Kafka Questions

<https://www.edureka.co/blog/interview-questions/top-apache-kafka-interview-questions-for-beginners/>

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.

<https://intellipaat.com/blog/interview-question/kafka-interview-questions/>

**Which are the elements of Kafka?**

The most important elements of Kafka:

* Topic – It is the bunch of similar kind of messages
* Producer – using this one can issue communications to the topic
* Consumer – it endures to a variety of topics and takes data from brokers.
* Brokers – this is the place where the issued messages are stored
* **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.

**4. What is Kafka?**

Kafka is a message divider project coded in Scala. Kafka is originally developed by LinkedIn and developed as an open sourced in early 2011. The purpose of the project is to achieve the best stand for conducting the real-time statistics nourishment.

**5. Why do you think the replications are dangerous in Kafka?**

Duplication assures that issued messages which are available are absorbed in the case of any appliance mistake, plan fault or recurrent software promotions.

**6. What major role a Kafka Producer API plays?**

It is responsible for covering the two producers- kafka.producer.SyncProducer and the kafka.producer.async.AsyncProducer. The main aim is to disclose all the producer performance through a single API to the clients.

**8. Describe partitioning key?**

Its role is to specify the target divider of the memo, within the producer. Usually, a hash-oriented divider concludes the divider ID according to the given factors. Consumers also use the tailored Partitions.

**9. Inside the manufacturer, when does the QueueFullException emerge?**

QueueFullException naturally happens when the manufacturer tries to propel communications at a speed which Broker can’t grip. Consumers need to insert sufficient brokers to collectively grip the amplified load since the Producer doesn’t block.

**10. 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.

**11. 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.

**12. How to start a Kafka server?**

Given that Kafka exercises Zookeeper, we have to start the Zookeeper’s server.  
Learn more in this [Zookeeper Tutorial now](https://intellipaat.com/tutorial/hbase-tutorial/installation/).

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.propertiesNow the Kafka server can start> bin/Kafka-server-start.shconfig/server.properties

**13. 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.

**14. Describe an Offset?**

The messages in the partitions will be given a sequential ID number known as an offset, the offset will be used to identify each message in the partition uniquely. With the aid of Zookeeper Kafka stores the offsets of messages consumed for a specific topic and partition by this consumer group.

**15. What do you know about partitioning key?**

A partition key can be precise to point to the aimed division of a communication, in Kafka producer. Usually, a hash-oriented divider concludes the division id with the input and people uses modified divisions also.

**16. 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 informationDurable: Messages are persistent and is replicated in the cluster to prevent record loss Distributed by Design: It provides fault tolerance guarantees and robust.

<https://data-flair.training/blogs/kafka-interview-questions/>

**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.

**Q.5 What is the role of the ZooKeeper in Kafka?**

**Ans.** Apache Kafka is a distributed system is built to use Zookeeper. Although, Zookeeper’s main role here is to build coordination between different nodes in a cluster. However, we also use Zookeeper to recover from previously committed offset if any node fails because it works as periodically commit offset.

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

**Ans.** It is impossible to bypass Zookeeper and connect directly to the Kafka server, so the answer is no. If somehow, ZooKeeper is down, then it is impossible to service any client request.

**Q.7 What do you know about Partition in Kafka?**

**Ans.** In every Kafka broker, there are few partitions available. And, here each partition in Kafka can be either a leader or a replica of a topic.

**Q.8 Why is Kafka technology significant to use?**

**Ans.** There are some advantages of Kafka, which makes it significant to use:

* **High-throughput**

We do not need any large hardware in Kafka, because it is capable of handling high-velocity and high-volume data. Moreover, it can also support message throughput of thousands of messages per second.

* **Low Latency**

Kafka can easily handle these messages with the very low latency of the range of milliseconds, demanded by most of the new use cases.

* **Fault-Tolerant**

Kafka is resistant to node/machine failure within a cluster.

* **Durability**

As Kafka supports messages replication, so,  messages are never lost. It is one of the reasons behind durability.

