**AWS Certified Data Analytics** - **Specialty Practice Questions**

**Requirement**: Share 10 DA Specialty practice questions.

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

**Delivery Timeline**: April-4

**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 DA Specialty 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 the number of expected answers. e.g. (Select TWO) or (Select THREE) etc. For single answers this is NOT required>

1. Option A...
2. Option B...
3. Option C...
4. 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].

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

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

**Domain:** Data collection systems

**Question text**:

You are a data scientist working on a project where you have two large tables (orders and products) that you need to load into Redshift from one of your S3 buckets. Your table files, which are both several million rows large, are currently on an EBS volume of one of your EC2 instances in a directory titled $HOME/myredshiftdata.

Since your table files are so large, what is the most efficient approach to first copy them to S3 from your EC2 instance?

1. Load your orders.tbl and products.tbl using the command: ‘aws s3 cp $HOME/myredshiftdata s3://dataanalytics/myredshiftdata --recursive’
2. Load your orders.tbl and products.tbl by first splitting each tbl file into smaller parts using the command: ‘split -d -l 5000000 -a 4 orders.tbl orders.tbl’ and ‘split -d -l 10000000 -a 4 products.tbl products.tbl’
3. Load your orders.tbl and products.tbl by first getting a count of the number of rows in each using the commands: ‘wc -l orders.tbl’ and ‘wc -l products.tbl’. Then splitting each tbl file into smaller parts using the command: ‘split -d -l # -a 4 orders.tbl orders.tbl’ and ‘split -d -l # -a 4 products.tbl products.tbl’ where # is replaced by the result of your wc command divided by 4.
4. Load your orders.tbl and products.tbl by first getting a count of the number of rows in each using the commands: ‘wc -l orders.tbl’ and ‘wc -l products.tbl’. Then splitting each tbl file into smaller parts using the command: ‘split -d -l # -a 4 orders.tbl orders.tbl-’ and ‘split -d -l # -a 4 products.tbl products.tbl-’ where # is replaced by the result of your wc command divided by 4.

**Answer:** D

**Explanation:**

Option A is incorrect because using the commands in this answer you don’t reduce the size of your tbl files before attempting to move them to S3. Also, when you attempt to move these files into Redshift from your S3 bucket the process will be less efficient because you haven’t split your files into more manageable sizes.

Option B is incorrect because when you attempt to split your files you haven’t determined the actual number of rows of each file. Therefore, your random selection of a split size will more than likely not be an efficient selection.

Option C is incorrect because your split command does not have a trailing ‘-’ at the end of the command. Therefore your smaller files on your S3 bucket will have names like ‘orders.tbl0001’ versus the more readable and manageable ‘orders.tbl-0001’ if you use a trailing ‘-’ in the split command.

Option D is correct because you have used the wc command to find the number of rows in each tbl file, and you have used the split command with the trailing ‘-’ to get the proper file name format on your S3 bucket in preparation for loading into Redshift.

**Reference:**

Please see the AWS Redshift Developer Guide titled **Tutorial: Loading Data from Amazon S3** (<https://docs.aws.amazon.com/redshift/latest/dg/tutorial-loading-data.html>), specifically step 2: Download the Data Files and Step 5: Run the Copy Commands where you’ll see that having the ‘-’ at the end of your split command will allow you to make your Redshift copy command more efficient.

Question: 2

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Determine appropriate data processing solution requirements**

**Domain:** Data processing solutions

**Question text**:

You are working on a project where you need to perform real-time analytics on your application server logs. Your application is split across several EC2 servers in an auto-scaling group and is behind an application load balancer as depicted in this diagram:

[](https://www.draw.io/?page-id=6_lzF5P4isDOIafqsJO-&scale=auto#G1GNODpLc0DVlcFq7mT79w1ym0bAFvR-pw)

You need to perform some transformation on the log data, such as joining rows of data, before you stream the data to your real-time dashboard.

What is the most efficient and performant solution to aggregate your application logs?

