UNIT 1
OVERVIEW OF VIRTUALIZATION

Q.1	**What is Virtualization? Explain the five stages Virtualization Process.**
A.	• "Half the work that is done in the world is to make things appear what they are not." (E. R. Beadle). • Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources* • It is the process by which one computer hosts the appearance of many computers. • Virtualization is used to improve IT throughput and costs by using physical resources as a pool from which virtual resources can be allocated. • Virtualization is a technology that transfers hardware into software. • Virtualization allows us to run multiple Operating Systems as VMs on single computer. • "Virtualization software makes it possible to run multiple operating systems and multiple applications on the same server at the same time". • "It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware." • The technology behind virtualization is known as a virtual machine monitor (VMM) or virtual manager, which separates compute environments from the actual physical infrastructure. **Five Stages of Virtualization Process:** **1**. **Discovery:** The first step begins with datacentre inventories and the identification of potential virtualization candidates. 2. **Virtualization:** The second step focuses on gaining a complete understanding of the value choices that virtualization can offer. 3. **Hardware maximization:** The third step focuses on hardware recovery and how you can make judicial investments when adding new hardware or replacing older systems. 4. **Architecture:** The fourth step looks to the architecture you must prepare to properly introduce virtualization technologies into your datacentre practices. 5. **Management:** The last step focuses on the update of the management tools you use to maintain complete virtualization scenarios in your new dynamic datacentre.

Q.2 **What are different types of Virtualization? Explain desktop Virtualization.**

A. **Different types of Virtualization:**

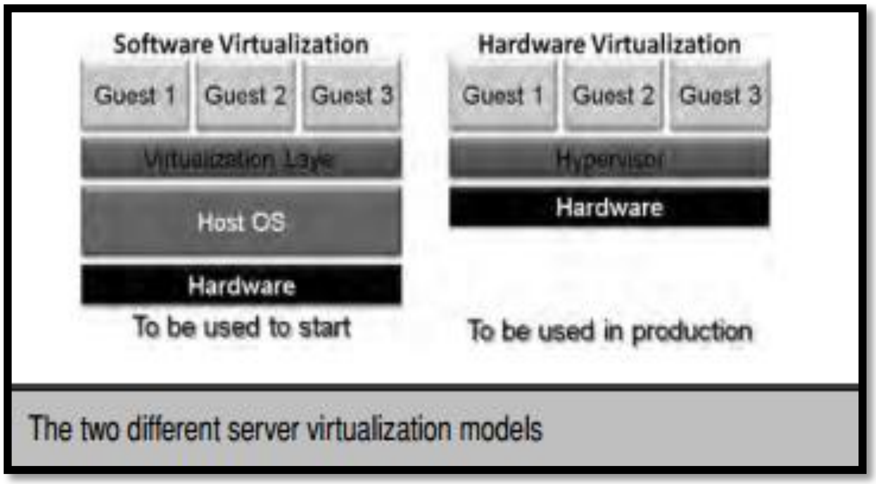
- **AppV (Application Virtualization)**
- **PresentV (Presentation Virtualization)**
- **DeskV (Desktop Virtualization)**
- **ManageV (Management Virtualization)**
- **NetV (Network Virtualization)**
- **StoreV (Storage Virtualization)**
- **SerV (Server Virtualization)**

Desktop Virtualization

- Desktop Virtualization (DeskV) allows you to rely on virtual machines to provision desktop systems. Desktop virtualization has several advantages, the least of which is the **ability to centralize desktop deployments and reduce distributed management costs** because user's access centralized desktops through a variety of thin or unmanaged devices.
- Desktop virtualization **centralizes desktop deployments so that you can gain complete control over them,** letting users rely on a variety of endpoints—thin computing devices, unmanaged PCs, home PCs, or public PCs—to access your corporate desktop infrastructure, once again through the **Remote Desktop Connection (RDC)**.
- The main difference between DeskV and PresentV, or presentation virtualization, often called **Terminal Services or server-based computing**, is that in PresentV, users must share the desktop environment with all of the other users connecting to the server. In DeskV, **each user gets access to their own desktop, limiting the potential impact of the applications they need on other desktop sessions.**
- DeskV can be quite a **time-saver** compared to the cost of managing distributed systems throughout your infrastructure. If you have existing desktops, you can turn them into unmanaged devices because all you need from the physical workstation are three things:
  1. A base operating system, which can be anything from Windows XP to Vista

