

CCDAK Exam Dumps Pdf - Discount CCDAK Code



BTW, DOWNLOAD part of VCE4Dumps CCDAK dumps from Cloud Storage: https://drive.google.com/open?id=1Zjbjq7EuYkLWcO4CsCE_1EEW9hTr-NC9

Will you feel nervous in the exam? If you do, just choose us, our CCDAK Soft test engine can stimulate the real exam environment, which will help you know the procedure of the exam, and will strengthen your confidence. Moreover CCDAK exam dumps are high-quality, and we have professional experts to compile them, and they can help you pass the exam just one time. We offer you free demo to have a try for CCDAK Exam Dumps, and free update for one year. If you indeed have questions, just contact with us.

The CCDAK certification exam is a valuable credential for developers who want to demonstrate their expertise in Kafka. By passing the exam, developers can show that they have a deep understanding of Kafka and its ecosystem, including its core components and the tools and frameworks used to build Kafka applications. Confluent Certified Developer for Apache Kafka Certification Examination certification also demonstrates the ability to design, develop, and deploy Kafka solutions to meet real-world business requirements.

The CCDAK exam is a comprehensive assessment of a developer's knowledge of the Kafka ecosystem and its various components, such as Kafka Connect, Kafka Streams, and Schema Registry. CCDAK exam covers a broad range of topics, including Kafka architecture, data modeling, event-driven design, and security. Developers who pass the CCDAK exam will have demonstrated their ability to develop Kafka-based applications that are scalable, reliable, and secure, and they will be recognized as experts in the Kafka ecosystem. The CCDAK Certification is an essential credential for developers who wish to work with Kafka and Confluent's platform and is highly valued by organizations that use Kafka as their messaging and streaming platform.

Confluent Certified Developer for Apache Kafka (CCDAK) certification exam is intended for developers who want to demonstrate their expertise in building and implementing applications that use Apache Kafka. CCDAK exam is designed to test the developer's knowledge of Kafka's key concepts, architecture, and APIs. The CCDAK certification is a valuable credential that can help developers stand out in the job market and demonstrate their ability to develop robust and reliable Kafka applications.

[**>> CCDAK Exam Dumps Pdf <<**](#)

CCDAK Valid Study Material & CCDAK Test Training Pdf & CCDAK Latest Pep Demo

Preparing for the exam may be not an easy thing for some candidates, if you choose us, we will do the things for you, what you need to do is practicing. We offer you free demo for CCDAK training materials, you can have a try before buying. And you will receive the downloading link and password within ten minutes after purchasing the CCDAK Exam Dumps. In addition, we have after-service stuff to resolve the confusions you have. If you fail to pass the exam, we are money back guaranteed, or if you have other exam to attend, we can also replace other 2 valid exam dumps for you.

Confluent Certified Developer for Apache Kafka Certification Examination Sample Questions (Q43-Q48):

NEW QUESTION # 43

What happens if you write the following code in your producer? producer.send(producerRecord).get()

- A. It will force all brokers in Kafka to acknowledge the producerRecord
- B. Throughput will be decreased
- C. Compression will be increased
- D. Batching will be increased

Answer: B

Explanation:

Using Future.get() to wait for a reply from Kafka will limit throughput.

NEW QUESTION # 44

When is the onCompletion() method called?

```
private class ProducerCallback implements Callback {  
    @Override  
    public void onCompletion(RecordMetadata recordMetadata, Exception e) {  
        if(e != null) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
ProducerRecord<String, String> record =  
new ProducerRecord<"topic1", "key1", "value1">;  
producer.send(record, new ProducerCallback());
```

- A. When the message is partitioned and batched successfully
- B. When message is serialized successfully
- C. When send() method is called
- D. When the broker response is received

Answer: D

Explanation:

Callback is invoked when a broker response is received.

NEW QUESTION # 45

You are writing to a topic with acks=all.

The producer receives acknowledgments but you notice duplicate messages.

You find that timeouts due to network delay are causing resends.

Which configuration should you use to prevent duplicates?

- A. retries=2147483647
max.in.flight.requests.per.connection=5
enable.idempotence=true
- B. enable.auto.commit=true
- C. retries=0
max.in.flight.requests.per.connection=5
enable.idempotence=true
- D. retries=2147483647
max.in.flight.requests.per.connection=1
enable.idempotence=false

Answer: A

Explanation:

To ensure exactly-once delivery and avoid duplicates even during retries:

- * `enable.idempotence=true` ensures deduplication on the broker
- * `retries=2147483647` allows unlimited retries on retriable errors
- * `max.in.flight.requests.per.connection=5` is the maximum value that preserves message order with idempotence. From Kafka Producer Config Docs:

"To achieve exactly-once semantics, set `enable.idempotence=true`, and `max.in.flight.requests.per.connection=5`."

- * A is unrelated (consumer-side)

- * C disables retries

- * D disables idempotence, leading to duplicates

Reference: Kafka Producer Configs > `enable.idempotence, retries`

NEW QUESTION # 46

Your Kafka cluster has five brokers. The topic t1 on the cluster has:

- * Two partitions
- * Replication factor = 4
- * `min.insync.replicas = 3` You need strong durability guarantees for messages written to topic t1. You configure a producer `acks=all` and all the replicas for t1 are in-sync. How many brokers need to acknowledge a message before it is considered committed?

- A. 0
- B. 1
- C. 2
- D. 3

Answer: A

Explanation:

With `acks=all`, the leader waits for `min.insync.replicas` to acknowledge the message. Since `min.insync.replicas=3`, Kafka will only commit the message once 3 brokers (leader + 2 followers) confirm they have the message.

From Kafka Documentation > Acks and Durability:

"If `acks=all` is specified, the producer will wait until the full set of in-sync replicas has acknowledged the record. The minimum number of in-sync replicas is controlled by `min.insync.replicas`." Even though the replication factor is 4, only 3 acknowledgments are needed, as defined by `min.insync.replicas`.

Reference: Apache Kafka Producer Configs > `acks, min.insync.replicas`

NEW QUESTION # 47

You are using JDBC source connector to copy data from 3 tables to three 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?

- A. 0
- B. 1
- C. 2
- D. 3

Answer: C

Explanation:

here, we have three tables, but the `max.tasks` is 2, so that's the maximum number of tasks that will be created

NEW QUESTION # 48

.....

Our CCDAK exam materials are the most reliable products for customers. If you need to prepare an exam, we hope that you can choose our CCDAK study guide as your top choice. In the past ten years, we have overcome many difficulties and never give up. And we have quickly grown up as the most influential company in the market. And our CCDAK preparation questions are the most

popular among the candidates.

Discount CCDAK Code: <https://www.vce4dumps.com/CCDAK-valid-torrent.html>

P.S. Free & New CCDAK dumps are available on Google Drive shared by VCE4Dumps: https://drive.google.com/open?id=1Zjbjq7EuYkLWcO4CsCE_1EEW9hTr-NC9