Batch: A-4 **Roll No.:** 16010122151

Experiment No. 02

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Virtualization in Cloud. (KVM, Virtual Box/VMWAre)

Objective: To perform VM Creation on windows and KVM on Linux Based Operating System.

Expected Outcome of Experiment:

CO	Outcome
CO2	Investigate the system virtualization and outline its role in enabling the cloud computing System model

Books/ Journals/ Websites referred:

Abstract:-

This experiment explores the concept of virtualization in cloud computing, focusing on three popular hypervisors: KVM (Kernel-based Virtual Machine), VirtualBox, and VMware. The primary goal is to evaluate the performance, scalability, and resource management capabilities of each virtualization platform within a cloud environment. The experiment involves setting up virtual machines (VMs) on each hypervisor and analyzing their ability to manage hardware resources, network communication, and storage. Key metrics such as CPU utilization, memory consumption, and network throughput are measured under varying loads to compare the efficiency and performance of each hypervisor. The findings aim to provide insights into the advantages and limitations of each platform, helping organizations make informed decisions about virtualization technologies suitable for cloud infrastructure.

Related Theory: -

Virtualization is a key technology in cloud computing that allows for the creation of virtual instances of computing resources, such as servers, storage, and networks. It enables efficient resource utilization, isolation, and flexibility within cloud environments. Virtualization involves abstracting the physical hardware to create multiple virtual machines (VMs) on a single host machine, allowing for more effective management and allocation of resources.

1. **Hypervisor Types:** There are two primary types of hypervisors used in virtualization:
    - **Type 1 (Bare-metal) Hypervisors:** These run directly on the host hardware and are typically more efficient due to their direct control over resources. Examples include VMware ESXi and KVM.
    - **Type 2 (Hosted) Hypervisors:** These run as applications on top of an existing operating system. Examples include VirtualBox and VMware Workstation. While they tend to have higher overhead, they are easier to set up and use for testing and development purposes.
2. **KVM (Kernel-based Virtual Machine):** KVM is a Type 1 hypervisor built into the Linux kernel. It leverages hardware virtualization extensions (Intel VT-x, AMD-V) to provide a robust environment for running multiple VMs on a Linux system. KVM's integration with the Linux kernel provides high performance, scalability, and security. It allows the use of various Linux distributions, Windows, and other operating systems as guest VMs.
3. **VirtualBox:** VirtualBox is a Type 2 hypervisor that runs on top of host operating systems like Windows, Linux, and macOS. It is open-source and widely used for personal or development environments. It supports a variety of guest operating systems and offers features like snapshots, seamless mode, and USB device support. VirtualBox is less efficient than KVM for large-scale

virtualization, but its ease of use makes it an excellent choice for smaller deployments or experimentation.

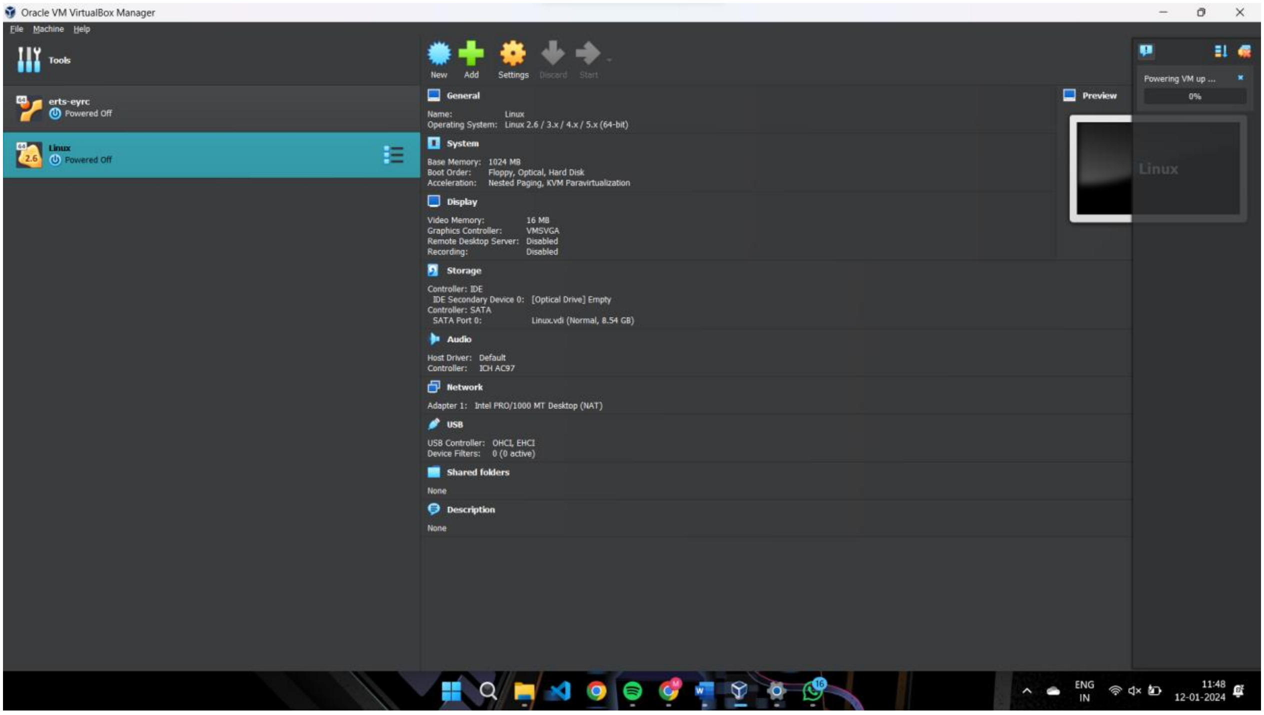
4. **VMware:** VMware offers both Type 1 and Type 2 hypervisors, with VMware ESXi being a Type 1 hypervisor and VMware Workstation being a Type 2 hypervisor. VMware is known for its enterprise-grade features, including robust management tools, VMotion (live migration of VMs), and DRS (Distributed Resource Scheduling). VMware's ESXi hypervisor is widely adopted in data centers due to its performance, scalability, and reliability.
5. **Cloud Virtualization and Resource Management:** Virtualization is the backbone of cloud computing, enabling resource pooling, efficient allocation, and dynamic scaling. Cloud providers use hypervisors to create isolated environments for each tenant, ensuring security and performance. Resource management techniques, such as CPU scheduling, memory overcommitment, and storage allocation, allow for optimal utilization of physical hardware while maintaining service quality for multiple users.
6. **Performance Metrics in Virtualization:** When evaluating the performance of virtualized environments, key metrics include:
   - **CPU Utilization:** Measures how efficiently the virtual machines are using the host's processing power.
   - **Memory Usage:** Indicates the memory consumption of the VMs and the host, highlighting any resource bottlenecks.
   - **Network Throughput:** Assesses the speed and efficiency of data transfer between virtual machines and across the host system.
   - **Storage I/O Performance:** Evaluates the read/write speeds and storage utilization under different workloads.

