DUMPSBOSS.

AWS Certified Generative AI DeveloperProfessional

Amazon Web Services AIP-C01

Version Demo

Total Demo Questions: 15

Total Premium Questions: 153

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Generative AI Application Development	75
Topic 2, Generative AI Model Development and Fine-Tuning	5
Topic 3, Generative AI Model Deployment and Inference	36
Topic 4, Generative AI Security, Compliance, and Governance	36
Topic 5, Mix Questions	1
Total	153

QUESTION NO: 1

A company is building a real-time voice assistant system to assist customer service representatives during customer calls. The system must convert audio calls to text with end-to-end latency of less than 500 ms. The system must use generative AI (GenAI) to produce response suggestions. Human supervisors must be able to rate the system ' s suggestions during a live customer call. The company must store all customer interactions to comply with auditing policies. Which solution will meet these requirements?

A. Use the Amazon Transcribe streaming API with standard settings to convert speech to text. Use Amazon Bedrock batch processing to perform inference. Store call recordings and metadata in Amazon S3. Use S3 Lifecycle policies to manage the storage.

B. Use the Amazon Transcribe streaming API with 100-ms audio chunks to optimize latency for the voice assistant. Call the Amazon Bedrock InvokeModelWithResponseStream operation to process client inquiries in real time. Store supervisor ratings in an Amazon DynamoDB table.

C. Use Amazon Transcribe batch processing to perform post-call analysis. Configure AWS Lambda functions to generate responses by using the Amazon Bedrock InvokeModel operation. Use Amazon CloudWatch to log supervisor feedback.

D. Use Amazon Transcribe to convert speech to text and to perform real-time analytics. Use Amazon Comprehend to perform sentiment analysis. Use Amazon SQS to queue processing tasks. Run the Amazon Bedrock InvokeModel operation to generate responses.

E. Use the Amazon Transcribe streaming API with 100-ms audio chunks to convert live call audio to text, and invoke Amazon Bedrock with InvokeModelWithResponseStream to generate response suggestions as streamed output. Store supervisor ratings in Amazon DynamoDB, and store call recordings, transcripts, generated suggestions, and associated interaction metadata in Amazon S3 for auditing.

ANSWER: E

Explanation:

Use the Amazon Transcribe streaming API with 100-ms audio chunks to convert live call audio to text, and invoke Amazon Bedrock with `InvokeModelWithResponseStream` to generate response suggestions as streamed output. Store supervisor ratings in Amazon DynamoDB, and store call recordings, transcripts, generated suggestions, and associated interaction metadata in Amazon S3 for auditing. This design uses streaming across the latency-sensitive path: Amazon Transcribe Streaming can send partial transcription results while audio is still being received, and Amazon Bedrock response streaming returns generated output incrementally instead of waiting for a complete model response. Small audio chunks help reduce the time before transcription processing begins. DynamoDB supports low-latency writes, making it suitable for supervisors submitting ratings during active calls. Amazon S3 provides durable, scalable storage for the complete interaction record required by audit policies, including original audio and artifacts produced during the call. The application should associate all stored objects and rating records with a common call or interaction identifier so that auditors can reconstruct the complete sequence of customer input, generated recommendations, and human feedback. See the Amazon Transcribe Streaming documentation and the Amazon Bedrock InvokeModelWithResponseStream API reference.

QUESTION NO: 2

A hotel company wants to enhance a legacy Java-based property management system (PMS) by adding AI capabilities. The company wants to use Amazon Bedrock Knowledge Bases to provide staff with room availability information and hotel-specific details. The solution must maintain separate access controls for each hotel that the company manages. The solution must provide room availability information in near real time and must maintain consistent performance during peak usage periods.

Which solution will meet these requirements?

A. Deploy a single Amazon Bedrock knowledge base that contains combined data for all hotels. Configure AWS Lambda functions to synchronize data from each hotelâ€™s PMS database through direct API connections. Implement AWS CloudTrail logging with hotel-specific filters to audit access logs for each hotelâ€™s data.

B. Create an Amazon EventBridge rule for each hotel that is invoked by changes to the PMS database. Configure the rule to send updates to a centralized Amazon Bedrock knowledge base in a management AWS account. Configure resource-based policies to enforce hotel-specific access controls.

C. Implement one Amazon Bedrock knowledge base for each hotel in a multi-account structure. Use direct data ingestion to provide near real-time room availability information. Schedule regular synchronization for less critical information.

