Actual4Dump

Choose the version that fits your needs	PDF Version	Desktop Test Engine	Online Test Engine
Latest and Up-to-Date exam dumps with real exam questions answers.	✓	✓	✓
Get 12-Months free updates without any extra charges.	✓	✓	✓
Experience same exam environment before appearing in the certification exam.	✗	✓	✓
100% exam passing guarante in the first attempt.	✓	✓	✓
20% discount on more than one license and 30% discount on 5+ license purchases.	✗	✓	✓
100% secure purchase on SSL.	✓	✓	✓
Completely private purchase without sharing your personal info with anyone.	✓	✓	✓

http://www.actual4dump.com

Superb Exam Dumps Materials lead you to get your certification easily - Actual4dump

Exam : **AIP-C01**

Title : AWS Certified Generative AI Developer - Professional

Vendor : Amazon

Version : DEMO

NO.1 A company is using Amazon Bedrock to design an application to help researchers apply for grants. The application is based on an Amazon Nova Pro foundation model (FM). The application contains four required inputs and must provide responses in a consistent text format. The company wants to receive a notification in Amazon Bedrock if a response contains bullying language. However, the company does not want to block all flagged responses.
The company creates an Amazon Bedrock flow that takes an input prompt and sends it to the Amazon Nova Pro FM. The Amazon Nova Pro FM provides a response.
Which additional steps must the company take to meet these requirements? (Select TWO.)
A. Use Amazon Bedrock Prompt Management to specify the required inputs as variables. Select an Amazon Nova Pro FM. Specify the output format for the response. Add the prompt to the prompts node of the flow.
B. Create an Amazon Bedrock guardrail that applies the hate content filter. Set the filter response to block.
Add the guardrail to the prompts node of the flow.
C. Create an Amazon Bedrock prompt router. Specify an Amazon Nova Pro FM. Add the required inputs as variables to the input node of the flow. Add the prompt router to the prompts node. Add the output format to the output node.
D. Create an Amazon Bedrock guardrail that applies the insults content filter. Set the filter response to detect. Add the guardrail to the prompts node of the flow.
E. Create an Amazon Bedrock application inference profile that specifies an Amazon Nova Pro FM.Specify the output format for the response in the description. Include a tag for each of the input variables. Add the profile to the prompts node of the flow.
Answer: A D
Explanation:
The correct answers are A and D because they collectively satisfy the requirements for structured inputs, consistent output formatting, and non-blocking detection of bullying language.
Option A is required because Amazon Bedrock Prompt Management enables prompt templates with explicit input variables and defined output formats. By defining the four required inputs as variables, the company ensures that every invocation of the Amazon Nova Pro FM receives the correct structured inputs. Specifying the output format ensures consistent responses, which is essential for a grants application workflow. Adding the managed prompt to the prompts node of the flow allows Bedrock Flows to invoke the model using this standardized configuration.
Option D addresses the requirement to receive notifications when bullying language is detected without blocking responses. Amazon Bedrock guardrails support content filters with configurable actions. By applying the insults content filter and setting the response action to detect, the system flags responses containing bullying or insulting language while still allowing the response to be returned. This enables monitoring, alerting, and auditing without interrupting application functionality.
Option B is incorrect because setting the filter response to block contradicts the requirement not to block all flagged responses. Option C introduces a prompt router, which is unnecessary because the application uses a single Amazon Nova Pro FM. Option E incorrectly attempts to enforce input variables and output formatting through an inference profile, which does not provide prompt-level variable enforcement or formatting guarantees.
Therefore, A and D together provide structured prompt management and non-blocking safety detection with minimal operational complexity.

