

Kafka Interview Questions

Download PDF

Print

Use the filter to quickly find topics like partitions, consumer groups, offsets, ISR, replication, retention, and more.

Filter questions (e.g., partitions, offsets, ISR, consumer

Showing 74 of 74

1. What is Apache Kafka?

Apache Kafka is a distributed, scalable, and high-throughput messaging system for handling real-time data streams.

2. What are the main components of Apache Kafka?

The main components of Apache Kafka are Topics, Producers, Consumers, and Brokers.

3. What is a Topic in Apache Kafka?

A Topic in Apache Kafka is a named feed for publishing and subscribing to streams of records.

4. What is a Producer in Apache Kafka?

A Producer in Apache Kafka is a client application that can publish a stream of records to one or more topics in a Kafka cluster.

5. What is a Consumer in Apache Kafka?

A Consumer in Apache Kafka is a client application that can subscribe to one or more topics and process the streams of records produced to those topics.

6. What is a Broker in Apache Kafka?

A Broker in Apache Kafka is a server instance that runs Kafka and is responsible for maintaining the publish-subscribe state and serving data to clients.

7. What is the use of Zookeeper in Apache Kafka?

Apache Zookeeper is used in Apache Kafka to manage and coordinate the cluster of Kafka brokers. It helps in maintaining the configuration information, provides group management, and ensures data consistency.

8. What is a partition in Apache Kafka?

A Partition in Apache Kafka is a horizontally divided unit of data within a topic that allows for parallel processing of data streams. Each partition can be processed by one or multiple consumers in a consumer group.

9. What is a Consumer Group in Apache Kafka?

A Consumer Group in Apache Kafka is a set of consumers that work together to consume a set of partitions from one or multiple topics. Each partition is consumed by only one consumer in a group.

10. What is the use of Offset in Apache Kafka?

The Offset in Apache Kafka is a unique identifier of a record within a partition and is used by consumers to track their progress within a partition. It helps in keeping track of which records have been consumed and which are yet to be consumed.

11. What is a Kafka Connector?

A Kafka Connector is a component that helps to move data in and out of Apache Kafka. It acts as a bridge between external systems and the Kafka cluster and can be used to import or export data to and from Kafka.

12. What is a Kafka Streams API?

The Kafka Streams API is a client library for building real-time and highly scalable data processing applications. It provides a high-level API for processing data streams from one or more topics and produces a stream of output records to one or more output topics.

13. What is a Kafka Serializer and Deserializer?

A Serializer in Apache Kafka is used to convert a record’s key or value to a byte array for sending to a topic. A Deserializer in Apache Kafka is used to convert a byte array received from a topic back to a key or value. Serializers and deserializers are used by producers and consumers to encode and decode records.

14. What are the different storage options available in Apache Kafka?

Apache Kafka provides two storage options: a write-ahead log and a compacted topic. A write-ahead log stores all records in their original form, while a compacted topic only stores the latest value for each key and discards old values.

15. What is a Kafka Cluster?

A Kafka Cluster is a group of one or more brokers that work together to manage the publish-subscribe state and serve data to clients. The cluster provides scalability, high availability, and data durability.

16. **What is the role of a Kafka Broker in a Cluster?**



In a Kafka Cluster, each broker plays a specific role. A broker is responsible for receiving records from producers, storing the records in one or more partitions, serving the records to consumers, and replicating data to other brokers for fault tolerance.

17. **What is the difference between a Kafka topic and a Kafka partition?**

A Topic in Apache Kafka is a named feed for publishing and subscribing to streams of records. Partitions are units of parallelism within a topic, providing horizontal scalability for the data streams. Each topic can be divided into multiple partitions, and each partition can be processed by one or multiple consumers in a consumer group.

18. **What is a Kafka Consumer Group and how does it work?**

A Consumer Group in Apache Kafka is a set of consumers that work together to consume a set of partitions from one or multiple topics. Each partition is consumed by only one consumer in a group, and the group collectively consumes the entire set of partitions. The consumer group allows for parallel processing and balancing the load among consumers.

