

## Cloud Mini Project -1

B00818310:

### Docker management :

Setup and installation:

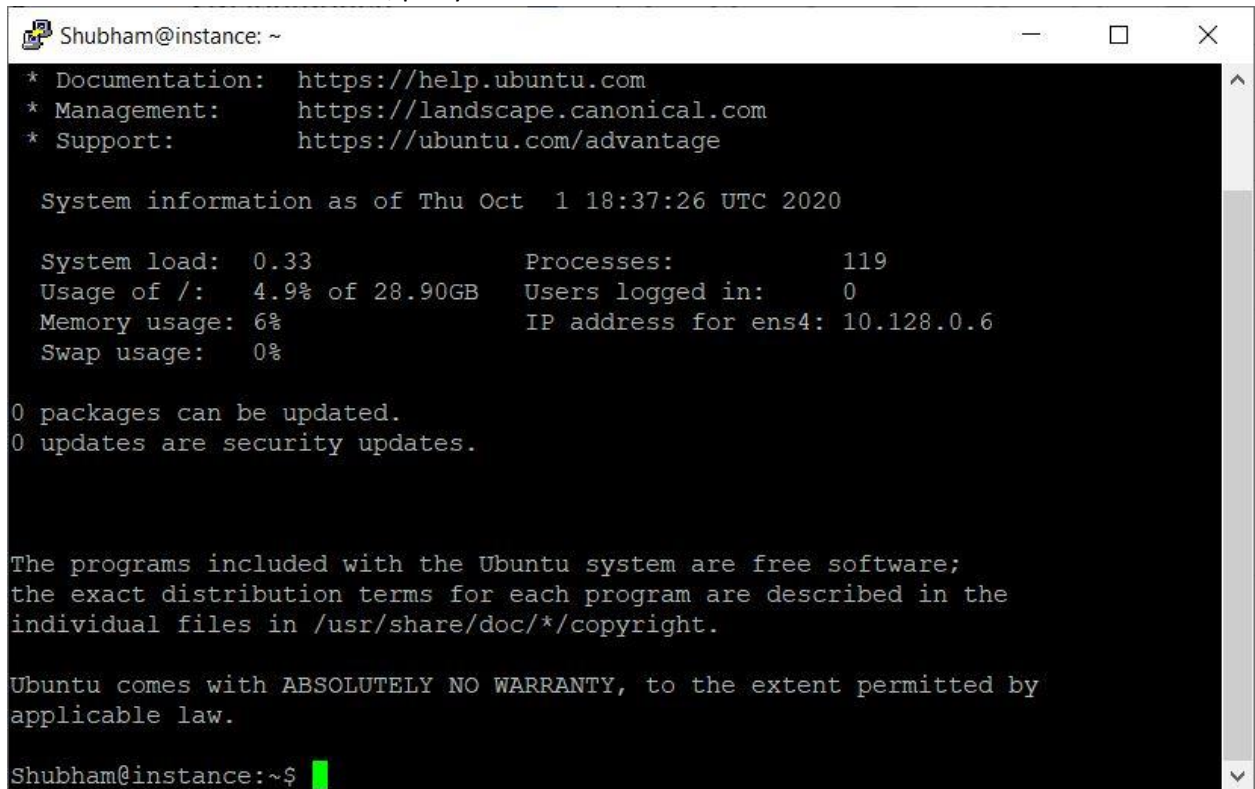
#### 1) Google instance setup and connection :

References:

[http://www.cs.binghamton.edu/~huilu/slidesfall2020/How\\_to\\_create\\_VM\\_in\\_GCP.pdf](http://www.cs.binghamton.edu/~huilu/slidesfall2020/How_to_create_VM_in_GCP.pdf)

[https://cloud.google.com/compute/docs/instances/create-start-instance#console\\_1](https://cloud.google.com/compute/docs/instances/create-start-instance#console_1)

- Firstly , the google instance was setup using 2 vCPU and 4GB memory with Ubuntu v18 as given.
- The google cloud SDK was installed since it was better than normal ssh and as well as browser connection.
- Once the google cloud shell was setup by configuring the default region and project, it was used to connect to the required google instance using the command: `gcloud compute ssh instance`
- Once the connection succeeded, putty was used for interaction.



```
Shubham@instance: ~
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu Oct  1 18:37:26 UTC 2020

System load:  0.33           Processes:            119
Usage of /:   4.9% of 28.90GB Users logged in:       0
Memory usage: 6%            IP address for ens4: 10.128.0.6
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Shubham@instance:~$
```

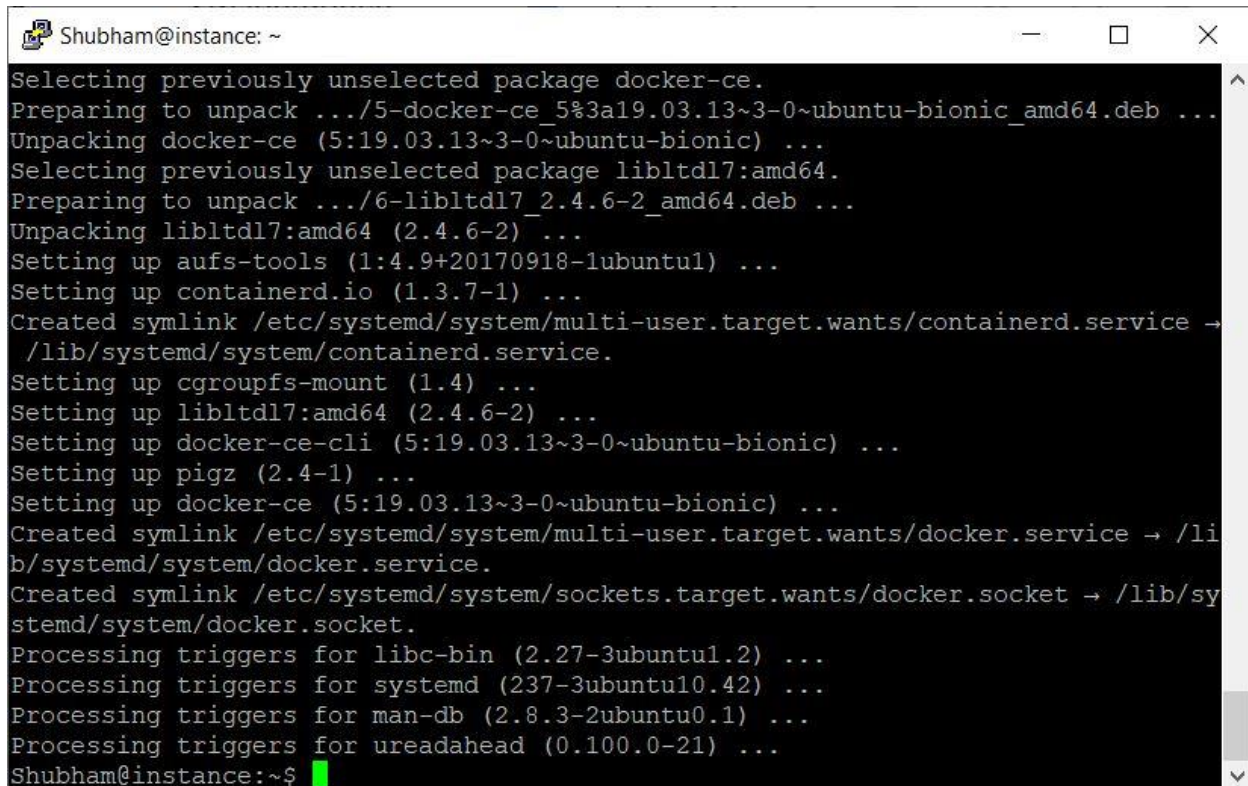
## 2) Docker installation inside google instance:

References:

<https://docs.docker.com/engine/install/ubuntu/>

<https://github.com/docker/labs/blob/master/beginner/chapters/alpine.md>

- Once google instance was setup, docker was installed using the given commands from the reference, one by one .
- Let's go through that one by one.
  - Firstly, some packages we setup to allow apt to use a repository over HTTPs.
  - Then, we Docker's official GPG key was added
  - It's fingerprint was verified
  - Then , finally the stable repository was setup.
- Now, for installation, docker engine was installed using :  
`sudo apt-get install docker-ce docker-ce-cli containerd.io`



```
Shubham@instance: ~
Selecting previously unselected package docker-ce.
Preparing to unpack .../5-docker-ce_5%3a19.03.13~3-0~ubuntu-bionic_amd64.deb ...
Unpacking docker-ce (5:19.03.13~3-0~ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../6-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.3.7-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
/lib/systemd/system/containerd.service.
Setting up cgroupfs-mount (1.4) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up docker-ce-cli (5:19.03.13~3-0~ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:19.03.13~3-0~ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy
stemd/system/docker.socket.
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu10.42) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Shubham@instance:~$
```

