Ishita Hardasmalani

C14-2103058

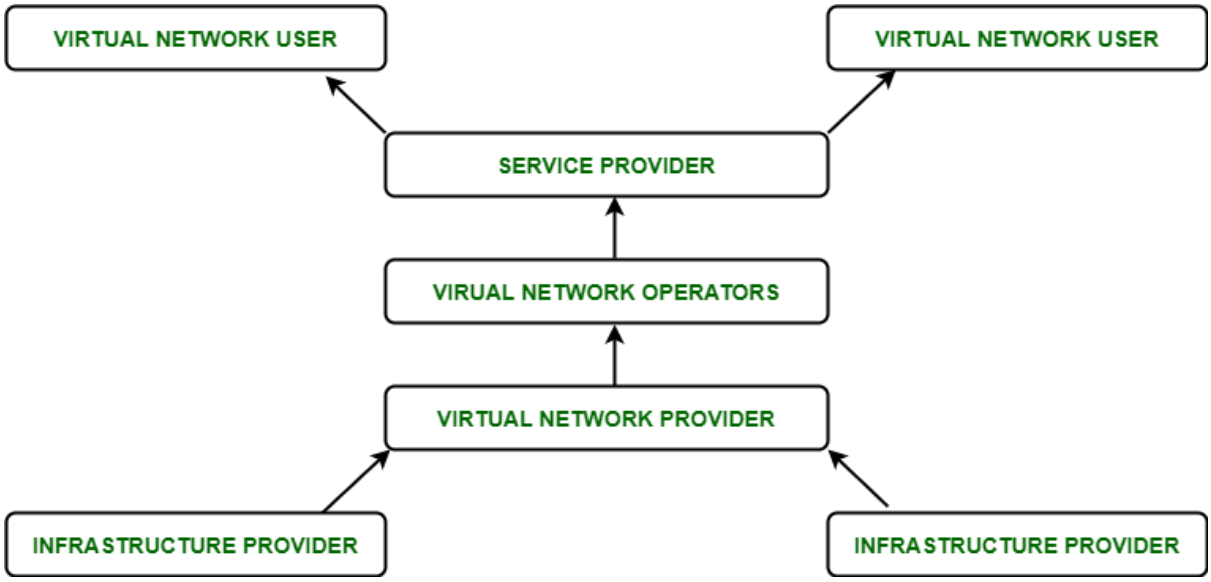
EXPERIMENT NO.2

Aim: To study and implement hosted virtualization using VirtualBox and KVM.

Theory:

Virtualization in cloud computing

Virtualization is a foundational technology in cloud computing that plays a crucial role in optimizing resource utilization, enhancing flexibility, and enabling efficient management of computing resources. At its core, virtualization involves creating a virtual (rather than physical) version of various computing components, such as servers, storage devices, or networks. These virtual instances can then be used to run multiple operating systems or applications simultaneously on a single physical machine.