	or beyond. This **base operating system must be patched and updated**, which is something you must do anyway. 2. **Proper virus protection**, another item you must manage anyway if you have physical desktops. 3. A Remote Desktop Connection client • Each single desktop system you remove from your environment will reduce your **power consumption by 650,000 kilowatt-hours per year**. • There are several other reasons for moving to virtual desktops. Here are a few: 1. You can create enterprise-standard lock-down PCs on unmanaged remote PCs. By centralizing the desktop through server virtualization, you can lock down the corporate PC image while letting users run free on the unmanaged physical desktops 2. You can create time-controlled PC images. 3. You can also secure information by keeping desktops in the datacentre. 4. You can encapsulate complex or sensitive applications, isolating them from any others. 5. DeskV can provide a new migration path to new operating systems. • DeskV is also a great model for testing and development since like any server virtualization machine, they support undoable disks and can easily be spawned when needed.
Q.3	**What are different types of Virtualization? Explain network Virtualization.**
A.	**Different types of Virtualization same as A.2** **Network Virtualization** • "The term network virtualization refers to the creation of logical isolated network partitions overlaid on top of a common physical infrastructure. Each partition is logically isolated from the others, and must behave and appear as a fully dedicated network to provide privacy, security, and an independent set of policies, service levels, and even routing decisions." • A network virtualization technology can aggregate, create, or segment one (or more) of the following planes: 1. **Data plane**: Handles the traffic that is traversing two or more interfaces of a network device (transit packets). Responsible for the majority of data influx on these devices, it is also known as **forwarding plane**. 2. **Control plane**: Processes traffic directed to the networking device itself and originated from other devices. It is exemplified with control packets from routing protocols and controls the behaviour of the data plane. 3. **Management plane:** Runs components meant for device management purposes, such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP). This plane usually interacts with third-party

	software and is able to modify the behaviour of both control and data planes. • Network virtualization introduces the possibility of **transforming physical connections and devices into simpler logical entities**, both improving resource utilization and reducing design complexities. These techniques include EtherChannel, Virtual PortChannel (vPC) ,Layer 2 multipathing with FabricPath. • Network virtualization techniques can help **network partitioning, resource optimization, management consolidation, and network extension** • Network Virtualization (NetV) lets you **control available bandwidth** by splitting it into independent channels that can be assigned to specific resources. For example, the simplest form of network virtualization is the virtual local area network (VLAN), which creates a logical segregation of a physical network.
Q.4	**What are different types of Virtualization? Explain Server Virtualization.**
A.	• Server Virtualization (SerV) is focused on **partitioning a physical instance of an operating system into a virtual instance** or virtual machine. True server virtualization products will let you virtualize any x86 or x64 operating system, such as Windows, Linux, and some forms of UNIX. There are two aspects of server virtualization: **1. Software Virtualization (SoftV)** runs the virtualized operating system on top of a software virtualization platform running on an existing operating system. **2. Hardware Virtualization (HardV)** runs the virtualized operating system on top of a software platform running directly on top of the hardware without an existing operating system. The **engine used to run hardware virtualization is usually referred to as a hypervisor**. The purpose of this engine is to expose hardware resources to the virtualized operating systems. • The first, software virtualization or SoftV, is often **used to begin virtualization projects** because it relies on simpler and often free technologies, but is less efficient because it requires an underlying host operating system (OS). This underlying host operating system also requires resources and, because of this, will impact the operation of the virtual machines running on top of it. • For this reason, organizations will not use this model unless it is for testing or development. Because it runs on top of an existing operating system, it is often simpler to rely on SoftV to learn how server virtualization technologies work. •



The two different server virtualization models

- Running a virtualization product on top of an existing OS makes all of the virtual machines subject to the update process for the host OS. If reboots are required, all of the VMs will be rebooted as well. Not the best of scenarios.
- Server virtualization is still the most popular of the different virtualization technologies, and rightly so.
- There are several benefits to server virtualization:

1. The first one is certainly at the deployment level. A virtual machine can often be built and customized in less than 20 minutes. You can deliver a virtual machine that is ready to work right away in considerably less time than with a physical machine.

2. Virtual machine mobility. You can move a VM from one host to another at any time. In some cases, you can move it while it is running.

3. Virtual machine mobility. You can move a VM from one host to another at any time. In some cases, you can move it while it is running.

4. Virtual machines support standard configurations. You can control the way VMs are built.

5. Virtual machines also support the concept of volatile services. If a tester or developer needs a virtual machine to perform a given series of tests, you can fire up a new VM, provide it to them in minutes, and then, when they are done with it, you simply delete it.

6. VMs are also secure because they can be completely isolated at any time.

7. VMs are also ideal for disaster recovery, since all you need to do is copy their files to another location, either within your datacentre or to another site entirely.

Q.5 **What are different types of Virtualization? Explain Storage Virtualization.**

A.
- Storage virtualization can be performed at the

1. **Storage device:** Symbolizes the location of data at rest.
2. **Host:** Represents the computer system that is effectively retrieving and saving

the data.

3. **Interconnect:** Encompasses the network or medium between the host and the storage device

- Storage virtualization also allows **multiple physical arrays to work together as a single system**, bringing advantages such as data redundancy and management consolidation.
- Storage Virtualization (StoreV) is used to **merge physical storage from multiple devices** so that they appear as one single storage pool. The storage in this pool can take several forms: direct attached storage (DAS), network attached storage (NAS), or storage area networks (SANs); and it can be linked to through several protocols: Fibre Channel, Internet SCSI (iSCSI), Fibre Channel on Ethernet, or even the Network File System (NFS).
- Though storage virtualization is not a requirement for server virtualization, one of the key strengths you will be able to obtain from storage virtualization is the ability to rely on thin provisioning or the assignation of a logical unit (LUN) of storage of a given size, but provisioning it only on an as-needed basis.
- For example, if you create a LUN of 100 gigabytes (GB) and you are only using 12GB, only 12GB of actual storage is provisioned. This significantly reduces the cost of storage since you only pay as you go.

