Mini Project 1: Containers and Virtual VMs

Name: Kanishk Barhanpurkar

Email: kbarhan1@binghamton.edu

B-Number: B-00863529

1. VM Instance Configuration Details

The VM-instance is created using the provided instructions and specified OS version (Ubuntu Bionic Beaver (18.04) with 30 GB disk space is enabled.

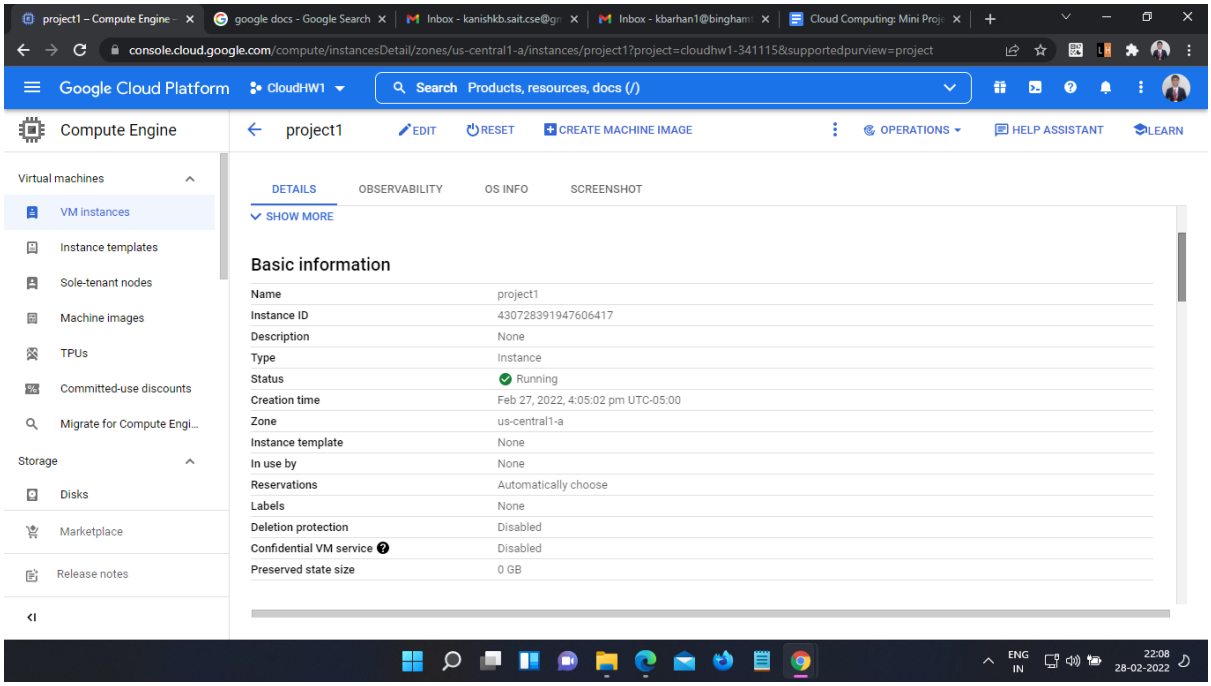

Figure 1: Basic Information about the created VM-instance "project1".

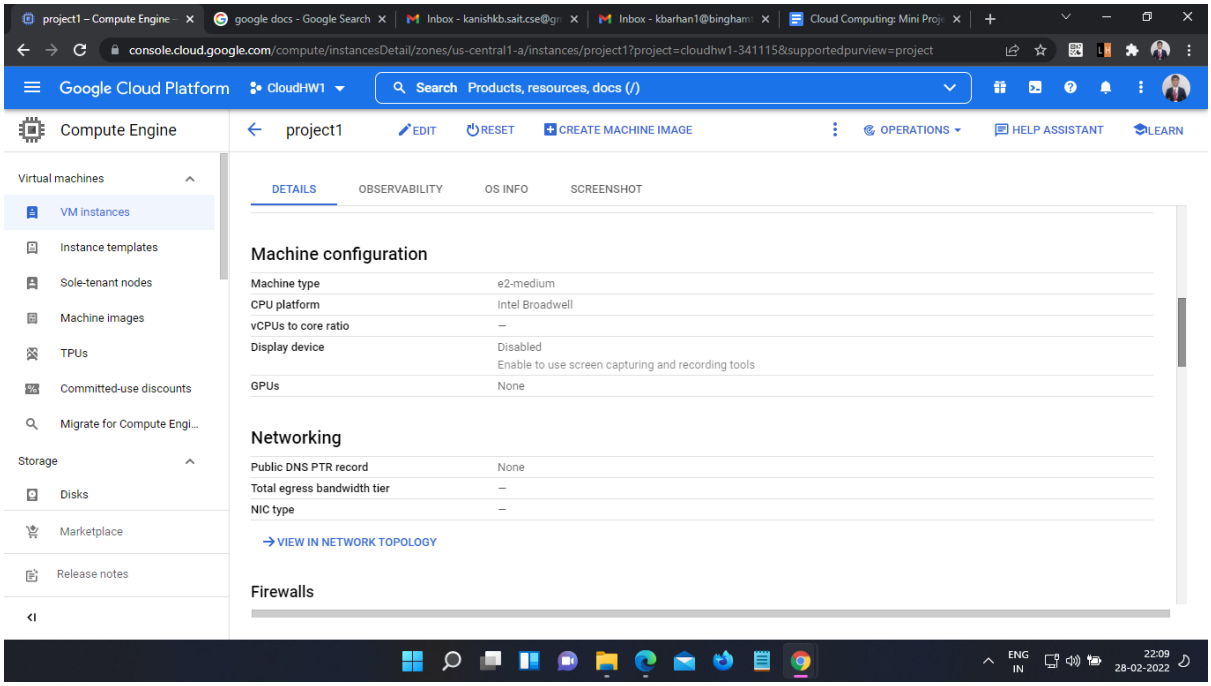

Figure 2: Machine configuration for the VM-instance “project1”.

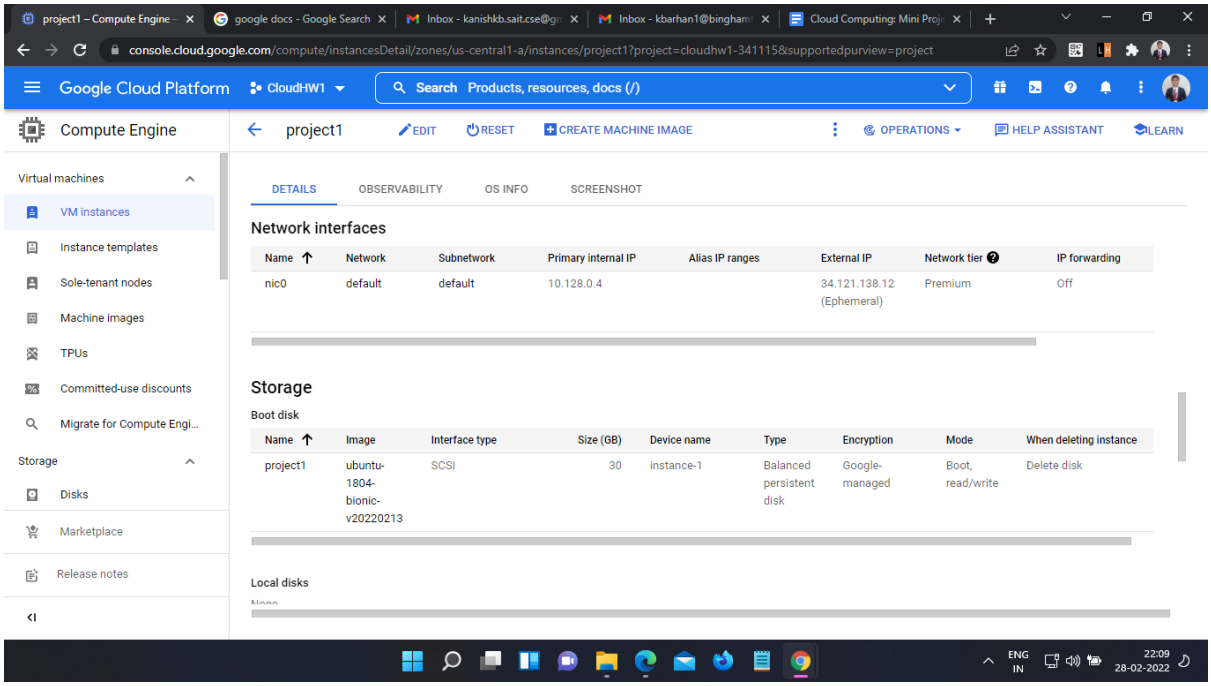

Figure 3: Storage and network interfaces details for the VM-instance “project1”.

