Cloud Computing

CHAPTER 3

WHAT IS VIRTUALIZATION

Dr. JAMIL S. ALAGHA

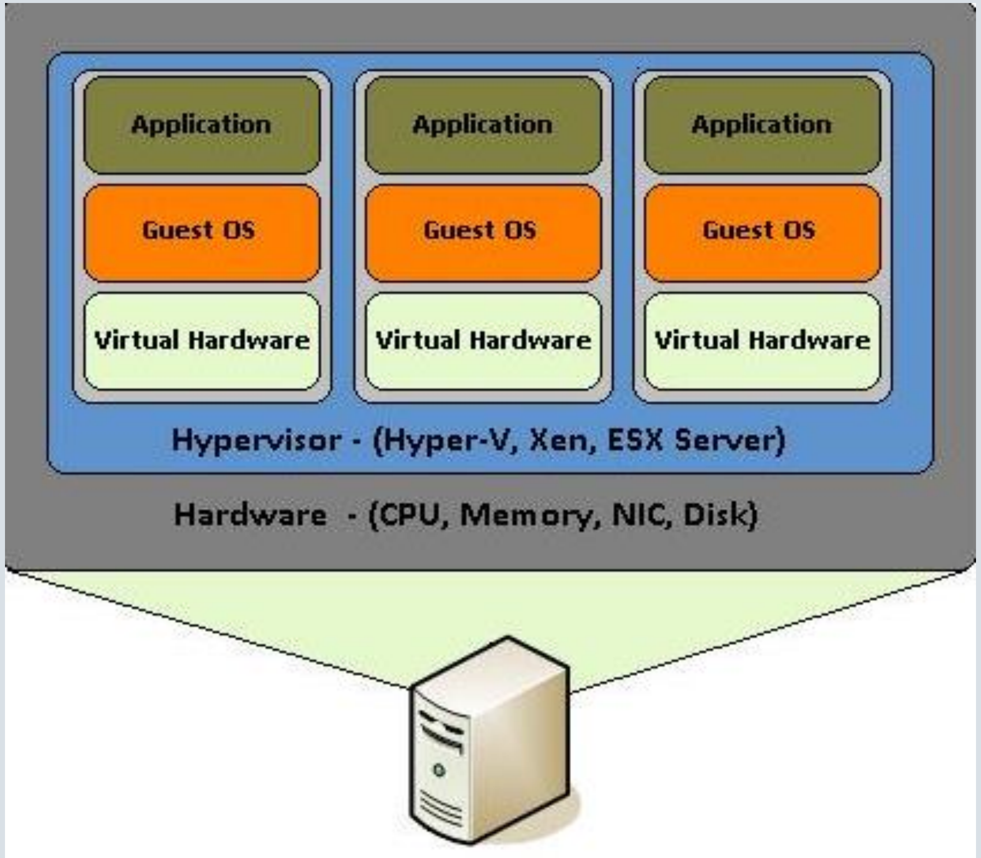
What is Virtualization?

Technique used to enable users to share physical instances of a technology resource

Virtualization creates a logical (or virtual) version of a resource, such as a storage device, server, computing desktop, operating system, or network resources that can be accessed by users transparently

Hardware Virtualization

Virtualization techniques used to create multiple virtual machines on an existing, physical computing platform is called hardware virtualization



