


CMR Institute of Technology, Bangalore			
Department(s): Computer Science and Engineering			
Semester: 6			
Subject: Cloud Computing		Code: BCS601	

VTU Solution
June/July 2025

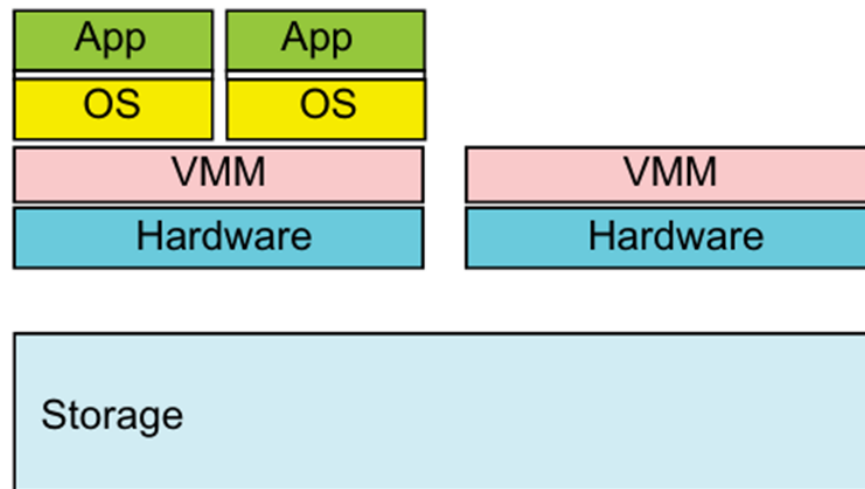
MODULE - 1																																
1	a.	<div>Explain critical cluster design issues and feasible implementation</div> <div>SOLUTION</div> <div>A cluster-wide OS for complete resource sharing is not available yet. Middleware or OS extensions were developed at the user space to achieve SSI at selected functional levels. Without this middleware, cluster nodes cannot work together effectively to achieve cooperative computing. The software environments and applications must rely on the middleware to achieve high performance. The cluster benefits come from scalable performance, efficient message passing, high system availability, seamless fault tolerance, and cluster-wide job management.</div> <div><table><tr><th colspan="3">Table 1.3 Critical Cluster Design Issues and Feasible Implementations</th></tr><tr><th>Features</th><th>Functional Characterization</th><th>Feasible Implementations</th></tr><tr><td>Availability and Support</td><td>Hardware and software support for sustained HA in cluster</td><td>Failover, fallback, check pointing, rollback recovery, nonstop OS, etc.</td></tr><tr><td>Hardware Fault Tolerance</td><td>Automated failure management to eliminate all single points of failure</td><td>Component redundancy, hot swapping, RAID, multiple power supplies, etc.</td></tr><tr><td>Single System Image (SSI)</td><td>Achieving SSI at functional level with hardware and software support, middleware, or OS extensions</td><td>Hardware mechanisms or middleware support to achieve DSM at coherent cache level</td></tr><tr><td>Efficient Communications</td><td>To reduce message-passing system overhead and hide latencies</td><td>Fast message passing, active messages, enhanced MPI library, etc.</td></tr><tr><td>Cluster-wide Job Management</td><td>Using a global job management system with better scheduling and monitoring</td><td>Application of single-job management systems such as LSF, Codine, etc.</td></tr><tr><td>Dynamic Load Balancing</td><td>Balancing the workload of all processing nodes along with failure recovery</td><td>Workload monitoring, process migration, job replication and gang scheduling, etc.</td></tr><tr><td>Scalability and Programmability</td><td>Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases</td><td>Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools</td></tr></table></div>	Table 1.3 Critical Cluster Design Issues and Feasible Implementations			Features	Functional Characterization	Feasible Implementations	Availability and Support	Hardware and software support for sustained HA in cluster	Failover, fallback, check pointing, rollback recovery, nonstop OS, etc.	Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.	Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level	Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing, active messages, enhanced MPI library, etc.	Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.	Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.	Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools	8	L 2	CO2
Table 1.3 Critical Cluster Design Issues and Feasible Implementations																																
Features	Functional Characterization	Feasible Implementations																														
Availability and Support	Hardware and software support for sustained HA in cluster	Failover, fallback, check pointing, rollback recovery, nonstop OS, etc.																														
Hardware Fault Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, multiple power supplies, etc.																														
Single System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions	Hardware mechanisms or middleware support to achieve DSM at coherent cache level																														
Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing, active messages, enhanced MPI library, etc.																														
Cluster-wide Job Management	Using a global job management system with better scheduling and monitoring	Application of single-job management systems such as LSF, Codine, etc.																														
Dynamic Load Balancing	Balancing the workload of all processing nodes along with failure recovery	Workload monitoring, process migration, job replication and gang scheduling, etc.																														
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a grid as the workload or data set increases	Use of scalable interconnect, performance monitoring, distributed execution environment, and better software tools																														
	b.	Describe VM Primitive operations	6	L	CO2																											

SOLUTION

Virtual Machine – Primitive Operations

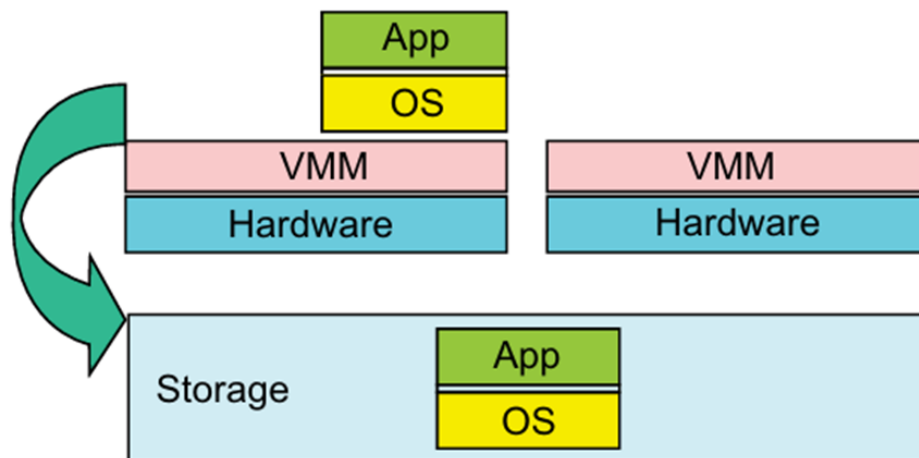
2

•**Multiplexing:** VMs can be multiplexed between hardware machines. Multiple VMs can be run on a single physical machine.



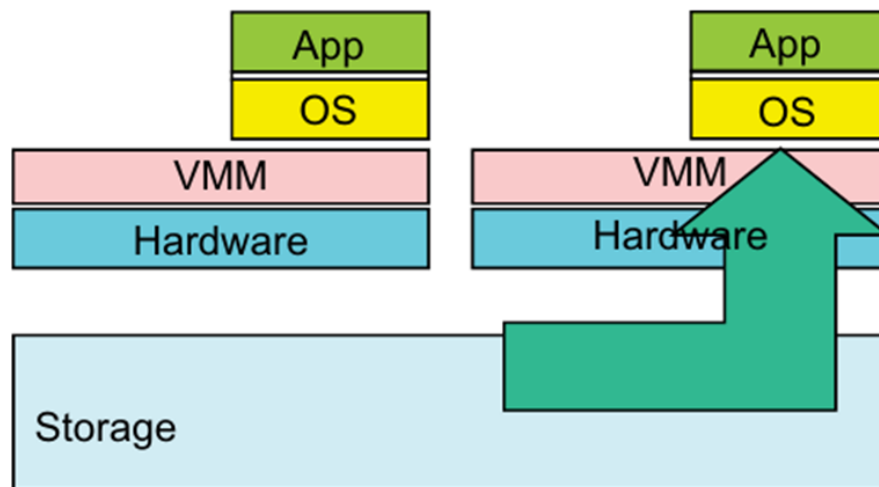
(a) Multiplexing

•**Suspending:** VM can be suspended (paused) and stored in the stable storage.



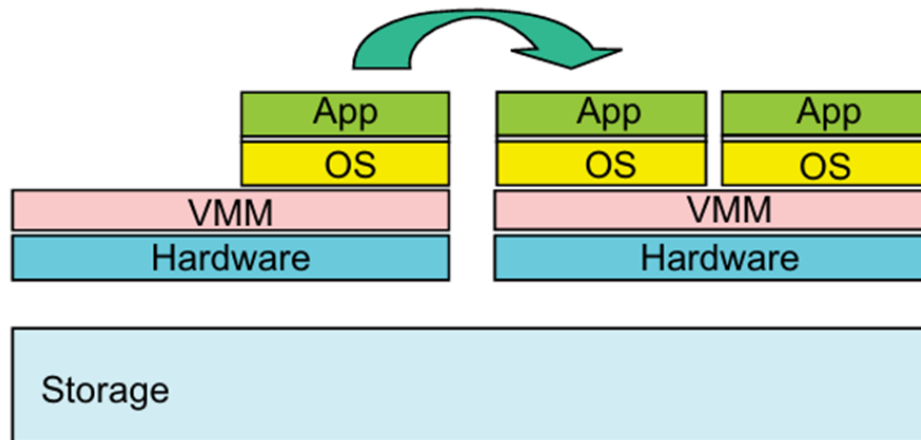
(b) Suspension (storage)

•**Resuming:** Suspended VM can be resumed or provisioned to a new hardware platform or the same machine.



(c) Provision (resume)

•**Migration:** A VM can be migrated from one platform to another.



(d) Live migration

c. Explain virtual machine with architectures of compared with traditional physical machine.

SOLUTION

- A **virtual machine** is a software-based computer that runs inside a real computer.
- It acts like a real machine but is actually just a program that emulates the computer program.

6 L
2

CO1

		<p>FIGURE 1.12</p> <ul style="list-style-type: none"> •Host machine is equipped with the physical hardware. •VM can be provided for any hardware system. •VM is built with virtual resources managed by guest OS. •Between VM and host platform, one needs to deploy a middleware called VMM (Virtual Machine Monitor). 			
OR					
2	a.	<p>Explain the following:</p> <p>i) Internet of Things</p> <p>ii) Cyber physical systems</p> <p>iii) Memory storage and wide area networking</p> <p>SOLUTION</p> <p>Internet of Things:</p> <p>The concept of the IoT was introduced in 1999 at MIT. The IoT refers to the networked interconnection of everyday objects, tools, devices, or computers. One can view the IoT as a wireless network of sensors that interconnect all things in our daily life. These things can be large or small and they vary with respect to time and place. The idea is to tag every object using RFID or a related sensor or electronic technology such as GPS. In the IoT era, all objects and devices are instrumented, interconnected, and interacted with each other intelligently. This communication can be made between people and things or among the things themselves. Three communication patterns co-exist: namely H2H (human-to-human), H2T</p>	2 2 6	L 1	CO1

