# Kafka Interview Questions

## Easy Category:

Q.1: What is Apache Kafka?
Ans: Apache Kafka is a distributed streaming platform that allows for the publishing and subscribing of streams of records.

Q.2: What are the key components of Kafka?
Ans: Kafka consists of topics, producers, consumers, brokers, and ZooKeeper (used for coordination and synchronisation).

Q.3: What is a Kafka topic?
Ans: A Kafka topic is a category or feed name to which records are published. It is divided into partitions for scalability and parallelism.

Q.4: Explain the role of Kafka producers.
Ans: Kafka producers are responsible for publishing messages to Kafka topics. They write records to Kafka brokers.

Q.5: What are Kafka consumers?
Ans: Kafka consumers subscribe to topics and consume records published to them. They read records from Kafka brokers.

Q.6:What is a Kafka broker?
Ans: A Kafka broker is a server that manages the storage and replication of Kafka topics. It is responsible for handling read and write requests.

Q.7:How does Kafka ensure fault tolerance?
Ans: Kafka achieves fault tolerance through replication. Each partition of a topic is replicated across multiple brokers.

Q.8:What is the role of ZooKeeper in Kafka?
Ans: ZooKeeper is used for managing the coordination and synchronization of Kafka brokers. It maintains the metadata and tracks the status of brokers.

Q.9:Explain the concept of message offset in Kafka.
Ans: Message offset represents the unique identifier of a message within a partition. It helps in tracking the position of a consumer in a partition.

Q.10:How does Kafka handle data retention?
Ans: Kafka has a configurable retention period for topics. It can retain messages for a certain duration or based on a certain message size.

Q.11:How can you monitor Kafka cluster health?
Ans: Kafka provides various metrics that can be monitored using tools like Kafka Manager, Confluent Control Center, or custom monitoring solutions.

Q.12:What is a Kafka consumer offset?
Ans: A consumer offset represents the position of a consumer within a partition. It tracks the last consumed message offset.

Q.13:Explain the role of the Kafka log compaction process.
Ans: The log compaction process in Kafka removes older, duplicate messages from a partition while retaining the latest message for each unique key.

Q.14:How can you control the number of consumers in a consumer group?
Ans: The number of consumers in a consumer group can be controlled by adding or removing consumer instances from the group.

Q.15:What is the purpose of Kafka Connect transformations?
Ans: Kafka Connect transformations allow for data manipulation during the process of moving data between Kafka and external systems.

Q.16:How does Kafka handle data compression?
Ans: Kafka supports various compression codecs, such as GZIP, Snappy, and LZ4, which can be configured to compress the messages before storage.

Q.17:What is a Kafka topic partition?
Ans: A Kafka topic partition is a logical division of a topic that allows for parallelism and scalability in handling message processing.

Q.18:Explain the concept of Kafka message durability.
Ans: Kafka ensures message durability by writing messages to disk before acknowledging the producers. This prevents data loss in case of failures.

Q.19:How can you ensure message delivery order across multiple Kafka partitions?
Ans: Kafka doesn't guarantee message delivery order across partitions. However, you can use a single partition or key-based partitioning to maintain order within a partition.

Q.20:What is the purpose of Kafka consumer offset commit intervals?
Ans: The offset commit interval determines how frequently the consumer commits its current offset. A shorter interval provides lower latency but higher commit overhead.

Q.21:What is the role of Kafka brokers in a Kafka cluster?
Ans: Kafka brokers are responsible for storing and serving the published messages. They handle read and write requests from producers and consumers.

Q.22:How does Kafka ensure high throughput and low latency?
Ans: Kafka achieves high throughput and low latency by utilizing a distributed architecture, parallel processing, and efficient disk I/O operations.

Q.23:Explain the concept of Kafka consumer offsets commit strategies.
Ans: Kafka consumer offsets can be committed using manual commit or automatic commit strategies. Manual commit gives more control, while automatic commit is simpler.

