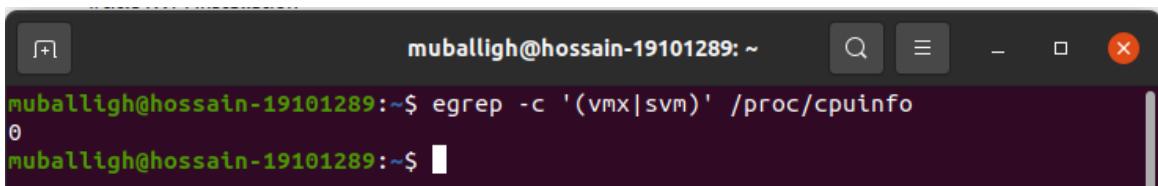


CSE 484
Assignment - 2
Md Muballigh Hossain Bhuyain
ID - 19101289

Task 1 - Installing KVM

Steps:

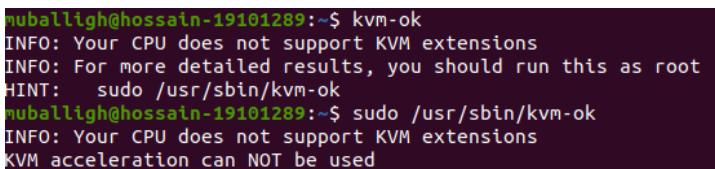
- In order for us to run KVM, our processor (in my instance AMD) needs to support hardware virtualisation, We will check using the following command:



```
muballigh@hossain-19101289:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
0
muballigh@hossain-19101289:~$
```

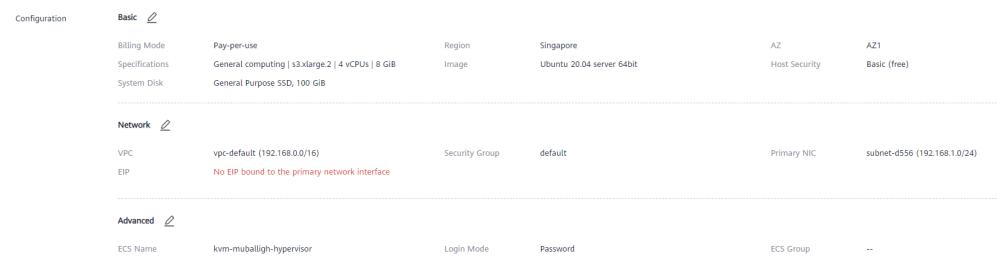
0 -> Signifies that CPU does not support hardware virtualisation

1-> It does support hardware virtualisation



```
muballigh@hossain-19101289:~$ kvm-ok
INFO: Your CPU does not support KVM extensions
INFO: For more detailed results, you should run this as root
HINT: sudo /usr/sbin/kvm-ok
muballigh@hossain-19101289:~$ sudo /usr/sbin/kvm-ok
INFO: Your CPU does not support KVM extensions
KVM acceleration can NOT be used
```

- In our Case KVM is not supported, so we will deploy an instance from a Cloud Provider
The specs of our instance are as follows:



Configuration Basic

Billing Mode	Pay-per-use	Region	AZ	AZ1
Specifications	General computing s3.xlarge.2 4 vCPUs 8 GiB	Singapore	Ubuntu 20.04 server 64bit	Host Security
System Disk	General Purpose SSD, 100 GiB			Basic (free)

Network

VPC	Security Group	Primary NIC	Subnet
vpc-default (192.168.0.0/16)	default	subnet-d556 (192.168.1.0/24)	
EIP			

Advanced

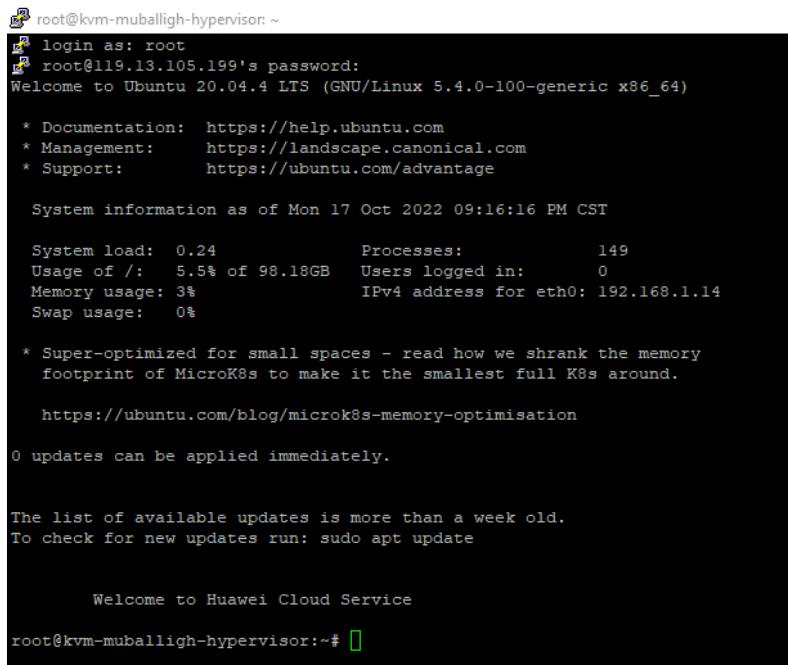
ECS Name	Login Mode	EC Group	...
kvm-muballigh-hypervisor	Dynamic BGP	bandwidth-9ca3	
		Traffic 300 Mbit/s	
		ECS	kvm-muballigh-hype...
			Pay-per-use
			Created on Oct 17, 2023
			Bind Unbind More

- Now we need an Elastic IP to access it publicly -



IP Address	Association	State	Region	Created On	Actions
119.13.105.199	kvm-muballigh-hypervisor	Bound	Dynamic BGP	bandwidth-9ca3	Traffic 300 Mbit/s
	67167eec-354d-4a81-b708-593c23619bd9				ECS
					kvm-muballigh-hype...
					Pay-per-use
					Created on Oct 17, 2023
					Bind Unbind More

- Accessing our deployed instance using Putty or MobaXterm-



```

root@kvm-muballigh-hypervisor: ~
[ ] login as: root
[ ] root@119.13.105.199's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-100-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Mon 17 Oct 2022 09:16:16 PM CST

 System load: 0.24          Processes:           149
 Usage of /: 5.5% of 98.18GB   Users logged in:      0
 Memory usage: 3%           IPv4 address for eth0: 192.168.1.14
 Swap usage: 0%

 * Super-optimized for small spaces - read how we shrank the memory
 footprint of MicroK8s to make it the smallest full K8s around.

 https://ubuntu.com/blog/microk8s-memory-optimisation

 0 updates can be applied immediately.

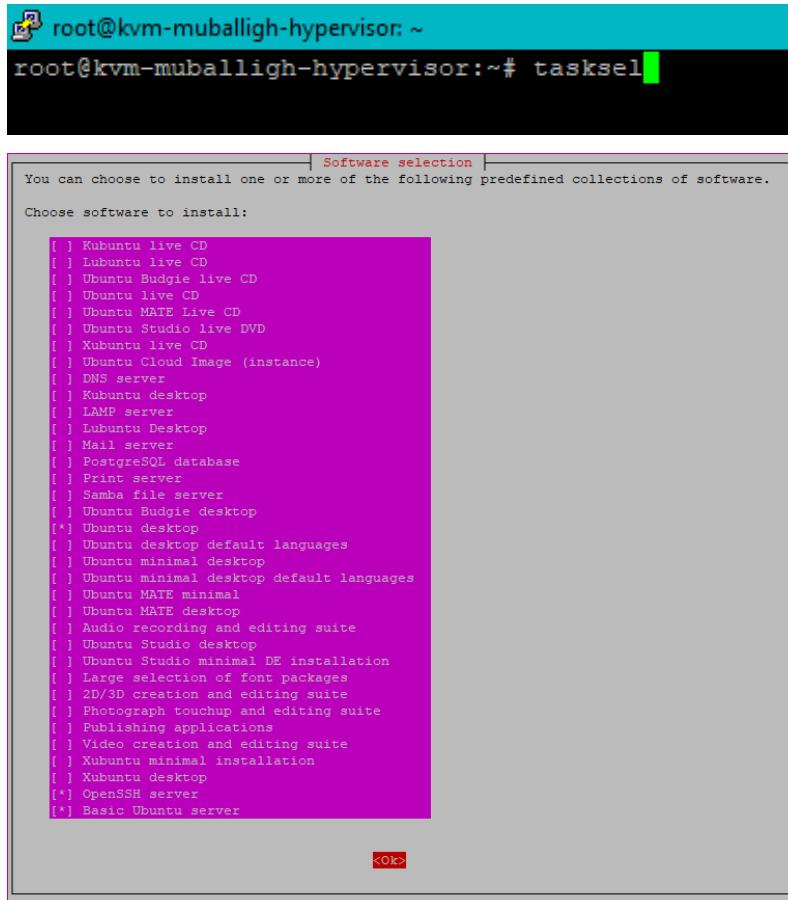
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Welcome to Huawei Cloud Service

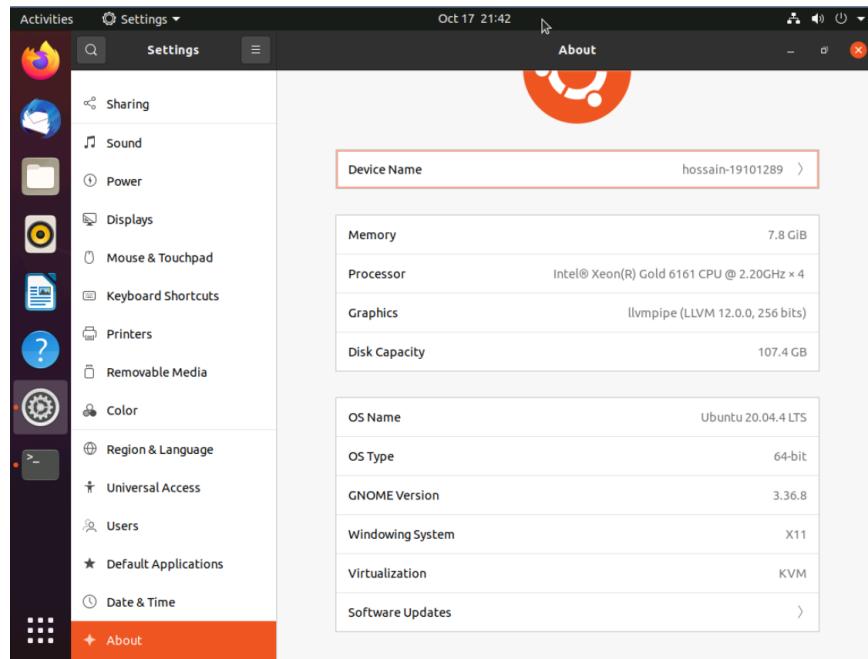
root@kvm-muballigh-hypervisor:~# [ ]

