FC Answer Bank

2-Mark Questions (Q1–Q10)

Q1. What is a hypervisor, and its role in virtualization?

- A hypervisor is special software that allows a single physical machine to run multiple virtual machines (VMs).

- It divides and allocates resources like CPU, RAM, storage, and network among different VMs.

- It ensures strong isolation, so problems in one VM don't affect others.

- It enables better resource use, saving cost and increasing efficiency.

Q2. State two challenges of edge computing.

- Security: Edge devices are often in remote or open areas, making them prone to tampering, theft, or hacking.

- Management: Thousands of devices across wide areas are hard to monitor and update.

- Troubleshooting failures in the field can take extra time and money.

- Keeping all devices patched and updated is complex.

Q3. Define Edge Computing.

- Edge computing means processing data near where it is generated instead of sending it to a distant cloud.

- It reduces latency and allows real-time decision-making.

- Only summarized or critical results are sent to the cloud, saving bandwidth.

- Examples include self-driving cars or smart factories.

Q4. What is virtualization in cloud computing?

- Virtualization lets one physical server run multiple virtual machines using a hypervisor.

- Each VM behaves like a separate computer with its own OS and apps.

- It improves hardware utilization and reduces cost.

- It provides strong isolation between workloads, increasing reliability.

Q5. Give two examples of an edge computing use case in smart cities.

- Smart traffic management: Edge devices process live traffic data and adjust signals in real-time.

- Public safety: AI-enabled CCTV cameras detect suspicious activities and alert authorities instantly.

- Both reduce cloud dependency and improve response speed.

- They also save bandwidth by only sending alerts to cloud.

Q6. Which edge computing platform is associated with Google Cloud?

- Google Cloud IoT Edge is the edge platform by Google.

- It processes data locally on devices before sending it to the cloud.

- It supports running ML models like TensorFlow Lite at the edge.

- It ensures very low latency actions, even offline.

Q7. What is the main purpose of processing data at the edge?

- To enable immediate local decision-making instead of waiting for the cloud.

- To reduce latency since data does not travel far.

- To save bandwidth by only sending summaries to the cloud.

- To improve privacy as raw data stays local.

Q8. Name two critical elements for an Edge architecture.

- Sufficient compute power: CPUs/GPUs to run AI and analytics locally.

- Strong networking: Reliable Wi-Fi, 5G, or Ethernet for communication.

- Without compute, tasks get delayed; without strong networking, cloud sync fails.

- Both are essential for reliable edge computing.

Q9. What is resource efficiency in the context of edge virtualization?

- Virtualization allows multiple apps to share limited hardware smoothly.

- It ensures CPU, RAM, and storage are utilized effectively.

- This reduces delays and improves processing speed at the edge.

- It also lowers cost and energy consumption.

Q10. What is the primary benefit of containerization for edge devices?

- Containers are lightweight and efficient compared to virtual machines.

- They start very quickly, making them suitable for real-time edge tasks.

- They are portable and can run across different hardware and OS easily.

- They allow multiple isolated apps to run on the same edge device.

5-Mark Questions (Q11–Q18)

Q11. Discuss the major advantages and disadvantages of edge computing.

- Advantages:

- 1. Reduces latency by processing data near its source.

- 2. Saves bandwidth since only filtered data goes to the cloud.

- 3. Improves reliability during weak or no internet connectivity.

- 4. Protects sensitive data locally, improving privacy.

- 5. Enables real-time applications like autonomous vehicles and healthcare.

- Disadvantages:

- 6. Limited CPU, RAM, and storage at edge devices.

- 7. High deployment and maintenance costs for large-scale edge networks.

- 8. Security challenges as devices are exposed to tampering or hacking.

- 9. Difficult to manage and update thousands of edge devices across regions.

Q12. Evolution of Edge Computing and its relation with IoT.

- Evolution of Edge Computing:

- 1. Early stage: On-premise servers were costly and hard to scale.

- 2. Cloud computing reduced cost but added latency and bandwidth usage.

- 3. Real-time needs led to edge computing for local processing.

- 4. Now widely used in self-driving cars, healthcare, and industrial IoT.

- Relation with IoT:

- 5. IoT devices generate massive amounts of continuous data.

- 6. Sending all data to the cloud is slow and costly.

- 7. Edge processes and filters data locally, sending only summaries to cloud.

