UNIT-1

Introduction to Cloud Computing

Defn [Imp]

Cloud computing is a model of computing that enables the delivery of on-demand computing resources over the internet. It allows users to access a variety of services, such as storage, processing power, software, and databases, without the need for local infrastructure or hardware. Instead of running applications or storing data on a personal computer or local server, cloud computing utilizes remote servers and networks to provide these resources.

The Cloud in cloud computing is a set of hardware, Network, Storage, interfaces etc. that enable the delivery of computing as service. The term "cloud" in cloud computing is used as a metaphor for the internet. When users access cloud services, they are essentially connecting to a network of servers located in data centers around the world.

#Evolution of Cloud Computing:

⇒ One of the first milestones for cloud computing was the arrival of Salesforce.com in 1999, which created the concept of delivering enterprise applications via a simple website.

⇒ The next development was Amazon Web Services in 2002, which provided a group of cloud-based services including storage, computation and even human intelligence.

⇒ Then in 2006, Amazon launched its Elastic Compute Cloud (EC2) as a commercial web service that allowed to rent computer system online to run computer applications.

⇒ Another big milestone came in 2009, as Web 2.0 hit its stride, and Google and others started to offer browser-based enterprise applications, through services such as Google Apps.

⇒ Then came mature virtualization technologies in 2009-tilldate that changed landscape of cloud computing.

#Characteristics of Cloud Computing: [Imp]

i) **On-Demand Service:** Users can arrange computing resources, such as processing power, storage, and applications, without requiring human intervention. Users can demand processing power, storage etc according to their need.

ii) **Broad Network Access:** Cloud computing services are accessible over the network, typically the internet, and can be accessed by various devices from anywhere with an internet connection.

iii) **Measured Service:** Cloud computing services are typically offered on a pay-per-use or usage-based pricing model. Users are charged based on their consumption of resources, such as storage, bandwith, or processing time. This allows for cost optimization and transparency, as users only pay for what they actually use.

iv) **Scalable Architecture:** Users can scale up their hardware, storage and other requirements according to time and need. This scalable architecture allows applications and services to handle increased workloads and maintain performance.

v) **Location Independent:** Cloud Services are location independent, they can be accessed from anywhere with an internet connection.

vi) **Security and Data Protection:** Cloud providers employ robust security measures to protect data and ensure the privacy of users.

#Types of Cloud: [V.Imp] ← OR Cloud Deployment Models

i) **Public Cloud:** It is held by an outsourced cloud provider and is available to numerous enterprises on a pay-per-use basis over the Internet. E.g: AWS, Google Cloud, Microsoft Azure etc.

PROS	CONS
→ Easy Scalable	→ Not the safest option.
→ Cost Effective	→ Sensitive data.

ii) **Private Cloud:** The cloud deployment architecture is controlled by a single company and it provides more regulated environment with more centralized access to IT resources inside the organization.

PROS	CONS
→ Improved level of security	→ Harder to access data from remote.
→ More control over server.	→ Requires IT expertise.

iii) Hybrid Cloud: It is an cloud environment that is suitable alternative for the enterprises that want benefits from both private and public cloud deployment.

PROS
→ Highly flexible & Scalable.
→ Enhanced Security

CONS
→ Communication may be conflicted as both system are used.

iv) Community Cloud: Community Cloud is not widely used but it is like a private cloud that operates similarly as public cloud. This platform is privately maintained either in data center or premises. Government Agencies, Healthcare organizations, Financial corporations use this system.

PROS
→ Scalable
→ Flexible

CONS
→ Shared storage and bandwidth.

Types of Cloud Services: [Imp]

i) Infrastructure as a Service (IaaS): IaaS provides virtualized computing resources over the internet. It allows users to rent and manage virtual machines (VMs), storage, and networking infrastructure. With IaaS, users have control over the operating systems, applications, and configurations of the virtualized infrastructure. Popular examples of IaaS providers include Amazon Web Services (AWS), Microsoft Azure VMs etc.

ii) Platform as a Service (PaaS): PaaS offers a platform for developing, testing, and deploying applications without the need to manage the underlying infrastructure. It provides a complete development and runtime environment, including tools, libraries, and frameworks. PaaS providers handle the infrastructure, scaling, and maintenance tasks, allowing developers to focus on writing code. Examples of PaaS offerings include Heroku, Google App Engine, and Microsoft Azure App Service.

iii) Software as a Service (SaaS): SaaS delivers software applications over the internet on a subscription basis. Users can access and use the software through web browsers or dedicated client applications. SaaS eliminates the need for users to install, maintain, and manage software on their own devices or servers. Popular examples of SaaS applications include customer relationship management (CRM) systems like Salesforce, collaboration tools like Google Workspace, and productivity applications like Microsoft Office 365.

Benefits/Advantages of Cloud Computing: [Imp]

→ High computing capabilities.
→ Reliability due to multiple redundant sites.
→ Scalable and Flexible.
→ Only pay for what we use.
→ No worries for maintenance.
→ Access through various devices (PCs, Mobile, Tablets etc.)
→ Broad Network Access
→ location Independent
→ Security and Data Protection.

Challenges/Disadvantages of Cloud Computing:

→ Internet connectivity is a must needed.
→ Vendor Lock in: Difficulty in transforming data from one vendor to another.
→ Limited control of users.
→ Security is a concern of everyone.
→ Higher ~~ong~~ ongoing operational cost if any firm needs large no. of services.

#Applications of Cloud Computing:

i) Arts: Cloud Computing offers various art applications to quickly and easily design attractive cards, booklets, and images.
E.g: Moo, Vistaprint, Adobe Creative Clouds etc.

ii) Business: Today every organization requires cloud business application to grow their business and ensures it is available 24*7 to their users. Eg: MailChimp, Salesforce, Slack, Paypal etc.

iii) Data Storage and Backup: Cloud Computing allows us to store information on cloud and access this information using internet.
E.g: Box.com, Google Workspace, Mozy etc.

iv) Education: Cloud computing in education sector is becoming very popular as it offers strong virtual classroom environment, ease of accessibility, scalability, secure data storage etc. for greater reach of student. E.g: Google apps for education, Chrome books.

v) Entertainment: Online Games, Video Conferencing apps etc.

vi) Social: Facebook, Twitter, Linkedin etc.

#Cloud Storage:

Cloud Storage is a service that allows users and organizations to store and access data on remote servers over the internet. Instead of storing data on local physical storage devices like hard ~~drivers~~ drives or servers, cloud storage enables users to store their files and data in a remote location, typically maintained by a third-party cloud storage provider.

Cloud Storage is delivered on demand with Just-In-Time capability and cost & eliminates buying and managing own physical data storage infrastructure. This gives us flexibility, scalability, durability, with anywhere anytime data access.

#Cloud Services Requirements: [Imp]

Cloud Computing consists of hardware and software resources made available on the Internet as managed third-party services. Cloud services requirements typically provide access to advanced software applications and high-end networks of server computers. Service providers create cloud computing systems to serve common business or research needs.

Examples of cloud computing services include:

i) Virtual IT: It configures and utilizes remote third-party servers as extensions to a company's local IT network.

ii) Software: They utilize commercial software applications or develop and remotely host custom-built applications.

iii) Network storage: The backup or archive data across the Internet to a provider without needing to know the physical location of storage.

Cloud computing systems in general are designed for scalability to support large numbers of customers and sudden & significant increase in demand. Some customers prefer this model because it limits their manageability burden.

#Cloud and dynamic infrastructure:

Cloud computing refers to the delivery of on-demand computing resources over the internet. It allows organizations to access and use computing infrastructure, such as virtual machines, storage, and networking, without having to invest in and ~~mat~~ maintain their own physical hardware.

A key aspect of cloud computing is its ability to provide a dynamic infrastructure. This means that the resources can be allocated, scaled, and deallocated quickly and easily based on the changing needs of an application or organization. Dynamic infrastructure results in improved scalability, cost-effectiveness, and flexibility.

In cloud environment, the infrastructure is typically virtualized, meaning that physical hardware resources are abstracted and provided as virtual instances. As a result, cloud providers can efficiently manage and distribute resources across multiple customers and applications. Cloud computing and dynamic infrastructure allows businesses to focus on their core competencies without the burden of infrastructure management, enabling them to develop and respond quickly to market demands.

#Cloud adoption: [Imp]

Cloud adoption refers to the process of organizations or individuals integrating cloud computing technologies and services into their existing IT infrastructure or workflows. It involves migrating data, applications, and processes from on-premises systems to the cloud or incorporating cloud services into new projects or operations.

The process of cloud adoption involves several steps, such as judging the existing environment, identifying suitable cloud solutions, planning the migration strategy, and executing the migration process. Successful cloud adoption requires careful planning, collaboration, etc. Cloud adoption can enable organizations to enhance the benefits of cloud computing, optimize resource utilization, and drive innovation in their operations.

Benefits/Advantages of cloud adoption:

OR factors that lead organizations to adopt cloud.

- → Scalability and Flexibility
- → Cost-Effectiveness.
- → Enhanced collaboration and accessibility
- → Reliability and availability.
- → Security and data protection.

UNIT-2

Cloud Computing Architecture

#Platform as service: (PaaS): [Imp]

(Introduction of PaaS already discussed in Unit-1)

Key Features of PaaS: ↔ OR Services of PaaS

→ It offers browser based development environment.
→ It provides built-in security, scalability and web services.
→ It provides built-in tools for defining workflow, approval process, and business rules.
→ It is easy to integrate with other applications.
→ It provides web services ~~inf~~ interface to connect applications outside the platform.

Benefits of PaaS:

→ Minimal Development Time.
→ Multiple Programming Language Support.
→ Enhanced Collaboration.

#Software as service (SaaS): [Imp]

(Introduction of SaaS also already discussed in Unit-1).