Then, csminpp/ubuntu-sysbench image was downloaded and I played with some commands over docker: csminpp/ubuntu-sysbench is a image for docker which comes with sysbench benchmark tool preinstalled.

```
root@instance: /home/Shubham
Digest: sha256:90fd06985472eec3aa99b665618c23f074deb326fcc87a5fb59d2be1f9d97435
Status: Downloaded newer image for csminpp/ubuntu-sysbench:latest
docker.io/csminpp/ubuntu-sysbench:latest
root@instance:/home/Shubham# service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Thu 2020-10-01 18:41:25 UTC; 5min ago
     Docs: https://docs.docker.com
   Main PID: 4037 (dockerd)
    Tasks: 13
   CGroup: /system.slice/docker.service
           └─4037 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contain

Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.067969278Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.067978058Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.068241962Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.205863996Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.293511160Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.335401005Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.335750508Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.369488749Z" le
Oct 01 18:41:25 instance systemd[1]: Started Docker Application Container Engine
Oct 01 18:46:00 instance dockerd[4037]: time="2020-10-01T18:46:00.424512852Z" le
lines 1-19/19 (END)
```

```
root@instance: /home/Shubham
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.293511160Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.335401005Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.335750508Z" le
Oct 01 18:41:25 instance dockerd[4037]: time="2020-10-01T18:41:25.369488749Z" le
Oct 01 18:41:25 instance systemd[1]: Started Docker Application Container Engine
Oct 01 18:46:00 instance dockerd[4037]: time="2020-10-01T18:46:00.424512852Z" le

root@instance:/home/Shubham# docker run csminpp/ubuntu-sysbench
root@instance:/home/Shubham# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
csminpp/ubuntu-sysbench  latest             2787c5e16909       4 years ago
336MB
root@instance:/home/Shubham# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
d8db2f53545d       csminpp/ubuntu-sysbench  "/bin/bash"        3 minutes ago
Exited (0) 3 minutes ago                                vibrant_napier
root@instance:/home/Shubham# docker run csminpp/ubuntu-sysbench
root@instance:/home/Shubham# docker run csminpp/ubuntu-sysbench
root@instance:/home/Shubham# docker run csminpp/ubuntu-sysbench echo "HELLO FROM
THE OTHER SIDE"
HELLO FROM THE OTHER SIDE
root@instance:/home/Shubham#
```

### 3) CPU and fileIO test cases:

References :

<https://imysql.com/wp-content/uploads/2014/10/sysbench-manual.pdf>

<https://linux.die.net/man/1/iostat>

<https://medium.com/@tejozarkar/simple-steps-for-connecting-to-google-cloud-vm-instance-through-mobaxterm-f1007d1d4fae>

- iostat was installed for measuring system performance(native).

```
root@instance:/home/Shubham
Command 'iostat' not found, but can be installed with:

apt install sysstat

root@instance:/home/Shubham# apt install sysstat
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  grub-pc-bin libnuma1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libsensors4
Suggested packages:
  lm-sensors isag
The following NEW packages will be installed:
  libsensors4 sysstat
0 upgraded, 2 newly installed, 0 to remove and 9 not upgraded.
Need to get 324 kB of archives.
After this operation, 1309 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-central1-gce.archive.ubuntu.com/ubuntu bionic/main amd64 libsensors4 amd64 1:3.4.0-4 [28.8 kB]
Get:2 http://us-central1-gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 sysstat amd64 11.6.1-lubuntu0.1 [295 kB]
Fetched 324 kB in 0s (4272 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libsensors4:amd64.
(Reading database ... 65787 files and directories currently installed.)
Preparing to unpack .../libsensors4_1:3.4.0-4_amd64.deb ...
Unpacking libsensors4:amd64 (1:3.4.0-4) ...
Selecting previously unselected package sysstat.
Preparing to unpack .../sysstat_11.6.1-lubuntu0.1_amd64.deb ...
Unpacking sysstat (11.6.1-lubuntu0.1) ...
Setting up libsensors4:amd64 (1:3.4.0-4) ...
Setting up sysstat (11.6.1-lubuntu0.1) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in
 auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /l
ib/systemd/system/sysstat.service.
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Processing triggers for systemd (237-3ubuntu1.0.42) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@instance:/home/Shubham#
```

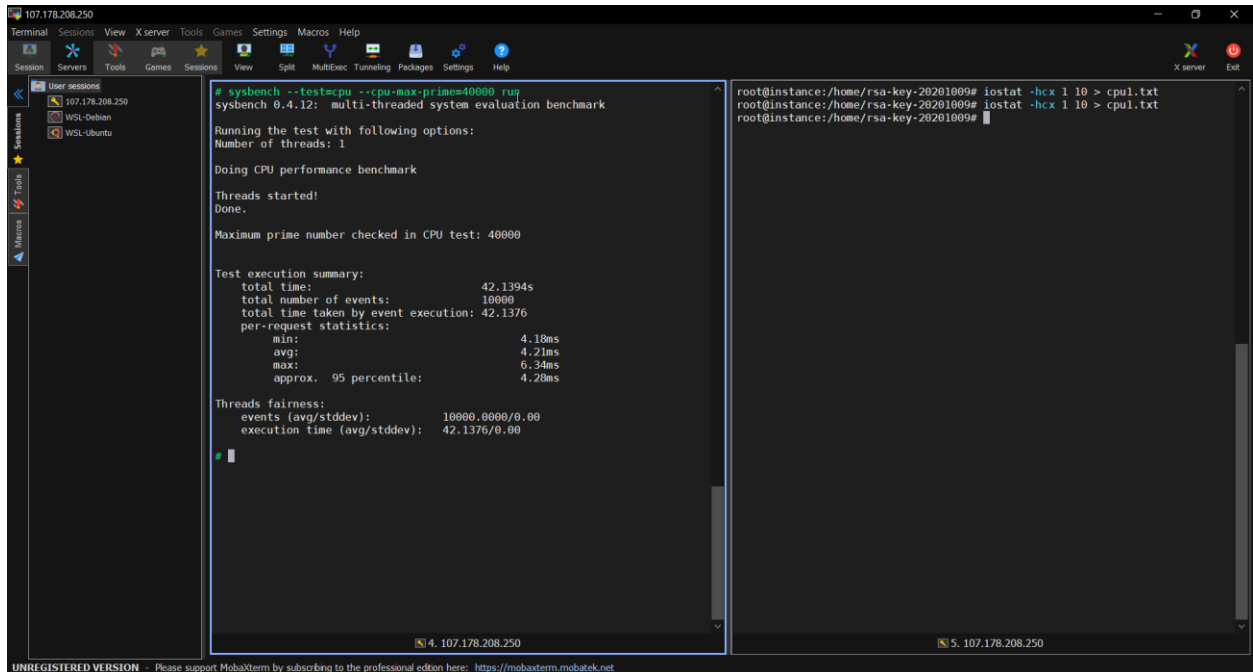
For CPU test, following command was used : sysbench --test=cpu --cpu-max-prime=40000 run

After some tests , '40000' was taken, since it generated time near to 42 secs.

Also, instance was connected using mobaxterm, since it was better for side by side viewing and taking better screenshots.

- Simultaneously, iostat was run in another terminal using the command :
  - iostat -hcx
  - -h : Make the NFS report displayed by option -n easier to read by a human.
  - -c : Display the CPU utilization report.
  - -x : Display extended statistics.
  - It was then all logged into a txt file .

## CPU Test 1:



The screenshot shows the MobaXterm interface with two terminal windows. The left window displays the output of the sysbench CPU test, and the right window shows the output of the iostat command.

```
# sysbench --test=cpu --cpu-max-prime=40000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

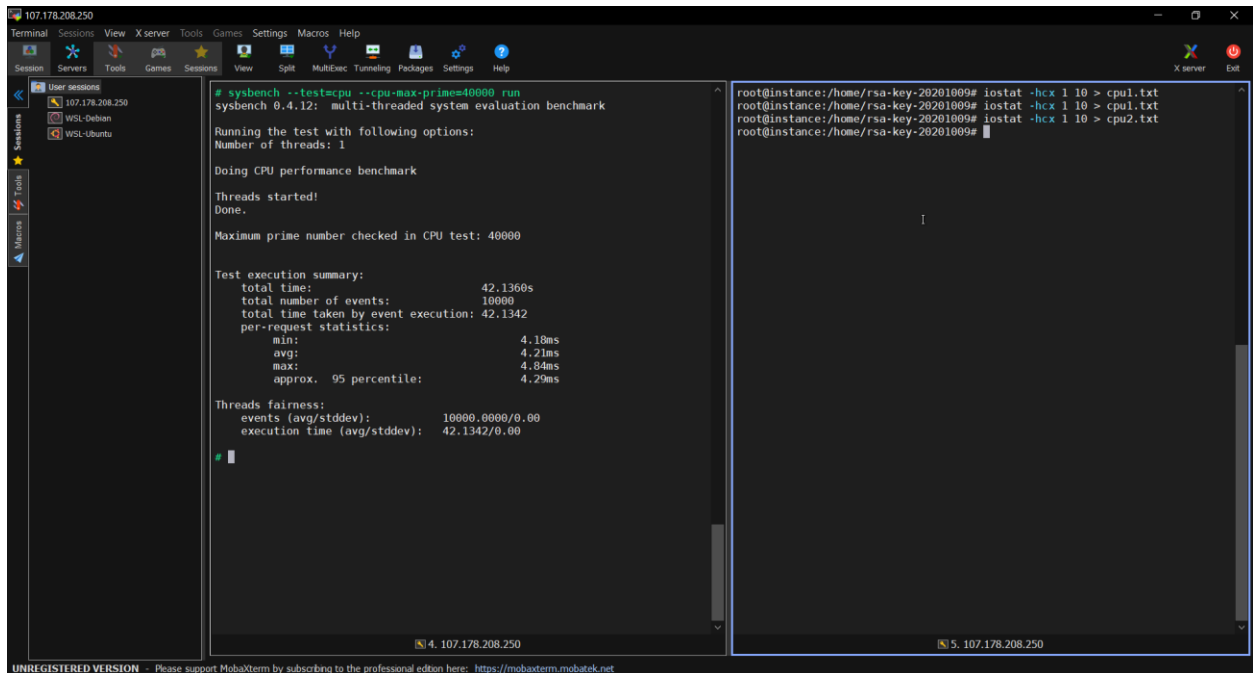
Maximum prime number checked in CPU test: 40000

Test execution summary:
total time: 42.1394s
total number of events: 10000
total time taken by event execution: 42.1376
per-request statistics:
  min: 4.18ms
  avg: 4.21ms
  max: 6.34ms
  approx. 95 percentile: 4.28ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 42.1376/0.00
```

```
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009#
```

- CPU Test 2:



