NAME: Mohit Narang

CLASS: TE-C24

ROLL NO: 2103124

Experiment 2: To study and implement Hosted Virtualization using VirtualBox& KVM.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment - 2

Aim = To Study and implement hosted virtualization using virtual Box and

* Virtualization in cloud computing
Virtualization is a technique how to separate a service from the underlying physical delivery of that service It's a process of creating a virtual version of something like computer hardw

* Benifits of virtualization.
→ More flexible and efficient allocation of resources
→ Enshance development productivity
→ It lowers the cost of IT infrastructure.
→ Remote access and rapid scalability
→ High availbility and distaster recovery
→ Pay peruse of the IT infrastructure dema
→ Enables running multiple operating system.

* What is hypervisor give examples?
A hypervisor is a from of virtualization software used in cloud hosting to divide and allocate the resources on various pieces of hardware.
Ex = Virtual Box Microsoft Hyper-V, KVM, Xen Windows Virtual PC.

* Explain the types of hypervisor
-Type 1 hypervisor:-
This kind of hypervisor runs directly on the underlyin host system. It is also known as a native hypervisor or Bare metal hyperrisor. It does not requrie any base Server operating system.
Ex= Citix xenserver Micro soft Hyper-v hypervisor

→ Type 2 hypervisor :-
A host operating system runs on the underlying host system. It is also known as hosted hypervisor such kin of hypervisor dose'nt run directly over the underlying hardware rather they run as an application in a host system (physical Machine)

Comparisons between virtual Box and kvm

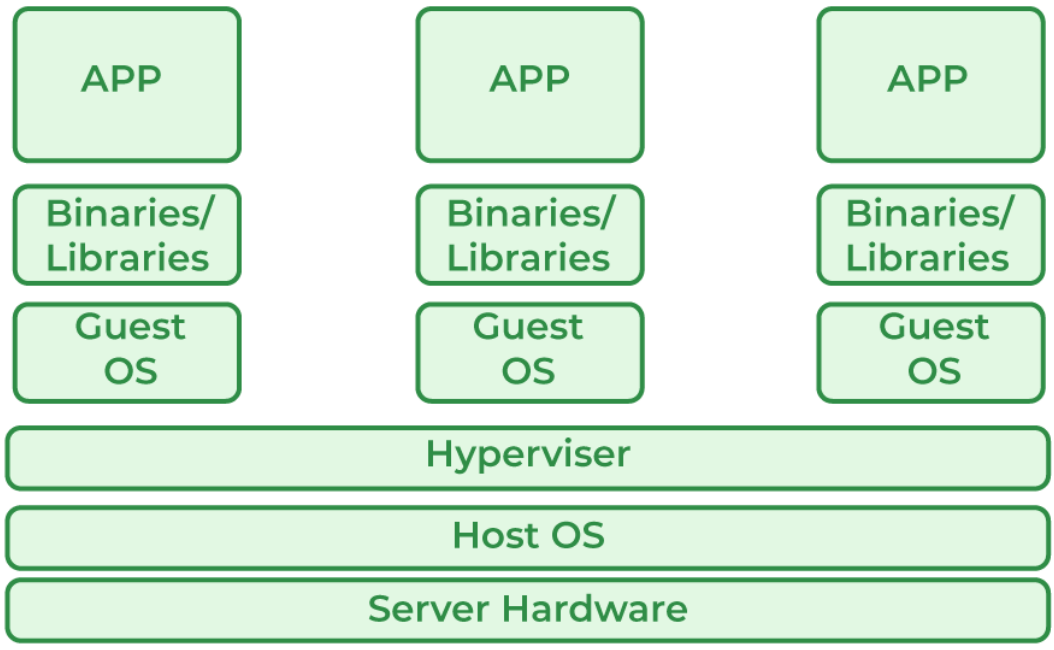
virtual Box	kvm
→ virtual Box is type 2 hypervisor	kvm is type 1 hypervisor which provides better performance.
→ virtual Box provides a user-friendly graphical interface that makes it easier for beginners to manage virtual machines	kvm is managed through command line tools and APIs. Making it suitable for advanced users and system administrators
→ virtual Box although it supports a wide range of hardware doesn't have the same level of hardware compatibility as kvm.	kvm has better hardware support including features like nested virtualization which allows running virtual machines within virtual machines.
→ virtual Box has limitations in resources management which is less suitable in resource intensive application.	kvm allows for dynamic allocation of resource like cpu and memory to virtual Machine

Experiment 2: To study and implement Hosted Virtualization using VirtualBox& KVM.