```

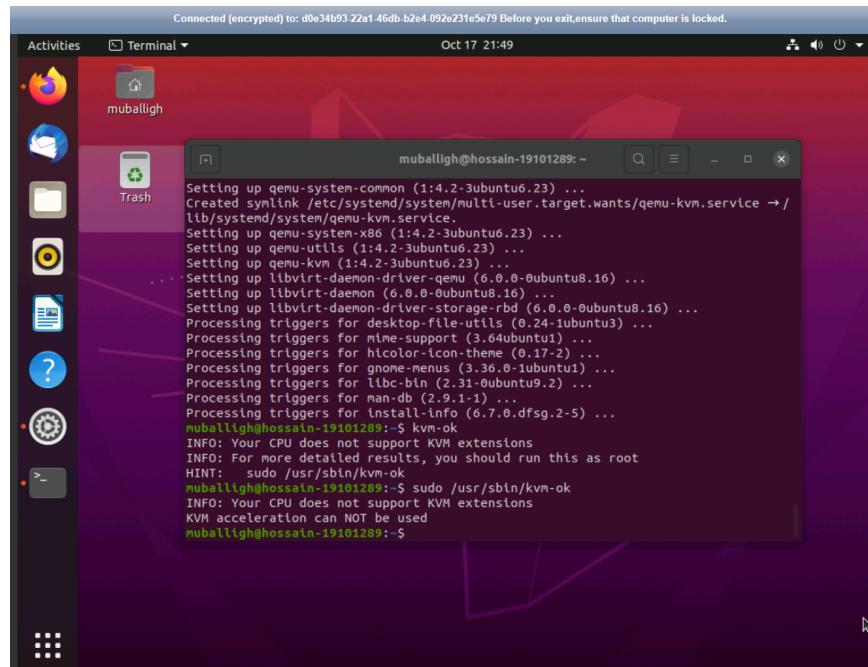
- GUI using Tasksel



- Our Cloud Instance is ready!



- We will now check if Nested Virtualization is supported on our cloud instance

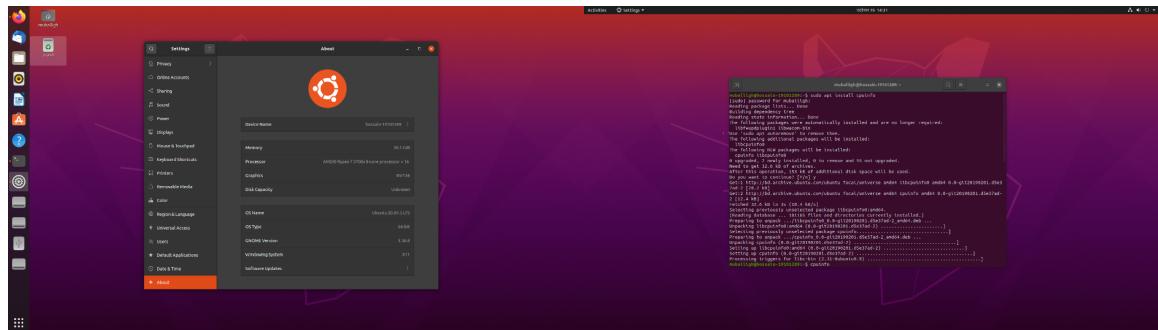


```
muballigh@hossain-19101289:~$ sudo /usr/sbin/kvm-ok
INFO: Your CPU does not support KVM extensions
KVM acceleration can NOT be used
muballigh@hossain-19101289:~$ grep -c '(vmx|svm)' /proc/cpuinfo
0
```

In our Cloud Instance also, KVM is not compatible. Our only option now is to avail AMD-V on our local machine.

KVM on Local Machine contd

- Ubuntu in dual mode installed



- We will see if our processor is 64-bit:

```
muballigh@muballigh-hossain-19101289:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
16
muballigh@muballigh-hossain-19101289:~$
```

0-> CPU is not 64-Bit

1 or higher -> It is 64-Bit

- To see if our running Kernel is 64 Bit : Yes, t is running the Kernel in 64 Bit

```
muballigh@muballigh-hossain-19101289:~$ uname -m
x86_64
```

- Now we shall install KVM (Cosmic 18.10 or later) using the following command:

```
muballigh@hossain-19101289:~$ sudo apt-get install qemu-kvm libvirt-daemon-syste
m libvirt-clients bridge-utils
[sudo] password for muballigh:
Reading package lists... Done
Building dependency tree
```

- Ensuring username is added to the group libvирtd

```
muballigh@hossain-19101289:~$ sudo adduser `id -un` libvirt
[sudo] password for muballigh: █
```

- Associating new member to kvm

```
muballigh@hossain-19101289:~$ sudo adduser `id -un` kvm
Adding user `muballigh' to group `kvm' ...
Adding user muballigh to group kvm
Done.
```

- Checking if username is added to to the kvm and libvирtd groups

```
muballigh@hossain-19101289:~$ groups
muballigh adm cdrom sudo dip plugdev lpadmin lxd sambashare libvirt
```

- Testing if installation was correct

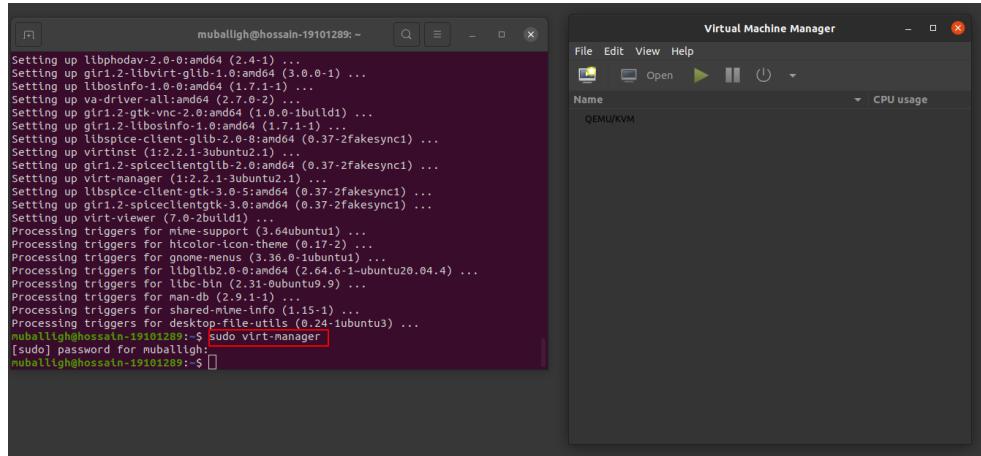
```
muballigh@hossain-19101289:~$ virsh list --all
 Id   Name    State
 -----
```

