DUMPSARENA

AWS Certified Data Engineer Associate (DEA-C01)

Amazon AWS Data-Engineer-Associate-DEA-C01

Version Demo

Total Demo Questions: 32

Total Premium Questions: 325

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Ingestion and Transformation	128
Topic 2, Data Store Management	84
Topic 3, Data Operations and Support	50
Topic 4, Data Security and Governance	63
Total	325

QUESTION NO: 1

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.

ANSWER: B D

Explanation:

Register the S3 path as an AWS Lake Formation location is correct because Lake Formation must first be made aware of the Amazon S3 location that contains the data lake objects before it can govern access to that data. Registering the location lets Lake Formation manage permissions for the underlying S3 data and integrate those permissions with IAM principals such as the unique HR department roles. Enable fine-grained access control in AWS Lake Formation. Add a data filter for each Region is also correct because Lake Formation data filters can enforce row-level, column-level, and cell-level access controls on Data Catalog tables. By creating a filter for each Region and granting each HR IAM role access through the matching filter, each HR department can query only the employee records for its own Region while the data remains in the same S3-based data lake. This approach centralizes access governance and avoids creating separate buckets or maintaining multiple complex S3 policies, which minimizes operational overhead. See the AWS documentation for registering Amazon S3 locations in Lake Formation and Lake Formation data filtering and fine-grained access control.

QUESTION NO: 2

A company receives marketing campaign data from a vendor. The company ingests the data into an Amazon S3 bucket every 40 to 60 minutes. The data is in CSV format. File sizes are between 100 KB and 300 KB.

A data engineer needs to set-up an extract, transform, and load (ETL) pipeline to upload the content of each file to Amazon Redshift.

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

A. Create an AWS Lambda function that connects to Amazon Redshift and runs a COPY command. Use Amazon EventBridge to invoke the Lambda function based on an Amazon S3 upload trigger.

B. Create an Amazon Data Firehose stream. Configure the stream to use an AWS Lambda function as a source to pull data from the S3 bucket. Set Amazon Redshift as the destination.

C.

Use Amazon Redshift Spectrum to query the S3 bucket. Configure an AWS Glue Crawler for the S3 bucket to update metadata in an AWS Glue Data Catalog.

D. Creates an AWS Database Migration Service (AWS DMS) task. Specify an appropriate data schema to migrate. Specify the appropriate type of migration to use.

ANSWER: A

Explanation:

Create an AWS Lambda function that connects to Amazon Redshift and runs a COPY command. Use Amazon EventBridge to invoke the Lambda function based on an Amazon S3 upload trigger is correct because the workload is small, infrequent, and naturally event driven. When a new CSV object lands in Amazon S3, an S3 object-created event can be routed through Amazon EventBridge to invoke a Lambda function without provisioning or managing servers. The Lambda function can then issue an Amazon Redshift COPY command to load the specific S3 object into the target table. COPY is the recommended and efficient mechanism for loading data from Amazon S3 into Amazon Redshift, and it supports CSV input directly.

This design has very low operational overhead because there is no continuously running ingestion service, cluster, or scheduler to maintain. The files are only 100 KB to 300 KB and arrive roughly once per hour, so Lambda's event-based execution model is a good fit. Amazon S3 supports sending events to EventBridge, and Amazon Redshift documents COPY from Amazon S3 as a standard bulk-loading approach. References: Amazon S3 EventBridge integration and Amazon Redshift COPY from Amazon S3.

QUESTION NO: 3

A company analyzes data in a data lake every quarter to perform inventory assessments. A data engineer uses AWS Glue DataBrew to detect any personally identifiable information (PII) about customers within the data. The company's privacy policy considers some custom categories of information to be PII. However, the categories are not included in standard DataBrew data quality rules.

The data engineer needs to modify the current process to scan for the custom PII categories across multiple datasets within the data lake.

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

A. Manually review the data for custom PII categories.

B. Implement custom data quality rules in Data Brew. Apply the custom rules across datasets.

C. Develop custom Python scripts to detect the custom PII categories. Call the scripts from DataBrew.

D. Implement regex patterns to extract PII information from fields during extract transform, and load (ETL) operations into the data lake.

ANSWER: B

Explanation:

Implement custom data quality rules in Data Brew. Apply the custom rules across datasets. is the correct solution because AWS Glue DataBrew supports reusable data quality rulesets that can be attached to profile jobs and applied consistently across datasets. For custom PII categories that are not covered by built-in profiling or standard checks, a data engineer can define rules that validate data against organization-specific patterns, such as regular expressions, value constraints, or other rule expressions. This keeps the detection process inside the existing DataBrew workflow instead of introducing a separate scanning application or manual review process.

Using DataBrew rulesets also minimizes operational overhead because the rules can be centrally created, reused, and run as part of scheduled profiling jobs against multiple datasets in the data lake. The results are produced through DataBrew's managed profiling and data quality reporting capabilities, which aligns well with a quarterly assessment process. AWS documents DataBrew rulesets and data quality rules as a mechanism for validating datasets during profiling, and DataBrew profiling includes data statistics and quality checks that can be automated. See AWS Glue DataBrew data quality rules and AWS Glue DataBrew profile jobs.

QUESTION NO: 4

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.

ANSWER: A C

Explanation:

Configure the third-party application to create the files in a columnar format is correct because Amazon Redshift Spectrum performs best when external data is stored in columnar formats such as Apache Parquet or ORC. The workload selects only subsets of more than 100 columns, so a columnar layout lets Spectrum read only the required columns instead of scanning full CSV rows. This reduces data scanned, improves query latency, and can lower cost for Spectrum queries. AWS specifically recommends using columnar, splittable, compressed file formats for Redshift Spectrum performance. See the AWS guidance for data files for Redshift Spectrum.

Partition the order data in the S3 bucket based on order date is also correct because users aggregate metrics based on daily orders. Partitioning the external table by order date allows Redshift Spectrum to prune irrelevant S3 prefixes and scan only the date partitions needed by a query. This is a low-effort structural change that aligns directly with the query access pattern and avoids unnecessary reads across all historical order files. AWS recommends partitioning external data by columns commonly used to restrict queries, as described in Amazon Redshift Spectrum external tables.

QUESTION NO: 5

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.

