

# Free PDF Quiz 2026 High-quality Confluent CCDAK: Confluent Certified Developer for Apache Kafka Certification Examination Valid Test Tutorial



What's more, part of that DumpsTests CCDAK dumps now are free: [https://drive.google.com/open?id=1sEd7eiGMx8l0Wzzl8\\_jomjlCNi3DXH](https://drive.google.com/open?id=1sEd7eiGMx8l0Wzzl8_jomjlCNi3DXH)

The students can give unlimited to track the performance of their last given tests in order to see their mistakes and try to avoid them while giving the final test. Customers of DumpsTests will receive updates till 1 year after their purchase. Anyone can try a free demo of the Confluent Certified Developer for Apache Kafka Certification Examination (CCDAK) practice material before making purchase. There is a 24/7 available support system that assists users whenever they are stuck in any problem or issues. This product is a complete package and a blessing for those who want to pass the Confluent CCDAK test in a single try.

The CCDAK Certification Exam is a hands-on exam that requires developers to demonstrate their skills in a real-world scenario. CCDAK exam consists of a set of tasks that must be completed within a set time frame. The tasks are designed to test a developer's ability to work with Kafka and Confluent's tools to build and manage Kafka-based applications. CCDAK exam is graded based on the number of tasks completed and the quality of the solutions provided.

Confluent CCDAK (Confluent Certified Developer for Apache Kafka Certification Examination) is a certification exam that validates a developer's knowledge and skills in building applications and solutions using Apache Kafka. CCDAK Exam is designed for developers who have experience in developing Kafka-based applications and want to demonstrate their proficiency in this technology. Confluent Certified Developer for Apache Kafka Certification Examination certification helps developers prove their expertise in Apache Kafka, which is a popular, open-source distributed streaming platform that is widely used in real-time data processing.

**>> CCDAK Valid Test Tutorial <<**

## Confluent CCDAK Exam Dumps with Guaranteed Success Result [2026]

Perhaps you still feel confused about our Confluent Certified Developer for Apache Kafka Certification Examination test questions when you browse our webpage. There must be many details about our products you would like to know. Do not hesitate and send us an email. Gradually, the report will be better as you spend more time on our CCDAK exam questions. As you can see, our system is so powerful and intelligent. What most important it that all knowledge has been simplified by our experts to meet all people's demands. So the understanding of the CCDAK Test Guide is very easy for you. Our products know you better.

The CCDAK Certification is an excellent way to demonstrate to potential employers that one has the skills and knowledge required

to work with the Kafka platform. It is also an excellent way to stay up-to-date with the latest developments in the Kafka community. Confluent Certified Developer for Apache Kafka Certification Examination certification is highly respected in the industry and is recognized by leading organizations worldwide.

## Confluent Certified Developer for Apache Kafka Certification Examination Sample Questions (Q57-Q62):

### NEW QUESTION # 57

Which configuration allows more time for the consumer poll to process records?

- A. **max.poll.interval.ms**
- B. session.timeout.ms
- C. fetch.max.wait.ms
- D. heartbeat.interval.ms

#### Answer: A

Explanation:

max.poll.interval.ms defines the maximum delay between invocations of poll() before the consumer is considered failed. It essentially gives consumers more time to process records before needing to poll again.

From Kafka Consumer Configuration Reference:

"max.poll.interval.ms: The maximum delay between invocations of poll() when using consumer group management. If poll() is not called before expiration, the consumer is considered failed."

\* session.timeout.ms and heartbeat.interval.ms relate to group coordination and heartbeats.

\* fetch.max.wait.ms affects how long the broker waits to accumulate data before sending a fetch response.

Reference: Apache Kafka Consumer Configs

### NEW QUESTION # 58

You are working on a Kafka cluster with three nodes. You create a topic named orders with:

replication.factor = 3

min.insync.replicas = 2

acks = all What exception will be generated if two brokers are down due to network delay?