- Installing Virt manager

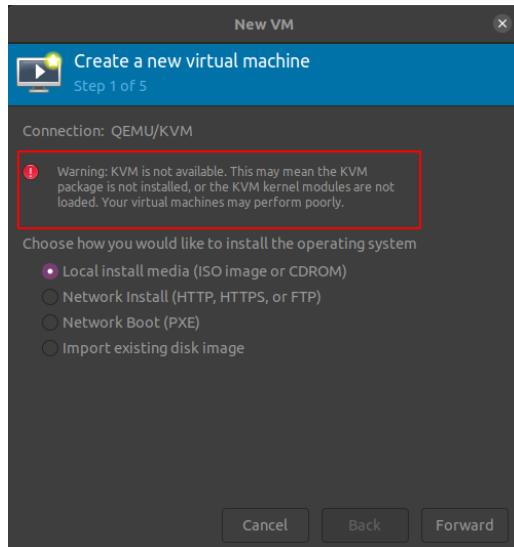
```
muballigh@hossain-19101289:~$ sudo apt-get install virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Task 2 - VM Using KVM GUI

- Starting Virt Manager and Select "New Virtual Machine" from the "File" Tab



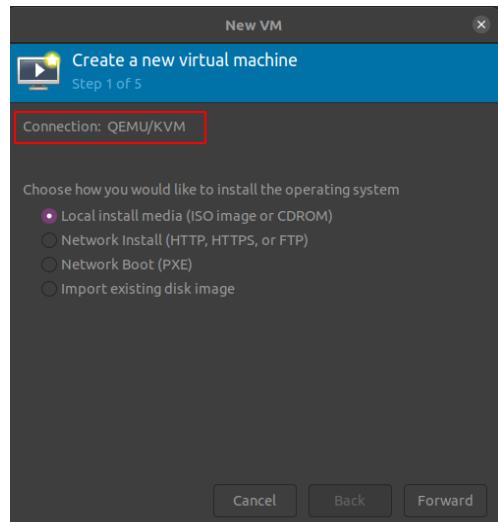
- VMM Shows this error:



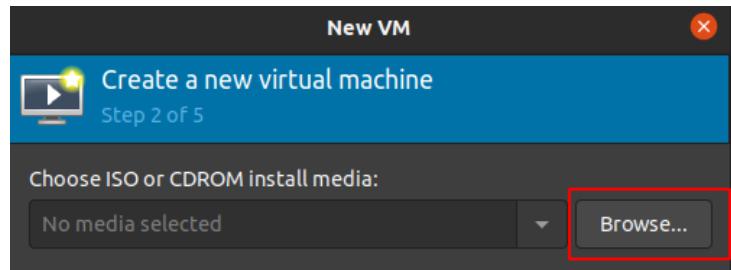
- We solve this by Enabling SVM from System BIOS



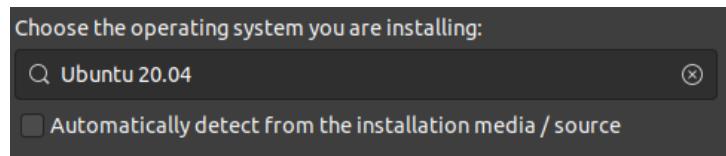
- Connection with Qemu/KVM is properly established. Select Local Install Media.



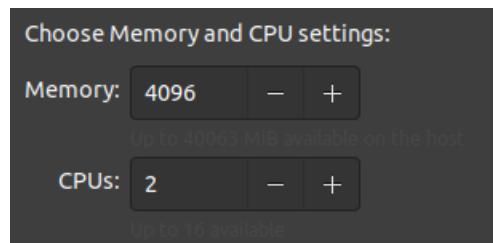
- Locate the iso Image from the local directory.



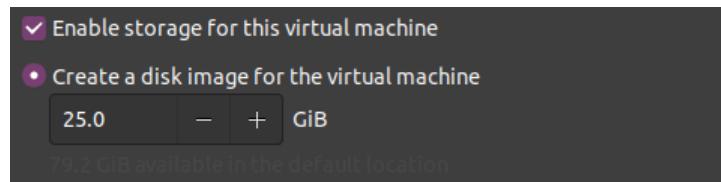
- Choose the OS



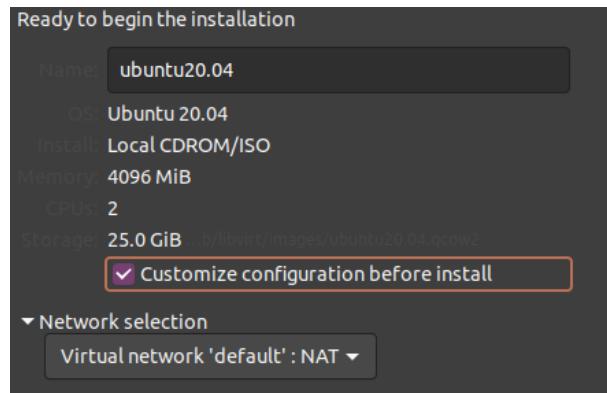
- Choose Specifications



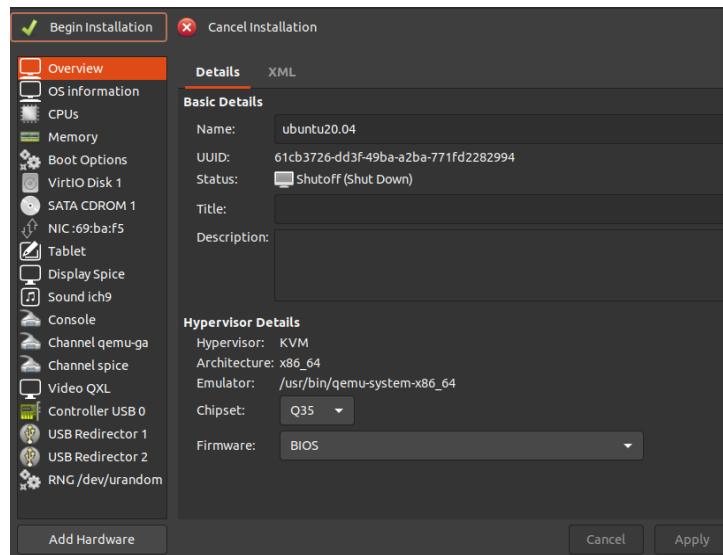
- Choose Storage



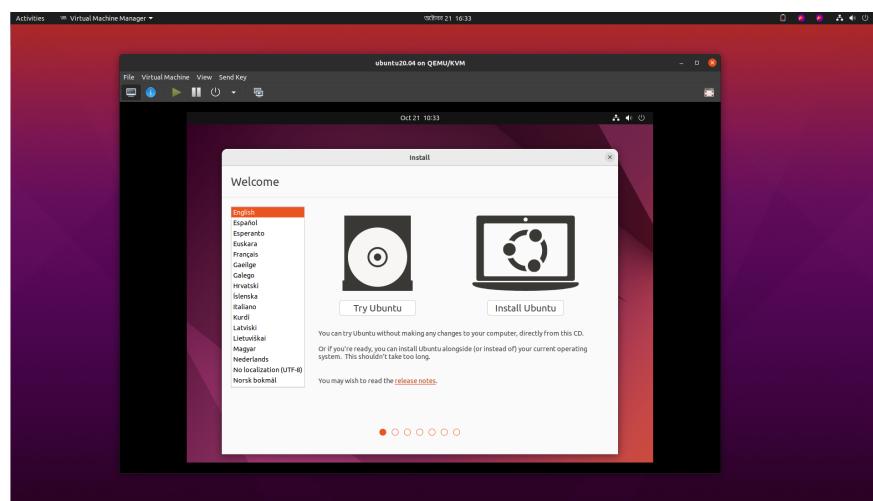
- Choose the Default NAT and check the final specs



- The Virtualized Instance:



- VM Started Successfully



- Installing Ubuntu Desktop LTS using VMM GUI

Install

Updates and other software

What apps would you like to install to start with?

Normal installation
Web browser, utilities, office software, games, and media players.

Minimal installation
Web browser and basic utilities.

Other options

