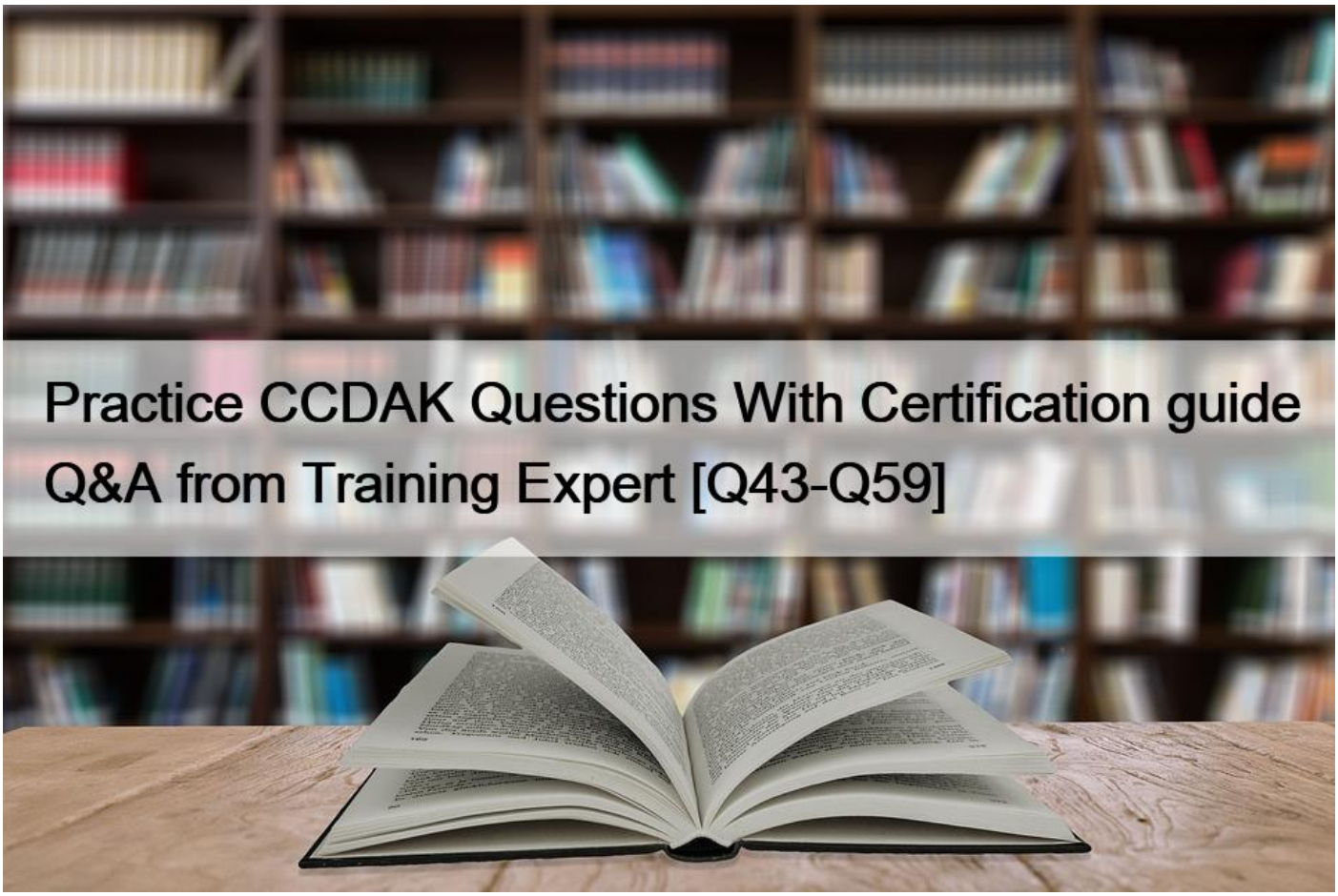
Practice CCDAK Questions With Certification guide Q&A from Training Expert [Q43-Q59



Practice CCDAK Questions With Certification guide Q&A from Training Expert ExamsLabs
Free Confluent CCDAK Test Practice Test Questions Exam Dumps

NO.43 A consumer starts and has auto.offset.reset=latest, and the topic partition currently has data for offsets going from 45 to 2311. The consumer group has committed the offset 643 for the topic before. Where will the consumer read from?
* it will crash
* offset 2311
* offset 643
* offset 45

The offsets are already committed for this consumer group and topic partition, so the property auto.offset.reset is ignored

NO.44 Compaction is enabled for a topic in Kafka by setting log.cleanup.policy=compact. What is true about log compaction?
* After cleanup, only one message per key is retained with the first value
* Each message stored in the topic is compressed
* Kafka automatically de-duplicates incoming messages based on key hashes
* After cleanup, only one message per key is retained with the latest value

Compaction changes the offset of messages

Explanation:

Log compaction retains at least the last known value for each record key for a single topic partition. All compacted log offsets remain valid, even if record at offset has been compacted away as a consumer will get the next highest offset.

