Number: DEA-C01
Passing Score: 800
Time Limit: 120
File Version: 3.0

Exam Code: DEA-C01

Exam Name: AWS Certified Data Engineer - Associate

Exam A

QUESTION 1
A company ingests data from multiple data sources and stores the data in an Amazon S3 bucket. An AWS Glue extract, transform, and load (ETL) job transforms the data and writes the transformed data to an Amazon S3 based data lake. The company uses Amazon Athena to query the data that is in the data lake.
The company needs to identify matching records even when the records do not have a common unique identifier.
Which solution will meet this requirement?

A. Use Amazon Made pattern matching as part of the ETL job.
B. Train and use the AWS Glue PySpark Filter class in the ETL job.
C. Partition tables and use the ETL job to partition the data on a unique identifier.
D. Train and use the AWS Lake Formation FindMatches transform in the ETL job.

Correct Answer: D
Section:
Explanation:
The problem described requires identifying matching records even when there is no unique identifier. AWS Lake Formation FindMatches is designed for this purpose. It uses machine learning (ML) to deduplicate and find matching records in datasets that do not share a common identifier.
Alternatives Considered:
A (Amazon Made pattern matching): Amazon Made is not a service in AWS, and pattern matching typically refers to regular expressions, which are not suitable for deduplication without a common identifier.
B (AWS Glue PySpark Filter class): PySpark's Filter class can help refine datasets, but it does not offer the ML-based matching capabilities required to find matches between records without unique identifiers.
C (Partition tables on a unique identifier): Partitioning requires a unique identifier, which the question states is unavailable.
AWS Glue Documentation on Lake Formation FindMatches
FindMatches in AWS Lake Formation
D Train and use the AWS Lake Formation FindMatches transform in the ETL job: FindMatches is a transform available in AWS Lake Formation that uses ML to discover duplicate records or related records that might not have a common unique identifier. It can be integrated into an AWS Glue ETL job to perform deduplication or matching tasks. FindMatches is highly effective in scenarios where records do not share a key, such as customer records from different sources that need to be merged or reconciled.

QUESTION 2
A company stores its processed data in an S3 bucket. The company has a strict data access policy. The company uses IAM roles to grant teams within the company different levels of access to the S3 bucket.
The company wants to receive notifications when a user violates the data access policy. Each notification must include the username of the user who violated the policy.
Which solution will meet these requirements?

A. Use AWS Config rules to detect violations of the data access policy. Set up compliance alarms.
B. Use Amazon CloudWatch metrics to gather object-level metrics. Set up CloudWatch alarms.
C. Use AWS CloudTrail to track object-level events for the S3 bucket. Forward events to Amazon CloudWatch to set up CloudWatch alarms.
D. Use Amazon S3 server access logs to monitor access to the bucket. Forward the access logs to an Amazon CloudWatch log group. Use metric filters on the log group to set up CloudWatch alarms.

Correct Answer: C
Section:
Explanation:
The requirement is to detect violations of data access policies and receive notifications with the username of the violator. AWS CloudTrail can provide object-level tracking for S3 to capture detailed API actions on specific S3 objects, including the user who performed the action.
AWS CloudTrail:
CloudTrail can monitor API calls made to an S3 bucket, including object-level API actions such as GetObject, PutObject, and DeleteObject. This will help detect access violations based on the API calls made by different users.
CloudTrail logs include details such as the user identity, which is essential for meeting the requirement of including the username in notifications.

The CloudTrail logs can be forwarded to Amazon CloudWatch to trigger alarms based on certain access patterns (e.g., violations of specific policies).
Amazon CloudWatch:
By forwarding CloudTrail logs to CloudWatch, you can set up alarms that are triggered when a specific condition is met, such as unauthorized access or policy violations. The alarm can include detailed information from the CloudTrail log, including the username.
Alternatives Considered:
A (AWS Config rules): While AWS Config can track resource configurations and compliance, it does not provide real-time, detailed tracking of object-level events like CloudTrail does.
B (CloudWatch metrics): CloudWatch does not gather object-level metrics for S3 directly. For this use case, CloudTrail provides better granularity.
D (S3 server access logs): S3 server access logs can monitor access, but they do not provide the real-time monitoring and alerting features that CloudTrail with CloudWatch alarms offer. They also do not include API-level granularity like CloudTrail.
AWS CloudTrail Integration with S3
Amazon CloudWatch Alarms

QUESTION 3
A company stores logs in an Amazon S3 bucket. When a data engineer attempts to access several log files, the data engineer discovers that some files have been unintentionally deleted.
The data engineer needs a solution that will prevent unintentional file deletion in the future.
Which solution will meet this requirement with the LEAST operational overhead?

A. Manually back up the S3 bucket on a regular basis.
B. Enable S3 Versioning for the S3 bucket.
C. Configure replication for the S3 bucket.
D. Use an Amazon S3 Glacier storage class to archive the data that is in the S3 bucket.

Correct Answer: B
Section:
Explanation:
To prevent unintentional file deletions and meet the requirement with minimal operational overhead, enabling S3 Versioning is the best solution.
S3 Versioning:
S3 Versioning allows multiple versions of an object to be stored in the same S3 bucket. When a file is deleted or overwritten, S3 preserves the previous versions, which means you can recover from accidental deletions or modifications.
Enabling versioning requires minimal overhead, as it is a bucket-level setting and does not require additional backup processes or data replication.
Users can recover specific versions of files that were unintentionally deleted, meeting the needs of the data engineer to avoid accidental data loss.
Alternatives Considered:
A (Manual backups): Manually backing up the bucket requires higher operational effort and maintenance compared to enabling S3 Versioning, which is automated.
C (S3 Replication): Replication ensures data is copied to another bucket but does not provide protection against accidental deletion. It would increase operational costs without solving the core issue of accidental deletion.
D (S3 Glacier): Storing data in Glacier provides long-term archival storage but is not designed to prevent accidental deletion. Glacier is also more suitable for archival and infrequently accessed data, not for active logs.
Amazon S3 Versioning Documentation
S3 Data Protection Best Practices

QUESTION 4
A company currently uses a provisioned Amazon EMR cluster that includes general purpose Amazon EC2 instances. The EMR cluster uses EMR managed scaling between one to five task nodes for the company's long-running Apache Spark extract, transform, and load (ETL) job. The company runs the ETL job every day.
When the company runs the ETL job, the EMR cluster quickly scales up to five nodes. The EMR cluster often reaches maximum CPU usage, but the memory usage remains under 30%.
The company wants to modify the EMR cluster configuration to reduce the EMR costs to run the daily ETL job.
Which solution will meet these requirements MOST cost-effectively?

A. Increase the maximum number of task nodes for EMR managed scaling to 10.
B. Change the task node type from general purpose EC2 instances to memory optimized EC2 instances.
C. Switch the task node type from general purpose EC2 instances to compute optimized EC2 instances.
D. Reduce the scaling cooldown period for the provisioned EMR cluster.

Correct Answer: C
Section:
Explanation:
The company's Apache Spark ETL job on Amazon EMR uses high CPU but low memory, meaning that compute-optimized EC2 instances would be the most cost-effective choice. These instances are designed for high-performance compute applications, where CPU usage is high, but memory needs are minimal, which is exactly the case here.
Compute Optimized Instances:
Compute-optimized instances, such as the C5 series, provide a higher ratio of CPU to memory, which is more suitable for jobs with high CPU usage and relatively low memory consumption.
Switching from general-purpose EC2 instances to compute-optimized instances can reduce costs while improving performance, as these instances are optimized for workloads like Spark jobs that perform a lot of computation.
Managed Scaling: The EMR cluster's scaling is currently managed between 1 and 5 nodes, so changing the instance type will leverage the current scaling strategy but optimize it for the workload.
Alternatives Considered:
A (Increase task nodes to 10): Increasing the number of task nodes would increase costs without necessarily improving performance. Since memory usage is low, the bottleneck is more likely the CPU, which compute-optimized instances can handle better.
B (Memory optimized instances): Memory-optimized instances are not suitable since the current job is CPU-bound, and memory usage remains low (under 30%).
D (Reduce scaling cooldown): This could marginally improve scaling speed but does not address the need for cost optimization and improved CPU performance.
Amazon EMR Cluster Optimization
Compute Optimized EC2 Instances

QUESTION 5
A company is building a data stream processing application. The application runs in an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. The application stores processed data in an Amazon DynamoDB table.
The company needs the application containers in the EKS cluster to have secure access to the DynamoDB table. The company does not want to embed AWS credentials in the containers.
Which solution will meet these requirements?

A. Store the AWS credentials in an Amazon S3 bucket. Grant the EKS containers access to the S3 bucket to retrieve the credentials.
B. Attach an IAM role to the EKS worker nodes. Grant the IAM role access to DynamoDB. Use the IAM role to set up IAM roles service accounts (IRSA) functionality.
C. Create an IAM user that has an access key to access the DynamoDB table. Use environment variables in the EKS containers to store the IAM user access key data.
D. Create an IAM user that has an access key to access the DynamoDB table. Use Kubernetes secrets that are mounted in a volume of the EKS cluster nodes to store the user access key data.

Correct Answer: B
Section:
Explanation:
In this scenario, the company is using Amazon Elastic Kubernetes Service (EKS) and wants secure access to DynamoDB without embedding credentials inside the application containers. The best practice is to use IAM roles for service accounts (IRSA), which allows assigning IAM roles to Kubernetes service accounts. This lets the EKS pods assume specific IAM roles securely, without the need to store credentials in containers.
IAM Roles for Service Accounts (IRSA):
With IRSA, each pod in the EKS cluster can assume an IAM role that grants access to DynamoDB without needing to manage long-term credentials. The IAM role can be attached to the service account associated with the pod. This ensures least privilege access, improving security by preventing credentials from being embedded in the containers.
Alternatives Considered:
A (Storing AWS credentials in S3): Storing AWS credentials in S3 and retrieving them introduces security risks and violates the principle of not embedding credentials.
C (IAM user access keys in environment variables): This also embeds credentials, which is not recommended.
D (Kubernetes secrets): Storing user access keys as secrets is an option, but it still involves handling long-term credentials manually, which is less secure than using IRSA.
IAM Best Practices for Amazon EKS
Secure Access to DynamoDB from EKS

QUESTION 6
A company is migrating its database servers from Amazon EC2 instances that run Microsoft SQL Server to Amazon RDS for Microsoft SQL Server DB instances. The company's analytics team must export large data elements every day until the migration is complete. The data elements are the result of SQL joins across multiple tables. The data must be in Apache Parquet format. The analytics team must store the data in Amazon S3.
Which solution will meet these requirements in the MOST operationally efficient way?

A. Create a view in the EC2 instance-based SQL Server databases that contains the required data elements. Create an AWS Glue job that selects the data directly from the view and transfers the data in Parquet format to an S3 bucket. Schedule the AWS Glue job to run every day.

B. Schedule SQL Server Agent to run a daily SQL query that selects the desired data elements from the EC2 instance-based SQL Server databases. Configure the query to direct the output .csv objects to an S3 bucket. Create an S3 event that invokes an AWS Lambda function to transform the output format from .csv to Parquet.

C. Use a SQL query to create a view in the EC2 instance-based SQL Server databases that contains the required data elements. Create and run an AWS Glue crawler to read the view. Create an AWS Glue job that retrieves the data and transfers the data in Parquet format to an S3 bucket. Schedule the AWS Glue job to run every day.

D. Create an AWS Lambda function that queries the EC2 instance-based databases by using Java Database Connectivity (JDBC). Configure the Lambda function to retrieve the required data, transform the data into Parquet format, and transfer the data into an S3 bucket. Use Amazon EventBridge to schedule the Lambda function to run every day.

Correct Answer: A
Section:
Explanation:
Option A is the most operationally efficient way to meet the requirements because it minimizes the number of steps and services involved in the data export process. AWS Glue is a fully managed service that can extract, transform, and load (ETL) data from various sources to various destinations, including Amazon S3. AWS Glue can also convert data to different formats, such as Parquet, which is a columnar storage format that is optimized for analytics. By creating a view in the SQL Server databases that contains the required data elements, the AWS Glue job can select the data directly from the view without having to perform any joins or transformations on the source data. The AWS Glue job can then transfer the data in Parquet format to an S3 bucket and run on a daily schedule.
Option B is not operationally efficient because it involves multiple steps and services to export the data. SQL Server Agent is a tool that can run scheduled tasks on SQL Server databases, such as executing SQL queries. However, SQL Server Agent cannot directly export data to S3, so the query output must be saved as .csv objects on the EC2 instance. Then, an S3 event must be configured to trigger an AWS Lambda function that can transform the .csv objects to Parquet format and upload them to S3. This option adds complexity and latency to the data export process and requires additional resources and configuration.
Option C is not operationally efficient because it introduces an unnecessary step of running an AWS Glue crawler to read the view. An AWS Glue crawler is a service that can scan data sources and create metadata tables in the AWS Glue Data Catalog. The Data Catalog is a central repository that stores information about the data sources, such as schema, format, and location. However, in this scenario, the schema and format of the data elements are already known and fixed, so there is no need to run a crawler to discover them. The AWS Glue job can directly select the data from the view without using the Data Catalog. Running a crawler adds extra time and cost to the data export process.
Option D is not operationally efficient because it requires custom code and configuration to query the databases and transform the data. An AWS Lambda function is a service that can run code in response to events or triggers, such as Amazon EventBridge. Amazon EventBridge is a service that can connect applications and services with event sources, such as schedules, and route them to targets, such as Lambda functions. However, in this scenario, using a Lambda function to query the databases and transform the data is not the best option because it requires writing and maintaining code that uses JDBC to connect to the SQL Server databases, retrieve the required data, convert the data to Parquet format, and transfer the data to S3. This option also has limitations on the execution time, memory, and concurrency of the Lambda function, which may affect the performance and reliability of the data export process.
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide
AWS Glue Documentation
Working with Views in AWS Glue
Converting to Columnar Formats

QUESTION 7
A data engineering team is using an Amazon Redshift data warehouse for operational reporting. The team wants to prevent performance issues that might result from long- running queries. A data engineer must choose a system table in Amazon Redshift to record anomalies when a query optimizer identifies conditions that might indicate performance issues.
Which table views should the data engineer use to meet this requirement?

A. STL USAGE CONTROL
B. STL ALERT EVENT LOG
C. STL QUERY METRICS
D. STL PLAN INFO

Correct Answer: B
Section:
Explanation:
The STL ALERT EVENT LOG table view records anomalies when the query optimizer identifies conditions that might indicate performance issues. These conditions include skewed data distribution, missing statistics, nested loop joins, and broadcasted data. The STL ALERT EVENT LOG table view can help the data engineer to identify and troubleshoot the root causes of performance issues and optimize the query execution plan. The other table views are not relevant for this requirement. STL USAGE CONTROL records the usage limits and quotas for Amazon Redshift resources. STL QUERY METRICS records the execution time and resource consumption of queries. STL PLAN INFO records the query execution plan and the steps involved in each query.Reference:
STL ALERT EVENT LOG

QUESTION 8
A data engineer must ingest a source of structured data that is in .csv format into an Amazon S3 data lake. The .csv files contain 15 columns. Data analysts need to run Amazon Athena queries on one or two columns of the dataset. The data analysts rarely query the entire file.
Which solution will meet these requirements MOST cost-effectively?

A. Use an AWS Glue PySpark job to ingest the source data into the data lake in .csv format.
B. Create an AWS Glue extract, transform, and load (ETL) job to read from the .csv structured data source. Configure the job to ingest the data into the data lake in JSON format.
C. Use an AWS Glue PySpark job to ingest the source data into the data lake in Apache Avro format.
D. Create an AWS Glue extract, transform, and load (ETL) job to read from the .csv structured data source. Configure the job to write the data into the data lake in Apache Parquet format.