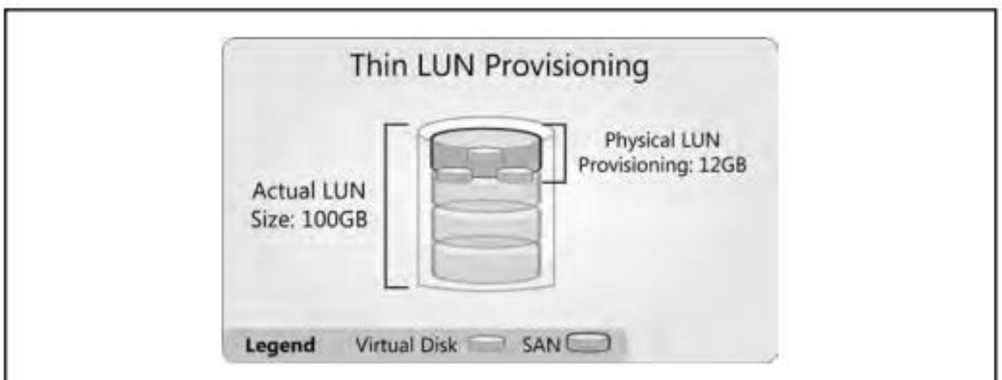


Figure 2-8. Using thin provisioning through Storage Virtualization can significantly reduce storage costs.

Q.6 **What are different types of Virtualization? Explain operating system Virtualization.**

A.

- Operating system virtualization often misconstrued as guest OS virtualization, this is nothing more than OS partitioning because it can only run one single OS type in parallel instances.
- The value of this type of "virtualization" is limited because you must have a need

	to run the particular OS. • Products of this type include Solaris Containers and Parallels Virtuozzo Containers, which runs the Virtuozzo OS—a version of Linux—in parallel.
Q.7	**What are different types of Virtualization? Explain application virtualization.**
A.	• The last key virtualization layer that most closely resembles server virtualization is application virtualization. Application virtualization, or AppV, creates software or service isolation on top of the OS through a special virtualization layer. • In this regard, AppV most closely resembles software server virtualization, or SoftV, because it requires an underlying operating system to function. The advantage AppV offers, however, is that it completely protects the OS from any changes applications can make during installation. That's because when you prepare an application for AppV, you don't capture the installation process as organizations did traditionally; instead, you capture the running state of the application or whatever is required to make the application functional on an OS. • Because of this, AppV-enabled applications can simply be Xcopied to endpoints since no installation is required. This provides a powerful model for distributed application management. AppV also provides support for application consolidation • Application Virtualization (AppV) uses the same principles as software based SerV, but instead of providing an engine to run an entire operating system, AppV decouples productivity applications from the operating system. • AppV transforms the distributed application management model because you only need to virtualize an application once. From then on, the application virtualization engine will make the virtualized application run on any version of Windows. • Work is also being done by major AppV vendors such as Microsoft, Citrix, InstallFree, Symantec, and VMware to apply AppV to server applications. While AppV only works on the 32-bit platform right now, work is also being done to make it work on 64-bit or x64 platforms