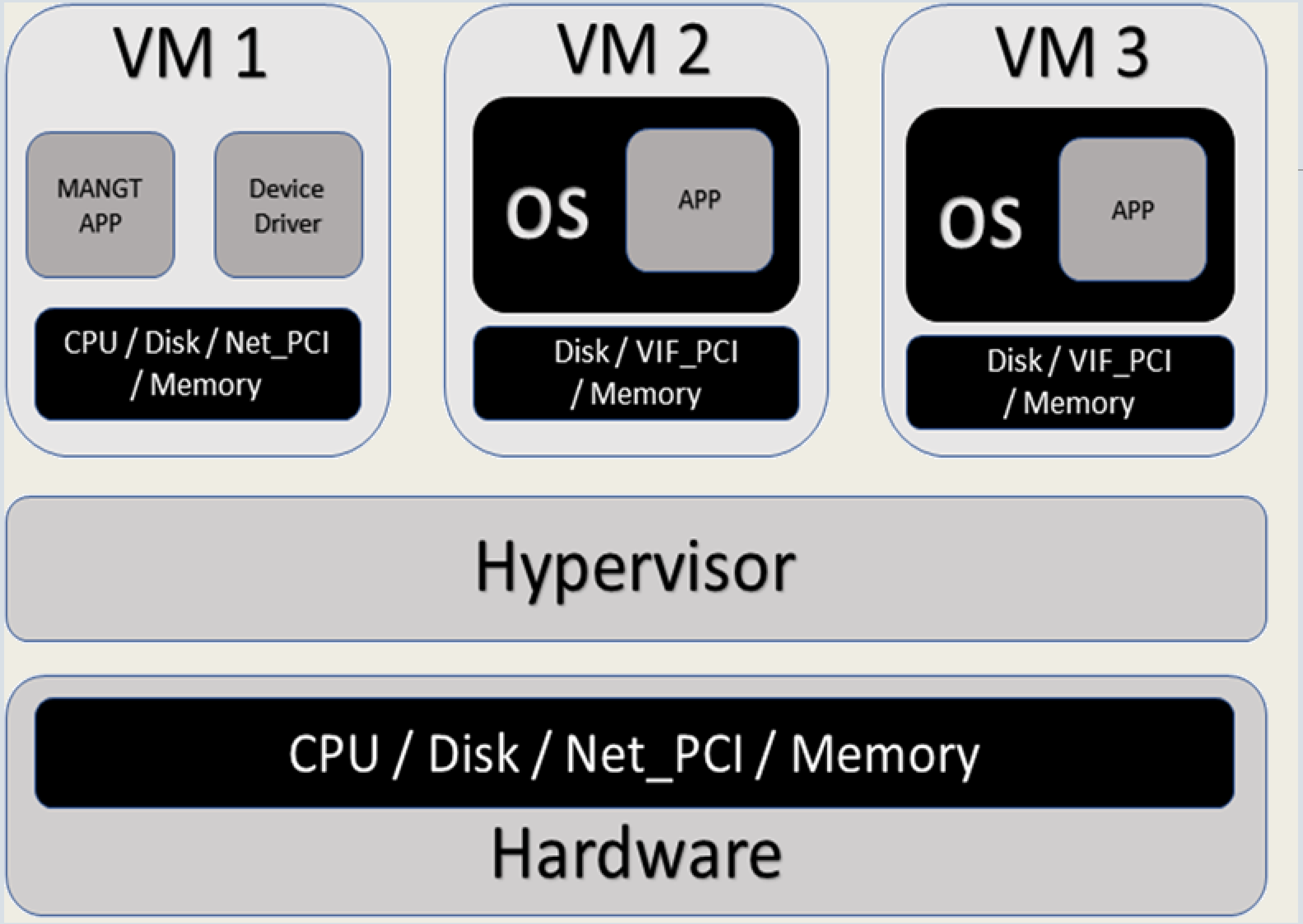
Hypervisor

Software, known as virtual machine manager (VMM) or hypervisor, installs on physical machine and creates virtual machines.

Virtual machines are controlled and monitored by the hypervisor.

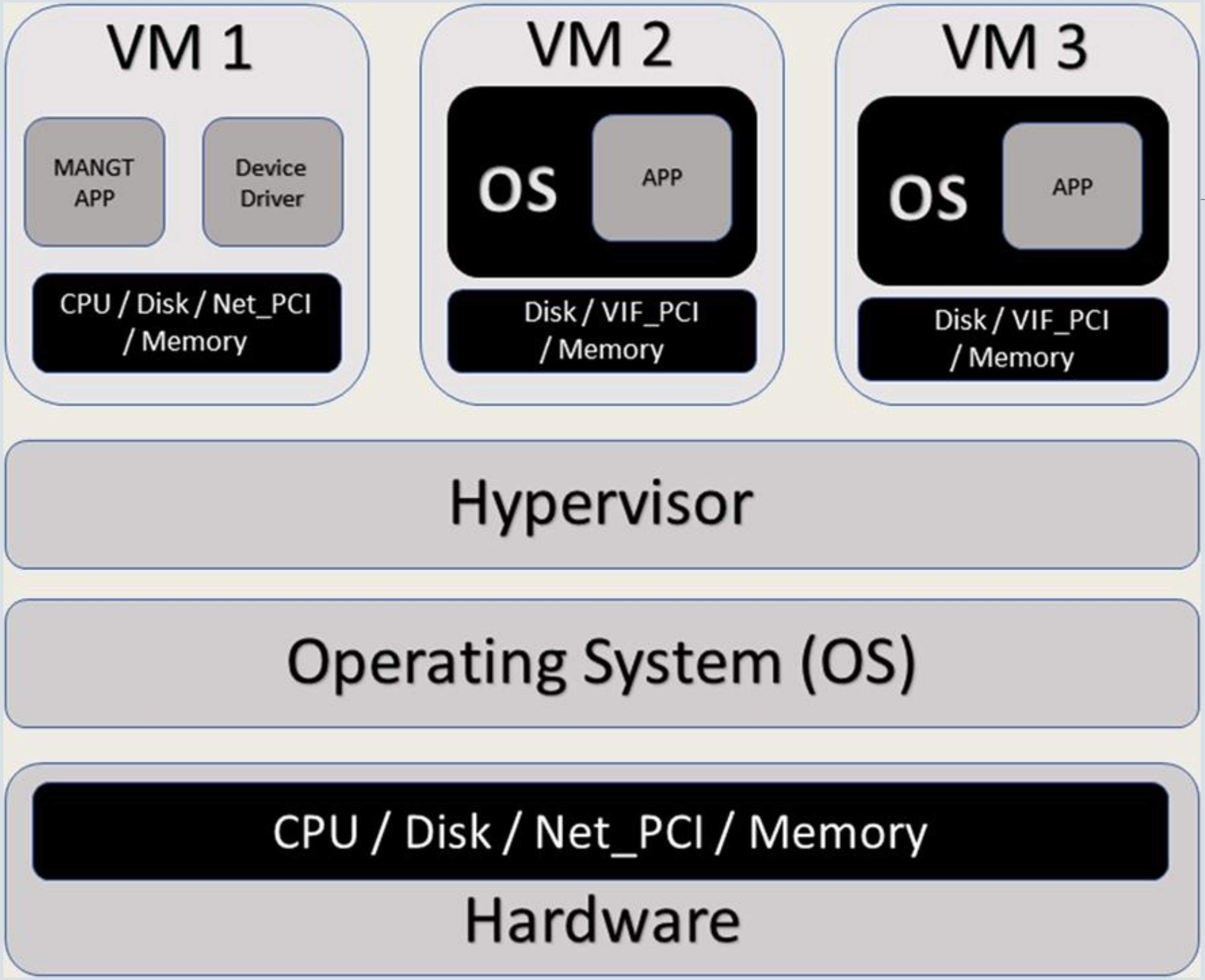
Utilization of the physical hardware (e.g. processor, memory, and other hardware resources) is tracked

Provisioned instances of virtual machines can be loaded with various operating systems and software applications tailored to user needs



Bare Metal Virtualization

Hypervisor installed as operating system directly on computing hardware



Hosted Virtualization

Hypervisor installed as a component of existing operating system on host

Integrated Hypervisors

Implemented through processor command set extensions

Developed by chip manufacturers to enhance speed performance in microchips used as servers

Example extensions include Intel Virtualization Technology (Intel-VT), AMD Virtualization (AMD-V), and VIA virtualization (VIA VT)