2. Docker Basics

The Docker has been installed in the VM-instance “project1” and basic commands are performed to understand the basics of Docker.

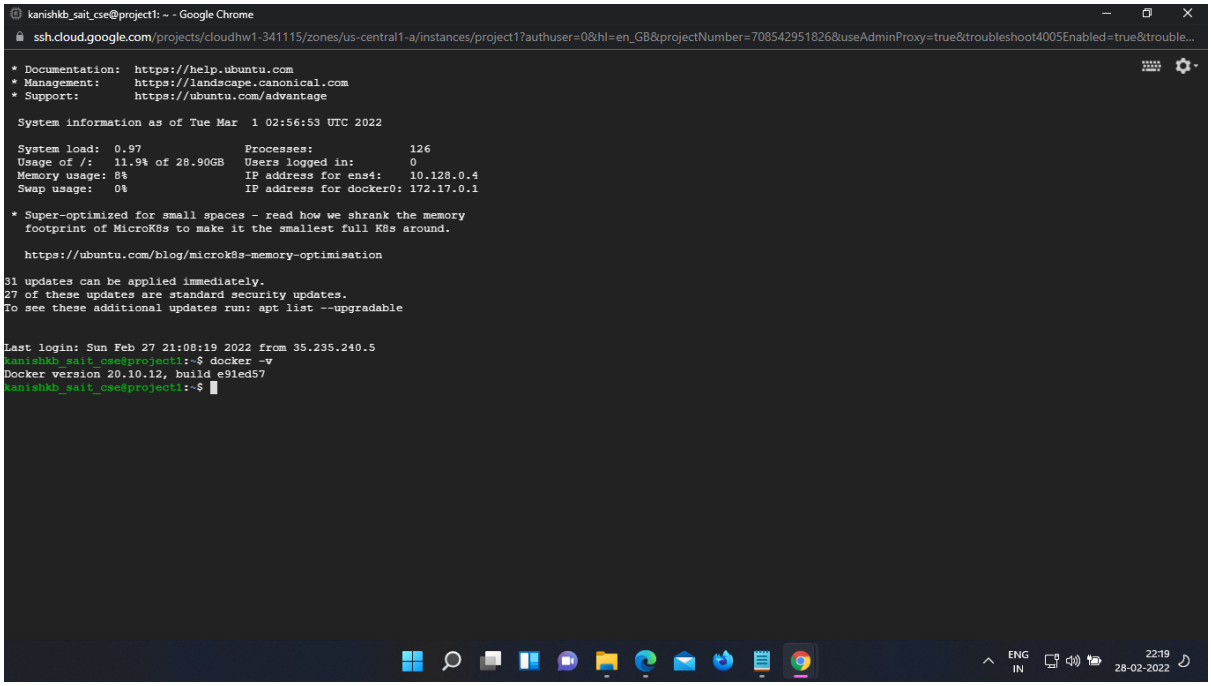

Figure 4: Command **“**docker –v**”** output to show that Docker installation.

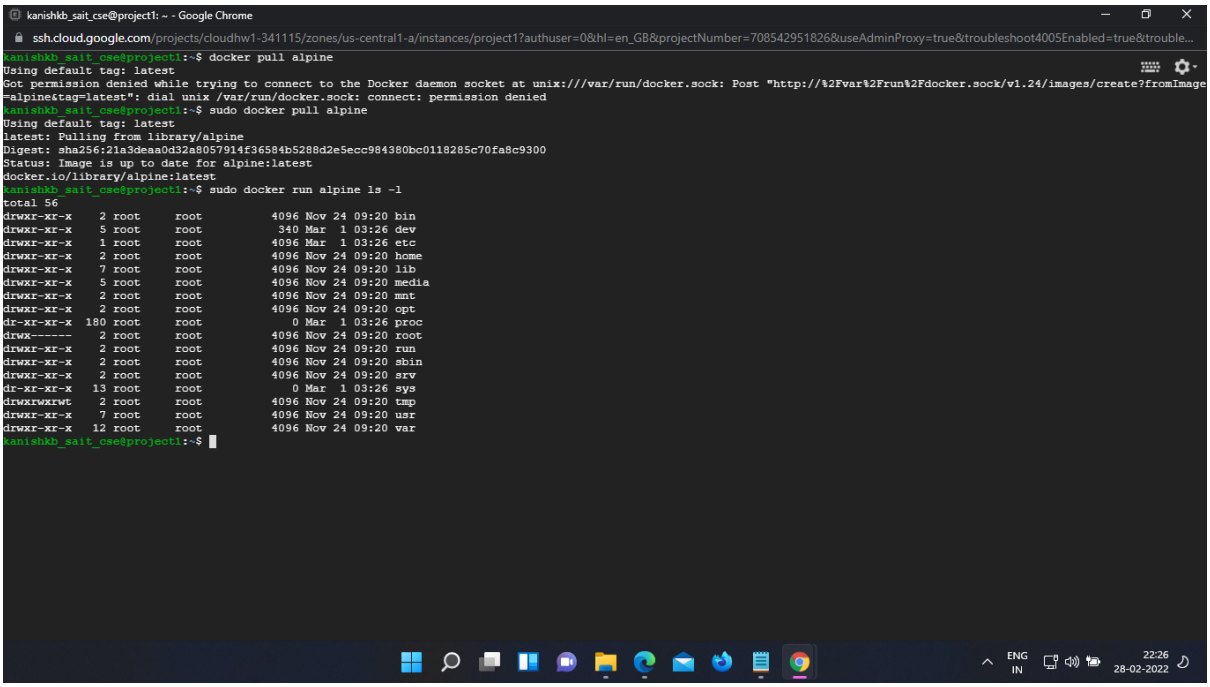

Figure 5: Docker and “alpine” as first container.

To run the first container and understand the set of commands associated with Docker, Alpine Container has been launched and then runs a command in Docker.