ANSWER: C D

Explanation:

The factors “QuickSiqht does not have access to the S3 bucket.” and “QuickSight does not have access to decrypt S3 data.” are correct because a QuickSight analysis or dashboard that queries Athena still depends on QuickSight being authorized to use the underlying AWS resources. For Athena data sources, QuickSight must have access to Athena and to the S3 locations that Athena reads from and writes query results to. In the QuickSight security and permissions settings, administrators explicitly grant access to selected S3 buckets; if the bucket containing the queried data is not allowed, Athena-backed visuals can fail with an insufficient permissions message. If the S3 objects are encrypted with AWS KMS, QuickSight also needs permission to use the relevant KMS key to decrypt the data. Without KMS decrypt permissions, the service cannot read the encrypted objects even when the bucket itself is reachable, so the dashboard cannot load the Athena query results successfully. AWS documents these requirements in the QuickSight guidance for connecting to Amazon Athena data sources and for granting QuickSight access to AWS resources.

QUESTION NO: 6

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.

ANSWER: B

Explanation:

Use AWS Certificate Manager (ACM) is the correct solution because ACM is designed to reduce the operational burden of managing SSL/TLS certificates in AWS. ACM can provision certificates, securely store the private keys, deploy certificates to supported AWS services, and automatically manage renewals when the certificate remains in use and validation requirements are met. For applications running on Amazon EC2, ACM is commonly used with an integrated front-end service such as Elastic Load Balancing, which terminates TLS for the EC2-hosted application while ACM handles certificate lifecycle tasks. This approach avoids building and maintaining custom certificate generation, distribution, installation, and renewal workflows.

ACM supports both public certificates and private certificates through integration with AWS Private Certificate Authority, making it suitable for securing external-facing endpoints as well as internal AWS communications that require trusted private certificates. Because renewal and deployment are handled by ACM and its integrated AWS services, the data engineer can meet the encryption-in-transit requirement with the least day-to-day operational effort. For details, see the AWS Certificate Manager User Guide and the list of AWS services integrated with ACM.

QUESTION NO: 7

A company uses AWS Key Management Service (AWS KMS) to encrypt an Amazon Redshift cluster. The company wants to configure a cross-Region snapshot of the Redshift cluster as part of disaster recovery (DR) strategy.

A data engineer needs to use the AWS CLI to create the cross-Region snapshot.

Which combination of steps will meet these requirements? (Select TWO.)

A. Create a KMS key and configure a snapshot copy grant in the source AWS Region.

B. In the source AWS Region, enable snapshot copying. Specify the name of the snapshot copy grant that is created in the destination AWS Region.

C. In the source AWS Region, enable snapshot copying. Specify the name of the snapshot copy grant that is created in the source AWS Region.

D. Create a KMS key and configure a snapshot copy grant in the destination AWS Region.

E. Convert the cluster to a Multi-AZ deployment.

ANSWER: B D

Explanation:

For an AWS KMS-encrypted Amazon Redshift cluster, cross-Region snapshot copy requires encryption permissions to be prepared in the destination Region and then referenced from the source Region when snapshot copying is enabled. “Create a KMS key and configure a snapshot copy grant in the destination AWS Region.” is correct because Amazon Redshift needs a KMS key in the target Region to encrypt the copied snapshot, and the snapshot copy grant authorizes Redshift to use that key for the copy operation. “In the source AWS Region, enable snapshot copying. Specify the name of the snapshot copy grant that is created in the destination AWS Region.” is also correct because the snapshot copy setting is enabled on the source cluster, but it must point to the grant that exists in the destination Region. With the AWS CLI, this corresponds to creating the snapshot copy grant in the destination Region and then running the enable snapshot copy operation for the source cluster with the destination Region and snapshot copy grant name. This workflow is described in the Amazon Redshift documentation for copying AWS KMS-encrypted snapshots across Regions and in the AWS CLI command reference for enable-snapshot-copy. See also the Redshift guidance for snapshot copy grants.

QUESTION NO: 8

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.
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]

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.

ANSWER: A

Explanation:

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. is the correct solution because it uses serverless AWS analytics services to provide SQL-style access without building and operating a custom ingestion or transformation pipeline. AWS Glue crawlers can infer schemas from data sources such as Amazon S3 objects and populate the AWS Glue Data Catalog, which Athena uses as a central metadata store. Athena then lets users run interactive SQL queries directly against cataloged data in Amazon S3. For the non-S3 sources in the scenario, Athena can also use federated query data source connectors to query sources such as relational databases, DynamoDB, and Amazon Redshift while presenting a SQL-based query experience. This aligns well with the requirement to let data scientists query across multiple data stores while minimizing infrastructure management. Athena is serverless, so there are no clusters to provision or maintain, and AWS Glue crawlers reduce the manual effort needed to define and maintain schemas. For semi-structured JSON data, Athena supports SQL-based querying of nested and semi-structured structures, including PartiQL-compatible access patterns. See Amazon Athena documentation and Athena federated query documentation.

QUESTION NO: 9

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

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

Which solution will meet these requirements with the LOWEST latency?

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

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

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

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

ANSWER: A

Explanation:

Use Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) to process the sensor data, write the results to Amazon Timestream, and visualize the Timestream data in Grafana is the correct lowest-latency solution. Amazon Managed Service for Apache Flink is designed for continuously processing streaming data from sources such as Amazon Kinesis Data Streams, so the manufacturing metrics can be computed as events arrive rather than after objects land in storage. This is well suited for real-time operational dashboards where sensor data must be transformed, aggregated, or enriched with minimal delay. Amazon Timestream is a serverless time-series database built for high-volume timestamped data such as IoT and industrial sensor readings, and it supports fast ingestion and SQL-based analytics over recent and historical measurements. Grafana integrates with Amazon Timestream as a data source, making it a strong fit for live operational dashboards on a facility display with frequent refresh intervals. Together, these services provide a streaming analytics path from Kinesis Data Streams to a time-series store and then to a real-time visualization layer. References: Amazon Managed Service for Apache Flink documentation and Using Grafana with Amazon Timestream.

QUESTION NO: 10

A company uses Amazon S3 to store data and Amazon QuickSight to create visualizations.

The company has an S3 bucket in an AWS account named Hub-Account. The S3 bucket is encrypted by an AWS Key Management Service (AWS KMS) key. The company's QuickSight instance is in a separate account named BI-Account

The company updates the S3 bucket policy to grant access to the QuickSight service role. The company wants to enable cross-account access to allow QuickSight to interact with the S3 bucket. Which combination of steps will meet this requirement? (Select TWO.)