NO.45 You are using JDBC source connector to copy data from a table to Kafka topic. There is one connector created with max.tasks equal to 2 deployed on a cluster of 3 workers. How many tasks are launched?
* 3
* 2
* 1
* 6

JDBC connector allows one task per table.

NO.46 Once sent to a topic, a message can be modified
* No
* Yes

Kafka logs are append-only and the data is immutable

NO.47 Which of the following is not an Avro primitive type?
* string
* long
* int
* date
* null

date is a logical type

NO.48 A Kafka producer application wants to send log messages to a topic that does not include any key. What are the properties that are mandatory to configure for the producer configuration? (select three)
* bootstrap.servers
* partition
* key.serializer
* value.serializer
* key
* value

Both key and value serializer are mandatory.

NO.49 Which of the following event processing application is stateless? (select two)
* Read events from a stream and modifies them from JSON to Avro
* Publish the top 10 stocks each day
* Read log messages from a stream and writes ERROR events into a high-priority stream and the rest of the events into a low-priority stream
* Find the minimum and maximum stock prices for each day of trading

Stateless means processing of each message depends only on the message, so converting from JSON to Avro or filtering a stream are both stateless operations

NO.50 We would like to be in an at-most once consuming scenario. Which offset commit strategy would you recommend?
* Commit the offsets on disk, after processing the data
* Do not commit any offsets and read from beginning
* Commit the offsets in Kafka, after processing the data

* Commit the offsets in Kafka, before processing the data
Here, we must commit the offsets right after receiving a batch from a call to .poll()

NO.51 You are using JDBC source connector to copy data from 2 tables to two Kafka topics. There is one connector created with max.tasks equal to 2 deployed on a cluster of 3 workers. How many tasks are launched?
* 6
* 1
* 2
* 3
we have two tables, so the max number of tasks is 2

NO.52 Suppose you have 6 brokers and you decide to create a topic with 10 partitions and a replication factor of 3. The brokers 0 and 1 are on rack A, the brokers 2 and 3 are on rack B, and the brokers 4 and 5 are on rack C.

If the leader for partition 0 is on broker 4, and the first replica is on broker 2, which broker can host the last replica? (select two)
* 6
* 1
* 2
* 5
* 0
* 3
When you create a new topic, partitions replicas are spreads across racks to maintain availability. Hence, the Rack A, which currently does not hold the topic partition, will be selected for the last replica

NO.53 How often is log compaction evaluated?
* Every time a new partition is created
* Every time a segment is closed
* Every time a message is sent to Kafka
* Every time a message is flushed to disk
Log compaction is evaluated every time a segment is closed. It will be triggered if enough data is “dirty” (see dirty ratio config)

NO.54 In Kafka Streams, by what value are internal topics prefixed by?
* tasks-<number>
* application.id
* group.id
* kafka-streams-
In Kafka Streams, the application.id is also the underlying group.id for your consumers, and the prefix for all internal topics (repartition and state)

NO.55 When auto.create.topics.enable is set to true in Kafka configuration, what are the circumstances under which a Kafka broker automatically creates a topic? (select three)
* Client requests metadata for a topic
* Consumer reads message from a topic
* Client alters number of partitions of a topic
* Producer sends message to a topic
A kafka broker automatically creates a topic under the following circumstances- When a producer starts writing messages to the topic – When a consumer starts reading messages from the topic – When any client requests metadata for the topic

NO.56 Where are the dynamic configurations for a topic stored?

* In Zookeeper
* In an internal Kafka topic __topic_configuratins
* In server.properties
* On the Kafka broker file system

Dynamic topic configurations are maintained in Zookeeper.

NO.57 A topic receives all the orders for the products that are available on a commerce site. Two applications want to process all the messages independently – order fulfilment and monitoring. The topic has 4 partitions, how would you organise the consumers for optimal performance and resource usage?

* Create 8 consumers in the same group with 4 consumers for each application
* Create two consumers groups for two applications with 8 consumers in each
* Create two consumer groups for two applications with 4 consumers in each
* Create four consumers in the same group, one for each partition – two for fulfilment and two for monitoring

two partitions groups – one for each application so that all messages are delivered to both the application. 4 consumers in each as there are 4 partitions of the topic, and you cannot have more consumers per groups than the number of partitions (otherwise they will be inactive and wasting resources)

NO.58 A consumer starts and has auto.offset.reset=none, and the topic partition currently has data for offsets going from 45 to 2311. The consumer group has committed the offset 10 for the topic before. Where will the consumer read from?

* offset 45
* offset 10
* it will crash
* offset 2311

auto.offset.reset=none means that the consumer will crash if the offsets it’s recovering from have been deleted from Kafka, which is the case here, as 10 < 45

NO.59 How will you find out all the partitions where one or more of the replicas for the partition are not in-sync with the leader?

* kafka-topics.sh –bootstrap-server localhost:9092 –describe –unavailable- partitions
* kafka-topics.sh –zookeeper localhost:2181 –describe –unavailable- partitions
* kafka-topics.sh –broker-list localhost:9092 –describe –under-replicated-partitions
* kafka-topics.sh –zookeeper localhost:2181 –describe –under-replicated-partitions