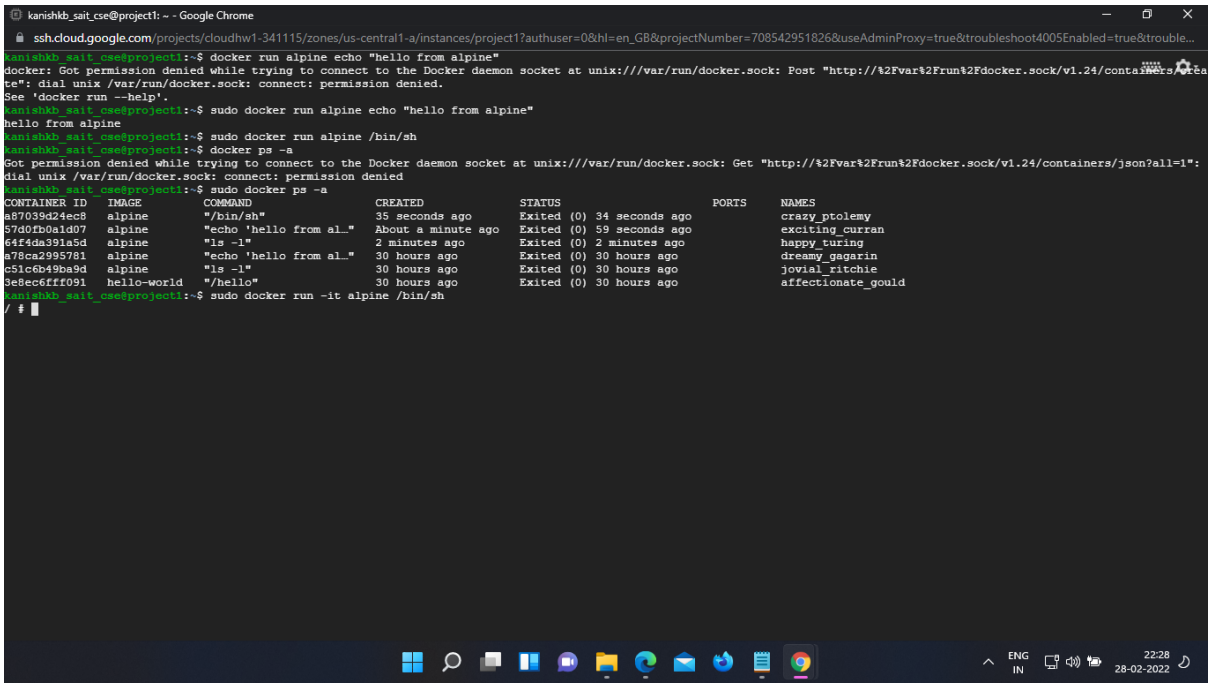

Figure 6: Different commands associated with Docker Alpine container.

3. QEMU Installation

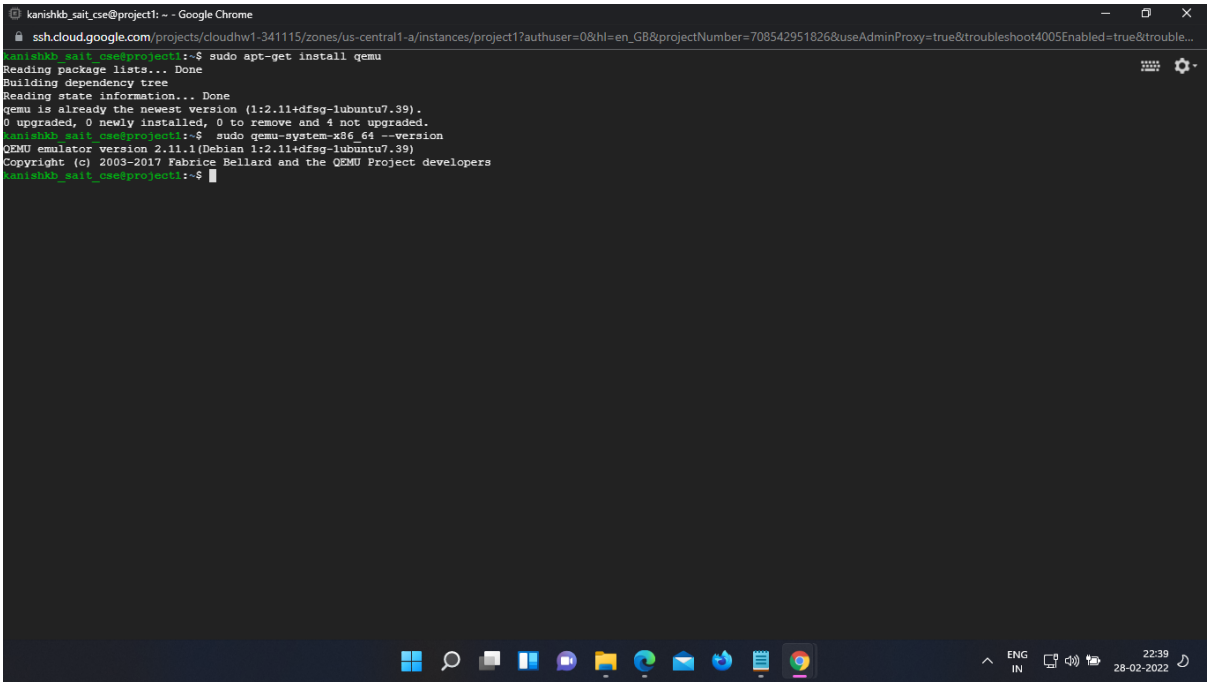

Figure 7: Commands to check about QEMU is installed.

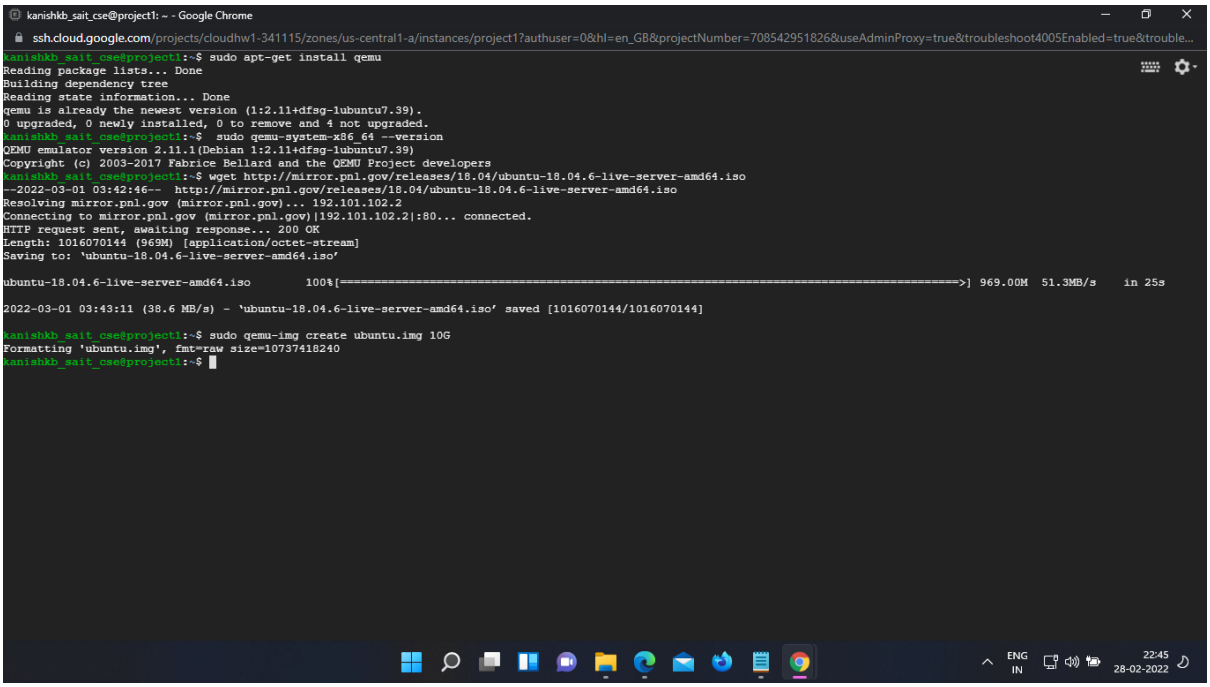

Figure 8: Commands to check about QEMU is installed.