Benefits of Virtualization

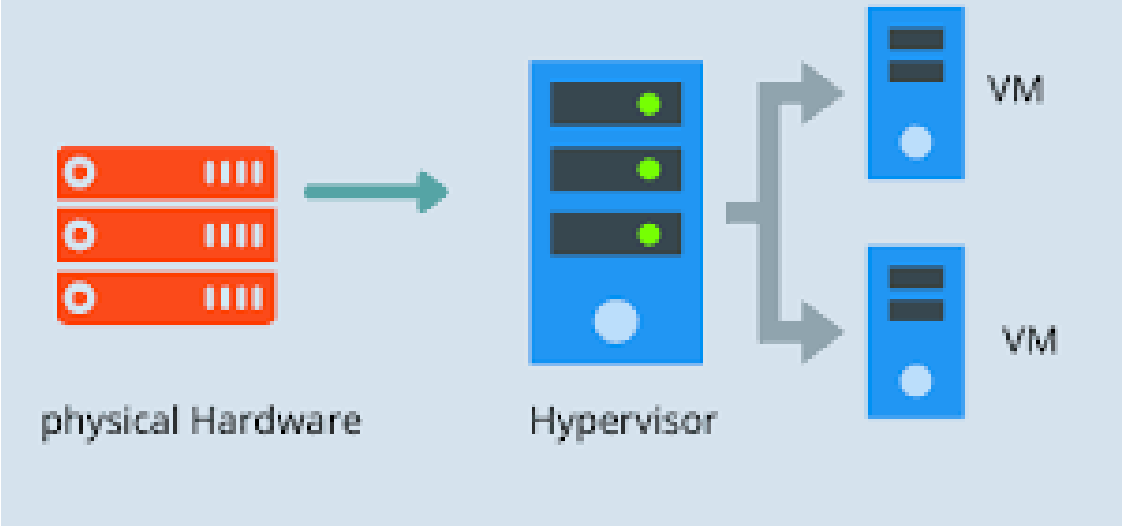
1. Resource Optimization and Cost Savings: Virtualization allows multiple virtual machines (VMs) to run on a single physical server, optimizing resource utilization. This consolidation reduces the need for additional hardware, leading to cost savings in terms of hardware acquisition, maintenance, and energy consumption.
2. Flexibility and Scalability: Virtualization provides flexibility by decoupling the operating system and applications from the underlying hardware. This allows for easy scaling up or down based on demand. New virtual instances can be quickly provisioned or de-provisioned to adapt to changing workloads.
3. Isolation and Security: Each virtual machine operates independently of others on the same physical server. This isolation enhances security by preventing one VM from

affecting others. If a security issue occurs in one VM, it does not necessarily impact the others.

4. Hardware Independence: Virtualization abstracts the underlying hardware, allowing virtual machines to run on different physical servers without being tied to specific hardware configurations. This independence makes it easier to migrate and manage virtualized workloads.
5. Improved Disaster Recovery and Business Continuity: Virtualization simplifies and enhances disaster recovery planning. Virtual machines can be easily replicated and moved between physical servers or even across data centers. This flexibility allows for quicker recovery times in the event of hardware failures, disasters, or system crashes. It ensures business continuity by minimizing downtime and data loss.

These benefits collectively contribute to the overall efficiency, agility, and cost-effectiveness of IT infrastructures in various industries, making virtualization a foundational technology in modern computing environments.

Hypervisor



A hypervisor, also known as a Virtual Machine Monitor (VMM), is a software or firmware layer that enables the creation and management of virtual machines (VMs) on a physical host machine. It abstracts the underlying hardware, allowing multiple operating systems to run independently on the same physical server. The hypervisor allocates and manages resources, such as CPU, memory, and storage, for each virtual machine. Hypervisors play a fundamental role in virtualization technology, allowing organizations to create and manage multiple virtualized environments on a single physical server for improved resource utilization and flexibility.

Examples of Hypervisors:

1. VMware vSphere/ESXi:

- Type: Type 1 Hypervisor

- Description: VMware ESXi is a bare-metal hypervisor used in enterprise environments. It provides a robust virtualization platform with features for resource management, scalability, and high availability.

2. Microsoft Hyper-V:

- Type: Type 1 (Standalone) and Type 2 (Hosted)

- Description: Hyper-V is a hypervisor developed by Microsoft. It can be used as a standalone hypervisor (Type 1) or installed as a feature on Windows operating systems (Type 2).

3. Xen:

- Type: Type 1 Hypervisor

- Description: Xen is an open-source hypervisor known for its performance and scalability. It is commonly used in server virtualization environments and supports paravirtualization.

4. Oracle VirtualBox:

- Type: Type 2 Hypervisor

- Description: Oracle VirtualBox is a free and open-source hosted hypervisor that runs on various host operating systems. It is widely used for desktop virtualization.

5. KVM (Kernel-based Virtual Machine):

- Type: Type 1 Hypervisor

- Description: KVM is a Linux kernel module that turns the host operating system into a hypervisor. It is integrated with the Linux kernel and supports hardware virtualization.

