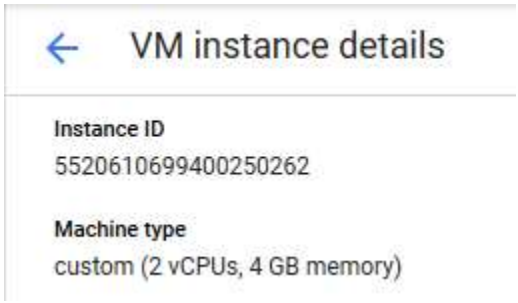
CS 580K Mini-Project 1

Ketan Deshpande
B00816854
kdeshpa5@binghamton.edu

Google cloud instance configurations:



Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.128.0.4	–	146.148.103.39 (ephemeral)	Premium	Off	View details

Boot disk

Name	Image	Size (GB)	Device name	Type	Encryption
nativesystem	ubuntu-1604-xenial-v20200223	20	nativesystem	Standard persistent disk	Google-managed

Steps to enable Docker container:

I have installed Docker for this assignment using the repository option. The detailed steps are as follows:

Reference: https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-using-the-repository

1. Update the apt package
2. Install packages to allow apt to use a repository over HTTPS
3. Add Docker's official GPG key (`9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88`)
4. Install the specific version of Docker Engine – Community by providing the version in the command
5. Verify the Docker Engine – Community is installed by running hello-world image
6. We need to use csminpp/ubuntu-sysbench, pre-installed image for benchmarks. This can be done using sudo docker pull and image name.

Steps to install QEMU:

1. Install QEMU by using sudo apt-get install qemu
2. Download the Ubuntu image to install into QEMU
3. Create image to install Ubuntu in QEMU

Steps to install GUI for native Google instance:

Reference: https://medium.com/google-cloud/graphical-user-interface-gui-for-google-compute-engine-instance-78fccda09e5c

1. Install the Gnome components
   `sudo apt-get install gnome-core`
2. Install VNC server to interact with desktop environment
   `sudo apt-get install vnc4server`
3. Run server by using `vncserver` command and set password for this instance
4. Verify it's working by netcat command
5. To work this GUI properly, we need to make some changes in the configuration. First kill the session.
   `vncserver -kill :1`
6. Open the xatartup file and make the changes as suggested in the assignment document as follows:

```
#!/bin/sh

Def

export XKL_XMODMAP_DISABLE=1

unset SESSION_MANAGER

unset DBUS_SESSION_BUS_ADDRESS

metacity &

gnome-settings-daemon &

gnome-panel &

nautilus &

gnome-terminal &
```

7. Install VNC client on local machine.
8. Add new firewall rule in the cloud by setting the configuration provided in the screenshot which is in the assignment document, as follows:

vnc-server

Description

Logs

Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver.Learn more

- ○ On
- ◉ Off

Network

default

Priority *

1000

Priority can be 0–65535Check priority of other firewall rules

Direction

Ingress

Action on match

Allow

Targets

All instances in the network

Source filter

IP ranges

Source IP ranges *

0.0.0.0/0 for example, 0.0.0.0/0, 192.168.2.0/24

Second source filter

None

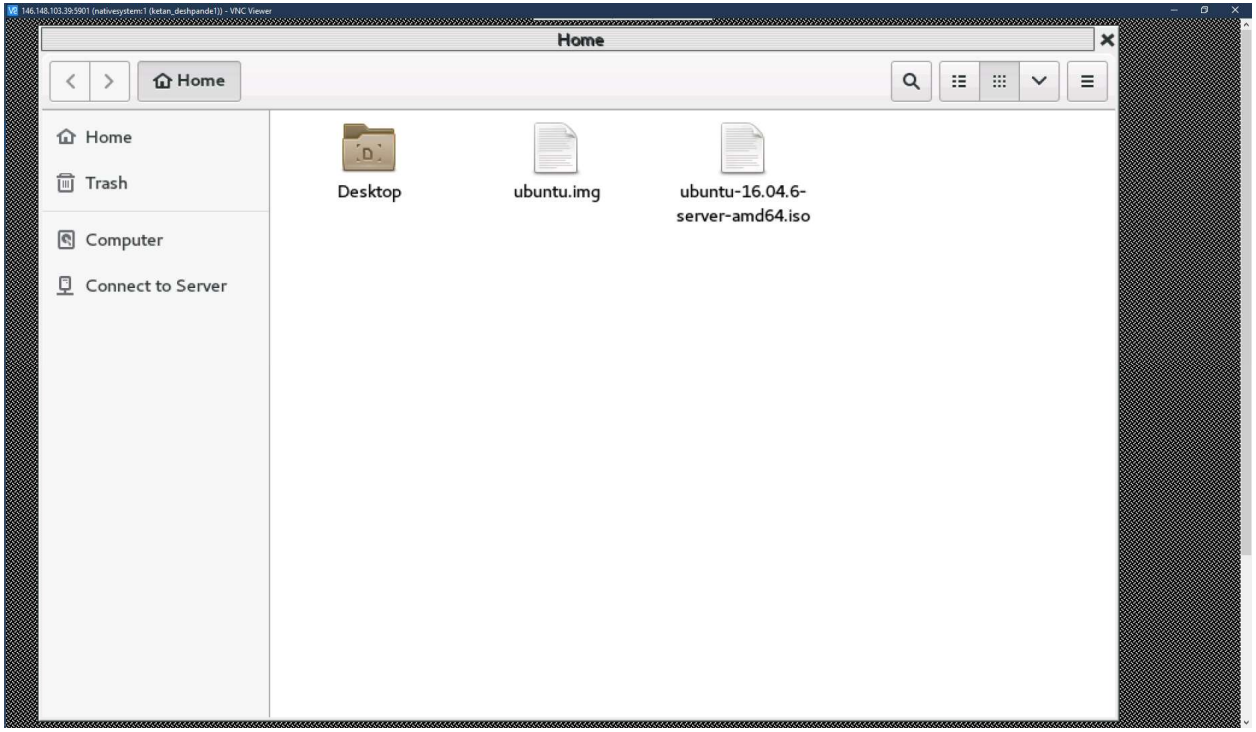
Protocols and ports

- ○ Allow all
- ◉ Specified protocols and ports

☑ tcp : 5000-10000

charged-skein-267904 all

9. Now check the connection by the firewall by netcat
   `nc 146.148.103.39 5901`
10. After installing VNC client, connect to the external IP through it. It'll open GUI for the Google instance.



Performance measurements in the native, Docker, and QEMU:

I have run the test cases on all the three environments to measure the system performance from CPU utilization and IO perspective.

Commands used for CPU utilization and IO:

1. sysbench --num-threads=2 --test=cpu --cpu-max-prime=30000 run
   This command calculates prime numbers up to the given number by using threads. Each thread executes the requests concurrently until either the total number of requests or the total execution time exceed the limits specified with the common command line options.
2. iostat -<option> <seconds>
   iostat command is used to monitor CPU utilization and I/O (input /output) statistics of all the disks and file systems. Options are c for CPU and d for devices. Seconds are the time interval.
3. sudo -i
   this command is used to change the user to root
4. echo 3 > /proc/sys/vm/drop_caches
   this command is used to clear the cache
5. sudo qemu-system-x86_64 -hda ubuntu.img -m 1536
   this command is used to start QEMU VM
6. sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=<mode> <prepare/run/cleanup>
   this command will create 128 files with total size of 3 GB in current directory, run the bench mark and clean the test files by provided modes.

I have run the commands and test cases. Please see below the screenshots for all of them:

