



Free Questions for [DAS-C01](#) by [go4braindumps](#)

Shared by [Mills](#) on [22-07-2024](#)

For More Free Questions and Preparation Resources

[Check the Links on Last Page](#)

Question 1

Question Type: MultipleChoice

A manufacturing company is storing data from its operational systems in Amazon S3. The company's business analysts need to perform one-time queries of the data in Amazon S3 with Amazon Athena

a. The company needs to access the Athena service from the on-premises network by using a JDBC connection. The company has created a VPC. Security policies mandate that requests to AWS services cannot traverse the internet.

Which combination of steps should a data analytics specialist take to meet these requirements? (Select TWO.)

Options:

- A-** Establish an AWS Direct Connect connection between the on-premises network and the VPC.
- B-** Configure the JDBC connection to connect to Athena through Amazon API Gateway.
- C-** Configure the JDBC connection to use a gateway VPC endpoint for Amazon S3.
- D-** Configure the JDBC connection to use an interface VPC endpoint for Athena.
- E-** Deploy Athena within a private subnet.

Answer:

A, D

Explanation:

AWS Direct Connect is a service that establishes a dedicated network connection between your on-premises network and AWS¹. It can help you reduce network costs, increase bandwidth throughput, and provide a more consistent network experience than internet-based connections¹. It can also help you meet the security policy that requires requests to AWS services not to traverse the internet.

An interface VPC endpoint is a type of VPC endpoint that enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink². It is represented by one or more Elastic Network Interfaces (ENIs) with private IP addresses in your VPC subnets². It can also help you meet the security policy that requires requests to AWS services not to traverse the internet.

Amazon Athena now provides an interface VPC endpoint that allows you to connect directly to Athena through an interface VPC endpoint in your VPC³. You can create an interface VPC endpoint to connect to Athena using the AWS console or AWS CLI commands⁴. You can also configure the JDBC connection to use the interface VPC endpoint for Athena by specifying the endpoint URL as the JDBC URL⁵.

Question 2

Question Type: MultipleChoice

A company receives datasets from partners at various frequencies. The datasets include baseline data and incremental data. The company needs to merge and store all the datasets without reprocessing the data.

Which solution will meet these requirements with the LEAST development effort?

Options:

- A- Use an AWS Glue job with a temporary table to process the datasets. Store the data in an Amazon RDS table.
- B- Use an Apache Spark job in an Amazon EMR cluster to process the datasets. Store the data in EMR File System (EMRFS).
- C- Use an AWS Glue job with job bookmarks enabled to process the datasets. Store the data in Amazon S3.
- D- Use an AWS Lambda function to process the datasets. Store the data in Amazon S3.

Answer:

C

Explanation:

AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy to prepare and load data for analytics¹. It can process datasets from various sources and formats, such as JDBC, Amazon S3, Amazon RDS, etc.

AWS Glue job bookmarks are a feature that helps AWS Glue track data that has already been processed during a previous run of an ETL job. This can prevent the reprocessing of old data and enable the processing of new data when rerunning on a scheduled interval².

Job bookmarks can handle both baseline data and incremental data from different sources.

Amazon S3 is a highly scalable, durable, and secure object storage service that can store any amount and type of data³. It can be used as a data lake to store the merged and processed datasets from AWS Glue. It can also integrate with other AWS services, such as Amazon Athena, Amazon Redshift Spectrum, Amazon EMR, etc., for further analysis and processing.

Question 3

Question Type: MultipleChoice

A financial company uses Amazon Athena to query data from an Amazon S3 data lake. Files are stored in the S3 data lake in Apache ORC format. Data analysts recently introduced nested fields in the data lake ORC files, and noticed that queries are taking longer to run in Athen

a. A data analysts discovered that more data than what is required is being scanned for the queries.

What is the MOST operationally efficient solution to improve query performance?

Options:

A- Flatten nested data and create separate files for each nested dataset.

- B-** Use the Athena query engine V2 and push the query filter to the source ORC file.
- C-** Use Apache Parquet format instead of ORC format.
- D-** Recreate the data partition strategy and further narrow down the data filter criteria.

Answer:

B

Explanation:

This solution meets the requirement because:

The Athena query engine V2 is a new version of the Athena query engine that introduces several improvements and new features, such as federated queries, geospatial functions, prepared statements, schema evolution support, and more¹.

One of the improvements of the Athena query engine V2 is that it supports predicate pushdown for nested fields in ORC files. Predicate pushdown is a technique that allows filtering data at the source before it is scanned and loaded into memory. This can reduce the amount of data scanned and processed by Athena, which can improve query performance and reduce cost².

By using the Athena query engine V2 and pushing the query filter to the source ORC file, the data analysts can leverage the predicate pushdown feature for nested fields and avoid scanning more data than what is required for the queries. This can improve query performance without changing the data format or partitioning strategy.