Most modern microchips include virtualization extensions

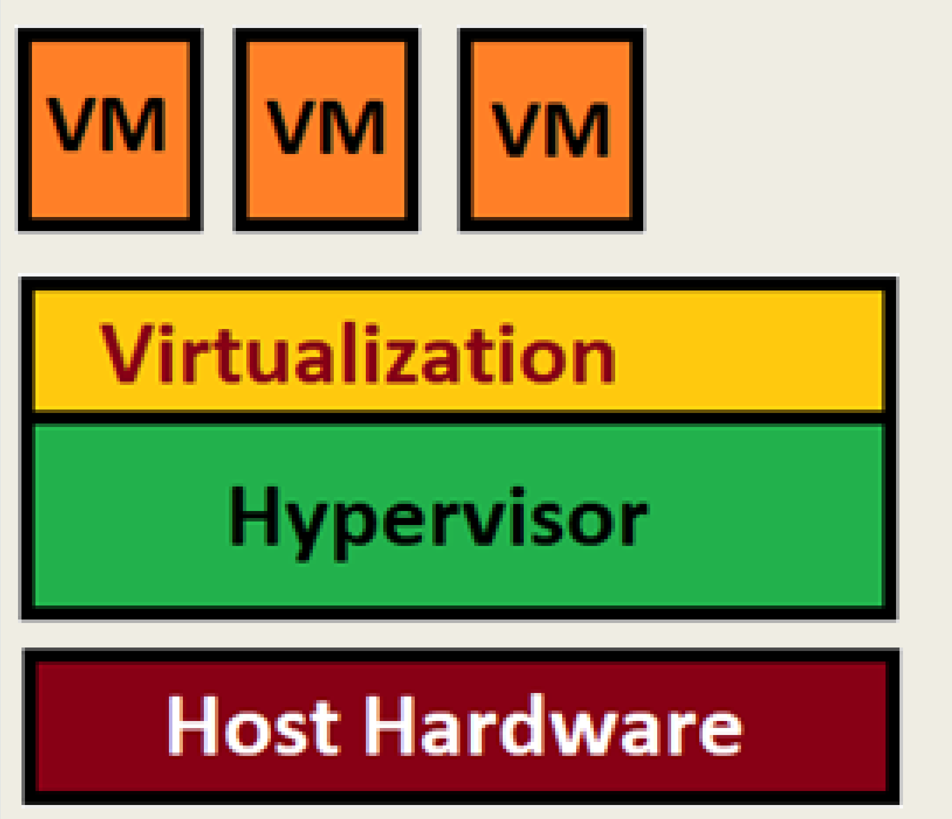
Full Virtualization

Emulates the full set of hardware needed to run a VM a hosted environment

Runs on top of a guest OS

Guest OS runs on hypervisor which runs on host's operating system and hardware platform

Meant to be indistinguishable from hardware platform implementation to user



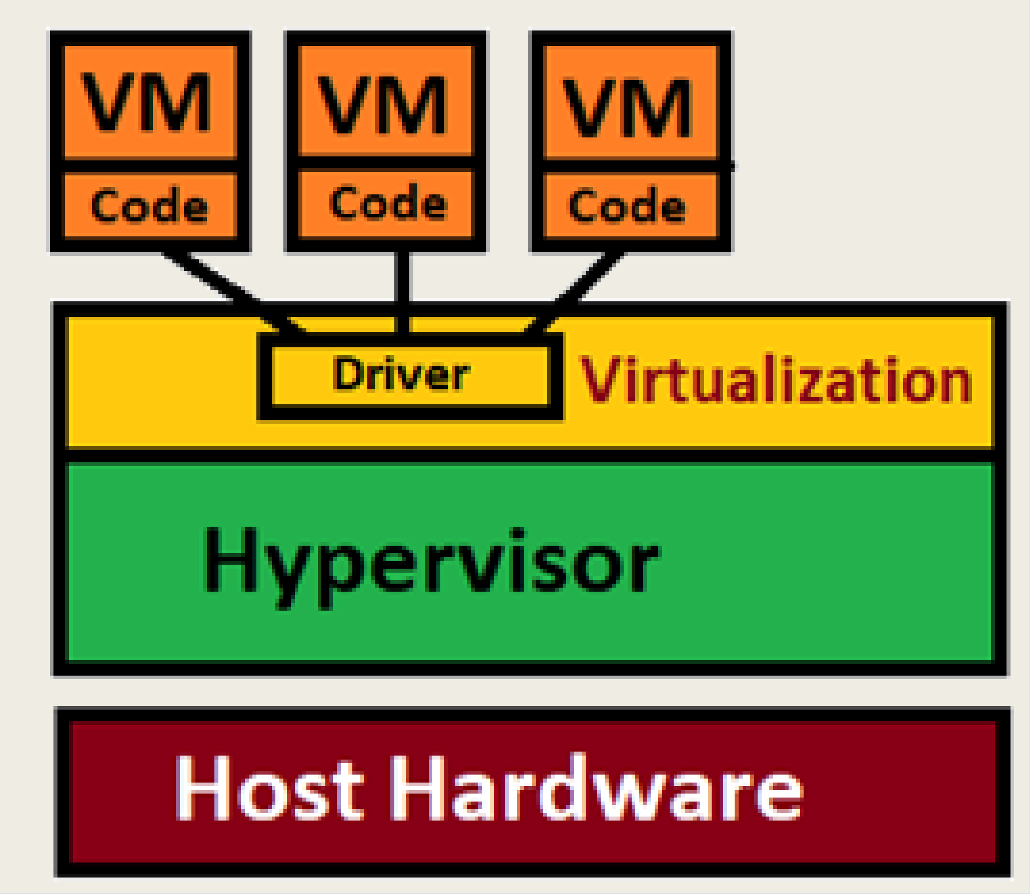
Paravirtualization

Requires guest operating systems to be modified to work on the virtual machines running host.

Hardware is not exactly emulated

Modifies guest machines' operating systems and replaces the existing code with instructions to enable calls to the VMM's APIs.

Calls between guest OSs and the hypervisor are hypercalls.



Hardware-assisted Virtualization

Uses the host computer's existing hardware to build a VM

First used 1970’s by IBM with its System/370 mainframes

Hardware-based solution to speed things up

Original hardware-assisted virtualization techniques obsolete

Influenced development of today’s microprocessors and their extended command sets that provide virtualization options

Graphics Virtualization Technology

Intel GVT-d, GVT-g and GVT-s

Graphics Processing Unit (GPU)

