AWS MLA-C01 Exam Questions

Total Questions: 70+
Demo Questions: 15
Version: Updated for 2025

Prepared and Verified by Cert Empire – Your Trusted IT Certification Partner

For Access to the full set of Updated Questions – Visit:
AWS MLA-C01 Exam Questions by Cert Empire

Question: 1

A company is gathering audio, video, and text data in various languages. The company needs to use a
large language model (LLM) to summarize the gathered data that is in Spanish.
Which solution will meet these requirements in the LEAST amount of time?

A. Train and deploy a model in Amazon SageMaker to convert the data into English text. Train and
deploy an LLM in SageMaker to summarize the text.

B. Use Amazon Transcribe and Amazon Translate to convert the data into English text. Use Amazon
Bedrock with the Jurassic model to summarize the text.

C. Use Amazon Rekognition and Amazon Translate to convert the data into English text. Use Amazon
Bedrock with the Anthropic Claude model to summarize the text.

D. Use Amazon Comprehend and Amazon Translate to convert the data into English text. Use Amazon Bedrock with the Stable Diffusion model to summarize the text.

Answer:

B

Explanation:

The goal is to create a summarization pipeline for audio, video, and text data in Spanish in the least amount of time. This requires using pre-trained, fully managed AWS services.
1. Data Ingestion: The input includes audio and video. The spoken content must be converted to text. Amazon Transcribe is the appropriate managed service for automatic speech recognition (ASR) to convert speech from audio and video files into text.
2. Translation: The transcribed text and original text are in Spanish. Amazon Translate is a neural machine translation service designed to translate text into a target language like English.
3. Summarization: Amazon Bedrock provides API access to various foundation models (FMs), including Large Language Models (LLMs), without the need for training or managing infrastructure. The Jurassic model from AI21 Labs is an LLM available in Bedrock that is well-suited for text summarization.
This combination of services (Transcribe, Translate, Bedrock) creates the required pipeline using serverless, pre-trained models, making it the fastest solution to implement.

Why Incorrect Options are Wrong:

A: Training and deploying custom models in Amazon SageMaker is a complex and time-consuming process, directly contradicting the requirement for the solution with the "LEAST amount of time."

C: Amazon Rekognition is a computer vision service for analyzing images and videos (e.g., object detection, text detection in images), not for transcribing spoken audio, which is the primary task here.

D: Amazon Comprehend is an NLP service for text analysis (e.g., entity recognition), not speech-to-text. Furthermore, Stable Diffusion is a text-to-image model, not an LLM for text summarization.

References:

1. Amazon Transcribe: "Amazon Transcribe is an automatic speech recognition (ASR) service that makes it easy for you to add speech-to-text capabilities to your applications."
Source: AWS Documentation, "What is Amazon Transcribe?", aws.amazon.com/transcribe/
2. Amazon Translate: "Amazon Translate is a neural machine translation service that delivers fast, high-quality, affordable, and customizable language translation."
Source: AWS Documentation, "What is Amazon Translate?", aws.amazon.com/translate/
3. Amazon Bedrock & Jurassic Models: "Amazon Bedrock is a fully managed service that offers a choice of high-performing foundation models (FMs) from leading AI companies like AI21 Labs... via a single API... The Jurassic-2 family of models from AI21 Labs... are multilingual and can generate text in Spanish, French, German, Portuguese, Italian, and Dutch... for tasks such as question answering, summarization, and code generation."
Source: AWS Documentation, "Amazon Bedrock", aws.amazon.com/bedrock/ and "Foundation models" section.
4. Stable Diffusion in Bedrock: "With Amazon Bedrock, you can easily experiment with a variety of popular FMs... for text and images. You can choose from the latest FMs including... Stable Diffusion to accelerate the development of generative AI applications." (Stable Diffusion is listed for image generation).
Source: AWS Documentation, "Amazon Bedrock", aws.amazon.com/bedrock/

Question: 2

A financial company receives a high volume of real-time market data streams from an external provider. The streams consist of thousands of JSON records every second.
The company needs to implement a scalable solution on AWS to identify anomalous data points.
Which solution will meet these requirements with the LEAST operational overhead?

A. Ingest real-time data into Amazon Kinesis data streams. Use the built-in RANDOMCUTFOREST
function in Amazon Managed Service for Apache Flink to process the data streams and to detect data
anomalies.

B. Ingest real-time data into Amazon Kinesis data streams. Deploy an Amazon SageMaker endpoint
for real-time outlier detection. Create an AWS Lambda function to detect anomalies. Use the data streams to invoke the Lambda function.

C. Ingest real-time data into Apache Kafka on Amazon EC2 instances. Deploy an Amazon SageMaker
endpoint for real-time outlier detection. Create an AWS Lambda function to detect anomalies. Use the data streams to invoke the Lambda function.

D. Send real-time data to an Amazon Simple Queue Service (Amazon SQS) FIFO queue. Create an
AWS Lambda function to consume the queue messages. Program the Lambda function to start an
AWS Glue extract, transform, and load (ETL) job for batch processing and anomaly detection.

