

Cloud Computing

Prepared by Dr. Sherif Mostafa

IT381

Lecture 2



Virtualization

Computer System

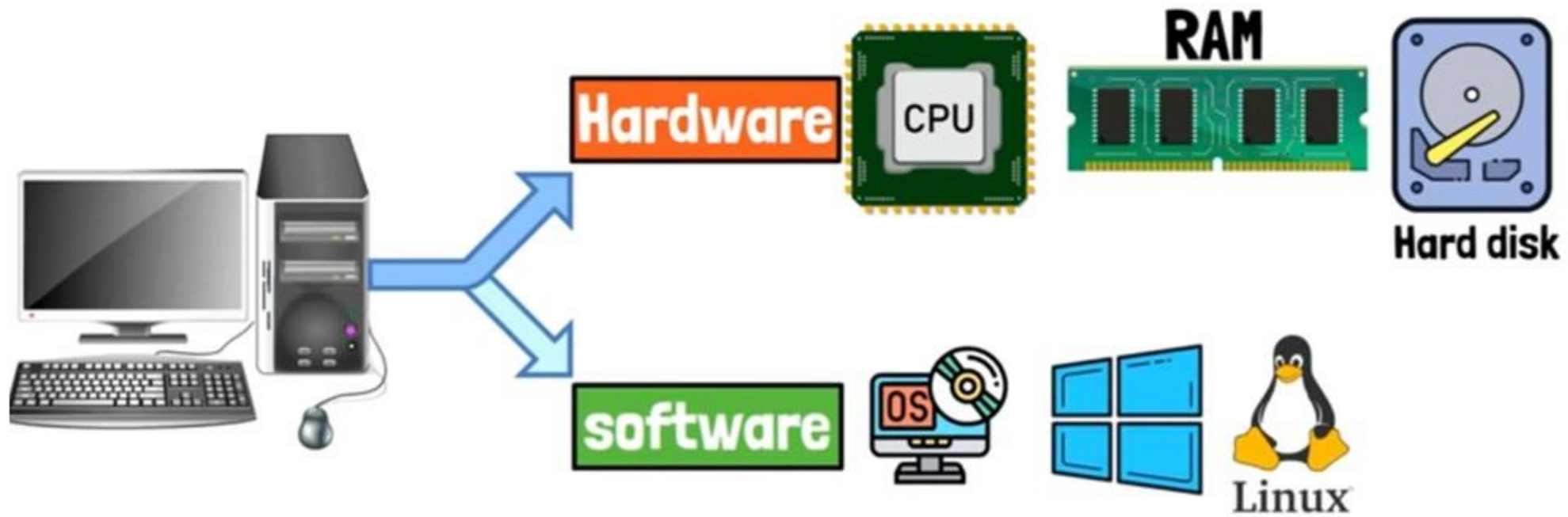
Virtual machine

Hypervisor

Adv. Of Virtualizations

Layering and Virtualization

Computer System



Computer System

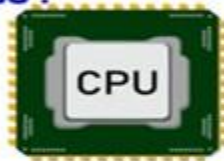
We all know that any computer consists of hardware and software. Hardware includes the device's resources such as CPU, RAM, and hard disk. Software includes the operating system, such as Windows, Linux, or others, and of course all the programs and applications on the device. It was natural for the entire device to be on one operating system, which supports a group of programs and applications

Computer Resources

1960

Computer Hardware resources :

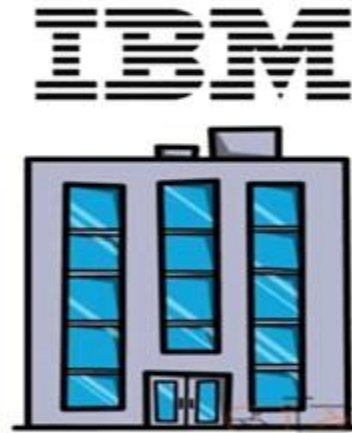
👉 CPU (Central Processing Unit)



👉 RAM (Random Access Memory)



👉 storage

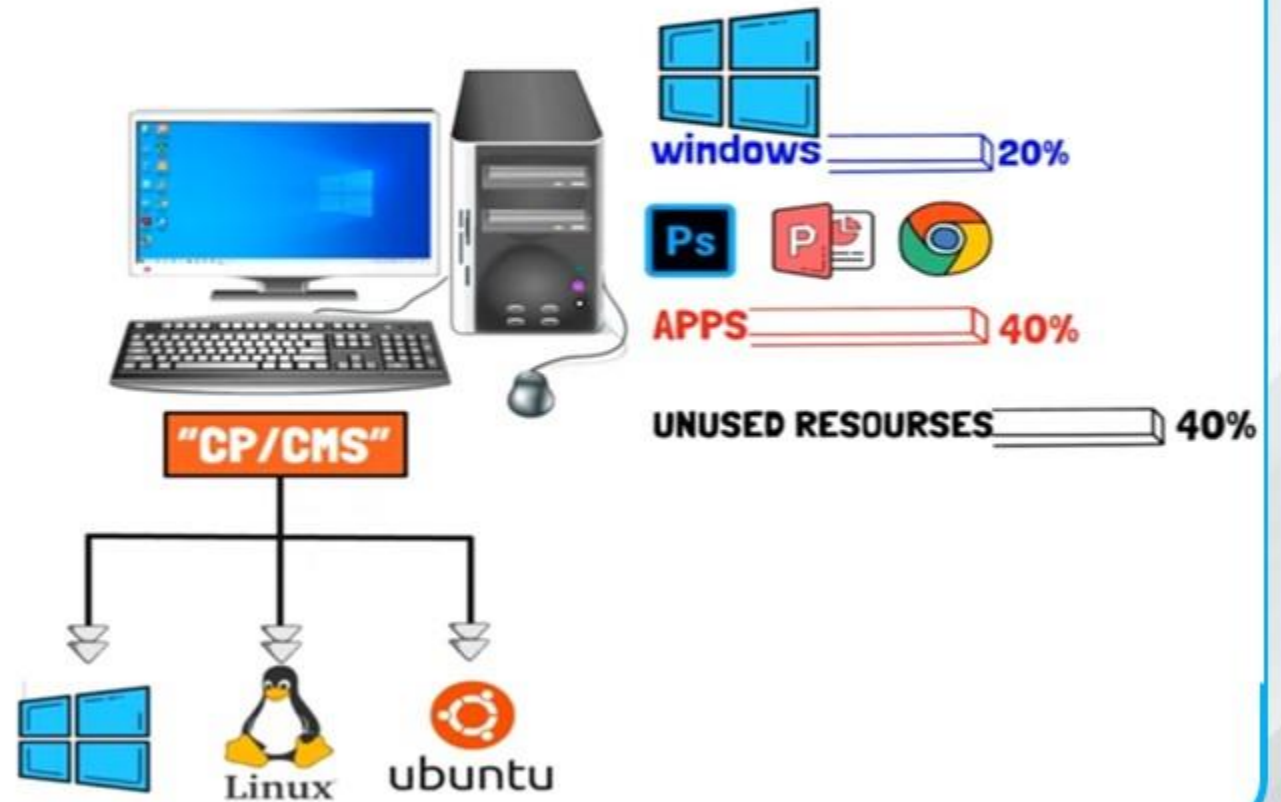


but in the sixties of the last century, a team from IBM thought that they wanted to exploit all the device's resources, which are, as we agreed, represented by the CPU, RAM, and storage, whether HDD or SSD

Computer Resources



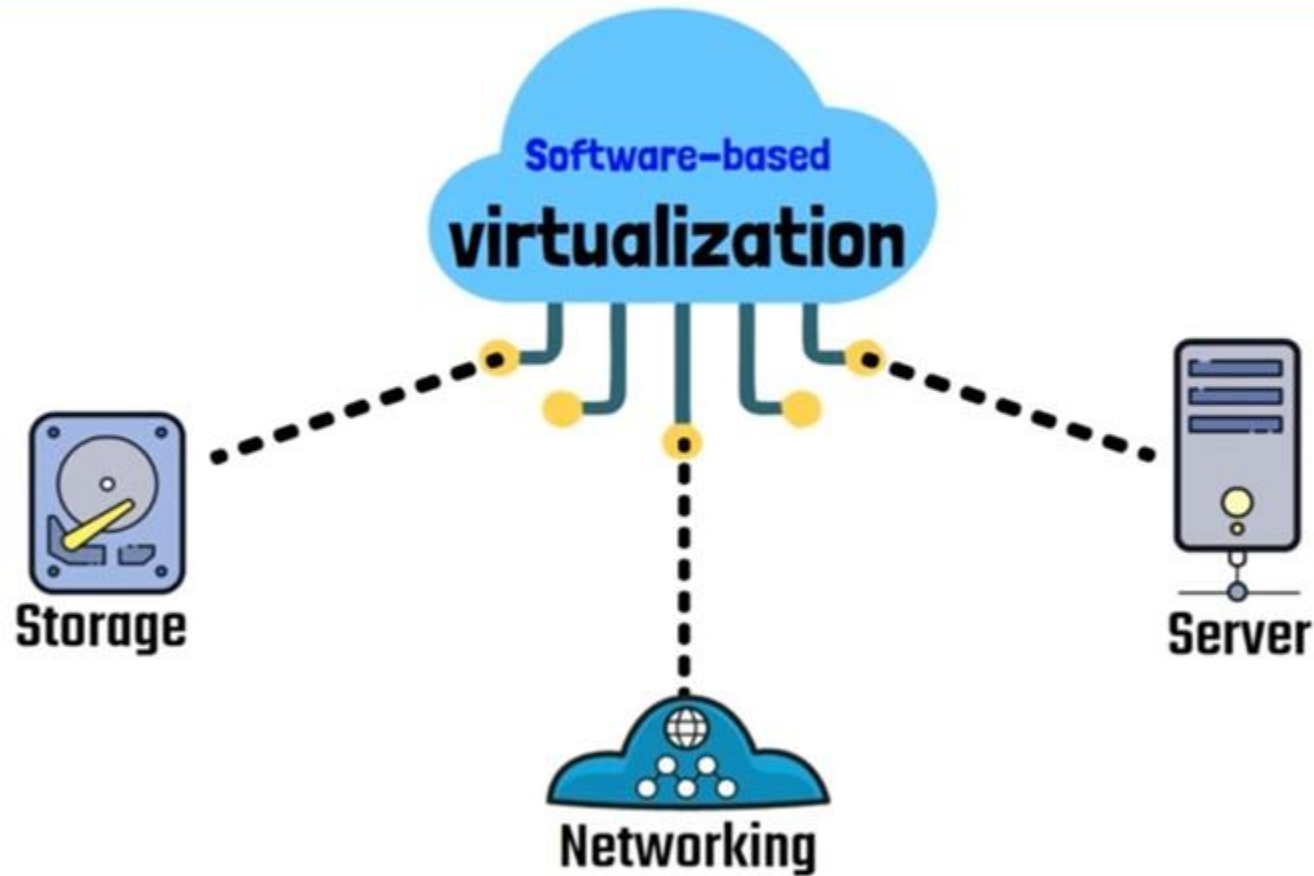
Server



Computer Resources

This is because when you download Windows and a group of programs, for example, you do not exploit all the device's resources, meaning that Windows will take some of the device's resources and programs as well, but a very large part will not be exploited. This is a simplification of the idea, but of course, the technology was originally aimed at providing the resources of servers and the huge storage devices that belong to large companies. This team thought of running more than one operating system on one device, and they implemented and developed a technology called CP/CMS. This technology gives me the ability to run more than one operating system, and thus they were able to exploit all the device's resources.