Hardware virtualization revolutionized video game speeds

Made it possible for video related processing to be moved off main CPU

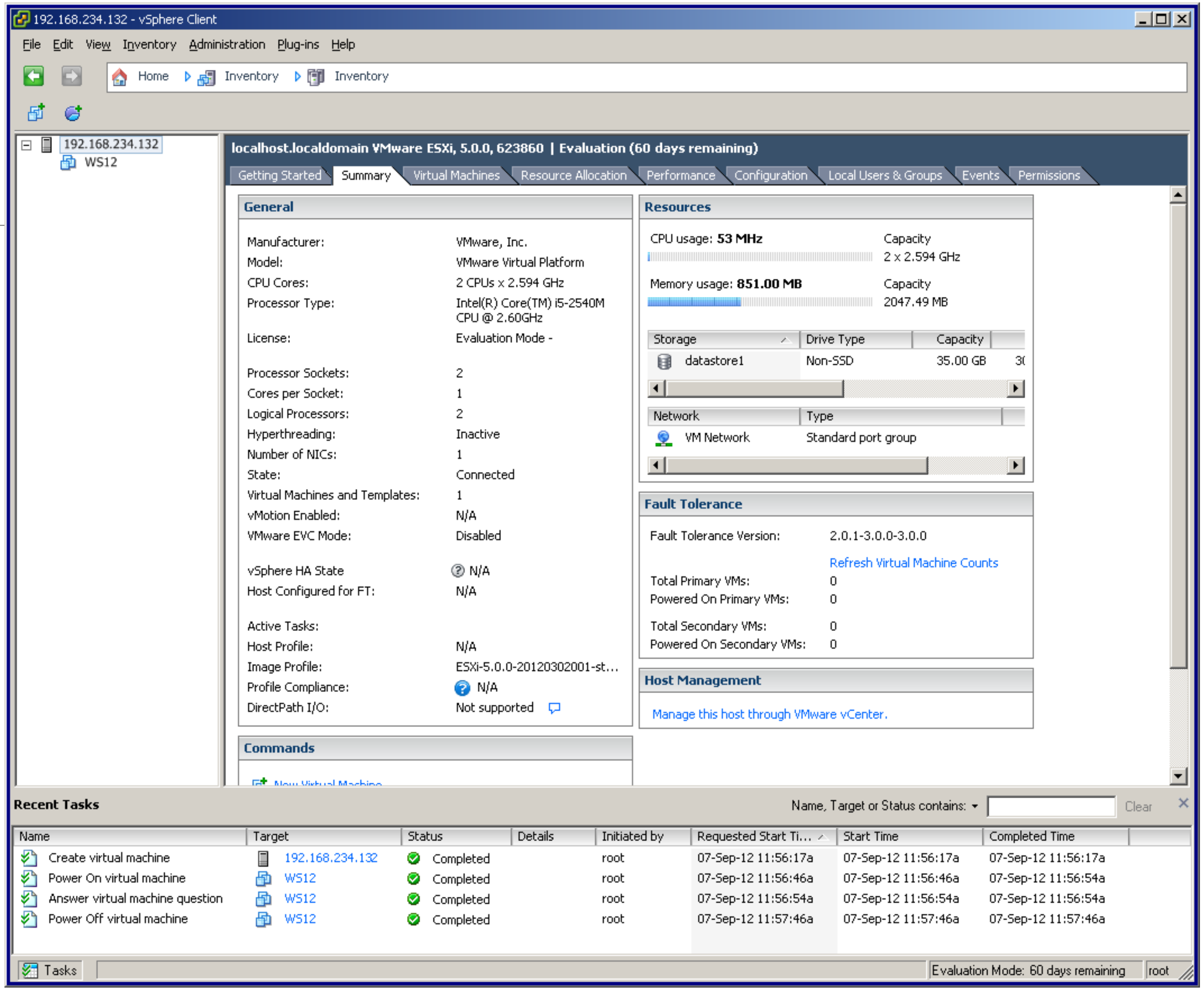
Example Hardware Virtualization Products

VMware ESXi: VMware’s leading hardware virtualization product. Installed on host computer and assumes complete control over the machine’s hardware resources.

Microsoft Hyper-V: Takes advantage of Microsoft’s knowledge of Windows operating systems using x86 architecture microchips. Creates partitions and isolates VMs. Operates in parent and child configuration. Parent partitions have direct access to all system hardware. Parent creates a child using a hypercall API.

Xen: Developed as open source product as part of the Linux kernel.

VMWare ESXi evaluation version screenshot



Hardware Virtualization Benefits

Independence of VM instances

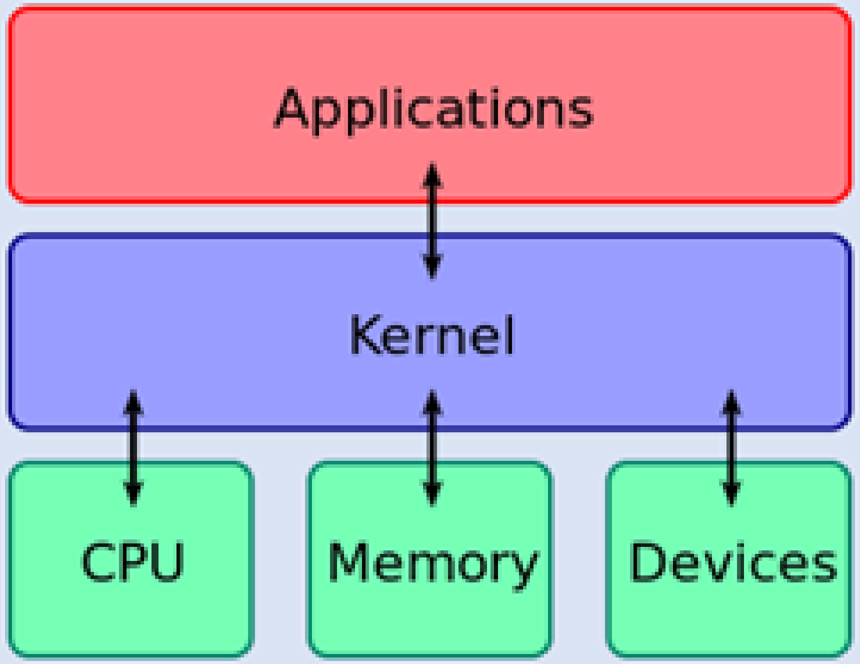
Enhances resource management and improve hardware utilization

Particularly true in cloud environments

Rather than buy multiple servers, an organization can host and manage multiple instances on one piece of hardware.

Key component of IaaS

Kernel: Centerpiece of a computer's operating system. This program controls all other programs. The kernel provides services that enable other programs to request various hardware items central to the computer. It also sends requests to device drivers that control attached hardware. The kernel regulates CPU use and enables multiple programs to use it simultaneously (e.g. multitasking). In everyday language, the kernel is the center of the computer's brain!