Answer:

A

Explanation:

This solution meets the requirements with the least operational overhead by using fully managed AWS services designed for this specific use case. Amazon Kinesis Data Streams is built for ingesting high-volume, real-time data. Amazon Managed Service for Apache Flink is a serverless, real-time processing service that directly integrates with Kinesis. Crucially, it includes the built-in RANDOMCUTFOREST function, which is an unsupervised algorithm specifically for detecting anomalies in streaming data. This combination provides a scalable, real-time anomaly detection pipeline without the need to provision servers, manage clusters, or develop, train, and deploy a separate machine learning model.

Why Incorrect Options are Wrong:

B: This approach introduces higher operational overhead by requiring the user to build, train, deploy, and manage a separate Amazon SageMaker endpoint, in addition to developing and managing the AWS Lambda integration code.

C: This option has the highest operational overhead because it involves self-managing an Apache Kafka cluster on Amazon EC2 instances, which requires significant effort for setup, scaling, patching, and maintenance.

D: This solution is incorrect because it uses AWS Glue for batch processing, which does not meet the real-time requirement. Amazon SQS is also less suited for real-time streaming analytics than Kinesis.

References:

1. Amazon Managed Service for Apache Flink, RANDOMCUTFOREST: "The RANDOMCUTFOREST function assigns an anomaly score to each record. Low scores indicate that the record is normal, and high scores indicate the presence of an anomaly in the data." This built-in function is designed for this exact purpose.

Source: AWS Documentation, Amazon Managed Service for Apache Flink Developer Guide, "RANDOMCUTFOREST".

URL: https://docs.aws.amazon.com/managed-flink/latest/apiv2/analytics-sql-reference-random-cut-forest.html

2. Amazon Kinesis Data Streams for Real-Time Processing: "You can use Kinesis Data Streams to collect and process large streams of data records in real time... The processed records can then be sent to dashboards, used to generate alerts, dynamically change pricing and advertising strategies, or send data to a variety of other AWS services."

Source: AWS Documentation, Amazon Kinesis Data Streams Developer Guide, "What Is Amazon Kinesis Data Streams?".

URL: https://docs.aws.amazon.com/streams/latest/dev/introduction.html

3. Operational Overhead of Self-Managed vs. Managed Services: Managed services like Kinesis and Managed Flink reduce operational burden compared to self-managing infrastructure like Apache Kafka on EC2. "Amazon Kinesis is fully managed and runs on a serverless architecture... With Apache Kafka, you are responsible for provisioning servers, managing cluster capacity..."

Source: AWS Documentation, Streaming Data Solutions on AWS with Amazon Kinesis, "Amazon Kinesis versus Apache Kafka".

URL: https://aws.amazon.com/kinesis/streaming-data/kinesis-vs-kafka/

Question: 3

A company has a large collection of chat recordings from customer interactions after a product release. An ML engineer needs to create an ML model to analyze the chat dat
a. The ML engineer needs to determine the success of the product by reviewing customer sentiments about the product.
Which action should the ML engineer take to complete the evaluation in the LEAST amount of time?

A. Use Amazon Rekognition to analyze sentiments of the chat conversations.

B. Train a Naive Bayes classifier to analyze sentiments of the chat conversations.

C. Use Amazon Comprehend to analyze sentiments of the chat conversations.

D. Use random forests to classify sentiments of the chat conversations.

Answer:

C

Explanation:

The question requires a solution for sentiment analysis of chat data that can be completed in the least amount of time. Amazon Comprehend is a managed Natural Language Processing (NLP) service that provides a pre-trained sentiment analysis API. Using this service eliminates the time-consuming tasks of data labeling, feature engineering, model training, and deployment. The ML engineer can simply send the chat text to the Comprehend API and receive sentiment scores, making it the most time-efficient solution among the choices.

Why Incorrect Options are Wrong:

A. Use Amazon Rekognition to analyze sentiments of the chat conversations.
Amazon Rekognition is a service for image and video analysis. It is not designed for analyzing sentiment from text data like chat recordings.
B. Train a Naive Bayes classifier to analyze sentiments of the chat conversations.
This involves building a custom model, which requires significant time for data preparation, labeling, training, and evaluation, contradicting the core requirement of speed.
D. Use random forests to classify sentiments of the chat conversations.
Similar to option B, this requires building a custom ML model from scratch, a process that is inherently more time-consuming than using a pre-built, managed service.

References:

1. Amazon Comprehend Documentation: "Amazon Comprehend is a natural language processing (NLP) service that uses machine learning to find insights and relationships in text.....You can use Amazon Comprehend to ... understand how positive or negative the text is. .. without having to develop and train your own models."
Source: AWS Documentation, "What is Amazon Comprehend?", https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html
2. Amazon Rekognition Documentation: "Amazon Rekognition makes it easy to add image and video analysis to your applications. .... You can detect objects, scenes, and faces in images and videos."
Source: AWS Documentation, "What Is Amazon Rekognition?", https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html
3. AWS AI Services Overview: AWS positions its AI services (like Comprehend) as the fastest way to add intelligence to applications, contrasting them with ML platforms (like SageMaker) used for building custom models. This highlights the speed advantage of using a managed service for a common task like sentiment analysis.
Source: AWS Machine Learning, "AI Services", https://aws.amazon.com/machine-learning/ai-services/

Question: 4

A company has a conversational AI assistant that sends requests through Amazon Bedrock to an Anthropic Claude large language model (LLM). Users report that when they ask similar questions multiple times, they sometimes receive different answers. An ML engineer needs to improve the responses to be more consistent and less random.
Which solution will meet these requirements?