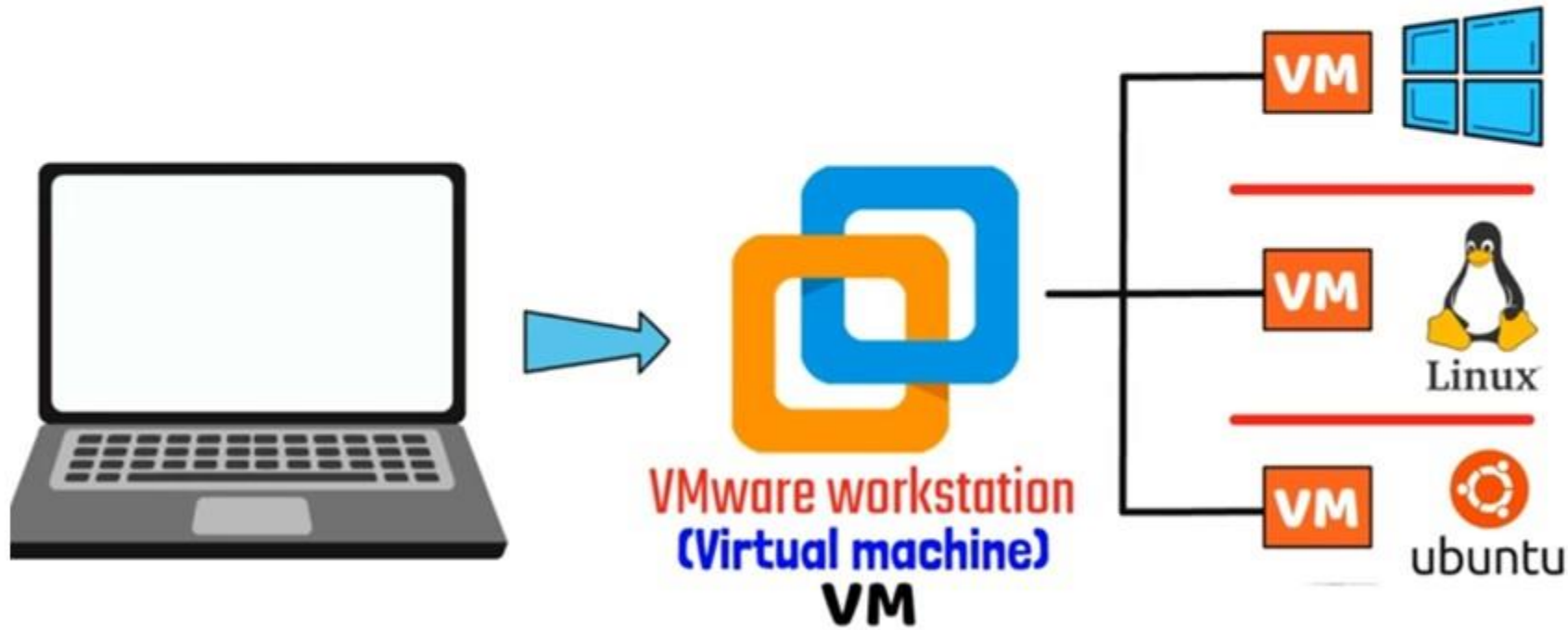
Virtual Machine



Virtual Machine

Here was the beginning of virtualization technology, which is the process of creating something virtual for anything physical, meaning software based, that plays the role of something physical or something normal hardware, whether it is a computer, which is servers, storage, or networking. All of this can be made software based or virtual, meaning virtual. After that, this technology continued to develop until the nineties.

Virtual Machine

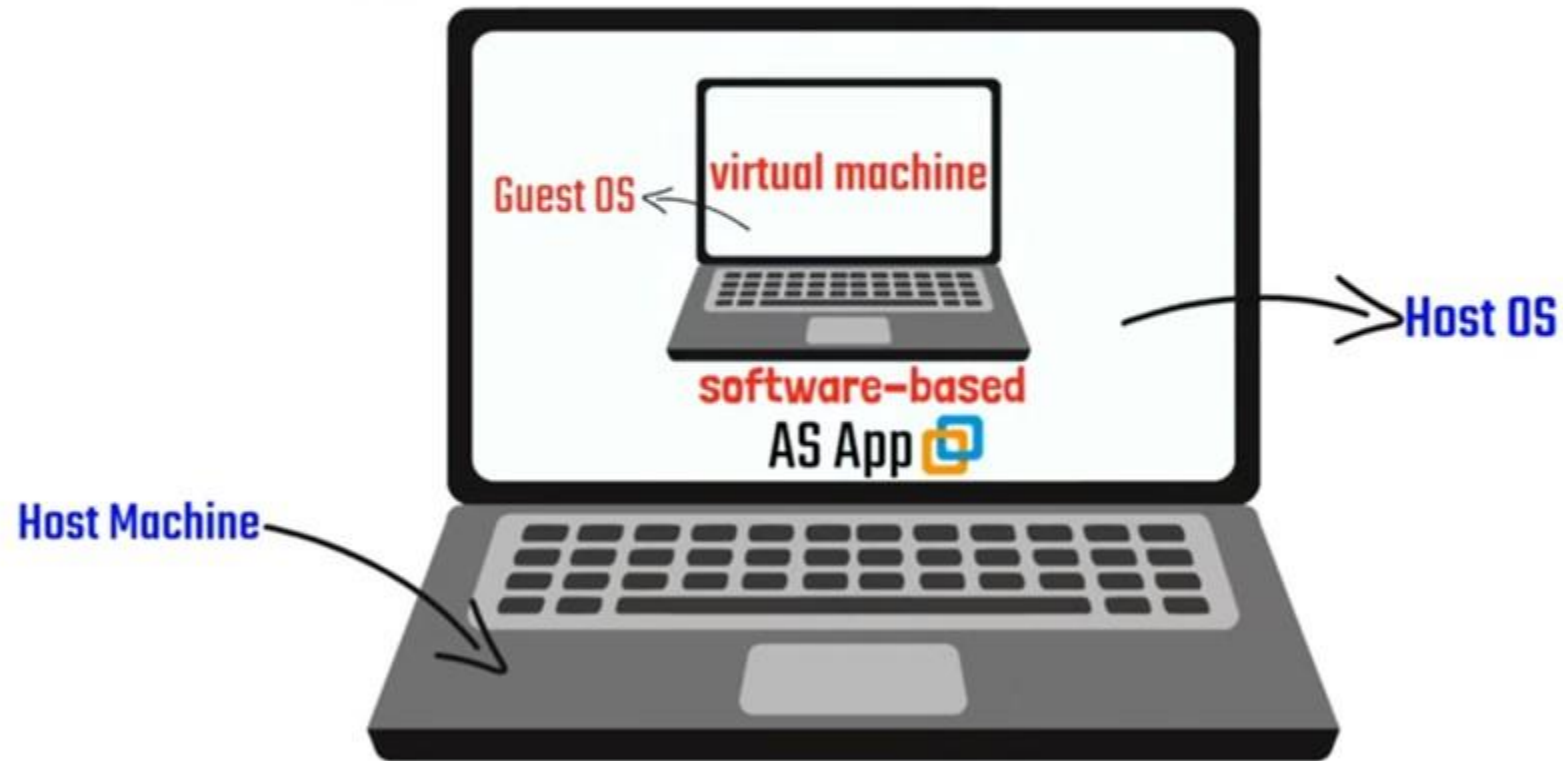


Virtual Machine

VMware came to us with a brilliant solution that would change a lot in the field of virtualization. They created a software program called VMware Workstation, which allows any device to have more than one virtual device, which we call a virtual machine, abbreviated as VM. Each VM can have a different operating system and be isolated from the rest of the VMs on the same physical device. This was, of course, a very big shift in the world of virtualization and IT, and it solved many problems. This is because some developers need to work on the Linux operating system, and the device itself has Windows on it, so they resorted to VM Workstation.

Virtual Machine

Virtual machine:



Virtual Machine

Let's understand together what the idea of a virtual machine is. The idea of a virtual machine is that there is a computer that has all the hardware elements, but it is software based, meaning it is software that runs on another computer and takes all the resources it needs and reserves them from the host device. My physical device is called the host machine and the operating system on it is called the host OS, but the device I put on is called the virtual machine and its operating system is called the guest OS.