NO.2 An elevator service company has developed an AI assistant application by using Amazon Bedrock. The application generates elevator maintenance recommendations to support the company's elevator technicians.
The company uses Amazon Kinesis Data Streams to collect the elevator sensor data.
New regulatory rules require that a human technician must review all AI-generated recommendations. The company needs to establish human oversight workflows to review and approve AI recommendations. The company must store all human technician review decisions for audit purposes.
Which solution will meet these requirements?
A. Create an AWS Glue workflow that has a human approval step. After the human technician review, integrate the application with an AWS Lambda function that calls the SendTaskSuccess API. Store all human technician review decisions in Amazon DynamoD
B. Create an AWS Step Functions workflow that has a human approval step that uses the waitForTaskToken API to pause execution. After a human technician completes a review, use an AWS Lambda function to call the SendTaskSuccess API with the approval decision. Store all review decisions in Amazon DynamoD
C. Configure Amazon EventBridge rules with custom event patterns to route AI recommendations to human technicians for review. Create AWS Glue jobs to process human technician approval queues.Use Amazon ElastiCache to cache all human technician review decisions.
D. Create a custom approval workflow by using AWS Lambda functions and Amazon SQS queues for human review of AI recommendations. Store all review decisions in Amazon DynamoDB for audit purposes.
Answer: B

NO.3 A financial services company uses multiple foundation models (FMs) through Amazon Bedrock for its generative AI (GenAI) applications. To comply with a new regulation for GenAI use with sensitive financial data, the company needs a token management solution.
The token management solution must proactively alert when applications approach model-specific token limits. The solution must also process more than 5,000 requests each minute and maintain token usage metrics to allocate costs across business units.
Which solution will meet these requirements?
A. Deploy an Amazon SQS dead-letter queue for failed requests. Configure an AWS Lambda function to analyze token-related failures. Use Amazon CloudWatch Logs Insights to generate reports on token usage patterns based on error logs from Amazon Bedrock API responses.
B. Develop model-specific tokenizers in an AWS Lambda function. Configure the Lambda function to estimate token usage before sending requests to Amazon Bedrock. Configure the Lambda function to publish metrics to Amazon CloudWatch and trigger alarms when requests approach thresholds. Store detailed token usage in Amazon DynamoDB to report costs.
C. Use Amazon API Gateway to create a proxy for all Amazon Bedrock API calls. Configure request throttling based on custom usage plans with predefined token quotas. Configure API Gateway to reject requests that will exceed token limits.
D. Implement Amazon Bedrock Guardrails with token quota policies. Capture metrics on rejected requests.Configure Amazon EventBridge rules to trigger notifications based on Amazon Bedrock Guardrails metrics. Use Amazon CloudWatch dashboards to visualize token usage trends across

models.

Answer: B

NO.4 A large ecommerce company has deployed a foundation model (FM) to generate product descriptions. The company ' s engineering team monitors technical metrics such as token usage, latency, and error rates by using Amazon CloudWatch. The company ' s marketing team tracks business metrics such as conversion rates and revenue impact in its own systems. The company needs a unified observability solution that correlates technical performance with business outcomes. The solution must provide automatic alerts to stakeholders when operational metrics indicate degradation. The solution must provide comprehensive visibility across both technical and business metrics. Which solution will meet these requirements?

A. Use Amazon Managed Grafana to visualize technical metrics from CloudWatch with business metrics from external sources. Configure Amazon Managed Grafana alerts to invoke AWS Lambda functions.
Configure the Lambda functions to remediate issues automatically when metrics exceed predefined thresholds.

B. Stream CloudWatch metrics to Amazon S3 by using CloudWatch metric streams. Create Amazon QuickSight dashboards to visualize the combined technical metrics and business metrics. Set up Amazon EventBridge rules to send notifications to stakeholders when metrics exceed predefined thresholds.

C. Create CloudWatch dashboards that include technical metrics and imported business metrics. Configure CloudWatch composite alarms that combine technical data and business data. Use Amazon SNS to set up notifications to stakeholders.

D. Configure CloudWatch custom dashboards that integrate operational metrics with imported business metrics. Set up CloudWatch composite alarms with anomaly detection. Use Amazon SNS to create alarm actions to notify stakeholders when correlated metrics indicate performance issues.

Answer: D

NO.5 A company has a generative AI (GenAI) application that uses Amazon Bedrock to provide real-time responses to customer queries. The company has noticed intermittent failures with API calls to foundation models (FMs) during peak traffic periods.
The company needs a solution to handle transient errors and provide detailed observability into FM performance. The solution must prevent cascading failures during throttling events and provide distributed tracing across service boundaries to identify latency contributors. The solution must also enable correlation of performance issues with specific FM characteristics.
Which solution will meet these requirements?