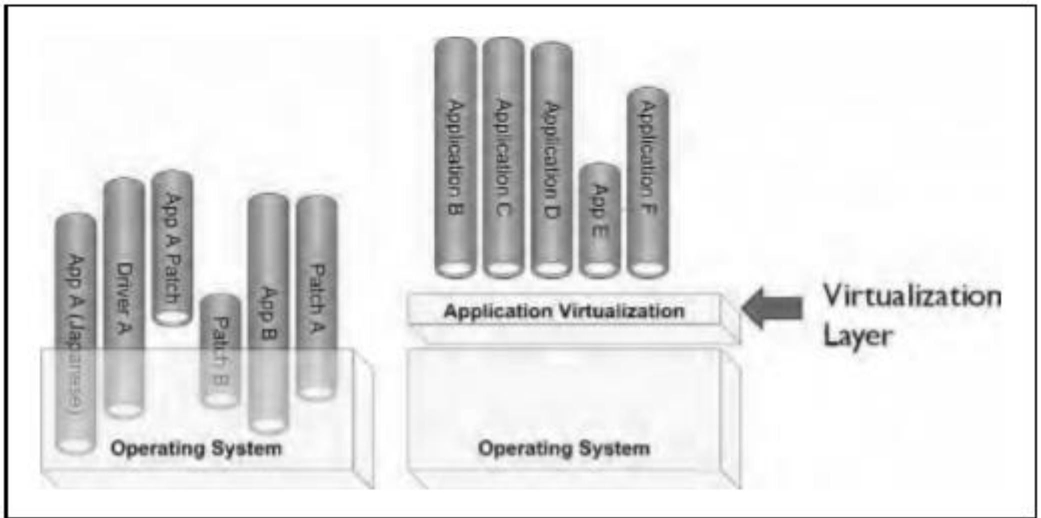


Figure 2-13. Relying on AppV to protect the target OS

- Basically, application virtualization creates a protective layer around the operating system to safeguard it against any changes that could be made by actually installing an application. The core concept of AppV is that instead of capturing a software installation, such as when packaging installations for the Windows Installer service, AppV captures the running state of an application.
- The application virtualization layer translates this running state into instructions the operating system and other applications can understand and interact with, keeping the core OS pristine at all times and containing application components in such a way that they can never cause conflicts with other applications.
- Another powerful advantage of AppV is that it can be both a stand-alone solution— working on its own on individual systems—or it can be integrated into both traditional (push installations) and streaming (pull installations) software delivery solutions.

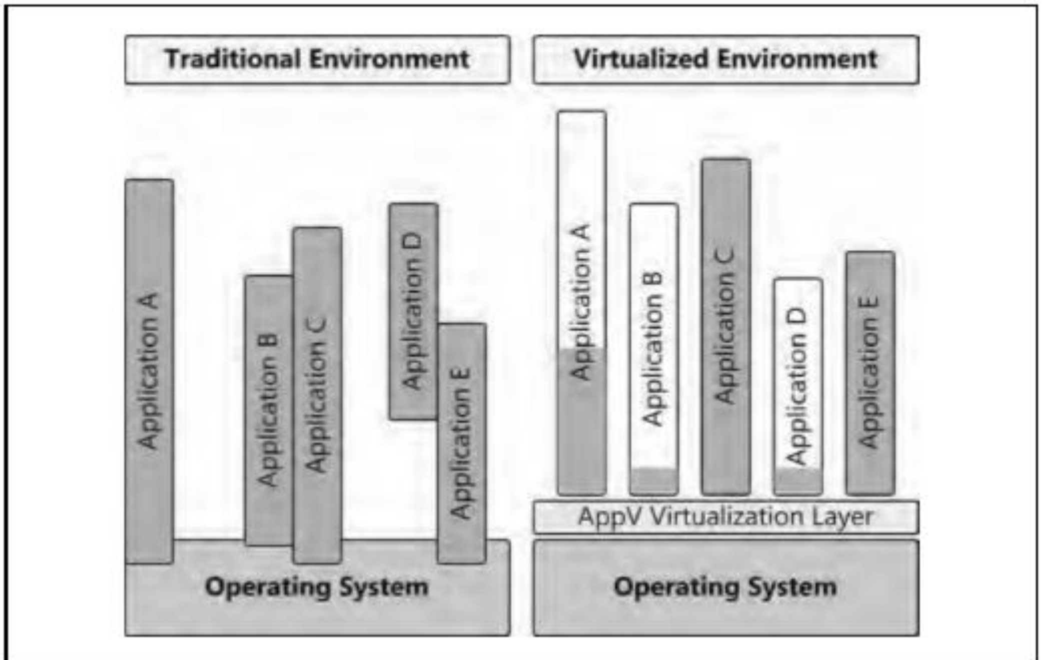


Figure 11-4. AppV creates a protective layer to safeguard the OS from changes.

Q.8 What are the advantages and disadvantages of Virtualization?

A. Virtualization is a computer technology which allows for the creation of a virtual as opposed to an actual or physical version of something, like an operating system, network resource or storage device. For instance a virtual Windows operating system can be easily accessed via a Mac which gives the user an opportunity to work with two operating systems on one computer / hardware. With this approach people and businesses are able to centralize the management of their network. It also reduces the dependency on additional hardware and software which cost extra both in the long and short term.

The Advantages Virtualization

- One of the top advantages of Virtualization is that it requires less hardware to run the same type and amount of software which brings down overall costs.
- Simple data recovery is another great advantage of this technology. For instance if your virtual server suddenly becomes corrupted you simply delete it and restore it from its virtual backup. You do not need to spend time and effort on restoring your entire system from scratch and then restore it from the latest backup. So a corrupted virtual system can be recovered in mere minutes.
- Virtualization provides you with a safe platform on which you can test various software configurations and on various platforms prior to deployment. So in effect you can tinker with the software until you get what

	you exactly want without inadvertently damaging your existing network. • Lower energy consumption since you're running less computer hardware to accomplish the same type of work. • Better system security and reliability. Virtualization systems do not crash due to corruption like device drivers or memory issues. **Some Disadvantages of Virtualization** • Even though rare, physical failures when they do happen can be devastating. For instance if your primary hard disk which contained all your virtual and physical data is suddenly stolen, burnt, broken or corrupted then all your servers both virtual and physical will need to be restored. • Virtualization is mainly dependent on processing power and memory. So you'll need to factor in both much more memory and processing power into your Virtualization strategy. • You'll need to invest in training existing network administrators who do not have the skills to administer a virtual network. • When something goes wrong with a virtualized system it requires complex troubleshooting. This requires expertise and experience of working and troubleshooting Virtualization problems.
Q.9	**Explain the taxonomy of virtual Machine.**
A.	➢ First, VMs are divided into the two major types: **Process VMs** and **System VMs**. In the first type, the VM supports the ABI – user instructions plus system/library calls, in the second, the VM supports a complete ISA – both user and system instructions. ➢ The remainder of the taxonomy is based on whether the guest and host use the same ISA. Using this taxonomy, Figure 0-15 shows the "space" of virtual machines that we have identified.