* **Scalability**

Kafka can be scaled-out, without incurring any downtime on the fly by adding additional nodes.

**Q.9 What are main APIs of Kafka?**

**Ans.** Apache Kafka has 4 main APIs:

1. Producer API
2. Consumer API
3. Streams API
4. Connector API

**Q.10 What are consumers or users?**

**Ans.** Mainly, [**Kafka Consumer**](https://data-flair.training/blogs/kafka-consumer/) subscribes to a topic(s), and also reads and processes messages from the topic(s). Moreover, with a consumer group name, Consumers label themselves. In other words, within each subscribing consumer group, each record published to a topic is delivered to one consumer instance. Make sure it is possible that Consumer instances can be in separate processes or on separate machines.

**Apache Kafka Interview Questions For Freshers.Q- 1, 2, 4, 7, 8, 9, 10**

**Apache Kafka Interview Questions for Experience.Q- 3, 5, 6**

### **3. Tricky Kafka Interview Questions and Answers**

**Q.11 Explain the concept of Leader and Follower.**

Ans. In every partition of Kafka, there is one server which acts as the Leader, and none or more servers plays the role as a Followers.

**Q.12 What ensures load balancing of the server in Kafka?**

Ans. As the main role of the Leader is to perform the task of all read and write requests for the partition, whereas Followers passively replicate the leader. Hence, at the time of Leader failing, one of the Followers takeover the role of the Leader. Basically, this entire process ensures load balancing of the servers.

**Q.13 What roles do Replicas and the ISR play?**

Ans. Basically, a list of nodes that replicate the log is Replicas. Especially, for a particular partition. However, they are irrespective of whether they play the role of the Leader.  
In addition, ISR refers to In-Sync Replicas. On defining ISR, it is a set of message replicas that are synced to the leaders.

**Q.14 Why are Replications critical in Kafka?**

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

**Q.15 If a Replica stays out of the ISR for a long time, what does it signify?**

Ans. Simply, it implies that the Follower cannot fetch data as fast as data accumulated by the Leader.

**Q.16 What is the process for starting a Kafka server?**

Ans. It is the very important step to initialize the ZooKeeper server because Kafka uses ZooKeeper.So, the process for starting a Kafka server is:  
**In order 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**

**Q.17 In the Producer, when does QueueFullException occur?**

Ans. whenever the [**Kafka Producer**](https://data-flair.training/blogs/kafka-producer/) attempts to send messages at a pace that the Broker cannot handle at that time QueueFullException typically occurs. However, to collaboratively handle the increased load, users will need to add enough brokers, since the Producer doesn’t block.

**Q.18 Explain the role of the Kafka Producer API.**

Ans. An API which permits an application to publish a stream of records to one or more Kafka topics is what we call Producer API.

**Q.20 Is Apache Kafka is a distributed streaming platform? if yes, what you can do with it?**

Ans. Undoubtedly, Kafka is a streaming platform. It can help:

1. To push records easily
2. Also, can store a lot of records without giving any storage problems
3. Moreover, it can process the records as they come in

### **4. Logical Kafka Interview Questions**

**Q. 21 What can you do with Kafka?**

**Ans.** It can perform in several ways, such as:  
>> In order to transmit data between two systems, we can build a real-time stream of data pipelines with it.  
>> Also, we can build a real-time streaming platform with Kafka, that can actually react to the data.

**Q.22 What is the purpose of retention period in Kafka cluster?**

**Ans.** However, retention period retains all the published records within the Kafka cluster. It doesn’t check whether they have been consumed or not. Moreover, the records can be discarded by using a configuration setting for the retention period. And, it results as it can free up some space.

**Q.23 Explain the maximum size of a message that can be received by the Kafka?**

**Ans.** The maximum size of a message that can be received by the Kafka is approx. 1000000 bytes.

**Q.24 What are the types of traditional method of message transfer?**

**Ans.** Basically, there are two methods of the traditional message transfer method, such as:

* **Queuing:** It is a method in which a pool of consumers may read a message from the server and each message goes to one of them.
* **Publish-Subscribe:** Whereas in Publish-Subscribe, messages are broadcasted to all consumers.