Download updates while installing Ubuntu
This saves time after installation.

Install third-party software for graphics and Wi-Fi hardware and additional media formats
This software is subject to license terms included with its documentation. Some is proprietary.

Installation type

This computer currently has no detected operating systems. What would you like to do?

Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

Advanced features... None selected

Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Who are you?

Your name:

Your computer's name:
The name it uses when it talks to other computers.

Pick a username:

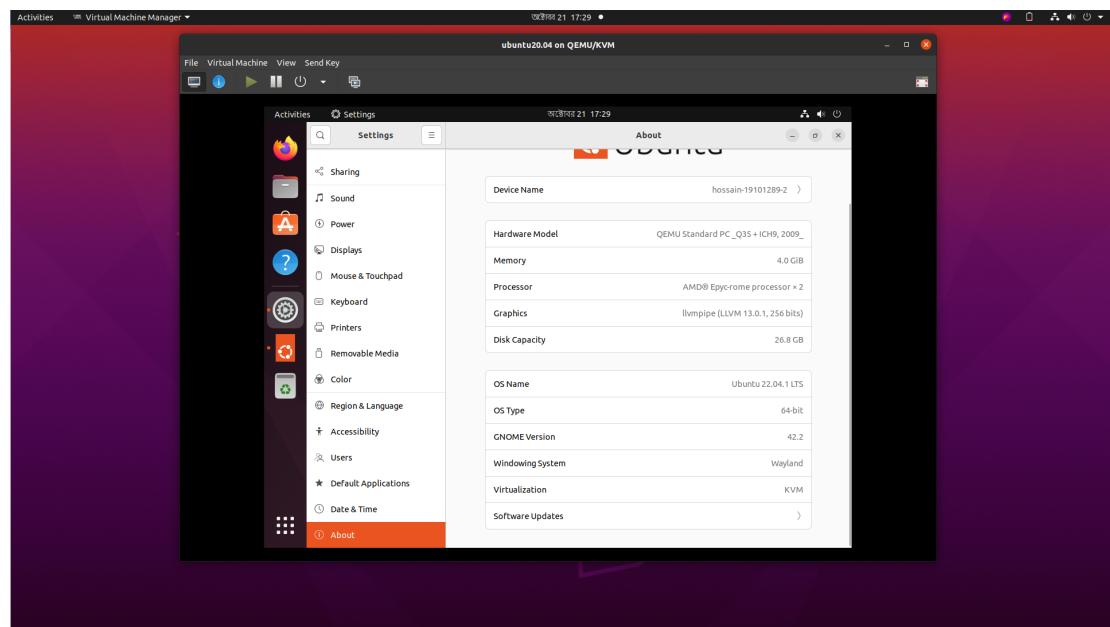
Choose a password:
(F)

Confirm your password:

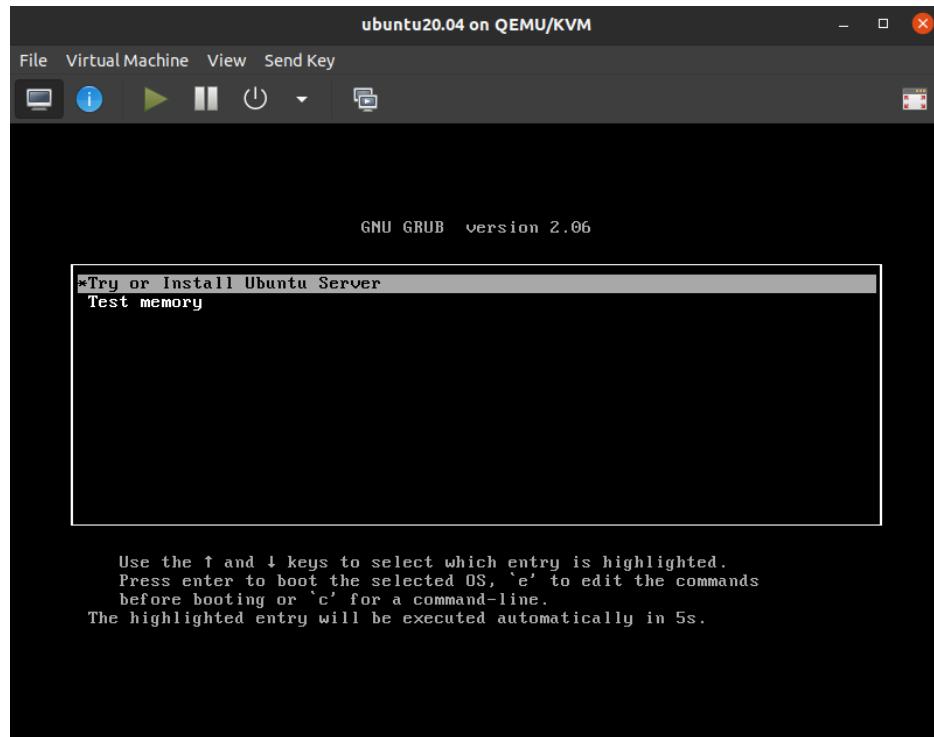
Log in automatically
 Require my password to log in

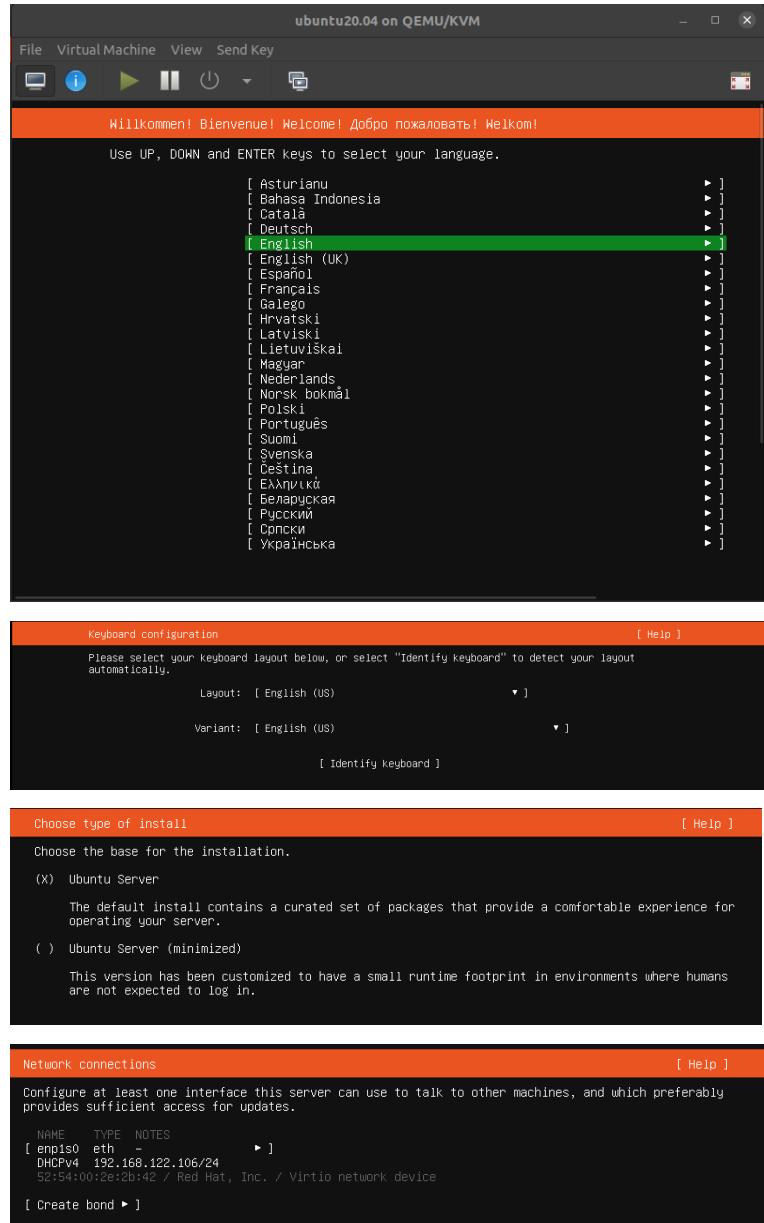
Use Active Directory
You'll enter domain and other details in the next step.

- After Installation:



- Installing Ubuntu Server Using VMM GUI (first few steps of selecting local iso image is similar to the previous case)





After this the Ubuntu Server will be installed and can be used as a regular instance using the VMM.

Task 3 - VM Using *virt-install* in CLI

- The command for installing a Ubuntu VM using *virt-install* in the CLI is as follows:

```
muballigh@hossain-19101289:~$ virt-install \
>           --connect qemu:///system \
>           --virt-type kvm \
>           --name ubuntu_demo \
>           --memory 2048 \
>           --disk size=25 \
>           --cdrom /home/muballigh/Downloads/ubuntu-22.04.1-desktop-amd64.iso \
>           --os-variant ubuntu20.04

Starting install...
```

Then a VM instance will be created. The installation procedure is similar to the above mentioned case. So, we are going to work with a different scenario, where we illustrate the installation procedure of CentOS (a different OS) using a URL instead of a disk iso image.

- The command for installing a CentOS 7 KVM from a URL using *virt-install* in the CLI is as follows:

```
muballigh@hossain-19101289:~$ virt-install \
>           --connect qemu:///system \
>           --memory 8192 \
>           --vcpus 8 \
>           --graphics vnc \
>           --os-variant centos7.0 \
>           --location http://mirror.centos.org/centos-7/7/os/x86_64/
Using default --name centos7.0
Using centos7.0 default --disk size=20

Starting install...
Retrieving file vmlinuz... | 6.5 MB 00:02
Retrieving file initrd.img 32% [=====] 1.8 MB/s | 17 MB 00:19 ETA
```

- A new Window will pop up

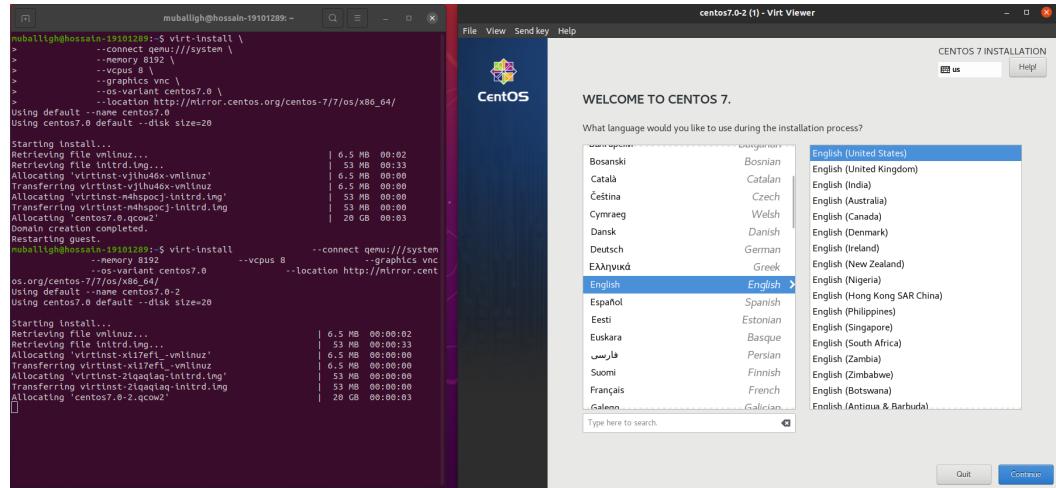
```

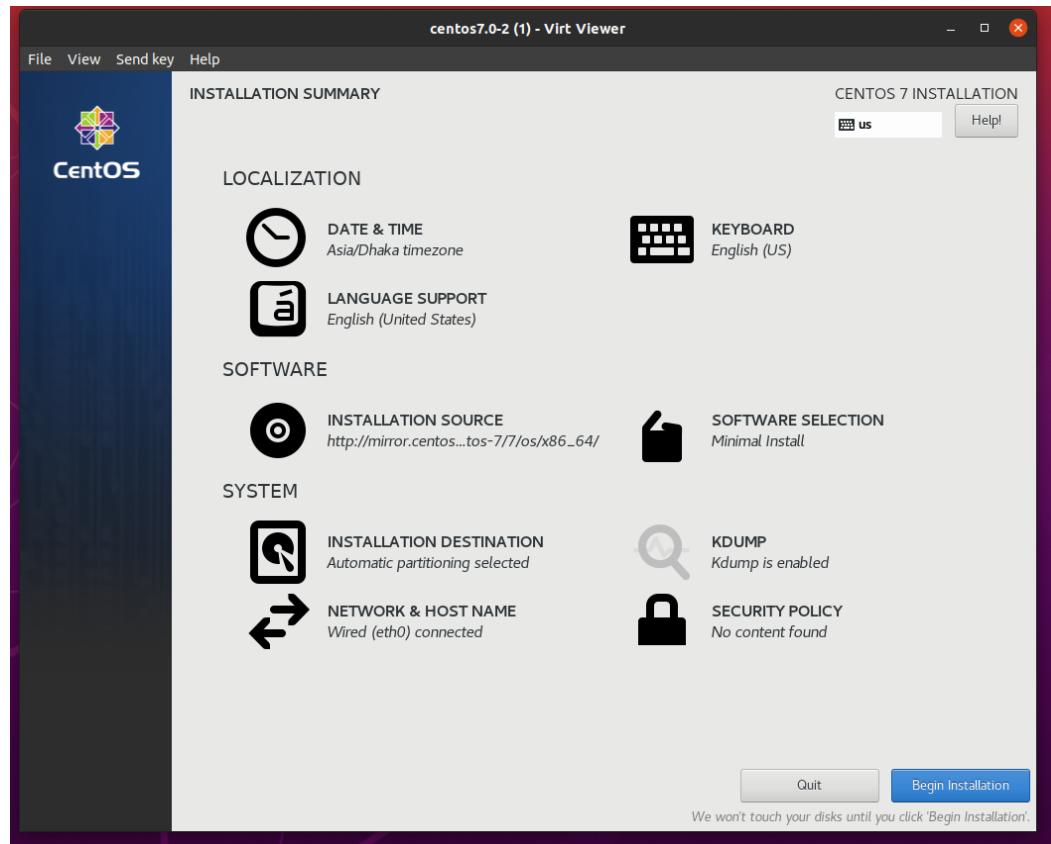
muballigh@hossain-19101289:~$ virt-install \
--connect qemu:///system \
--memory 8192 \
--vcpus 8 \
--graphics vnc \
--os-variant centos7.0 \
--location http://mirror.centos.org/centos-7/7/os/x86_64/
Using default -name centos7.0
Using centos7.0 default -disk size=20
Starting install...
Retrieving file vmlinuz...
Retrieving file initrd.img...
Allocating 'virtinst-vjhu4ex-vmlinuz' | 6.5 MB 00:02
Transferring virtinst-vjhu4ex-vmlinuz | 6.5 MB 00:00
Allocating 'virtinst-m4spocj-initrd.img' | 53 MB 00:00
Transferring virtinst-m4spocj-initrd.img | 53 MB 00:00
Allocating 'centos7.0.qcow2' | 20 GB 00:03
[...]

```

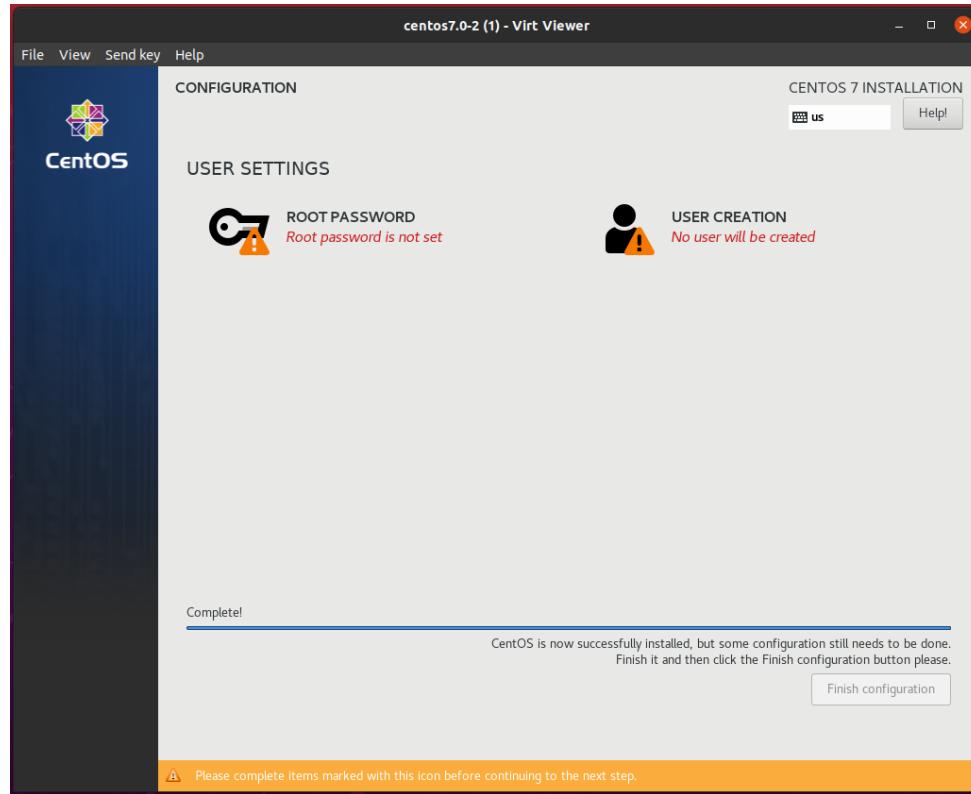
The terminal output continues with more log entries related to the installation process, including kernel boot messages and disk allocation details.

- Install CentOS like any other Linux Distros

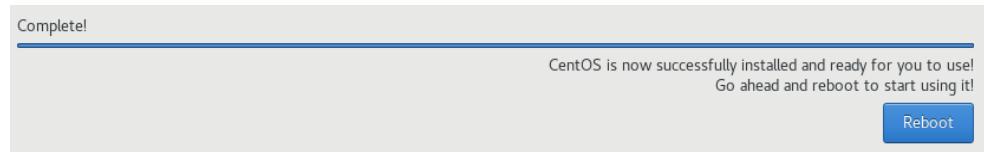




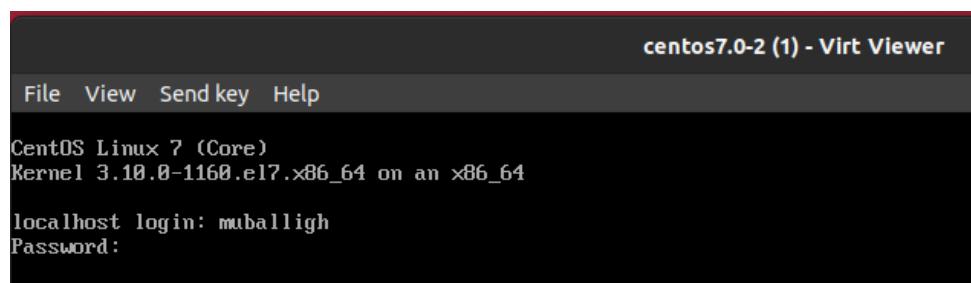
- Set User Information and Root Password



- Reboot the system



Enter the system using username and password



- See CPU Info using `cat /proc/cpuinfo`