1. Install the Kinesis Agent on your application servers to watch your logs and use Kinesis Data Firehose to stream the logs directly to S3. Use Kinesis Data Analytics queries to build your real-time analytics dashboard.
2. Write a Kinesis Data Streams producer application that reads the application logs and pushes the data directly into your Kinesis data stream. Use Kinesis Data Analytics queries to build your real-time analytics dashboard.
3. Install the Kinesis Agent on your application servers to watch your logs and ingest the log data. Write a Kinesis Data Analytics application that reads the application log data from the agent, performs the required transformations, and pushes the data into your Kinesis data output stream. Use Kinesis Data Analytics queries to build your real-time analytics dashboard.
4. Use a CloudWatch dashboard that uses your application’s CloudWatch logs as the data source.

**Answer:** C

**Explanation:**

Option A is incorrect because with this approach you don’t have a capability to perform the required transformations. You could write a lambda function to perform the transformations but the answer doesn’t specify these details.

Option B is incorrect because the answer is missing the Kinesis Agent part of the solution. You could write your Kinesis producer application to read the application log files, but using the Kinesis Agent is much more efficient.

Option C is correct. The Kinesis Agent ingests the application log data, the Kinesis Analytics application transforms the data, and Kinesis Analytics queries are used to build your dashboard.

Option D is incorrect since while a CloudWatch dashboard could be used to build this solution simply, it lacks the real-time capability. CloudWatch high-resolution metrics log in intervals of 1 second, 5 seconds, 10 seconds, 30 seconds, or multiples of 60 seconds. Also, this solution lacks the ability to perform the required transformations of the log data.

**Reference:**

Please see the **Amazon CloudWatch FAQs** (<https://aws.amazon.com/cloudwatch/faqs/>), 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 blog titled **Implement Serverless Log Analytics Using Kinesis Analytics** (<https://aws.amazon.com/blogs/big-data/implement-serverless-log-analytics-using-amazon-kinesis-analytics/>), and the **Amazon Kinesis Data Streams overview page** (<https://aws.amazon.com/kinesis/data-streams/>)

Question: 3

**Main​ ​Topic​ ​:​** Data Analytics

**Sub​ ​Topic​ ​:​ Design a solution for transforming and preparing data for analysis**

**Domain:** Data processing solutions

**Question text**:

You are a data scientist on a team where you are responsible for ingesting IoT streamed data into a data lake for use in an EMR cluster. The data in the data lake will be used to allow your company to do business intelligence analytics on the IoT data. Due to the large amount of data being streamed to your application you will need to compress the data on the fly as you process it into your EMR cluster.

How would you most efficiently collect the data from your IoT devices?

1. Use the Kinesis REST API to get the IoT device records from the IoT devices and stream the data to your data lake through Kinesis Data Streams, then use Apache DistCp to move the data from S3 to your EMR cluster.
2. Use the AWS IoT service to get the device data from the IoT devices, use Kinesis Data Firehose to stream the data to your data lake, then use S3DistCp to move the data from S3 to your EMR cluster.
3. Use the Kinesis Producer Library to create a Kinesis producer application that reads the data from the IoT devices and stream the data to your data lake through Kinesis Data Streams, then use S3DistCp to move the data from S3 to your EMR cluster.
4. Use the Kinesis Client Library to get the device data from the IoT devices, use Kinesis Data Firehose to stream the data to your data lake, then use Apache DistCp to move the data from S3 to your EMR cluster.

**Answer:** B

**Explanation:**

Option A is incorrect because the Kinesis REST API is not the most efficient way to gather the IoT device data from your set of devices. Also, Apache DistCp does not offer the compression option that S3DistCp offers.

Option B is correct. The AWS IoT service ingests the device data, Kinesis Data Firehose streams the data to your S3 data lake, then the S3DistCp command is used to compress and move the data inno your EMR cluster

Option C is incorrect. The Kinesis Producer Library is not the most efficient way to gather the IoT device data from your set of devices.

Option D is incorrect. The Kinesis Client Library is used to consume Kinesis Stream data, not to produce data for consumption into the data stream. Also, Apache DistCp does not offer the compression option that S3DistCp offers.