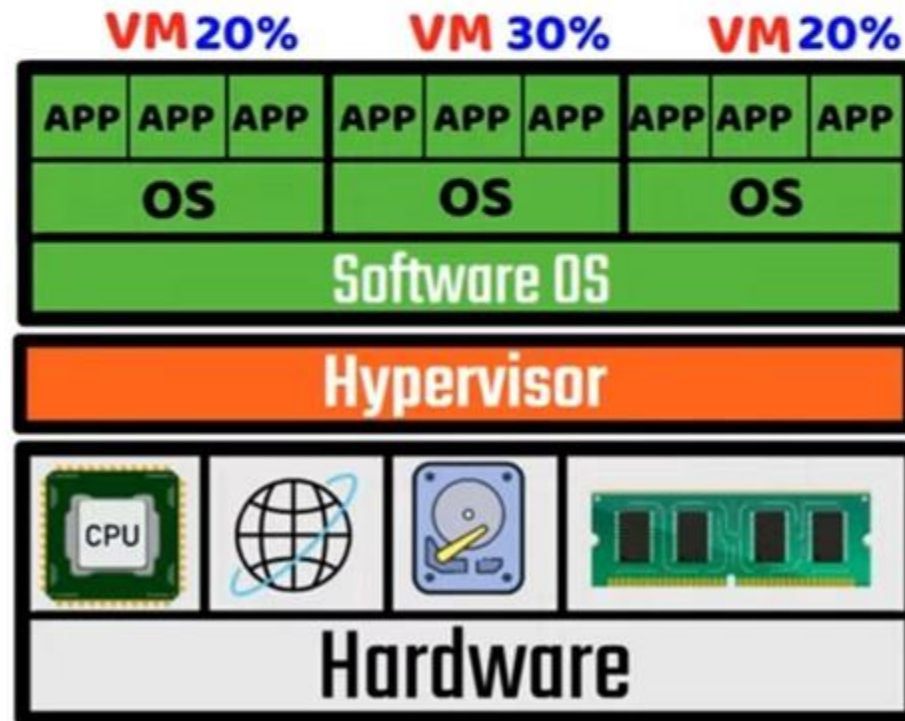
Hypervisor

Virtualization

Hypervisor:



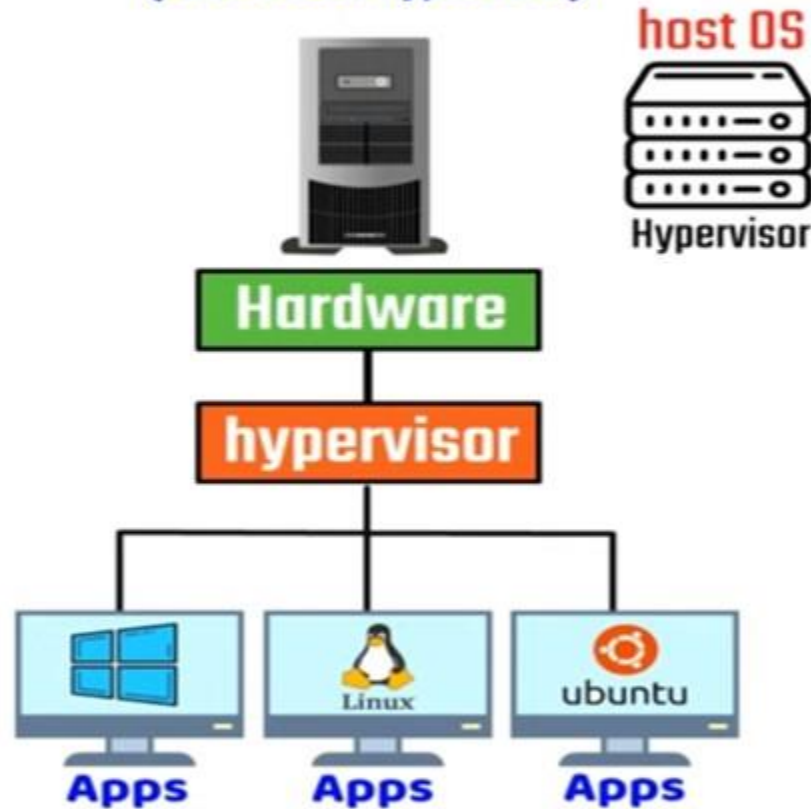
Virtualization Software



100 %

Now we can understand virtualization in general. Let's see how the process itself is done. The first and most important element in this process is the hypervisor, which is the software that performs virtualization, meaning you can consider it a layer like the one that exists between the software OS and the hardware, which is, as we agreed, the computer's resources. Its job is to manage the hardware resources so that it allocates to each VM the resources it needs according to the configuration that I do when I create the VM. We have two main types, type 1 and type 2.

Hypervisor (Type 1 & 2)

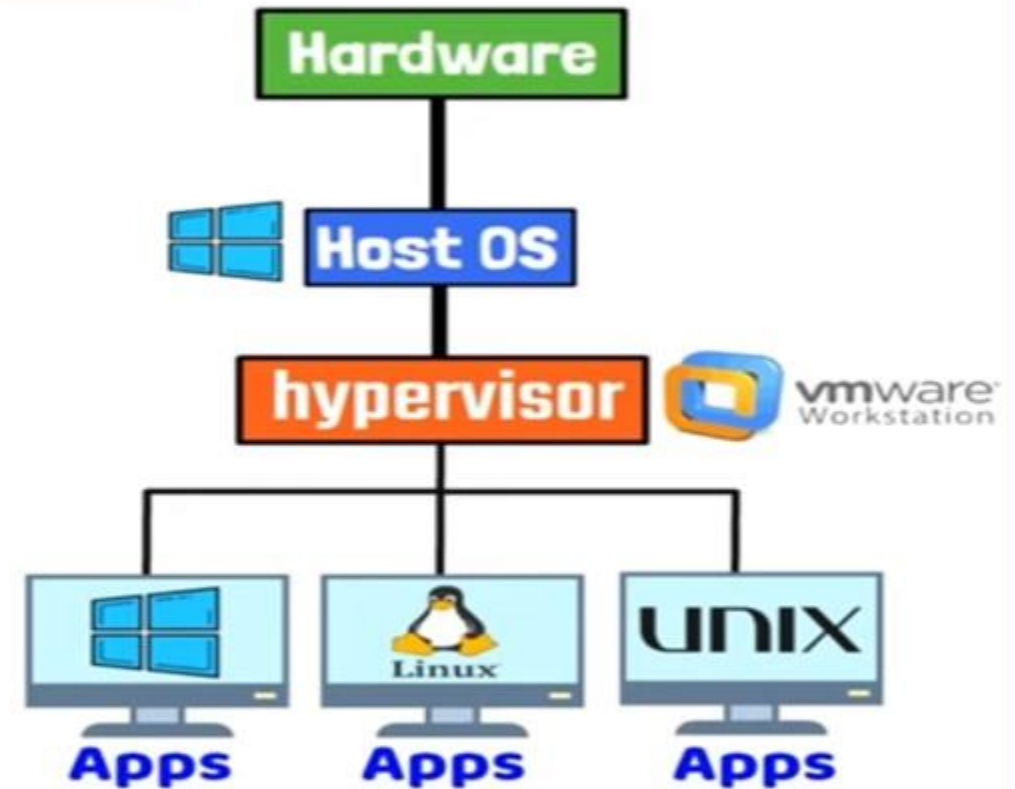
Type-1
(bare-metal hypervisor)

Examples



hypervisor

Type-2



examples

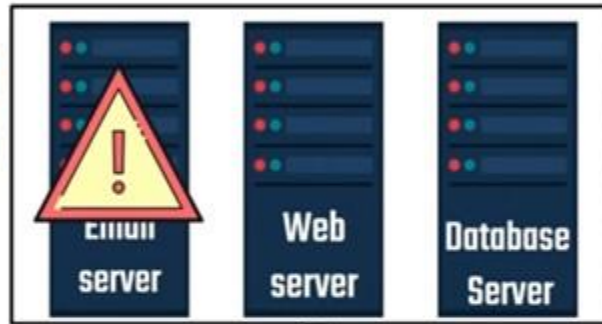
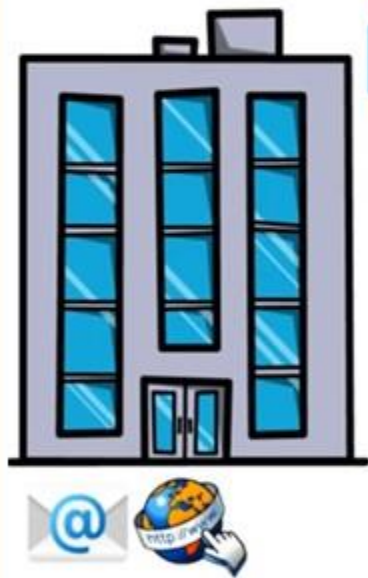


We start with the type 1 hypervisor, which is called (bare-metal hypervisor) because it is installed directly on the hardware itself as if it is the host operating system, and of course after I install it, it will allow me to run more than one virtual machine. Each VM may have a different OS and its own applications. Type 1 is the best and is always used in companies because it is applied to servers. The most famous examples of type 1 hypervisor are ESXi from VMware and Hyper-v from Microsoft, and there is also KVM.

Chapter

02

Hypervisor (Type 1): Example



3



UNIX



+3

Type-1
(bare-metal hypervisor)
one server

ESXi hypervisor
vmware



+3

Hardware

hypervisor



Hypervisor (Type 1): Example

Okay, let's take an example together to understand the subject more. We have company X that wants to create an email for employees. The company will need to have an email server to install an OS, for example Windows, and then install the email application, such as Exchange, for example. The same thing is if we want to create a website for the company. Here we will need to bring a webserver and put an OS on it, which will be Linux, for example. Of course, any company needs a database for its employees, so here we need another server to put on it, for example a Unix OS dedicated to the database.

Hypervisor (Type 1): Example

So, suppose any of these servers crashed or had any problem, then there would be a lot of work that would stop. Here the company would bring another physical server and put on it the same OS and the same application, and it would be on standby so that if the main server crashed, the second one would take its place. Here, the minimum we will need is 3 servers, and of course, if I want to ensure the continuity of work, I will need three more like them. But if I apply the concept of virtualization, I can bring just one physical server and install a type 1 hypervisor on it, such as ESXI from VMware. After that, I can create VMs, and each VM will have a specific OS and a specific application. Here, I can use only one server and get the same result as if I had three separate servers.

