Read Less, Learn More..!

CLOUD COMPUTING

BY - ANSHIKA CHAUDHARY

KRAZY KAKSHA

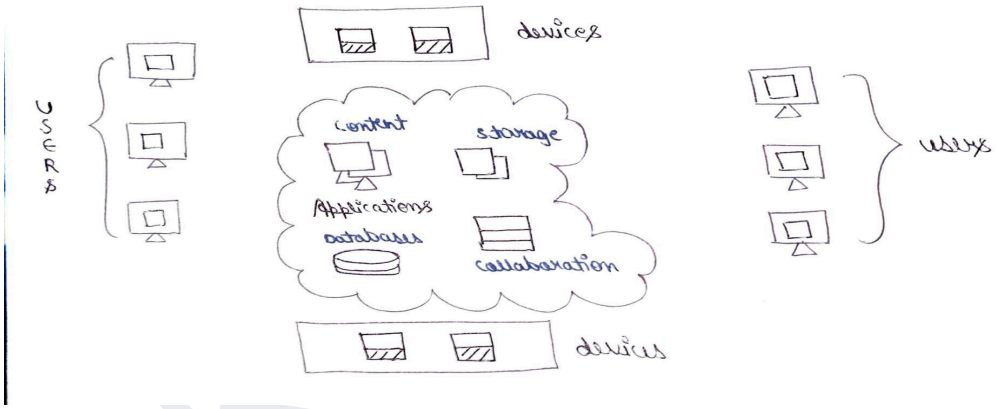
Unit 1: Introduction To Cloud Computing

1. Definition of Cloud Computing

- **Cloud Computing** is the delivery of computing services like storage, software, and processing power over the internet, allowing you to access and use them on-demand without needing physical hardware.

2. Cloud Characteristics:

- **On-demand Self-service:** Users can provision resources as needed, without requiring human intervention.
- **Broad Network Access:** Cloud services are available over the network, accessible from a variety of devices.
- **Resource Pooling:** Cloud resources are pooled to serve multiple customers using multi-tenant models.
- **Rapid Elasticity:** Resources can be rapidly and elastically provisioned to scale up or down as needed.
- **Measured Service:** Cloud resources are metered, and customers only pay for what they use, often based on usage patterns.



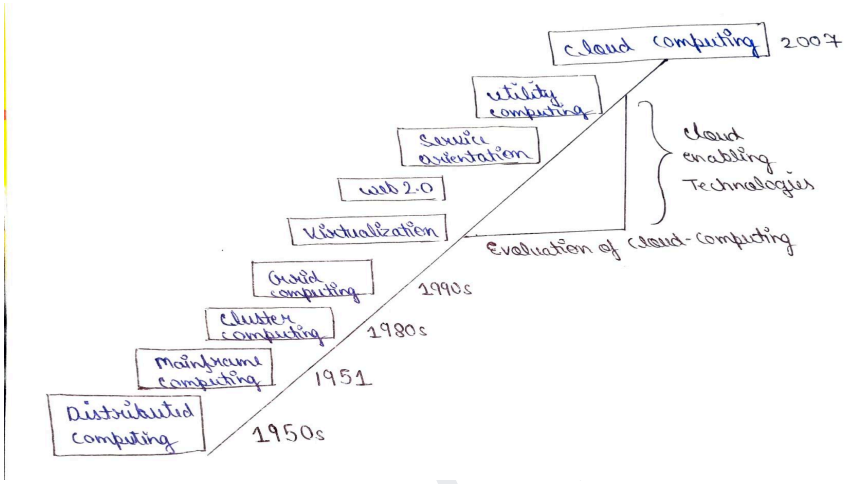
3. Advantages of Cloud Computing:

1. **Cost Efficiency**: No upfront hardware investment; pay-as-you-go model.
2. **Scalability**: Easily adjust resources based on demand.
3. **Accessibility**: Access services from anywhere with an internet connection.
4. **Automatic Updates**: Providers manage software and security updates.
5. **Disaster Recovery**: Built-in backup and recovery options.

4. Disadvantages of Cloud Computing:

1. **Security Risks**: Potential exposure of data to breaches.
2. **Downtime**: Service outages can disrupt access.
3. **Limited Control**: Less control over infrastructure and management.
4. **Data Privacy**: Concerns about data laws and privacy in remote locations.
5. **Ongoing Costs**: Subscription fees can add up over time.

5. Evoluation of Cloud Computing:

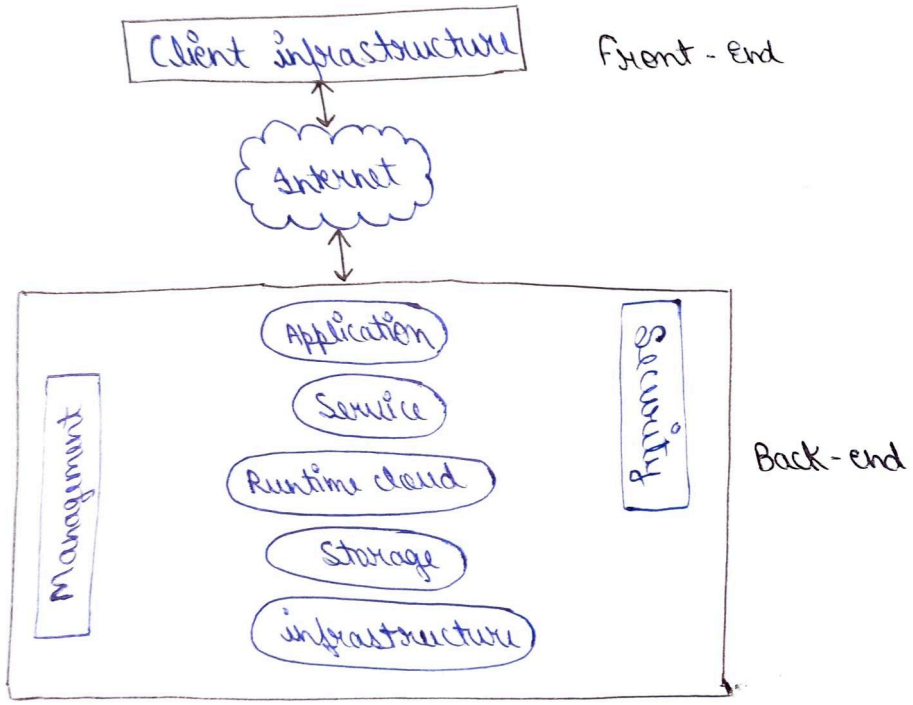


1. **Distributed Computing (1950s)**:
   Early computing systems used multiple machines to share workloads, allowing parallel processing.
   Problem: It was challenging to manage, and network speeds were limited, making it difficult to scale resources efficiently.
2. **Mainframe Computing (1960s-1970s)**:
   Mainframe computers centralized computing power, enabling large organizations to serve many users at once.
   Problem: Mainframes were expensive, required dedicated space, and lacked flexibility, making them inaccessible for smaller businesses.
3. **Cluster Computing (1980s)**:
   Multiple computers were connected to work together as a single system, improving performance and fault tolerance.
   Problem: Managing clusters was complex and costly, requiring high hardware investment, and scalability remained limited by physical constraints.
4. **Grid Computing (1990s)**:
   Grid computing connected distributed systems across networks to pool resources efficiently, using idle computing power from various locations.
   Problem: Despite efficiency, it required significant management, lacked real-time scalability, and was not dynamic enough for variable workloads.
5. **Virtualization (2000s)**:
   Virtualization allowed multiple virtual machines to run on a single physical server, optimizing resource use and flexibility.
   Problem: It still needed advanced software and infrastructure management, and scaling up resources quickly wasn't straightforward.

- **Utility computing** is a cloud service model where computing resources (e.g., storage, processing, and applications) are provided on-demand and billed based on usage, similar to utilities like electricity or water. It allows businesses to pay only for what they use, eliminating the need for upfront infrastructure investment.
- **Web 2.0** introduced interactive and dynamic web pages, allowing users to create, share, and interact with content. It acts as a bridge between users and cloud services, making applications like SaaS easily accessible through web browsers and enhancing user engagement.

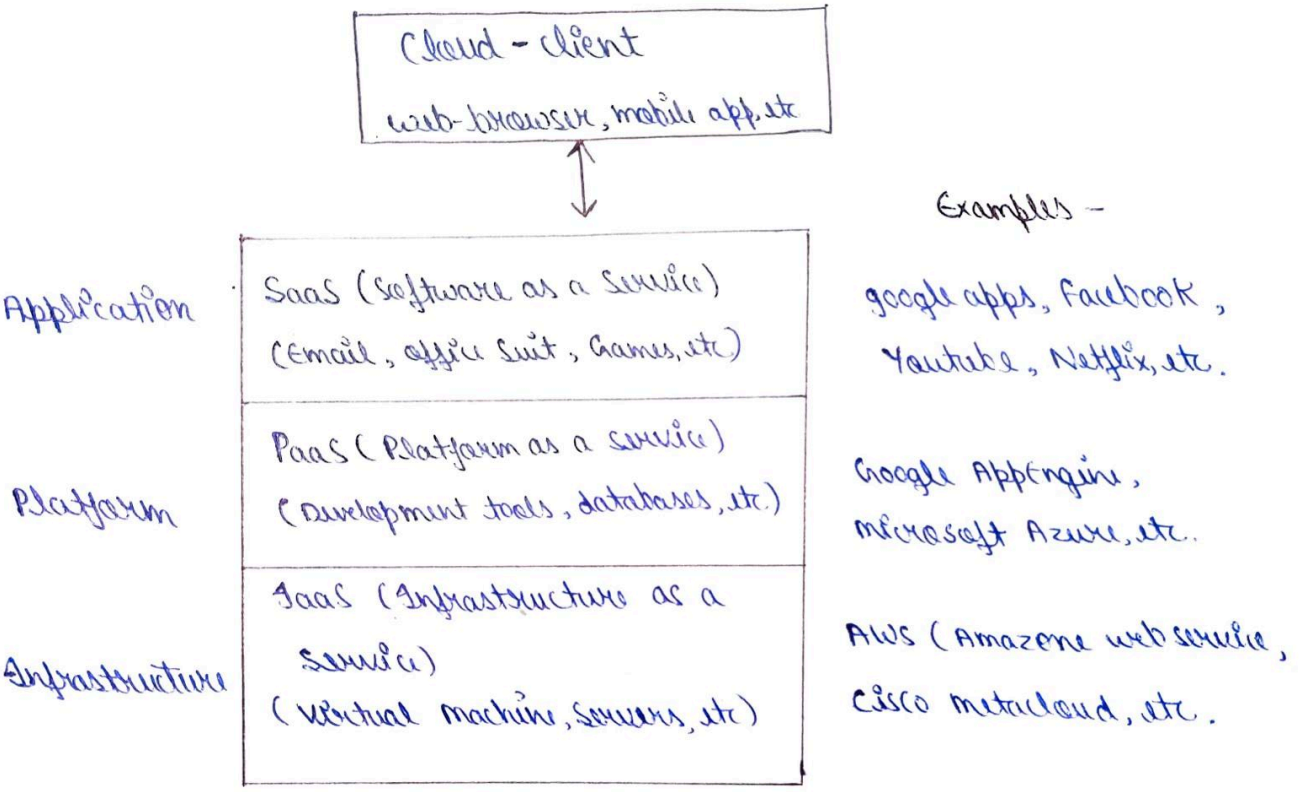
6. **Cloud Computing (2006-Present)**:
   Cloud computing enables on-demand access to computing resources like servers, storage, and applications via the internet, allowing for easy scalability and flexibility.
   Advantage: Cloud computing solved the issues of previous technologies by removing the need for physical infrastructure, allowing rapid scaling, and offering cost-efficient, on-demand services.

6. Components of Cloud Computing Architecture:



1. **Front-End**: User interface and devices accessing cloud services.
2. **Back-End**: Servers and data centers managing cloud resources.
3. **Cloud Management**: Tools for monitoring and controlling cloud services.
4. **Security**: Protection of data and services in the cloud.
5. **Infrastructure**: Hardware resources like servers and networking.
6. **Storage**: Data storage systems in the cloud.
7. **Runtime**: Environment for running applications.
8. **Cloud Service**: Services like IaaS, PaaS, SaaS provided to users.
9. **Application**: Software programs hosted on the cloud.
10. **Internet**: Network enabling access to cloud resources.
11. **Client**: Devices used by users to access cloud services.
12. **Infrastructure (Cloud Infrastructure)**: Foundation resources supporting cloud services.

7. Service/Delivery Models of Cloud Computing:



1. SaaS (Software as a Service): SaaS delivers fully managed software applications over the internet. Users access the software via a web browser without worrying about underlying infrastructure.

Characteristics:

- Fully managed by the provider.
- No need to install or maintain hardware or software.
- Accessible from any device with internet access.
- Examples: Google Workspace, Microsoft 365, Dropbox.

Advantages:

- Cost-effective (no need for hardware/software infrastructure).
- Scalable and easy to upgrade.
- Accessible from any device with the internet.

Disadvantages:

- Limited customization.
- Data security is dependent on third-party providers.
- Requires a stable internet connection.

2. PaaS (Platform as a Service): PaaS provides a platform for developers to build, test, and deploy applications without managing the underlying hardware or software infrastructure.

Characteristics:

- Managed platform with tools for app development.
- Focus on coding and development, not on managing infrastructure.
- Scalability for app hosting.
- Examples: Google App Engine, Microsoft Azure, Heroku.

Advantages:

- Speeds up app development with pre-built tools.
- Scalable based on demand.
- No need to manage underlying infrastructure.

Disadvantages:

- Vendor lock-in can make migration difficult.
- Limited control over the environment.
- Potential security risks from shared resources.

3. IaaS (Infrastructure as a Service): IaaS offers virtualized computing resources like virtual machines, storage, and networking. Users manage the OS and applications, while the provider manages the physical infrastructure.

Characteristics:

- Provides virtual machines, storage, and networking.
- High flexibility and scalability.
- Users manage the OS, storage, and applications.
- Examples: AWS EC2, Google Compute Engine, Microsoft Azure.

Advantages:

- Full control over infrastructure and virtual machines.
- Scalable based on demand.
- Pay-per-use model, cost-effective.

