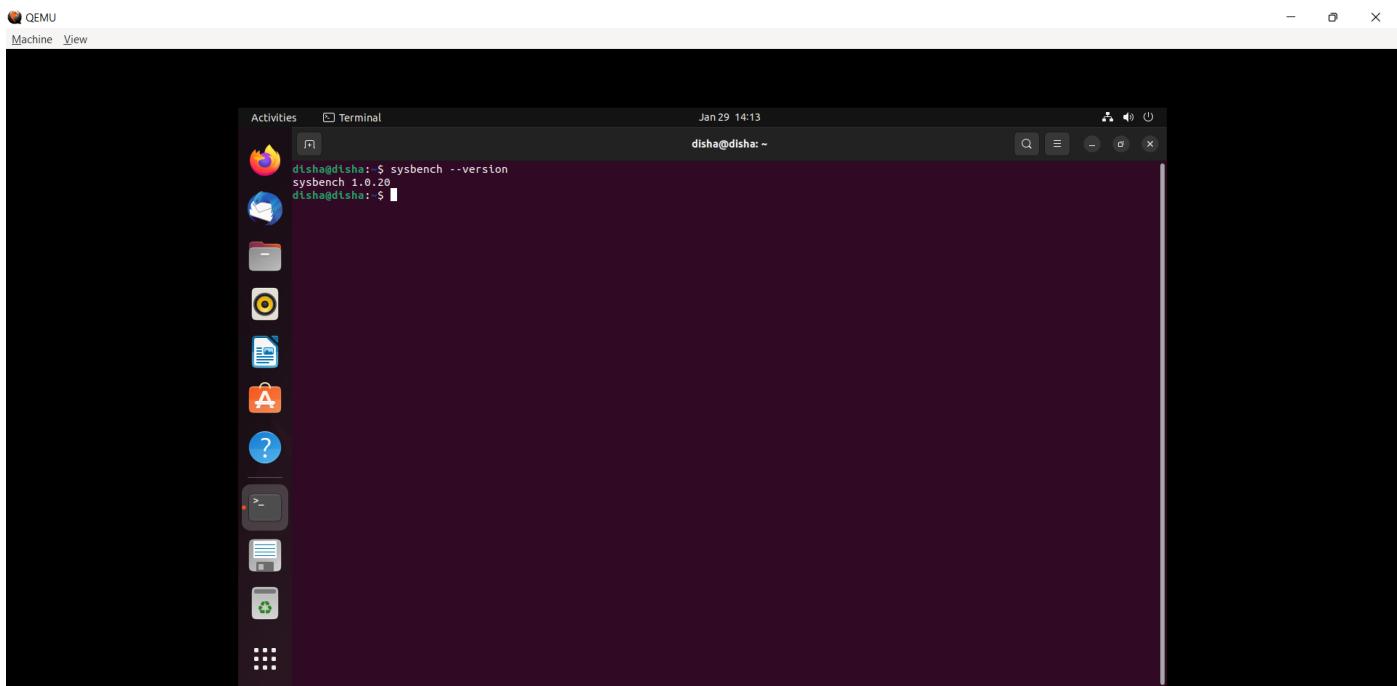
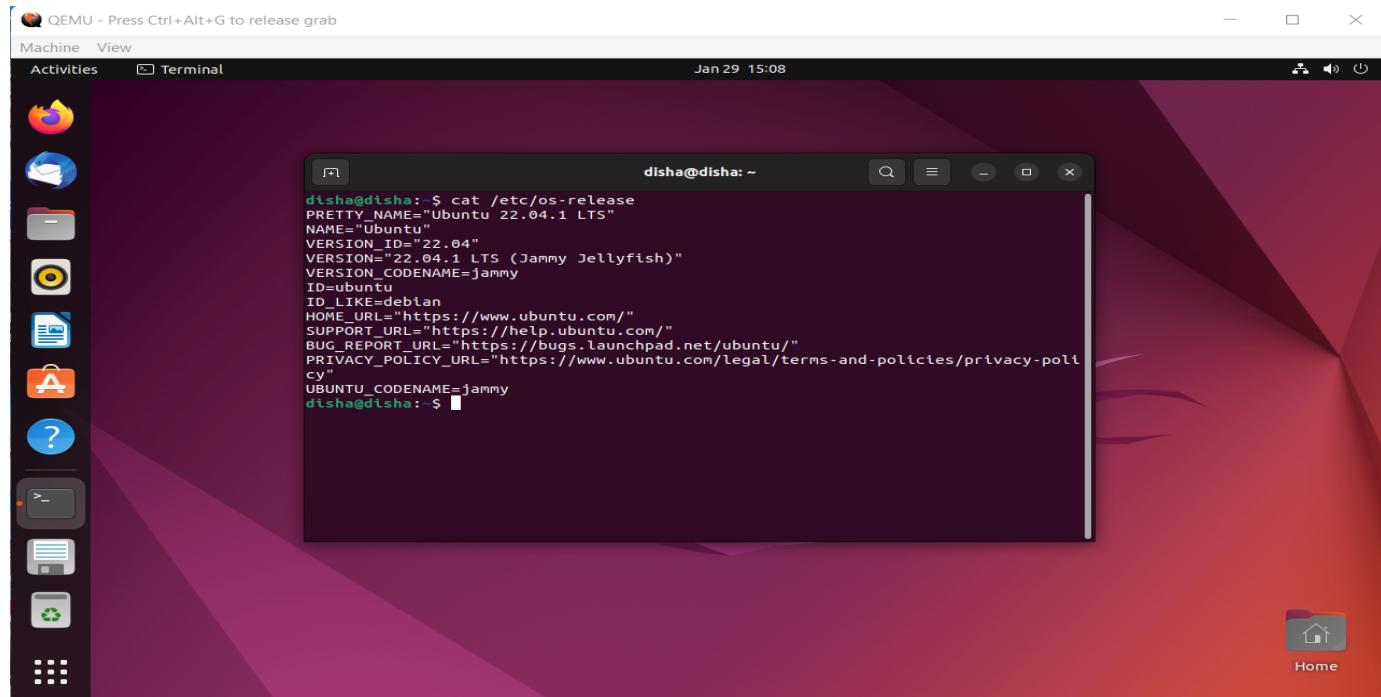
To check if the image creation was successful or not:

```
docker images
```

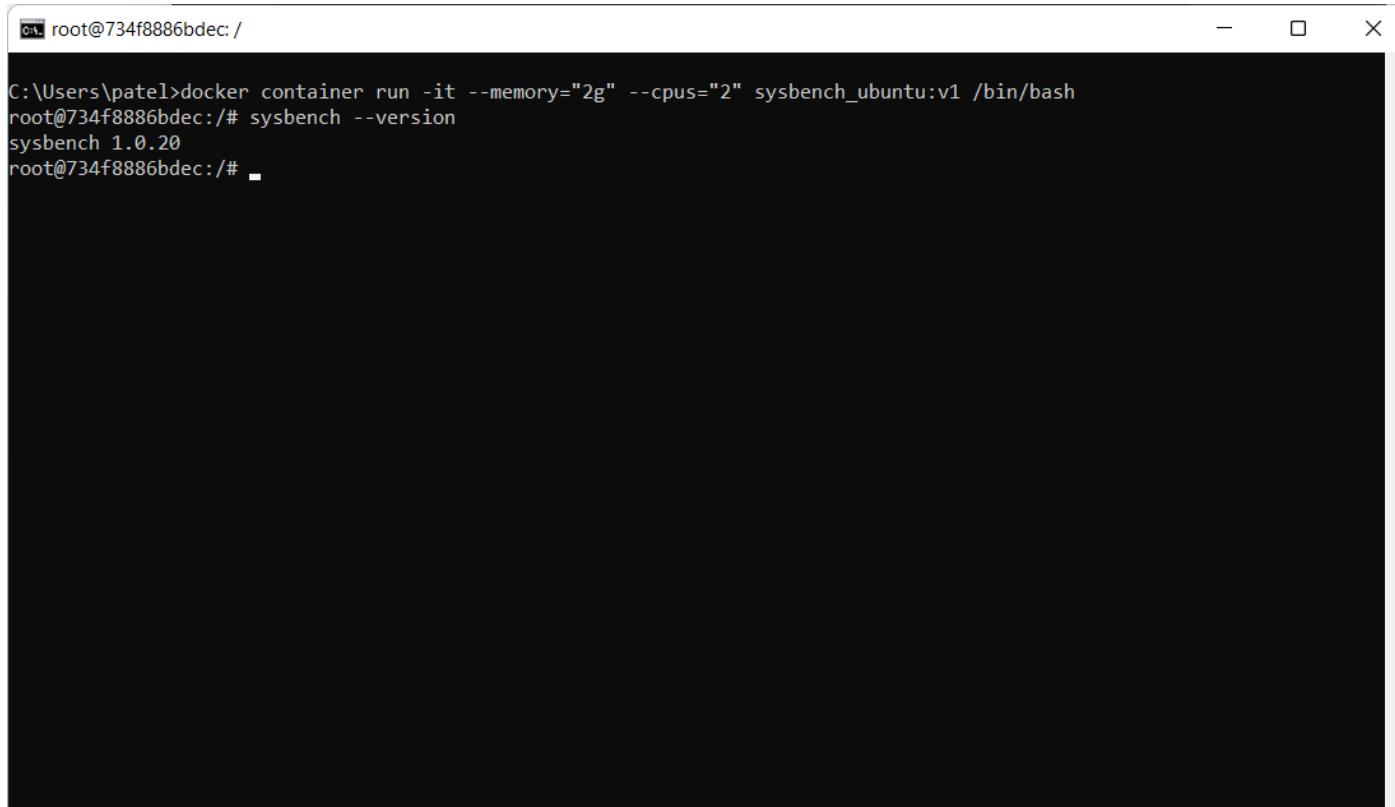
```
PS C:\Users\patel> docker commit 988cf8277625 sysbench_ubuntu:v1
sha256:191781a4f5646ca0e22b62fb2c6c0fae90c19767c32ec2d24425d380fd32132
PS C:\Users\patel> docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
sysbench_ubuntu  v1       191781a4f564  13 seconds ago  131MB
ubuntu          latest   6b7dfa7e8fdb  7 weeks ago   77.8MB
PS C:\Users\patel>
```

