USN | | | | | | | | | | |

CMRIT CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – October 2024

Sub:	**CLOUD COMPUTING**				Sub Code:	**21CS72**	Branch:	ISE
Date:	15/10/2024	Duration:	90 min's	Max Marks: 50	Sem/Sec:	VII / A, B, C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1. a)	With neat diagram, explain the cloud computing reference model.	6M	CO1	L2
1. b)	Describe the main characteristics and benefits of cloud computing.	4M	CO1	L2
2.	Define Cloud computing. With a neat diagram, explain major deployment models for cloud computing.	10M	CO2	L2
3.	Briefly explain the core technologies that play an important role in the realization of cloud computing.	10M	CO1	L2
4. a)	Explain the characteristics of virtualized environment.	4M	CO4	L2
4. b)	Briefly explain the different hardware virtualization techniques.	6M	CO4	L2
5.	Discuss operating System level virtualization and Application level virtualization.	10M	CO4	L2
6.	Explain in brief full virtualization and para virtualization with its pros and con's.	10M	CO4	L2

Faculty Signature CCI Signature HOD Signature

CLOUD COMPUTING - 21CS72 Academic Year: Odd 2024-25

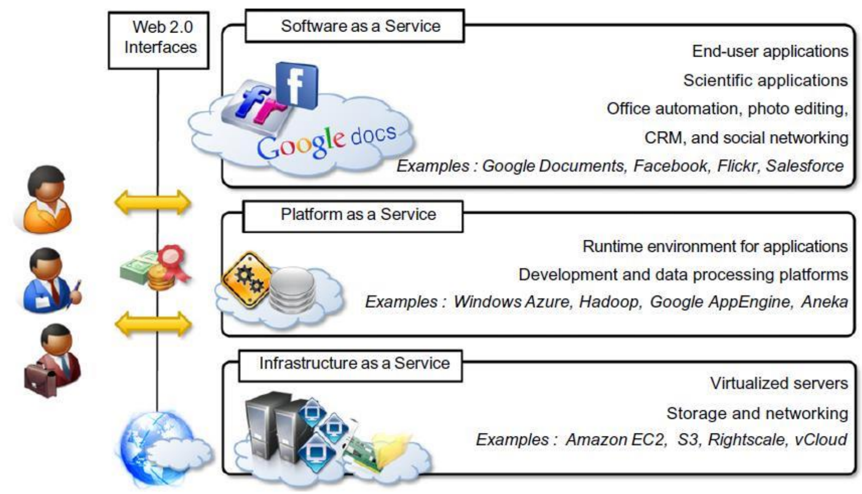
Scheme of Evaluation – IAT1

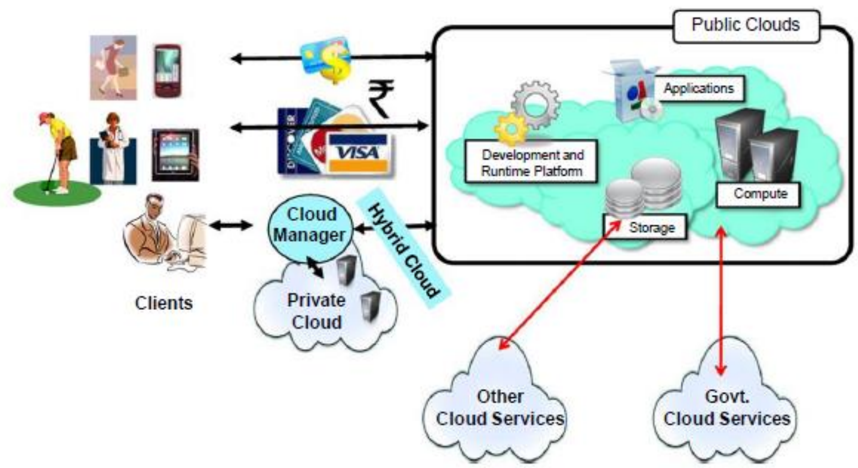
CMR INSTITUTE OF TECHNOLOGY

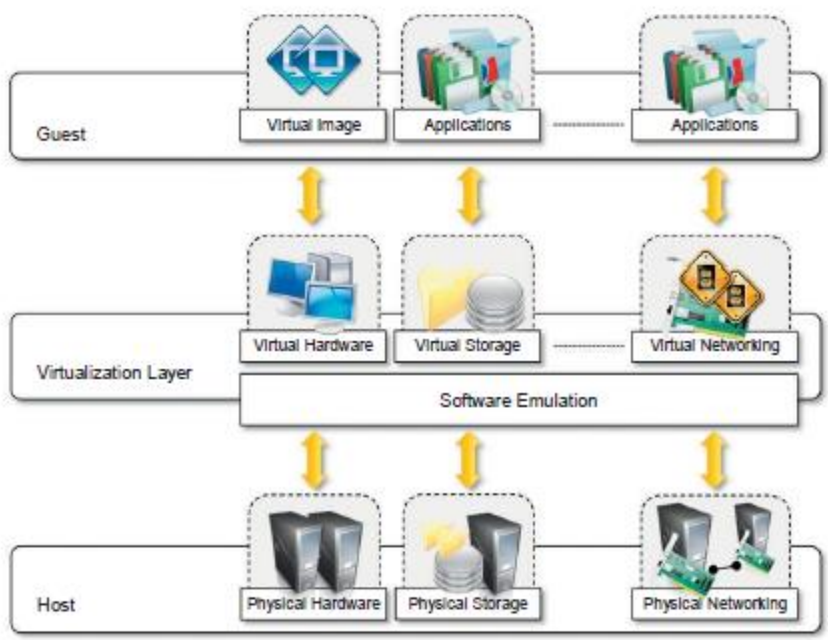
Department of Information Science and Engineering

CLOUD COMPUTING - 21CS72

Scheme of Evaluation – IAT1

Q. No.	Answers	Marks Distribution	Marks
1. a)	**Infrastructure-as-a-Service (IaaS),** **Platform-as-a-Service (PaaS),** **Software-as-a-Service (SaaS).**  FIGURE 1.5 The Cloud Computing Reference Model.	3*2=6M	6M