Types of Hypervisors

Hypervisors, also known as Virtual Machine Monitors (VMMs), come in two main types: Type 1 (Bare Metal) and Type 2 (Hosted). These types differ in their architecture and deployment.

The choice between Type 1 and Type 2 hypervisors depends on the specific requirements of the use case, the level of control needed over hardware resources, and performance considerations.

Type 1 Hypervisor (Bare Metal Hypervisor):

Characteristics:

- Installation: Installs directly on the physical hardware of the host machine.

- Resource Management: Has direct control over hardware resources.

- Performance: Typically considered more efficient and provides better performance compared to Type 2 hypervisors.

- Use Cases:Commonly used in enterprise environments, data centers, and cloud infrastructure.

Examples:

1. VMware ESXi: A leading enterprise-grade hypervisor with advanced features for resource management, scalability, and high availability.

2. Microsoft Hyper-V (Standalone): Microsoft's standalone hypervisor, suitable for virtualization in Windows-centric environments.

3. Xen: Open-source hypervisor known for its performance and scalability. Supports paravirtualization for optimized performance.

4. KVM (Kernel-based Virtual Machine): A Linux kernel module that transforms the Linux OS into a hypervisor. KVM supports hardware virtualization and is integrated into the Linux kernel.

Type 2 Hypervisor (Hosted Hypervisor):

Characteristics:

- Installation: Runs on top of a host operating system.

- Resource Management: Relies on the host OS for resource management.

- Performance: Typically has more overhead compared to Type 1 hypervisors.

- Use Cases: Commonly used for development, testing, and desktop virtualization.

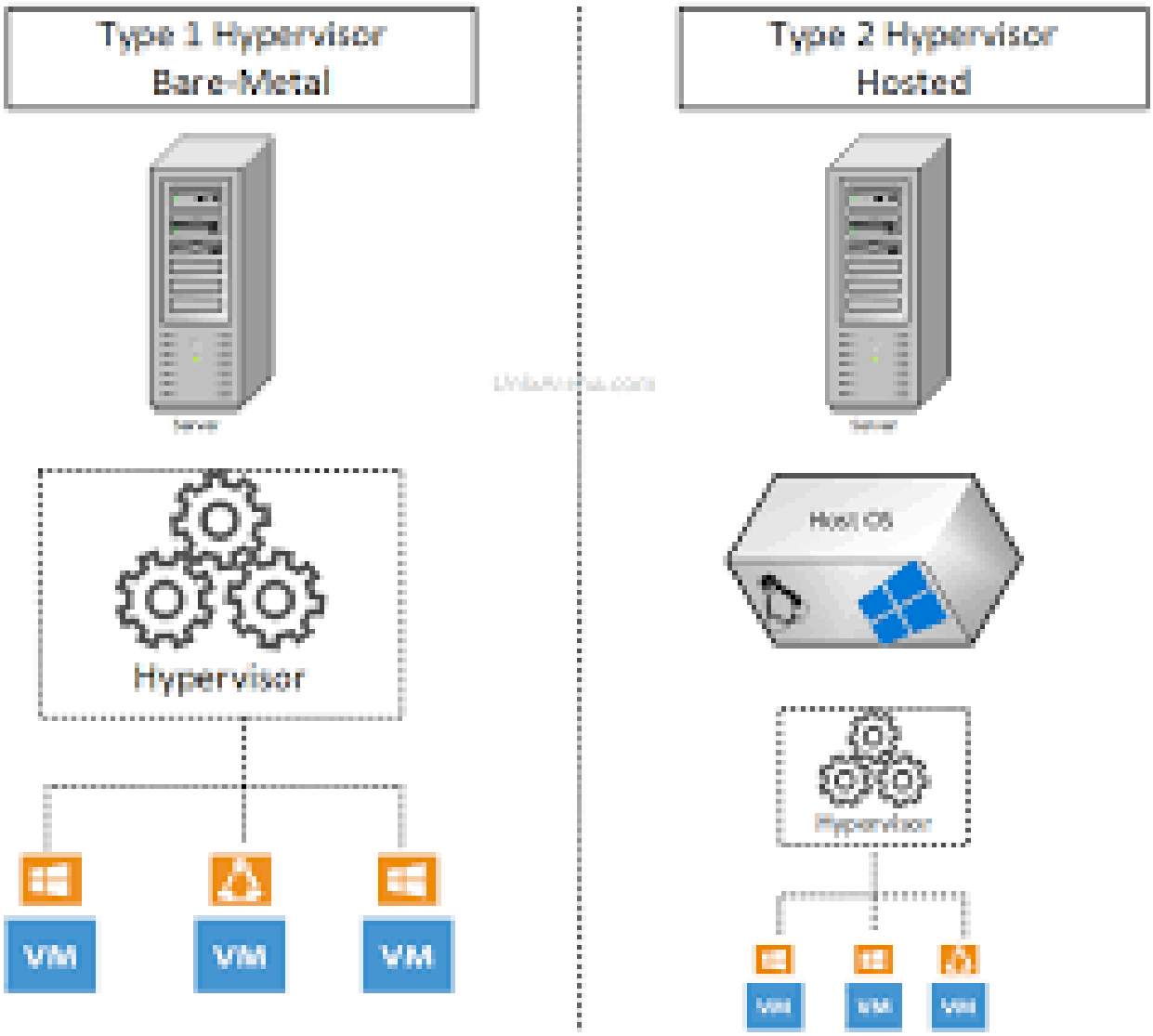
Examples:

