Apache Kafka

Q.1 What is Apache Kafka?
Ans. Apache Kafka is a publish-subscribe open source message broker application. This messaging application was coded in "**Scala**".

Explain partitions in Apache Kafka.

- Topics in Kafka are divided into partitions. One or more consumers can read data from a Kafka topic simultaneously by reading from each partition.
- The partitions are separated in order. While creating a topic, you need to specify the number of partitions, although this number is arbitrary and can be changed later.

Q.2 Enlist the several components in Kafka.

- **Topic –** Kafka Topic is the bunch or a collection of messages.
- **Producer –** In Kafka, Producers issue communications as well as publishes messages to a Kafka topic.
- **Consumer –** Kafka Consumers subscribes to a topic(s) and also reads and processes messages from the topic(s).
- **Brokers –** While it comes to manage storage of messages in the topic(s) we use Kafka Brokers.

Q.3 Explain the role of the offset.
Ans. There is a sequential ID number given to the messages in the partitions what we call, an offset. So, to identify each message in the partition uniquely, we use these offsets.

Q.4 What is a Consumer Group?
Ans. The concept of Consumer Groups is exclusive to Apache Kafka. Basically, every Kafka consumer group consists of one or more consumers that jointly consume a set of subscribed topics.

Mention the APIs provided by Apache Kafka.

- Kafka Producer API: The producer API allows applications to publish messages in the form of a stream of records to one or more Kafka topics.

- Kafka Consumer API: The consumer API allows applications to subscribe to one or more Kafka topics.
- Kafka Streams API: The Kafka streams API allows applications to process data in a stream processing paradigm.
- Kafka Connector API: The connector API helps connect applications to Kafka topics. It provides features for managing the running of producers and consumers and handling the connections between them.

What is meant by ZooKeeper in Apache Kafka?

- The ZooKeeper in Kafka is responsible for managing and coordinating the Kafka cluster.
- It ensures coordination between different nodes in a cluster.
- The changes include when brokers or topics are removed or added.

How is load balancing maintained in Kafka?

- Load balancing in Kafka is handled by the producers. The message load is spread out between the various partitions while maintaining the order of the message.
- By default, the producer selects the next partition to take up message data using a round-robin approach.

What are some differences between Apache Kafka and Flume?

Apache Kafka	**Apache Flume**
Kafka is optimized to ingest data and process streaming data in real-time.	Flume is mainly used for collecting and aggregating large amounts of log data from multiple sources.
Easy to scale.	Not as easy to scale as Kafka.

It can be supported across various applications.	Specifically designed for Hadoop.
Apache Kafka runs as a cluster and supports automatic recovery if resilient to node failure.	Tool to collect log data from distributed web servers.

What is the role of the Partitioning Key?

- Messages are sent to various partitions associated with a topic in a round-robin fashion.
- If there is a requirement to send a message to a particular partition, then it is possible to associate a key with the message.
- The key determines which partition that particular message will go to.
- All messages with the same key will go to the same partition. If a key is not specified for a message, the producer will choose the partition in a round-robin fashion.

Explain the roles of leader and follower in Apache Kafka.

- Every partition in the Kafka server has one server that plays the role of a leader.
- The leader performs all the read and write data tasks for a partition.
- A partition can have no followers, one follower, or more than one follower.
- The job of the follower is to replicate the leader. In such a case, if there is a failure in the leader, then one of the followers can take on the leader's load.

What is the purpose of ISR in Apache Kafka?

- ISR - in-synch replicas refers to all the replicated partitions that are completely synced up with the leader.

- A replica has to be fully caught up with the leader within a configurable amount of time.
- By default, this time is 10 seconds. After this period of time, if a follower is not caught up with the leader, the leader will drop the follower from its ISR and writes will continue on the remaining replicas in the ISR.

What is meant by partition offset in Apache Kafka?

- Every time message or record is assigned to a partition in Kafka, it is provided with an offset.
- The offset denotes the position of the record in that partition.
- A record can be uniquely identified within a partition using the offset value.
- Records are always added to the ends of partitions, and therefore, older records will have a lower offset.

Explain fault tolerance in Apache Kafka.

- In Kafka, the partition data is copied to other brokers, which are known as replicas.
- If there is a point of failure in the partition data in one node, then there are other nodes that will provide a backup and ensure that the data is still available

What is the best way to start the Kafka server?