Correct Answer: D
Section:
Explanation:
Amazon Athena is a serverless interactive query service that allows you to analyze data in Amazon S3 using standard SQL. Athena supports various data formats, such as CSV, JSON, ORC, Avro, and Parquet. However, not all data formats are equally efficient for querying. Some data formats, such as CSV and JSON, are row-oriented, meaning that they store data as a sequence of records, each with the same fields. Row-oriented formats are suitable for loading and exporting data, but they are not optimal for analytical queries that often access only a subset of columns. Row-oriented formats also do not support compression or encoding techniques that can reduce the data size and improve the query performance. On the other hand, some data formats, such as ORC and Parquet, are column-oriented, meaning that they store data as a collection of columns, each with a specific data type. Column-oriented formats are ideal for analytical queries that often filter, aggregate, or join data by columns. Column-oriented formats also support compression and encoding techniques that can reduce the data size and improve the query performance. For example, Parquet supports dictionary encoding, which replaces repeated values with numeric codes, and run-length encoding, which replaces consecutive identical values with a single value and a count. Parquet also supports various compression algorithms, such as Snappy, GZIP, and ZSTD, that can further reduce the data size and improve the query performance. Therefore, creating an AWS Glue extract, transform, and load (ETL) job to read from the .csv structured data source and writing the data into the data lake in Apache Parquet format will meet the requirements most cost-effectively. AWS Glue is a fully managed service that provides a serverless data integration platform for data preparation, data cataloging, and data loading. AWS Glue ETL jobs allow you to transform and load data from various sources into various targets, using either a graphical interface (AWS Glue Studio) or a code-based interface (AWS Glue console or AWS Glue API). By using AWS Glue ETL jobs, you can easily convert the data from CSV to Parquet format, without having to write or manage any code. Parquet is a column-oriented format that allows Athena to scan only the relevant columns and skip the rest, reducing the amount of data read from S3. This solution will also reduce the cost of Athena queries, as Athena charges based on the amount of data scanned from S3. The other options are not as cost-effective as creating an AWS Glue ETL job to write the data into the data lake in Parquet format. Using an AWS Glue PySpark job to ingest the source data into the data lake in .csv format will not improve the query performance or reduce the query cost, as .csv is a row-oriented format that does not support columnar access or compression. Creating an AWS Glue ETL job to ingest the data into the data lake in JSON format will not improve the query performance or reduce the query cost, as JSON is also a row-oriented format that does not support columnar access or compression. Using an AWS Glue PySpark job to ingest the source data into the data lake in Apache Avro format will improve the query performance, as Avro is a column-oriented format that supports compression and encoding, but it will require more operational effort, as you will need to write and maintain PySpark code to convert the data from CSV to Avro format.Reference: Amazon Athena Choosing the Right Data Format AWS Glue [AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide], Chapter 5: Data Analysis and Visualization, Section 5.1: Amazon Athena

QUESTION 9
A company has five offices in different AWS Regions. Each office has its own human resources (HR) department that uses a unique IAM role. The company stores employee records in a data lake that is based on Amazon S3 storage.
A data engineering team needs to limit access to the records. Each HR department should be able to access records for only employees who are within the HR department's Region.
Which combination of steps should the data engineering team take to meet this requirement with the LEAST operational overhead? (Choose two.)

A. Use data filters for each Region to register the S3 paths as data locations.
B. Register the S3 path as an AWS Lake Formation location.
C. Modify the IAM roles of the HR departments to add a data filter for each department's Region.
D. Enable fine-grained access control in AWS Lake Formation. Add a data filter for each Region.
E. Create a separate S3 bucket for each Region. Configure an IAM policy to allow S3 access. Restrict access based on Region.

Correct Answer: B, D
Section:

Explanation:
AWS Lake Formation is a service that helps you build, secure, and manage data lakes on Amazon S3. You can use AWS Lake Formation to register the S3 path as a data lake location, and enable fine-grained access control to limit access to the records based on the HR department's Region. You can use data filters to specify which S3 prefixes or partitions each HR department can access, and grant permissions to the IAM roles of the HR departments accordingly.This solution will meet the requirement with the least operational overhead, as it simplifies the data lake management and security, and leverages the existing IAM roles of the HR departments12.
The other options are not optimal for the following reasons:
A . Use data filters for each Region to register the S3 paths as data locations. This option is not possible, as data filters are not used to register S3 paths as data locations, but to grant permissions to access specific S3 prefixes or partitions within a data location. Moreover, this option does not specify how to limit access to the records based on the HR department's Region.
C . Modify the IAM roles of the HR departments to add a data filter for each department's Region. This option is not possible, as data filters are not added to IAM roles, but to permissions granted by AWS Lake Formation. Moreover, this option does not specify how to register the S3 path as a data lake location, or how to enable fine-grained access control in AWS Lake Formation.
E . Create a separate S3 bucket for each Region. Configure an IAM policy to allow S3 access. Restrict access based on Region. This option is not recommended, as it would require more operational overhead to create and manage multiple S3 buckets, and to configure and maintain IAM policies for each HR department. Moreover, this option does not leverage the benefits of AWS Lake Formation, such as data cataloging, data transformation, and data governance.
1: AWS Lake Formation
2: AWS Lake Formation Permissions
: AWS Identity and Access Management
: Amazon S3

QUESTION 10
A company uses AWS Step Functions to orchestrate a data pipeline. The pipeline consists of Amazon EMR jobs that ingest data from data sources and store the data in an Amazon S3 bucket. The pipeline also includes EMR jobs that load the data to Amazon Redshift.
The company's cloud infrastructure team manually built a Step Functions state machine. The cloud infrastructure team launched an EMR cluster into a VPC to support the EMR jobs. However, the deployed Step Functions state machine is not able to run the EMR jobs.
Which combination of steps should the company take to identify the reason the Step Functions state machine is not able to run the EMR jobs? (Choose two.)

A. Use AWS CloudFormation to automate the Step Functions state machine deployment. Create a step to pause the state machine during the EMR jobs that fail. Configure the step to wait for a human user to send approval through an email message. Include details of the EMR task in the email message for further analysis.
B. Verify that the Step Functions state machine code has all IAM permissions that are necessary to create and run the EMR jobs. Verify that the Step Functions state machine code also includes IAM permissions to access the Amazon S3 buckets that the EMR jobs use. Use Access Analyzer for S3 to check the S3 access properties.
C. Check for entries in Amazon CloudWatch for the newly created EMR cluster. Change the AWS Step Functions state machine code to use Amazon EMR on EKS. Change the IAM access policies and the security group configuration for the Step Functions state machine code to reflect inclusion of Amazon Elastic Kubernetes Service (Amazon EKS).
D. Query the flow logs for the VPC. Determine whether the traffic that originates from the EMR cluster can successfully reach the data providers. Determine whether any security group that might be attached to the Amazon EMR cluster allows connections to the data source servers on the informed ports.
E. Check the retry scenarios that the company configured for the EMR jobs. Increase the number of seconds in the interval between each EMR task. Validate that each fallback state has the appropriate catch for each decision state. Configure an Amazon Simple Notification Service (Amazon SNS) topic to store the error messages.

Correct Answer: B, D
Section:
Explanation:
To identify the reason why the Step Functions state machine is not able to run the EMR jobs, the company should take the following steps:
Verify that the Step Functions state machine code has all IAM permissions that are necessary to create and run the EMR jobs. The state machine code should have an IAM role that allows it to invoke the EMR APIs, such as RunJobFlow, AddJobFlowSteps, and DescribeStep. The state machine code should also have IAM permissions to access the Amazon S3 buckets that the EMR jobs use as input and output locations.The company can use Access Analyzer for S3 to check the access policies and permissions of the S3 buckets12. Therefore, option B is correct.
Query the flow logs for the VPC. The flow logs can provide information about the network traffic to and from the EMR cluster that is launched in the VPC. The company can use the flow logs to determine whether the traffic that originates from the EMR cluster can successfully reach the data providers, such as Amazon RDS, Amazon Redshift, or other external sources. The company can also determine whether any security group that might be attached to the EMR cluster allows connections to the data source servers on the informed ports.The company can use Amazon VPC Flow Logs or Amazon CloudWatch Logs Insights to query the flow logs3. Therefore, option D is correct.
Option A is incorrect because it suggests using AWS CloudFormation to automate the Step Functions state machine deployment. While this is a good practice to ensure consistency and repeatability of the deployment, it does not help to identify the reason why the state machine is not able to run the EMR jobs. Moreover, creating a step to pause the state machine during the EMR jobs that fail and wait for a human user to send approval through an email message is not a reliable way to troubleshoot the issue. The company should use the Step Functions console or API to monitor the execution history and status of the state machine, and use Amazon CloudWatch to

view the logs and metrics of the EMR jobs .
Option C is incorrect because it suggests changing the AWS Step Functions state machine code to use Amazon EMR on EKS. Amazon EMR on EKS is a service that allows you to run EMR jobs on Amazon Elastic Kubernetes Service (Amazon EKS) clusters. While this service has some benefits, such as lower cost and faster execution time, it does not support all the features and integrations that EMR on EC2 does, such as EMR Notebooks, EMR Studio, and EMRFS. Therefore, changing the state machine code to use EMR on EKS may not be compatible with the existing data pipeline and may introduce new issues.
Option E is incorrect because it suggests checking the retry scenarios that the company configured for the EMR jobs. While this is a good practice to handle transient failures and errors, it does not help to identify the root cause of why the state machine is not able to run the EMR jobs. Moreover, increasing the number of seconds in the interval between each EMR task may not improve the success rate of the jobs, and may increase the execution time and cost of the state machine. Configuring an Amazon SNS topic to store the error messages may help to notify the company of any failures, but it does not provide enough information to troubleshoot the issue.
1: Manage an Amazon EMR Job - AWS Step Functions
2: Access Analyzer for S3 - Amazon Simple Storage Service
3: Working with Amazon EMR and VPC Flow Logs - Amazon EMR
[4]: Analyzing VPC Flow Logs with Amazon CloudWatch Logs Insights - Amazon Virtual Private Cloud
[5]: Monitor AWS Step Functions - AWS Step Functions
[6]: Monitor Amazon EMR clusters - Amazon EMR
[7]: Amazon EMR on Amazon EKS - Amazon EMR

QUESTION 11
A company is developing an application that runs on Amazon EC2 instances. Currently, the data that the application generates is temporary. However, the company needs to persist the data, even if the EC2 instances are terminated.
A data engineer must launch new EC2 instances from an Amazon Machine Image (AMI) and configure the instances to preserve the data.
Which solution will meet this requirement?

A. Launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume that contains the application data. Apply the default settings to the EC2 instances.
B. Launch new EC2 instances by using an AMI that is backed by a root Amazon Elastic Block Store (Amazon EBS) volume that contains the application data. Apply the default settings to the EC2 instances.
C. Launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume. Attach an Amazon Elastic Block Store (Amazon EBS) volume to contain the application data. Apply the default settings to the EC2 instances.
D. Launch new EC2 instances by using an AMI that is backed by an Amazon Elastic Block Store (Amazon EBS) volume. Attach an additional EC2 instance store volume to contain the application data. Apply the default settings to the EC2 instances.

Correct Answer: C
Section:
Explanation:
Amazon EC2 instances can use two types of storage volumes: instance store volumes and Amazon EBS volumes. Instance store volumes are ephemeral, meaning they are only attached to the instance for the duration of its life cycle. If the instance is stopped, terminated, or fails, the data on the instance store volume is lost. Amazon EBS volumes are persistent, meaning they can be detached from the instance and attached to another instance, and the data on the volume is preserved. To meet the requirement of persisting the data even if the EC2 instances are terminated, the data engineer must use Amazon EBS volumes to store the application data. The solution is to launch new EC2 instances by using an AMI that is backed by an EC2 instance store volume, which is the default option for most AMIs. Then, the data engineer must attach an Amazon EBS volume to each instance and configure the application to write the data to the EBS volume. This way, the data will be saved on the EBS volume and can be accessed by another instance if needed. The data engineer can apply the default settings to the EC2 instances, as there is no need to modify the instance type, security group, or IAM role for this solution. The other options are either not feasible or not optimal. Launching new EC2 instances by using an AMI that is backed by an EC2 instance store volume that contains the application data (option A) or by using an AMI that is backed by a root Amazon EBS volume that contains the application data (option B) would not work, as the data on the AMI would be outdated and overwritten by the new instances. Attaching an additional EC2 instance store volume to contain the application data (option D) would not work, as the data on the instance store volume would be lost if the instance is terminated.Reference:
Amazon EC2 Instance Store
Amazon EBS Volumes
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide, Chapter 2: Data Store Management, Section 2.1: Amazon EC2

QUESTION 12
A company uses Amazon Athena to run SQL queries for extract, transform, and load (ETL) tasks by using Create Table As Select (CTAS). The company must use Apache Spark instead of SQL to generate analytics.
Which solution will give the company the ability to use Spark to access Athena?

A. Athena query settings
B. Athena workgroup
C. Athena data source
D. Athena query editor

Correct Answer: C
Section:
Explanation:
Athena data source is a solution that allows you to use Spark to access Athena by using the Athena JDBC driver and the Spark SQL interface. You can use the Athena data source to create Spark DataFrames from Athena tables, run SQL queries on the DataFrames, and write the results back to Athena. The Athena data source supports various data formats, such as CSV, JSON, ORC, and Parquet, and also supports partitioned and bucketed tables. The Athena data source is a cost-effective and scalable way to use Spark to access Athena, as it does not require any additional infrastructure or services, and you only pay for the data scanned by Athena.
The other options are not solutions that give the company the ability to use Spark to access Athena. Option A, Athena query settings, is a feature that allows you to configure various parameters for your Athena queries, such as the output location, the encryption settings, the query timeout, and the workgroup. Option B, Athena workgroup, is a feature that allows you to isolate and manage your Athena queries and resources, such as the query history, the query notifications, the query concurrency, and the query cost. Option D, Athena query editor, is a feature that allows you to write and run SQL queries on Athena using the web console or the API. None of these options enable you to use Spark instead of SQL to generate analytics on Athena.Reference:
Using Apache Spark in Amazon Athena
Athena JDBC Driver
Spark SQL
Athena query settings
[Athena workgroups]
[Athena query editor]

QUESTION 13
A company needs to partition the Amazon S3 storage that the company uses for a data lake. The partitioning will use a path of the S3 object keys in the following format: s3://bucket/prefix/year=2023/month=01/day=01.
A data engineer must ensure that the AWS Glue Data Catalog synchronizes with the S3 storage when the company adds new partitions to the bucket.
Which solution will meet these requirements with the LEAST latency?

A. Schedule an AWS Glue crawler to run every morning.
B. Manually run the AWS Glue CreatePartition API twice each day.
C. Use code that writes data to Amazon S3 to invoke the Boto3 AWS Glue create partition API call.
D. Run the MSCK REPAIR TABLE command from the AWS Glue console.

Correct Answer: A
Section:
Explanation:
The best solution to ensure that the AWS Glue Data Catalog synchronizes with the S3 storage when the company adds new partitions to the bucket with the least latency is to use code that writes data to Amazon S3 to invoke the Boto3 AWS Glue create partition API call. This way, the Data Catalog is updated as soon as new data is written to S3, and the partition information is immediately available for querying by other services.The Boto3 AWS Glue create partition API call allows you to create a new partition in the Data Catalog by specifying the table name, the database name, and the partition values1. You can use this API call in your code that writes data to S3, such as a Python script or an AWS Glue ETL job, to create a partition for each new S3 object key that matches the partitioning scheme.
Option A is not the best solution, as scheduling an AWS Glue crawler to run every morning would introduce a significant latency between the time new data is written to S3 and the time the Data Catalog is updated.AWS Glue crawlers are processes that connect to a data store, progress through a prioritized list of classifiers to determine the schema for your data, and then create metadata tables in the Data Catalog2. Crawlers can be scheduled to run periodically, such as daily or hourly, but they cannot run continuously or in real-time. Therefore, using a crawler to synchronize the Data Catalog with the S3 storage would not meet the requirement of the least latency.
Option B is not the best solution, as manually running the AWS Glue CreatePartition API twice each day would also introduce a significant latency between the time new data is written to S3 and the time the Data Catalog is updated. Moreover, manually running the API would require more operational overhead and human intervention than using code that writes data to S3 to invoke the API automatically.
Option D is not the best solution, as running the MSCK REPAIR TABLE command from the AWS Glue console would also introduce a significant latency between the time new data is written to S3 and the time the Data Catalog is updated.The MSCK REPAIR TABLE command is a SQL command that you can run in the AWS Glue console to add partitions to the Data Catalog based on the S3 object keys that match the partitioning scheme3. However, this command is not meant to be run frequently or in real-time, as it can take a long time to scan the entire S3 bucket and add the partitions. Therefore, using this command to synchronize the Data Catalog with the S3 storage would not meet the requirement of the least latency.Reference:
AWS Glue CreatePartition API

Populating the AWS Glue Data Catalog
MSCK REPAIR TABLE Command
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide

QUESTION 14
A media company uses software as a service (SaaS) applications to gather data by using third-party tools. The company needs to store the data in an Amazon S3 bucket. The company will use Amazon Redshift to perform analytics based on the data.
Which AWS service or feature will meet these requirements with the LEAST operational overhead?

A. Amazon Managed Streaming for Apache Kafka (Amazon MSK)
B. Amazon AppFlow
C. AWS Glue Data Catalog
D. Amazon Kinesis

Correct Answer: B
Section:
Explanation:
Amazon AppFlow is a fully managed integration service that enables you to securely transfer data between SaaS applications and AWS services like Amazon S3 and Amazon Redshift. Amazon AppFlow supports many SaaS applications as data sources and targets, and allows you to configure data flows with a few clicks. Amazon AppFlow also provides features such as data transformation, filtering, validation, and encryption to prepare and protect your data. Amazon AppFlow meets the requirements of the media company with the least operational overhead, as it eliminates the need to write code, manage infrastructure, or monitor data pipelines.Reference:
Amazon AppFlow
Amazon AppFlow | SaaS Integrations List
Get started with data integration from Amazon S3 to Amazon Redshift using AWS Glue interactive sessions

QUESTION 15
A data engineer is using Amazon Athena to analyze sales data that is in Amazon S3. The data engineer writes a query to retrieve sales amounts for 2023 for several products from a table named sales_data. However, the query does not return results for all of the products that are in the sales_data table. The data engineer needs to troubleshoot the query to resolve the issue.
The data engineer's original query is as follows:
SELECT product_name, sum(sales_amount)
FROM sales_data
WHERE year = 2023
GROUP BY product_name
How should the data engineer modify the Athena query to meet these requirements?