19. **What is the role of Apache Zookeeper in Apache Kafka?**

Apache Zookeeper is used in Apache Kafka to manage and coordinate the cluster of brokers. It helps in maintaining configuration information, provides group management, and ensures data consistency. Zookeeper is used for managing and resolving broker failures, maintaining the metadata about the partitions and replicas, and electing a controller for the cluster.

20. **What are the main features of Apache Kafka?**

The main features of Apache Kafka are: scalability, high performance, fault tolerance, low latency, and high throughput. It supports real-time data streams, provides a publish-subscribe model, and allows for parallel processing of data streams. It also supports data compression and serialization, and provides a well-defined API for integrating with other systems.

21. **What is Apache Kafka Compression and why is it important?**

Apache Kafka Compression is the process of reducing the size of messages being sent over the network. This is important for two reasons: it reduces the amount of bandwidth required to transmit the data and it speeds up the processing of data. Apache Kafka supports several compression algorithms, including Gzip, Snappy, and LZ4, which can be configured on a per-topic basis.

22. **What is a Kafka Consumer offset and why is it important?**

A Consumer offset in Apache Kafka is the position of a consumer in a partition. It represents the location of the next record that the consumer will receive from the partition. The offset is important because it allows consumers to pick up where they left off in case of a failure, and it provides a mechanism for committing the position of a consumer in a partition to ensure that data is not re-processed.

23. **What is Apache Kafka Streams API?**

The Apache Kafka Streams API is a library for building streaming applications. It provides a simple and easy-to-use API for transforming and processing data streams in real-time. It allows developers to write applications that process data from one or more input topics, perform transformations on the data, and produce output to one or more output topics.

24. **What are the benefits of using Apache Kafka for real-time data processing?**

Apache Kafka provides several benefits for real-time data processing, including scalability, high performance, fault tolerance, low latency, and high throughput. It allows for parallel processing of data streams and provides a publish-subscribe model. It also supports data compression and serialization, and provides a well-defined API for integrating with other systems.

25. **What is the role of a Kafka Producer in a Kafka Cluster?**

A Kafka Producer is responsible for publishing data to one or more topics in a Kafka Cluster. The producer is responsible for serializing the data, dividing the data into partitions, and sending the data to the appropriate brokers. The producer may also choose to wait for acknowledgements from the brokers before considering the data as sent.

26. **What is the purpose of a Kafka topic partition?**

The purpose of a Kafka topic partition is to provide horizontal scalability for the data streams within a topic. By dividing the data into partitions, it allows for parallel processing of the data by multiple consumers in a consumer group. Each partition can be processed by one or multiple consumers, and each consumer in a group will only receive a subset of the data, providing load balancing for the consumers.

27. **What is Apache ZooKeeper and how does it relate to Apache Kafka?**

Apache ZooKeeper is a distributed coordination service used to manage large distributed systems. It provides services such as naming, configuration management, and group services. Apache Kafka uses ZooKeeper to manage the cluster configuration, including broker discovery, topic configuration, and other metadata.

28. **What is a Kafka Broker?**

A Kafka Broker is a node in a Kafka Cluster that is responsible for storing and serving the data for one or more partitions. The broker receives data from producers, stores the data, and serves the data to consumers. The broker also manages the partition assignments for

 the consumers, and coordinates the distribution of data between the partitions.

29. What is a Kafka Consumer Group?

A Consumer Group in Apache Kafka is a set of consumers that work together to consume data from one or more topics. Each consumer in the group receives a unique subset of the data, allowing for parallel processing of the data. The group also provides load balancing between the consumers, allowing for the distribution of the data processing workload.

30. What is a Kafka Partition Leader?

A Partition Leader in Apache Kafka is a broker that is responsible for serving data for a specific partition. The partition leader is responsible for responding to consumer requests, as well as coordinating the distribution of data to the followers. The partition leader is elected by the other brokers in the cluster, and can change dynamically if the current leader fails or becomes unavailable.

31. What is a Kafka Partition Replica?

A Partition Replica in Apache Kafka is a copy of a partition that is stored on a different broker in the cluster. The replicas provide fault tolerance for the data, as the data can be recovered from the replicas in the event of a failure of the partition leader. The replicas also help to distribute the data processing load, as the consumers can read from any of the replicas.