Disadvantages:

- Requires expertise to manage virtual machines and networks.
- Security is the user's responsibility.
- Mismanagement can lead to resource inefficiencies.

8. Differences between SaaS, PaaS, and IaaS:

Feature	SaaS	PaaS	IaaS
Primary Offering	Software applications	Platform for building applications	Virtualized infrastructure resources
Management Responsibility	Fully managed by the provider	Managed platform, user manages apps	User manages OS, storage, and apps
Target Users	End-users	Developers	IT administrators, businesses
Control over Infrastructure	None	Limited control over platform	Full control over virtual infrastructure
Customization	Limited to software features	Can customize applications	Full customization of resources
Examples	Google Workspace, Dropbox	Google App Engine, Heroku	AWS EC2, Google Compute Engine

9. Underlying Principles of Parallel and Distributed Computing

Serial Computing:

- In serial computing, tasks are processed one after the other, sequentially, on a single processor.
- **Problem with Serial Computing**:
  - **Limited Speed**: The speed of execution is limited by the processing power of a single processor.
  - **Inefficient for Large Tasks**: Large or complex tasks take longer to process as each step depends on the previous one.

Requirements of Parallel Computing:

1. **Multiple Processing Units**: Multiple processors or cores to execute tasks simultaneously.

2. **Synchronization Mechanisms**: To ensure proper coordination among tasks.
3. **Divisible Tasks**: The workload should be divisible into smaller tasks that can be executed concurrently.
4. **Communication**: Efficient data exchange between processing units.

Parallel Computing:

- Parallel computing involves breaking a task into smaller sub-tasks that are processed simultaneously by multiple processors or cores.
- **Goal**:
    - **Faster Computation**: Complete tasks in less time by performing computations in parallel.
    - **Efficiency**: Utilizes multiple processors to increase computational efficiency.
    - **Scalability**: Can handle larger datasets and more complex problems by increasing the number of processors.

Distributed Computing:

- Distributed computing involves a network of independent computers (nodes) that work together to complete a task. Each node processes part of the task and communicates to achieve the overall result.
- **Requirement**:
    - **Network Connectivity**: Reliable communication between nodes.
    - **Resource Distribution**: Efficient allocation of resources across nodes.
    - **Fault Tolerance**: The system must continue working even if some nodes fail.

10. Differences between Parallel and Distributed Computing:

Feature	Parallel Computing	Distributed Computing
Definition	Multiple processors work together on the same task.	Multiple independent computers work on different tasks.
Scope	Single system with multiple processors or cores.	Multiple systems (nodes) connected via a network.
Task Division	Task is divided into smaller sub-tasks for parallel execution.	Task is divided into parts and distributed across different systems.
Communication	Requires high-speed communication between processors.	Communication is done over a network, often slower than parallel systems.

Resource Sharing	All processors share the same memory and resources.	Resources are distributed and may be physically separated.
Failure Handling	If one processor fails, the whole process may be disrupted.	Failure of one node can be handled without affecting the entire system.
Synchronization	Strong synchronization required between processors.	Nodes may operate asynchronously, with less strict synchronization.
System Complexity	Less complex, as it uses a single system.	More complex due to multiple interconnected systems.
Scalability	Limited scalability; adding more processors increases complexity.	Highly scalable by adding more nodes to the network.
Performance	Suitable for tasks requiring intensive computation.	Suitable for tasks that can be divided into independent subtasks.

11. Distributed Computing Architecture

1. Software Architecture

Defines the organization of software components in distributed systems.

- **Layered Architecture**:
    - Software is divided into layers, where each layer performs specific functions and interacts with adjacent layers.
    - Example: Presentation Layer, Business Logic Layer, and Data Layer in web applications.
- **Data-Centered Architecture**:
    - All components interact with a central data repository to share and process data.
    - Example: Database systems where multiple clients access and manipulate the same data.

2. System Architecture

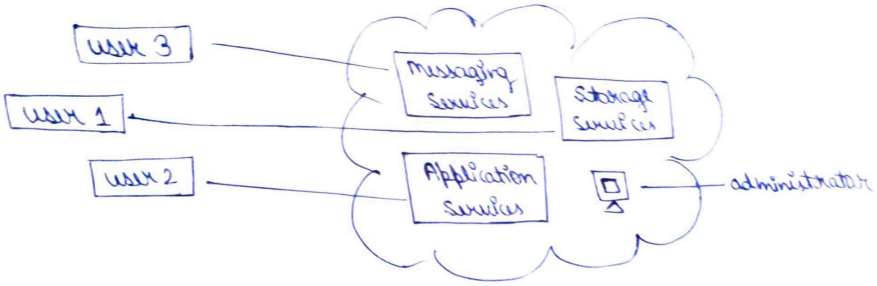
Defines the physical structure of distributed systems, focusing on hardware and network connections.

- **Client-Server Architecture**:
    - Clients request services, and servers provide responses.
    - Example: A web browser (client) accessing a web server.
- **Peer-to-Peer (P2P) Architecture**:
    - All nodes act as both clients and servers, sharing resources directly without a centralized server.
    - Example: File-sharing networks like BitTorrent.

12. Deployment Models of Cloud Computing

1. Public Cloud:

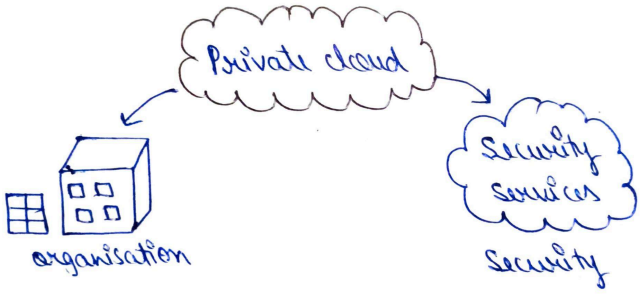
- A cloud environment that is available to the general public over the internet and owned by third-party providers.
- **Example**: Microsoft Azure, AWS.



- **Advantages**:
    - Low cost.
    - Scalable and flexible.
    - Easy to access.
- **Disadvantages**:
    - Less secure.
    - Limited control over resources.

2. Private Cloud:

- A dedicated cloud infrastructure operated solely for one organization, offering greater control and security.
- **Example**: Internal company data centers.

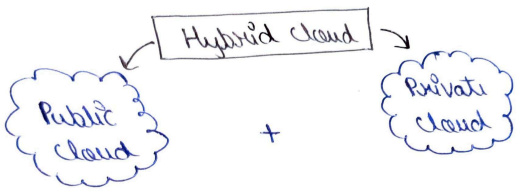


- **Advantages**:
  - High security and privacy.
  - Greater control.
- **Disadvantages**:
  - High cost.
  - Limited scalability.

3. Hybrid Cloud:

A combination of public and private clouds, allowing data and applications to be shared between them for flexibility and optimization.

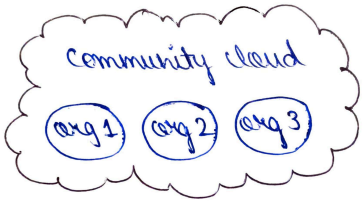
Example: A private cloud for sensitive data and a public cloud for scalability.



- **Advantages**:
  - Cost-effective.
  - Scalable and secure.
- **Disadvantages**:
  - Complex management.
  - Dependency on network connectivity.

4. Community Cloud:

- A shared cloud infrastructure tailored for a group of organizations with common goals or regulatory needs.
- **Example**: Healthcare organizations sharing a cloud for compliance and data sharing.



- **Advantages**:
  - Cost-sharing among users.
  - Improved security compared to public clouds.
- **Disadvantages**:
  - Limited scalability.
  - Complex agreements among members.

13. Elasticity in Computer Networking

Elasticity refers to the ability of a cloud system to dynamically scale resources up or down based on demand. This means that the system can automatically adjust its capacity (e.g., computing power, storage) in real-time to handle changes in workload.

Elasticity allows for:

- **Scaling up**: Increasing resources to meet higher demand.
- **Scaling down**: Reducing resources when demand decreases, optimizing cost.

This on-demand provisioning helps businesses to efficiently manage resources, improve performance, and reduce costs by only using the necessary resources at any given time.

14. On-Demand Provisioning

On-demand provisioning in cloud computing is the automatic allocation and deallocation of resources as needed, ensuring efficiency and cost-effectiveness.

Types of On-Demand Provisioning

1. **Vertical Scaling (Scaling Up/Down)**
   - Adjusts resources like CPU or RAM within a single server.
   - Example: Upgrading a server's RAM for higher performance.
2. **Horizontal Scaling (Scaling Out/In)**
   - Adds or removes servers to distribute workload.
   - Example: Adding more instances to handle increased traffic.

IMPORTANT AND PYQ'S QUESTIONS

1. Define Cloud Computing and explain its Evolution with a neat diagram?

2. Demonstrate Cloud Computing delivery model with advantages and disadvantages.

3. Describe in detail about major Deployment Models and services for cloud computing.

4. Describe in detail about cloud computing reference model with diagram.

5. List out and discuss the innovative characteristics of cloud computing.

6. Define the need of cloud computing and explain its evolution with suitable diagrams.

7. Discuss cloud computing delivery model with advantages and disadvantages.

8. Describe in detail about cloud computing reference models with a neat diagram.

9 Compare parallel computing and distributed computing. [2M]

10. List the characteristics of cloud computing. [2M]

11. Compare parallel computing and distributed computing. [2M]

12. What are the service models available in cloud computing? [2M]

13. What is utility computing? [2M]

14. What is IaaS, PaaS, and SaaS? [2M]

15. Differentiate between Parallel computing and Grid computing? [2M]

Unit 2: Cloud Enabling Technologies Service Oriented Architecture

1. Technologies used in Cloud Computing

1. Virtualization
2. Grid Computing
3. Service-Oriented Architecture (SOA)
4. Utility Computing

2. Virtualization

1. Virtualization creates virtual versions of physical resources like servers, storage, or networks.
2. Multiple virtual machines can run on a single physical machine.
3. Each virtual machine acts like a separate, independent computer.
4. It helps in better resource utilization and improved efficiency.
5. It enables easier scaling and management of cloud services.
6. In simple terms, it splits physical hardware into multiple virtual parts.

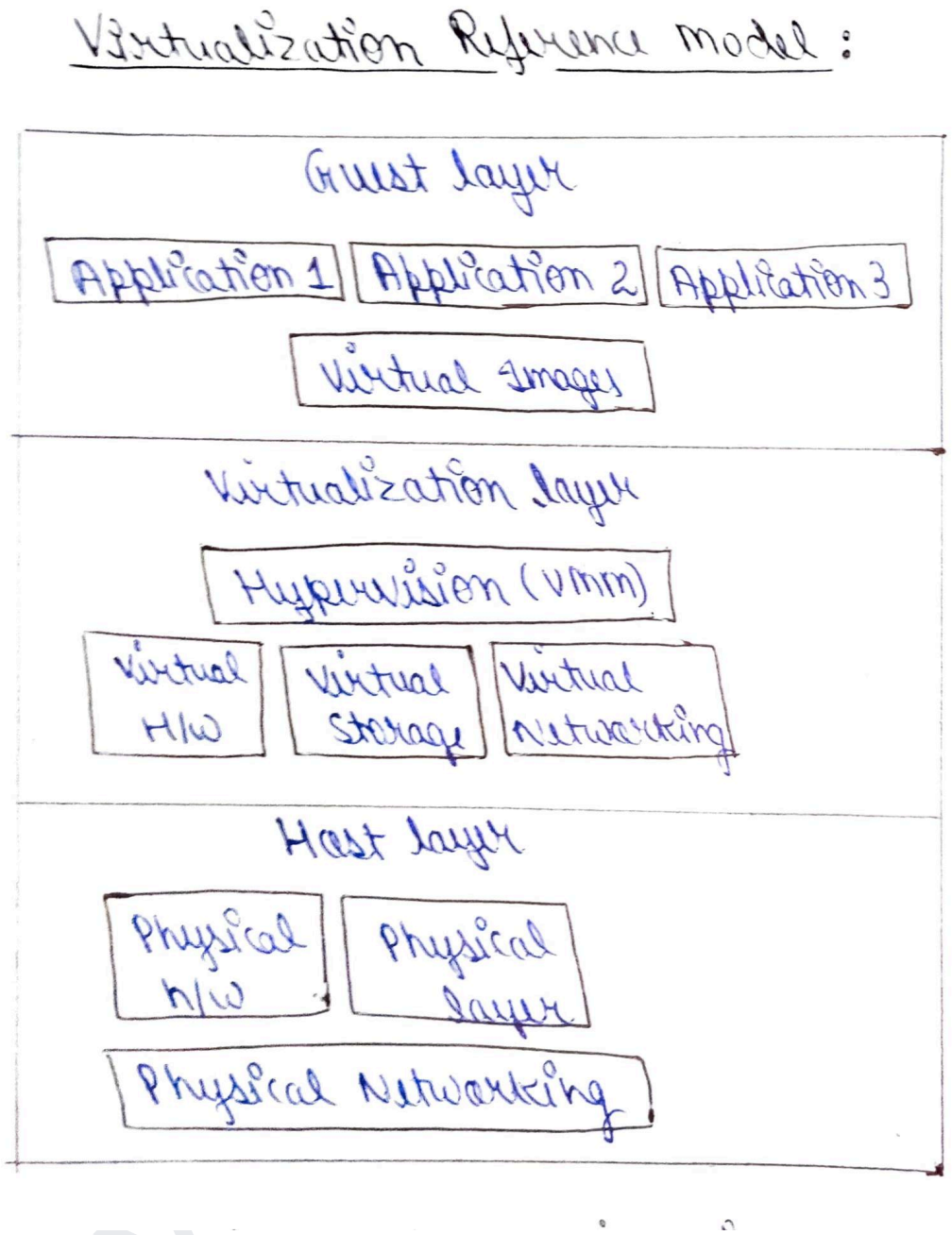
3. Need for Virtualization

1. Efficient resource utilization by allowing multiple virtual machines on a single physical machine.
2. Improved scalability and flexibility for running applications and workloads.
3. Cost savings by reducing the need for physical hardware.
4. Easy management and provisioning of resources.
5. Isolation of applications and workloads for better security.
6. Quick recovery and backup options in case of failures.