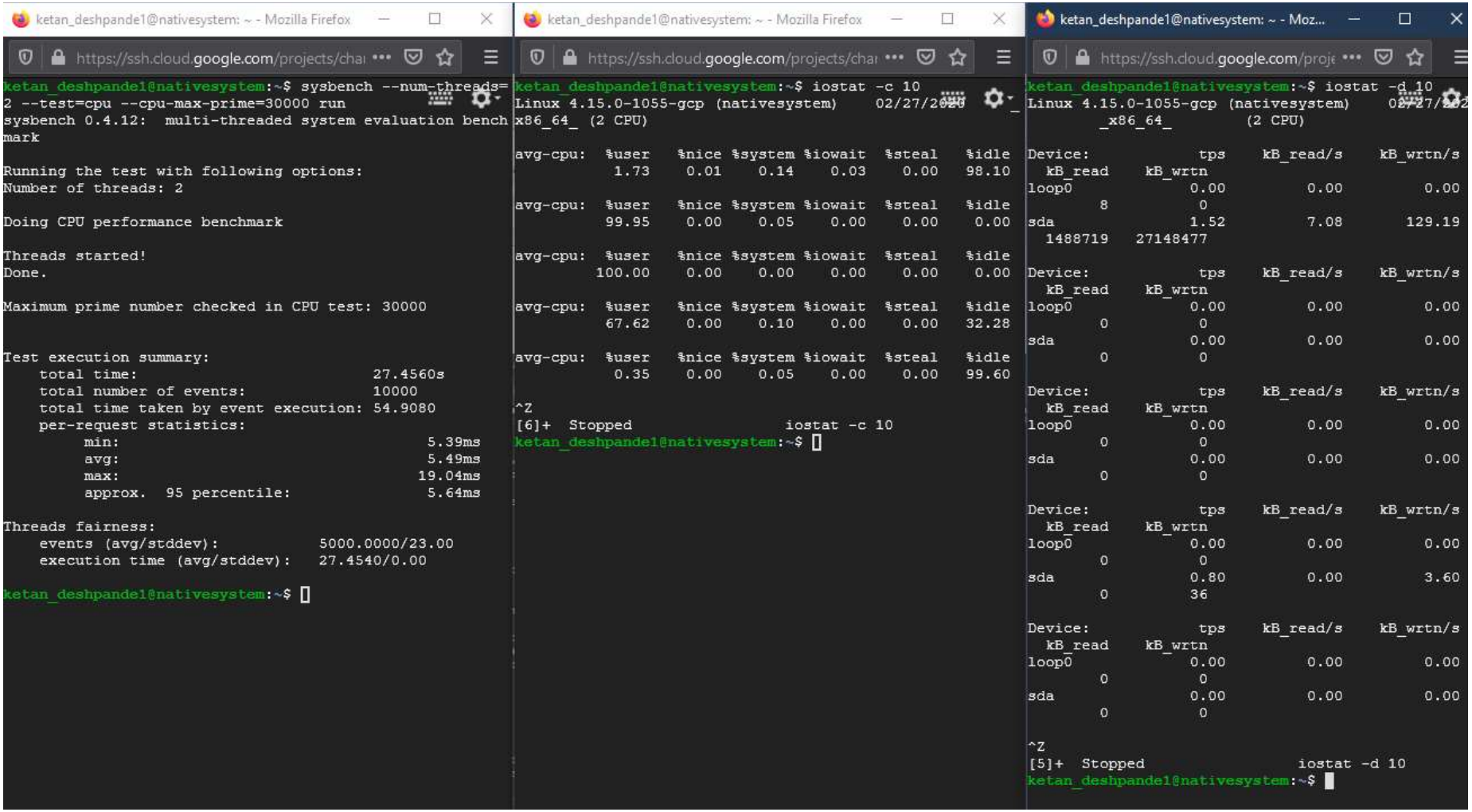
CPU Test 1

Command - sysbench --num-threads=2 --test=cpu --cpu-max-prime=30000 run

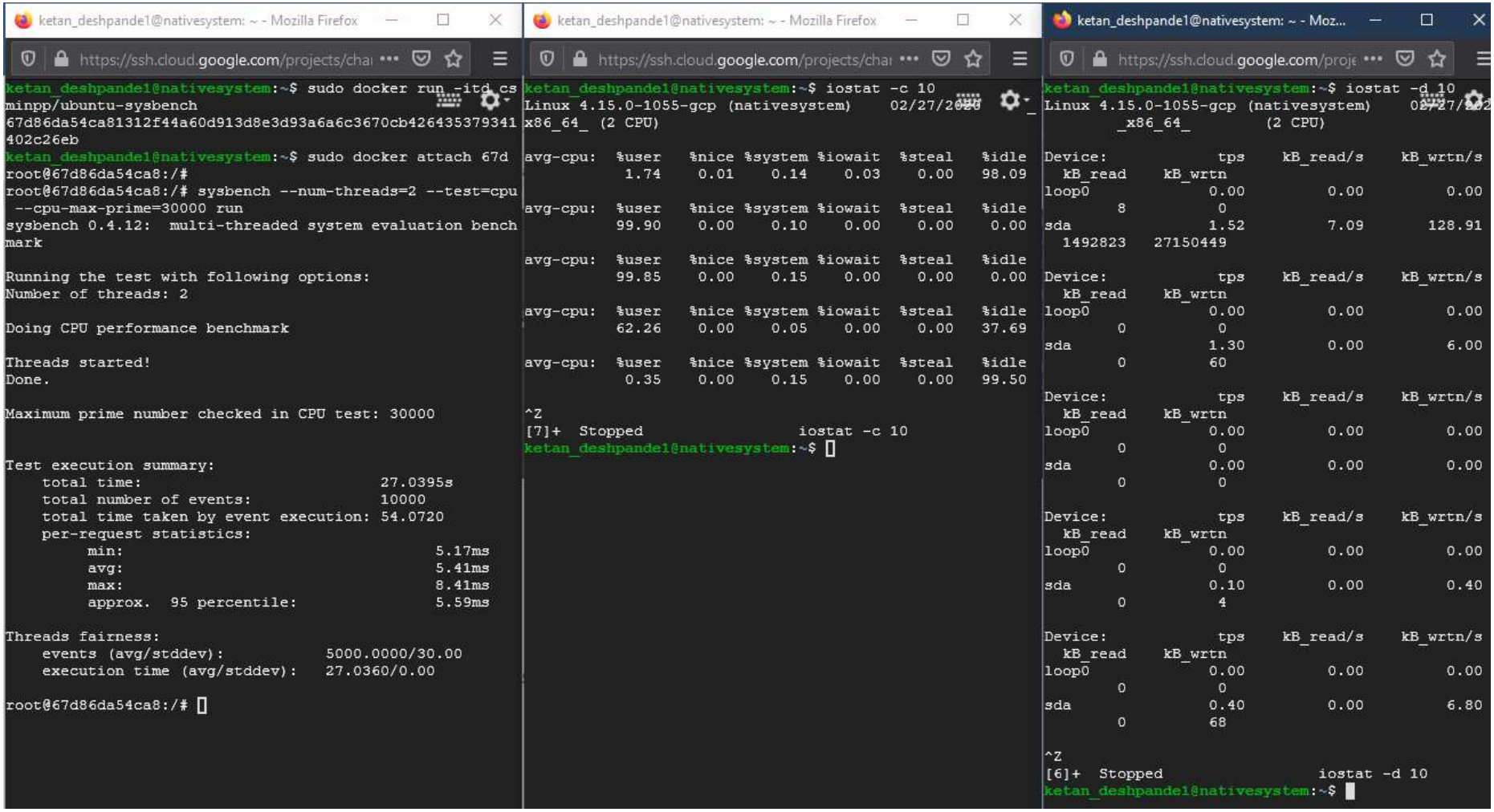
iostat -c 10

iostat -d 10

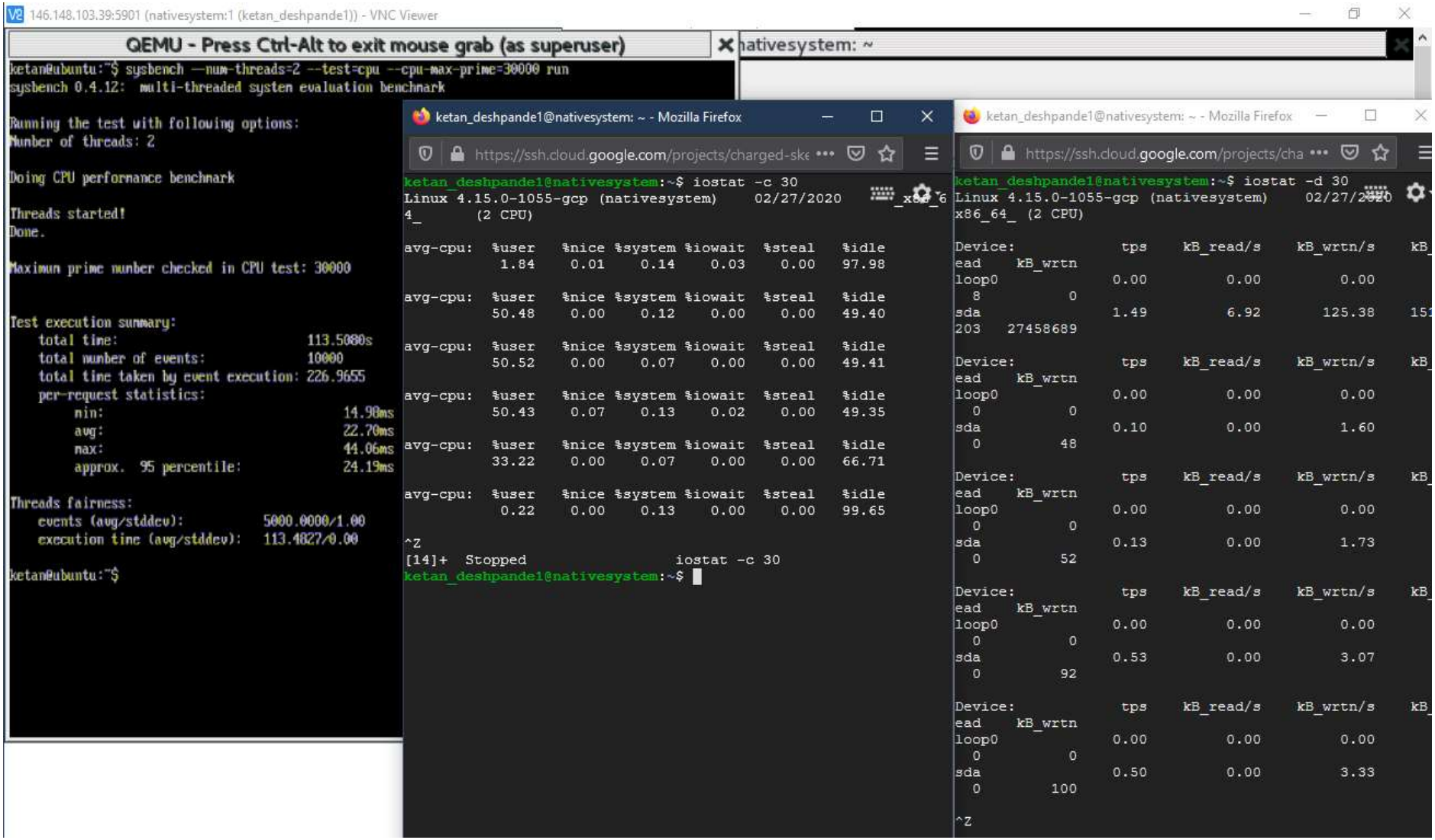
Native System: Total time taken: 27.45s