Virtualization in Cloud Computing and Types

Virtualization is a technique of how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on the same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

In other words, one of the main cost-effective, hardware-reducing, and energy-saving techniques used by cloud providers is Virtualization. Virtualization allows sharing of a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to physical storage and providing a pointer to that physical resource on demand. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.



Virtualization

Virtualization

- Host Machine: The machine on which the virtual machine is going to be built is known as Host Machine.
- Guest Machine: The virtual machine is referred to as a Guest Machine.

Work of Virtualization in Cloud Computing

Virtualization has a prominent impact on Cloud Computing. In the case of cloud computing, users store data in the cloud, but with the help of Virtualization, users have the extra benefit of sharing the infrastructure. Cloud Vendors take care of the required physical resources, but these cloud providers charge a huge amount for these services which impacts every user or organization. Virtualization helps Users or Organisations in maintaining those services which are required by a company through external (third-party) people, which helps in reducing costs to the company. This is the way through which Virtualization works in Cloud Computing.

Benefits of Virtualization

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.

- Pay peruse of the IT infrastructure on demand.
- Enables running multiple operating systems.

Drawback of Virtualization

- High Initial Investment: Clouds have a very high initial investment, but it is also true that it will help in reducing the cost of companies.
- Learning New Infrastructure: As the companies shifted from Servers to Cloud, it requires highly skilled staff who have skills to work with the cloud easily, and for this, you have to hire new staff or provide training to current staff.
- Risk of Data: Hosting data on third-party resources can lead to putting the data at risk, it has the chance of getting attacked by any hacker or cracker very easily.

Hypervisor

A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware. The program which provides partitioning, isolation, or abstraction is called a virtualization hypervisor. The hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. A hypervisor is sometimes also called a virtual machine manager(VMM).

Types of Hypervisor –

TYPE-1 Hypervisor:
The hypervisor runs directly on the underlying host system. It is also known as a “Native Hypervisor” or “Bare metal hypervisor”. It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor.

Pros & Cons of Type-1 Hypervisor:

Pros: Such kinds of hypervisors are very efficient because they have direct access to the physical hardware resources(like Cpu, Memory, Network, and Physical storage). This causes the empowerment of the security because there is nothing

any kind of the third party resource so that attacker couldn't compromise with anything.

Cons: One problem with Type-1 hypervisors is that they usually need a dedicated separate machine to perform their operation and to instruct different VMs and control the host hardware resources.

TYPE-2 Hypervisor:

A Host operating system runs on the underlying host system. It is also known as 'Hosted Hypervisor". Such kind of hypervisors doesn't run directly over the underlying hardware rather they run as an application in a Host system(physical machine). Basically, the software is installed on an operating system. Hypervisor asks the operating system to make hardware calls. An example of a Type 2 hypervisor includes VMware Player or Parallels Desktop. Hosted hypervisors are often found on endpoints like PCs. The type-2 hypervisor is very useful for engineers, and security analysts (for checking malware, or malicious source code and newly developed applications).

Pros & Cons of Type-2 Hypervisor:

Pros: Such kind of hypervisors allows quick and easy access to a guest Operating System alongside the host machine running. These hypervisors usually come with additional useful features for guest machines. Such tools enhance the coordination between the host machine and the guest machine.

Cons: Here there is no direct access to the physical hardware resources so the efficiency of these hypervisors lags in performance as compared to the type-1 hypervisors, and potential security risks are also there an attacker can compromise the security weakness if there is access to the host operating system so he can also access the guest operating system.

Choosing the right hypervisor :

Type 1 hypervisors offer much better performance than Type 2 ones because there's no middle layer, making them the logical choice for mission-critical applications and workloads. But that's not to say that hosted hypervisors don't have their place – they're much simpler to set up, so they're a good bet if, say, you need to deploy a test environment quickly. One of the best ways to determine which hypervisor meets your needs is to compare their performance metrics. These include CPU overhead, the amount of maximum host and guest memory, and support for virtual processors. The following factors should be examined before choosing a suitable hypervisor:

1. Understand your needs: The company and its applications are the reason for the data center (and your job). Besides your company's needs, you (and your co-workers in IT) also have your own needs. Needs for a virtualization hypervisor are:

a. Flexibility
b. Scalability
c. Usability
d. Availability
e. Reliability
f. Efficiency
g. Reliable support

2. The cost of a hypervisor: For many buyers, the toughest part of choosing a hypervisor is striking the right balance between cost and functionality. While a number of entry-level solutions are free, or practically free, the prices at the opposite end of the market can be staggering. Licensing frameworks also vary, so it's important to be aware of exactly what you're getting for your money.

3. Virtual machine performance: Virtual systems should meet or exceed the performance of their physical counterparts, at least in relation to the applications within each server. Everything beyond meeting this benchmark is profit.