4. Advantages of Virtualization

1. **Better resource utilization** by running multiple virtual machines on a single physical machine.
2. **Cost savings** on hardware and energy.
3. **Improved scalability and flexibility** for managing workloads.
4. **Enhanced security** by isolating applications and workloads.
5. **Faster backup, recovery, and disaster recovery.**

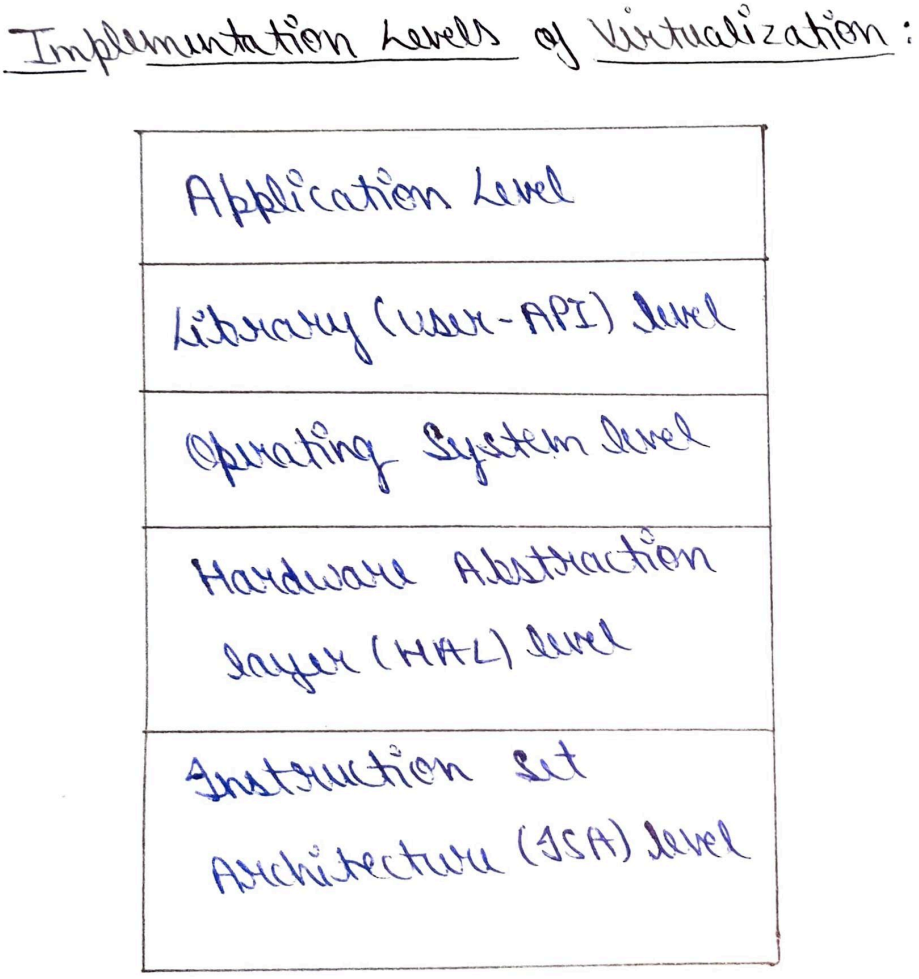
5. Virtualization Reference Model



The Virtualization Reference Model has four major components:

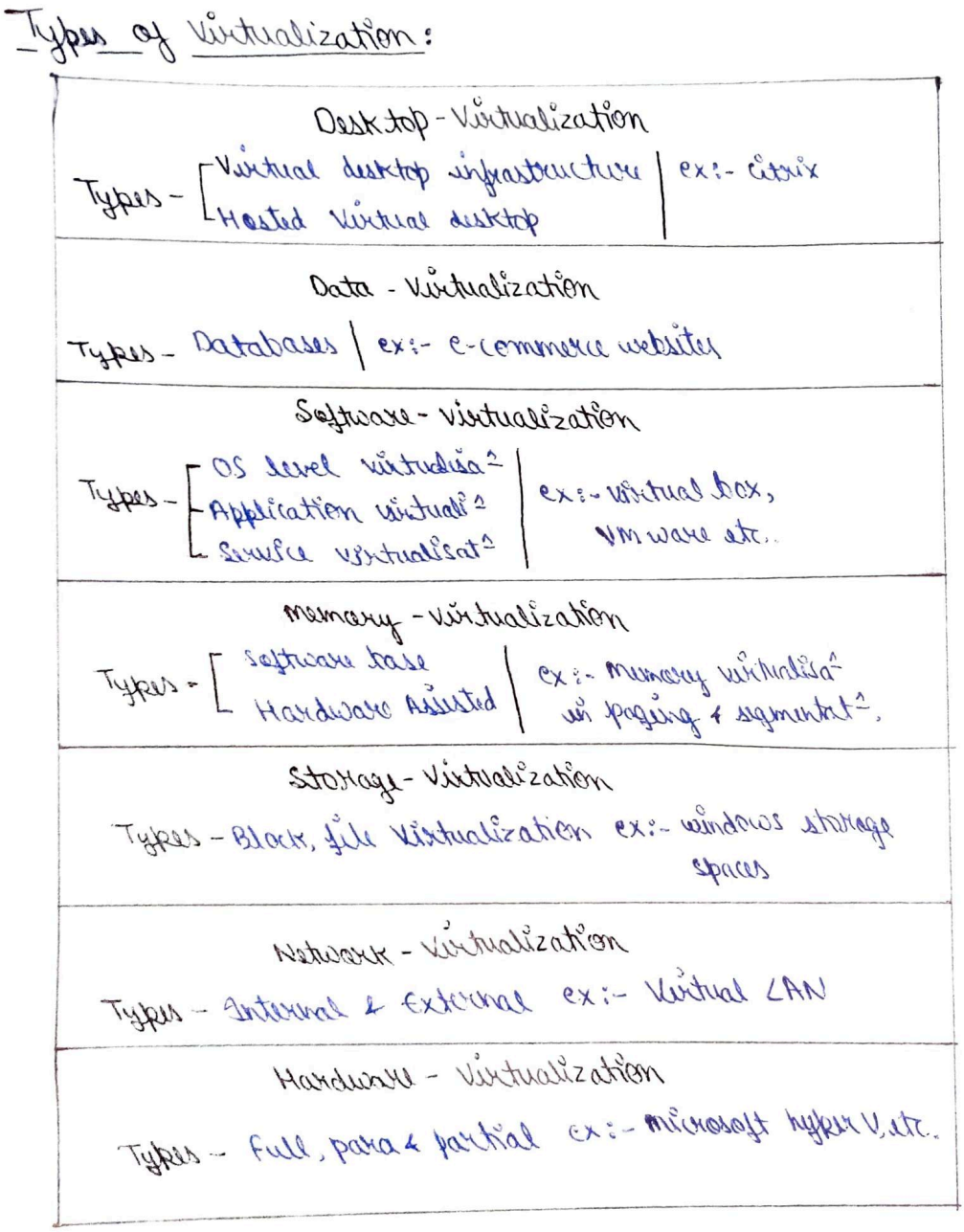
1. **Host Layer**: This is the original physical environment, like a physical computer or server, where everything runs. It includes all the hardware, such as the CPU, memory, and networking resources.
2. **Guest Layer**: These are the virtual machines that run on the host layer. Each guest acts like a separate computer, even though it shares the host's physical resources.
3. **Virtualization Layer**: This layer acts as a bridge between the host and guest layers. It manages the resources and lets multiple virtual machines run on a single physical machine.
4. **Hypervisor**: This is the software that makes virtualization possible. It controls the virtual machines, allocates resources to them, and keeps them isolated from each other.

6. Implementation levels of virtualization:



1. **Application Level**: Virtualization occurs at the application layer. Each application runs as if it is on its own machine, but in reality, they share the underlying system. This type is typically seen in application containers.
2. **Library Level**: Virtualization happens at the library level, where libraries (software components) are virtualized. Applications interact with virtual libraries instead of the physical ones, making the software portable across different environments.
3. **Operating System Level**: In this type, the operating system itself is virtualized to run multiple instances of itself, or multiple different operating systems, on the same hardware. It uses containers or virtualization tools like Docker.
4. **Hardware Abstraction Layer Level**: This layer abstracts the physical hardware from the software, providing a virtualized environment that allows the operating system to interact with virtualized hardware resources instead of physical ones.
5. **Instruction Set Architecture Level**: This level virtualizes the CPU and the instruction set, meaning software instructions are handled by a virtual processor rather than directly by the physical hardware, enabling multiple operating systems to run on the same machine.

7. Types of Virtualization



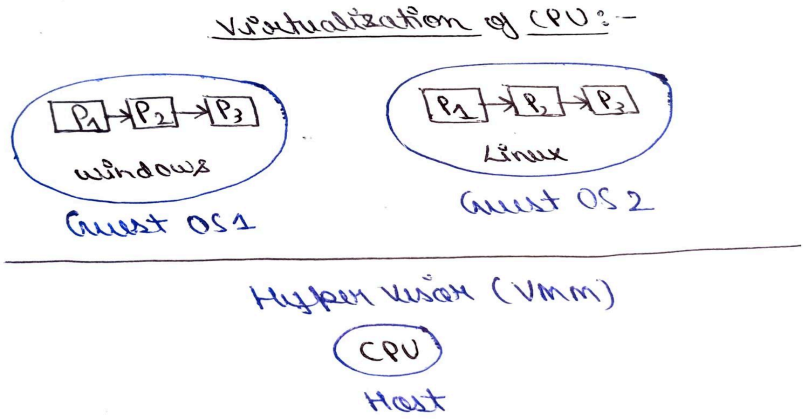
1. **Desktop Virtualization:**
   - Allows users to access and manage their desktop environments from anywhere, on any device, without being tied to a specific physical machine.
   - **Type:** Remote Desktop Virtualization (accessing desktops remotely), Virtual Desktop Infrastructure (VDI) (hosting desktops in a centralized server).
2. **Data Virtualization:**
   - Combines data from multiple sources into a single, unified view, without moving the actual data, allowing users to access and query it as if it's in one place.
   - **Type:** Virtual Data Warehouses, Data Integration Tools.
3. **Software Virtualization:**
   - Runs applications or software in a virtual environment, separating them from the underlying operating system, so they can run independently.

- **Type:** Application Virtualization (isolating apps from OS), OS-Level Virtualization (running software on a virtual OS).

4. **Memory Virtualization:**
   - Combines physical memory resources across multiple devices and makes them appear as a single memory resource to the operating system.
   - **Type:** Virtual Memory, Memory Management Techniques (like swapping and paging).
5. **Storage Virtualization:**
   - Combines physical storage devices into one virtual storage pool, allowing easier management and allocation of storage resources.
   - **Type:** Block-Level Storage Virtualization, File-Level Storage Virtualization.
6. **Network Virtualization:**
   - Creates a virtual network that separates the management of network resources from the physical hardware, allowing better flexibility, scalability, and security.
   - **Type:** Software-Defined Networking (SDN), Network Function Virtualization (NFV).
7. **Hardware Virtualization:**
   - Enables running multiple operating systems on the same physical machine by creating virtual versions of hardware components, like CPUs or memory.
   - **Type:** Full Virtualization (complete isolation), Para-virtualization (OS aware of virtualization).

8. Virtualization of CPU

CPU virtualization allows multiple virtual machines (VMs) to run on a single physical CPU, giving each VM its own virtual CPU. This means that the CPU's resources are divided and allocated to different VMs, allowing them to run independently as if they had their own dedicated CPUs.



Types of CPU Virtualization:

1. **Software-Based CPU Virtualization**:
   - This method uses software (called a **Hypervisor**) to manage and allocate CPU resources to virtual machines. The hypervisor translates the instructions from the virtual machines into instructions that the physical CPU can understand.

2. **Hardware-Assisted CPU Virtualization**:

- In this method, the physical CPU includes special features to support virtualization. These features help the hypervisor run virtual machines more efficiently by providing hardware support for virtualizing the CPU, improving performance and reducing the overhead of software-based virtualization.

9. Virtualization of Memory

Memory virtualization is the process of making physical memory resources (RAM) available to virtual machines, allowing them to use memory as if they have their own dedicated resources, even though they share the physical memory of the host machine.

Three Layers of Memory in Virtualization:

1. **Host Machine Memory**: The actual physical memory (RAM) on the physical machine that is shared among all virtual machines.
2. **Guest Operating System (Virtual Memory)**: The virtual memory seen by the guest operating system in each virtual machine, which appears as if it has its own dedicated memory.
3. **Physical Memory**: The real physical RAM that the host machine uses.

10. Process of Memory Virtualization

Two concept are used in process of Memory Virtualization:

1. **Mapping** is the process of connecting the memory that a virtual machine (VM) thinks it has (virtual memory) to the real physical memory on the host machine. This way, when the VM needs to access memory, the system knows where to find it in the actual physical memory.
2. A **shadow page table** is like a "translator" between the virtual memory used by the VM and the real memory of the host machine. It helps the system figure out which part of the host's physical memory the VM is asking for, making sure the right memory is accessed.

11. Virtualization of Input-Output Devices

Input-Output (I/O) device virtualization allows virtual machines (VMs) to use physical I/O devices (like keyboards, mice, storage, and network interfaces) as if they have their own dedicated devices, even though multiple VMs share the same physical devices.