A. Implement a custom retry mechanism with a fixed delay of 1 second between retries. Configure Amazon CloudWatch alarms to monitor the application's error rates and latency metrics.

B. Configure the AWS SDK with standard retry mode and exponential backoff with jitter. Use AWS X-Ray tracing with annotations to identify and filter service components.

C. Implement client-side caching of all FM responses. Add custom logging statements in the application code to record API call durations.

D. Configure the AWS SDK with adaptive retry mode. Use AWS CloudTrail distributed tracing to monitor throttling events.

Answer: B
Explanation:
Option B best meets the combined resiliency and observability requirements because it applies AWS-recommended retry behavior for transient throttling and enables true distributed tracing across service boundaries. During peak traffic, intermittent failures are commonly caused by throttling and other transient conditions. The AWS SDK standard retry mode provides exponential backoff with jitter, which reduces synchronized retry storms, prevents cascading failures, and improves overall system stability. Jitter is important because it spreads retry attempts over time, reducing load amplification during throttling events.
For observability, AWS X-Ray provides distributed tracing that follows a request across components such as API Gateway or load balancers, application services, and downstream calls to Amazon Bedrock. X-Ray can identify where latency is being introduced and which downstream call is contributing most to end-to-end response time. This is required to "identify latency contributors" and isolate performance issues under load.
The requirement also states that the company must correlate performance issues with specific FM characteristics. X-Ray annotations are designed for this purpose: the application can annotate traces with the model ID, inference parameters, region, or inference profile used. This enables filtering and analysis (for example, comparing latency or error patterns by model, parameter set, or endpoint configuration) without building a separate telemetry system.
Option A's fixed-delay retries increase synchronized retry behavior and do not provide distributed tracing.
Option C does not prevent cascading failures and cannot provide cross-service tracing. Option D is incorrect because CloudTrail is an audit logging service and does not provide distributed tracing for request latency analysis.
Therefore, Option B provides the correct combination of resilient retries and deep, model-correlated distributed observability for Amazon Bedrock workloads.

NO.6 A medical device company wants to feed reports of medical procedures that used the company's devices into an AI assistant. To protect patient privacy, the AI assistant must expose patient personally identifiable information (PII) only to surgeons. The AI assistant must redact PII for engineers. The AI assistant must reference only medical reports that are less than 3 years old.
The company stores reports in an Amazon S3 bucket as soon as each report is published. The company has already set up an Amazon Bedrock Knowledge Bases. The AI assistant uses Amazon Cognito to authenticate users.
Which solution will meet these requirements?
A. Enable Amazon Macie PII detection on the S3 bucket. Use an S3 trigger to invoke an AWS Lambda function that redacts PII from the reports. Configure the Lambda function to delete outdated documents and invoke knowledge base syncing.
B. Invoke an AWS Lambda function to sync the S3 bucket and the knowledge base when a new report is uploaded. Use a second Lambda function with Amazon Comprehend to redact PII for engineers. Use S3 Lifecycle rules to remove reports older than 3 years.
C. Set up an S3 Lifecycle configuration to remove reports that are older than 3 years. Schedule an AWS Lambda function to run daily syncs between the bucket and the knowledge base. When users interact with the AI assistant, apply a guardrail configuration selected based on the user's Cognito user group to redact PII from responses when required.

D. Create a second knowledge base. Use Lambda and Amazon Comprehend to redact PII before syncing to the second knowledge base. Route users to the appropriate knowledge base based on Cognito group membership.

Answer: C

Explanation:

Option C is the correct solution because it enforces privacy controls at inference time, not at ingestion time, which is required when different user roles require different visibility into the same underlying data.

Using an S3 Lifecycle configuration ensures that documents older than 3 years are automatically removed, guaranteeing that the knowledge base references only compliant, recent medical reports. Scheduling Lambda- based syncs keeps the knowledge base aligned with the bucket contents without introducing complex per- upload orchestration.

The most important requirement is role-based PII exposure. Amazon Bedrock guardrails support dynamic application at inference time, allowing the system to select a guardrail configuration based on the authenticated user's Amazon Cognito group. Surgeons can receive full responses, while engineers receive responses with PII masked-without duplicating data or maintaining multiple knowledge bases.