The screenshot shows the MobaXterm interface with two terminal windows. The left window displays the output of the sysbench CPU test, and the right window shows the output of the iostat command.

```
# sysbench --test=cpu --cpu-max-prime=40000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 40000

Test execution summary:
total time: 42.1360s
total number of events: 10000
total time taken by event execution: 42.1342
per-request statistics:
  min: 4.18ms
  avg: 4.21ms
  max: 4.84ms
  approx. 95 percentile: 4.29ms

Threads fairness:
  events (avg/stddev): 10000.0000/0.00
  execution time (avg/stddev): 42.1342/0.00
```

```
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu2.txt
root@instance:/home/rsa-key-20201009#
```

- CPU Test 3:

The screenshot displays the MobaXterm interface with two terminal windows. The left window, titled '107.178.208.250', shows the execution of the `sysbench` CPU benchmark. The command used is `sysbench 0.4.12: multi-threaded system evaluation benchmark` with options `--test=cpu --cpu-max-prime=40000 run`. The output indicates that the test was run with 1 thread and a maximum prime number of 40,000. The test execution summary shows a total time of 42.1999s, 10,000 events, and a total time taken by event execution of 42.1982s. The per-request statistics show a minimum of 4.18ms, an average of 4.22ms, a maximum of 6.28ms, and an approximate 95th percentile of 4.31ms. The threads fairness is also reported.

```
# sysbench --test=cpu --cpu-max-prime=40000 run
sysbench 0.4.12: multi-threaded system evaluation benchmark

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

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 40000

Test execution summary:
total time:                42.1999s
total number of events:    10000
total time taken by event execution: 42.1982
per-request statistics:
  min:                     4.18ms
  avg:                     4.22ms
  max:                     6.28ms
  approx. 95 percentile:   4.31ms

Threads fairness:
  events (avg/stddev):      10000.0000/0.00
  execution time (avg/stddev): 42.1982/0.00
```

The right window, also titled '107.178.208.250', shows the execution of the `iostat` command to monitor disk I/O. The command used is `iostat -hcx 1 10 > cpu1.txt`. The output shows the disk I/O statistics for the system.

```
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu1.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu2.txt
root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > cpu3.txt
root@instance:/home/rsa-key-20201009#
```



- file IO: same procedure was done.
- We go into the docker image shell
- Then , we run the commands for rndrw mode and iostat while running the tests.

Commands :

`sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test-mode=rndrw prepare`

`sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test-mode=rndrw run`

`sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test-mode=rndrw cleanup`

Then , after each test , cache needs to be cleared from native using :

`echo 3 > /proc/sys/vm/drop_caches`

- Also, using the command `df -h`, disk utilisation was check before preparing the files, after running the test and after cleaning up the files.

- File IO test 1:

The screenshot shows a MobaXterm terminal window with two panes. The left pane shows the execution of sysbench file IO tests. The right pane shows the output of the `df -h` command before and after the tests.

**Left Pane Output:**

```
# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 128Mb each
15Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
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 r/w test
Threads started!
Done.

Operations performed: 6016 Read, 4001 Write, 12000 Other = 22017 Total
Read 94Mb Written 62.510Mb Total transferred 156.52Mb (4.8707Mb/sec)
311.72 Requests/sec executed

Test execution summary:
total time: 32.1343s
total number of events: 10017
total time taken by event execution: 386.7444
per-request statistics:
  min: 0.00ms
  avg: 38.61ms
  max: 251.58ms
  approx. 95 percentile: 134.42ms

Threads fairness:
  events (avg/stddev): 626.0625/43.31
  execution time (avg/stddev): 24.1715/0.43

# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark
```

**Right Pane Output:**

```
root@instance:/home/rsa-key-20201009# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0 2.0G   0% /dev
tmpfs           394M 1012K 393M   1% /run
/dev/sda1       29G   2.6G  27G   9% /
tmpfs           2.0G   0 2.0G   0% /dev/shm
tmpfs           5.0M   0 5.0M   0% /run/lock
tmpfs           2.0G   0 2.0G   0% /sys/fs/cgroup
/dev/loop0      56M   56M   0 100% /snap/core18/1885
/dev/loop1     31M   31M   0 100% /snap/snapd/9279
/dev/loop2     127M  127M   0 100% /snap/google-cloud-sdk/151
/dev/sda15     105M   3.6M  101M   4% /boot/efi
/dev/loop3     128M  128M   0 100% /snap/google-cloud-sdk/154
tmpfs          394M   0 394M   0% /run/user/1002
overlay        29G   2.6G  27G   9% /var/lib/docker/overlay2/378ebc9daa94951cfe9a8c403a2169db0079724a7c15f90f402c60bfd4e05cc9/merged

root@instance:/home/rsa-key-20201009# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0 2.0G   0% /dev
tmpfs           394M 1012K 393M   1% /run
/dev/sda1       29G   18G  12G  61% /
tmpfs           2.0G   0 2.0G   0% /dev/shm
tmpfs           5.0M   0 5.0M   0% /run/lock
tmpfs           2.0G   0 2.0G   0% /sys/fs/cgroup
/dev/loop0      56M   56M   0 100% /snap/core18/1885
/dev/loop1     31M   31M   0 100% /snap/snapd/9279
/dev/loop2     127M  127M   0 100% /snap/google-cloud-sdk/151
/dev/sda15     105M   3.6M  101M   4% /boot/efi
/dev/loop3     128M  128M   0 100% /snap/google-cloud-sdk/154
tmpfs          394M   0 394M   0% /run/user/1002
overlay        29G   18G  12G  61% /var/lib/docker/overlay2/378ebc9daa94951cfe9a8c403a2169db0079724a7c15f90f402c60bfd4e05cc9/merged

root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > fileiol.txt
root@instance:/home/rsa-key-20201009# echo 3 > /proc/sys/vm/drop_caches
root@instance:/home/rsa-key-20201009#
```

- File IO test 2:

```

# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test
-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 120Mb each
15Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
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 r/w test
Threads started!
Done.

Operations performed: 6016 Read, 4004 Write, 12000 Other = 22020 Total
Read 94Mb Written 62.562Mb Total transferred 156.56Mb (4.932Mb/sec)
315.65 Requests/sec executed

Test execution summary:
total time: 31.7441s
total number of events: 10020
total time taken by event execution: 376.5449
per-request statistics:
min: 0.01ms
avg: 37.58ms
max: 273.07ms
approx. 95 percentile: 135.27ms

Threads fairness:
events (avg/stddev): 626.2500/35.56
execution time (avg/stddev): 23.5341/0.51

# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test
-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark
  
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

- FileIO Test 3:

```

# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test
-mode=rndrw run
sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 120Mb each
15Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
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 r/w test
Threads started!
Done.

Operations performed: 6024 Read, 4006 Write, 12001 Other = 22031 Total
Read 94.125Mb Written 62.594Mb Total transferred 156.72Mb (4.803Mb/sec)
307.39 Requests/sec executed

Test execution summary:
total time: 32.6294s
total number of events: 10030
total time taken by event execution: 394.9717
per-request statistics:
min: 0.00ms
avg: 39.38ms
max: 278.44ms
approx. 95 percentile: 136.80ms

Threads fairness:
events (avg/stddev): 626.8750/43.83
execution time (avg/stddev): 24.6857/0.60

# sysbench --num-threads=16 --test=fileio --file-total-size=15G --file-test
-mode=rndrw cleanup
sysbench 0.4.12: multi-threaded system evaluation benchmark
  
```

```

root@instance:/home/rsa-key-20201009# iostat -hcx 1 10 > fileio3.txt
root@instance:/home/rsa-key-20201009# echo 3 > /proc/sys/vm/drop_caches
root@instance:/home/rsa-key-20201009# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            2.0G   0  2.0G   0% /dev
tmpfs           394M 1012K 393M   1% /run
/dev/sda1       29G   2.6G   27G   9% /
tmpfs           2.0G   0  2.0G   0% /dev/shm
tmpfs           5.0M   0   5.0M   0% /run/lock
tmpfs           2.0G   0  2.0G   0% /sys/fs/cgroup
/dev/loop0      56M   56M   0 100% /snap/core18/1885
/dev/loop1      31M   31M   0 100% /snap/snapd/9279
/dev/loop2     127M  127M   0 100% /snap/google-cloud-sdk/151
/dev/sda15     105M   3.6M  101M   4% /boot/efi
/dev/loop3     128M  128M   0 100% /snap/google-cloud-sdk/154
tmpfs          394M   0  394M   0% /run/user/1002
overlay        20G   2.6G   27G   9% /var/lib/docker/overlay2/378ebc9daa94951cfe9a8c403a2169db0079724a7c15f90f402c60bf04e05cc9/merged
root@instance:/home/rsa-key-20201009#
  
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>



## QEMU installation :

### References :

<https://qemu.weilnetz.de/doc/>

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

- First go into root using : `sudo su`
- Run : `sudo apt-get install qemu`
- Download ubuntu iso using :  
`wget http://mirror.pnl.gov/releases/18.04/ubuntu-18.04.5-live-server-amd64.iso`
- Run : `sudo qemu-img create ubuntu.img 10G`  
To format the image (create) before installing the QEMU VM
- Run : `$ sudo apt-get install gnome-core`  
This is to install the gnome components of our virtual desktop
- Since we need to interact with our virtual environment, we have to do it with vncserver,  
To install, run this command:  
`$ sudo apt-get install vnc4server`
- Setup a password for future connection, run :  
`$ vncserver`
- The vncserver is now listening on port 5901, we can verify it by running,  
`$ nc localhost 5901`  
And check if we get the following response : RFB 003.008
- A tweak must be made to the startup script to enable the gui properly .So, lets kill the server first using:  
`$ vncserver -kill :1`
- Edit the file using :  
`$ vim .vnc/xstartup`
- Replace the whole content to the given content in the question statement.
- Then , we have add firewall rule in our google instance for accepting tcp connection for port 5901.
- Once that is done, we can start our server using `$ vncserver`

I have used realvnc to connect to the server.