**Q.25 What does ISR stand in Kafka environment?**

**Ans.** ISR refers to In sync replicas. These are generally classified as a set of message replicas which are synced to be leaders.

**Q.26 What is Geo-Replication in Kafka?**

**Ans.** For our cluster, Kafka MirrorMaker offers geo-replication. Basically, messages are replicated across multiple data centers or cloud regions, with MirrorMaker. So, it can be used in active/passive scenarios for backup and recovery; or also to place data closer to our users, or support data locality requirements.

**Q.27 Explain Multi-tenancy?**

**Ans.** We can easily deploy Kafka as a multi-tenant solution. However, by configuring which topics can produce or consume data, Multi-tenancy is enabled. Also, it provides operations support for quotas.

**Q.28 What is the role of Consumer API?**

**Ans.** An API which permits an application to subscribe to one or more topics and also to process the stream of records produced to them is what we call Consumer API.

**Q.29 Explain the role of Streams API?**

**Ans.** An API which permits an application to act as a stream processor, and also consuming an input stream from one or more topics and producing an output stream to one or more output topics, moreover, transforming the input streams to output streams effectively, is what we call Streams API.

**Q.30 What is the role of Connector API?**

**Ans.** An API which permits to run as well as build the reusable producers or consumers which connect Kafka topics to existing applications or data systems is what we call the Connector API.

**Apache Kafka Interview Questions For Freshers.Q- 21, 23, 25, 26, 27, 28, 29, 30**

**Apache Kafka Interview Questions for Experience.Q- 24, 22**

**Q.31 Explain Producer?**

**Ans.** The main role of Producers is to publish data to the topics of their choice. Basically, its duty is to select the record to assign to partition within the topic.

**Q.34 Why Should we use Apache Kafka Cluster?**

**Ans.** In order to overcome the challenges of collecting the large volume of data, and analyzing the collected data we need a messaging system. Hence Apache Kafka came in the story. Its benefits are:

* It is possible to track web activities just by storing/sending the events for real-time processes.
* Through this, we can Alert as well as report the operational metrics.
* Also, we can transform data into the standard format.
* Moreover, it allows continuous processing of streaming data to the topics.

Due to its this wide use, it is ruling over some of the most popular applications like ActiveMQ, RabbitMQ, AWS etc.

**Q.35 Explain the term “Log Anatomy”.**

**Ans.** We view log as the partitions. Basically, 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. Here, below diagram shows a log is being written by the data source and the log is being read by consumers at different offsets.

**Q.36 What is Data Log in Kafka?**

**Ans.** As we know, messages are retained for a considerable amount of time in Kafka. Moreover, there is flexibility for consumers that they can read as per their convenience. Although, there is a possible case that if Kafka is configured to keep messages for 24 hours and possibly that time consumer is down for time greater than 24 hours, then the consumer may lose those messages. However, still, we can read those messages from last known offset, but only at a condition that the downtime on part of the consumer is just 60 minutes. Moreover, on what consumers are reading from a topic Kafka doesn’t keep state.

**Q.37 Explain how to Tune Kafka for Optimal Performance.**

**Ans.** So, ways to tune Apache Kafka it is to tune its several components:

1. Tuning Kafka Producers
2. Kafka Brokers Tuning
3. Tuning Kafka Consumers

<https://www.janbasktraining.com/blog/kafka-interview-questions-answers/>

#### 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 assigneda 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 stored 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.

#### 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.

#### 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.

<https://mindmajix.com/apache-kafka-interview-questions>

##### **Q1) Explain what is Kafka?**

Kafka is a publish-subscribe messaging application which is coded in “[Scala](https://mindmajix.com/apache-scala-tutorial)”. It is an open source message broker project which was started by the Apache software. The design pattern of Kafka is mainly based on the design of the transactional log.