Q.24:How can you scale Kafka consumers?
Ans: Kafka consumers can be scaled horizontally by adding more consumer instances to a consumer group, allowing for parallel processing and increased throughput.

Q.25:What is the purpose of Kafka message serialisation?
Ans: Kafka requires messages to be serialised before being sent. Serialisation converts messages from their object representation to a format that can be transmitted.

Q.26:How does Kafka handle data retention policy enforcement?
Ans: Kafka enforces the data retention policy by periodically checking the timestamp or size of messages in a topic and removing them if they exceed the configured limits.

Q.27:Explain the role of Kafka ZooKeeper in consumer coordination.
Ans: Kafka ZooKeeper is used for consumer coordination, such as tracking consumer group membership, partition ownership, and leader election.

Q.28:How can you handle duplicate messages in Kafka consumers?
Ans: Kafka consumers can handle duplicate messages by deduplicating based on unique identifiers or using idempotent processing logic to handle duplicate records.

Q.29:What is the purpose of Kafka record headers?
Ans: Kafka record headers allow for attaching additional metadata to Kafka messages, enabling flexible processing and enrichment of messages.

Q.30:How can you monitor Kafka consumer lag?
Ans: Kafka consumer lag can be monitored by comparing the current consumer offset with the latest available offset for each partition using Kafka consumer lag monitoring tools.

## Medium Category:

Q.1:What is the role of a Kafka producer callback?
Ans: A producer callback is invoked after a record is sent to Kafka. It allows for handling success or failure notifications.

Q.2:How does Kafka handle backpressure?
Ans: Kafka employs backpressure at the consumer end by allowing consumers to control their own consumption rates.

Q.3:What is the significance of the Kafka Connect API?
Ans: Kafka Connect API is used for building and running connectors that move data in and out of Kafka. It provides a scalable and fault-tolerant solution for data integration.

Q.4:Explain the concept of a Kafka consumer group.
Ans: A consumer group is a group of consumers that work together to consume records from Kafka topics. Each consumer in a group processes a subset of partitions.

Q.5:How does Kafka handle data replication?
Ans: Kafka uses a leader-follower replication model. Each partition has one leader and multiple followers, ensuring data redundancy and fault tolerance.

Q.6:What is the purpose of the Kafka Streams library?
Ans: Kafka Streams is a client library for building stream processing applications that can process and analyze real-time data from Kafka topics.

Q.7:How can you ensure message ordering within a Kafka partition?
Ans: Kafka guarantees message ordering within a partition. Multiple messages from the same producer will be written to the same partition and processed in order.

Q.8:What is a Kafka offset commit?
Ans: Offset commit is the process of acknowledging the successful consumption of messages by a Kafka consumer. It helps in tracking the progress of consumers.

Q.9:Explain the concept of message retention policies in Kafka.
Ans: Kafka provides two retention policies: log compaction and delete. Log compaction retains the latest message for each key, while delete retains messages based on time or size.

Q.10:How does Kafka handle failover scenarios?
Ans: In case of broker failure, Kafka ensures failover by electing a new leader for affected partitions. Consumers automatically recover and continue from the last committed offset.

Q.11:How can you handle Kafka message retries in case of failures?
Ans: You can configure the Kafka producer to retry failed messages using the "retries" configuration parameter. Additionally, you can set a callback to handle failures.

Q.12:Explain the concept of Kafka log compaction cleanup policy.
Ans: The log compaction cleanup policy in Kafka determines when to remove obsolete messages during log compaction. It can be based on time or log size.

Q.13:How does Kafka handle consumer rebalancing?
Ans: Kafka rebalances consumers in a consumer group by redistributing partitions among the consumers when new consumers join or existing consumers leave the group.

Q.14:What is the role of Kafka Streams windowing operations?
Ans: Kafka Streams windowing operations allow for aggregating data over fixed time intervals or key-specific session windows, enabling time-based analysis.