Just use the “externalip:5901” to connect to the server.

```
root@instance: /home/Shubham
Shubham@instance:~$ sudo su
root@instance:/home/Shubham#
```

```
Shubham@instance: ~
Processing triggers for install-info (6.5.0.dfsg.1-2) ...
Processing triggers for libglib2.0-0:amd64 (2.56.4-0ubuntu0.18.04.6) ...
Setting up glib-networking:amd64 (2.56.0-1ubuntu0.1) ...
Setting up libsoup2.4-1:amd64 (2.62.1-1ubuntu0.4) ...
Setting up gstreamer1.0-plugins-good:amd64 (1.14.5-0ubuntu1~18.04.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1.2) ...
Shubham@instance:~$ wget http://mirror.pnl.gov/releases/18.04/ubuntu-18.04.5-live-server-amd64.iso
--2020-10-08 01:45:39-- http://mirror.pnl.gov/releases/18.04/ubuntu-18.04.5-live-server-amd64.iso
Resolving mirror.pnl.gov (mirror.pnl.gov)... 192.101.102.2
Connecting to mirror.pnl.gov (mirror.pnl.gov)|192.101.102.2|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 990904320 (945M) [application/octet-stream]
Saving to: 'ubuntu-18.04.5-live-server-amd64.iso'

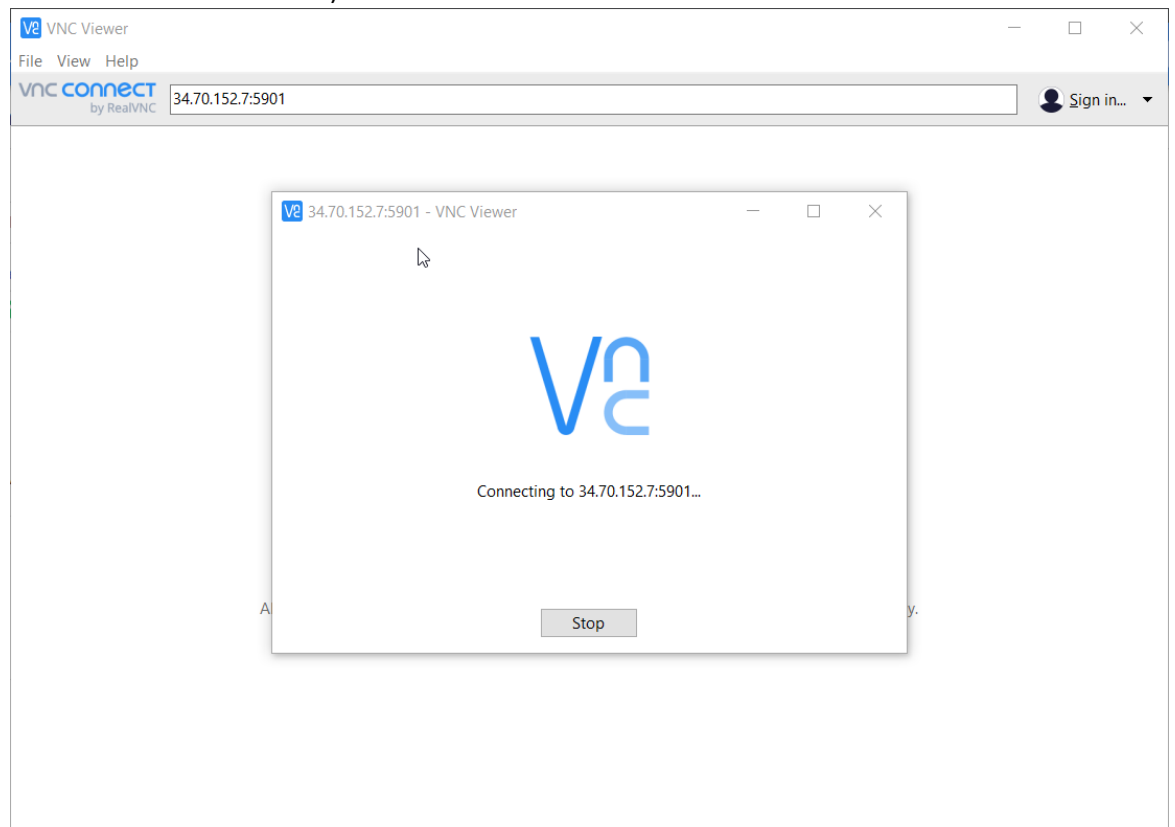
ubuntu-18.04.5-live 100%[=====>] 945.00M 54.2MB/s in 23s

2020-10-08 01:46:02 (41.9 MB/s) - 'ubuntu-18.04.5-live-server-amd64.iso' saved [990904320/990904320]

Shubham@instance:~$ sudo qemu-img create ubuntu.img 10G
Formatting 'ubuntu.img', fmt=raw size=10737418240
Shubham@instance:~$
```

```
Shubham@instance: ~  
Verify:  
xauth: file /home/Shubham/.Xauthority does not exist  
  
New 'instance:1 (Shubham)' desktop is instance:1  
  
Creating default startup script /home/Shubham/.vnc/xstartup  
Starting applications specified in /home/Shubham/.vnc/xstartup  
Log file is /home/Shubham/.vnc/instance:1.log  
  
Shubham@instance:~$ nc localhost 5901  
RFB 003.008  
^C  
Shubham@instance:~$ vncserver -kill :1  
Killing Xvnc4 process ID 19161  
Shubham@instance:~$ vim .vnc/xstartup  
Shubham@instance:~$ vim .vnc/xstartup  
Shubham@instance:~$ vncserver  
  
New 'instance:1 (Shubham)' desktop is instance:1  
  
Starting applications specified in /home/Shubham/.vnc/xstartup  
Log file is /home/Shubham/.vnc/instance:1.log  
  
Shubham@instance:~$
```

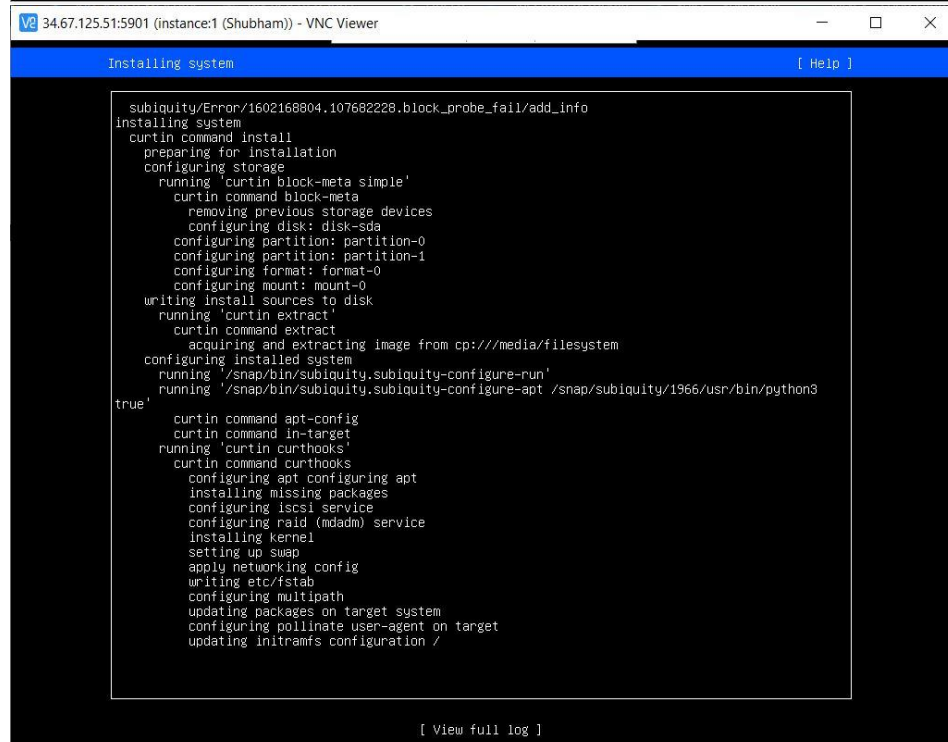
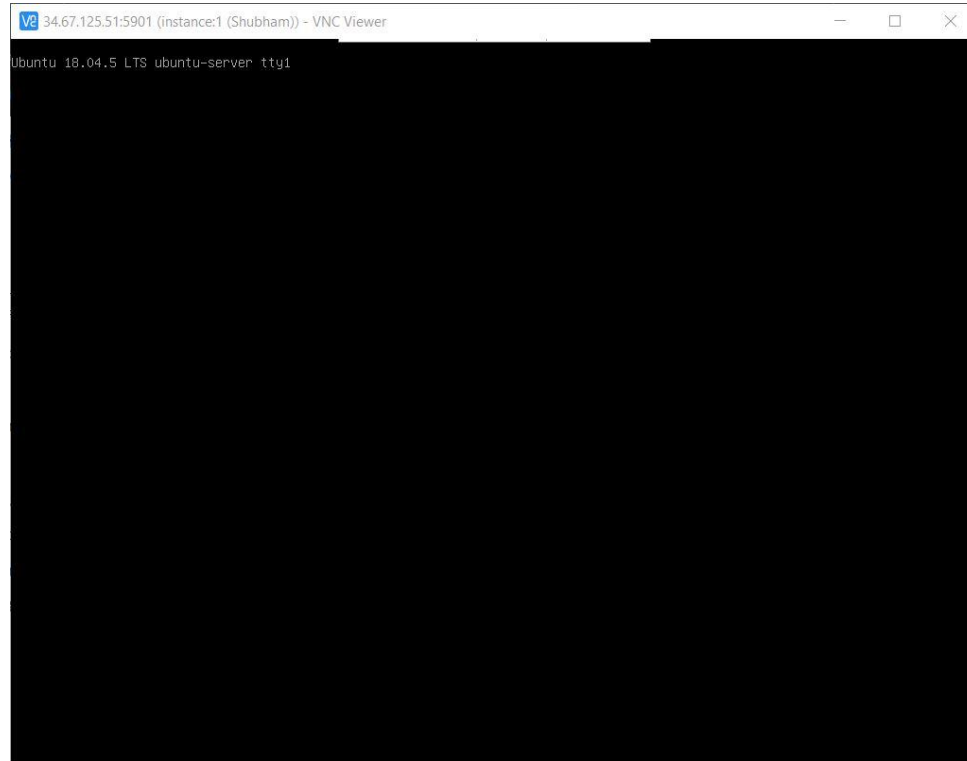
After vncserver has started, I used realvnc to connect to the instance.(ip is different, screenshot was taken later)