A. Replace sum(sales amount) with count(*J for the aggregation.
B. Change WHERE year = 2023 to WHERE extractlyear FROM sales data) = 2023.
C. Add HAVING sumfsales amount) > 0 after the GROUP BY clause.
D. Remove the GROUP BY clause

Correct Answer: B
Section:
Explanation:
The original query does not return results for all of the products because the year column in the sales_data table is not an integer, but a timestamp. Therefore, the WHERE clause does not filter the data correctly, and only returns the products that have a null value for the year column. To fix this, the data engineer should use the extract function to extract the year from the timestamp and compare it with 2023. This way, the query will return the correct results for all of the products in the sales_data table. The other options are either incorrect or irrelevant, as they do not address the root cause of the issue. Replacing sum with count does not change the filtering condition, adding HAVING clause does not affect the grouping logic, and removing the GROUP BY clause does not solve the problem of missing products.Reference:
Troubleshooting JSON queries - Amazon Athena(Section: JSON related errors)
When I query a table in Amazon Athena, the TIMESTAMP result is empty(Section: Resolution)
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide (Chapter 7, page 197)

QUESTION 16
A data engineer has a one-time task to read data from objects that are in Apache Parquet format in an Amazon S3 bucket. The data engineer needs to query only one column of the data.
Which solution will meet these requirements with the LEAST operational overhead?

A. Configure an AWS Lambda function to load data from the S3 bucket into a pandas dataframe- Write a SQL SELECT statement on the dataframe to query the required column.
B. Use S3 Select to write a SQL SELECT statement to retrieve the required column from the S3 objects.
C. Prepare an AWS Glue DataBrew project to consume the S3 objects and to query the required column.
D. Run an AWS Glue crawler on the S3 objects. Use a SQL SELECT statement in Amazon Athena to query the required column.

Correct Answer: B
Section:
Explanation:
Option B is the best solution to meet the requirements with the least operational overhead because S3 Select is a feature that allows you to retrieve only a subset of data from an S3 object by using simple SQL expressions. S3 Select works on objects stored in CSV, JSON, or Parquet format. By using S3 Select, you can avoid the need to download and process the entire S3 object, which reduces the amount of data transferred and the computation time. S3 Select is also easy to use and does not require any additional services or resources.
Option A is not a good solution because it involves writing custom code and configuring an AWS Lambda function to load data from the S3 bucket into a pandas dataframe and query the required column. This option adds complexity and latency to the data retrieval process and requires additional resources and configuration. Moreover, AWS Lambda has limitations on the execution time, memory, and concurrency, which may affect the performance and reliability of the data retrieval process.
Option C is not a good solution because it involves creating and running an AWS Glue DataBrew project to consume the S3 objects and query the required column. AWS Glue DataBrew is a visual data preparation tool that allows you to clean, normalize, and transform data without writing code. However, in this scenario, the data is already in Parquet format, which is a columnar storage format that is optimized for analytics. Therefore, there is no need to use AWS Glue DataBrew to prepare the data. Moreover, AWS Glue DataBrew adds extra time and cost to the data retrieval process and requires additional resources and configuration.
Option D is not a good solution because it involves running an AWS Glue crawler on the S3 objects and using a SQL SELECT statement in Amazon Athena to query the required column. An AWS Glue crawler is a service that can scan data sources and create metadata tables in the AWS Glue Data Catalog. The Data Catalog is a central repository that stores information about the data sources, such as schema, format, and location. Amazon Athena is a serverless interactive query service that allows you to analyze data in S3 using standard SQL. However, in this scenario, the schema and format of the data are already known and fixed, so there is no need to run a crawler to discover them. Moreover, running a crawler and using Amazon Athena adds extra time and cost to the data retrieval process and requires additional services and configuration.
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide
S3 Select and Glacier Select - Amazon Simple Storage Service
AWS Lambda - FAQs
What Is AWS Glue DataBrew? - AWS Glue DataBrew
Populating the AWS Glue Data Catalog - AWS Glue
What is Amazon Athena? - Amazon Athena

QUESTION 17
A company uses Amazon Redshift for its data warehouse. The company must automate refresh schedules for Amazon Redshift materialized views.
Which solution will meet this requirement with the LEAST effort?

A. Use Apache Airflow to refresh the materialized views.
B. Use an AWS Lambda user-defined function (UDF) within Amazon Redshift to refresh the materialized views.
C. Use the query editor v2 in Amazon Redshift to refresh the materialized views.
D. Use an AWS Glue workflow to refresh the materialized views.

Correct Answer: B
Section:
Explanation:
The query editor v2 in Amazon Redshift is a web-based tool that allows users to run SQL queries and scripts on Amazon Redshift clusters. The query editor v2 supports creating and managing materialized views, which are precomputed results of a query that can improve the performance of subsequent queries. The query editor v2 also supports scheduling queries to run at specified intervals, which can be used to refresh materialized views automatically. This solution requires the least effort, as it does not involve any additional services, coding, or configuration. The other solutions are more complex and require more operational overhead. Apache Airflow is an open-source platform for orchestrating workflows, which can be used to refresh materialized views, but it requires setting up and managing an Airflow environment, creating DAGs (directed acyclic graphs) to define the workflows, and integrating with Amazon Redshift. AWS Lambda is a serverless compute service that can run code in response to events, which can be used to refresh materialized views, but it requires creating and deploying

Lambda functions, defining UDFs within Amazon Redshift, and triggering the functions using events or schedules. AWS Glue is a fully managed ETL service that can run jobs to transform and load data, which can be used to refresh materialized views, but it requires creating and configuring Glue jobs, defining Glue workflows to orchestrate the jobs, and scheduling the workflows using triggers.Reference:
Query editor V2
Working with materialized views
Scheduling queries
[AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide]

QUESTION 18
A data engineer must orchestrate a data pipeline that consists of one AWS Lambda function and one AWS Glue job. The solution must integrate with AWS services.
Which solution will meet these requirements with the LEAST management overhead?

A. Use an AWS Step Functions workflow that includes a state machine. Configure the state machine to run the Lambda function and then the AWS Glue job.
B. Use an Apache Airflow workflow that is deployed on an Amazon EC2 instance. Define a directed acyclic graph (DAG) in which the first task is to call the Lambda function and the second task is to call the AWS Glue job.
C. Use an AWS Glue workflow to run the Lambda function and then the AWS Glue job.
D. Use an Apache Airflow workflow that is deployed on Amazon Elastic Kubernetes Service (Amazon EKS). Define a directed acyclic graph (DAG) in which the first task is to call the Lambda function and the second task is to call the AWS Glue job.

Correct Answer: A
Section:
Explanation:
AWS Step Functions is a service that allows you to coordinate multiple AWS services into serverless workflows. You can use Step Functions to create state machines that define the sequence and logic of the tasks in your workflow. Step Functions supports various types of tasks, such as Lambda functions, AWS Glue jobs, Amazon EMR clusters, Amazon ECS tasks, etc. You can use Step Functions to monitor and troubleshoot your workflows, as well as to handle errors and retries.
Using an AWS Step Functions workflow that includes a state machine to run the Lambda function and then the AWS Glue job will meet the requirements with the least management overhead, as it leverages the serverless and managed capabilities of Step Functions. You do not need to write any code to orchestrate the tasks in your workflow, as you can use the Step Functions console or the AWS Serverless Application Model (AWS SAM) to define and deploy your state machine. You also do not need to provision or manage any servers or clusters, as Step Functions scales automatically based on the demand.
The other options are not as efficient as using an AWS Step Functions workflow. Using an Apache Airflow workflow that is deployed on an Amazon EC2 instance or on Amazon Elastic Kubernetes Service (Amazon EKS) will require more management overhead, as you will need to provision, configure, and maintain the EC2 instance or the EKS cluster, as well as the Airflow components. You will also need to write and maintain the Airflow DAGs to orchestrate the tasks in your workflow. Using an AWS Glue workflow to run the Lambda function and then the AWS Glue job will not work, as AWS Glue workflows only support AWS Glue jobs and crawlers as tasks, not Lambda functions.Reference:
AWS Step Functions
AWS Glue
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide, Chapter 6: Data Integration and Transformation, Section 6.3: AWS Step Functions

QUESTION 19
A company needs to set up a data catalog and metadata management for data sources that run in the AWS Cloud. The company will use the data catalog to maintain the metadata of all the objects that are in a set of data stores. The data stores include structured sources such as Amazon RDS and Amazon Redshift. The data stores also include semistructured sources such as JSON files and .xml files that are stored in Amazon S3.
The company needs a solution that will update the data catalog on a regular basis. The solution also must detect changes to the source metadata.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon Aurora as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the Aurora data catalog. Schedule the Lambda functions to run periodically.
B. Use the AWS Glue Data Catalog as the central metadata repository. Use AWS Glue crawlers to connect to multiple data stores and to update the Data Catalog with metadata changes. Schedule the crawlers to run periodically to update the metadata catalog.
C. Use Amazon DynamoDB as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the DynamoDB data catalog. Schedule the Lambda functions to run periodically.
D. Use the AWS Glue Data Catalog as the central metadata repository. Extract the schema for Amazon RDS and Amazon Redshift sources, and build the Data Catalog. Use AWS Glue crawlers for data that is in Amazon S3 to infer the schema and to automatically update the Data Catalog.

Correct Answer: B
Section:
Explanation:
This solution will meet the requirements with the least operational overhead because it uses the AWS Glue Data Catalog as the central metadata repository for data sources that run in the AWS Cloud. The AWS Glue Data Catalog is a fully managed service that provides a unified view of your data assets across AWS and on-premises data sources. It stores the metadata of your data in tables, partitions, and columns, and enables you to access and query your data using various AWS services, such as Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum. You can use AWS Glue crawlers to connect to multiple data stores, such as Amazon RDS, Amazon Redshift, and Amazon S3, and to update the Data Catalog with metadata changes. AWS Glue crawlers can automatically discover the schema and partition structure of your data, and create or update the corresponding tables in the Data Catalog.You can schedule the crawlers to run periodically to update the metadata catalog, and configure them to detect changes to the source metadata, such as new columns, tables, or partitions12.
The other options are not optimal for the following reasons:
A . Use Amazon Aurora as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the Aurora data catalog. Schedule the Lambda functions to run periodically. This option is not recommended, as it would require more operational overhead to create and manage an Amazon Aurora database as the data catalog, and to write and maintain AWS Lambda functions to gather and update the metadata information from multiple sources. Moreover, this option would not leverage the benefits of the AWS Glue Data Catalog, such as data cataloging, data transformation, and data governance.
C . Use Amazon DynamoDB as the data catalog. Create AWS Lambda functions that will connect to the data catalog. Configure the Lambda functions to gather the metadata information from multiple sources and to update the DynamoDB data catalog. Schedule the Lambda functions to run periodically. This option is also not recommended, as it would require more operational overhead to create and manage an Amazon DynamoDB table as the data catalog, and to write and maintain AWS Lambda functions to gather and update the metadata information from multiple sources. Moreover, this option would not leverage the benefits of the AWS Glue Data Catalog, such as data cataloging, data transformation, and data governance.
D . Use the AWS Glue Data Catalog as the central metadata repository. Extract the schema for Amazon RDS and Amazon Redshift sources, and build the Data Catalog. Use AWS Glue crawlers for data that is in Amazon S3 to infer the schema and to automatically update the Data Catalog. This option is not optimal, as it would require more manual effort to extract the schema for Amazon RDS and Amazon Redshift sources, and to build the Data Catalog. This option would not take advantage of the AWS Glue crawlers' ability to automatically discover the schema and partition structure of your data from various data sources, and to create or update the corresponding tables in the Data Catalog.
1: AWS Glue Data Catalog
2: AWS Glue Crawlers
: Amazon Aurora
: AWS Lambda
: Amazon DynamoDB

QUESTION 20
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
Section:
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 no minimum 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.
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 21
A company is planning to migrate on-premises Apache Hadoop clusters to Amazon EMR. The company also needs to migrate a data catalog into a persistent storage solution.
The company currently stores the data catalog in an on-premises Apache Hive metastore on the Hadoop clusters. The company requires a serverless solution to migrate the data catalog.
Which solution will meet these requirements MOST cost-effectively?

A. Use AWS Database Migration Service (AWS DMS) to migrate the Hive metastore into Amazon S3. Configure AWS Glue Data Catalog to scan Amazon S3 to produce the data catalog.
B. Configure a Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use AWS Glue Data Catalog to store the company's data catalog as an external data catalog.
C. Configure an external Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use Amazon Aurora MySQL to store the company's data catalog.
D. Configure a new Hive metastore in Amazon EMR. Migrate the existing on-premises Hive metastore into Amazon EMR. Use the new metastore as the company's data catalog.

Correct Answer: A
Section:
Explanation:
AWS Database Migration Service (AWS DMS) is a service that helps you migrate databases to AWS quickly and securely. You can use AWS DMS to migrate the Hive metastore from the on-premises Hadoop clusters into Amazon S3, which is a highly scalable, durable, and cost-effective object storage service. AWS Glue Data Catalog is a serverless, managed service that acts as a central metadata repository for your data assets. You can use AWS Glue Data Catalog to scan the Amazon S3 bucket that contains the migrated Hive metastore and create a data catalog that is compatible with Apache Hive and other AWS services. This solution meets the requirements of migrating the data catalog into a persistent storage solution and using a serverless solution. This solution is also the most cost-effective, as it does not incur any additional charges for running Amazon EMR or Amazon Aurora MySQL clusters. The other options are either not feasible or not optimal. Configuring a Hive metastore in Amazon EMR (option B) or an external Hive metastore in Amazon EMR (option C) would require running and maintaining Amazon EMR clusters, which would incur additional costs and complexity. Using Amazon Aurora MySQL to store the company's data catalog (option C) would also incur additional costs and complexity, as well as introduce compatibility issues with Apache Hive. Configuring a new Hive metastore in Amazon EMR (option D) would not migrate the existing data catalog, but create a new one, which would result in data loss and inconsistency.Reference:

Using AWS Database Migration Service
Populating the AWS Glue Data Catalog
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide, Chapter 4: Data Analysis and Visualization, Section 4.2: AWS Glue Data Catalog

QUESTION 22
A company uses an Amazon Redshift provisioned cluster as its database. The Redshift cluster has five reserved ra3.4xlarge nodes and uses key distribution.
A data engineer notices that one of the nodes frequently has a CPU load over 90%. SQL Queries that run on the node are queued. The other four nodes usually have a CPU load under 15% during daily operations.
The data engineer wants to maintain the current number of compute nodes. The data engineer also wants to balance the load more evenly across all five compute nodes.
Which solution will meet these requirements?

A. Change the sort key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement.
B. Change the distribution key to the table column that has the largest dimension.
C. Upgrade the reserved node from ra3.4xlarqe to ra3.16xlarqe.
D. Change the primary key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement.

Correct Answer: B
Section:
Explanation:
Changing the distribution key to the table column that has the largest dimension will help to balance the load more evenly across all five compute nodes. The distribution key determines how the rows of a table are distributed among the slices of the cluster. If the distribution key is not chosen wisely, it can cause data skew, meaning some slices will have more data than others, resulting in uneven CPU load and query performance. By choosing the table column that has the largest dimension, meaning the column that has the most distinct values, as the distribution key, the data engineer can ensure that the rows are distributed more uniformly across the slices, reducing data skew and improving query performance.
The other options are not solutions that will meet the requirements. Option A, changing the sort key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement, will not affect the data distribution or the CPU load. The sort key determines the order in which the rows of a table are stored on disk, which can improve the performance of range-restricted queries, but not the load balancing. Option C, upgrading the reserved node from ra3.4xlarge to ra3.16xlarge, will not maintain the current number of compute nodes, as it will increase the cost and the capacity of the cluster. Option D, changing the primary key to be the data column that is most often used in a WHERE clause of the SQL SELECT statement, will not affect the data distribution or the CPU load either. The primary key is a constraint that enforces the uniqueness of the rows in a table, but it does not influence the data layout or the query optimization.Reference:
Choosing a data distribution style
Choosing a data sort key
Working with primary keys

QUESTION 23
A security company stores IoT data that is in JSON format in an Amazon S3 bucket. The data structure can change when the company upgrades the IoT devices. The company wants to create a data catalog that includes the IoT data. The company's analytics department will use the data catalog to index the data.
Which solution will meet these requirements MOST cost-effectively?

