**What is Apache Kafka?**

* Apache Kafka is a distributed publish-subscribe messaging system. It is a scalable, fault-tolerant, publish-subscribe messaging system which enables us to build distributed applications. It is an Apache Top Level project. Kafka is suitable for both offline and online message consumption.

**What are the advantages of using Apache Kafka?**

* The Advantages of using Apache Kafka are as follows-**High Throughput-**  
  The design of Kafka enables the platform to process messages at very fast speed. The processing rates in Kafka can exceed beyond 100k/seconds. The data is processed in a partitioned and ordered fashion.
* **Scalability-**  
  The scalability can be achieved in Kafka at various levels. Multiple producers can write to the same topic. Topics can be partitioned. Consumers can be grouped to consume individual partitions.
* **Fault Tolerance-**  
  Kafka is a distributed architecture which means there are several nodes running together to serve the cluster. Topics inside Kafka are replicated. Users can choose the number of replicas for each topic to be safe in case of a node failure. Node failure in cluster won’t impact. Integration with Zookeeper provides producers and consumers accurate information about the cluster. Internally each topic has its own leader which takes care of the writes. Failure of node ensures new leader election.
* **Durability-**  
  Kafka offers data durability as well. The message written in Kafka can be persisted. The persistence can be configured. This ensures re-processing, if required, can be performed.

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.

**What role ZooKeeper plays in a cluster of Kafka?**

* Kafka is an open source system and also a distributed system is built to use Zookeeper. The basic responsibility of Zookeeper is to build coordination between different nodes in a cluster. Since Zookeeper works as periodically commit offset so that if any node fails, it will be used to recover from previously committed to offset.
* The ZooKeeper is also responsible for configuration management, leader detection, detecting if any node leaves or joins the cluster, synchronization, etc.

**Can Kafka be utilized without Zookeeper?**

* It is impossible to use [Kafka without Zookeeper](https://intellipaat.com/tutorial/kafka-tutorials/kafka-configuration/) because it is not feasible to go around Zookeeper and attach in a straight line to the server. If the Zookeeper is down for a number of causes, then we will not be able to serve any customer demand.

**Elaborate Kafka architecture.**

* A cluster contains multiple brokers since it is a distributed system. Topic in the system will get divided into multiple partitions and each broker store one or more of those partitions so that multiple producers and consumers can publish and retrieve messages at the same time.

**How to start a Kafka server?**

* Given that Kafka exercises Zookeeper, we have to start the Zookeeper’s server.
* One can use the convince script packaged with Kafka to get a crude but effective single node Zookeeper instance> bin/zookeeper-server-start.shconfig/zookeeper.properties Now the Kafka server can start> bin/Kafka-server-start.shconfig/server.properties

**What are consumers or users?**

* Kafka provides single consumer abstractions that discover both queuing and publish-subscribe Consumer Group. They tag themselves with a user group and every communication available on a topic is distributed to one user case within every promising user group. User instances are in disconnected process. We can determine the messaging model of the consumer based on the consumer groups.
* If all consumer instances have the same consumer set, then this works like a conventional queue adjusting load over the consumers.
* If all customer instances have dissimilar consumer groups, then this works like a publish-subscribe and all messages are transmitted to all the consumers.

**Why is Kafka technology significant to use?**

* Kafka being distributed publish-subscribe system has the advantages as below. Fast: Kafka comprises of a broker and a single broker can serve thousands of clients by handling megabytes of reads and writes per second. Scalable: facts are partitioned and streamlined over a cluster of machines to enable large information Durable: Messages are persistent and is replicated in the cluster to prevent record loss Distributed by Design: It provides fault tolerance guarantees and robust.

**What is Kafka Logs?**

* An important concept for Apache Kafka is â€œlogâ€. This is not related to application log or system log. This is a log of the data. It creates a loose structure of the data which is consumed by Kafka. The notion of â€œlogâ€ is an ordered, append-only sequence of data. The data can be anything because for Kafka it will be just an array of bytes.

**When not to use Apache Kafka?**

* Kafka doesn’t number the messages. It has a notion of offset inside the log which identifies the messages.
* Consumers consume the data from topics but Kafka does not keep track of the message consumption. Kafka does not know which consumer consumed which message from the topic. The consumer or consumer group has to keep a track of the consumption.
* There are no random reads from Kafka. Consumer has to mention the offset for the topic and Kafka starts serving the messages in order from the given offset.
* Kafka does not offer the ability to delete. The message stays via logs in Kafka till it expires (until the retention time defined).

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.

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.

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.

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.

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.

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