**1. What is Kafka?**

Wikipedia defines Kafka as “an open-source message broker project developed by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) written in Scala and is a distributed publish-subscribe messaging system.

|  |  |
| --- | --- |
| **Kafka Salient Features** | |
| **Feature** | **Description** |
| ***High Throughput*** | Support for millions of messages with modest hardware |
| ***Scalability*** | Highly scalable distributed systems with no downtime |
| ***Replication*** | Messages are replicated across the cluster to provide support for multiple subscribers and balances the consumers in case of failures |
| ***Durability*** | Provides support for persistence of message to disk |
| ***Stream Processing*** | Used with real-time streaming applications like Apache Spark & Storm |
| ***Data Loss*** | Kafka with proper configurations can ensure zero data loss |

**2. 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.

**3. 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.

**4. What is a Consumer Group?**

Consumer Groups is a concept exclusive to Kafka.  Every Kafka consumer group consists of one or more consumers that jointly consume a set of subscribed topics.

**5. What is the role of the ZooKeeper?**

Kafka uses Zookeeper to store offsets of messages consumed for a specific topic and partition by a specific Consumer Group.

**6. Is it possible to use Kafka without ZooKeeper?**

No, it is not possible to bypass Zookeeper and connect directly to the Kafka server. If, for some reason, ZooKeeper is down, you cannot service any client request.

**7. 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.

**8. What roles do Replicas and the ISR play?**

Replicas are essentially a list of nodes that replicate the log for a particular partition irrespective of whether they play the role of the Leader. On the other hand, ISR stands for In-Sync Replicas. It is essentially a set of message replicas that are synced to the leaders.

**9. Why are Replications critical in Kafka?**

Replication ensures that published messages are not lost and can be consumed in the event of any machine error, program error or frequent software upgrades.

**10. If a Replica stays out of the ISR for a long time, what does it signify?**

It means that the Follower is unable to fetch data as fast as data accumulated by the Leader.

**11. What is the process for starting a Kafka server?**

Since Kafka uses ZooKeeper, it is essential to initialize the ZooKeeper server, and then fire up the Kafka server.

* To start the ZooKeeper server: > bin/zookeeper-server-start.sh config/zookeeper.properties
* Next, to start the Kafka server: > bin/kafka-server-start.sh config/server.properties

**12. 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.

**13. 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.

**14. 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.

**15. What is the main difference between Kafka and Flume?**

Even though both are used for real-time processing, Kafka is scalable and ensures message durability.

These are some of the frequently asked Apache Kafka interview questions with answers. You can brush up on your knowledge of Apache Kafka with [these](https://www.edureka.co/blog/category/apache-kafka/)blogs.