Docker: Total time taken: 27.03s



QEMU: Total time taken: 113.50s



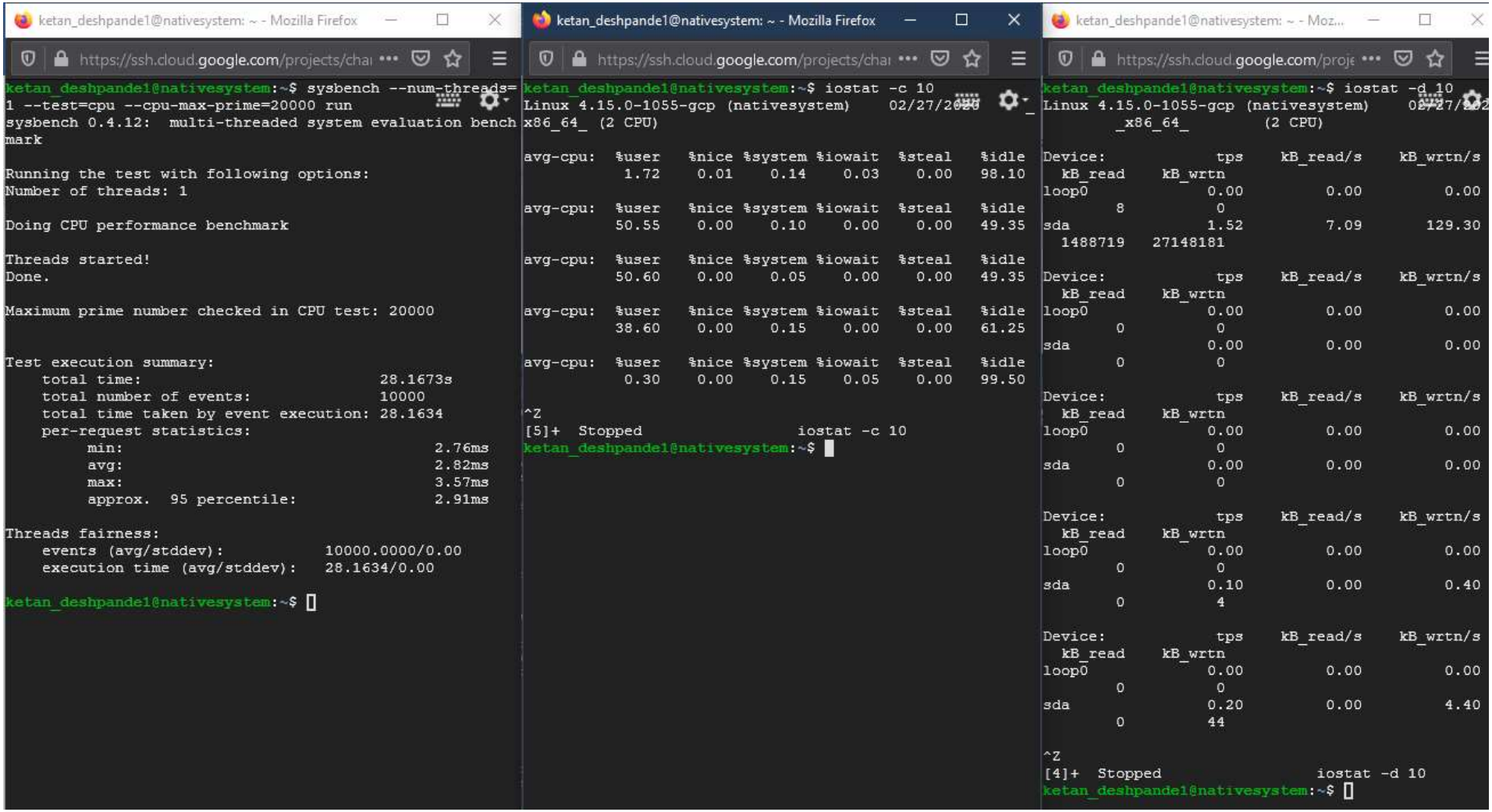
CPU Test 2

Command - sysbench --num-threads=1 --test=cpu --cpu-max-prime=20000 run

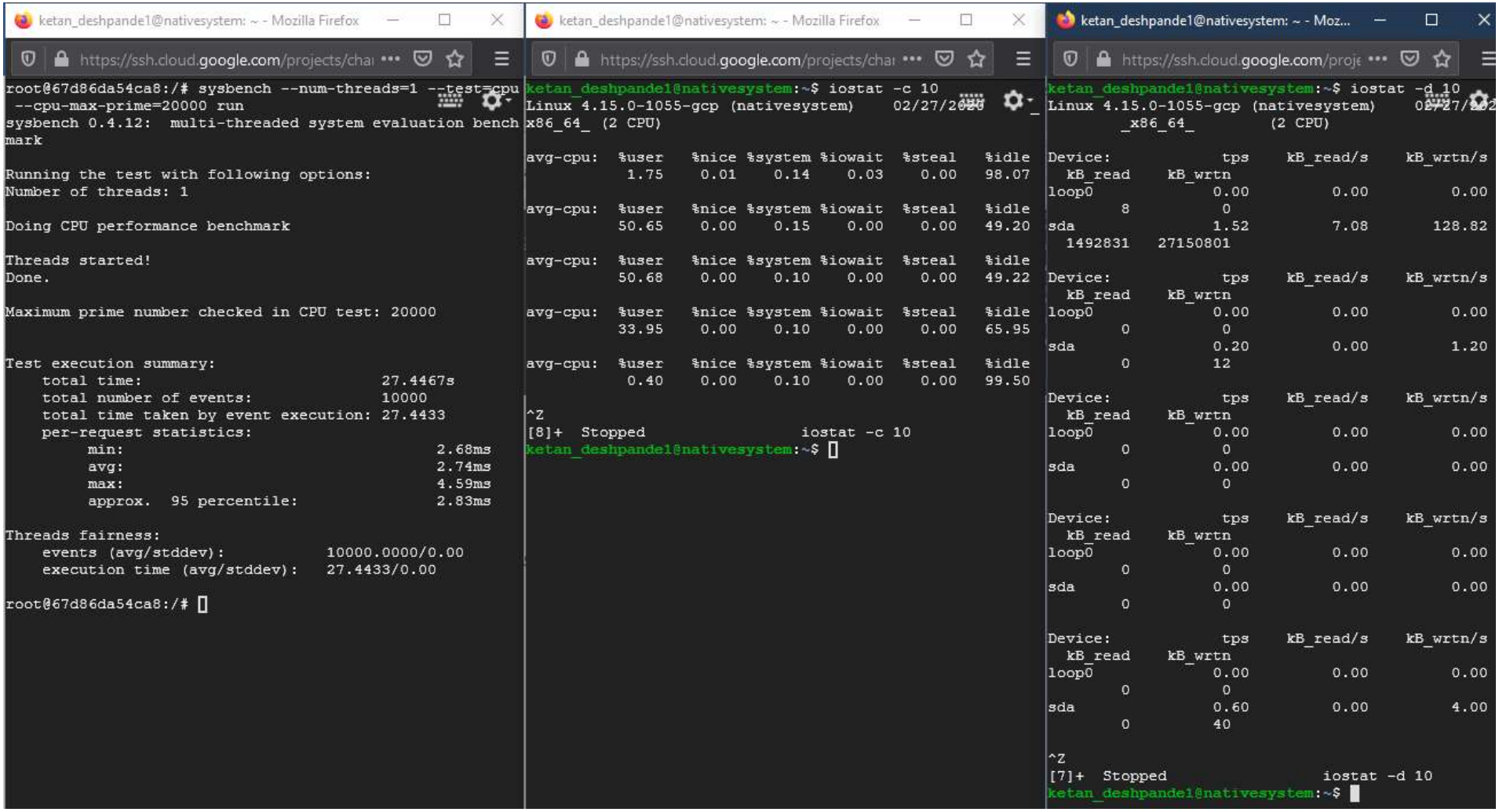