This approach preserves a single source of truth for medical reports while enforcing privacy through response- level controls. It also maintains full auditability of access and redaction behavior.

Option A permanently removes PII and violates surgeon access requirements. Option B redacts data inconsistently and couples privacy logic to ingestion. Option D doubles storage, increases cost, and introduces data drift risk.

Therefore, Option C best meets privacy, compliance, scalability, and operational efficiency requirements.

NO.7 A media company is launching a platform that allows thousands of users every hour to upload images and text content. The platform uses Amazon Bedrock to process the uploaded content to generate creative compositions.

The company needs a solution to ensure that the platform does not process or produce inappropriate content.

The platform must not expose personally identifiable information (PII) in the compositions. The solution must integrate with the company ' s existing Amazon S3 storage workflow.

Which solution will meet these requirements with the LEAST infrastructure management overhead?

A. Enable the Enhanced Monitoring tool. Use an Amazon CloudWatch alarm to filter traffic to the platform. Use Amazon Comprehend PII detection to pre-process the data. Create a CloudWatch alarm to monitor for Amazon Comprehend PII detection events. Create an AWS Step Functions workflow that includes an Amazon Rekognition image moderation step.

B. Use an Amazon API Gateway HTTP API with request validation templates to screen content before storing the uploaded content in Amazon S3. Use Amazon SageMaker AI to build custom content moderation models that process content before sending the processed content to Amazon Bedrock.

C. Create an Amazon Cognito user pool that uses pre-authentication AWS Lambda functions to run content moderation checks. Use Amazon Textract to filter text content and Amazon Rekognition to filter image content before allowing users to upload content to the platform.

D. Create an AWS Step Functions workflow that uses built-in Amazon Bedrock guardrails to filter content. Use Amazon Comprehend PII detection to pre-process the content. Use Amazon Rekognition

image moderation.

Answer: D

Explanation:

Option D is the correct solution because it relies primarily on managed, purpose-built AWS services and minimizes custom infrastructure and model management. Amazon Bedrock guardrails provide native, configurable content safety controls that can block or redact disallowed content before or after model inference. This directly ensures that the platform does not process or produce inappropriate outputs while maintaining low operational overhead.

Using Amazon Comprehend PII detection as a preprocessing step integrates cleanly with an Amazon S3- based ingestion workflow. Comprehend is a fully managed service that detects and optionally redacts PII in text without requiring custom models or pipelines. This ensures that sensitive information is removed before content is passed to Amazon Bedrock for generation.

Amazon Rekognition image moderation is purpose-built for detecting unsafe or inappropriate visual content and integrates naturally into Step Functions workflows. Step Functions provides orchestration without requiring servers or long-running infrastructure, allowing the company to integrate text and image moderation steps in a clear, auditable pipeline.

Option A introduces redundant monitoring logic and alarms that do not directly enforce content safety. Option B requires building and maintaining custom SageMaker models, increasing complexity and operational burden. Option C applies moderation at authentication time and uses services like Textract that are not designed for content moderation, increasing latency and management overhead.

Therefore, Option D best satisfies content safety, PII protection, S3 integration, and minimal infrastructure management requirements.

NO.8 A financial services company is developing a customer service AI assistant by using Amazon Bedrock. The AI assistant must not discuss investment advice with users. The AI assistant must block harmful content, mask personally identifiable information (PII), and maintain audit trails for compliance reporting. The AI assistant must apply content filtering to both user inputs and model responses based on content sensitivity.

The company requires an Amazon Bedrock guardrail configuration that will effectively enforce policies with minimal false positives. The solution must provide multiple handling strategies for multiple types of sensitive content.

Which solution will meet these requirements?

A. Configure a single guardrail and set content filters to high for all categories. Set up denied topics for investment advice and include sample phrases to block. Set up sensitive information filters that apply the block action for all PII entities. Apply the guardrail to all model inference calls.

B. Configure multiple guardrails by using tiered policies. Create one guardrail and set content filters to high. Configure the guardrail to block PII for public interactions. Configure a second guardrail and set content filters to medium. Configure the second guardrail to mask PII for internal use. Configure multiple topic-specific guardrails to block investment advice and set up contextual grounding checks.