A. Use the existing AWS KMS key to encrypt connections from QuickSight to the S3 bucket.

B. Add the 53 bucket as a resource that the QuickSight service role can access.

C. Use AWS Resource Access Manager (AWS RAM) to share the S3 bucket with the BI-Account account.

D. Add an IAM policy to the QuickSight service role to give QuickSight access to the KMS key that encrypts the S3 bucket.

E. Add the KMS key as a resource that the QuickSight service role can access.

ANSWER: B E

Explanation:

To allow Amazon QuickSight in BI-Account to use an Amazon S3 bucket in Hub-Account, the bucket must be explicitly configured as a resource that QuickSight is allowed to access from the QuickSight account. Updating the S3 bucket policy in Hub-Account grants the cross-account resource-side permission, but QuickSight still needs the bucket added to its permitted S3 resources so that its service role can make the required S3 API calls. Because the bucket is encrypted with an AWS KMS key, QuickSight also needs access to the KMS key used to encrypt the objects. Adding the KMS key as a resource that the QuickSight service role can access allows QuickSight to decrypt the S3 objects when it reads the data for analysis or ingestion into SPICE. This combination aligns with Amazon QuickSight's documented process for authorizing S3 buckets, including cross-account buckets, and configuring access to encrypted data sources. See the AWS documentation for troubleshooting and configuring QuickSight access to Amazon S3 and using AWS KMS keys with Amazon QuickSight.

QUESTION NO: 11

A company has as JSON file that contains personally identifiable information (PIT) data and non-PII data. The company needs to make the data available for querying and analysis. The non-PII data must be available to everyone in the company. The PII data must be available only to a limited group of employees. Which solution will meet these requirements with the LEAST operational overhead?

A. Store the JSON file in an Amazon S3 bucket. Configure AWS Glue to split the file into one file that contains the PII data and one file that contains the non-PII data. Store the output files in separate S3 buckets. Grant the required access to the buckets based on the type of user.

B. Store the JSON file in an Amazon S3 bucket. Use Amazon Macie to identify PII data and to grant access based on the type of user.

C. Store the JSON file in an Amazon S3 bucket. Catalog the file schema in AWS Lake Formation. Use Lake Formation permissions to provide access to the required data based on the type of user.

D. Create two Amazon RDS PostgreSQL databases. Load the PII data and the non-PII data into the separate databases. Grant access to the databases based on the type of user.

ANSWER: C

Explanation:

Store the JSON file in an Amazon S3 bucket. Catalog the file schema in AWS Lake Formation. Use Lake Formation permissions to provide access to the required data based on the type of user is correct because AWS Lake Formation is designed to centrally manage secure access to data lakes on Amazon S3 with fine-grained authorization. After the JSON data is cataloged in the AWS Glue Data Catalog and governed by Lake Formation, administrators can grant permissions at the database, table, column, row, or cell level. This allows the company to expose non-PII fields broadly while restricting PII fields to only the authorized employee group, without creating and maintaining separate storage locations, duplicating

datasets, or building custom access-control logic. Lake Formation permissions also integrate with supported analytics services such as Amazon Athena, AWS Glue, Amazon Redshift Spectrum, and Amazon EMR, so users can query the same governed dataset while Lake Formation enforces the appropriate access controls. This provides the required separation of sensitive and non-sensitive data with the least operational overhead. See the AWS documentation for Lake Formation permissions and Lake Formation data filtering.

QUESTION NO: 12

A company builds a new data pipeline to process data for business intelligence reports. Users have noticed that data is missing from the reports.

A data engineer needs to add a data quality check for columns that contain null values and for referential integrity at a stage before the data is added to storage.

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

A. Use Amazon SageMaker Data Wrangler to create a Data Quality and Insights report.

B. Use AWS Glue ETL jobs to perform a data quality evaluation transform on the data. Use an IsComplete rule on the requested columns. Use a ReferentialIntegrity rule for each join.

C. Use AWS Glue ETL jobs to perform a SQL transform on the data to determine whether requested columns contain null values. Use a second SQL transform to check referential integrity.

D. Use Amazon SageMaker Data Wrangler and a custom Python transform to create custom rules to check for null values and referential integrity.

ANSWER: B

Explanation:

Use AWS Glue ETL jobs to perform a data quality evaluation transform on the data. Use an IsComplete rule on the requested columns. Use a ReferentialIntegrity rule for each join is correct because AWS Glue Data Quality is designed to validate data directly inside AWS Glue ETL pipelines before the data is written to the target storage layer. The built-in data quality evaluation transform can apply Data Quality Definition Language rules without requiring custom validation code or separate data profiling workflows. The IsComplete rule specifically checks whether a column has complete values, which addresses null-value detection for required columns. The ReferentialIntegrity rule validates whether values in one dataset are present in a referenced dataset, which addresses relationship consistency before the pipeline commits data for reporting use. This approach keeps the checks close to the ETL process, supports automated pass/fail outcomes, and minimizes operational overhead by using managed AWS Glue capabilities rather than manually implementing SQL or Python-based validation logic. AWS documents these capabilities in the AWS Glue Data Quality rule types and ETL integration guidance: AWS Glue Data Quality rule types and AWS Glue Data Quality in ETL jobs.

QUESTION NO: 13

A company stores customer records in Amazon S3. The company must not delete or modify the customer record data for 7 years after each record is created. The root user also must not have the ability to delete or modify the data.

A data engineer wants to use S3 Object Lock to secure the data.

Which solution will meet these requirements?

A. Enable governance mode on the S3 bucket. Use a default retention period of 7 years.

B. Enable compliance mode on the S3 bucket. Use a default retention period of 7 years.

C. Place a legal hold on individual objects in the S3 bucket. Set the retention period to 7 years.

D. Set the retention period for individual objects in the S3 bucket to 7 years.

ANSWER: B

Explanation:

Enable compliance mode on the S3 bucket. Use a default retention period of 7 years is correct because Amazon S3 Object Lock in compliance mode enforces write-once-read-many protection for the full retention period. While an object version is protected by compliance mode, it cannot be overwritten or deleted by any user, including the AWS account root user. This directly satisfies the requirement that customer records must remain unmodified and undeleted for 7 years after creation and that even the root user must not be able to bypass the protection. Configuring a default retention period on the bucket ensures that new objects uploaded to the bucket automatically receive the required 7-year retention setting, which is appropriate when every customer record must be protected consistently from the time it is created. AWS documentation states that compliance mode prevents deletion or overwrite of protected object versions by any user until the retention date passes. For more details, see Amazon S3 Object Lock and Object Lock retention modes.