Kernel's role in a CPU

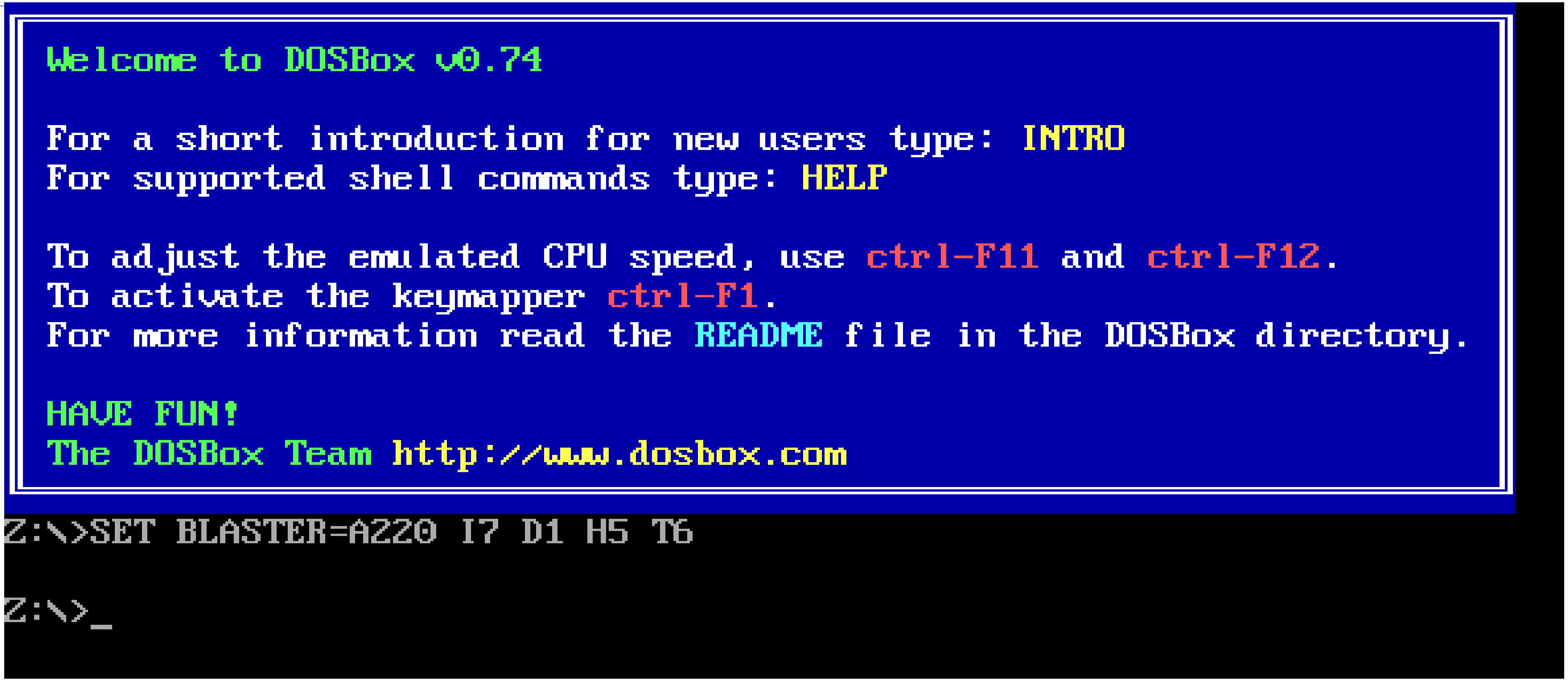
Operating System Virtualization

Describes technique where VM software installed on host computer’s operating system in contrast to bare-metal virtualization

Example: Virtual DOS machine (VDM), allows a virtual 16-bit/32-bit version of DOS to run software using another host Windows-based platform

Permits variety of virtual operating systems to share virtualized hardware resources

DOSBox running on a host computer with a Windows operating system



Operating-System-Level-Virtualization (e.g. Containerization)

Uses host operating system feature where the kernel permits creation of isolated, user-space instances called containers.

