

# Latest Version: 6.0

## Question: 1

Select the Kafka Streams joins that are always windowed joins.

- A. KStream-KTable join
- B. KStream-GlobalKTable
- C. KStream-KStream join
- D. KTable-KTable join

**Answer: C**

Explanation:

See: <https://docs.confluent.io/current/streams/developer-guide/dsl-api.html#joining>

## Question: 2

You are sending messages with keys to a topic. To increase throughput, you decide to increase the number of partitions of the topic. Select all that apply.

- A. New records may get written to a different partition
- B. All the existing records will get rebalanced among the partitions to balance load
- C. New records with the same key will get written to the partition where old records with that key were written
- D. Old records will stay in their partitions

**Answer: A,D**

Explanation:

Increasing the number of partition causes new messages keys to get hashed differently, and breaks the guarantee "same keys goes to the same partition". Kafka logs are immutable and the previous messages are not re-shuffled

## Question: 3

To read data from a topic, the following configuration is needed for the consumers

- A. the list of brokers that have the data, the topic name and the partitions list
- B. any broker, and the list of topic partitions
- C. any broker to connect to, and the topic name

D. all brokers of the cluster, and the topic name

**Answer: C**

Explanation:

All brokers can respond to Metadata request, so a client can connect to any broker in the cluster.

### Question: 4

When using plain JSON data with Connect, you see the following error message: `org.apache.kafka.connect.errors.DataException: JsonSerializer with schemas.enable requires "schema" and "payload" fields and may not contain additional fields.` How will you fix the error?

- A. Use Single Message Transforms to add schema and payload fields in the message
- B. Set `key.converter`, `value.converter` to `AvroConverter` and the schema registry url
- C. Set `key.converter`, `value.converter` to `JsonConverter` and the schema registry url
- D. Set `key.converter.schemas.enable` and `value.converter.schemas.enable` to false

**Answer: D**

Explanation:

You will need to set the `schemas.enable` parameters for the converter to false for plain text with no schema.

### Question: 5

A kafka topic has a replication factor of 3 and `min.insync.replicas` setting of 2. How many brokers can go down before a producer with `acks=all` can't produce?

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

**Answer: A**

Explanation:

`acks=all` and `min.insync.replicas=2` means we must have at least 2 brokers up for the partition to be available

### Question: 6

The kafka-console-consumer CLI, when used with the default options

- A. does not use a group id
- B. uses a random group id
- C. always uses the same group id

**Answer: B**

Explanation:

If a group is not specified, the kafka-console-consumer generates a random consumer group.

### Question: 7

How will you find out all the partitions without a leader?

- A. kafka-topics.sh --broker-list localhost:9092 --describe --under-replicated-partitions
- B. kafka-topics.sh --bootstrap-server localhost:2181 --describe --unavailable-partitions
- C. kafka-topics.sh --zookeeper localhost:2181 --describe --under-replicated-partitions
- D. kafka-topics.sh --zookeeper localhost:2181 --describe --unavailable-partitions

**Answer: D**

Explanation:

Please note that as of Kafka 2.2, the --zookeeper option is deprecated and you can now use: kafka-topics.sh --bootstrap-server localhost:9092 --describe --unavailable-partitions

### Question: 8

Your topic is log compacted and you are sending a message with the key K and value null. What will happen?

- A. The broker will delete the message with the key K and null value only upon cleanup
- B. The producer will throw a Runtime exception
- C. The broker will delete all messages with the key K upon cleanup
- D. The message will get ignored by the Kafka broker

**Answer: C**

Explanation:

Sending a message with the null value is called a tombstone in Kafka and will ensure the log compacted topic does not contain any messages with the key K upon compaction

## Question: 9

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

**Answer: D**

Explanation:

Callback is invoked when a broker response is received.

## Question: 10

A producer application was sending messages to a partition with a replication factor of 2 by connecting to Broker 1 that was hosting partition leader. If the Broker 1 goes down, what will happen?

- A. The topic will be unavailable
- B. The producer will stop working
- C. The producer will automatically produce to the broker that has been elected leader

**Answer: C**

Explanation:

Once the client connects to any broker, it is connected to the entire cluster and in case of leadership changes, the clients automatically do a Metadata Request to an available broker to find out who is the new leader for the topic. Hence the producer will automatically keep on producing to the correct Kafka Broker

## Question: 11

Which actions will trigger partition rebalance for a consumer group? (select three)

- A. Increase partitions of a topic
- B. A consumer in a consumer group shuts down
- C. Remove a broker from the cluster
- D. Add a broker to the cluster
- E. Add a new consumer to consumer group

**Answer: A,B,E**

Explanation:

Rebalance occurs when a new consumer is added, removed or consumer dies or partitions increased.

## Question: 12

Kafka is configured with following parameters - log.retention.hours = 168 log.retention.minutes = 168 log.retention.ms = 168 How long will the messages be retained for?

- A. 168 minutes
- B. Broker will not start due to bad configuration
- C. 168 ms
- D. 168 hours

**Answer: C**

Explanation:

If more than one similar config is specified, the smaller unit size will take precedence.

## Question: 13

Compaction is enabled for a topic in Kafka by setting log.cleanup.policy=compact. What is true about log compaction?

- A. Kafka automatically de-duplicates incoming messages based on key hashes
- B. Each message stored in the topic is compressed
- C. After cleanup, only one message per key is retained with the first value
- D. After cleanup, only one message per key is retained with the latest value
- E. Compaction changes the offset of messages

**Answer: D**

Explanation:

Log compaction retains at least the last known value for each record key for a single topic partition. All compacted log offsets remain valid, even if record at offset has been compacted away as a consumer will get the next highest offset.

## Question: 14

You have a Zookeeper cluster that needs to be able to withstand the loss of 2 servers and still be able to function. What size should your Zookeeper cluster have?

- A. 3
- B. 5
- C. 2
- D. 6
- E. 4

**Answer: B**

Explanation:

Your Zookeeper cluster needs to have an odd number of servers, and must maintain a majority of servers up to be able to vote. Therefore, a  $2N+1$  zookeeper cluster can survive to  $N$  zookeeper being down, so here the right answer is  $N=2$ ,  $2*N+1=5$

## Question: 15

How can you gracefully make a Kafka consumer to stop immediately polling data from Kafka and gracefully shut down a consumer application?

- A. Kill the consumer thread
- B. Call `consumer.poll()` in another thread
- C. Call `consumer.wakeup()` and catch a `WakeupException`

**Answer: C**

Explanation:

See <https://stackoverflow.com/a/37748336/3019499>