Proof of Experiment

QEMU Running Environment



Docker Running Environment



A screenshot of a terminal window titled "root@734f8886bdec: /". The window shows the following command-line session:

```
C:\Users\patel>docker container run -it --memory="2g" --cpus="2" sysbench_ubuntu:v1 /bin/bash
root@734f8886bdec:/# sysbench --version
sysbench 1.0.20
root@734f8886bdec:/# ■
```

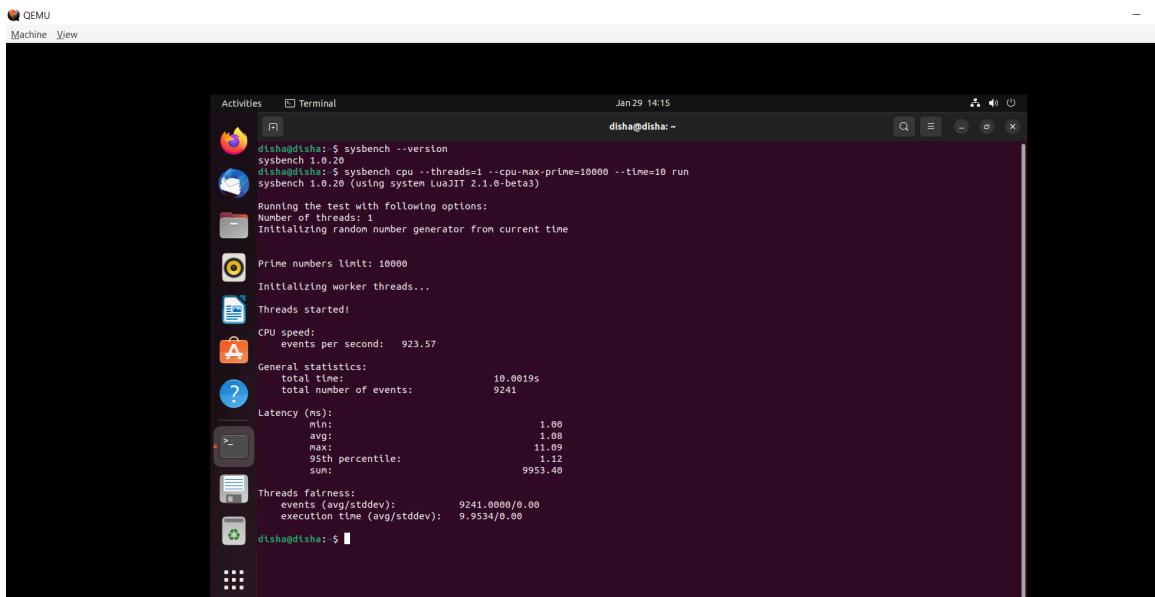
Experiment using sysbench

CPU Test:

1. CPU Test in QEMU VM

a. Test 1:

```
sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
```



The screenshot shows a terminal window in a QEMU VM. The terminal output is as follows:

```
disha@disha: ~$ sysbench --version
sysbench 1.0.20
disha@disha: ~$ sysbench cpu --threads=1 --cpu-max-prime=10000 --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: 10000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 923.57

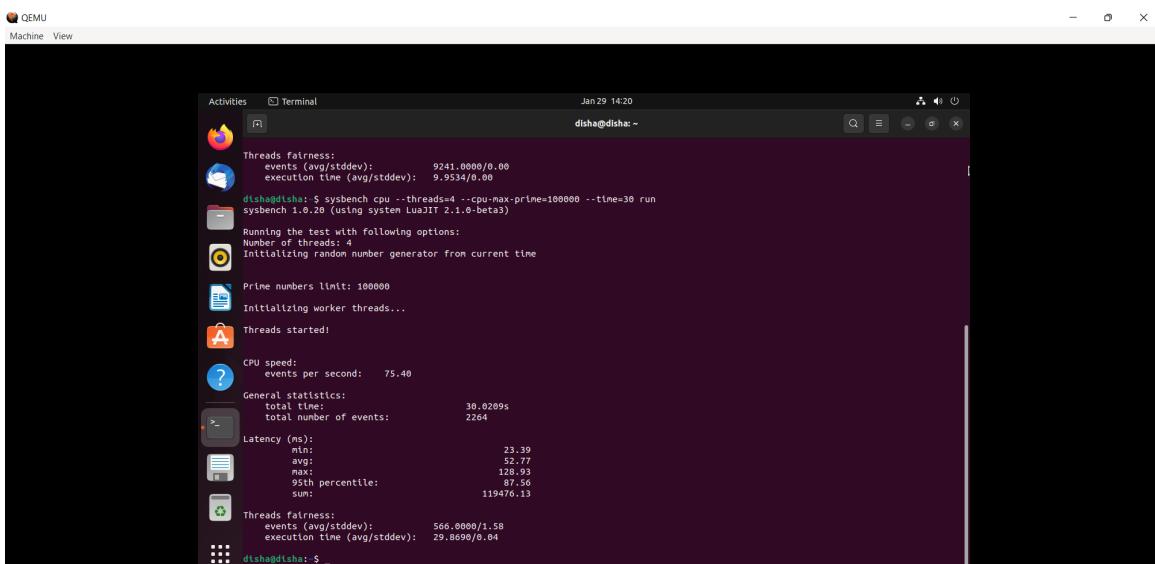
General statistics:
total time: 10.0019s
total number of events: 9241

Latency (ms):
min: 1.00
avg: 1.08
max: 1.09
95th percentile: 1.12
sum: 9953.40

Threads fairness:
events (avg/stddev): 9241.0000/0.00
execution time (avg/stddev): 9.9534/0.00
disha@disha: ~
```

b. Test 2:

```
sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run
```



The screenshot shows a terminal window in a QEMU VM. The terminal output is as follows:

```
disha@disha: ~$ sysbench --version
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Threads fairness:
events (avg/stddev): 9241.0000/0.00
execution time (avg/stddev): 9.9534/0.00

disha@disha: ~$ sysbench cpu --threads=4 --cpu-max-prime=100000 --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: 100000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 75.40

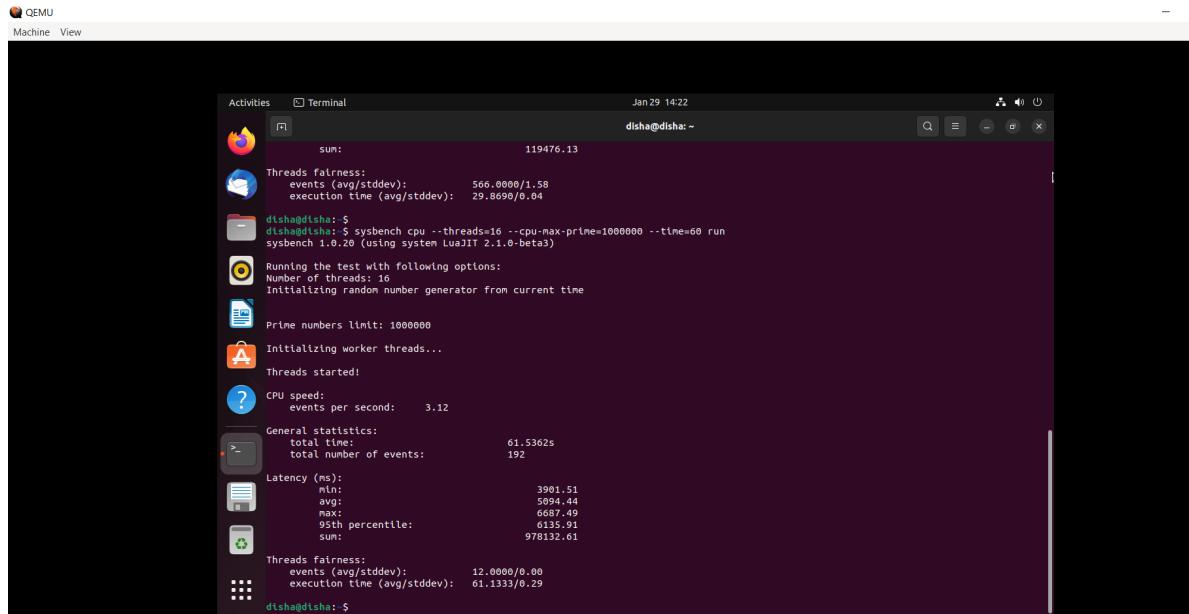
General statistics:
total time: 30.0209s
total number of events: 2264

Latency (ms):
min: 23.39
avg: 52.77
max: 128.93
95th percentile: 87.56
sum: 119476.13

Threads fairness:
events (avg/stddev): 566.0000/1.58
execution time (avg/stddev): 29.8690/0.04
disha@disha: ~
```

c. Test 3:

```
sysbench cpu --threads=16 --cpu-max-prime=1000000 --time=60 run
```



```
Activities Terminal Jan 29 14:22
disha@disha: ~
sum: 119476.13
Threads fairness:
events (avg/stddev): 566.0000/1.58
execution time (avg/stddev): 29.8690/0.84
disha@disha: $ sysbench cpu --threads=16 --cpu-max-prime=1000000 --time=60 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)
Running the test with following options:
Number of threads: 16
Initializing random number generator from current time
Prime numbers limit: 1000000
Initializing worker threads...
Threads started!
CPU speed:
events per second: 3.12
General statistics:
total time: 61.5362s
total number of events: 192
Latency (ms):
min: 3901.51
avg: 5994.44
max: 6687.49
95th percentile: 6135.91
sum: 978132.61
Threads fairness:
events (avg/stddev): 12.0000/0.00
execution time (avg/stddev): 61.1333/0.29
disha@disha: $
```

