Microservices and Apache Kafka Interview Questions & Answers

Microservices Interview Questions & Answers

Q: What are microservices?
A: Microservices is an architectural style where an application is divided into small, loosely coupled, independently deployable services. Each service has its own database, business logic, and can communicate with other services (usually via REST, gRPC, or messaging systems like Kafka).

Q: What are the advantages of microservices architecture?
A: Independent deployment, scalability, technology diversity, fault isolation, and better team ownership.

Q: What are the challenges in microservices?
A: Complex communication, distributed transactions, monitoring/logging, deployment complexity, and security management.

Q: How do microservices communicate with each other?
A: Synchronous: REST API, gRPC. Asynchronous: Kafka, RabbitMQ, JMS.

Q: How do you handle data consistency in microservices?
A: Event-driven architecture, Saga pattern, two-phase commit.

Q: What is API Gateway in microservices?
A: API Gateway acts as a single entry point for all microservices. Handles routing, authentication, rate limiting, etc.

Q: How do you secure microservices?
A: OAuth2, JWT tokens, TLS/SSL.

Q: What is Circuit Breaker in microservices?
A: Prevents cascading failures. If a service fails repeatedly, the circuit opens and requests fail immediately. Tools: Hystrix, Resilience4j.

Q: How do you deploy microservices?
A: Using Docker containers, Kubernetes orchestration, Service Mesh (Istio, Linkerd).

Q: What design patterns are commonly used in microservices?
A: API Gateway, Saga, Strangler, Circuit Breaker, Event Sourcing & CQRS.

Apache Kafka Interview Questions & Answers

Q: What is Apache Kafka?
A: Kafka is a distributed event streaming platform used for high-throughput, fault-tolerant, real-time data pipelines and messaging.

Q: What are Kafka components?
A: Producer, Consumer, Broker, Topic, Partition, ZooKeeper (legacy).

Q: What are Kafka partitions and why are they used?
A: Partitions allow parallelism and scalability. Messages in a partition are ordered.

Q: How does Kafka ensure fault tolerance?
A: Replication, leader-follower model, automatic failover.

Q: Difference between Kafka and traditional brokers?

A: Kafka is designed for high throughput & streaming. RabbitMQ is message-queue focused. Kafka stores/replays messages.

Q: What is Kafka Consumer Group?
A: A group of consumers with same group ID. Each partition is consumed by only one consumer in a group.

Q: What is offset in Kafka?
A: Unique ID of a record within a partition. Tracks consumer progress.

Q: What are Kafka delivery semantics?
A: At most once, At least once, Exactly once.

Q: How do you achieve exactly-once processing in Kafka?
A: Enable idempotent producer, use transactions, commit offsets after success.

Q: How is Kafka used in Microservices?
A: For event-driven communication, decoupling services, eventual consistency, and implementing Saga patterns.