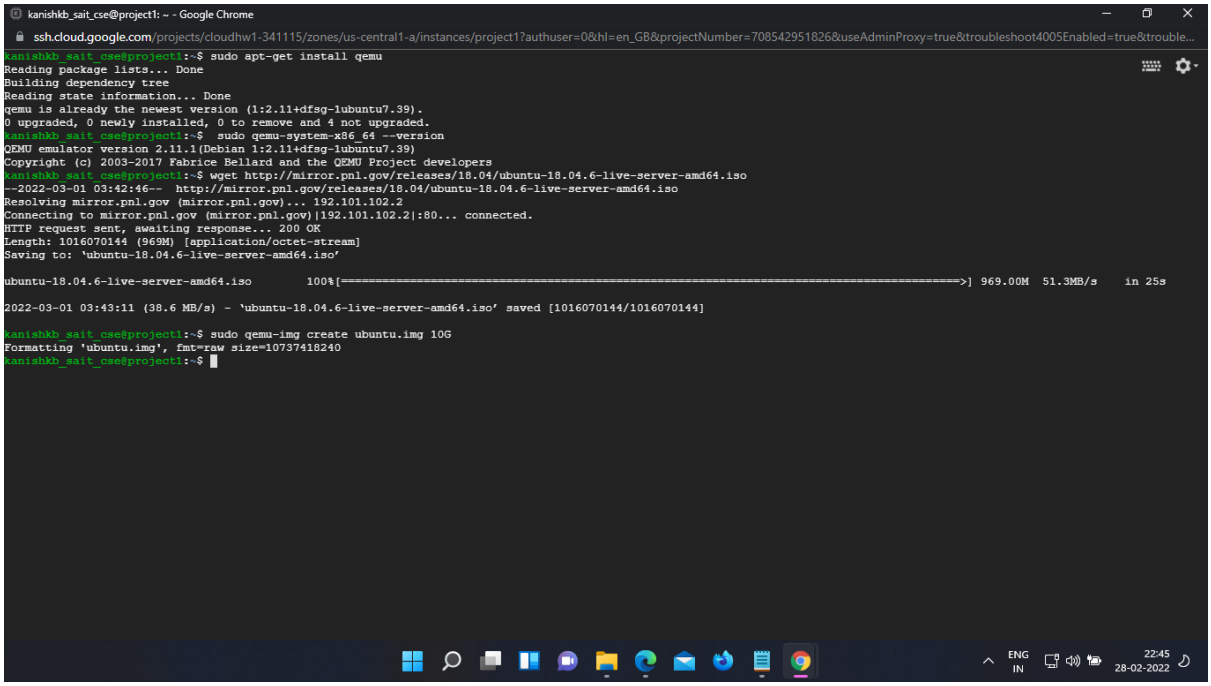

Figure 9: To create an image before installation for QEMU VM.

The following commands are performed for QEMU installation

- $sudo apt-get update
- $sudo apt-get install qemu
- $wget http://mirror.pnl.gov/releases/18.04/ubuntu-18.04.6-live-server-amd64.iso
- $sudo qemu-img create ubuntu.img 10G

To enable the QEMU VM following commands are used-

- $sudo apt-get update
- $sudo apt-get install gnome-core
- $sudo apt-get install gnome-panel
- $sudo apt-get install vnc4server
- $vncserver

4. Sysbench Installation

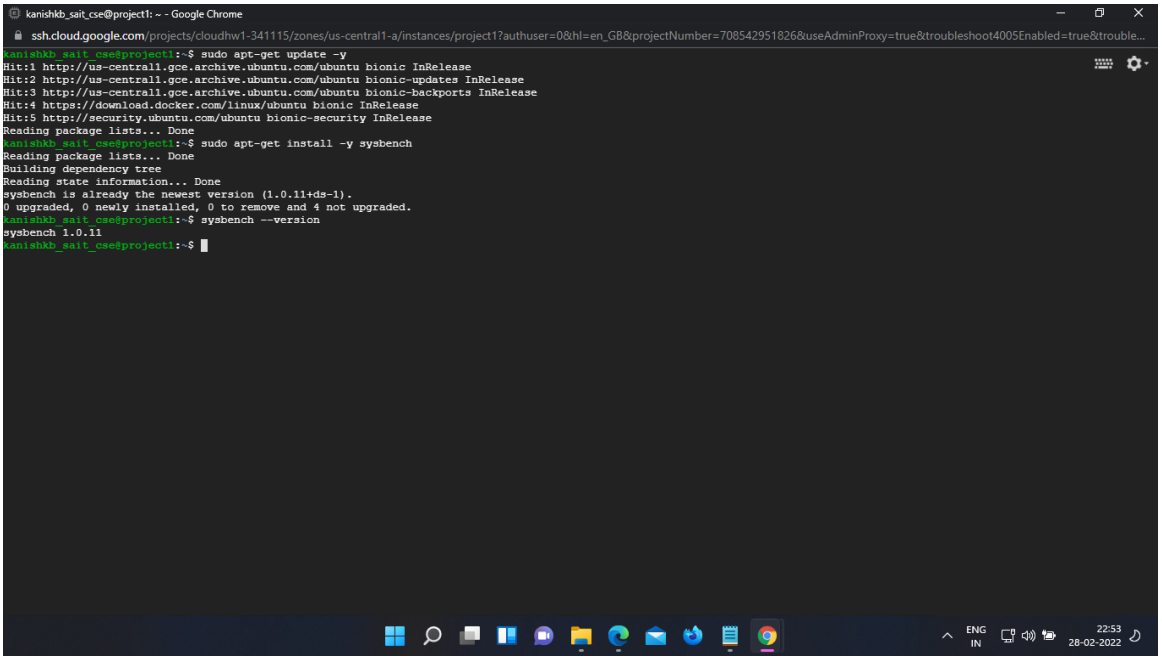

Figure 10: Status of Sysbench installation.

5. Iostat installation and basic operations

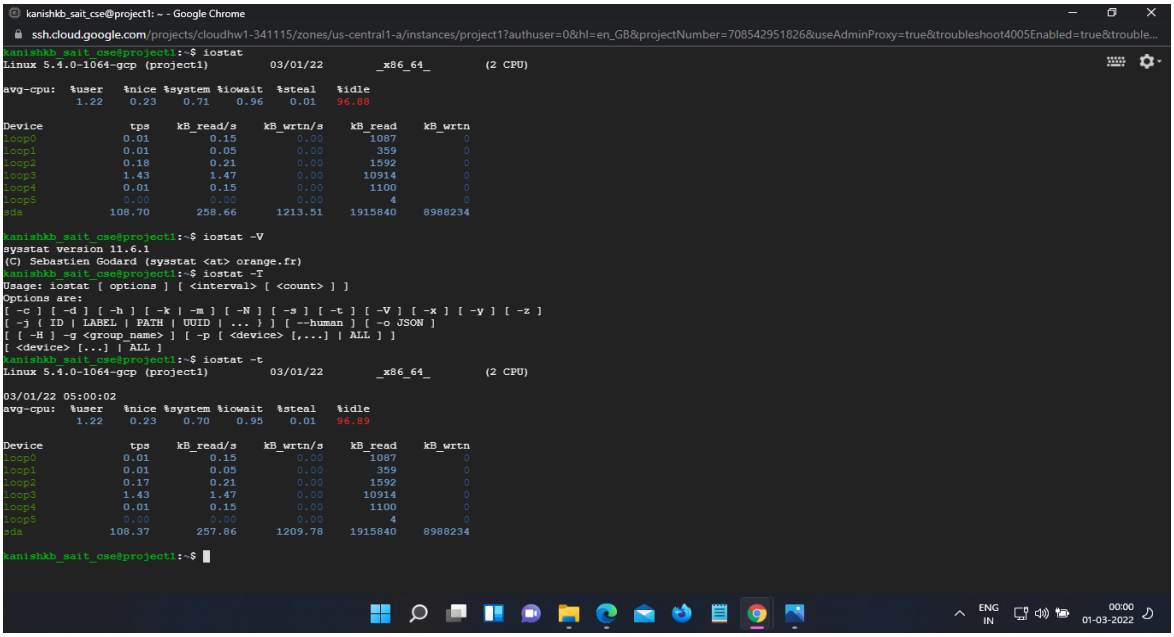

Figure 11: Iostat basic operations.

