

1. How will you define Kafka?

Kafka is an open-source message broker project that is written in Scala programming language and it is an initiative by Apache Software Foundation. A unique set of features make it most suitable choice for the data integration and one of the leading data processing tools of choice.

2. What are the main features of Kafka that make it suitable for data integration and data processing in real-time?

Some of the most highlighting features of Kafka that make it popular worldwide includes – data partitioning, scalability, low-latency, high throughputs etc. These features are the reason why Kafka had become the most suitable choice for data integration and data processing in the real-time.

3. What are the major components of Kafka integration product?



- **Topic**—The messages stream that belongs to the same pattern.
- **Producer**—It helps in publishing messages to the topic.
- **Broker**—This is a set of various servers where all published data is stored.
- **Consumer**—It subscribes to the different topics and fetch data from the brokers.

4. Explain the offset in Kafka data integration tool?

Messages are stored in partitions and assigned a unique ID to each of them for quick and easy access. That unique number is named as the offset that is responsible to identify each of the messages in the partition.

5. What is Zookeeper and Is it possible to run Kafka without Zookeeper?

Zookeeper is used to store offset values of messages. There is no alternative of Zookeeper in Kafka. In case, Zookeeper is down then this is not possible to serve any of the client requests.

6. What is the meaning of Leader and Follower in Kafka?

Every partition in Kafka has one main server that is named as “Leader” and one or more non-connected servers that are named as the “Followers”.

7. What is a consumer group in Kafka?

A Consumer group is made up of one or more consumers that together subscribe to the different topics and fetch data from the brokers.

8. How to balance loads in Kafka when one server fails?

Every partition in Kafka has one main server that plays the role of a leader and one or more non-connected servers that are named as the followers. Here, the leading server sets the permission and rest of the servers just follow him accordingly. In case, leading server fails then followers take the responsibility of the main server.

9. Do you know any traditional technique of message transfer?

Yes, these techniques are queuing, and publish-subscribe. However, Kafka generalizes both of the techniques through consumer group.

10. How is Kafka preferred over traditional message transfer techniques?

Kafka product is more scalable, faster, robust and distributed by design.

11. Explain the meaning of broker in Kafka?

Broker and server have the meaning in Kafka.

12. Explain the maximum size of a message that can be received by the Kafka?

It is approx. 1000000 bytes.

13. Do you know how to improve the throughput of the remote consumer?

Well, it is interesting and advance concept in Kafka. If the consumer is located in the distant location then you need to optimize the socket buffer size to tune the overall throughput of a remote consumer.

14. Do replication is necessary or just a waste of time in Kafka?

Replicating messages is a good practice in Kafka that assure that messages will never lose even if the main server fails.

15. Is it possible to get the offset value of the message once it is produced already?

No, we cannot do that.

16. What is the main difference between Kafka and Fume?

Both products are used to process data in the real-time but Kafka is proven more scalable and ensures durability.

17. Explain the role of producer API in Kafka?

A producer API exposes the functionalities of all producers through a single API to the client.

18. In the producer, when there comes the situation of queue fullness?

If there are not enough number of servers added for load balancing, there comes a situation of queue fullness.

19. How to initiate the Kafka server? Do you know the process?

Yes, I know. To initiate the Kafka server, you need to initiate the Zookeeper server first then you could fire up the Kafka server.

20. How will you explain the Kafka architecture?

Kafka product is based on a distributed design where one cluster has multiple brokers/servers associated with it. The 'Topic' will be divided into plenty of partitions to store the messages and there is one consumer group to fetch the messages from brokers.

21. What are partitions and offsets in Kafka Topics?

Kafka cluster maintains a partition of logs for each Kafka Topic. Each partition is a structured commit log containing a sequence of ordered, immutable records that is continuously appended to.

Each of the records in the partition are assigned a sequence number, called the offset, that uniquely identifies each record within each partition.



Producers and Consumers of a Kafka Topic