4. Ecosystem: It's tempting to overlook the role of a hypervisor's ecosystem – that is, the availability of documentation, support, training, third-party developers and consultancies, and so on – in determining whether or not a solution is cost-effective in the long term.

5. Test for yourself: You can gain basic experience from your existing desktop or laptop. You can run both VMware vSphere and Microsoft Hyper-V in either VMware Workstation or VMware Fusion to create a nice virtual learning and testing environment.

HYPERVISOR REFERENCE MODEL :
There are 3 main modules coordinates in order to emulate the underlying hardware:

1. **DISPATCHER:**
   The dispatcher behaves like the entry point of the monitor and reroutes the instructions of the virtual machine instance to one of the other two modules.

2. **ALLOCATOR:**
   The allocator is responsible for deciding the system resources to be provided to the virtual machine instance. It means whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with the virtual machine, the allocator is invoked by the dispatcher.

3. **INTERPRETER:**
   The interpreter module consists of interpreter routines. These are executed, whenever a virtual machine executes a privileged instruction.

COMPARISON BETWEEN KVM AND VIRTUAL BOX:

KVM (Kernel-based Virtual Machine):

- **Type:** KVM is a type 1 hypervisor, also known as a "bare metal" hypervisor.
- **Linux Integration:** KVM is tightly integrated with the Linux kernel.
- **Performance and Efficiency:**
    - KVM runs directly on the host machine's physical hardware, bypassing the need for an underlying operating system. This direct access to hardware resources results in excellent performance and efficiency.
    - It doesn't contend with other software layers (such as additional operating systems or device drivers) for virtualization.
- **Maturity and Support:**
    - KVM has been around for over 15 years and boasts a large community of 1,000+ code contributors.
    - Its maturity ensures robust development, extensive debugging, and ample support from experts.
- **Scalability:**
    - As the number of virtual machines (VMs) increases, KVM automatically scales to handle heavy workloads.
    - It supports clustering for thousands of nodes, laying the groundwork for cloud-based infrastructures.
- **Security:**

    - KVM benefits from the rigorous development and testing processes of the Linux kernel.
    - Continuous security patching ensures a secure environment.
- **Cost:** KVM is open source and available as a Linux kernel module, making it free out of the box.

VirtualBox:

- **Type:** VirtualBox is a type 2 hypervisor, also known as a “hosted” hypervisor.
- **Operating System Abstraction:**
    - VirtualBox runs on a conventional OS (like other computer programs) and abstracts guest operating systems from the host OS.
    - It relies on the host machine’s pre-existing OS to manage CPU, memory, storage, and network resources.
- **Guest OS Support:**
    - VirtualBox supports a wide range of guest operating systems, including both Linux and non-Linux systems.
- **Ease of Use:**
    - VirtualBox provides an easy-to-use GUI for everyday VM operations.
    - It’s suitable for less technical users who want to get started quickly.
- **Scalability:**
    - While VirtualBox is more scalable than KVM, it’s not as performant in large environments.
    - It excels for small-scale use cases.
- **Choosing Between KVM and VirtualBox:**
    - KVM is ideal for:
        - Large environments requiring maximum performance and scalability.
        - Linux-centric setups.
        - Those comfortable with command-line tools.
    - VirtualBox is great for:
        - Small-scale deployments.

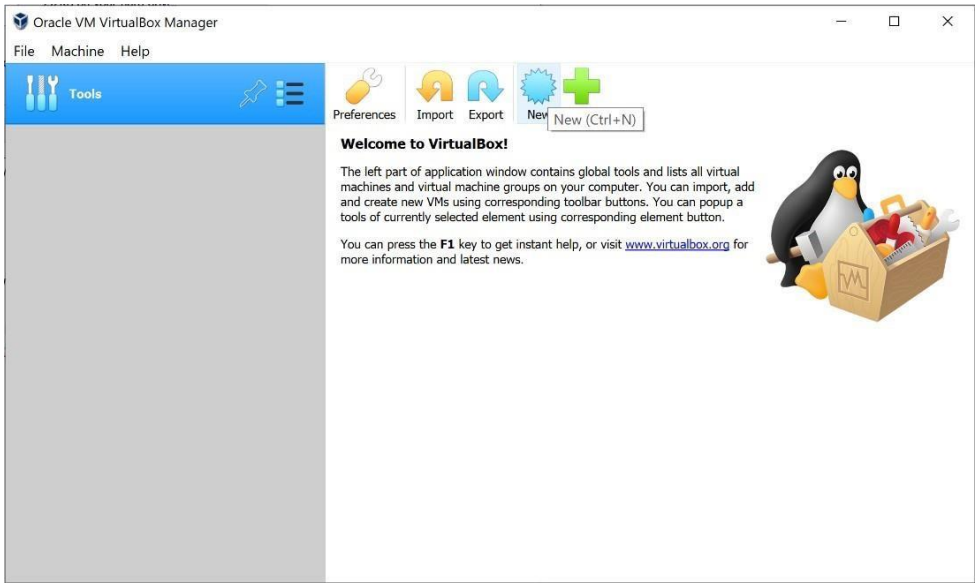
- User-friendly interfaces.
- Broad platform support.

