

---

## Question Bank

### UNIT – 2

#### 1. What are cloud computing technologies? justify few cloud computing technologies and its applications

##### Cloud Computing Technologies and Their Applications

Cloud computing refers to the delivery of computing services—such as servers, storage, databases, networking, software, and analytics—over the internet, enabling users to access and use resources without needing physical infrastructure.

- **Cloud Computing Technologies**

1. Virtualization

- Virtualization allows multiple virtual machines (VMs) to run on a single physical machine.
- Example: VMware, Microsoft Hyper-V, KVM.
- Application: Used in data centers for efficient resource management, allowing businesses to run multiple OS environments on the same hardware.

2. Containerization

- Containers package applications and their dependencies, ensuring consistency across environments.
- Example: Docker, Kubernetes.
- Application: Used in DevOps for CI/CD (Continuous Integration/Continuous Deployment), making applications scalable and portable across different cloud environments.

3. Serverless Computing

- Allows developers to run applications without managing the underlying infrastructure.
- Example: AWS Lambda, Google Cloud Functions, Azure Functions.
- Application: Used in event-driven applications like chatbots, IoT data processing, and real-time notifications.

4. Edge Computing

- Processes data closer to the source rather than in centralized cloud data centers.
- Example: AWS IoT Greengrass, Azure IoT Edge.
- Application: Used in real-time analytics for IoT devices, smart city infrastructure, and autonomous vehicles.

## 5. Cloud Security Technologies

- Focuses on securing cloud environments with encryption, authentication, and access controls.
- Example: Cloudflare, Microsoft Defender for Cloud.
- Application: Protects sensitive data in cloud storage, prevents cyberattacks, and ensures compliance with regulations like GDPR.

- **Applications of Cloud Computing**

- Healthcare:
  - Cloud-based AI for medical image analysis (e.g., detecting respiratory diseases using deep learning).
  - Telemedicine and remote patient monitoring.
- Finance:
  - Online banking, fraud detection, and risk analysis using cloud-based AI models.
  - High-frequency trading powered by cloud computing.
- Education:
  - Online learning platforms like Google Classroom, Coursera, and edX.
  - Virtual labs for hands-on experiments.
- Entertainment & Media:
  - Cloud-based video streaming services (Netflix, YouTube).
  - Gaming platforms using cloud gaming (Google Stadia, NVIDIA GeForce Now).
- AI & Machine Learning:
  - Cloud-based ML model training (Google AI, AWS SageMaker).
  - Speech recognition, language translation, and recommendation systems.

## **2. Justify the process of connecting multiple inline servers across external servers via ISP (Internet Service provider) network.**

Connecting multiple inline servers across external servers via an Internet Service Provider (ISP) involves establishing a reliable and secure network infrastructure that enables seamless communication, data transfer, and resource sharing between geographically dispersed servers. This process is often justified by the following reasons:

- 1) **Scalability:** Connecting multiple servers across external locations allows organizations to scale their operations horizontally. Inline servers can share the workload, and additional servers can be added as demand increases.
- 2) **Improved Security and Compliance:** ISPs often provide secure connections (e.g.,

VPNs, encrypted tunnels) that protect data in transit. Additionally, distributing servers across locations can help organizations comply with data sovereignty laws and regulations.

- 3) **Cost Efficiency:** Leveraging an ISP network to connect servers can be more cost-effective than building and maintaining a private network. ISPs provide the necessary infrastructure, reducing the need for significant capital investment.
- 4) **Centralized Management and Monitoring:** By connecting servers via an ISP network, organizations can centralize management and monitoring tools, making it easier to oversee the entire infrastructure.
- 5) **Flexibility & Remote Access:** ISP networks support remote access, enabling employees and administrators to manage servers from different locations without being physically present.
- 6) **Disaster Recovery & Redundancy:** Distributing servers across multiple locations improves fault tolerance. If one server fails, traffic can be redirected to another, ensuring business continuity and minimizing downtime.
- 7) **Improved Performance & Reduced Latency:** Connecting servers through an ISP allows organizations to place servers closer to end-users. This reduces latency, improves response times, and enhances user experience.

### **3. What are the connectionless packet switching and route based interconnectivity? Justify with flow diagram to demonstrate the packet traveling via a datagram network.**

- Connectionless Packet Switching (Datagram Switching)

In a connectionless packet-switched network, each packet is treated independently and is routed based on the destination address it carries. There is no need to establish a dedicated connection before sending packets, making the network more flexible but potentially leading to variable delays and out-of-order delivery.

- **Characteristics:**
  - Each packet is independent and can take different routes.
  - No prior setup or teardown of connections is required.
  - Packets may arrive in a different order.
  - Suitable for bursty traffic and fault-tolerant applications.

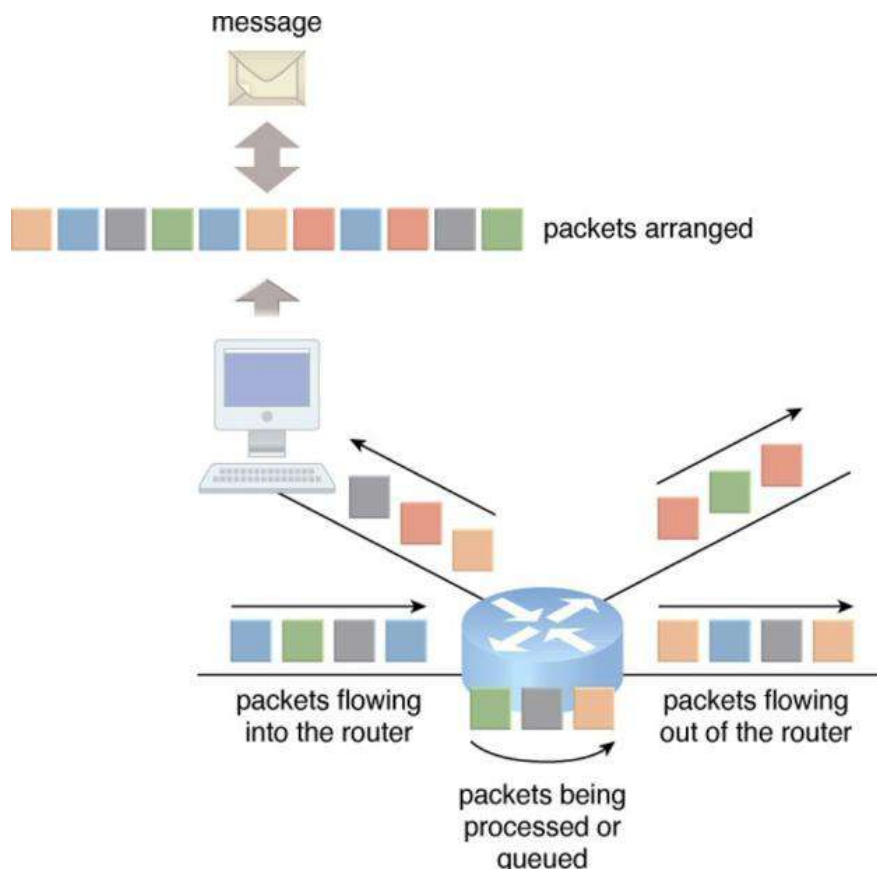
- Route-Based Interconnectivity

Route-based interconnectivity refers to the method of routing packets through a network based on the routing tables maintained at each node. The routers use protocols such as RIP, OSPF, or BGP to determine the best path for forwarding

packets dynamically.

- Justification with Flow Diagram

Below is a flow diagram demonstrating packet traveling via a datagram network:



#### 4. Explain with an architectural diagram of the internetworking with a diagram.

Internetworking refers to the practice of connecting multiple networks together to create a larger, integrated network that allows communication across different systems and environments. It involves using various devices such as routers, switches, gateways, and protocols to manage traffic and enable devices in different networks to exchange information. The goal of internetworking is to make sure that devices on separate networks can communicate efficiently and securely, whether those networks are local (LANs), wide-area (WANs), or even the global internet.