2. CPU Test in Docker Container

a. Test 1:

```
sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
```

```
root@f9e0c51fa7d2:/# sysbench cpu --threads=1 --cpu-max-prime=10000 --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: 10000
Initializing worker threads...

Threads started!

CPU speed:
events per second: 3847.83

General statistics:
total time: 10.0002s
total number of events: 38484

Latency (ms):
min: 0.25
avg: 0.26
max: 1.54
95th percentile: 0.28
sum: 9993.11

Threads fairness:
events (avg/stddev): 38484.0000/0.00
execution time (avg/stddev): 9.9931/0.00

root@f9e0c51fa7d2:/# |
```

b. Test 2:

```
sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run
```

```
root@f9e0c51fa7d2:/# sysbench cpu --threads=4 --cpu-max-prime=100000 --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: 100000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 472.72

General statistics:
total time: 30.0058s
total number of events: 14185

Latency (ms):
min: 6.13
avg: 8.46
max: 14.17
95th percentile: 12.30
sum: 120008.70

Threads fairness:
events (avg/stddev): 3546.2500/70.90
execution time (avg/stddev): 30.0022/0.00

root@f9e0c51fa7d2:/#
```

c. Test 3:

```
sysbench cpu --threads=16 --cpu-max-prime=1000000 --time=60 run
```

```
root@f9e0c51fa7d2:/# sysbench cpu --threads=16 --cpu-max-prime=1000000 --time=60 run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 16
Initializing random number generator from current time

Prime numbers limit: 1000000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 42.25

General statistics:
total time: 60.3256s
total number of events: 2549

Latency (ms):
min: 272.05
avg: 377.69
max: 532.82
95th percentile: 419.45
sum: 962719.82

Threads fairness:
events (avg/stddev): 159.3125/0.77
execution time (avg/stddev): 60.1700/0.09

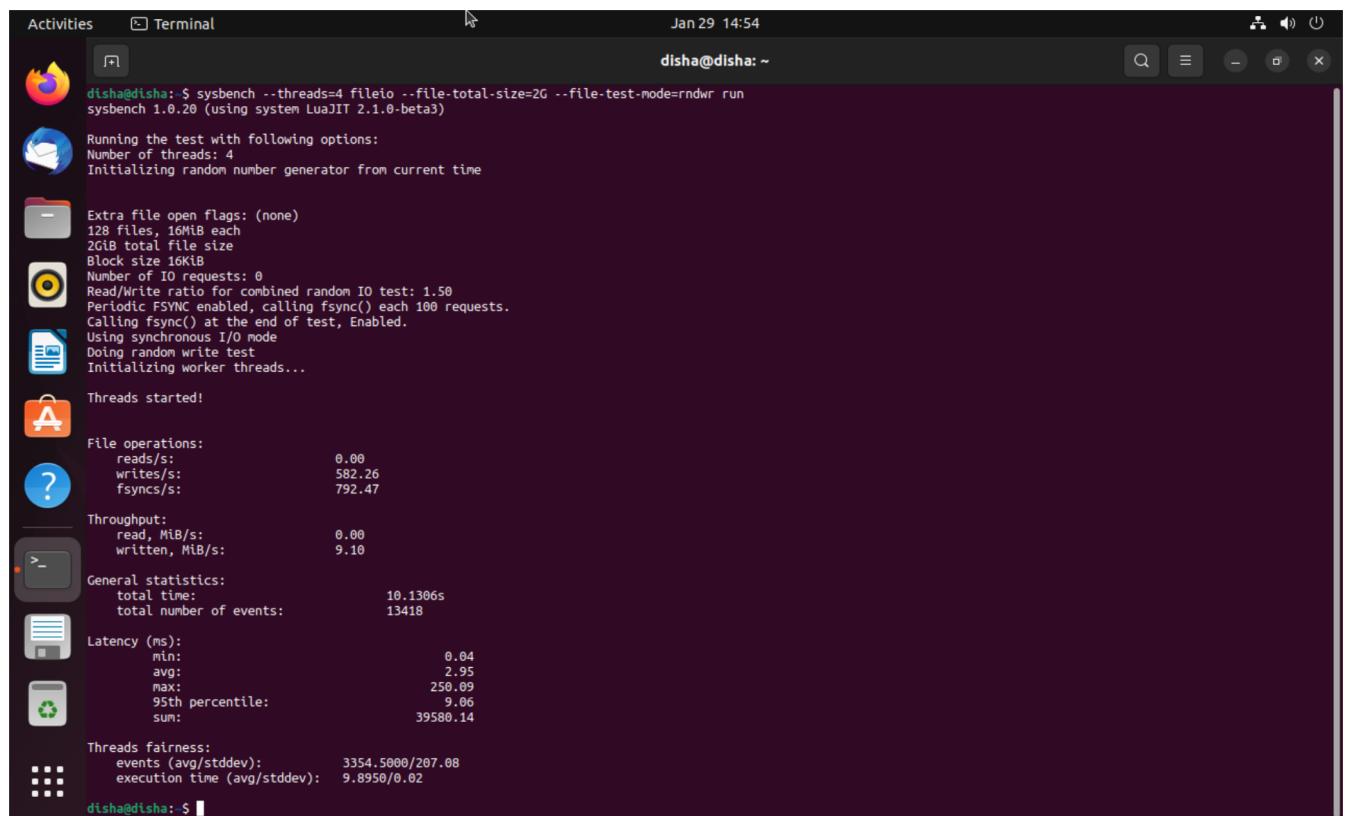
root@f9e0c51fa7d2:/#
```

File IO Test:

1. File IO Test in QEMU VM

a. Test 1:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup
```



```
disha@disha: ~$ sysbench --threads=4 fileio --file-total-size=2G --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, 16MiB each
2GiB 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:        582.26
  fsyncs/s:       792.47

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   9.10

General statistics:
  total time:           10.1306s
  total number of events: 13418

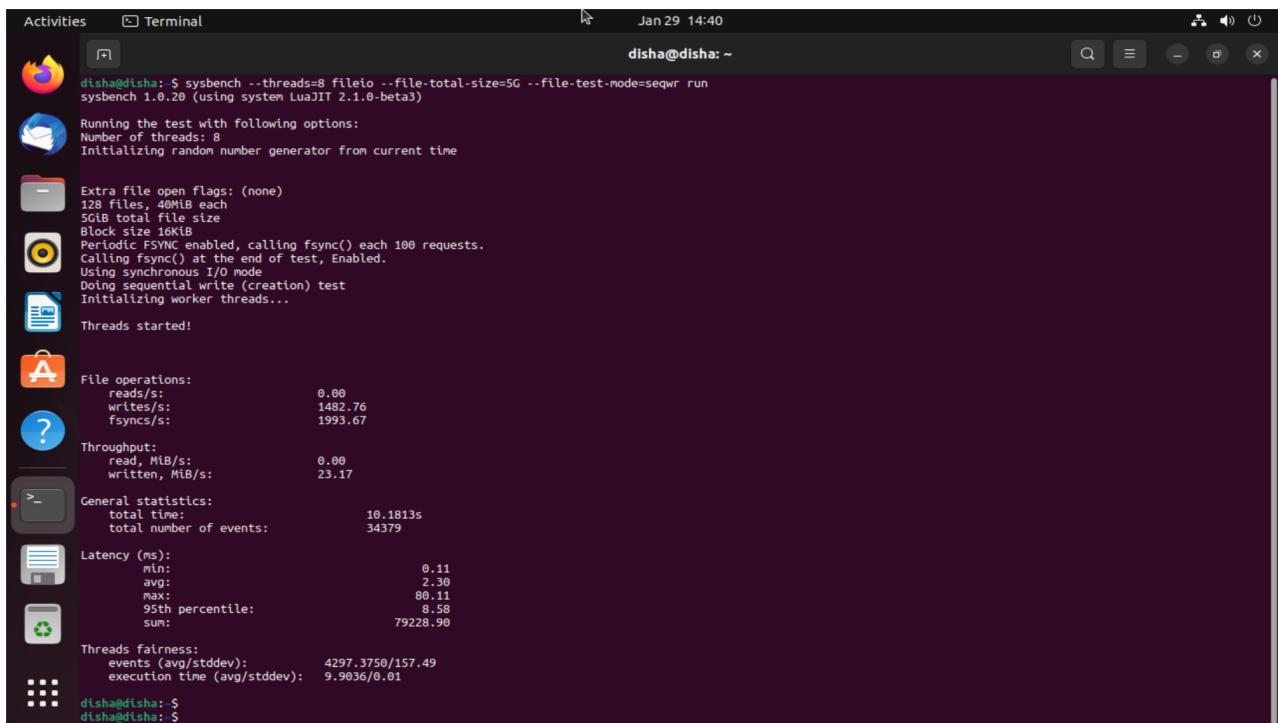
Latency (ms):
  min:                 0.04
  avg:                 2.95
  max:                250.09
  95th percentile:     9.06
  sum:            39580.14

Threads fairness:
  events (avg/stddev): 3354.5000/207.08
  execution time (avg/stddev): 9.8950/0.02

disha@disha: ~$
```

b. Test 2:

```
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare  
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run  
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup
```

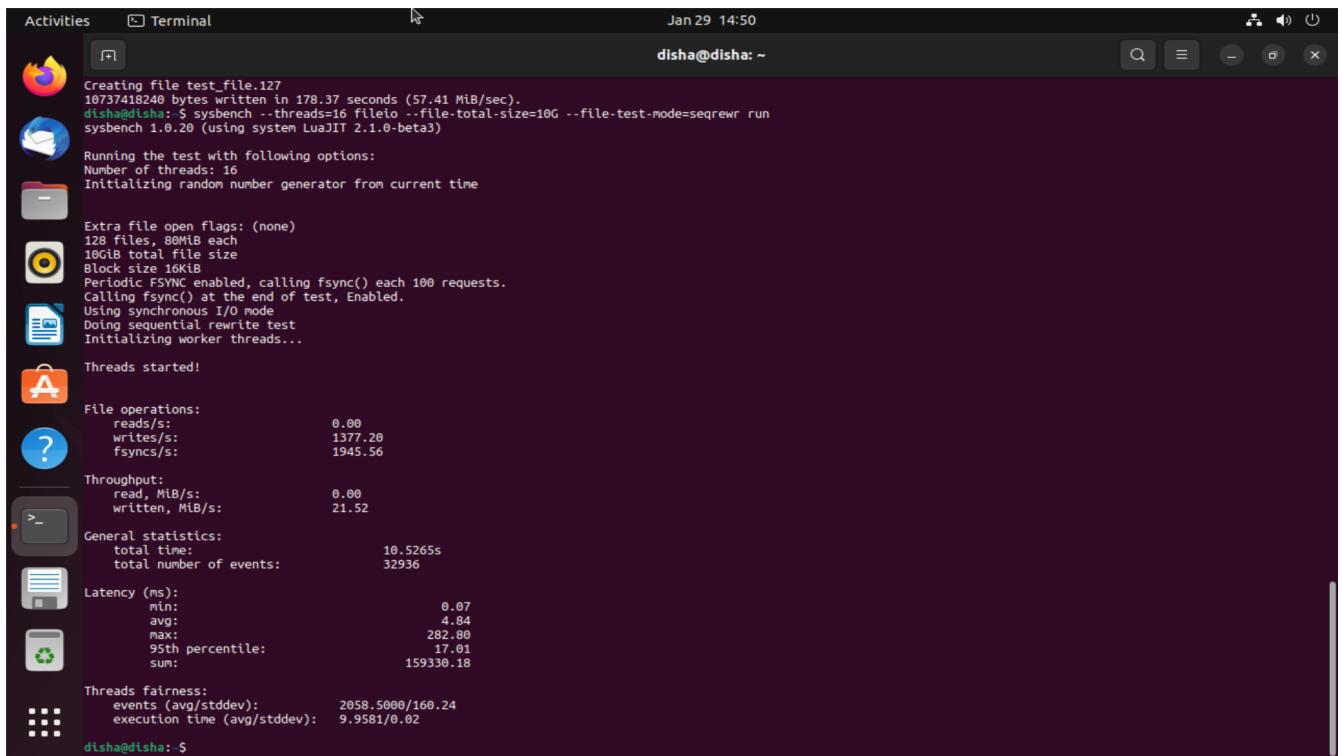


The screenshot shows a terminal window titled "disha@disha: ~" running on a Linux system. The window displays the output of a sysbench fileio test with 8 threads, 5GB total size, and sequential write mode. The output includes configuration details like file open flags, block size, and fsync settings, followed by throughput and general statistics. The throughput section shows reads/s, writes/s, and fsyncs/s. General statistics include total time and number of events. Latency statistics provide min, avg, max, and 95th percentile values. Thread fairness metrics show events and execution time per thread.

```
disha@disha: $ sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr 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, 40MB each  
5GB 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: 1482.76  
fsyncs/s: 1993.67  
  
Throughput:  
read, MiB/s: 0.00  
written, MiB/s: 23.17  
  
General statistics:  
total_time: 10.1813s  
total_number_of_events: 34379  
  
Latency (ms):  
min: 0.11  
avg: 2.30  
max: 80.11  
95th_percentile: 8.58  
sum: 79228.96  
  
Threads fairness:  
events (avg/stddev): 4297.3750/157.49  
execution_time (avg/stddev): 9.9036/0.01  
disha@disha: $  
disha@disha: $
```

c. Test 3:

```
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr prepare  
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr run  
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr cleanup
```



The screenshot shows a terminal window titled "Terminal" with the command "disha@disha: ~". The output of the sysbench test is displayed:

```
Creating file test_file.127  
10737418240 bytes written in 178.37 seconds (57.41 MiB/sec).  
disha@disha: $ sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 16  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 8MiB each  
10GiB 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 rewrite test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
  reads/s:          0.00  
  writes/s:        1377.20  
  fsyncs/s:       1945.56  
  
Throughput:  
  read, MiB/s:      0.00  
  written, MiB/s:   21.52  
  
General statistics:  
  total time:           10.5265s  
  total number of events: 32936  
  
Latency (ms):  
  min:                 0.07  
  avg:                4.84  
  max:               282.80  
  95th percentile:    17.01  
  sum:            159330.18  
  
Threads fairness:  
  events (avg/stddev): 2058.5000/160.24  
  execution time (avg/stddev): 9.9581/0.02  
disha@disha: $
```

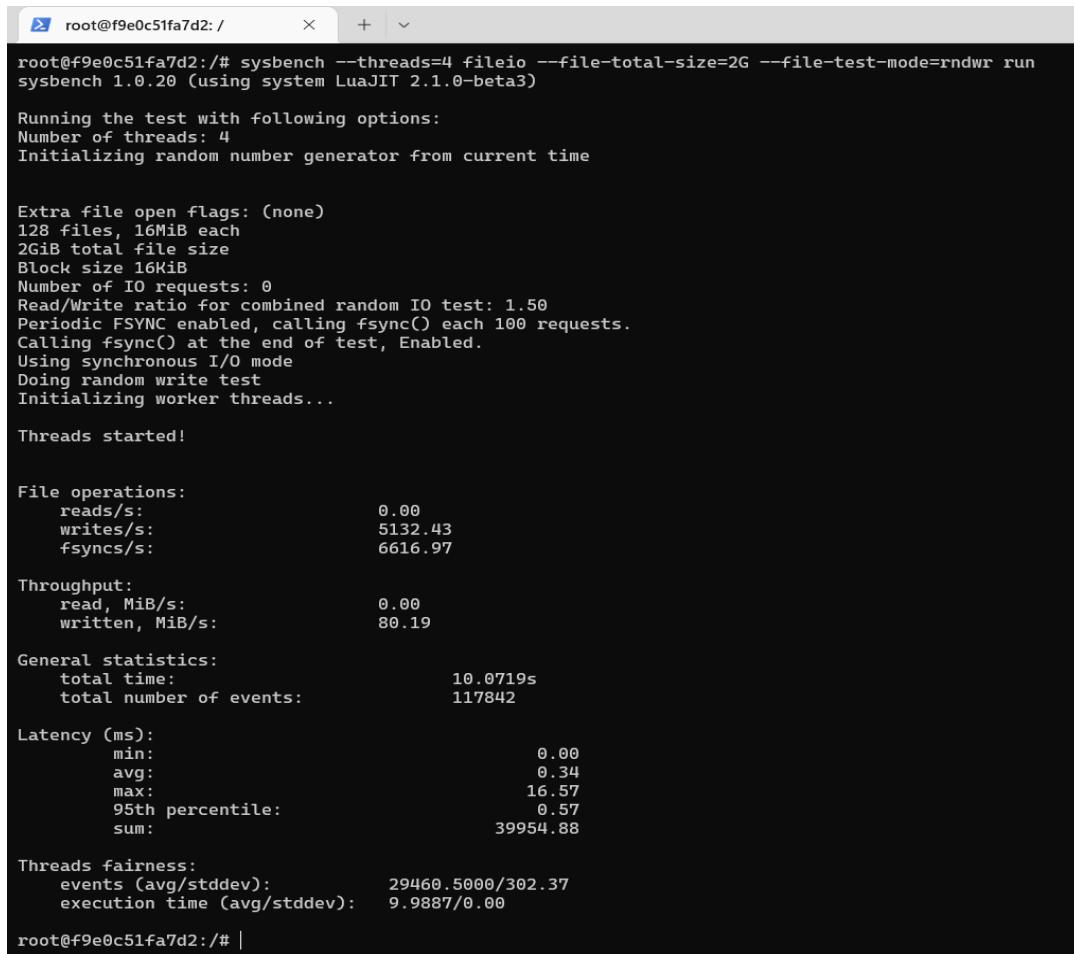
2. File IO Test in Docker Container

a. Test 1:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare
```

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run
```

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup
```



The screenshot shows a terminal window with the following output:

```
root@f9e0c51fa7d2:/# sysbench --threads=4 fileio --file-total-size=2G --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, 16MiB each
2GiB 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:        5132.43
  fsyncs/s:        6616.97

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   80.19

General statistics:
  total time:       10.0719s
  total number of events: 117842

Latency (ms):
  min:              0.00
  avg:             0.34
  max:             16.57
  95th percentile: 0.57
  sum:            39954.88

Threads fairness:
  events (avg/stddev): 29460.5000/302.37
  execution time (avg/stddev): 9.9887/0.00

root@f9e0c51fa7d2:/# |
```

b. Test 2:

```
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr prepare
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr cleanup
```

```
root@f9e0c51fa7d2:/# sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr 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, 40MiB each
5GiB 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:         3762.98
  fsyncs/s:         4915.08

Throughput:
  read, MiB/s:      0.00
  written, MiB/s:   58.80

General statistics:
  total time:        10.1237s
  total number of events: 86841

Latency (ms):
  min:                0.00
  avg:                0.92
  max:               974.37
  95th percentile:    2.48
  sum:              79957.83

Threads fairness:
  events (avg/stddev): 10855.1250/555.74
  execution time (avg/stddev): 9.9947/0.00

root@f9e0c51fa7d2:/#
```

c. Test 3:

```
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr prepare  
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr run  
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr cleanup
```

```
root@f9e0c51fa7d2:/# sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr run  
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)  
  
Running the test with following options:  
Number of threads: 16  
Initializing random number generator from current time  
  
Extra file open flags: (none)  
128 files, 80MiB each  
10GiB 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 rewrite test  
Initializing worker threads...  
  
Threads started!  
  
File operations:  
    reads/s:          0.00  
    writes/s:        5229.95  
    fsyncs/s:       6883.15  
  
Throughput:  
    read, MiB/s:      0.00  
    written, MiB/s:   81.72  
  
General statistics:  
    total time:           10.2821s  
    total number of events: 122519  
  
Latency (ms):  
    min:                  0.00  
    avg:                 1.31  
    max:                 29.34  
    95th percentile:     3.55  
    sum:                159951.40  
  
Threads fairness:  
    events (avg/stddev): 7657.4375/649.70  
    execution time (avg/stddev): 9.9970/0.00  
  
root@f9e0c51fa7d2:/# |
```

Analysis of Results

CPU Test:

Following parameters are considered while performing the cpu test

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

The experiment is done by altering the values of the above parameters and running each test case 5 times to obtain the average, min, max and std. values of the results.