- A. NetworkException
- **B. NotEnoughReplicasException**
- C. NotCoordinatorException
- D. NotLeaderForPartitionException

#### Answer: B

Explanation:

With acks=all and min.insync.replicas=2, Kafka requires at least two in-sync replicas to acknowledge a write.

If only one broker is alive, the condition fails, and NotEnoughReplicasException is thrown by the producer.

From Kafka Producer Exception Docs:

"NotEnoughReplicasException is thrown when the number of in-sync replicas is insufficient to satisfy acks=all with min.insync.replicas." NetworkException is generic and not raised here.

NotCoordinatorException is related to consumer group coordination.

NotLeaderForPartitionException is unrelated unless accessing an unassigned leader.

Reference: Kafka Producer Error Handling

### NEW QUESTION # 59

You have a consumer group with default configuration settings reading messages from your Kafka cluster.

You need to optimize throughput so the consumer group processes more messages in the same amount of time.

Which change should you make?

- A. Decrease the session timeout of each consumer.
- **B. Increase the number of bytes the consumers read with each fetch request.**
- C. Disable auto commit and have the consumers manually commit offsets.

- D. Remove some consumers from the consumer group.

**Answer: B**

Explanation:

To increase consumer throughput, one effective strategy is to increase the amount of data fetched in each poll by raising `fetch.max.bytes` or `max.partition.fetch.bytes`. This allows each poll to retrieve more records per request, improving processing efficiency.

From Kafka Consumer Config Docs:

"Increasing fetch size allows consumers to retrieve larger batches of messages, improving throughput and reducing request overhead."

- \* Removing consumers (A) may reduce parallelism.
- \* Manual commit (C) adds complexity, not throughput.
- \* Decreasing session timeout (D) risks unnecessary rebalances.

Reference: Kafka Consumer Configuration > `fetch.max.bytes`

**NEW QUESTION # 60**

You need to configure a sink connector to write records that fail into a dead letter queue topic. Requirements:

Topic name: DLQ-Topic

Headers containing error context must be added to the messages. Which three configuration parameters are necessary? (Select three.)

- A. `errors.tolerance=none`
- B. `errors.deadletterqueue.context.headers.enable=true`
- C. `errors.log.include.messages=true`
- D. `errors.tolerance=all`
- E. `errors.deadletterqueue.topic.name=DLQ-Topic`
- F. `errors.log.enable=true`

**Answer: B,D,E**

Explanation:

To send failed records to a dead letter queue (DLQ), you must configure:

`errors.tolerance=all`: Tells the connector to not fail on errors but handle them (e.g., send to DLQ).

`errors.deadletterqueue.topic.name=DLQ-Topic`: Specifies the DLQ topic.

`errors.deadletterqueue.context.headers.enable=true`: Includes error context in message headers.

From Kafka Connect Error Handling Docs:

"Kafka Connect supports directing problematic records to a separate topic (DLQ) using `errors.*` configs.

Headers can include failure metadata."

Options D, E, F are related to logging, not DLQ behavior.

Reference: Kafka Connect Configurations > Error Handling

**NEW QUESTION # 61**

What is the risk of increasing `max.in.flight.requests.per.connection` while also enabling retries in a producer?

- A. Reduce throughput
- B. Message order not preserved
- C. At least once delivery is not guaranteed
- D. Less resilient

**Answer: B**

Explanation:

Some messages may require multiple retries. If there are more than 1 requests in flight, it may result in messages received out of order. Note an exception to this rule is if you enable the producer setting `enable.idempotence=true`.

`enable.idempotence=true` which takes care of the out of ordering case on its own. See <https://issues.apache.org/jira/browse/KAFKA-5494>

**NEW QUESTION # 62**

**CCDAK Torrent:** <https://www.dumpstests.com/CCDAK-latest-test-dumps.html>

What's more, part of that DumpsTests CCDAK dumps now are free: [https://drive.google.com/open?id=1sEd7eiGMx8l0Wzzl8\\_jomijlCNi3DXH](https://drive.google.com/open?id=1sEd7eiGMx8l0Wzzl8_jomijlCNi3DXH)