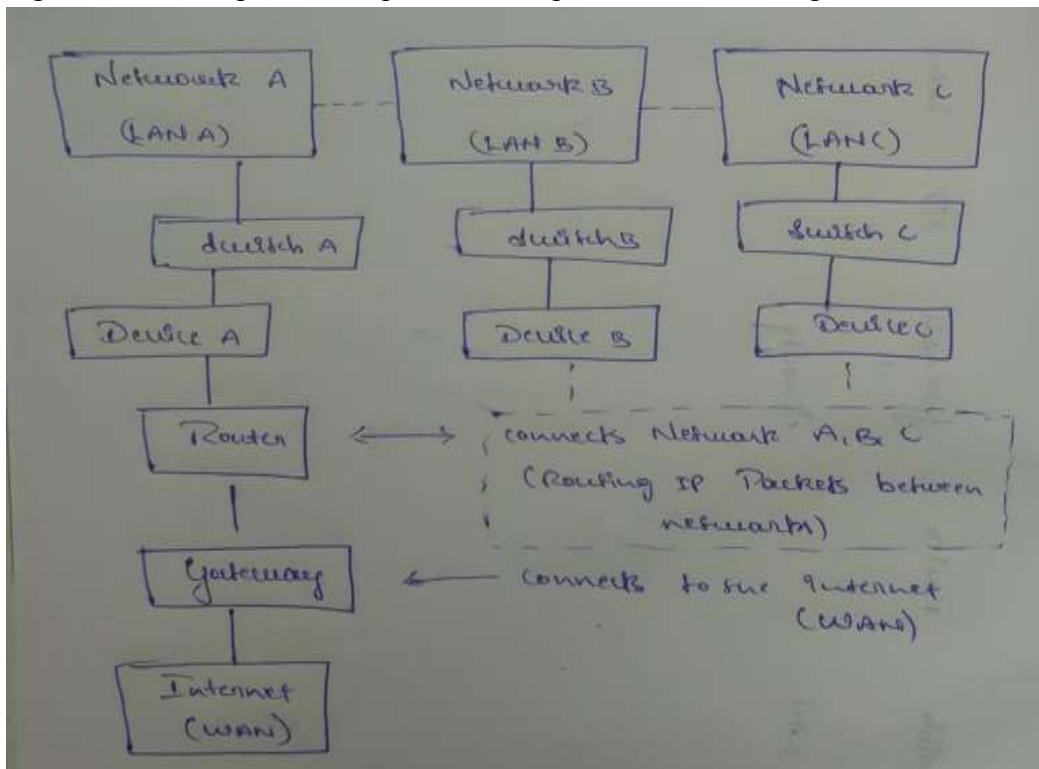
It consists of

- Routers: Routers are devices that connect multiple networks and route data between them. They determine the most efficient path for data packets to travel from the source to the destination.
- Switches: Switches operate within a single network and connect multiple devices.

They forward data based on MAC (Media Access Control) addresses and ensure that data is delivered to the correct device.

- Gateways: Gateways are devices that connect different types of networks, especially when they use different communication protocols. They enable communication between networks that cannot otherwise directly communicate (e.g., LAN to WAN).
- Protocols: Protocols are the rules that define how data is transmitted across the network. Common protocols used in internetworking include TCP/IP, HTTP, DNS, and FTP. These protocols ensure that data is formatted, routed, and delivered correctly.
- Devices:

Devices such as computers, smartphones, printers, or servers are the endpoints in a network. These devices communicate via switches, routers, and other components, sending and receiving data based on predefined protocols. Explanation of the Diagram:



- a) Network A, B, and C (LANs): Local Area Networks (LANs) are confined to a small geographic area, such as a building, office, or campus. In the diagram, Network A, Network B, and Network C represent different local networks. Each network can have its own set of devices and equipment, and they operate independently but need to communicate with other networks.
- b) Switches (Switch A, B, C): Switches are devices used to connect multiple devices (computers, printers, etc.) within a network. Switches operate primarily at the Data Link Layer (Layer 2 of the OSI model) and use MAC addresses to forward data. Switch A connects devices in Network A, Switch B connects devices in Network B, and so on.

Switches improve the network's efficiency by forwarding data only to the specific device that needs it (unlike hubs, which broadcast to all devices).

- c) Devices (A, B, C): These are the end-user devices that participate in the network. Examples of devices include computers, printers, servers, and smartphones. Device A, for instance, could be a computer in Network A, while Device B might be a printer in Network B.
- d) Router: The router connects different networks. It operates primarily at the Network Layer (Layer 3) and uses IP addresses to route data between different networks. In the diagram, the router connects Network A, Network B, and Network C to ensure that data packets can travel between these networks. Routers use routing tables and protocols like IP to determine the best path for data to reach its destination.
- e) Gateway: A gateway is a device that connects networks that use different protocols (e.g., a LAN to the internet or between different types of LANs). In the diagram, the gateway connects the local networks (A, B, C) to the Internet (WAN). It translates communication from one protocol (used within the LANs) to another (used in the internet or WAN). Gateways are also responsible for security functions, such as firewalls and Network Address Translation (NAT).
- f) Internet (WAN): The internet is a vast Wide Area Network (WAN) that connects devices globally. It is a network of networks that facilitates global communication. In the diagram, the Gateway connects the local networks to the Internet, allowing data to flow between devices in different LANs and external systems on the internet.

### 5. Compare the cloud services of "on premise IT services" & "cloud base IT services".

| Feature                  | On-Premises IT Services                          | Cloud-Based IT Services                              |
|--------------------------|--|--|
| Infrastructure Ownership | Managed and maintained by the organization       | Managed by a cloud provider (AWS, Azure, GCP)        |
| Initial Cost             | High (hardware, software, setup, maintenance)    | Low (pay-as-you-go or subscription model)            |
| Scalability              | Limited; requires purchasing additional hardware | Highly scalable; resources can be adjusted as needed |
| Maintenance & Upgrades   | IT team is responsible for updates and security  | Managed by the cloud provider                        |
| Security & Compliance    | Full control over security measures              | Security managed by provider, but compliance varies  |
| Accessibility            | Limited to company premises or VPN access        | Accessible from anywhere with an internet connection |
| Performance              | Faster for local processing; lower latency       | May experience latency, depending on internet speed  |

|                   |  |  |
|-------------------|--|--|
| Disaster Recovery | Requires in-house backup and disaster recovery plans | Built-in redundancy and disaster recovery options              |
| Customization     | Highly customizable to business needs                | Limited customization depending on provider                    |
| Data Control      | Full control over data location and security         | Data stored on third-party servers; may have compliance issues |
| Deployment Time   | Slow; requires procurement, installation, and setup  | Fast; services can be deployed in minutes                      |

- Which One to Choose ?
  - Choose On-Premises if you require full control, high security, customization, and are willing to manage infrastructure.
  - Choose Cloud-Based if you prefer scalability, low upfront costs, automatic updates, and remote accessibility.

#### **6. What is data? Is accumulation of data across multiple servers and extending these clusters are known as data centre justify?**

Data refers to raw facts, figures, or pieces of information that can be processed or analyzed to gain insights (information). It can be of many forms, including numbers, text, images, audio, or even video. Data can be collected, stored, and analyzed to support decision-making, problem-solving, and other activities across various domains like business, science, and technology.

A data center is a facility used to store and manage large volumes of data across multiple servers, networking equipment, and other IT infrastructure. A data centre is not just about accumulating (increasing) data but involves a sophisticated (experience) environment that handles the following:

- **Storage:** Storing vast amounts of data across physical and virtual servers.
- **Processing:** Using powerful servers to process the data for various applications.
- **Networking:** Ensuring that data is transmitted efficiently between servers, users, and other data centres.
- **Management:** Managing the flow of data, ensuring security, and maintaining the health of the infrastructure.
- **data accumulation across multiple servers and the extension of clusters,** it refers to the way data is stored in distributed systems. These systems involve multiple servers or nodes that work together as a cluster, allowing data to be stored in different physical locations but managed as a unified whole.
- **Scalability:** Data centres are designed to grow as demand increases. By extending clusters and adding more servers, a data centre can handle more data and more traffic.



- **Redundancy:** Storing data across multiple servers ensures that even if one server or part of the infrastructure fails, the system can continue to function without losing critical data.
- **High Availability:** Extending clusters across multiple servers or even multiple data centres can ensure that data is always available, minimizing downtime.
- **Distributed Computing:** Large data operations (such as big data analytics) require distributed computing, where clusters of servers work in parallel to process large volumes of data quickly.

The accumulation of data across multiple servers and extending these clusters to handle growing amounts of data is a key characteristic of data centres. These centres ensure efficient storage, processing, and retrieval of data, enabling businesses and other organizations to handle increasing data demands effectively.

**7. What is virtualization? Is virtualization a process of replanting physical hardware resources or integrating physical hardware resources. Justify.**

Virtualization is the creation of virtual versions of physical hardware resources. This can include virtual machines (VMs), virtual storage devices, virtual networks, and more. It allows multiple virtual instances to run on a single physical machine, creating the illusion of dedicated resources even though they are shared.