Key Features of SaaS:

→ Software and apps are delivered to consumers via a subscription model by SaaS companies.
→ Data is secure in the cloud; failure of equipment does not result in data loss.
→ Resources may be scaled based on requirements.
→ Applications may be accessed from anywhere with internet connected device.

Benefits of SaaS:

→ Affordable.
→ Accessibility from anywhere.
→ Ready to Use.

#Infrastructure as service (IaaS): [Imp]

(Introduction of IaaS also already discussed in Unit-1).

Key features of IaaS:

→ Instead of purchasing hardware completely, users pay for IaaS on demand.
→ Infrastructure is scalable depending on processing and storage needs.
→ Saves eterprises, the costs of buying and maintaining their hardware.
→ Because data is on the cloud, there can be no single point of failure.

Benefits of IaaS:

→ Minimize Costs
→ Enhanced Scalability
→ Simple Deployment.

#Deployment Models of Cloud:

i) Public cloud.
ii) Private cloud.
iii) Hybrid cloud.
iv) Community cloud

} Already discussed in Unit-1 with pros and cons.

In addition you can have a look at detailed diagrams or architectures of these in kee book

#Cloud Computing Architecture:

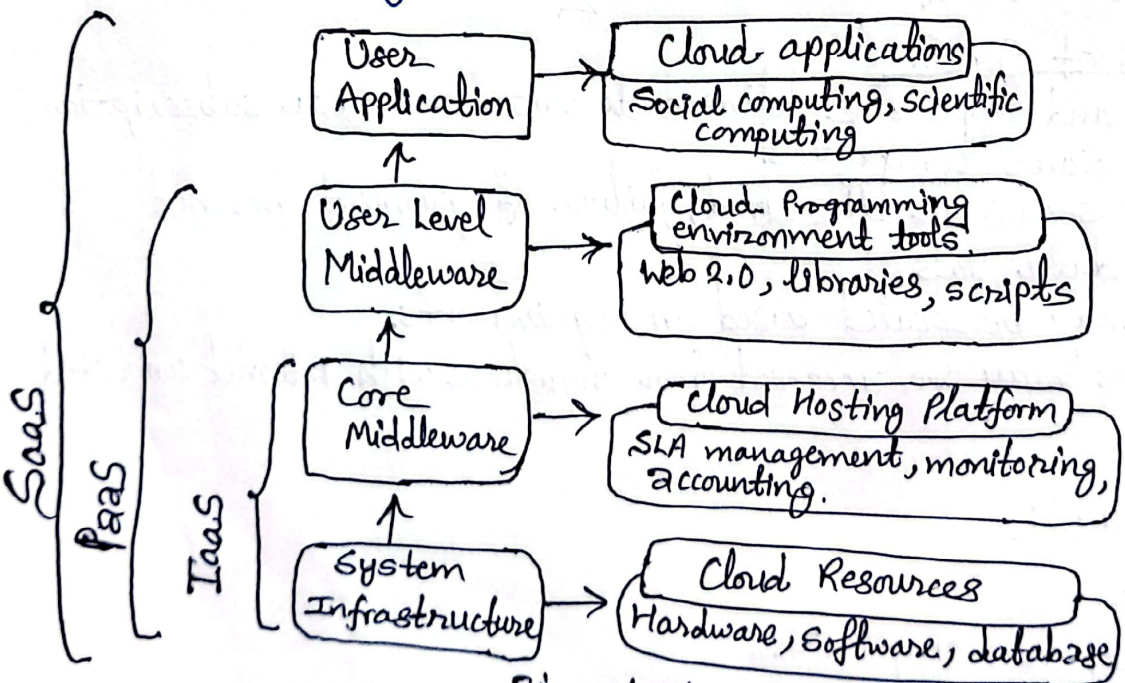


Fig: Cloud Computing architecture.

If asked in exam describe in short about fig above and describe in short about IaaS, PaaS, and SaaS.

#Security: [Imp] Security is defined as preservation of confidentiality, integrity, and availability of information. It involves other properties in addition such as authenticity, accountability, non-repudation and reliability. Security in cloud computing refers to the measures and practices implemented to protect data, applications, and infrastructure in cloud environments. Security measures is crucial to safeguard against potential threats and risks. Organizations must carefully assess and implement appropriate security measures based on their specific requirements and the sensitivity of the data and applications being stored and processed in the cloud. Key aspects of security are: Data encryption, Identity and access management, Vulnerability management, Data backup and recovery etc.

#Trust and Privacy: [Imp]

Trust refers to the confidence, reliance, and belief that one person or entity has in another person or organization. Trust in cloud computing refers to the confidence, reliance, and belief towards cloud services and its infrastructures. Trust is a broader notation than security that involves human psychology, brand loyality and friendliness. Establishing trust is essential to encourage the adoption and utilization of cloud services. Key factors that contribute to trust include: security measures, Data privacy, Service reliability and availability, Data ownership and control etc.

Privacy is the ability to keep information and activities confidential, and protected from unauthorized access. It encompasses the right to control and manage one's personal data, communications and online presence. Privacy in cloud computing refers to the protection and control of personal and sensitive data stored, processed, and transmitted through cloud services. It is important for users to assess the privacy practices and capabilities of cloud service providers. Some key aspects to consider regarding privacy are: Data Confidentiality, Data Ownership and control, Transparency and Auditing, Data Breach Response, Regulatory Compliance etc.

#Cloud Design and Implementation using SOA :

Cloud design and implementation using Service-Oriented Architecture (SOA) involves taking advantage of the benefits of cloud computing to design, deploy and manage services in a distributed environment. SOA is an architectural approach that promotes modular, loosely coupled services that can be reused and composed to build complex applications.

Following are the key steps involved in designing and implementing a cloud solution using SOA:

i) **Identify service boundaries:**
- ↳ Analyze the system requirements and identify logical boundaries for different services.
- ↳ Define responsibilities and scope of each service.

ii) **Define service contracts:**
- ↳ Specify the interfaces and contracts for each service, including message formats, protocols, and service-level agreements (SLAs).

iii) **Implement services:**
- ↳ Services can be implemented using various technologies and languages, such as RESTful APIs, SOAP, or microservices.

iv) **Deploy services in the cloud:**
- ↳ Choose a cloud platform that supports our service deployment requirements. Platforms like AWS, Azure etc.

v) **Implement service governance:**
- ↳ Establish policies and procedures for managing the lifecycle of services.

vi) **Implement service composition:**
- ↳ Compose services to create larger business processes or applications.

vii) **Implement security measures:**
- ↳ Apply appropriate security measures to protect the services and data in the cloud.

viii) **Monitor and manage services:**
- ↳ Implement monitoring and management tools to track the performance, availability, and usage of deployed services.

Differences between Public cloud and Private cloud. [Imp]

Feature	Private Cloud	Public Cloud
Ownership	Owned and operated by a single organization.	Owned and operated by a cloud service provider.
Infrastructure	Dedicated hardware and resources	Shared hardware and resources.
Customization	Highly customizable to specific requirements.	Limited customization options as per provider's offerings.
Accessibility	Accessible within the organizations network or via VPN.	Accessible via the internet from anywhere.
Cost	Higher upfront and maintenance costs.	Pay-as-you-go model, cost based on usage.
Security	Higher level of control and security.	Relies on the security measures provided by cloud service provider.

Cloud Computing Refrence Model (Cloud Computing General Architecture):

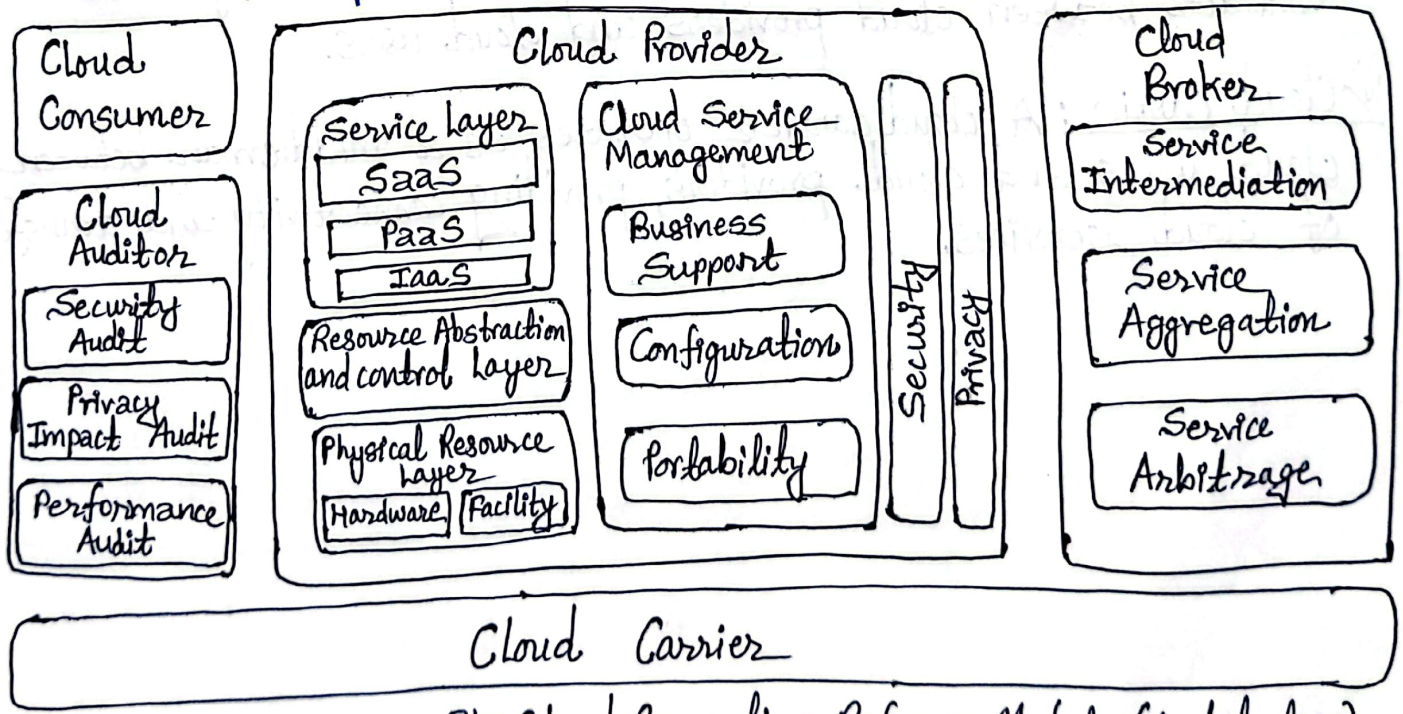


Fig: Cloud Computing Refrence Model (Architecture).