In terminal ; use command :

```
$sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./ubuntu-18.04.5-live-server-amd64.iso -m 1536
```

to install the image.I have done proper selections so as to get the image installed. Some images :



After installation is done, kill the server and run the server again and run the command:

`$sudo qemu-system-x86_64 -hda ubuntu.img -m 1536`

Bootup:

```
104.198.33.84:5901 (instance-1:1 (root)) - VNC Viewer

Ubuntu 18.04.5 LTS shubham tty1

shubham login: [ 186.064474] cloud-init[1019]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~18.04.1 running 'modules:config' at Fri, 09 Oct 2020 23:36:09 +0000. Up 171.97 seconds.
ci-info: no authorized SSH keys fingerprints found for user shubham.
<14>Oct 9 23:36:42 ec2: #####
<14>Oct 9 23:36:42 ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----
<14>Oct 9 23:36:42 ec2: 1024 SHA256:HWtGlxJ/y2x6JT1YJsfmYrkHB9WR2JAY9I6b8iPCcTA root@shubham (DSA)
<14>Oct 9 23:36:42 ec2: 256 SHA256:K1k22308F8f66Y5Y0uz7IOFjsTY1hBoulr6WaxiMDg root@shubham (ECDSA)
<14>Oct 9 23:36:43 ec2: 256 SHA256:+i8BFXLZEhuNbXNus9mV9PzZ0PoC4y7+ZKcjQ96i9nM root@shubham (ED25519)
<14>Oct 9 23:36:43 ec2: 2048 SHA256:jNbrfzUtv0Q+2K5XuPse0i2WcgFwtVExrq0iNlUGffa root@shubham (RSA)
<14>Oct 9 23:36:43 ec2: -----END SSH HOST KEY FINGERPRINTS-----
<14>Oct 9 23:36:43 ec2: #####
-----BEGIN SSH HOST KEY KEYS-----
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBMIF1e5iwoEAcfsQVRUTacRfntV+g/eWXIkx0Be9WVV0jTrR70MCon
s2fh7ItoYk6d2sE9055LQ2kvm8BDueQ= root@shubham
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIKj5kdmF1K+ggX93mjfQYzIE703h24EueqKsTcMmKg9 root@shubham
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCaETsvUeIt1btqricU+bUPMzoH9aMBNFY2emde3LoKIIUs1M2gp1MN+mma6Cj3YiPQ69Mp97pw8a8FAqShnjGDUBPs
EBkRRcNxxgIXCX+TJU8BFMOJgogDI9JG0Mh7xG2FonfAuWp6iXYtfhyBuZQ7o/SIb21Qk54D3T30sUS0IX645hIKU0oNHKs00an3H9Arh4tPskX4sJZEmiQJgR0W+W2wj
t2/Vk/XX32RYVJSa9Zq2eeMj2T1BN1vS1Tjleyb5/IvUM2PSe9Z/EibSkKNfFuKn6lH+4KH3KobK9R/HfUJmT+WhdzbjbzFy4dl1k6BeWPsho4uAQ1QKIBWAGWsj root@shubham
-----END SSH HOST KEY KEYS-----
[ 206.295910] cloud-init[1078]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~18.04.1 running 'modules:final' at Fri, 09 Oct 2020 23:36:37 +0000. Up 200.47 seconds.
[ 206.301820] cloud-init[1078]: ci-info: no authorized SSH keys fingerprints found for user shubham.
[ 206.306763] cloud-init[1078]: Cloud-init v. 20.2-45-g5f7825e2-0ubuntu1~18.04.1 finished at Fri, 09 Oct 2020 23:36:43 +0000. Datasource DataSourceNone. Up 206.07 seconds
[ 206.315455] cloud-init[1078]: 2020-10-09 23:36:43,706 - cc_final_message.py[WARNING]: Used fallback datasource
```

QEMU statistics :

It took approx. 5-6 mins to boot up for the first time.

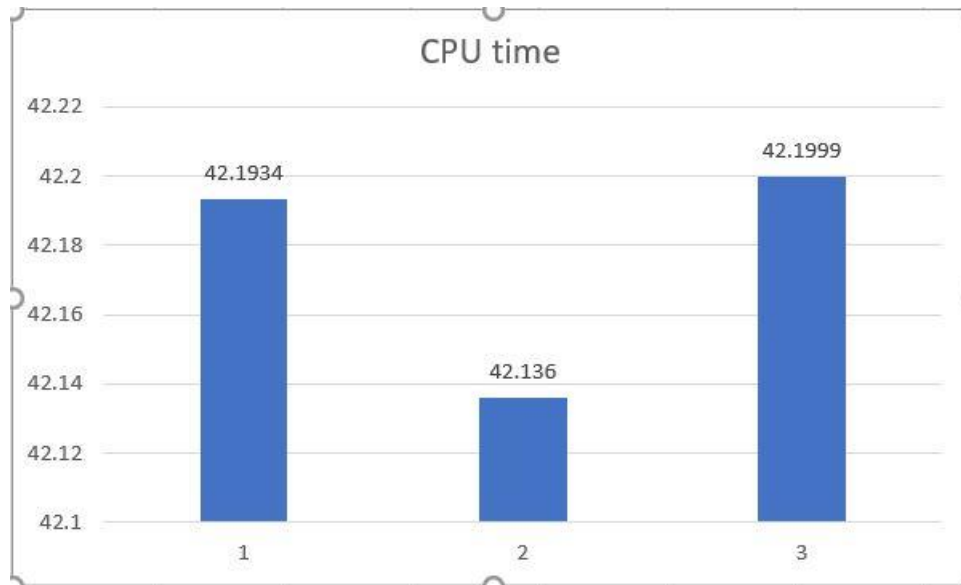
It took me approximately 2 and half hours to complete the whole installation.

## Test Statistics:

- **CPU User level using sysbench:**

Total time taken for all 3 tests = 126.47 sec

Avg time for all 3 tests = 42.1584



- **Kernel Level :**

Before starting sysbench :

Linux 5.4.0-1025-gcp (instance)				10/09/20		_x86_64_		(2 CPU)								
avg-cpu:		%user	%nice	%system	%iowait	%steal	%idle									
		3.3%	0.1%	0.4%	0.2%	0.0%	95.9%									
Device		r/s	w/s	rkB/s	wkB/s	rrqm/s	wrqm/s	%rrqm	%wrqm	r_await	w_await	aqu-sz	rareq-sz	wareq-sz	svctm	%util
loop0		0.72	0.00	0.8k	0.0k	0.00	0.00	0.0%	0.0%	0.11	0.00	0.00	1.2k	0.0k	0.09	0.0%
loop1		0.02	0.00	0.1k	0.0k	0.00	0.00	0.0%	0.0%	0.22	0.00	0.00	7.4k	0.0k	0.35	0.0%
loop2		0.02	0.00	0.4k	0.0k	0.00	0.00	0.0%	0.0%	0.02	0.00	0.00	17.5k	0.0k	0.20	0.0%
sda		3.24	5.24	184.4k	979.8k	1.43	12.58	30.6%	70.6%	1.10	3.98	0.01	56.9k	187.1k	1.58	1.3%

We can see that before starting the test, idle cpu was 95.9 % and user was using 3.3%

Now, in-between sysbench tests,



```

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           49.7%    0.0%    0.0%    0.0%    0.0%   50.3%

```

```

Device      r/s    w/s    kB/s    kB/s    rrqm/s  wrqm/s  %rrqm  %wrqm  r_await w_await aqu-sz  rareq-sz  wareq-sz  svctm  %util
loop0
           0.00    0.00    0.0k    0.0k     0.00    0.00    0.0%   0.0%    0.00    0.00    0.00    0.0k    0.0k    0.00    0.0%
loop1
           0.00    0.00    0.0k    0.0k     0.00    0.00    0.0%   0.0%    0.00    0.00    0.00    0.0k    0.0k    0.00    0.0%
loop2
           0.00    0.00    0.0k    0.0k     0.00    0.00    0.0%   0.0%    0.00    0.00    0.00    0.0k    0.0k    0.00    0.0%
sda
           0.00    0.00    0.0k    0.0k     0.00    0.00    0.0%   0.0%    0.00    0.00    0.00    0.0k    0.0k    0.00    0.0%