Understanding the underlying theory of virtualization technologies, their resource management strategies, and performance metrics is essential for making informed decisions about which hypervisor to use in cloud infrastructure.

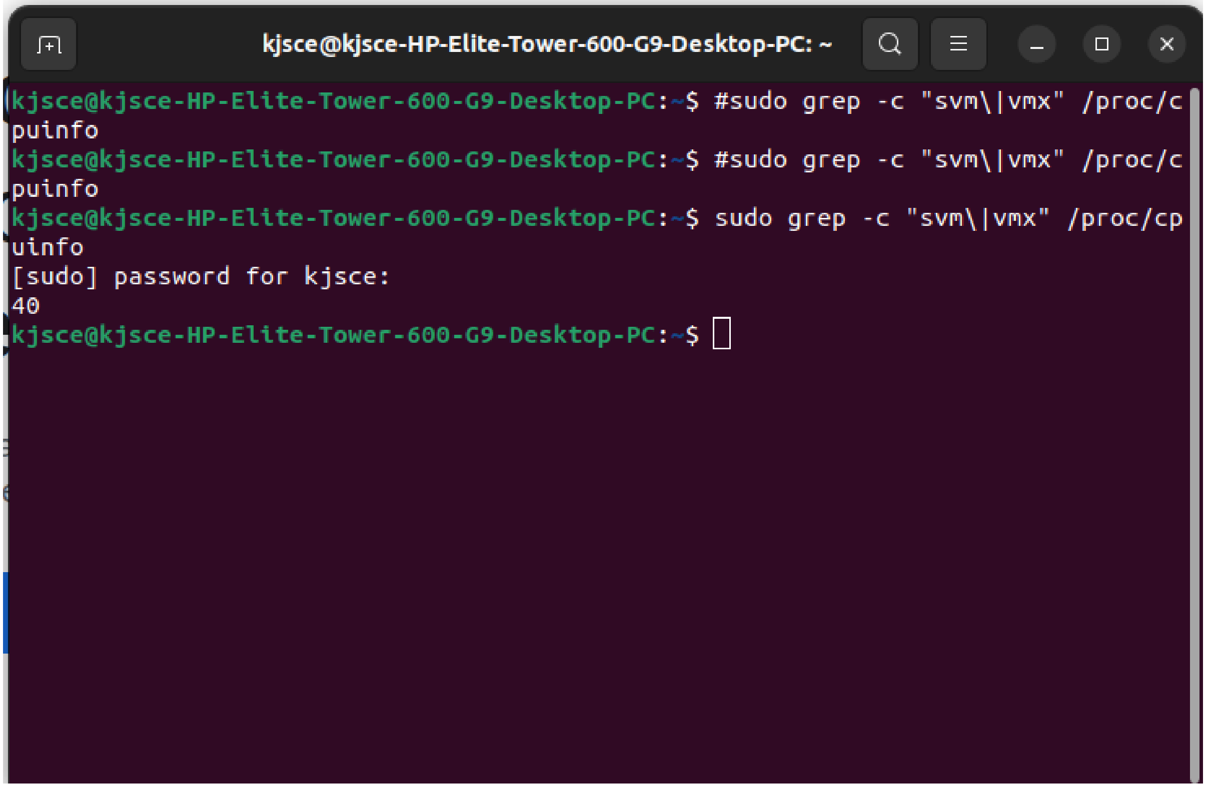
Implementation Details:

Installing VirtualBox:

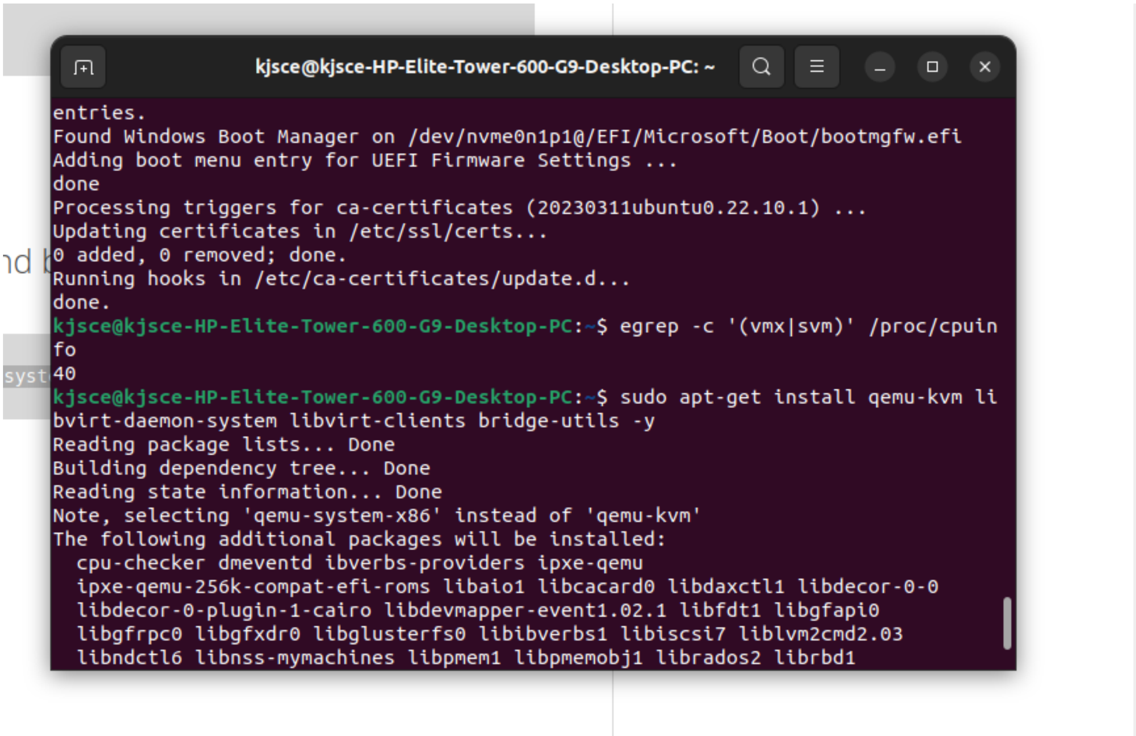
Creating Virtual Machine:

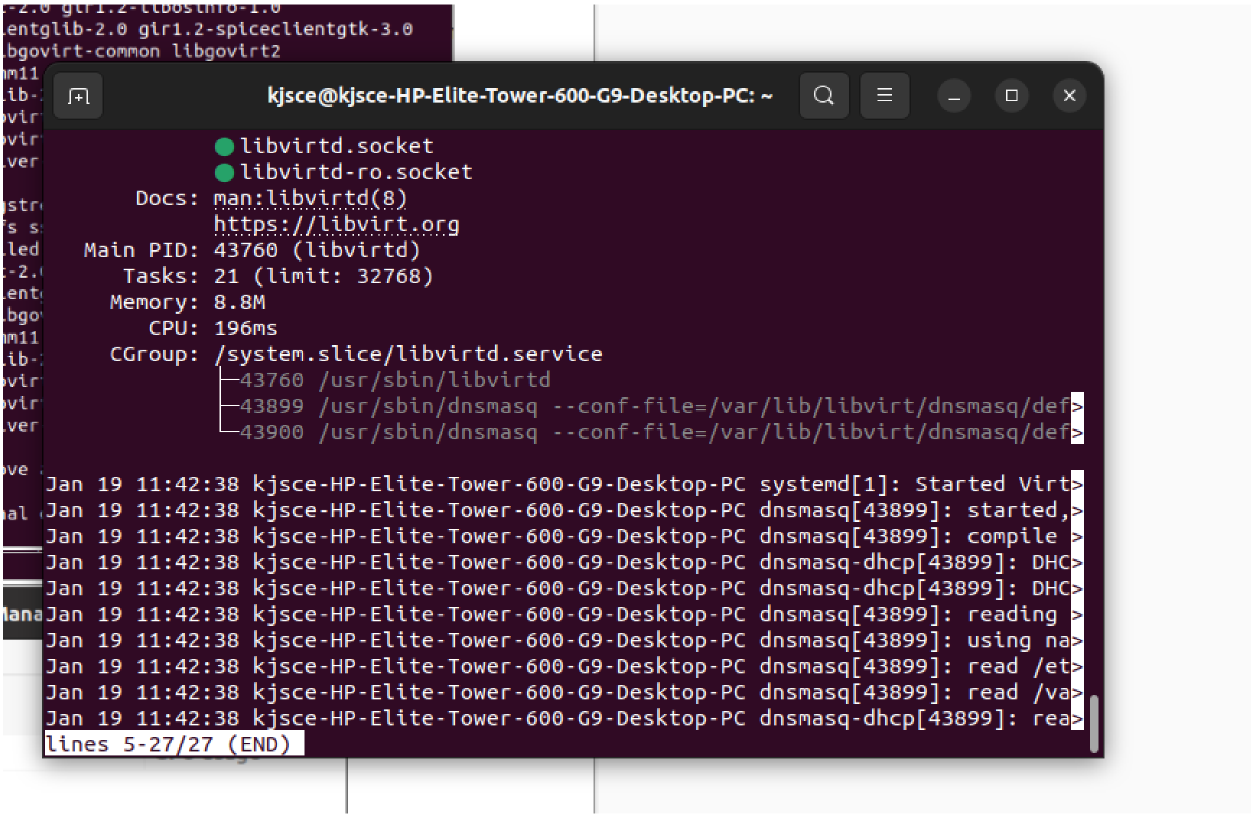


Checking if svm or vmx is activated:

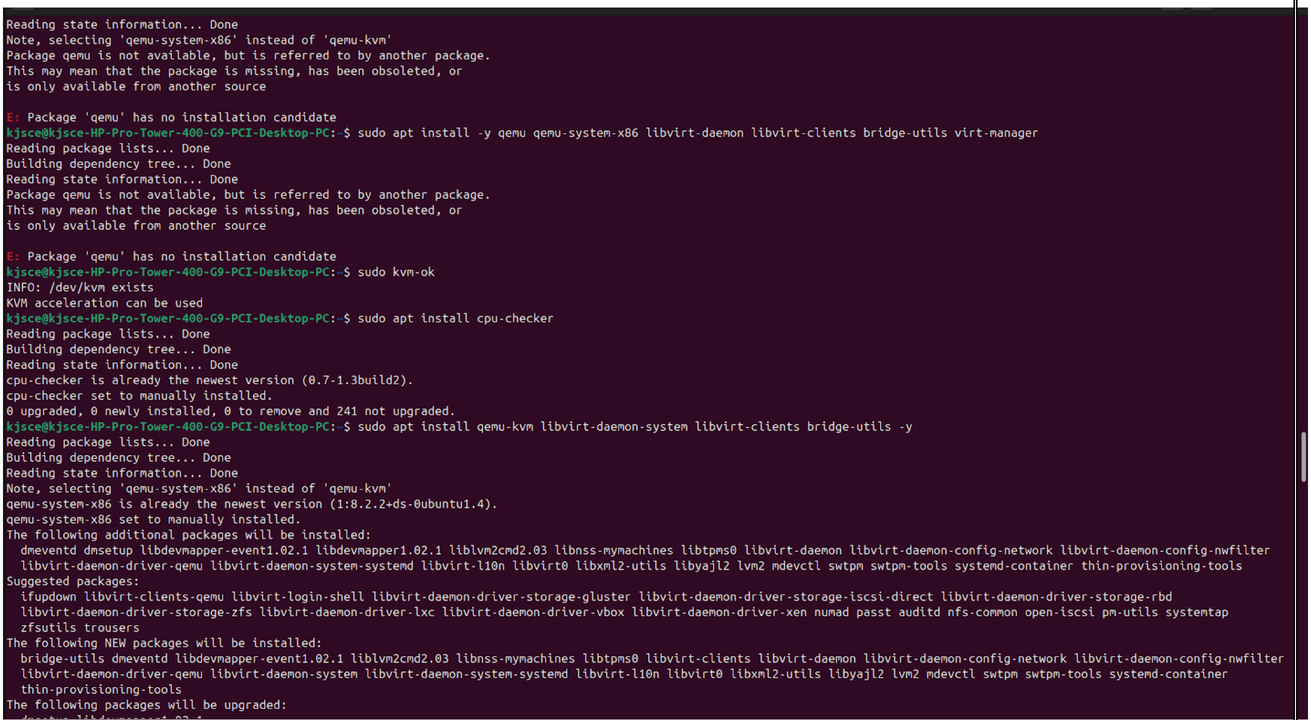


Installing KVM:

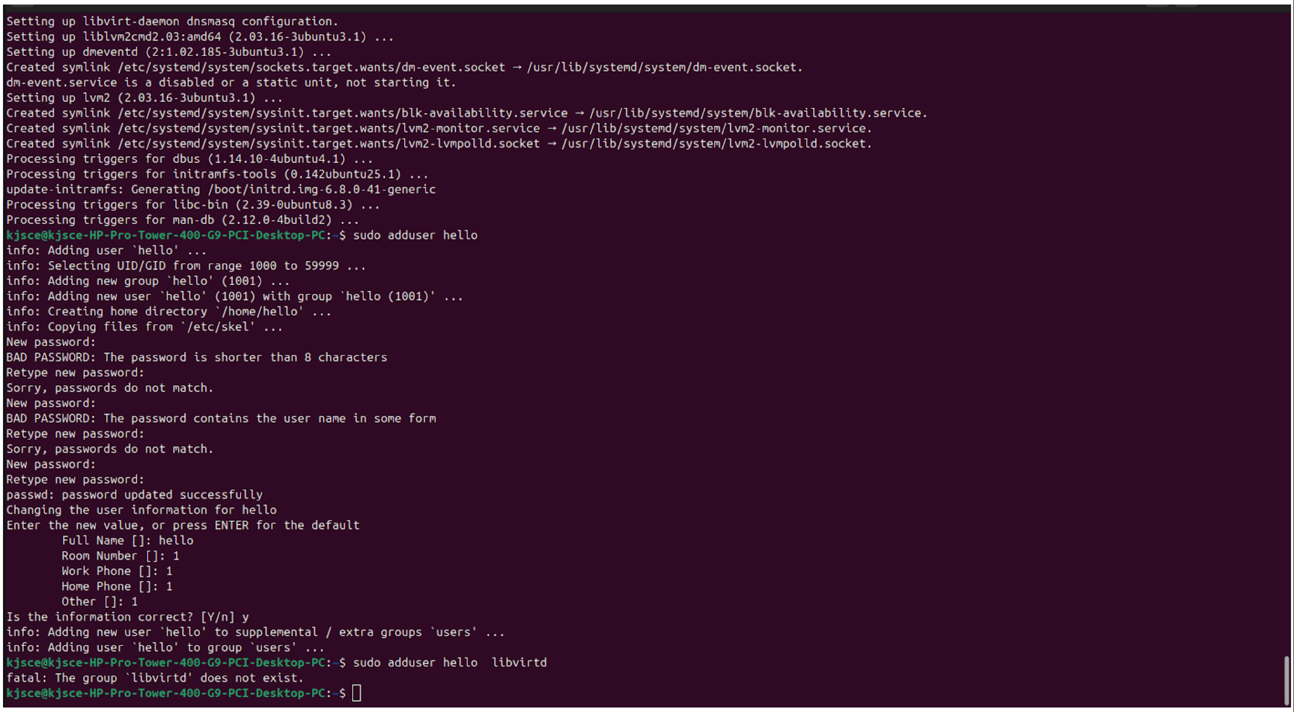


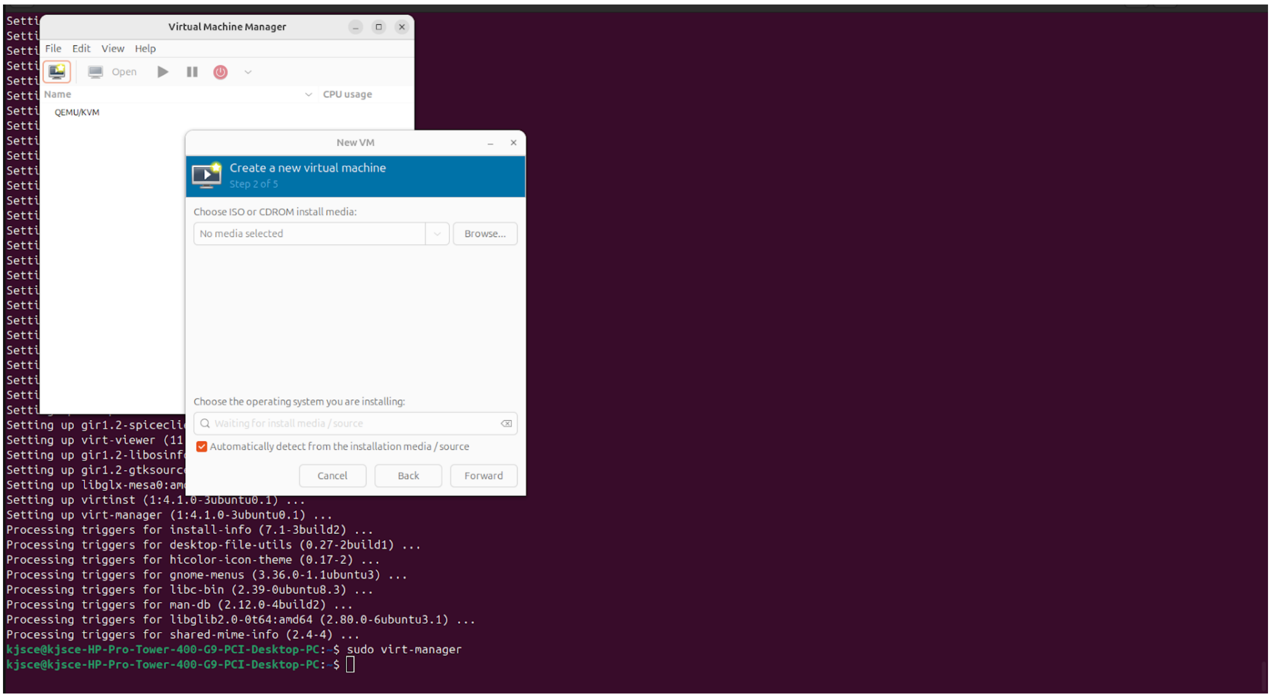


```
kjsce@kjsce-HP-Elite-Tower-600-G9-Desktop-PC: ~
             ● libvirtd.socket
             ● libvirtd-ro.socket
       Docs: man:libvirtd(8)
             https://libvirt.org
   Main PID: 43760 (libvirtd)
      Tasks: 21 (limit: 32768)
     Memory: 8.8M
        CPU: 196ms
     CGroup: /system.slice/libvirtd.service
             ├─43760 /usr/sbin/libvirtd
             ├─43899 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/def>
             └─43900 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dnsmasq/def>

Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC systemd[1]: Started Virt>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: started,>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: compile >
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq-dhcp[43899]: DHC>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq-dhcp[43899]: DHC>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: reading >
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: using na>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: read /et>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq[43899]: read /va>
Jan 19 11:42:38 kjsce-HP-Elite-Tower-600-G9-Desktop-PC dnsmasq-dhcp[43899]: rea>
lines 5-27/27 (END)
```



```
Reading state information... Done
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'
Package qemu is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'qemu' has no installation candidate
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo apt install -y qemu qemu-system-x86 libvirt-daemon libvirt-clients bridge-utils virt-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package qemu is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'qemu' has no installation candidate