The cloud computing refrence model is a general high-level architecture and is meant to aid understanding of the cloud computing needs, usage, features, and standards.

Above architecture is an overview of the NIST cloud computing refrence architecture, which outlines the primary actor and their cloud computing activities and roles. The NIST cloud computing refrence architecture identifies five primary players:

i) Cloud Consumer: A cloud consumer is a person or organization who has a commercial connection with a cloud provider and consumes it's services. A cloud consumer browses and uses a cloud provider's service and makes payment appropriately.

ii) Cloud Provider: A cloud provider is a person or an organization that is responsible for making a service available to interested parties. Cloud provider deploys, configures, maintains, and upgrades the cloud software and infrastructures.

iii) Cloud Auditor: A cloud auditor is a party who can do analysis of cloud service controls to provide an opinion on them. Audits are carried out to ~~check~~ ensure that standards are met in terms of security measures, performance, privacy etc.

iv) Cloud Broker: A cloud broker is a company that handles the usage, performance, and delivery of cloud services and negotiates contracts between cloud providers and cloud users.

v) Cloud Carrier: A cloud carrier operates as a middleman between cloud users and cloud providers, providing connectivity and transfer of cloud services.

UNIT-3

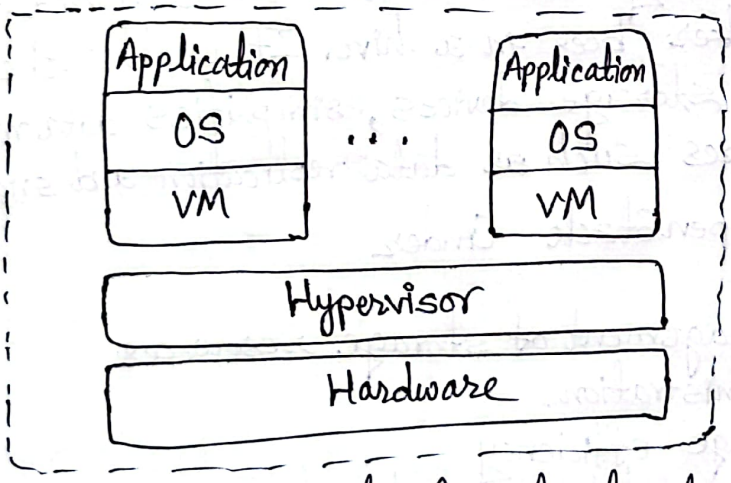
Cloud Virtualization Technology

#Introduction to Virtualization:

Concept Imp!

Virtualization in cloud computing refers to the technique of creating virtual versions or representations of various computing resources, such as servers, storage devices, networks and operating systems. It involves abstracting the physical infrastructure and dividing it into multiple virtual environments that can be used independently by different users or applications.

The virtualization process utilizes specialized software called hypervisors or virtual machine monitors (VMMs) to create and manage virtual machines (VMs). These virtualized entities operate as self-contained units, complete with their own operating systems and applications, while sharing the underlying physical resources.



the term "provisioned" used in VM below refers to act of allocating or providing resources.

these VMM, VM, Container are the components of Virtualization

Fig: Concept of Virtualization.

Hypervisor or VMM: It is responsible for managing the virtualized infrastructure. The hypervisor runs on the physical server and enables the creation, allocation, and monitoring of multiple VMs or containers.

Virtual Machines (VMs): These are the virtualized instances of servers, operating systems, and applications. Each VM runs independently and can be provisioned, migrated, and scaled as needed.

Containers: Containers are lightweight and isolated virtual environments that share the host operating system's kernel.

#Types of Virtualization: [Imp]

OR Server and Machine Virtualization

i) **Server Virtualization:** It involves dividing a physical server into multiple virtual machines (VMs), each running its own operating system and applications. This enables efficient utilization of server resources and allows multiple users or applications to share the same physical server.

Example: VMware vSphere, Microsoft Hyper-V.

Advantages:

- → Higher server ability
- → Cheaper operating costs
- → Eliminate server complexity
- → Increased application performance.

ii) **Storage Virtualization:** Storage virtualization is an array of servers that are managed by virtual storage system. The server aren't aware of exactly where their data is stored and instead function more like worker bees in a hive. It enables the centralization and management of storage devices, simplifies data management, and allows for features such as data replication and snapshotting.

Example: EMC ViPR, OpenStack Cinder

Advantages:

- → Centralized management of storage resources.
- → Simplified administration.
- → Increased storage efficiency
- → Improved data availability and disaster recovery.

iii) **Network Virtualization:** Network virtualization refers to the management and monitoring of an entire computer network as a single administrative entity. Network virtualization allows the creation of virtual networks that can be allocated, managed and configured independently. It provides flexibility, scalability and isolation for different applications sharing the same physical network.

Example: VMware NSX, Cisco ACI

Advantages:

- → Enhanced flexibility, scalability and isolation.
- → Simplified network management.

iv) Desktop Virtualization: Desktop virtualization involves delivering desktop environments to end-users from a centralized server. The user's desktop environment is virtualized and runs on a server, and the user accesses it remotely using thin clients or other devices. Desktop virtualization provides flexibility, security, and simplified management of desktop environments.

Example: VMware Horizon, Citrix Virtual Apps and Desktops.

Advantages:

→ Centralized management and delivery of desktop environments.
→ Reduced administration efforts.
→ Enhanced security
→ Access to desktop environments from various devices.

v) Application Virtualization: This type of virtualization abstracts applications from the underlying operating system and encapsulates them into self-contained units. These units, known as virtualized applications or containers, can run on different operating systems or platforms without the need for installation or conflicts with other applications.

Example: Docker, Kubernetes.

Advantages:

→ Simplified application deployment.
→ Increased portability and compatibility.
→ Efficient utilization of resources.

vi) Data Virtualization: It involves creating a unified view of data from multiple sources, such as databases, files, or web services, without physically moving or copying the data. Data virtualization provides a layer of abstraction that allows applications to access and manipulate data from different sources as if it were in a single location.

Example: Denodo, Oracle Data Virtualization

Advantages:

→ Unified view of data from various sources.
→ Easy data integration and access.
→ Reduced data duplication and redundancy.
→ Real-time data access and analysis.

⊛ Types of Server and Machine Virtualization:

i) Full-Virtualization: In full-virtualization, a hypervisor is installed on the physical server. The hypervisor allows multiple virtual machines (VMs) to run on the server independently, each with it's own operating system (OS) and applications.

ii) Para-Virtualization: In para-virtualization, the guest operating systems are modified to be aware that they are running in a virtualized environment. The guest operating systems communicate with the hypervisor, enabling optimized interactions and resource management.

iii) Hardware-assisted Virtualization: Hardware-assisted virtualization utilizes specific hardware features, such as Intel VT-x or AMD-V, to improve virtualization performance and security.

iv) Container-based Virtualization: It enables the creation and execution of lightweight, isolated containers that share the host OS kernel. Containers provide an efficient and portable way to package and deploy applications.

⊛ Types of Storage Virtualization:

i) Block-Level Storage Virtualization: It operates at the block level of the storage system, abstracting the physical storage devices and presenting them as logical storage volume or blocks. It allows for aggregation and management of multiple storage devices, regardless of their manufacturers.

ii) File-Level Storage Virtualization: It abstracts the underlying physical storage systems and presents them as logical file systems. It enables the integration and centralized management of file-based storage resources, allowing access and movement of files across different storage devices.

#Implementation Levels of Virtualization Structures:

There are 5 implementation levels of virtualization structures:

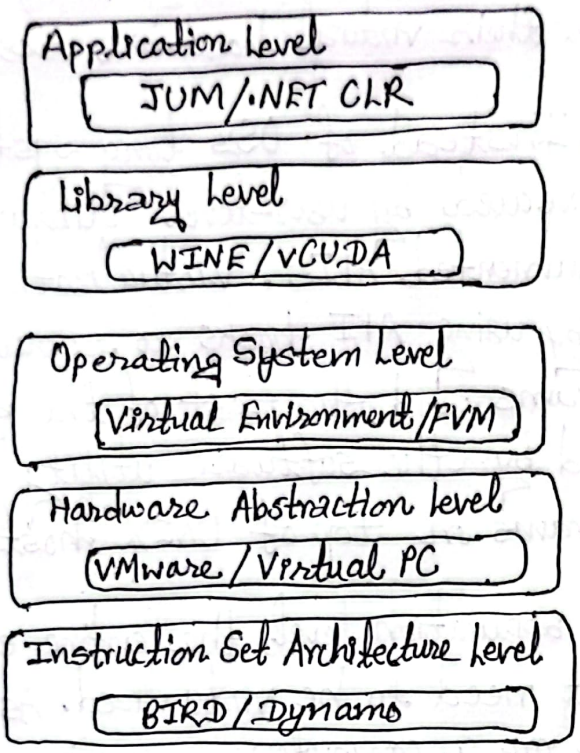


Fig: Levels of virtualization.

