**AWS Certified Data Analytics** - **Specialty (DAS-C01) Practice Questions**

**Requirement**: Create & Share 20 DAS-C01 practice questions.

**Important Note**: The practice questions should appropriately belong to Data Analytics Specialty in terms of exam objectives & difficulty level.

**Delivery Timeline**: May-17

**Question Response Types**

There are two types of questions:

● Multiple Choice Single Response – **1** correct answer **3** incorrect responses (distractors).

● Multiple Choice Multiple Response – **2** or more correct answers out of **5** or more options.

**Important Note**

● Do write Question Number for quick identification. Q# 1, Q# 2 …. & so on.

● Please mention Domain (based on Data Analytics exam blueprint), Topic & Sub-Topic (If Applicable) with every question.

● Note that we’re expecting standard scenario based questions & NOT straight-forward definition kind of questions.

● The options should not have any obviously incorrect option. We need to word the options so that all of them should appear correct for the students, but a subtle point should mark the correct answer without any ambiguity. So, one answer should be the best choice without any doubt.

● The answer / explanation section should contain explanations on why the answer is correct and others are incorrect. It should also contain the relevant resource link (for details) preferably from AWS documentation.

○ Example

■ Option A is incorrect because..

■ Option B is CORRECT because…

■ Option C is incorrect because..

■ Option D is incorrect because..

● Try to balance the domains based on weightage % defined in the exam blueprint.

● Any AWS service or feature must be approximately 6 months old to figure out in Practice Tests. Put a note in the explanation for any latest service or feature that might be on the borderline of appearing in the real exam.

● **Plagiarism** in any form - Question or in Explanation will be **rejected.** Questions & Explanations should reflect your own professional experience & AWS skills. Author’s who indulge in plagiarism will be **blacklisted** & dropped from our author’s list.

● The ownership of the questions once approved & published on Whizlabs LMS platform, lies solely with Whizlabs Software Pvt. Ltd. You can’t share or publish it elsewhere in any circumstances.

**Sample Format of** **Questions**

----------------------------------------------------------------------------------------------------------------------------

**Question​ ​:​** #

**Main​ ​Topic​ ​:​** < >

**Sub​ ​Topic​ ​:​** [optional]

**Domain:** < >

**Question text**:

<Scenario based. Should be clear in terms of requirements. No ambiguity. No duplicate options. In case of multiple answers, at the end, you should include number of expected answers. e.g. (Select TWO) or (Select THREE) etc. For single answers this is NOT required>

**A)** Option A...

**B)** Option B...

**C)** Option C...

**D)** Option D...

**Answer:** A and C

**Explanation:**

**Option A is CORRECT because...**

**Option B is incorrect because...**

**Option C is CORRECT because...**

**Option D is incorrect because...**

[Insert the explanation in clear and lucid language here.]

**Diagram:** [Optional] [Insert the architectural or conceptual diagram here.]

**Reference:** [Insert the references here - which may include links to AWS Documentation, Blog, re:Invent video, Authority YouTube video].

----------------------------------------------------------------------------------------------------------------------------

**Data Analytics** **Specialty has 5 Domains**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Name of the Domain** | **Weight** | **Estimated No. of Questions**  (Out of 65 as per weightage %) |
| 1 | Collection | 18% | 12 |
| 2 | Storage and Data Management | 22% | 14 |
| 3 | Processing | 24% | 15 |
| 4 | Analysis and Visualization | 18% | 12 |
| 5 | Security | 18% | 12 |

--------------------------------------Question Section Starts-----------------------------------------------------

Question: 1

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Determine the operational characteristics of a storage solution for analytics**

**Domain:** Collection

**Question text**:

You are a data scientist working for a healthcare services company that produces a Software as a Service (SaaS) offering to its industry affiliates. Your current architecture is built upon an aging infrastructure that is all run on-premises using local physical servers. Your SaaS solution provides data analytics and visualization services for your partner healthcare firms. These firms send you healthcare data for consolidation. Your service ingests the data from these disparate sources, transforms the data, processes it, and finally produces visualizations for consumption by your partners.