```

centos7.0-2 (1) - Virt Viewer

cpu        : yes
cpu_exception : yes
cpu_id_level : 13
mp        : yes
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx
mcext fxsr_opt pdpe1gb rdtsvp la art rep_good nopl exld_apcid eagerfpu pni pclmulqdq ssse3 fmw cx16 sse4_1 sse4_2 x2apic mohc
popcnt tsc deadline_timer aes xsavec xss f1fc rdrand hypervisor lahf_lm cmp_legacy sse cr8 legacy abm sse4a misalignsse 3dnowpref
etcksm topoext perfctr_core retpoline_and_smmh Ibp8 stibp vncall fsgsbbase tsc_adjust bmi1 avx2 ssepm bm12 rdseed adx smpclif
tushtopf clwb sha_ni xsaveopt xssvec sgdtbv1 clzero xssveprtr arat nptr nrip_save umip intel_stibp arch_capabilities
bogomips   : 2106.49
TLB size    : 1024 4K pages
clfflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management:

processor : 7
vendor_id : authentechRMD
cpu family : 23
model    : 49
model name : AMD EPYC-Rome Processor
stepping : 4
microcode : 0x10000065
cpu MHz   : 3593.248
cache size : 512 KB
physical id : 7
siblings   : 1
core id   : 0
cpu cores : 1
apicid    : 7
initial apicid : 7
cpu        : yes
cpu_exception : yes
cpu_id_level : 13
mp        : yes
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx
mcext fxsr_opt pdpe1gb rdtsvp la art rep_good nopl exld_apcid eagerfpu pni pclmulqdq ssse3 fmw cx16 sse4_1 sse4_2 x2apic mohc
popcnt tsc deadline_timer aes xsavec xss f1fc rdrand hypervisor lahf_lm cmp_legacy sse cr8 legacy abm sse4a misalignsse 3dnowpref
etcksm topoext perfctr_core retpoline_and_smmh Ibp8 stibp vncall fsgsbbase tsc_adjust bmi1 avx2 ssepm bm12 rdseed adx smpclif
tushtopf clwb sha_ni xsaveopt xssvec sgdtbv1 clzero xssveprtr arat nptr nrip_save umip intel_stibp arch_capabilities
bogomips   : 2106.49
TLB size    : 1024 4K pages
clfflush size : 64
cache_alignment : 64
address sizes : 48 bits physical, 48 bits virtual
power management:

[mballigh@localhost ~]$
```

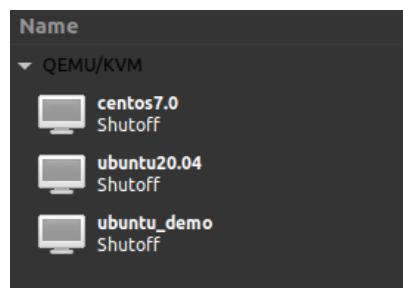
- See Memory Info using `cat /proc/meminfo`

```

centos7.0-2 (1) - Virt Viewer

MemTotal:       8088844 kB
MemFree:        7438212 kB
MemAvailable:  7391288 kB
Buffers:         1024 kB
Cached:        178348 kB
SwapCached:      0 kB
Active:        115988 kB
Inactive:      115312 kB
Active(anon):  6316 kB
Inactive(anon): 8536 kB
Active(file):  52876 kB
Inactive(file): 110776 kB
Unevictable:     0 kB
Migrable:        0 kB
SleepTotal:    2897148 kB
SwapFree:       2897148 kB
Dirty:          0 kB
Writeback:       0 kB
AnonPages:      62949 kB
Mapped:        24268 kB
Shmem:          6884 kB
Slab:           68252 kB
SpecialKmable: 24444 kB
NormalKmable:  53230 kB
KernelStack:   2944 kB
PageTables:    3912 kB
NFS_Unstable:   0 kB
CommitLimit:   6191168 kB
Committed_AS:  388844 kB
VmallocTotal:  34359738367 kB
VmallocUsed:   416384 kB
VmallocChunk:  353520 kB
Percpu:         2088 kB
HardwareCorrupted: 0 kB
AnonHugePages:  6144 kB
CmaTotal:       0 kB
CmaFree:        0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize:   2048 kB
DirectMap4G:    147312 kB
DirectMap2M:   5895424 kB
DirectMap1G:   5242888 kB
[mballigh@localhost ~]$
```

- So as we can see, our CentOS is properly configured. The `centos7.0` & `ubuntu_demo` were created using the CLI & `ubuntu20.04` was created using VMM GUI.



Complete set of commands and their explanation can be found here -
<https://manpages.ubuntu.com/manpages/focal/man1/virt-install.1.html>

Commands -

Ubuntu :

```
virt-install \
    --connect qemu:///system \
    --virt-type kvm \
    --name ubuntu_demo \
    --memory 2048 \
    --disk size=25 \
    --cdrom /home/muballigh/Downloads/ubuntu-22.04.1-desktop-amd64.iso \
    --os-variant ubuntu20.04
```

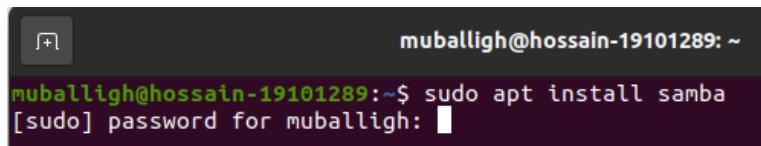
CentOS:

```
virt-install \
    --connect qemu:///system \
    --memory 8192 \
    --vcpus 8 \
    --graphics vnc \
    --os-variant centos7.0 \
    --location http://mirror.centos.org/centos-7/7/os/x86\_64/
```

Task 5 – Shared Folder Between Host and Guest Machine

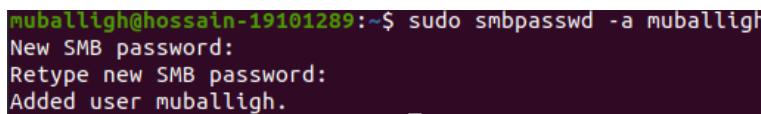
We will do this using Samba.

- Installing Samba in Host OS using CLI



```
muballigh@hossain-19101289:~$ sudo apt install samba
[sudo] password for muballigh: [REDACTED]
```

- Setting Password for Samba



```
muballigh@hossain-19101289:~$ sudo smbpasswd -a muballigh
New SMB password:
Retype new SMB password:
Added user muballigh.
```

- Creating a folder that will be shared