```

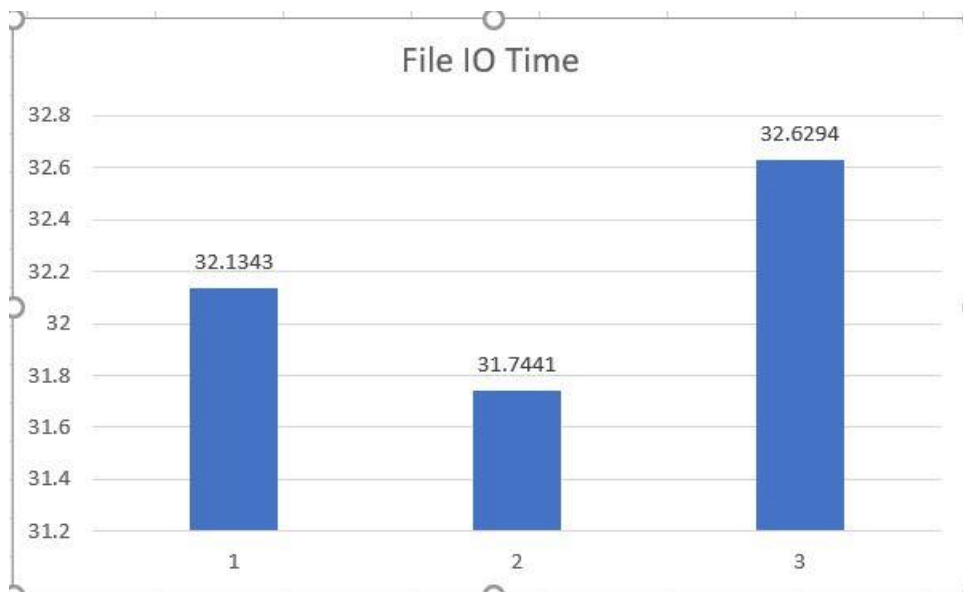
The user is using 49.7% cpu and idle time is 50.3%

### FILE IO Test :

- **User level :**

Total time taken for all 3 tests : 90.5078

Avg. time taken for all 3 tests : 32.1692



- **Kernel Level :**

For file IO, the iostat was computed for one sec for 10 intervals. After starting the test, it looks like below:

avg-cpu:	%user 0.6%	%nice 0.1%	%system 3.9%	%iowait 11.8%	%steal 0.1%	%idle 83.5%										
Device	r/s	w/s	rkB/s	wkB/s	rrqm/s	wrqm/s	%rrqm	%wrqm	r_await	w_await	aqu-sz	rareq-sz	wareq-sz	svctm	%util	
loop0	0.01	0.00	0.1k	0.0k	0.00	0.00	0.0%	0.0%	1.05	0.00	0.00	8.0k	0.0k	1.24	0.0%	
loop1	0.64	0.00	0.7k	0.0k	0.00	0.00	0.0%	0.0%	0.69	0.00	0.00	1.2k	0.0k	0.15	0.0%	
loop2	0.02	0.00	0.4k	0.0k	0.00	0.00	0.0%	0.0%	1.54	0.00	0.00	17.5k	0.0k	0.98	0.0%	
loop3	0.02	0.00	0.4k	0.0k	0.00	0.00	0.0%	0.0%	0.14	0.00	0.00	16.3k	0.0k	0.61	0.0%	
sda	21.91	265.93	596.2k	51.2M	5.05	13.03	18.7%	4.7%	49.87	2.15	1.20	27.2k	197.0k	0.61	17.4%	
avg-cpu:	%user 0.5%	%nice 0.0%	%system 3.0%	%iowait 95.0%	%steal 0.5%	%idle 1.0%										
Device	r/s	w/s	rkB/s	wkB/s	rrqm/s	wrqm/s	%rrqm	%wrqm	r_await	w_await	aqu-sz	rareq-sz	wareq-sz	svctm	%util	
loop0	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop1	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop2	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop3	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
sda	145.00	482.00	2.3M	2.1M	7.00	50.00	4.6%	9.4%	80.21	2.94	12.34	16.0k	4.4k	1.09	68.4%	
avg-cpu:	%user 0.0%	%nice 0.0%	%system 2.0%	%iowait 96.5%	%steal 0.5%	%idle 1.0%										
Device	r/s	w/s	rkB/s	wkB/s	rrqm/s	wrqm/s	%rrqm	%wrqm	r_await	w_await	aqu-sz	rareq-sz	wareq-sz	svctm	%util	
loop0	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop1	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop2	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
loop3	0.00	0.00	0.0k	0.0k	0.00	0.00	0.0%	0.0%	0.00	0.00	0.00	0.0k	0.0k	0.00	0.0%	
sda	146.00	512.00	2.3M	2.1M	0.00	38.00	0.0%	6.9%	70.32	3.47	11.28	16.0k	4.3k	1.03	67.6%	

We can see that read/sec and write/sec increases and idle time reduces as well as iowait increases, which is expected for file io test.

I have also mentioned disk utilization in the images given above.

The full iostat statistics are uploaded here :

<https://drive.google.com/drive/folders/1f28ppUWzhEDrXrT6xjrkaWAVYS1GirFS?usp=sharing>

**Please answer the question, why is the QEMU based VM so slow (to install and to execute)?**

**Answer :**

We know and understand that QEMU is a GUI based full virtual machine which consist of CPU-scheduler to schedule and manage processes, other tools which a full VM possesses, while docker does not.

Hence, QEMU takes more time compare to docker .It also took me 5-6 mins to boot up which is slow.

## MiniDocker:

## References:

<https://man7.org/linux/man-pages/man2/unshare.2.html>

[http://www.cs.binghamton.edu/~huilu/slidesfall2020/Lecture\\_4\\_5\\_Containerization.pdf](http://www.cs.binghamton.edu/~huilu/slidesfall2020/Lecture_4_5_Containerization.pdf)

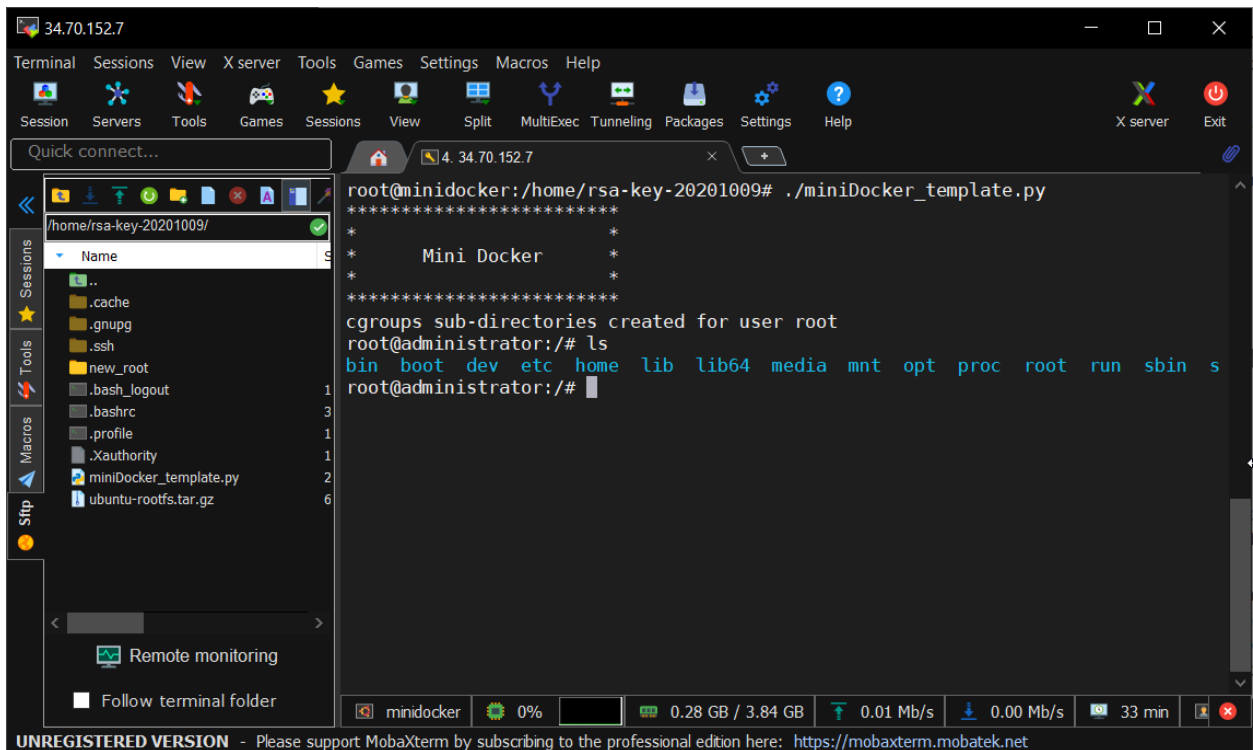
<https://binghamton.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=2a2ca5ae-c017-44c2-8ffc-ac2f016628fb>

<https://docs.python.org/3/library/os.html>

I have used the following references for coding the mini docker.