Hypervisor (Type 1)

Advantages

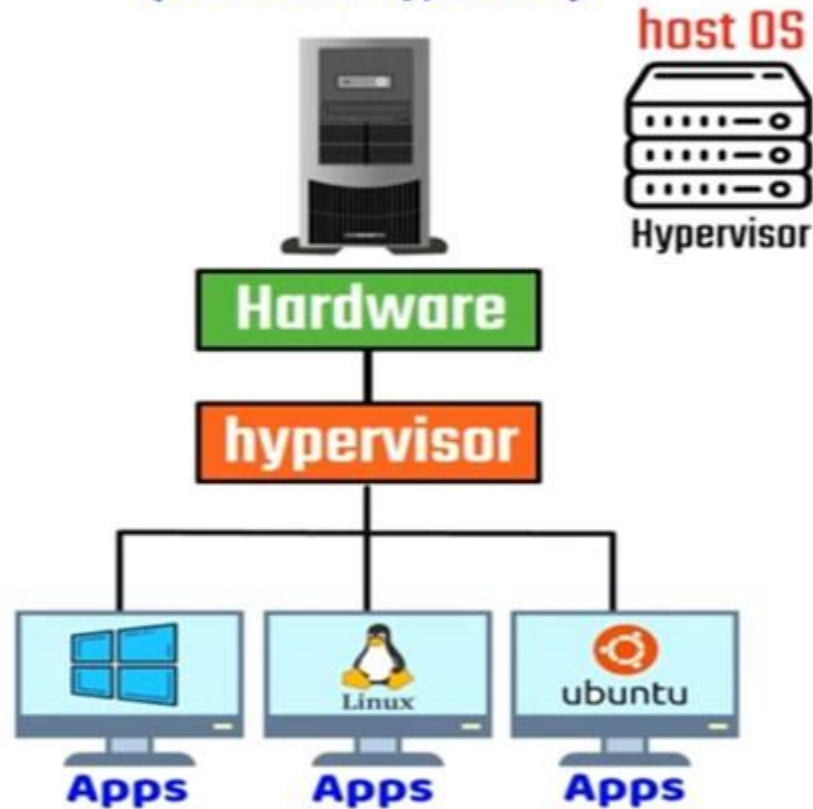


Management

Here, of course, the company saved the money for the servers and saved the cooling and energy that they would have needed to operate. Also, the maintenance of the servers, in addition to the fact that their management will be much simpler.

Hypervisor (Type 2)

Type-1
(bare-metal hypervisor)

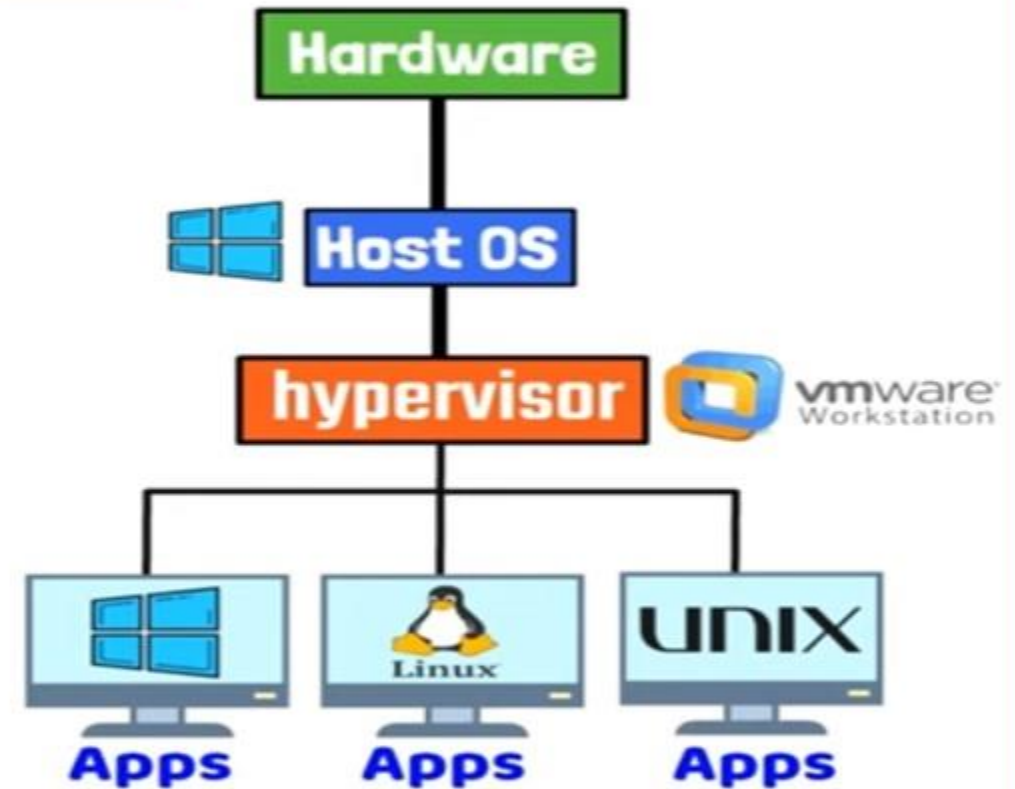


Examples



hypervisor

Type-2



examples



Now let's move on to our second type, which is type 2 hypervisor. Here, I install the host OS on the host, such as Windows, and on this OS, I install the hypervisor, such as VM workstation, which we talked about a while ago. It is a program that is downloaded on my OS and allows me to create virtual machines. Of course, I can download the guest OS on each virtual machine and then install the application for each VM. This is exactly what you can apply on your device.

Hypervisor (Type 2)

Uses



**Testing
Virtualized environment**



windows

linux

Type-2
Hypervisor



Hypervisor (Type 2): Example

For example, if you have a host OS that is Windows, you can install the VM workstation program, and this will be the type 2 hypervisor that will give you the ability to create different virtual machines inside your device. Of course, type 2 is much slower than type 1 because the host OS will take hardware resources, and of course it will not be reliable at all for me to rely on it in a large datacenter. So, what can we use type 2 hypervisor for? Most of the use of type2 hypervisors is in testing, meaning that if developers want to try a specific SW, they can create a virtual environment to test this SW on more than one operating system.

Hypervisor (Type 2): Example

The biggest use is for students for educational purposes. For example, server administration courses that require students to install Win Server on their Win10 device, so they resort to type 2 hypervisors, the most famous types of which are Client Hyper-V, VM Workstation, and Oracle Virtual Box.