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo apt install cpu-checker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cpu-checker is already the newest version (0.7-1.3build2).
cpu-checker set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 241 not upgraded.
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'
qemu-system-x86 is already the newest version (1:8.2.2+ds-0ubuntu1.4).
qemu-system-x86 set to manually installed.
The following additional packages will be installed:
  dmeventd dmsetup libdevmapper-event1.02.1 libdevmapper1.02.1 liblvm2cmd2.03 libnss-mymachines libtpms0 libvirt-daemon libvirt-daemon-config-network libvirt-daemon-config-nwfilter
  libvirt-daemon-driver-qemu libvirt-daemon-system-systemd libvirt-l10n libvirt0 libxml2-utils libyajl2 lvm2 mdevctl swtpm swtpm-tools systemd-container thin-provisioning-tools
Suggested packages:
  ifupdown libvirt-clients-qemu libvirt-login-shell libvirt-daemon-driver-storage-gluster libvirt-daemon-driver-storage-iscsi-direct libvirt-daemon-driver-storage-rbd
  libvirt-daemon-driver-storage-zfs libvirt-daemon-driver-lxc libvirt-daemon-driver-vbox libvirt-daemon-driver-xen numad passt auditd nfs-common open-iscsi pm-utils systemtap
  zfsutils trousers
The following NEW packages will be installed:
  bridge-utils dmeventd libdevmapper-event1.02.1 liblvm2cmd2.03 libnss-mymachines libtpms0 libvirt-clients libvirt-daemon libvirt-daemon-config-network libvirt-daemon-config-nwfilter
  libvirt-daemon-driver-qemu libvirt-daemon-system libvirt-daemon-system-systemd libvirt-l10n libvirt0 libxml2-utils libyajl2 lvm2 mdevctl swtpm swtpm-tools systemd-container
  thin-provisioning-tools
The following packages will be upgraded:
```