Test Cases :

1) ls:

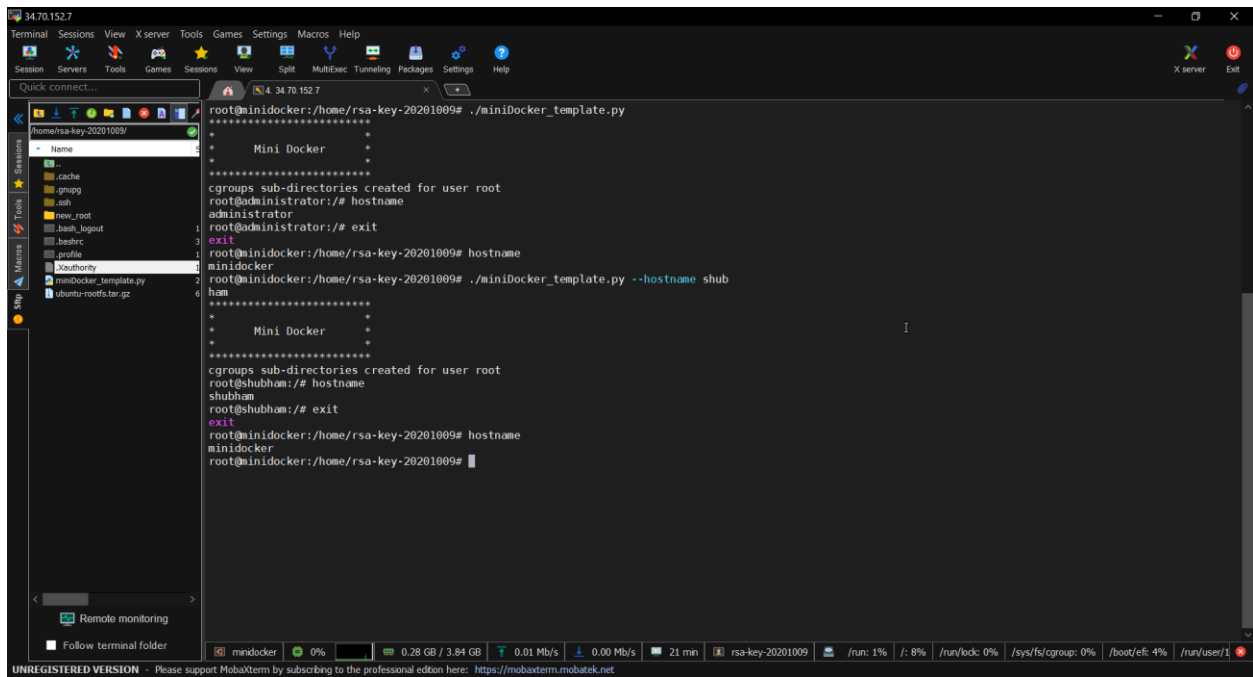


The screenshot shows a MobaXterm terminal window with a dark theme. The title bar indicates the IP address 34.70.152.7. The terminal interface includes a menu bar (Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help) and a toolbar with icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, Help, X server, and Exit. A sidebar on the left shows a file explorer for the directory /home/rsa-key-20201009/, listing files like .cache, .gnupg, .ssh, new\_root, .bash\_logout, .bashrc, .profile, .Xauthority, miniDocker\_template.py, and ubuntu-rootfs.tar.gz. The main terminal area shows the execution of the script ./miniDocker\_template.py. The script output includes a separator line of asterisks, the title 'Mini Docker', another separator line, and a message 'cgroups sub-directories created for user root'. It then shows the prompt 'root@administrator:/#' followed by the command 'ls' and its output: 'bin boot dev etc home lib lib64 media mnt opt proc root run sbin s'. The status bar at the bottom shows 'UNREGISTERED VERSION' and a link to the professional edition, along with system metrics like 'minidocker', '0%', '0.28 GB / 3.84 GB', '0.01 Mb/s', '0.00 Mb/s', and '33 min'.

```
34.70.152.7
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/rsa-key-20201009/
Name
..
.cache
.gnupg
.ssh
new_root
.bash_logout
.bashrc
.profile
.Xauthority
miniDocker_template.py
ubuntu-rootfs.tar.gz
Remote monitoring
Follow terminal folder
minidocker 0% 0.28 GB / 3.84 GB 0.01 Mb/s 0.00 Mb/s 33 min
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

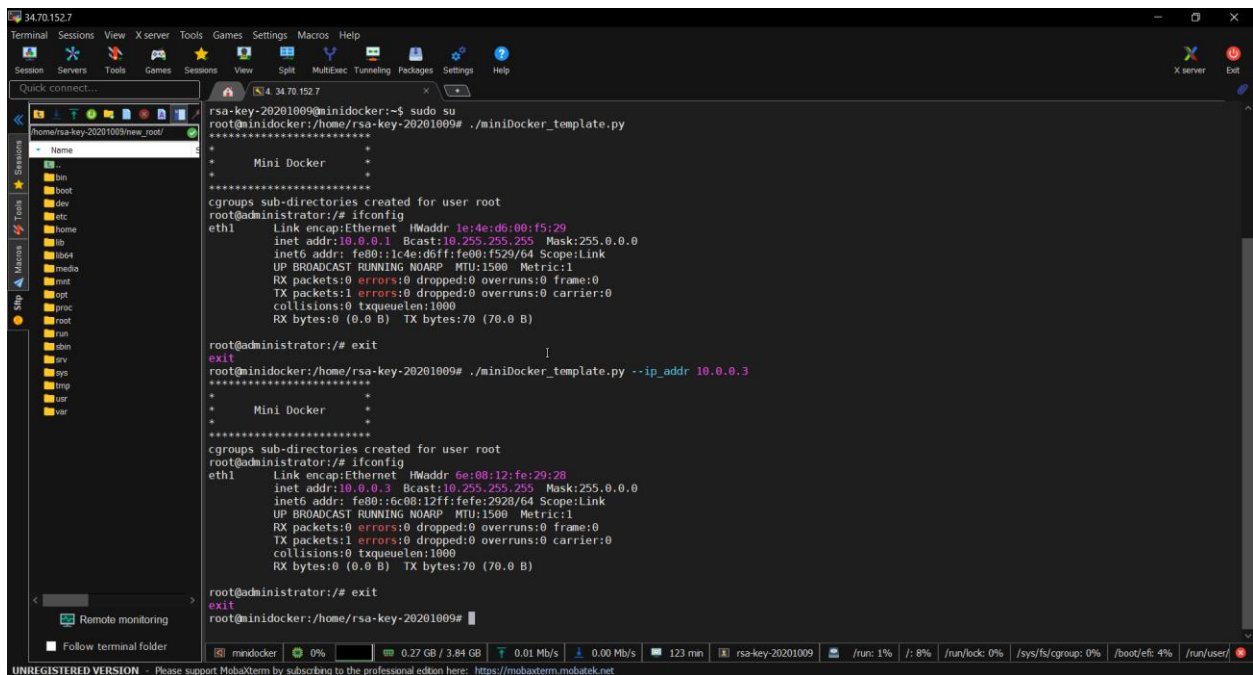
root@minidocker:/home/rsa-key-20201009# ./miniDocker_template.py
*****
*
*      Mini Docker
*
*
*****
cgroups sub-directories created for user root
root@administrator:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin s
root@administrator:/#
```

## 2) Hostname :



```
root@minidocker:/home/rsa-key-20201009# ./miniDocker_template.py
*****
* Mini Docker *
*****
cgroups sub-directories created for user root
root@administrator:/# hostname
administrator
root@administrator:/# exit
exit
root@minidocker:/home/rsa-key-20201009# hostname
minidocker
root@minidocker:/home/rsa-key-20201009# ./miniDocker_template.py --hostname shub
*****
* Mini Docker *
*****
cgroups sub-directories created for user root
root@shubham:/# hostname
shubham
root@shubham:/# exit
exit
root@minidocker:/home/rsa-key-20201009# hostname
minidocker
root@minidocker:/home/rsa-key-20201009#
```

## 3) Network namespace : ifconfig:

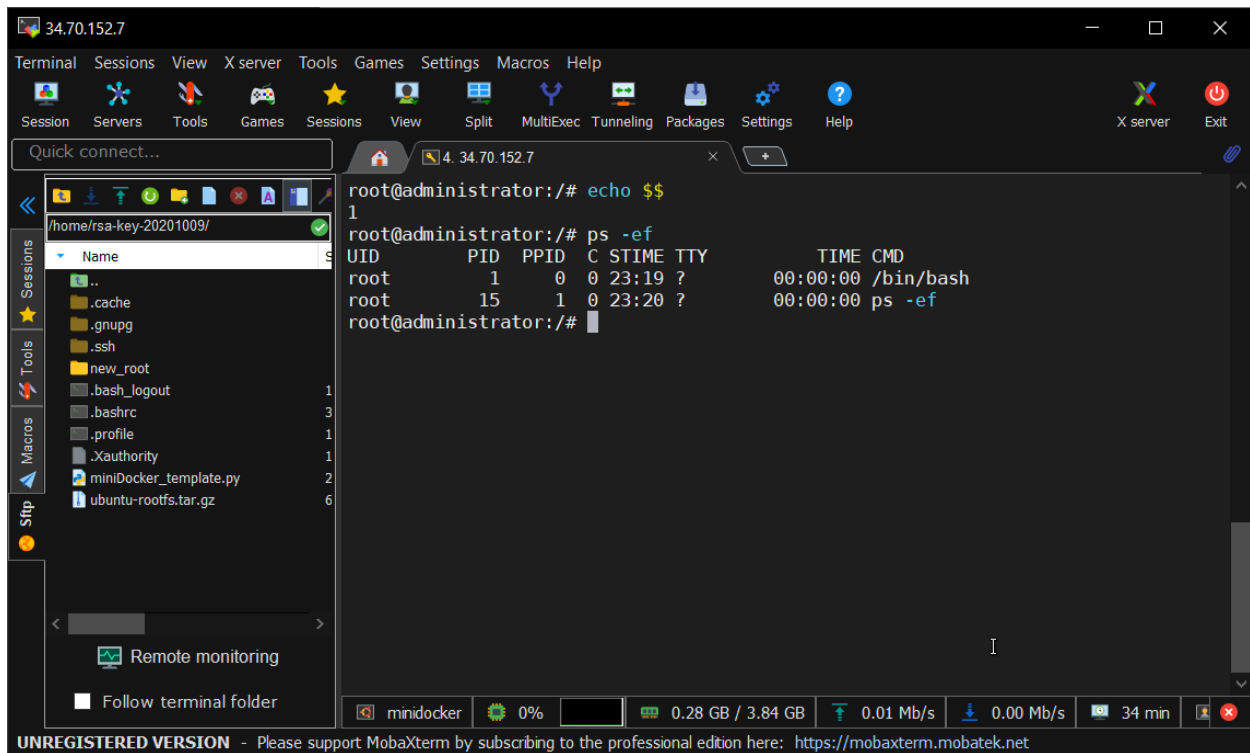


```
rsa-key-20201009@minidocker:~$ sudo su
root@minidocker:/home/rsa-key-20201009# ./miniDocker_template.py
*****
* Mini Docker *
*****
cgroups sub-directories created for user root
root@administrator:/# ifconfig
eth1  Link encap:Ethernet  HWaddr 1e:4e:d6:00:f5:29
      inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
      inet6 addr: fe80::1c4e:d6ff:fe00:f529/64 Scope:Link
      UP BROADCAST RUNNING NOARP MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:70 (70.0 B)

root@administrator:/# exit
exit
root@minidocker:/home/rsa-key-20201009# ./miniDocker_template.py --ip_addr 10.0.0.3
*****
* Mini Docker *
*****
cgroups sub-directories created for user root
root@administrator:/# ifconfig
eth1  Link encap:Ethernet  HWaddr 6e:08:12:fe:29:28
      inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
      inet6 addr: fe80::6e08:12ff:fe29:28/64 Scope:Link
      UP BROADCAST RUNNING NOARP MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:0 (0.0 B)  TX bytes:70 (70.0 B)

root@administrator:/# exit
exit
root@minidocker:/home/rsa-key-20201009#
```