A. Create an AWS Glue Data Catalog. Configure an AWS Glue Schema Registry. Create a new AWS Glue workload to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless.
B. Create an Amazon Redshift provisioned cluster. Create an Amazon Redshift Spectrum database for the analytics department to explore the data that is in Amazon S3. Create Redshift stored procedures to load the data into Amazon Redshift.
C. Create an Amazon Athena workgroup. Explore the data that is in Amazon S3 by using Apache Spark through Athena. Provide the Athena workgroup schema and tables to the analytics department.
D. Create an AWS Glue Data Catalog. Configure an AWS Glue Schema Registry. Create AWS Lambda user defined functions (UDFs) by using the Amazon Redshift Data API. Create an AWS Step Functions job to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless.

Correct Answer: C
Section:
Explanation:
The best solution to meet the requirements of creating a data catalog that includes the IoT data, and allowing the analytics department to index the data, most cost-effectively, is to create an Amazon Athena workgroup, explore the data that is in Amazon S3 by using Apache Spark through Athena, and provide the Athena workgroup schema and tables to the analytics department.

Amazon Athena is a serverless, interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL or Python1.Amazon Athena also supports Apache Spark, an open-source distributed processing framework that can run large-scale data analytics applications across clusters of servers2. You can use Athena to run Spark code on data in Amazon S3 without having to set up, manage, or scale any infrastructure.You can also use Athena to create and manage external tables that point to your data in Amazon S3, and store them in an external data catalog, such as AWS Glue Data Catalog, Amazon Athena Data Catalog, or your own Apache Hive metastore3.You can create Athena workgroups to separate query execution and resource allocation based on different criteria, such as users, teams, or applications4.You can share the schemas and tables in your Athena workgroup with other users or applications, such as Amazon QuickSight, for data visualization and analysis5.

Using Athena and Spark to create a data catalog and explore the IoT data in Amazon S3 is the most cost-effective solution, as you pay only for the queries you run or the compute you use, and you pay nothing when the service is idle1. You also save on the operational overhead and complexity of managing data warehouse infrastructure, as Athena and Spark are serverless and scalable. You can also benefit from the flexibility and performance of Athena and Spark, as they support various data formats, including JSON, and can handle schema changes and complex queries efficiently.

Option A is not the best solution, as creating an AWS Glue Data Catalog, configuring an AWS Glue Schema Registry, creating a new AWS Glue workload to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless, would incur more costs and complexity than using Athena and Spark.AWS Glue Data Catalog is a persistent metadata store that contains table definitions, job definitions, and other control information to help you manage your AWS Glue components6.AWS Glue Schema Registry is a service that allows you to centrally store and manage the schemas of your streaming data in AWS Glue Data Catalog7.AWS Glue is a serverless data integration service that makes it easy to prepare, clean, enrich, and move data between data stores8.Amazon Redshift Serverless is a feature of Amazon Redshift, a fully managed data warehouse service, that allows you to run and scale analytics without having to manage data warehouse infrastructure9. While these services are powerful and useful for many data engineering scenarios, they are not necessary or cost-effective for creating a data catalog and indexing the IoT data in Amazon S3.AWS Glue Data Catalog and Schema Registry charge you based on the number of objects stored and the number of requests made67.AWS Glue charges you based on the compute time and the data processed by your ETL jobs8.Amazon Redshift Serverless charges you based on the amount of data scanned by your queries and the compute time used by your workloads9. These costs can add up quickly, especially if you have large volumes of IoT data and frequent schema changes. Moreover, using AWS Glue and Amazon Redshift Serverless would introduce additional latency and complexity, as you would have to ingest the data from Amazon S3 to Amazon Redshift Serverless, and then query it from there, instead of querying it directly from Amazon S3 using Athena and Spark.

Option B is not the best solution, as creating an Amazon Redshift provisioned cluster, creating an Amazon Redshift Spectrum database for the analytics department to explore the data that is in Amazon S3, and creating Redshift stored procedures to load the data into Amazon Redshift, would incur more costs and complexity than using Athena and Spark.Amazon Redshift provisioned clusters are clusters that you create and manage by specifying the number and type of nodes, and the amount of storage and compute capacity10.Amazon Redshift Spectrum is a feature of Amazon Redshift that allows you to query and join data across your data warehouse and your data lake using standard SQL11.Redshift stored procedures are SQL statements that you can define and store in Amazon Redshift, and then call them by using the CALL command12. While these features are powerful and useful for many data warehousing scenarios, they are not necessary or cost-effective for creating a data catalog and indexing the IoT data in Amazon S3.Amazon Redshift provisioned clusters charge you based on the node type, the number of nodes, and the duration of the cluster10.Amazon Redshift Spectrum charges you based on the amount of data scanned by your queries11. These costs can add up quickly, especially if you have large volumes of IoT data and frequent schema changes. Moreover, using Amazon Redshift provisioned clusters and Spectrum would introduce additional latency and complexity, as you would have to provision and manage the cluster, create an external schema and database for the data in Amazon S3, and load the data into the cluster using stored procedures, instead of querying it directly from Amazon S3 using Athena and Spark.

Option D is not the best solution, as creating an AWS Glue Data Catalog, configuring an AWS Glue Schema Registry, creating AWS Lambda user defined functions (UDFs) by using the Amazon Redshift Data API, and creating an AWS Step Functions job to orchestrate the ingestion of the data that the analytics department will use into Amazon Redshift Serverless, would incur more costs and complexity than using Athena and Spark.AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers13. AWS Lambda UDFs are Lambda functions that you can invoke from within an Amazon Redshift query. Amazon Redshift Data API is a service that allows you to run SQL statements on Amazon Redshift clusters using HTTP requests, without needing a persistent connection. AWS Step Functions is a service that lets you coordinate multiple AWS services into serverless workflows. While these services are powerful and useful for many data engineering scenarios, they are not necessary or cost-effective for creating a data catalog and indexing the IoT data in Amazon S3.AWS Glue Data Catalog and Schema Registry charge you based on the number of objects stored and the number of requests made67.AWS Lambda charges you based on the number of requests and the duration of your functions13.Amazon Redshift Serverless charges you based on the amount of data scanned by your queries and the compute time used by your workloads9. AWS Step Functions charges you based on the number of state transitions in your workflows. These costs can add up quickly, especially if you have large volumes of IoT data and frequent schema changes. Moreover, using AWS Glue, AWS Lambda, Amazon Redshift Data API, and AWS Step Functions would introduce additional latency and complexity, as you would have to create and invoke Lambda functions to ingest the data from Amazon S3 to Amazon Redshift Serverless using the Data API, and coordinate the ingestion process using Step Functions, instead of querying it directly from Amazon S3 using Athena and Spark.Reference:

What is Amazon Athena?

Apache Spark on Amazon Athena

Creating tables, updating the schema, and adding new partitions in the Data Catalog from AWS Glue ETL jobs

Managing Athena workgroups

Using Amazon QuickSight to visualize data in Amazon Athena

AWS Glue Data Catalog

AWS Glue Schema Registry

What is AWS Glue?

Amazon Redshift Serverless

Amazon Redshift provisioned clusters

Querying external data using Amazon Redshift Spectrum

Using stored procedures in Amazon Redshift

What is AWS Lambda?

[Creating and using AWS Lambda UDFs]

[Using the Amazon Redshift Data API]

[What is AWS Step Functions?]
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide

QUESTION 24
A company stores details about transactions in an Amazon S3 bucket. The company wants to log all writes to the S3 bucket into another S3 bucket that is in the same AWS Region.
Which solution will meet this requirement with the LEAST operational effort?

A. Configure an S3 Event Notifications rule for all activities on the transactions S3 bucket to invoke an AWS Lambda function. Program the Lambda function to write the event to Amazon Kinesis Data Firehose. Configure Kinesis Data Firehose to write the event to the logs S3 bucket.
B. Create a trail of management events in AWS CloudTraiL. Configure the trail to receive data from the transactions S3 bucket. Specify an empty prefix and write-only events. Specify the logs S3 bucket as the destination bucket.
C. Configure an S3 Event Notifications rule for all activities on the transactions S3 bucket to invoke an AWS Lambda function. Program the Lambda function to write the events to the logs S3 bucket.
D. Create a trail of data events in AWS CloudTraiL. Configure the trail to receive data from the transactions S3 bucket. Specify an empty prefix and write-only events. Specify the logs S3 bucket as the destination bucket.

Correct Answer: D
Section:
Explanation:
This solution meets the requirement of logging all writes to the S3 bucket into another S3 bucket with the least operational effort. AWS CloudTrail is a service that records the API calls made to AWS services, including Amazon S3. By creating a trail of data events, you can capture the details of the requests that are made to the transactions S3 bucket, such as the requester, the time, the IP address, and the response elements. By specifying an empty prefix and write-only events, you can filter the data events to only include the ones that write to the bucket. By specifying the logs S3 bucket as the destination bucket, you can store the CloudTrail logs in another S3 bucket that is in the same AWS Region. This solution does not require any additional coding or configuration, and it is more scalable and reliable than using S3 Event Notifications and Lambda functions.Reference:
Logging Amazon S3 API calls using AWS CloudTrail
Creating a trail for data events
Enabling Amazon S3 server access logging

QUESTION 25
A data engineer needs to maintain a central metadata repository that users access through Amazon EMR and Amazon Athena queries. The repository needs to provide the schema and properties of many tables. Some of the metadata is stored in Apache Hive. The data engineer needs to import the metadata from Hive into the central metadata repository.
Which solution will meet these requirements with the LEAST development effort?

A. Use Amazon EMR and Apache Ranger.
B. Use a Hive metastore on an EMR cluster.
C. Use the AWS Glue Data Catalog.
D. Use a metastore on an Amazon RDS for MySQL DB instance.

Correct Answer: C
Section:
Explanation:
The AWS Glue Data Catalog is an Apache Hive metastore-compatible catalog that provides a central metadata repository for various data sources and formats. You can use the AWS Glue Data Catalog as an external Hive metastore for Amazon EMR and Amazon Athena queries, and import metadata from existing Hive metastores into the Data Catalog. This solution requires the least development effort, as you can use AWS Glue crawlers to automatically discover and catalog the metadata from Hive, and use the AWS Glue console, AWS CLI, or Amazon EMR API to configure the Data Catalog as the Hive metastore. The other options are either more complex or require additional steps, such as setting up Apache Ranger for security, managing a Hive metastore on an EMR cluster or an RDS instance, or migrating the metadata manually.Reference:
Using the AWS Glue Data Catalog as the metastore for Hive(Section: Specifying AWS Glue Data Catalog as the metastore)
Metadata Management: Hive Metastore vs AWS Glue(Section: AWS Glue Data Catalog)
AWS Glue Data Catalog support for Spark SQL jobs(Section: Importing metadata from an existing Hive metastore)
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide (Chapter 5, page 131)

QUESTION 26
A company needs to build a data lake in AWS. The company must provide row-level data access and column-level data access to specific teams. The teams will access the data by using Amazon Athena, Amazon Redshift

Spectrum, and Apache Hive from Amazon EMR.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon S3 for data lake storage. Use S3 access policies to restrict data access by rows and columns. Provide data access through Amazon S3.
B. Use Amazon S3 for data lake storage. Use Apache Ranger through Amazon EMR to restrict data access by rows and columns. Provide data access by using Apache Pig.
C. Use Amazon Redshift for data lake storage. Use Redshift security policies to restrict data access by rows and columns. Provide data access by using Apache Spark and Amazon Athena federated queries.
D. Use Amazon S3 for data lake storage. Use AWS Lake Formation to restrict data access by rows and columns. Provide data access through AWS Lake Formation.

Correct Answer: D
Section:
Explanation:
Option D is the best solution to meet the requirements with the least operational overhead because AWS Lake Formation is a fully managed service that simplifies the process of building, securing, and managing data lakes. AWS Lake Formation allows you to define granular data access policies at the row and column level for different users and groups. AWS Lake Formation also integrates with Amazon Athena, Amazon Redshift Spectrum, and Apache Hive on Amazon EMR, enabling these services to access the data in the data lake through AWS Lake Formation.
Option A is not a good solution because S3 access policies cannot restrict data access by rows and columns. S3 access policies are based on the identity and permissions of the requester, the bucket and object ownership, and the object prefix and tags. S3 access policies cannot enforce fine-grained data access control at the row and column level.
Option B is not a good solution because it involves using Apache Ranger and Apache Pig, which are not fully managed services and require additional configuration and maintenance. Apache Ranger is a framework that provides centralized security administration for data stored in Hadoop clusters, such as Amazon EMR. Apache Ranger can enforce row-level and column-level access policies for Apache Hive tables. However, Apache Ranger is not a native AWS service and requires manual installation and configuration on Amazon EMR clusters. Apache Pig is a platform that allows you to analyze large data sets using a high-level scripting language called Pig Latin. Apache Pig can access data stored in Amazon S3 and process it using Apache Hive. However, Apache Pig is not a native AWS service and requires manual installation and configuration on Amazon EMR clusters.
Option C is not a good solution because Amazon Redshift is not a suitable service for data lake storage. Amazon Redshift is a fully managed data warehouse service that allows you to run complex analytical queries using standard SQL. Amazon Redshift can enforce row-level and column-level access policies for different users and groups. However, Amazon Redshift is not designed to store and process large volumes of unstructured or semi-structured data, which are typical characteristics of data lakes. Amazon Redshift is also more expensive and less scalable than Amazon S3 for data lake storage.
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide
What Is AWS Lake Formation? - AWS Lake Formation
Using AWS Lake Formation with Amazon Athena - AWS Lake Formation
Using AWS Lake Formation with Amazon Redshift Spectrum - AWS Lake Formation
Using AWS Lake Formation with Apache Hive on Amazon EMR - AWS Lake Formation
Using Bucket Policies and User Policies - Amazon Simple Storage Service
Apache Ranger
Apache Pig
What Is Amazon Redshift? - Amazon Redshift

QUESTION 27
An airline company is collecting metrics about flight activities for analytics. The company is conducting a proof of concept (POC) test to show how analytics can provide insights that the company can use to increase on-time departures.
The POC test uses objects in Amazon S3 that contain the metrics in .csv format. The POC test uses Amazon Athena to query the data. The data is partitioned in the S3 bucket by date.
As the amount of data increases, the company wants to optimize the storage solution to improve query performance.
Which combination of solutions will meet these requirements? (Choose two.)

A. Add a randomized string to the beginning of the keys in Amazon S3 to get more throughput across partitions.
B. Use an S3 bucket that is in the same account that uses Athena to query the data.
C. Use an S3 bucket that is in the same AWS Region where the company runs Athena queries.
D. Preprocess the .csv data to JSON format by fetching only the document keys that the query requires.
E. Preprocess the .csv data to Apache Parquet format by fetching only the data blocks that are needed for predicates.

Correct Answer: C, E
Section:

Explanation:
Using an S3 bucket that is in the same AWS Region where the company runs Athena queries can improve query performance by reducing data transfer latency and costs. Preprocessing the .csv data to Apache Parquet format can also improve query performance by enabling columnar storage, compression, and partitioning, which can reduce the amount of data scanned and fetched by the query. These solutions can optimize the storage solution for the POC test without requiring much effort or changes to the existing data pipeline. The other solutions are not optimal or relevant for this requirement. Adding a randomized string to the beginning of the keys in Amazon S3 can improve the throughput across partitions, but it can also make the data harder to query and manage. Using an S3 bucket that is in the same account that uses Athena to query the data does not have any significant impact on query performance, as long as the proper permissions are granted. Preprocessing the .csv data to JSON format does not offer any benefits over the .csv format, as both are row-based and verbose formats that require more data scanning and fetching than columnar formats like Parquet.Reference:
Best Practices When Using Athena with AWS Glue
Optimizing Amazon S3 Performance
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide

QUESTION 28
A company uses Amazon RDS for MySQL as the database for a critical application. The database workload is mostly writes, with a small number of reads.
A data engineer notices that the CPU utilization of the DB instance is very high. The high CPU utilization is slowing down the application. The data engineer must reduce the CPU utilization of the DB Instance.
Which actions should the data engineer take to meet this requirement? (Choose two.)

A. Use the Performance Insights feature of Amazon RDS to identify queries that have high CPU utilization. Optimize the problematic queries.
B. Modify the database schema to include additional tables and indexes.
C. Reboot the RDS DB instance once each week.
D. Upgrade to a larger instance size.
E. Implement caching to reduce the database query load.

