DEA-C01 Q&As

SnowPro Advanced: Data Engineer Certification Exam

Pass Snowflake DEA-C01 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/dea-c01.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Snowflake Official Exam Center

QUESTION 1

A data engineer is configuring an AWS Glue job to read data from an Amazon S3 bucket. The data engineer has set up the necessary AWS Glue connection details and an associated IAM role. However, when the data engineer attempts to run the AWS Glue job, the data engineer receives an error message that indicates that there are problems with the Amazon S3 VPC gateway endpoint.

The data engineer must resolve the error and connect the AWS Glue job to the S3 bucket.

Which solution will meet this requirement?

A. Update the AWS Glue security group to allow inbound traffic from the Amazon S3 VPC gateway endpoint.

B. Configure an S3 bucket policy to explicitly grant the AWS Glue job permissions to access the S3 bucket.

C. Review the AWS Glue job code to ensure that the AWS Glue connection details include a fully qualified domain name.

D. Verify that the VPC\\'s route table includes inbound and outbound routes for the Amazon S3 VPC gateway endpoint.

Correct Answer: D

Explanation: The error message indicates that the AWS Glue job cannot access the Amazon S3 bucket through the VPC endpoint. This could be because the VPC\\'s route table does not have the necessary routes to direct the traffic to the endpoint. To fix this, the data engineer must verify that the route table has an entry for the Amazon S3 service prefix (com.amazonaws.region.s3) with the target as the VPC endpoint ID. This will allow the AWS Glue job to use the VPC endpoint to access the S3 bucket without going through the internet or a NAT gateway. For more information, see Gateway endpoints. References: Troubleshoot the AWS Glue error "VPC S3 endpoint validation failed" Amazon VPC endpoints for Amazon S3 [AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide]

QUESTION 2

A company stores data from an application in an Amazon DynamoDB table that operates in provisioned capacity mode. The workloads of the application have predictable throughput load on a regular schedule. Every Monday, there is an immediate increase in activity early in the morning. The application has very low usage during weekends.

The company must ensure that the application performs consistently during peak usage times. Which solution will meet these requirements in the MOST cost-effective way?

A. Increase the provisioned capacity to the maximum capacity that is currently present during peak load times.

B. Divide the table into two tables. Provision each table with half of the provisioned capacity of the original table. Spread queries evenly across both tables.

C. Use AWS Application Auto Scaling to schedule higher provisioned capacity for peak usage times. Schedule lower capacity during off-peak times.

D. Change the capacity mode from provisioned to on-demand. Configure the table to scale up and scale down based on the load on the table.

Correct Answer: C

Explanation: Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB offers two capacity modes for throughput capacity: provisioned and on-

demand. In provisioned capacity mode, you specify the number of read and write capacity units per second that you expect your application to require. DynamoDB reserves the resources to meet your throughput needs with consistent

performance. In on-demand capacity mode, you pay per request and DynamoDB scales the resources up and down automatically based on the actual workload. On-demand capacity mode is suitable for unpredictable workloads that can vary

significantly over time1.

The solution that meets the requirements in the most cost-effective way is to use AWS Application Auto Scaling to schedule higher provisioned capacity for peak usage times and lower capacity during off-peak times. This solution has the

following advantages:

It allows you to optimize the cost and performance of your DynamoDB table by adjusting the provisioned capacity according to your predictable workload patterns. You can use scheduled scaling to specify the date and time for the scaling

actions, and the new minimum and maximum capacity limits. For example, you can schedule higher capacity for every Monday morning and lower capacity for weekends2.

It enables you to take advantage of the lower cost per unit of provisioned capacity mode compared to on-demand capacity mode. Provisioned capacity mode charges a flat hourly rate for the capacity you reserve, regardless of how much you

use. On-demand capacity mode charges for each read and write request you consume, with nominimum capacity required. For predictable workloads, provisioned capacity mode can be more cost-effective than on-demand capacity mode1.

It ensures that your application performs consistently during peak usage times by having enough capacity to handle the increased load. You can also use auto scaling to automatically adjust the provisioned capacity based on the actual

utilization of your table, and set a target utilization percentage for your table or global secondary index. This way, you can avoid under-provisioning or over- provisioning your table2.

Option A is incorrect because it suggests increasing the provisioned capacity to the maximum capacity that is currently present during peak load times. This solution has the following disadvantages:

It wastes money by paying for unused capacity during off-peak times. If you provision the same high capacity for all times, regardless of the actual workload, you are over-provisioning your table and paying for resources that you don\\'t need1.

It does not account for possible changes in the workload patterns over time. If your peak load times increase or decrease in the future, you may need to manually adjust the provisioned capacity to match the new demand. This adds

operational overhead and complexity to your application2.