#### 4) echo \$\$ and ps -ef

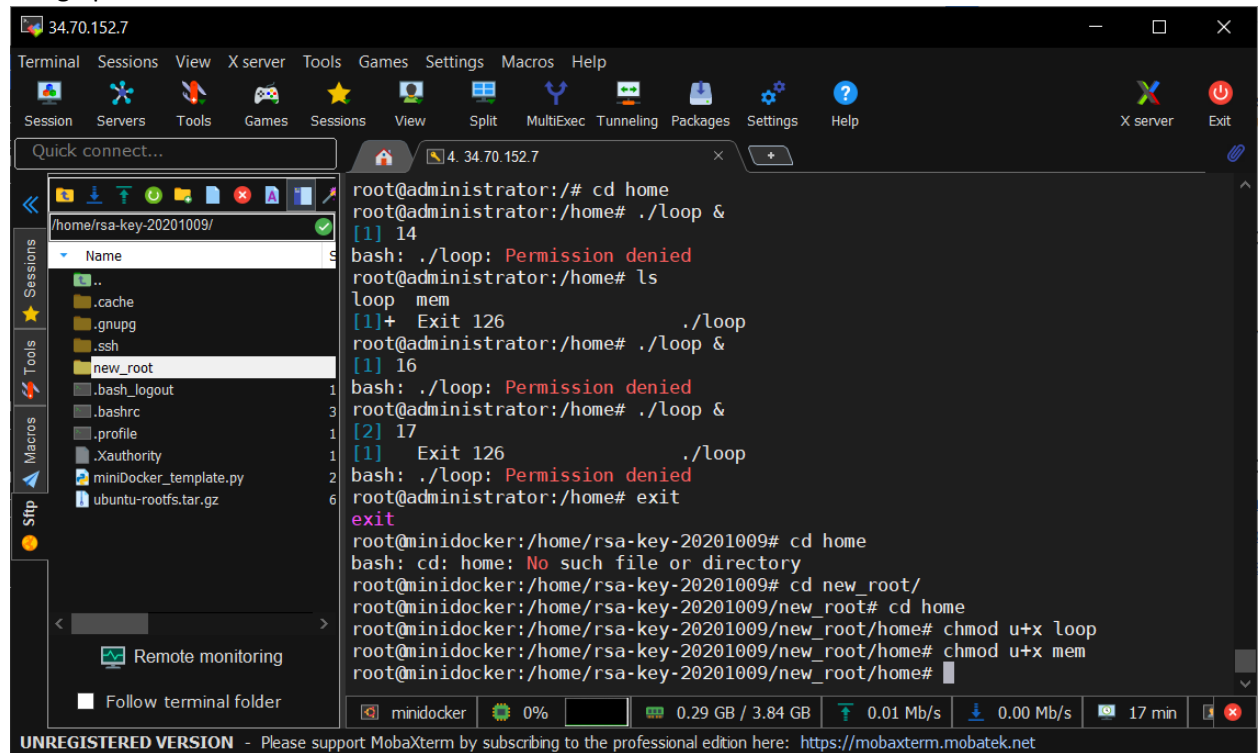


The screenshot shows a MobaXterm terminal window with the title '34.70.152.7'. The terminal is running a shell session as 'root@administrator'. The user has entered the command 'echo \$\$', which outputs '1'. Then, the user enters 'ps -ef', which displays a table of running processes. The table has columns for UID, PID, PPID, C, STIME, TTY, TIME, and CMD. The output shows two processes: a /bin/bash process with PID 1 and a ps -ef process with PID 15. The left sidebar shows a file explorer with a tree view of the file system, including directories like .cache, .gnupg, .ssh, and new\_root. The bottom status bar indicates 'UNREGISTERED VERSION' and provides a link to the professional edition.

```
root@administrator:/# echo $$
1
root@administrator:/# ps -ef
UID          PID    PPID  C   STIME TTY          TIME CMD
root           1         0   0  23:19 ?           00:00:00 /bin/bash
root          15         1   0  23:20 ?           00:00:00 ps -ef
root@administrator:/#
```

#### 5) CPUSET : Loop and mem

Assign permission first :



The screenshot shows a MobaXterm terminal window with the title '34.70.152.7'. The terminal is running a shell session as 'root@administrator'. The user has entered the command 'cd home', which outputs 'root@administrator:/home#'. Then, the user enters './loop &', which outputs '[1] 14'. The user then enters 'ls', which outputs 'bash: ./loop: Permission denied'. The user then enters 'ls', which outputs 'root@administrator:/home# ls'. The user then enters 'loop mem', which outputs '[1]+ Exit 126 ./loop'. The user then enters './loop &', which outputs '[1] 16'. The user then enters 'ls', which outputs 'bash: ./loop: Permission denied'. The user then enters './loop &', which outputs '[2] 17'. The user then enters 'ls', which outputs '[1] Exit 126 ./loop'. The user then enters 'bash: ./loop: Permission denied'. The user then enters 'exit', which outputs 'root@administrator:/home# exit'. The user then enters 'exit', which outputs 'root@minidocker:/home/r...# cd home'. The user then enters 'cd: home: No such file or directory'. The user then enters 'cd new\_root/', which outputs 'root@minidocker:/home/r...# cd new\_root/'. The user then enters 'cd home', which outputs 'root@minidocker:/home/r.../new\_root# cd home'. The user then enters 'chmod u+x loop', which outputs 'root@minidocker:/home/r.../new\_root/home# chmod u+x loop'. The user then enters 'chmod u+x mem', which outputs 'root@minidocker:/home/r.../new\_root/home#'. The left sidebar shows a file explorer with a tree view of the file system, including directories like .cache, .gnupg, .ssh, and new\_root. The bottom status bar indicates 'UNREGISTERED VERSION' and provides a link to the professional edition.

```
root@administrator:/# cd home
root@administrator:/home# ./loop &
[1] 14
bash: ./loop: Permission denied
root@administrator:/home# ls
loop mem
[1]+ Exit 126 ./loop
root@administrator:/home# ./loop &
[1] 16
bash: ./loop: Permission denied
root@administrator:/home# ./loop &
[2] 17
[1] Exit 126 ./loop
bash: ./loop: Permission denied
root@administrator:/home# exit
exit
root@minidocker:/home/r...# cd home
bash: cd: home: No such file or directory
root@minidocker:/home/r...# cd new_root/
root@minidocker:/home/r.../new_root# cd home
root@minidocker:/home/r.../new_root/home# chmod u+x loop
root@minidocker:/home/r.../new_root/home# chmod u+x mem
root@minidocker:/home/r.../new_root/home#
```



I had messed up with my password for VM . So I used google instance (ubuntu 18.04.5) for previous tasks which ran flawlessly. But, the './mem &' was getting killed in google instance for unknown reason. Later, I tried with college VM from another user and it worked.

## 1) Mem :

```
cgroups sub-directories created for user root
root@administrator:/# cd home
root@administrator:/home# ./mem &
[1] 13
root@administrator:/home# ./mem &
[2] 14
root@administrator:/home#
```

As we can see, we have default limit to 10MB which is 0.2% of MEM

```
top - 02:39:55 up 29 days, 6:28, 0 users, load average: 1.57, 0.61, 0.23
Tasks: 4 total, 3 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 99.7 us, 0.2 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
KiB Mem: 4039072 total, 3338612 used, 700460 free, 292264 buffers
KiB Swap: 1003516 total, 404708 used, 598808 free. 1855568 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14	root	20	0	209064	7128	992	R	99.6	0.2	0:34.95	mem
13	root	20	0	209064	916	916	R	99.2	0.0	0:36.02	mem
1	root	20	0	18224	3088	2860	S	0.0	0.1	0:00.00	bash
15	root	20	0	19884	2488	2132	R	0.0	0.1	0:00.00	top

## 2) Loop :

```
cgroups sub-directories created for user root
root@administrator:/# cdhome
bash: cdhome: command not found
root@administrator:/# cd home
root@administrator:/home# ./loop &
[1] 14
root@administrator:/home# ./loop &
[2] 15
root@administrator:/home#
```

Tried , but failed to create cgroup for cpu and loop failed.

```
top - 02:42:53 up 29 days, 6:31, 0 users, load average: 1.93, 1.16, 0.51
Tasks: 4 total, 3 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
KiB Mem: 4039072 total, 3330696 used, 708376 free, 292264 buffers
KiB Swap: 1003516 total, 484 used, 1003032 free. 1855572 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14	root	20	0	4200	800	720	R	99.6	0.0	2:19.09	loop
15	root	20	0	4200	628	548	R	99.6	0.0	2:17.87	loop
1	root	20	0	18224	3364	2840	S	0.0	0.1	0:00.00	bash
16	root	20	0	19884	2388	2028	R	0.0	0.1	0:00.01	top