QUESTION NO: 14

A company is designing a serverless data processing workflow in AWS Step Functions that involves multiple steps. The processing workflow ingests data from an external API, transforms the data by using multiple AWS Lambda functions, and loads the transformed data into Amazon DynamoDB.

The company needs the workflow to perform specific steps based on the content of the incoming data.

Which Step Functions state type should the company use to meet this requirement?

A. Parallel

B. Choice

C. Task

D. Map

ANSWER: B

Explanation:

Choice is correct because an AWS Step Functions Choice state is designed specifically for conditional branching in a workflow. When a state machine needs to inspect incoming data and decide which step or path to run next, the Choice state evaluates rules against fields in the state input. Based on those rule evaluations, Step Functions transitions to the matching next state. This fits the requirement to perform specific steps based on the content of the incoming data, such as routing records to different Lambda transformation functions or skipping certain processing paths depending on attributes returned from an external API.

In Amazon States Language, a Choice state contains one or more choice rules and can also define a default transition if no rule matches. This makes it the standard way to implement if/else-style logic in Step Functions while keeping the workflow serverless and declarative. AWS documents Choice states as the mechanism for adding branching logic to state machines, including comparisons against JSON input values and JSONata or JSONPath-based conditions. For more details, see the AWS Step Functions documentation on Choice workflow states and the Amazon States Language.

QUESTION NO: 15

A data engineer needs to use Amazon Neptune to develop graph applications.

Which programming languages should the engineer use to develop the graph applications? (Select TWO.)

A. Gremlin

B. SQL

C. ANSI SQL

D. SPARQL

E. Spark SQL

ANSWER: A D

Explanation:

Gremlin and SPARQL are correct because Amazon Neptune is a fully managed graph database service that supports multiple graph query languages for building graph applications. Gremlin is the traversal language of Apache TinkerPop and is used with Neptune's property graph model. It lets developers express graph traversals such as finding connected vertices, walking relationships, filtering paths, and updating graph structures. This makes Gremlin a natural choice for applications that work with highly connected property graph data, such as recommendation engines, fraud detection, and network analysis.

SPARQL is also correct because Neptune supports RDF graphs and provides a SPARQL endpoint for querying RDF data. SPARQL is the standard query language for RDF and is commonly used when graph data is represented as triples and queried using semantic relationships. Neptune's support for both Gremlin and SPARQL allows engineers to choose the graph model and query approach that best fits the application design. AWS documents Gremlin support in the Amazon Neptune Gremlin documentation and SPARQL support in the Amazon Neptune SPARQL documentation.

QUESTION NO: 16

A company uses Amazon RDS to store transactional data. The company runs an RDS DB instance in a private subnet. A developer wrote an AWS Lambda function with default settings to insert, update,

or delete data in the DB instance.

The developer needs to give the Lambda function the ability to connect to the DB instance privately without using the public internet.

Which combination of steps will meet this requirement with the LEAST operational overhead? (Choose two.)

A. Turn on the public access setting for the DB instance.

B. Update the security group of the DB instance to allow only Lambda function invocations on the database port.

C. Configure the Lambda function to run in the same subnet that the DB instance uses.

D. Attach the same security group to the Lambda function and the DB instance. Include a selfreferencing rule that allows access through the database port.

E. Update the network ACL of the private subnet to include a self-referencing rule that allows access through the database port.

ANSWER: C D

Explanation:

Configure the Lambda function to run in the same subnet that the DB instance uses is correct because a Lambda function with default settings runs outside the customer VPC and cannot directly reach private resources such as an Amazon RDS DB instance in a private subnet. By configuring the function with VPC access and selecting the private subnet, Lambda creates managed elastic network interfaces in that VPC so the function can communicate with private IP addresses inside the VPC without traversing the public internet. Attach the same security group to the Lambda function and the DB instance. Include a selfreferencing rule that allows access through the database port is also correct because security groups control allowed traffic between the Lambda-managed network interface and the RDS instance. A self-referencing inbound rule on the shared security group allows resources associated with that security group to communicate with each other on the database port, while keeping access private and tightly scoped. This combination uses native VPC networking and security group controls, which keeps operational overhead low and avoids introducing public exposure or additional networking components. For details, see Configuring a Lambda function to access resources in a VPC and Controlling access with security groups.

QUESTION NO: 17

A retail company stores order information in an Amazon Aurora table named Orders. The company needs to create operational reports from the Orders table with minimal latency. The Orders table contains billions of rows, and over 100,000 transactions can occur each second.

A marketing team needs to join the Orders data with an Amazon Redshift table named Campaigns in the marketing team's data warehouse. The operational Aurora database must not be affected.

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

A. Use AW5 Database Migration Service (AWS DMS) Serverless to replicate the Orders table to Amazon Redshift. Create a materialized view in Amazon Redshift to join with the Campaigns table.

B. Use the Aurora zero-ETL integration with Amazon Redshift to replicate the Orders table. Create a materialized view in Amazon Redshift to join with the Campaigns table.

C. Use AWS Glue to replicate the Orders table to Amazon Redshift. Create a materialized view in Amazon Redshift to join with the Campaigns table.

D. Use federated queries to query the Orders table directly from Aurora. Create a materialized view in Amazon Redshift to join with the Campaigns table.

ANSWER: B

Explanation:

Use the Aurora zero-ETL integration with Amazon Redshift to replicate the Orders table. Create a materialized view in Amazon Redshift to join with the Campaigns table. is correct because Aurora zero-ETL integration is purpose-built to make transactional data from Amazon Aurora available in Amazon Redshift with near real-time replication and without requiring a custom data pipeline. This fits the need for minimal latency reporting on a very high-volume Orders table while avoiding analytical query pressure on the operational Aurora database. After the data is replicated into Amazon Redshift, the marketing team can join the replicated Orders data with the existing Campaigns table inside the Redshift data warehouse, where analytics workloads are designed to run at scale. Creating a materialized view in Redshift can further optimize repeated reporting queries and joins, improving performance for operational reporting use cases. This approach also provides the least operational effort because AWS manages the integration mechanics, reducing the need to build, schedule, monitor, and troubleshoot separate extract-and-load jobs. AWS describes Aurora zero-ETL integrations as enabling near real-time analytics in Amazon Redshift from Aurora data; see Aurora zero-ETL integrations and Using zero-ETL integrations with Amazon Redshift.