iostat -c 10

iostat -d 10

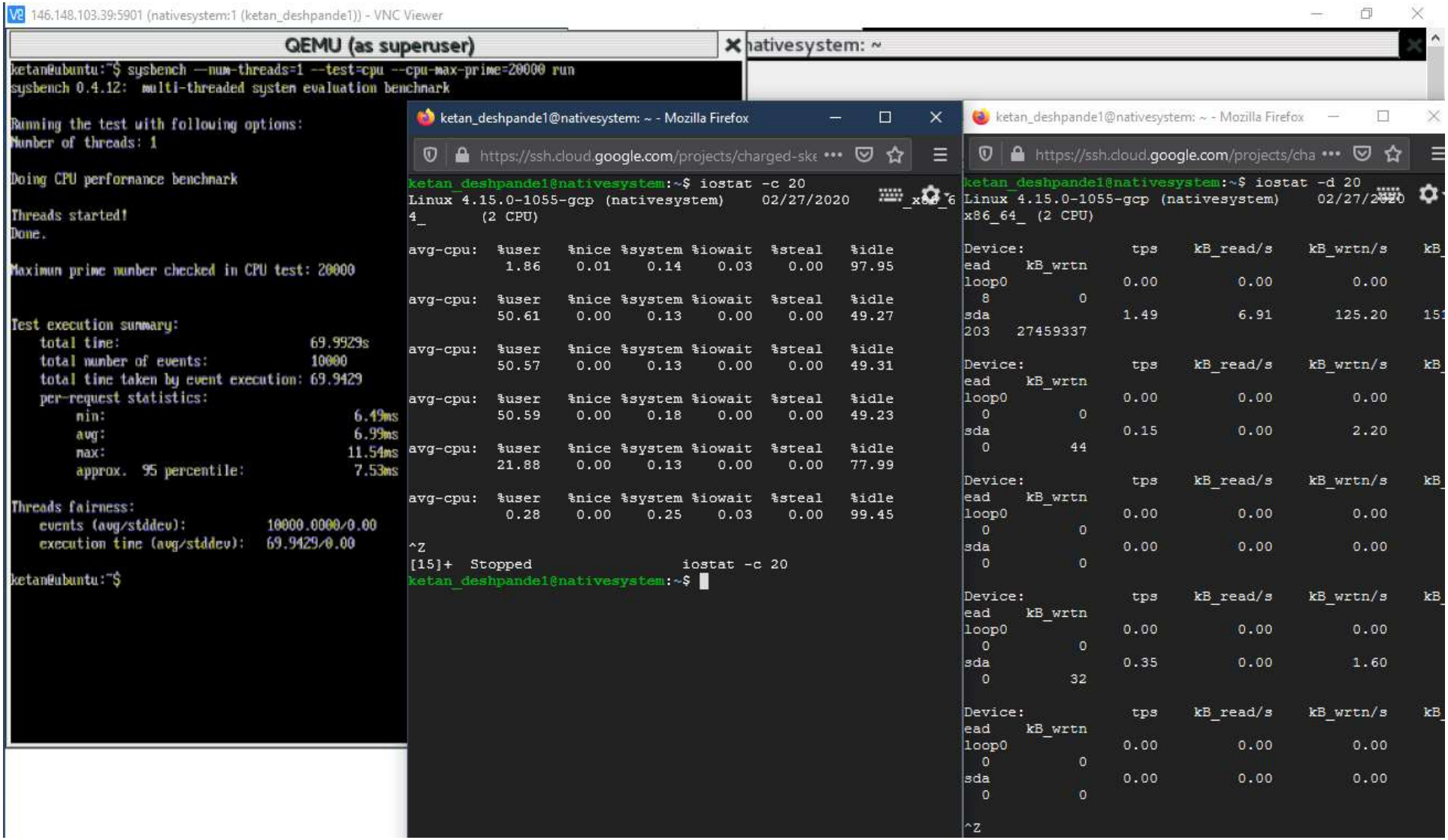
Native System: Total time taken: 28.16s



Docker: Total time taken: 27.44s



QEMU: Total time taken: 69.99s



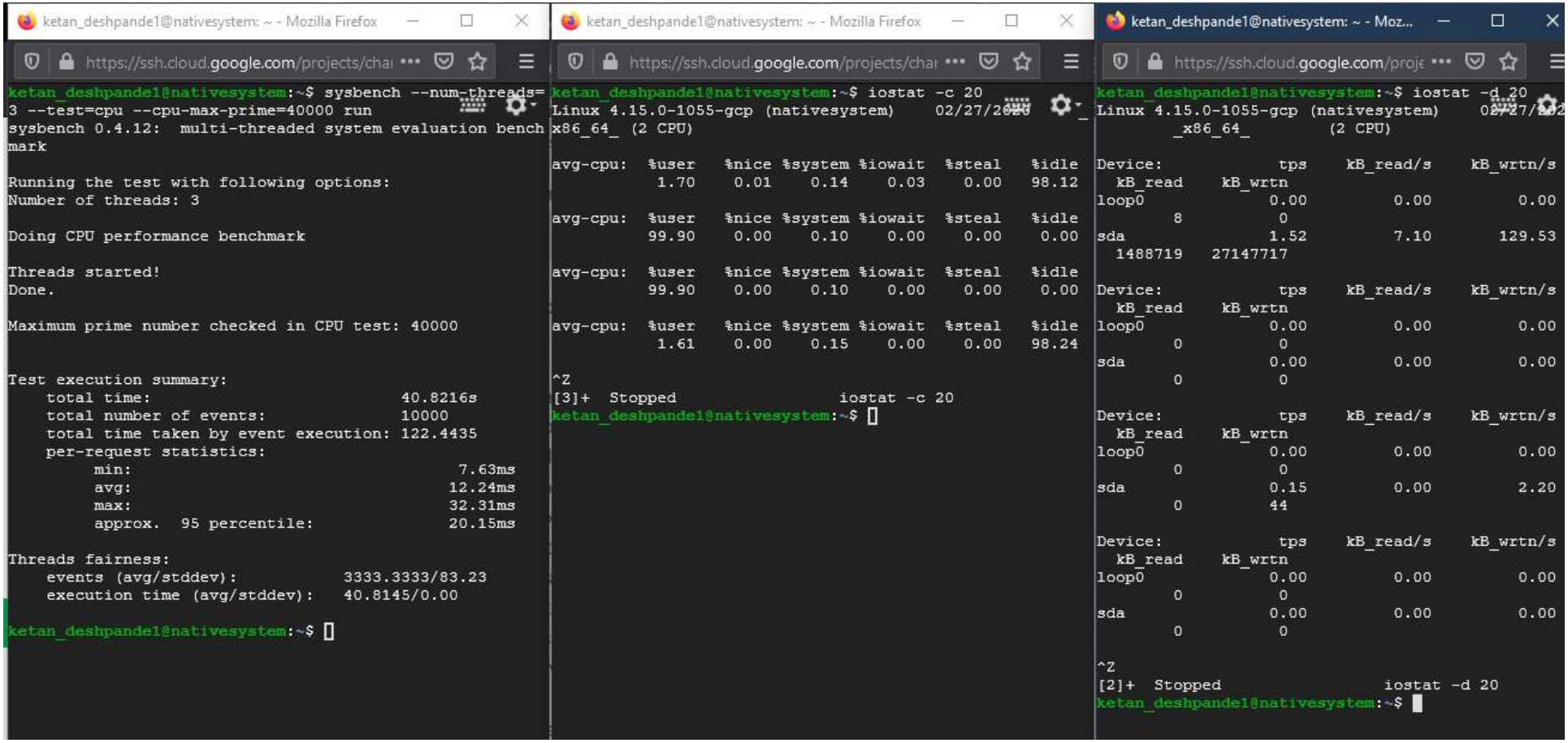
CPU Test 3