A. Increase the temperature parameter and the topk parameter.

B. Increase the temperature parameter. Decrease the topk parameter.

C. Decrease the temperature parameter. Increase the topk parameter.

D. Decrease the temperature parameter and the topk parameter.

Answer:

D

Explanation:

To make a large language model's (LLM) output more consistent and less random, the generation process must be made more deterministic. Decreasing the temperature parameter makes the model more likely to select the tokens with the highest probability, reducing creativity and randomness. Similarly, decreasing the topk parameter restricts the model's choices to a smaller, more probable set of next tokens. The combination of decreasing both temperature and topk is the most effective strategy to minimize variability and ensure the model produces more consistent and predictable responses, directly addressing the stated problem.

Why Incorrect Options are Wrong:

A: Increasing both temperature and topk would significantly increase the randomness and diversity of the responses, which is the opposite of the required outcome.
B: Increasing the temperature parameter introduces more randomness, directly contradicting the goal of improving consistency, even with a lower topk.
C: While decreasing temperature is correct, increasing topk expands the pool of potential tokens, which can work against the goal of reducing randomness.

References:

1. Amazon Bedrock User Guide, "Inference parameters for foundation models": This official documentation specifies the function of parameters for models like Anthropic Claude. It states that a lower temperature results in more predictable text. It also defines Top K as limiting the token selection pool, where a smaller value restricts choice.
URL: https://docs.aws.amazon.com/bedrock/latest/userguide/model-parameters.html (Specifically, see the sections on "Top K" and "Temperature" under the Anthropic Claude model parameters.)

2. Holtzman, A., et al. (2020). "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). This foundational academic paper on text generation discusses sampling strategies. It explains that Top-K sampling truncates the probability distribution to the K most likely tokens, and a smaller K makes the output less random and more focused.
URL: https://arxiv.org/abs/1904.09751 (Section 3: "Decoding Strategies")
3. Zhao, W. X., et al. (2023). "A Survey of Large Language Models." ACM Computing Surveys, 56(9), 1-37. This survey article explains decoding strategies for LLMs. It clarifies that "a low temperature value makes the generation more deterministic," and describes Top-K sampling as a method to control the diversity of generated text.
URL: https://dl.acm.org/doi/10.1145/3626232 (Section 3.2.1: "Decoding Strategy")

Question: 5

A company is using ML to predict the presence of a specific weed in a farmer's field. The company is
using the Amazon SageMaker linear learner built-in algorithm with a value of multiclassdassifier for
the predictorjype hyperparameter.
What should the company do to MINIMIZE false positives?

A. Set the value of the weight decay hyperparameter to zero.

B. Increase the number of training epochs.

C. Increase the value of the targetprecision hyperparameter.

D. Change the value of the predictorjype hyperparameter to regressor.

Answer:

C

Explanation:

The primary goal is to minimize false positives, which are instances incorrectly predicted as positive (e.g., predicting a weed is present when it is not). This is equivalent to maximizing the precision metric, defined as True Positives / (True Positives + False Positives). To increase the precision score, the model must reduce the number of false positives. The Amazon SageMaker Linear Learner algorithm provides hyperparameters to optimize the model's classification threshold for specific metrics. By increasing the value of a precision-oriented hyperparameter, the training process is explicitly guided to favor a model that makes fewer false positive errors.

Why Incorrect Options are Wrong:

A. Set the value of the weight decay hyperparameter to zero: This removes L2 regularization, which can increase the risk of overfitting. It is a general model complexity control, not a direct tool for minimizing false positives.
B. Increase the number of training epochs: Increasing epochs can lead to better model convergence but also risks overfitting. It does not specifically target the trade-off between precision and recall to reduce false positives.
D. Change the value of the predictortype hyperparameter to regressor: This changes the problem from classification (predicting a category) to regression (predicting a continuous value), making the concept of false positives inapplicable.

References:

1. Amazon SageMaker Developer Guide: The Linear Learner documentation details hyperparameters for optimizing the model. For classification, the model can be optimized for metrics like precision. The guide states, "For the binaryclassifier predictor type, you can also optimize the model for a specific metric (precision, recall, fbeta) by setting the binaryclassifiermodelselectioncriteria hyperparameter." While the question specifies multiclassclassifier, the principle of optimizing for precision to reduce false positives is a core, intended mechanism presented in the algorithm's capabilities.
Source: Amazon SageMaker Developer Guide, "Linear Learner Algorithm," section on "Optimization Hyperparameters."
2. Stanford University CS229 Course Notes: Course materials on machine learning evaluation metrics define precision and its relationship to false positives. "Precision measures, of all the examples the classifier labeled as positive, what fraction were actually positive... A model with high precision will have a low number of false positives."
Source: Stanford University, CS229 Machine Learning, "Evaluation Metrics" by Andrew Ng. (Specifically, discussions on Precision-Recall).
3. Peer-Reviewed Publication: In "The Elements of Statistical Learning," the concepts of Type I (false positive) and Type II (false negative) errors are fundamental. Optimizing for precision is a standard strategy to control the rate of Type I errors.
Source: Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer. (See Chapter 7, "Model Assessment and Selection").