```
muballigh@hossain-19101289:~$ cd /
muballigh@hossain-19101289:/$ ls
bin dev lib libx32 mnt root snap sys var
boot etc lib32 lost+found opt run srv tmp
cdrom home lib64 media proc sbin swapfile usr
muballigh@hossain-19101289:/$ mkdir /home/muballigh/samba
```

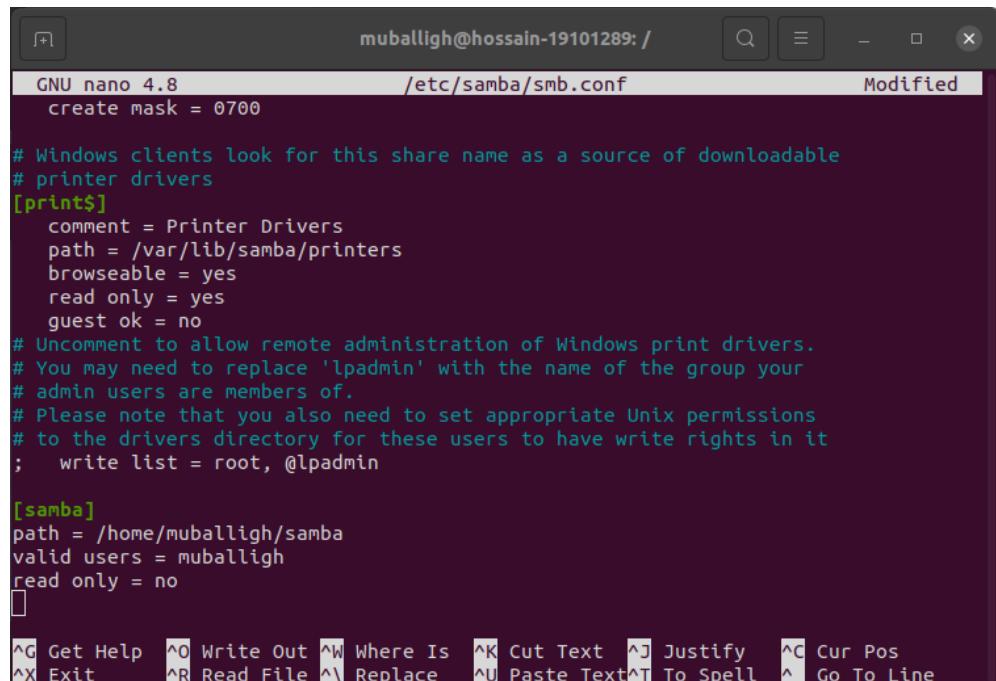
- Creating a backup as a standard practice

```
muballigh@hossain-19101289:/$ sudo cp /etc/samba/smb.conf ~
```

- Modifying the configuration file

```
muballigh@hossain-19101289:/$ sudo nano /etc/samba/smb.conf
```

- Adding path and user and setting permissions



```
GNU nano 4.8          /etc/samba/smb.conf          Modified
create mask = 0700

# Windows clients look for this share name as a source of downloadable
# printer drivers
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no

# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
;   write list = root, @lpadmin

[samba]
path = /home/muballigh/samba
valid users = muballigh
read only = no
;
```

Toolbar:

- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^C Cur Pos
- ^X Exit
- ^R Read File
- ^V Replace
- ^U Paste Text
- ^T To Spell
- ^L Go To Line

- Restarting the samba service to enact the changes made.

```
muballigh@hossain-19101289:/$ sudo service smbd restart
```

- Checking if the conf file has any error or not using testparm

```
muballigh@hossain-19101289:/$ sudo testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed
WARNING: The 'netbios name' is too long (max. 15 chars).

Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions
```

- Installing smbclient using CLI

```
muballigh@hossain-19101289:/$ sudo apt install smbclient
```

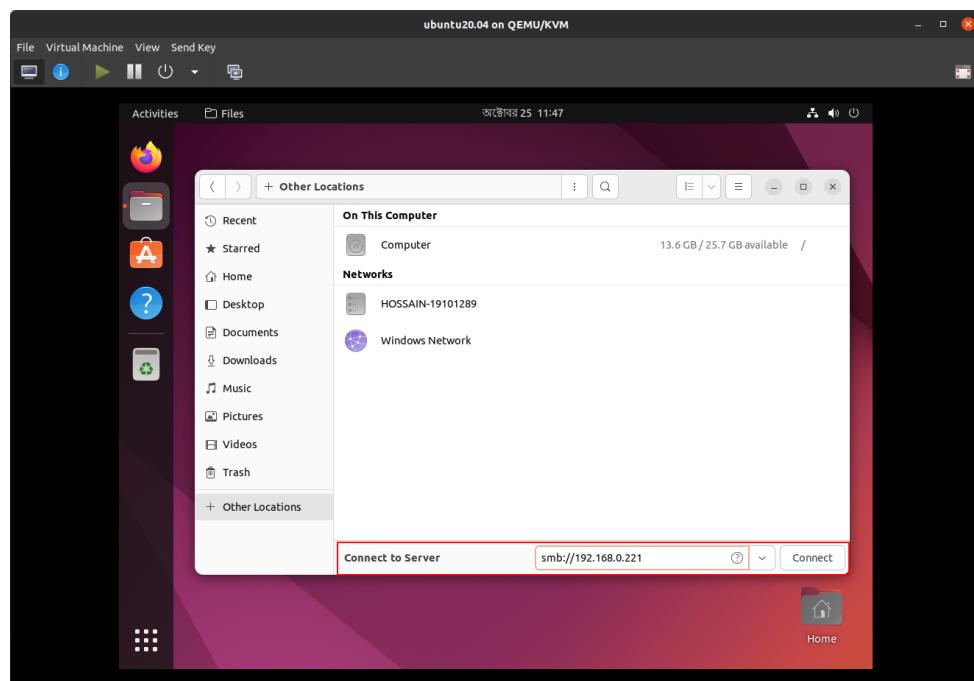
- Installing net-tools to see IP and MAC address

```
muballigh@hossain-19101289:/$ sudo apt install -y net-tools  
Reading package lists... Done
```

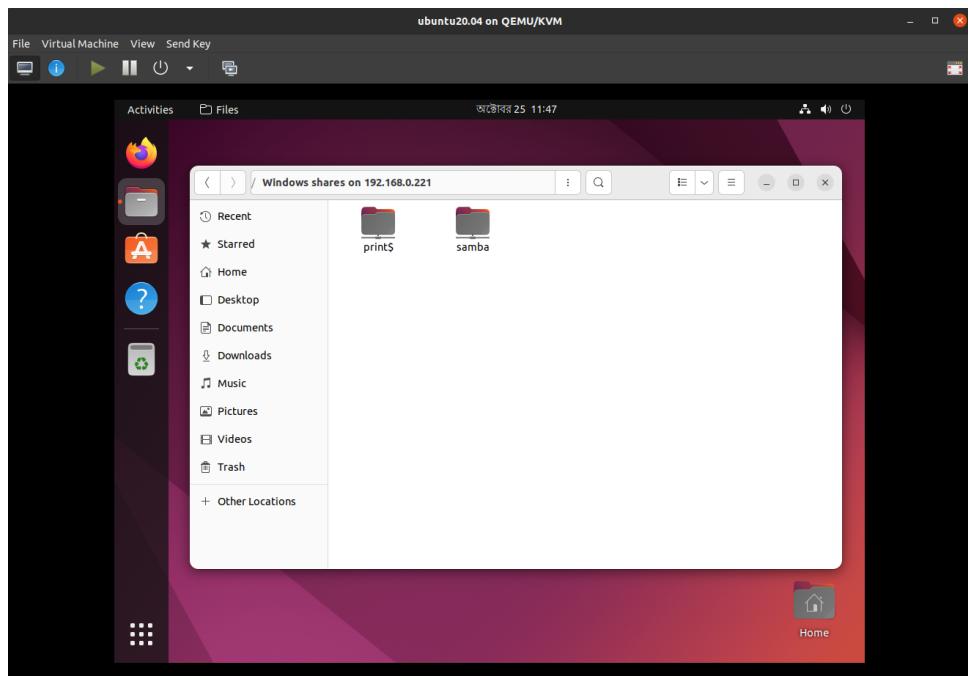
- Using ifconfig command to view IP address