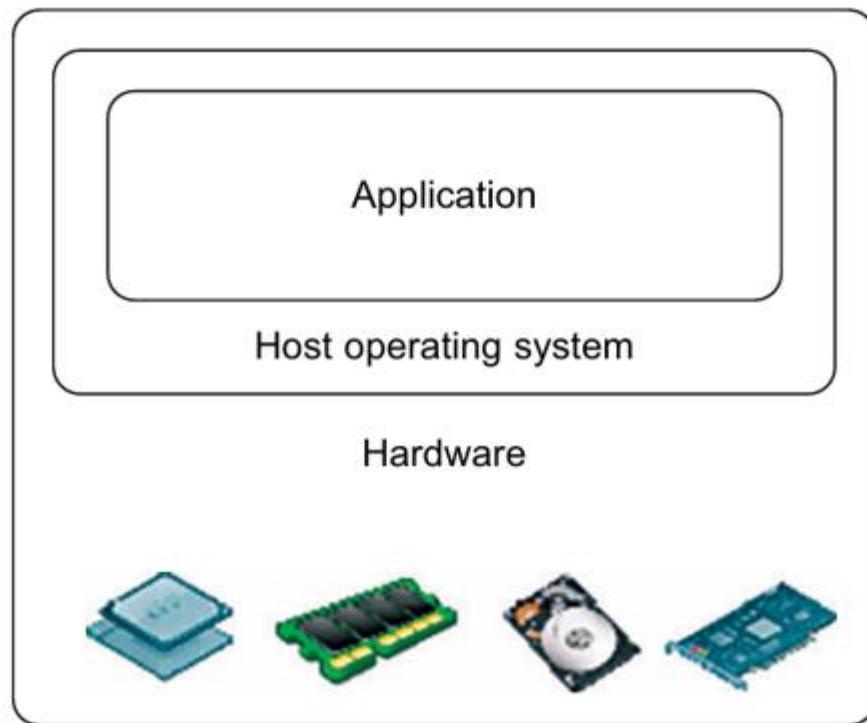
	<p>(human-to thing), and T2T (thing-to-thing). Here things include machines such as PCs and mobile phones. The idea here is to connect things (including human and machine objects) at any time and any place intelligently with low cost. Any place connections include at the PC, indoor (away from PC), outdoors, and on the move. Any time connections include daytime, night, outdoors and indoors, and on the move as well.</p> <p>Cyber Physical Systems:</p> <p>A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world. A CPS integrates “cyber” (heterogeneous, asynchronous) with “physical” (concurrent and information-dense) objects. A CPS merges the “3C” technologies of computation, communication, and control into an intelligent closed feedback system between the physical world and the information world, a concept which is actively explored in the United States. The IoT emphasizes various networking connections among physical objects, while the CPS emphasizes exploration of virtual reality (VR) applications in the physical world. We may transform how we interact with the physical world just like the Internet transformed how we interact with the virtual world.</p>			
b.	<p>Explain computing paradigm distinctions</p> <p>SOLUTION</p> <p>The high-technology community has argued for many years about the precise definitions of centralized computing, parallel computing, distributed computing, and cloud computing. The following list defines these terms more clearly; their architectural and operational differences are discussed further in subsequent chapters.</p> <ul style="list-style-type: none"> • Centralized computing This is a computing paradigm by which all computer resources are centralized in one physical system. All resources (processors, memory, and storage) are fully shared and tightly coupled within one integrated OS. Many data centers and supercomputers are centralized systems, but they are used in parallel, distributed, and cloud computing applications. • Parallel computing In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Some authors refer to this discipline as parallel processing. Interprocessor communication is accomplished through shared memory or via message passing. A computer system capable of parallel computing is commonly known as a parallel computer. Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often referred to as parallel programming. 	5	L 2	CO2

		<ul style="list-style-type: none"> • Distributed computing This is a field of computer science/engineering that studies distributed systems. A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through message passing. A computer program that runs in a distributed system is known as a distributed program. The process of writing distributed programs is referred to as distributed programming. • Cloud computing An Internet cloud of resources can be either a centralized or a distributed computing system. The cloud applies parallel or distributed computing, or both. Clouds can be built with physical or virtualized resources over large data centers that are centralized or distributed. Some authors consider cloud computing to be a form of utility computing or service computing. As an alternative to the preceding terms, some in the high-tech community prefer the term concurrent computing or concurrent programming. <p>These terms typically refer to the union of parallel computing and distributing computing, although biased practitioners may interpret them differently. Ubiquitous computing refers to computing with pervasive devices at any place and time using wired or wireless communication. The Internet of Things (IoT) is a networked connection of everyday objects including computers, sensors, humans, etc. The IoT is supported by Internet clouds to achieve ubiquitous computing with any object at any place and time. Finally, the term Internet computing is even broader and covers all computing paradigms over the Internet. This book covers all the aforementioned computing paradigms, placing more emphasis on distributed and cloud computing and their working systems, including the clusters, grids, P2P, and cloud systems.</p>			
	c.	Describe the classification of parallel and distributed computing systems <p>In parallel computing, all processors are either tightly coupled with centralized shared memory or loosely coupled with distributed memory. Some authors refer to this discipline as parallel processing. Interprocessor communication is accomplished through shared memory or via message passing. A computer system capable of parallel computing is commonly known as a parallel computer. Programs running in a parallel computer are called parallel programs. The process of writing parallel programs is often referred to as parallel programming</p> <p>Distributed Computing This is a field of computer science/engineering that studies distributed systems. A distributed system consists of multiple autonomous computers, each having its own private memory, communicating through a computer network. Information exchange in a distributed system is accomplished through message passing. A computer program that runs in a distributed system is known as a distributed program. The process of writing distributed programs is referred to as distributed programming.</p>	5	L 2	CO2
MODULE -2					
3	a.	Explain implementation levels of virtualization	5	L	CO2

		<p>SOLUTION</p> <p>Virtualization can be implemented in following levels:</p> <p>•Instruction-set Architecture Level (ISA):</p> <ul style="list-style-type: none"> •Virtualization is performed by emulating a given ISA, by the ISA of the host machine. •Basic emulation happens through code interpretation. •Interpreter program interprets source instructions to target instructions. •One source instructions may require tens and hundreds of target instructions to perform its functions. •Process is relatively slow. •Dynamic binary translation is needed for runtime translation. •Translates basic blocks of dynamic source instructions to target instructions. •A virtual ISA, requires adding a processor specific software translation layer to the compiler. •MIPS -> X86 •Android ->PC ->bluestacks <p>•Hardware level</p> <ul style="list-style-type: none"> •Performed on the top of hardware •Provides a virtual hardware environment for VM. •Virtualizes memory, processors and I/O devices. •To upgrade hardware utilization rate by multiple users concurrently. •IBM VM/370 •Xen Hypervisor <p>•Operating System Level</p> <ul style="list-style-type: none"> •Abstraction layer between traditional OS and user applications. •Creates isolated containers on a single physical server and OS instances to utilize software and hardware. 	2	
--	--	--	---	--

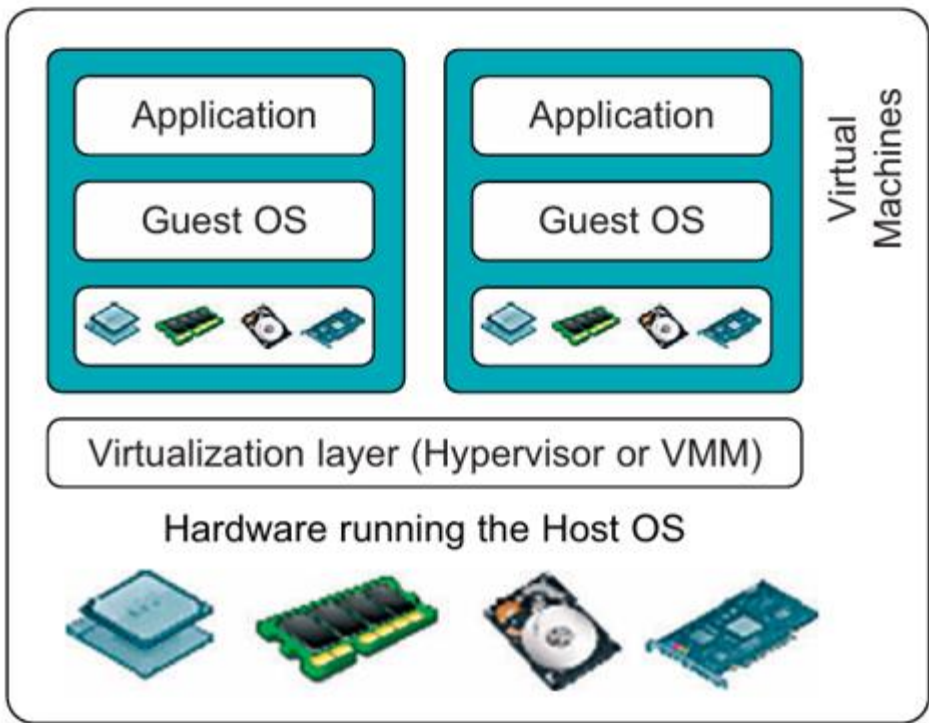
		<ul style="list-style-type: none"> •Ex: Dockers •Containers behave like a real servers. •Commonly used in creating a virtual environment to allocate hardware resources among multiple users. <p>•Library Support Level</p> <ul style="list-style-type: none"> •Controlling the communication link between applications and rest of system using APIs. •WINE supports windows applications on top of UNIX hosts. •vCUDA allows applications executing within VMs to leverage GPU acceleration. <p>•Application Level</p> <ul style="list-style-type: none"> •Virtualizes an application as VM. •Also known as process-level virtualization •JVM, .Net •Application isolation, sandboxing, streaming 		
--	--	---	--	--

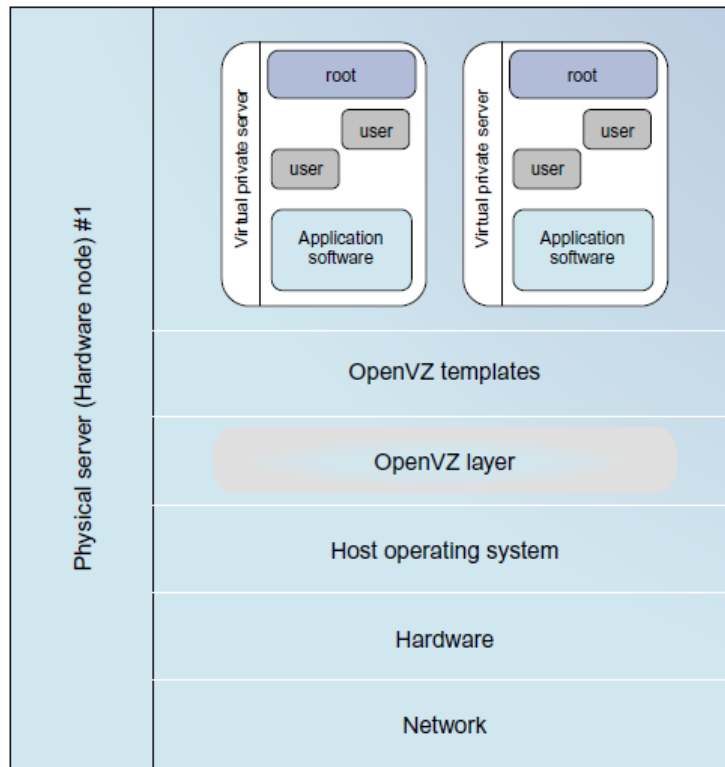
		<div>Application level</div> <div>JVM / .NET CLR / Panot</div> <div>Library (user-level API) level</div> <div>WINE/ WABI/ LxRun / Visual MainWin / vCUDA</div> <div>Operating system level</div> <div>Jail / Virtual Environment / Ensim's VPS / FVM</div> <div>Hardware abstraction layer (HAL) level</div> <div>VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux</div> <div>Instruction set architecture (ISA) level</div> <div>Bochs / Crusoe / QEMU / BIRD / Dynamo</div>			
	b.	Draw architecture of computer before and after virtualization. SOLUTION Before Virtualization	5	L 3	CO2



(a) Traditional computer

After Virtualization

		 <p>(b) After virtualization</p>			
c.	<p>Explain how virtualization support at OS level</p> <p>SOLUTION</p> <p>Operating System Level This refers to an abstraction layer between the traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users. It is also used, to a lesser extent, in consolidating server hardware by moving services on separate hosts into containers or VMs on one server.</p> <p>Operating system virtualization inserts a virtualization layer inside an operating system to partition a machine's physical resources. It enables multiple isolated VMs within a single operating system kernel. This kind of VM is often called a virtual execution environment (VE), Virtual Private System (VPS), or simply container. From the user's point of view, VEs look like real servers. This means a VE has its own set of processes, file system, user accounts, network interfaces with IP addresses, routing tables, firewall rules, and other personal settings. Although VEs can be customized for different people, they share the same operating system kernel. Therefore, OS-level virtualization is also called single-OS image virtualization</p>	1 0	L 3	CO2	