Virtualization is more about replicating physical hardware resources rather than directly integrating them.

- **Replication of Resources:** Virtualization involves creating virtual instances of physical resources (e.g., CPU, memory, storage) through software. These virtual instances, known as virtual machines or containers, behave as though they have dedicated hardware, even though they share the underlying physical resources.
- For instance, a single physical server (with, say, 64GB of RAM and multiple CPU cores) can host multiple virtual machines, each with its own operating system and application, as though they were running on separate physical hardware.
- **Integration of Resources (Less Relevant in Virtualization):** While virtualization does allow better utilization of the physical resources (i.e., integrating these resources in a more efficient manner), the primary concept is not about integrating them but rather abstracting, partitioning, and allocating them in a flexible, isolated manner.
- Virtualization abstracts the physical hardware, integrating it with software to make it more flexible and efficient, but it does not combine or merge the resources in a way that changes their fundamental physical nature.



**8. An hypervisor is an interfacing entity between the virtual server and physical servers. Can network architecture under a hypervisor be improved with respect to minimum physical resources?**

Yes, network architecture under a hypervisor can be improved to use minimal physical resources. This can be achieved through techniques like network virtualization, software-defined networking (SDN) (centralized control of the network), and network function virtualization (NFV) (virtualizing network services). These methods reduce the need for additional physical hardware and make better use of the existing physical resources

- **Virtual Switches (vSwitches)**  
A virtual switch (vSwitch) allows multiple virtual machines (VMs) to communicate internally with one another and with the physical network through a given physical server. vSwitches avoid the necessity for a physical network switch, thereby reducing overall hardware requirements.
- **Distributed Virtual Routers**  
Distributed virtual routers are not confined to a single physical router; they could be distributed across the virtualized infrastructure. This evenly distributes the networking load, minimizing the use of large, centralized physical network hardware, hence improving the efficiency of resource use.
- **Network Function Virtualization (NFV)**  
Network functions like firewalls, load balancers, and routers can be performed on virtual network functions instead of on dedicated physical appliances. This cuts down on the need for additional hardware while also consolidating network services onto fewer physical servers.
- **Software-Defined Networking**  
SDN centralizes control of the network, issuing commands for traffic instead of letting individual physical networking devices function independently. SDN is dynamic in resource allocation to active users, thus boosting physical network efficiency and reducing hardware dependence.

**9. In data centers, standardization, modularity, automation, remote operation, and management are demanding entities. Justify each of these with a supporting case study.**

- **Justification of Entities with Case Studies**
  - Modern data centers are built on principles of standardization, modularity, automation, and remote operation & management to ensure efficiency, scalability, and reliability. Each of these entities plays a crucial role in optimizing performance, reducing operational costs, and improving service delivery. Below is a justification of each entity along with a supporting case

---

study.

- **Standardization:** Standardization ensures uniformity in hardware, software, and networking components, reducing complexity, and enhancing interoperability. It enables seamless scaling and easier maintenance while ensuring compliance with global standards.
- **Case Study: Facebook Data Centers**  
Facebook adopted the Open Compute Project (OCP), which standardizes server and data center hardware designs. This initiative helped reduce power consumption and improved efficiency by 38%, demonstrating the advantages of a standardized approach.
- **Modularity:** Modularity allows data centers to expand and adapt to changing needs without disrupting operations. Using modular infrastructure, organizations can add capacity dynamically and optimize space and power utilization.
- **Case Study: Google’s Modular Data Centers**  
Google developed modular data centers in shipping containers to rapidly scale operations. These modules are pre-configured with servers, cooling systems, and power supplies, enabling quick deployment and efficient resource management.
  - Automation enhances efficiency by reducing manual intervention in routine operations such as provisioning, monitoring, and troubleshooting. AI-driven automation also predicts failures and optimizes resource allocation.
- **Case Study: Microsoft’s AI-Powered Data Centers**  
Microsoft leverages AI-driven automation in Azure data centers to monitor server health, predict failures, and optimize workloads. This has significantly reduced downtime and operational costs while improving energy efficiency.
  - **Remote Operation & Management** Remote operation and management allow administrators to monitor and control data centers from anywhere, enhancing resilience and disaster recovery capabilities. This is crucial for large-scale and geographically distributed data centers.
- **Case Study: Equinix’s Remote Data Center Management**  
Equinix uses Smart Hands, a remote management solution, to provide 24/7 monitoring and troubleshooting. This minimizes the need for physical presence, ensuring continuity of operations even in emergency situations.

In conclusion, these key entities—standardization, modularity, automation, and remote management—are indispensable in modern data center operations. Organizations implementing these principles benefit from enhanced efficiency, cost savings, and improved reliability, as demonstrated by the case studies above.

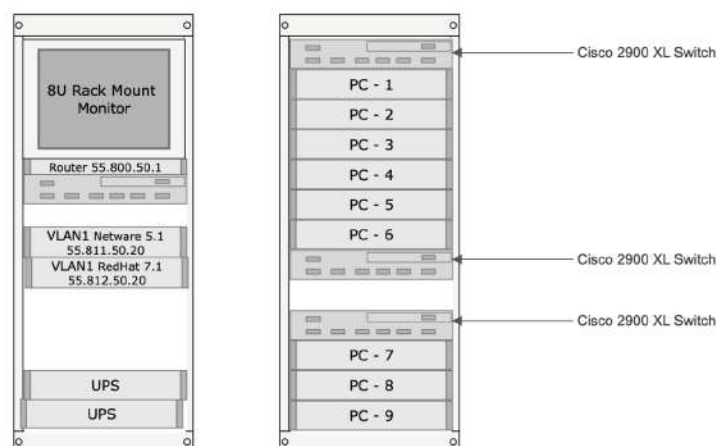
**10. Is computing hardware a synonym for cloud hardware services? Justify the hardware capabilities rackmount server & Blade server with a neat diagram.**

No, computing hardware is not a synonym for cloud hardware services. Computing hardware refers to the physical components of a computer system, such as processors, memory, storage, and networking equipment. This includes desktops, laptops, servers, and embedded systems. Cloud hardware services refer to virtualized computing resources provided by cloud service providers (e.g., AWS, Azure, Google Cloud). These services run on specialized data center hardware, including rackmount servers and blade servers, but they offer computing as a service over the internet.

Rackmount Server vs. Blade Server

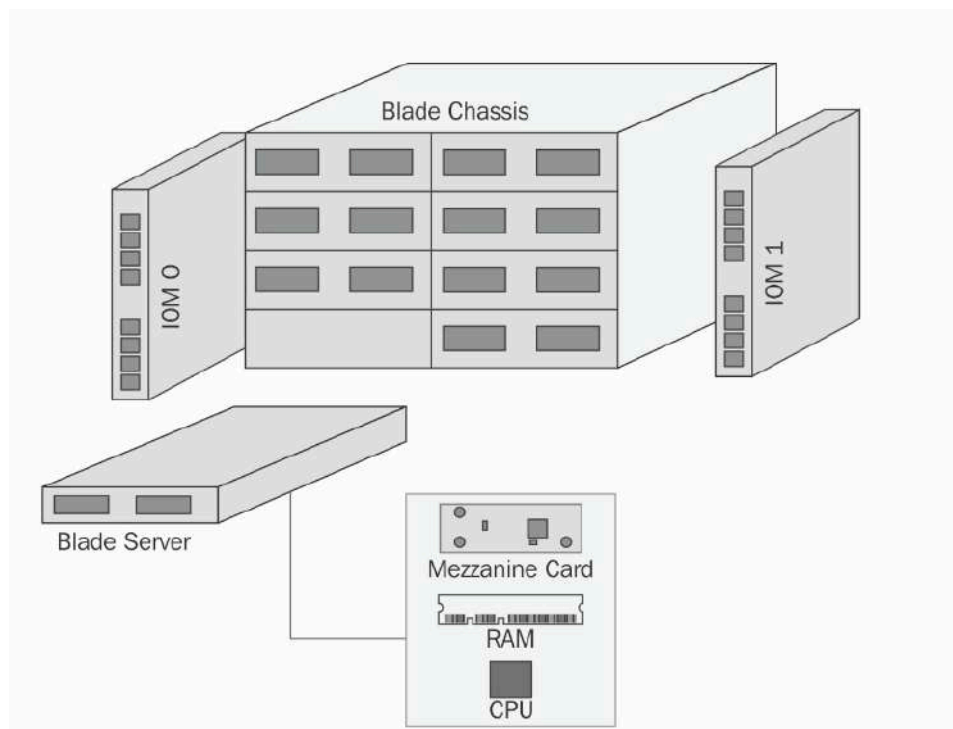
| Feature         | Rackmount Server                                      | Blade server                                  |
|-----------------|---|---|
| Structure       | Independent Servers housed in a rack                  | Modular Servers (blades) in a chassis         |
| Size            | 1U, 2U,4U (based on height)                           | Slim blades, high density                     |
| Power & cooling | Individual power supply & cooling per unit            | Shared power & cooling systems                |
| Scalability     | Moderate (limited by rack space)                      | High (blades can be added easily)             |
| Cost Efficiency | Costlier in power & cooling per unit                  | More cost-effective for large-scale computing |
| Ideal Use Case  | Small to medium businesses, general-purpose computing | Large data centers, cloud computing, HPC      |