1. b)	**Characteristics and benefits** Cloud computing has some interesting characteristics that bring benefits to both cloud service consumers (CSCs) and cloud service providers (CSPs). These characteristics are: • No up-front commitments • On-demand access • Nice pricing • Simplified application acceleration and scalability • Efficient resource allocation • Energy efficiency • Seamless creation and use of third-party services	4*1=4M	4M
2.	**Cloud computing** is a technological advancement it is based on the concept of dynamic provisioning, which is applied not only to services but also to compute capability, storage, networking, and information technology (IT) infrastructure in general. Resources are made available through the Internet and offered on a pay-per-use basis from cloud computing vendors.	Definition-2M Diagram-2M	10M

	 The three major models for deploying and accessing cloud computing environments are **public clouds,** **private/enterprise clouds,** **and hybrid clouds**	Explanation-3*2=6M	
3.	five core technologies that played an important role in the realization of cloud computing. These technologies are **distributed systems - ,** **virtualization,** **Web 2.0,** **service orientation,** **and utility computing.** 1.2.1 Distributed systems Clouds are essentially large distributed computing facilities that make available their services to third parties on demand. 1.2.2 Virtualization *Virtualization* is another core technology for cloud computing. It encompasses a collection of solutions allowing the abstraction of some of the fundamental elements for computing, such as hardware, runtime environments, storage, and networking. 1.2.3 Web 2.0 The Web is the primary interface through which cloud computing delivers its services. 1.2.4 Service-oriented computing	2*5=10M	10M