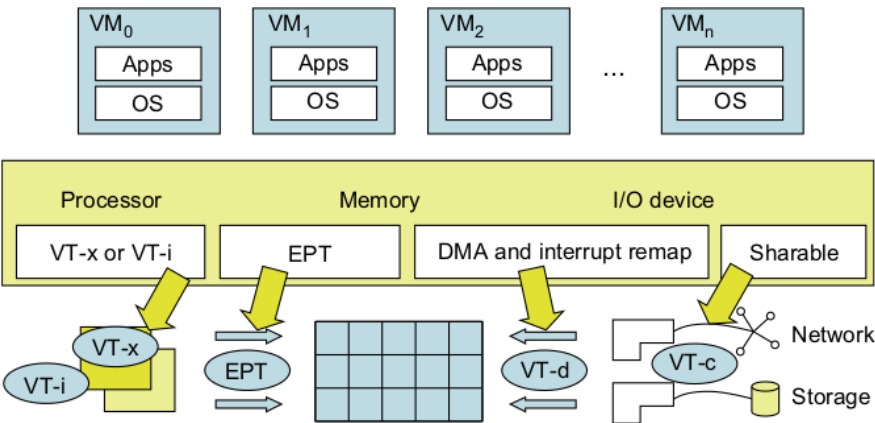


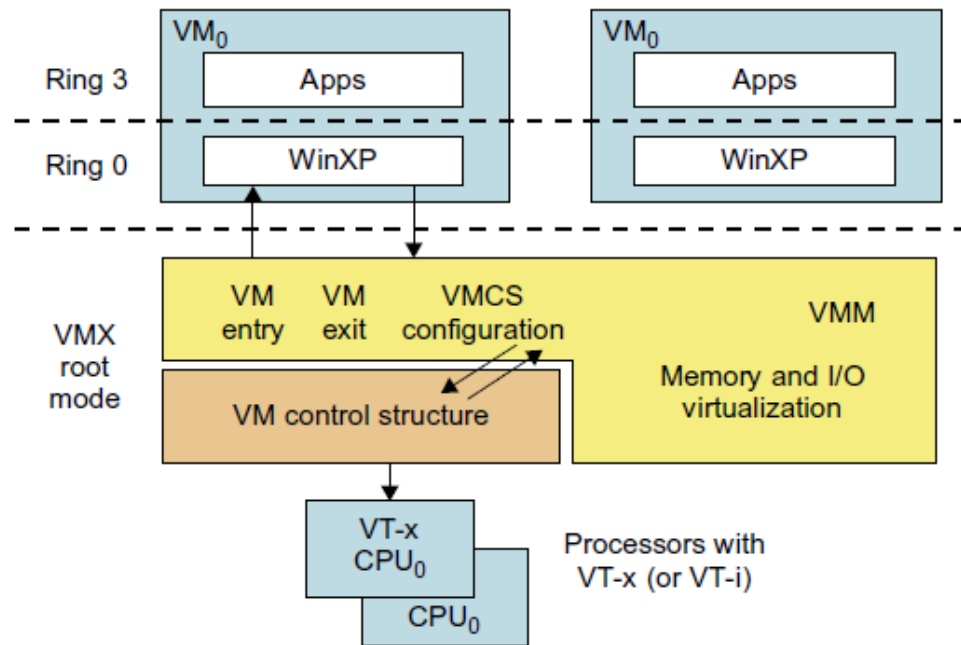
The most reported OS-level virtualization systems are Linux-based. Virtualization support on the Windows-based platform is still in the research stage. The Linux kernel offers an abstraction layer to allow software processes to work with and operate on resources without knowing the hardware details. New hardware may need a new Linux kernel to support. Therefore, different Linux platforms use patched kernels to provide special support for extended functionality.

Table 3.3 Virtualization Support for Linux and Windows NT Platforms

Virtualization Support and Source of Information	Brief Introduction on Functionality and Application Platforms
Linux vServer for Linux platforms (http://linux-vserver.org/)	Extends Linux kernels to implement a security mechanism to help build VMs by setting resource limits and file attributes and changing the root environment for VM isolation
OpenVZ for Linux platforms [65]; http://ftp.openvz.org/doc/OpenVZ-Users-Guide.pdf	Supports virtualization by creating <i>virtual private servers</i> (VPSes); the VPS has its own files, users, process tree, and virtual devices, which can be isolated from other VPSes, and checkpointing and live migration are supported
FVM (Feather-Weight Virtual Machines) for virtualizing the Windows NT platforms [78]	Uses system call interfaces to create VMs at the NT kernel space; multiple VMs are supported by virtualized namespace and copy-on-write

OR

4	a.	<p>Explain virtualization of CPU/memory and I/O devices</p>  <p>Modern operating systems and processors permit multiple processes to run simultaneously. If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash. Therefore, all processors have at least two modes, user mode and supervisor mode, to ensure controlled access of critical hardware. Instructions running in supervisor mode are called privileged instructions. Other instructions are unprivileged instructions. In a virtualized environment, it is more difficult to make OSes and applications run correctly because there are more layers in the machine stack.</p> <p>CPU Virtualization</p> <p>A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode. When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM. In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system. However, not all CPU architectures are virtualizable.</p> <p>Hardware Assisted CPU virtualization</p>	1 0	L 2 CO3
---	----	---	--------	-------------------

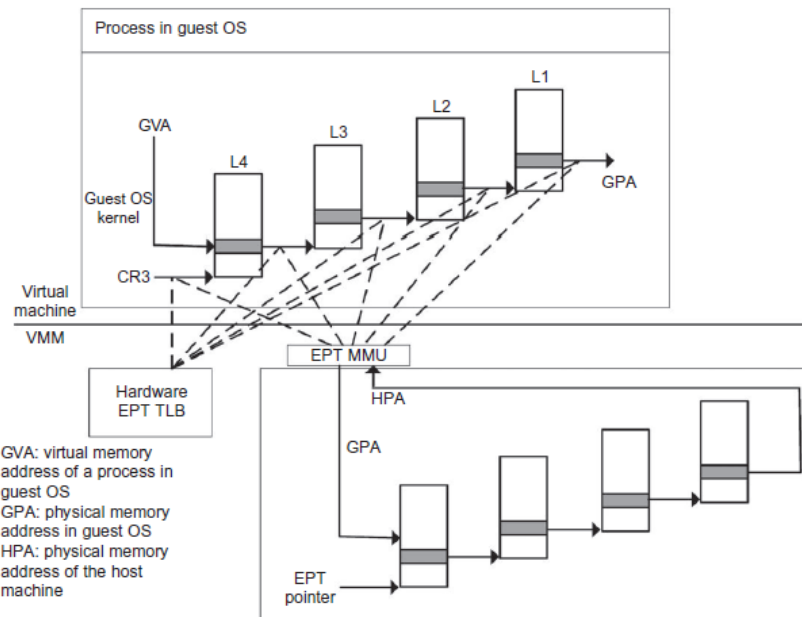


Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance. However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs. That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.

Furthermore, MMU virtualization should be supported, which is transparent to the guest OS. The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs. But the guest OS cannot directly access the actual machine memory. The VMM is responsible for mapping the guest physical memory to the actual machine memory.

Two-level memory mapping procedure.

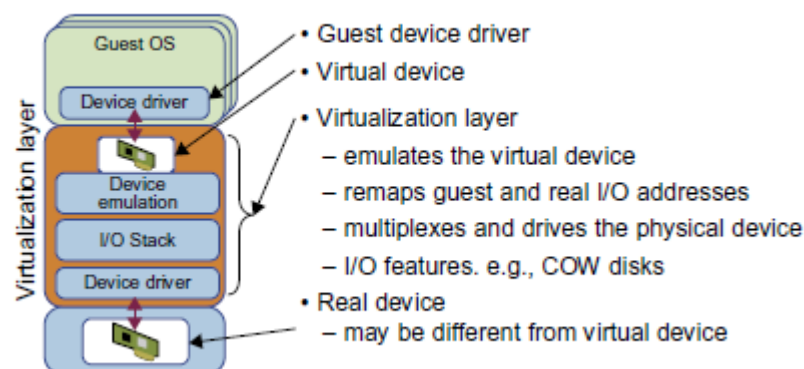


I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware. At the time of this writing, there are three ways to implement I/O virtualization: full device emulation, para-virtualization, and direct I/O. Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices.

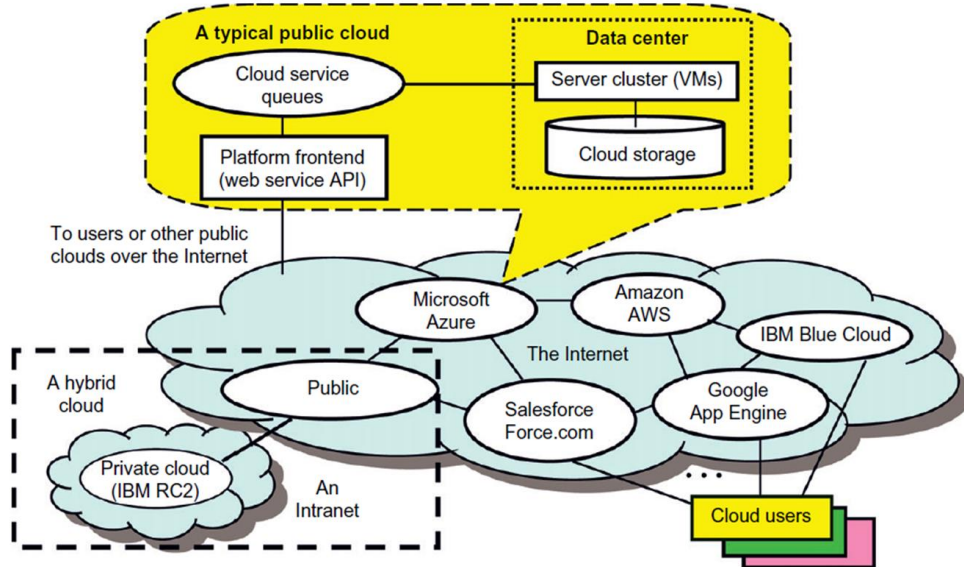
All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.