In summary, if you're installing a binary Linux distribution as a guest, KVM is the better choice due to its speed and Linux integration. However, for broader compatibility and ease of use, VirtualBox is a solid option. Choose based on your specific needs!

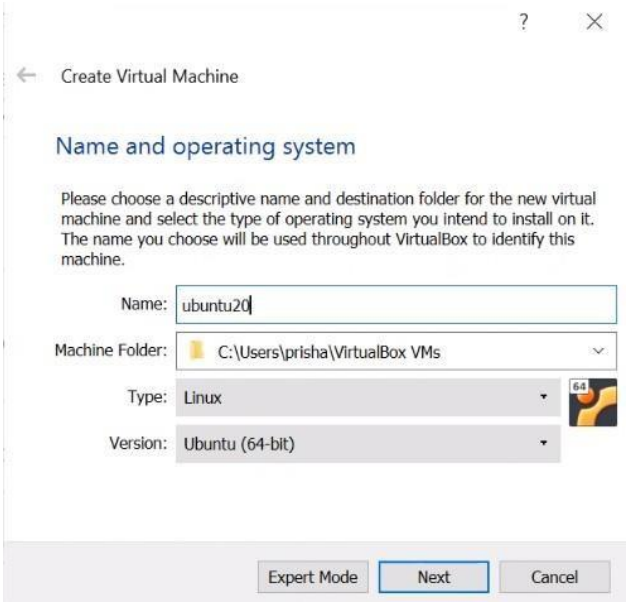
Steps :

1. Download virtual box.

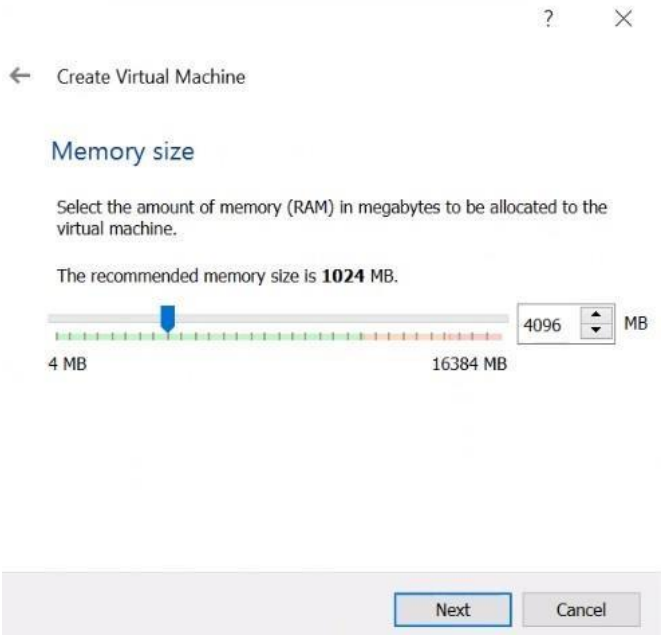
2. Create a new virtual machine.