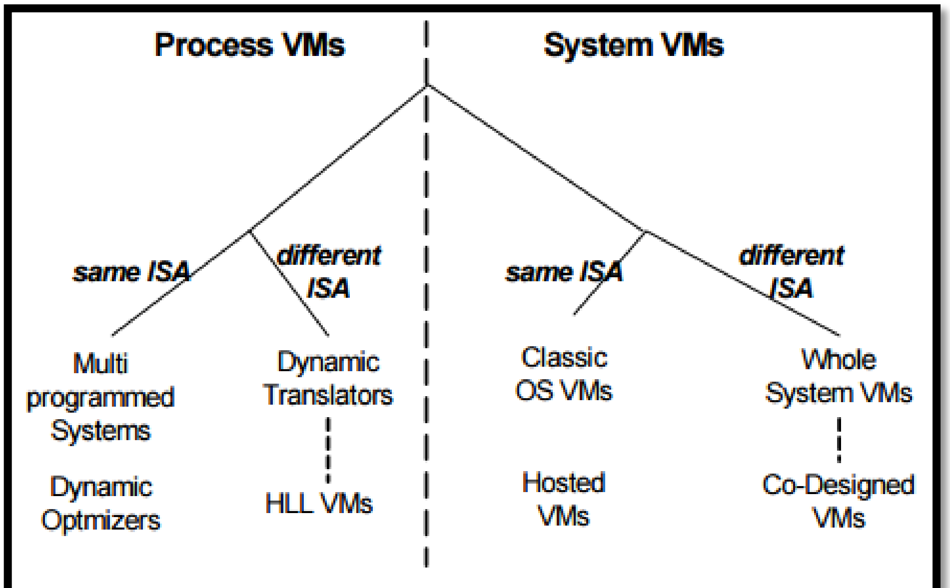


Figure 0-15. A taxonomy of virtual machine architectures.

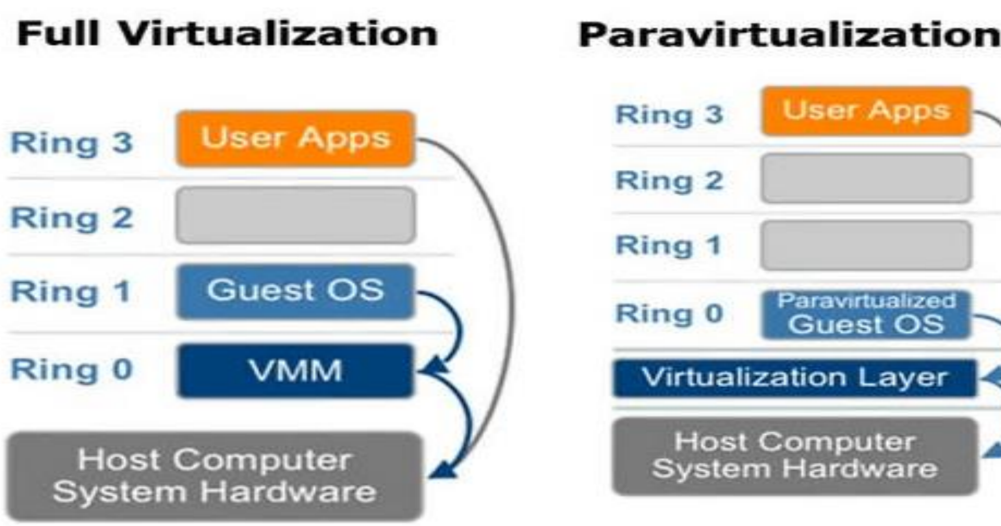
- On the left side of the figure are process VMs. There are two types of process VMs where the host and guest instruction sets are the same.
- The **first** is multi-programmed systems, where virtualization is a natural part of multiprogramming and is supported on most of today's systems.
- The **second** is dynamic optimizers, which transform guest instructions only by optimizing them, and then execute them natively.
- The two types of process VMs that do provide emulation are **dynamic translators** and **HLL VMs**. HLL VMs are connected to the VM taxonomy via a "dotted line" because their process level interface is at a different, higher level than the other process VMs.
- On the right side of the figure are system VMs. These range from Classic OS VMs and Hosted VMs, where replication – and providing isolated system environments – is the goal, to Whole System VMs and CoDesigned VMs where emulation is the goal.
- With **Whole System VMs**, performance is often secondary, in favor of accurate functionality, while with Co-Designed VMs, performance (or power efficiency) is the major goal.
- Here, **Co-Designed VMs** are "dotted line" connected because their interface is at a lower level than other system VMs.