Server administration
courses



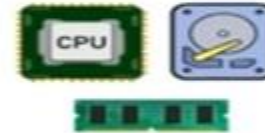
Hypervisor (Type 2): disadvantages

Type-2

Disadvantages



Slower than type-1



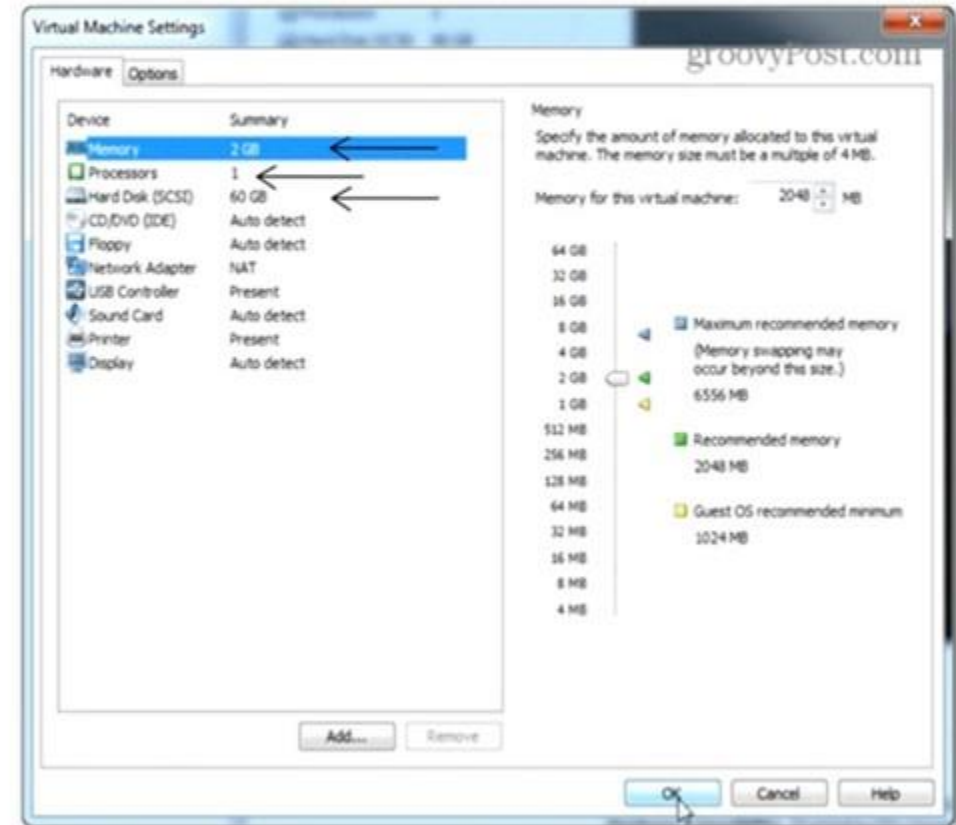
Unreliable



Chapter

02

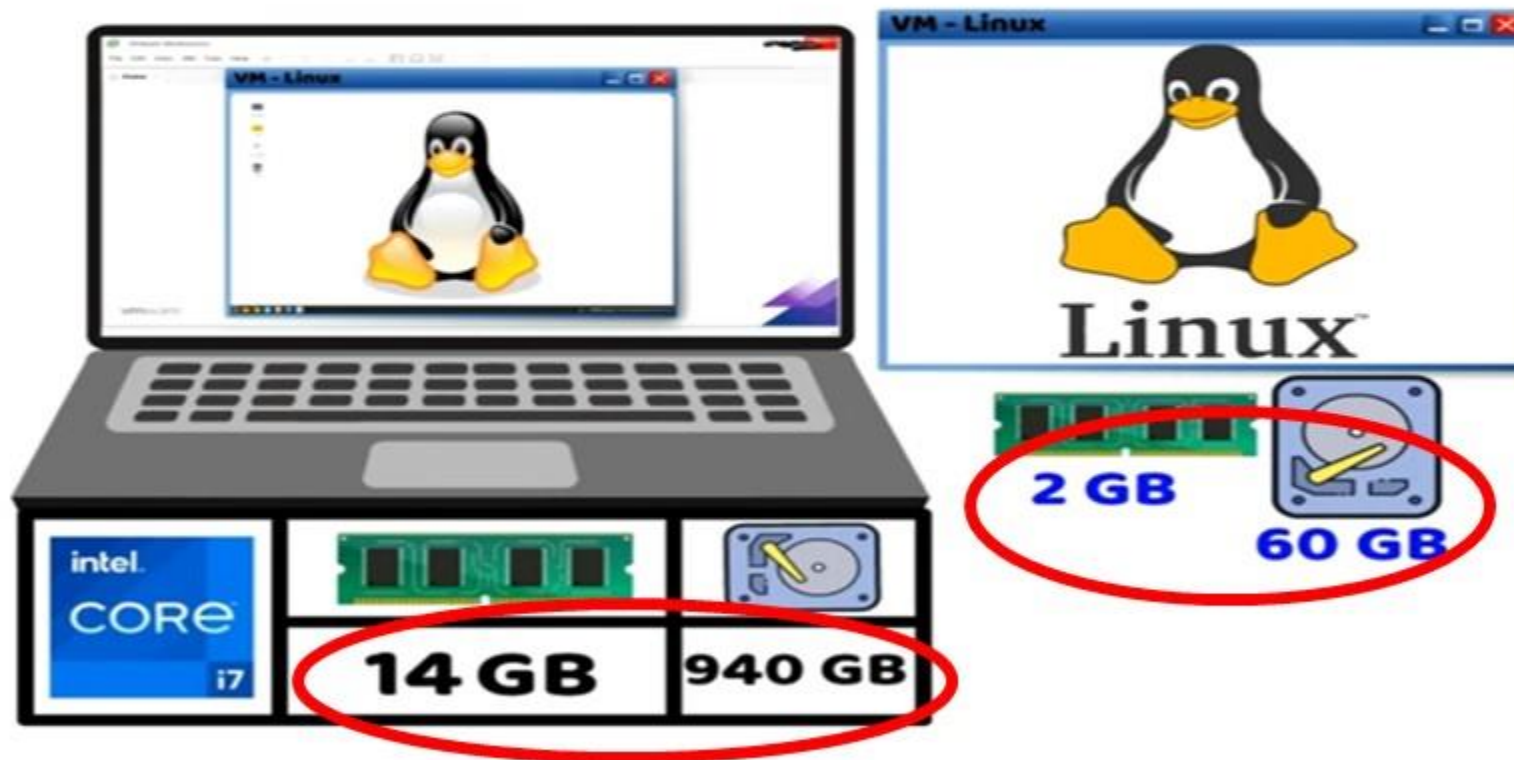
Example on Type-2



Example on Type-2

Let's take an example of type 2 hypervisor to understand more. Now you have your device installed with Win10 and its resources are CPU core i7, RAM 16 GB, and hard disk 1 terabyte. And you wanted to learn Linux, for example. Instead of buying another device to install Linux on, you will install a type 2 hypervisor like VM workstation, and you will start creating a new virtual machine. When you create a new virtual machine, you choose all its resources as if you were buying a regular computer and you choose how much CPU it will take, how much RAM you will allocate to it, and how much hard disk, and now I have a VM with its resources.

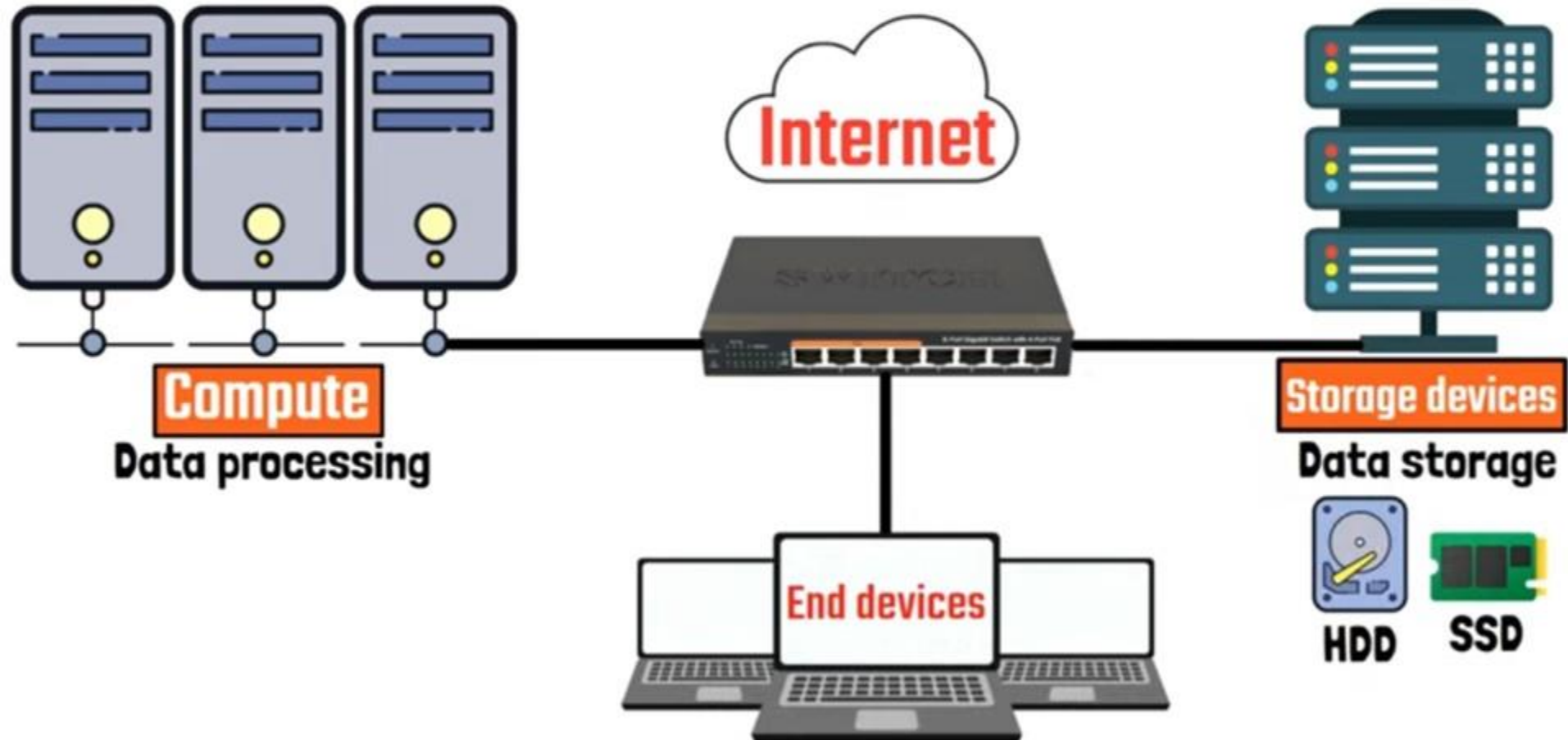
Example on Type-2



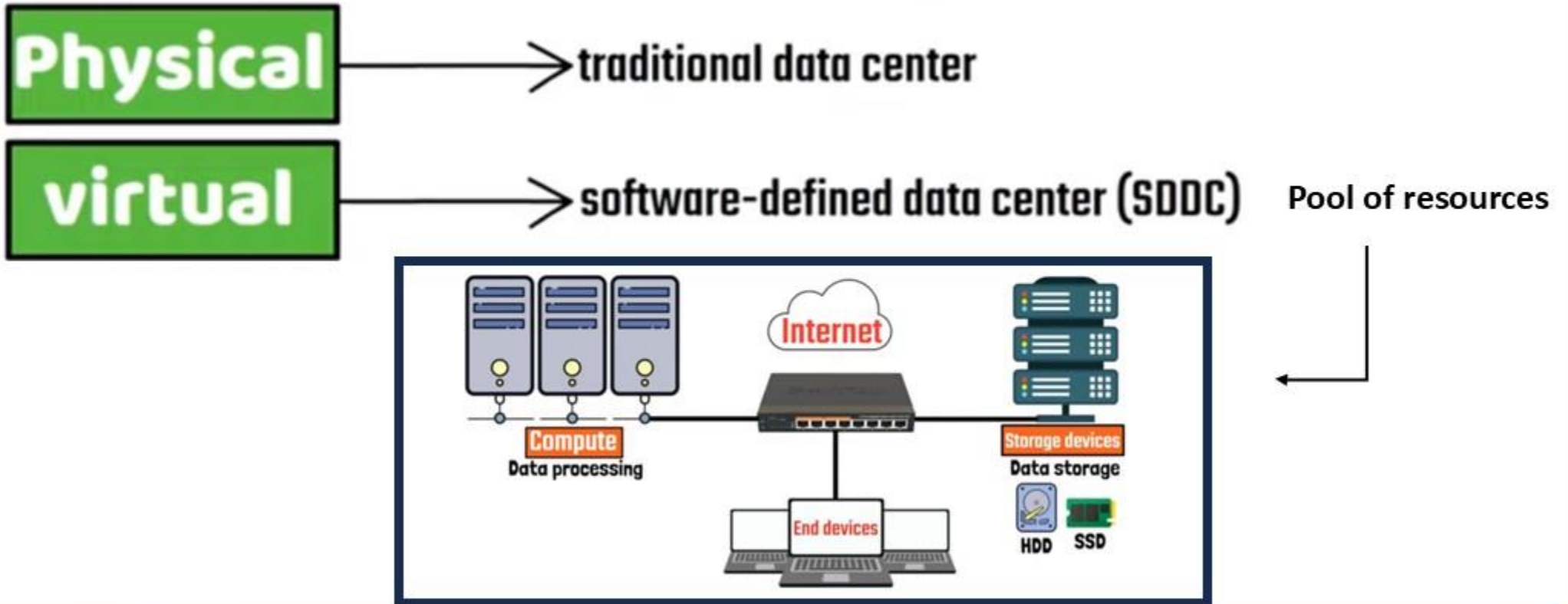
Example on Type-2

Now I have a VM on my device. Let's assume that the new VM has 2 gigabytes of RAM and 60 gigabytes of hard disk, so it has taken from the resources on your device, so the ram has become only 14 gigabytes, and the hard disk is 940 gigabytes.