32. What is a Kafka topic?

A topic in Apache Kafka is a named channel to which data is published. The data is divided into partitions, and each partition can be consumed by one or multiple consumers. Topics provide a way to logically organize the data in the Kafka Cluster, allowing for easier management of the data streams.

33. What is a Kafka Producer?

A Kafka Producer is a client application that is responsible for publishing data to one or more topics in a Kafka Cluster. The producer is responsible for dividing the data into partitions, serializing the data, and sending the data to the appropriate brokers for storage.

34. What is a Kafka Record?

A Record in Apache Kafka is the unit of data that is stored and served by the brokers. Records are organized into topics and partitions, and consist of a key, a value, and a timestamp. The key and value can be of any data type, and are used to store the actual data, while the timestamp is used to record the time the data was produced.

35. What is a Kafka Topic Partition?

A Topic Partition in Apache Kafka is a unit of data storage that is used to organize the data within a topic. The data within a partition is ordered and immutable, meaning that once a record is written to a partition it cannot be changed. Partitions allow for parallel processing of data by consumers, and provide a way to scale the data storage and processing capabilities of a Kafka Cluster.

36. What is the purpose of a Kafka offset?

A Kafka offset is a unique identifier for a record within a partition. The offset is used by the consumers to track the current position in the partition, allowing them to resume reading from the last consumed record in the event of a failure or restart. The offset is also used by the broker to determine which records have been consumed and can be safely deleted.

37. What is the role of a Kafka Controller?

A Kafka Controller is a broker that is responsible for managing the state of the Kafka Cluster. The controller is responsible for assigning partitions to brokers, handling broker failures, and maintaining the current state of the cluster metadata. The controller is elected by the other brokers in the cluster, and if the current controller fails, a new controller will be elected to take its place.

38. What is the Kafka Request/Response model?

The Kafka Request/Response model is a communication pattern used by the brokers and clients in a Kafka Cluster. In this model, clients send requests to the brokers, and the brokers send responses back to the clients. The requests can be for tasks such as producing data, consuming data, or fetching metadata, while the responses provide the results of the requested operations.

39. What is the Kafka Broker Leader Election process?

The Kafka Broker Leader Election process is the mechanism by which a leader broker is elected for each partition in a Kafka Cluster. The leader broker is responsible for handling write requests for the partition and serving read requests to the consumers. The leader election process is handled automatically by the Kafka Cluster and is triggered in the event of a broker failure or when a new broker is added to the cluster.

40. What is a Kafka Consumer Group?

A Kafka Consumer Group is a group of consumers that work together to consume data from one or more partitions of a topic. The consumers within a consumer group work together to provide fault tolerance and scalability, as each consumer can take on a portion of the load and consume data in parallel. The consumer group also provides a mechanism for load balancing, as the partitions are automatically distributed among the consumers in the group.

41. What is a Kafka Replica?

X A Replica in Apache Kafka is a copy of a partition that is stored on a different broker in the same cluster. Replicas provide redundancy and fault tolerance, as in the event of a broker failure the replicas can be used to serve data to the consumers. Replicas also provide improved performance, as read requests can be served by the closest replica to the consumer.

42. What is the Kafka Serialization and Deserialization process?

The Kafka Serialization and Deserialization process is the process by which data is converted from its original format into a binary format suitable for transmission over the network, and then converted back into its original format by the consumer. This process is performed by the producer using a serializer to convert the data into binary form, and by the consumer using a deserializer to convert the binary data back into its original format.

43. What is the difference between a Kafka Topic and a Kafka Partition?

A Topic in Apache Kafka is a unit of data organization that is used to categorize the data within a Kafka Cluster. Topics are used to group similar data together for easier management and processing. On the other hand, a Partition in Apache Kafka is a unit of data storage that is used to physically store the data within a topic. Partitions allow for parallel processing of data by consumers, and provide a way to scale the data storage and processing capabilities of a Kafka Cluster.

44. What is the role of a Kafka Zookeeper?

Apache ZooKeeper is a distributed coordination service that is often used in conjunction with Apache Kafka. The role of ZooKeeper in a Kafka Cluster is to maintain configuration information, provide distributed synchronization, and handle failover. ZooKeeper provides a centralized and consistent view of the state of the Kafka Cluster, allowing the brokers and clients to detect and respond to changes in the cluster state.