Full Device Emulation Approach



	<p>b. Describe virtualization for data center automation.</p> <p>SOLUTION</p> <p>Data centers have grown rapidly in recent years, and all major IT companies are pouring their resources into building new data centers. In addition, Google, Yahoo!, Amazon, Microsoft, HP, Apple, and IBM are all in the game. All these companies have invested billions of dollars in datacenter construction and automation. Data-center automation means that huge volumes of hardware, software, and database resources in these data centers can be allocated dynamically to millions of Internet users simultaneously, with guaranteed QoS and cost-effectiveness.</p> <p>Server Consolidation in Data Centers</p> <p>In data centers, a large number of heterogeneous workloads can run on servers at various times. These heterogeneous workloads can be roughly divided into two categories: chatty workloads and noninteractive workloads. Chatty workloads may burst at some point and return to a silent state at some other point. A web video service is an example of this, whereby a lot of people use it at night and few people use it during the day. Noninteractive workloads do not require people's efforts to make progress after they are submitted.</p> <p>High-performance computing is a typical example of this. At various stages, the requirements for resources of these workloads are dramatically different. However, to guarantee that a workload will always be able to cope with all demand levels, the workload is statically allocated enough resources so that peak demand is satisfied</p> <p>server virtualization has the following side effects:</p> <ul style="list-style-type: none"> • Consolidation enhances hardware utilization. Many underutilized servers are consolidated into fewer servers to enhance resource utilization. Consolidation also facilitates backup services and disaster recovery. • This approach enables more agile provisioning and deployment of resources. In a virtual environment, the images of the guest OSES and their applications are readily cloned and reused. • The total cost of ownership is reduced. In this sense, server virtualization causes deferred purchases of new servers, a smaller data-center footprint, lower maintenance costs, and lower power, cooling, and cabling requirements. • This approach improves availability and business continuity. The crash of a guest OS has no effect on the host OS or any other guest OS. It becomes easier to transfer a VM from one server to another, because virtual servers are unaware of the underlying hardware. <p>To automate data-center operations, one must consider resource scheduling, architectural support, power management, automatic or autonomic resource management, performance of analytical models, and so on. In virtualized data centers, an efficient, on-demand, fine-grained scheduler is one of the key factors to improve resource utilization. Scheduling and reallocations can be done in a wide range of levels in a set of data centers. The levels match at least at the VM level, server level, and data-center level. Ideally, scheduling and resource reallocations should be done at all levels.</p> <p>However, due to the complexity of this, current techniques only focus on a single level or, at most, two levels.</p> <p>Dynamic CPU allocation is based on VM utilization and application-level QoS metrics. One method considers both CPU and memory flowing as well as automatically adjusting resource overhead based on varying workloads in hosted services. Another scheme uses a two-level resource management system to handle the complexity involved. A local controller at the VM level and a global controller at the server level are designed. They implement autonomic resource allocation via the interaction of the local and global controllers. Multicore and virtualization are two cutting techniques that can enhance each other.</p> <p>Virtual Storage Management</p>	10	L2	CO2
--	--	----	----	-----

		<p>The term “storage virtualization” was widely used before the renaissance of system virtualization. Yet the term has a different meaning in a system virtualization environment. Previously, storage virtualization was largely used to describe the aggregation and repartitioning of disks at very coarse time scales for use by physical machines. In system virtualization, virtual storage includes the storage managed by VMMs and guest OSes. Generally, the data stored in this environment can be classified into two categories: VM images and application data.</p> <p>Cloud OS for Virtualized Data Centers</p> <p>Data centers must be virtualized to serve as cloud providers</p> <table><tr><th colspan="6">Table 3.6 VI Managers and Operating Systems for Virtualizing Data Centers [9]</th></tr><tr><th>Manager/ OS, Platforms, License</th><th>Resources Being Virtualized, Web Link</th><th>Client API, Language</th><th>Hypervisors Used</th><th>Public Cloud Interface</th><th>Special Features</th></tr><tr><td>Nimbus Linux, Apache v2</td><td>VM creation, virtual cluster, www.nimbusproject.org/</td><td>EC2 WS, WSRF, CLI</td><td>Xen, KVM</td><td>EC2</td><td>Virtual networks</td></tr><tr><td>Eucalyptus Linux, BSD</td><td>Virtual networking (Example 3.12 and [41]), www.eucalyptus.com/</td><td>EC2 WS, CLI</td><td>Xen, KVM</td><td>EC2</td><td>Virtual networks</td></tr><tr><td>OpenNebula Linux, Apache v2</td><td>Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/</td><td>XML-RPC, CLI, Java</td><td>Xen, KVM</td><td>EC2, Elastic Host</td><td>Virtual networks, dynamic provisioning</td></tr><tr><td>vSphere 4 Linux, Windows, proprietary</td><td>Virtualizing OS for data centers (Example 3.13), www.vmware.com/products/vsphere/ [66]</td><td>CLI, GUI, Portal, WS</td><td>VMware ESX, ESXi</td><td>VMware vCloud partners</td><td>Data protection, vStorage, VMFS, DRM, HA</td></tr></table> <p>These VI managers are used to create VMs and aggregate them into virtual clusters as elastic resources.</p> <p>Trust Management in Virtualized Data Centers</p> <p>A VMM changes the computer architecture. It provides a layer of software between the operating systems and system hardware to create one or more VMs on a single physical platform. A VM entirely encapsulates the state of the guest operating system running inside it. Encapsulated machine state can be copied and shared over the network and removed like a normal file, which proposes a challenge to VM security.</p>	Table 3.6 VI Managers and Operating Systems for Virtualizing Data Centers [9]						Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features	Nimbus Linux, Apache v2	VM creation, virtual cluster, www.nimbusproject.org/	EC2 WS, WSRF, CLI	Xen, KVM	EC2	Virtual networks	Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and [41]), www.eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks	OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML-RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning	vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www.vmware.com/products/vsphere/ [66]	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA		
Table 3.6 VI Managers and Operating Systems for Virtualizing Data Centers [9]																																								
Manager/ OS, Platforms, License	Resources Being Virtualized, Web Link	Client API, Language	Hypervisors Used	Public Cloud Interface	Special Features																																			
Nimbus Linux, Apache v2	VM creation, virtual cluster, www.nimbusproject.org/	EC2 WS, WSRF, CLI	Xen, KVM	EC2	Virtual networks																																			
Eucalyptus Linux, BSD	Virtual networking (Example 3.12 and [41]), www.eucalyptus.com/	EC2 WS, CLI	Xen, KVM	EC2	Virtual networks																																			
OpenNebula Linux, Apache v2	Management of VM, host, virtual network, and scheduling tools, www.opennebula.org/	XML-RPC, CLI, Java	Xen, KVM	EC2, Elastic Host	Virtual networks, dynamic provisioning																																			
vSphere 4 Linux, Windows, proprietary	Virtualizing OS for data centers (Example 3.13), www.vmware.com/products/vsphere/ [66]	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vCloud partners	Data protection, vStorage, VMFS, DRM, HA																																			
MODULE - 03 Dr. Preethi Sheba Hepsiba																																								
5	a.	Explain cloud service models with the diagram.			5 L 2 CO2																																			



A public cloud is built over the Internet and can be accessed by any user who has paid for the service.

Public clouds are owned by service providers and are accessible through a subscription.

public cloud delivers a selected set of business processes.

The application and infrastructure services are offered on a flexible price on a pay-per-use basis.

Eg. Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com

A private cloud is built within the domain of an intranet owned by a single organization. It is client owned and managed, and its access is limited to the owning clients and their partners.

Private clouds give local users a flexible and agile private infrastructure

They can run service workloads within their administrative domains

It may impact the cloud standardization, while retaining greater customization and organizational control Eg. U.S. National Aeronautics and Space Administration (NASA) was a pioneer in private cloud in 2009 called Nebula in 2009 at the Ames Research Center (Ames) – shut down in 2012

European Council for Nuclear Research - CERN – private cloud using OpenStack

A hybrid cloud is built with both public and private clouds

In this model, a private cloud's local infrastructure is supplemented with computing capacity from an external public cloud.

A hybrid cloud provides access to clients, the partner network, and third

		parties.			
b.	Explain cloud deployment models Infrastructure-as-a-Service(IaaS) Applications can be run over their own OS environment user does not manage or control the underlying cloud infrastructure has control over the OS, storage, deployed applications, and possibly select networking components storage as a service, compute instances as a service, and communication as a service		5	L 2	CO2

Table 4.1 Public Cloud Offerings of IaaS [10,18]			
Cloud Name	VM Instance Capacity	API and Access Tools	Hypervisor, Guest OS
Amazon EC2	Each instance has 1–20 EC2 processors, 1.7–15 GB of memory, and 160–1.69 TB of storage.	CLI or web Service (WS) portal	Xen, Linux, Windows
GoGrid	Each instance has 1–6 CPUs, 0.5–8 GB of memory, and 30–480 GB of storage.	REST, Java, PHP, Python, Ruby	Xen, Linux, Windows
Rackspace Cloud	Each instance has a four-core CPU, 0.25–16 GB of memory, and 10–620 GB of storage.	REST, Python, PHP, Java, C#, .NET	Xen, Linux
FlexiScale in the UK	Each instance has 1–4 CPUs, 0.5–16 GB of memory, and 20–270 GB of storage.	web console	Xen, Linux, Windows
Joyent Cloud	Each instance has up to eight CPUs, 0.25–32 GB of memory, and 30–480 GB of storage.	No specific API, SSH, Virtual/Min	OS-level virtualization, OpenSolaris

PaaS – Platform as a Service

To develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment

includes operating system and runtime library support.

user application can be developed on this virtualized cloud platform using some programming languages and software tools supported by the provider (e.g., Java, Python, .NET).

User does not manage the underlying cloud infrastructure

Enables collaborated software development platform for users from different parts of the world

Encourages third parties to provide software management, integration, and service monitoring

Table 4.2 Five Public Cloud Offerings of PaaS [10,18]

Cloud Name	Languages and Developer Tools	Programming Models Supported by Provider	Target Applications and Storage Option
Google App Engine	Python, Java, and Eclipse-based IDE	MapReduce, web programming on demand	Web applications and BigTable storage
Salesforce.com's Force.com	Apex, Eclipse-based IDE, web-based Wizard	Workflow, Excel-like formula, Web programming on demand	Business applications such as CRM
Microsoft Azure	.NET, Azure tools for MS Visual Studio	Unrestricted model	Enterprise and web applications
Amazon Elastic MapReduce	Hive, Pig, Cascading, Java, Ruby, Perl, Python, PHP, R, C++	MapReduce	Data processing and e-commerce
Aneka	.NET, stand-alone SDK	Threads, task, MapReduce	.NET enterprise applications, HPC

Software as a Service(SaaS)

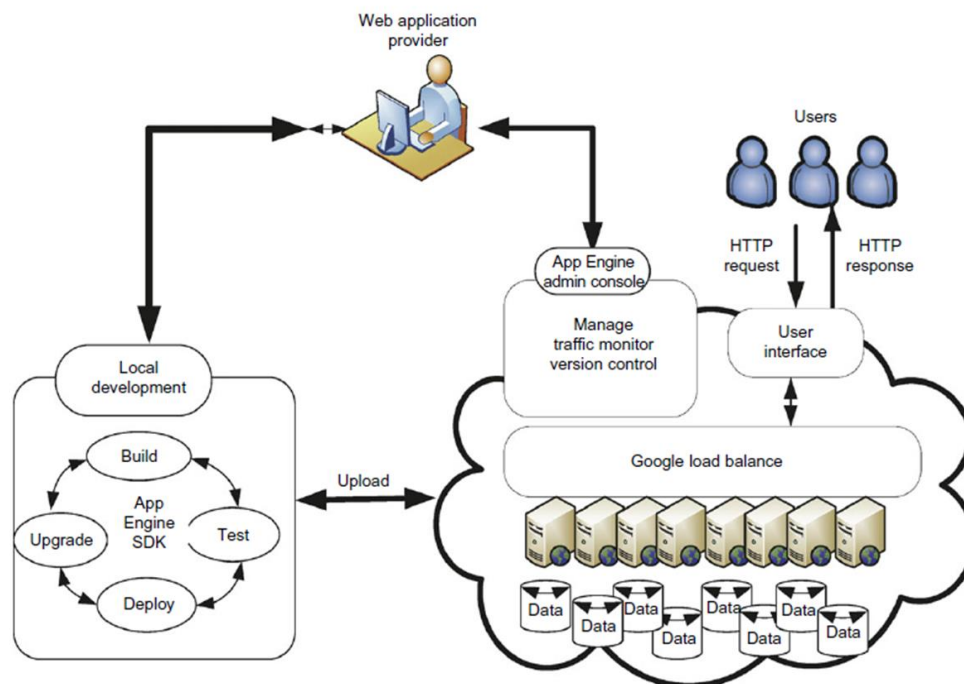
browser-initiated application software over thousands of cloud customers on the customer side, there is no upfront investment in servers or software licensing

provider side, costs are kept rather low, compared with conventional hosting of user applications

Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS

Eg. Google Gmail and docs, Microsoft SharePoint, and the CRM software from Salesforce.com

c. Write a note on public cloud platforms, GAE, AWS and Azure.