Process of Input-Output Device Virtualization:

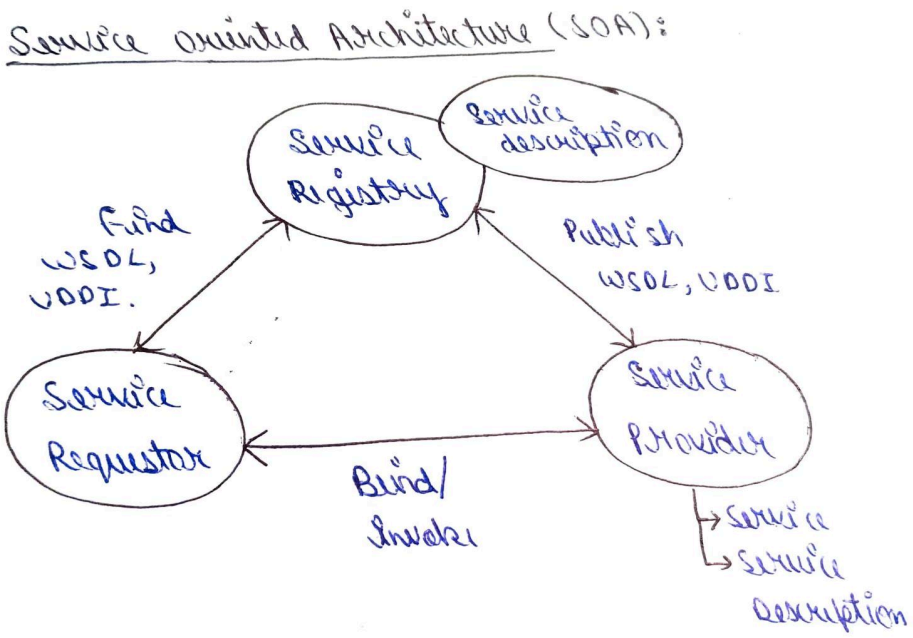
1. The user makes an I/O request through the guest operating system (OS).
2. The guest OS sends the request to the hypervisor.
3. The hypervisor manages and forwards the request to the physical I/O device.
4. The physical device processes the request and sends the response back through the hypervisor.
5. The hypervisor returns the response to the guest OS, completing the I/O operation.

Disaster recovery:

Disaster Recovery (DR) means having a plan to fix and restore your computer systems, data, and operations after something bad happens, like a natural disaster, hacking, or technical failure. It helps a business get back to work quickly and reduces damage.

12. Service-Oriented Architecture (SOA)

SOA is a design style where software components (called services) communicate over a network to work together. Each service is a small, self-contained unit that performs a specific task and can be used by different applications.



Components of SOA:

1. **Services**: Independent, reusable units that perform specific functions.
2. **Service Registry**: A directory where services are listed for easy discovery.
3. **Service Consumer**: Applications or systems that request services.
4. **Service Provider**: The system that hosts and delivers services.
5. **Communication Protocols**: Methods (like HTTP or SOAP) for services to interact.

Benefits of SOA:

1. **Reusability**: Services can be reused across different applications.
2. **Flexibility**: Easy to add or update services without affecting the whole system.
3. **Scalability**: Can scale services individually as needed.
4. **Cost Efficiency**: Reduces duplication by using shared services.

Advantages of SOA:

1. **Improved Agility**: Easier to modify or replace individual services.
2. **Integration**: Simplifies connecting different systems and technologies.
3. **Faster Development**: Reusing existing services speeds up development.

Disadvantages of SOA:

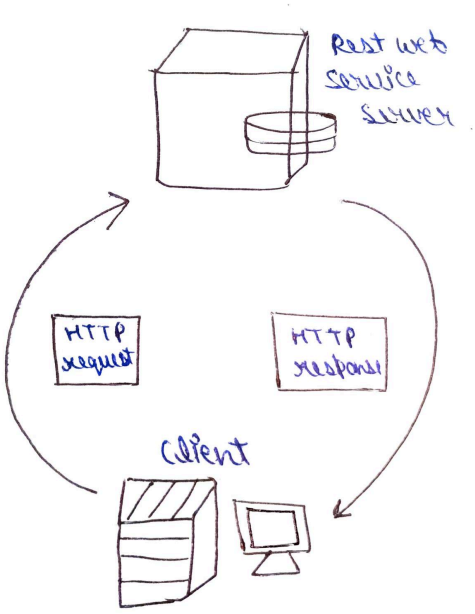
1. **Complexity**: Managing many services and their interactions can become complex.
2. **Performance Overhead**: Network communication between services can slow down the system.
3. **Security Risks**: More services mean more points of access, potentially increasing security concerns.

13. REST (Representational State Transfer)

REST is a way of designing web services that allows systems to communicate with each other over the internet. It uses standard web protocols like HTTP and makes it easy for different software systems to interact by sending requests and receiving responses.

REST Architecture:

There are two main part:
1. A client who requests for resources.
2. A server who has the resources.



HTTP methods:

1. **GET**: Retrieves data from the server.
2. **POST**: Sends data to the server to create a resource.
3. **PUT**: Updates or replaces an existing resource.
4. **PATCH**: Partially updates an existing resource.
5. **DELETE**: Removes a resource from the server.

14. Difference between RESTful and RESTless web services:

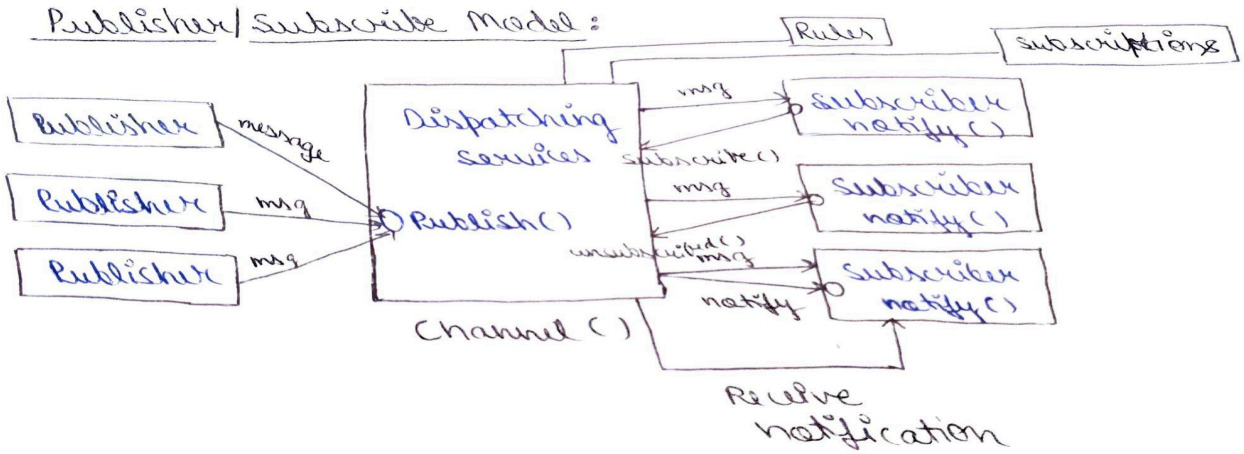
Feature	RESTful Web Services	RESTless Web Services
Definition	Strictly adheres to REST principles.	Does not strictly follow REST principles.
HTTP Methods	Uses standard HTTP methods (GET, POST, etc.).	May use non-standard or custom methods.
Communication	Stateless communication between client and server.	May include stateful interactions.
Resource Access	Resources are accessed via URLs.	Resource access may not be URL-based.
Scalability	Highly scalable due to standardization.	Scalability may be limited.

15. Architectural Constraints of RESTful APIs

1. **Statelessness**:
   Each request made by the client must contain all the information needed for the server to understand and respond. The server doesn't remember anything from previous requests.
2. **Client-Server**:
   The client (like your web browser or app) and the server (where data is stored) work separately. This makes it easier to improve or change either one without affecting the other.
3. **Cacheability**:
   The server can tell the client whether it's okay to save the response (cache it) for later use. This helps speed up repeated requests by using saved data instead of asking the server again.
4. **Uniform Interface**:
   There is a simple, consistent way that clients and servers talk to each other. This makes it easy for different apps or systems to understand each other and work together.
5. **Layered System**:
   There can be different layers (like middle servers) between the client and the main server. The client doesn't need to know about these layers, just that it gets the right data.

6. **Code on Demand**:
   Sometimes, the server can send some code (like JavaScript) to the client to help it do something. This is optional and not always used.

16. Publisher-Subscriber Model



1. **Publisher**: Sends messages.
2. **Subscriber**: Receives messages it subscribed to.
3. **Channel**: Connects publishers and subscribers.

Process of Publisher-Subscriber Model:

1. The **publisher** creates and sends messages or data to a central **channel** without knowing the subscribers.
2. The **channel** organizes messages based on topics or categories.
3. Subscribers **register** with the channel, specifying the topics they are interested in.
4. When a message is published, the channel **delivers** it only to the subscribers of the relevant topic.
5. The **channel handles communication** between publishers and subscribers, ensuring proper message delivery.

Advantages:

1. Publishers and subscribers are independent, making the system flexible.
2. Easy to add or remove subscribers without affecting others.
3. Efficient for broadcasting messages to multiple receivers.

Disadvantages:

1. Hard to manage many subscribers or complex channels.
2. Delays may occur if the channel is overloaded.
3. Debugging issues can be challenging.

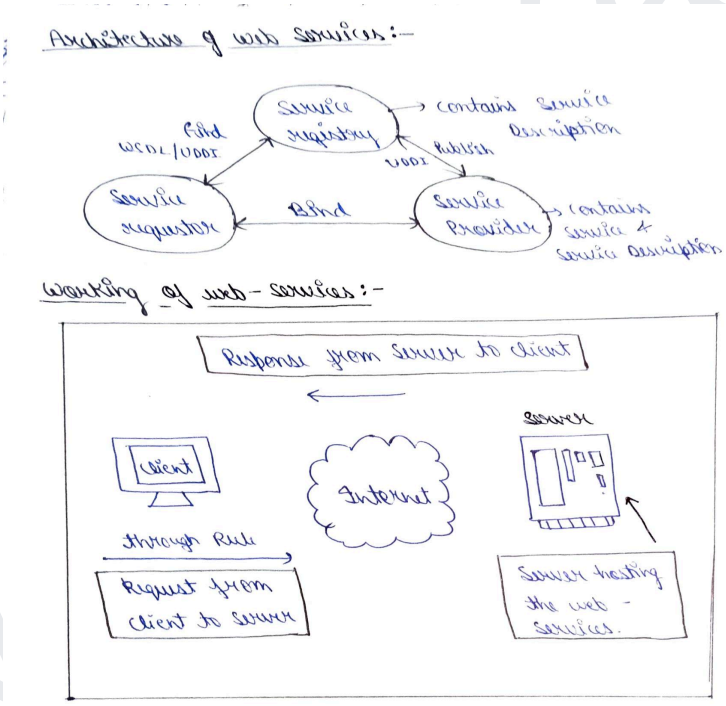
17. What are Web Services

Web services are a way for applications to communicate with each other over the internet, using standard protocols like HTTP. They allow different systems to exchange data and perform tasks, even if they are built using different technologies.

Types of Web Services:

1. **RESTful Web Services**:
   - Use simple URLs and HTTP methods (GET, POST, PUT, DELETE).
   - Data is usually exchanged in lightweight formats like JSON or XML.
   - Easy to use and faster due to its simplicity.
2. **SOAP Web Services**:
   - Use XML-based messages for communication.
   - Follow strict rules and require more bandwidth.
   - More secure and reliable, suitable for complex tasks.

Working of Web Services:



1. **Request**: A client sends a request to the web service over the internet using a protocol like HTTP.
2. **Processing**: The web service processes the request and interacts with the server or database if needed.
3. **Response**: The web service sends back the processed data or result to the client.

Architecture of Web Services:

1. **Web Provider**: The service that hosts and provides the functionality or data.
2. **Web Requestor**: The client or application that requests the service from the provider.

3. **Web Registry**: A directory where web services are listed, helping requestors discover and connect to providers.

18. Difference between SOAP and REST

Aspect	SOAP	REST
Protocol	A protocol with strict rules (SOAP protocol).	An architectural style, not a protocol.
Data Format	Only supports XML for data exchange.	Supports multiple formats like JSON, XML.
Transport Protocol	Works with protocols like HTTP, SMTP, etc.	Uses only HTTP for communication.
Complexity	More complex due to strict standards.	Simpler and easier to use.
Performance	Slower because of XML overhead.	Faster due to lightweight formats.
State	Can support both stateful and stateless operations.	Always stateless (each request is independent).
Security	Provides built-in security (e.g., WS-Security).	Relies on standard HTTP security mechanisms.
Use Case	Ideal for complex, secure, and enterprise systems.	Suitable for web, mobile, and lightweight apps.

IMPORTANT AND PYQ'S QUESTIONS

1. What is virtual desktop infrastructure? Explain with a diagram.

2. Explain Infrastructure virtualization and cloud computing solutions with the help of diagrams.

3. What are the different techniques used for implementation of hardware virtualization? Explain them in detail.

4. Describe in detail about the REST, a software architecture style for distributed systems.

5. Analyze the pros and cons of virtualization in detail.

6. Describe the different techniques used for implementation of Hardware virtualization. Explain them with a diagram.

7. Discuss Service-oriented architecture (SOA). Also explain the building block of SOAP.

8. Explain Architectural constraints of web services.

9. Define disaster recovery. [2M]

10. List types of Virtualizations. [2M]

11. Express the levels of virtualization. [2M]

12. Illustrate Web services. [2M]

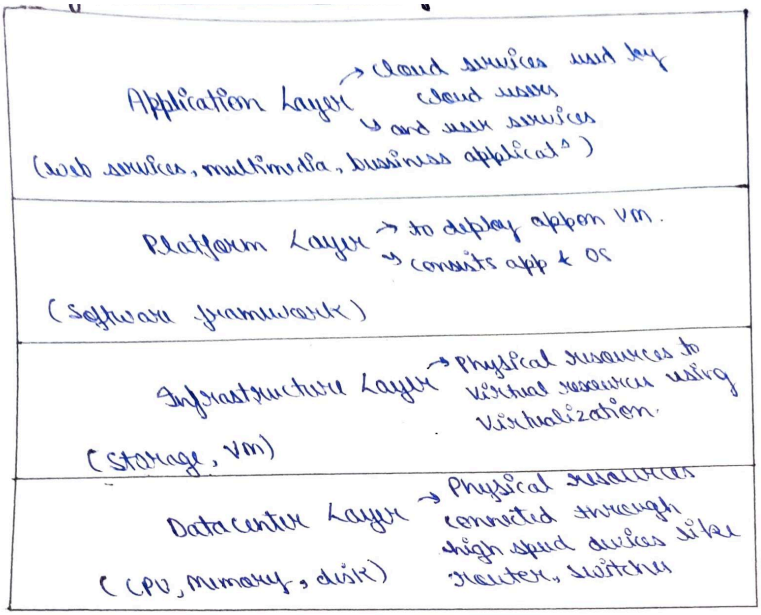
13. Define Virtualization.Demonstrate implementation levels of virtualization.