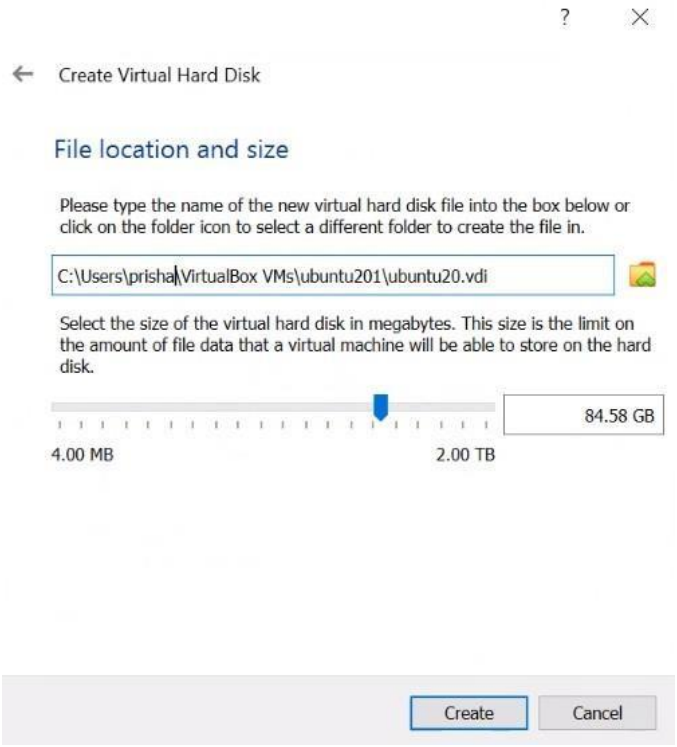


3. Choose Linux Ubuntu OS as KVM runs on Linux

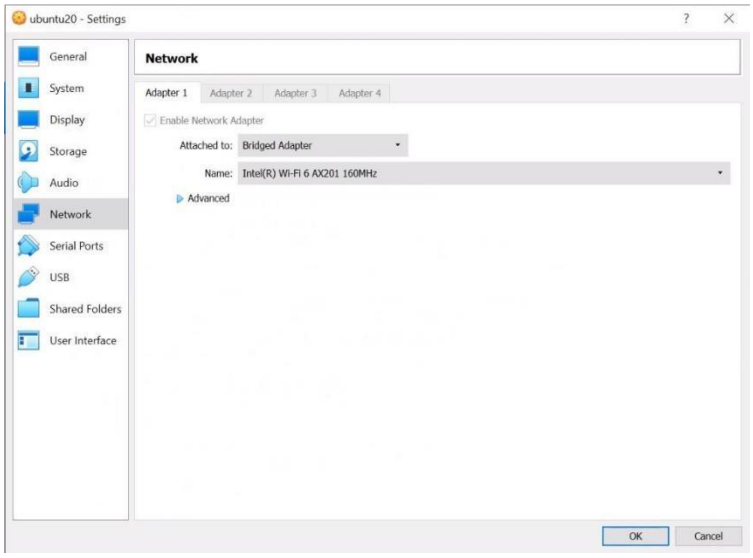


4. Select around 4GB Ram and create virtual hard disk of around 80 GB

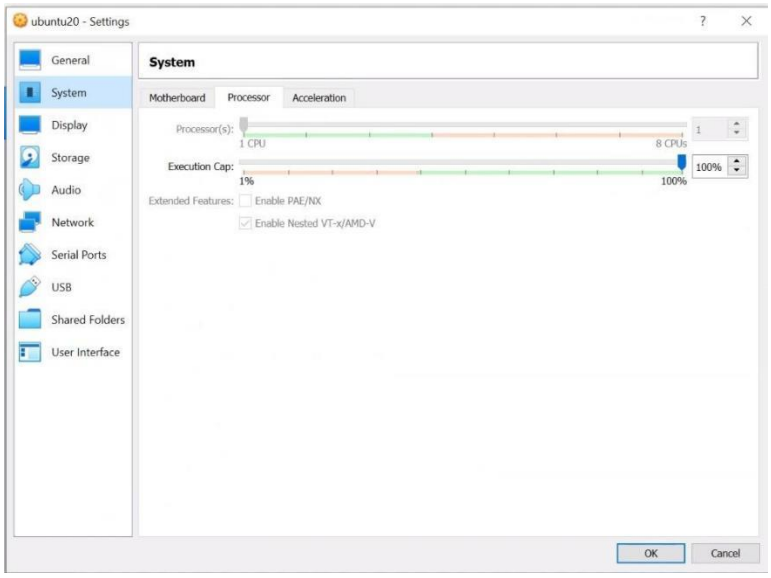

Create Virtual Machine
Memory size
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.
The recommended memory size is 1024 MB.
4096
MB
4 MB
16384 MB
Next
Cancel


Create Virtual Hard Disk
File location and size
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.
C:\Users\prisha\VirtualBox VMs\ubuntu201\ubuntu20.vdi
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.
84.58 GB
4.00 MB
2.00 TB
Create
Cancel