QUESTION NO: 18

A company uses AWS Glue Apache Spark jobs to handle extract, transform, and load (ETL) workloads. The company has enabled logging and monitoring for all AWS Glue jobs. One of the AWS Glue jobs begins to fail. A data engineer investigates the error and wants to examine metrics for all individual stages within the job. How can the data engineer access the stage metrics?

A. Examine the AWS Glue job and stage details in the Spark UI.

B. Examine the AWS Glue job and stage metrics in Amazon CloudWatch.

C. Examine the AWS Glue job and stage logs in AWS CloudTrail logs.

D. Examine the AWS Glue job and stage details by using the run insights feature on the job.

ANSWER: A

Explanation:

Examine the AWS Glue job and stage details in the Spark UI is correct because AWS Glue Apache Spark jobs produce Spark event data that can be viewed through the Apache Spark web UI. The Spark UI is designed specifically for Spark-level troubleshooting and performance analysis, including details about Spark applications, jobs, stages, tasks, executors, storage, and SQL execution. When a Glue job fails or performs unexpectedly, the Spark UI lets a data engineer drill into

each stage to inspect metrics such as task duration, input and output records, shuffle read/write behavior, failed tasks, executor activity, and related diagnostic details. This is the appropriate tool when the requirement is to examine metrics for all individual stages within a Spark-based AWS Glue job rather than only high-level job run status. AWS Glue supports using the Spark UI by generating Spark event logs and making them available for review, including through Glue job run monitoring workflows. For more details, see the AWS documentation for monitoring jobs using the Apache Spark web UI and the Spark documentation for the Spark web UI.

QUESTION NO: 19

A company is planning to use a provisioned Amazon EMR cluster that runs Apache Spark jobs to perform big data analysis. The company requires high reliability. A big data team must follow best practices for running cost-optimized and long-running workloads on Amazon EMR. The team must

find a solution that will maintain the company's current level of performance.

Which combination of resources will meet these requirements MOST cost-effectively? (Choose two.)

A. Use Hadoop Distributed File System (HDFS) as a persistent data store.

B. Use Amazon S3 as a persistent data store.

C. Use x86-based instances for core nodes and task nodes.

D. Use Graviton instances for core nodes and task nodes.

E. Use Spot Instances for all primary nodes.

ANSWER: B D

Explanation:

Use Amazon S3 as a persistent data store is correct because Amazon EMR best practices favor separating durable storage from transient cluster compute. Storing input, output, and intermediate durable datasets in Amazon S3 provides high durability and availability while allowing the EMR cluster to scale, terminate, or be replaced without risking persistent data loss. Amazon EMR integrates with Amazon S3 through EMRFS and supports Spark workloads that read from and write to S3, which is a common pattern for reliable and cost-optimized analytics architectures. See the AWS documentation for EMR file systems.

Use Graviton instances for core nodes and task nodes is also correct because AWS Graviton-based EC2 instances are designed to provide strong price performance for scale-out workloads, including Apache Spark on Amazon EMR. For long-running provisioned clusters, improved price performance directly reduces compute cost while maintaining the required processing capability. AWS documents Graviton support and price-performance benefits for Amazon EMR, making it an appropriate choice for cost-optimized big data processing when the application stack is compatible. See Amazon EMR instance planning guidelines.

QUESTION NO: 20

A company generates reports from 30 tables in an Amazon Redshift data warehouse. The data source is an operational Amazon Aurora MySQL database that contains 100 tables. Currently, the company refreshes all data from Aurora to Redshift every hour, which causes delays in report generation.

Which combination of steps will meet these requirements with the LEAST operational overhead? (Select TWO.)

A. Use AWS Database Migration Service (AWS DMS) to create a replication task. Select only the required tables.

B. Create a database in Amazon Redshift that uses the integration.

C. Create a zero-ETL integration in Amazon Aurora. Select only the required tables.

D. Use query editor v2 in Amazon Redshift to access the data in Aurora.

E. Create an AWS Glue job to transfer each required table. Run an AWS Glue workflow to initiate the jobs every 5 minutes.

ANSWER: B C

Explanation:

The best combination is to create a zero-ETL integration in Amazon Aurora and select only the required tables, then create a database in Amazon Redshift that uses the integration. Aurora zero-ETL integrations with Amazon Redshift are designed for near real-time analytics without requiring custom extract, transform, and load pipelines. This directly addresses the reporting delays caused by hourly full refreshes because changes from the Aurora MySQL source can be replicated continuously into Redshift with much less operational management. Selecting only the required tables also reduces unnecessary data movement because the reports depend on only 30 of the 100 source tables. After the integration is created, Amazon Redshift needs a target database that is associated with the integration so analysts and reporting workloads can query the replicated data in Redshift. This approach is aligned with AWS guidance for simplifying analytics on operational data while minimizing pipeline maintenance. For more details, see the AWS documentation for Aurora zero-ETL integrations with Amazon Redshift and creating a destination database in Amazon Redshift.

QUESTION NO: 21

An ecommerce company processes millions of orders each day. The company uses AWS Glue ETL to collect data from multiple sources, clean the data, and store the data in an Amazon S3 bucket in CSV format by using the S3 Standard storage class. The company uses the stored data to conduct daily analysis.

The company wants to optimize costs for data storage and retrieval. Which solution will meet this requirement?

A. Transition the data to Amazon S3 Glacier Flexible Retrieval.

B. Transition the data from Amazon S3 to an Amazon Aurora cluster.

C. Configure AWS Glue ETL to transform the incoming data to Apache Parquet format.

D. Configure AWS Glue ETL to use Amazon EMR to process incoming data in parallel.

ANSWER: C

Explanation:

Configure AWS Glue ETL to transform the incoming data to Apache Parquet format is the correct solution because Parquet is a columnar, compressed file format that is optimized for analytics workloads on Amazon S3. Since the company performs daily analysis on millions of orders, reducing the amount of data stored and scanned directly lowers cost and improves retrieval/query performance. CSV is row-based and typically requires analytics engines to read more data than necessary, even when only a subset of columns is needed. Parquet stores data by column and supports efficient compression and predicate pushdown, so services such as Amazon Athena, Amazon EMR, Amazon Redshift Spectrum, and AWS Glue can read less data for many analytical queries.

This approach keeps the data in Amazon S3 for scalable, durable storage while making the dataset more efficient for repeated analysis. AWS specifically recommends using columnar formats such as Apache Parquet for better performance and lower query costs with analytics services. See the AWS Athena guidance on using columnar storage formats and the AWS Glue documentation for Parquet format support.