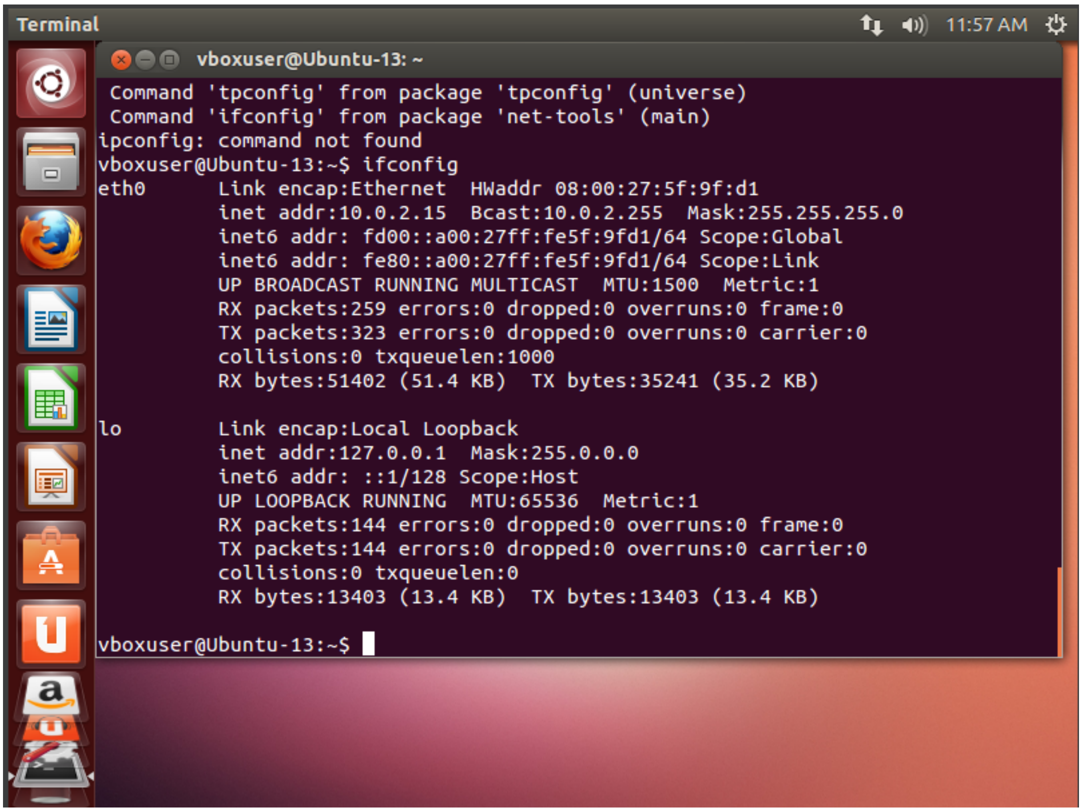

Setting up libvirt-daemon dnsmasq configuration.
Setting up liblvm2cmd2.03:amd64 (2.03.16-3ubuntu3.1) ...
Setting up dmeventd (2:1.02.185-3ubuntu3.1) ...
Created symlink /etc/systemd/system/sockets.target.wants/dm-event.socket → /usr/lib/systemd/system/dm-event.socket.
dm-event.service is a disabled or a static unit, not starting it.
Setting up lvm2 (2.03.16-3ubuntu3.1) ...
Created symlink /etc/systemd/system/sysinit.target.wants/blk-availability.service → /usr/lib/systemd/system/blk-availability.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-monitor.service → /usr/lib/systemd/system/lvm2-monitor.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket → /usr/lib/systemd/system/lvm2-lvmpolld.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for initramfs-tools (0.142ubuntu25.1) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-41-generic
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo adduser hello
info: Adding user `hello' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `hello' (1001) ...
info: Adding new user `hello' (1001) with group `hello (1001)' ...
info: Creating home directory `/home/hello' ...
info: Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hello
Enter the new value, or press ENTER for the default
Full Name []: hello
Room Number []: 1
Work Phone []: 1
Home Phone []: 1
Other []: 1
Is the information correct? [Y/n] y
info: Adding new user `hello' to supplemental / extra groups `users' ...
info: Adding user `hello' to group `users' ...
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo adduser hello libvirtd
fatal: The group `libvirtd' does not exist.
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$

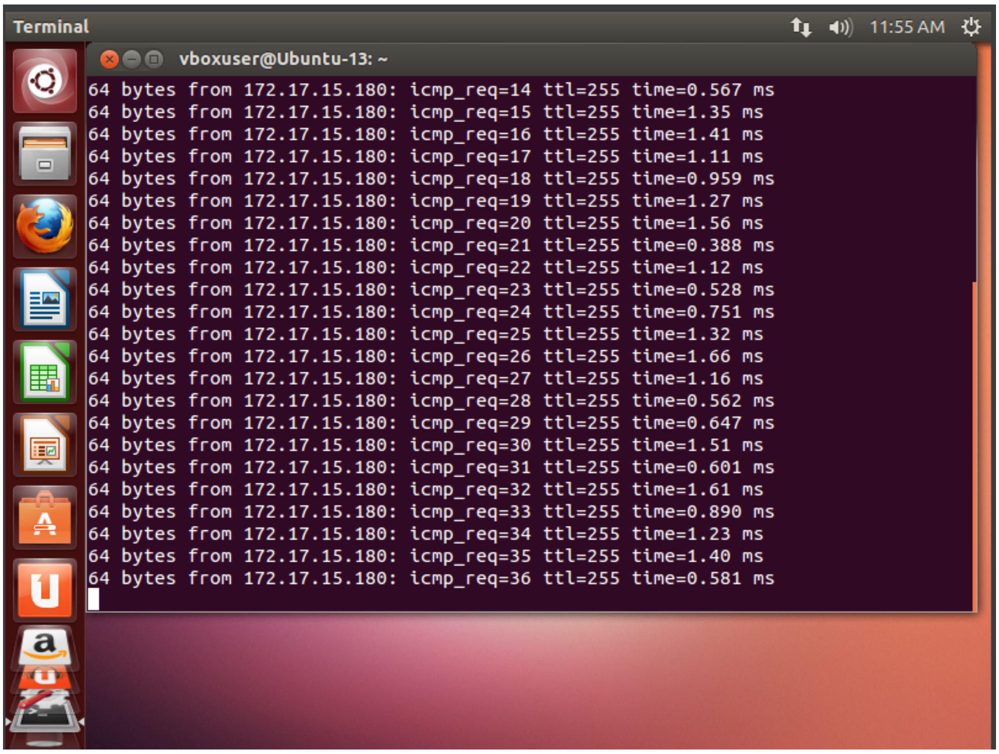

Virtual Machine Manager
File Edit View Help
Open
Name
CPU usage
QEMU/KVM
New VM
Create a new virtual machine
Step 2 of 5
Choose ISO or CDROM install media:
No media selected
Browse...
Choose the operating system you are installing:
Waiting for install media / source
Automatically detect from the installation media / source
Cancel
Back
Forward
Setting up virtinst (1:4.1.0-3ubuntu0.1) ...
Setting up virt-manager (1:4.1.0-3ubuntu0.1) ...
Processing triggers for install-info (7.1-3build2) ...
Processing triggers for desktop-file-utils (0.27-2build1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1.1ubuntu3) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libglib2.0-0t64:amd64 (2.80.0-6ubuntu3.1) ...
Processing triggers for shared-mime-info (2.4-4) ...
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$ sudo virt-manager
kjsce@kjsce-HP-Pro-Tower-400-G9-PCI-Desktop-PC:~$