Unfortunately, your current solution, depicted in the diagram below, is prone to failures and is not easily scaled. You also have a difficult optimizing the costs of your SaaS offering because you are using sunk costs (the physical servers). You have been tasked with modernizing your analytics solution.

[](https://www.draw.io/?page-id=B8PfgTc1ObyOHXG35zZY&scale=auto#G1dfCJuuEcrWGeKUHvVPOuYyu-bpN3l92R)

Which option provides the most cost effective, performant, and scalable solution?

1. Healthcare data sources -> Kinesis Data Firehose -> S3 -> EMR running Hive -> S3 Data Lake -> Redshift -> Athena
2. Healthcare data sources -> Kinesis Data Firehose -> S3 -> EMR running Spark -> S3 Data Lake -> Redshift -> QuickSight
3. Healthcare data sources -> Kinesis Data Streams -> S3 -> EMR running Hive -> S3 Data Lake -> Redshift -> QuickSight
4. Healthcare data sources -> Kinesis Data Streams -> S3 -> EMR running Spark -> S3 Data Lake -> Redshift -> Athena

**Answer:** B

**Explanation:**

Option A is incorrect. Apache Hive is used to manage Hive clusters, but it wouldn’t give you the capability to transform your data in preparation for integration with Redshift. Also, using Athena to produce visualizations will be less cost effective than using QuickSight because Athena will require more development work on your part.

Option B is correct. Using Kinesis Data Firehose to collect your healthcare data, Spark running on your EMR cluster to transform your data, and Redshift to source your QuickSight visualizations is the most cost effective, scalable, and performant option.

Option C is incorrect. Apache Hive is used to manage Hive clusters, but it wouldn’t give you the capability to transform your data in preparation for integration with Redshift. Also, Kinesis Data Streams will require you to write producer and consumer applications to complete your data collection pipeline. This will be less cost effective than using Kinesis Data Firehose.

Option D is incorrect. Kinesis Data Streams will require you to write producer and consumer applications to complete your data collection pipeline. This will be less cost effective than using Kinesis Data Firehose. Also, using Athena to produce visualizations will be less cost effective than using QuickSight because Athena will require more development work on your part.

**Reference:**

Please see the **Amazon EMR overview page** (<https://aws.amazon.com/emr/>), the Amazon EMR features page titled **Apache Hive on AMazon EMR** (<https://aws.amazon.com/emr/features/hive/>), the article titled **Tutorial: Using Redshift and Amazon QuickSight to deliver business analytics** (<https://www.stitchdata.com/blog/tutorial-using-redshift-and-amazon-quicksight-to-deliver-business-analytics/>)

Question: 2

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Select a collection system that handles the frequency, volume, and source of data**

**Domain:** Collection

**Question text**:

You are a data scientist working for a securities trading firm that receives trading data from multiple market data producer sources. Your task is to consume the data from these producers cost effectively while also maximizing the performance of your data collection system. Your data collection system must deliver the aggregated producer data to your firm’s data lake for analytics application use.

You have built a collection system as follows: a Kinesis Producer Library application that writes records to your Kinesis Data Streams stream shards. You have configured your Kinesis Producer constructor as follows:

RecordMaxBufferedTime = 200

MaxConnections = 2

RequestTimeout = 5000

Region = us-east-1

Your Kinesis Data Streams writes to Kinesis Data Firehose. Kinesis Data Firehose uses a Lambda function to transform your data into the Avro format before writing it to your S3 bucket in your data lake.

You have noticed that your data collection pipeline is not performing as well as you had expected. What may be the cause, and what can you do to improve the situation?

1. Your RecordMaxBufferedTime value is too low, resulting in lower aggregation efficiency, so your pipeline throughput is slow. Dynamically change the RecordMaxBufferedTime to 3000. This allows the KinesisProducer to deliver larger aggregate packages to your Kinesis Data Stream.
2. Your RecordMaxBufferedTime value is too high, resulting in lower aggregation efficiency, so your pipeline throughput is slow. Change the RecordMaxBufferedTime to 50 and restart your KPL application. This allows the KinesisProducer to deliver larger aggregate packages to your Kinesis Data Stream.
3. Your RecordMaxBufferedTime value is too low, resulting in lower aggregation efficiency, so your pipeline throughput is slow. Change the RecordMaxBufferedTime to 3000 and restart your KPL application. This allows the KinesisProducer to deliver larger aggregate packages to your Kinesis Data Stream.
4. Your RequestTimeout value is too high, resulting in lower aggregation efficiency, so your pipeline throughput is slow. Dynamically change the RequestTimeout to 2000. This allows the KinesisProducer to deliver larger aggregate packages to your Kinesis Data Stream.