D. Build a centralized Amazon Bedrock Agents solution that uses multiple knowledge bases. Implement AWS IAM Identity Center with hotel-specific permission sets to control staff access.

ANSWER: C

Explanation:

Implementing one Amazon Bedrock knowledge base for each hotel in a multi-account structure provides a clear isolation boundary for both hotel data and the AWS identities authorized to access it. Each hotel account can use its own IAM roles and policies to control access to its individual knowledge base and underlying data sources, supporting independent administration and preventing one hotel's staff permissions from applying to another hotel's data. AWS documents that accounts are a fundamental isolation boundary and recommends using multiple accounts to separate workloads and resources where appropriate.

For room availability, direct ingestion lets the application submit updated documents to the applicable knowledge base as PMS availability changes, rather than waiting for a broad periodic refresh of all content. Less dynamic hotel information can use scheduled synchronization, reducing unnecessary ingestion activity while retaining current reference data. Amazon Bedrock Knowledge Bases manages the ingestion, indexing, and retrieval workflow, so the solution does not require the company to operate retrieval infrastructure or manually scale vector-search capacity during busy periods. Keeping each hotel's retrieval workload in its own knowledge base also provides an operationally clean way to scale and manage hotels independently. See the AWS guidance on direct ingestion into Knowledge Bases and using multiple AWS accounts for workload isolation.

QUESTION NO: 3

A financial services company is developing a Retrieval Augmented Generation (RAG) application to help investment analysts query complex financial relationships across multiple investment vehicles, market sectors, and regulatory environments. The dataset contains highly interconnected entities that have multi-hop relationships. Analysts must examine relationships holistically to provide accurate investment guidance. The application must deliver comprehensive answers that capture indirect relationships between financial entities and must respond in less than 3 seconds.

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

A. Use Amazon Bedrock Knowledge Bases with GraphRAG and Amazon Neptune Analytics to store financial data. Analyze multi-hop relationships between entities and automatically identify related information across documents.

B. Use Amazon Bedrock Knowledge Bases and an Amazon OpenSearch Service vector store to implement custom relationship identification logic that uses AWS Lambda to query multiple vector embeddings in sequence.

C. Use Amazon OpenSearch Serverless vector search with k-nearest neighbor (k-NN). Implement manual relationship mapping in an application layer that runs on Amazon EC2 Auto Scaling.

D. Use Amazon DynamoDB to store financial data in a custom indexing system. Use AWS Lambda to query relevant records. Use Amazon SageMaker to generate responses.

ANSWER: A

Explanation:

Use Amazon Bedrock Knowledge Bases with GraphRAG and Amazon Neptune Analytics to store financial data. Analyze multi-hop relationships between entities and automatically identify related information across documents. This approach is designed for retrieval problems in which relationships among entities are as important as similarity between text chunks. GraphRAG enriches conventional RAG retrieval by using a knowledge graph that represents entities and their connections, allowing retrieval to incorporate indirect and multi-hop relationships. This is especially appropriate for financial data, where an analyst may need to connect an investment vehicle to its holdings, sectors, counterparties, and applicable regulatory context before forming a complete answer.

Amazon Bedrock Knowledge Bases provides the managed ingestion, retrieval, and foundation-model augmentation workflow, while Neptune Analytics is purpose-built for fast graph analytics and traversal at scale. The managed GraphRAG

integration avoids the operational work of building, maintaining, and orchestrating a custom graph-extraction pipeline, sequential vector searches, and application-side relationship traversal. Neptune Analytics is optimized for low-latency graph queries, helping the application retrieve relevant connected context within the required response time when combined with an efficient Bedrock generation workflow. AWS documents GraphRAG support in Knowledge Bases and its Neptune Analytics integration at Amazon Bedrock Knowledge Bases GraphRAG and describes graph analytics capabilities at What is Amazon Neptune Analytics?

QUESTION NO: 4

A GenAI developer is evaluating Amazon Bedrock foundation models (FMs) to enhance a Europe-based company ' s internal business application. The company has a multi-account landing zone in AWS Control Tower. The company uses Service Control Policies (SCPs) to allow its accounts to use only the eu-north-1 and eu-west-1 Regions. All customer data must remain in private networks within the approved AWS Regions.

The GenAI developer selects an FM based on analysis and testing and hosts the model in the eu-central-1 Region and the eu-west-3 Region. The GenAI developer must enable access to the FM for the company ' s employees. The GenAI developer must ensure that requests to the FM are private and remain within the same Regions as the FM.