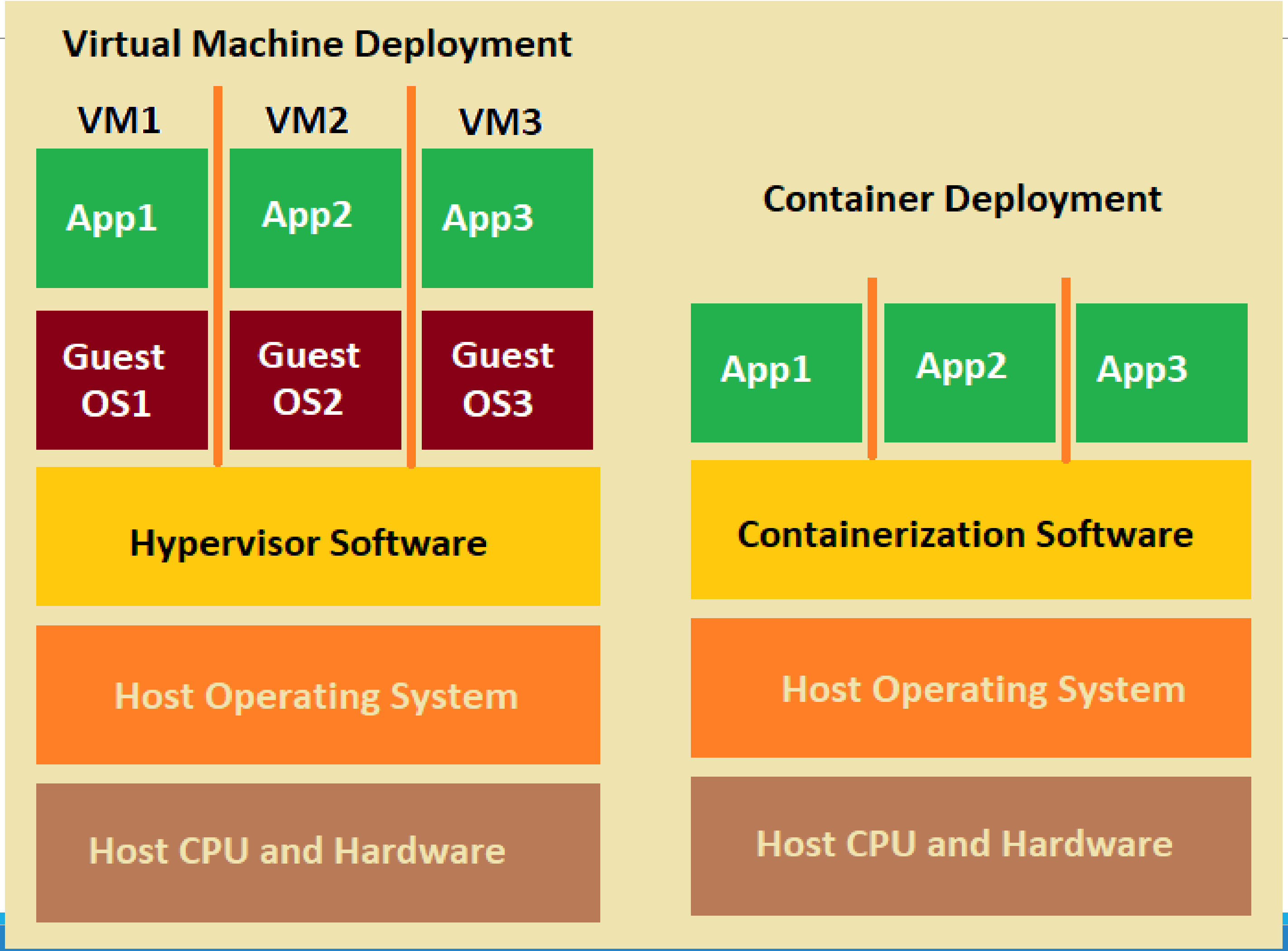
Provides space for applications

Applications do not experience differences from running solo on a physical machine

Applications have access to all resources permitted by container

Kernel controls all allocations.

Containerization Differs from Virtual Machine Deployment



Example Containerization Software

Docker: Widely used to test web servers and web applications. Docker implements a container that runs a web server and application. A second container runs a database server used by the application.

Linux Containers (LXC): System container that occupies space between VMs and application containers. A system container can run an OS but does not emulate the underlying hardware specific to an OS.

Hyper-V and Windows Server containers: Microsoft containers use a small footprint VM to provide application isolation, making it like LXC's system container approach. Application isolation in Windows Server containers uses namespaces, resource control and other tools provided by the server software.

Kubernetes: Although not technically containerization software, it organizes containers into pods located on nodes. Used for many container functions and to help automate, deploy, scale, maintain, and operate application containers. Kubernetes is considered an orchestration tool.

Docker's Components

Daemon: Docker uses a persistent process to manage user-defined containers and container objects. This process, the Docker daemon, is called *dockerd*. It listens for requests sent to the Docker API.

Client Program: Docker's client program, also called "docker", is a line interface that permits users to directly interact with the daemon.

Objects: Objects are used to build applications within Docker. A wide range of objects can be used. The primary object classes include containers, images, and services.

- Containers: Docker containers are a standardized, encapsulated environments used to run applications. A container instance is managed using the Docker API.
- Images: These are read-only templates used to build the contents within a container. Images are used to deploy applications after testing and debugging usually in multiple locations.
- Services: These permit scaling containers across multiple Docker daemons. A swarm of cooperating daemons are created, and these daemons communicate through Docker's API.

Registries: Docker Registries are Docker's repository for retaining images. Clients connect to registries to allow images to be downloaded (or uploaded). Docker offers both public and private registries.

Hub: Docker Hub is the default public registry.

Cloud: Docker Cloud is a native solution used to deploy, monitor, and manage Dockerized applications.

Containers versus VMs

Containers take less memory than equivalent VM.

VMs emulate a hardware system so multiple OSs can run in independent environments on same physical machine.

Containers share the hosts' OSs.

VMs take more time to run and create but have added flexibility of permitting multiple OSs from various vendors.

VMs can run multiple versions of OSs, including older ones.

Web application may be used on a variety of client machines on various browsers being run on various OSs. In this case, a VM makes more sense.

If a homogenous environment used for an in-house software project, containers are better.

Pros and Cons of Containers

Benefits of containers

- Can be more efficient than VMs since they share the host OS kernel.
- More portable due to encapsulation of the application operating code.
- No guest OS environment variables or library dependencies need to be managed.
- More efficient use of memory, CPU, and storage.
- More containers can run on same infrastructure as VMs.
- Consistent through entire application lifecycle (particularly in agile development).
- Better for distributed applications.

Disadvantages of containers

- Lack of isolation from the host OS.
- Security threats may be greater than for VMs
- Lack of OS flexibility. Each container generally uses same OS as host
- More difficult to monitor activity on containers since many more can simultaneously exist.
- Large number of vendors and orchestrators in this field make the technology complicated.

Containers work well with Microservices

Microservices

Microservice architecture describes a software development approach where an application is broken into smaller, specialized parts that communicate across REST-based interfaces or APIs. In general, microservices manage their own data, authenticate and maintain logs. Microservices are ideal for container environments.