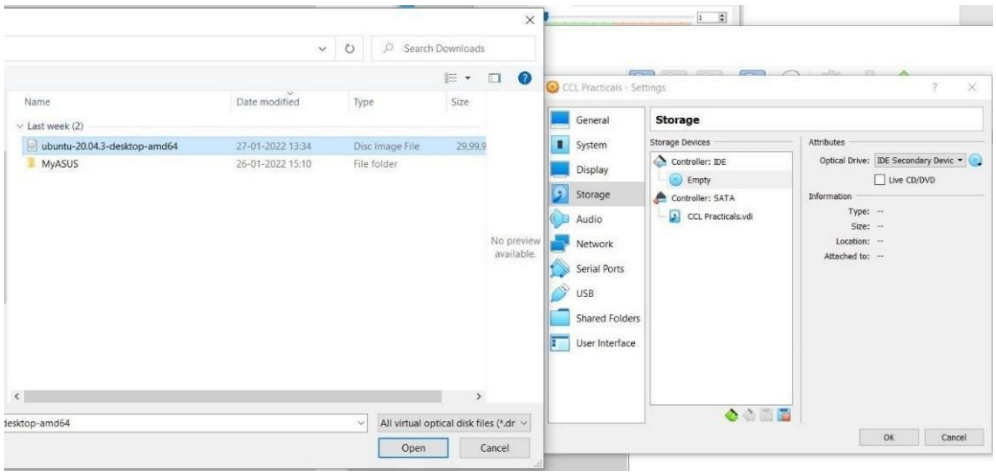
5. Click on Settings and go to Network and switch to Bridged Adapter



6. Click on settings and go to System, Processor, and Enable Nested VT-x/AMD-V



7. Download Ubuntu ISO from the official site and then click on [Optical Drive], followed by choose a disk file and select the ubuntu ISO.


Search Downloads
Name
Date modified
Type
Size
Last week (2)
ubuntu-20.04.3-desktop-amd64
27-01-2022 13:34
Disc Image File
MyASUS
26-01-2022 15:10
File folder
No preview available.
All virtual optical disk files (*.dr
Open
Cancel
CCL Practicals - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
Storage
Storage Devices
Controller: IDE
Empty
Controller: SATA
CCL Practicals.vdi
Attributes
Optical Drive:
IDE Secondary Devic
Live CD/DVD
Information
Type:
Size:
Location:
Attached to:
OK
Cancel

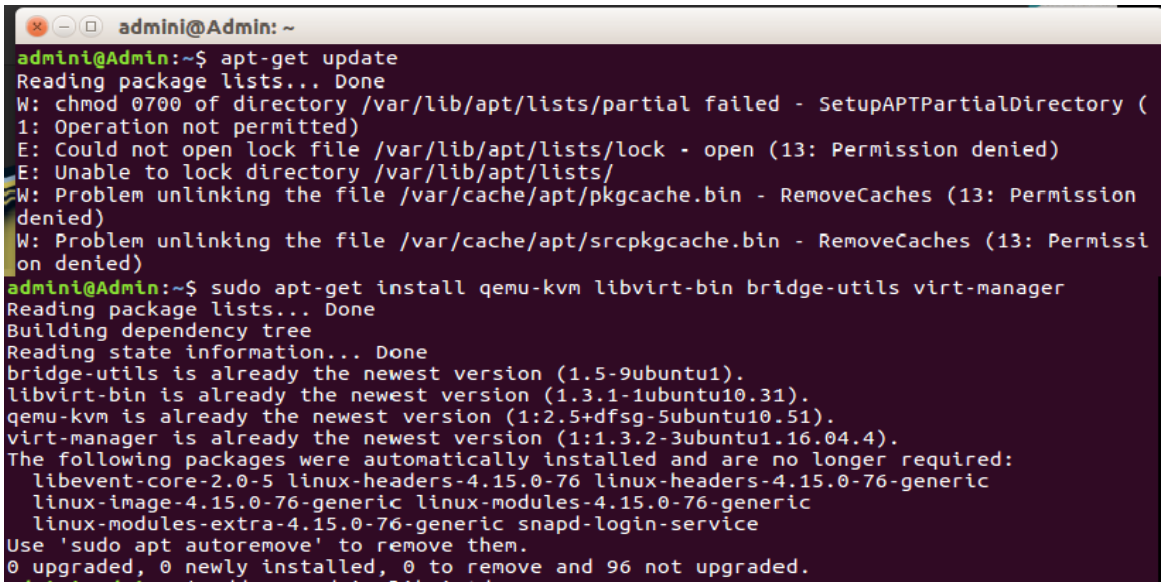
8. Click on Start and it will run Ubuntu OS

9. Check whether CPU has hardware virtualization support. KVM only works if your CPUhas hardware virtualization support –either Intel VT-x or AMD-V. To determine whether your CPU includes these features, run the command as shown below. To ensure that the VM can use all cores, we should get a number above 0. Here we get 2.