Pinging from windows to Ubuntu

```
PS C:\Users\KJSCE> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::de5:d159:2a91:bb31%2
   IPv4 Address. . . . . . . . . . . : 172.17.15.180
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 172.17.15.254

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::50cf:1a65:47c1:6657%13
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Wi-Fi:
```

```
PS C:\Users\KJSCE> ping 10.0.2.255

Pinging 10.0.2.255 with 32 bytes of data:
Reply from 10.0.2.255: bytes=32 time=21ms TTL=62
Reply from 10.0.2.255: bytes=32 time=171ms TTL=62
Reply from 10.0.2.255: bytes=32 time=16ms TTL=62
Reply from 10.0.2.255: bytes=32 time=17ms TTL=62

Ping statistics for 10.0.2.255:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 16ms, Maximum = 171ms, Average = 56ms
PS C:\Users\KJSCE>
```


Terminal
11:57 AM
vboxuser@Ubuntu-13: ~
 Command 'tpconfig' from package 'tpconfig' (universe)
 Command 'ifconfig' from package 'net-tools' (main)
ipconfig: command not found
vboxuser@Ubuntu-13:~$ ifconfig
eth0   Link encap:Ethernet HWaddr 08:00:27:5f:9f:d1
          inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fd00::a00:27ff:fe5f:9fd1/64 Scope:Global
          inet6 addr: fe80::a00:27ff:fe5f:9fd1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:259 errors:0 dropped:0 overruns:0 frame:0
          TX packets:323 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:51402 (51.4 KB) TX bytes:35241 (35.2 KB)
lo   Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:144 errors:0 dropped:0 overruns:0 frame:0
          TX packets:144 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13403 (13.4 KB) TX bytes:13403 (13.4 KB)
vboxuser@Ubuntu-13:~$


Terminal
11:55 AM
vboxuser@Ubuntu-13: ~
64 bytes from 172.17.15.180: icmp_req=14 ttl=255 time=0.567 ms
64 bytes from 172.17.15.180: icmp_req=15 ttl=255 time=1.35 ms
64 bytes from 172.17.15.180: icmp_req=16 ttl=255 time=1.41 ms
64 bytes from 172.17.15.180: icmp_req=17 ttl=255 time=1.11 ms
64 bytes from 172.17.15.180: icmp_req=18 ttl=255 time=0.959 ms
64 bytes from 172.17.15.180: icmp_req=19 ttl=255 time=1.27 ms
64 bytes from 172.17.15.180: icmp_req=20 ttl=255 time=1.56 ms
64 bytes from 172.17.15.180: icmp_req=21 ttl=255 time=0.388 ms
64 bytes from 172.17.15.180: icmp_req=22 ttl=255 time=1.12 ms
64 bytes from 172.17.15.180: icmp_req=23 ttl=255 time=0.528 ms
64 bytes from 172.17.15.180: icmp_req=24 ttl=255 time=0.751 ms
64 bytes from 172.17.15.180: icmp_req=25 ttl=255 time=1.32 ms
64 bytes from 172.17.15.180: icmp_req=26 ttl=255 time=1.66 ms
64 bytes from 172.17.15.180: icmp_req=27 ttl=255 time=1.16 ms
64 bytes from 172.17.15.180: icmp_req=28 ttl=255 time=0.562 ms
64 bytes from 172.17.15.180: icmp_req=29 ttl=255 time=0.647 ms
64 bytes from 172.17.15.180: icmp_req=30 ttl=255 time=1.51 ms
64 bytes from 172.17.15.180: icmp_req=31 ttl=255 time=0.601 ms
64 bytes from 172.17.15.180: icmp_req=32 ttl=255 time=1.61 ms
64 bytes from 172.17.15.180: icmp_req=33 ttl=255 time=0.890 ms
64 bytes from 172.17.15.180: icmp_req=34 ttl=255 time=1.23 ms
64 bytes from 172.17.15.180: icmp_req=35 ttl=255 time=1.40 ms
64 bytes from 172.17.15.180: icmp_req=36 ttl=255 time=0.581 ms

Conclusion:-

In conclusion, hardware-assisted virtualization technologies like SVM (Secure Virtual Machine) for AMD processors and VMX (Virtual Machine Extensions) for Intel processors play a crucial role in enhancing the performance, efficiency, and security of virtualized environments. By enabling direct hardware access for virtual machines, these technologies allow hypervisors to manage multiple virtual machines more effectively, with reduced overhead and improved resource utilization. Both SVM and VMX are fundamental in modern cloud computing, enabling scalable, isolated, and efficient virtual infrastructures.