C. Configure a guardrail and set content filters to medium for harmful content. Set up denied topics for investment advice and include clear definitions and sample phrases to block. Configure sensitive information filters to mask PII in responses and to block financial information in inputs. Enable both input and output evaluations that use custom blocked messages for audits.

D. Create a separate guardrail for each use case. Create one guardrail that applies a harmful content

filter.Create a guardrail to apply topic filters for investment advice. Create a guardrail to apply sensitive information filters to block PII. Use AWS Step Functions to chain the guardrails sequentially.

Answer: C

Explanation:

Option C is the correct solution because it uses a single, well-tuned Amazon Bedrock guardrail that applies different actions to different content types, which is the recommended approach for minimizing false positives while enforcing strong policy controls.

Setting content filters to medium rather than high reduces overblocking of benign customer conversations while still preventing harmful content. Amazon Bedrock guardrails are designed to balance precision and recall, and medium sensitivity is commonly recommended for customer-facing financial services use cases.

Denied topics explicitly prevent the assistant from discussing investment advice, which is a regulatory requirement. Including definitions and sample phrases improves detection accuracy and reduces ambiguity.

Sensitive information filters support different actions per context. Masking PII in responses preserves conversational usefulness for legitimate customer support while preventing exposure of sensitive data.

Blocking sensitive financial information in inputs prevents downstream processing of disallowed content before it reaches the foundation model.

Critically, enabling both input and output evaluation ensures that guardrails are applied consistently at every stage of interaction. Custom blocked messages and audit logging provide clear compliance evidence for regulators and internal audits.

Option A causes excessive false positives by blocking all PII outright. Option B introduces unnecessary complexity and is not how Bedrock guardrails are intended to be applied. Option D uses orchestration logic that Bedrock guardrails already handle natively.

Therefore, Option C best satisfies enforcement, flexibility, auditability, and accuracy requirements.

NO.9 A company uses Amazon Bedrock to build a Retrieval Augmented Generation (RAG) system. The RAG system uses an Amazon Bedrock Knowledge Bases that is based on an Amazon S3 bucket as the data source for emergency news video content. The system retrieves transcripts, archived reports, and related documents from the S3 bucket.

The RAG system uses state-of-the-art embedding models and a high-performing retrieval setup. However, users report slow responses and irrelevant results, which cause decreased user satisfaction. The company notices that vector searches are evaluating too many documents across too many content types and over long periods of time.

The company determines that the underlying models will not benefit from additional fine-tuning. The company must improve retrieval accuracy by applying smarter constraints and wants a solution that requires minimal changes to the existing architecture.

Which solution will meet these requirements?

A. Enhance embeddings by using a domain-adapted model that is specifically trained on emergency news content for improved vector similarity.

B. Migrate to Amazon OpenSearch Service. Use vector fields and metadata filters to define the scope of results retrieval.

C. Enable metadata-aware filtering within the Amazon Bedrock knowledge base by indexing S3 object metadata.

D. Migrate to an Amazon Q Business index to perform structured metadata filtering and document

categorization during retrieval.

Answer: C

Explanation:

Option C is the correct solution because it directly addresses the root cause of the problem-overly broad retrieval-while requiring minimal architectural change. Amazon Bedrock Knowledge Bases support metadata-aware filtering, which allows the system to constrain retrieval queries based on indexed metadata such as content type, publication date, source, or category.

By indexing Amazon S3 object metadata, the company can restrict vector searches to relevant subsets of the corpus, such as recent emergency reports, specific content formats, or trusted sources. This significantly reduces the number of documents evaluated during retrieval, which improves both latency and result relevance without changing embedding models or retrieval infrastructure.

This approach aligns with AWS best practices for optimizing RAG systems: when embeddings are already strong, retrieval quality is often improved by narrowing the candidate set rather than increasing model complexity. Metadata filtering reduces noise and ensures that retrieved documents are more contextually aligned with user queries.

Option A requires retraining or adapting embedding models, which the company has already determined will not provide additional benefit. Option B introduces a migration to OpenSearch, which adds operational overhead and deviates from the existing Bedrock knowledge base architecture. Option D requires moving to a different indexing service, increasing complexity and implementation effort.