Q.10	**What is process virtual machine? Explain.**
A.	• In computing, a **virtual machine** (**VM**) is an emulation of a particular computer

system. Virtual machines operate based on the computer architecture and functions of a real or hypothetical computer and their implementations may involve specialized hardware, software, or a combination of both.

- Classification of virtual machines can be based on the degree to which they implement functionality of targeted real machines. That way, system virtual machines (also known as full virtualization VMs) provide a complete substitute for the targeted real machine and a level of functionality required for the execution of a complete operating system. On the other hand, process virtual machines are designed to execute a single computer program by providing an abstracted and platform-independent program execution environment.
- Different virtualization techniques are used based on the desired usage. Native execution is based on direct virtualization of the underlying raw hardware, thus it provides multiple "instances" of the same architecture a real machine is based on, capable of running complete operating systems. Some virtual machines can also emulate different architectures and allow execution of software applications and operating systems written for another CPU or architecture. Operating-system-level virtualization allows the resources of a computer to be partitioned via kernel's support for multiple isolated user space instances, which are usually called containers and may look and feel like real machines to the end users.
- Some computer architectures are capable of hardware-assisted virtualization, which enables efficient full virtualization by using virtualization-specific hardware capabilities, primarily from the host CPUs.
- A process VM, sometimes called an application virtual machine, or Managed Runtime Environment (MRE), runs as a normal application inside a host OS and supports a single process. It is created when that process is started and destroyed when it exits. Its purpose is to provide a platform-independent programming environment that abstracts away details of the underlying hardware or operating system, and allows a program to execute in the same way on any platform.
- A process VM provides a high-level abstraction – that of a high-level programming language (compared to the low-level ISA abstraction of the system VM). Process VMs are implemented using an interpreter; performance comparable to compiled programming languages is achieved by the use of just-in-time compilation.
- This type of VM has become popular with the Java programming language, which is implemented using the Java virtual machine. Other examples include the Parrot virtual machine, and the .NET Framework, which runs on a VM called the Common Language Runtime. All of them can serve as an abstraction layer for any computer language.
- A special case of process VMs are systems that abstract over the communication

	mechanisms of a (potentially heterogeneous) computer cluster. Such a VM does not consist of a single process, but one process per physical machine in the cluster. They are designed to ease the task of programming concurrent applications by letting the programmer focus on algorithms rather than the communication mechanisms provided by the interconnect and the OS. They do not hide the fact that communication takes place, and as such do not attempt to present the cluster as a single machine. • Unlike other process VMs, these systems do not provide a specific programming language, but are embedded in an existing language; typically such a system provides bindings for several languages (e.g., C and FORTRAN). Examples are PVM (Parallel Virtual Machine) and MPI (Message Passing Interface). They are not strictly virtual machines, as the applications running on top still have access to all OS services, and are therefore not confined to the system model.
Q.11	**What is system virtual machine? Explain.**
A.	• A virtual machine (VM) is a software implementation of a machine (for example, a computer) that executes programs like a physical machine. Virtual machines are separated into two major classes, based on their use and degree of correspondence to any real machine • A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS). These usually emulate an existing architecture, and are built with the purpose of either providing a platform to run programs where the real hardware is not available for use (for example, executing on otherwise obsolete platforms), or of having multiple instances of virtual machines leading to more efficient use of computing resources, both in terms of energy consumption and cost effectiveness (known as hardware virtualization, the key to a cloud computing environment), or both. • A process virtual machine (also, language virtual machine) is designed to run a single program, which means that it supports a single process. Such virtual machines are usually closely suited to one or more programming languages and built with the purpose of providing program portability and flexibility (amongst other things). An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine—it cannot break out of its virtual environment. • A VM was originally defined by Popek and Goldberg as "an efficient, isolated duplicate of a real machine". Current use includes virtual machines which have no direct correspondence to any real hardware.

	System virtual machines **System virtual machine advantages:** • Multiple OS environments can co-exist on the same primary hard drive, with a virtual partition that allows sharing of files generated in either the "host" operating system or "guest" virtual environment. Adjunct software installations, wireless connectivity, and remote replication, such as printing and faxing, can be generated in any of the guest or host operating systems. Regardless of the system, all files are stored on the hard drive of the host OS. • Application provisioning, maintenance, high availability and disaster recovery are inherent in the virtual machine software selected. • Can provide emulated hardware environments different from the host's instruction set architecture (ISA), through emulation or by using just-in-time compilation. **The main disadvantages of VMs are:** • A virtual machine is less efficient than an actual machine when it accesses the host hard drive indirectly. • When multiple VMs are concurrently running on the hard drive of the actual host, adjunct virtual machines may exhibit a varying and/or unstable performance (speed of execution and malware protection). This depends on the data load imposed on the system by other VMs, unless the selected VM software provides temporal isolation among virtual machines. • Malware protections for VMs are not necessarily compatible with the "host", and may require separate software.
Q.12	**Explain full virtualization and paravirtualization.**
A.	