14. Explain virtualization of CPU,Memory and I/O devices in detail.

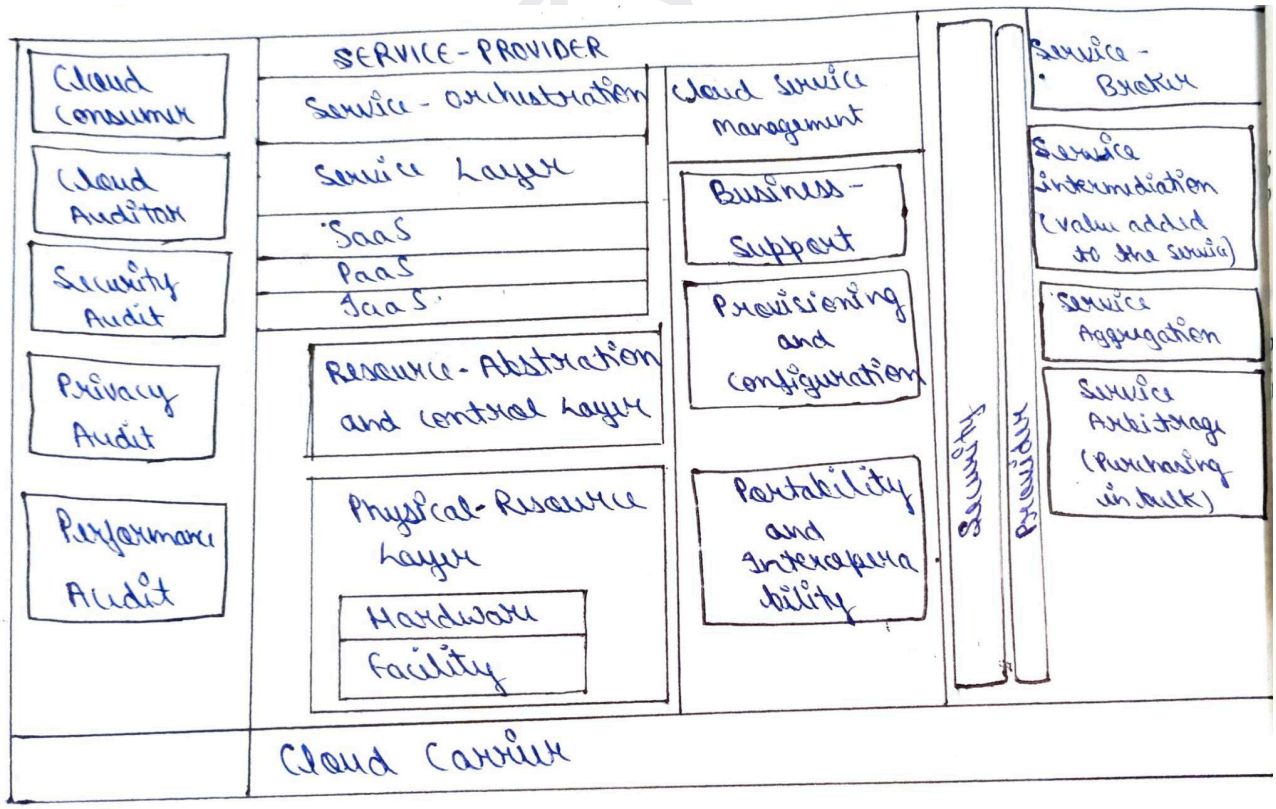
15. What is the role of Service-oriented architecture in cloud computing? [2M]

Unit 3: Cloud Architecture, Services and Storage

1. Layered Architecture of Cloud



2. Layered Architecture-NIST cloud-computing reference Architecture



The NIST Cloud Computing Reference Model explains how different components interact to deliver cloud services efficiently.

1. **Cloud Consumer**
   Cloud consumers are individuals or organizations that use cloud services such as storage, applications, or computing power. For example, a company using Google Drive to store files.
2. **Cloud Provider**
   Cloud providers are companies that offer cloud services, managing resources like servers, storage, and software. Examples include AWS, Microsoft Azure, and Google Cloud.
3. **Cloud Auditor**
   Cloud auditors are independent entities that evaluate the security, compliance, and performance of cloud services. They ensure providers meet required standards, such as GDPR compliance.
4. **Service Broker**
   Service brokers act as intermediaries, helping consumers choose and manage cloud services. They often combine services from multiple providers, like in cloud marketplaces.

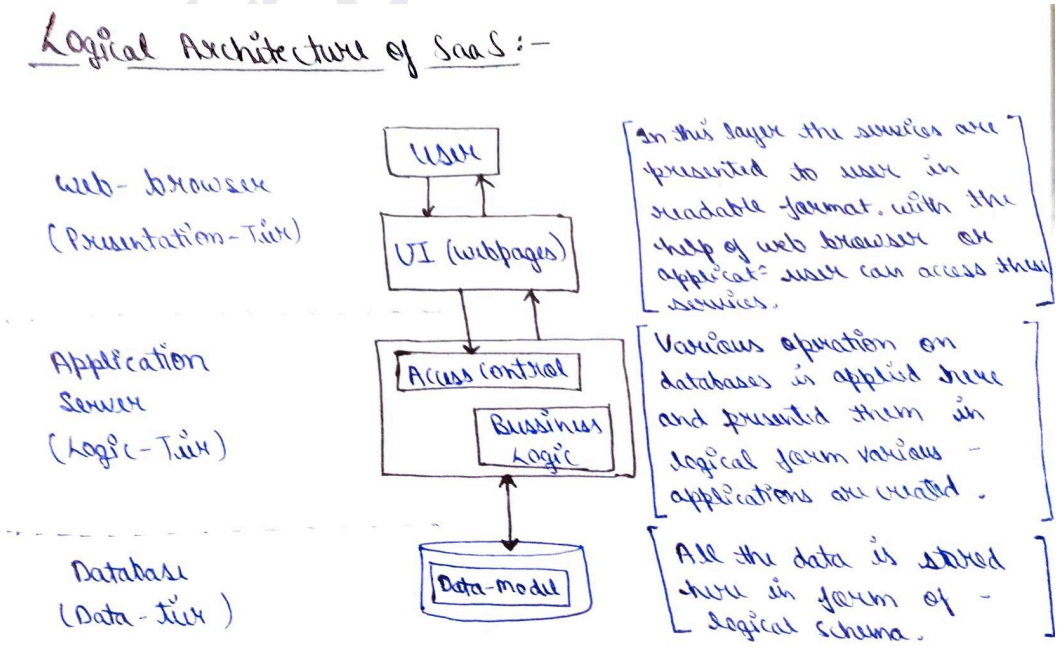
3. Software-as-a-Service (SaaS) in Cloud Computing

SaaS is a cloud model where software is hosted online and accessed via the internet without installing it locally.

Features:

- Accessible over the internet anytime, anywhere.
- Subscription-based payment model.
- Automatic software updates and maintenance.
- Scalable to user needs.
- Multiple users share the same application securely.

Logical Architecture of SaaS:



Advantages:

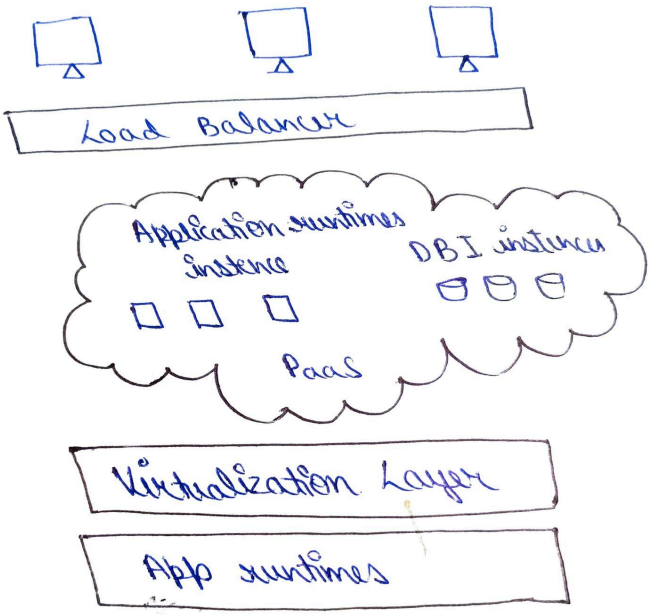
- Reduces hardware and maintenance costs.
- Accessible from any internet-enabled device.
- Quick to deploy without installation.
- Updates and security are managed by the provider.

Disadvantages:

- Requires constant internet connectivity.
- Limited options for customization.
- Storing data on third-party servers may pose risks.
- Switching providers can be difficult.

4. Platform-as-a-Service (PaaS)

PaaS is a cloud service model that provides a platform allowing developers to build, run, and manage applications without dealing with underlying infrastructure.**Example:**Google App Engine



Features:

- Provides a development environment with tools and libraries.
- Automatic scaling based on demand.
- Supports various programming languages.

Advantages:

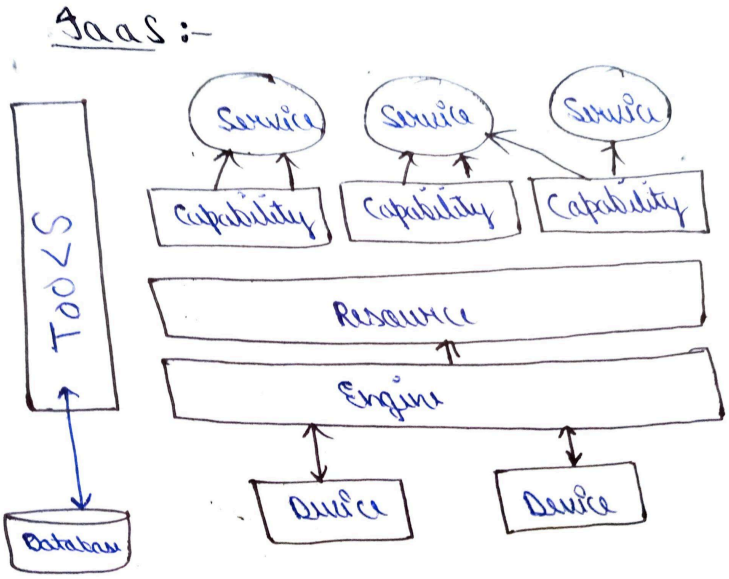
- Faster development and deployment.
- No need to manage infrastructure.
- Scalable and flexible.

Disadvantages:

- Limited control over the underlying infrastructure.
- Can be expensive as usage grows.
- Limited compatibility with certain technologies.

5. Infrastructure-as-a-Service (IaaS)

IaaS is a cloud service model that provides virtualized computing resources like servers, storage, and networking over the internet.**Example:**Amazon Web Services (AWS EC2)



Features:

- Virtual machines, storage, and networking provided on-demand.
- Pay-as-you-go pricing model.
- Full control over the infrastructure.

Advantages:

- Eliminates the cost of buying physical hardware.
- Scalable resources based on business needs.
- Offers flexibility and control over the environment.

Disadvantages:

- Requires technical expertise for setup and management.
- High costs if not managed properly.
- Security risks due to shared resources in the cloud.

6. Cloud Storage

Cloud storage allows users to store data remotely on servers that can be accessed via the internet, instead of using local storage devices like hard drives or USB drives.

Features:

- Stores data remotely on servers accessed through the internet.
- Accessible from any device with an internet connection.
- Types of cloud storage: Public, Private, and Hybrid Cloud.
- Provides options for file sharing, editing, and backup.

Advantages:

- **Remote Access**: Data is accessible from any device with internet access.
- **Scalability**: Easily increase or decrease storage based on needs.
- **Automatic Backup**: Data is backed up automatically to prevent loss.
- **Cost-Efficiency**: No need for physical storage devices, reducing hardware costs.
- **Security**: Providers offer strong security measures like encryption and access control.

Disadvantages:

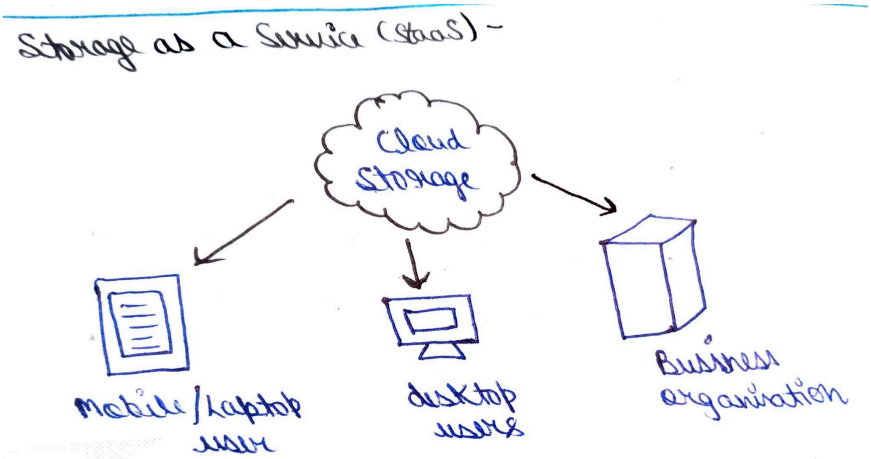
- **Internet Dependency**: Requires a stable internet connection to access data.
- **Privacy Concerns**: Data stored on third-party servers may raise privacy issues.
- **Limited Free Storage**: Free storage is limited, and additional space costs extra.
- **Service Downtime**: Cloud services may experience outages, blocking data access.
- **Speed**: Speed depends on internet connection quality.
- **Compliance Issues**: Storing regulated or sensitive data on third-party servers may face legal challenges.

Cloud Storage Providers:

1. Amazon Web Services (AWS)
2. Google Cloud Storage
3. Microsoft OneDrive
4. Dropbox
5. IBM Cloud Storage

7. Storage-as-a-Service (STaaS)

Storage-as-a-Service (STaaS) is a cloud-based service that allows businesses and individuals to rent storage space from a cloud provider. Users can store, access, and manage their data on remote servers, without the need to invest in physical storage hardware.