Therefore, Option C provides the most effective and low-effort solution to improve retrieval accuracy and performance in the existing Amazon Bedrock RAG system.

NO.10 A media company must use Amazon Bedrock to implement a robust governance process for AI-generated content. The company needs to manage hundreds of prompt templates. Multiple teams use the templates across multiple AWS Regions to generate content. The solution must provide version control with approval workflows that include notifications for pending reviews. The solution must also provide detailed audit trails that document prompt activities and consistent prompt parameterization to enforce quality standards.

Which solution will meet these requirements?

A. Configure Amazon Bedrock Studio prompt templates. Use Amazon CloudWatch dashboards to display prompt usage metrics. Store approval status in Amazon DynamoDB. Use AWS Lambda functions to enforce approvals.

B. Use Amazon Bedrock Prompt Management to implement version control. Configure AWS CloudTrail for audit logging. Use AWS Identity and Access Management policies to control approval permissions.

Create parameterized prompt templates by specifying variables.

C. Use AWS Step Functions to create an approval workflow. Store prompts in Amazon S3. Use tags to implement version control. Use Amazon EventBridge to send notifications.

D. Deploy Amazon SageMaker Canvas with prompt templates stored in Amazon S3. Use AWS CloudFormation for version control. Use AWS Config to enforce approval policies.

Answer: B

Explanation:

Option B is the correct solution because Amazon Bedrock Prompt Management is purpose-built to manage, govern, and standardize prompt usage at scale across teams and Regions. It provides native

version control, allowing teams to track prompt changes over time and ensure that only approved versions are used in production workflows.
Prompt Management supports approval workflows that align with enterprise governance requirements.
Approval permissions can be enforced through IAM policies, ensuring that only authorized reviewers can approve or publish prompt versions. This removes the need for custom workflow engines or external storage systems, significantly reducing operational overhead.
Parameterized prompt templates enable consistent prompt structure while allowing controlled variation through defined variables. This ensures consistent quality standards and reduces prompt drift, which is critical when hundreds of prompts are reused across multiple applications and teams.
AWS CloudTrail integrates natively with Amazon Bedrock to provide immutable audit logs for prompt creation, updates, approvals, and usage. These detailed audit trails satisfy compliance requirements and allow security and governance teams to trace prompt activity across Regions and users.
Option A requires significant custom development to coordinate approvals and maintain state.
Option C relies on general-purpose workflow services and manual versioning mechanisms that are error-prone and difficult to scale. Option D uses services not designed for large-scale GenAI prompt governance and introduces unnecessary complexity.
Therefore, Option B best meets the requirements for scalable, auditable, and low-overhead governance of AI- generated content using Amazon Bedrock.

NO.11 A company is using Amazon Bedrock to develop an AI-powered application that uses a foundation model that supports cross-Region inference and provisioned throughput. The application must serve users in Europe and North America with consistently low latency. The application must comply with data residency regulations that require European user data to remain within Europe-based AWS Regions.
During testing, the application experiences service degradation when Regional traffic spikes reach service quotas. The company needs a solution that maintains application resilience and minimizes operational complexity.
Which solution will meet these requirements?

A. Deploy separate Amazon Bedrock instances in North American and European Regions. Use a custom routing layer that directs traffic based on user location. Configure Amazon CloudWatch alarms to monitor Regional service usage. Use Amazon SNS to send email alerts to the company when usage approaches specified thresholds.

B. Use Amazon Bedrock cross-Region inference profiles by specifying geographical codes in profile IDs when the application calls the InvokeModel API. Configure separate Amazon API Gateway HTTP APIs to direct European and North American users to the appropriate Regional endpoints.

C. Deploy a multi-Region Amazon API Gateway HTTP API and AWS Lambda functions that implement retry logic to handle throttling. Configure the Lambda functions to call the foundation model in the nearest secondary Region when the application reaches service quotas in the primary Region. Use intelligent routing to ensure compliance with data residency requirements.

D. Configure provisioned throughput for Amazon Bedrock in multiple Regions. Implement failover logic in the application code to switch between Regions when throttling occurs. Use AWS Global Accelerator to route traffic to the appropriate endpoints based on user location.