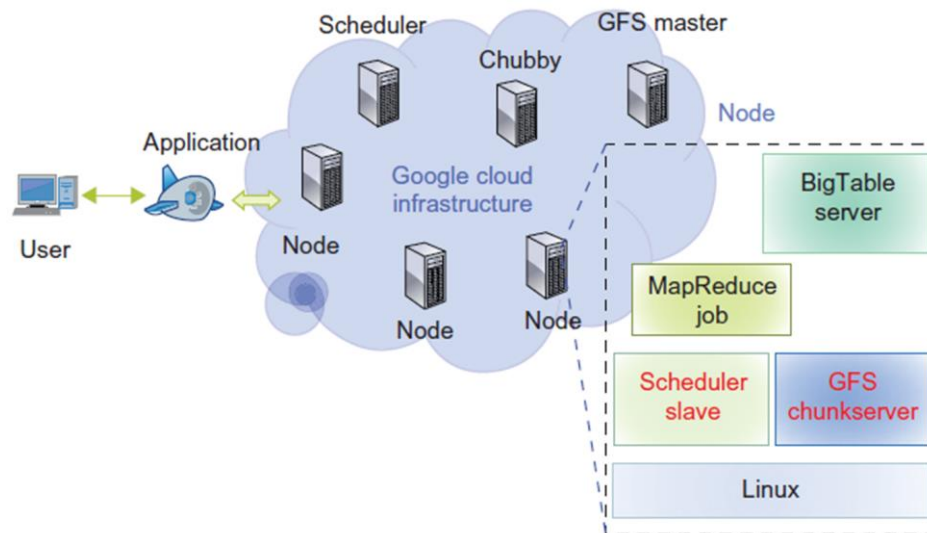


Google provides a fully featured local development environment that simulates GAE on the developer's computer.

1 L
0 2

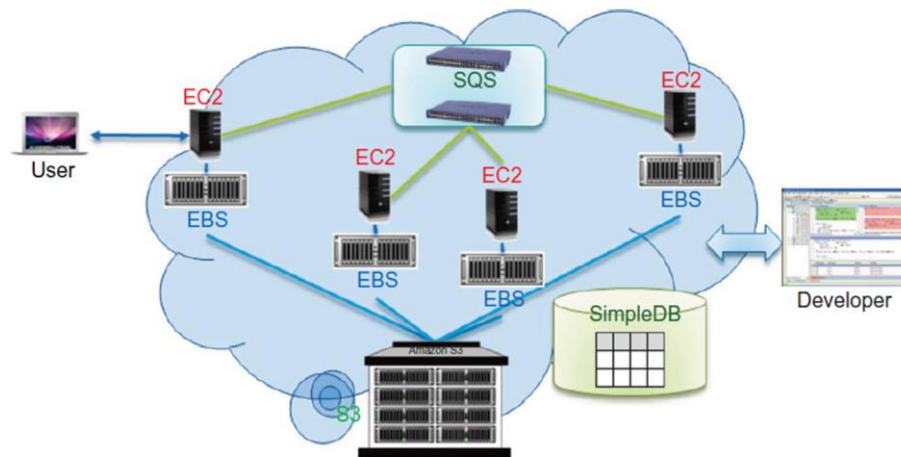
CO3

coding and debugging stages can be performed locally
 SDK provided provides a tool for uploading the user's application to
 Google's infrastructure where the applications are actually deployed
 additional third-party capabilities, including software management,
 integration, and service monitoring solutions



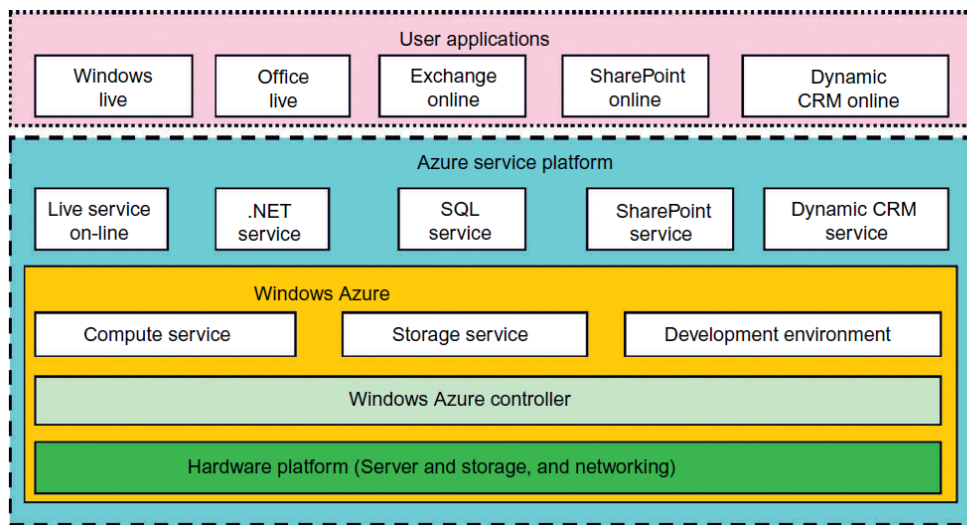
GFS is used for storing large amounts of data
 MapReduce is for use in application program development
 Chubby is used for distributed application lock services
 BigTable offers a storage service for accessing structured data
 Functional Modules of GAE
 Datastore
 Application runtime environment
 Software Development Kit (SDK)
 Administration console
 GAE Web service infrastructure
 service is free within a quota
 Well known GAE applications : Google Search Engine, Google Docs,
 Google Earth, and Gmail
 can support large numbers of users simultaneously
 Third-party application providers can use GAE to build cloud applications
 for providing services

AWS



Amazon applies the IaaS model in providing its services
 EC2 provides the virtualized platforms to the host VMs where the cloud application can run.
 S3 (Simple Storage Service) provides the object-oriented storage service for users.
 EBS (Elastic Block Service) provides the block storage
 SQS stands for Simple Queue Service, and its job is to ensure a reliable message service between two processes.
 ELB automatically distributes incoming application traffic across multiple Amazon EC2 instances
 CloudWatch monitors running instances
 Amazon (like Azure) offers a Relational Database Service (RDS)

Azure



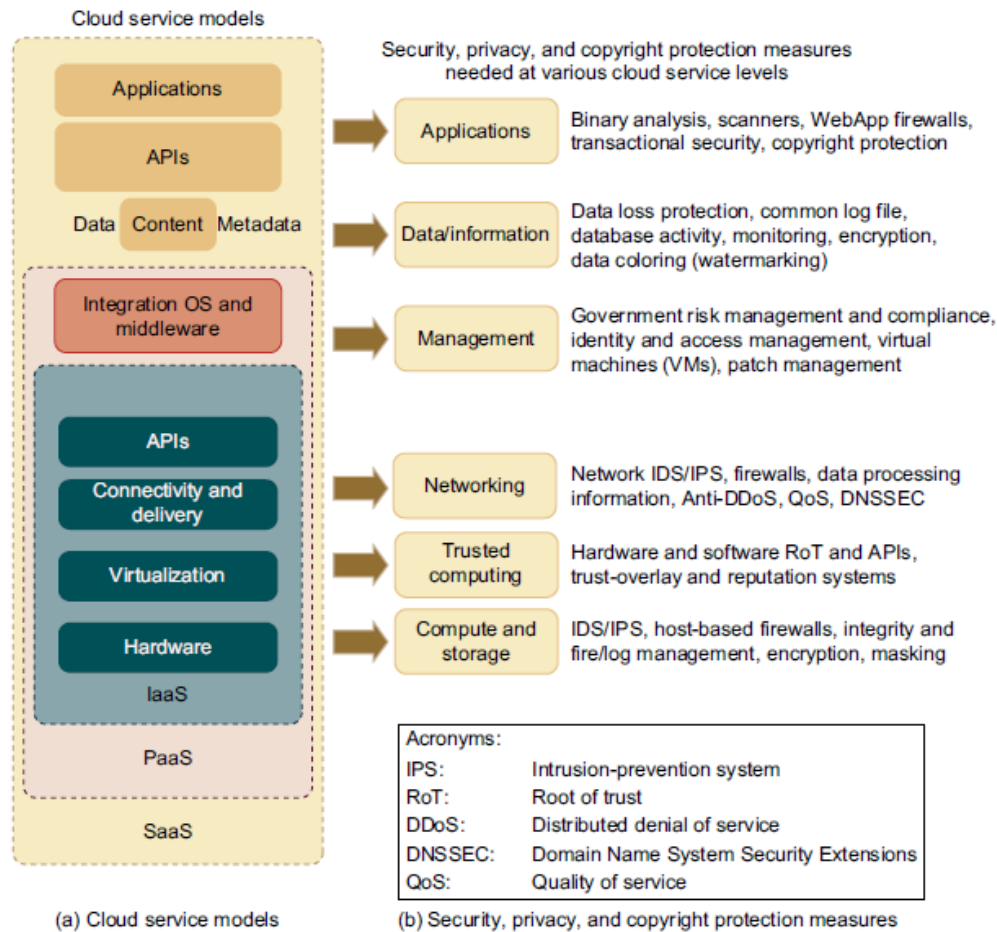
		<ul style="list-style-type: none"> • The platform is divided into three major component platforms • Services <ul style="list-style-type: none"> – Live service Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently. – .NET service This package supports application development on local hosts and execution on cloud machines. – SQL Azure This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud – SharePoint service This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services. – Dynamic CRM service This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions. • applies the standard web communication protocols SOAP and REST 			
OR					
6	a.	<p>Define cloud computing and list the characteristics</p> <p>The concept of cloud computing has evolved from cluster, grid, and utility computing. Cluster and grid computing leverage the use of many computers in parallel to solve problems of any size. Utility and Software as a Service (SaaS) provide computing resources as a service with the notion of pay per use. Cloud computing leverages dynamic resources to deliver large numbers of services to end users.</p> <p>Cloud computing is a high-throughput computing (HTC) paradigm whereby the infrastructure provides the services through a large data center or server farms. The cloud computing model enables users to share access to resources from anywhere at any time through their connected devices.</p> <p>Characteristics of cloud :</p> <ol style="list-style-type: none"> 1. Desired location in areas with protected space and higher energy efficiency 2. Sharing of peak-load capacity among a large pool of users, improving overall utilization 3. Separation of infrastructure maintenance duties from domain-specific application development 4. Significant reduction in cloud computing cost, compared with traditional computing paradigms 5. Cloud computing programming and application development 6. Service and data discovery and content/service distribution 7. Privacy, security, copyright, and reliability issues 8. Service agreements, business models, and pricing policies 	5	L 1	CO1

	<p>b. Write benefits and challenges of each service</p> <p>IaaS This model allows users to use virtualized IT resources for computing, storage, and networking. In short, the service is performed by rented cloud infrastructure. The user can deploy and run his applications over his chosen OS environment. The user does not manage or control the underlying cloud infrastructure, but has control over the OS, storage, deployed applications, and possibly select networking components. This IaaS model encompasses storage as a service, compute instances as a service, and communication as a service.</p> <p>PaaS To be able to develop, deploy, and manage the execution of applications using provisioned resources demands a cloud platform with the proper software environment. Such a platform includes operating system and runtime library support. This has triggered the creation of the PaaS model to enable users to develop and deploy their user applications.</p> <p>SaaS Services and tools offered by PaaS are utilized in construction of applications and management of their deployment on resources offered by IaaS providers. The SaaS model provides software applications as a service. As a result, on the customer side, there is no upfront investment in servers or software licensing. On the provider side, costs are kept rather low, compared with conventional hosting of user applications. Customer data is stored in the cloud that is either vendor proprietary or publicly hosted to support PaaS and IaaS.</p> <p>Issues and Challenges</p> <ul style="list-style-type: none"> • difficult for cloud customers to determine in advance the best location for hosting their services as they may not know the origin of consumers of their services. • no single cloud infrastructure provider will be able to establish its data centers at all possible locations throughout the world • SaaS providers may not be able to meet the QoS expectations of their service consumers originating from multiple geographical locations • Solution : seamless federation of data centers of a cloud provider or providers supporting dynamic scaling of applications across multiple domains in order to meet QoS targets of cloud customers. • cloud providers will be able to dynamically expand or resize their 	5	L 1	CO1
--	---	---	--------	-----

