

CertKillers

Guaranteed Success with Accurate & Updated Questions.

Confluent

CCOAK

CCOAK Confluent Certified Operator for Apache Kafka

Questions & Answers PDF

For More Information - Visit:
<https://www.certkillers.net/>

Latest Version: 6.0

Question: 1

Which of the following are functions of the Controller in Apache Kafka? (select three)

- A. Sending updates to Kafka brokers after the elections
- B. Sending updates to Zookeeper after the elections
- C. Perform administrative tasks such as reassigning partitions
- D. Sending updates to producers and consumers after the elections
- E. Elect which partitions are leader and replicas

Answer: A,C,E

Explanation:

In a Kafka cluster, one of the brokers serves as the controller, which is responsible for managing the states of partitions and replicas and for performing administrative tasks like reassigning partitions. The following describes the states of partitions and replicas, and the kind of operations going through the controller.

KafkaController uses the KafkaZkClient (Zookeeper Client) to be notified about changes in the state of a Kafka cluster (that are reflected in changes in znodes of Apache Zookeeper) and propagate the state changes to other brokers (through Zookeeper).

Read more: - <https://cwiki.apache.org/confluence/display/KAFKA/Kafka+Controller+Internals> - <https://jaceklaskowski.gitbooks.io/apache-kafka/kafka-controller-KafkaController.html>

Question: 2

A consumer has `auto.offset.reset=latest`, and the topic partition currently has data for offsets going from 45 to 2311. The consumer group never committed offsets for the topic before. Where will the consumer read from?

- A. it will crash
- B. offset 0
- C. offset 2311
- D. offset 45

Answer: C

Explanation:

The latest means that data retrievals will start from where the offsets currently end.

Question: 3

Which of the following is correct? (select three)

- A. An offline consumer will resume reading where the offsets were last committed
- B. A consumer deletes messages in Kafka after reading them
- C. An offline consumer will miss data
- D. A consumer reading messages in Kafka won't delete messages
- E. A producer can produce to Kafka if a consumer is offline
- F. A producer cannot produce to Kafka if a consumer is offline

Answer: A,D

Explanation:

Data in Kafka is not deleted after being read, and offsets are committed to checkpoint progress. Consumers and Producers are decoupled and Producers are not aware of Consumers.

Question: 4

Consumers in your consumer group are rebalancing. What can be a possible cause? (select two)

- A. A producer has come online
- B. A Consumer left the group cleanly
- C. A Zookeeper node has come offline
- D. A Consumer has joined the group

Answer: B,D

Explanation:

Source: <https://stackoverflow.com/a/46774113/3019499>

Every consumer in a consumer group is assigned one or more topic partitions exclusively, and Rebalance is the re-assignment of partition ownership among consumers.

A Rebalance happens when:

- a consumer JOINS the group
- a consumer SHUTS DOWN cleanly
- a consumer is considered DEAD by the group coordinator. This may happen after a crash or when the consumer is busy with long-running processing, which means that no heartbeats have been sent in the meanwhile by the consumer to the group coordinator within the configured session interval
- new partitions are added

Being a group coordinator (one of the brokers in the cluster) and a group leader (the first consumer that joins a group) designated for a consumer group, Rebalance can be more or less described as follows:

- the leader receives a list of all consumers in the group from the group coordinator (this will include all consumers that sent a heartbeat recently and which are therefore considered alive) and is responsible for assigning a subset of partitions to each consumer.

- After deciding on the partition assignment (Kafka has a couple of built-in partition assignment policies), the group leader sends the list of assignments to the group coordinator, which sends this information to all the consumers.

Question: 5

In what circumstances is Zookeeper currently used in Apache Kafka and newer consumer groups? (select two)

- A. Maintain a list of all brokers
- B. Maintain a list of all consumer offsets
- C. Maintain a list of all topics
- D. Maintain a list of all consumer groups

Answer: A,C

Explanation:

You can find all the data structure of Zookeeper here:

<https://cwiki.apache.org/confluence/display/KAFKA/Kafka+data+structures+in+Zookeeper>

Old consumer groups used to store information in Zookeeper, but the new consumer groups store information directly in the Kafka brokers in the `__consumer_offsets` topic

Question: 6

What is true about replicas under the default consuming scheme ?

- A. The follower replica handles all consumers requests
- B. The producers and consumers requests are load-balanced between Leader and Follower replicas
- C. The leader replica handles all producers and consumers requests
- D. The producer requests can be done to the replicas that are followers

Answer: C

Explanation:

Replicas are passive - they don't handle produce or consume requests. Produce and consume requests get sent to the node hosting partition leader.

Question: 7

In Avro, adding an element to an enum without a default is a `__` schema evolution

- A. Full

- B. forward
- C. backward
- D. breaking

Answer: D

Explanation:

Since Confluent 5.4.0, Avro 1.9.1 is used. Since default value was added to enum complex type , the schema resolution changed from:

(<1.9.1) if both are enums:** if the writer's symbol is not present in the reader's enum, then an error is signalled. **(>=1.9.1) if both are enums:

if the writer's symbol is not present in the reader's enum and the reader has a default value, then that value is used, otherwise an error is signalled.

Question: 8

A consumer inside a group is reading from Kafka. Each message processing can take a long time, and you would like to make sure your consumer is not disconnected during this time. What should you do? (select two)

- A. Increase session.timeout.ms
- B. Shut down the consumer after receiving each message, and put it back online after it's done processing the message.
- C. Disable auto commit of messages
- D. After receiving a message, call .pause() and keep on calling .poll() in a separate thread.
- E. Increase max.poll.interval.ms

Answer: C,D

Explanation:

For use cases where message processing time varies unpredictably the recommended way to handle these cases is to move message processing to another thread, which allows the consumer to continue calling poll while the processor is still working. Some care must be taken to ensure that committed offsets do not get ahead of the actual position. Typically, you must disable automatic commits and manually commit processed offsets for records only after the thread has finished handling them (depending on the delivery semantics you need). Note also that you will need to pause the partition so that no new records are received from poll until after thread has finished handling those previously returned.

Source: <https://kafka.apache.org/10/javadoc/org/apache/kafka/clients/consumer/KafkaConsumer.html>

Question: 9

There are five brokers in a cluster, a topic with 10 partitions and a replication factor of 3, and a quota of `producer_bytes_rate` of 1 MB/sec has been specified for the client. What is the maximum throughput allowed for the client?

- A. 10 MB/s
- B. 0.33 MB/s
- C. 5 MB/s
- D. 1 MB/s

Answer: C

Explanation:

Each producer is allowed to produce @ 1MB/s to a broker (the quota). Max throughput $5 * 1MB$, because we have 5 brokers.

Question: 10

Your manager would like to have topic availability over consistency. Which setting do you need to change to enable that?

- A. `min.insync.replicas`
- B. `compression.type`
- C. `unclean.leader.election.enable`

Answer: C

Explanation:

`unclean.leader.election.enable=true` allows non ISR replicas to become leader, ensuring availability but losing consistency as data loss will occur