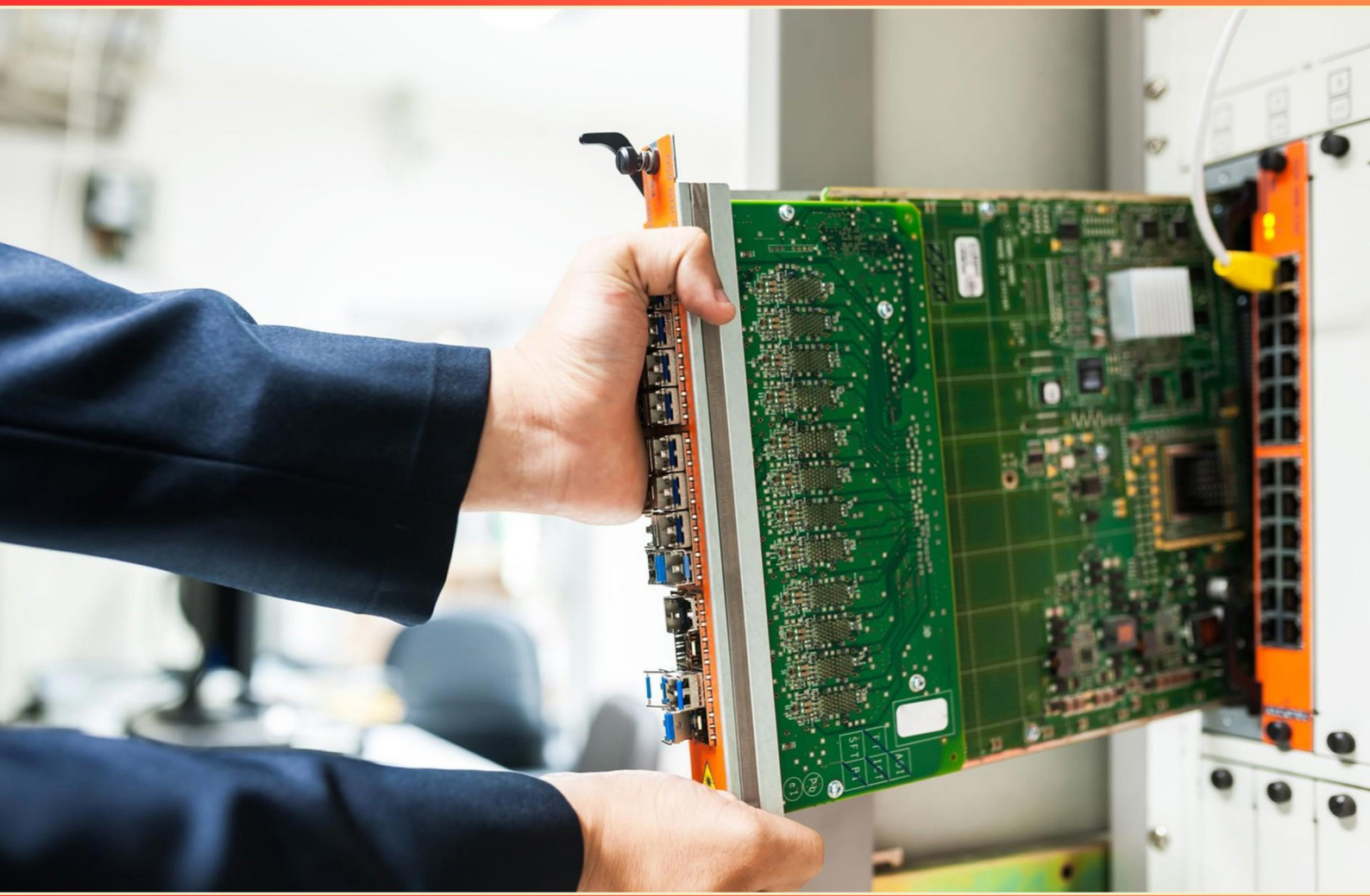
CCDAK APACHE KAFKA PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://ccdakapachekafka.examzify.com

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright 1
Table of Contents 2
Introduction 3
How to Use This Guide 4
Questions 5
Answers 8
Explanations 10
Next Steps 15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

SAMPLE

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

SAMPLE

1. Does increasing the number of partitions affect the guarantee that all messages with the same key go to the same partition?

A. No

B. Yes

C. Only if keys are hashed

D. It never affects

2. Which of the following is NOT a stateless operator in Kafka Streams?

A. foreach

B. map

C. join

D. branch

3. Which statement best describes the relationship between backward, forward, and full compatibility in schema evolution?

A. BACKWARD means newer schema can read data written with the old schema; FORWARD means older schema can read data written with the newer schema; FULL means both.

B. BACKWARD means newer schema can read data written with the old schema; FORWARD means older schema can read data written with the newer schema; FULL means both.