Advantages of Storage-as-a-Service (STaaS):

- Scalable storage capacity.
- Cost-effective with no hardware investment.
- Accessible from anywhere with the internet.
- Strong security measures by providers.
- Pay-as-you-go pricing model.

8. Amazon S3 (Simple Storage Service)

Amazon S3 is a scalable cloud storage service provided by AWS that allows users to store and retrieve any amount of data at any time, from anywhere on the web.

Features of Amazon S3:

- Scalable storage for growing data.
- Provides encryption and access control.
- Pay only for the storage you use.
- Supports data backup.
- Accessible globally from anywhere.
- Cost-Effective

9. Comparison between Public Cloud, Private Cloud, and Hybrid Cloud:

Aspect	Public Cloud	Private Cloud	Hybrid Cloud
Ownership	Owned by third-party cloud providers.	Owned and operated by a single organization.	Combination of both public and private clouds.
Cost	Lower cost due to shared resources.	Higher cost due to dedicated infrastructure.	Mixed cost, depending on usage of both.
Scalability	Highly scalable, easily adjusts to demand.	Limited scalability based on private resources.	Flexible scalability across both environments.
Security	Security depends on the provider.	High security, managed in-house or privately.	Security can be customized for sensitive data.
Control	Limited control over the infrastructure.	Full control over infrastructure and security.	Provides control over sensitive data while using public resources for other needs.
Resource Management	Shared resources with multiple customers.	Dedicated resources for a single organization.	Combination of shared and dedicated resources.
Accessibility	Accessible from anywhere via the internet.	Limited to internal or authorized users.	Accessible from both public and private networks.
Customization	Limited customization options.	Full customization to meet specific needs.	Offers some customization, but less than private cloud.

Compliance	May face challenges meeting strict regulations.	Easier to meet compliance and regulatory standards.	Easier to comply with regulations for sensitive data while using public resources for non-sensitive workloads.
Maintenance	Managed and maintained by the cloud provider.	Managed and maintained by the organization.	Maintenance split between the organization and the provider.

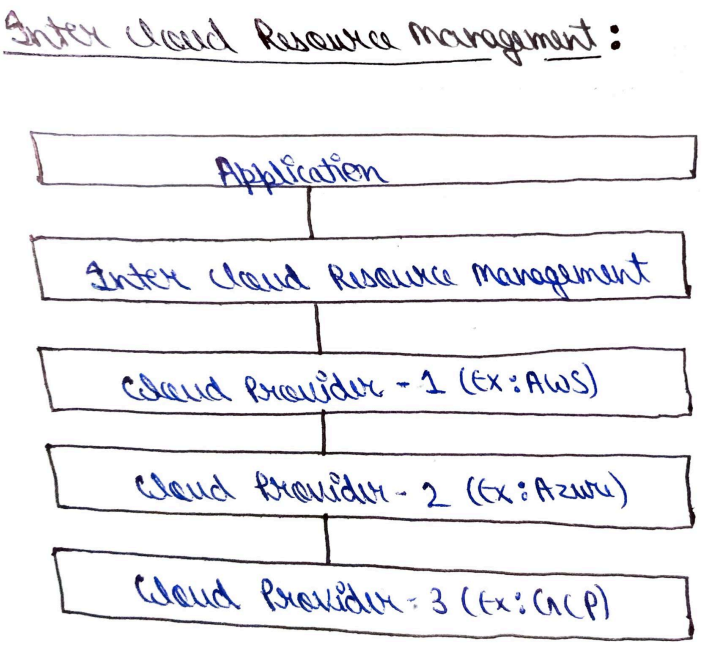
IMPORTANT AND PYQ'S QUESTIONS

1. Give the diagram of Cloud Computing Reference Architecture. Illustrate in detail about the Conceptual Reference Model of cloud.
2. List and discuss the principles for designing public cloud, private cloud and hybrid cloud.
3. Describe in detail about cloud computing reference model with diagram.
4. Differentiate Public cloud and Private cloud. [2M]
5. List out the characteristics of SaaS. [2M]
6. Define a community cloud with an example. [2M]
7. List the types of services provided by cloud. [2M]
8. Explain the major goal of NIST. Also explain the different layers in cloud computing.
9. Describe the components of Cloud Provider. Also explain the responsibility of cloud providers for SaaS, PaaS, and IaaS.
10. Explain Amazon S3. Also describe the advantages of cloud storage.
11. What is IaaS, PaaS, and SaaS? [2M]
12. List the layers used in Layered cloud architecture? [2M]
13. What do you mean by Cloud storage? Describe its types.
14. Illustrate NIST Cloud Computing Reference Architecture in detail.
15. What do you mean by Cloud Service Provider? Explain all the Cloud Service Provider with its features in detail.

Unit 4: Resource Management And Security In Cloud

1. Inter-Cloud Resource Management

Inter-cloud resource management refers to the process of managing and optimizing resources across multiple cloud environments (public, private, or hybrid clouds). It ensures that resources from different clouds are utilized efficiently, enabling seamless integration and collaboration.



Importance:

- **Optimizes Resource Use**: Efficiently allocates resources across clouds.
- **Scalable**: Adjusts resources based on demand.
- **Flexible**: Leverages the strengths of different clouds.
- **Avoids Vendor Lock-In**: Reduces dependency on a single provider.

Benefits:

- **Cost-Efficient**: Optimizes cloud costs.
- **Better Performance**: Distributes workloads for optimal performance.
- **High Availability**: Ensures reliability through resource distribution.
- **Disaster Recovery**: Supports better backup and recovery.

Function:

1. **Resource Allocation & Optimization**: Distributes resources across clouds.
2. **Resource Monitoring & Reporting**: Monitors cloud resources.
3. **Cost & Performance Optimization**: Balances cost and performance.
4. **Load Balancing & Failover**: Ensures workload distribution and failure recovery.
5. **Seamless Integration**: Allows smooth interaction between clouds.

Types of Inter-Cloud Resource Management

1. **Cloud Federation**: Multiple cloud providers collaborate to share resources, providing a unified platform for users.
2. **Multi-Cloud**: Using services from multiple cloud providers to avoid reliance on a single vendor, enhancing flexibility and redundancy.

Technologies Used in Inter-Cloud Architecture

1. **Peer-to-Peer Intercloud Federation**: A decentralized approach where clouds directly share resources with one another, allowing for direct interactions and resource exchange between independent clouds.
2. **Centralized Intercloud Federation**: Involves a central entity managing resources and services across multiple clouds, ensuring efficient resource allocation and coordination.
3. **Multi-Cloud Services**: The use of services from multiple cloud providers to improve flexibility, performance, and cost-efficiency while avoiding vendor lock-in.
4. **Multi-Cloud Libraries**: Pre-built libraries and tools that facilitate interaction and management across multiple cloud environments, simplifying the integration of services and resources.

2. What is Resource Provisioning?

Resource provisioning is the process of allocating and managing cloud resources (e.g., compute power, storage, network) as needed by users or applications to ensure efficient performance and scalability.

Example: A company deploys a web application in the cloud, provisioning virtual machines (VMs) and storage. As traffic increases, additional VMs are provisioned to handle the load.

Key Steps in Resource Provisioning

1. **Resource Allocation**
   Assigning the required resources (CPU, storage, etc.) to meet the demand of applications or users.
2. **Resource Configuration**
   Setting up and configuring resources according to specific needs or workloads.
3. **Resource Scaling**
   Adjusting resources (scaling up or down) based on real-time demand.
4. **Resource Monitoring**
   Continuously tracking the utilization and performance of resources.

Methods of Resource Provisioning

1. **On-Demand Provisioning**
   Resources are allocated only when required, and they are released once no longer needed.
2. **Auto-Scaling**
   Resources are automatically adjusted (scaled up or down) based on workload demand.
3. **Static Provisioning**
   Resources are allocated based on predicted or fixed needs, regardless of real-time demand.
4. **Manual Provisioning**
   Administrators manually allocate resources based on predefined needs.
5. **Automated Provisioning**
   Use of scripts or tools to automatically allocate resources based on application requirements.

6. **Dynamic Provisioning**
   Resources are allocated dynamically, adjusting based on fluctuating workloads.
7. **Event-Driven Resource Provisioning**

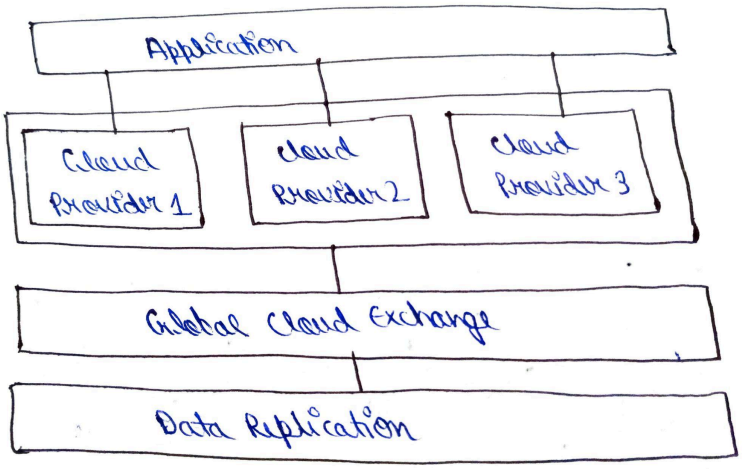
   Resources are allocated automatically in response to specific events or triggers, like a sudden increase in website traffic.

8. **Demand-Driven Resource Provisioning**

   Resources are provisioned based on user demand or workload requirements, ensuring availability when usage increases.

3. Global Exchange of Cloud Resources

The **Global Exchange of Cloud Resources** is the sharing and allocation of cloud resources across different geographic regions and cloud service providers. It enables seamless access to resources, ensuring efficient distribution and utilization of computing, storage, and networking resources on a global scale.



Key Concepts of Global Exchange of Cloud Resources:

1. **Federation**: Collaboration between multiple cloud providers to share and integrate resources.
2. **Interoperability**: Ensuring different cloud platforms can work together smoothly.
3. **Data Mobility**: The ability to move data across different clouds or regions.
4. **Application Portability**: Moving applications between different cloud environments without modification.
5. **Portability**: Transferring workloads and data between cloud providers or on-premise systems.

Benefits of Global Exchange of Cloud Resources:

1. **Cost Efficiency**: Reduces costs by optimizing resource use.
2. **Improved Performance**: Faster access through regional resources.
3. **Flexibility**: Access to diverse cloud services.
4. **High Availability**: Minimizes downtime with distributed resources.
5. **Scalability**: Adjust resources based on demand.

4. Cloud Security Overview

Cloud security refers to the practices, technologies, and policies used to protect data, applications, and services hosted in the cloud. It ensures that cloud environments are safe from unauthorized access, breaches, and data loss, while maintaining confidentiality, integrity, and availability of the resources.

Key elements of cloud security include:

1. **Data Protection:** Safeguarding sensitive information with encryption and access control.
2. **Identity & Access Management (IAM):** Controlling who can access cloud resources and what actions they can perform.
3. **Compliance:** Ensuring cloud systems meet regulatory standards for data privacy and protection.
4. **Incident Response:** Managing and responding to security breaches or attacks in the cloud environment.
5. **Monitoring & Auditing:** Continuously monitoring cloud infrastructure for potential threats and vulnerabilities.
6. **Risk Management:** Identifying and mitigating risks related to cloud adoption and use.

Cloud Security Challenges :

1. **Data Breaches:** Unauthorized access to sensitive data.
2. **IAM Issues:** Poor identity and access management.
3. **Data Loss:** Loss of data due to lack of backups or attacks.
4. **Compliance Issues:** Failure to meet legal or regulatory requirements.
5. **Insecure APIs:** Vulnerabilities in cloud interfaces.
6. **Shared Responsibility:** Unclear security roles between provider and customer.
7. **Insider Threats:** Security risks from trusted users.
8. **Limited Visibility:** Difficulty in monitoring cloud resources.
9. **DoS Attacks:** Overloading cloud services, causing downtime.
10. **Misconfigurations:** Incorrect cloud setup leading to vulnerabilities.

5. Software-as-a-Service(SaaS) Security

SaaS Security protects cloud-based software and data from threats, ensuring confidentiality, integrity, and availability. It includes measures like encryption, access control, user authentication, and continuous monitoring to prevent unauthorized access and data breaches.

Software-as-a-Service (SaaS) Security Key Aspects:

1. **Data Encryption**: Ensures that data is encrypted both in transit and at rest to prevent unauthorized access.
2. **Identity & Access Management (IAM)**: Controls who can access the service and defines their permissions.
3. **Multi-Tenant Security**: Ensures isolation of data between different users or tenants sharing the same SaaS platform.
4. **Compliance**: Adherence to legal and regulatory requirements like GDPR or HIPAA.
5. **Secure APIs**: Protection of APIs used for integrating with third-party services to prevent security vulnerabilities.
6. **Incident Response**: Procedures to address security breaches or incidents swiftly.

7. **User Authentication**: Implementing multi-factor authentication (MFA) to secure user logins.
8. **Monitoring & Auditing**: Continuous monitoring of SaaS applications and auditing access logs for suspicious activities.

6. Security Governance

Security Governance is the framework of policies, procedures, and controls that ensure effective management of security risks within an organization.

Purpose of Security Governance:

1. **Manages Risk**: Identifies and mitigates potential security risks.
2. **Ensures Compliance**: Ensures adherence to legal and regulatory requirements.
3. **Complements IT Strategy**: Aligns security with business goals and IT strategies.
4. **Educates Employees**: Raises awareness and provides training on security policies.
5. **Controls Access**: Manages user permissions and access to sensitive data.

Key Aspects of Security Governance:

1. **Risk Management**: Identifying, assessing, and mitigating risks.
2. **Compliance & Standards**: Adherence to laws, regulations, and industry standards.
3. **Policy Development**: Creation and enforcement of security policies.
4. **Incident Management**: Effective response to security breaches or incidents.
5. **Monitoring & Auditing**: Continuous tracking and auditing of security activities.

Importance of Security Governance:

- **Protects Data**: Ensures data is secure and protected from breaches.
- **Reduces Risks**: Minimizes the likelihood of security threats and vulnerabilities.
- **Maintains Compliance**: Helps organizations meet regulatory and legal requirements.
- **Improves Efficiency**: Streamlines security management across the organization.
- **Builds Trust**: Enhances trust with customers and stakeholders through strong security practices.