- 8. IoT and Edge together enable smart cities, factories, and healthcare.

Q13. Compare and contrast Edge Computing with traditional Cloud Computing.

- 1. Processing: Edge → near data source; Cloud → centralized data centers.

- 2. Latency: Edge has very low latency; Cloud has higher latency.

- 3. Bandwidth: Edge saves bandwidth; Cloud consumes more.

- 4. Reliability: Edge works offline; Cloud needs stable internet.

- 5. Privacy: Edge improves local data privacy; Cloud exposes more data.

- 6. Resources: Edge has limited compute/storage; Cloud has massive resources.

- 7. Best Use: Edge for real-time apps; Cloud for AI training & big data.

- 8. Scalability: Edge scales by adding devices; Cloud scales within data centers.

- 9. Approach: Both complement each other for complete solutions.

Q14. What are the challenges and opportunities in deploying edge computing solutions?

- Challenges:

- 1. Limited computing resources on edge devices.

- 2. Security risks as devices are often exposed.

- 3. Difficult to monitor and update thousands of nodes.

- 4. High deployment and operational costs.

- Opportunities:

- 5. Real-time systems like robotics and autonomous vehicles.

- 6. Reduced bandwidth cost by local processing.

- 7. Improved privacy as sensitive data stays local.

- 8. Supports IoT expansion with millions of connected devices.

- 9. Enables reliable smart cities and industrial automation.

Q15. Compare and contrast AWS IoT Greengrass and Azure IoT Edge.

- 1. Cloud integration: Greengrass with AWS Lambda & SageMaker; IoT Edge with Azure Functions & Cognitive Services.

- 2. Application model: Greengrass supports Lambda + containers; IoT Edge uses containerized modules.

- 3. ML support: Greengrass deploys SageMaker ML models; IoT Edge supports Azure ML & AI services.

- 4. Offline support: Both run locally and sync with cloud when online.

- 5. Deployment: Both scalable for manufacturing, smart cities, and vehicles.

- 6. Security: Both platforms ensure secure device-cloud communication.

- 7. Portability: IoT Edge modules are highly portable across devices.

- 8. Greengrass emphasizes integration with AWS IoT Core.

- 9. Azure IoT Edge emphasizes enterprise cloud + AI ecosystem.

Q16. Describe a typical Edge Computing Architecture.

- 1. IoT devices/sensors continuously generate data.

- 2. Data flows to local edge hardware or gateways for processing.

- 3. Fog nodes (intermediate layer) handle distributed analytics.

- 4. Cloud receives only summaries for storage, AI training, and reporting.

- 5. Ensures low latency for local tasks and scalability for large data.

- 6. Provides security by processing sensitive data locally.

- 7. Components include devices, edge servers, fog layer, and cloud.

- 8. Interaction ensures smooth, fast, and reliable system functioning.

- 9. Examples: smart cities, industrial IoT, and healthcare monitoring.

Q17. Critical elements for edge architecture (SCANC model).

- 1. Storage: Temporary storage for local caching (e.g., CCTV footage buffer).

- 2. Compute: Local CPUs/GPUs for analytics (e.g., industrial sensors).

- 3. Acceleration: AI/ML accelerators (GPU/TPU) for high-speed inference.

- 4. Networking: Reliable 5G/Wi-Fi/Ethernet for communication.

- 5. Control: Local actions like emergency shutdowns without cloud delays.

- 6. Examples: Raspberry Pi, Jetson Nano, Intel NUC with all SCANC features.

- 7. Ensures reduced latency and reliability.

- 8. Supports scalability by adding more nodes.

- 9. Critical for real-time decision-making environments.

Q18. Virtualization and containerization in edge computing with advantages.

- Virtualization:

- 1. Runs multiple VMs on one machine using a hypervisor.

- 2. Provides isolation, supports legacy apps, improves hardware use.

- 3. Each VM runs its own OS independently.

- Containerization:

- 4. Lightweight packaging of apps + dependencies into containers.

- 5. Containers share OS but remain isolated, portable, and fast.

- Advantages:

- 6. Virtualization → strong isolation, multi-OS support, efficient hardware use.

- 7. Containerization → quick start, portability, easier updates.

- 8. Containers use fewer resources, ideal for edge devices.

- 9. Both together improve flexibility, scalability, and efficiency at the edge.