```
muballigh@hossain-19101289:/$ ifconfig  
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
      inet 192.168.0.221 netmask 255.255.255.0 broadcast 192.168.0.255  
        inet6 fe80::7c61:1757:be59 prefixlen 64 scopeid 0x20<link>
```

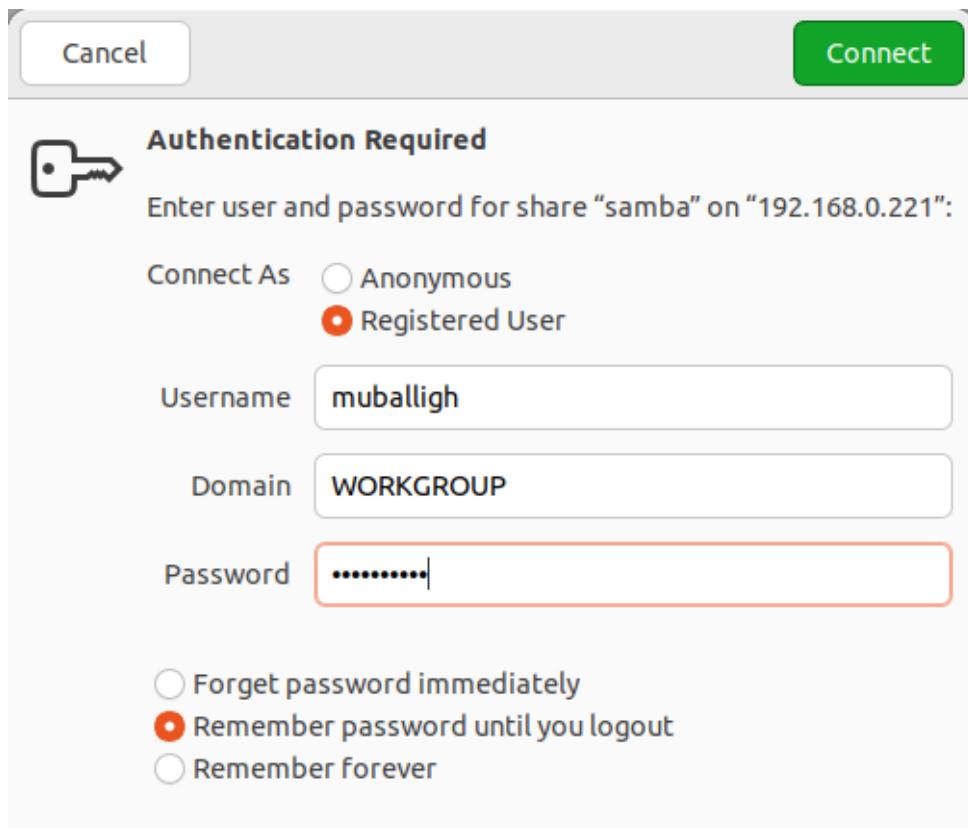
- Connecting to the samba folder from Guest OS using the IP of Host OS



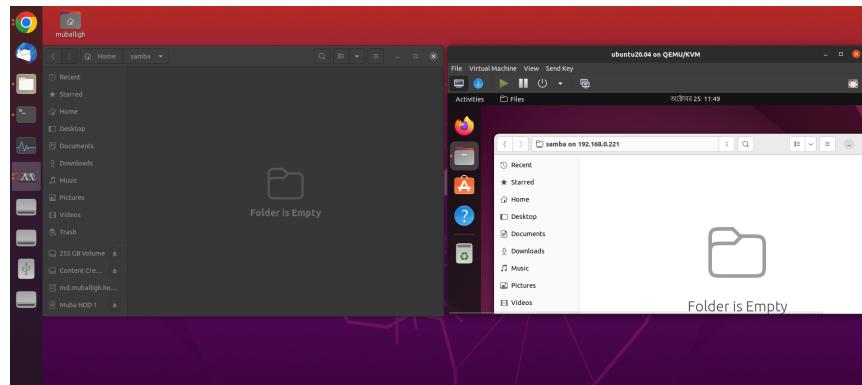
- Shared folders from the Host OS can be seen



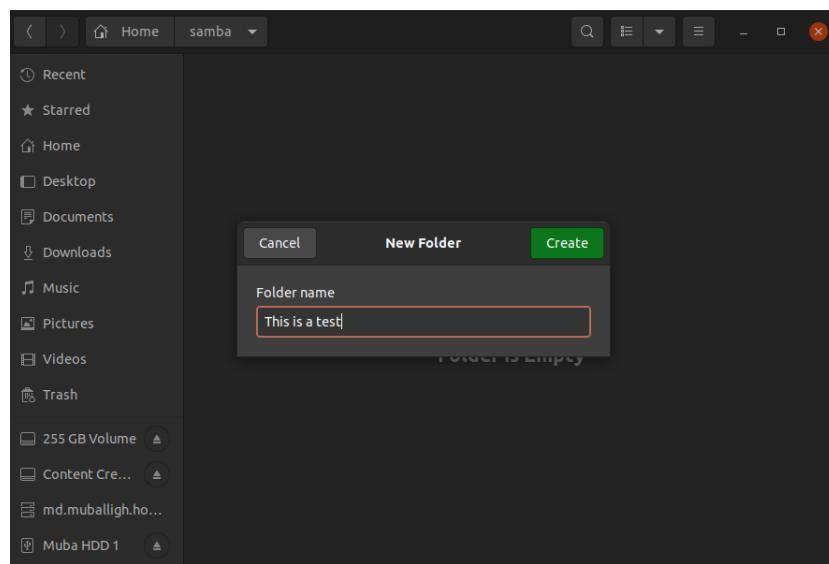
- In order to access the folder we need to input our credentials



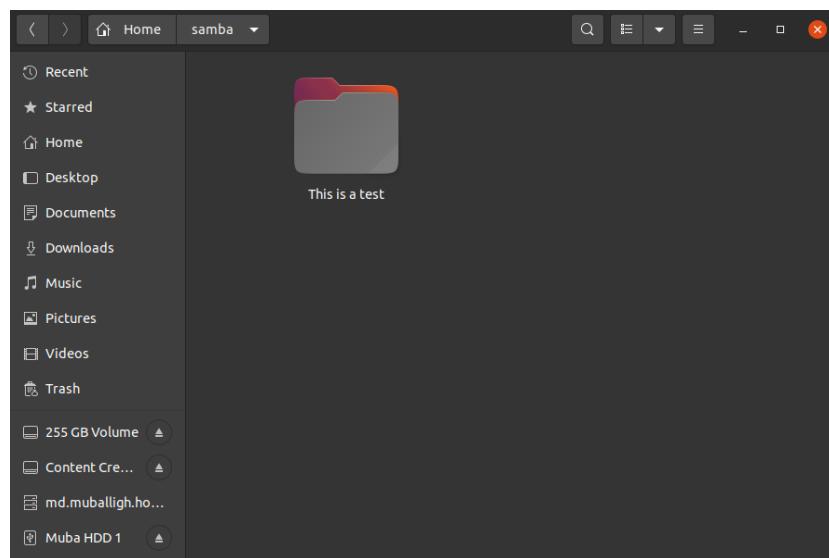
- Left one is Host OS and right is the Guest OS



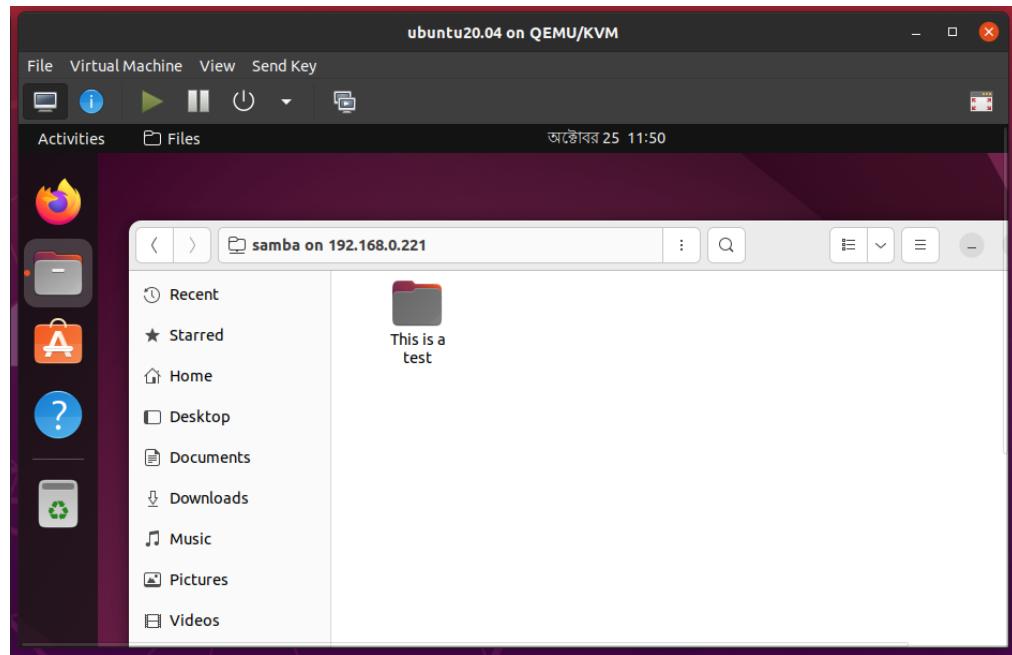
- Creating new folder from Host's end



- New folder in Host OS



- The test folder can also be seen from the Guest OS



Thus we have properly shared a folder between Host and Guest OS. If changes are made to the folder, it gets updated in both the ends.