



CERTSWARRIOR

# Amazon MLA-C01

**AWS Certified Machine Learning Engineer - Associate**

**Questions&AnswersPDF**

**ForMoreInformation:**

**<https://www.certswarrior.com/>**

## **Features:**

- 90DaysFreeUpdates
- 30DaysMoneyBackGuarantee
- InstantDownloadOncePurchased
- 24/7OnlineChat Support
- ItsLatestVersion

# Latest Version: 6.0

## Question: 1

A machine learning engineer is tasked with building a fraud detection system. The system needs to ingest high-velocity streaming data (transactions) from multiple sources. The data arrives in JSON format, but with varying schemas and occasional missing fields. The engineer needs to store this raw, unstructured data for future analysis and model retraining. The raw data must be preserved in its original form. Which AWS service and data format combination is MOST suitable for this scenario, minimizing initial transformation overhead and ensuring scalability?

- A. Amazon S3 with CSV format, using AWS Glue to infer schema and handle missing fields.
- B. Amazon Kinesis Data Firehose with Parquet format, configured to transform data using AWS Lambda before storage in S3.
- C. Amazon Kinesis Data Firehose with JSON format, delivering directly to an S3 bucket with automatic schema discovery via AWS Glue crawlers.
- D. Amazon RDS with JSON datatype columns to store the raw JSON strings. Create a separate table and use AWS Glue to create data catalog.
- E. Amazon DynamoDB with JSON document structure

**Answer: C**

Explanation:

Option C is the most suitable. Kinesis Data Firehose allows direct ingestion of JSON data to S3 without mandatory transformation, which fits the requirement of preserving the raw data. Storing in S3 provides scalability and cost-effectiveness. AWS Glue crawlers can then be used to discover the schema of the JSON data stored in S3 for later analysis using services like Athena or EMR. Other formats (Parquet) require transformation, which contradicts the requirement to preserve the raw data initially. RDS is not ideal for unstructured data storage. DynamoDB could be used, but S3 is more cost effective for raw data archival.

## Question: 2

You are developing a real-time image classification model. Images are streamed from IoT devices to AWS. The system needs to preprocess these images before feeding them into the model. The pre-processing steps involve resizing, normalization, and format conversion. To optimize performance and cost, you need to choose the most efficient data format for storing the pre-processed images in S3. Which data format offers the best balance of storage efficiency, read/write performance, and support for image data?

- A. CSV (Comma Separated Values)
- B. JSON (JavaScript Object Notation)
- C. Parquet
- D. RecordIO

E. Apache Avro

**Answer: D**

Explanation:

RecordIO is optimized for storing sequences of records, especially binary data like images. It's designed for efficient read/write operations in machine learning workloads. While Parquet is good for columnar storage and analytics, it's less suitable for storing raw image data directly. CSV and JSON are not appropriate for image data. Apache Avro is a good option for serializing data structures and supports schema evolution but is not as optimized for binary data compared to RecordIO, especially when used within the SageMaker ecosystem.

### Question: 3

A data engineer is setting up a data pipeline to ingest sensor data from a fleet of connected devices. The data is semi-structured, containing readings along with device metadata, and is received as JSON. The engineer needs to efficiently store and query the data in a cost-effective manner for ad-hoc analysis and model training. Given the need for complex queries involving nested JSON fields and historical data analysis, which of the following combinations of AWS services and data formats would BEST address these requirements? (Choose TWO)

- A. Amazon S3 with JSON format, using Amazon Athena for querying.
- B. Amazon Redshift with VARCHAR columns to store the JSON strings, using JSON functions for querying.
- C. Amazon S3 with Apache Parquet format, converting JSON data to Parquet using AWS Glue.
- D. Amazon DynamoDB with JSON document structure and using DynamoDB Accelerator (DAX).
- E. Amazon S3 with Apache Avro format, converting JSON data to Avro using AWS Glue.

**Answer: C,A**

Explanation:

Option A and C are the best choices: Storing data in S3 with Apache Parquet (Option C) offers several advantages. Parquet is a columnar storage format, which is highly efficient for analytical queries. AWS Glue can be used to transform the JSON data into Parquet format. This improves query performance and reduces storage costs, especially when dealing with large datasets. Amazon S3 with JSON format, using Amazon Athena for querying. S3 stores the raw data, and AWS Athena allows to query the data directly using SQL. No need for transformation or to use AWS Glue to infer the schema since JSON is self describing. This is a cost effective way to create ad-hoc queries.