6. Sysbench CPU performance testing (inside Docker)

[Run 3 times; results attached for 1 iteration]

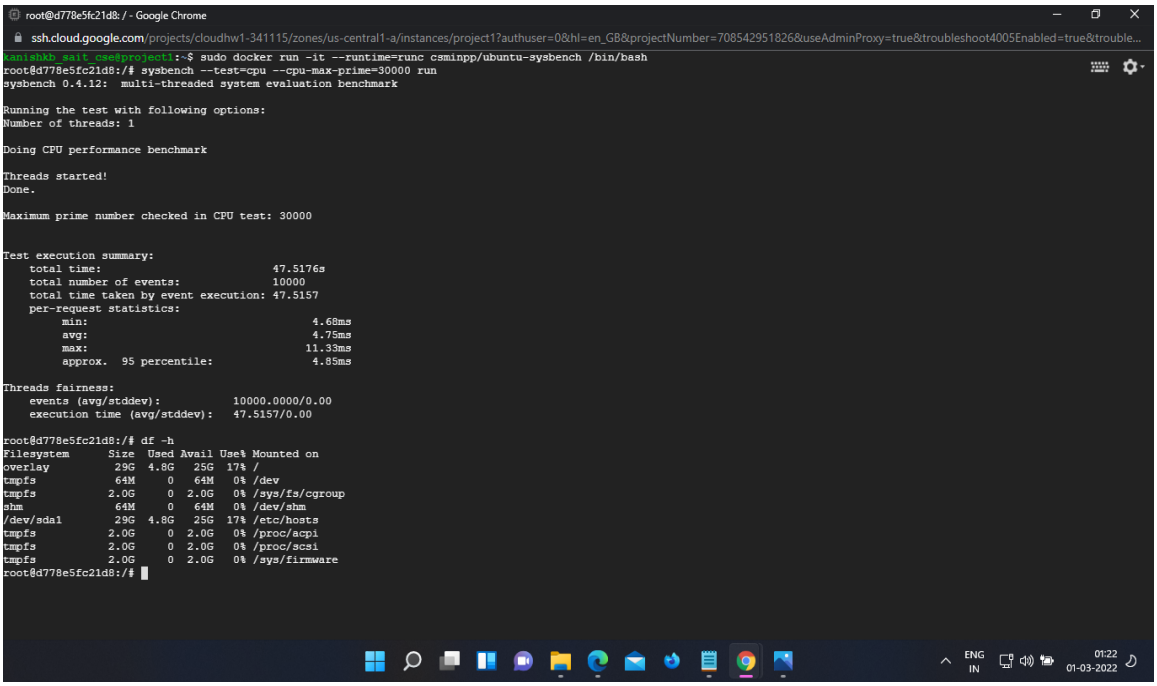

Figure 12: Output for following commands-

- $sudo docker run -it --runtime=runc csminpp/ubuntu-sysbench /bin/bash
- $sysbench --test=cpu --cpu-max-prime=30000 run
- $df –h

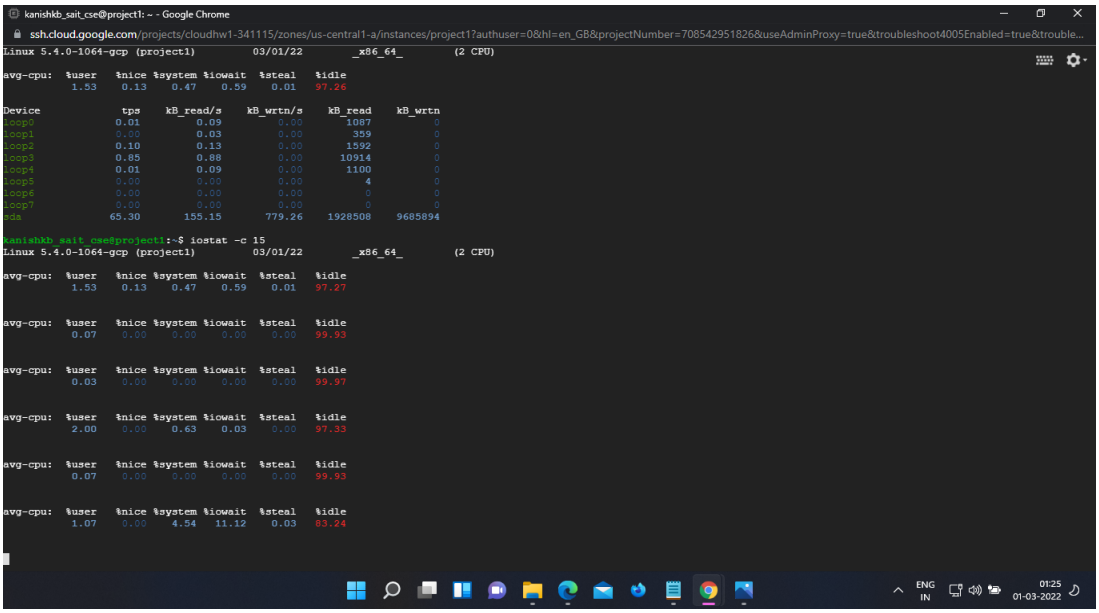

Figure 13: Output for the following commands-

- $iostat ALL
- $iostat -c 15

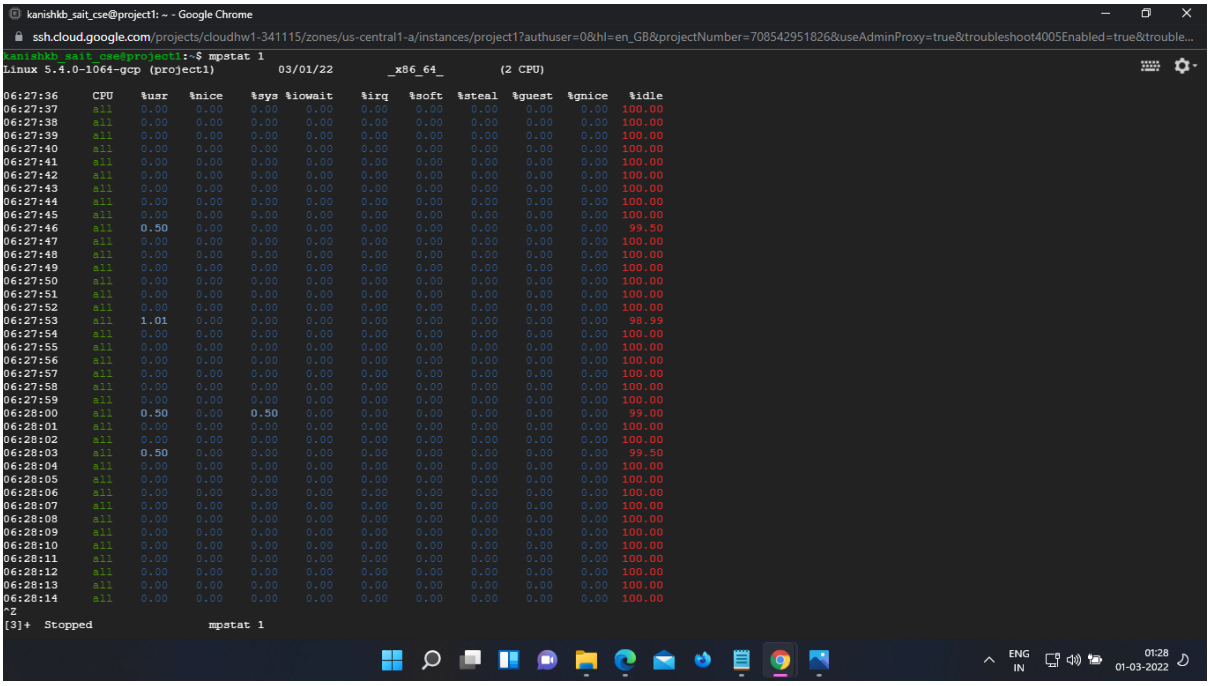

Figure 14: Output for the following commands-

- $mpstat 1

7. Sysbench CPU commands (native system)

[Run 3 times; results attached for 1 iteration]