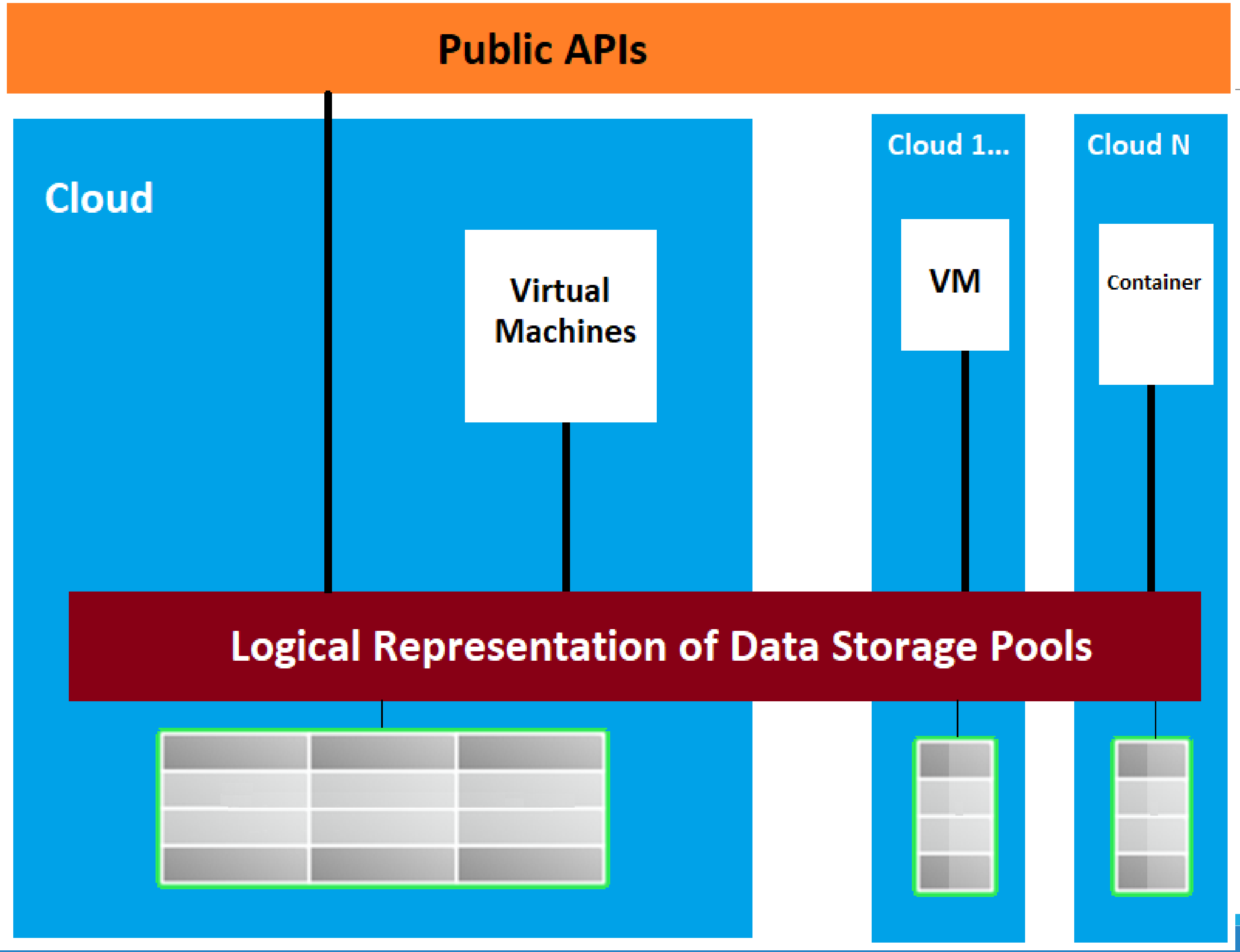
Containers as a Service (CaaS)

Cloud-based service model where containers and related applications can be uploaded, organized, and managed to facilitate use of this technology.

Storage Virtualization

- Form of hardware virtualization
- Implemented using combination of specialty hardware (appliances) and software applications
- Goals are to:
  - (1) enable and enhance storage management in environments where not all hardware is the same (e.g. heterogeneous)
  - (2) provide managed system for equipment replacement and downtime reductions
  - (3) offer better storage utilization

Separation of Physical Storage from User



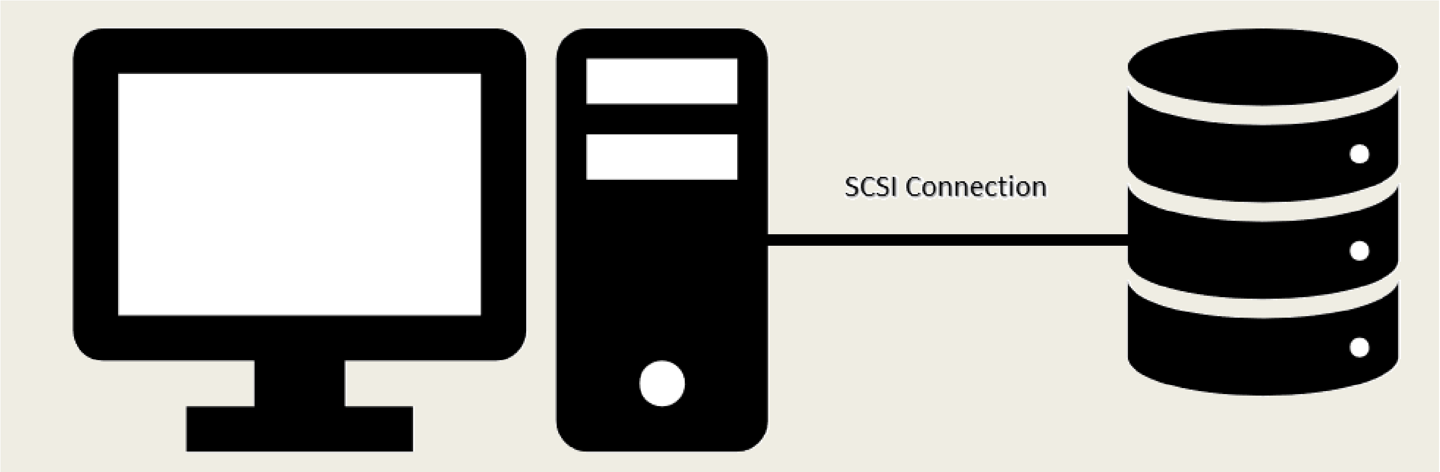
Direct Attached Storage (DAS)

Storage devices directly attached to one computer

Not meant to be networked

Used exclusively by one computer

Does not make good cloud storage



SCSI: A trustworthy interface often used with DAS

In the days when storage was costly and slow, most application servers had internal storage devices. These storage devices used a protocol (e.g. convention for communicating) known as Small Computer System Interface or SCSI. SCSI is a standard to ensure devices can speak the 'same language'. Many hard drives, CDs, DVDs, and other storage devices can be attached using SCSI. SCSI interfaces require use of a microprocessor-controlled, smart bus, that permits peripheral hardware devices to be daisy-chained to a computer. In addition to storage devices, scanners, printers, and other items use SCSI. In the desktop and laptop markets, SCSI is losing out to USB and FireWire interfaces. In the server arena, SCSI remains popular, particularly for building in redundancy. For instance, several hard drives can operate on a single server in a RAID (Redundant Array of Independent Disks) configuration. If a drive fails, a new one can be added without losing data while the system operates (at least from the user's perspective) as normal.