1) Instruction Set Architecture Level (ISA): In this level virtualization in ISA is accomplished by ISA emulation. This is useful for running large amounts of legacy code that was orginally developed for various hardware configurations. These programs can be executed on the virtual machine through an ISA. The fundamental emulation, requires the use of an interpreter. This translator translates the source code and turns it into a hardware-readable format for processing.

2) Hardware Abstraction Level (HAL): Hardware-level virtualization occurs directly on the top of the hardware. This method creates a virtual hardware environment for a virtual machine as well as handles the underlying hardware through virtualization. The goal is to virtualize the resources of a computer, such as its processors, memory and I/O devices.

3) Operating System level: The virtualization at operating system level, establishes an abstract layer between the programs and the OS. It functions as a seperate container on the actual server and

operating system, utilizing hardware and software. To allocate hardware resources, OS-level virtualization is often utilized in the creation of virtual environments. Every user gets their virtual environment with their virtual hardware resources.

4) Library Support level: Instead of OS's long system calls, most programs use APIs revealed by user-level libraries as most systems have well-documented APIs. Virtualization using library interfaces is achieved by using API hooks to regulate the communication channel between programs and the rest of the system. This concept has been applied by the software utility WINE to support Windows programs on top of UNIX hosts.

5) Application level: In application level, the entire environment of the platform does not need to be virtualized. As applications run as a single process on a computer's operating system this can also be referred to as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs.

[Imp]
Benefits / Advantages of virtualization:

(Also Ans for Why virtualization is perfect for cloud computing?)

↳ It is cheaper: because it does not need the usage or installation of physical hardware components. No need to allocate significant amount of storage, we just buy license and access it.

↳ It keeps costs predictable: as virtualization is provided by third party and cost is almost fixed.

↳ It reduces workload: Immediate updates to hardware and software are maintained by service providers. This allows individuals to focus on their work reducing workload.

↳ It offers a better uptime: Most service providers guarantee a 99.99% uptime.

↳ It allows for faster deployment of resources.

↳ Easier backup and disaster recovery.

↳ It provides energy savings.

↳ More efficient IT operations.

#Disadvantages of Virtualization:

↳ It can have a high cost of implementation.
↳ It creates security risk.
↳ It creates an availability issue: what if assets are unavailable?
↳ It creates a scalability issue.
↳ It takes time than that of local system.

#Virtualization Software:

Virtual software refers to the technology that enables the creation and management of virtual instances or virtual machines (VMs) on physical hardware infrastructure. It allows for efficient utilization of computing resources by abstracting the underlying hardware and providing a virtual environment for running multiple operating systems and applications.

Virtualization software allows a single host computer to create and run one or more virtual environments. It is most often used to emulate a complete computer system in order to allow a guest OS to be run.
E.g: Linux to run as guest on top of Windows PC.

#Types of Hypervisor (or VMM types): [Imp]

A hypervisor, also known as virtual machine monitor (VMM) is a computer hardware platform virtualization software that allows several OS to share a single hardware host. Each OS appears to have the hosts processor, memory and resources to it. The hypervisor is controlling the host processor and resources, distributing what is needed to each OS and ensure that guest OS. does not disrupt eachother.

Hypervisor is responsible for managing virtualized infrastructure. The hypervisor runs on physical server and enables the creation, allocation, and monitoring of multiple VMs or containers.
There are two types of hypervisors:
1) Type 1 hypervisor (native, bare metal, or embedded hypervisor).
2) Type 2 hypervisor (hosted hypervisor).

Type 1 Hypervisor:

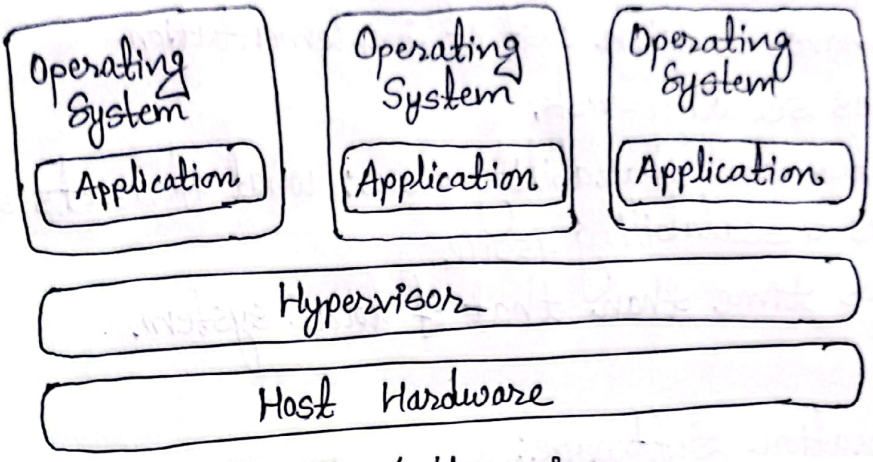


Fig: Type 1 Hypervisor.

Type 1 hypervisor are those that operate directly on system hardware. They are frequently referred to as 'native', 'bare metal', or 'embedded' hypervisor. On the top of this layer we can install many virtual machines. The machines are not connected in any way and can have different instances of OS and acts as different application servers. It is highly secure. Example: Oracle VM, Microsoft Hyper-V etc.

Type 2 Hypervisor:

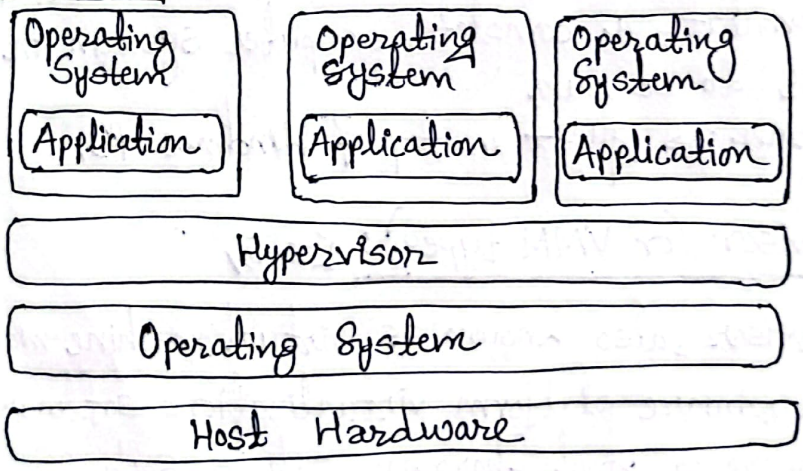


Fig: Type 2 Hypervisor.

Type 2 hypervisor is installed on top of existing OS. It allows users to utilize their personal computer OS serves as a host for VM. Type 2 hypervisors are primarily used on client systems where efficiency is not as important, or on systems where support for a wide range of I/O devices is required. This is less secure than Type 1 hypervisor. Example: VMware Server, Microsoft Virtual PC etc.

#Load Balancing: [Imp]

Load Balancing refers to the process of distributing a set of task over a set of resources with the aim of making their overall processing more efficient. Load balancing can optimize the response time and avoid unevenly overloading some computed nodes while other nodes are left idle. It refers to distributing incomming network traffic across a group of backend servers.

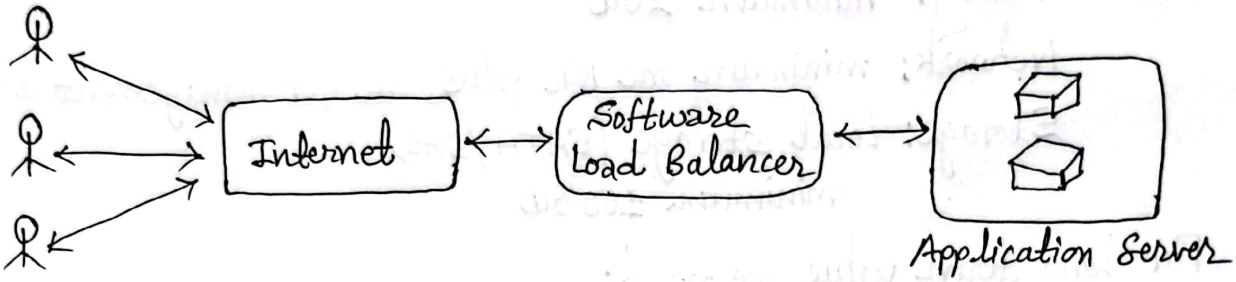


Fig: Load Balancing

Load Balancer performs following functions:

→ Distributes client request or network load efficiently across multiple servers.
→ Ensures high availability and reliability by sending requests only to servers that are online.
→ Provides flexibility to add or subtract servers as demand.

Benefits:

→ Reduced downtime
→ Scalable
→ Flexible
→ Efficient

#Infrastructure requirement for virtualization:

Virtual Infrastructure is a collection of software defined components that make-up an enterprise IT environment. A virtual infrastructure provides same IT capabilities as physical resources, but with software IT team can allocate these virtual resources quickly and across multiple system based on varying needs of enterprise.

Virtualization product have strict requirements on backend infrastructure components including storage, backup, system management, security and time sync. Ensuring that these components ~~are~~ require critical configuration for successful implementation.

Hardware Requirement:

CPU: 4 or more core

Memory: minimum 2GB

Network: minimum one Nic plus, one for management interface.

Storage: Local storage (SATA/SAS)
minimum 100 GB

For long term value we need:

Plan ahead: When designing a VI, IT team should consider how business growth, market fluctation, and advancement in technology might impact their hardware requirement, reliance on computer network and storage resources.

Look for ways to cut costs: Cost cutting initiatives may range from replacing old servers and renegotiating vendor aggrements to automating time consuming server management tasks.

Prepare for failure: IT team should prepare for worst-case scenario by taking advantage of monitoring tools, purchase extra hardware and relying on clusters to better manage host resource.