7. Virtual Machine Security

Virtual machine security refers to the protection of virtual machines (VMs) from threats and attacks, ensuring that the VM environment remains secure, isolated, and free from vulnerabilities.

Components of Virtual Machine Security:

1. **Hypervisor Security**: Safeguards the virtual machine manager.
2. **VM Isolation**: Keeps VMs separated to prevent interference.

3. **Access Control**: Restricts who can access VMs.
4. **Encryption**: Protects data within and across VMs.
5. **Patch Management**: Updates VMs with security patches regularly.
6. **Monitoring & Logging**: Tracks activities to detect malicious actions.
7. **Backup & Recovery**: Ensures VMs can be restored after failures.

Benefits of Virtual Machine Security:

1. **Protects from Attacks**: Shields VMs from external threats.
2. **Ensures Data Privacy**: Keeps data confidential and secure.
3. **Meets Compliance**: Fulfills security regulations.
4. **Isolation of Workloads**: Prevents one VM from affecting others.
5. **Improves Control**: Gives better management over VM access.

8. What is Identity and Access Management (IAM)

1. **IAM stands for Identity and Access Management**. It is a framework of policies and technologies to manage digital identities and control access to resources.
2. **IAM** helps ensure that only authorized users can access specific systems or data.
3. **IAM** integrates user authentication, authorization, and auditing to keep data secure.

Key Components of IAM:

1. **Authentication**: Verifies the identity of users.
2. **Authorization**: Defines what resources users can access.
3. **User Management**: Manages user identities and roles.
4. **Audit and Monitoring**: Tracks user activities for security and compliance.

Benefits of IAM:

- It helps to manage who can access what systems easily.
- It reduces the chances of unauthorized access to sensitive data.
- It makes the process of granting and revoking access faster.
- It ensures users only have access to what they need, improving security.
- It helps the organization meet legal requirements for data protection.
- It improves the overall safety of systems and reduces security risks.

9. Cloud Security Standards

Cloud security standards are rules and guidelines that organizations follow to protect their cloud systems and data. These standards help ensure that cloud services are secure, comply with laws, and protect sensitive information from unauthorized access.

Prominent Cloud Security Standards:

1. **ISO/IEC 27001**: A global standard for managing information security.
2. **NIST Cybersecurity Framework**: A set of guidelines to improve cybersecurity in the cloud.
3. **SOC 2**: Focuses on the security, availability, and confidentiality of cloud services.

4. **GDPR**: European regulation that protects personal data and privacy in the cloud.
5. **PCI-DSS**: A standard for securing payment card information in the cloud.

IMPORTANT AND PYQ'S QUESTIONS

1. What are security services in the cloud? [2M]
2. What is virtual desktop infrastructure? Explain with a diagram.
3. What are the different security challenges in cloud computing? Discuss each in brief.
4. List the security issues in the cloud. [2M]
5. Define security governance. [2M]
6. Illustrate the following in detail. i. Demand-Driven Resource Provisioning ii. Event-Driven Resource Provisioning
7. Explain in detail about security monitoring and incident response.
8. Explain resource provisioning and resource provisioning methods.
9. Explain inter cloud resource management in cloud computing.
10. Discuss security challenges and security governance in the cloud computing environment.
11. Define VM security. [2M]
12. List the security threats in cloud computing. [2M]
13. What do you mean by intercloud resource management? Write name of its type.
14. Explain IAM and its advantages and disadvantages. What do you mean by cloud computing security? Explain its risk.
15. Describe cloud computing security architecture with a neat schematic diagram. What do you mean by cloud security governance?
16. Explain key objectives and challenges of cloud security governance in detail.

Unit 5: Cloud Technologies And Advancements Hadoop

1. History of Hadoop

1. **2003–2004:** Google published papers on the Google File System (GFS) and MapReduce, inspiring the development of Hadoop.
2. **2005:** Doug Cutting and Mike Cafarella implemented GFS and MapReduce in Nutch, a web-crawling project.
3. **2006:** Hadoop became a standalone project at Yahoo!, named after Doug's son's toy elephant.
4. **2008:** First stable version of Hadoop released; it became an Apache Software Foundation project.
5. **2011:** Expansion of the Hadoop ecosystem with tools like Hive, Pig, and HBase.
6. **2012–Present:** Hadoop became the backbone of big data analytics, widely adopted by tech giants and integrated into cloud platforms like AWS and Azure.

2. Apache Hadoop

Apache Hadoop is an open-source framework designed for distributed storage and processing of big data using a cluster of computers. It uses a **master-slave architecture** and processes data in parallel.

Common Frameworks of Hadoop

1. **HDFS (Hadoop Distributed File System)**
    - Stores large datasets across multiple nodes.
    - Ensures high fault tolerance and scalability.
2. **MapReduce**
    - A programming model for processing big data.
3. **Apache Hive**
    - A SQL-like tool for querying and analyzing big data.
    - Suitable for non-programmers working with Hadoop.
4. **Apache Pig**
    - A scripting platform for writing data processing logic.
    - Simplifies MapReduce programming.
5. **Apache Spark**
    - A fast, in-memory data processing framework.
    - Works with Hadoop and offers better speed for iterative tasks.

Challenges of Hadoop

1. **Complex Programming**: Requires expertise in Java and MapReduce.
2. **Slow Processing**: Inefficient for real-time data due to disk-based processing.
3. **Small File Problem**: Struggles with handling a large number of small files.

4. **Security Gaps**: Limited built-in security features.
5. **Resource Management**: May face inefficiency in resource allocation.

Solutions

1. **Simplified Tools**: Use Hive or Pig for easier data querying.
2. **In-Memory Processing**: Integrate Hadoop with **Apache Spark** for faster performance.
3. **Optimized Storage**: Combine small files or use tools like HBase for better storage management.
4. **Enhanced Security**: Implement Kerberos authentication and third-party tools for data protection.
5. **Improved Resource Allocation**: Leverage **YARN** for better resource management.

Advantages of Hadoop

1. **Scalable**: Easily handles growing data by adding more nodes.
2. **Cost-Effective**: Uses inexpensive commodity hardware.
3. **Fault-Tolerant**: Automatically replicates data to ensure reliability.
4. **Flexible**: Works with all data types (structured, semi-structured, unstructured).
5. **Fast Processing**: Processes data in parallel across distributed systems.

Disadvantages of Hadoop

1. **Complex to Use**: Requires technical skills in Java and MapReduce.
2. **Slow for Small Tasks**: Disk-based processing is inefficient for real-time or small-scale jobs.
3. **Security Limitations**: Lacks robust built-in security features.
4. **High Energy Usage**: Consumes significant power to operate large clusters.
5. **Small File Problem**: Struggles to handle numerous small files efficiently.

Components of Hadoop

1. **HDFS (Hadoop Distributed File System)**
    - Stores large datasets across multiple nodes.
    - Ensures fault tolerance by replicating data.
2. **MapReduce**
    - Processes data in parallel.
    - Splits tasks into smaller parts for fast computation.
3. **YARN (Yet Another Resource Negotiator)**
    - Manages resources in a Hadoop cluster.
    - Allocates tasks to different nodes.

3. HDFS(Hadoop Distributed File System)

HDFS (Hadoop Distributed File System) is a storage system in Hadoop designed to store and manage large amounts of data across multiple computers in a cluster.

Why HDFS?

1. Handles **big data** efficiently by splitting it into blocks.
2. Ensures **data reliability** through replication.
3. Works well with distributed computing for faster processing.

Characteristics of HDFS

1. **Distributed Storage**: Stores data across multiple machines.
2. **Fault Tolerance**: Replicates data to avoid loss during failures.
3. **High Scalability**: Can handle growing data by adding more nodes.
4. **Write-Once, Read-Many**: Optimized for reading large datasets frequently.
5. **Cost-Effective**: Works on inexpensive commodity hardware.

4. MapReduce

MapReduce is a programming model used in Hadoop to process and generate large datasets in a distributed manner. It splits the data processing into two main tasks: **Map** and **Reduce**, which are executed in parallel across a cluster of computers.

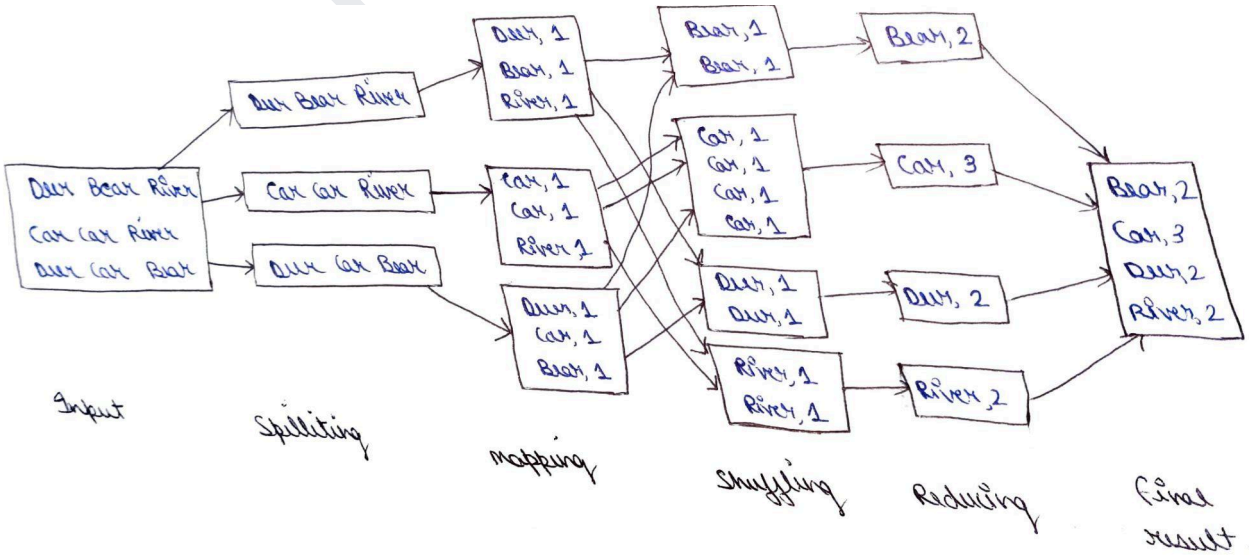
Phases of MapReduce

1. **Map Phase**: Processes and filters data.
2. **Shuffle and Sort Phase**: Organizes and transfers data to Reducer.
3. **Reduce Phase**: Aggregates and finalizes the results.

Characteristics of MapReduce

1. **Parallel Processing**: Splits tasks across multiple nodes for faster processing.
2. **Fault Tolerance**: Automatically re-executes failed tasks.
3. **Scalable**: Handles large datasets effectively.
4. **Data Localization**: Processes data where it is stored to reduce data transfer.
5. **Simple Programming**: Provides an easy-to-understand model for developers.

How mapReduce works



5. VirtualBox

VirtualBox is an open-source virtualization software developed by Oracle. It allows users to run multiple operating systems (virtual machines) on a single physical machine simultaneously. It is widely used for testing, development, and running isolated environments.

Typical VirtualBox Deployment Components

1. **Host Machine**:
    - The physical machine where VirtualBox is installed.
2. **Guest Machines**:
    - Virtual machines (VMs) running different operating systems.
3. **Virtual Disk**:
    - Storage allocated for each VM, acting as its hard drive.
4. **VirtualBox Manager**:
    - GUI tool for managing and configuring VMs.
5. **Shared Folders**:
    - Facilitates file sharing between the host and guest machines.

6. Google App Engine (GAE)

Google App Engine is a **Platform-as-a-Service (PaaS)** offering from Google Cloud. It allows developers to build, deploy, and run applications on Google's infrastructure without managing servers or hardware. It is designed for web applications and mobile backend services.

Key Features of Google App Engine

1. **Fully Managed Environment**:
    - Google handles server management, scaling, patching, and security updates.
2. **Automatic Scaling**:
    - Apps scale automatically based on the traffic or workload.
3. **Supports Multiple Languages**:
    - Supports popular programming languages like Python, Java, PHP, Go, and more.
4. **Integrated with Google Cloud Services**:
    - Works seamlessly with services like Datastore, Cloud SQL, and BigQuery.
5. **Pay-As-You-Go Pricing**:
    - Charges only for the resources used, such as compute, storage, and network bandwidth.

Advantages of Google App Engine

- **No Server Management**: Focus on app development, not infrastructure.
- **Automatic Scaling**: Ideal for apps with fluctuating traffic.
- **Global Reach**: Deploy apps closer to users worldwide for better performance.
- **Security**: Google ensures robust security for apps.

Disadvantages of Google App Engine

- **Vendor Lock-In**: Dependency on Google's infrastructure.
- **Limited Customization**: May not suit complex apps requiring deep server control.
- **Cost**: Can become expensive for high-traffic apps.

Challenges of Google App Engine

1. **Vendor Lock-In**: Difficult to migrate to other platforms.
2. **Limited Customization**: Restricted access to underlying infrastructure.
3. **Language Support**: Limited compatibility with some frameworks or libraries.
4. **Cost**: Auto-scaling can lead to unpredictable costs.
5. **Debugging Issues**: Limited access to hardware and logs complicates debugging.

Google App Engine Services:

1. **Cloud Datastore**
2. **Cloud SQL**
3. **Cloud Storage**
4. **Cloud Firestore**
5. **Memorystore**

7. OpenStack

OpenStack is an open-source platform used for building and managing private and public clouds. It provides Infrastructure-as-a-Service (IaaS) by pooling virtualized resources such as compute, storage, and networking through a web-based dashboard or APIs.

Core Components of OpenStack

1. **Nova (Compute)**
    - Manages virtual machines and computing power.
    - Provides APIs for provisioning and managing instances.
2. **Swift (Object Storage)**
    - Stores and retrieves large amounts of unstructured data.
    - Ideal for backups, archiving, and static content.
3. **Cinder (Block Storage)**
    - Provides persistent storage to virtual machines.
    - Similar to hard drives attached to servers.
4. **Neutron (Networking)**
    - Handles networking for virtual machines.
    - Provides features like load balancing, firewalls, and VPNs.
5. **Horizon (Dashboard)**
    - Web-based interface for managing OpenStack resources.