Datacenter

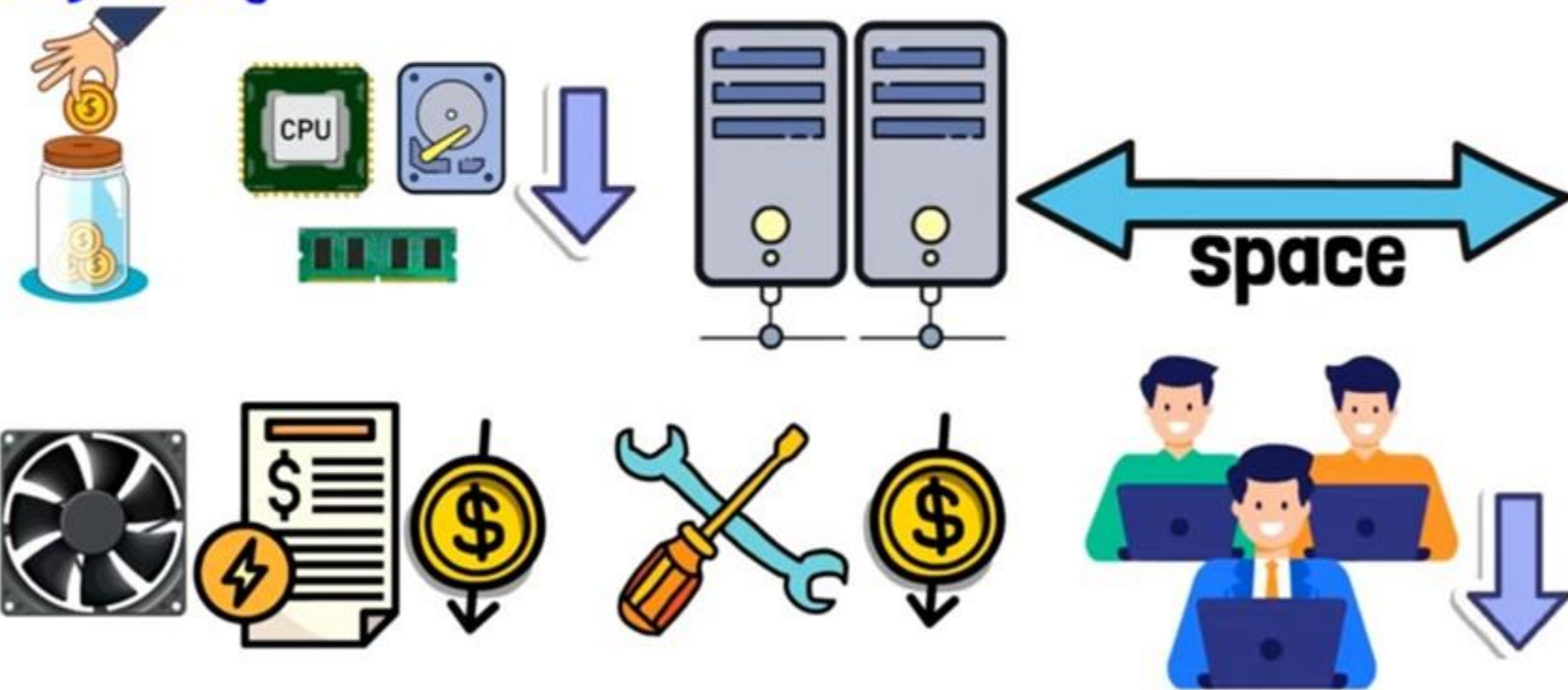


Motivation



Advantages

1- Money-saving

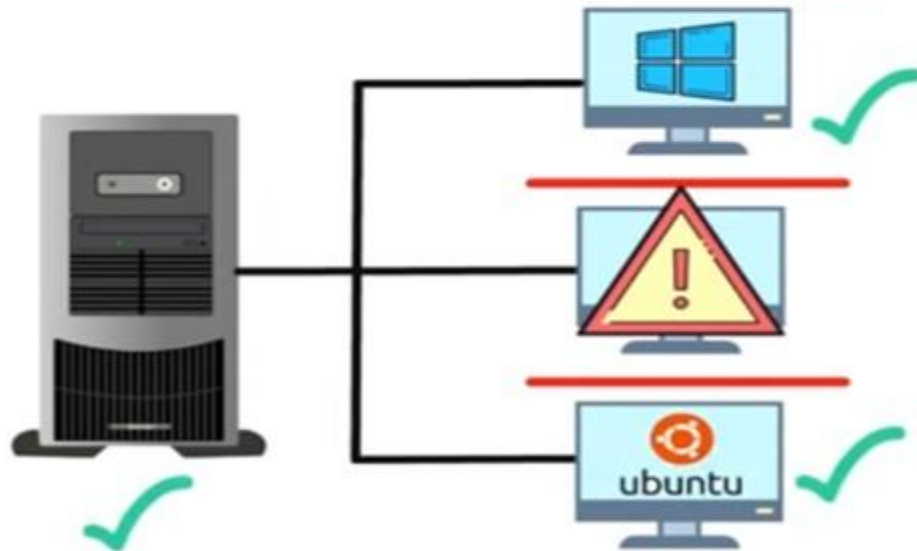


Advantages

2- Business Continuity

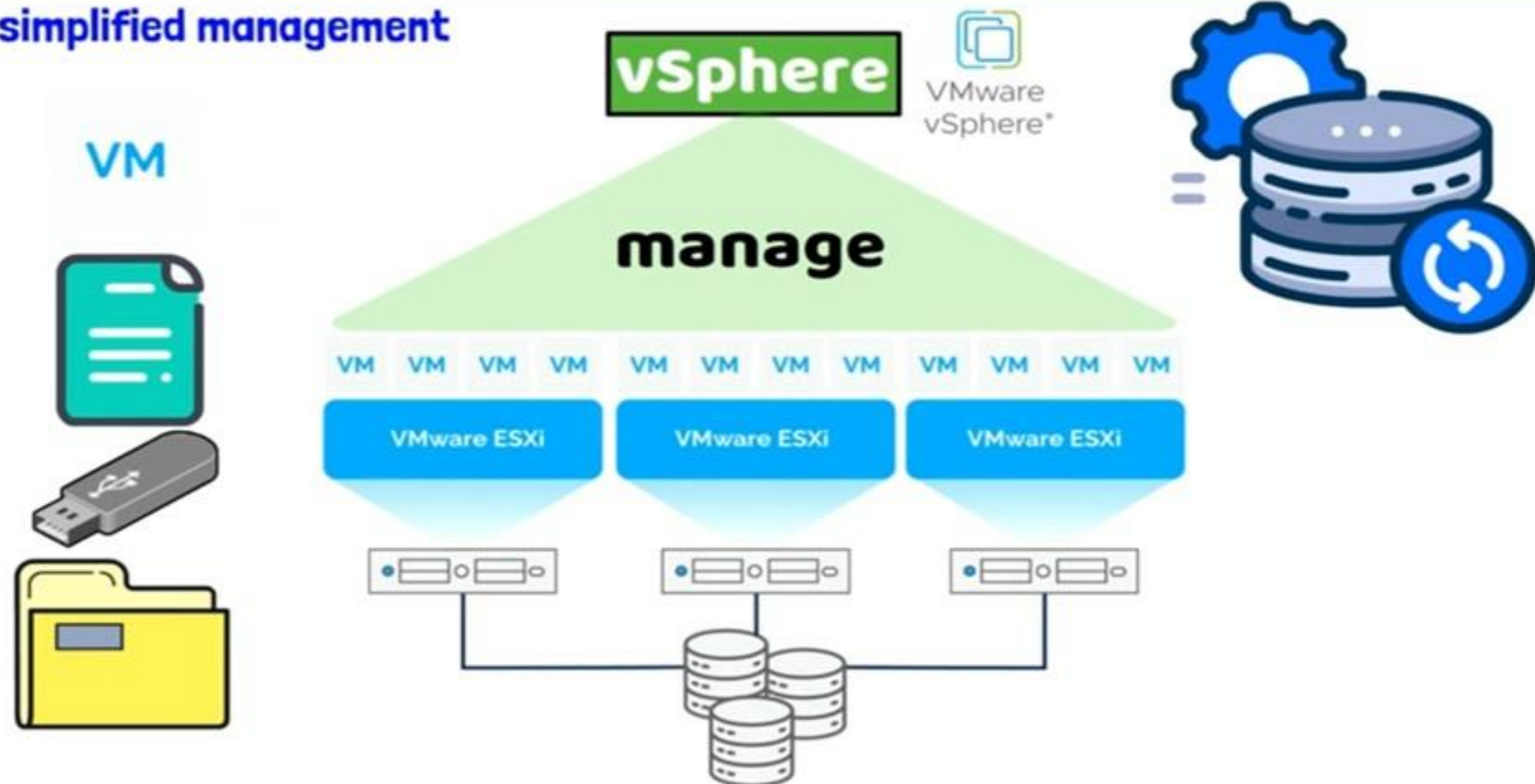


decrease down time



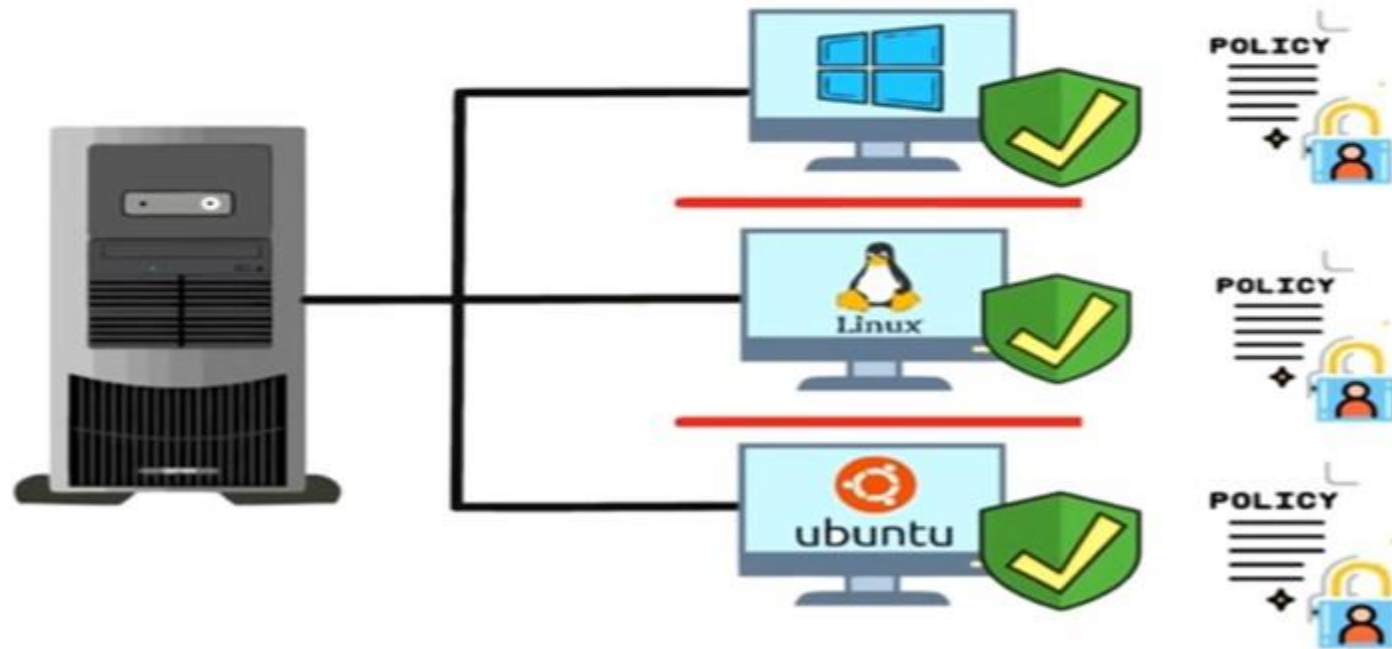
Advantages

3- simplified management



Advantages

4- security



Motivation

- Three fundamental abstractions are necessary to describe the operation of a computing systems:
 - (1) interpreters/processors, (2) memory, (3) communications links
- As the scale of a system and the size of its users grows, it becomes very challenging to manage its recourses (see three points above)
- Resource management issues:
 - provision for peak demands → **overprovisioning**
 - **heterogeneity** of hardware and software
 - machine failures

Motivation

- **Virtualization is a basic enabler of Cloud Computing; it simplifies the management of physical resources for the three abstractions**
- For example, the state of a virtual machine (VM) running under a virtual machine monitor (VMM) can be saved and migrated to another server to balance the load
- For example, virtualization allows users to operate in environments they are familiar with, rather than forcing them to specific ones

Motivation

- *"**Virtualization**, in computing, refers to the act of creating a **virtual (rather than actual) version of something**, including but not limited to a virtual computer hardware platform, operating system (OS), storage device, or computer network resources."* from Wikipedia
- Virtualization abstracts the underlying resources; simplifies their use; isolates users from one another; and supports replication which increases the elasticity of a system

Motivation

- Cloud resource virtualization is important for:
 - Performance isolation
 - as we can dynamically assign and account for resources across different applications
 - System security:
 - as it allows isolation of services running on the same hardware
 - Performance and reliability:
 - as it allows applications to migrate from one platform to another
 - The development and management of services offered by a provider

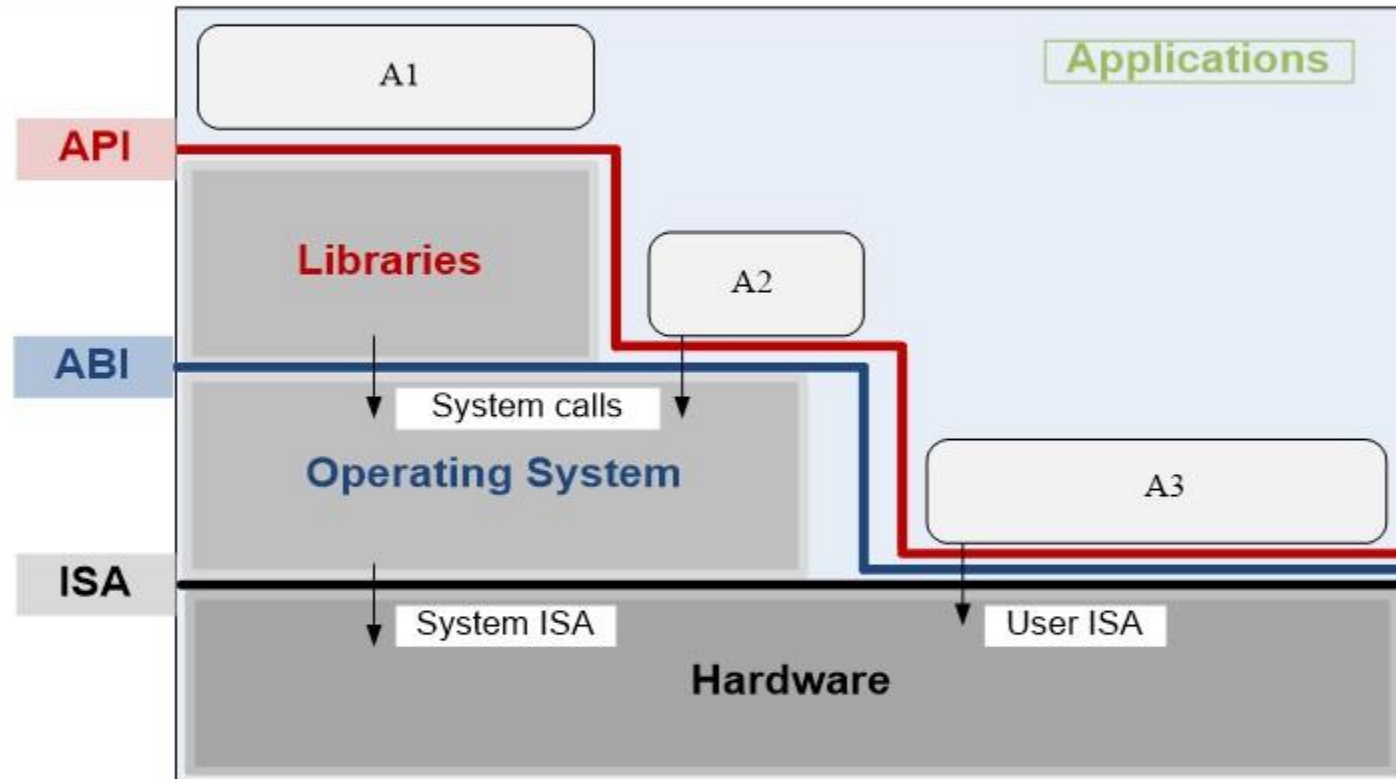
