CCDAK$^{\text{Q\&As}}$

Confluent Certified Developer for Apache Kafka Certification Examination

Pass Confluent CCDAK Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/ccdak.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Confluent Official Exam Center

QUESTION 1

You want to send a message of size 3 MB to a topic with default message size configuration. How does KafkaProducer handle large messages?

A. KafkaProducer divides messages into sizes of max.request.size and sends them in order

B. KafkaProducer divides messages into sizes of message.max.bytes and sends them in order

C. MessageSizeTooLarge exception will be thrown, KafkaProducer will not retry and return exception immediately

D. MessageSizeTooLarge exception will be thrown, KafkaProducer retries until the number of retries are exhausted

Correct Answer: C

MessageSizeTooLarge is not a retryable exception.

QUESTION 2

A kafka topic has a replication factor of 3 and min.insync.replicas setting of 1. What is the maximum number of brokers that can be down so that a producer with acks=all can still produce to the topic?

A. 3

B. 0

C. 2

D. 1

Correct Answer: C

Two brokers can go down, and one replica will still be able to receive and serve data

QUESTION 3

In Kafka, every broker... (select three)

A. contains all the topics and all the partitions

B. knows all the metadata for all topics and partitions

C. is a controller

D. knows the metadata for the topics and partitions it has on its disk

E. is a bootstrap broker

F. contains only a subset of the topics and the partitions

Correct Answer: BEF

Kafka topics are divided into partitions and spread across brokers. Each brokers knows about all the metadata and each broker is a bootstrap broker, but only one of them is elected controller

QUESTION 4

```
StreamsBuilder builder = new StreamsBuilder();

KStream textLines = builder.stream("word-count-input");

KTable wordCounts = textLines

.mapValues(textLine -> textLine.toLowerCase())

.flatMapValues(textLine -> Arrays.asList(textLine.split("\W+"))) .selectKey((key, word) -> word)

.groupByKey()

.count(Materialized.as("Counts"));

wordCounts.toStream().to("word-count-output", Produced.with(Serdes.String(), Serdes.Long()));

builder.build();
```

What is an adequate topic configuration for the topic word-count-output?

A. max.message.bytes=10000000

B. cleanup.policy=delete

C. compression.type=lz4

D. cleanup.policy=compact

Correct Answer: D

Result is aggregated into a table with key as the unique word and value its frequency. We have to enable log compaction for this topic to align the topic\\'s cleanup policy with KTable semantics.

QUESTION 5

You have a Kafka cluster and all the topics have a replication factor of 3. One intern at your company stopped a broker, and accidentally deleted all the data of that broker on the disk. What will happen if the broker is restarted?

A. The broker will start, and other topics will also be deleted as the broker data on the disk got deleted

B. The broker will start, and won\\'t be online until all the data it needs to have is replicated from other leaders

C. The broker will crash

D. The broker will start, and won\\'t have any data. If the broker comes leader, we have a data loss

Correct Answer: B

Kafka replication mechanism makes it resilient to the scenarios where the broker lose data on disk, but can recover from replicating from other brokers. This makes Kafka amazing!