1. VMware Workstation: Allows users to run multiple operating systems on a single host machine. Suited for development, testing, and training.

2. Oracle VirtualBox: A free, open-source hypervisor that supports various host operating systems. Widely used for desktop virtualization.

3. Microsoft Hyper-V (Hosted): Hyper-V can be installed as a feature on Windows operating systems, serving as a Type 2 hypervisor.

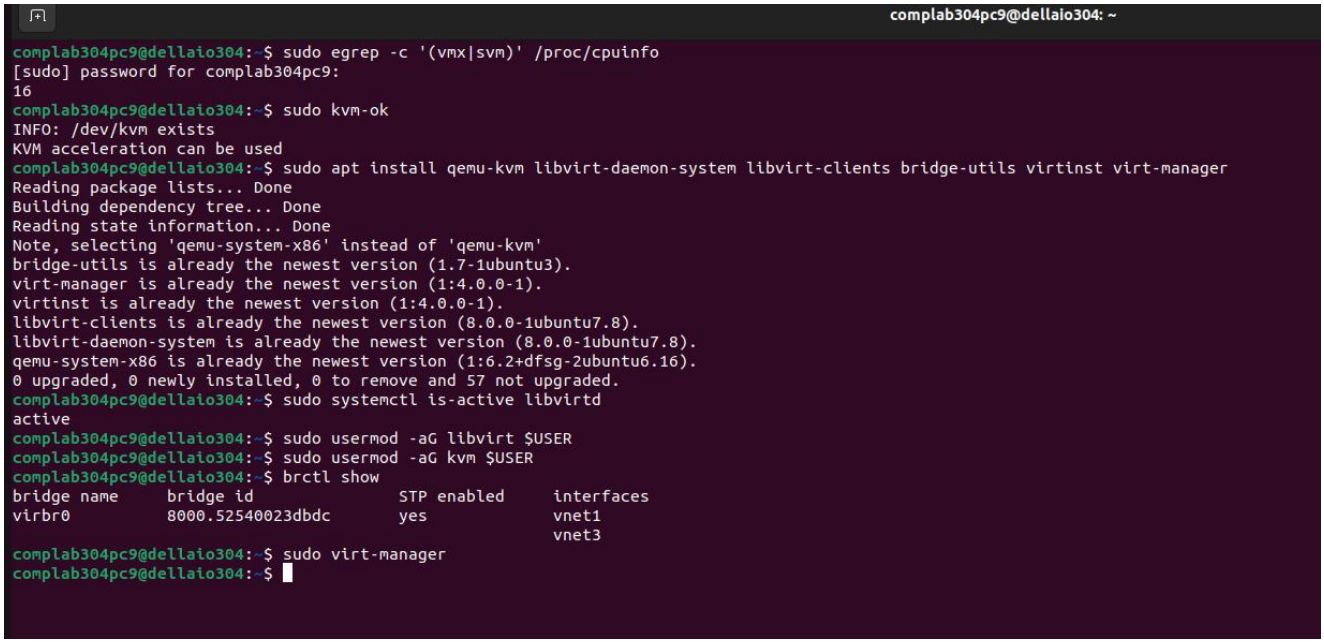
4. Parallels Desktop: Designed for macOS, Parallels Desktop allows users to run Windows and other operating systems on a Mac.