##### **Q2) What are the different components that are available in Kafka?**

The different components that are available in Kafka are as follows:

1. **Topic:** this is nothing but a stream of messages that belong to the same type
2. **Producer:** this is used for publishing messages to a specific topic
3. **Brokers:** It is a set of servers which has the capability of storing publishers messages.

Consumer- responsible for subscribing to various topics and pulls the data from different brokers

##### **Q3) What is the role of offset in Kafka?**

Offset is nothing but a unique id that is assigned to the partitions. The messages are contained in this partitions. The important aspect or use of offset is that it identifies every message with the id which is available within the partition.

##### **Q4) What is a consumer group?**

A consumer group is nothing but an exclusive concept of Kafka.

Within each and every Kafka consumer group, we will have one or more consumers who actually consume subscribed topics.

##### **Q5) Explain the role of the zookeeper in Kafka?**

Within the Kafka environment, the zookeeper is used to store offset related information which is used to consume a specific topic and by a specific consumer group.

##### **Q6) Would it be possible to use Kafka without the zookeeper?**

No, it is not possible to use Kafka without the zookeeper. The user will not able to connect directly to the Kafka server in the absence of zookeeper. For some reason, if zookeeper is down then the individual will not able to access any of the client requests.

##### **Q7) Elaborate on the terms leader and follower in Kafka environment?**

The concept of leader and follower is maintained in Kafka environment so that the overall system ensures load balancing on the servers.

* For every partition in Kafka environment, one server plays the role as leader and rest of the servers act as followers.
* All the data read and write commands are executed at the leader level and the rest of the followers just have to replicate the process.
* At the time of any server faults and the leader is not able to function appropriately then one of the followers will take the place of the leaders. Thus making the system stable and also helps in load balancing of the server.

##### **Q8) What does ISR stand in Kafka environment?**

[ISR](https://en.wikipedia.org/wiki/Replication_(computing)) stands for In sync replicas.

They are classified as a set of message replicas which are synched to be leaders.

##### **Q9) What is the replica? What does it do?**

A replica can be defined as a list of essential nodes that are responsible to log for a particular partition, and it doesn't matter whether they actually play the role of a leader or not.

##### **Q10) Why are the replications are considered critical in Kafka environment?**

The main reason why replications are needed because they can be consumed again in an uncertain event of machine error or program malfunction or the system is down due to frequent software upgrades. So to make sure to overcome these, replication makes sure that the messages published are not lost.

##### **Q11) If the replica stays out of the ISR for a very long time, then what does it tell us?**

If the replica stays out of the ISR for a very long time, or replica is not in synch with the ISR then it means that the follower server is not able to grasp data as fast the leader is doing. So basically the follower is not able to come up with the leader activities.

##### **Q12) What is the process of starting a Kafka server?**

As the Kafka environment is run on zookeeper, one has to make sure to run zookeeper server first and then ignite Kafka server.

##### **Q13) Explain what is a partitioning key?**

Within the available producer, the main function of partitioning key is to validate and direct the destination partition of the message. Normally, a hashing based partitioner is used to assess the partition Id if the key is provided.

##### **Q14) Within the producer can you explain when will you experience QueueFullException occur?**

Well, if the producer is sending more messages to the broker and if it cannot handle this in the flow of the messages then we will experience QueueFullException.

The producers don't have any limitation so it doesn't know when to stop the overflow of the messages. So to overcome this problem one should add multiple brokers so that the flow of the messages can be handled perfectly and we won't fall into this exception again.

##### **Q15) Define the role of Kafka producer API?**

Kafka procedure API aims to do the producer functionality through one API call to the client.

In specific, Kafka producer API actually combines the efforts of kafka.producer.SyncProducer and the kafka.producer.async.Async Producer

##### **Q16) Explain the main difference between Kafka and Flume?**

Both Kafka and Flume are used for real-time processing where Kafka seems to be more scalable and you can trust on the message durability.

##### **Q17) Explain the Kafka architecture?**

Kafka is nothing but a cluster which holds multiple brokers as it is called as a distributed system.  
The topics within the system will hold multiple partitions.