Rack Diagram





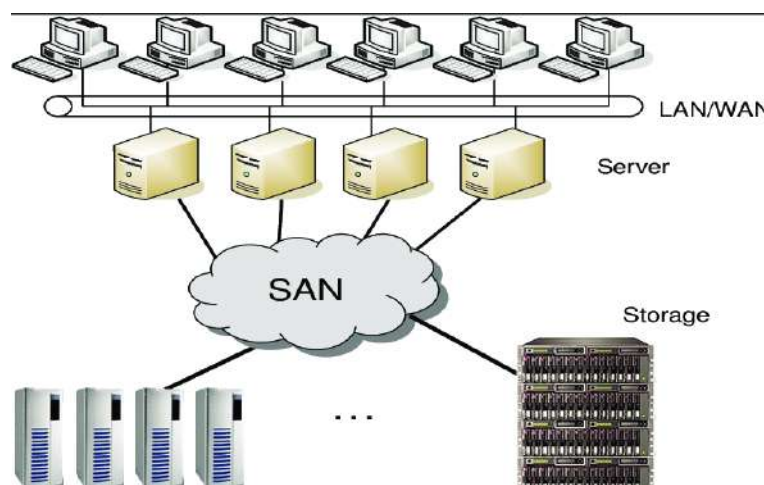
### Blade Server's



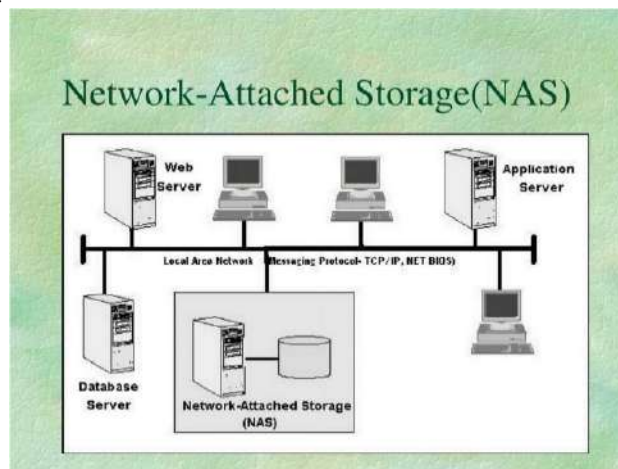
### 11. What is storage area network and network attached storage. Justify with neat diagram.

When managing data in an IT environment, understanding storage options is crucial. Storage Area Network (SAN) and Network Attached Storage (NAS) are two primary technologies used for data storage. SAN is a high-speed network that connects servers to storage devices, making it ideal for enterprises with high data demands. NAS, on the other hand, is a cost-effective solution that connects to a network, enabling easy file sharing. Both have distinct advantages depending on storage needs.

- What is a Storage Area Network (SAN)?  
SAN is a high-speed network used for data transfer between servers and storage devices through fiber channels and switches. In SAN, data is identified by disk block, and common protocols include SCSI and SATA.
- Components of SAN:
  - Node ports
  - Cables
  - Interconnect devices (Hubs, switches, directors)
  - Storage arrays
  - SAN management software
- How SAN Works?
  - SAN is a dedicated high-speed network that links multiple servers to storage devices, avoiding network congestion. It uses Fibre Channel or iSCSI protocols for fast data transfer, improving scalability, performance, and centralized storage management.
  - SAN Use Cases:
    - Disaster Recovery: SAN helps in data replication across geographically dispersed sites, ensuring data availability.
    - Enterprise Applications: SAN is used in large databases, virtualization, and high-speed computing environments.



- What is Network Attached Storage (NAS)?  
NAS identifies data by file name and byte offset. The file system is managed by the Head Unit (CPU & Memory) instead of the server. Unlike SAN, NAS uses file-based storage protocols.
- Components of NAS:
  - Head Unit (CPU, Memory)
  - Network Interface Card (NIC)
  - Optimized OS
  - Storage Protocols: ATA, SCSI, Fibre Channel
- How NAS Works?  
NAS connects directly to a network via Ethernet, allowing multiple devices to access shared files. It supports SMB, NFS, and AFP protocols, offering file redundancy (RAID), backups, and access controls. Users can access it like a local drive, managed via a web-based interface.
- NAS Use Cases:
  - File Sharing: Ideal for small offices, home setups, and team collaborations.
  - Data Backup & Recovery: Used for centralized backups and disaster recovery.
  - Media Streaming: Functions as a home media hub for music, videos, and photos.



Key Differences Between SAN and NAS

| Feature                | SAN<br>(Storage Area Network)      | NAS<br>(Network Attached Storage) |
|------------------------|------------------------------------|-----------------------------------|
| Data Identification    | Disk block                         | File name & byte offset           |
| File System Management | Managed by servers                 | Managed by the Head Unit          |
| Cost                   | More expensive                     | Cost-effective                    |
| Complexity             | Complex                            | Simple                            |
| Protocols Used         | SCSI, SATA                         | CIFS, NFS, SMB                    |
| Backup & Recovery      | Block-by-block copying             | File-based backup                 |
| Performance            | High-speed                         | Network-dependent                 |
| Management             | Requires expertise                 | Easy to manage                    |
| Network Dependency     | Uses Fibre Channel (independent of | Uses TCP/IP, depends on           |



|                               |                             |                                 |
|-------------------------------|-----------------------------|---------------------------------|
|                               | LAN)                        | LAN                             |
| <b>Use Cases</b>              | Enterprises, virtualization | Small businesses, homes         |
| <b>Latency</b>                | Low                         | Higher compared to SAN          |
| <b>Virtualization Support</b> | Yes                         | No                              |
| <b>Network Traffic Impact</b> | Not affected                | Affected by network bottlenecks |

**12. As network hardware dependencies monitored via external service providers web tier load balancing, LAN fabric, SAN fabric and NAS gateway. Justify?**

- Network hardware dependencies are monitored via external service providers to ensure smooth operation and reliability of the system. Monitoring helps in detecting issues early, improving performance, and reducing downtime.
- Load balancing distributes network traffic across multiple servers, preventing overload and ensuring efficient resource utilization. This improves response times and keeps the system running smoothly even during high traffic.
- LAN fabric is the network infrastructure that connects different devices within a local area, enabling fast and secure communication between computers, servers, and other network components. It ensures data flows efficiently without congestion.
- SAN fabric is a specialized high-speed network that connects storage devices to servers, allowing quick and reliable data access. It enhances storage performance and ensures data is available whenever needed.
- NAS gateway provides shared access to storage over a network, making it easy for users and applications to retrieve and manage files. It supports centralized data storage, improving accessibility and security.
- By monitoring these components externally, organizations can ensure better performance, security, and minimal downtime.

**13. With a neat labelled diagram explain the process of virtual machine management and extend with an example of a hypervisor.**

- Overview of Virtual Machine Management (VMM)  
Virtual Machine Management (VMM) refers to the process of creating, configuring, monitoring, and controlling virtual machines (VMs) within a computing environment. It helps in optimizing resource allocation, improving security, and ensuring efficient operation of multiple virtual machines on a single physical system.
- Steps in Virtual Machine Management
- Step 1: Resource Allocation  
The Hypervisor (Hyper-V) allocates CPU, RAM, storage, and network resources to each VM.  
Ensures fair distribution among Parent VM and Child VMs.
- Step 2: Virtual Machine Creation



A new VM is created using the VMM Admin Console.

The administrator selects VM hardware specifications (CPU, RAM, storage).