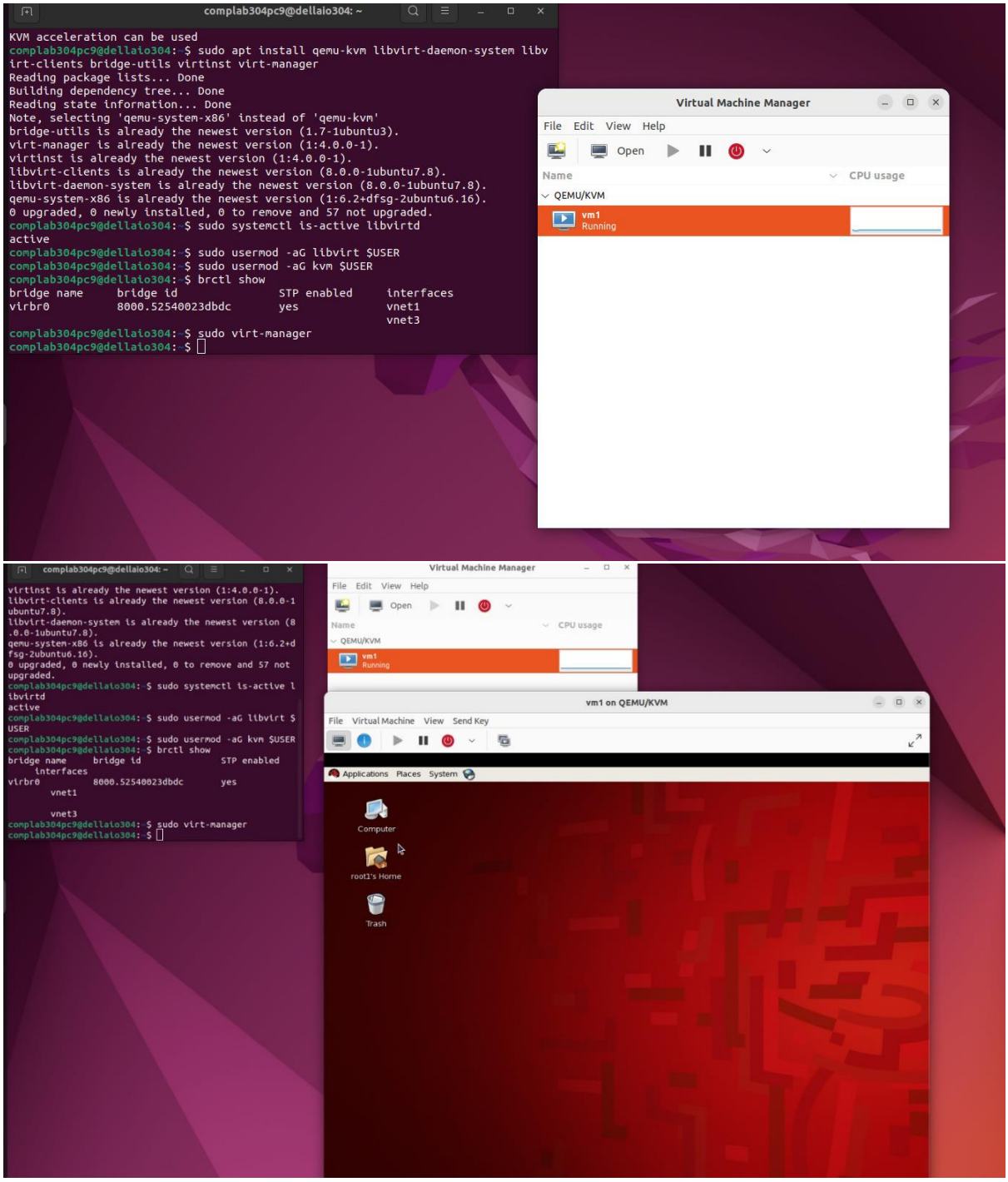

Type 1 Hypervisor
Bare-Metal
Hypervisor
VM
VM
VM
Type 2 Hypervisor
Hosted
Host OS
Hypervisor
VM
VM
VM

VirtualBox VS KVM

Feature	Oracle VirtualBox	KVM (Kernel-based Virtual Machine)
Type of Hypervisor	Type 2 (Hosted Hypervisor)	Type 1 (Bare Metal Hypervisor)
Installation	Installs on top of a host operating system	Part of the Linux kernel, integrated into the host OS
Platform Support	Windows, macOS, Linux	Primarily Linux-based, supports various Linux distributions
Performance	Generally has more overhead compared to Type 1 hypervisors	Typically offers better performance due to being a Type 1 hypervisor
Resource Utilization	May have higher resource consumption due to running on top of an OS	Efficient resource utilization as it has direct access to hardware
Ease of Use	User-friendly with a graphical interface	Often managed through command-line tools; additional management tools available
Guest OS Support	Supports a wide range of guest operating systems, including Windows, Linux, macOS	Primarily designed for Linux guests, supports Windows and other OS with varying degrees of optimization
Snapshot Management	Provides snapshot functionality for creating and managing snapshots of VM states	Offers snapshot functionality for creating and restoring VM states
Networking Options	Provides a range of networking options and configurations through a GUI	Networking configurations often handled through command-line tools, may require additional setup
USB Support	Supports USB passthrough for connecting USB devices to VMs	Offers USB passthrough, but setup may be more manual compared to VirtualBox