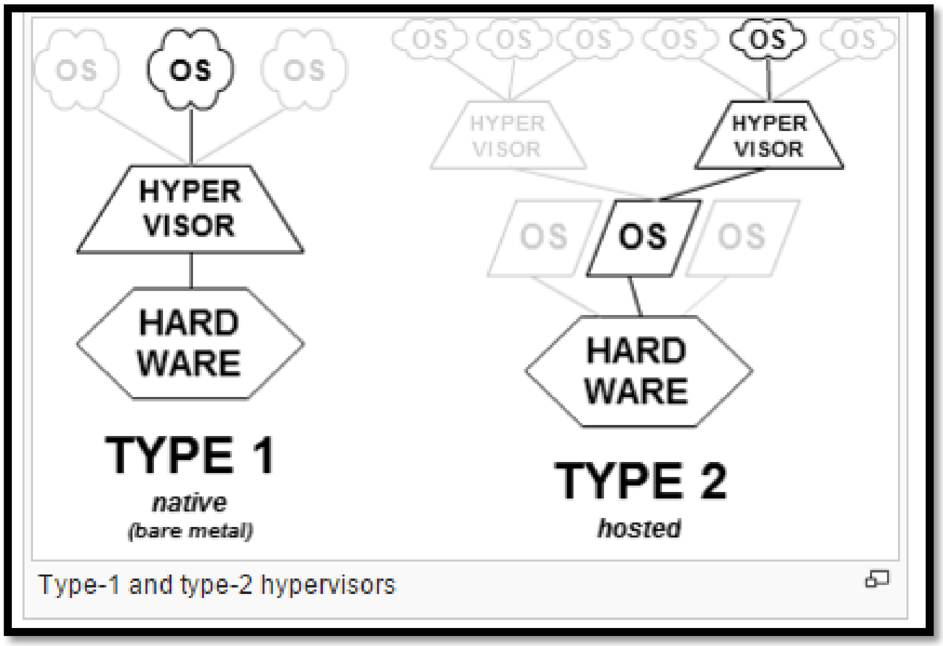
1.Paravirtualization:

- **Paravirtualization** is virtualization in which the guest operating system (the one being virtualized) is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issue commands directly to the host operating system. This also includes memory and thread management as well, which usually require unavailable privileged instructions in the processor.
- Paravirtualization is different from full virtualization, where the unmodified OS does not know it is virtualized and sensitive OS calls are trapped using binary translation at run time.
- In paravirtualization, these instructions are handled at compile time when the non-virtualizable OS instructions are replaced with hypercalls.
- The advantage of paravirtualization is lower virtualization overhead, but the performance advantage of paravirtualization over full virtualization can vary greatly depending on the workload. Most user space workloads gain very little, and near native performance is not achieved for all workloads.
- As paravirtualization cannot support unmodified operating systems (e.g. Windows 2000/XP), its compatibility and portability is poor.
- Paravirtualization can also introduce significant support and maintainability issues in production environments as it requires deep OS kernel modifications.
- The invasive kernel modifications tightly couple the guest OS to the hypervisor with data structure dependencies, preventing the modified guest OS from running on other hypervisors or native hardware.
- The open source Xen project is an example of paravirtualization that virtualizes the processor and memory using a modified Linux kernel and virtualizes the I/O using custom guest OS device drivers.

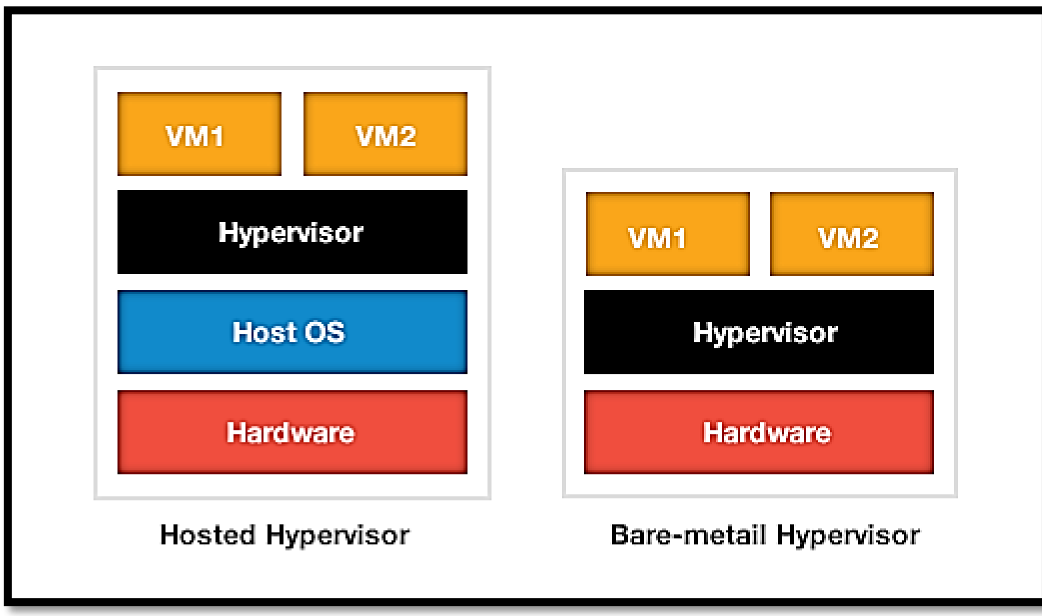
2.Full Virtualization:

- **Full Virtualization** is virtualization in which the guest operating system is unaware that it is in a virtualized environment, and therefore hardware is virtualized by the host operating system so that the guest can issue commands to what it thinks is actual hardware, but really are just simulated hardware devices created by the host.
- Full Virtualization is done with a hardware emulation tool and processor-based virtualization support that allows you to run unmodified guest kernels that are not “aware” they are being virtualized. The result is that you give up performance on these platforms.
- Windows, NetWare, and most closed-source OSs require full virtualization. Many of these guests have PV drivers available, though, which allow for

	devices like disks, network cards, etc., to run with improved performance. ▪ Full virtualization is called "full" because the entire system's resources are abstracted by the virtualization software layer. ▪ Full virtualization has proven highly successful for: ✓ sharing a computer system among multiple users; ✓ isolating users from each other (and from the control program); ✓ Emulating new hardware to achieve improved reliability, security and productivity.
Q.13	**What are hypervisors? Explain.**
A.	✓ A hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. ✓ The guest OS shares the hardware of the host computer, such that each OS appears to have its own processor, memory and other hardware resources. ✓ A hypervisor is also known as a virtual machine manager (VMM). ✓ A hypervisor, also called a virtual machine manager, is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. ✓ However, the hypervisor is actually controlling the host processor and resources, allocating what are needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other. ✓ The hypervisor program installed on the computer allowed the sharing of its memory. ✓ The hypervisor installed on the server hardware controls the guest operating system running on the host machine. Its main job is to cater to the needs of the guest operating system and effectively manage it such that the instances of multiple operating systems do not interrupt one another. ✓ **Hypervisors can be divided into two types**:



Type-1 and type-2 hypervisors



- **Type 1:**
  Also known as native or bare-metal hypervisors, these run directly on the host computer's hardware to control the hardware resources and to manage guest operating systems. For this reason, they are sometimes called bare metal hypervisors. A guest operating system runs as a process on the host. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor.
- **Type 2:**
  Also known as hosted hypervisors, these run within a formal operating

	system environment. In this type, the hypervisor runs as a distinct second layer while the operating system runs as a third layer above the hardware. Type-2 hypervisors abstract guest operating systems from the host operating system. VMware Workstation, VMware Player and Virtual Box are examples of type-2 hypervisors.
Q.14	**Explain virtualization, virtual machine and virtual machine monitor.**
A.	**<u>Virtualization:-</u>** • Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources* • It is the process by which one computer hosts the appearance of many computers. • Virtualization is used to improve IT throughput and costs by using physical resources as a pool from which virtual resources can be allocated. • Virtualization is a technology that transfers hardware into software. • Virtualization allows us to run multiple Operating Systems as VMs on single computer. • "Virtualization software makes it possible to run multiple operating systems and multiple applications on the same server at the same time". • "It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware." • The technology behind virtualization is known as a virtual machine monitor (VMM) or virtual manager, which separates compute environments from the actual physical infrastructure. **<u>Virtual Machine:-</u>** • A virtual machine (VM) is a software program or operating system that not only exhibits the behaviour of a separate computer, but is also capable of performing tasks such as running applications and programs like a separate computer. • A virtual machine, usually known as a guest is created within another computing environment referred as a "host." Multiple virtual machines can exist within a single host at one time. • A virtual machine is also known as a guest. **<u>Virtual Machine Monitor:-</u>** • A Virtual Machine Monitor (VMM) is a software program that enables the creation, management and governance of virtual machines (VM) and manages the operation of a virtualized environment on top of a physical host machine. • VMM is also known as Virtual Machine Manager and Hypervisor. However, the provided architectural implementation and services differ by vendor

	product. • VMM is the primary software behind virtualization environments and implementations. When installed over a host machine, VMM facilitates the creation of VMs, each with separate operating systems (OS) and applications. VMM manages the backend operation of these VMs by allocating the necessary computing, memory, storage and other input/output (I/O) resources. • VMM also provides a centralized interface for managing the entire operation, status and availability of VMs that are installed over a single host or spread across different and interconnected hosts.
Q.15	**Compare process virtual machines and system virtual machines.**
A	**<u>System Virtual Machines:</u>** • A system platform that supports the sharing of the host computer's physical resources between multiple virtual machines, each running with its own copy of the operating system. • The virtualization technique is provided by a software layer known as a hypervisor, which can run either on bare hardware or on top of an operating system. **<u>Process Virtual Machine:</u>** • Designed to provide a platform-independent programming environment that masks the information of the underlying hardware or operating system and allows program execution to take place in the same way on any given platform. • A process virtual machine or application Virtual machine is designed to run a single program with a single process. • It runs just like a regular application within the host OS as a process. • The VM is created when process is initiated and destroyed when the process exits or dies. • A Process VM is sometimes referred to as application virtual machine. • This VM mainly aims at providing a platform-independent development environment. • Java programming language is platform independent as it implements Java Virtual Machine (JVM) which is a process VM.