		<p>provisioning capability based on sudden spikes in workload demands by leasing available computational and storage capabilities from other cloud</p> <ul style="list-style-type: none">operate as part of a market-driven resource leasing federation, where application service providers such as Salesforce.com host their services based on negotiated SLA contracts driven by competitive market prices; and deliver on-demand, reliable, cost-effective, and QoS-aware services based on virtualization technologies while ensuring high QoS standards and minimizing service costs.																													
c.	<p>Write a note on Inter cloud resource management.</p> <p>Extended Cloud Computing Services Resource Provisioning and Platform Deployment Virtual Machine Creation and Management Global Exchange of Cloud Resources</p> <table><tr><td colspan="3">Cloud application (SaaS)</td><td>Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.</td></tr><tr><td colspan="3">Cloud software environment (PaaS)</td><td>Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay</td></tr><tr><td colspan="3">Cloud software infrastructure</td><td rowspan="2">Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth</td></tr><tr><td>Computational resources (IaaS)</td><td>Storage (DaaS)</td><td>Communications (Caas)</td></tr><tr><td colspan="3">Collocation cloud services (LaaS)</td><td>Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main</td></tr><tr><td colspan="3">Network cloud services (NaaS)</td><td>Owest, AT&T, AboveNet</td></tr><tr><td colspan="3">Hardware/Virtualization cloud services (HaaS)</td><td>VMware, Intel, IBM, XenEnterprise</td></tr></table> <p>1. Extended Cloud Services bottommost layer provides Hardware as a Service (HaaS). next layer is for interconnecting all the hardware components - Network as a Service (NaaS). Virtual LANs fall within the scope of NaaS. next layer up offers Location as a Service (LaaS) - provides a collocation service to house, power, and secure all the physical hardware and network resources. This layer provides Security as a Service (“SaaS”). The cloud infrastructure layer can be further subdivided Data as a Service (DaaS) Communication as a Service (CaaS) Cloud Service Tasks – SaaS, PaaS,LaaS, NaaS, HaaS</p>	Cloud application (SaaS)			Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.	Cloud software environment (PaaS)			Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay	Cloud software infrastructure			Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth	Computational resources (IaaS)	Storage (DaaS)	Communications (Caas)	Collocation cloud services (LaaS)			Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main	Network cloud services (NaaS)			Owest, AT&T, AboveNet	Hardware/Virtualization cloud services (HaaS)			VMware, Intel, IBM, XenEnterprise	10	L3	CO3
Cloud application (SaaS)			Concur, RightNOW, Teleo, Kenexa, Webex, Blackbaud, salesforce.com, Netsuite, Kenexa, etc.																												
Cloud software environment (PaaS)			Force.com, App Engine, Facebook, MS Azure, NetSuite, IBM BlueCloud, SGI Cyclone, eBay																												
Cloud software infrastructure			Amazon AWS, OpSource Cloud, IBM Ensembles, Rackspace cloud, Windows Azure, HP, Banknorth																												
Computational resources (IaaS)	Storage (DaaS)	Communications (Caas)																													
Collocation cloud services (LaaS)			Savvis, Internap, NTTCommunications, Digital Realty Trust, 365 Main																												
Network cloud services (NaaS)			Owest, AT&T, AboveNet																												
Hardware/Virtualization cloud services (HaaS)			VMware, Intel, IBM, XenEnterprise																												

	<p>cloud mashup is practiced in vertical cloud applications</p> <p>Software Stack for Cloud computing</p> <p>platform for running cloud computing services can be either physical servers or virtual servers.</p> <p>Software layer on top of the platform is the layer for storing massive amounts of data.</p> <p>Other layers running on top of the file system are the layers for executing cloud computing applications.</p> <p>Runtime Support Services</p> <p>Runtime support is software needed in browser-initiated applications applied by thousands of cloud customers</p> <p>2. Resource Provisioning and Platform deployment</p> <p>Cloud architecture puts more emphasis on the number of processor cores or VM instances</p> <p>Provisioning of compute instances (VMs)</p> <p>must commit sufficient resources such as CPU, memory, and bandwidth that the user can use for a preset period</p> <p>Underprovisioning of resources will lead to broken SLAs and penalties.</p> <p>Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider</p> <p>Efficient VM provisioning depends on the cloud architecture and management of cloud infrastructures.</p> <p>In a virtualized cluster of servers, this demands efficient installation of VMs, live VM migration, and fast recovery from failures</p> <p>EC2 platform, some predefined VM templates are also provided</p> <p>IBM's Blue Cloud does not provide any VM templates.</p> <p>In general, any type of VM can run on top of Xen</p> <p>Power-efficient schemes for caching, query processing, and thermal management are mandatory due to increasing energy waste by heat dissipation from data centers.</p> <p>Dynamic Resource Deployment - IGG (Intergrid gateway)</p> <p>Can achieve scalability in performance</p> <p>IGG developed by Melbourne University group – Java implemented software that creates a cloud environment for all participating grid resources.</p> <p>deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases, and (3) deploying the VMs as requested.</p> <p>Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider</p> <p>A grid has predefined peering arrangements with other grids, which the IGG manages</p> <p>The InterGrid allocates and provides a distributed virtual environment (DVE).</p> <p>This is a virtual cluster of VMs that runs isolated from other virtual clusters.</p>		
--	--	--	--

		DVE manager performs resource allocation and management on behalf of specific user applications.			
MODULE 4					
7	a.	<p>Summarize cloud data encryption and challenges in data encryption.</p> <p>SOLUTION</p> <p>data protection was done by encryption or decryption which is computationally expensive. The data coloring takes a minimal number of calculations to color or decolor the data objects. Cryptography and watermarking or coloring can be used jointly in a cloud environment. With shared files and data sets, privacy, security, and copyright information could be compromised in a cloud computing environment. Users desire to work in a trusted software environment that provides useful tools to build cloud applications over protected data sets. In the past, watermarking was mainly used for digital copyright management. The system generates special colors for each data object. Data coloring means labeling each data object by a unique color. Differently colored data objects are thus distinguishable. The user identification is also colored to be matched with the data colors. This color matching process can be applied to implement different trust management events. Cloud storage provides a process for the generation, embedding, and extraction of the watermarks in colored objects.</p>	8	L 2	CO1
	b.	<p>Write a note on cloud security defense strategies.</p> <p>SOLUTION</p>	6	L 2	CO1



Basic Cloud Security

Three basic cloud security enforcements are expected. First, facility security in data centers demands on-site security year round. Biometric readers, CCTV (close-circuit TV), motion detection, and man traps are often deployed. Also, network security demands fault-tolerant external firewalls, intrusion detection systems (IDSes), and third-party vulnerability assessment. Finally, platform security demands SSL and data decryption, strict password policies, and system trust certification.

Security Challenges in VMs

As we discussed earlier in this chapter, traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits. Another type of attack is the man-in-the-middle attack for VM migrations.

Cloud Defense Methods

Virtualization enhances cloud security. But VMs add an additional layer of software that could become a single point of failure. With virtualization, a single physical machine can be divided or partitioned into multiple VMs (e.g., server consolidation).

Table 4.9 Physical and Cyber Security Protection at Cloud/Data Centers

Protection Schemes	Brief Description and Deployment Suggestions
Secure data centers and computer buildings	Choose hazard-free location, enforce building safety. Avoid windows, keep buffer zone around the site, bomb detection, camera surveillance, earthquake-proof, etc.
Use redundant utilities at multiple sites	Multiple power and supplies, alternate network connections, multiple databases at separate sites, data consistency, data watermarking, user authentication, etc.
Trust delegation and negotiation	Cross certificates to delegate trust across PKI domains for various data centers, trust negotiation among certificate authorities (CAs) to resolve policy conflicts
Worm containment and DDoS defense	Internet worm containment and distributed defense against DDoS attacks to secure all data centers and cloud platforms
Reputation system for data centers	Reputation system could be built with P2P technology; one can build a hierarchy of reputation systems from data centers to distributed file systems
Fine-grained file access control	Fine-grained access control at the file or object level; this adds to security protection beyond firewalls and IDSes
Copyright protection and piracy prevention	Piracy prevention achieved with peer collusion prevention, filtering of poisoned content, nondestructive read, alteration detection, etc.
Privacy protection	Uses double authentication, biometric identification, intrusion detection and disaster recovery, privacy enforcement by data watermarking, data classification, etc.

Defense with Virtualization

The VM is decoupled from the physical hardware. The entire VM can be represented as a software component and can be regarded as binary or digital data. The VM can be saved, cloned, encrypted, moved, or restored with ease.

Privacy and Copyright Protection

The user gets a predictable configuration before actual system integration. Yahoo!'s Pipes is a good example of a lightweight cloud platform. With shared files and data sets, privacy, security, and copyright data could be compromised in a cloud computing environment.

Distributed Intrusion/Anomaly Detection

Data security is the weakest link in all cloud models. We need new cloud security standards to apply common API tools to cope with the data lock-in problem and network attacks or abuses.

The IaaS model represented by Amazon is most sensitive to external attacks.

Distributed Defense against DDoS Flooding Attacks

A DDoS defense system must be designed to cover multiple network domains spanned by a given cloud platform. These network domains cover the edge networks where cloud resources are connected. DDoS attacks come with widespread worms. The flooding traffic is large enough to crash the victim server by buffer overflow, disk exhaustion, or connection saturation

Data and Software Protection Techniques

Data Integrity and Privacy Protection

Such software should offer the following features:

- Special APIs for authenticating users and sending e-mail using commercial accounts
- Fine-grained access control to protect data integrity and deter intruders or hackers
- Shared data sets protected from malicious alteration, deletion, or copyright violation

Data Coloring and Cloud Watermarking

With shared files and data sets, privacy, security, and copyright information could be compromised in a cloud computing environment.

Data Lock-in Problem and Proactive Solutions

Cloud computing moves both the computation and the data to the server clusters maintained by cloud service providers. Once the data is moved into the cloud, users cannot easily extract their data and programs from cloud servers to run on another platform.

Reputation-Guided Protection of Data Centers

Trust is a personal opinion, which is very subjective and often biased. Trust can be transitive but not necessarily symmetric between two parties.