- Start the ZooKeeper service:

```
$bin/zookeeper-server-start.sh config/zookeeper.properties
```

- Open another terminal and run the following to start the Kafka broker service:

```
$ bin/kafka-server-start.sh config/server.properties
```

What is Geo-Replication in Kafka?

- Geo-Replication in Kafka is a process by which you can duplicate messages in one cluster across other data centers or cloud regions.
- Geo-replication involves copying all the files and allows them to be stored across the globe if required.
- Geo-replication is a way to ensure that the data is backed up.

What is meant by multi-tenancy in Apache Kafka?

- Multi-tenancy refers to the mode of operation of software where there are multiple instances of one or multiple applications operating in a shared environment independent from each other.
- The instances are said to be logically isolated but physically integrated.
- Kafka can be said to be multi-tenant as it allows for configuring different topics for which data can be consumed or produced on the same cluster.

Explain the topic replication factor.

- Topic replication factor refers to the number of copies of the topic that are present over multiple brokers.
- The replication factor should be greater than 1 for fault tolerance.

Mention some real-world use cases of Apache Kaka.

- Message Broker: Kafka is capable of appropriate metadata handling, i.e., a large volume of similar types of messages or data, due to its high throughput value.
- Monitor Operational Data: Kafka can be used to monitor the metrics associated with certain technologies, such as security logs.

- Tracking website activities: Kafka can be used to ensure that data is successfully sent and received by websites.
- Data logging: Kafka's feature of data replication across the nodes can be used to restore data on failed nodes.
- Kafka for Stream Processing: Kafka can handle streaming data wherein data is read from a topic, processed, and written to a new topic. The new topic containing the processed data will be available to users and applications.

Define the role of Kafka Streams API and Kafka Connector API.

- Streams API enables an application to work as a stream processor by efficiently changing input streams into output streams.
- Streams API is responsible for receiving input streams from one or more topics and sending output streams to one or more output topics.
- Connector API Connects Kafka topics to applications. The connector API enables the execution and building of reusable producers or consumers that link Kafka topics to pre-existing applications or data systems.

What is the Kafka Mirror Maker?

- The Kafka Mirror Maker is a stand-alone tool that allows data to be copied from one Apache Kafka cluster to another.
- The Kafka Mirror Maker will read data from topics in the original cluster and write the topics to a destination cluster with the same topic name.

What is the Confluent Replicator?

- The Confluent Replicator allows easy and reliable replication of topics from a source cluster to a destination cluster.

- It continuously copies messages from the source to the destination and even assigns the same names to the topics in the destination cluster.

What is the command to start ZooKeeper?

bin/zookeeper-server-start.sh

Explain how topic configurations can be modified in Apache Kafka.

- To add a config: bin/kafka-configs.sh --zookeeper localhost:2181 --topics --topic_name --alter --add-config x=y
- To remove a config: bin/kafka-configs.sh --zookeeper localhost:2181 --topics --topic_name --alter --delete-config x

Explain the term "Log Anatomy".

- a data source writes messages to the log. One of the advantages is, at any time one or more consumers read from the log they select.

Explain message compression in Apache Kafka.

- In Apache Kafka, producer applications write data to the brokers in JSON format.
- The data in the JSON format is stored in string form, which can result in several duplicated records getting stored in the Kafka topic.
- This leads to an increased occupation of disk space.
- Message compression is done on the producer side, and hence there is no need to make any changes to the configuration of the consumer or the broker.

What is meant by Kafka Connect?

- Kafka Connect is a tool provided by Apache Kafka to allow scalable and reliable streaming data to move between Kafka and other systems.

- It makes it easier to define connectors that are responsible for moving large collections of data in and out of Kafka.
- Kafka Connect is able to process entire databases as input.

Explain producer batch in Apache Kafka.

- Producers write messages to Kafka, one at a time.
- Kafka waits for the messages that are being sent to Kafka, creates a batch and puts the messages into the batch, and waits until this batch becomes full.
- Only then is the batch sent to Kafka. The batch here is known as the producer batch. The default size of a producer batch is 16KB, but it can be modified.

Mention some use cases where Apache Kafka is not suitable.