Question: 6

A company has implemented a data ingestion pipeline for sales transactions from its ecommerce website. The company uses Amazon Data Firehose to ingest data into Amazon OpenSearch Service.
The buffer interval of the Firehose stream is set for 60 seconds. An OpenSearch linear model generates real-time sales forecasts based on the data and presents the data in an OpenSearch dashboard.
The company needs to optimize the data ingestion pipeline to support sub-second latency for the real-time dashboard.
Which change to the architecture will meet these requirements?

A. Use zero buffering in the Firehose stream. Tune the batch size that is used in the PutRecordBatch
operation.

B. Replace the Firehose stream with an AWS DataSync task. Configure the task with enhanced fan-out
consumers.

C. Increase the buffer interval of the Firehose stream from 60 seconds to 120 seconds.

D. Replace the Firehose stream with an Amazon Simple Queue Service (Amazon SQS) queue.

Answer:

A

Explanation:

With Amazon Data Firehose the only controllable contributor to end-to-end latency is the delivery stream's buffering hint. Setting the buffer interval to 0 seconds (and using the smallest practical PutRecordBatch size) causes each record to be forwarded to Amazon OpenSearch Service as soon as it is received, eliminating the 60-second wait and allowing sub-second propagation to the dashboard.

Why Incorrect Options are Wrong:

B. AWS DataSync is a scheduled bulk-transfer service; it has no concept of streaming or enhanced fan-out consumers, so cannot achieve sub-second latency.
C. Doubling the buffer interval to 120 s increases, not decreases, latency.
D. SQS introduces queue polling latency; a consumer would still need to batch or long-poll messages, so sub-second delivery is not guaranteed.

References:

1. Amazon Kinesis Data Firehose Developer Guide - "BufferingHints: ... you can set IntervalInSeconds to 0 to disable buffering and have records delivered immediately." https://docs.aws.amazon.com/firehose/latest/dev/create-destination.html#es-buffering
2. AWS Big Data Blog - "Design patterns for near-real-time analytics using Amazon Kinesis Data Firehose and Amazon OpenSearch Service" (section: 'Reducing latency with zero-second buffering'). https://aws.amazon.com/blogs/big-data/design-patterns-for-near-real-time-analytics-using-amazon-kinesis-data-firehose-and-amazon-opensearch-service/

Question: 7

A company has trained an ML model in Amazon SageMaker. The company needs to host the model to provide inferences in a production environment. The model must be highly available and must respond with minimum latency. The size of each request will be between 1 KB and 3 M B. The model will receive unpredictable bursts of requests during the day. The inferences must adapt proportionally to the changes in demand. How should the company deploy the model into production to meet these requirements?

A. Create a SageMaker real-time inference endpoint. Configure auto scaling. Configure the endpoint to present the existing model.

B. Deploy the model on an Amazon Elastic Container Service (Amazon ECS) cluster. Use ECS scheduled scaling that is based on the CPU of the ECS cluster.

C. Install SageMaker Operator on an Amazon Elastic Kubernetes Service (Amazon EKS) cluster. Deploy the model in Amazon EKS. Set horizontal pod auto scaling to scale replicas based on the memory metric.

D. Use Spot Instances with a Spot Fleet behind an Application Load Balancer (ALB) for inferences. Use the ALBRequestCountPerTarget metric as the metric for auto scaling.

Answer:

A

Explanation:

The scenario requires a highly available, low-latency inference solution that can automatically scale to handle unpredictable traffic bursts. Amazon SageMaker real-time inference endpoints are specifically designed for this purpose. They provide a fully managed environment for hosting models, are optimized for low-latency responses, and can be deployed across multiple Availability Zones for high availability. Configuring auto scaling for the endpoint allows it to dynamically adjust the number of instances based on the workload, such as the number of invocations, ensuring it can adapt proportionally to the described traffic patterns. This solution directly and efficiently meets all the stated requirements using native SageMaker features.

References:

1. Amazon SageMaker Developer Guide - Deploy models for inference: "For inferences in real time, you can deploy your model to Amazon SageMaker endpoints. These endpoints are fully managed and can serve inferences in real-time with low latency." This supports the use of real-time endpoints for the latency requirement.
URL: https://docs.aws.amazon.com/sagemaker/latest/dg/deploy-model.html
2. Amazon SageMaker Developer Guide - Automatically Scale Amazon SageMaker Models: "Amazon SageMaker supports automatic scaling (auto scaling) for your production variants. Auto

scaling dynamically adjusts the number of instances provisioned for a production variant in response to changes in your workload... For high availability, deploy multiple instances for each production variant and deploy them across multiple Availability Zones." This confirms the solution for high availability and scaling for bursty traffic.
URL: https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling.html
3. Amazon SageMaker Developer Guide - Using Batch Transform: "Use batch transform when you need to get inferences for an entire dataset... Batch transform is ideal for scenarios where you have large datasets of data and don't need sub-second latency". This confirms why option B is incorrect.
URL: https://docs.aws.amazon.com/sagemaker/latest/dg/batch-transform.html

Question: 8

An ML engineer needs to use an Amazon EMR cluster to process large volumes of data in batches.
Any data loss is unacceptable.
Which instance purchasing option will meet these requirements MOST cost-effectively?