Test 1:

```
sysbench cpu --threads=1 --cpu-max-prime=10000 --time=10 run
```

Platform	min	avg	max	total no.of events
QEMU	1.00	1.08	11.09	9241
Docker	0.25	0.26	1.54	38484

Test 2:

```
sysbench cpu --threads=4 --cpu-max-prime=100000 --time=30 run
```

Platform	min	avg	max	total no.of events
QEMU	23.39	52.77	128.93	2264
Docker	6.13	8.46	14.17	14185

Test 3:

```
sysbench cpu --threads=16 --cpu-max-prime=1000000 --time=60 run
```

Platform	min	avg	max	total no.of events
QEMU	3901.51	5094.44	6687.49	192
Docker	272.05	377.69	532.82	2549

FileIO Test:

Following parameters are considered to perform FileIO Test

- a. --file-total-size: This parameter sets the total size of the test file in megabytes.
- file-test-mode: This parameter sets the type of file I/O test to run (e.g. "seqwr", "seqrewr").
- b. --threads: This parameter sets the total number of threads used for the process.

The experiment is done by altering the values of the above parameters and running each test case 5 times to obtain the average, min, max and std. values of the results.

Test 1:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run
```

Platform	min	avg	max	total no.of events
QEMU	0.04	2.95	250.09	13418
Docker	0.0	0.34	16.57	117842

Test 2:

```
sysbench --threads=8 fileio --file-total-size=5G --file-test-mode=seqwr run
```

Platform	min	avg	max	total no.of events
QEMU	0.11	2.30	80.11	34379
Docker	0.0	0.92	974.37	86841

Test 3:

```
sysbench --threads=16 fileio --file-total-size=10G --file-test-mode=seqrewr run
```

Platform	min	avg	max	total no.of events
QEMU	0.07	4.84	282.80	32936
Docker	0.0	1.31	29.34	122519

Conclusion of CPU & File IO Tests

1. The docker container was determined to be quicker than the QEMU VM in all three of the CPU test cases.
2. The number of events for CPU tests declines as the number of threads rises.
3. For CPU tests, the latency numbers, min, max, and average, grow in tandem with the number of threads and time allotted to complete a case.
4. QEMU VM seemed to perform well in case of File IO test cases.
5. In case of File IO tests, QEMU VM took less number of overall events as compared to Docker Container.
6. Given that they have the same resources, such as CPU and RAM, docker containers perform better than QEMU VMs.

Shell Scripts of the Experiment

1. Shell Script: CPU Tests:

```
#!/bin/bash

echo "CPU TEST MODE"

PRIMES=("10000" "100000" "1000000")
MAX_TIME=("10" "30" "60")
THREADS=("1" "4" "16")

for ((i=1; i<=3; i++))
do
    echo "~ Starting ${i} Test Case ~"
    for ((j=1; j <=5; j++ ))
    do
        echo "Run: ${j} of Test Case ${i}"
        sysbench cpu --threads=${THREADS[$i]} --cpu-max-prime=${PRIMES[$i]}
--time=${MAX_TIME[$i]} run
        echo "Completed run:${j} of Test Case ${i}"
    done
    echo "~ Completed ${i} Test Case ~"
done
```

2. Shell Script: File IO Tests:

```
#! /bin/bash

echo "FILE TEST MODE"

TOTAL_FILE_SIZE=("2G" "5G" "10G")
MODES=("rndwr" "seqwr" "seqrewr")
THREADS=("4" "8" "16")

for ((i=1; i<=3; i++))
do
    echo "~ Starting ${i}st Test Case ~"
    for (( j=1; j <=5; j++ ))
    do
        echo "Run: ${j} of Test Case ${i}"
        sysbench --threads=${THREADS[$i]} fileio
--file-total-size=${TOTAL_FILE_SIZE[$i]} --file-test-mode=${MODES[i]} prepare
        sysbench --threads=${THREADS[$i]} fileio
--file-total-size=${TOTAL_FILE_SIZE[$i]} --file-test-mode=${MODES[i]} run
        sysbench --threads=${THREADS[$i]} fileio
--file-total-size=${TOTAL_FILE_SIZE[$i]} --file-test-mode=${MODES[i]} cleanup
        echo "Completed run:${j} of Test Case ${i}"
    done
    echo "~ Completed ${i}st Test Case ~"
done
```

System Performance Tools and Analysis

CPU Utilization:

We may view the CPU utilization in User Mode, System Mode, and Idle Mode with the 'top' command in Linux.

1. QEMU

a. QEMU VM running sysbench cpu test

```
top - 18:59:45 up 20 min, 1 user, load average: 0.67, 1.36, 1.23
Tasks: 173 total, 1 running, 172 sleeping, 0 stopped, 0 zombie
%CPU(s): 47.6 us, 5.5 sy, 0.0 ni, 46.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
M1B Mem : 7950.0 total, 6108.6 free, 709.5 used, 1131.9 buff/cache
M1B Swap: 2048.0 total, 2048.0 free, 0.0 used. 6954.2 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
1519 disha    20   0 4499652 344248 136156 S 103.3  4.2 8:29.11 gnome-shell
2526 disha    20   0 569148 58652 45208 S  4.6  0.7 0:14.24 gnome-terminal-
3099 disha    20   0 21760 3996 3228 R  1.0  0.0 0:00.72 top
368 systemd+  20   0 14824 6012 5212 S  0.7  0.1 0:07.55 systemd-oomd
17 root     20   0     0     0     0 I  0.3  0.0 0:00.72 kworker/0:1-mm_percpu_wq
23 root     20   0     0     0     0 I  0.3  0.0 0:00.95 kworker/1:0-events
101 root    20   0     0     0     0 I  0.3  0.0 0:00.74 kworker/u4:3-events_unbound
1 root     20   0 166560 11580 8116 S  0.0  0.1 0:08.27 systemd
2 root     20   0     0     0     0 S  0.0  0.0 0:00.01 kthreadd
3 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 rCU_gp
4 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 rCU_par_gp
5 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 slub_flushwq
6 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 netns
8 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 kworker/0:0H-events_highpri
10 root    0 -20     0     0     0 I  0.0  0.0 0:00.0 mm_percpu_wq
11 root    20   0     0     0     0 S  0.0  0.0 0:00.0 rCU_tasks_rude_
12 root    20   0     0     0     0 S  0.0  0.0 0:00.0 rCU_tasks_trace
13 root    20   0     0     0     0 S  0.0  0.0 0:01.04 ksoftirqd/0
14 root    20   0     0     0     0 I  0.0  0.0 0:03.64 rCU_sched
15 root    rt  0     0     0     0 S  0.0  0.0 0:00.0 migration/0
16 root    -51  0     0     0     0 S  0.0  0.0 0:00.0 idle_inject/0
18 root    20   0     0     0     0 S  0.0  0.0 0:00.0 cpuhp/0
19 root    20   0     0     0     0 S  0.0  0.0 0:00.0 cpuhp/1
20 root    -51  0     0     0     0 S  0.0  0.0 0:00.0 idle_inject/1
21 root    rt  0     0     0     0 S  0.0  0.0 0:00.31 migration/1
22 root    20   0     0     0     0 S  0.0  0.0 0:00.64 ksoftirqd/1
24 root    0 -20     0     0     0 I  0.0  0.0 0:00.0 kworker/1:0H-events_highpri
25 root    20   0     0     0     0 S  0.0  0.0 0:00.0 kdevtmpfs
26 root    0 -20     0     0     0 I  0.0  0.0 0:00.0 inet_frag_wq
27 root    20   0     0     0     0 S  0.0  0.0 0:00.0 kaudild
29 root    20   0     0     0     0 S  0.0  0.0 0:00.0 khungtaskd
30 root    20   0     0     0     0 S  0.0  0.0 0:00.0 oom_reaper
```

b. QEMU VM NOT running sysbench cpu test