Q.15:How can you handle schema evolution in Kafka using Apache Avro?
Ans: Apache Avro, a schema-based serialisation framework, enables schema evolution by supporting backward and forward compatibility when reading and writing data.

Q.16:Explain the purpose of Kafka Connect converters.
Ans: Kafka Connect converters are responsible for serialising and deserializing data between Kafka and external systems. They handle data format conversions.

Q.17:How does Kafka ensure data consistency and durability?
Ans: Kafka ensures data consistency and durability through replication, which replicates each partition across multiple brokers, ensuring fault tolerance.

Q.18:What is the role of Kafka Streams Interactive Queries Server?
Ans: The Kafka Streams Interactive Queries Server provides a REST API for querying the state stores maintained by Kafka Streams applications.

Q.19:Explain the purpose of Kafka Streams stateful operations.
Ans: Kafka Streams stateful operations allow for maintaining and updating state while processing data streams, enabling complex processing tasks.

Q.20:How can you handle late-arriving events in Kafka Streams?
Ans: Kafka Streams provides windowing operations that allow you to define a grace period for late-arriving events, ensuring accurate processing.

Q.21:What is the purpose of Kafka Streams state store changelog topics?
Ans: Kafka Streams state stores use changelog topics to persist the local state of a stream processing application, ensuring fault tolerance and state recovery.

Q.22:How does Kafka handle leader elections in a broker cluster?
Ans: Kafka uses ZooKeeper for leader election in a broker cluster. When a leader fails, ZooKeeper helps elect a new leader for the affected partition.

Q.23:Explain the concept of Kafka message timestamping.

Ans: Kafka allows messages to be time stamped with a user-defined timestamp or the producer's system timestamp. It can be used for message ordering and time-based processing.

Q.24:How can you ensure message delivery guarantees in Kafka producers?
Ans: Kafka producers can ensure message delivery guarantees by configuring the "acks" parameter appropriately and handling producer errors and retries.

Q.25:What is the role of Kafka Connect converters in serialization and deserialization?
Ans: Kafka Connect converters handle the serialization and deserialization of data between Kafka and external systems, ensuring compatibility and data integrity.

Q.26:How does Kafka handle data partitioning across brokers?
Ans: Kafka uses consistent hashing to assign partitions to brokers, ensuring even distribution and fault tolerance by replicating partitions across multiple brokers.

Q.27:Explain the purpose of Kafka Streams processor API.
Ans: Kafka Streams processor API provides a low-level, fine-grained interface for building custom stream processors, enabling advanced stream processing operations.

Q.28:How can you handle dead-lettering in Kafka consumers?
Ans: Dead-lettering in Kafka consumers involves redirecting failed or unprocessable messages to a separate topic or storage for further analysis or manual intervention.

Q.29:What is the role of Kafka Connect in sink and source connectors?
Ans: Kafka Connect acts as a framework for building sink and source connectors that facilitate the movement of data between Kafka and external systems.

Q.30:How does Kafka handle message offsets for Kafka Streams?
Ans: Kafka Streams tracks message offsets internally, ensuring the processing of each message exactly once by maintaining the offset state for fault tolerance.

## Hard Category:

Q.1:How does Kafka handle data replication across multiple data centers?
Ans: Kafka allows for data replication across multiple data centers through the use of MirrorMaker or the built-in MirrorMaker 2.0.

Q.2:Explain the role of the Kafka Controller.
Ans: The Kafka Controller is responsible for managing and coordinating the actions of brokers in the Kafka cluster. It handles tasks such as leader election and partition reassignment.

Q.3:What is the purpose of the Kafka Streams DSL?
Ans: Kafka Streams DSL is a high-level abstraction for building stream processing applications. It provides an easy-to-use interface for defining data transformations and aggregations.

Q.4:How does Kafka handle semantics exactly once?
Ans: Kafka provides exactly once semantics through the use of idempotent producers and transactional producers. Consumers can also use the commitSync() method for exactly once processing.