C. BACKWARD means no compatibility checks; FORWARD means strict checks; FULL means some checks.

D. BACKWARD means older schema can read newer data; FORWARD means newer can read older data.

4. Which statement about the Consumer assign() method is correct?

A. It commits offsets automatically

B. It subscribes to partitions

C. It closes the consumer

D. It creates a new consumer group

5. Which command resets a consumer group's committed offset to an earlier point?

A. Restart the broker

B. Delete the consumer group

C. kafka-consumer-groups --reset-offsets

D. Change the topic replication factor

SAMPLE

6. Which command lists all consumer groups?

A. bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list

B. bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe

C. bin/kafka-consumer-groups.sh --bootstrap-server localhost:2181 --list

D. bin/kafka-consumer-groups.sh --zookeeper localhost:2181 --list

7. Which window type is time-based, fixed size, non-overlapping, and gapless?

A. Hopping Window

B. Sliding Window

C. Session Window

D. Tumbling Window

8. The KStream abstraction represents a record stream from a subset of partitions of a topic.

A. True

B. False

C. It represents the entire topic

D. It represents a stream from all partitions

9. What is the default Confluent Schema Registry compatibility type?

A. None

B. Forward

C. Full

D. Backward

10. Which AdminClient operation would you use to create a topic with 3 partitions and replication factor 2, and what code would look like?

A. Use AdminClient.deleteTopics to create.

B. Use AdminClient.createTopics with a NewTopic("topic-name", 3, 2). Then call topics to create; handle exceptions.

C. Use AdminClient.createPartitions with a topic name.

D. Use AdminClient.createTopics with a NewTopic("topic-name", 2, 3).

SAMPLE

Answers

SAMPLE

1. B
2. C
3. B
4. B
5. C
6. A
7. D
8. A
9. D
10. B

SAMPLE

Explanations

SAMPLE

1. Does increasing the number of partitions affect the guarantee that all messages with the same key go to the same partition?

A. No

B. Yes

C. Only if keys are hashed

D. It never affects

Partitioning in Kafka uses a hash of the key modulo the number of partitions to decide the target partition. When you increase the number of partitions, the modulo base changes, so the partition chosen for a given key can change. As a result, messages with the same key sent after the change may land in a different partition than earlier messages with that key. The guarantee holds only while the partition count stays fixed; once you add partitions, that cross-time guarantee is no longer preserved, though within a single fixed set of partitions the mapping remains deterministic and the same key will always go to the same partition.

2. Which of the following is NOT a stateless operator in Kafka Streams?

A. foreach

B. map

C. join

D. branch

Kafka Streams differentiates stateless and stateful operators. Stateless operators process each record independently and don't rely on previously seen data. Map applies a function to each record and emits a transformed one; foreach performs side effects per record without storing state; branch routes each input to different outputs based on a predicate and doesn't need to remember past events. Join, however, is stateful because producing joined results requires keeping past records from one or both streams (often within a window) in a state store to be matched together. So the operator that is not stateless is join.

SAMPLE

3. Which statement best describes the relationship between backward, forward, and full compatibility in schema evolution?

A. BACKWARD means newer schema can read data written with the old schema; FORWARD means older schema can read data written with the newer schema; FULL means both.

B. BACKWARD means newer schema can read data written with the old schema; FORWARD means older schema can read data written with the newer schema; FULL means both.

C. BACKWARD means no compatibility checks; FORWARD means strict checks; FULL means some checks.

D. BACKWARD means older schema can read newer data; FORWARD means newer can read older data.

Compatibility in schema evolution describes whether a consumer and producer using different versions of a schema can still interoperate. Backward compatibility means the newer schema can read data written with the older schema. This lets you upgrade the reader to a newer version while still being able to access existing, older data. Forward compatibility means the older schema can read data written with the newer schema. This is typically achieved when new fields are optional or have safe defaults, so old readers can ignore the added information. Full compatibility means both directions work: data written with either version can be read by a reader using either version. This mapping is what the statement expresses: the newer schema reading old data (backward), the older schema reading newer data (forward), and both directions available for full compatibility.

4. Which statement about the Consumer assign() method is correct?

A. It commits offsets automatically

B. It subscribes to partitions

C. It closes the consumer

D. It creates a new consumer group

Assign() is about fixing which partitions a consumer will read from. In Kafka, there are two ways to tell a consumer what data to process: subscribing to topics, which uses the consumer group protocol to automatically assign partitions, and manually assigning partitions with assign(). When you call assign() and pass a set of TopicPartition objects, you bind this consumer to those specific partitions and it will only consume from them. This bypasses the dynamic partition rebalancing that happens with subscribe(). It's not about committing offsets automatically, closing the consumer, or creating a new consumer group—the latter two are done by other methods/properties. Therefore, the statement that best describes assign() is that it assigns partitions for the consumer to read from.