Virtualization

- Virtualization simulates the interface to a physical object by:
 - Multiplexing: creates multiple virtual objects from one instance of a physical object. Many virtual objects to one physical. Example - a processor is multiplexed among a number of processes or threads.
 - Aggregation: creates one virtual object from multiple physical objects. One virtual object to many physical objects. Example - a number of physical disks are aggregated into a RAID disk.
 - Emulation: constructs a virtual object of a certain type from a different type of a physical object. Example - a physical disk emulates a Random Access Memory (RAM).
 - Multiplexing and emulation. Examples - virtual memory with paging multiplexes real memory and disk; a virtual address emulates a real address.

Layering and Virtualization

- Layering – a common approach to manage system complexity:
 - Simplifies the description of the subsystems; each subsystem is abstracted through its interfaces with the other subsystems
 - Minimises the interactions among the subsystems of a complex system
 - With layering we are able to design, implement, and modify the individual subsystems independently
- Layering in a computer system:
 - Hardware
 - Software
 - Operating system
 - Libraries
 - Applications

Layering and Interfaces



Application Programming Interface (API), Application Binary Interface (ABI), and Instruction Set Architecture (ISA). An application uses library functions (A1), makes system calls (A2), and executes machine instructions (A3) (from book)

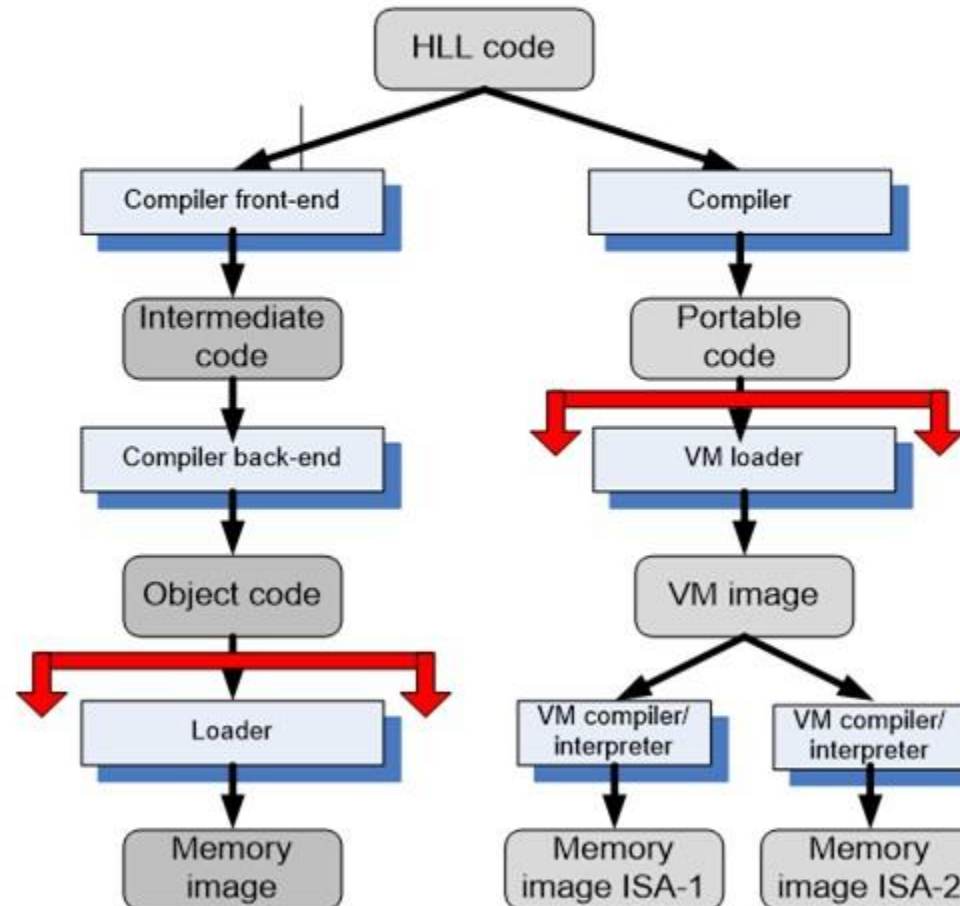
Interfaces

- **Instruction Set Architecture (ISA)** – at the boundary between hardware and software.
- **Application Binary Interface (ABI)** – allows the ensemble consisting of the application and the library modules to access the hardware; the ABI does not include *privileged* system instructions, instead it invokes system calls.
- **Application Program Interface (API)** - defines the set of instructions the hardware was designed to execute and gives the application access to the ISA; it includes high-level language (HLL) library calls which often invoke system calls