Option B is incorrect because it suggests dividing the table into two tables and provisioning each table with half of the provisioned capacity of the original table. This solution has the following disadvantages:

It complicates the data model and the application logic by splitting the data into two separate tables. You need to ensure that the queries are evenly distributed across both tables, and that the data is consistent and synchronized between them. This adds extra development and maintenance effort to your application3. It does not solve the problem of adjusting the provisioned capacity according to the workload patterns. You still need to manually or automatically scale the

capacity of each table based on the actual utilization and demand. This may result in under- provisioning or over-provisioning your tables2.

Option D is incorrect because it suggests changing the capacity mode from provisioned to on-demand. This solution has the following disadvantages:

It may incur higher costs than provisioned capacity mode for predictable workloads. On-demand capacity mode charges for each read and write request you consume, with no minimum capacity required. For predictable workloads,

provisioned capacity mode can be more cost-effective than on-demand capacity mode, as you can reserve the capacity you need at a lower rate1. It may not provide consistent performance during peak usage times, as on- demand capacity

mode may take some time to scale up the resources to meet the sudden increase in demand. On-demand capacity mode uses adaptive capacity to handle bursts of traffic, but it may not be able to handle very large spikes or sustained high

throughput. In such cases, you may experience throttling or increased latency.

References:

1: Choosing the right DynamoDB capacity mode - Amazon DynamoDB

2: Managing throughput capacity automatically with DynamoDB auto scaling - Amazon DynamoDB

3: Best practices for designing and using partition keys effectively - Amazon DynamoDB [4]: On-demand mode guidelines - Amazon DynamoDB [5]: How to optimize Amazon DynamoDB costs - AWS Database Blog [6]: DynamoDB adaptive capacity: How it works and how it helps - AWS Database Blog [7]: Amazon DynamoDB pricing - Amazon Web Services (AWS)

QUESTION 3

A manufacturing company collects sensor data from its factory floor to monitor and enhance operational efficiency. The company uses Amazon Kinesis Data Streams to publish the data that the sensors collect to a data stream. Then Amazon Kinesis Data Firehose writes the data to an Amazon S3 bucket.

The company needs to display a real-time view of operational efficiency on a large screen in the manufacturing facility.

Which solution will meet these requirements with the LOWEST latency?

A. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Use a connector for Apache Flink to write data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard.

B. Configure the S3 bucket to send a notification to an AWS Lambda function when any new object is created. Use the Lambda function to publish the data to Amazon Aurora. Use Aurora as a source to create an Amazon QuickSight dashboard.

C. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Create a new Data Firehose delivery stream to publish data directly to an Amazon Timestream database. Use the Timestream database as a source to create an Amazon QuickSight dashboard.

D. Use AWS Glue bookmarks to read sensor data from the S3 bucket in real time. Publish the data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard.

Correct Answer: C

Explanation: This solution will meet the requirements with the lowest latency because it uses Amazon Managed Service for Apache Flink to process the sensor data in real time and write it to Amazon Timestream, a fast, scalable, and

serverless time series database. Amazon Timestream is optimized for storing and analyzing time series data, such as sensor data, and can handle trillions of events per day with millisecond latency. By using AmazonTimestream as a source,

you can create an Amazon QuickSight dashboard that displays a real-time view of operational efficiency on a large screen in the manufacturing facility. Amazon QuickSight is a fully managed business intelligence service that can connect to

various data sources, including Amazon Timestream, and provide interactive visualizations and insights123.

The other options are not optimal for the following reasons:

A. Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data. Use a connector for Apache Flink to write data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard. This option is similar to option C, but it uses Grafana instead of Amazon QuickSight to create the dashboard. Grafana is an open source visualization tool that can also connect to Amazon Timestream, but it requires additional steps to set up and configure, such as deploying a Grafana server on Amazon EC2, installing the Amazon Timestream plugin, and creating an IAM role for Grafana to access Timestream. These steps can increase the latency and complexity of the solution. B. Configure the S3 bucket to send a notification to an AWS Lambda function when any new object is created. Use the Lambda function to publish the data to Amazon Aurora. Use Aurora as a source to create an Amazon QuickSight dashboard. This option is not suitable for displaying a real-time view of operational efficiency, as it introduces unnecessary delays and costs in the data pipeline. First, the sensor data is written to an S3 bucket by Amazon Kinesis Data Firehose, which can have a buffering interval of up to 900 seconds. Then, the S3 bucket sends a notification to a Lambda function, which can incur additional invocation and execution time. Finally, the Lambda function publishes the data to Amazon Aurora, a relational database that is not optimized for time series data and can have higher storage and performance costs than Amazon Timestream . D. Use AWS Glue bookmarks to read sensor data from the S3 bucket in real time. Publish the data to an Amazon Timestream database. Use the Timestream database as a source to create a Grafana dashboard. This option is also not suitable for displaying a real-time view of operational efficiency, as it uses AWS Glue bookmarks to read sensor data from the S3 bucket. AWS Glue bookmarks are a feature that helps AWS Glue jobs and crawlers keep track of the data that has already been processed, so that they can resume from where they left off. However, AWS Glue jobs and crawlers are not designed for real-time data processing, as they can have a minimum frequency of 5 minutes and a variable start-up time. Moreover, this option also uses Grafana instead of Amazon QuickSight to create the dashboard, which can increase the latency and complexity of the solution . References:

1: Amazon Managed Streaming for Apache Flink

2: Amazon Timestream

3: Amazon QuickSight : Analyze data in Amazon Timestream using Grafana : Amazon Kinesis Data Firehose : Amazon Aurora : AWS Glue Bookmarks : AWS Glue Job and Crawler Scheduling

QUESTION 4

A company maintains multiple extract, transform, and load (ETL) workflows that ingest data from the company\\'s operational databases into an Amazon S3 based data lake. The ETL workflows use AWS Glue and Amazon EMR to process data.