⊛ Differences between full-virtualization and para-virtualization: [Imp]

Full Virtualization	Para Virtualization
i) Virtual machine permits the execution of instruction with running of unmodified OS in an entirely isolated way.	i) Virtual machine does not implement full isolation of OS rather provides different API which is utilized when OS is subjected to alternation.
ii) Less Secure	ii) More Secure.
iii) Slower	iii) Faster
iv) More portable and compatible.	iv) less portable and compatible.
v) Eg. Microsoft and parallel system	v) Eg. VMware and Xen.

⊛ Differences between Type 1 hypervisor and Type 2 hypervisor. [Imp]

Type 1 Hypervisor	Type 2 Hypervisor
i) A hypervisor that run directly on hosts hardware to control the hardware and to manage guest OS.	i) A hypervisor that run on a conventional OS just as other computer programs do.
ii) Faster	ii) Slower
iii) High performance as there is no middle layer.	iii) Comparatively reduced performance.
iv) Better Scalability.	iv) No so much because it's reliance on underlying OS.
v) Hardware Virtualization	v) OS virtualization.
vi) Eg. Microsoft Hyper-V	vi) Eg. VMware workstation

UNIT-4

Cloud Programming Models

1) Thread Programming: [Imp]

A thread is a sequence of instructions that can be executed independently by a computer's central processing unit (CPU). A thread refers to a lightweight unit of execution within a process.

Processes, in turn, are instances of programs running on a computer. Each process has it's own memory space and resources, and multiple processes can run concurrently on a computer system. Threads share the same memory space and resources within a process.

Definition: Thread programming is a technique that allows multiple threads of execution to run concurrently within a single process. In cloud programming, thread programming refers to the practice of utilizing threads within a program to achieve concurrency and parallelism. Thread programming involves creating and managing threads to perform parallel processing and improve performance, scalability, and responsiveness of cloud-based applications. It enables different parts of an application to execute simultaneously, allowing for efficient utilization of computing resources and faster completion of task.

⊛ How thread programming is done? → Or lifecycle of thread.

Ans: Thread programming involves creating and managing multiple threads within a program to achieve concurrent execution of tasks. In general thread programming is done as follows:

1) **Thread Creation:** Threads are typically created by the operating system or a thread management library provided by the programming language.

2) **Thread Synchronization:** Threads access shared resources, so thread synchronization mechanisms are required to prevent conflicts and ensure data integrity. Common synchronization techniques include locks, mutexes, semaphores, and condition variables.

iii) Thread Execution: The schedular determines the order and duration of execution for each thread based on factors such as priority, time slicing, and the availability of resources.

iv) Thread Communication: This can be achieved through meachanisms such as shared memory, message passing etc.

v) Thread Termination: Threads can be terminated explicitly by the program or automatically by the operating system when they have completed their tasks.

#Task Programming: Imp

Task programming is an approach to parallel programming where tasks, rather than threads, are the fundamental unit of work. Tasks represent individual units of work that can be executed independently and concurrently, allowing for efficient utilization of resources.

In task programming, we typically define tasks and their dependencies, and a task schedular manages the execution of these tasks. The schedular automatically determines the order of task execution based on their dependencies and the availability of resources. Following are the key concepts and steps involved in task programming:

i) Task Definition: Tasks represent discrete units of work that can be executed independently. Tasks can be defined using language-specific constructs or libraries that provide task parallelism support.

ii) Task Dependencies: Tasks can have dependencies on other tasks, A task may need the output of another task as input before it can start its execution.

iii) Task Schedular: The task schedular is responsible for managing the execution of tasks. It analyzes the task dependencies and schedules tasks for execution.

Advantages:

→ No explicit threading, tasks are created, not the threads.
→ Schedules the tasks to the processors so that the code scales to the number of available processors.
→ Multiple algorithms written using task programming model can run at same time without significant performance impact.
→ Complex problems can be solved without requiring traditional synchronization tools such as Mutexes and conditional variable.
→ Task function are simple.

#Map-reduce programming: [Imp]

Map-reduce programming is a model and framework that is used for processing and analyzing large-scale data sets in a distributed computing environment, typically in the cloud. It was first introduced by Google and has become popular for its ability to handle massive amounts of data efficiently.

In the MapReduce programming model, computations are divided into two main stages: the map stage and the reduce stage:

Map Stage: In this stage, the input data is divided into smaller chunks and processed independently by multiple map tasks. Each map task applies a map function to the input data chunk and generates a set of intermediate key-value pairs.

Reduce Stage: In this stage, the intermediate key-value pairs are processed by reduce tasks. Each reduce task applies a reduce function to the grouped intermediate data. The reduce function combines the values associated with each key, performing operations like aggregation, summarization, or generating the final output.

Map reduce is often used for large-scale data processing tasks like batch processing, log analysis, web indexing, and data transformations. It provides a scalable and fault-tolerant approach to big data processing. Various implementations of MapReduce exist, including Apache Hadoop and Apache Spark.

Advantages of Map Reduce:

→ It is fault tolerent, i.e, it can handle the failure.
→ Each node sends status update to master node regularly.
→ MapReduce processes GB's of unstructured huge quantities of data in a matter of minutes.
→ Scalable approach to big data processing.

#Parallel efficiency of Map-Reduce:

Let us assume that the data produced after the map phase is σ times the original data size D. i.e, σD data (postmap). Further we assume that there are P processors. Assume WD is useful work needed to be done.

Now after the map operation, each mapper writes $\frac{\sigma D}{P}$ data to their local disk.

Overheads: $\frac{\sigma D}{P}$

Next this data has to be read by each reducer before it can begin 'reduce operations', so each reducer has to read $\frac{\sigma D}{P^2}$ that is one pth of the data from particular mapper. Since, there are P different reducers, the communication time that a reducer spends getting the data it requires from different mappers as: $\frac{\sigma D}{P}$.

i.e, $\frac{\sigma D}{P^2} \times P = \frac{\sigma D}{P}$

Now, Efficiency of Map-Reduce (EMR) $= \frac{WD}{P\left(\frac{WD}{P} + \frac{2c\sigma D}{P}\right)} = \frac{1}{1+\frac{2c}{W}\sigma}$

Where, $\frac{2\sigma D}{P}$ is extra overhead and c is constant.

Scalable: Efficiency approaches, as useful work per data-item W grows independent of P.

If n document, m words, occuring f times per document on average then $D = nmf$.

The map phase produces mp partial counts:

$$\sigma = \frac{mp}{nmf} = \frac{p}{nf} \quad \text{and} \quad EMR = \frac{1}{1+\frac{2cP}{Wnf}} = \frac{1}{1+\frac{2P}{nf}}$$

⇒here assuming W and c are same.

Hence, we notice that the ratio $\frac{n}{P}$ goes very large, as expected for parallel algorithm

#Enterprise batch processing using Map-Reduce: [Imp]

The data generated by today's enterprises has been increasing at exponential rates in size. Bioinformatics applications mine databases containing terabytes of data. The transaction data of an ecommerce site may exceed millions per month. This volume of data is mined not only for billing purposes, but also to find events, trends, and patterns that help these firms provide better service.

With such a large data volume, it will be difficult for a single server – or node – to handle it. MapReduce is a framework that lets users develop code that runs on numerous nodes without worrying about fault tolerance, dependability, synchronization, or availability. Batch processing is a type of automated task that performs computations regularly. It executes the processing code on a top of inputs known as batch. The task will often read batch data from a database and save the results in the same or a seperate database.

Batch processing is done with the cumulative transactions in a group, once batch processing has begun, no user participation is necessary. The batch processing mechanism processes the data blocks that have previously been stored over some time. Apache Hadoop's MapReduce is the most widely used batch processing framework. Hadoop's MapReduce framework data processing is shown below:

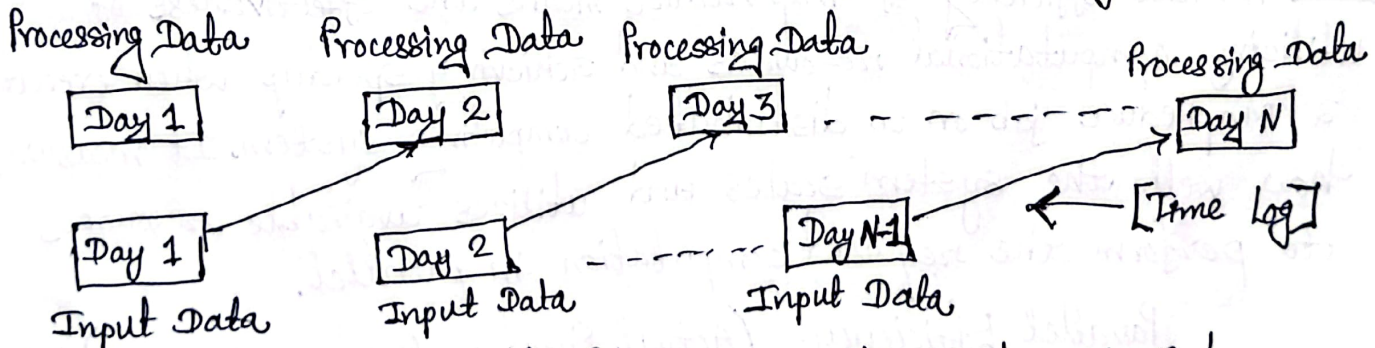


Fig: Data Processing using Hadoop MapReduce.

Batch processing is a way of waiting and performing regularly everything periodically such as the end of day, week, or month. Batch processing framework are great for handling exceedingly big datasets that need a substantial amount of computation.

#Comparisons between Thread, Task and Map Reduce : [Imp]

Feature	Thread	Task	MapReduce
Parallelism	Enables concurrent execution	Enables concurrent execution	Enables distributed execution.
Communication	Shared memory	Message passing	Message passing
Synchronization	Requires explicit synchronization	Requires implicit synchronization	Requires implicit synchronization.
Programming Model	Low-level	High-level	High-level.
Fault Tolerance	No built-in fault tolerance.	No built-in fault tolerance	Built-in fault tolerance.
Scalability	Limited scalability	Moderate scalability	High scalability.
Data Processing	Suitable for small-scale tasks.	Suitable for moderate-scale tasks.	Suitable for large-scale tasks.
Use Cases	Multithreaded Applications.	Asynchronous programming	Big data processing

⊛ What is parallel efficiency of map reduce means?