A. Run the primary node, core nodes, and task nodes on On-Demand Instances.

B. Run the primary node, core nodes, and task nodes on Spot Instances.

C. Run the primary node on an On-Demand Instance. Run the core nodes and task nodes on Spot
Instances.

D. Run the primary node and core nodes on On-Demand Instances. Run the task nodes on Spot Instances.

Answer:

D

Explanation:

This configuration provides the optimal balance between cost-effectiveness and the strict requirement of no data loss. The primary node, which manages the cluster, and the core nodes, which store data in the Hadoop Distributed File System (HDFS), are placed on reliable On-Demand Instances to prevent cluster failure or data loss. Task nodes, which only perform computation and do not store data, are run on cost-effective Spot Instances. If a Spot-based task node is terminated, the work can be rescheduled on another node without any loss of persistent data, thus meeting all requirements.

Why Incorrect Options are Wrong:

A: This option is highly reliable but is the most expensive. It is not the most cost-effective solution as it fails to leverage Spot Instances for interruptible workloads.
B: This option is the cheapest but is unacceptable because running the primary and core nodes on Spot Instances creates a high risk of cluster termination and HDFS data loss.
C: Running core nodes on Spot Instances is risky. The termination of a core node can lead to the loss of data stored on its HDFS component, violating the "no data loss" requirement.

References:

1. Amazon EMR Management Guide - Understanding node types: primary, core, and task nodes: This document defines the roles. The core node "stores data in the Hadoop Distributed File System (HDFS)," while the task node "runs tasks and doesn't store data in HDFS." This distinction is critical.

URL: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-overview-nodes.html
2. Amazon EMR Management Guide - Use Spot Instances: This guide provides best practices. It states, "Because task nodes don't host data, they are a good choice to run on Spot Instances." It also warns, "If the core node that is running the HDFS DataNode daemon terminates, you risk data loss."
URL: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-spot-instances.html
3. Amazon EMR Management Guide - Instance fleets: This documentation reinforces the strategy: "For the primary node group, On-Demand is the only allowed purchasing option... For core and task node groups, you can choose On-Demand, Spot, or a combination." The recommended practice for data durability is to use On-Demand for core nodes.
URL: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-instance-fleet.html

Question: 9

A company wants to improve the sustainability of its ML operations.
Which actions will reduce the energy usage and computational resources that are associated with the company's training jobs? (Choose two.)

A. Use Amazon SageMaker Debugger to stop training jobs when non-converging conditions are detected.

B. Use Amazon SageMaker Ground Truth for data labeling.

C. Deploy models by using AWS Lambda functions.

D. Use AWS Trainium instances for training.

E. Use PyTorch or TensorFlow with the distributed training option.

Answer:

A, D

Explanation:

Improving sustainability in ML operations involves minimizing energy and computational waste. AWS Trainium instances are purpose-built accelerators designed specifically for deep learning training, offering significantly better performance per watt compared to general-purpose CPUs or GPUs. This directly reduces the energy consumed for the same training task. Amazon SageMaker Debugger contributes by monitoring training jobs for issues like non-convergence. By automatically stopping a job that is no longer learning effectively, it prevents the continued, wasteful consumption of computational resources and energy for a futile training run.

Why Incorrect Options are Wrong:

B: Amazon SageMaker Ground Truth is a data labeling service used before training begins; it does not reduce the computational resources consumed during the training job itself.
C: AWS Lambda is a serverless compute service typically used for model deployment and inference, not for resource-intensive model training. The question specifically asks about training jobs.
E: Distributed training primarily aims to reduce the wall-clock time of training by using more resources in parallel. This often increases the total energy consumption due to communication overhead between nodes.

References:

AWS Trainium: According to official AWS documentation, "AWS Trainium is the second-generation machine learning (ML) accelerator that AWS purpose built for high-performance deep learning training." Purpose-built hardware is inherently more

energy-efficient for its specific task. (AWS, "AWS Trainium," https://aws.amazon.com/machine-learning/trainium/).
Amazon SageMaker Debugger: The official documentation states, "You can also configure SageMaker Debugger to automatically stop a training job when it detects an error... This helps you to avoid manually monitoring jobs and save on costs for wasted training runs." (AWS, "Debug and Profile Training Jobs with Amazon SageMaker Debugger," https://docs.aws.amazon.com/sagemaker/latest/dg/train-debugger.html).
Distributed Training: AWS documentation on distributed training focuses on speed: "Distributed training helps reduce the time it takes to train a model." It does not claim to reduce total energy or computational resources. (AWS, "Distributed Training," https://docs.aws.amazon.com/sagemaker/latest/dg/distributed-training.html).

Question: 10

A company is planning to create several ML prediction models. The training data is stored in Amazon
S3. The entire dataset is more than 5 in size and consists of CSV, JSON, Apache Parquet, and simple text files.
The data must be processed in several consecutive steps. The steps include complex manipulations
that can take hours to finish running. Some of the processing involves natural language processing
(NLP) transformations. The entire process must be automated.
Which solution will meet these requirements?