The company wants to improve the existing architecture to provide automated orchestration and to require minimal manual effort.

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

A. AWS Glue workflows

B. AWS Step Functions tasks

C. AWS Lambda functions

D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA) workflows

Correct Answer: A

Explanation: AWS Glue workflows are a feature of AWS Glue that enable you to create and visualize complex ETL pipelines using AWS Glue components, such as crawlers, jobs, triggers, anddevelopment endpoints. AWS Glue workflows provide automated orchestration and require minimal manual effort, as they handle dependency resolution, error handling, state management, and resource allocation for your ETL workflows. You can use AWS Glue workflows to ingest data from your operational databases into your Amazon S3 based data lake, and then use AWS Glue and Amazon EMR to process the data in the data lake. This solution will meet the requirements with the least operational overhead, as it leverages the serverless and fully managed nature of AWS Glue, and the scalability and flexibility of Amazon EMR12. The other options are not optimal for the following reasons:

B. AWS Step Functions tasks. AWS Step Functions is a service that lets you coordinate multiple AWS services into serverless workflows. You can use AWS Step Functions tasks to invoke AWS Glue and Amazon EMR jobs as part of your ETL workflows, and use AWS Step Functions state machines to define the logic and flow of your workflows. However, this option would require more manual effort than AWS Glue workflows, as you would need to write JSON code to define your state machines, handle errors and retries, and monitor the execution history and status of your workflows3.

C. AWS Lambda functions. AWS Lambda is a service that lets you run code without provisioning or managing servers. You can use AWS Lambda functions to trigger AWS Glue and Amazon EMR jobs as part of your ETL workflows, and use AWS Lambda event sources and destinations to orchestrate the flow of your workflows. However, this option would also require more manual effort than AWS Glue workflows, as you would need to write code to implement your business logic, handle errors and retries, and monitor the invocation and execution of your Lambda functions. Moreover, AWS Lambda functions have limitations on the execution time, memory, and concurrency, which may affect the performance and scalability of your ETL workflows.

D. Amazon Managed Workflows for Apache Airflow (Amazon MWAA) workflows. Amazon MWAA is a managed service that makes it easy to run open source Apache Airflow on AWS. Apache Airflow is a popular tool for creating and managing complex ETL pipelines using directed acyclic graphs (DAGs). You can use Amazon MWAA workflows to orchestrate AWS Glue and Amazon EMR jobs as part of your ETL workflows, and use the Airflow web interface to visualize and monitor your workflows. However, this option would have more operational overhead than AWS Glue workflows, as you would need to set up and configure your Amazon MWAA environment, write Python code to define your DAGs, and manage the dependencies and versions of your Airflow plugins and operators. References:

1: AWS Glue Workflows

2: AWS Glue and Amazon EMR

3: AWS Step Functions : AWS Lambda : Amazon Managed Workflows for Apache Airflow

QUESTION 5

A company uses Amazon Athena for one-time queries against data that is in Amazon S3. The company has several use cases. The company must implement permission controls to separate query processes and access to query history among users, teams, and applications that are in the same AWS account.

Which solution will meet these requirements?

A. Create an S3 bucket for each use case. Create an S3 bucket policy that grants permissions to appropriate individual IAM users. Apply the S3 bucket policy to the S3 bucket.

B. Create an Athena workgroup for each use case. Apply tags to the workgroup. Create an 1AM policy that uses the tags to apply appropriate permissions to the workgroup.

C. Create an JAM role for each use case. Assign appropriate permissions to the role for each use case. Associate the role with Athena.

D. Create an AWS Glue Data Catalog resource policy that grants permissions to appropriate individual IAM users for each use case. Apply the resource policy to the specific tables that Athena uses.

Correct Answer: B

Explanation: Athena workgroups are a way to isolate query execution and query history among users, teams, and applications that share the same AWS account. By creating a workgroup for each use case, the company can control the access and actions on the workgroup resource using resource-level IAM permissions or identity-based IAM policies. The company can also use tags to organize and identify the workgroups, and use them as conditions in the IAM policies to grant or deny permissions to the workgroup. This solution meets the requirements of separating query processes and access to query history among users, teams, and applications that are in the same AWS account. References: Athena Workgroups IAM policies for accessing workgroups Workgroup example policies