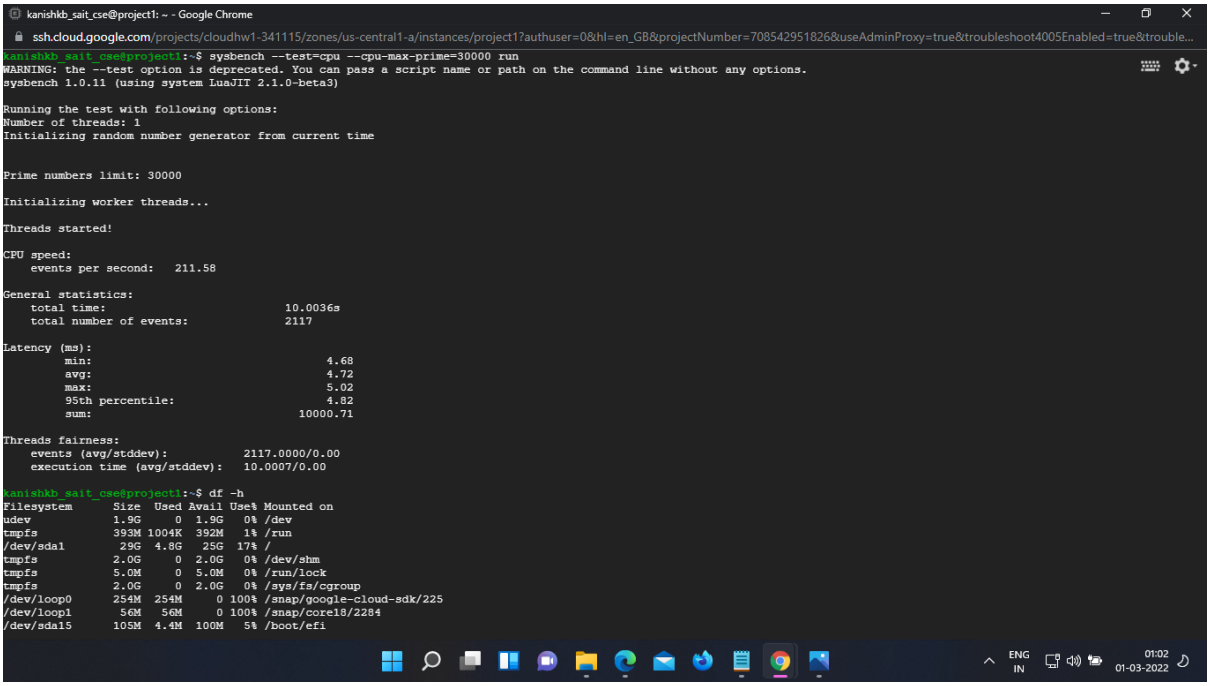

Figure 15: Output for the following commands-

- $ sysbench --test=cpu --cpu-max-prime=30000 run
- $df –h

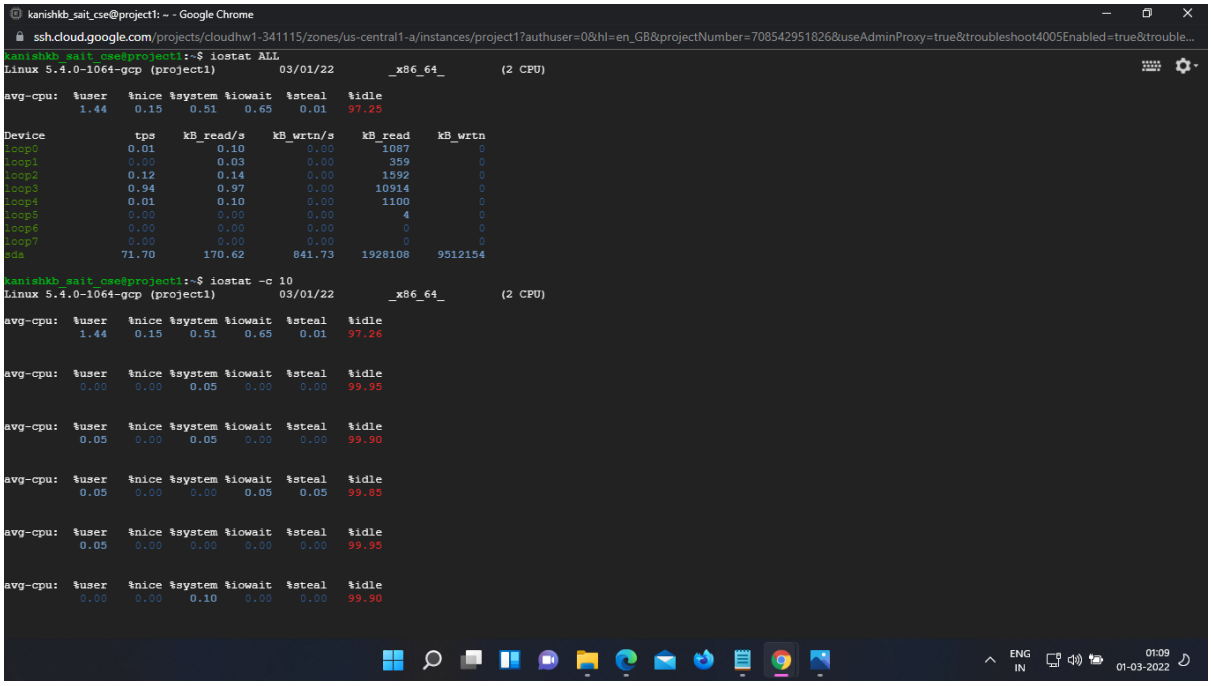

Figure 16: Output for the following commands-

- $ iostat ALL
- $ iostat –c 10

Major Insights

- Total time (average) for the CPU operation under Docker is 47.83 seconds.
- Total time (average) for the CPU operation under native system is 10.578 seconds.
- Hence, it concludes that the performance tools in the native system is running faster than the Docker system for the mandatory condition that the number of runs=30000 remains constant.

8. Sysbench fileio commands (under docker system)
[Run 3 times; results attached for 1 iteration]

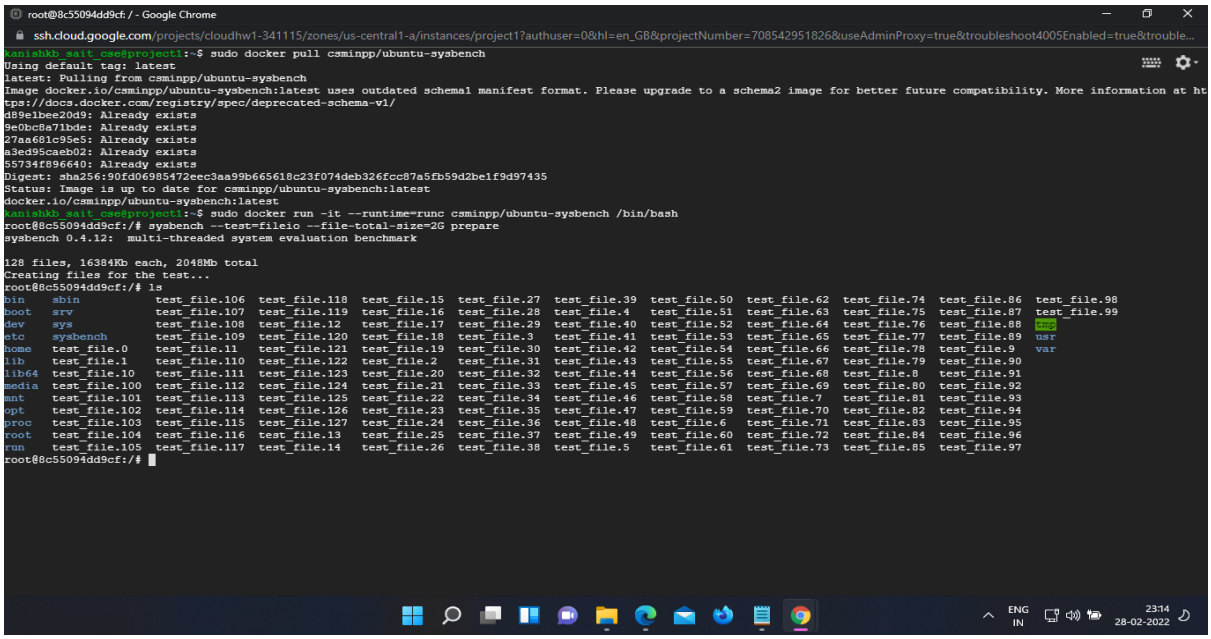

Figure 17: Output for the following command-

- $sudo docker pull csminpp/ubuntu-sysbench
- $sudo docker run -it --runtime=runc csminpp/ubuntu-sysbench /bin/bash
- $sysbench --test=fileio --file-total-size=2G prepare