Ans: Parallel efficiency of map reduce means the effectiveness of utilizing computational resources and achieving speedup when executing a MapReduce job on a distributed computing system. It measures how well the system scales and utilizes available resources to perform the required computation in parallel.

Parallel Efficiency = (Actual Speedup / Number of Processors) * 100

Here, SpeedUp represents the improvement in performance achieved by parallel execution compared to a sequential execution. A parallel efficiency of 100% indicates that parallel execution is utilizing all available resources effictively. However, in practice achieving 100% is challenging.

UNIT-5

Cloud Security

Cloud security refers to the set of measures and practices designed to protect data, applications, and infrastructure in cloud computing environments. As organizations increasingly adopt cloud services, ensuring the security and privacy of their data becomes a critical concern, because the ~~clout~~ cloud is shared resource. Cloud security involves a comprehensive ~~rangle~~ range of rules, technologies, and procedures used to safeguard data, applications, and the related computing infrastructure. Cloud security is subdomain of computer security, network security, and information security in general.

Cloud Security Issues: [Imp]

i) **Data breaches:** Unauthorized access to sensitive data stored in the cloud is a significant concern. Breaches can occur due to weak access controls, inadequate encryption, or insider threats.

ii) **Insecure interfaces and APIs:** Cloud services often provide interfaces and APIs to interact with their platform. If these interfaces and APIs are not properly secured, they can be exploited by attackers to gain unauthorized access or manipulate data.

iii) **Insider threats:** While cloud providers invest heavily in security measures, the actions of authorized users within an organization can still pose a risk. Insiders may intentionally or accidently expose sensitive information or compromise the security of cloud resources.

iv) **Data loss:** Cloud service providers can experience data loss due to factors such as hardware failures, natural disasters, or software bugs. Organizations should have proper backup and disaster recovery strategies to mitigate the risk of permanant data loss.

v) Inadequate identity and access management: Weak identity and access management practices can lead to unauthorized access to cloud resources. This includes issues like poor password policies, insufficient authentication mechanisms, etc.

vi) Shared infrastructure vulnerabilities: Cloud computing environments are built on shared infrastructure, which means that vulnerabilities affecting one customer's data or application can potentially impact others.

Cloud Security Challenges and Risks: [Imp]

Challanges and Issues are same, so if asked cloud security challenges we will write points of cloud security issues that we just studied before this topic.

Cloud Security Risks:

i) The effect on Company's ROI: A security breach or data loss can have a significant impact on a company's return on investement (ROI). These looses can include direct costs such as legal fees, potential regulatory fines etc. Indirect costs may arise from reputational damage, customer churn etc.

ii) Compatibility: Going to cloud, may not be compatible with organization's current technology. It should be studied properly.

iii) Trust: Customers entrust their sensitive data to businesses that utilize cloud services. If a company experiences a data breach or security incident, it can erode customer trust.

iv) Reputation: A security breach or data loss can damage a company's reputation in the market.

v) Account hijacking: If an attacker gains control over a user's cloud account credentials, they can misuse the account to manipulate resources, access sensitive data, or launch further attacks.

#Software-as-a-Service Security:

Software-as-a-Service (SaaS) security refers to the measures and practices implemented to protect the security and privacy of data and application in a SaaS environment. SaaS is a cloud computing environment or model where software applications are hosted and provided to customers over internet, eliminating the need for on-premises infrastructure and software installation.

Future cloud computing models will most likely integrate the usage of SaaS. As a result of shift to cloud computing, new business models emerge, then there will be need for new security requirements and concerns. To avoid losing their data, companies or end-users will need to examine vendor rules on data security. The security risks to be discussed with cloud computing vendor are;

i) Compatibility: Migration to the cloud may cause compatibility issues with an existing IT infrastructure, as well as with a company's security needs and organizational regulations.

ii) Trust: Not all service providers are created equal. Unforeseen incidents may cause disruptions to cloud computing services.

iii) Security: The entire structure should be evaluated. Where will your data be stored? Who will have access to data? What security and protection does the cloud provider provide? etc.

iv) Lack of control over performance: There is always the possibility that the system quality is unable to deliver excellent services at all times.

v) Lack of control over the quality: A company must have faith in the quality standards that a provider can supply over time.

#Security Monitoring: [Imp]

Security monitoring is the process of actively monitoring systems, to detect and respond to security events and incidents. The goal of security monitoring is to identify potential security threats, breaches, or suspicious activities in real-time, allowing timely investigation and response to mitigate the impact of an incident.

Centralized security information management systems should be utilized to offer security vulnerability verification and to continiously monitor systems using automated ~~or~~ methods to identify possible concerns. Some of the key aspects of security monitoring are as follows:

i) Log monitoring: Security monitoring involves analyzing logs generated by various systems and devices, such as servers, firewalls, intrusion detection systems, and antivirus software.

ii) Intrusion Detection and Prevention Systems: IDS/IPS solutions monitor network traffic in real-time, searching for patterns or signatures that indicate potential attacks or security breaches.

iii) Threat Intelligence: Threat intelligence feeds and databases helps security teams identify and respond to emerging threats more effectively.

iv) Continious Monitoring: Security monitoring should be done continiously. It is an ongoing process, not a one-time activity.

v) User Behaviour Analytics (UBA): UBA focuses on analyzing user behaviour patterns to identify anomalous activities that may indicate insider threats or compromised accounts.

#Security Architecture Design: [Imp]

A security architecture design in cloud should be established with consideration of processes such as enterprise authentication and authorization, access control, confidentiality, integrity, accountability, privacy, availability etc.

The development of a secure architecture gives engineers, data centers operations staff, and network operations staff a standardized blueprint for designing, building, and testing the security of applications and systems. Following are the principles to be ensured during cloud security architecture design:

i) Vulnerability Assessment: Vulnerability assessment categorizes network assets to better prioritize vulnerability-mitigation initiatives such as patching and system upgrades. It assesses the success of risk mitigation by establishing objectives such as reduced vulnerability exposure and faster mitigation.

ii) Data Privacy: A risk assessment, as well as a gap analysis of controls and processes, must be conducted. Formal privacy processes and activities must be defined and maintained based on this data. Privacy controls and protection, like security, must be incorporated into the secure architectural design.

Data, Application, & Virtual Machine Security detail में देखें better

iii) Data Security: Enterprises will need to bring security to the data level to ensure that their data is secure wherever it travels. It can also compel the encryption of particular types of data and restrict access to the data to only specific people.

iv) Application Security: The security features and requirements are defined here, and the application security test results are examined. Application security methods, secure coding rules, training etc. are are often developed together by the security and development teams.

v) Virtual Machine Security: Firewalls, intrusion detection and prevention, integrity monitoring, and log inspection may all be implemented as software on virtual machines to boost server and application protection. By applying this conventional line of protection to the virtual machine itself, we can safeguard the migration of essential programs and data to the cloud.

#Legal Issues and Aspects:

i) Data Privacy and Security: When storing data in the cloud, organizations must ensure compliance with applicable data protection laws and regulations.

विस्तार से

ii) Jurisdictional Issues: The global nature of cloud services can raise jurisdictional issues. Data stored in the cloud may be subject to the laws and regulations of multiple countries, which can conflict with one another.

iii) Cross border transfer of data: Before doing cross border transfer of data, the ~~user~~ user should be noticed, provided choice, confirmed safety, data integrity, access etc.

iv) Data Ownership and Control: Determining ownership and control of data stored in the cloud can be complex. Organization should carefully review their agreements with cloud service providers to understand rights and responsibilities regarding data ownership and control.

v) Compliance and Regulatory Requirements: Organizations operating in specific industries or regions may have industry-specific or regulatory compliance requirements that must be met when using cloud services.

Multi-tenancy issues:

Multi-tenancy means multiple customers of a cloud vendor are using the same computing resources. Despite the fact that they share resources cloud customer aren't aware of each other and their data is kept totally separate. It helps to better use of resources and lower costs. But, it has following some issues:

i) Security: There is always a risk of data loss, data theft, and hacking.

ii) Performance: SaaS application are at different places, and it affects the response time. It usually takes longer to respond and are much slower than server applications.

iii) Less powerful: It lacks many essential computing features which makes it less powerful.

iv) Noisy neighbour effect: If tenant uses a lots of computing resources, other tenants may suffer because of their lowered computing power.

v) Monitoring: Constant monitoring is vital for cloud service provider to check if there is an issue in multi-tenancy system. If any problem arises, it must get solved immediately without disturbing system efficiency.

UNIT–6
Cloud Platforms and Applications

#Web services: ↔ OR Web Server

Web service refers to the software system designed to support interoperable machine-to-machine interaction over the computer network and World Wide Web (www). A server running on the network listens for requests at certain port and serves web content such as HTML, JSON, XML etc. Web services are client-server applications that interact using the HTTP over www.