```
top - 18:59:03 up 19 min, 1 user, load average: 0.80, 1.51, 1.27
Tasks: 175 total, 1 running, 174 sleeping, 0 stopped, 0 zombie
%CPU(s): 7.5 us, 0.8 sy, 0.0 ni, 91.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
M1B Mem : 7950.0 total, 6108.0 free, 709.3 used, 1131.9 buff/cache
M1B Swap: 2048.0 total, 2048.0 free, 0.0 used. 6954.4 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
1519 disha    20   0 4499636 344360 136156 S 14.1  4.2 8:08.93 gnome-shell
3099 disha    20   0 21760 3996 3228 R  1.3  0.0 0:00.21 top
368 systemd+  20   0 14824 6012 5212 S  0.7  0.1 0:07.32 systemd-oomd
2526 disha    20   0 569148 58652 45208 S  0.7  0.7 0:13.66 gnome-terminal-
14 root     20   0     0     0     0 I  0.3  0.0 0:00.61 rCU_sched
23 root     20   0     0     0     0 I  0.3  0.0 0:00.92 kworker/1:0-events
1 root     20   0 166560 11580 8116 S  0.0  0.1 0:08.27 systemd
2 root     20   0     0     0     0 S  0.0  0.0 0:00.01 kthreadd
3 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 rCU_gp
4 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 rCU_par_gp
5 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 slub_flushwq
6 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 netns
8 root     0 -20     0     0     0 I  0.0  0.0 0:00.0 kworker/0:0H-events_highpri
9 root    20   0     0     0     0 I  0.0  0.0 0:02.91 kworker/u4:0-ext4-rsv-conversion
10 root    0 -20     0     0     0 I  0.0  0.0 0:00.0 mm_percpu_wq
11 root    20   0     0     0     0 S  0.0  0.0 0:00.0 rCU_tasks_rude_
12 root    20   0     0     0     0 S  0.0  0.0 0:00.0 rCU_tasks_trace
13 root    20   0     0     0     0 S  0.0  0.0 0:01.04 ksoftirqd/0
15 root    rt  0     0     0     0 S  0.0  0.0 0:00.04 migration/0
16 root    -51  0     0     0     0 S  0.0  0.0 0:00.00 idle_inject/0
17 root    20   0     0     0     0 I  0.0  0.0 0:00.69 kworker/0:1-events
18 root    20   0     0     0     0 S  0.0  0.0 0:00.00 cpuhp/0
19 root    20   0     0     0     0 S  0.0  0.0 0:00.00 cpuhp/1
20 root    -51  0     0     0     0 S  0.0  0.0 0:00.00 idle_inject/1
21 root    rt  0     0     0     0 S  0.0  0.0 0:00.31 migration/1
22 root    20   0     0     0     0 S  0.0  0.0 0:00.64 ksoftirqd/1
24 root    0 -20     0     0     0 I  0.0  0.0 0:00.00 kworker/1:0H-events_highpri
25 root    20   0     0     0     0 S  0.0  0.0 0:00.00 kdevtmpfs
26 root    0 -20     0     0     0 I  0.0  0.0 0:00.00 inet_frag_wq
27 root    20   0     0     0     0 S  0.0  0.0 0:00.00 kaudild
29 root    20   0     0     0     0 S  0.0  0.0 0:00.00 khungtaskd
30 root    20   0     0     0     0 S  0.0  0.0 0:00.00 oom_reaper
```

2. Docker Container

a. Docker container running sysbench cpu test

```
top - 19:08:47 up 13:44, 0 users, load average: 4.50, 1.66, 1.00
Tasks: 2 total, 1 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.7 us, 0.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7796.3 total, 2253.0 free, 1690.9 used, 3852.5 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 5821.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	4620	3864	3316	S	0.0	0.0	0:00.03	bash
9	root	20	0	7308	3512	2952	R	0.0	0.0	0:00.02	top

b. Docker container NOT running sysbench cpu test

```
top - 22:41:19 up 2:48, 0 users, load average: 1.49, 0.68, 0.37
Tasks: 2 total, 1 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.1 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7796.3 total, 5388.6 free, 1343.5 used, 1064.2 buff/cache
MiB Swap: 2048.0 total, 2047.7 free, 0.3 used. 6207.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	4620	3804	3236	S	0.0	0.0	0:00.02	bash
37	root	20	0	7300	3400	2864	R	0.0	0.0	0:00.02	top

Observations from the above cases:

1. Docker Container CPU

Parameter	Running Sysbench CPU test	Not Running Sysbench CPU test
User level CPU Usage %	99.7	0.0
Kernel level CPU Usage %	0.3	0.1
Idle CPU %	0.0	99.9

2. QEMU VM

Parameter	Running Sysbench CPU test	Not Running Sysbench CPU test
User level CPU Usage %	47.6	7.5
Kernel level CPU Usage %	5.5	0.8
Idle CPU %	46.9	91.7

Conclusion:

1. In the case of a docker container, running sysbench cpu test increases the user and kernel level CPU percentage as compared to the docker container not running sysbench cpu test. The idle time also decreases when the sysbench cpu test is run inside the docker container.
2. In the case of QEMU VM, running sysbench cpu test increases the user and kernel level CPU percentage as compared to not running sysbench cpu test. The idle time also decreases when the sysbench cpu test is run inside the QEMU VM.
3. The increase in the usage of User level CPU is more significant as compared to the usage of Kernel level CPU while running the sysbench cpu test inside the QEMU VM. This shows that the QEMU VM doesn't use the Kernel level CPU of the host OS.

IO Utilization:

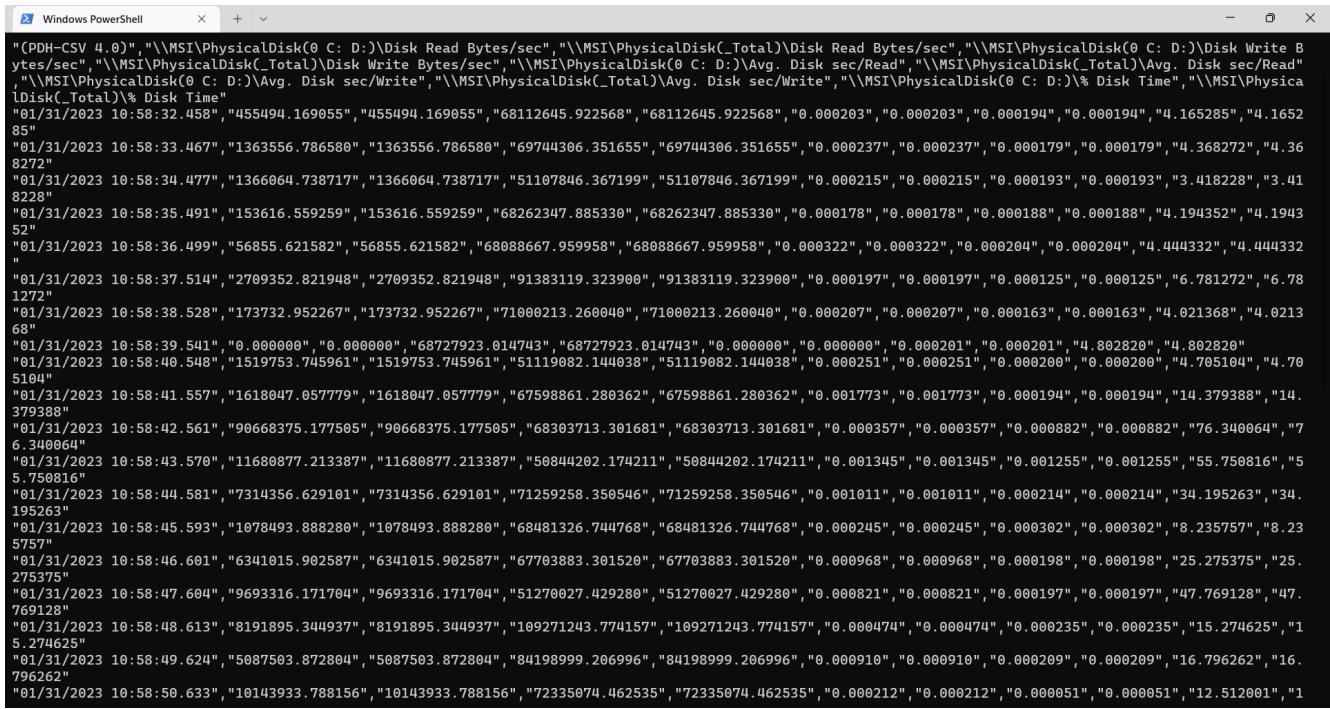
We may view the real time disk utilization using the following command of Windows:

```
typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec"  
"\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk  
Time"
```

1. QEMU VM

a. Prepare Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare
```



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The command entered was "sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare". The output is a large block of CSV data, starting with the header "(PDH-CSV 4.0)", followed by numerous rows of disk performance metrics. The data includes columns for Disk Read Bytes/sec, Disk Write Bytes/sec, Avg. Disk sec/Read, Avg. Disk sec/Write, and % Disk Time, all recorded at 1-second intervals from January 31, 2023, at 10:58:32.458 to 10:58:50.633.