Community/Support	Large community support and documentation	Community support, especially strong within the Linux community
Use Cases	Ideal for desktop virtualization, development, and testing	Commonly used in server environments, data centers, and cloud infrastructure

STEPS FOR IMPLEMENTING:




complab304pc9@dellaio304: ~
KVM acceleration can be used
complab304pc9@dellaio304:~$ sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virtinst virt-manager
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'qemu-system-x86' instead of 'qemu-kvm'
bridge-utils is already the newest version (1.7-1ubuntu3).
virt-manager is already the newest version (1:4.0.0-1).
virtinst is already the newest version (1:4.0.0-1).
libvirt-clients is already the newest version (8.0.0-1ubuntu7.8).
libvirt-daemon-system is already the newest version (8.0.0-1ubuntu7.8).
qemu-system-x86 is already the newest version (1:6.2+dfsg-2ubuntu6.16).
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
complab304pc9@dellaio304:~$ sudo systemctl is-active libvirtd
active
complab304pc9@dellaio304:~$ sudo usermod -aG libvirt $USER
complab304pc9@dellaio304:~$ sudo usermod -aG kvm $USER
complab304pc9@dellaio304:~$ brctl show
bridge name bridge id STP enabled interfaces
virbr0 8000.52540023dbdc yes vnet1
vnet3
complab304pc9@dellaio304:~$ sudo virt-manager
complab304pc9@dellaio304:~$
Virtual Machine Manager
File Edit View Help
Open
Name
CPU usage
QEMU/KVM
vm1
Running
vm1 on QEMU/KVM
File Virtual Machine View Send Key
Applications Places System
Computer
root1's Home
Trash