Q.5:What is the purpose of Kafka Connect converters?
Ans: Kafka Connect converters are used to serialize and deserialize data between Kafka and external systems. They ensure compatibility and seamless integration.

Q.6:Explain the concept of Kafka log compaction.
Ans: Log compaction in Kafka retains only the latest message for each unique key in a topic. It is useful for scenarios where the latest state of a record is important.

Q.7:How does Kafka handle data rebalancing in consumer groups?
Ans: When a consumer joins or leaves a consumer group, Kafka triggers a rebalance process. It redistributes the partitions among the remaining consumers in a group.

Q.8:What is the role of Kafka ACLs (Access Control Lists)?
Ans: Kafka ACLs are used to control access to topics and perform authorization checks. They provide granular control over read, write, and admin operations.

Q.9:Explain the purpose of Kafka Streams Interactive Queries.
Ans: Kafka Streams Interactive Queries allow applications to query state stores created by stream processing tasks. It provides real-time access to the current state of the data.

Q.10:What is the role of the Kafka Connect framework?
Ans: Kafka Connect is a framework for building and running scalable and fault-tolerant data pipelines between Kafka and other systems. It provides a plugin-based architecture for easy extensibility.

Q.11:How does Kafka handle data compaction in the presence of tombstone messages?
Ans: Kafka uses tombstone messages (special delete markers) during log compaction to mark records for deletion. Tombstones take precedence over non-tombstone messages.

Q.12:Explain the concept of Kafka message format evolution using Schema Registry.
Ans: The Schema Registry in Kafka allows for message format evolution by supporting schema compatibility checks and providing a centralised schema repository.

Q.13:How can you handle out-of-order data in Kafka Streams processing?
Ans: Kafka Streams provides a time windowing feature that allows for handling out-of-order data by defining a window size that accommodates late-arriving events.

Q.14:What is the purpose of Kafka Streams state stores?
Ans: Kafka Streams state stores enable the storage and querying of application-specific state during stream processing, facilitating stateful transformations.

Q.15:How can you configure Kafka to ensure at least once message delivery semantics?
Ans: Kafka provides at least once message delivery semantics by enabling the "acks" configuration parameter and implementing proper error handling and retries.

Q.16:Explain the role of Kafka Connect converters in Avro serialisation.
Ans: Kafka Connect converters handle Avro serialisation/deserialization by interacting with the Avro schema registry to ensure compatibility and schema evolution.

Q.17:How does Kafka handle partition rebalancing during a consumer group rebalance?
Ans: Kafka follows a group coordination protocol, where the group coordinator orchestrates the partition assignment and rebalancing process during consumer group rebalances.

Q.18:What is the purpose of Kafka Streams state store suppliers?
Ans: State store suppliers in Kafka Streams define how the state stores are created and restored, allowing for customization and integration with external systems.

Q.19:Explain the concept of Kafka transactional messaging.
Ans: Kafka transactional messaging ensures atomicity and consistency by allowing producers and consumers to participate in transactions across multiple partitions.

Q.20:How can you handle stateful aggregations in Kafka Streams?
Ans: Kafka Streams provides built-in stateful aggregation operations, such as count, sum, and average, which can be used to maintain and update aggregations over time.

Q.21:What is the purpose of Kafka's log cleaner and log compaction?
Ans: Kafka's log cleaner is responsible for removing obsolete log segments, while log compaction retains the latest message for each unique key within a topic.

Q.22:Explain the concept of Kafka message compression.
Ans: Kafka supports message compression to reduce network bandwidth and storage costs. It offers configurable compression codecs like GZIP, Snappy, and LZ4.

Q.23:How can you handle transactional messaging in Kafka consumers?
Ans: Kafka consumers can participate in transactions by using the Kafka transactional API, which allows for atomic and consistent processing across multiple partitions.

Q.24:What is the purpose of Kafka Streams interactive queries?
Ans: Kafka Streams interactive queries enable applications to query the current state of the data stored in the state stores maintained by Kafka Streams.