A. Process data at each step by using Amazon SageMaker Data Wrangler. Automate the process by
using Data Wrangler jobs.

B. Use Amazon SageMaker notebooks for each data processing step. Automate the process by using
Amazon EventBridge.

C. Process data at each step by using AWS Lambda functions. Automate the process by using AWS
Step Functions and Amazon EventBridge.

D. Use Amazon SageMaker Pipelines to create a pipeline of data processing steps. Automate the pipeline by using Amazon EventBridge.

Answer:

D

Explanation:

Amazon SageMaker Pipelines is the purpose-built service for creating, automating, and managing end-to-end machine learning (ML) workflows. It allows for the definition of a sequence of steps, such as data processing, model training, and evaluation, as a directed acyclic graph (DAG). Each step can run as a long-running job on dedicated infrastructure, which is necessary for processing a 5 TB dataset for hours. The entire pipeline can be versioned, shared, and automated. Triggers for automation can be configured using Amazon EventBridge, fulfilling all the scenario's requirements for a complex, automated, multi-step process.

Why Incorrect Options are Wrong:

A: Amazon SageMaker Data Wrangler is primarily an interactive tool for data preparation and feature engineering, not for orchestrating complex, multi-hour, end-to-end ML workflows that go beyond data prep.

B: Amazon SageMaker notebooks are interactive development environments. While they can be run on a schedule, they are not designed to be a robust, production-grade orchestration system for complex, dependent workflows.

C: AWS Lambda functions have a maximum execution timeout of 15 minutes, which makes them unsuitable for data processing steps that are expected to run for hours as described in the scenario.

References:

Amazon SageMaker Pipelines: According to the AWS Documentation, "Amazon SageMaker Pipelines is a continuous integration and continuous delivery (CI/CD) service that is purpose-built for machine learning (ML)... You can create a pipeline, a series of interconnected steps... These ML pipelines can be automated to run in response to events using Amazon EventBridge."
Source: AWS Documentation, "Amazon SageMaker Pipelines," https://docs.aws.amazon.com/sagemaker/latest/dg/pipelines.html

SageMaker Processing Jobs (within Pipelines): "With Amazon SageMaker, you can run processing jobs for data pre- or post-processing and for model evaluation... SageMaker manages the startup and teardown of the infrastructure." These jobs are designed for large-scale, long-running tasks.
Source: AWS Documentation, "Data Processing with Amazon SageMaker," https://docs.aws.amazon.com/sagemaker/latest/dg/processing-job.html

AWS Lambda Quotas: The official documentation states the maximum timeout for a Lambda function.
Source: AWS Documentation, "Lambda quotas," https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html (See "Timeout" row, value: 900 seconds / 15 minutes).

Question: 11

An ML engineer needs to use AWS CloudFormation to create an ML model that an Amazon SageMaker endpoint will host.
Which resource should the ML engineer declare in the CloudFormation template to meet this requirement?

A. AWS::SageMaker::Model

B. AWS::SageMaker::Endpoint

C. AWS::SageMaker::NotebookInstance

D. AWS::SageMaker::Pipeline

Answer:

A

Explanation:

To create an ML model artifact in Amazon SageMaker using AWS CloudFormation, the AWS::SageMaker::Model resource is used. This resource defines the model, including the location of the model artifacts (e.g., in an S3 bucket) and the inference container image. Creating this model resource is a prerequisite before it can be associated with an endpoint configuration (AWS::SageMaker::EndpointConfig) and then deployed to a real-time endpoint (AWS::SageMaker::Endpoint). The question specifically asks for the resource to create the model, not the endpoint that hosts it.

Why Incorrect Options are Wrong:

B. AWS::SageMaker::Endpoint: This resource creates the hosting service (the endpoint) but requires an EndpointConfig which, in turn, requires an existing Model to be specified.
C. AWS::SageMaker::NotebookInstance: This resource provisions a managed Jupyter notebook environment for model development and experimentation, not the deployable model artifact itself.
D. AWS::SageMaker::Pipeline: This resource defines an MLOps workflow for building, training, and deploying models, but the fundamental resource to declare the model artifact is AWS::SageMaker::Model.

References:

AWS CloudFormation User Guide: The AWS::SageMaker::Model resource is explicitly defined for creating a model in Amazon SageMaker. The documentation states, "The AWS::SageMaker::Model resource creates a model in Amazon SageMaker. In the request, you specify a name for the model and describe one or more containers."
Source: AWS CloudFormation User Guide, AWS::SageMaker::Model resource type. URL: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-sagemaker-model.ht

ml
AWS CloudFormation User Guide: The documentation for AWS::SageMaker::Endpoint shows that it depends on an EndpointConfig, which in turn depends on a Model. This confirms the sequential dependency where the model must be created first.
Source: AWS CloudFormation User Guide, AWS::SageMaker::Endpoint resource type. URL: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-sagemaker-endpoint.html

Question: 12

An advertising company uses AWS Lake Formation to manage a data lake. The data lake contains
structured data and unstructured dat
a. The company's ML engineers are assigned to specific advertisement campaigns.
The ML engineers must interact with the data through Amazon Athena and by browsing the data directly in an Amazon S3 bucket. The ML engineers must have access to only the resources that are
specific to their assigned advertisement campaigns.
Which solution will meet these requirements in the MOST operationally efficient way?