45. What is a Kafka Partition Leader?

A Partition Leader in Apache Kafka is the broker that is responsible for handling write requests for a particular partition and serving read requests to the consumers. The leader is elected by the brokers in the cluster and can change in the event of a broker failure or when a new broker is added to the cluster. The partition leader plays a crucial role in ensuring the reliability and availability of data in a Kafka Cluster.

46. What is a Kafka Producer?

A Kafka Producer is a client application that is responsible for sending data to a Kafka Cluster. The producer is responsible for choosing which partition to send data to and for serializing the data into a binary format suitable for transmission over the network. Producers can also choose to send data to specific partitions or to allow the Kafka Cluster to automatically balance the data across partitions.

47. What is a Kafka Offset?

An Offset in Apache Kafka is a unique identifier for a message within a partition. The offset is used by consumers to keep track of their position within the partition and to ensure that they are processing the data in order. The offset is also used by the Kafka Cluster to determine which messages are stored on which broker, allowing the data to be efficiently distributed and balanced.

48. What is a Kafka Connector?

A Kafka Connector is a pre-built integration that allows data to be easily moved into or out of Apache Kafka. Connectors are used to connect to various data sources or data sinks, such as databases, message queues, or file systems, and to transfer data into or out of the Kafka Cluster. Connectors can be run standalone or as part of a larger data processing pipeline, making it easy to integrate data into a Kafka-based data architecture.

49. What is a Kafka Streams Application?

A Kafka Streams Application is a type of application that is built using the Kafka Streams API. This API allows developers to easily build real-time data processing pipelines that can process data from one or more Kafka topics, perform various transformations on the data, and write the results to one or more output topics. Kafka Streams Applications are designed to be highly scalable, fault-tolerant, and easy to develop, making them a powerful tool for building real-time data processing systems.

50. What is the role of a Kafka Cluster Manager?

A Cluster Manager in Apache Kafka is responsible for managing the lifecycle of the Kafka Cluster. This includes tasks such as adding or removing brokers, reassigning partitions, and handling failover. The cluster manager also provides a centralized view of the cluster state, allowing the brokers and clients to detect and respond to changes in the cluster configuration.

51. What is a Kafka Topic?

A Topic in Apache Kafka is a category or feed name to which messages can be published by producers and from which messages can be consumed by consumers. Topics are used to logically partition the data in a Kafka Cluster, allowing for parallel processing and efficient storage. Topics can also be used to organize the data into meaningful categories or streams for easy consumption by multiple consumers.

52. What is a Kafka Consumer Group?

X A Consumer Group in Apache Kafka is a group of consumers that work together to consume a set of partitions from one or more topics. The messages within a partition are assigned to the consumers in a consumer group in a round-robin fashion, ensuring that the load is balanced across all members of the group. Consumer groups are used to allow multiple consumers to work together to consume the data in a scalable and parallel manner.

53. What is a Kafka Broker?

A Broker in Apache Kafka is a node in the Kafka Cluster that is responsible for storing and serving data to the consumers. Brokers are used to distribute the data in the cluster, allowing for parallel processing and high availability. Brokers are also responsible for maintaining the metadata of the topics and partitions, allowing the producers and consumers to interact with the cluster.

54. What is a Kafka Compaction?

Compaction in Apache Kafka is a process that is used to reduce the storage usage of a topic by removing old and redundant data. Compaction works by only retaining the latest version of each record based on its key and discarding older versions of the same record. Compaction is performed automatically by the Kafka Cluster and can be configured for each topic based on the specific requirements of the data.

55. What is the role of a Kafka ZooKeeper?

Apache ZooKeeper is a distributed coordination service that is often used in conjunction with Apache Kafka. The role of ZooKeeper in a Kafka Cluster is to coordinate the state of the cluster, including managing configuration information, detecting and responding to failures, and electing a leader for partitions and topics. ZooKeeper helps to ensure the high availability and reliability of the Kafka Cluster by providing a centralized source of truth for the state of the cluster.