- **Step 3: Operating System Installation**

The selected OS (e.g., Windows Server 2003/2008 or Linux) is installed on child VMs. Each VM operates independently inside the physical host.

- **Step 4: Virtual Machine Configuration**

Software applications, security settings, and network configurations are applied to each VM.

VMM Server ensures proper settings are implemented.

- **Step 5: Performance Monitoring & Optimization**

VMM Agent tracks CPU, memory, and disk usage of each VM.

The VMM Server ensures load balancing across VMs.

- **Step 6: VM Backup & Migration**

VMM Library stores snapshots and backup files of VMs.

Hyper-V supports Live Migration, allowing VMs to move between servers without downtime.

- **Step 7: Security & Isolation**

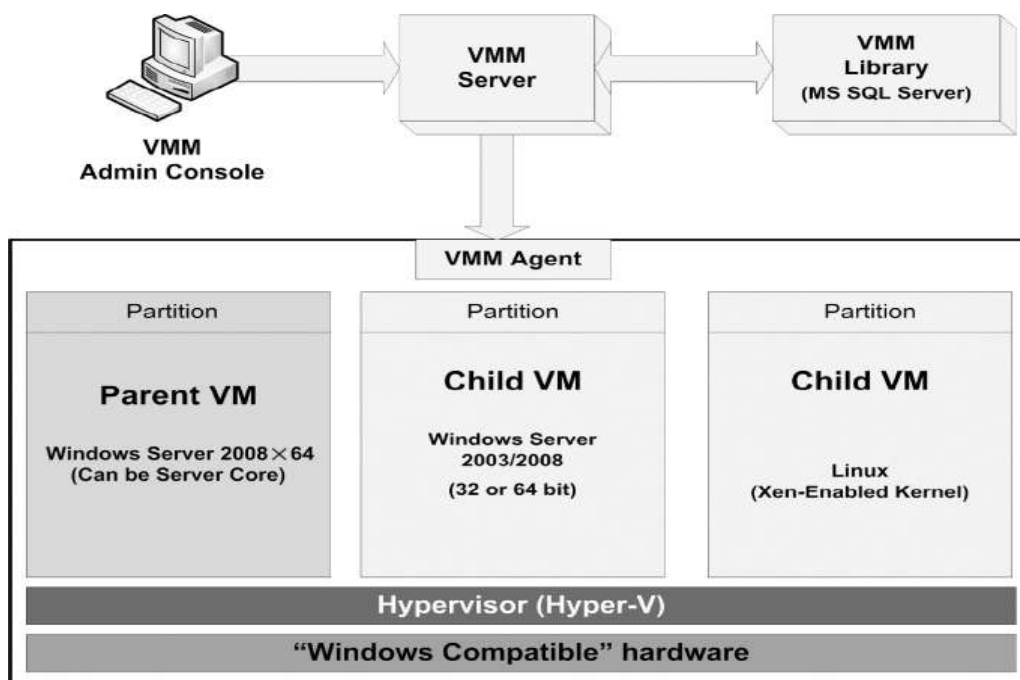
Each VM is in its own Partition (Parent VM and Child VMs are isolated).

Security features like firewalls, encryption, and access control prevent unauthorized access.

- **Step 8: VM Deletion or Reallocation**

Unused VMs are deleted to free up resources.

VMM Server reallocates resources for efficiency.



- Example of a Hypervisor – Microsoft Hyper-V
  - Hyper-V is a Type-1 Hypervisor by Microsoft that allows multiple virtual machines to run on a single physical server.
  - Example Scenario
  - A company installs Hyper-V on a Windows Server and manages multiple VMs:
    - VM 1 (Parent VM): Windows Server 2008 (Handles hardware interaction).
    - VM 2 (Child VM): Windows Server 2003 (Used for company applications).
    - VM 3 (Child VM): Linux OS (Runs web hosting services).
  - How Hyper-V Manages These VMs:
    - ✓ Efficient resource allocation among Windows and Linux VMs.
    - ✓ Secure isolation, preventing one VM from affecting another.
    - ✓ Supports Live Migration, allowing VMs to be moved without downtime.



**14. What is web-technology? justify: URL, HTTP, HTMP, XML using a DNS split the given URL into protocol domain, path and indexing.**

Web technology includes the tools, programming languages, and protocols that allow websites and web applications to function. It enables communication between web browsers (like Chrome or Firefox) and web servers, making it possible to create, deploy, and manage websites.

Explanation of Key Terms

- URL (Uniform Resource Locator)
  - A URL is the web address you type in your browser to access a website or an online resource. It consists of different parts like the protocol, domain name, and path.
  - 📌 Example:  
https://www.example.com/path/page.html?query=123
  - https:// → Protocol (secure communication)
  - www.example.com → Domain (website name)
  - /path/page.html → Path (specific page on the website)
  - ?query=123 → Query parameters (optional data sent to the server)
- HTTP (Hypertext Transfer Protocol)
  - HTTP is the set of rules that governs how web browsers and servers communicate. Whenever you visit a website, your browser sends an HTTP request to the server, which then responds with the webpage data.
  - 📌 Example:

When you type `www.google.com` in your browser and hit Enter, your browser sends an HTTP request to Google's servers, which send back the search engine page.

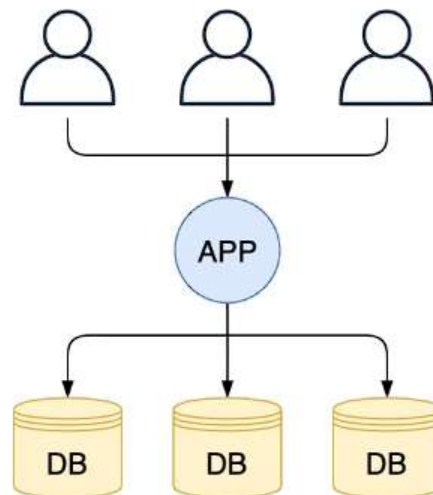
- **HTML (HyperText Markup Language)**
  - HTML is the building block of web pages. It tells the browser how to display content using different tags like `<p>` for paragraphs, `<img>` for images, and `<a>` for links.
  -  Example of an HTML snippet:
  - `<p>This is a paragraph.</p>`
  - `<a href="https://www.example.com">Click here</a>`
- **XML (eXtensible Markup Language)**
  - XML is used to store and transport data in a structured format. Unlike HTML, which is designed for displaying content, XML is used to organize and share data between applications.
  -  Example of XML data:
  - `<student>`
  - `<name>John</name>`
  - `<age>21</age>`
  - `</student>`
  - This XML file represents a student's information, which can be easily shared between systems.

### **15. What is multitenant technology. Justify with a neat labelled diagram and a case study?**

Multitenant technology is an architectural approach in which a single software application or infrastructure serves multiple customers (tenants), each with its own separate data, configurations, and access controls. This technology is commonly used in cloud computing and Software as a Service (SaaS) applications, allowing multiple users or organizations to share computing resources efficiently while maintaining data security and isolation.

- **Key Characteristics of Multitenant Architecture**
  - **Resource Sharing:** A single application instance runs on a server and serves multiple tenants.
  - **Data Isolation:** Each tenant's data is logically separated to ensure security.
  - **Customizability:** Tenants can customize some aspects of the application without affecting others.

- Scalability: The system can dynamically allocate resources based on demand.
- Cost Efficiency: Reduces infrastructure costs as resources are shared.



- Case Study: Salesforce – A Leading Multitenant SaaS Provider
  - Background

Salesforce, a cloud-based Customer Relationship Management (CRM) platform, is a prime example of a multitenant application. It allows multiple businesses (tenants) to use the same cloud-based CRM software while keeping their data private and secure.
  - How Multitenancy Works in Salesforce

Salesforce runs a single software instance on the cloud and serves multiple businesses.
  - Each business (tenant) accesses the application through a web interface.
  - Salesforce ensures that every tenant's data is logically separated using unique identifiers.
  - Customization features allow each tenant to tailor the application to their needs without affecting others.

## 16. What is REST (Representational State Transfer) services? Explain each in detail?

Representational State Transfer (REST) is an architectural style for designing networked applications, particularly web services, introduced by Roy Fielding in his 2000 dissertation. It operates on a stateless, client-server model, typically using HTTP as its protocol. REST centers on resources—entities like users, products, or documents—identified by URIs (Uniform Resource Identifiers). These resources are manipulated using standard HTTP methods aligned with CRUD operations (Create, Read, Update, Delete). REST is not a strict protocol but a set of constraints that promote scalability, simplicity, and interoperability, resulting in RESTful systems widely adopted in modern API design.