**Answer:** C

**Explanation:**

Option A is incorrect. Changing your RecordMaxBufferedTime to a higher value will increase your aggregate package size, thereby improving the performance of your pipeline throughput. However, you cannot dynamically change your KinesisProducer once it has been initialized without restarting your KPL application.

Option B is incorrect. Lowering your RecordMaxBufferedTime will actually result in a smaller aggregate package size. This will make your KPL producer to Kinesis Data Streams pipe even slower.

Option C is correct. Changing your RecordMaxBufferedTime to a higher value will increase your aggregate package size, thereby improving the performance of your pipeline throughput. Also, you must restart your KPL application if you want to change any of the KinesisProducerConfiguration values.

Option D is incorrect. Changing the RequestTimeout will not improve the performance of your pipeline throughput. It will allow your KPL application to wait longer before failing when trying to write to your Kinesis Data Streams stream.

**Reference:**

Please see the Amazon Kinesis Data Streams developer guide titled **Developing Producers Using the Amazon Kinesis Producer Library** (<https://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html>), the Amazon Kinesis Data Streams developer guide titled **Configuring the Kinesis Producer Library** (<https://docs.aws.amazon.com/streams/latest/dev/kinesis-kpl-config.html>), the Amazon Kinesis Data Streams developer guide titled **Using the KPL with Kinesis Data Firehose** (<https://docs.aws.amazon.com/streams/latest/dev/kpl-with-firehose.html>), and the Amazon Kinesis Data Firehose developer guide titled **Writing to Kinesis Data Firehose Using Kinesis Data Streams** (<https://docs.aws.amazon.com/firehose/latest/dev/writing-with-kinesis-streams.html>)

Question: 3

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Select a collection system that addresses the key properties of data, such as order, format, and compression**

**Domain:** Collection

**Question text**:

You are a data scientist working for a transportation company that specializes in delivering cargo to manufacturing companies. You have been tasked with building a data collection system to gather all of your logistics data into a data lake. This data will be used by analytics applications to perform operations management tasks such as solving the “traveling salesman” problem, where your analytics application needs to find the optimal path for your delivery truck to take to all of its destinations. This optimal path needs to maximize cost efficiency as well as meeting delivery timelines.

You have constructed a Kinesis Data Streams infrastructure with KPL producer applications delivering the transportation data into your Kinesis shards. You are in the process of building your Kinesis Consumer Library application code to consume the streaming data from Kinesis and write the data to your S3 buckets. What happens when your KCL worker code fails in the middle of retrieving a record from your Kinesis stream?

1. Your KCL implementation takes advantage of checkpointing, where KCL stores a cursor on S3 to keep track of records that you have read from a shard. To recover from a failed KCL read, a new KCL worker uses the cursor to restart from the record so you don’t lose the record from the shard.
2. Your KCL implementation takes advantage of checkpointing, where KCL stores a cursor in DynamoDB to keep track of records that you have read from a shard. To recover from a failed KCL read, a new KCL worker uses the cursor to restart from the record so you don’t lose the record from the shard.
3. Your KCL worker will throw an exception from its processRecords call. You need to handle the exception in your KCL application code.
4. Your KCL worker shuts down with shutdown reason ZOMBIE and the KCL throws a ShutdownException.

**Answer:** B

**Explanation:**

Option A is incorrect. Your KCL worker does take advantage of checkpointing, but it persists its checkpoint cursor data to DynamoDB, not S3.

Option B is correct. Your KCL worker takes advantage of checkpointing, persisting its checkpoint cursor data to DynamoDB. The KCL will use the cursor information to restart at the exact record where the previous worker failed.