56. What is the role of a Kafka Replica?

A Replica in Apache Kafka is a duplicate of a partition that is stored on another broker in the cluster. Replicas are used to provide high availability of data in the event of a broker failure. The replicas of a partition are kept in sync with each other, allowing the partition leader to quickly failover to one of the replicas in the event of a failure. Replicas also provide a mechanism for distributing the load of serving data to the consumers, allowing for increased scalability.

57. What is the purpose of a Kafka Partition?

A Partition in Apache Kafka is a unit of parallelism in the data storage and processing of a topic. Partitions allow for horizontal scalability by allowing the data within a topic to be split across multiple brokers in the cluster. Each partition is an ordered, immutable sequence of records, with each record having a unique offset. Partitions also provide a mechanism for load balancing, allowing multiple consumers to consume the data in parallel and ensuring that the load is balanced across the consumers.

58. What is a Kafka Producer?

A Producer in Apache Kafka is a client application that is responsible for publishing data to one or more topics in the Kafka Cluster. Producers are responsible for choosing which partition to send the data to and for handling the partitioning and replication of the data within the cluster. Producers can be configured to ensure that the data is sent to the cluster in a reliable and scalable manner, using mechanisms such as batching and compression.

59. What is a Kafka Record?

A Record in Apache Kafka is a unit of data that is stored and processed in the cluster. Records consist of a key, a value, and a timestamp, and are stored in a partition within a topic. Records are the basic unit of data that is processed by the Kafka Cluster and are the means by which producers and consumers interact with the cluster.

60. What is the Kafka Offset?

The Offset in Apache Kafka is a unique identifier for each record within a partition. The offset is used by the consumers to track their progress within a partition and to ensure that they are processing the data in the correct order. The offset is also used by the cluster to provide guarantees around the delivery of data to the consumers, such as exactly-once delivery.

61. What is the difference between Kafka and RabbitMQ?

Apache Kafka and RabbitMQ are both messaging systems that are used for distributed data processing and communication between microservices. However, there are several key differences between the two systems. Kafka is designed for high-throughput, real-time data streaming, while RabbitMQ is designed for lower-throughput, queuing and routing of messages. Kafka is optimized for write-intensive workloads and has better scalability, while RabbitMQ is optimized for read-intensive workloads and has better flexibility. Finally, Kafka has a simpler model for data storage and processing, while RabbitMQ has a more flexible and feature-rich model.

62. What is the Kafka Consumer Lag?

The Consumer Lag in Apache Kafka is the difference between the highest offset that has been processed by a consumer and the latest offset in the partition. The consumer lag represents the amount of data that the consumer has not yet processed, and is an important metric for monitoring the health of a consumer and for ensuring that the data is being processed in a timely manner. The consumer lag can be used to identify issues with the consumer, such as slow processing or connectivity issues, and can also be used to trigger alerts or to take other actions to address the issue.

63. What is the role of ZooKeeper in Apache Kafka?



ZooKeeper is an integral part of the Apache Kafka ecosystem, serving as a coordination service for the cluster. ZooKeeper is used to manage the configuration of the cluster, such as the location of the brokers, and to maintain the state of the cluster, such as the current leader of a partition. ZooKeeper is also used to handle cluster membership and to handle failover in the event of a broker failure, ensuring that the cluster remains available and that data is not lost.

64. What is the Kafka Consumer Group?

A Consumer Group in Apache Kafka is a set of consumers that jointly consume the data from one or more topics. Each partition in a topic can be consumed by only one consumer in the group at any given time, and each consumer in the group will receive a unique set of partitions to consume. This allows for load balancing and parallel processing of the data, as well as for high availability in the event of a consumer failure.

65. What is the Kafka Retention Policy?

The Retention Policy in Apache Kafka determines how long data is stored in the cluster and when it is automatically deleted. The retention policy can be set on a per-topic basis and can be configured to keep data for a specific amount of time, or to keep all data indefinitely. The retention policy is a critical aspect of managing the data in a Kafka cluster, as it determines how much data is stored in the cluster and the amount of storage required. It is important to balance the retention policy with the data retention needs of the application, to ensure that the data is stored for the appropriate amount of time and to avoid running out of storage.