Storage Area Network (SAN)

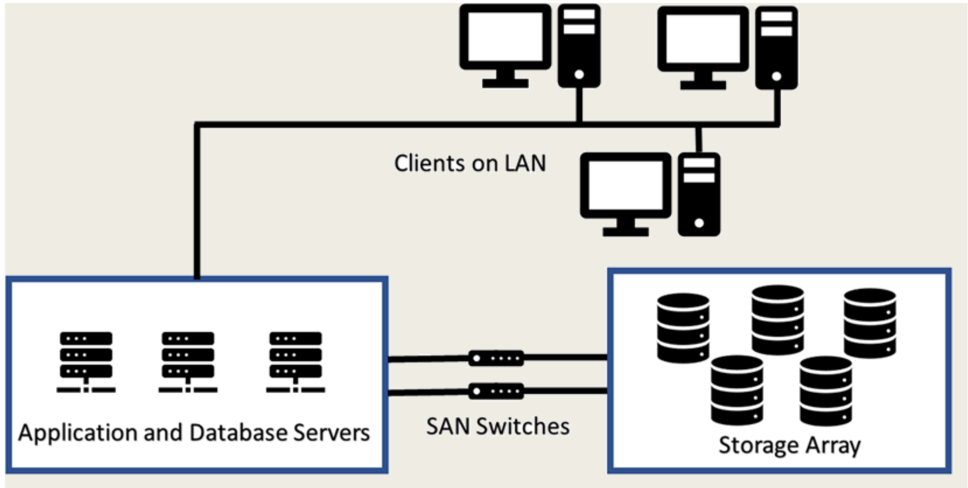
Enable users to share storage arrays in flexible and scalable ways

Ensure data redundancy

Share storage among application servers

Enhance performance

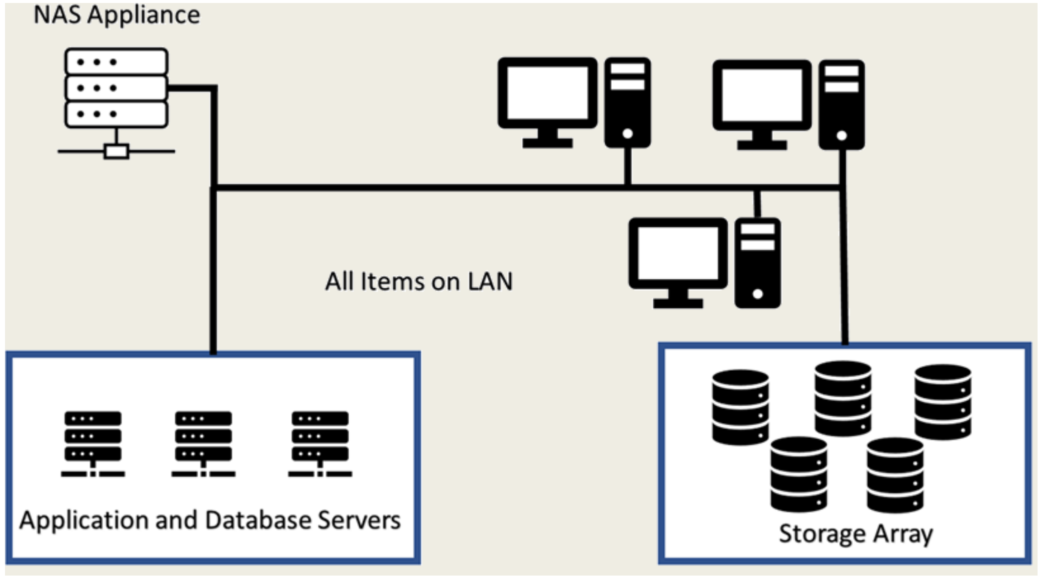
Require specialized staff members due to their complexity



Network Attached Storage (NAS)

Uses dedicated piece of hardware, called an appliance, to add commonly accessible storage to a network.

NAS appliance can help manage file requests, provide additional security, and offer methods for configuring and overseeing a shared storage system



NAS Head

In some circumstances, NAS and SAN technologies can be synergistically combined. One way of doing this is through a NAS head device. A NAS head is a protocol handler gateway specifically engineered to interface with a SAN. A NAS head does not have any client storage availability, instead it provides a NAS-style interface that, from the end-user's perspective, makes storage appear to occur at the file-level. The NAS head offers the load-balancing and protocol handling needed to transparently change file-level access to more efficient, block-level format access. From the user's perspective, it looks like a giant NAS storage device. At the same time, IT specialists see the block-level data stored on the SAN so all their SAN management tools can be used to ensure back-up, recovery, storage efficiencies and other high-end features offered by a SAN. For an SME (or more likely larger businesses), it may be the best of both worlds. In many ways, this is a private cloud.

SAN Virtualization

Applied at various levels within a SAN and to different storage functions (e.g. physical storage, RAID groupings, logical units and so forth).

Can be broken into four layers:

1. Physical storage devices
2. Block aggregation layer
3. File/record layer
4. Application layer

NAS Virtualization

Like creation of a private cloud for storage

Software products exist to enable an existing NAS device to abstract stored files from their physical location to an online virtual location

Offers same benefits as a private cloud

- no third party is involved
- data remains private on the organization's own storage devices behind a firewall
- virtualized NAS devices can be attached through virtual private networks (VPNs) for added security

File versus Block Level Virtualization

SANs (and DAS) use block-level virtualization and NAS systems use file level storage.

Block Level

- Block-level storage virtualization adds a logical layer to the SAN which can be thought to sit on top of the storage arrays
- Servers using the SAN are directed to virtualized logical unit numbers (LUNs)
- Virtualization at this level causes separate physical devices to be logically ‘regrouped’ using LUNs

File Level

- Usually done with NAS devices, virtualizes link between the files and physical storage location
- Logical drives, composed of various physical storage areas, seen by the users
- Physical locations remained masked
- Enables location independence
- Enhance users’ capability to move and access files

Chapter 3 Summary

Virtualization is a primary characteristic of cloud computing

Hardware virtualization, operating system virtualization, operating-system-level virtualization (containerization), and storage virtualization all examples of approaches to add a logical layer that redefines, regroups, and consolidates resources in ways that add to efficiency, management capabilities, and expands usage possibilities

Virtualization ensures resources are shared more effectively