QUESTION NO: 22

A company is planning to upgrade its Amazon Elastic Block Store (Amazon EBS) General Purpose SSD storage from gp2 to gp3. The company wants to prevent any interruptions in its Amazon EC2 instances that will cause data loss during the migration to the upgraded storage.

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

A. Create snapshots of the gp2 volumes. Create new gp3 volumes from the snapshots. Attach the new gp3 volumes to the EC2 instances.

B. Create new gp3 volumes. Gradually transfer the data to the new gp3 volumes. When the transfer is complete, mount the new gp3 volumes to the EC2 instances to replace the gp2 volumes.

C. Change the volume type of the existing gp2 volumes to gp3. Enter new values for volume size, IOPS, and throughput.

D. Use AWS DataSync to create new gp3 volumes. Transfer the data from the original gp2 volumes to the new gp3 volumes.

ANSWER: C

Explanation:

Change the volume type of the existing gp2 volumes to gp3. Enter new values for volume size, IOPS, and throughput. is the correct solution because Amazon EBS Elastic Volumes supports modifying an existing EBS volume in place, including changing the volume type from gp2 to gp3 and configuring gp3 performance settings. This avoids creating replacement volumes, copying data, detaching disks, or remounting storage, which makes it the lowest-operational-overhead approach. In most cases, the volume remains attached and available while the modification is applied, so the EC2 instance can continue running and applications can continue accessing the data during the transition. gp3 also separates storage size from performance, so the company can explicitly set IOPS and throughput values that meet workload requirements after the conversion. AWS documents that you can use Elastic Volumes to modify volume type, size, IOPS, and throughput without detaching the volume, and the gp3 documentation describes the configurable baseline and provisioned performance characteristics. See Modify an Amazon EBS volume using Elastic Volumes and General Purpose SSD volumes.

QUESTION NO: 23

A banking company uses an application to collect large volumes of transactional data. The company uses Amazon Kinesis Data Streams for real-time analytics. The company's application uses the PutRecord action to send data to Kinesis Data Streams.

A data engineer has observed network outages during certain times of day. The data engineer wants to configure exactly-once delivery for the entire processing pipeline.

Which solution will meet this requirement?

A. Design the application so it can remove duplicates during processing by embedding a unique ID in each record at the source.

B. Update the checkpoint configuration of the Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics) data collection application to avoid duplicate processing of events.

C.

Design the data source so events are not ingested into Kinesis Data Streams multiple times.

D. Stop using Kinesis Data Streams. Use Amazon EMR instead. Use Apache Flink and Apache Spark Streaming in Amazon EMR.

ANSWER: A

Explanation:

Design the application so it can remove duplicates during processing by embedding a unique ID in each record at the source is the correct solution. Amazon Kinesis Data Streams can support highly durable ingestion and ordered processing within a shard, but producers that use PutRecord can still create duplicate records when retries occur after network timeouts or ambiguous failures. For example, the producer might successfully write a record to the stream but fail to receive the acknowledgement because of a network outage; a retry can then write the same business event again. Because of this distributed-systems behavior, exactly-once delivery across the full pipeline is achieved by making the application idempotent.

Adding a stable, source-generated unique identifier to each transactional event lets downstream consumers detect whether an event has already been processed and safely ignore duplicates. This is the recommended design pattern for building

reliable streaming applications where retries, failures, and replays are possible. AWS explicitly recommends handling duplicate records in consumers and using application-level logic to make processing idempotent when using Kinesis Data Streams. See the AWS documentation on handling duplicate records in Kinesis Data Streams consumers and the PutRecord API behavior.

QUESTION NO: 24

A company stores employee data in Amazon Redshift A table named Employee uses columns named Region ID, Department ID, and Role ID as a compound sort key. Which queries will MOST increase the speed of a query by using a compound sort key of the table? (Select TWO.)

A. Select * from Employee where Region ID='North America';

B. Select * from Employee where Region ID='North America' and Department ID=20;

C. Select * from Employee where Department ID=20 and Region ID='North America';

D. Select '' from Employee where Role ID=50;

E. Select * from Employee where Region ID='North America' and Role ID=50;

ANSWER: B C

Explanation:

The correct queries are **Select * from Employee where Region ID='North America' and Department ID=20;** and **Select * from Employee where Department ID=20 and Region ID='North America';**. In Amazon Redshift, a compound sort key stores data in sorted order based on the leading sort key column first, then the next column, and so on. Because the table is sorted by Region ID, then Department ID, then Role ID, queries that filter on the leading prefix of that sort key can take the best advantage of block pruning through Redshift zone maps. Filtering on both Region ID and Department ID matches the first two columns of the compound sort key, so Redshift can skip larger portions of table data and scan fewer blocks.

The order of predicates in the SQL WHERE clause does not need to match the sort key definition. Redshift's optimizer can evaluate the predicates logically and still use the compound sort key effectively when the leading sort key columns are included. This is why both query forms provide the same sort key benefit even though the conditions appear in a different textual order. For more details, see the AWS documentation on sorting data in Amazon Redshift and sort key best practices.

QUESTION NO: 25

A company has an application that uses a microservice architecture. The company hosts the application on an Amazon Elastic Kubernetes Services (Amazon EKS) cluster.

The company wants to set up a robust monitoring system for the application. The company needs to analyze the logs from the EKS cluster and the application. The company needs to correlate the cluster's logs with the application's traces to identify points of failure in the whole application request flow.

Which combination of steps will meet these requirements with the LEAST development effort? (Select TWO.)

A. Use FluentBit to collect logs. Use OpenTelemetry to collect traces.

B. Use Amazon CloudWatch to collect logs. Use Amazon Kinesis to collect traces.

C. Use Amazon CloudWatch to collect logs. Use Amazon Managed Streaming for Apache Kafka (Amazon MSK) to collect traces.

D. Use Amazon OpenSearch to correlate the logs and traces.

E. Use AWS Glue to correlate the logs and traces.

ANSWER: A D

Explanation:

Use FluentBit to collect logs. Use OpenTelemetry to collect traces. is correct because Fluent Bit is the standard lightweight log router commonly used with Amazon EKS to collect Kubernetes node, pod, and container logs and forward them to observability backends. OpenTelemetry is the AWS-supported, vendor-neutral approach for collecting distributed traces from microservice applications, and AWS provides the AWS Distro for OpenTelemetry to reduce setup and operational effort for instrumenting and exporting telemetry from workloads running on EKS. Together, these tools provide log and trace collection without requiring a custom ingestion framework. See Amazon EKS logging with Fluent Bit and AWS Distro for OpenTelemetry.