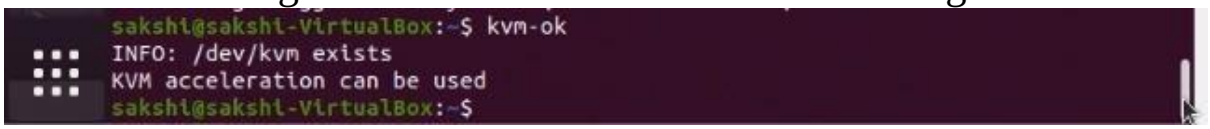
```
admini@Admin:~$ sudo grep -c "svm\|vmx" /proc/cpuinfo
[sudo] password for admini:
6
```

10. Run following commands to install KVM and supporting packages:

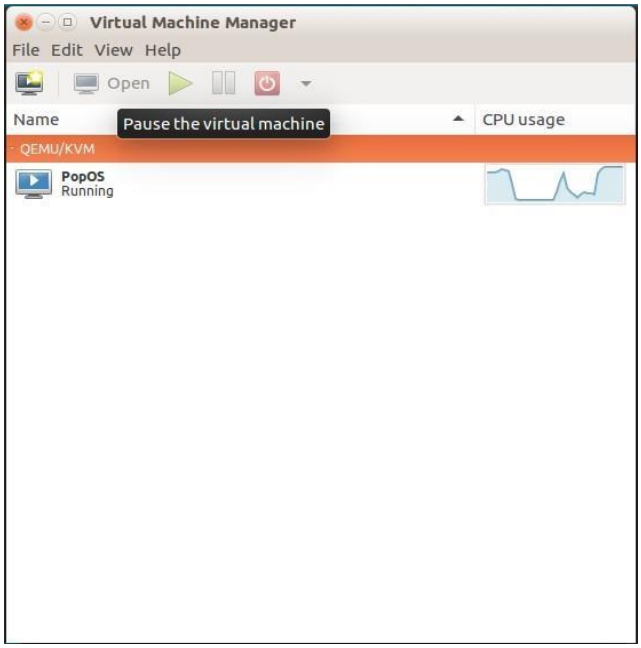
a. sudo apt update
b. apt install cpu-checker
c. sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clientsbridge-utilsvirtinst virt-manager


admini@Admin: ~
admini@Admin:~$ apt-get update
Reading package lists... Done
W: chmod 0700 of directory /var/lib/apt/lists/partial failed - SetupAPTPartialDirectory (
1: Operation not permitted)
E: Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin - RemoveCaches (13: Permission
denied)
W: Problem unlinking the file /var/cache/apt/srcpkgcache.bin - RemoveCaches (13: Permissi
on denied)
admini@Admin:~$ sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager
Reading package lists... Done
Building dependency tree
Reading state information... Done
bridge-utils is already the newest version (1.5-9ubuntu1).
libvirt-bin is already the newest version (1.3.1-1ubuntu10.31).
qemu-kvm is already the newest version (1:2.5+dfsg-5ubuntu10.51).
virt-manager is already the newest version (1:1.3.2-3ubuntu1.16.04.4).
The following packages were automatically installed and are no longer required:
libevent-core-2.0-5 linux-headers-4.15.0-76 linux-headers-4.15.0-76-generic
linux-image-4.15.0-76-generic linux-modules-4.15.0-76-generic
linux-modules-extra-4.15.0-76-generic snapd-login-service
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 96 not upgraded.