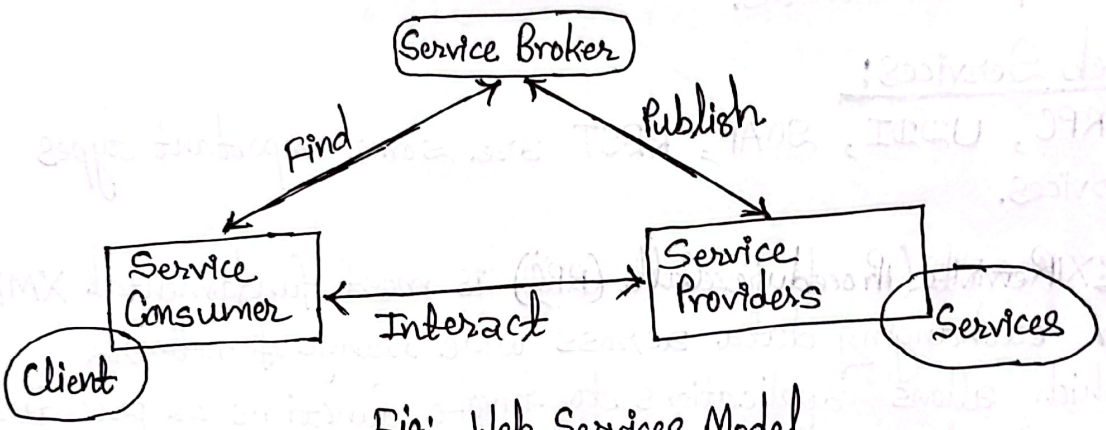


Fig: Web Services Model.

A key aspect of web services is that applications developed in different languages connect by transferring data over a web service between clients and servers. A client invokes a web service by submitting an XML request, and the service responds with an XML response. A web service must perform following functions:

→ Accessible through the Internet or intranet networks.
→ An XML communications protocol that is standardized.
→ Self-description using the standard XML language.
→ Discoverable through a simple location method.

Characteristics:

→ Web services are XML-Based.
→ Web service is loosely coupled.
→ Web service can be synchronous or asynchronous.
→ Web service supports Remote Procedure Calls (RPCs)
→ Web service supports Document Exchange.

Benefits/Advantages:

→ Reduces development time.
→ Simpler Infrastructure, which helps to realize greater ROI.
→ Web services provide Interoperability ie, applications developed in one programming language can communicate with the application developed in other programming languages.
→ Web services interact via regular Internet technology.
→ It allows effective technology distribution over a complete network in a B2B business where both parties understand how the process works.

Types of Web Services:

XML-RPC, UDDI, SOAP, REST are some important types of web services.

XML-RPC: XML-RPC (Remote Procedure Call) is most fundamental XML protocol for exchanging data across wide range of network devices, which allows applications to invoke functions or procedures across a network.

XML-RPC performs RPCs using XML messages. In the XML request, the XML-RPC client gives a procedure name and arguments, and the server delivers either a fault or a response in the XML response. This platform-independent web service uses HTTP to transmit data and communicate other information from the client to the server in a timely and efficient manner.

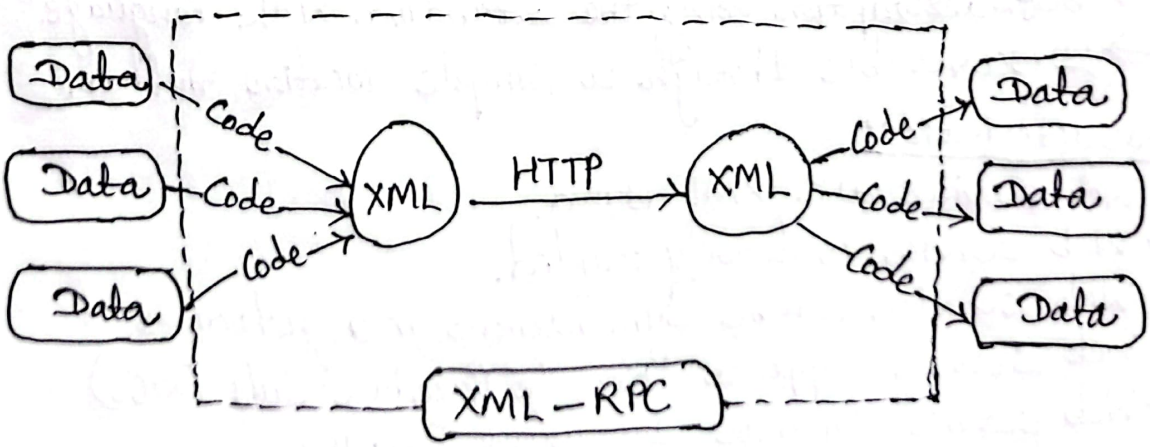


Fig: XML-RPC

ii) **UDDI:** Universal Description, Discription, and Integration (UDDI) is a XML-based standard used to describe, publish, and discover web services. UDDI is platform-independent and open framework. It serves as an online register ~~firm for~~ for firm across the world. The major purpose is to improve the efficiency of digital transactions and e-commerce across enterprise platforms. UDDI communicates via SOAP, COBRA, and the JAVA RMI Protocol. UDDI uses WSDL to describe web service interfaces.

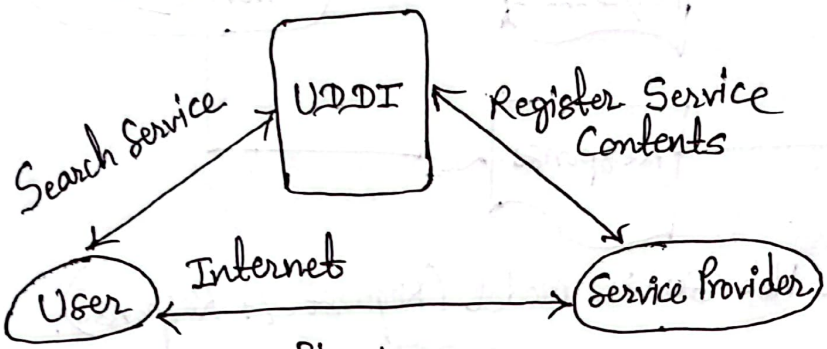


Fig: UDDI

iii) **SOAP:** SOAP stands for Simple Object Access protocol. It is a W3C standard, XML-based. Web service protocol for exchanging structured data and documents through HTTP or SMTP. As, it is language independent, it enables separate programs running on different platforms to communicate using XML. It is a platform-independent format for sending messages. SOAP also uses Web Service description model using WSDL (Web Services Description Language) documents.

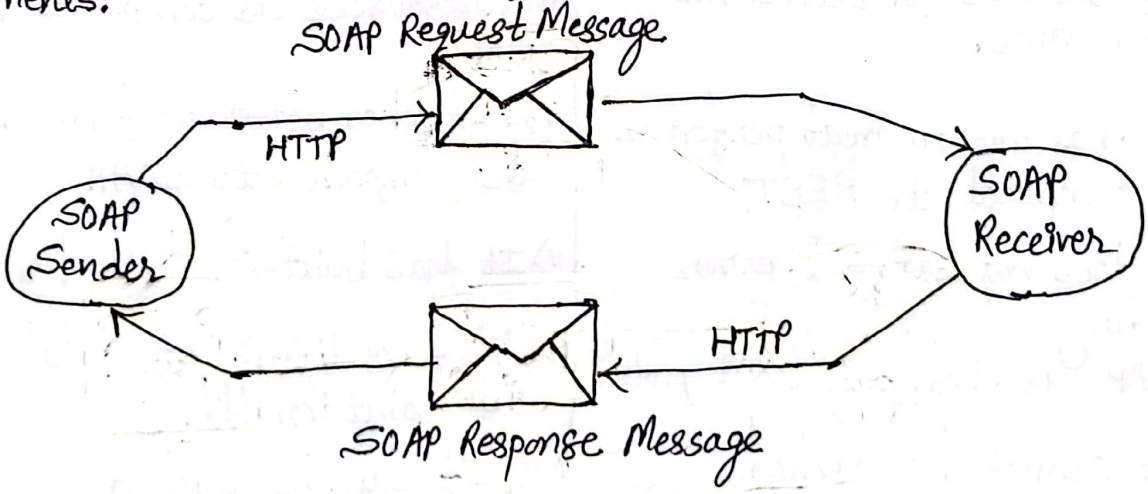


Fig: Basic of SOAP Communication.

iv) REST: REST (Representational State Transfer), is an architectural paradigm in which each unique URL represents a distinct object of some type. REST allows communication and connections between devices and the Internet for API-based operations. A REST web service communicates through HTTP and supports various HTTP methods, including GET, POST, PUT, and DELETE. It also provides basic CRUD functions.

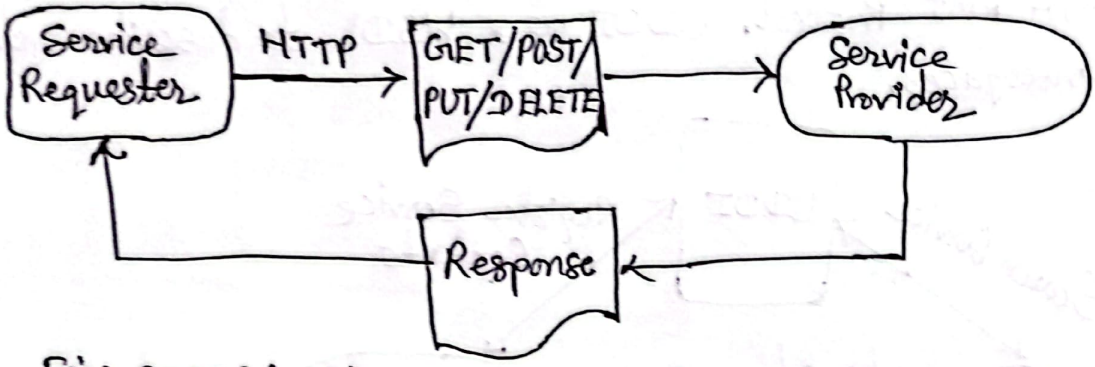


Fig: RESTful Web Service Model (Request & Response).

#SOAP vs. REST [Imp]

SOAP	REST
i) It stands for Simple Object Access Protocol.	i) It stands for Representational State Transfer.
ii) It is an XML-based messaging protocol.	ii) It is not a protocol. It is an architectural style for distributed hypermedia systems.
iii) It needs more bandwith and resources for better web performance.	iii) REST requires less bandwidth and resources as compared to SOAP.
iv) It has not a great performance as compared to REST.	iv) It has better performance as compared to SOAP.
v) It does not support error handling	v) It has built-in error handling.
vi) SOAP is heavyweight XML protocol.	vi) Rest is lightweight, scalable, and maintainable.
vii) It cannot be cached	vii) It can be cached.