	Service orientation is the core reference model for cloud computing systems. This approach adopts the concept of services as the main building blocks of application and system development. 1.2.5 Utility-oriented computing *Utility computing* is a vision of computing that defines a service-provisioning model for compute services in which resources such as storage, compute power, applications, and infrastructure are packaged and offered on a pay-per-use basis.		
4a.	**Virtualization** is a broad concept that refers to the creation of a virtual version of something, whether hardware, a software environment, storage, or a network. In a virtualized environment there are three major components: *guest*, *host*, and *virtualization layer*. The *guest* represents the system component that interacts with the virtualization layer rather than with the host, as would normally happen. The *host* represents the original environment where the guest is supposed to be managed. The *virtualization layer* is responsible for recreating the same or a different environment where the guest will operate (see Figure 3.1).  FIGURE 3.1 The virtualization reference model.	Diagram – 2M Explanation – 2M	4M
4b.	**Hardware virtualization techniques** **Full virtualization**	2*3=6M	6M

	refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and without any modification, as though it were run on the raw hardware. **Paravirtualization** This is a not-transparent virtualization solution that allows implementing thin virtual machine managers. **Partial virtualization.** Partial virtualization. Partial virtualization provides a partial emulation of the underlying hard- ware, thus not allowing the complete execution of the guest operating system in complete isolation.		
5.	**Operating system-level virtualization** Operating system-level virtualization offers the opportunity to create different and separated execu- tion environments for applications that are managed concurrently. Differently from hardware virtua- lization, there is no virtual machine manager or hypervisor, and the virtualization is done within a single operating system, where the OS kernel allows for multiple isolated user space instances. The kernel is also responsible for sharing the system resources among instances and for limiting the impact of instances on each other. A user space instance in general contains a proper view of the file system, which is completely isolated, and separate IP addresses, software configurations, and access to devices. Operating systems supporting this type of virtualization are general-purpose, time- shared operating systems with the capability to provide stronger namespace and resource isolation. ***3.3.1.4 Application-level virtualization*** Application-level virtualization is a technique allowing applications to be run in runtime environ- ments that do not natively support all the features required by such applications. In this scenario, applications are not installed in the expected runtime environment but are run as though they were. In general, these techniques are mostly concerned with partial file systems, libraries, and operating system component emulation. Such emulation is performed by a thin layer—a program or an operating system component—that is in charge of executing the application.	5*2=10M	10M
6.	**Full virtualization.** *Full virtualization* refers to the ability to run a program, most likely an operating system, directly on top of a virtual machine and without any modification, as though it were run on the raw hardware. To make this possible, virtual machine managers are required to provide a complete emulation of the entire underlying hardware. The principal advantage of full virtualization is complete isolation, which leads to enhanced security, ease of emulation of different architectures, and	5*2=10M	10M

	coexistence of different systems on the same platform. Whereas it is a desired goal for many virtualization solutions, full virtualization poses important concerns related to perfor- mance and technical implementation. A key challenge is the interception of privileged instructions such as I/O instructions: Since they change the state of the resources exposed by the host, they have to be contained within the virtual machine manager. A simple solution to achieve full virtuali- zation is to provide a virtual environment for all the instructions, thus posing some limits on perfor- mance. A successful and efficient implementation of full virtualization is obtained with a combination of hardware and software, not allowing potentially harmful instructions to be executed directly on the host. This is what is accomplished through hardware-assisted virtualization. **Paravirtualization.** This is a not-transparent virtualization solution that allows implementing thin virtual machine managers. Paravirtualization techniques expose a software interface to the vir- tual machine that is slightly modified from the host and, as a consequence, guests need to be modi- fied. The aim of paravirtualization is to provide the capability to demand the execution of performance-critical operations directly on the host, thus preventing performance losses that would otherwise be experienced in managed execution. Partial virtualization. Partial virtualization provides a partial emulation of the underlying hard- ware, thus not allowing the complete execution of the guest operating system in complete isolation. Partial virtualization allows many applications to run transparently, but not all the features of the operating system can be supported, as happens with full virtualization.		