Apache Kafka Interview Questions

Q.1- Mention what Apache Kafka is?

Apache Kafka is a publish-subscribe messaging system developed by Apache written in Scala. It is a distributed, partitioned and replicated log service.

Q.2- Mention what is the traditional method of message transfer?

The traditional method of message transfer includes two methods

- Queuing: In a queuing, a pool of consumers may read message from the server and each message goes to one of them
- Publish-Subscribe: In this model, messages are broadcasted to all consumers

Kafka caters single consumer abstraction that generalized both of the above- the consumer group.

Q.3- Mention what are the benefits of Apache Kafka over the traditional technique?

Apache Kafka has following benefits above traditional messaging technique

- Fast: A single Kafka broker can serve thousands of clients by handling megabytes of reads and writes per second
- Scalable: Data are partitioned and streamlined over a cluster of machines to enable larger data
- Durable: Messages are persistent and is replicated within the cluster to prevent data loss
- Distributed by Design: It provides fault tolerance guarantees and durability

Q.4- Mention what is the meaning of Broker in Kafka?

In the Kafka cluster, broker term is used to refer to Server.

Q.5- Mention what is the Maximum Size of the Message does Kafka server can Receive?

The maximum size of the message that Kafka server can receive is 1000000 bytes.

Q.6- Explain what is Zookeeper in Kafka and can we use Kafka without Zookeeper?

Zookeeper is an open source, high-performance co-ordination service used for distributed applications adapted by Kafka. No, it is not possible to bye-pass Zookeeper and connect straight to the Kafka broker. Once the Zookeeper is down, it cannot serve client requests. Zookeeper is basically used to communicate between different nodes in a cluster In Kafka, it is used to commit offset, so if node fails in any case it can be retrieved from the previously committed offset Apart from this it also does other activities like leader detection, distributed synchronization, configuration management, identifies when a new node leaves or joins, the cluster, node status in real time, etc.

Q.7- Explain how messages are consumed by consumers in Kafka?

Transfer of messages in Kafka is done by using sendfile API. It enables the transfer of bytes from the socket to disk via kernel space saving copies and calls between kernel users back to the kernel.

Q.8- Explain how you can improve the throughput of a remote consumer?

If the consumer is located in a different data centre from the broker, you may require to tune the socket buffer size to amortize the long network latency.

Q.9- Explain how you can get Exactly Once Messaging from Kafka during data production?

During data production to get exactly once messaging from Kafka you have to follow two things: avoiding duplicates during data consumption and avoiding duplication during data production. Here are the two ways to get exactly one semantics while data production: Avail a single writer per partition, every time you get a network error checks the last message in that partition to see if your last write succeeded In the message include a primary key (UUID or something) and de-duplicate on the consumer

Q.10- Explain how you can reduce churn in Isr and when does Broker leave the Isr?

ISR is a set of message replicas that are completely synced up with the leaders, in other word ISR has all messages that are committed. ISR should always include all replicas until there is a real failure. A replica will be dropped out of ISR if it deviates from the leader.

Q.11- Why Replication is required in Kafka?

Replication of messages in Kafka ensures that any published message does not lose and can be consumed in case of machine error, program error or more common software upgrades.

Q.12- What does it indicate if a replica stays out of Isr for a long time?

If a replica remains out of ISR for an extended time, it indicates that the follower is unable to fetch data as fast as data accumulated at the leader.

Q.13- Mention what happens if the preferred replica is not in the Isr?

If the preferred replica is not in the ISR, the controller will fail to move leadership to the preferred replica.

Q.14- Is it possible to get the Message Offset after Producing?

You cannot do that from a class that behaves as a producer like in most queue systems, its role is to fire and forget the messages. The broker will do the rest of the work like appropriate metadata handling with id’s, offsets, etc. As a consumer of the message, you can get the offset from a Kafka broker. If you look in the SimpleConsumer class, you will notice it fetches MultiFetchResponse objects that include offsets as a list. In addition to that, when you iterate the Kafka Message, you will have MessageAndOffset objects that include both, the offset and the message sent.

Q.15- Mention what is the difference between Apache Kafka and Apache Storm?

Apache Kafka: It is a distributed and robust messaging system that can handle huge amounts of data and allows passage of messages from one end-point to another. Apache Storm: It is a real time message processing system, and you can edit or manipulate data in real time. Apache storm pulls the data from Kafka and applies some required manipulation.

Q.16- List the various components in Kafka?

The four major components of Kafka are:

- Topic – a stream of messages belonging to the same type
- Producer – that can publish messages to a topic
- Brokers – a set of servers where the publishes messages are stored
- Consumer – that subscribes to various topics and pulls data from the brokers.

Q.17- Explain the role of the Offset?

Messages contained in the partitions are assigned a unique ID number that is called the offset. The role of the offset is to uniquely identify every message within the partition.

Q.18- Explain the concept of Leader and Follower?

Every partition in Kafka has one server which plays the role of a Leader, and none or more servers that act as Followers. The Leader performs the task of all read and write requests for the partition, while the role of the Followers is to passively replicate the leader. In the event of the Leader failing, one of the Followers will take on the role of the Leader. This ensures load balancing of the server.

Q.19- How do you define a Partitioning Key?

Within the Producer, the role of a Partitioning Key is to indicate the destination partition of the message. By default, a hashing-based Partitioner is used to determine the partition ID given the key. Alternatively, users can also use customized Partitions.

Q.20- In the Producer when does Queuefullexception occur?

QueueFullException typically occurs when the Producer attempts to send messages at a pace that the Broker cannot handle. Since the Producer doesn't block, users will need to add enough brokers to collaboratively handle the increased load.

Q.21- Explain the role of the Kafka Producer Api.

The role of Kafka's Producer API is to wrap the two producers – kafka.producer.SyncProducer and the kafka.producer.async.AsyncProducer. The goal is to expose all the producer functionality through a single API to the client.