A. Configure IAM policies on an AWS Glue Data Catalog to restrict access to Athena based on the ML
engineers' campaigns.

B. Store users and campaign information in an Amazon DynamoDB table. Configure DynamoDB Streams to invoke an AWS Lambda function to update S3 bucket policies.

C. Use Lake Formation to authorize AWS Glue to access the S3 bucket. Configure Lake Formation tags
to map ML engineers to their campaigns.

D. Configure S3 bucket policies to restrict access to the S3 bucket based on the ML engineers' campaigns.

Answer:

C

Explanation:

The most operationally efficient solution is to use AWS Lake Formation's native capabilities for fine-grained access control. Lake Formation Tag-Based Access Control (LF-TBAC) is designed for this exact scenario. By assigning LF-Tags to Data Catalog resources (which represent the data in Amazon S3) based on campaign, and then granting permissions on those tags to the corresponding ML engineers' roles, the company can manage access centrally. This single policy framework is enforced for users accessing data through both Amazon Athena and directly in S3 (via Lake Formation's credential vending), meeting all requirements in a unified and scalable manner.

Why Incorrect Options are Wrong:

A: This approach is incomplete as it only addresses access through Athena and the AWS Glue Data Catalog, but not direct S3 bucket access.
B: This is an overly complex, custom-built solution that is not operationally efficient. It also only solves for S3 access, ignoring the Athena requirement.
D: This method only controls direct S3 bucket access and does not manage permissions for queries run through Athena. It also becomes difficult to manage at scale.

References:

1. AWS Lake Formation Developer Guide, "Tag-based access control with AWS Lake Formation": This guide explicitly describes using LF-Tags to manage permissions at scale. It states, "You can define LF-tags and assign them to Data Catalog resources... You then grant permissions on those LF-tags to principals. Lake Formation allows operations on the resources if the principal has permissions on the LF-tags for that resource." This directly supports the methodology in option C.
URL: https://docs.aws.amazon.com/lake-formation/latest/dg/tag-based-access-control.html
2. AWS Lake Formation Developer Guide, "How data access control works in Lake Formation": This document explains that Lake Formation provides a single place to manage access controls for data in an S3 data lake, which are then enforced across multiple services like Amazon Athena. This highlights the operational efficiency of a centralized model.
URL: https://docs.aws.amazon.com/lake-formation/latest/dg/how-it-works-access-control.html

3. AWS Big Data Blog, "Govern your data lake centrally with AWS Lake Formation tag-based access control": This official blog post details the exact use case of using tags to grant permissions to different groups (like ML engineers for specific campaigns), confirming it as the intended and most efficient pattern.
URL: https://aws.amazon.com/blogs/big-data/govern-your-data-lake-centrally-with-aws-lake-formation-tag-based-access-control/

Question: 13

An ML engineer needs to use data with Amazon SageMaker Canvas to train an ML model. The data is
stored in Amazon S3 and is complex in structure. The ML engineer must use a file format that minimizes processing time for the data.
Which file format will meet these requirements?

A. CSV files compressed with Snappy

B. JSON objects in JSONL format

C. JSON files compressed with gzip

D. Apache Parquet files

Answer:

D

Explanation:

Apache Parquet is a columnar storage file format optimized for performance in large-scale data processing. For complex data structures, Parquet's columnar layout significantly minimizes processing time by allowing Amazon SageMaker to read only the specific columns needed for analysis, rather than scanning entire rows. This drastically reduces I/O operations and accelerates data retrieval. Its support for efficient, splittable compression further enhances performance in distributed environments like those used by AWS services.

Why Incorrect Options are Wrong:

A. CSV files compressed with Snappy: CSV is a row-based format and is inefficient for querying specific columns from complex data, which increases I/O and processing time despite the fast Snappy compression.
B. JSON objects in JSONL format: JSONL is also row-based. While it handles complex structures, processing requires reading and parsing entire JSON objects, making it slower than Parquet's selective column reads.
C. JSON files compressed with gzip: A single, large gzip-compressed file is not splittable. This prevents parallel processing, creating a bottleneck and significantly increasing processing time for large datasets.

References:

1. AWS Big Data Blog: In a post on performance tuning, AWS states, "We recommend you use Apache Parquet or Apache ORC... Parquet is a columnar data format that is optimized for fast retrieval of data." This directly supports using Parquet to minimize processing time.
Source: AWS Big Data Blog, "Top 10 performance tuning tips for Amazon Athena," Tip #2. (URL:

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/)
2. Apache Parquet Official Documentation: The official documentation describes Parquet as "a columnar storage format... designed for efficient as well as performant flat columnar storage format of data compared to row based files like CSV or TSV files."
Source: The Apache Software Foundation, "Apache Parquet Documentation." (URL: https://parquet.apache.org/docs/)
3. Amazon SageMaker Developer Guide: The guide explains that for efficient data processing, especially with services like AWS Glue, converting data to Apache Parquet is a best practice. This reduces the amount of data scanned and improves performance.
Source: Amazon SageMaker Developer Guide, "Input and Output Data." (URL: https://docs.aws.amazon.com/sagemaker/latest/dg/cdf-training.html - Note: While this link discusses data formats for training, the principle of Parquet's performance advantage is a core concept across AWS analytics and ML services).