Answer: B
Explanation:

Option B best meets the latency, resilience, and data residency requirements while keeping operational complexity low by using built-in Amazon Bedrock cross-Region inference behavior through inference profiles . Cross-Region inference profiles are designed to provide higher availability and better traffic absorption when a single Region experiences throttling, transient capacity constraints, or quota-related degradation. By selecting the appropriate geography-scoped inference profile (for example, a Europe-scoped profile for European users and a North America-scoped profile for North American users), the application can keep inference traffic within the required geographic boundary. This directly supports EU data residency needs because European requests can be served only by Europe-based Regions while still benefiting from multi-Region resilience inside Europe.
The question also highlights degradation when Regional traffic spikes hit quotas. Cross-Region inference profiles help mitigate these conditions by allowing Bedrock to serve requests from another Region within the same geography, improving continuity during spikes without requiring the company to implement custom retry-and-failover logic across Regions. This reduces development and operational burden compared to building and maintaining a bespoke routing and fallback system.
Using separate Amazon API Gateway HTTP APIs to direct European and North American users to the correct endpoints simplifies request routing and provides a clean boundary for compliance controls, logging, and monitoring. It also allows each geography to scale independently and maintain consistently low latency by keeping users close to the entry point and the Bedrock geography they must use.
Option A requires custom routing and manual operational monitoring and does not inherently solve quota- driven degradation. Option C adds significant complexity by embedding throttling retries and cross-Region selection logic in Lambda while still needing careful controls to prevent cross-border routing mistakes. Option D introduces the highest operational complexity and can inadvertently violate residency if failover crosses geographies unless additional safeguards are implemented.

NO.12 A company is building a generative AI (GenAI) application that processes financial reports and provides summaries for analysts. The application must run two compute environments. In one environment, AWS Lambda functions must use the Python SDK to analyze reports on demand. In the second environment, Amazon EKS containers must use the JavaScript SDK to batch process multiple reports on a schedule. The application must maintain conversational context throughout multi-turn interactions, use the same foundation model (FM) across environments, and ensure consistent authentication.
Which solution will meet these requirements?

A. Use the Amazon Bedrock InvokeModel API with a separate authentication method for each environment. Store conversation states in Amazon DynamoDB. Use custom I/O formatting logic for each programming language.

B. Use the Amazon Bedrock Converse API directly in both environments with a common authentication mechanism that uses IAM roles. Store conversation states in Amazon ElastiCache. Create programming language-specific wrappers for model parameters.

C. Create a centralized Amazon API Gateway REST API endpoint that handles all model interactions by using the InvokeModel API. Store interaction history in application process memory in each Lambda function or EKS container. Use environment variables to configure model parameters.

D. Use the Amazon Bedrock Converse API and IAM roles for authentication. Pass previous messages in the request messages array to maintain conversational context. Use programming language-

specific SDKs to establish consistent API interfaces.

Answer: D

Explanation:

Option D is the correct solution because the Amazon Bedrock Converse API is purpose-built for multi-turn conversational interactions and is designed to work consistently across SDKs and compute environments. The Converse API standardizes how messages, roles, and context are represented, which ensures consistent behavior whether the application is running in AWS Lambda with Python or in Amazon EKS with JavaScript.

By passing previous messages in the messages array, the application explicitly maintains conversational context across turns without relying on external state stores. This approach is recommended by AWS for conversational GenAI workflows because it avoids state synchronization complexity and ensures deterministic model behavior across environments.

Using IAM roles for authentication provides a single, consistent security model for both Lambda and EKS.

IAM roles integrate natively with AWS SDKs, eliminating the need for custom authentication logic or environment-specific credentials. This aligns with AWS best practices for least privilege and simplifies governance.

Option A introduces inconsistent authentication and custom formatting logic, increasing complexity. Option B unnecessarily introduces ElastiCache for state management, which is not required when using the Converse API correctly. Option C stores state in process memory, which is unsafe and unreliable for serverless and containerized workloads.

Therefore, Option D best satisfies the requirements for conversational consistency, multi-environment support, shared model usage, and consistent authentication with minimal operational overhead.