Correct Answer: A, E
Section:
Explanation:
Amazon RDS is a fully managed service that provides relational databases in the cloud. Amazon RDS for MySQL is one of the supported database engines that you can use to run your applications. Amazon RDS provides various features and tools to monitor and optimize the performance of your DB instances, such as Performance Insights, Enhanced Monitoring, CloudWatch metrics and alarms, etc.
Using the Performance Insights feature of Amazon RDS to identify queries that have high CPU utilization and optimizing the problematic queries will help reduce the CPU utilization of the DB instance. Performance Insights is a feature that allows you to analyze the load on your DB instance and determine what is causing performance issues. Performance Insights collects, analyzes, and displays database performance data using an interactive dashboard. You can use Performance Insights to identify the top SQL statements, hosts, users, or processes that are consuming the most CPU resources. You can also drill down into the details of each query and see the execution plan, wait events, locks, etc. By using Performance Insights, you can pinpoint the root cause of the high CPU utilization and optimize the queries accordingly. For example, you can rewrite the queries to make them more efficient, add or remove indexes, use prepared statements, etc.
Implementing caching to reduce the database query load will also help reduce the CPU utilization of the DB instance. Caching is a technique that allows you to store frequently accessed data in a fast and scalable storage layer, such as Amazon ElastiCache. By using caching, you can reduce the number of requests that hit your database, which in turn reduces the CPU load on your DB instance. Caching also improves the performance and availability of your application, as it reduces the latency and increases the throughput of your data access. You can use caching for various scenarios, such as storing session data, user preferences, application configuration, etc. You can also use caching for read-heavy workloads, such as displaying product details, recommendations, reviews, etc.
The other options are not as effective as using Performance Insights and caching. Modifying the database schema to include additional tables and indexes may or may not improve the CPU utilization, depending on the nature of the workload and the queries. Adding more tables and indexes may increase the complexity and overhead of the database, which may negatively affect the performance. Rebooting the RDS DB instance once each week will not reduce the CPU utilization, as it will not address the underlying cause of the high CPU load. Rebooting may also cause downtime and disruption to your application. Upgrading to a larger instance size may reduce the CPU utilization, but it will also increase the cost and complexity of your solution. Upgrading may also not be necessary if you can optimize the queries and reduce the database load by using caching.Reference:
Amazon RDS
Performance Insights
Amazon ElastiCache
[AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide], Chapter 3: Data Storage and Management, Section 3.1: Amazon RDS

QUESTION 29
A company has used an Amazon Redshift table that is named Orders for 6 months. The company performs weekly updates and deletes on the table. The table has an interleaved sort key on a column that contains AWS Regions.
The company wants to reclaim disk space so that the company will not run out of storage space. The company also wants to analyze the sort key column.

Which Amazon Redshift command will meet these requirements?

A. VACUUM FULL Orders
B. VACUUM DELETE ONLY Orders
C. VACUUM REINDEX Orders
D. VACUUM SORT ONLY Orders

Correct Answer: C
Section:
Explanation:
Amazon Redshift is a fully managed, petabyte-scale data warehouse service that enables fast and cost-effective analysis of large volumes of data. Amazon Redshift uses columnar storage, compression, and zone maps to optimize the storage and performance of data. However, over time, as data is inserted, updated, or deleted, the physical storage of data can become fragmented, resulting in wasted disk space and degraded query performance.To address this issue, Amazon Redshift provides the VACUUM command, which reclaims disk space and resorts rows in either a specified table or all tables in the current schema1.
The VACUUM command has four options: FULL, DELETE ONLY, SORT ONLY, and REINDEX. The option that best meets the requirements of the question is VACUUM REINDEX, which re-sorts the rows in a table that has an interleaved sort key and rewrites the table to a new location on disk. An interleaved sort key is a type of sort key that gives equal weight to each column in the sort key, and stores the rows in a way that optimizes the performance of queries that filter by multiple columns in the sort key. However, as data is added or changed, the interleaved sort order can become skewed, resulting in suboptimal query performance. The VACUUM REINDEX option restores the optimal interleaved sort order and reclaims disk space by removing deleted rows.This option also analyzes the sort key column and updates the table statistics, which are used by the query optimizer to generate the most efficient query execution plan23.
The other options are not optimal for the following reasons:
A . VACUUM FULL Orders. This option reclaims disk space by removing deleted rows and resorts the entire table. However, this option is not suitable for tables that have an interleaved sort key, as it does not restore the optimal interleaved sort order. Moreover, this option is the most resource-intensive and time-consuming, as it rewrites the entire table to a new location on disk.
B . VACUUM DELETE ONLY Orders. This option reclaims disk space by removing deleted rows, but does not resort the table. This option is not suitable for tables that have any sort key, as it does not improve the query performance by restoring the sort order. Moreover, this option does not analyze the sort key column and update the table statistics.
D . VACUUM SORT ONLY Orders. This option resorts the entire table, but does not reclaim disk space by removing deleted rows. This option is not suitable for tables that have an interleaved sort key, as it does not restore the optimal interleaved sort order. Moreover, this option does not analyze the sort key column and update the table statistics.
1: Amazon Redshift VACUUM
2: Amazon Redshift Interleaved Sorting
3: Amazon Redshift ANALYZE

QUESTION 30
A manufacturing company wants to collect data from sensors. A data engineer needs to implement a solution that ingests sensor data in near real time.
The solution must store the data to a persistent data store. The solution must store the data in nested JSON format. The company must have the ability to query from the data store with a latency of less than 10 milliseconds.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use a self-hosted Apache Kafka cluster to capture the sensor data. Store the data in Amazon S3 for querying.
B. Use AWS Lambda to process the sensor data. Store the data in Amazon S3 for querying.
C. Use Amazon Kinesis Data Streams to capture the sensor data. Store the data in Amazon DynamoDB for querying.
D. Use Amazon Simple Queue Service (Amazon SQS) to buffer incoming sensor data. Use AWS Glue to store the data in Amazon RDS for querying.

Correct Answer: C
Section:
Explanation:
Amazon Kinesis Data Streams is a service that enables you to collect, process, and analyze streaming data in real time. You can use Kinesis Data Streams to capture sensor data from various sources, such as IoT devices, web applications, or mobile apps. You can create data streams that can scale up to handle any amount of data from thousands of producers.You can also use the Kinesis Client Library (KCL) or the Kinesis Data Streams API to write applications that process and analyze the data in the streams1.
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. You can use DynamoDB to store the sensor data in nested JSON format, as DynamoDB supports document data types, such as lists and maps. You can also use DynamoDB to query the data with a latency of less than 10 milliseconds, as DynamoDB offers single-digit millisecond performance for any scale of data.You can use the DynamoDB API or the AWS SDKs to perform queries on the data, such as using key-value lookups, scans, or queries2.
The solution that meets the requirements with the least operational overhead is to use Amazon Kinesis Data Streams to capture the sensor data and store the data in Amazon DynamoDB for querying. This solution has the

following advantages:
It does not require you to provision, manage, or scale any servers, clusters, or queues, as Kinesis Data Streams and DynamoDB are fully managed services that handle all the infrastructure for you. This reduces the operational complexity and cost of running your solution.
It allows you to ingest sensor data in near real time, as Kinesis Data Streams can capture data records as they are produced and deliver them to your applications within seconds.You can also use Kinesis Data Firehose to load the data from the streams to DynamoDB automatically and continuously3.
It allows you to store the data in nested JSON format, as DynamoDB supports document data types, such as lists and maps. You can also use DynamoDB Streams to capture changes in the data and trigger actions, such as sending notifications or updating other databases.
It allows you to query the data with a latency of less than 10 milliseconds, as DynamoDB offers single-digit millisecond performance for any scale of data. You can also use DynamoDB Accelerator (DAX) to improve the read performance by caching frequently accessed data.
Option A is incorrect because it suggests using a self-hosted Apache Kafka cluster to capture the sensor data and store the data in Amazon S3 for querying. This solution has the following disadvantages:
It requires you to provision, manage, and scale your own Kafka cluster, either on EC2 instances or on-premises servers. This increases the operational complexity and cost of running your solution.
It does not allow you to query the data with a latency of less than 10 milliseconds, as Amazon S3 is an object storage service that is not optimized for low-latency queries. You need to use another service, such as Amazon Athena or Amazon Redshift Spectrum, to query the data in S3, which may incur additional costs and latency.
Option B is incorrect because it suggests using AWS Lambda to process the sensor data and store the data in Amazon S3 for querying. This solution has the following disadvantages:
It does not allow you to ingest sensor data in near real time, as Lambda is a serverless compute service that runs code in response to events. You need to use another service, such as API Gateway or Kinesis Data Streams, to trigger Lambda functions with sensor data, which may add extra latency and complexity to your solution.
It does not allow you to query the data with a latency of less than 10 milliseconds, as Amazon S3 is an object storage service that is not optimized for low-latency queries. You need to use another service, such as Amazon Athena or Amazon Redshift Spectrum, to query the data in S3, which may incur additional costs and latency.
Option D is incorrect because it suggests using Amazon Simple Queue Service (Amazon SQS) to buffer incoming sensor data and use AWS Glue to store the data in Amazon RDS for querying. This solution has the following disadvantages:
It does not allow you to ingest sensor data in near real time, as Amazon SQS is a message queue service that delivers messages in a best-effort manner. You need to use another service, such as Lambda or EC2, to poll the messages from the queue and process them, which may add extra latency and complexity to your solution.
It does not allow you to store the data in nested JSON format, as Amazon RDS is a relational database service that supports structured data types, such as tables and columns. You need to use another service, such as AWS Glue, to transform the data from JSON to relational format, which may add extra cost and overhead to your solution.
1: Amazon Kinesis Data Streams - Features
2: Amazon DynamoDB - Features
3: Loading Streaming Data into Amazon DynamoDB - Amazon Kinesis Data Firehose
[4]: Capturing Table Activity with DynamoDB Streams - Amazon DynamoDB
[5]: Amazon DynamoDB Accelerator (DAX) - Features
[6]: Amazon S3 - Features
[7]: AWS Lambda - Features
[8]: Amazon Simple Queue Service - Features
[9]: Amazon Relational Database Service - Features
[10]: Working with JSON in Amazon RDS - Amazon Relational Database Service
[11]: AWS Glue - Features

QUESTION 31
A company stores data in a data lake that is in Amazon S3. Some data that the company stores in the data lake contains personally identifiable information (PII). Multiple user groups need to access the raw data. The company must ensure that user groups can access only the PII that they require.
Which solution will meet these requirements with the LEAST effort?

A. Use Amazon Athena to query the data. Set up AWS Lake Formation and create data filters to establish levels of access for the company's IAM roles. Assign each user to the IAM role that matches the user's PII access requirements.
B. Use Amazon QuickSight to access the data. Use column-level security features in QuickSight to limit the PII that users can retrieve from Amazon S3 by using Amazon Athena. Define QuickSight access levels based on the PII access requirements of the users.
C. Build a custom query builder UI that will run Athena queries in the background to access the data. Create user groups in Amazon Cognito. Assign access levels to the user groups based on the PII access requirements of the users.
D. Create IAM roles that have different levels of granular access. Assign the IAM roles to IAM user groups. Use an identity-based policy to assign access levels to user groups at the column level.

Correct Answer: A
Section:
Explanation:
Amazon Athena is a serverless, interactive query service that enables you to analyze data in Amazon S3 using standard SQL. AWS Lake Formation is a service that helps you build, secure, and manage data lakes on AWS. You can use AWS Lake Formation to create data filters that define the level of access for different IAM roles based on the columns, rows, or tags of the data. By using Amazon Athena to query the data and AWS Lake Formation to create data filters, the company can meet the requirements of ensuring that user groups can access only the PII that they require with the least effort. The solution is to use Amazon Athena to query the data in the data lake that is in Amazon S3. Then, set up AWS Lake Formation and create data filters to establish levels of access for the company's IAM roles. For example, a data filter can allow a user group to access only the columns that contain the PII that they need, such as name and email address, and deny access to the columns that contain the PII that they do not need, such as phone number and social security number. Finally, assign each user to the IAM role that matches the user's PII access requirements. This way, the user groups can access the data in the data lake securely and efficiently. The other options are either not feasible or not optimal. Using Amazon QuickSight to access the data (option B) would require the company to pay for the QuickSight service and to configure the column-level security features for each user. Building a custom query builder UI that will run Athena queries in the background to access the data (option C) would require the company to develop and maintain the UI and to integrate it with Amazon Cognito. Creating IAM roles that have different levels of granular access (option D) would require the company to manage multiple IAM roles and policies and to ensure that they are aligned with the data schema.Reference:
Amazon Athena
AWS Lake Formation
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide, Chapter 4: Data Analysis and Visualization, Section 4.3: Amazon Athena

QUESTION 32
A data engineer uses Amazon Redshift to run resource-intensive analytics processes once every month. Every month, the data engineer creates a new Redshift provisioned cluster. The data engineer deletes the Redshift provisioned cluster after the analytics processes are complete every month. Before the data engineer deletes the cluster each month, the data engineer unloads backup data from the cluster to an Amazon S3 bucket.
The data engineer needs a solution to run the monthly analytics processes that does not require the data engineer to manage the infrastructure manually.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon Step Functions to pause the Redshift cluster when the analytics processes are complete and to resume the cluster to run new processes every month.
B. Use Amazon Redshift Serverless to automatically process the analytics workload.
C. Use the AWS CLI to automatically process the analytics workload.
D. Use AWS CloudFormation templates to automatically process the analytics workload.

Correct Answer: B
Section:
Explanation:
Amazon Redshift Serverless is a new feature of Amazon Redshift that enables you to run SQL queries on data in Amazon S3 without provisioning or managing any clusters. You can use Amazon Redshift Serverless to automatically process the analytics workload, as it scales up and down the compute resources based on the query demand, and charges you only for the resources consumed. This solution will meet the requirements with the least operational overhead, as it does not require the data engineer to create, delete, pause, or resume any Redshift clusters, or to manage any infrastructure manually.You can use the Amazon Redshift Data API to run queries from the AWS CLI, AWS SDK, or AWS Lambda functions12.
The other options are not optimal for the following reasons:
A . Use Amazon Step Functions to pause the Redshift cluster when the analytics processes are complete and to resume the cluster to run new processes every month. This option is not recommended, as it would still require the data engineer to create and delete a new Redshift provisioned cluster every month, which can incur additional costs and time. Moreover, this option would require the data engineer to use Amazon Step Functions to orchestrate the workflow of pausing and resuming the cluster, which can add complexity and overhead.
C . Use the AWS CLI to automatically process the analytics workload. This option is vague and does not specify how the AWS CLI is used to process the analytics workload. The AWS CLI can be used to run queries on data in Amazon S3 using Amazon Redshift Serverless, Amazon Athena, or Amazon EMR, but each of these services has different features and benefits. Moreover, this option does not address the requirement of not managing the infrastructure manually, as the data engineer may still need to provision and configure some resources, such as Amazon EMR clusters or Amazon Athena workgroups.
D . Use AWS CloudFormation templates to automatically process the analytics workload. This option is also vague and does not specify how AWS CloudFormation templates are used to process the analytics workload. AWS CloudFormation is a service that lets you model and provision AWS resources using templates. You can use AWS CloudFormation templates to create and delete a Redshift provisioned cluster every month, or to create and configure other AWS resources, such as Amazon EMR, Amazon Athena, or Amazon Redshift Serverless. However, this option does not address the requirement of not managing the infrastructure manually, as the data engineer may still need to write and maintain the AWS CloudFormation templates, and to monitor the status and performance of the resources.
1: Amazon Redshift Serverless
2: Amazon Redshift Data API
: Amazon Step Functions
: AWS CLI

: AWS CloudFormation

QUESTION 33
A company receives a daily file that contains customer data in .xls format. The company stores the file in Amazon S3. The daily file is approximately 2 GB in size.
A data engineer concatenates the column in the file that contains customer first names and the column that contains customer last names. The data engineer needs to determine the number of distinct customers in the file.
Which solution will meet this requirement with the LEAST operational effort?

A. Create and run an Apache Spark job in an AWS Glue notebook. Configure the job to read the S3 file and calculate the number of distinct customers.
B. Create an AWS Glue crawler to create an AWS Glue Data Catalog of the S3 file. Run SQL queries from Amazon Athena to calculate the number of distinct customers.
C. Create and run an Apache Spark job in Amazon EMR Serverless to calculate the number of distinct customers.
D. Use AWS Glue DataBrew to create a recipe that uses the COUNT_DISTINCT aggregate function to calculate the number of distinct customers.

