1.Differentiate between traditional IT infrastructure, virtualization, and cloud computing.

The basic difference between the IT infrastructure , virtualization and cloud computing is represented below with the categories:

Aspect	Traditional IT Infrastructure	Virtualization	Cloud Computing
Definition	Physical hardware and servers are set up and maintained on-premises.	Software-based method of creating virtual versions of physical resources like servers, storage, and networks.	On-demand access to computing resources (like servers, storage, and applications) over the internet.
Hardware Management	Requires physical management and maintenance of all hardware.	Physical hardware is abstracted, but resources are managed virtually.	Hardware is managed by the cloud provider; users access it remotely.
Cost	High upfront costs for hardware, software, and maintenance.	Lower hardware costs, as virtualization allows for better resource utilization.	Pay-as-you-go model with no upfront hardware costs. Users pay for what they use.
Scalability	Limited scalability; adding more resources requires purchasing and installing new hardware.	Scalable within the limits of the physical hardware; can add virtual machines.	Highly scalable; resources can be increased or decreased on-demand.
Flexibility	Less flexible; upgrades and changes are more complex and time-consuming.	More flexible; resources can be allocated and reallocated easily.	Extremely flexible; resources and services can be adjusted quickly as needed.

Maintenance	Requires ongoing maintenance, updates, and troubleshooting of physical hardware.	Easier to maintain since hardware failure doesn't directly impact virtual resources.	Maintenance is handled by the cloud provider, with minimal user involvement.
Deployment Time	Slow deployment; requires installation and configuration of hardware and software.	Faster deployment compared to traditional infrastructure; virtual resources can be spun up quickly.	Very fast deployment; users can instantly access cloud resources.
Examples	On-premises data centers with physical servers and storage.	VMware, Hyper-V, VirtualBox (virtualizing servers, storage, etc.).	Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP).

2. A food delivery app gets 10 orders → 1,000 → 100,000.
Which is a better option?

a.Horizontal scaling (add more servers).
b.Vertical scaling (upgrade one server).
c.Explain advantages/disadvantages of each.

Which scaling strategy is better for long term IoT data streams from millions of devices?

1. Horizontal Scaling (Add More Servers)

Horizontal scaling involves adding more servers or machines to distribute the load and handle increased traffic.

- Advantages:
    - **Scalability:** More servers can be added as needed, making it highly scalable.

    - **Redundancy:** If one server fails, others can take over, ensuring high availability and reliability.

- **Cost-effective:** Over time, it may be cheaper to add multiple smaller servers than to keep upgrading one powerful server.
  - **Flexibility:** Easily adjusts to changing workloads and traffic spikes.

- Disadvantages:
  - Complexity: Managing multiple servers requires complex load balancing, networking, and synchronization.
  - Maintenance Overhead: More servers mean more resources needed to monitor, update, and maintain them.
  - Data Consistency: Distributing data across multiple servers can lead to challenges in ensuring consistent data across all servers.

2. Vertical Scaling (Upgrade One Server)

Vertical scaling involves upgrading the existing server by adding more resources (CPU, RAM, storage) to handle increased load.

- Advantages:
  - Simplicity: Easier to manage since only one server is involved.
  - Less Complex: No need for load balancing or handling multiple machines, reducing operational complexity.
  - Performance Boost: Upgrading a server's resources can provide an immediate performance increase for applications with high CPU or memory needs.

- Disadvantages:
  - Limited by Hardware: There's a physical limit to how much you can upgrade a single server, and eventually, it won't be enough.
  - Single Point of Failure: If the server goes down, the entire system might go down, impacting availability.
  - Cost: High-end hardware upgrades can be very expensive and might not provide a proportional increase in performance for large-scale growth.

Which Scaling Strategy is Better for Long-Term IoT Data Streams from Millions of Devices?

For long-term IoT data streams (which can involve millions of devices sending continuous data), horizontal scaling is typically the better option for the following reasons:

- **Scalability:** IoT systems generate massive amounts of data, and horizontal scaling allows you to add more servers to handle the growing volume of data as needed. This is much more efficient than trying to upgrade a single server.

- **Fault Tolerance:** Horizontal scaling increases redundancy. If one server fails, the data and load can be distributed to other servers, minimizing the risk of system downtime. This is crucial in IoT systems where continuous data availability is essential.

- **Flexibility:** As IoT devices scale up, more resources can be dynamically allocated, and servers can be added or removed without disrupting the system. This makes horizontal scaling a flexible solution for handling unpredictable spikes in data.

- **Cost-Efficiency:** As the number of IoT devices grows, using multiple smaller, less expensive servers often becomes more cost-effective than continually upgrading one expensive server.

For a food delivery app scaling from 10 orders to 100,000 orders, horizontal scaling is likely the better choice as it can handle growth more efficiently and provide better fault tolerance. For long-term IoT data streams from millions of devices, horizontal scaling is even more advantageous due to its scalability, flexibility, and fault tolerance.

3. Why can't IoT devices just store everything locally? Why do they need the cloud?

IoT devices cannot store everything locally because they typically have limited storage, processing power, and computational resources. Storing large amounts of data locally would quickly exhaust their capacity and hinder their ability to process complex data. The cloud provides virtually unlimited storage, powerful computational capabilities for data analysis, real-time access from anywhere, and easier synchronization across multiple devices. Additionally, the cloud offers scalability, backup, security, and integration with other systems, which are essential as IoT networks grow in size and complexity. Thus, cloud storage and processing are necessary for managing the vast amounts of data generated by IoT devices efficiently.

4.What is the relationship between speed and power consumptions in electronics/embedded devices? Define why we should limit current in the circuit?

In electronics and embedded devices, **speed** (how fast a processor or circuit operates) and **power consumption** are tightly linked:

- **Dynamic Power Consumption**
  - Formula:
  - $P_{dynamic} = C * V^2 * f$
    - where C = capacitance, V = supply voltage, f = clock frequency.
    - As **speed (frequency)** increases, the device switches more often → more charging/discharging of capacitors → higher power draw.
  - **Voltage Scaling**
    - To achieve higher speeds, circuits often require higher voltage.
    - Since power is proportional to V^2, even small increases in voltage cause large jumps in power consumption.
  - **Heat & Efficiency Trade-off**
  - Faster operation = more heat generated.
  - Designers balance speed vs. energy efficiency (e.g., dynamic voltage and frequency scaling in CPUs).

Limiting current is critical for both **safety** and **device reliability**:

- **Prevent Component Damage**
  - Excess current overheats resistors, transistors, ICs, and wires → permanent damage or fire risk.
- **Protect Power Supply**
  - Overcurrent can overload batteries or regulators, reducing lifespan or causing failure.
- **Ensure Stable Operation**
  - High current spikes cause voltage drops (Ohm's law: $V=I\cdot R$) → unstable logic levels in digital circuits.
- **Safety for Users**
  - Limiting current prevents hazards like electric shocks or burns in exposed circuits.

- **Circuit Design Practice**
  - Use resistors, fuses, current limiters, or constant-current drivers (e.g., for LEDs) to keep current within safe bounds.

5.Design a small project idea that combines both cloud computing and IoT. Outline what problem it solves, what devices or sensors you would use, and how the cloud would store or analyze the data. (Students are requested to pitch their idea in front of the class).

[Answer by Yourself]