(PDH-CSV 4.0)	\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec	\PhysicalDisk(_Total)\Disk Read Bytes/sec	\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec	\PhysicalDisk(_Total)\Disk Write Bytes/sec	\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read	\PhysicalDisk(_Total)\Avg. Disk sec/Read	\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write	\PhysicalDisk(_Total)\Avg. Disk sec/Write	\PhysicalDisk(0 C: D:)% Disk Time	\PhysicalDisk(_Total)% Disk Time
"01/31/2023 10:58:32.458"	"455494.169055"	"455494.169055"	"68112645.922568"	"68112645.922568"	"0.000203"	"0.000203"	"0.000194"	"0.000194"	"4.165285"	"4.165285"
"01/31/2023 10:58:33.458"	"455494.169055"	"455494.169055"	"68112645.922568"	"68112645.922568"	"0.000203"	"0.000203"	"0.000194"	"0.000194"	"4.165285"	"4.165285"
"01/31/2023 10:58:33.467"	"1363556.786580"	"1363556.786580"	"69744306.351655"	"69744306.351655"	"0.000237"	"0.000237"	"0.000179"	"0.000179"	"4.368272"	"4.368272"
"01/31/2023 10:58:34.477"	"1366064.738717"	"1366064.738717"	"51107846.367199"	"51107846.367199"	"0.000215"	"0.000215"	"0.000193"	"0.000193"	"3.418228"	"3.418228"
"01/31/2023 10:58:34.828"	"153616.559259"	"153616.559259"	"68262347.885330"	"68262347.885330"	"0.000178"	"0.000178"	"0.000188"	"0.000188"	"4.194352"	"4.194352"
"01/31/2023 10:58:35.491"	"153616.559259"	"153616.559259"	"68262347.885330"	"68262347.885330"	"0.000178"	"0.000178"	"0.000188"	"0.000188"	"4.194352"	"4.194352"
"01/31/2023 10:58:36.499"	"56855.621582"	"56855.621582"	"68088667.959958"	"68088667.959958"	"0.000322"	"0.000322"	"0.000204"	"0.000204"	"4.444332"	"4.444332"
"01/31/2023 10:58:37.514"	"2709352.821948"	"2709352.821948"	"91383119.323900"	"91383119.323900"	"0.000197"	"0.000197"	"0.000125"	"0.000125"	"6.781272"	"6.781272"
"01/31/2023 10:58:38.528"	"173732.952267"	"173732.952267"	"71000213.260040"	"71000213.260040"	"0.000207"	"0.000207"	"0.000163"	"0.000163"	"4.021368"	"4.021368"
"01/31/2023 10:58:39.541"	"0.000000"	"0.000000"	"68727923.014743"	"68727923.014743"	"0.000000"	"0.000000"	"0.000201"	"0.000201"	"4.802820"	"4.802820"
"01/31/2023 10:58:40.548"	"1519753.745961"	"1519753.745961"	"51119082.144038"	"51119082.144038"	"0.000251"	"0.000251"	"0.000200"	"0.000200"	"4.765104"	"4.765104"
"01/31/2023 10:58:41.557"	"1618047.057779"	"1618047.057779"	"67598861.280362"	"67598861.280362"	"0.001773"	"0.001773"	"0.000194"	"0.000194"	"14.379388"	"14.379388"
"01/31/2023 10:58:42.561"	"90668375.177505"	"90668375.177505"	"68303713.301681"	"68303713.301681"	"0.000357"	"0.000357"	"0.000882"	"0.000882"	"76.340064"	"76.340064"
"01/31/2023 10:58:43.570"	"11680877.213387"	"11680877.213387"	"50844202.174211"	"50844202.174211"	"0.001345"	"0.001345"	"0.001255"	"0.001255"	"55.750816"	"55.750816"
"01/31/2023 10:58:44.581"	"7314356.629101"	"7314356.629101"	"71259258.350546"	"71259258.350546"	"0.001011"	"0.001011"	"0.000214"	"0.000214"	"34.195263"	"34.195263"
"01/31/2023 10:58:45.593"	"1078493.888280"	"1078493.888280"	"68481326.744768"	"68481326.744768"	"0.000245"	"0.000245"	"0.000302"	"0.000302"	"8.235757"	"8.235757"
"01/31/2023 10:58:46.601"	"6341015.902587"	"6341015.902587"	"67703883.301520"	"67703883.301520"	"0.000968"	"0.000968"	"0.000198"	"0.000198"	"25.275375"	"25.275375"
"01/31/2023 10:58:47.604"	"9693316.171704"	"9693316.171704"	"51270027.429280"	"51270027.429280"	"0.000821"	"0.000821"	"0.000197"	"0.000197"	"47.769128"	"47.769128"
"01/31/2023 10:58:48.613"	"8191895.344937"	"8191895.344937"	"109271243.774157"	"109271243.774157"	"0.000474"	"0.000474"	"0.000235"	"0.000235"	"15.274625"	"15.274625"
"01/31/2023 10:58:49.624"	"5087503.872804"	"5087503.872804"	"84198999.206996"	"84198999.206996"	"0.000910"	"0.000910"	"0.000209"	"0.000209"	"16.796262"	"16.796262"
"01/31/2023 10:58:50.633"	"10143933.788156"	"10143933.788156"	"72335074.462535"	"72335074.462535"	"0.000212"	"0.000212"	"0.000051"	"0.000051"	"12.512001"	"12.512001"

b. Run Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run
```

```
Windows PowerShell
```

```
PS C:\Users\patel> typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec" "\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk Time"

"(PDH-CSV 4.0)", "\MSI\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(0 C: D:)% Disk Time", "\MSI\PhysicalDisk(_Total)% Disk Time"
"01/31/2023 11:01:34.727", "1276275.801937", "11969192.471236", "11969192.471236", "0.000268", "0.000268", "0.000023", "0.000023", "3.254926", "3.254926"
"01/31/2023 11:01:35.729", "466329.890168", "466329.890168", "12071399.174427", "12071399.174427", "0.000300", "0.000300", "0.000023", "2.494446", "2.494446"
"01/31/2023 11:01:36.732", "673614.982352", "673614.982352", "11271824.038020", "11271824.038020", "0.000276", "0.000276", "0.000027", "2.883976", "2.883976"
"01/31/2023 11:01:37.742", "567859.643285", "567859.643285", "9896982.354403", "9896982.354403", "0.000411", "0.000411", "0.000024", "0.000024", "2.427346", "2.427346"
"01/31/2023 11:01:38.750", "247935.639299", "247935.639299", "11982201.059911", "11982201.059911", "0.000193", "0.000193", "0.000054", "0.000054", "5.349751", "5.349751"
"01/31/2023 11:01:39.756", "0.000000", "0.000000", "12547765.601138", "12547765.601138", "0.000000", "0.000000", "0.000028", "0.000028", "2.780449", "2.780449"
"01/31/2023 11:01:40.762", "215976.541522", "215976.541522", "12652965.309922", "12652965.309922", "0.000282", "0.000282", "0.000025", "0.000025", "2.816125", "2.816125"
"01/31/2023 11:01:41.768", "1155805.471711", "1155805.471711", "12526652.260299", "12526652.260299", "0.000715", "0.000715", "0.000025", "0.000025", "2.776611", "2.776611"
"01/31/2023 11:01:42.777", "4245126.925789", "4245126.925789", "5982138.708043", "5982138.708043", "0.000234", "0.000234", "0.000022", "0.000022", "2.828397", "2.828397"
"01/31/2023 11:01:43.782", "3430103.293010", "3430103.293010", "1124356.898896", "1124356.898896", "0.000376", "0.000376", "0.000050", "0.000050", "2.714524", "2.714524"
"24"
```

c. Cleanup Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup
```

```
Windows PowerShell
```

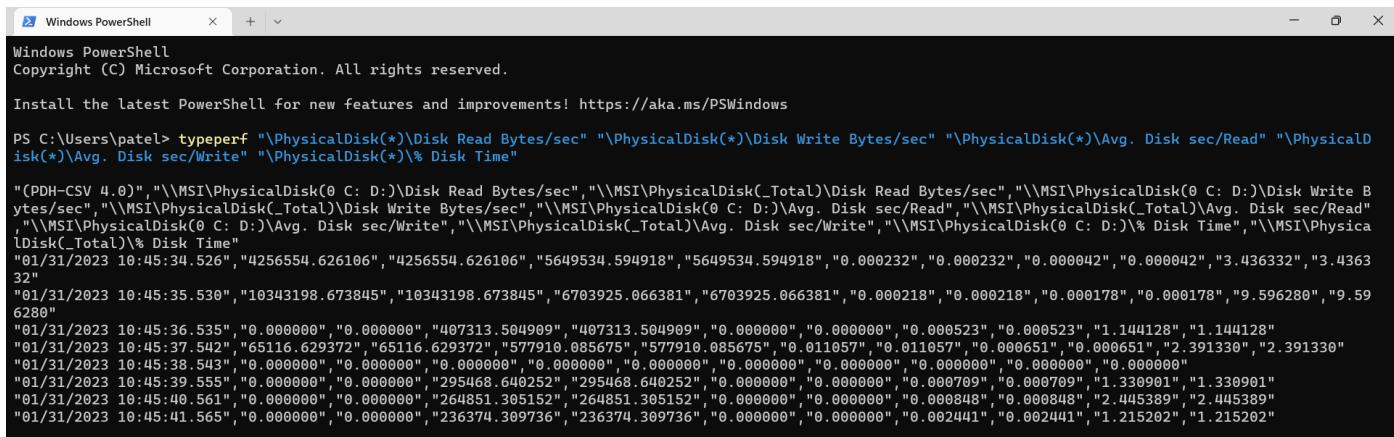
```
PS C:\Users\patel> typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec" "\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk Time"

"(PDH-CSV 4.0)", "\MSI\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(0 C: D:)% Disk Time", "\MSI\PhysicalDisk(_Total)% Disk Time"
"01/31/2023 11:03:10.015", "433900.066322", "433900.066322", "0.000000", "0.000000", "0.002072", "0.002072", "0.000000", "0.000000", "1.435806", "1.435806"
"01/31/2023 11:03:11.019", "983717.679436", "983717.679436", "216336.253154", "216336.253154", "0.000957", "0.000957", "0.000939", "0.000939", "2.835950", "2.835950"
"01/31/2023 11:03:12.023", "248592.157471", "248592.157471", "199688.782231", "199688.782231", "0.002910", "0.002910", "0.000829", "0.000829", "2.313833", "2.313833"
"01/31/2023 11:03:13.037", "258763.079236", "258763.079236", "109165.674053", "109165.674053", "0.003093", "0.003093", "0.002576", "0.002576", "3.507279", "3.507279"
"01/31/2023 11:03:14.051", "351236.278724", "351236.278724", "355273.477330", "355273.477330", "0.002051", "0.002051", "0.000114", "0.000114", "1.437561", "1.437561"
```

2. Docker Container

a. Prepare Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr prepare
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