Command - sysbench --num-threads=3 --test=cpu --cpu-max-prime=40000 run

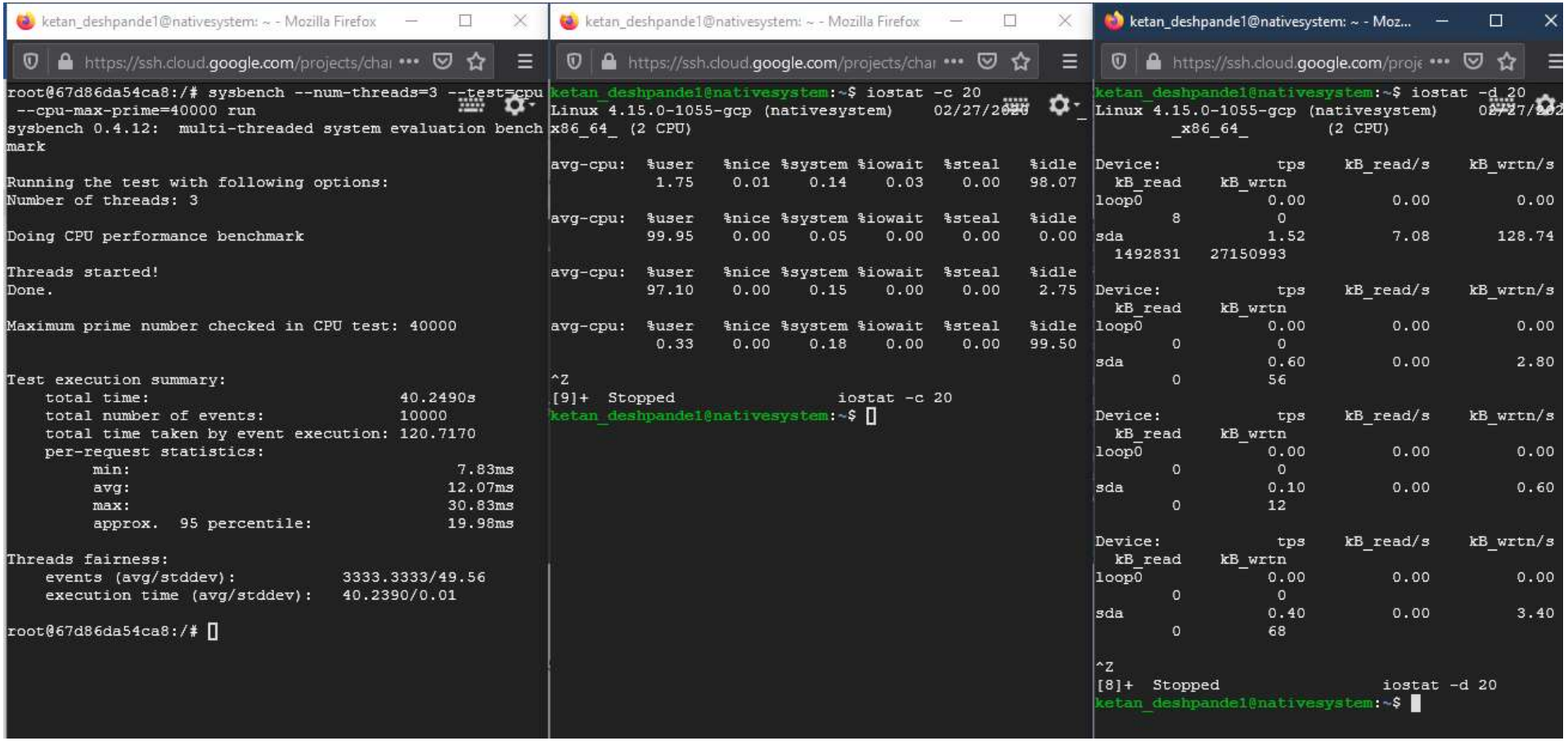
iostat -c 20

iostat -d 20

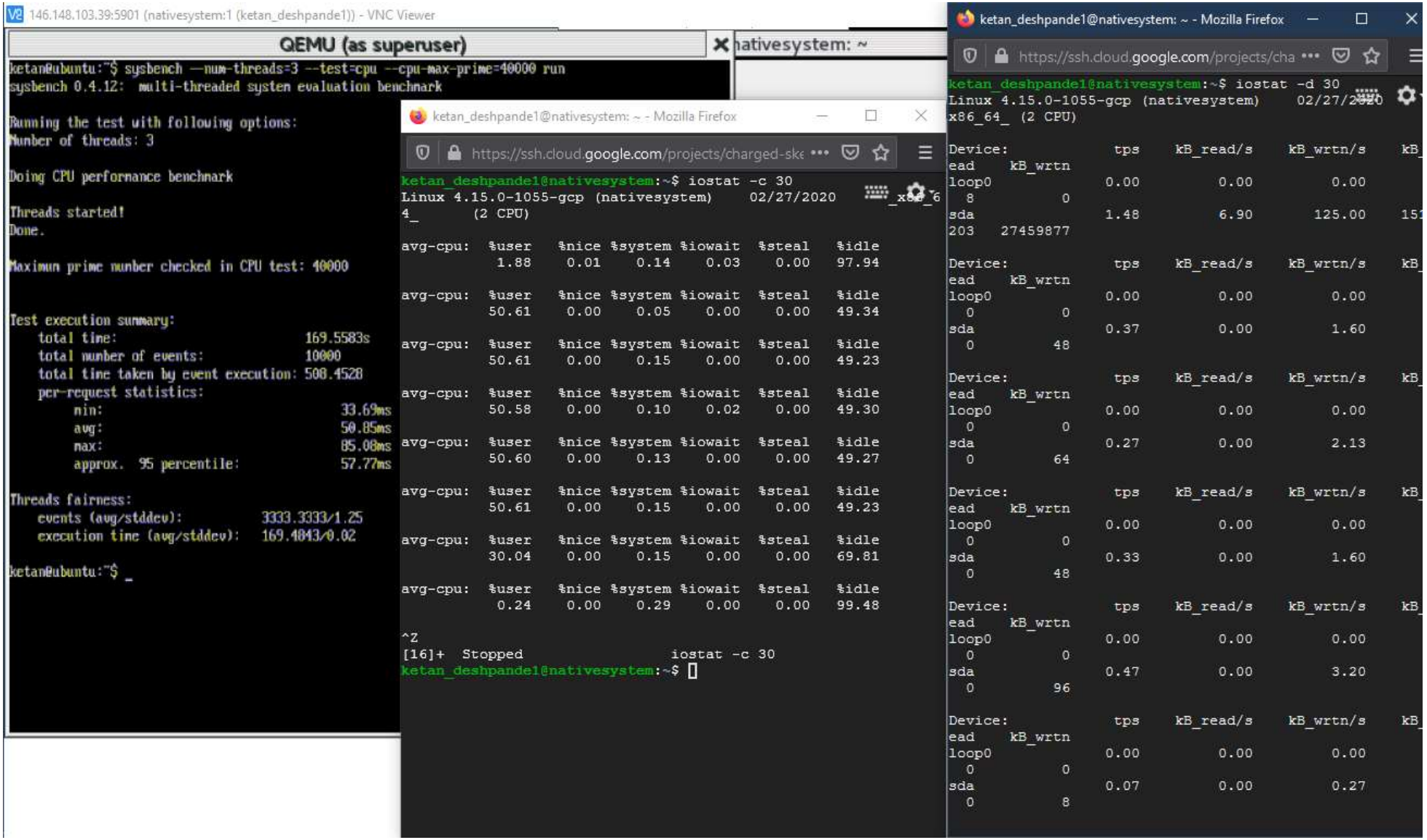
Native System: Total time taken: 40.82s



Docker: Total time taken: 40.24s