Question 4

Question Type: MultipleChoice

A machinery company wants to collect data from sensors. A data analytics specialist needs to implement a solution that aggregates the data in near-real time and saves the data to a persistent data store. The data must be stored in nested JSON format and must be queried from the data store with a latency of single-digit milliseconds.

Which solution will meet these requirements?

Options:

- A-** Use Amazon Kinesis Data Streams to receive the data from the sensors. Use Amazon Kinesis Data Analytics to read the stream, aggregate the data, and send the data to an AWS Lambda function. Configure the Lambda function to store the data in Amazon DynamoDB.
- B-** Use Amazon Kinesis Data Firehose to receive the data from the sensors. Use Amazon Kinesis Data Analytics to aggregate the data. Use an AWS Lambda function to read the data from Kinesis Data Analytics and store the data in Amazon S3.
- C-** Use Amazon Kinesis Data Firehose to receive the data from the sensors. Use an AWS Lambda function to aggregate the data during capture. Store the data from Kinesis Data Firehose in Amazon DynamoDB.
- D-** Use Amazon Kinesis Data Firehose to receive the data from the sensors. Use an AWS Lambda function to aggregate the data during capture. Store the data in Amazon S3.

Answer:

C

Explanation:

This solution meets the requirements because:

Amazon Kinesis Data Firehose is a fully managed service that can capture, transform, and load streaming data into AWS data stores, such as Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Amazon DynamoDB¹. It can receive data from sensors and other sources and deliver it to a destination with near-real time latency.

AWS Lambda is a serverless compute service that can run code in response to events and automatically manage the underlying compute resources². It can be used to perform custom transformations on the data during capture by Kinesis Data Firehose³. It can aggregate the data according to the desired logic and output format.

Amazon DynamoDB is a fully managed NoSQL database service that supports key-value and document data models⁴. It can store nested JSON data as document attributes and provide single-digit millisecond latency for queries. It can be used as a persistent data store for the aggregated sensor data.

Question 5

Question Type: MultipleChoice

An event ticketing website has a data lake on Amazon S3 and a data warehouse on Amazon Redshift. Two datasets exist: events data and sales data. Each dataset has millions of records.

The entire events dataset is frequently accessed and is stored in Amazon Redshift. However, only the last 6 months of sales data is frequently accessed and is stored in Amazon Redshift. The rest of the sales data is available only in Amazon S3.

A data analytics specialist must create a report that shows the total revenue that each event has generated in the last 12 months. The report will be accessed thousands of times each week.

Which solution will meet these requirements with the LEAST operational effort?

Options:

- A-** Create an AWS Glue job to access sales data that is older than 6 months from Amazon S3 and to access event and sales data from Amazon Redshift. Load the results into a new table in Amazon Redshift.
- B-** Create a stored procedure to copy sales data that is older than 6 months and newer than 12 months from Amazon S3 to Amazon Redshift. Create a materialized view with the autorefresh option
- C-** Create an AWS Lambda function to copy sales data that is older than 6 months and newer than 12 months to an Amazon Kinesis Data Firehose delivery stream. Specify Amazon Redshift as the destination of the delivery stream. Create a materialized view with the autorefresh option.
- D-** Create a materialized view in Amazon Redshift with the autorefresh option. Use Amazon Redshift Spectrum to include sales data that is older than 6 months.

Answer:

D

Explanation:

This solution meets the requirements because:

A materialized view is a database object that contains the results of a query. It can be used to improve query performance and reduce data processing costs by caching the query results and refreshing them periodically¹.

The autorefresh option enables Amazon Redshift to automatically refresh materialized views with up-to-date data from its base tables when materialized views are created with or altered to have this option. Amazon Redshift autorefreshes materialized views as soon as possible after base tables change².

Amazon Redshift Spectrum enables you to use your existing Amazon Redshift SQL queries to analyze data that is stored in Amazon S3. You can create external tables in your Amazon Redshift cluster and join them with other tables, including materialized views³.

By creating a materialized view in Amazon Redshift with the autorefresh option, the data analytics specialist can precompute and cache the report query results and keep them updated automatically. This can improve the report performance and reduce the load on the Amazon Redshift cluster.

By using Amazon Redshift Spectrum to include sales data that is older than 6 months, the data analytics specialist can access the data that is stored in Amazon S3 without loading it into Amazon Redshift. This can reduce the storage costs and avoid data duplication.

Question 6

Question Type: MultipleChoice

A company plans to store quarterly financial statements in a dedicated Amazon S3 bucket. The financial statements must not be modified or deleted after they are saved to the S3 bucket.

Which solution will meet these requirements?