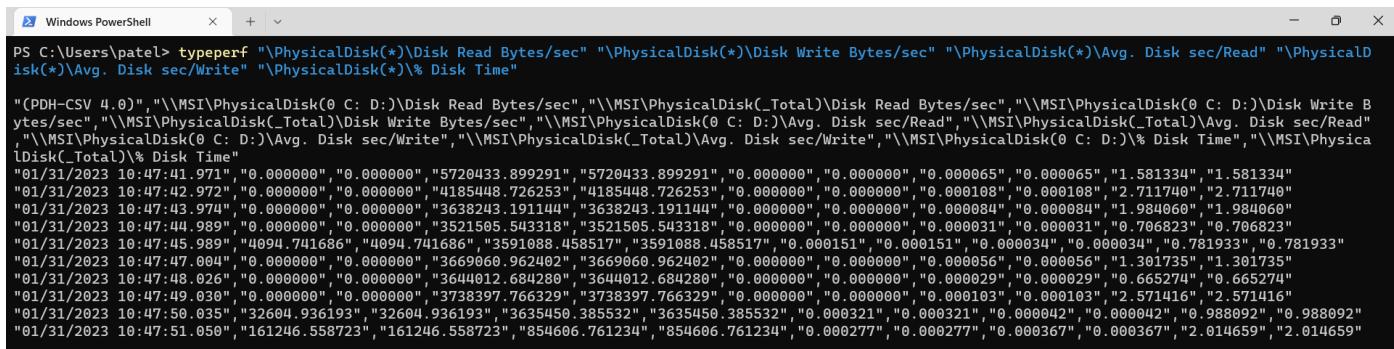
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\patel> typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec" "\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk Time"

"(PDH-CSV 4.0)", "\MSI\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(0 C: D:)% Disk Time", "\MSI\PhysicalDisk(_Total)% Disk Time"
"01/31/2023 10:45:34.526", "4256554.626106", "4256554.626106", "5649534.594918", "5649534.594918", "0.000232", "0.000232", "0.000042", "0.000042", "3.436332", "3.436332"
"01/31/2023 10:45:35.530", "10343198.673845", "10343198.673845", "6703925.066381", "6703925.066381", "0.000218", "0.000218", "0.000178", "0.000178", "9.596280", "9.596280"
"01/31/2023 10:45:36.535", "0.000000", "0.000000", "407313.504909", "407313.504909", "0.000000", "0.000000", "0.000523", "0.000523", "1.144128", "1.144128"
"01/31/2023 10:45:37.542", "65116.629372", "577910.085675", "577910.085675", "0.011057", "0.011057", "0.000651", "0.000651", "2.391330", "2.391330"
"01/31/2023 10:45:38.542", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000"
"01/31/2023 10:45:39.555", "0.000000", "0.000000", "295468.648252", "295468.648252", "0.000000", "0.000000", "0.000709", "0.000709", "1.330901", "1.330901"
"01/31/2023 10:45:40.561", "0.000000", "0.000000", "264851.305152", "264851.305152", "0.000000", "0.000000", "0.000848", "0.000848", "2.445389", "2.445389"
"01/31/2023 10:45:41.565", "0.000000", "0.000000", "236374.309736", "236374.309736", "0.000000", "0.000000", "0.002441", "0.002441", "1.215202", "1.215202"
```

b. Run Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr run
```



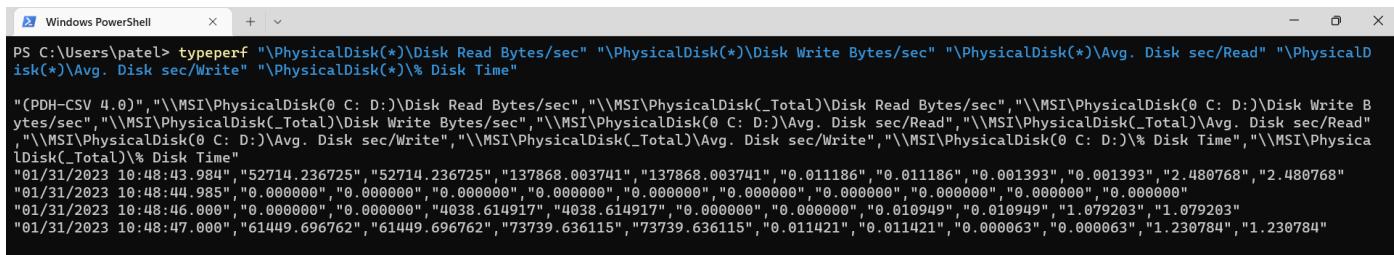
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\patel> typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec" "\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk Time"

"(PDH-CSV 4.0)", "\MSI\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(0 C: D:)% Disk Time", "\MSI\PhysicalDisk(_Total)% Disk Time"
"01/31/2023 10:47:41.971", "0.000000", "5720433.899291", "5720433.899291", "0.000000", "0.000000", "0.000065", "0.000065", "1.581334", "1.581334"
"01/31/2023 10:47:42.972", "0.000000", "4185448.726253", "4185448.726253", "0.000000", "0.000000", "0.000108", "0.000108", "2.711740", "2.711740"
"01/31/2023 10:47:43.974", "0.000000", "3638243.191144", "3638243.191144", "0.000000", "0.000000", "0.000084", "0.000084", "1.984060", "1.984060"
"01/31/2023 10:47:44.982", "0.000000", "3521505.543318", "3521505.543318", "0.000000", "0.000000", "0.000031", "0.000031", "0.706823", "0.706823"
"01/31/2023 10:47:45.989", "4094.741686", "3591088.458517", "3591088.458517", "0.000151", "0.000151", "0.000034", "0.000034", "0.781933", "0.781933"
"01/31/2023 10:47:47.004", "0.000000", "3669060.962402", "3669060.962402", "0.000000", "0.000000", "0.000056", "0.000056", "1.301735", "1.301735"
"01/31/2023 10:47:48.026", "0.000000", "3644012.684280", "3644012.684280", "0.000000", "0.000000", "0.000029", "0.000029", "0.665274", "0.665274"
"01/31/2023 10:47:49.030", "0.000000", "3738397.766329", "3738397.766329", "0.000000", "0.000000", "0.000103", "0.000103", "2.571416", "2.571416"
"01/31/2023 10:47:50.035", "32604.936193", "32604.936193", "3635450.385532", "3635450.385532", "0.000321", "0.000321", "0.000042", "0.000042", "0.988092", "0.988092"
"01/31/2023 10:47:51.050", "161246.558723", "161246.558723", "854606.761234", "854606.761234", "0.000277", "0.000277", "0.000367", "0.000367", "2.014659", "2.014659"
```

c. Cleanup Stage:

```
sysbench --threads=4 fileio --file-total-size=2G --file-test-mode=rndwr cleanup
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\patel> typeperf "\PhysicalDisk(*)\Disk Read Bytes/sec" "\PhysicalDisk(*)\Disk Write Bytes/sec" "\PhysicalDisk(*)\Avg. Disk sec/Read" "\PhysicalDisk(*)\Avg. Disk sec/Write" "\PhysicalDisk(*)%\ Disk Time"

"(PDH-CSV 4.0)", "\MSI\PhysicalDisk(0 C: D:)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Read Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(_Total)\Disk Write Bytes/sec", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Read", "\MSI\PhysicalDisk(0 C: D:)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(_Total)\Avg. Disk sec/Write", "\MSI\PhysicalDisk(0 C: D:)% Disk Time", "\MSI\PhysicalDisk(_Total)% Disk Time"
"01/31/2023 10:48:43.984", "52714.236725", "52714.236725", "137868.003741", "137868.003741", "0.011186", "0.011186", "0.001393", "0.001393", "2.480768", "2.480768"
"01/31/2023 10:48:44.985", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000", "0.000000"
"01/31/2023 10:48:46.000", "0.000000", "4038.614917", "4038.614917", "0.000000", "0.000000", "0.010949", "0.010949", "1.079203", "1.079203"
"01/31/2023 10:48:47.000", "61449.696762", "61449.696762", "73739.636115", "73739.636115", "0.011421", "0.011421", "0.000063", "0.000063", "1.230784", "1.230784"
```

Observations from the above cases:

1. QEMU VM

Stage	I/O Throughput (MB/sec)	I/O Latency (Bytes/sec)	I/O Disk Utilization
Prepare	68.56	0.000397	4.16
Run	12.547	0.000028	2.49
Cleanup	0.448	0.0037	2.31

2. Docker Container

Stage	I/O Throughput (MB/sec)	I/O Latency (Bytes/sec)	I/O Disk Utilization
Prepare	17.04	0.000396	9.596
Run	3.73	0.000185	2.71
Cleanup	0.13	0.011484	2.48

Conclusion:

1. Regarding three stages(prepare, run and cleanup): I/O throughput, latency and disk utilization vary based on the stage and the highest value of all the parameters is shown by the prepare stage followed by run and cleanup stage.
2. The Disk Utilization was more by containers as compared to QEMU VM while running above mentioned test cases.

Automation

Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/hirsute64"
  config.vm.provider "virtualbox" do |vb|
    vb.memory = "2048"
    vb.cpus = 2
  end

  #Below command will sync the local folder qemu to the /vagrant_folder folder
  #on the virtual machine.
  #Hence, any files added to the qemu folder on your host machine will be
  #available in the /vagrant_folder folder on the virtual machine.
  config.vm.synced_folder "qemu", "/vagrant_folder"

  config.vm.provision "shell", path: "vagrant_setup.sh"

end
```

Dockerfile

```
FROM sysbench_ubuntu:v1

COPY main_docker.sh /main_docker.sh
COPY cpu.sh /cpu.sh
COPY file.sh /file.sh

RUN chmod +x main_docker.sh
RUN chmod +x cpu.sh
RUN chmod +x file.sh

ENTRYPOINT bash main_docker.sh
```