**Reference:**

Please see the **AWS IoT overview page** (<https://aws.amazon.com/iot/>), the Amazon Premium Support Knowledge Center article titled **How can I copy large amounts of data from Amazon S3 into HDFS on my Amazon EMR cluster?**

(<https://aws.amazon.com/premiumsupport/knowledge-center/copy-s3-hdfs-emr/>), the Amazon EMR Release Guide titled **S3DistCp (s3-dist-cp)**

(<https://docs.aws.amazon.com/emr/latest/ReleaseGuide/UsingEMR_s3distcp.html>), the AWS Big Data blog titled **Seven Tips for Using S3DistCp on Amazon EMR to Move Data Efficiently Between HDFS and Amazon S3** (<https://aws.amazon.com/blogs/big-data/seven-tips-for-using-s3distcp-on-amazon-emr-to-move-data-efficiently-between-hdfs-and-amazon-s3/>), and the AWS Solutions page titled **Real-Time IoT Device Monitoring with Kinesis Data Analytics** (<https://aws.amazon.com/solutions/real-time-iot-device-monitoring-with-kinesis/>)

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:** Data collection systems

**Question text**:

You are a data scientist working for a mobile gaming company that is developing a new mobile gaming app that will need to handle thousands of messages per second arriving in your application data store. Due to the user interactivity of your game, all changes to the game datastore must be recorded with a before-change and after-change view of the data record. These data store changes will be used to deliver a near-real-time usage dashboard of the app for your management team.

What application collection system infrastructure best delivers these capabilities in the most performant and cost effective way?

1. Kinesis Firehose -> S3 -> EMR with Spark -> S3 -> Redshift -> QuickSight
2. DynamoDB -> DynamoDB Streams -> Lambda -> Kinesis Firehose -> Redshift -> QuickSight
3. Kinesis Firehose -> Aurora MySQL -> Lambda -> Kinesis Firehose -> Redshift -> QuickSight
4. Kinesis Data Streams -> Aurora MySQL -> Lambda->Kinesis Firehose -> Redshift -> QuickSight

**Answer:** B

**Explanation:**

Option A is incorrect because none of the collection systems listed easily allow for the before-change and after-change views of your applications data store changes. Also, there is no data store other than S3 in the listed collection system components. S3 is not the most cost effective data store for this type of application.

Option B is correct. Your application will write its game activity data to your DynamoDB table which will have DynamoDB streams enabled. DynamoDB Streams will record both the new and old (or before and after) images of any item in the DynamoDB table that is changed. Your Lambda function will be triggered by DynamoDB Streams. Your Lambda function will use the Firehose client to write to your Firehose stream. Firehose will stream your data to Redshift. Quicksite will visualize your data in near-real-time.

Option C is incorrect. Kinesis Firehose does not have the capability to write directly to Aurora. You would have to write your stream data to S3 then write a Lambda function, triggered on each write, to consume the data stream and then write the stream data to your Aurora data store. You could also use the Amazon Database Migration service to move your data from S3 to Aurora. Also, you would have to write custom code to record the before-change information.

Option D is incorrect. Kinesis Data Streams does not have the capability to write directly to Aurora. You would have to write a Kinesis consumer client using the Kinesis Consumer Library (KCL) to consume the data stream and then write the stream data to your Aurora data store. Also, you would have to write custom code to record the before-change information.

**Reference:**

Please see the Amazon DynamoDB developer guide titled **Capturing Table Activity with DynamoDB Streams**

(<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html#Streams.Processing>), the Medium.com article titled **Data Transfer Dynamodb to Redshift**

(<https://medium.com/@ananthsrinivas/data-transfer-dynamodb-to-redshift-5424d7fdf673>), the **Amazon Redshift overview page** (<https://aws.amazon.com/redshift/>), the AWS Database blog titled **Stream data into an Aurora PostgreSQL Database using AWS DMS and Amazon Kinesis Data Firehose** (<https://aws.amazon.com/blogs/database/stream-data-into-an-aurora-postgresql-database-using-aws-dms-and-amazon-kinesis-data-firehose/>), the AWS Database blog titled **Capturing Data Changes in Amazon Aurora Using AWS Lambda**

(<https://aws.amazon.com/blogs/database/capturing-data-changes-in-amazon-aurora-using-aws-lambda/>), the **Kinesis Data Firehose overview page** (<https://aws.amazon.com/kinesis/data-firehose/>), and the **Kinesis Data Streams overview page** (<https://aws.amazon.com/kinesis/data-streams/>)