QEMU: Total time taken: 169.55s



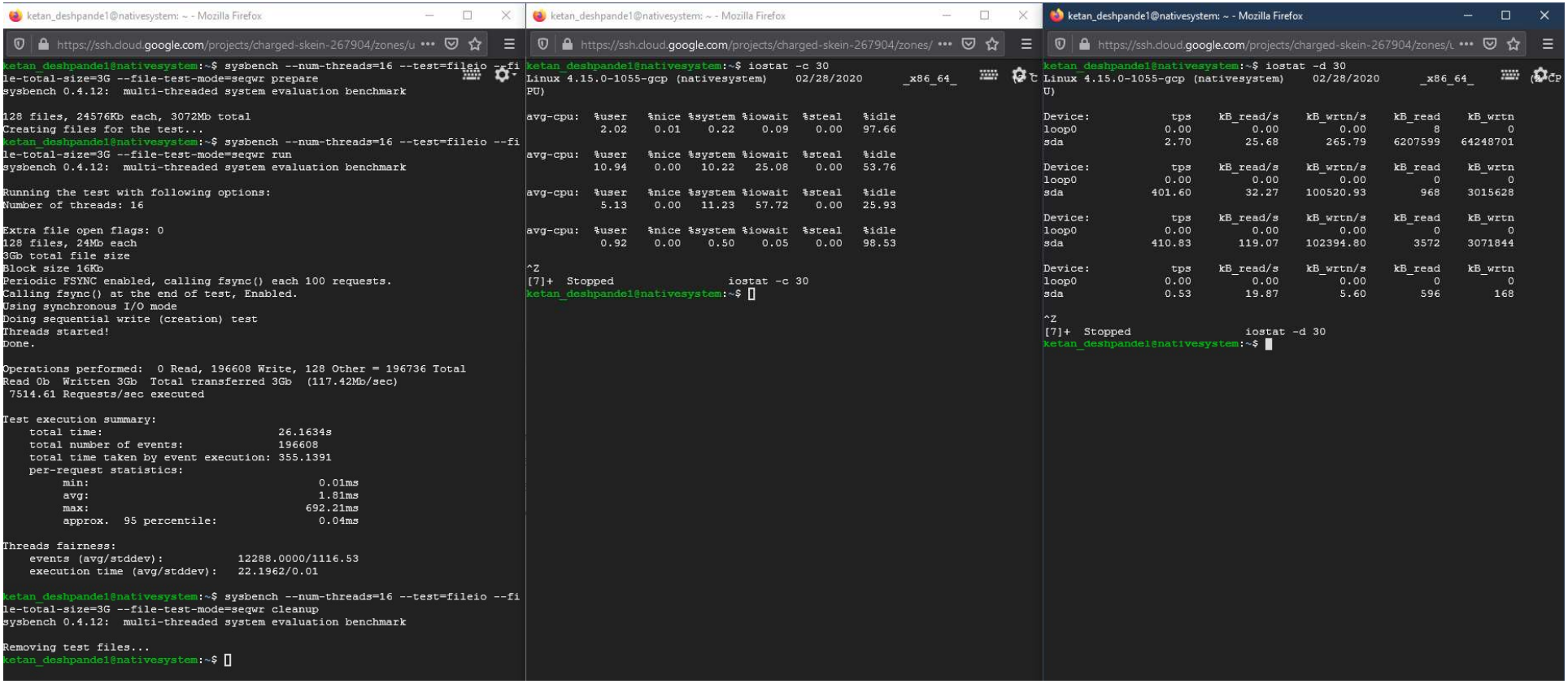
File IO Test 1

Commands - sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqwr prepare

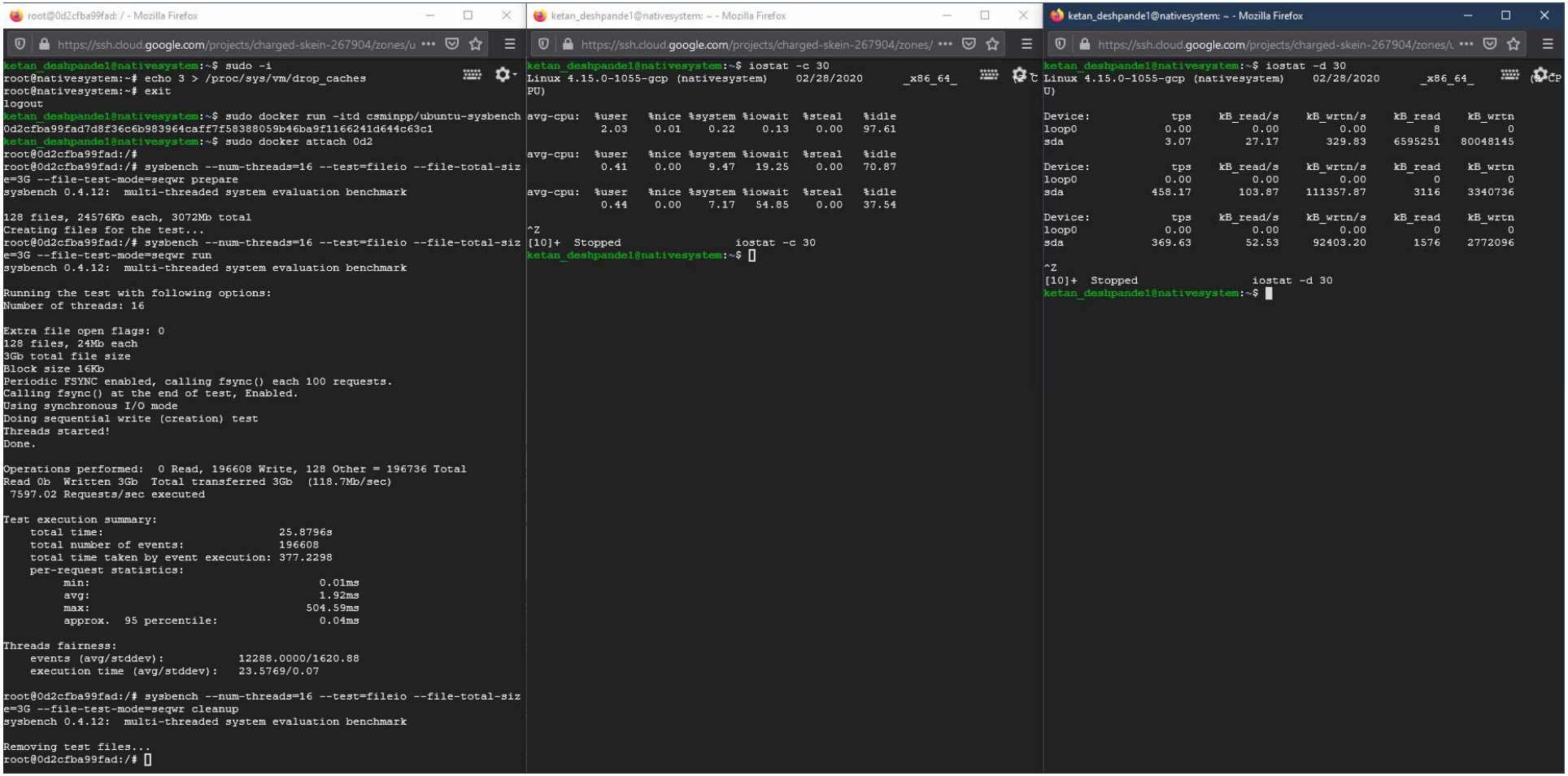
- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqwr run

- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqwr cleanup

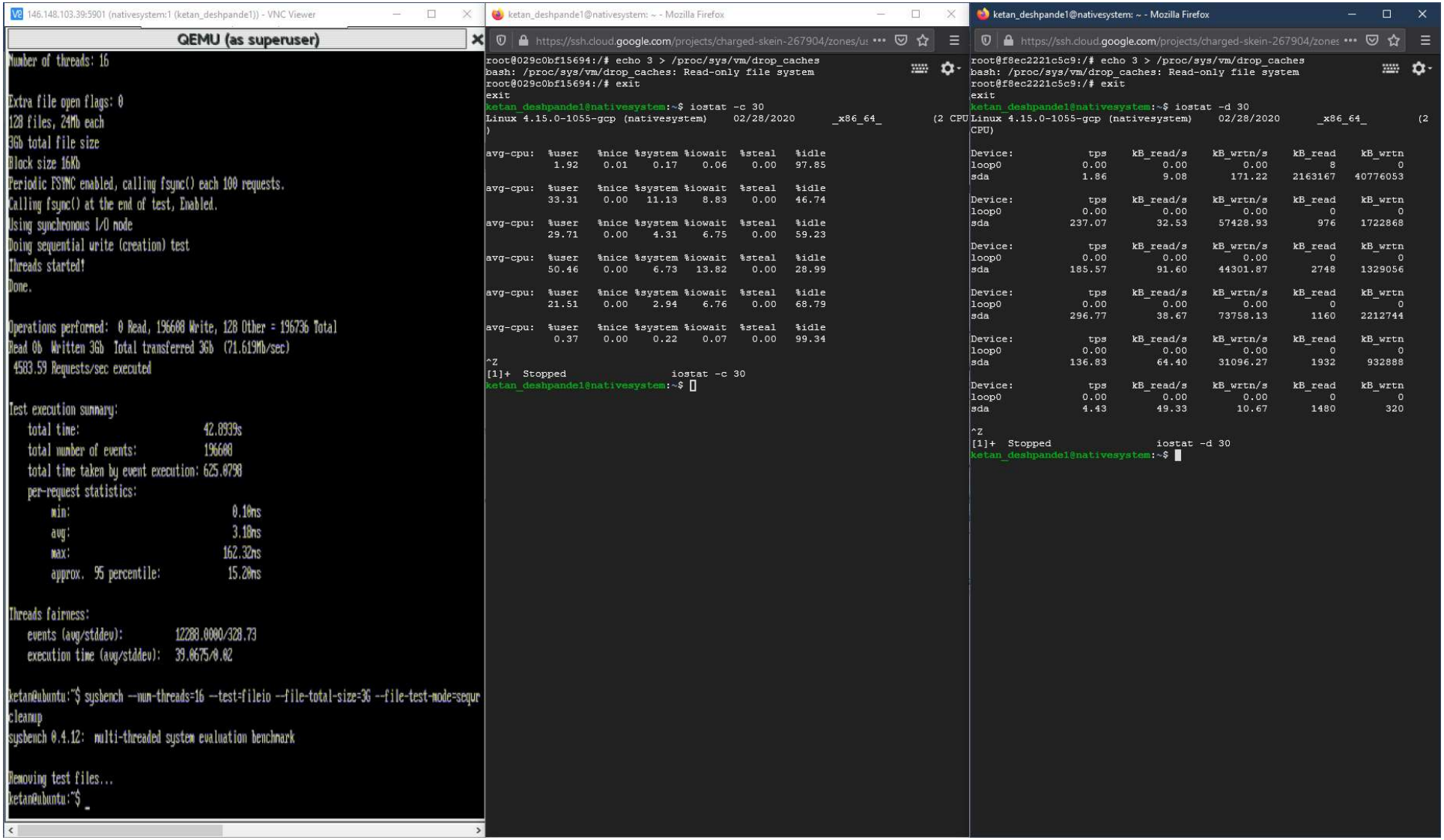
Native System: Total time taken: 26.16s



Docker: Total time taken: 25.87s



QEMU: Total time taken: 42.89s



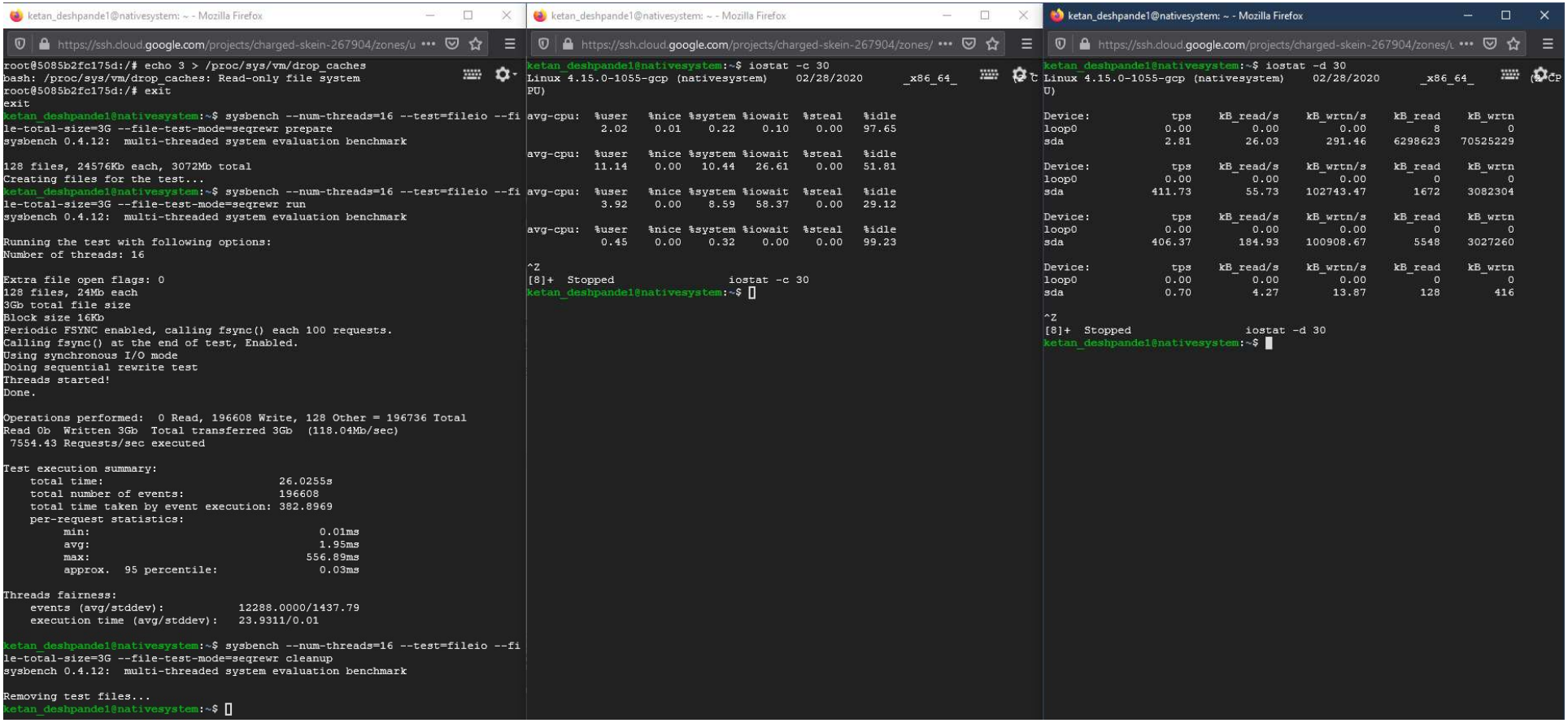
File IO Test 2

Commands - sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqrewr prepare

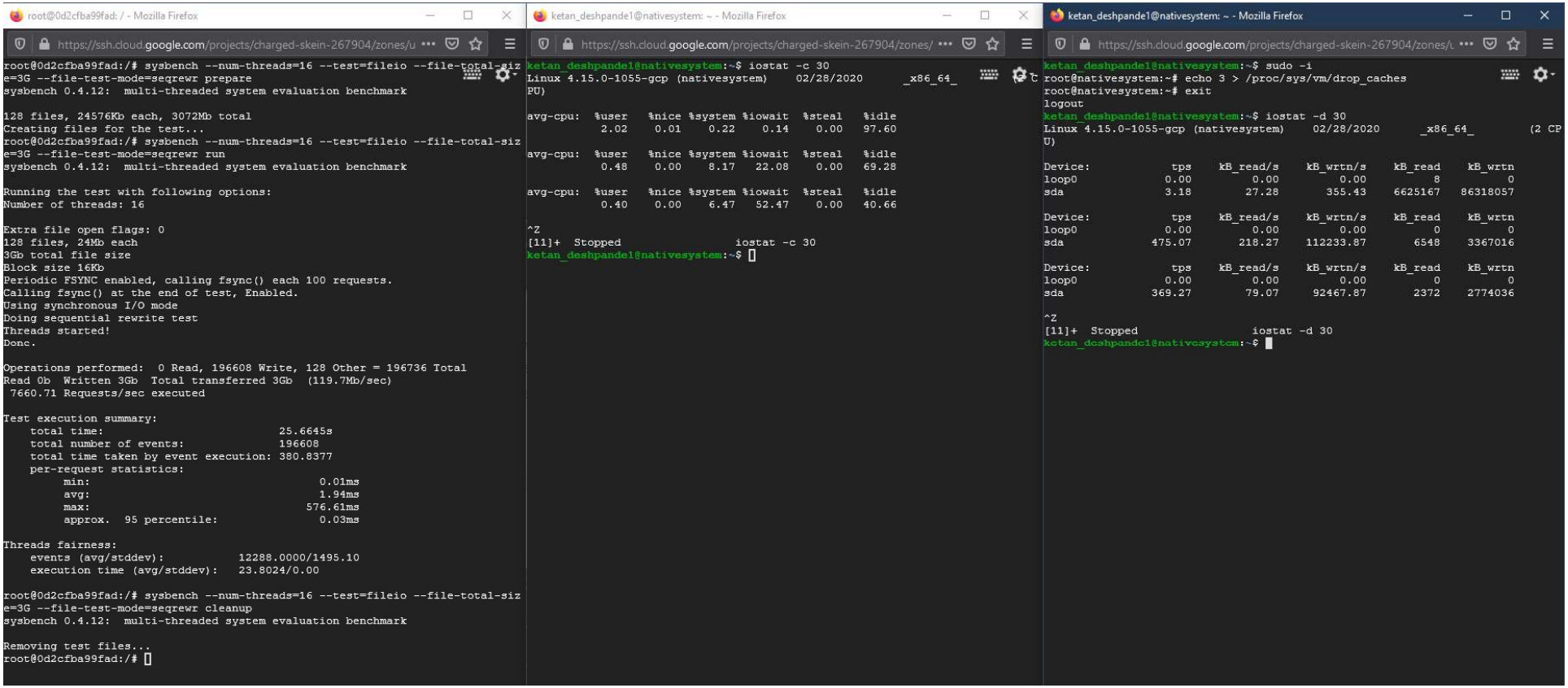
- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqrewr run

- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=seqrewr cleanup

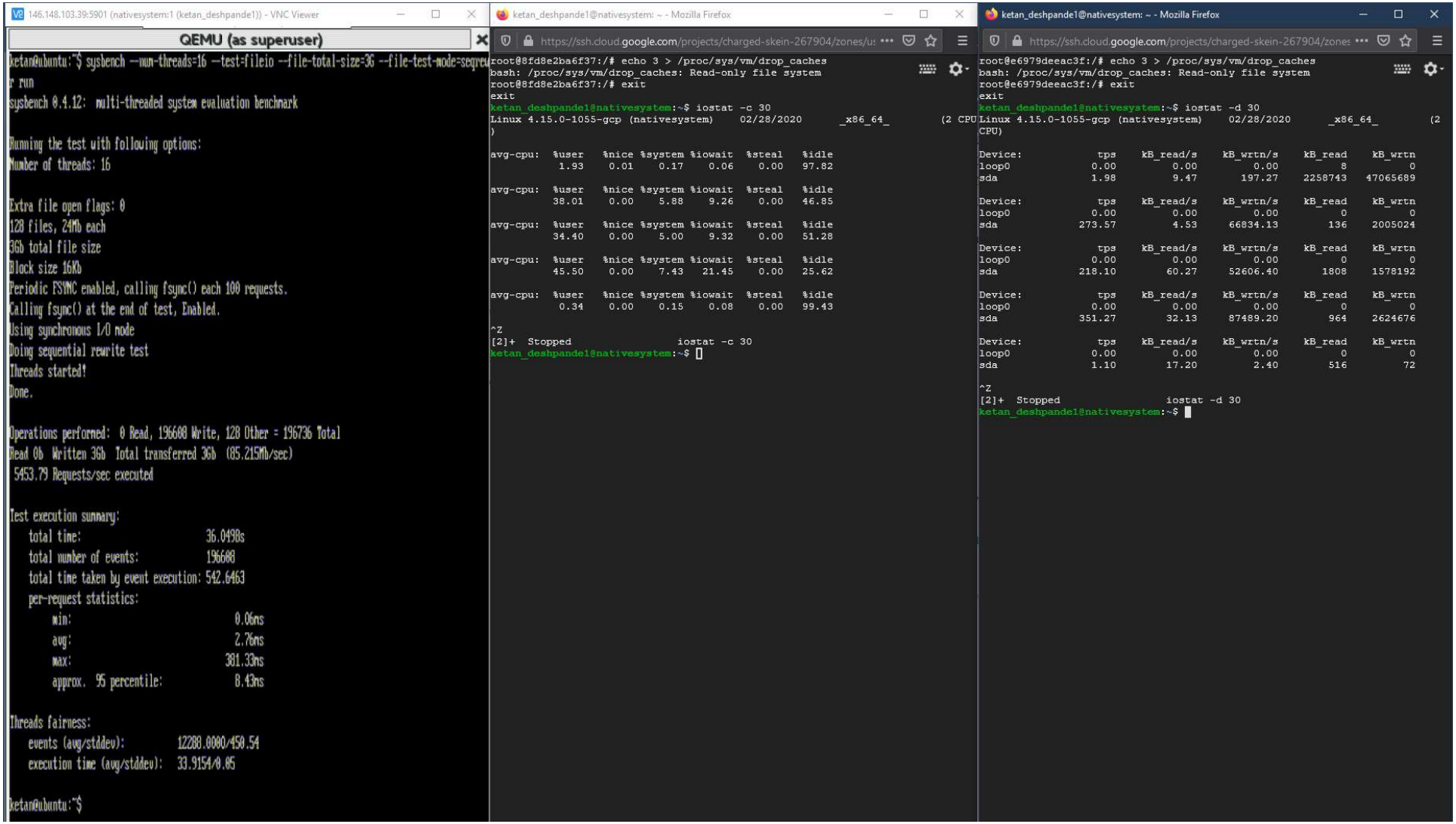
Native System: Total time taken: 26.02s



Docker: Total time taken: 25.66s



QEMU: Total time taken: 36.04s



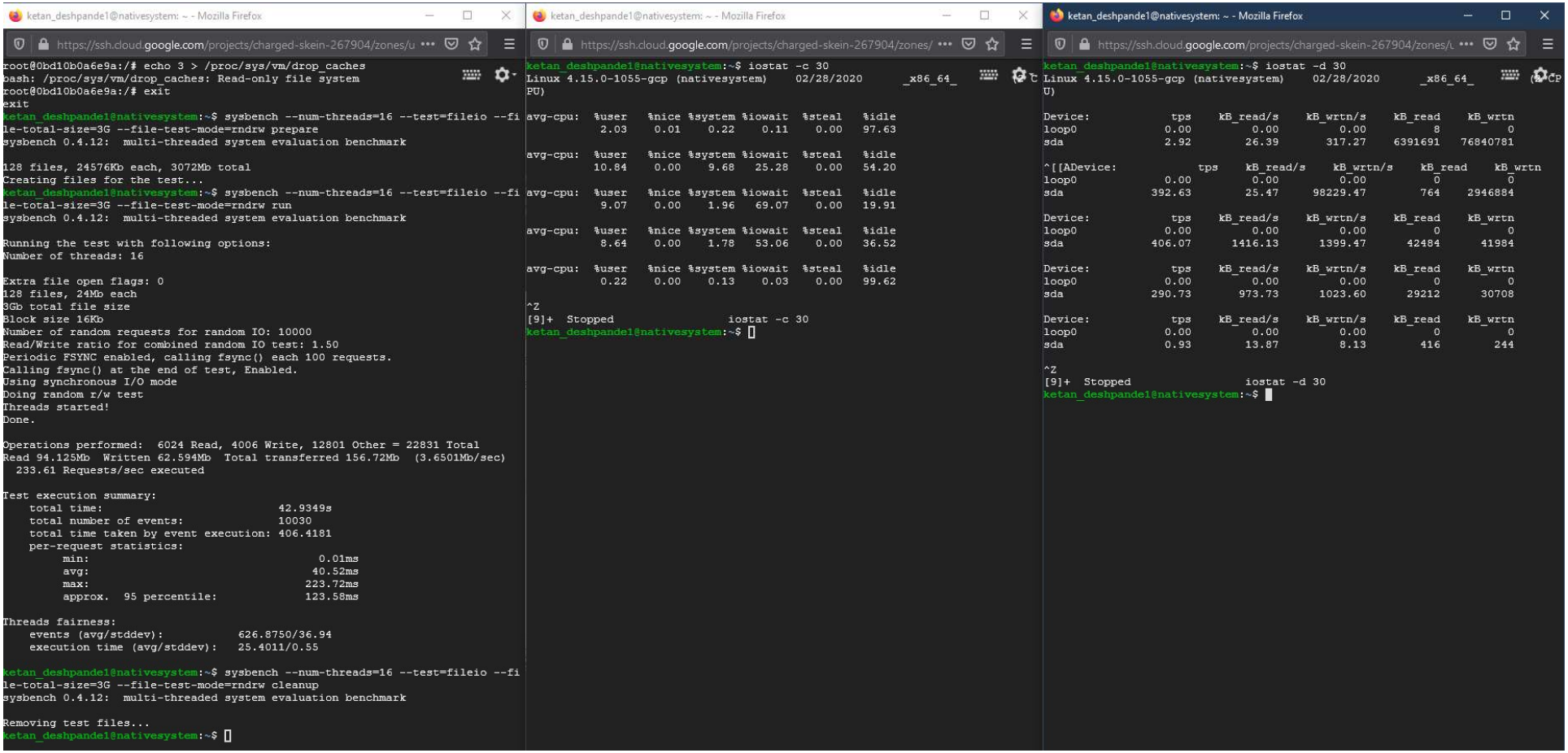
File IO Test 3

Commands - sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=rndrw prepare

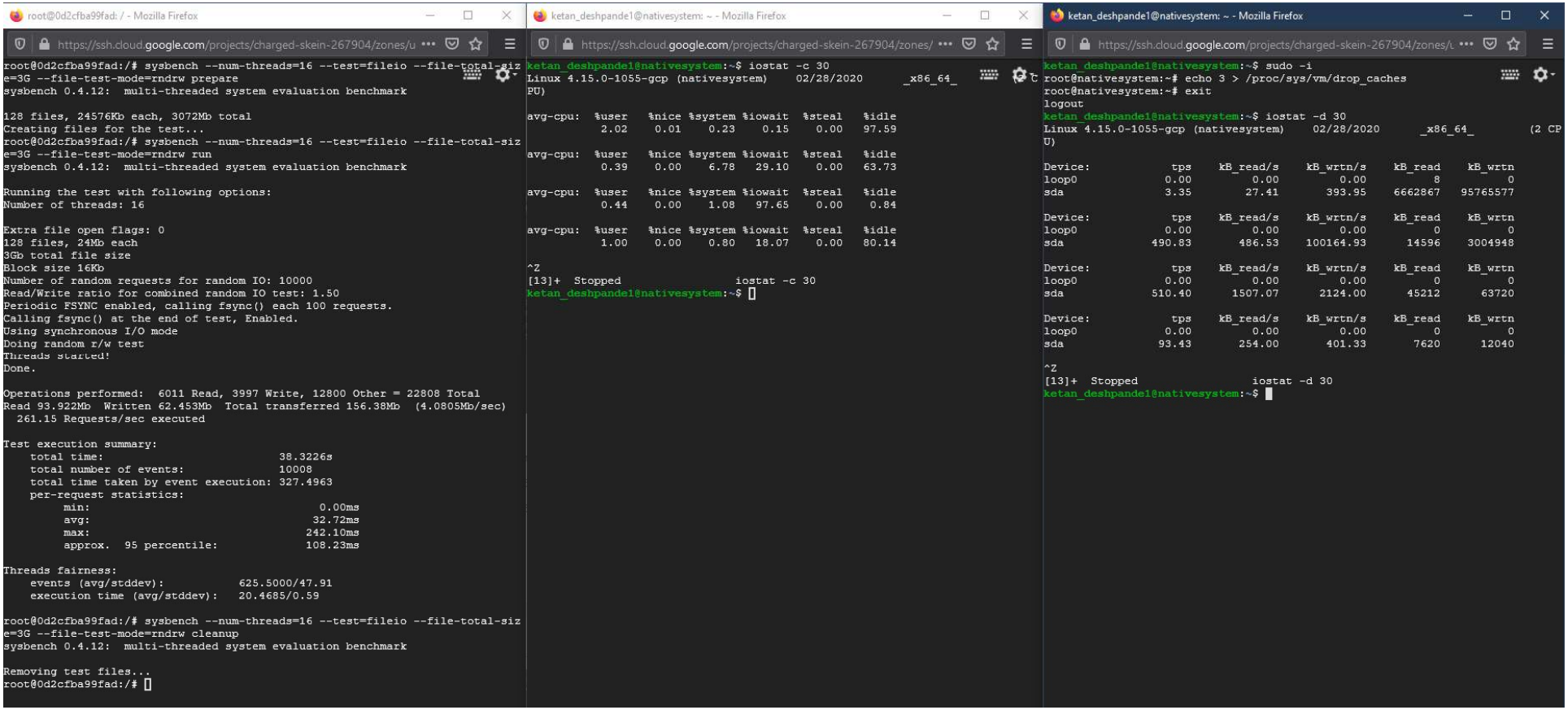
- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=rndrw run

- sysbench --num-threads=16 --test=fileio --file-total-size=3G --file-test-mode=rndrw cleanup

Native System: Total time taken: 42.93s



Docker: Total time taken: 38.32s



QEMU: Total time taken: 37.80s