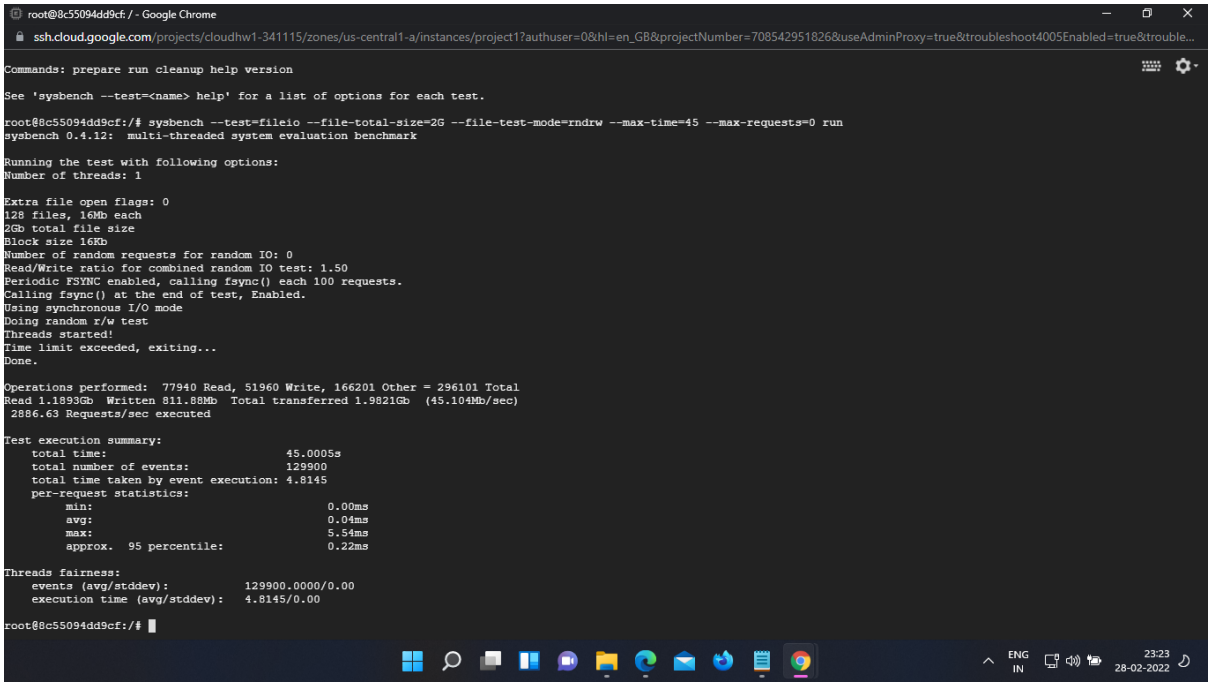

Figure 18: Output for the following command-

$ sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=45 --max-requests=0 run

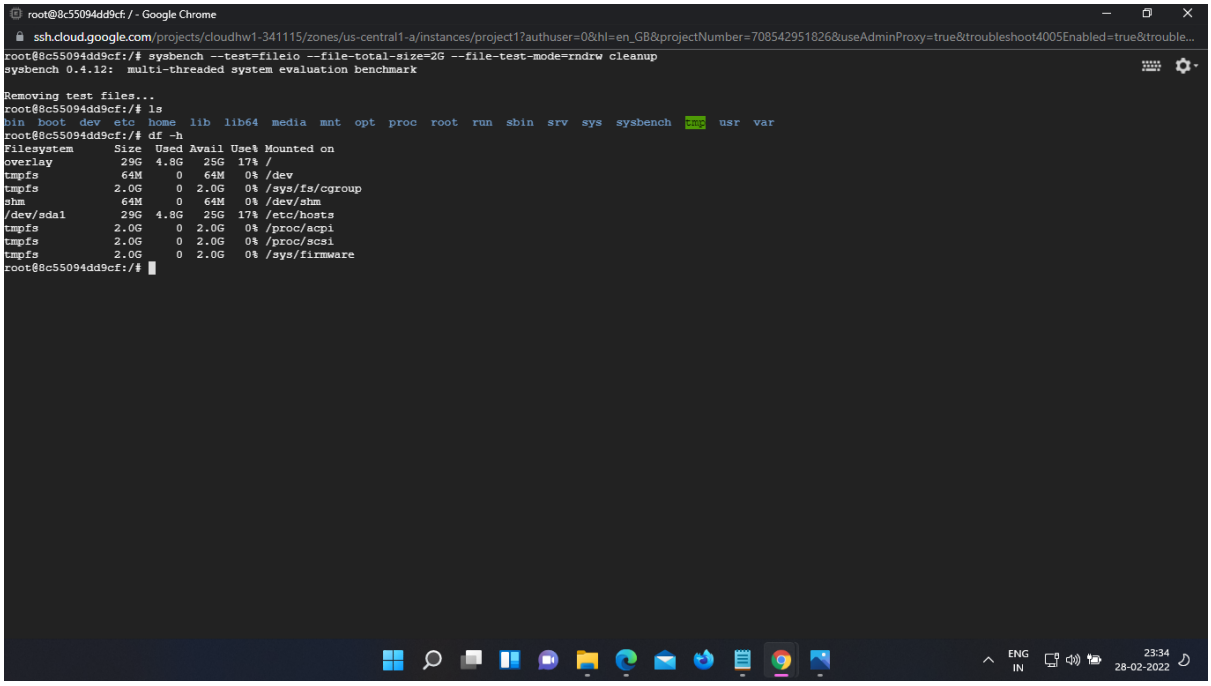

Figure 19: Output for the following command-

- $ sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw cleanup

9. Sysbench fileio commands (file system)

[Run 3 times; results attached for 1 iteration]