Correct Answer: D
Section:
Explanation:
AWS Glue DataBrew is a visual data preparation tool that allows you to clean, normalize, and transform data without writing code. You can use DataBrew to create recipes that define the steps to apply to your data, such as filtering, renaming, splitting, or aggregating columns. You can also use DataBrew to run jobs that execute the recipes on your data sources, such as Amazon S3, Amazon Redshift, or Amazon Aurora.DataBrew integrates with AWS Glue Data Catalog, which is a centralized metadata repository for your data assets1.
The solution that meets the requirement with the least operational effort is to use AWS Glue DataBrew to create a recipe that uses the COUNT_DISTINCT aggregate function to calculate the number of distinct customers. This solution has the following advantages:
It does not require you to write any code, as DataBrew provides a graphical user interface that lets you explore, transform, and visualize your data.You can use DataBrew to concatenate the columns that contain customer first names and last names, and then use the COUNT_DISTINCT aggregate function to count the number of unique values in the resulting column2.
It does not require you to provision, manage, or scale any servers, clusters, or notebooks, as DataBrew is a fully managed service that handles all the infrastructure for you.DataBrew can automatically scale up or down the compute resources based on the size and complexity of your data and recipes1.
It does not require you to create or update any AWS Glue Data Catalog entries, as DataBrew can automatically create and register the data sources and targets in the Data Catalog.DataBrew can also use the existing Data Catalog entries to access the data in S3 or other sources3.
Option A is incorrect because it suggests creating and running an Apache Spark job in an AWS Glue notebook. This solution has the following disadvantages:
It requires you to write code, as AWS Glue notebooks are interactive development environments that allow you to write, test, and debug Apache Spark code using Python or Scala. You need to use the Spark SQL or the Spark DataFrame API to read the S3 file and calculate the number of distinct customers.
It requires you to provision and manage a development endpoint, which is a serverless Apache Spark environment that you can connect to your notebook. You need to specify the type and number of workers for your development endpoint, and monitor its status and metrics.
It requires you to create or update the AWS Glue Data Catalog entries for the S3 file, either manually or using a crawler. You need to use the Data Catalog as a metadata store for your Spark job, and specify the database and table names in your code.
Option B is incorrect because it suggests creating an AWS Glue crawler to create an AWS Glue Data Catalog of the S3 file, and running SQL queries from Amazon Athena to calculate the number of distinct customers. This solution has the following disadvantages:
It requires you to create and run a crawler, which is a program that connects to your data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in the Data Catalog. You need to specify the data store, the IAM role, the schedule, and the output database for your crawler.
It requires you to write SQL queries, as Amazon Athena is a serverless interactive query service that allows you to analyze data in S3 using standard SQL. You need to use Athena to concatenate the columns that contain customer first names and last names, and then use the COUNT(DISTINCT) aggregate function to count the number of unique values in the resulting column.
Option C is incorrect because it suggests creating and running an Apache Spark job in Amazon EMR Serverless to calculate the number of distinct customers. This solution has the following disadvantages:
It requires you to write code, as Amazon EMR Serverless is a service that allows you to run Apache Spark jobs on AWS without provisioning or managing any infrastructure. You need to use the Spark SQL or the Spark DataFrame API to read the S3 file and calculate the number of distinct customers.
It requires you to create and manage an Amazon EMR Serverless cluster, which is a fully managed and scalable Spark environment that runs on AWS Fargate. You need to specify the cluster name, the IAM role, the VPC, and the subnet for your cluster, and monitor its status and metrics.
It requires you to create or update the AWS Glue Data Catalog entries for the S3 file, either manually or using a crawler. You need to use the Data Catalog as a metadata store for your Spark job, and specify the database and table names in your code.
1: AWS Glue DataBrew - Features
2: Working with recipes - AWS Glue DataBrew
3: Working with data sources and data targets - AWS Glue DataBrew

[4]: AWS Glue notebooks - AWS Glue
[5]: Development endpoints - AWS Glue
[6]: Populating the AWS Glue Data Catalog - AWS Glue
[7]: Crawlers - AWS Glue
[8]: Amazon Athena - Features
[9]: Amazon EMR Serverless - Features
[10]: Creating an Amazon EMR Serverless cluster - Amazon EMR
[11]: Using the AWS Glue Data Catalog with Amazon EMR Serverless - Amazon EMR

QUESTION 34
A company hosts its applications on Amazon EC2 instances. The company must use SSL/TLS connections that encrypt data in transit to communicate securely with AWS infrastructure that is managed by a customer.
A data engineer needs to implement a solution to simplify the generation, distribution, and rotation of digital certificates. The solution must automatically renew and deploy SSL/TLS certificates.
Which solution will meet these requirements with the LEAST operational overhead?

A. Store self-managed certificates on the EC2 instances.
B. Use AWS Certificate Manager (ACM).
C. Implement custom automation scripts in AWS Secrets Manager.
D. Use Amazon Elastic Container Service (Amazon ECS) Service Connect.

Correct Answer: B
Section:
Explanation:
The best solution for managing SSL/TLS certificates on EC2 instances with minimal operational overhead is to use AWS Certificate Manager (ACM). ACM simplifies certificate management by automating the provisioning, renewal, and deployment of certificates.
AWS Certificate Manager (ACM):
ACM manages SSL/TLS certificates for EC2 and other AWS resources, including automatic certificate renewal. This reduces the need for manual management and avoids operational complexity.
ACM also integrates with other AWS services to simplify secure connections between AWS infrastructure and customer-managed environments.
Alternatives Considered:
A (Self-managed certificates): Managing certificates manually on EC2 instances increases operational overhead and lacks automatic renewal.
C (Secrets Manager automation): While Secrets Manager can store keys and certificates, it requires custom automation for rotation and does not handle SSL/TLS certificates directly.
D (ECS Service Connect): This is unrelated to SSL/TLS certificate management and would not address the operational need.
AWS Certificate Manager Documentation

QUESTION 35
A company saves customer data to an Amazon S3 bucket. The company uses server-side encryption with AWS KMS keys (SSE-KMS) to encrypt the bucket. The dataset includes personally identifiable information (PII) such as social security numbers and account details.
Data that is tagged as PII must be masked before the company uses customer data for analysis. Some users must have secure access to the PII data during the preprocessing phase. The company needs a low-maintenance solution to mask and secure the PII data throughout the entire engineering pipeline.
Which combination of solutions will meet these requirements? (Select TWO.)

A. Use AWS Glue DataBrew to perform extract, transform, and load (ETL) tasks that mask the PII data before analysis.
B. Use Amazon GuardDuty to monitor access patterns for the PII data that is used in the engineering pipeline.
C. Configure an Amazon Made discovery job for the S3 bucket.
D. Use AWS Identity and Access Management (IAM) to manage permissions and to control access to the PII data.
E. Write custom scripts in an application to mask the PII data and to control access.

Correct Answer: A, D
Section:

Explanation:
To address the requirement of masking PII data and ensuring secure access throughout the data pipeline, the combination of AWS Glue DataBrew and IAM provides a low-maintenance solution.
A . AWS Glue DataBrew for Masking:
AWS Glue DataBrew provides a visual tool to perform data transformations, including masking PII data. It allows for easy configuration of data transformation tasks without requiring manual coding, making it ideal for this use case.
D . AWS Identity and Access Management (IAM):
Using IAM policies allows fine-grained control over access to PII data, ensuring that only authorized users can view or process sensitive data during the pipeline stages.
Alternatives Considered:
B (Amazon GuardDuty): GuardDuty is for threat detection and does not handle data masking or access control for PII.
C (Amazon Macie): Macie can help discover sensitive data but does not handle the masking of PII or access control.
E (Custom scripts): Custom scripting increases the operational burden compared to a built-in solution like DataBrew.
AWS Glue DataBrew for Data Masking
IAM Policies for PII Access Control

QUESTION 36
A data engineer needs to onboard a new data producer into AWS. The data producer needs to migrate data products to AWS.
The data producer maintains many data pipelines that support a business application. Each pipeline must have service accounts and their corresponding credentials. The data engineer must establish a secure connection from the data producer's on-premises data center to AWS. The data engineer must not use the public internet to transfer data from an on-premises data center to AWS.
Which solution will meet these requirements?

A. Instruct the new data producer to create Amazon Machine Images (AMIs) on Amazon Elastic Container Service (Amazon ECS) to store the code base of the application. Create security groups in a public subnet that allow connections only to the on-premises data center.
B. Create an AWS Direct Connect connection to the on-premises data center. Store the service account credentials in AWS Secrets manager.
C. Create a security group in a public subnet. Configure the security group to allow only connections from the CIDR blocks that correspond to the data producer. Create Amazon S3 buckets than contain presigned URLS that have one-day expiration dates.
D. Create an AWS Direct Connect connection to the on-premises data center. Store the application keys in AWS Secrets Manager. Create Amazon S3 buckets that contain resigned URLS that have one-day expiration dates.

Correct Answer: B
Section:
Explanation:
For secure migration of data from an on-premises data center to AWS without using the public internet, AWS Direct Connect is the most secure and reliable method. Using Secrets Manager to store service account credentials ensures that the credentials are managed securely with automatic rotation.
AWS Direct Connect:
Direct Connect establishes a dedicated, private connection between the on-premises data center and AWS, avoiding the public internet. This is ideal for secure, high-speed data transfers.
AWS Secrets Manager:
Secrets Manager securely stores and rotates service account credentials, reducing operational overhead while ensuring security.
Alternatives Considered:
A (ECS with security groups): This does not address the need for a secure, private connection from the on-premises data center.
C (Public subnet with presigned URLs): This involves using the public internet, which does not meet the requirement.
D (Direct Connect with presigned URLs): While Direct Connect is correct, presigned URLs with short expiration dates are unnecessary for this use case.
AWS Direct Connect Documentation
AWS Secrets Manager Documentation

QUESTION 37
A company uses AWS Glue Data Catalog to index data that is uploaded to an Amazon S3 bucket every day. The company uses a daily batch processes in an extract, transform, and load (ETL) pipeline to upload data from external sources into the S3 bucket.
The company runs a daily report on the S3 data. Some days, the company runs the report before all the daily data has been uploaded to the S3 bucket. A data engineer must be able to send a message that identifies any incomplete data to an existing Amazon Simple Notification Service (Amazon SNS) topic.
Which solution will meet this requirement with the LEAST operational overhead?

A. Create data quality checks for the source datasets that the daily reports use. Create a new AWS managed Apache Airflow cluster. Run the data quality checks by using Airflow tasks that run data quality queries on the columns data type and the presence of null values. Configure Airflow Directed Acyclic Graphs (DAGs) to send an email notification that informs the data engineer about the incomplete datasets to the SNS topic.
B. Create data quality checks on the source datasets that the daily reports use. Create a new Amazon EMR cluster. Use Apache Spark SQL to create Apache Spark jobs in the EMR cluster that run data quality queries on the columns data type and the presence of null values. Orchestrate the ETL pipeline by using an AWS Step Functions workflow. Configure the workflow to send an email notification that informs the data engineer about the incomplete datasets to the SNS topic.
C. Create data quality checks on the source datasets that the daily reports use. Create data quality actions by using AWS Glue workflows to confirm the completeness and consistency of the datasets. Configure the data quality actions to create an event in Amazon EventBridge if a dataset is incomplete. Configure EventBridge to send the event that informs the data engineer about the incomplete datasets to the Amazon SNS topic.
D. Create AWS Lambda functions that run data quality queries on the columns data type and the presence of null values. Orchestrate the ETL pipeline by using an AWS Step Functions workflow that runs the Lambda functions. Configure the Step Functions workflow to send an email notification that informs the data engineer about the incomplete datasets to the SNS topic.

Correct Answer: C
Section:
Explanation:
AWS Glue workflows are designed to orchestrate the ETL pipeline, and you can create data quality checks to ensure the uploaded datasets are complete before running reports. If there is an issue with the data, AWS Glue workflows can trigger an Amazon EventBridge event that sends a message to an SNS topic.
AWS Glue Workflows:
AWS Glue workflows allow users to automate and monitor complex ETL processes. You can include data quality actions to check for null values, data types, and other consistency checks.
In the event of incomplete data, an EventBridge event can be generated to notify via SNS.
Alternatives Considered:
A (Airflow cluster): Managed Airflow introduces more operational overhead and complexity compared to Glue workflows.
B (EMR cluster): Setting up an EMR cluster is also more complex compared to the Glue-centric solution.
D (Lambda functions): While Lambda functions can work, using Glue workflows offers a more integrated and lower operational overhead solution.
AWS Glue Workflow Documentation

QUESTION 38
Two developers are working on separate application releases. The developers have created feature branches named Branch A and Branch B by using a GitHub repository's master branch as the source.
The developer for Branch A deployed code to the production system. The code for Branch B will merge into a master branch in the following week's scheduled application release.
Which command should the developer for Branch B run before the developer raises a pull request to the master branch?

A. git diff branchB master git commit -m <message>
B. git pull master
C. git rebase master
D. git fetch -b master

Correct Answer: C
Section:
Explanation:
To ensure that Branch B is up to date with the latest changes in the master branch before submitting a pull request, the correct approach is to perform a git rebase. This command rewrites the commit history so that Branch B will be based on the latest changes in the master branch.
git rebase master:
This command moves the commits of Branch B to be based on top of the latest state of the master branch. It allows the developer to resolve any conflicts and create a clean history.
Alternatives Considered:
A (git diff): This will only show differences between Branch B and master but won't resolve conflicts or bring Branch B up to date.
B (git pull master): Pulling the master branch directly does not offer the same clean history management as rebase.
D (git fetch -b): This is an incorrect command.
Git Rebase Best Practices

QUESTION 39

A healthcare company uses Amazon Kinesis Data Streams to stream real-time health data from wearable devices, hospital equipment, and patient records.
A data engineer needs to find a solution to process the streaming data. The data engineer needs to store the data in an Amazon Redshift Serverless warehouse. The solution must support near real-time analytics of the streaming data and the previous day's data.
Which solution will meet these requirements with the LEAST operational overhead?

A. Load data into Amazon Kinesis Data Firehose. Load the data into Amazon Redshift.
B. Use the streaming ingestion feature of Amazon Redshift.
C. Load the data into Amazon S3. Use the COPY command to load the data into Amazon Redshift.
D. Use the Amazon Aurora zero-ETL integration with Amazon Redshift.

Correct Answer: B
Section:
Explanation:
The streaming ingestion feature of Amazon Redshift enables you to ingest data from streaming sources, such as Amazon Kinesis Data Streams, into Amazon Redshift tables in near real-time. You can use the streaming ingestion feature to process the streaming data from the wearable devices, hospital equipment, and patient records. The streaming ingestion feature also supports incremental updates, which means you can append new data or update existing data in the Amazon Redshift tables. This way, you can store the data in an Amazon Redshift Serverless warehouse and support near real-time analytics of the streaming data and the previous day's data. This solution meets the requirements with the least operational overhead, as it does not require any additional services or components to ingest and process the streaming data. The other options are either not feasible or not optimal. Loading data into Amazon Kinesis Data Firehose and then into Amazon Redshift (option A) would introduce additional latency and cost, as well as require additional configuration and management. Loading data into Amazon S3 and then using the COPY command to load the data into Amazon Redshift (option C) would also introduce additional latency and cost, as well as require additional storage space and ETL logic. Using the Amazon Aurora zero-ETL integration with Amazon Redshift (option D) would not work, as it requires the data to be stored in Amazon Aurora first, which is not the case for the streaming data from the healthcare company.Reference:
Using streaming ingestion with Amazon Redshift
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide, Chapter 3: Data Ingestion and Transformation, Section 3.5: Amazon Redshift Streaming Ingestion

QUESTION 40
A data engineer needs to use an Amazon QuickSight dashboard that is based on Amazon Athena queries on data that is stored in an Amazon S3 bucket. When the data engineer connects to the QuickSight dashboard, the data engineer receives an error message that indicates insufficient permissions.
Which factors could cause to the permissions-related errors? (Choose two.)

A. There is no connection between QuickSgqht and Athena.
B. The Athena tables are not cataloged.
C. QuickSiqht does not have access to the S3 bucket.
D. QuickSight does not have access to decrypt S3 data.
E. There is no 1AM role assigned to QuickSiqht.

Correct Answer: C, D
Section:
Explanation:
QuickSight does not have access to the S3 bucket and QuickSight does not have access to decrypt S3 data are two possible factors that could cause the permissions-related errors. Amazon QuickSight is a business intelligence service that allows you to create and share interactive dashboards based on various data sources, including Amazon Athena. Amazon Athena is a serverless query service that allows you to analyze data stored in Amazon S3 using standard SQL. To use an Amazon QuickSight dashboard that is based on Amazon Athena queries on data that is stored in an Amazon S3 bucket, you need to grant QuickSight access to both Athena and S3, as well as any encryption keys that are used to encrypt the S3 data. If QuickSight does not have access to the S3 bucket or the encryption keys, it will not be able to read the data from Athena and display it on the dashboard, resulting in an error message that indicates insufficient permissions.
The other options are not factors that could cause the permissions-related errors. Option A, there is no connection between QuickSight and Athena, is not a factor, as QuickSight supports Athena as a native data source, and you can easily create a connection between them using the QuickSight console or the API. Option B, the Athena tables are not cataloged, is not a factor, as QuickSight can automatically discover the Athena tables that are cataloged in the AWS Glue Data Catalog, and you can also manually specify the Athena tables that are not cataloged. Option E, there is no IAM role assigned to QuickSight, is not a factor, as QuickSight requires an IAM role to access any AWS data sources, including Athena and S3, and you can create and assign an IAM role to QuickSight using the QuickSight console or the API.Reference:
Using Amazon Athena as a Data Source

Granting Amazon QuickSight Access to AWS Resources
Encrypting Data at Rest in Amazon S3

QUESTION 41
A company stores datasets in JSON format and .csv format in an Amazon S3 bucket. The company has Amazon RDS for Microsoft SQL Server databases, Amazon DynamoDB tables that are in provisioned capacity mode, and an Amazon Redshift cluster. A data engineering team must develop a solution that will give data scientists the ability to query all data sources by using syntax similar to SQL.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use Amazon Athena to query the data. Use SQL for structured data sources. Use PartiQL for data that is stored in JSON format.
B. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use Redshift Spectrum to query the data. Use SQL for structured data sources. Use PartiQL for data that is stored in JSON format.
C. Use AWS Glue to crawl the data sources. Store metadata in the AWS Glue Data Catalog. Use AWS Glue jobs to transform data that is in JSON format to Apache Parquet or .csv format. Store the transformed data in an S3 bucket. Use Amazon Athena to query the original and transformed data from the S3 bucket.
D. Use AWS Lake Formation to create a data lake. Use Lake Formation jobs to transform the data from all data sources to Apache Parquet format. Store the transformed data in an S3 bucket. Use Amazon Athena or Redshift Spectrum to query the data.