Option C is incorrect. A processRecords exception results in records being skipped by your consumer code. If your worker fails in the middle of retrieving a record from the Kinesis shard, the KCL uses its cursor stored in DynamoDB to recover from that failed record.

Option D is incorrect. The ZOMBIE reason code is returned by the KCL when a worker has its shards taken by another worker. This scenario describes the situation where a worker fails in the middle of reading a record from your Kinesis shard, which is a different failure scenario.

**Reference:**

Please see the AWS Big Data blog titled **Persist Streaming Data to Amazon S3 using Amazon Kinesis Firehose and AWS Lambda** (<https://aws.amazon.com/blogs/big-data/persist-streaming-data-to-amazon-s3-using-amazon-kinesis-firehose-and-aws-lambda/>), the Amazon Kinesis Data Streams developer guide titled **Troubleshooting Kinesis Data Streams Consumers** (<https://docs.aws.amazon.com/streams/latest/dev/troubleshooting-consumers.html>), and the Amazon Kinesis Data Streams getting started guide titled **Getting started with Amazon Kinesis Data Streams** (<https://aws.amazon.com/kinesis/data-streams/getting-started/>)

Question: 4

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Select a collection system that addresses the key properties of data, such as order, format, and compression**

**Domain:** Collection

**Question text**:

You are a data scientist working for a sports media publishing company. Your company owns a series of sports websites that give sports fans access to data about their sport such as game schedules, player statistics, game results, live streaming game data, etc.

You have been tasked with collecting the sports data from your 100+ websites in real-time, process it, and then store it in your S3 data lake. You also need to store the raw data you receive in a separate S3 prefix so that the raw data can be used by other analytics applications in the future.

Which of the following options is the most cost effective and simplest to implement that addresses your requirements?

1. Recieve sports site data into Kinesis Data Streams. Simultaneously read the Kinesis Data Stream into your EMR cluster running Spark AND your KCL/KCL-C worker application. The EMR Spark instance transforms the raw data and writes it to your S3 bucket processed prefix. The KCL/KCL-C application writes the raw data to the raw prefix on your S3 bucket.
2. Recieve sports site data into Kinesis Data Streams. Simultaneously read the Kinesis Data Stream into your EMR cluster using Kinesis Data Analytics AND your Kinesis Data Firehose stream. The EMR Kinesis Data Analytics instance transforms the raw data and writes it to your S3 bucket processed prefix. The Kinesis Data Firehose leverages a lambda function you write to aggregate the raw data and then write the raw data to the raw prefix on your S3 bucket.
3. Recieve sports site data into Kinesis Data Streams. Simultaneously read from Kinesis Data Streams into your EMR cluster running Spark AND your KPL/KPL-C worker application. The EMR Spark instance transforms the raw data and writes it to your S3 bucket processed prefix. The KPL/KPL-C application writes the raw data to the raw prefix on your S3 bucket.
4. Recieve sports site data into Kinesis Data Streams. Simultaneously read the Kinesis Data Stream into your EMR cluster running Spark AND your Kinesis Data Firehose stream. The EMR Spark instance transforms the raw data and writes it to your S3 bucket processed prefix. The Kinesis Data Firehose leverages a lambda function you write to aggregate the raw data and then write the raw data to the raw prefix on your S3 bucket.

**Answer:** D

**Explanation:**

Option A is incorrect. While this option technically works, using the KCL/KCL-C application code requires more coding effort than using Kinesis Data Firehose and Lambda. Also, with this option you have to provision, maintain, and pay for EC2 servers to run your KCL/KCL-C application.

Option B is incorrect. This option is technically incorrect. You cannot use Kinesis Data Analytics on an EMR cluster to perform data transformation.

Option C is incorrect. While you can use the Kinesis Producer Library (KPL) to produce data to place into your Kinesis Data Stream, it will not work as a consumer as depicted in this option. Also, there is no KPL-C producer library.

Option D is correct. The Kinesis Data Firehose with Lambda function transformation requires less coding work when compared to writing the KCL/KCL-C worker. Also, Kinesis Data Firehose and Lambda are serverless, therefore they require no EC2 provisioning, maintenance, or cost.