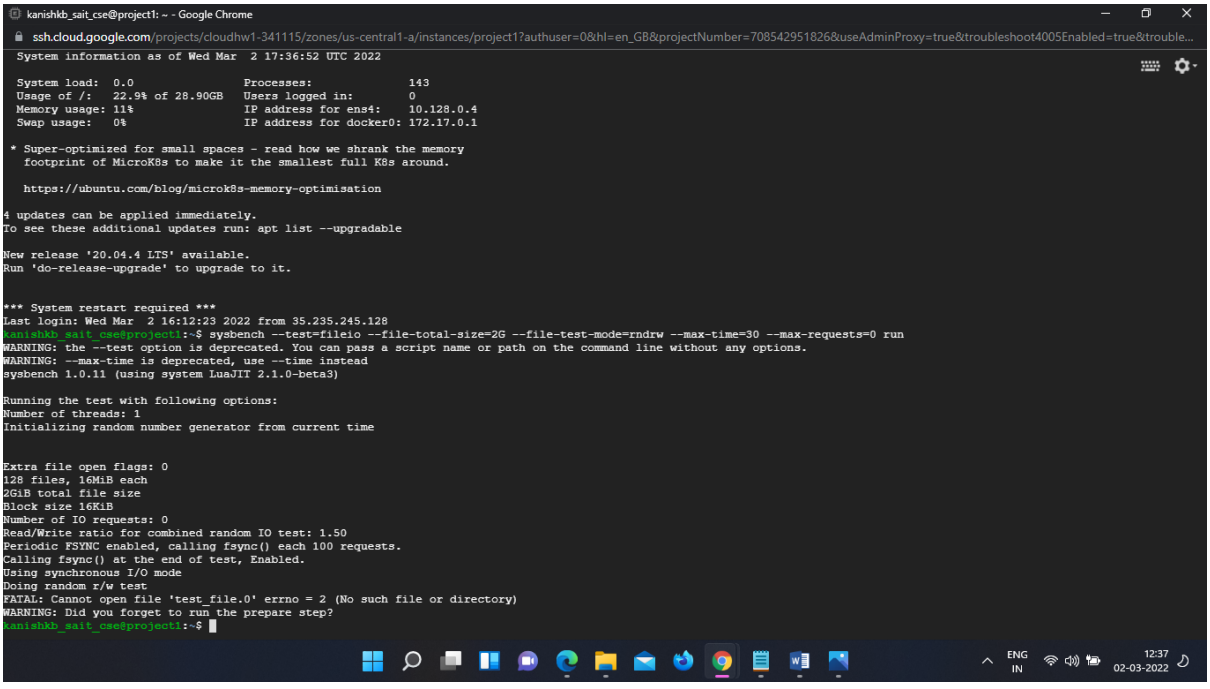

Figure 20: Output for the following command-

- $ sysbench --test=fileio --file-total-size=2G --file-test-mode=rndrw --max-time=30 --max-requests=0 run

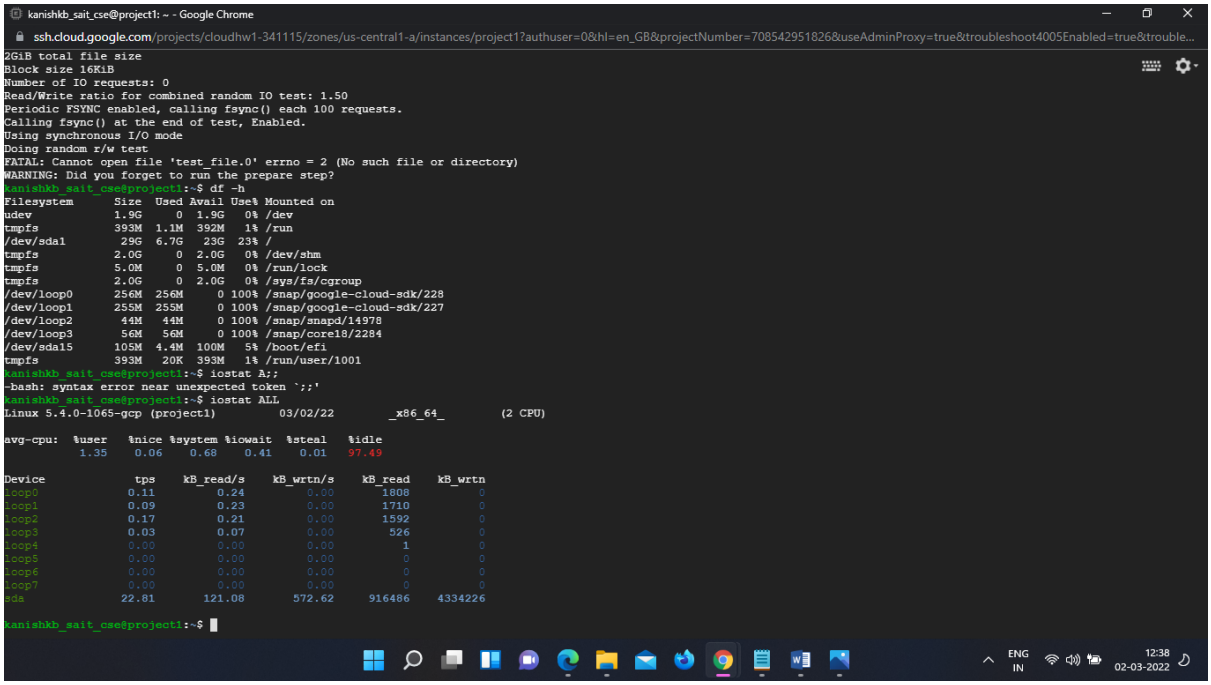

Figure 21: Output for the following command-

- $ iostat ALL

Major insights

- The average fileio parameter total time under Docker system for reading files is 47.63 seconds.
- The average fileio parameter total time in native system for reading files is 13.58 seconds.
- Hence the results shows that native system works faster than Docker system.

Reason for late execution of QEMU VM: The QEMU VM emulates all devices due to which it is needed to run a VM guest. The OS executes a system call and it does not connect with the hardware. The system call is going to the hypervisor and the host OS before the call gets executed. Thus, it increases the delay time.

Total Time to install QEMU VM: ~4 Hours (with VNCserver)

Part-II: miniDocker.py

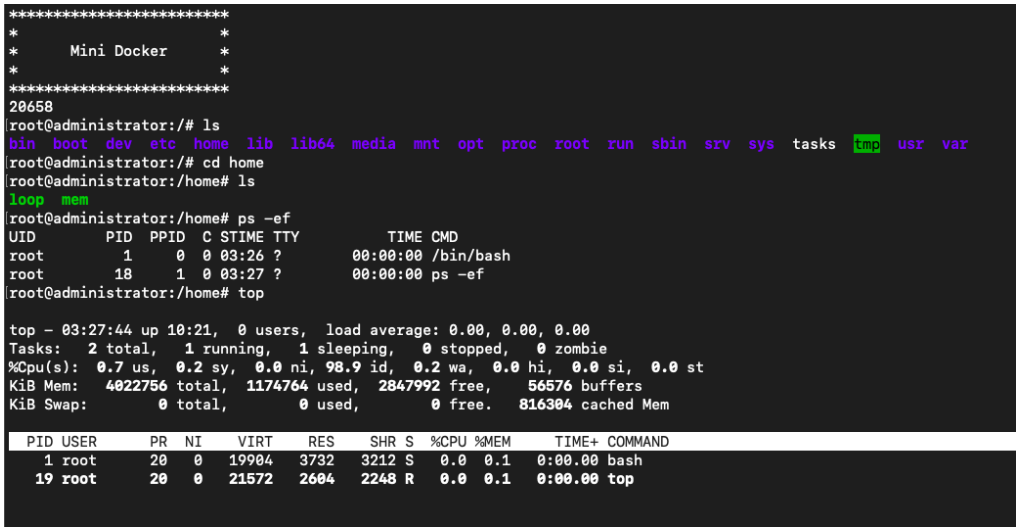

Figure 23: Executing miniDocker.py

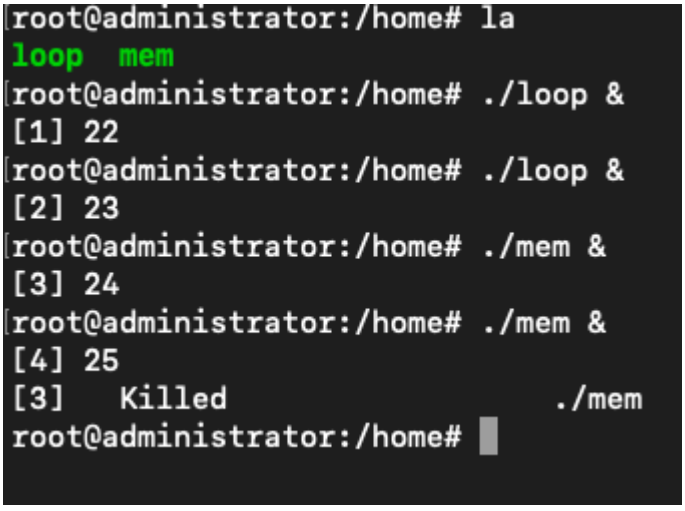

Figure 24: Test cpuset Cgroup.