		<p>Reputation System Design Options</p> <p>Public opinion on the character or standing (such as honest behavior or reliability) of an entity could be the reputation of a person, an agent, a product, or a service. It represents a collective evaluation by a group of people/ agents and resource owners. Many reputation systems have been proposed in the past mainly for P2P, multiagent, or e-commerce systems.</p> <p>Reputation Systems for Clouds</p> <p>Redesigning the aforementioned reputation systems for protecting data centers offers new opportunities for expanded applications beyond P2P networks. Data consistency is checked across multiple databases.</p> <p>Trust Overlay Networks</p> <p>Reputation represents a collective evaluation by users and resource owners. Many reputation systems have been proposed in the past for P2P, multiagent, or e-commerce systems. To support trusted cloud services,</p>			
	c.	<p>Explain anomaly detection techniques in cloud.</p> <p>SOLUTION</p> <p>The IaaS model represented by Amazon is most sensitive to external attacks. Role-based interface tools alleviate the complexity of the provisioning system. For example, IBM's Blue Cloud provisions through a role-based web portal. A SaaS bureau may order secretarial services from a common cloud platform. Many IT companies are now offering cloud services with no guaranteed security.</p> <p>Security threats may be aimed at VMs, guest OSes, and software running on top of the cloud. IDSes attempt to stop these attacks before they take effect. Both signature matching and anomaly detection can be implemented on VMs dedicated to building IDSes. Signature-matching IDS technology is more mature, but require frequent updates of the signature databases. Network anomaly detection reveals abnormal traffic patterns, such as unauthorized episodes of TCP connection sequences, against normal traffic patterns. Distributed IDSes are needed to combat both types of intrusions.</p>	6	L 3	CO3
OR					

8	a.	<p>Describe data and software protection techniques</p> <p>SOLUTION</p> <p>Data and Software Protection Techniques In this section, we will introduce a data coloring technique to preserve data integrity and user privacy. Then we will discuss a watermarking scheme to protect software files from being widely distributed in a cloud environment.</p> <p>1. Data Integrity and Privacy Protection Users desire a software environment that provides many useful tools to build cloud applications over large data sets. In addition to application software for MapReduce, BigTable, EC2, 3S, Hadoop, AWS, GAE, and WebSphere2, users need some security and privacy protection software for using the cloud. Such software should offer the following features:</p> <ul style="list-style-type: none"> • Special APIs for authenticating users and sending e-mail using commercial accounts • Fine-grained access control to protect data integrity and deter intruders or hackers • Shared data sets protected from malicious alteration, deletion, or copyright violation <p>Ability to secure the ISP or cloud service provider from invading users' privacy</p> <ul style="list-style-type: none"> • Personal firewalls at user ends to keep shared data sets from Java, JavaScript, and ActiveX applets • A privacy policy consistent with the cloud service provider's policy, to protect against identity theft, spyware, and web bugs • VPN channels between resource sites to secure transmission of critical data objects <p>2. Data Coloring and Cloud Watermarking With shared files and data sets, privacy, security, and copyright information could be compromised in a cloud computing environment. Users desire to work in a trusted software environment that provides useful tools to build cloud applications over protected data sets. In the past, watermarking was mainly used for digital copyright management. As shown in Figure 4.35, the system generates special colors for each data object. Data coloring means labeling each data object by a unique color. Differently colored data objects are thus distinguishable.</p> <p>The user identification is also colored to be matched with the data colors. This color matching process can be applied to implement different trust management events. Cloud storage provides a process for the generation, embedding, and extraction of the watermarks in colored objects.</p>	8	L 2	CO2
---	----	--	---	--------	-----

	<p>Interested readers may refer to the articles by Hwang and Li [36] for details on the data coloring and matching process. In general, data protection was done by encryption or</p> <p>FIGURE 4.35 Data coloring with cloud watermarking for trust management at various security clearance levels in data centers.</p> <p>decryption which is computationally expensive. The data coloring takes a minimal number of calculations to color or decolor the data objects. Cryptography and watermarking or coloring can be used jointly in a cloud environment.</p> <p>3. Data Lock-in Problem and Proactive Solutions</p> <p>Cloud computing moves both the computation and the data to the server clusters maintained by cloud service providers. Once the data is moved into the cloud, users cannot easily extract their data and programs from cloud servers to run on another platform. This leads to a data lock-in problem. This has hindered the use of cloud computing. Data lock-in is attributed to two causes: lack of interoperability, whereby each cloud vendor has its proprietary API that limits users to extract data once submitted; and lack of application compatibility, in that most computing clouds expect users to write new applications from scratch, when they switch cloud platforms.</p> <p>One possible solution to data lock-in is the use of standardized cloud APIs. This requires building standardized virtual platforms that adhere to OVF, a platform-independent, efficient, extensible, and open format for VMs. This will enable efficient, secure software distribution, facilitating the mobility of VMs. Using OVF one can move data from one application to another. This will enhance QoS, and thus enable cross-cloud applications, allowing workload migration among data centers to user-specific storage. By deploying applications, users can access and intermix applications across different cloud services.</p>		
	<p>b. Briefly explain reputation-guided protection of data centers.</p> <p>SOLUTION</p> <p>Reputation-Guided Protection of Data Centers</p> <p>Trust is a personal opinion, which is very subjective and often biased. Trust can be transitive but not necessarily symmetric between two parties. Reputation is a public opinion, which is more objective and often relies on a large opinion aggregation process to evaluate. Reputation may change or decay over time. Recent reputation should be given more preference than past reputation. In this section, we review the reputation systems for protecting data centers or cloud user communities.</p> <p>1. Reputation System Design Options</p>	6 L 2	CO1

Figure 4.36 provides an overview of reputation system design options. Public opinion on the character or standing (such as honest behavior or reliability) of an entity could be the reputation of a person, an agent, a product, or a service. It represents a collective evaluation by a group of people/ agents and resource owners. Many reputation systems have been proposed in the past mainly for P2P, multiagent, or e-commerce systems. To address reputation systems for cloud services, a systematic approach is based on the design criteria and administration of the reputation systems. Figure 4.36 shows a two-tier classification of existing reputation systems that have been proposed in recent years. Most of them were designed for P2P or social networks. These reputation systems can be converted for protecting cloud computing applications. In general, the reputation systems are classified as centralized or distributed depending on how they are implemented. In a centralized system, a single central authority is responsible for managing the reputation system, while the distributed model involves multiple control centers working collectively. Reputation-based trust management and techniques for securing P2P and social networks could be merged to defend data centers and cloud platforms against attacks from the open network.

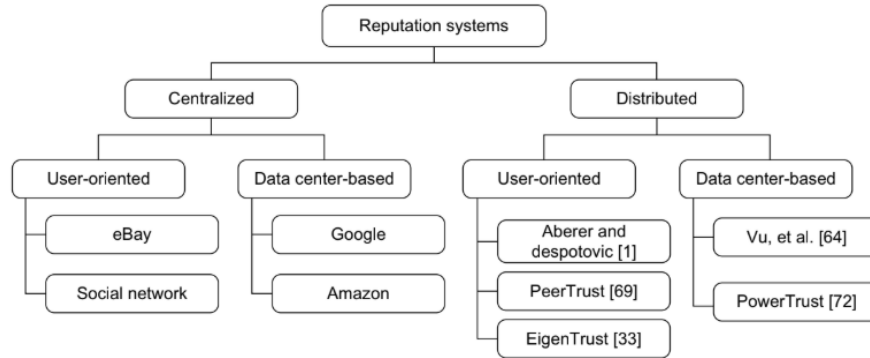


FIGURE 4.36

Design options of reputation systems for social networks and cloud platforms.

A centralized reputation system is easier to implement, but demands more powerful and reliable server resources; a distributed reputation system is much more complex to build. Distributed systems are more scalable and reliable in terms of handling failures. At the second tier, reputation systems are further classified by the scope of reputation evaluation. User-oriented reputation systems focus on individual users or agents. Most P2P reputation systems belong to this category. In data centers, reputation is modeled for the resource site as a whole. This reputation applies to products or services offered by the cloud. Commercial reputation systems have been built by eBay, Google, and Amazon in connection with the services they provide. These are centralized reputation systems. Distributed reputation systems are mostly developed by academic research communities. Aberer and Despotovic have proposed a model to manage trust in P2P systems. The Eigentrust reputation system was developed at Stanford University using a trust matrix approach. The PeerTrust system was developed at Georgia Institute of Technology for supporting e-commerce applications. The PowerTrust system was developed at the University of Southern California based on Power law characteristics of Internet traffic for P2P applications. Vu, et al. proposed a QoS-based ranking system for P2P Transactions.

2. Reputation Systems for Clouds

	<p>Redesigning the aforementioned reputation systems for protecting data centers offers new opportunities for expanded applications beyond P2P networks. Data consistency is checked across multiple databases. Copyright protection secures wide-area content distributions. To separate user data from specific SaaS programs, providers take the most responsibility in maintaining data integrity and consistency. Users can switch among different services using their own data. Only the users have the keys to access the requested data.</p> <p>The data objects must be uniquely named to ensure global consistency. To ensure data consistency, unauthorized updates of data objects by other cloud users are prohibited. The reputation system can be implemented with a trust overlay network. A hierarchy of P2P reputation systems is suggested to protect cloud resources at the site level and data objects at the file level. This demands both coarse-grained and fine-grained access control of shared resources. These reputation systems keep track of security breaches at all levels. The reputation system must be designed to benefit both cloud users and data centers. Data objects used in cloud computing reside in multiple data centers over a SAN. In the past, most reputation systems were designed for P2P social networking or for online shopping services. These reputation systems can be converted to protect cloud platform resources or user applications in the cloud. A centralized reputation system is easier to implement, but demands more powerful and reliable server resources. Distributed reputation systems are more scalable and reliable in terms of handling failures. The five security mechanisms presented earlier can be greatly assisted by using a reputation system specifically designed for data centers.</p> <p>However, it is possible to add social tools such as reputation systems to support safe cloning of VMs. Snapshot control is based on the defined RPO. Users demand new security mechanisms to protect the cloud. For example, one can apply secured information logging, migrate over secured virtual LANs, and apply ECC-based encryption for secure migration. Sandboxes provide a safe execution platform for running programs. Further, sandboxes can provide a tightly controlled set of resources for guest operating systems, which allows a security test bed to test the application code from third-party vendors.</p> <p>3. Trust Overlay Networks</p> <p>Reputation represents a collective evaluation by users and resource owners. Many reputation systems have been proposed in the past for P2P, multiagent, or e-commerce systems. To support trusted cloud services, Hwang and Li [36] have suggested building a trust overlay network to model trust relationships among data-center modules. This trust overlay could be structured with a distributed hash table (DHT) to achieve fast aggregation of global reputations from a large number of local reputation scores. This trust overlay design was first introduced in [12].</p> <p>Here, the designer needs to have two layers for fast reputation aggregation, updating, and dissemination to all users. Figure 4.37 shows construction of the two layers of the trust overlay network.</p> <p>At the bottom layer is the trust overlay for distributed trust negotiation and reputation aggregation over multiple resource sites. This layer handles user/server authentication, access authorization, trust delegation, and data integrity control. At the top layer is an overlay for fast virus/worm signature generation and dissemination and for piracy detection. This overlay facilitates worm containment and IDSes against viruses, worms, and DDoS attacks. The content poisoning technique [6] is</p>		
--	---	--	--

	<p>reputation-based. This protection scheme can stop copyright violations in a cloud environment over multiple data centers.</p> <p>The reputation system enables trusted interactions between cloud users and data-center owners. Privacy is enforced by matching colored user identifications with the colored data objects. The use of content poisoning was suggested to protect copyright of digital content [46]. The security-aware cloud architecture (see Figure 4.14) is specially tailored to protect virtualized cloud infrastructure.</p> <p>The trust of provided cloud platforms comes from not only SLAs, but also from effective enforcement of security policies and deployment of countermeasures to defend against network attacks. By varying security control standards, one can cope with the dynamic variation of cloud operating</p> <p>FIGURE 4.37 HT-based trust overlay networks built over cloud resources provisioned from multiple data centers for trust management and distributed security enforcement</p> <p>conditions. The design is aimed at a trusted cloud environment to ensure high-quality services, including security. The cloud security trend is to apply virtualization support for security enforcement in data centers. Both reputation systems and data watermarking mechanisms can protect data-center access at the coarse-grained level and to limit data access at the fine-grained file level. In the long run, a new Security as a Service is desired. This “SaaS” is crucial to the universal acceptance of web-scale cloud computing in personal, business, community, and government applications. Internet clouds are certainly in line with IT globalization and efficient computer outsourcing. However, interoperability among different clouds relies on a common operational standard by building a healthy cloud ecosystem.</p>			
c.	<p>Explain access control and identity access management.</p> <p>SOLUTION</p> <p>Identity and Access Management (IAM) controls and defines user roles, permissions, and</p>	6	L 1	CO2