### Question: 4

Your Machine Learning (ML) application processes high-resolution satellite imagery stored on-premises. The data volume is 50 TB and is growing by 1 TB per week. You need to migrate this data to AWS for model training and real-time inference. The training pipeline requires frequent reads and writes during the feature engineering stage, demanding low latency. You also need to support a large number of concurrent users accessing the data for ad-hoc analysis. Considering cost-effectiveness and

performance, which storage solution is MOST suitable for the initial data ingestion and ongoing storage during the ML lifecycle?

- A. AWS Snowball Edge to transfer the initial 50 TB to S3, followed by using S3 Intelligent-Tiering for ongoing storage.
- B. AWS Direct Connect to transfer the initial 50 TB to EFS, followed by using EFS Standard for ongoing storage.
- C. AWS Storage Gateway to transfer the initial 50 TB to S3, followed by using S3 Glacier Deep Archive for ongoing storage.
- D. AWS Direct Connect to transfer the initial 50 TB to FSx for NetApp ONTAP, followed by using its built-in tiering to S3.
- E. AWS Snowball Edge to transfer the initial 50 TB to FSx for NetApp ON TAP, followed by using FSx for ONTAP as the primary storage.

**Answer: D**

Explanation:

FSx for NetApp ONTAP, coupled with Direct Connect for initial transfer, offers a blend of high performance and cost efficiency through its built-in tiering to S3. It delivers low latency for the feature engineering stage and supports concurrent user access. S3 alone (Option A) might not provide the required low latency for the initial data processing. EFS (Option B) can become expensive at this scale. S3 Glacier (Option C) is unsuitable for frequent reads/writes. Transferring directly to FSx ONTAP via Snowball Edge is also feasible, but using direct connect provides ongoing bandwidth as data increases.

### Question: 5

You are building a distributed training pipeline for a large language model (LLM). The training data, consisting of billions of text files, is stored in an S3 bucket. Each training instance needs to access a specific subset of these files, and the training script uses the 'bot03' library to interact with S3. Due to the massive scale, you are encountering throttling issues and slow data retrieval times from S3. Which of the following strategies can you implement to optimize data ingestion and improve the training performance? (Select TWO)

- A. Enable S3 Transfer Acceleration to optimize data transfer speeds across long distances.
- B. Use the 's3transfer' library within 'bot03' to enable parallel downloads from S3.
- C. Implement an SQS queue to buffer requests to S3, reducing the request rate.
- D. Configure the S3 bucket for request rate scaling using adaptive request rate.
- E. Switch to using FSx for Lustre linked to the S3 bucket as a high-performance file system for training.

**Answer: B, E**

Explanation:

Using 's3transfer' (Option B) enables parallel downloads, significantly improving throughput compared to single-threaded downloads. While S3 Transfer Acceleration (Option A) helps with long-distance transfers, parallelization is crucial for large datasets within AWS. FSx for Lustre (Option E) provides a high-performance, shared file system that is optimized for ML workloads, significantly reducing data

access latency compared to directly accessing S3. SQS (Option C) adds latency and is not the best approach for data ingestion. While Adaptive request rate can help, it won't address the fundamental bottlenecks like high-latency S3 accesses as effectively as FSx or parallel downloads (Option D).

## Question: 6

You are developing a machine learning model that requires access to financial market data. This data is stored in a NetApp ONTAP system on-premises. You need to make this data accessible to your AWS-based machine learning environment with minimal latency and without migrating the entire dataset. You require versioning and snapshot capabilities for reproducibility of your experiments. Which configuration is MOST appropriate to meet these requirements?

- A. Use AWS Storage Gateway File Gateway to provide file-based access to the NetApp ONTAP data and replicate it to S3. Use S3 versioning for data reproducibility.
- B. Implement a VPN connection between your on-premises network and AWS, and directly access the NetApp ONTAP system using NFS from your EC2 instances.
- C. Use FSx for NetApp ONTAP in AWS and configure data replication from the on-premises NetApp ONTAP system using SnapMirror. Utilize FSx for ONTAP's snapshot capabilities for experiment reproducibility.
- D. Use AWS DataSync to periodically synchronize the data from the on-premises NetApp ONTAP to an EFS file system. Take EFS snapshots for reproducibility.
- E. Mount on-premise NetApp ONTAP datastore using AWS Storage gateway and configure access from the on-premise ML environment.