- **GET - Retrieve a Resource:** The GET method retrieves a representation of a resource or a collection of resources from the server. It's a read-only operation, ensuring no changes occur on the server, which makes it safe and idempotent. The client specifies the desired resource via a URI, and the server provides the data in a format negotiated between the two, such as JSON or XML. This method supports caching, a key REST principle, allowing clients or intermediaries to store responses for efficiency. It's fundamental for accessing information without altering the system's state.
- **POST - Create a Resource:** The POST method facilitates the creation of new resources on the server. The client sends data in the request body to a URI, often targeting a collection rather than a specific resource, and the server generates a new entity based on this input. Unlike GET, POST is neither safe nor idempotent, as repeated requests can produce multiple resources. This method is versatile, supporting complex payloads and triggering server-side logic beyond simple creation, such as initiating processes or workflows tied to the new resource.
- **PUT - Update a Resource:** The PUT method updates an existing resource or creates one if it doesn't exist, depending on the server's design. The client sends a complete representation of the resource to a specific URI, and the server overwrites the existing data with this new version. Its idempotent nature ensures that multiple identical requests yield the same result, providing reliability in update operations. PUT assumes the client has full knowledge of the resource's state, making it suitable for scenarios requiring a definitive replacement rather than incremental changes.
- **PATCH - Partially Update a Resource:** The PATCH method modifies a resource partially, targeting specific fields rather than replacing the entire entity. The client submits a request body with only the changes, and the server applies them to the existing resource. Unlike PUT, PATCH is not inherently idempotent unless the server enforces consistency, as the outcome depends on the resource's current state and the update instructions. This method offers flexibility, reducing bandwidth by sending minimal data, and is often used when full resource updates are unnecessary or impractical.
- **DELETE - Remove a Resource:** The DELETE method removes a resource from the server, identified by its URI. It's idempotent, meaning once the resource is deleted, further requests have no additional impact, enhancing predictability in operations. The server may retain internal records or mark the resource as inactive, but from the client's perspective, it's no longer accessible. DELETE is a straightforward way to manage resource lifecycle, though servers might impose restrictions or require authentication to prevent unauthorized removal.

Additional REST Considerations: Beyond these core methods, REST emphasizes constraints like statelessness, where each request contains all necessary information without relying on server-stored session data. Resources can have multiple representations (e.g., JSON, XML, or

plain text), and clients can negotiate the preferred format. REST also supports hypermedia, where responses include links to related resources, enabling dynamic navigation. These features collectively enhance REST's adaptability, making it a cornerstone of distributed systems and microservices architectures.

### **17. What is a cloud infrastructure mechanism made of? Explain supporting elements in detail.**

Cloud infrastructure mechanisms are the fundamental components that enable cloud computing services. They include various hardware and software resources that work together to provide computing power, storage, and networking capabilities. The key components of cloud infrastructure mechanisms are:

- **Hardware**  
Hardware includes the equipment needed to connect machines to a single cloud. Hardware components include servers, power supplies, memory and storage, and central processing units (CPUs). All of these features need to work together to provide the performance, security, and availability cloud users need.
- **Virtualization**  
Virtualization makes it possible to decouple computing infrastructure from the hardware that runs them. This is perhaps the most important part of a cloud infrastructure. Virtualization software plays an essential role, because it decouples data storage and computing power from the hardware itself. Virtualization also allows operators to manage their cloud infrastructure through a central user interface.
- **Storage**  
Cloud storage holds data, keeps the latest version of a file or data item and possibly also previous versions, and enables remote access as needed. Virtualization provides a link between hardware and cloud storage, enabling the three major cloud storage models:
  - Block storage—sorts data into blocks instead of complete files. This is an ideal solution for storing data that is static and does not change regularly.
  - File storage—like the file manager systems used with regular PCs.
  - Object storage—suitable for storing unstructured data, or data that needs to frequently change.
- **Network**  
Network infrastructure is also important for providing cloud computing services. This includes both internal networks, within the cloud environment, and external network connections to enable remote access. Network infrastructure covers many different types of hardware, including routers, switches, load balancers, and physical cables.

**18. What is an virtual server and justify with an architecture the process of connecting the virtual server with an active cloud service with minimal cloud balancing.**

A virtual server is a software-based server that emulates the functionality of a physical server. It runs an operating system and applications just like a physical server, but instead of being tied to specific hardware, it resides on a hypervisor (virtual machine monitor) that allows multiple virtual servers to run on the same physical hardware. Virtual servers are part of virtualization technology and are commonly used in cloud computing environments to provide scalable, flexible, and cost-effective infrastructure.

- Architecture for Connecting a Virtual Server with an Active Cloud Service (Minimal Cloud Balancing)
- To connect a virtual server to an active cloud service with minimal cloud balancing, the architecture can be broken down into the following components:
- Cloud Service Provider:
  - A cloud provider like AWS, Azure, or Google Cloud offers virtual machines (VMs) as part of their infrastructure services.
  - The cloud provider will have the necessary networking, storage, and compute resources to run virtual servers and other services.
- Virtual Machine/Server:
  - The virtual server is instantiated as a virtual machine (VM) on the cloud service provider's infrastructure.
  - The virtual server could be running on a hypervisor, such as VMware or KVM, within the cloud environment.
- Virtualization Layer (Hypervisor):
  - The hypervisor is responsible for abstracting the physical hardware and running the virtual servers. It allocates resources like CPU, memory, and storage to the virtual server.
- Cloud Load Balancer:
  - A minimal cloud load balancing mechanism can be employed to distribute traffic evenly to different virtual servers (if needed). This can be done through a software-based load balancer that operates at the application or network layer.
  - In the case of minimal balancing, you might choose a simple round-robin or least connections approach to distribute traffic between multiple instances of the virtual server.
- Cloud Active Service :
  - The active cloud service could be any cloud-based resource, such as a database service (e.g., Amazon RDS, Azure SQL Database) or an application (e.g., a microservice hosted on Kubernetes).
  - The cloud service could be accessed by the virtual server using secure API calls,



web service endpoints, or database connections.

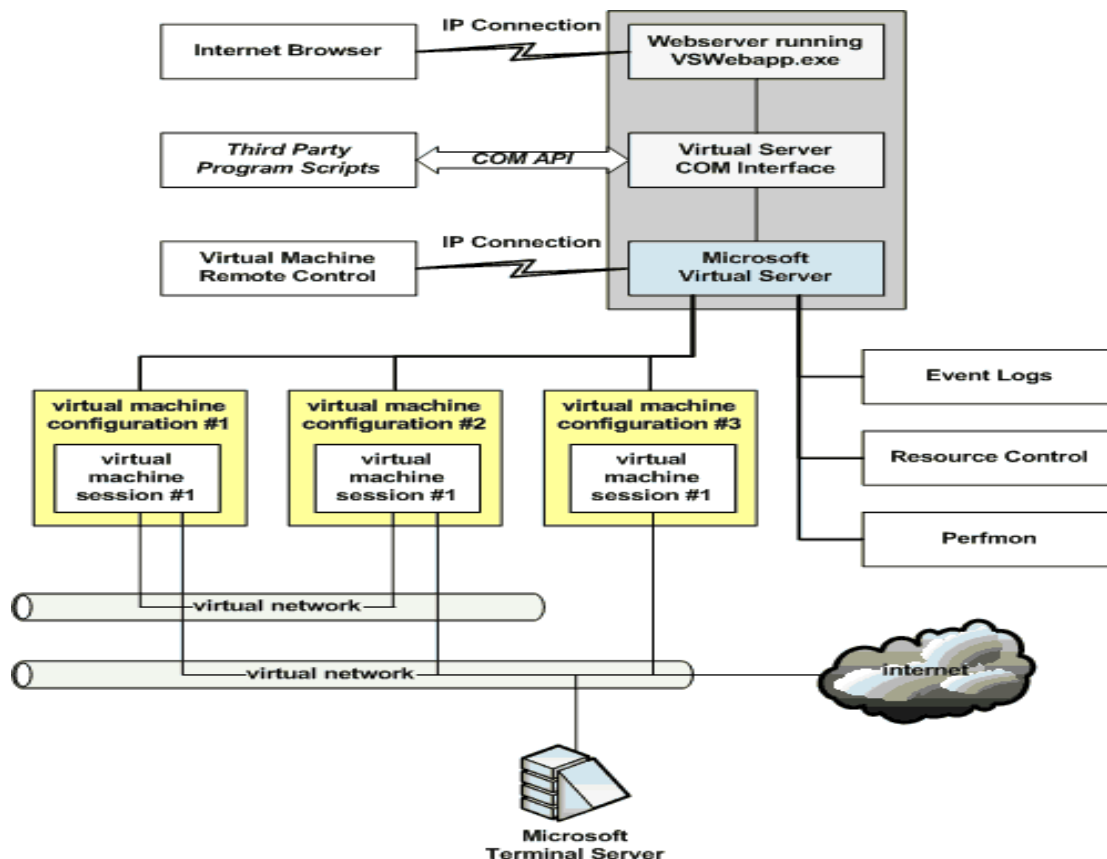
- e.g., Database, Application, or Storage
- Networking Layer:
  - Virtual Private Cloud (VPC): The virtual server resides inside a VPC, which is a logically isolated network within the cloud. VPC enables secure communication between the virtual server and other cloud resources, like cloud services or other virtual machines.
  - Subnets: The virtual server and active cloud service are often placed in specific subnets for organization and security. Public and private subnets can be used to isolate services from each other.
  - Security Groups and Firewall: Virtual servers and cloud services need to have firewalls or security groups configured to control inbound and outbound traffic. Only authorized traffic should be allowed between the virtual server and the cloud service.