Every broker within the system will hold multiple partitions. Based on this the producers and consumers actually exchange the message at the same time and the overall execution happens seamlessly.

##### **Q18) What are the advantages of Kafka technology?**

The following are the advantages of using Kafka technology:

1. It is fast
2. It comprises of brokers. Every single broker is capable of handling megabytes of data.
3. It is scalable
4. A large dataset can be easily analyzed
5. It is durable
6. It has a distributed design which is robust in nature

##### **Q19) Is apache Kafka is a distributed streaming platform? if yes, what you can do with it?**

Yes, Apache Kafka is a streaming platform. A streaming platform contains the vital three capabilities, they are as follows:

1. It will help you to push records easily
2. It will help you store a lot of records without giving any storage problems
3. It will help you to process the records as they come in

##### **Q20) What can you do with Kafka?**

With the help of Kafka technology we can do the below:

1. We can build a real-time stream of data pipelines which will help to transmit data between two systems
2. Build a real-time streaming platform which can actually react to the data

###### **Q21) What is the core API in Kafka?**

They are four main core API’s:

1. Producer API
2. Consumer API
3. Streams API
4. Connector API

All the communications between the clients happen over through high-performance language via TCP protocol.

###### **Q22) Explain the functionality of producer API in Kafka?**

The **producer API** is responsible where it will allow the application to push a stream of records to one of the Kafka topics.

###### **Q23) Explain the functionality of Consumer API in Kafka?**

The **Consumer API** is responsible where it allows the application to receive one or more topics and at the same time process the stream of data that is produced.

###### **Q24) Explain the functionality of Streams API in Kafka?**

The**Streams API** is responsible where it allows the application to act as a processor and within the process, it will be effectively transforming the input streams to output streams.

###### **Q25) Explain the functionality of the Connector API in Kafka?**

The **Connector API** is responsible where it allows the application to stay connected and keeping a track of all the changes that happen within the system. For this to happen, we will be using reusable producers and consumers which stays connected to the Kafka topics.

###### **Q26) Explain what is a topic?**

A topic is nothing but a category classification or it can be a feed name out of which the records are actually published. Topics are always classified, the multi-subscriber.

###### **Q27) What is the purpose of the retention period in the Kafka cluster?**

Within the Kafka [cluster](https://mindmajix.com/spark/exploring-apache-spark-standalone-cluster), it retains all the published records. It doesn’t check whether they have been consumed or not. Using a configuration setting for the retention period, the records can be discarded. The main reason to discard the records from the Kafka cluster is that it can free up some space.

###### **Q28) Highlights of Kafka system?**

1. It is dedicated to high performance
2. Low latency system
3. Scalable storage system

###### **Q29) What are the main components where the data is processed seamlessly in Kafka?**

The main components where the data is processed seamlessly is:

1. Producers
2. Consumers

<https://career.guru99.com/top-14-kafka-interview-questions/>

**6) Explain what is Zookeeper in Kafka? 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 request.

•  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.

**7) Explain how message is consumed by consumer 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 call between kernel user back to the kernel.

**8) Explain how you can improve the throughput of a remote consumer?**

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

**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

**10) Explain how you can reduce churn in ISR? 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.

**11) Why replication is required in Kafka?**

Replication of message 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.

**12) What does it indicate if 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.

**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.

**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 gaze 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.

<https://www.educba.com/kafka-interview-questions/>

#### Q1. What is Kafka and what are the various components of Kafka?

**Answer:**  
Kafka is said to be a pub-sub messaging model which was [developed using Scala](https://www.educba.com/uses-of-scala/). It is an open source application which was started by Apache software. Kafka is mainly designed upon transactional logs design. It has unique features which make it the best choice for data integration these days and is among the famous data processing tools. The important features are data partitioning, scalability, low-latency, high throughputs, stream processing, durability, zero data loss, etc. The main components of Kafka are:

* **Topic:** A bunch of messages which are of the same type come under the same topic.
* **Producer:** A producer as the name suggests, produces messages and can issue a communication to the selected topic.
* **Brokers:** These act as a channel between the producers and consumers. They are a set of servers where the published messages are stored.
* **Consumer:** Consumer is the one who is going to the consumer the published data. It can subscribe to different topics and then pull data from the brokers.