		<p>authentication methods.</p> <p>Use Principle of Least Privilege (PoLP) to limit user access rights.</p> <p>Role-Based Access Control (RBAC)</p> <ul style="list-style-type: none"> • Assigns permissions based on predefined user roles. • Used in AWS IAM Roles, Azure RBAC, and Google Cloud IAM. <p>Attribute-Based Access Control (ABAC)</p> <ul style="list-style-type: none"> • Uses attributes (e.g., department, location, device type) to define access policies. • Provides more granular access control compared to RBAC. <p>Multi-Factor Authentication (MFA) and Single Sign-On (SSO)</p> <ul style="list-style-type: none"> • MFA adds an extra layer of security by requiring a second form of authentication (e.g., SMS, authenticator app, biometric). • SSO simplifies user authentication across multiple cloud services using a central identity provider (e.g., Okta, Azure AD, Google Workspace). 			
--	--	--	--	--	--

MODULE - 5

9	a.	<p>Write difference between cloud and grid computing.</p> <p>SOLUTION</p> <table><tr><th>Cloud Computing</th><th>Grid Computing</th></tr><tr><td>A model that provides on-demand network access to a shared pool of configurable computing resources (servers, storage, applications, services).</td><td>A distributed system in which resources of many computers (often geographically dispersed) are used to reach a common goal.</td></tr><tr><td>Managed by cloud service provider (centralized control).</td><td>Managed in a decentralized manner across multiple organizations or systems.</td></tr><tr><td>Provides services as IaaS, PaaS, SaaS.</td><td>Provides computational power and storage for large-scale problems (scientific, research-based).</td></tr><tr><td>Highly scalable; resources can be provisioned dynamically.</td><td>Limited scalability; depends on number of systems connected in grid.</td></tr></table>	Cloud Computing	Grid Computing	A model that provides on-demand network access to a shared pool of configurable computing resources (servers, storage, applications, services).	A distributed system in which resources of many computers (often geographically dispersed) are used to reach a common goal.	Managed by cloud service provider (centralized control).	Managed in a decentralized manner across multiple organizations or systems.	Provides services as IaaS, PaaS, SaaS .	Provides computational power and storage for large-scale problems (scientific, research-based).	Highly scalable; resources can be provisioned dynamically.	Limited scalability; depends on number of systems connected in grid.	6	L 1	CO2
Cloud Computing	Grid Computing														
A model that provides on-demand network access to a shared pool of configurable computing resources (servers, storage, applications, services).	A distributed system in which resources of many computers (often geographically dispersed) are used to reach a common goal.														
Managed by cloud service provider (centralized control).	Managed in a decentralized manner across multiple organizations or systems.														
Provides services as IaaS, PaaS, SaaS .	Provides computational power and storage for large-scale problems (scientific, research-based).														
Highly scalable; resources can be provisioned dynamically.	Limited scalability; depends on number of systems connected in grid.														

		<table><tr><td>Resources accessed via the internet (pay-per-use).</td><td>Resources accessed through grid middleware and protocols.</td><td></td><td></td><td></td></tr><tr><td>Suitable for business applications, web apps, storage, virtualization, enterprise solutions.</td><td>Suitable for scientific computations, simulations, research problems requiring high computing power.</td><td></td><td></td><td></td></tr><tr><td>Utility-based (pay for what you use).</td><td>Often collaborative (shared resources without monetary cost).</td><td></td><td></td><td></td></tr><tr><td>Owned and maintained by cloud provider.</td><td>Owned by multiple organizations or individuals.</td><td></td><td></td><td></td></tr></table>	Resources accessed via the internet (pay-per-use).	Resources accessed through grid middleware and protocols.				Suitable for business applications, web apps, storage, virtualization, enterprise solutions.	Suitable for scientific computations, simulations, research problems requiring high computing power.				Utility-based (pay for what you use).	Often collaborative (shared resources without monetary cost).				Owned and maintained by cloud provider.	Owned by multiple organizations or individuals.						
Resources accessed via the internet (pay-per-use).	Resources accessed through grid middleware and protocols.																								
Suitable for business applications, web apps, storage, virtualization, enterprise solutions.	Suitable for scientific computations, simulations, research problems requiring high computing power.																								
Utility-based (pay for what you use).	Often collaborative (shared resources without monetary cost).																								
Owned and maintained by cloud provider.	Owned by multiple organizations or individuals.																								
b.	<p>Explain the following:</p> <p>i) Server keys computing</p> <p>ii) Edge computing</p> <p>iii) AI/ML in cloud</p> <p>iv) Containerization with Docker and Kubernetes</p> <p>v) Quantum computing in cloud</p> <p>SOLUTION</p> <p>i) Serverless Computing</p> <ul style="list-style-type: none">• Serverless computing is a cloud execution model where the cloud provider dynamically manages the allocation of servers.• Developers only write code as functions (Function-as-a-Service, FaaS) without managing infrastructure.• Example: AWS Lambda, Azure Functions.• Benefits: Auto-scaling, cost-effective (pay only for execution time), faster deployment. <p>ii) Edge Computing</p> <ul style="list-style-type: none">• Edge computing refers to processing data closer to the source of data generation (e.g., IoT devices, sensors) rather than sending all data to centralized cloud servers.• Reduces latency and bandwidth usage.• Example: Smart cameras processing video feeds locally before sending summary data to the cloud.• Benefits: Faster response, reliability in real-time applications (autonomous cars, healthcare monitoring). <p>iii) AI/ML in Cloud</p>	10	L2	CO2																					

		<ul style="list-style-type: none"> Cloud providers offer AI and Machine Learning as services to enable organizations to use predictive analytics and intelligent applications without building complex infrastructure. Example services: Google AI Platform, AWS SageMaker, Azure AI Services. Benefits: Scalability for model training, pre-built APIs for vision, speech, NLP, and cost-effective computing power (GPUs/TPUs on demand). <p>iv) Containerization with Docker and Kubernetes</p> <ul style="list-style-type: none"> Containerization packages an application with all its dependencies into a lightweight, portable container. Docker is a popular containerization tool for creating and running containers. Kubernetes is a container orchestration platform that automates deployment, scaling, and management of containerized applications. Benefits: Portability, efficient resource usage, microservices-based architecture, easy scaling. <p>v) Quantum Computing in Cloud</p> <ul style="list-style-type: none"> Quantum computing uses quantum-mechanical principles (superposition, entanglement) to solve problems that are extremely hard for classical computers. Cloud providers now offer Quantum Computing-as-a-Service (QCaaS). Example: IBM Quantum Experience, Microsoft Azure Quantum, Amazon Braket. Applications: Cryptography, drug discovery, optimization, complex simulations. Benefit: Provides access to powerful quantum machines without owning expensive hardware. 			
	c.	<p>Explain AWS services</p> <p>SOLUTION</p> <p>Amazon Web Services (AWS) is a leading public cloud platform offering a wide range of cloud services to build, deploy, and manage applications.</p> <p>Key categories include:</p> <ul style="list-style-type: none"> Compute: e.g., Amazon EC2, offering scalable virtual server instances quickly configurable to user needs. Storage: e.g., Amazon S3, an object storage service for storing and retrieving any volume of data securely and durably. Database: e.g., Amazon RDS, a managed relational database service that automates tasks like patching, backups, and scaling. Serverless Compute: AWS Lambda, which enables running code in response to events without managing servers. <p>Benefits of AWS include its scalable infrastructure, pay-as-you-go pricing model, global infrastructure, and rich ecosystem of integrated services.</p>	4	L 2	CO2

OR

10	a.	Explain the features of cloud and grid computing SOLUTION Features of Cloud Computing <ol style="list-style-type: none">1. On-demand self-service – Users can provision computing resources (e.g., servers, storage) automatically without human intervention.2. Broad network access – Services are available over the internet and can be accessed from anywhere using standard devices (PC, mobile).3. Resource pooling – Cloud providers pool resources to serve multiple customers using a multi-tenant model.4. Rapid elasticity & scalability – Resources can be scaled up or down dynamically according to demand.5. Measured service (Pay-per-use) – Users pay only for the resources they consume, with usage monitored and billed transparently.6. Virtualization support – Efficient utilization of hardware through virtual machines and containers.7. Reliability & Availability – High uptime, redundancy, and fault tolerance ensured by providers. Features of Grid Computing <ol style="list-style-type: none">1. Resource Sharing – Enables sharing of computational power, storage, and data across multiple distributed systems.2. Geographical Distribution – Resources are often located in different locations but connected via networks.3. Heterogeneity – Grid integrates diverse hardware, software, and networks into a unified system.4. Scalability – Ability to handle large-scale scientific and engineering problems by aggregating resources.5. Parallel Processing – Tasks are divided and processed simultaneously across multiple machines.6. Decentralized Management – No single owner; resources belong to different organizations and are managed autonomously.7. High Throughput Computing – Optimized for executing large numbers of tasks, such as simulations and data analysis.	10	L1	CO2
	b.	Distinguish between AWS, Azure, GCP, IBM cloud.	63	L3	CO3

SOLUTION				
Aspect	AWS (Amazon Web Services)	Microsoft Azure	GCP (Google Cloud Platform)	IBM Cloud
Launch & Focus	Launched 2006; leading IaaS/PaaS provider with the widest global reach and maturity.	Strong hybrid-cloud integration; excels in enterprise ecosystems linked with Windows, .NET, and Office 365.	Early mover in AI/ML and data analytics platforms (TensorFlow, BigQuery); excels in big data processing.	Emphasizes hybrid cloud and enterprise AI (Watson); strong for highly regulated industries.
Compute Services	EC2, Lambda (FaaS) for scalable VM and serverless computing.	Azure Virtual Machines, Azure Functions (serverless), strong PaaS support.	Compute Engine, Cloud Functions, GKE (containers).	IBM Cloud Virtual Servers; container orchestration via OpenShift.
Storage Offerings	S3 (object), EBS (block), EFS (file).	Azure Blob (object), Disk (block), File Storage (managed file).	Google Cloud Storage, plus file and block options; integrated with BigQuery/data services.	IBM Cloud Object Storage for scalable unstructured storage.
Networking & Security	Offers VPCs, CDNs (CloudFront), Direct Connect, IAM, KMS, DDoS protection.	Virtual Networks, ExpressRoute, Azure AD, Security Center, encryption and compliance services.	VPC, Cloud CDN, Interconnect, Cloud IAM, KMS, integrated with Google's secure network backbone.	Software-defined networking, IAM, encryption, and enterprise-grade compliance and governance tools.
Unique Strengths	Market leader with vast service portfolio, global zones, and strong ecosystem/niche tools.	Deep integration with Microsoft ecosystem; excels in hybrid-cloud workloads.	Leading edge in analytics, AI, data processing; managed Kubernetes (GKE).	Focused on hybrid cloud, AI (Watson), and solutions for enterprises needing strict regulatory compliance.

	c.	List out best practices for cloud software development. SOLUTION Best Practices for Cloud Software Development are: <ol style="list-style-type: none"> 1. Design for Scalability & Elasticity – Applications should automatically scale up or down based on workload demand. 2. Use Microservices & Containerization – Break applications into smaller, independent services using Docker/Kubernetes for portability. 3. Ensure Security & Compliance – Implement authentication, encryption, IAM (Identity and Access Management), and follow compliance standards. 4. Automate with DevOps – Use CI/CD pipelines, Infrastructure as Code (IaC) for faster and reliable deployments. 5. Monitor & Optimize – Continuously track performance, availability, and costs using monitoring tools (e.g., CloudWatch, Azure Monitor). 	4	L 3	CO3
--	----	--	---	--------	-----