Code portability

- Binaries created by a compiler for a specific ISA and a specific operating systems are not portable
- It is possible, though, to compile a HLL program for a virtual machine (VM) environment where portable code is produced and distributed and then converted by binary translators to the ISA of the host system
- A **dynamic binary translation** converts blocks of guest instructions from the portable code to the host instruction and leads to a significant performance improvement, as such blocks are cached and reused

HLL Language Translations



History of Virtualization

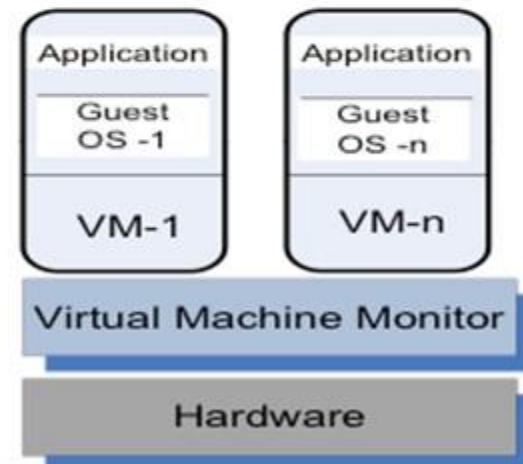
- **1960's, IBM: CP/CMS** control program: a virtual machine operating system for the IBM System/360 Model 67
- **2000, IBM: z-series** with 64-bit virtual address spaces and backward compatible with the System/360
- **1974: Popek and Golberg** from UCLA published "*Formal Requirements for Virtualizable Third Generation Architectures*" where they listed the conditions a computer architecture should satisfy to support virtualization efficiently. The popular x86 architecture that originated in the 1970s did not support these requirements for decades.

History of Virtualization

- **1990's, Stanford researchers, VMware:** Researchers developed a new hypervisor and founded VMware, the biggest virtualization company of today's. First virtualization solution was in 1999 for x86.
- Today many virtualization solutions: Xen from Cambridge, KVM, Hyper-V, ...
- IBM was the first to produce and sell virtualization for the mainframe. But, VMware popularized virtualization for the masses.

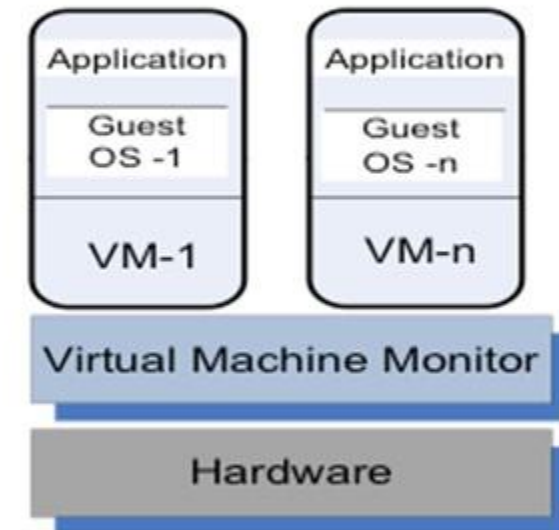
Virtual Machine Monitor (VMM / Hypervisor)

- A **virtual machine monitor (VMM/hypervisor)** partitions the resources of computer system into one or more **virtual machines (VMs)**. Allows several operating systems to run concurrently on a single hardware platform
- A VM is an execution environment that runs an OS
- VM – an isolated environment that appears to be a whole computer, but only has access to a portion of the computer resources



Virtual Machine Monitor (VMM / Hypervisor)

- A VMM allows:
 - Multiple services to share the same platform
 - Live migration - the movement of a server from one platform to another
 - System modification while maintaining backward compatibility with the original system
 - Enforces isolation among the systems, thus security
- A **guest operating system** is an OS that runs in a VM under the control of the VMM.

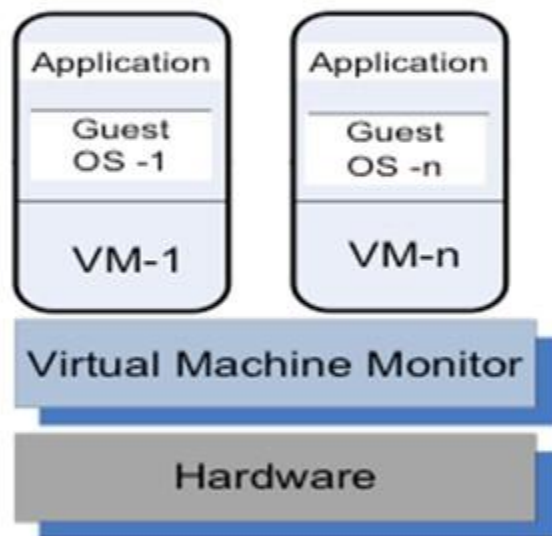


VMM Virtualizes the CPU and the Memory

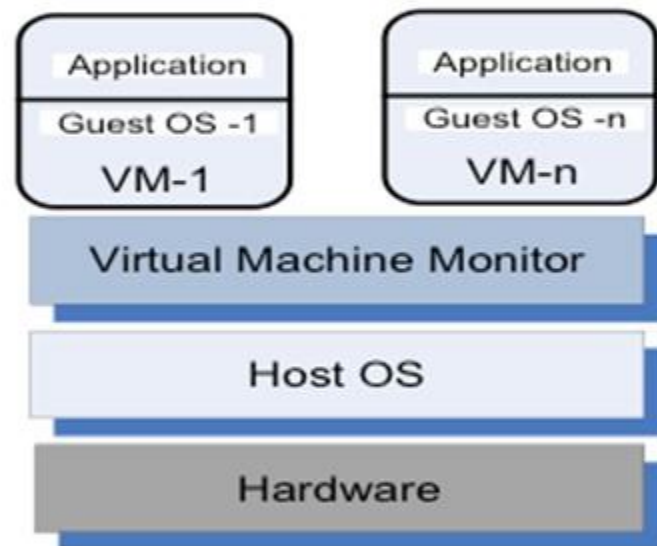
- A VMM (also hypervisor) (how to):
 - Traps the privileged instructions executed by a guest OS and enforces the correctness and safety of the operation
 - Traps interrupts and dispatches them to the individual guest operating systems
 - Controls the virtual memory management
 - Maintains a shadow page table for each guest OS and replicates any modification made by the guest OS in its own shadow page table. This shadow page table points to the actual page frame, and it is used by the Memory Management Unit (MMU) for dynamic address translation.
 - Monitors the system performance and takes corrective actions to avoid performance degradation. For example, the VMM may swap out a VM to avoid thrashing.

Type 1 and 2 Hypervisors

Type 1 Hypervisor



Type 2 Hypervisor



Taxonomy of VMMs:

1. Type 1 Hypervisor (bare metal, native): supports multiple virtual machines and runs directly on the hardware (e.g., VMware ESX, Xen, Denali)
2. Type 2 Hypervisor (hosted) VM - runs under a host operating system (e.g., user-mode Linux)

Examples of Hypervisors

Name	Host ISA	Guest ISA	Host OS	guest OS	Company
Integrity VM	<i>x86-64</i>	<i>x86-64</i>	HP-Unix	Linux, Windows HP Unix	HP
Power VM	Power	Power	No host OS	Linux, AIX	IBM
z/VM	z-ISA	z-ISA	No host OS	Linux on z-ISA	IBM
Lynx Secure	<i>x86</i>	<i>x86</i>	No host OS	Linux, Windows	LinuxWorks
Hyper-V Server	<i>x86-64</i>	<i>x86-64</i>	Windows	Windows	Microsoft
Oracle VM	<i>x86, x86-64</i>	<i>x86, x86-64</i>	No host OS	Linux, Windows	Oracle
RTS Hypervisor	<i>x86</i>	<i>x86</i>	No host OS	Linux, Windows	Real Time Systems
SUN xVM	<i>x86, SPARC</i>	same as host	No host OS	Linux, Windows	SUN
VMware EX Server	<i>x86, x86-64</i>	<i>x86, x86-64</i>	No host OS	Linux, Windows Solaris, FreeBSD	VMware
VMware Fusion	<i>x86, x86-64</i>	<i>x86, x86-64</i>	MAC OS <i>x86</i>	Linux, Windows Solaris, FreeBSD	VMware
VMware Server	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Workstation	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Player	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux Windows	Linux, Windows Solaris, FreeBSD	VMware
Denali	<i>x86</i>	<i>x86</i>	Denali	ILVACO, NetBSD	University of Washington
Xen	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux Solaris	Linux, Solaris NetBSD	University of Cambridge

Performance and Security Isolation

- The run-time behavior of an application is affected by other applications running concurrently on the same platform and competing for CPU cycles, cache, main memory, disk and network access. Thus, it is difficult to predict the completion time!
- Performance isolation - a critical condition for QoS guarantees in shared computing environments
- A VMM is a much simpler and better specified system than a traditional operating system. Example - Xen has approximately 60,000 lines of code; Denali has only about half: 30,000
- The security vulnerability of VMMs is considerably reduced as the systems expose a much smaller number of privileged functions. For example, Xen VMM has 28 hyper calls while Linux has 100s of system calls



Thank You