#### **Process of Connecting Virtual Server to Active Cloud Service:**

- Create the Virtual Server:
- Deploy a virtual machine (VM) in the cloud provider's environment. You can configure the VM with the necessary resources, like CPU, RAM, and storage.
- Connect to Active Cloud Service:
- The virtual server connects to the active cloud service (e.g., a database or an application). This can be done via:
  - API Calls: If the cloud service provides a RESTful API or similar, the virtual server can make HTTP requests to interact with the service.
  - Database Connection: If the cloud service is a database, the virtual server can establish a direct connection using secure protocols (e.g., SSL for MySQL or PostgreSQL).
- Minimal Cloud Balancing:
  - A minimal load balancing approach can be employed in the following ways:
  - Single Active Instance: If the load is manageable, only a single instance of the virtual server may be running, and the cloud service can communicate directly with it.
  - Round Robin DNS or Simple Load Balancer: If there are multiple virtual server instances, a simple load balancer or round-robin DNS can distribute the load with minimal complexity.
- Network Configuration:
  - Ensure that networking components like VPCs, subnets, and security groups are configured to allow traffic between the virtual server and the active cloud service. Public IPs or private IPs may be assigned to the virtual server, depending on whether the service is exposed to the internet.
- Monitoring and Scaling (Optional):

- If scaling is needed, cloud services like auto-scaling groups can be used, but in this case, the focus is on minimal cloud balancing, so only basic scaling may be employed.

Example Architecture Diagram:



In this architecture:

- The Cloud Load Balancer distributes the traffic between the virtual servers (VMs).
- The Virtual Server(s) communicate with the Active Cloud Service (e.g., a database or application) to fetch or send data.
- The communication happens over a secure network setup within the cloud.
- Justification:
- Flexibility: The virtual server model allows easy scaling and isolation of workloads, ideal for cloud environments.
- Cost-Effectiveness: Virtual servers make use of underlying hardware more efficiently, sharing resources among multiple virtual instances, thus reducing hardware costs.
- Minimized Cloud Balancing: The load balancing setup in this architecture is minimal, meaning it focuses on low-cost or low-complexity methods like round-robin DNS or a simple load balancer to distribute traffic without heavy complexity or unnecessary resource consumption.

### 19. What are cloud based management interfaces. Justify with a diagram

Cloud-based management interfaces are centralized platforms that enable users to interact with and control cloud resources efficiently. These interfaces provide tools for provisioning, monitoring, scaling, and securing cloud infrastructure.

Organizations use cloud-based management interfaces to handle various cloud services like virtual machines, databases, networking, and security configurations. These interfaces ensure that businesses can operate their cloud environments without needing direct access to the underlying hardware.

They are essential for:

- Ease of Use – Web portals provide a user-friendly interface, while APIs and CLI enable automation.
- Scalability – Dynamically scales resources based on demand.
- Security – Implements strict access control, encryption, and compliance policies.
- Cost Optimization – Helps monitor cloud spending and optimize costs.
- Automation – Supports DevOps workflows, allowing continuous deployment.

Types of Cloud-Based Management Interfaces

- Web-Based Graphical User Interface (GUI)

Definition: A visual dashboard accessed via a web browser that allows users to manage cloud services through clicks and menus.

Use Case: Best for beginners and administrators who prefer a user-friendly interface.

Examples:

- AWS Management Console
- Microsoft Azure Portal
- Google Cloud Console

- Command-Line Interface (CLI)

Definition: A text-based interface that allows users to execute cloud commands through scripts or shell terminals.

Use Case: Preferred by developers and system administrators for automation and bulk operations.

Examples:

- AWS CLI
- Azure CLI
- Google Cloud SDK

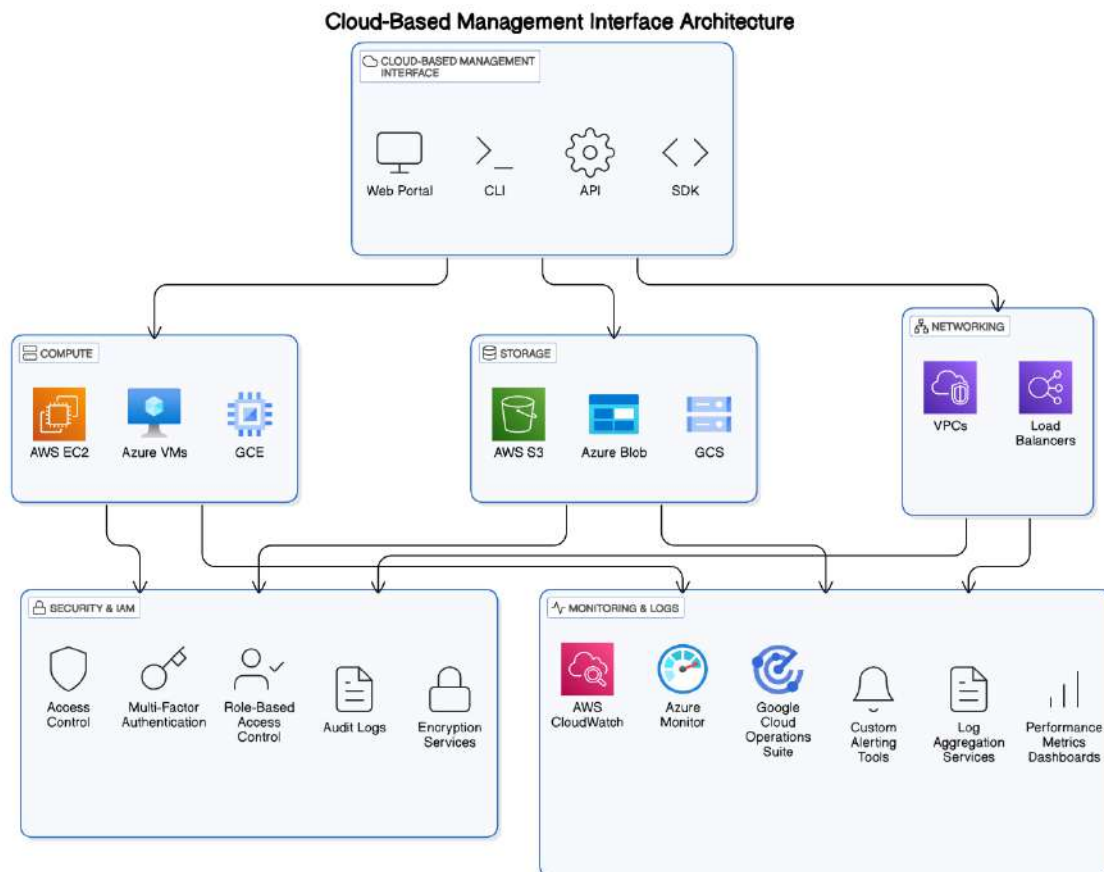
- Application Programming Interfaces (API) & Software Development Kits (SDK)

Definition: APIs provide programmatic access to cloud services, while SDKs offer pre-built libraries to integrate cloud functionalities into applications.

Use Case: Used by developers to integrate cloud functionalities into applications.

Examples:

- AWS API Gateway
- Azure REST API
- Google Cloud Client Libraries



### 1. Cloud-Based Management Interface (Top Layer)

- Users interact with the cloud through different interfaces like web portals, CLI, APIs, or SDKs.
- These interfaces allow provisioning, monitoring, and managing cloud resources.