#AppEngine: [Imp]

AppEngine also called Google App Engine (GAE) is a cloud computing platform as a service (PaaS) that allows to create and run web apps on Google managed data centers. App Engine provides automatic scaling for web applications, which means that when the number of requests for an application rises, App Engine automatically assigns more resources to the web application to accommodate the increased demand. AppEngine helps to build highly scalable applications which generally supports apps written in Go, PHP, Java, Python, Node.js, .NET, and Ruby. The service is free up to a particular number of consumed resources. The most impressive feature of GAE is its capacity to handle built-in apps in Google's data centers.

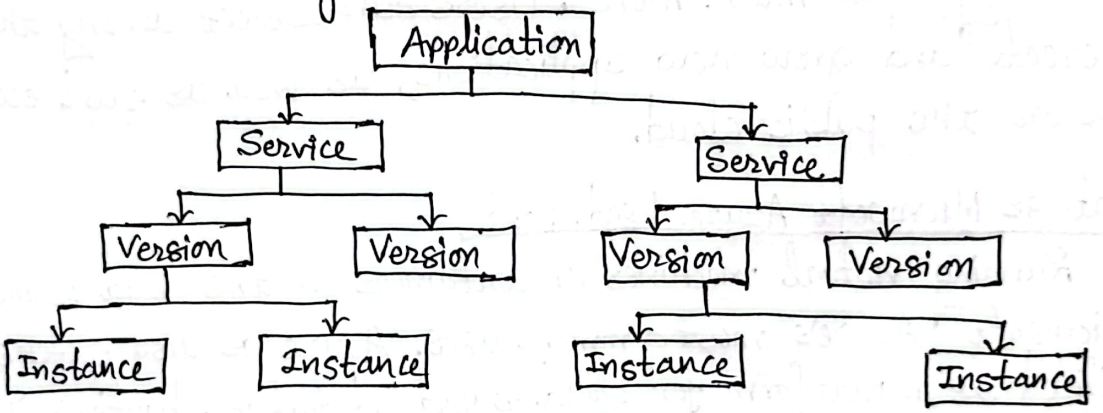


Fig: The App Engine architecture in cloud computing.

Services Provided by App Engine:

→ PaaS to build and deploy scalable applications.
→ Hosting facility is fully managed data centers.
→ A fully managed, Scalable environment or platform.
→ Support in the form of popular development languages and developer tools.

Features of App Engine:

→ Supports wide range of programming languages
→ Fully managed and scalable.
→ Pay-Per-Use
→ Reliable diagnostics services to help to identify and resolve bugs.
→ Traffic segmentation which automatically routes incomming traffic to different versions of the apps.

Benefits/Advantages:

→ High Availability.
→ User Friendly Platform.
→ Wide Range of APIs.
→ Enhanced Scalability
→ Pricing Intelligence
→ Increased Savings.

#Azures Platform: [Imp]

Microsoft Azure Platform (MAP), originally known as Windows Azure, is a public cloud computing platform developed by Microsoft. It provides solutions such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These services can be used for analytics, virtual computing, storage, networking and much more. Users can select among these services to create and grow new applications, as well as run existing apps on the public cloud.

What is Microsoft Azure Used For?

- Running virtual machines or containers in the cloud is one of Microsoft Azure's most common uses. Azure is also extensively utilized as a platform for cloud-based database hosting. This platform is also commonly used for disaster recovery and backup.

Azure products and services:

i) Computer: Allows users to deploy and manage VMs, Containers, and batch tasks.
ii) Mobile: Aid developers in the creation of cloud apps for mobile.
iii) Web: Aid in the creation and deployment of web apps.
iv) Storage: Offers scalable cloud storage for organized and unstructured data.
v) Analytics: Realtime analytics like ML, IoT, data warehousing etc.
vi) Networking: Comprises of virtual network, traffic management etc.
vii) Identity: Guarentees that only authorized users can access services.
viii) IoT: Assists customers in collecting, monitoring, and analyzing IoT data from sensors and other devices.

#Aneka: [Imp] → OR Aneka Container

Aneka is a cloud-based platform and framework for constructing distributed applications, developed by Manjarasoft. Aneka serves as an Application Platform as a Service for Cloud Computing. Aneka provides a variety of programming paradigms such as Task Programming, Thread Programming, and MapReduce Programming. It also provides tools for quick application development and smooth deployment on private or public clouds to distribute applications.

Aneka makes use of the idle CPU cycles of a heterogenous network of desktop PCs, servers, and data centers. Aneka provides a comprehensive set of APIs for developers. Aneka technology primarily consists of two key components:

→ SDK (Software Development Kit) containing application programming interfaces (APIs) and tools essential for rapid development of applications.

→ A Runtime Engine and Platform for managing deployment and execution of applications on private or public clouds.

Types of services hosted in Aneka container:

i) Web applications: Aneka containers can host web applications built with different technologies such as Java, .NET, PHP, Python and more. These containers provide a runtime environment for executing web applications and handling HTTP requests.

ii) Microservices: Aneka supports the deployment of microservices, which are small, independent, and loosely coupled services that work together to form larger applications.

iii) Database services: Aneka containers can host databases, allowing for efficient data storage and retrival.

iv) Content Management Systems (CMS): Aneka containers can host content management systems like WordPress, Durpal, or Joomla.

#Open Challenges of Cloud:

i) Security and Privacy: Customers need to trust that their data is protected from unauthorized access, data breaches, and other security threats. Cloud providers must implement robust security measures, encryption tools, privacy policies etc. to ensure the confidentiality and integrity of customer data. Security remains a significant concern in the cloud.

ii) Data Governance and Compliance: Cloud computing involves storing and processing vast amounts of data. Ensuring compliance with data protection regulations can be challenging.

iii) Performance and Reliability: Cloud services need to deliver consistent performance and reliability to meet the demands of users and applications. Issues like latency, network congestion can impact performance and reliability.

iv) Cost Management and Optimization: Managing cloud costs can be complex. Organizations need to optimize resource allocation, monitor usage, and choose appropriate pricing models to avoid unexpected expenses.

v) Data Transfer and Bandwidth Limitations: Transfering large volumes of data to and from the cloud can be time-consuming and costly, especially when dealing with limited bandwidth or data transfer fees.

Applications of Cloud Computing: [Imp]

1) Scientific Applications:

i) Healthcare: ECG1 analysis in Cloud: In Healthcare field, Cloud applications are being used to support scientists in developing preventive solutions for diseases.

The ECG1 machine measures the electrical activity of hearts Cordium and as a result, a waveform of heartbeat is created. Here, Cloud Computing technology allows the remote monitoring of patient heartbeat data and the patient at risk can be constantly monitored without going to hospital for ECG1 analysis. Also, doctors will be instantly notified of cases that need their attention.

ii) Biology: Protein Structure Prediction (PSP): A protein is made up of peptide bonds that connect lengthy sequences of amino acids. Proteins primary structures are created first, secondry, tertiary, & quaternary from fundamental structure.

Protein structures predictions are made in this manner and it employs a variety of technologies including neural network, AI, ML, bioinformatics and many more.

The use of Cloud Computing in PSP is because PSP requires high computing capabilities and extensive I/O operations. Structure Prediction is computationally tough task and also help in making 3D model structures very fast than manual.

iii) Geo Science: Satellite Image Processing: Massive volumes of geographic and non-spatial data are collected, produced, and analyzed by geoscience applications. To process this large volume of data, the more advanced technology is needed.

The fundamental component of geoscience is GIS and all sorts of spatially linked ~~objects~~ data may be captured, stored, manipulated, analyzed, managed, and presented using GIS application. Cloud Computing offers the necessary infrastructure to support these types of applications and several technologies are integrated to it.

That huge amount of data (in TBs) should be stored in cloud because it is difficult to do so in physical storage devices which is less reliable, expensive, and difficult to manage.

2) Business and Consumer Applications:

i) CRM and ERP: CRM (customer relationship management) and ERP (enterprise resource planning) systems are two industry categories that are booming in the cloud. Cloud CRM programs provide small businesses and start-ups a terrific way to get fully working CRM software with small expenditures and monthly fees.

Cloud based ERP systems are less mature, and they must compete with well established inhouse solutions. Their purpose is to give a unified prespective and access to all activities required to keep a complicated company running.

Social Networking: In previous several years, social networking programs have evolved to become the most popular websites on the Internet. Twitter and Facebook have used cloud computing technology to maintain their traffic and smoothly serve millions of users. For social networks that are continually growing their user base, cloud has the ability to expand while systems are operating in the most appealing aspect, with this feature of cloud social networks can start with low cloud resources at their initial phase and can increase resources as number of users go on increasing. This helps in cost management as well as efficient use of resources.

⊛ What is bucket in Amazon simple storage service? How addressing of a bucket is done? [Model set question. Q.No. 9] [Imp]

Ans: In Amazon Simple Storage Service (S3), a bucket is a container for storing objects (files) in the cloud. It acts as the top-level folder or directory within S3, and we can think of it as a logical unit where we can store our data. Each bucket has a globally unique name, which means that no two buckets in S3 can have the same name.

To address or access a bucket in Amazon S3, we need to use its unique bucket name in the URL. The general format for addressing a bucket is:

http://<bucket-name>.s3.amazonaws.com

For example if we have a bucket named "my-bucket", we would address it as:

http://my-bucket.s3.amazonaws.com

We can also use the AWS Management Console, AWS SDKs (Software Development Kits), or AWS CLI (Command Line Interface) to interact with S3 buckets, perform operations like creating, deleting, or modifying buckets, and upload or retrive objects from the bucket.