11. Run following command to check if KVM is working:

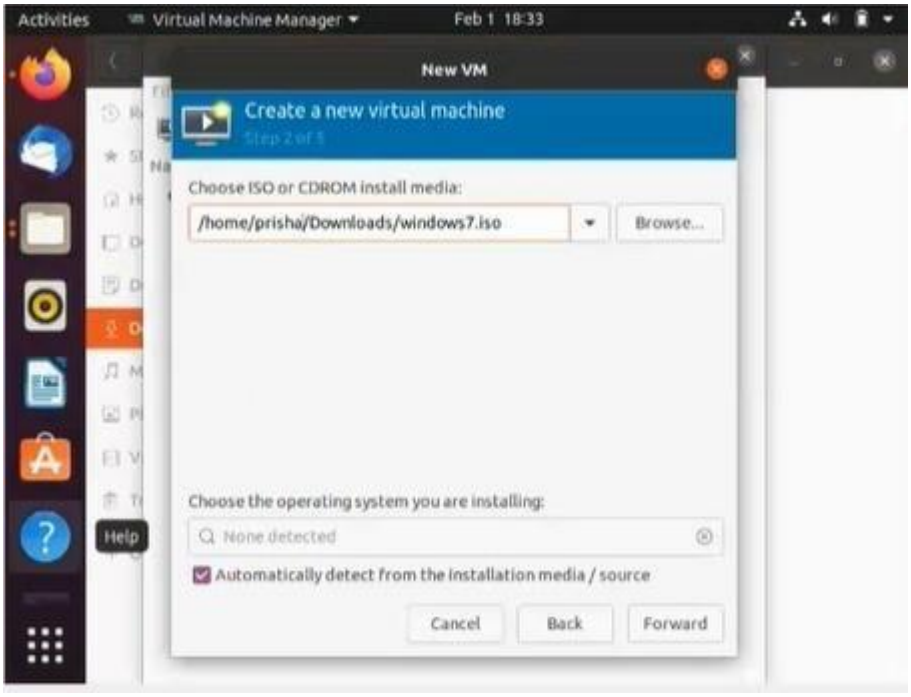


12. Open Virtual Machine Manager application and Create Virtual Machine

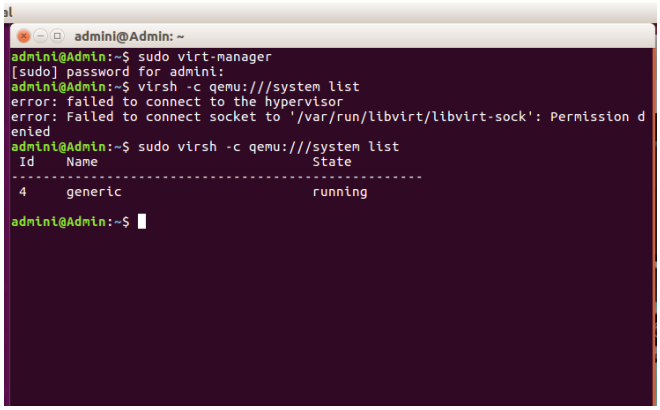


13. Download another OS ISO (such as POP! OS) and set up a VM, like we did with Virtual Box and install the OS, for a virtual Hard Disk do not exceed the amount of free space available. Click on File, New Virtual Machine to create the VM

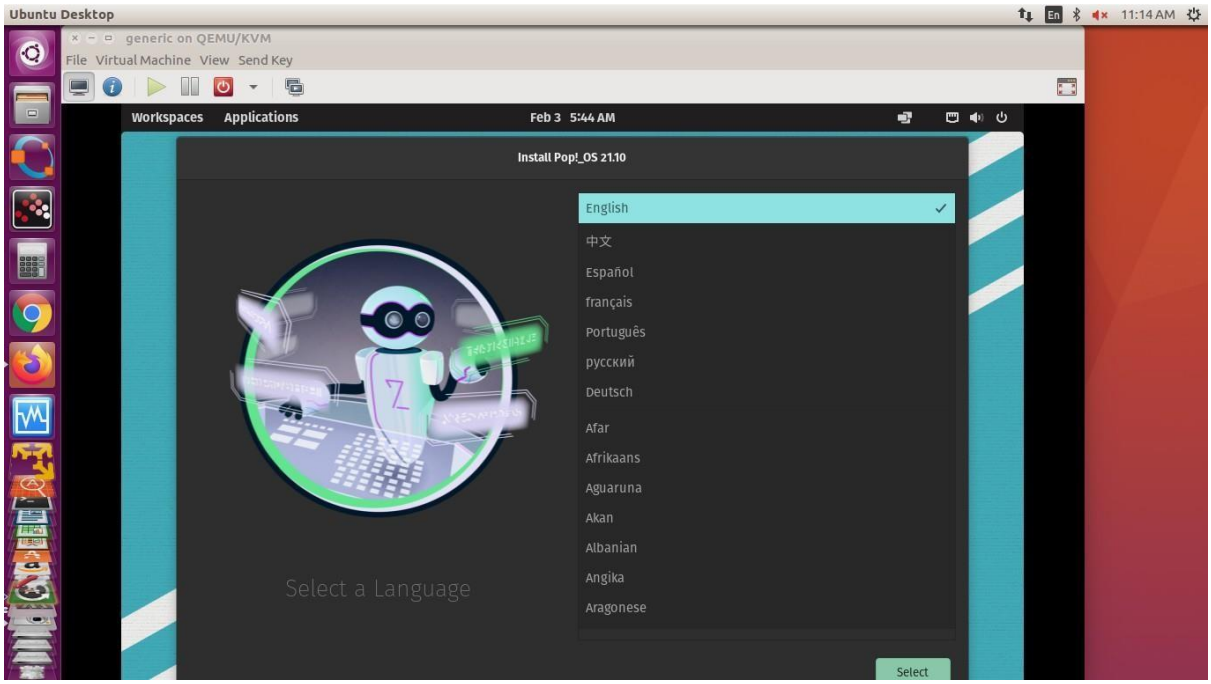
14. Locate the ISO file and select Generic default OS



15. After installation is complete we can see the VM running using followingcommand: sudo virsh -c qemu:///system list



16. Open VM and complete installation steps of POP! OS.



Conclusion: Hence understood the concept of Virtualization and successfully implemented HostedVirtualization using VirtualBox and KVM.