Use Amazon OpenSearch to correlate the logs and traces. is also correct because Amazon OpenSearch Service provides observability capabilities for searching, analyzing, visualizing, and correlating operational data such as logs and traces. OpenSearch supports trace analytics and log analytics workflows, making it suitable for following a request across microservices and identifying where failures occur in the end-to-end request path. This managed service approach minimizes development effort by using built-in observability features instead of building a custom correlation and visualization layer.

QUESTION NO: 26

A company has a data processing pipeline that runs multiple SQL queries in sequence against an Amazon Redshift cluster. The company merges with a second company. The original company modifies a query that aggregates sales revenue data to join sales tables from both companies.

The sales table for the first company is named Table S1 and contains 10 billion records. The sales table for the second company is named Table S2 and contains 900 million records. The query becomes slow after the modification.

A data engineer must improve the query performance.

Which solutions will meet these requirements? (Select TWO)

A. Use the KEY distribution style for both sales tables. Select a low-cardinality column to use for the join.

B. Use the KEY distribution style for both sales tables. Select a high-cardinality column to use for the join.

C. Use the EVEN distribution style for Table S1. Use the ALL distribution style for Table S2.

D. Use the Amazon Redshift query optimizer to review and select optimizations to implement.

E. Use Amazon Redshift Advisor to review and select optimizations to implement.

ANSWER: B E

Explanation:

Using the KEY distribution style for both sales tables and selecting a high-cardinality column to use for the join is correct because Amazon Redshift performs best when rows that are joined together are physically colocated on the same compute nodes. For very large tables such as these sales tables, choosing the join column as the distribution key can reduce or eliminate expensive data redistribution during query execution. A high-cardinality distribution key also helps spread rows more evenly across slices, which reduces data skew and improves parallel processing efficiency. AWS guidance recommends choosing distribution keys based on common join columns and evaluating whether the key distributes data evenly across the cluster. See the Amazon Redshift documentation on choosing the best distribution key.

Using Amazon Redshift Advisor to review and select optimizations to implement is also correct because Advisor analyzes the cluster workload and provides recommendations based on observed query patterns and table design. These recommendations can include changes to distribution keys, sort keys, compression, and other physical design choices that improve performance. This is especially useful after a major workload change, such as adding a large new table to an existing join query. For more information, see Amazon Redshift Advisor.

QUESTION NO: 27

A company stores sales data in an Amazon RDS for MySQL database. The company needs to start a reporting process between 6:00 A.M. and 6:10 A.M. every Monday. The reporting process must generate a CSV file and store the file in an Amazon S3 bucket.

Which combination of steps will meet these requirements with the LEAST operational overhead? (Select TWO.)

A. Create an Amazon EventBridge rule to run every Monday at 6:00 A.M.

B. Create an Amazon EventBridge Scheduler to run every Monday at 6:00 A.M.

C. Create and invoke an AWS Batch job that runs a script in an Amazon Elastic Container Service (Amazon ECS) container. Configure the script to generate the report and to save it to the S3 bucket.

D. Create and invoke an AWS Glue ETL job to generate the report and to save it to the S3 bucket.

E. Create and invoke an Amazon EMR Serverless job to generate the report and to save it to the S3 bucket.

ANSWER: B D

Explanation:

Create an Amazon EventBridge Scheduler to run every Monday at 6:00 A.M. is correct because EventBridge Scheduler is a fully managed scheduling service that can invoke AWS services on recurring cron-based schedules without requiring servers or custom polling logic. Scheduling the workflow for 6:00 A.M. satisfies the requirement to start the process between 6:00 A.M. and 6:10 A.M. every Monday. Create and invoke an AWS Glue ETL job to generate the report and to save it to the S3 bucket is also correct because AWS Glue is a managed serverless data integration service that can connect to Amazon RDS for MySQL through JDBC, run the required extraction and transformation logic, and write the resulting CSV output to Amazon S3. This combination minimizes operational overhead because AWS manages the scheduling infrastructure and the ETL execution environment, so the company does not need to provision or maintain compute clusters, containers, or orchestration servers. For more details, see the AWS documentation for Amazon EventBridge Scheduler and AWS Glue jobs.

QUESTION NO: 28

A company uses an Amazon QuickSight dashboard to monitor usage of one of the company's applications. The company uses AWS Glue jobs to process data for the dashboard. The company stores the data in a single Amazon S3 bucket. The company adds new data every day.

A data engineer discovers that dashboard queries are becoming slower over time. The data engineer determines that the root cause of the slowing queries is long-running AWS Glue jobs.

Which actions should the data engineer take to improve the performance of the AWS Glue jobs? (Choose two.)

A. Partition the data that is in the S3 bucket. Organize the data by year, month, and day.

B. Increase the AWS Glue instance size by scaling up the worker type.

C. Convert the AWS Glue schema to the DynamicFrame schema class.

D. Adjust AWS Glue job scheduling frequency so the jobs run half as many times each day.

E. Modify the 1AM role that grants access to AWS glue to grant access to all S3 features.

ANSWER: A B

Explanation:

Partition the data that is in the S3 bucket. Organize the data by year, month, and day. is correct because the dataset grows daily, and date-based partitioning lets AWS Glue and downstream query engines read only the relevant slices of data instead of scanning the full bucket contents. When the AWS Glue Data Catalog contains partition metadata, jobs can use partition pruning and pushdown predicates to reduce S3 reads, Spark processing time, and overall job duration. This is a

standard optimization for time-series or incremental datasets stored in Amazon S3. See the AWS Glue documentation on managing partitions for ETL output in AWS Glue.

Increase the AWS Glue instance size by scaling up the worker type. is also correct because long-running AWS Glue jobs can benefit from more compute, memory, and parallelism when the workload is constrained by available resources. AWS Glue worker types provide different amounts of vCPU, memory, disk, and Spark executors, and selecting a larger worker type can improve job throughput for larger datasets or heavier transformations. AWS documents worker types and their resource characteristics in Defining job properties in AWS Glue.

QUESTION NO: 29

A company uses Amazon S3 as a data lake. The company sets up a data warehouse by using a multinode Amazon Redshift cluster. The company organizes the data files in the data lake based on the

data source of each data file.