Q.25:How does Kafka handle partition leadership rebalancing?
Ans: Kafka handles partition leadership rebalancing by electing a new leader when the current leader fails or when new brokers join or leave the cluster.

Q.26:Explain the concept of Kafka consumer group coordination protocol.
Ans: Kafka consumer group coordination protocol uses ZooKeeper or the newer group coordination protocol based on the Kafka brokers to manage consumer group membership and rebalancing.

Q.27:What is the role of Kafka Streams KTables?
Ans: Kafka Streams KTables represent the state of a stream at a specific point in time. They provide a materialized view of the data for efficient querying.

Q.28:How can you achieve near real-time processing with Kafka Streams?
Ans: Near real-time processing can be achieved with Kafka Streams by configuring appropriate time windows, reducing processing time, and optimizing the stream topology.

Q.29:Explain the purpose of Kafka Streams punctuators.
Ans: Kafka Streams punctuators allow for the insertion of user-defined processing logic at regular intervals, enabling periodic actions and time-driven transformations.

Q.30:What is the purpose of Kafka's high-level consumer API?
Ans: Kafka's high-level consumer API provides a simpler interface for building Kafka consumers by handling complex details such as partition assignment and offset management.

Q.31:What is the purpose of Kafka Connect converters in Avro serialization?
Ans: Kafka Connect converters are responsible for serializing and deserializing data between Kafka and external systems. In the case of Avro, converters interact with the Avro schema registry to ensure compatibility and schema evolution during serialization.

Q.32:How does Kafka handle message deduplication?
Ans: Kafka provides idempotent producers, which assign a unique identifier (message key) to each message. By using the message key, Kafka can identify and filter out duplicate messages during processing.

Q.33:Explain the concept of Kafka's Exactly Once Semantics.
Ans: Kafka's Exactly Once Semantics guarantees that messages are processed exactly once, without duplication or loss. This is achieved through the combination of idempotent producers, transactional producers, and transactional consumer offsets.

Q.34:What is the purpose of Kafka Connect Sinks and how are they used?
Ans: Kafka Connect Sinks are connectors that move data from Kafka to external systems. They enable the integration of Kafka with databases, file systems, and other data stores, allowing data to be consumed by other applications.

Q.35:How does Kafka Streams handle state restoration after a failure?
Ans: Kafka Streams leverages Kafka's built-in log compaction and changelog topics to persist and restore the state of stream processing applications. The application's state is continuously maintained and can be restored in case of failure.

Q.36:Explain the concept of Kafka Streams Global State Stores.
Ans: Kafka Streams Global State Stores are replicated and fault-tolerant storage mechanisms that allow all instances of a Kafka Streams application to access the same state. They enable global aggregations and lookups across all partitions.

Q.37:What is the purpose of Kafka's MirrorMaker tool?
Ans: Kafka's MirrorMaker tool is used for data replication between Kafka clusters. It allows data to be mirrored from one cluster to another, enabling data backup, disaster recovery, and data center migration scenarios.

Q.38:How does Kafka handle schema evolution in Avro serialization?
Ans: Kafka's Avro serialization, along with the Schema Registry, supports schema evolution by allowing compatible schema changes without breaking compatibility with existing data. This ensures backward and forward compatibility during data serialization.

Q.39:Explain the role of Kafka Streams in event-driven microservices architectures.
Ans: Kafka Streams provides a lightweight and scalable stream processing library that enables event-driven microservices architectures. It allows the processing and analysis of real-time data streams, facilitating event-driven communication and data processing.

Q.40:What are the considerations for scaling Kafka clusters and applications in a production environment?
Ans: Scaling Kafka clusters and applications involves considerations such as adding more brokers, increasing the number of partitions, optimizing hardware resources, and fine-tuning configuration parameters. Load balancing and monitoring tools are also essential for managing scalability effectively.