Question: 14

An ML engineer is evaluating several ML models and must choose one model to use in production.
The cost of false negative predictions by the models is much higher than the cost of false positive predictions.
Which metric finding should the ML engineer prioritize the MOST when choosing the model?

A. Low precision

B. High precision

C. Low recall

D. High recall

Answer:

D

Explanation:

The question states that the cost of false negatives (FN) is significantly higher than the cost of false positives (FP). Recall, also known as sensitivity, is the metric that measures a model's ability to correctly identify all actual positive cases. The formula for recall is True Positives / (True Positives + False Negatives). To minimize the number of costly false negatives, one must maximize the recall score. Therefore, the ML engineer should prioritize the model with the highest recall.

Why Incorrect Options are Wrong:

A. Low precision: Prioritizing low precision is never a desirable goal; it implies a high rate of false positives, which, while less costly here, is not the objective.
B. High precision: This metric focuses on minimizing false positives (TP / (TP + FP)). It would be prioritized if the cost of false positives were high, which is not the case here.
C. Low recall: This would mean the model has a high number of false negatives, which is the exact outcome the engineer must avoid due to its high cost.

References:

1. Official Vendor Documentation (AWS): Amazon SageMaker Developer Guide, "Evaluate models". The guide states, "For use cases where you want to avoid false negatives, you would want a high recall." This directly supports prioritizing high recall when false negatives are costly.
URL:
https://docs.aws.amazon.com/sagemaker/latest/dg/model-monitor-model-quality-metrics.html
(Under the "Confusion matrix" section, which explains the base metrics for recall).
2. University Courseware (Stanford): Stanford University, CS229 Machine Learning Course Notes.

The notes explain that in scenarios like cancer detection, where a false negative (missing a tumor) is far more critical than a false positive, optimizing for high recall is the correct strategy.
URL: http://cs229.stanford.edu/notes2019fall/cs229-notes-eval-metrics.pdf (Section 1.1, "Precision and Recall").
3. Peer-Reviewed Academic Publication (ACM): Davis, J., & Goadrich, M. (2006). "The relationship between Precision-Recall and ROC curves." Proceedings of the 23rd international conference on Machine learning. This paper discusses the importance of the Precision-Recall space in evaluating models, especially when there is a high cost associated with missing positive instances (i.e., requiring high recall).
URL: https://dl.acm.org/doi/10.1145/1143844.1143874 (Section 1, Introduction).

Question: 15

A company has trained and deployed an ML model by using Amazon SageMaker. The company needs
to implement a solution to record and monitor all the API call events for the SageMaker endpoint. The solution also must provide a notification when the number of API call events breaches a threshold.
Use SageMaker Debugger to track the inferences and to report metrics. Create a custom rule to provide a notification when the threshold is breached.
Which solution will meet these requirements?

A. Use SageMaker Debugger to track the inferences and to report metrics. Create a custom rule to
provide a notification when the threshold is breached.

B. Use SageMaker Debugger to track the inferences and to report metrics. Use the tensorvariance
built-in rule to provide a notification when the threshold is breached.

C. Log all the endpoint invocation API events by using AWS CloudTrail. Use an Amazon CloudWatch
dashboard for monitoring. Set up a CloudWatch alarm to provide notification when the threshold is
breached.

D. Add the Invocations metric to an Amazon CloudWatch dashboard for monitoring. Set up a CloudWatch alarm to provide notification when the threshold is breached.

Answer:

D

Explanation:

Amazon SageMaker endpoints automatically publish operational metrics to Amazon CloudWatch. The Invocations metric specifically counts the number of requests sent to a SageMaker endpoint. This solution directly addresses the requirements by using the native, purpose-built metric for monitoring invocation counts. A CloudWatch dashboard can be used for visualization, and a CloudWatch alarm can be configured on the Invocations metric to send a notification when a specified threshold is breached. This is the most direct and efficient method.

Why Incorrect Options are Wrong:

A: SageMaker Debugger is designed to capture and analyze internal model tensors during training or inference for debugging purposes, not for counting external API calls.
B: The tensorvariance rule is a specific SageMaker Debugger feature for monitoring tensor statistics during training, which is irrelevant to counting endpoint API invocations.
C: While AWS CloudTrail logs API calls for auditing, using it for high-frequency operational metrics is less efficient than using the dedicated CloudWatch metrics that SageMaker provides automatically.

References:

1. Amazon SageMaker Developer Guide: "Amazon SageMaker monitors your endpoints and sends metrics to Amazon CloudWatch. For real-time inference, these metrics are by variant... Invocations: The number of invocations of this variant."
Source: AWS Documentation, "Monitor Amazon SageMaker with Amazon CloudWatch," section on "Endpoint Invocation Metrics."
2. Amazon CloudWatch User Guide: "You can create a CloudWatch alarm that watches a single CloudWatch metric... If the metric breaches the threshold for a specified number of evaluation periods, the alarm performs one or more actions (such as sending a notification to an Amazon SNS topic)."
Source: AWS Documentation, "Using Amazon CloudWatch alarms."

3. Amazon SageMaker Developer Guide (on Debugger): "Amazon SageMaker Debugger provides tools to debug your training scripts in near real time... Debugger gains visibility into the training process by capturing tensors."
Source: AWS Documentation, "Debug and Profile Training Jobs Using Amazon SageMaker Debugger."