Correct Answer: A
Section:
Explanation:
The best solution to meet the requirements of giving data scientists the ability to query all data sources by using syntax similar to SQL with the least operational overhead is to use AWS Glue to crawl the data sources, store metadata in the AWS Glue Data Catalog, use Amazon Athena to query the data, use SQL for structured data sources, and use PartiQL for data that is stored in JSON format.
AWS Glue is a serverless data integration service that makes it easy to prepare, clean, enrich, and move data between data stores1.AWS Glue crawlers are processes that connect to a data store, progress through a prioritized list of classifiers to determine the schema for your data, and then create metadata tables in the Data Catalog2.The Data Catalog is a persistent metadata store that contains table definitions, job definitions, and other control information to help you manage your AWS Glue components3. You can use AWS Glue to crawl the data sources, such as Amazon S3, Amazon RDS for Microsoft SQL Server, and Amazon DynamoDB, and store the metadata in the Data Catalog.
Amazon Athena is a serverless, interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL or Python4. Amazon Athena also supports PartiQL, a SQL-compatible query language that lets you query, insert, update, and delete data from semi-structured and nested data, such as JSON. You can use Amazon Athena to query the data from the Data Catalog using SQL for structured data sources, such as .csv files and relational databases, and PartiQL for data that is stored in JSON format. You can also use Athena to query data from other data sources, such as Amazon Redshift, using federated queries.
Using AWS Glue and Amazon Athena to query all data sources by using syntax similar to SQL is the least operational overhead solution, as you do not need to provision, manage, or scale any infrastructure, and you pay only for the resources you use.AWS Glue charges you based on the compute time and the data processed by your crawlers and ETL jobs1. Amazon Athena charges you based on the amount of data scanned by your queries. You can also reduce the cost and improve the performance of your queries by using compression, partitioning, and columnar formats for your data in Amazon S3.
Option B is not the best solution, as using AWS Glue to crawl the data sources, store metadata in the AWS Glue Data Catalog, and use Redshift Spectrum to query the data, would incur more costs and complexity than using Amazon Athena. Redshift Spectrum is a feature of Amazon Redshift, a fully managed data warehouse service, that allows you to query and join data across your data warehouse and your data lake using standard SQL. While Redshift Spectrum is powerful and useful for many data warehousing scenarios, it is not necessary or cost-effective for querying all data sources by using syntax similar to SQL.Redshift Spectrum charges you based on the amount of data scanned by your queries, which is similar to Amazon Athena, but it also requires you to have an Amazon Redshift cluster, which charges you based on the node type, the number of nodes, and the duration of the cluster5. These costs can add up quickly, especially if you have large volumes of data and complex queries. Moreover, using Redshift Spectrum would introduce additional latency and complexity, as you would have to provision and manage the cluster, and create an external schema and database for the data in the Data Catalog, instead of querying it directly from Amazon Athena.
Option C is not the best solution, as using AWS Glue to crawl the data sources, store metadata in the AWS Glue Data Catalog, use AWS Glue jobs to transform data that is in JSON format to Apache Parquet or .csv format, store the transformed data in an S3 bucket, and use Amazon Athena to query the original and transformed data from the S3 bucket, would incur more costs and complexity than using Amazon Athena with PartiQL. AWS Glue jobs are ETL scripts that you can write in Python or Scala to transform your data and load it to your target data store.Apache Parquet is a columnar storage format that can improve the performance of analytical queries by reducing the amount of data that needs to be scanned and providing efficient compression and encoding schemes6. While using AWS Glue jobs and Parquet can improve the performance and reduce the cost of your queries, they would also increase the complexity and the operational overhead of the data pipeline, as you would have to write, run, and monitor the ETL jobs, and store the transformed data in a separate location in Amazon S3. Moreover, using AWS Glue jobs and Parquet would introduce additional latency, as you would have to wait for the ETL jobs to finish before querying the transformed data.
Option D is not the best solution, as using AWS Lake Formation to create a data lake, use Lake Formation jobs to transform the data from all data sources to Apache Parquet format, store the transformed data in an S3 bucket, and use Amazon Athena or Redshift Spectrum to query the data, would incur more costs and complexity than using Amazon Athena with PartiQL.AWS Lake Formation is a service that helps you centrally govern, secure, and globally share data for analytics and machine learning7. Lake Formation jobs are ETL jobs that you can create and run using the Lake Formation console or API. While using Lake Formation and Parquet can improve the performance and reduce the cost of your queries, they would also increase the complexity and the operational overhead of the data pipeline, as you would have to create, run, and monitor the Lake Formation jobs, and store the transformed data in a separate location in Amazon S3. Moreover, using Lake Formation and Parquet would introduce additional latency, as you would have to wait for the Lake Formation jobs to finish before querying the transformed data. Furthermore, using Redshift Spectrum to query the data would also incur the same costs and complexity as mentioned in option B.Reference:
What is Amazon Athena?

Data Catalog and crawlers in AWS Glue
AWS Glue Data Catalog
Columnar Storage Formats
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide
AWS Glue Schema Registry
What is AWS Glue?
Amazon Redshift Serverless
Amazon Redshift provisioned clusters
[Querying external data using Amazon Redshift Spectrum]
[Using stored procedures in Amazon Redshift]
[What is AWS Lambda?]
[PartiQL for Amazon Athena]
[Federated queries in Amazon Athena]
[Amazon Athena pricing]
[Top 10 performance tuning tips for Amazon Athena]
[AWS Glue ETL jobs]
[AWS Lake Formation jobs]

QUESTION 42
A data engineer is configuring Amazon SageMaker Studio to use AWS Glue interactive sessions to prepare data for machine learning (ML) models.
The data engineer receives an access denied error when the data engineer tries to prepare the data by using SageMaker Studio.
Which change should the engineer make to gain access to SageMaker Studio?

A. Add the AWSGlueServiceRole managed policy to the data engineer's IAM user.
B. Add a policy to the data engineer's IAM user that includes the sts:AssumeRole action for the AWS Glue and SageMaker service principals in the trust policy.
C. Add the AmazonSageMakerFullAccess managed policy to the data engineer's IAM user.
D. Add a policy to the data engineer's IAM user that allows the sts:AddAssociation action for the AWS Glue and SageMaker service principals in the trust policy.

Correct Answer: B
Section:
Explanation:
This solution meets the requirement of gaining access to SageMaker Studio to use AWS Glue interactive sessions. AWS Glue interactive sessions are a way to use AWS Glue DataBrew and AWS Glue Data Catalog from within SageMaker Studio. To use AWS Glue interactive sessions, the data engineer's IAM user needs to have permissions to assume the AWS Glue service role and the SageMaker execution role. By adding a policy to the data engineer's IAM user that includes the sts:AssumeRole action for the AWS Glue and SageMaker service principals in the trust policy, the data engineer can grant these permissions and avoid the access denied error. The other options are not sufficient or necessary to resolve the error.Reference:
Get started with data integration from Amazon S3 to Amazon Redshift using AWS Glue interactive sessions
Troubleshoot Errors - Amazon SageMaker
AccessDeniedException on sagemaker:CreateDomain in AWS SageMaker Studio, despite having SageMakerFullAccess

QUESTION 43
A company extracts approximately 1 TB of data every day from data sources such as SAP HANA, Microsoft SQL Server, MongoDB, Apache Kafka, and Amazon DynamoDB. Some of the data sources have undefined data schemas or data schemas that change.
A data engineer must implement a solution that can detect the schema for these data sources. The solution must extract, transform, and load the data to an Amazon S3 bucket. The company has a service level agreement (SLA) to load the data into the S3 bucket within 15 minutes of data creation.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon EMR to detect the schema and to extract, transform, and load the data into the S3 bucket. Create a pipeline in Apache Spark.
B. Use AWS Glue to detect the schema and to extract, transform, and load the data into the S3 bucket. Create a pipeline in Apache Spark.
C. Create a PvSpark proqram in AWS Lambda to extract, transform, and load the data into the S3 bucket.

D. Create a stored procedure in Amazon Redshift to detect the schema and to extract, transform, and load the data into a Redshift Spectrum table. Access the table from Amazon S3.

Correct Answer: B
Section:
Explanation:
AWS Glue is a fully managed service that provides a serverless data integration platform. It can automatically discover and categorize data from various sources, including SAP HANA, Microsoft SQL Server, MongoDB, Apache Kafka, and Amazon DynamoDB. It can also infer the schema of the data and store it in the AWS Glue Data Catalog, which is a central metadata repository. AWS Glue can then use the schema information to generate and run Apache Spark code to extract, transform, and load the data into an Amazon S3 bucket. AWS Glue can also monitor and optimize the performance and cost of the data pipeline, and handle any schema changes that may occur in the source data. AWS Glue can meet the SLA of loading the data into the S3 bucket within 15 minutes of data creation, as it can trigger the data pipeline based on events, schedules, or on-demand. AWS Glue has the least operational overhead among the options, as it does not require provisioning, configuring, or managing any servers or clusters. It also handles scaling, patching, and security automatically.Reference:
AWS Glue
[AWS Glue Data Catalog]
[AWS Glue Developer Guide]
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide

QUESTION 44
A company has multiple applications that use datasets that are stored in an Amazon S3 bucket. The company has an ecommerce application that generates a dataset that contains personally identifiable information (PII). The company has an internal analytics application that does not require access to the PII.
To comply with regulations, the company must not share PII unnecessarily. A data engineer needs to implement a solution that with redact PII dynamically, based on the needs of each application that accesses the dataset.
Which solution will meet the requirements with the LEAST operational overhead?

A. Create an S3 bucket policy to limit the access each application has. Create multiple copies of the dataset. Give each dataset copy the appropriate level of redaction for the needs of the application that accesses the copy.
B. Create an S3 Object Lambda endpoint. Use the S3 Object Lambda endpoint to read data from the S3 bucket. Implement redaction logic within an S3 Object Lambda function to dynamically redact PII based on the needs of each application that accesses the data.
C. Use AWS Glue to transform the data for each application. Create multiple copies of the dataset. Give each dataset copy the appropriate level of redaction for the needs of the application that accesses the copy.
D. Create an API Gateway endpoint that has custom authorizers. Use the API Gateway endpoint to read data from the S3 bucket. Initiate a REST API call to dynamically redact PII based on the needs of each application that accesses the data.

Correct Answer: B
Section:
Explanation:
Option B is the best solution to meet the requirements with the least operational overhead because S3 Object Lambda is a feature that allows you to add your own code to process data retrieved from S3 before returning it to an application. S3 Object Lambda works with S3 GET requests and can modify both the object metadata and the object data. By using S3 Object Lambda, you can implement redaction logic within an S3 Object Lambda function to dynamically redact PII based on the needs of each application that accesses the data. This way, you can avoid creating and maintaining multiple copies of the dataset with different levels of redaction.
Option A is not a good solution because it involves creating and managing multiple copies of the dataset with different levels of redaction for each application. This option adds complexity and storage cost to the data protection process and requires additional resources and configuration. Moreover, S3 bucket policies cannot enforce fine-grained data access control at the row and column level, so they are not sufficient to redact PII.
Option C is not a good solution because it involves using AWS Glue to transform the data for each application. AWS Glue is a fully managed service that can extract, transform, and load (ETL) data from various sources to various destinations, including S3. AWS Glue can also convert data to different formats, such as Parquet, which is a columnar storage format that is optimized for analytics. However, in this scenario, using AWS Glue to redact PII is not the best option because it requires creating and maintaining multiple copies of the dataset with different levels of redaction for each application. This option also adds extra time and cost to the data protection process and requires additional resources and configuration.
Option D is not a good solution because it involves creating and configuring an API Gateway endpoint that has custom authorizers. API Gateway is a service that allows you to create, publish, maintain, monitor, and secure APIs at any scale. API Gateway can also integrate with other AWS services, such as Lambda, to provide custom logic for processing requests. However, in this scenario, using API Gateway to redact PII is not the best option because it requires writing and maintaining custom code and configuration for the API endpoint, the custom authorizers, and the REST API call. This option also adds complexity and latency to the data protection process and requires additional resources and configuration.
AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide
Introducing Amazon S3 Object Lambda -- Use Your Code to Process Data as It Is Being Retrieved from S3
Using Bucket Policies and User Policies - Amazon Simple Storage Service
AWS Glue Documentation
What is Amazon API Gateway? - Amazon API Gateway

QUESTION 45
A data engineer needs to build an extract, transform, and load (ETL) job. The ETL job will process daily incoming .csv files that users upload to an Amazon S3 bucket. The size of each S3 object is less than 100 MB.
Which solution will meet these requirements MOST cost-effectively?

A. Write a custom Python application. Host the application on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster.
B. Write a PySpark ETL script. Host the script on an Amazon EMR cluster.
C. Write an AWS Glue PySpark job. Use Apache Spark to transform the data.
D. Write an AWS Glue Python shell job. Use pandas to transform the data.

Correct Answer: D
Section:
Explanation:
AWS Glue is a fully managed serverless ETL service that can handle various data sources and formats, including .csv files in Amazon S3. AWS Glue provides two types of jobs: PySpark and Python shell. PySpark jobs use Apache Spark to process large-scale data in parallel, while Python shell jobs use Python scripts to process small-scale data in a single execution environment. For this requirement, a Python shell job is more suitable and cost-effective, as the size of each S3 object is less than 100 MB, which does not require distributed processing. A Python shell job can use pandas, a popular Python library for data analysis, to transform the .csv data as needed. The other solutions are not optimal or relevant for this requirement. Writing a custom Python application and hosting it on an Amazon EKS cluster would require more effort and resources to set up and manage the Kubernetes environment, as well as to handle the data ingestion and transformation logic. Writing a PySpark ETL script and hosting it on an Amazon EMR cluster would also incur more costs and complexity to provision and configure the EMR cluster, as well as to use Apache Spark for processing small data files. Writing an AWS Glue PySpark job would also be less efficient and economical than a Python shell job, as it would involve unnecessary overhead and charges for using Apache Spark for small data files.Reference:
AWS Glue
Working with Python Shell Jobs
pandas
[AWS Certified Data Engineer - Associate DEA-C01 Complete Study Guide]

QUESTION 46
The company stores a large volume of customer records in Amazon S3. To comply with regulations, the company must be able to access new customer records immediately for the first 30 days after the records are created.
The company accesses records that are older than 30 days infrequently.
The company needs to cost-optimize its Amazon S3 storage.
Which solution will meet these requirements MOST cost-effectively?

A. Apply a lifecycle policy to transition records to S3 Standard Infrequent-Access (S3 Standard-IA) storage after 30 days.
B. Use S3 Intelligent-Tiering storage.
C. Transition records to S3 Glacier Deep Archive storage after 30 days.
D. Use S3 Standard-Infrequent Access (S3 Standard-IA) storage for all customer records.

Correct Answer: A
Section:
Explanation:
The most cost-effective solution in this case is to apply a lifecycle policy to transition records to Amazon S3 Standard-IA storage after 30 days. Here's why:
Amazon S3 Lifecycle Policies: Amazon S3 offers lifecycle policies that allow you to automatically transition objects between different storage classes to optimize costs. For data that is frequently accessed in the first 30 days and infrequently accessed after that, transitioning from the S3 Standard storage class to S3 Standard-Infrequent Access (S3 Standard-IA) after 30 days makes the most sense. S3 Standard-IA is designed for data that is accessed less frequently but still needs to be retained, offering lower storage costs than S3 Standard with a retrieval cost for access.
Cost Optimization: S3 Standard-IA offers a lower price per GB than S3 Standard. Since the data will be accessed infrequently after 30 days, using S3 Standard-IA will lower storage costs while still allowing for immediate retrieval when necessary.
Compliance with Regulations: Since the records need to be immediately accessible for the first 30 days, the use of S3 Standard for that period ensures compliance with regulatory requirements. After 30 days, transitioning to S3 Standard-IA continues to meet access requirements for infrequent access while reducing storage costs.
Alternatives Considered:
Option B (S3 Intelligent-Tiering): While S3 Intelligent-Tiering automatically moves data between access tiers based on access patterns, it incurs a small monthly monitoring and automation charge per object. It could be a viable option, but transitioning data to S3 Standard-IA directly would be more cost-effective since the pattern of access is well-known (frequent for 30 days, infrequent thereafter).