Advantages of OpenStack

- **Cost-Effective:** No licensing fees.
- **Vendor Neutral:** Works across various hardware and software vendors.
- **Customizable:** Tailored to specific business needs.
- **Large Community:** Continuous updates and support from contributors.

Disadvantages of OpenStack

- **Complex Deployment:** Requires expertise to set up and manage.
- **High Maintenance:** Needs ongoing support and monitoring.
- **Hardware Dependency:** Performance depends on underlying infrastructure.

8. Cloud Federation

Cloud Federation is the integration and interconnection of multiple cloud environments (public, private, or hybrid) to allow seamless sharing of resources and services between them. It enables organizations to distribute workloads, share data, and move resources dynamically between different cloud platforms. Cloud federation facilitates a unified cloud environment by connecting various cloud services, allowing users to access and utilize them as if they are part of a single, integrated system.

Types of Federation in cloud:

1. Trusted Federation

- In trusted federation, clouds or organizations are pre-approved to share resources based on mutual trust. These clouds are typically known to each other and have established agreements or protocols for secure data exchange. The focus is on seamless integration, but there might not be much independent verification.
- Example: Two companies with private clouds trust each other to share certain resources without heavy encryption or additional security checks.

2. Verified Federation

- Verified federation requires that each participating cloud undergo independent verification through third-party assessments or certifications (like ISO or SOC2). This ensures that each cloud meets specific security, compliance, and operational standards before being integrated into the federation.
- Example: A cloud provider must pass an audit to confirm it complies with security protocols before being allowed to federate resources with other clouds.
- Example: A hybrid cloud where both private and public clouds are linked, sharing resources under a common security and management framework.

3. Permissive Federation

- In permissive federation, clouds share resources freely with minimal restrictions or stringent security measures. These clouds trust each other enough to allow broad access to resources, making the federation more flexible but potentially less secure. It typically involves fewer encryption protocols.

- Example: A company allows multiple clouds to access its resources with simple access controls, relying on the trust between the clouds rather than enforcing strong security measures.

4. Encrypted Federation

- Encrypted federation places a strong emphasis on security. Data shared between clouds is encrypted to ensure that sensitive information is protected during transfer and storage. This type of federation ensures that no unauthorized parties can access or tamper with the data.
- Example: Data being shared between a private cloud and a public cloud is encrypted, ensuring privacy and compliance with security standards such as GDPR or HIPAA.

Advantages of federated cloud

1. **Cost Efficiency:** Reduces costs by sharing resources across multiple cloud providers, avoiding the need for duplicate infrastructure.
2. **Scalability:** Enables seamless scaling of resources across federated clouds based on demand.
3. **Flexibility:** Allows organizations to choose the best services from different cloud providers, optimizing performance and cost.
4. **Resource Optimization:** Facilitates efficient use of resources by distributing workloads across multiple clouds.
5. **Improved Availability:** Enhances service reliability by leveraging multiple cloud providers, ensuring minimal downtime.

9. What are Web Services

Web services are a way for applications to communicate with each other over the internet, using standard protocols like HTTP. They allow different systems to exchange data and perform tasks, even if they are built using different technologies.

Types of Web Services:

3. **RESTful Web Services**:
   - Use simple URLs and HTTP methods (GET, POST, PUT, DELETE).
   - Data is usually exchanged in lightweight formats like JSON or XML.
   - Easy to use and faster due to its simplicity.
4. **SOAP Web Services**:
   - Use XML-based messages for communication.
   - Follow strict rules and require more bandwidth.
   - More secure and reliable, suitable for complex tasks.

10. Comparison between Web Services and APIs:

Aspect	Web Services	APIs (Application Programming Interfaces)
Definition	A standardized medium for communication between two devices over a network.	A set of protocols and tools for building software applications.
Scope	Web services are a subset of APIs designed to operate over the web.	APIs can be web-based, library-based, or local and not necessarily web-oriented.
Communication	Uses protocols like SOAP, REST, and XML-RPC for communication.	Can use any protocol: HTTP, HTTPS, FTP, WebSocket, etc.
Transport	Relies primarily on web protocols like HTTP and HTTPS.	Can work with various transport mechanisms beyond the web.
Platform Dependency	Requires a network to operate (internet or intranet).	May work locally without a network or over a network.
Format	Commonly uses XML or JSON for data exchange.	Can use various formats like XML, JSON, YAML, plain text, etc.
Purpose	Specifically for network communication between applications.	Designed for enabling functionality between software components or applications.
Implementation	Must be hosted on a server and accessed via a network.	Can be implemented locally or remotely, depending on its design.
Usage Examples	SOAP-based web service, RESTful web service.	REST API, GraphQL API, library-based APIs like Java SDKs.

Example Protocols	SOAP (Simple Object Access Protocol), REST (Representational State Transfer).	REST, SOAP, GraphQL, gRPC, and others.

IMPORTANT AND PYQ'S QUESTIONS

1. Take a suitable example and explain the concept of map reduces
2. Give a suitable definition of cloud federation stack and explain it in detail.
3. Explain the major cloud features of Google applications engine.
4. List the functional models of GAE. [2M]
5. Elaborate HDFS concepts with suitable illustrations.
6. Describe the following in detail i. Google Cloud Infrastructure ii. GAE Architecture
7. List the security threats in cloud computing. [2M]
8. Explain Hadoop and its history. Also illustrate Hadoop architecture.
9. Give a suitable definition of cloud federation stack and explain it in detail.
10. Explain Web services in detail. Differentiate Web services and APIs.
11. Define virtual box. [2M]
12. Explain web services. Also, write the difference between APIs and web services in detail?
13. What do you mean by Hadoop and its history? Why is it important ?
14. What do you mean by Google App Engine and Open Stack? Explain both advantages and disadvantages and openStack components.

Previous Year Question Papers

B. TECH
(SEM-VII) THEORY EXAMINATION 2020-21
CLOUD COMPUTING

Time: 3 Hours ***Total Marks: 70***

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief. 2 x 7 = 14

a.	Why is cloud computing required? List the five characteristics of cloud computing.
b.	Differentiate between parallel computing and grid computing.
c.	What is the role of service-oriented architecture in cloud computing?
d.	List the layer used in layered cloud architecture.
e.	Write steps to ensure virtual machine security in cloud computing.
f.	What do you mean by inter cloud resource management? Write name of its types.
g.	What is distributed computing?

SECTION B

2. Attempt any *three* of the following: 7 x 3 = 21

a.	Define cloud computing and explain its evolution with neat diagram.
b.	What do you mean by cloud federation? Explain the four levels and future of federation in details.
c.	What do you mean by cloud storage? Describe its types.
d.	Explain IAM and its advantages & disadvantages. What do you mean cloud computing security? Explain its risks.
e.	Illustrate Web services in details. Why is Web Services required? Differentiate between API and Web services.

SECTION C

3. Attempt any *one* part of the following: 7 x 1 = 7

(a)	Demonstrate cloud computing delivery model with advantages and disadvantages.
(b)	What are challenges faced in architectural of cloud computing? Discuss.

4. Attempt any *one* part of the following: 7 x 1 = 7

(a)	Define virtualization. Demonstrate implementation level of virtualization.
(b)	Explain virtualization of CPU, Memory, and I/O devices in details.

5. Attempt any *one* part of the following: 7 x 1 = 7

(a)	Illustrate NIST Cloud computing reference architecture in details.
(b)	What do you mean by cloud service provider? Explain all the cloud service provider with its feature in details.

6. Attempt any *one* part of the following: 7 x 1 = 7

(a)	Describe cloud computing security architecture with neat schematic diagram.
(b)	What do you mean by cloud security governance? Explain key Objectives and challenges of cloud security governance in details.

7. Attempt any *one* part of the following: 7 x 1 = 7

(a)	What do you mean by Hadoop and its history? Why it is important? Illustrate Hadoop architecture.
(b)	What do you mean by Google App Engine (GAE) and Open stack? Explain both advantages & disadvantages and open stack components.

B.TECH
(SEM VII) THEORY EXAMINATION 2021-22
CLOUD COMPUTING

Time: 3 Hours ***Total Marks: 100***

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief. 2 x 10 = 20

a. Define threat Agents? Explain their role in cloud security.
b. What are security services in the cloud?
c. What is encryption? Define the types of encryption.
d. What is utility computing?
e. What is IaaS, PaaS, and SaaS?
f. What is VM network routing?
g. Give some examples of Web 2.0 applications.
h. What are IaaS, PaaS and SaaS ?
i. What are security services in the cloud?
j. What are modules of Hadoop?

SECTION B

2. Attempt any *three* of the following: 10 x 3 = 30

a. What is virtual desktop infrastructure? Explain with diagram.
b. How does an unauthorized access can be detected by the help of virtualization techniques?
c. Take a suitable example and explain the concept of map reduces.
d. Explain Cloud Computing reference model with diagram.
e. What is the difference between cloud computing and distributed computing?

SECTION C

3. Attempt any *one* part of the following: 10 x 1 = 10

(a) Explain Cloud computing security architecture.
(b) Explain Infrastructure virtualization and cloud computing solutions with the help of diagram.

4. Attempt any *one* part of the following: 10 x 1 = 10

(a) What are the different techniques used for implementation of hardware virtualization? Explain them in detail.
(b) What is load balancing? What are the advantages of load balancing?

5. Attempt any *one* part of the following: 10 x 1 = 10

(a) What are the different security challenges in cloud computing? Discuss each in brief.
(b) What is Honeypot? What are the different types of Honeypot?

6. Attempt any *one* part of the following: 10 x 1 = 10

(a) What do you mean by third party cloud services? Give suitable examples.
(b) Explain the major cloud features of Google applications engine.

7. Attempt any *one* part of the following: 10 x 1 = 10

(a) Give a suitable definition of cloud federation stack and explain it in detail.
(b) Write short notes on any two of the followings:
i. HADOO ii.Microsoft Azure

B.TECH.
(SEM VII) THEORY EXAMINATION 2022-23
CLOUD COMPUTING

Time: 3 Hours ***Total Marks: 100***

Note: Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief. 2 x 10 = 20

(a) Compare parallel computing and distributed computing.
(b) What are the service models available in cloud computing?
(c) Express the levels of virtualization.
(d) Illustrate Web services.
(e) Differentiate Public cloud and Private cloud.
(f) List out the characteristics of SaaS.
(g) List the security issues in cloud.
(h) Define security governance.
(i) List the functional models of GAE.
(j) What is the use of cloud Watch in Amazon EC2?

SECTION B

2. Attempt any *three* of the following: 10x3 = 30

(a) Describe in detail about major Deployment Models and services for cloud computing.
(b) Illustrate the three major components of virtualized environment.
(c) Give the diagram of Cloud Computing Reference Architecture. Illustrate in detail about the Conceptual Reference Model of cloud.
(d) Illustrate the following in detail
i. Demand-Driven Resource Provisioning
ii. Event-Driven Resource Provisioning
(e) Elaborate HDFS concepts with suitable illustrations.

SECTION C

3. Attempt any *one* part of the following: 10x1=10

(a) Describe in detail about cloud computing reference model with diagram.
(b) List out and discuss the innovative characteristic of cloud computing.

4. Attempt any *one* part of the following: 10x1=10

(a) Describe in detail about the REST a software architecture style for distributed systems.
(b) Analyze the pros and cons of virtualization in detail.

5. Attempt any *one* part of the following: 10x1=10

(a) List and discuss the principles for designing public cloud, private cloud and hybrid cloud.

(b) Describe Cloud deployment models with neat diagrams.

6. Attempt any *one* part of the following: 10x1=10

(a) Describe the Secure Software Development Life Cycle with neat diagram.

(b) Explain in detail about security monitoring and incident response.

7. Attempt any *one* part of the following: 10x1=10

(a) Describe the following in detail

i. Google Cloud Infrastructure

ii. GAE Architecture

(b) Illustrate any five web services of Amazon in detail.

BTECH
(SEM VII) THEORY EXAMINATION 2023-24
CLOUD COMPUTING

TIME: 3 HRS **M.MARKS: 100**

Note: 1. Attempt all Sections. If require any missing data; then choose suitably.

SECTION A

1. Attempt *all* questions in brief.

Qno.	Question	Marks	CO
a.	Compare parallel computing and distributed computing.	2	1
b.	List the characteristics of cloud computing.	2	1
c.	Define disaster recovery.	2	2
d.	List types of Virtualizations.	2	2
e.	Define community cloud with example.	2	3
f.	List the types of services provided by cloud.	2	3
g.	Define VM security.	2	4
h.	List the security threats in cloud computing.	2	4
i.	List four levels of Federation.	2	5
j.	Define virtual box.	2	5

SECTION B

2. Attempt any *three* of the following:

a.	Define the need of cloud computing and explain its evolution with suitable diagram.	10	1
b.	Describe the different techniques used for implementation of Hardware virtualization. Explain them with diagram.	10	2
c.	Explain the major goal of NIST. Also explain the different layers in cloud computing.	10	3
d.	Explain resource provisioning and resource provisioning methods.	10	4
e.	Explain Hadoop and its history. Also illustrate Hadoop architecture.	10	5

SECTION C

3. Attempt any *one* part of the following:

a.	Discuss cloud computing delivery model with advantages and disadvantages.	10	1
b.	Describe in detail about cloud computing reference model with a neat diagram.	10	1

4. Attempt any *one* part of the following:

a.	Discuss Service-oriented architecture (SOA). Also explain building block of SOAP.	10	2
b.	Explain Architectural constraints of web services.	10	2

5. Attempt any *one* part of the following:

a.	Describe the components of Cloud Provider. Also explain the responsibility of cloud provider for SaaS, PaaS, and IaaS.	10	3
b.	Explain Amazon S3. Also describe the advantages of cloud storage.	10	3

6. Attempt any *one* part of the following:

a.	Explain inter cloud resource management in cloud computing.	10	4
b.	Discuss security challenges and security governance in cloud computing environment.	10	4

7. Attempt any *one* part of the following:

a.	Give a suitable definition of cloud federation stack and explain it in detail.	10	5
b.	Explain Web services in detail. Differentiate Web services and APIs.	10	5