### 2. Core Cloud Services (Middle Layer)

- Compute: Virtual Machines (AWS EC2, Azure VMs, Google Compute Engine) for running applications.
- Storage: Cloud storage services (AWS S3, Azure Blob Storage, Google Cloud Storage) for data management.
- Networking: Load balancers, Virtual Private Cloud (VPC), and firewalls ensure secure network communication.

### 3. Security and Monitoring (Bottom Layer)

- Security & IAM: Ensures access control through authentication and authorization.
- Monitoring & Logs: Collects performance data and logs for system monitoring and troubleshooting.

## **20. Is cloud usage monitoring an important entity for infrastructural mechanism? Justify with the help of monitoring, resource and cooling agent.**

Yes, cloud usage monitoring is an essential entity for the infrastructural mechanism. It ensures efficient resource utilization, cost optimization, and system reliability. The justification can be explained using the following aspects:

- Monitoring:
  - Tracks CPU, memory, storage, and network usage in real time.
  - Detects system anomalies and potential failures before they impact services.
  - Helps in optimizing workload distribution across virtual machines.
- Resource Management:
  - Ensures dynamic resource allocation based on demand.
  - Prevents resource over-provisioning and under-utilization.
  - Enhances cloud elasticity by scaling resources up or down as needed.
- Cooling Agent:
  - Monitors data center temperature and energy consumption.
  - Optimizes cooling mechanisms to prevent overheating and reduce power usage.
  - Contributes to sustainability by minimizing energy waste.

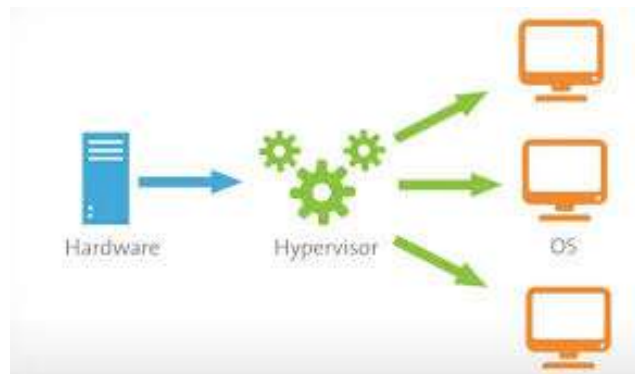
## **21. What is resource replication? With a diagram justify the role of hypervisor in bi-parting physical servers and cloud and services across virtual services.**

Resource replication refers to the process of duplicating computing resources such as storage, virtual machines (VMs), or applications across multiple locations to ensure availability, reliability, and fault tolerance in cloud computing. This technique is widely used in distributed systems and cloud environments to enhance performance and redundancy.

### **Role of Hypervisor in Virtualization**

A hypervisor (or Virtual Machine Monitor, VMM) is software that enables multiple virtual machines (VMs) to run on a single physical server. It creates an abstraction layer that divides physical resources (CPU, RAM, storage) among VMs, enabling efficient resource utilization and scalability in cloud environments.

### Diagram Representation



### Justification of the Role of Hypervisor

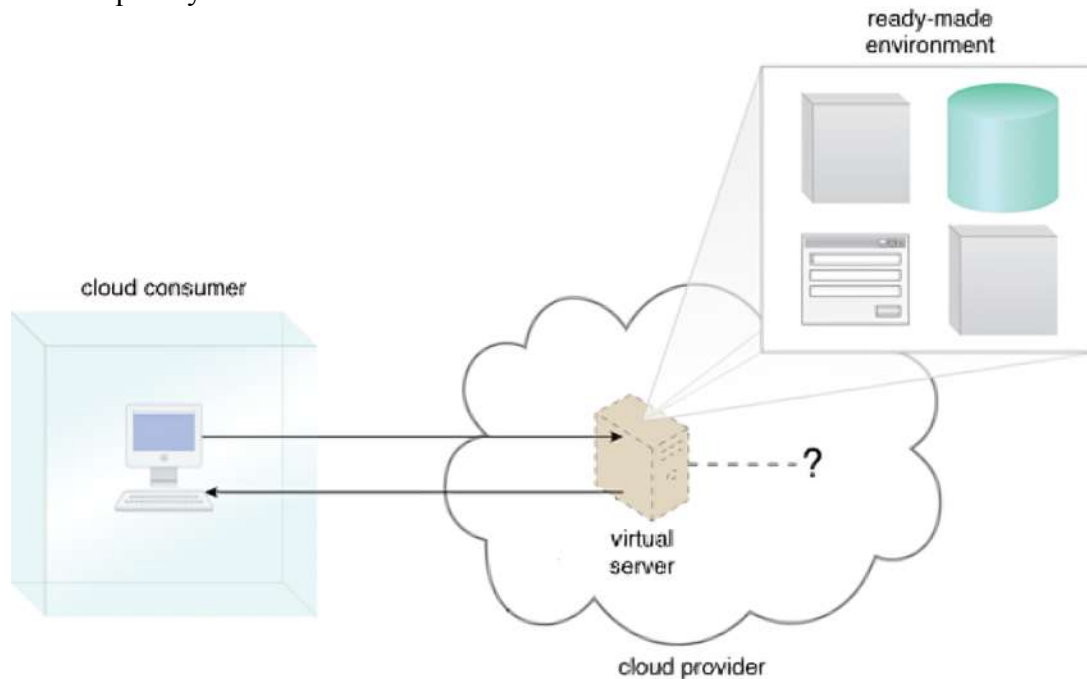
- Partitioning of Resources – The hypervisor allocates CPU, memory, and storage dynamically among VMs, ensuring optimal utilization of physical resources.
- Isolation – Each VM operates independently with its own OS, preventing conflicts and ensuring security.
- Scalability – Cloud providers can create and manage multiple VMs on demand, scaling resources as needed.
- Fault Tolerance – In case of hardware failure, VMs can be migrated to another physical server with minimal downtime.
- Multi-Tenancy – Different users can run multiple workloads on the same physical infrastructure, reducing operational costs.

The hypervisor plays a crucial role in cloud computing by dividing a physical server into multiple virtual machines, enabling efficient resource replication, scalability, and service deployment across the cloud.

### **22. What is a readymade environment or plug and play services? Explain with a neat diagram for a case study.**

- The ready-made environment mechanism is a defining component of the PaaS cloud delivery model that represents a pre-defined, cloud-based platform comprised of a set of already installed IT resources, ready to be used and customized by a cloud consumer.
- These environments are utilized by cloud consumers to remotely develop and deploy their own services and applications within a cloud. Typical ready-made environments include pre-installed IT resources, such as databases, middleware, development tools, and governance tools. A ready-made environment is generally equipped with a complete software development kit (SDK) that provides cloud consumers with programmatic access to the development technologies that comprise their preferred programming stacks.
- These environments eliminate the need for manual installation and configuration, allowing developers and businesses to focus on building and deploying applications

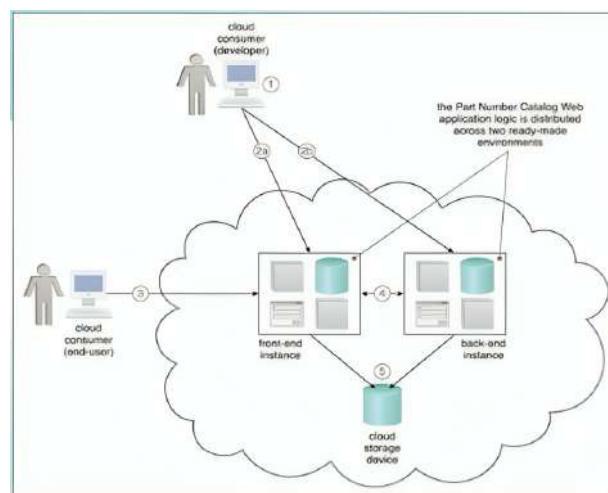
quickly.



### Case study:

The developer uses the provided SDK to develop the Part Number Catalog Web application

- The application software is deployed on a Web platform that was established by two ready-made environments called the front-end instance and the back-end instance
- The application is made available for usage and one end-user accesses its front-end instance
- The software running in the front-end instance invokes a long-running task at the back-end instance that corresponds to the processing required by the end-user
- The application software deployed at both the front-end and back-end instances is backed by a cloud storage device that provides persistent storage of the application data



\*\*\*\*\*