Option C (S3 Glacier Deep Archive): Glacier Deep Archive is the lowest-cost storage class, but it is not suitable in this case because the data needs to be accessed immediately within 30 days and on an infrequent basis thereafter. Glacier Deep Archive requires hours for data retrieval, which is not acceptable for infrequent access needs.
Option D (S3 Standard-IA for all records): Using S3 Standard-IA for all records would result in higher costs for the first 30 days, as the data is frequently accessed. S3 Standard-IA incurs retrieval charges, making it less suitable for frequently accessed data.
Amazon S3 Lifecycle Policies
S3 Storage Classes
Cost Management and Data Optimization Using Lifecycle Policies
AWS Data Engineering Documentation

QUESTION 47
A retail company is expanding its operations globally. The company needs to use Amazon QuickSight to accurately calculate currency exchange rates for financial reports. The company has an existing dashboard that includes a visual that is based on an analysis of a dataset that contains global currency values and exchange rates.
A data engineer needs to ensure that exchange rates are calculated with a precision of four decimal places. The calculations must be precomputed. The data engineer must materialize results in QuickSight super-fast, parallel, in-memory calculation engine (SPICE).
Which solution will meet these requirements?

A. Define and create the calculated field in the dataset.
B. Define and create the calculated field in the analysis.
C. Define and create the calculated field in the visual.
D. Define and create the calculated field in the dashboard.

Correct Answer: A
Section:

QUESTION 48
A retail company uses an Amazon Redshift data warehouse and an Amazon S3 bucket. The company ingests retail order data into the S3 bucket every day.
The company stores all order data at a single path within the S3 bucket. The data has more than 100 columns. The company ingests the order data from a third-party application that generates more than 30 files in CSV format every day. Each CSV file is between 50 and 70 MB in size.
The company uses Amazon Redshift Spectrum to run queries that select sets of columns. Users aggregate metrics based on daily orders. Recently, users have reported that the performance of the queries has degraded. A data engineer must resolve the performance issues for the queries.
Which combination of steps will meet this requirement with LEAST developmental effort? (Select TWO.)

A. Configure the third-party application to create the files in a columnar format.
B. Develop an AWS Glue ETL job to convert the multiple daily CSV files to one file for each day.
C. Partition the order data in the S3 bucket based on order date.
D. Configure the third-party application to create the files in JSON format.
E. Load the JSON data into the Amazon Redshift table in a SUPER type column.

Correct Answer: A, C
Section:
Explanation:
The performance issue in Amazon Redshift Spectrum queries arises due to the nature of CSV files, which are row-based storage formats. Spectrum is more optimized for columnar formats, which significantly improve performance by reducing the amount of data scanned. Also, partitioning data based on relevant columns like order date can further reduce the amount of data scanned, as queries can focus only on the necessary partitions.
A . Configure the third-party application to create the files in a columnar format:
Columnar formats (like Parquet or ORC) store data in a way that is optimized for analytical queries because they allow queries to scan only the columns required, rather than scanning all columns in a row-based format like CSV.
Amazon Redshift Spectrum works much more efficiently with columnar formats, reducing the amount of data that needs to be scanned, which improves query performance.
C . Partition the order data in the S3 bucket based on order date:

Partitioning the data on columns like order date allows Redshift Spectrum to skip scanning unnecessary partitions, leading to improved query performance.
By organizing data into partitions, you minimize the number of files Spectrum has to read, further optimizing performance.
Alternatives Considered:
B (Develop an AWS Glue ETL job): While consolidating files can improve performance by reducing the number of small files (which can be inefficient to process), it adds additional ETL complexity. Switching to a columnar format (Option A) and partitioning (Option C) provides more significant performance improvements with less development effort.
D and E (JSON-related options): Using JSON format or the SUPER type in Redshift introduces complexity and isn't as efficient as the proposed solutions, especially since JSON is not a columnar format.
Amazon Redshift Spectrum Documentation
Columnar Formats and Data Partitioning in S3

QUESTION 49
A technology company currently uses Amazon Kinesis Data Streams to collect log data in real time. The company wants to use Amazon Redshift for downstream real-time queries and to enrich the log data.
Which solution will ingest data into Amazon Redshift with the LEAST operational overhead?

A. Set up an Amazon Data Firehose delivery stream to send data to a Redshift provisioned cluster table.
B. Set up an Amazon Data Firehose delivery stream to send data to Amazon S3. Configure a Redshift provisioned cluster to load data every minute.
C. Configure Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to send data directly to a Redshift provisioned cluster table.
D. Use Amazon Redshift streaming ingestion from Kinesis Data Streams and to present data as a materialized view.

Correct Answer: D
Section:
Explanation:
The most efficient and low-operational-overhead solution for ingesting data into Amazon Redshift from Amazon Kinesis Data Streams is to use Amazon Redshift streaming ingestion. This feature allows Redshift to directly ingest streaming data from Kinesis Data Streams and process it in real-time.
Amazon Redshift Streaming Ingestion:
Redshift supports native streaming ingestion from Kinesis Data Streams, allowing real-time data to be queried using materialized views.
This solution reduces operational complexity because you don't need intermediary services like Amazon Kinesis Data Firehose or S3 for batch loading.
Alternatives Considered:
A (Data Firehose to Redshift): This option is more suitable for batch processing but incurs additional operational overhead with the Firehose setup.
B (Firehose to S3): This involves an intermediate step, which adds complexity and delays the real-time requirement.
C (Managed Service for Apache Flink): This would work but introduces unnecessary complexity compared to Redshift's native streaming ingestion.
Amazon Redshift Streaming Ingestion from Kinesis
Materialized Views in Redshift

QUESTION 50
A company has three subsidiaries. Each subsidiary uses a different data warehousing solution. The first subsidiary hosts its data warehouse in Amazon Redshift. The second subsidiary uses Teradata Vantage on AWS. The third subsidiary uses Google BigQuery.
The company wants to aggregate all the data into a central Amazon S3 data lake. The company wants to use Apache Iceberg as the table format.
A data engineer needs to build a new pipeline to connect to all the data sources, run transformations by using each source engine, join the data, and write the data to Iceberg.
Which solution will meet these requirements with the LEAST operational effort?

A. Use native Amazon Redshift, Teradata, and BigQuery connectors to build the pipeline in AWS Glue. Use native AWS Glue transforms to join the data. Run a Merge operation on the data lake Iceberg table.
B. Use the Amazon Athena federated query connectors for Amazon Redshift, Teradata, and BigQuery to build the pipeline in Athena. Write a SQL query to read from all the data sources, join the data, and run a Merge operation on the data lake Iceberg table.
C. Use the native Amazon Redshift connector, the Java Database Connectivity (JDBC) connector for Teradata, and the open source Apache Spark BigQuery connector to build the pipeline in Amazon EMR. Write code in PySpark to join the data. Run a Merge operation on the data lake Iceberg table.
D. Use the native Amazon Redshift, Teradata, and BigQuery connectors in Amazon Appflow to write data to Amazon S3 and AWS Glue Data Catalog. Use Amazon Athena to join the data. Run a Merge operation on the data lake Iceberg table.

Correct Answer: B

Section:
Explanation:
Amazon Athena provides federated query connectors that allow querying multiple data sources, such as Amazon Redshift, Teradata, and Google BigQuery, without needing to extract the data from the original source. This solution is optimal because it offers the least operational effort by avoiding complex data movement and transformation processes.
Amazon Athena Federated Queries:
Athena's federated queries allow direct querying of data stored across multiple sources, including Amazon Redshift, Teradata, and BigQuery. With Athena's support for Apache Iceberg, the company can easily run a Merge operation on the Iceberg table.
The solution reduces complexity by centralizing the query execution and transformation process in Athena using SQL queries.
Alternatives Considered:
A (AWS Glue pipeline): This would work but requires more operational effort to manage and transform the data in AWS Glue.
C (Amazon EMR): Using EMR and writing PySpark code introduces more operational overhead and complexity compared to a SQL-based solution in Athena.
D (Amazon AppFlow): AppFlow is more suitable for transferring data between services but is not as efficient for transformations and joins as Athena federated queries.
Amazon Athena Documentation
Federated Queries in Amazon Athena

QUESTION 51
A company has a data lake in Amazon S3. The company collects AWS CloudTrail logs for multiple applications. The company stores the logs in the data lake, catalogs the logs in AWS Glue, and partitions the logs based on the year. The company uses Amazon Athena to analyze the logs.
Recently, customers reported that a query on one of the Athena tables did not return any data. A data engineer must resolve the issue.
Which combination of troubleshooting steps should the data engineer take? (Select TWO.)

A. Confirm that Athena is pointing to the correct Amazon S3 location.
B. Increase the query timeout duration.
C. Use the MSCK REPAIR TABLE command.
D. Restart Athena.
E. Delete and recreate the problematic Athena table.

Correct Answer: A, C
Section:
Explanation:
The problem likely arises from Athena not being able to read from the correct S3 location or missing partitions. The two most relevant troubleshooting steps involve checking the S3 location and repairing the table metadata.
A . Confirm that Athena is pointing to the correct Amazon S3 location:
One of the most common issues with missing data in Athena queries is that the query is pointed to an incorrect or outdated S3 location. Checking the S3 path ensures Athena is querying the correct data.
C . Use the MSCK REPAIR TABLE command:
When new partitions are added to the S3 bucket without being reflected in the Glue Data Catalog, Athena queries will not return data from those partitions. The MSCK REPAIR TABLE command updates the Glue Data Catalog with the latest partitions.
Alternatives Considered:
B (Increase query timeout): Timeout issues are unrelated to missing data.
D (Restart Athena): Athena does not require restarting.
E (Delete and recreate table): This introduces unnecessary overhead when the issue can be resolved by repairing the table and confirming the S3 location.
Athena Query Fails to Return Data

QUESTION 52
A company is using an AWS Transfer Family server to migrate data from an on-premises environment to AWS. Company policy mandates the use of TLS 1.2 or above to encrypt the data in transit.
Which solution will meet these requirements?

A. Generate new SSH keys for the Transfer Family server. Make the old keys and the new keys available for use.
B. Update the security group rules for the on-premises network to allow only connections that use TLS 1.2 or above.
C. Update the security policy of the Transfer Family server to specify a minimum protocol version of TLS 1.2.

D. Install an SSL certificate on the Transfer Family server to encrypt data transfers by using TLS 1.2.

Correct Answer: C
Section:
Explanation:
The AWS Transfer Family server's security policy can be updated to enforce TLS 1.2 or higher, ensuring compliance with company policy for encrypted data transfers.
AWS Transfer Family Security Policy:
AWS Transfer Family supports setting a minimum TLS version through its security policy configuration. This ensures that only connections using TLS 1.2 or above are allowed.
Alternatives Considered:
A (Generate new SSH keys): SSH keys are unrelated to TLS and do not enforce encryption protocols like TLS 1.2.
B (Update security group rules): Security groups control IP-level access, not TLS versions.
D (Install SSL certificate): SSL certificates ensure secure connections, but the TLS version is controlled via the security policy.
AWS Transfer Family Documentation

QUESTION 53
A data engineer configured an AWS Glue Data Catalog for data that is stored in Amazon S3 buckets. The data engineer needs to configure the Data Catalog to receive incremental updates.
The data engineer sets up event notifications for the S3 bucket and creates an Amazon Simple Queue Service (Amazon SQS) queue to receive the S3 events.
Which combination of steps should the data engineer take to meet these requirements with LEAST operational overhead? (Select TWO.)

A. Create an S3 event-based AWS Glue crawler to consume events from the SQS queue.
B. Define a time-based schedule to run the AWS Glue crawler, and perform incremental updates to the Data Catalog.
C. Use an AWS Lambda function to directly update the Data Catalog based on S3 events that the SQS queue receives.
D. Manually initiate the AWS Glue crawler to perform updates to the Data Catalog when there is a change in the S3 bucket.
E. Use AWS Step Functions to orchestrate the process of updating the Data Catalog based on 53 events that the SQS queue receives.

Correct Answer: A, C
Section:
Explanation:
The requirement is to update the AWS Glue Data Catalog incrementally based on S3 events. Using an S3 event-based approach is the most automated and operationally efficient solution.
A . Create an S3 event-based AWS Glue crawler:
An event-based Glue crawler can automatically update the Data Catalog when new data arrives in the S3 bucket. This ensures incremental updates with minimal operational overhead.
C . Use an AWS Lambda function to directly update the Data Catalog:
Lambda can be triggered by S3 events delivered to the SQS queue and can directly update the Glue Data Catalog, ensuring that new data is reflected in near real-time without running a full crawler.
Alternatives Considered:
B (Time-based schedule): Scheduling a crawler to run periodically adds unnecessary latency and operational overhead.
D (Manual crawler initiation): Manually starting the crawler defeats the purpose of automation.
E (AWS Step Functions): Step Functions add complexity that is not needed when Lambda can handle the updates directly.
AWS Glue Event-Driven Crawlers
Using AWS Lambda to Update Glue Catalog

QUESTION 54
A company uploads .csv files to an Amazon S3 bucket. The company's data platform team has set up an AWS Glue crawler to perform data discovery and to create the tables and schemas.
An AWS Glue job writes processed data from the tables to an Amazon Redshift database. The AWS Glue job handles column mapping and creates the Amazon Redshift tables in the Redshift database appropriately.
If the company reruns the AWS Glue job for any reason, duplicate records are introduced into the Amazon Redshift tables. The company needs a solution that will update the Redshift tables without duplicates.
Which solution will meet these requirements?

A. Modify the AWS Glue job to copy the rows into a staging Redshift table. Add SQL commands to update the existing rows with new values from the staging Redshift table.
B. Modify the AWS Glue job to load the previously inserted data into a MySQL database. Perform an upsert operation in the MySQL database. Copy the results to the Amazon Redshift tables.

C. Use Apache Spark's DataFrame dropDuplicates() API to eliminate duplicates. Write the data to the Redshift tables.
D. Use the AWS Glue ResolveChoice built-in transform to select the value of the column from the most recent record.

Correct Answer: A
Section:
Explanation:
To avoid duplicate records in Amazon Redshift, the most effective solution is to perform the ETL in a way that first loads the data into a staging table and then uses SQL commands like MERGE or UPDATE to insert new records and update existing records without introducing duplicates.
Using Staging Tables in Redshift:
The AWS Glue job can write data to a staging table in Redshift. Once the data is loaded, SQL commands can be executed to compare the staging data with the target table and update or insert records appropriately. This ensures no duplicates are introduced during re-runs of the Glue job.
Alternatives Considered:
B (MySQL upsert): This introduces unnecessary complexity by involving another database (MySQL).
C (Spark dropDuplicates): While Spark can eliminate duplicates, handling duplicates at the Redshift level with a staging table is a more reliable and Redshift-native solution.
D (AWS Glue ResolveChoice): The ResolveChoice transform in Glue helps with column conflicts but does not handle record-level duplicates effectively.
Amazon Redshift MERGE Statements
Staging Tables in Amazon Redshift

QUESTION 55
A financial company recently added more features to its mobile app. The new features required the company to create a new topic in an existing Amazon Managed Streaming for Apache Kafka (Amazon MSK) cluster.
A few days after the company added the new topic, Amazon CloudWatch raised an alarm on the RootDiskUsed metric for the MSK cluster.
How should the company address the CloudWatch alarm?

A. Expand the storage of the MSK broker. Configure the MSK cluster storage to expand automatically.
B. Expand the storage of the Apache ZooKeeper nodes.
C. Update the MSK broker instance to a larger instance type. Restart the MSK cluster.
D. Specify the Target-Volume-in-GiB parameter for the existing topic.

Correct Answer: A
Section:
Explanation:
The RootDiskUsed metric for the MSK cluster indicates that the storage on the broker is reaching its capacity. The best solution is to expand the storage of the MSK broker and enable automatic storage expansion to prevent future alarms.
Expand MSK Broker Storage:
AWS Managed Streaming for Apache Kafka (MSK) allows you to expand the broker storage to accommodate growing data volumes. Additionally, auto-expansion of storage can be configured to ensure that storage grows automatically as the data increases.
Alternatives Considered:
B (Expand Zookeeper storage): Zookeeper is responsible for managing Kafka metadata and not for storing data, so increasing Zookeeper storage won't resolve the root disk issue.
C (Update instance type): Changing the instance type would increase computational resources but not directly address the storage problem.
D (Target-Volume-in-GiB): This parameter is irrelevant for the existing topic and will not solve the storage issue.
Amazon MSK Storage Auto Scaling