The company loads all the data files into one table in the Redshift cluster by using a separate COPY command for each data file location. This approach takes a long time to load all the data files into the table. The company must increase the speed of the data ingestion. The company does not want to increase the cost of the process.

Which solution will meet these requirements?

A. Use a provisioned Amazon EMR cluster to copy all the data files into one folder. Use a COPY command to load the data into Amazon Redshift.

B. Load all the data files in parallel into Amazon Aurora. Run an AWS Glue job to load the data into Amazon Redshift.

C. Use an AWS Glue job to copy all the data files into one folder. Use a COPY command to load the data into Amazon Redshift.

D. Create a manifest file that contains the data file locations. Use a COPY command to load the data into Amazon Redshift.

ANSWER: D

Explanation:

Create a manifest file that contains the data file locations. Use a COPY command to load the data into Amazon Redshift. is correct because Amazon Redshift COPY is optimized for parallel loading from Amazon S3, and a manifest file lets a single COPY operation explicitly reference files that are spread across multiple S3 prefixes or locations. In this scenario, the company is currently running separate COPY commands for each data file location, which adds overhead and can reduce overall ingestion efficiency. By listing all required objects in a manifest file, Redshift can process the load as one coordinated operation while still reading data in parallel across the multinode cluster.

This approach directly addresses the performance problem without introducing additional services, intermediate storage, cluster provisioning, or extra ETL processing costs. Redshift manifest files are designed for cases where the files to load are not conveniently represented by one common S3 prefix, or when you need precise control over which files are included in a load. AWS documents that COPY can use a manifest file in JSON format to specify the S3 object URLs to load. For more details, see the Amazon Redshift documentation for COPY from Amazon S3 and using a manifest to specify data files.

QUESTION NO: 30

A data engineer is processing a large amount of log data from web servers. The data is stored in an Amazon S3 bucket. The data engineer uses AWS services to process the data every day. The data engineer needs to extract specific fields from the raw log data and load the data into a data warehouse for analysis.

A. Use Amazon EMR to run Apache Hive queries on the raw log files in the S3 bucket to extract the specified fields. Store the output as ORC files in the original S3 bucket.

B. Use AWS Step Functions to orchestrate a series of AWS Batch jobs to parse the raw log files. Load the specified fields into an Amazon RDS for PostgreSQL database.

C. Use an AWS Glue crawler to parse the raw log data in the S3 bucket and to generate a schema. Use AWS Glue ETL jobs to extract and transform the data and to load it into Amazon Redshift.

D. Use AWS Glue DataBrew to run AWS Glue ETL jobs on a schedule to extract the specified fields from the raw log files in the S3 bucket. Load the data into partitioned tables in Amazon Redshift.

ANSWER: C

Explanation:

Use an AWS Glue crawler to parse the raw log data in the S3 bucket and to generate a schema. Use AWS Glue ETL jobs to extract and transform the data and to load it into Amazon Redshift. is the correct choice because AWS Glue is designed for serverless data integration workflows involving data stored in Amazon S3. A crawler can scan the raw log files, infer their structure, and create or update metadata in the AWS Glue Data Catalog. That cataloged schema can then be used by AWS Glue ETL jobs to read the source data reliably, extract only the required fields, transform the records as needed, and write the curated result to a target analytics store.

Amazon Redshift is an AWS data warehouse service, so loading the transformed log data into Redshift directly satisfies the requirement to support downstream analysis. AWS Glue jobs can be scheduled to run daily, which matches the recurring processing requirement without needing to manage infrastructure. This pattern is a common AWS-native approach for building repeatable extract, transform, and load pipelines from S3 into a warehouse. For more details, see the AWS documentation for AWS Glue crawlers and using AWS Glue with Amazon Redshift.

QUESTION NO: 31

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.

ANSWER: A D

Explanation:

Use AWS Glue DataBrew to perform extract, transform, and load (ETL) tasks that mask the PII data before analysis is correct because DataBrew provides managed, no-code/low-code data preparation capabilities, including recipe actions for detecting, redacting, replacing, hashing, and otherwise masking personally identifiable information. This fits the requirement for a low-maintenance way to transform sensitive customer data before it is used for analytics, without requiring the company to build and operate custom masking logic across the pipeline. See the AWS documentation for DataBrew PII transformations at AWS Glue DataBrew PII recipe actions.

Use AWS Identity and Access Management (IAM) to manage permissions and to control access to the PII data is also correct because IAM policies can grant only approved users and roles access to the original encrypted S3 objects, AWS KMS keys, and preprocessing resources. This supports secure access for authorized preprocessing users while keeping broader analysis users limited to masked datasets. IAM is the standard AWS service for centrally managing permissions to AWS resources, including S3 and KMS. See IAM policies and permissions.

QUESTION NO: 32

A data engineer is using an Apache Iceberg framework to build a data lake that contains 100 TB of data. The data engineer wants to run AWS Glue Apache Spark Jobs that use the Iceberg framework. What combination of steps will meet these requirements? (Select TWO.)

A. Create a key named -conf for an AWS Glue job. Set Iceberg as a value for the --datalake-formats job parameter.

B. Specify the path to a specific version of Iceberg by using the --extra-Jars job parameter. Set Iceberg as a value for the ~ datalake-formats job parameter.

C. Set Iceberg as a value for the -datalake-formats job parameter.

D. Set the -enable-auto-scaling parameter to true.

E. Add the -job-bookmark-option: job-bookmark-enable parameter to an AWS Glue job.

ANSWER: A C

Explanation:

To run Apache Iceberg workloads in AWS Glue Apache Spark jobs, the job must be configured to load AWS Glue's built-in Iceberg integration and Spark must be given the Iceberg-related configuration through the job parameters. "Create a key named -conf for an AWS Glue job. Set Iceberg as a value for the --datalake-formats job parameter." is correct because AWS Glue requires the Iceberg data lake format to be enabled and the Spark session to be configured with Iceberg extensions and catalog settings. "Set Iceberg as a value for the -datalake-formats job parameter." is also correct because the core AWS Glue job parameter for enabling Iceberg support is the data lake formats parameter with Iceberg as the value. In AWS documentation, this is shown as setting `--datalake-formats` to `iceberg`, along with using `--conf` for Spark/Iceberg catalog configuration. This enables AWS Glue Spark jobs to read and write Iceberg tables using supported AWS Glue runtimes and the AWS Glue Data Catalog. See the AWS Glue documentation for using the Iceberg framework in AWS Glue and the AWS Glue job parameters reference.