Options:

- A-** Create the S3 bucket with S3 Object Lock in governance mode.
- B-** Create the S3 bucket with MFA delete enabled.
- C-** Create the S3 bucket with S3 Object Lock in compliance mode.
- D-** Create S3 buckets in two AWS Regions. Use S3 Cross-Region Replication (CRR) between the buckets.

Answer:

A

Explanation:

This solution meets the requirements because:

S3 Object Lock is a feature in Amazon S3 that allows users and businesses to store files in a highly secure, tamper-proof way. It's used for situations in which businesses must be able to prove that data has not been modified or destroyed after it was written, and it relies on a model known as write once, read many (WORM)¹.

S3 Object Lock provides two ways to manage object retention: retention periods and legal holds. A retention period specifies a fixed period of time during which an object remains locked. A legal hold provides the same protection as a retention period, but it has no expiration date².

S3 Object Lock has two retention modes: governance mode and compliance mode. Governance mode allows users with specific IAM permissions to overwrite or delete an object version before its retention period expires. Compliance mode prevents anyone, including the root user of the account that owns the bucket, from overwriting or deleting an object version or altering its lock settings until the retention period expires².

By creating the S3 bucket with S3 Object Lock in compliance mode, the company can ensure that the quarterly financial statements are stored in a WORM model and cannot be modified or deleted by anyone until the retention period expires or the legal hold is removed. This can help meet regulatory requirements that require WORM storage, or to add another layer of protection against object changes and deletion².

Question 7

Question Type: MultipleChoice

A company collects data from parking garages. Analysts have requested the ability to run reports in near real time about the number of vehicles in each garage.

The company wants to build an ingestion pipeline that loads the data into an Amazon Redshift cluster. The solution must alert operations personnel when the number of vehicles in a particular garage exceeds a specific threshold. The alerting query will use garage threshold values as a static reference. The threshold values are stored in

Amazon S3.

What is the MOST operationally efficient solution that meets these requirements?

Options:

A- Use an Amazon Kinesis Data Firehose delivery stream to collect the data and to deliver the data to Amazon Redshift. Create an Amazon Kinesis Data Analytics application that uses the same delivery stream as an input source. Create a reference data source in Kinesis Data Analytics to temporarily store the threshold values from Amazon S3 and to compare the number of vehicles in a particular garage to the corresponding threshold value. Configure an AWS Lambda function to publish an Amazon Simple Notification Service (Amazon SNS) notification if the number of vehicles exceeds the threshold.

B- Use an Amazon Kinesis data stream to collect the data. Use an Amazon Kinesis Data Firehose delivery stream to deliver the data to Amazon Redshift. Create another Kinesis data stream to temporarily store the threshold values from Amazon S3. Send the delivery stream and the second data stream to Amazon Kinesis Data Analytics to compare the number of vehicles in a particular garage to the corresponding threshold value. Configure an AWS Lambda function to publish an Amazon Simple Notification Service (Amazon SNS) notification if the number of vehicles exceeds the threshold.

C- Use an Amazon Kinesis Data Firehose delivery stream to collect the data and to deliver the data to Amazon Redshift. Automatically initiate an AWS Lambda function that queries the data in Amazon Redshift. Configure the Lambda function to compare the number of vehicles in a particular garage to the corresponding threshold value from Amazon S3. Configure the Lambda function to also publish an Amazon Simple Notification Service (Amazon SNS) notification if the number of vehicles exceeds the threshold.

D- Use an Amazon Kinesis Data Firehose delivery stream to collect the data and to deliver the data to Amazon Redshift. Create an Amazon Kinesis Data Analytics application that uses the same delivery stream as an input source. Use Kinesis Data Analytics to compare the number of vehicles in a particular garage to the corresponding threshold value that is stored in a table as an in-application stream. Configure an AWS Lambda function as an output for the application to publish an Amazon Simple Queue Service (Amazon SQS) notification if the number of vehicles exceeds the threshold.

Answer:

A

Explanation:

This solution meets the requirements because:

It uses Amazon Kinesis Data Firehose to collect and deliver data to Amazon Redshift in near real time, without requiring any coding or server management¹.

It uses Amazon Kinesis Data Analytics to process and analyze streaming data using SQL queries or Apache Flink applications². It can also create a reference data source that allows joining streaming data with static data stored in Amazon S3. This way, it can compare the number of vehicles in each garage with the corresponding threshold value from the reference data source.

It uses AWS Lambda to create a serverless function that can be triggered by Kinesis Data Analytics as an output destination⁴. The Lambda function can then publish an Amazon SNS notification to alert operations personnel when the number of vehicles exceeds the threshold⁵.

To Get Premium Files for DAS-C01 Visit

<https://www.p2pexams.com/products/das-c01>

For More Free Questions Visit

<https://www.p2pexams.com/amazon/pdf/das-c01>