**Answer: C**

Explanation:

FSx for NetApp ONTAP (Option C) allows you to replicate data directly from your on-premises NetApp ONTAP system using SnapMirror. This provides minimal latency access to the data within AWS, leverages the existing NetApp infrastructure, and allows you to use the familiar snapshot capabilities for experiment reproducibility. Storage Gateway (Option A) replicates data to S3, which may introduce latency and might not be ideal for frequently accessed data. A VPN connection with direct NFS access (Option B) can be complex to manage and may not provide optimal performance. AWS DataSync to EFS (Option D) is suitable for periodic synchronization but does not provide the tight integration and low latency of FSx for ONTAP/SnapMirror. Option E is incorrect because AWS Storage Gateway provides access to AWS services from on-premise environments, not the other way around.

## Question: 7

You are tasked with merging two large datasets in S3 for a machine learning project. One dataset contains customer transaction data (transaction\_data.parquet), and the other contains product information (product\_info.csv). The transaction data has columns like 'customer\_id', 'product\_id', and 'transaction\_amount', while the product information data has columns like 'product\_id', 'product\_name', and You want to perform a join on 'product\_id' and store the resulting merged dataset back in S3 as a Parquet file. Due to the size of the data, you need to leverage Apache Spark on AWS

Glue. Select the code snippets that, when combined correctly, will achieve this data merging and storage using AWS Glue's dynamic frames.

A.

```
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from pyspark.context import SparkContext
```

```
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
```

```
transaction_data = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://your-bucket/transaction_data.parquet"]},
    format="parquet"
)
```

B.

```
product_info = spark.read.csv("s3://your-bucket/product_info.csv", header=True, inferSchema=True)
product_info_df = DynamicFrame.fromDF(product_info, glueContext, "product_info")
```

C.

```
joined_data = Join.apply(transaction_data, product_info, 'product_id', 'product_id')
```

D.

```
glueContext.write_dynamic_frame.from_options(
    frame=joined_data,
    connection_type="s3",
    connection_options={"path": "s3://your-bucket/merged_data/"},
    format="parquet"
)
```

E.

```
from awsglue.transforms import Join
```

**Answer: A,B,C,D,E**

Explanation:

The correct answer is to select all options. The code snippets, when combined, form a complete AWS Glue ETL job to merge the two datasets. A: Initializes the Glue context and reads the transaction data from S3 as a DynamicFrame. B: Reads the product information from S3 as a Spark DataFrame and converts it to a DynamicFrame. E: Imports the necessary 'Join' transform from sawsglue.transforms. C: Performs the join operation using the 'Join.apply' function on the 'product\_id' column. D: Writes the merged data to a new S3 location as a Parquet file. These snippets together demonstrate the end-to-end process of reading data from multiple sources, merging them, and writing the result back to S3 using AWS Glue and DynamicFrames.

## Question: 8

You are building a real-time fraud detection system that ingests transaction data from multiple sources. The data volume is highly variable, with peak loads exceeding 10 TB per hour. You've chosen to use Kinesis Data Streams to ingest the data, followed by Kinesis Data Firehose to deliver the data to S3 for storage. However, you are experiencing frequent throttling errors in Kinesis Data Streams and delayed data delivery to S3 during peak periods. You need to optimize the Kinesis Data Streams configuration to handle the variable load and ensure timely data delivery. Which of the following strategies would be MOST effective?

- A. Increase the number of shards in the Kinesis Data Stream significantly, and implement exponential backoff retry logic in the producers. Also, configure Kinesis Data Firehose to use dynamic partitioning based on time-based prefixes to improve S3 write performance.
- B. Reduce the Kinesis Data Streams shard count and enable enhanced fan-out consumers. Configure Kinesis Data Firehose to use a larger buffer interval and buffer size to batch records before writing to S3.
- C. Implement Kinesis Data Analytics to aggregate and filter the data before sending it to Kinesis Data Firehose. Increase the Kinesis Data Streams shard count linearly as the data ingestion rate increases.
- D. Switch to Kinesis Data Firehose direct PUT to S3 to bypass Kinesis Data Streams throttling issues. Configure S3 lifecycle policies to manage storage costs.
- E. Implement a custom solution using AWS Lambda to buffer and batch records before writing to Kinesis Data Streams. Use CloudWatch alarms to scale the Lambda function based on ingestion rate.