#### Q2. What is leader and follower in Kafka?

**Answer:**  
[Kafka creates partitions](https://www.educba.com/what-is-kafka/) based on offset and consumer groups. Every partition in Kafka has a server which plays the role of leader. One of them being the leader, there can be none or more servers which will act as a follower. The leader has assigned to itself tasks which read and write requests for partition. Followers, on the other hand, need to follow the leader and replicate what is being told by a leader. If at all the leader fails, like the real life one of the followers need to take over as the role of leader. This can happen at the time of server faults. This ensures that the load is balanced properly on the server and also ensures the system’s stability.

#### Q3. What is a replica? Why are the replications considered to be critical in Kafka environment?

**Answer:**  
A list of essential nodes which are responsible to log for any particular partition is known as a replica. A replica node does not matter whether it plays the role of leader or follower. The vital reason for the need of replication is that they can be consumed again in any uncertain event of machine error or program malfunction or system is down due to usual frequent updates. In order to make sure that no data is lost or corrupted replication makes sure that all messages are published properly and are not lost.

#### Q4. What is Zookeeper in Kafka? Can Kafka be used without Zookeeper?

**Answer:**  
This is the basic Kafka Interview Question asked in an interview. Zookeeper is used for distributed applications which are adapted by Kafka. It helps Kafka in managing all sources properly. Zookeeper is an open source, high performance and provides a complete coordination service.

No, it is impossible to skip the Zookeeper and go directly to the Kafka broker. Zookeeper manages all Kafka resources and hence if Zookeeper is down it cannot serve any client service requests. The main job of zookeeper is to be a channel of communication for the different nodes which are existing in a cluster. Zookeeper in Kafka is used to commit to the offset. If at all a node fails it can be easily retrieved from the offset which was previously committed. In addition to this zookeeper also takes care of activities like leader detection, distributed synchronization, configuration management, etc. With all of these, it also does the job of identifying the new node which leaves or joins the cluster nodes, the status of all nodes, etc.

#### Q5. How are the messages consumed by a consumer in Kafka?

**Answer:**  
By making use of send file API transfer of messages is done in Kafka. Using this file the transfer of bytes takes place from the socket to disk through the kernel space saving copies and the calls between kernel user and back to the kernel.

#### Q6. What is SerDes?

**Answer:**  
SerDes stands for serializer and deserializer. For any Kafka stream to materialize the data whenever necessary it is vital to provide SerDes for all data types or record and record values.

#### Q7. What is the way to send large messages with Kafka?

**Answer:**  
In order to send larges messages using Kafka, you must adjust a few properties. By making these changes you will not face any exceptions and will be able to send all messages successfully. Below are the properties which require a few changes:

At the Consumer end – fetch.message.max.bytes

At the Broker, end to create replica– replica.fetch.max.bytes

At the Broker, the end to create a message – message.max.bytes

At the Broker end for every topic – max.message.bytes

Let us move to the next Kafka Interview Questions

#### Q8. What is offset?

**Answer:**  
An offset can be called as a unique identifier which is assigned to all different partitions. These partitions contain messages. The most important use of offset is that it can help identify the messages through the offset id. These offset ids are available in all the partitions.

#### Q9. What is multi-tenancy?

**Answer:**  
This is the most asked Kafka Interview Questions in an interview. Kafka can be deployed easily as a multi-tenant solution. The configuration for different topics on which data is to be produced or consumed this feature is enabled. With all this, it also provides operational support for different quotas.

#### Q10. For its optimal performance, how will you tune Kafka?

**Answer:**  
There are different components which are present in Kafka. In order to tune Kafka, it is important to tune its components first. This includes tuning Kafka producers, Tuning Kafka consumers and also tuning the Kafka brokers.