66. What is the Kafka Replication Factor?

The Replication Factor in Apache Kafka determines the number of copies of the data that are stored in the cluster. The replication factor is set on a per-topic basis and determines how many brokers in the cluster will store a copy of the data. A higher replication factor provides increased fault tolerance, as it ensures that multiple copies of the data are available in the event of a broker failure. However, a higher replication factor also requires more storage and more resources to manage the data.

67. What is the Kafka Compression Type?

The Compression Type in Apache Kafka determines the type of compression that is applied to the data before it is stored in the cluster. Compression can help to reduce the size of the data and to improve the performance of the cluster by reducing the amount of data that needs to be transferred and stored. Apache Kafka supports several types of compression, including Snappy, LZ4, and Gzip, and the appropriate compression type can be selected based on the needs of the application and the network conditions.

68. What is a Kafka Partition Leader?

A Partition Leader in Apache Kafka is the broker that is responsible for handling write requests for a particular partition and serving read requests to the consumers. The leader is elected by the brokers in the cluster and can change in the event of a broker failure or when a new broker is added to the cluster. The partition leader plays a crucial role in ensuring the reliability and availability of data in a Kafka Cluster.

69. What is a Kafka Broker?

A Broker in Apache Kafka is a server that runs the Kafka software and acts as a central repository for data in the cluster. The brokers are responsible for managing the storage of the data, serving data to consumers, and handling the distribution of data across the cluster. In a typical Kafka cluster, there will be multiple brokers to provide increased fault tolerance and to handle the storage and processing of large amounts of data.

70. What is a Kafka Topic?

A Topic in Apache Kafka is a named channel for storing data. Topics are used to logically organize the data in the cluster and to provide a way for producers and consumers to interact with the data. Topics are partitioned and each partition can be stored on one or more brokers in the cluster, providing a way to horizontally scale the storage and processing of data.

71. What is a Kafka Partition?

A Partition in Apache Kafka is a horizontally-divided subset of a topic's data. Partitions are used to scale the storage and processing of data in the cluster, as each partition can be stored on a different broker and can be processed in parallel by multiple consumers. Each partition is ordered and has a unique identifier, allowing for the data in the partition to be read in a specific order.

72. What is a Kafka Producer?

A Producer in Apache Kafka is a client application that writes data to a topic in the cluster. Producers are responsible for sending data to the broker, including the topic and partition to which the data should be written. Producers can be configured to control the way that data is written to the cluster, such as the batch size, the number of acknowledgements required, and the compression type.

73. What is a Kafka Consumer?

A Consumer in Apache Kafka is a client application that reads data from a topic in the cluster. Consumers subscribe to one or more topics and pull data from the broker as it becomes available. Consumers can be part of a consumer group, which allows for parallel processing of data and for load balancing across multiple consumers. Consumers can be configured to control the way that data is read from the cluster, such as the batch size and the starting offset.

74. What is a Kafka Offset?

An Offset in Apache Kafka is a unique identifier for a particular message within a partition. Offsets are used by consumers to keep track of their position within a partition and to ensure that they receive all of the messages in the partition. The offset of a message is automatically managed by the broker and is stored with the message. Consumers can start consuming data from a specific offset, allowing them to resume processing from where they left off in the event of a failure.

Tip: Sketch broker/partition diagrams when answering replication or rebalancing questions.

SEARCH

Search topics...

Go

INTERVIEW TOPICS

Java	Q&A
Java 8	Q&A
Spring	Q&A
Hibernate	Q&A
React JS	Q&A
SQL	Q&A
MongoDB	Q&A
JavaScript	Q&A
Kafka	Q&A
Design Patterns	Q&A
Microservices	Q&A
REST API	Q&A
Behavioral	Q&A

QUICK ACTIONS

Start System Design

Practice Coding

Earn a Certificate

RECOMMENDED READING

[Learn Java 17 Programming](#)

[Learn JavaScript Quickly](#)

[SQL QuickStart Guide](#)

[System Design Interview](#)

THANK YOU!

Your comments and complements are always the inspiration that keeps us motivated.

ABOUT

[PROJECTS](#)

[ABOUT US](#)

[BLOG](#)

[AWARDS](#)

ADDRESS

- 

Sterling, Virginia, 20166, US
- 

info@techventurellc.com
- 

641 451 3463
- 

641 451 3463

FOLLOW US

- 
- 
- 
- 