**Reference:**

Please see the AWS Big Data blog titled **Persist Streaming Data to Amazon S3 using Amazon Kinesis Firehose and AWS Lambda** (<https://aws.amazon.com/blogs/big-data/persist-streaming-data-to-amazon-s3-using-amazon-kinesis-firehose-and-aws-lambda/>), the Amazon Kinesis Data Streams getting started guide titled **Getting started with Amazon Kinesis Data Streams** (<https://aws.amazon.com/kinesis/data-streams/getting-started/>), and the Amazon EMR release guide titled **Kinesis** (<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-kinesis.html>)

Question: 5

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Determine the operational characteristics of a storage solution for analytics**

**Domain:** Storage and Data Management

**Question text**:

You recently started working as a data scientist for a large real estate company. Your real estate brokers need near real-time streaming data on interest rates and loan offerings for their regional markets. They also need near real-time streaming data describing their regional real estate inventory, for example on the market, sold, pending sale, etc.

You have constructed a Kinesis Data Firehose data collection pipeline to gather the data. You now wish to store the data in a DynamoDB database for access via REST APIs by your real estate agents out in the field using their mobile devices. You have implemented the REST APIs using API Gateway.

When you run your first canary deployment of your Lambda function you notice that your Lambda function attempts to process your buffered Kinesis records 3 times and then skips the batch of records. What might be the cause of the problem, and how can you correct the issue?

1. Your Kinesis Firehose buffer interval is set to 60 seconds. This setting is too high, causing your buffer to grow too large for your Lambda invocation and resulting in an invocation limit error. Lower your Kinesis Firehose buffer interval.
2. Your Kinesis Firehose buffer size is set to 1 MB. This setting is too low, causing your Kinesis Firehose to invoke your Lambda function in asynchronous invocation mode. Increase your Kinesis Firehose buffer size.
3. Your Kinesis Firehose buffer size is set to 7 MB. This setting is too high, causing your Lambda function to fail with an invocation limit error. Lower your Kinesis Firehose buffer size.
4. Your Lambda function is returning a response payload of 3 MB. This payload is too large, causing your Lambda function to fail with an invocation limit error. Lower the size of your Lambda function response payload.

**Answer:** C

**Explanation:**

Option A is incorrect. The buffer interval range for Kinesis Data Firehose is 60 to 900 seconds. Therefore, you cannot lower your buffer interval.

Option B is incorrect. Kinesis Firehose invokes your Lambda function in synchronous mode, not asynchronous mode.

Option C is correct. Lambda has an invocation payload limit of 6 MB for synchronous invocations. Kinesis Firehose invokes your Lambda function in synchronous invocation mode. This type of data transformation failure results in three tries before skipping the batch of records. Lowering your Kinesis Firehose buffer size to a value 6 MB or less will solve the issue.

Option D is incorrect. The Lambda function response payload limit is 6 MB. Therefore, a response payload of 3 MB would not cause your Lambda function to fail with an invocation limit error.

**Reference:**

Please see the AWS Database blog titled **Query your AWS database from your serverless application** (<https://aws.amazon.com/blogs/database/query-your-aws-database-from-your-serverless-application/>), the AWS Lambda developer guide titled **Invoking AWS Lambda functions** (<https://docs.aws.amazon.com/lambda/latest/dg/lambda-invocation.html>), the AWS Architecture blog titled **Understanding the Different Ways to Invoke Lambda Functions** (<https://aws.amazon.com/blogs/architecture/understanding-the-different-ways-to-invoke-lambda-functions/>), the Amazon API Gateway developer guide titled **Creating a REST API in Amazon API Gateway** (<https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-create-api.html>), the Amazon Kinesis Data Firehose developer guide titled **Amazon Kinesis Data Firehose Data Transformation** (<https://docs.aws.amazon.com/firehose/latest/dev/data-transformation.html>), the AWS Lambda developer guide titled **AWS Lambda limits** (<https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html>), the Amazon Kinesis Data Firehose developer guide titled **Configure Settings** (<https://docs.aws.amazon.com/firehose/latest/dev/create-configure.html>), and the **Amazon Kinesis Data Firehose FAQs** (<https://aws.amazon.com/kinesis/data-firehose/faqs/>)