SAMPLE

5. Which command resets a consumer group's committed offset to an earlier point?

A. Restart the broker

B. Delete the consumer group

C. kafka-consumer-groups --reset-offsets

D. Change the topic replication factor

Resetting a consumer group's committed offset to an earlier point is done with the kafka-consumer-groups tool by using the --reset-offsets option. This command targets a specific group (and topic, if you want to narrow it) and moves the committed position backward, such as to the earliest available offset or to a specific offset you choose. In practice you'd run something like --reset-offsets --to-earliest (often paired with --execute to apply the change, or --dry-run to preview). Once applied, the next time the group's consumers read messages, they will start from that earlier point, effectively reprocessing messages from there. Other approaches don't accomplish this targeted offset movement: restarting the broker doesn't change what offsets are committed for a group; deleting the consumer group removes its metadata and offsets rather than repositioning them; and changing the topic replication factor affects storage and replication, not the consumer's committed offsets.

6. Which command lists all consumer groups?

A. bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --list

B. bin/kafka-consumer-groups.sh --bootstrap-server localhost:9092 --describe

C. bin/kafka-consumer-groups.sh --bootstrap-server localhost:2181 --list

D. bin/kafka-consumer-groups.sh --zookeeper localhost:2181 --list

Listing all consumer groups is done by querying the cluster through the broker and asking the tool to return just the group IDs. Use the kafka-consumer-groups.sh script with the broker connection flag and the list option. Point it at a running broker (the common port is 9092) and request a simple list. This returns all current consumer group IDs registered in the cluster. Connecting to the Zookeeper port (2181) or using the Zookeeper option is not correct for this operation because modern tooling communicates with the broker API, not Zookeeper. If you used describe, you'd get detailed information about each group (members, state, offsets) rather than a simple list. The Zookeeper-based approach is deprecated in newer Kafka versions. So the best command is the one that uses --bootstrap-server with a broker address and --list.

7. Which window type is time-based, fixed size, non-overlapping, and gapless?

A. Hopping Window

B. Sliding Window

C. Session Window

D. Tumbling Window

Windowing in streaming systems groups events into fixed time blocks. Tumbling windows are defined with a fixed duration and align to start times, producing consecutive blocks that do not overlap and leave no gaps between them. This makes them time-based with a fixed size, and ensures each moment in time belongs to exactly one window, giving gapless coverage. Other options behave differently: hopping windows use a window length with a stride that can create overlaps; sliding windows continuously move the window forward, causing the same events to appear in multiple windows; session windows are driven by inactivity gaps and can vary in length, so they're not fixed-size or gapless.

SAMPLE

8. The KStream abstraction represents a record stream from a subset of partitions of a topic.

A. True

B. False

C. It represents the entire topic

D. It represents a stream from all partitions

Kafka Streams models a stream as a sequence of records from topic partitions, and in a distributed setup each processing task handles a subset of partitions. This means a KStream represents the records from a topic, but the data you process within a given application instance comes from only the partitions assigned to that instance. So, the stream you work with is effectively from a subset of partitions of the topic. If a topic has just one partition, that subset is the whole topic, but with multiple partitions the workload is split across instances. This is why describing the KStream as a stream from a subset of partitions is the accurate view.

9. What is the default Confluent Schema Registry compatibility type?

A. None

B. Forward

C. Full

D. Backward

Schema compatibility in Confluent Schema Registry determines how schemas can evolve without breaking readers or writers. The default setting is backward. This means that when you register a new version of a schema, it must be able to deserialize data written with the previous version, so newer readers can still read data produced by older schemas. In practice, adding an optional field with a default value preserves backward compatibility, because old data can be read by the new schema without issues. If you remove a field or change its type without a default, you would break backward compatibility. The other options describe different directions of compatibility (forward or full) or no guarantees, but the default is backward.

10. Which AdminClient operation would you use to create a topic with 3 partitions and replication factor 2, and what code would look like?

A. Use AdminClient.deleteTopics to create.

B. Use AdminClient.createTopics with a NewTopic("topic-name", 3, 2). Then call topics to create; handle exceptions.

C. Use AdminClient.createPartitions with a topic name.

D. Use AdminClient.createTopics with a NewTopic("topic-name", 2, 3).

Creating a topic with the AdminClient is done with createTopics and a NewTopic that specifies both the number of partitions and the replication factor. For this scenario you want 3 partitions and a replication factor of 2, so you construct a NewTopic with those values and pass it to createTopics. The replication factor must be provided as a short, so you'd use (short) 2. After invoking createTopics, you typically wait for the operation to complete (for example, result.all().get()) and handle any exceptions to catch issues like the topic already existing or connectivity problems. Code outline: AdminClient adminClient = AdminClient.create(properties); NewTopic newTopic = new NewTopic("topic-name", 3, (short) 2); CreateTopicsResult result = adminClient.createTopics(Collections.singleton(newTopic)); try { result.all().get(); } catch (Exception e) { // handle error } Why this fits best: this approach directly creates a new topic with the exact desired configuration in a single API call, which is what topic creation requires. The other options don't create a new topic with the specified partitions and replication factor: deleting topics is for removal, creating partitions is for adjusting an existing topic, and the last option swaps the numbers (2 partitions with a replication factor of 3), which doesn't meet the requirement.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://ccdakapachekafka.examzify.com

We wish you the very best on your exam journey. You've got this!

SAMPLE