- Kafka is built to handle high volumes of data. If the requirement is to process only a small number of messages per day, a traditional messaging system would be more suitable.
- Kafka has a streaming API, but it is not suitable for performing data transformation operations. Kafka is to be avoided for ETL (extract, transform, load) jobs.
- In cases where a simple task queue is needed, there are better alternatives like the RabbitMQ.
- Kafka is not good if long-term storage is required. It supports saving data only for a specified retention period and not longer than that.

Why is the Kafka broker said to be “dumb”?

- The Kafka broker does not keep a tab of which the consumers have read messages. It simply keeps all of the messages in its queue for a fixed time, known as the retention time, after which the messages are deleted.

Explain customer serialization and deserialization in Kafka.

- In Kafka, message transfer among the producer, broker, and consumers is done by making use of a standardized binary message format.
- The process of converting the data into a stream of bytes for the purpose of the transmission is known as serialization.
- Deserialization is the process of converting the bytes of arrays into the desired data format.

Enlist all Apache Kafka Operations.

Ans. Apache Kafka Operations are:

1. Addition and Deletion of Kafka Topics
2. How to modify the Kafka Topics
3. Distinguished Turnoff
4. Mirroring Data between Kafka Clusters
5. Finding the position of the Consumer
6. Expanding Your Kafka Cluster
7. Migration of Data Automatically
8. Retiring Servers
9. Datacenters

What is meant by the Kafka schema registry?

- For both the producers and consumers associated with a Kafka cluster, a Schema Registry is present, which stores Avro schemas.
- Kafka Schema Registry is used to ensure that there is no difference in the schema that is being used by the consumer and the one that is being used by the producer.
- The consumer uses the schema ID to look up the corresponding schema in the Schema Registry.

What is the ZooKeeper ensemble?

- ZooKeeper works as a coordination system for distributed systems and is a distributed system on its own.

- It follows a simple client-server model, where clients are the machines that make use of the service, and the servers are nodes that provide the service.
- The collection of ZooKeeper servers forms the ZooKeeper ensemble. Each ZooKeeper server is capable of handling a large number of clients.

What are Znodes?

- Nodes in a ZooKeeper tree are referred to as znodes. Znodes maintain a structure that contains version numbers for data changes, acl changes, and also timestamps.
- The version number, along with the timestamp, allows ZooKeeper to validate the cache and ensure that updates are coordinated.
- The version number associated with Znode increases each time the znode's data changes.

What are ZooKeeper barriers?

- In ZooKeeper, barriers are primitives that allow a group of processes to remain in synchronization at the start and the end of a computation.

Explain Apache Kafka Use Cases?

- **Kafka Metrics**

It is possible to use Kafka for operational monitoring data. Also, to produce centralized feeds of operational data, it involves aggregating statistics from distributed applications.

- **Kafka Log Aggregation**

Moreover, to gather logs from multiple services across an organization.

- **Stream Processing**

While stream processing, Kafka's strong durability is very useful.

Features of Kafka Stream.

Ans. Some best features of Kafka Stream are

- Kafka Streams are highly scalable and fault-tolerant.
- Kafka deploys to containers, VMs, bare metal, cloud.

- We can say, Kafka streams are equally viable for small, medium, & large use cases.
- Also, it is fully in integration with Kafka security.
- Write standard Java applications.
- Exactly-once processing semantics.

What do you mean by Stream Processing in Kafka?
Ans. The type of processing of data continuously, real-time, concurrently, and in a record-by-record fashion is what we call Kafka Stream processing.

What are the types of System tools?
Ans. There are three types of System tools:

- **Kafka Migration Tool**

It helps to migrate a broker from one version to another.

- **Mirror Maker**

Mirror Maker tool helps to offer to mirror of one Kafka cluster to another.

- **Consumer Offset Checker**

What are Replication Tool and its types?
Ans. For the purpose of stronger durability and higher availability, replication tool is available here. Its types are –

- Create Topic Tool
- List Topic Tool
- Add Partition Tool

Explain some Kafka Streams real-time Use Cases.
Ans. So, the use cases are:

- **The New York Times**
  - This company uses it to store and distribute, in real-time, published content to the various applications and systems that make it available to the readers. Basically, it uses Apache Kafka and the Kafka Streams both.
- **Zalando**
  - As an ESB (Enterprise Service Bus) as the leading online fashion retailer in Europe Zalando uses Kafka.
- **LINE**
  - Basically, to communicate to one another LINE application uses **Apache Kafka** as a central data hub for their services.