**Answer: A**

Explanation:

Option A is the most effective because increasing the number of shards allows Kinesis Data Streams to handle higher write throughput. Implementing exponential backoff in the producers prevents overwhelming the stream during bursts. Dynamic partitioning in Kinesis Data Firehose optimizes S3 write performance by preventing hot partitions. Option B is incorrect because reducing shards would exacerbate throttling. Option C is incorrect because Kinesis Data Analytics adds unnecessary complexity and latency. Option D bypasses the stream which is not always desirable when ordering matters. Option E, while functional, is more complex than simply scaling the Kinesis Data Streams appropriately.

## Question: 9

You are building a machine learning model that requires access to sensitive customer data stored in an S3 bucket. The data science team needs read-only access to the data for model training, but you must ensure that the data remains secure and compliant with regulatory requirements. You've implemented server-side encryption (SSE-KMS) with a customer-managed KMS key. Which of the following steps are NECESSARY to grant the data science team secure and auditable access to the S3 data while adhering to the principle of least privilege? (Select TWO)

- A. Create an IAM role for the data science team with read-only access to the S3 bucket. Attach a policy to the role that allows the 's3:GetObject' action on the bucket and objects.

- B. Grant the data science team direct access to the KMS key used to encrypt the data by adding their IAM users to the key policy with 'kms:Decrypt' permission.
- C. Add a condition to the IAM role policy that allows the 's3:GetObject' action only if the request includes the KMS key ID used for encryption in the 's3:x-amz-server-side-encryption-aws-kms-key-id' header.
- D. Grant the data science team 'kms:GenerateDataKey' permission on the KMS key, allowing them to create temporary decryption keys.
- E. Grant the IAM role assumed by the data science team 'kms:Decrypt' permission on the KMS key. Modify the S3 bucket policy to explicitly allow access from the data science team's role.

**Answer: A,E**

Explanation:

Options A and E are correct. Option A creates an IAM role with read-only access to the S3 bucket. Option E grants the role the necessary 'kms:Decrypt' permission on the KMS key. Both of these actions in conjunction allow the team to access the encrypted data, and the principle of least privilege. Option B is incorrect because granting direct access to the KMS key to individual users increases the risk of key compromise and is harder to manage. Option C is unnecessary because AWS handles KMS decryption automatically if the IAM role has 'kms:Decrypt' permission and the S3 data is encrypted with SSE-KMS. Option D is incorrect because it provides excessive permissions; the data science team only needs to decrypt the data, not generate new keys.

## Question: 10

You are tasked with preparing a dataset for a regression model to predict customer churn for a telecommunications company. The dataset contains a 'MonthlyCharges' feature with some missing values. You have decided to use imputation to fill in these missing values. Considering the following information: The 'MonthlyCharges' feature is normally distributed. The missing values constitute about 10% of the total data. There are no other highly correlated features that can be readily used for more sophisticated imputation methods. Which of the following imputation strategies, along with the corresponding Python code snippet using scikit-learn, is the MOST appropriate and computationally efficient for handling the missing values in the 'MonthlyCharges' feature?

A. Using Mean Imputation:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
data['MonthlyCharges'] = imputer.fit_transform(data[['MonthlyCharges']])
```

B. Using Median Imputation:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='median')
data['MonthlyCharges'] = imputer.fit_transform(data[['MonthlyCharges']])
```

C. Using K-Nearest Neighbors (KNN) Imputation:

```
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=5)
data['MonthlyCharges'] = imputer.fit_transform(data[['MonthlyCharges']])
```

D. Removing rows with missing values:

```
data = data.dropna(subset=['MonthlyCharges'])
```

E. Using a Constant Value Imputation:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='constant', fill_value=0)
data['MonthlyCharges'] = imputer.fit_transform(data[['MonthlyCharges']])
```

<b>Answer: A</b>
------------------

Explanation:

Since 'MonthlyCharges' is normally distributed and the missing values are only 10% of the data, mean imputation is the most appropriate and computationally efficient choice. It avoids introducing significant bias and is simple to implement. Median imputation (B) is better suited for skewed data. KNN imputation (C) is more computationally expensive and requires careful selection of the number of neighbors. Removing rows with missing values (D) may lead to a significant loss of data, which is undesirable. Constant value imputation (E) can introduce significant bias if the constant value is not representative of the true distribution.



# CERTSWARRIOR

## *FULL PRODUCT INCLUDES:*

Money Back Guarantee



Instant Download after Purchase



90 Days Free Updates



PDF Format Digital Download



24/7 Live Chat Support



Latest Syllabus Updates



**For More Information – Visit link below:**

**<https://www.certswarrior.com>**

**16 USD Discount Coupon Code: U89DY2AQ**