Which solution will meet these requirements?

A. Deploy an AWS Lambda function that is exposed by a private Amazon API Gateway REST API to a VPC in eu-north-1. Create a VPC endpoint for the selected FM in eu-central-1 and eu-west-3. Extend existing SCPs to allow employees to use the FM. Integrate the REST API with the business application.

B. Deploy the FM on Amazon EC2 instances in eu-north-1. Deploy a private Amazon API Gateway REST API in front of the EC2 instances. Configure an Amazon Bedrock VPC endpoint. Integrate the REST API with the business application.

C. Configure the FM to use cross-Region inference through a Europe-scoped endpoint. Configure an Amazon Bedrock VPC endpoint. Extend existing SCPs to allow employees to use the FM through inference profiles in Europe-based Regions where the FM is available. Use an inference profile to integrate Amazon Bedrock with the business application.

D. Deploy the FM in Amazon SageMaker in eu-north-1. Configure a SageMaker VPC endpoint. Extend existing SCPs to allow employees to use the SageMaker endpoint. Integrate the FM in SageMaker with the business application.

ANSWER: C

Explanation:

Configure the FM to use cross-Region inference through a Europe-scoped endpoint. Configure an Amazon Bedrock VPC endpoint. Extend existing SCPs to allow employees to use the FM through inference profiles in Europe-based Regions where the FM is available. Use an inference profile to integrate Amazon Bedrock with the business application.

Amazon Bedrock inference profiles support cross-Region inference while applying a geography-level routing boundary. A Europe-scoped inference profile routes eligible requests only to supported AWS Regions in Europe, allowing the application to use the selected model capacity in the European geography rather than routing inference to a non-European Region. The organization must update its SCP region guardrails to permit the European Regions that can participate in that inference profile; SCPs otherwise prevent the required Bedrock requests.

An interface VPC endpoint for Amazon Bedrock provides private connectivity from the application VPC to Bedrock Runtime using AWS PrivateLink. The application can invoke the model by specifying the inference profile, without requiring public internet access or operating custom routing infrastructure. This combines private application-to-Bedrock connectivity with the inference profile’s Europe-only geographic scope and provides a managed, resilient integration pattern. AWS documents cross-Region inference profiles and their geographic routing behavior in the Amazon Bedrock User Guide, and documents private Bedrock access through interface VPC endpoints.

QUESTION NO: 5

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:

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. A managed prompt can define variables that are supplied at runtime, making the four application inputs explicit and reusable. Prompt Management also lets the company define the prompt configuration for the selected model and include instructions that establish the required response structure. Associating this prompt with the flow's prompt node applies the standardized prompt configuration whenever the flow invokes Amazon Nova Pro, supporting consistently formatted grant-application responses. Amazon Bedrock documents prompt variables and the use of managed prompts in flows at Amazon Bedrock Prompt Management.

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. The insults filter is the relevant configurable content filter for bullying or insulting language. Using a detect action permits Bedrock to identify and report the intervention in the response metadata or trace while allowing the generated response to continue rather than automatically blocking it. The application can use that detection result to trigger its required notification or review workflow. Guardrails content-filter behavior and actions are described in Use content filters in Amazon Bedrock Guardrails.

QUESTION NO: 6

A medical company is creating a generative AI (GenAI) system by using Amazon Bedrock. The system processes data from various sources and must maintain end-to-end data lineage. The system must also use real-time personally identifiable information (PII) filtering and audit trails to automatically report compliance.

Which solution will meet these requirements?

A. Use AWS Glue Data Catalog to register all data sources and track lineage. Use Amazon Bedrock Guardrails PII filters. Enable AWS CloudTrail logging for all Amazon Bedrock API calls with Amazon S3 integration. Use Amazon Macie to scan stored data for sensitive information and publish findings to Amazon CloudWatch Logs. Create CloudWatch dashboards to visualize the findings and generate automated compliance reports.

B. Use AWS Config to track data source configurations and changes. Use AWS WAF with custom rules to filter PII at the application layer before Amazon Bedrock processes the data. Configure Amazon EventBridge to capture and route audit events to Amazon S3. Use Amazon Comprehend Medical with scheduled AWS Lambda functions to analyze stored outputs for compliance violations.

C. Use AWS DataSync to replicate data sources to track lineage. Configure Amazon Macie to scan Amazon Bedrock outputs for sensitive information. Use AWS Systems Manager Session Manager to log user interactions. Deploy Amazon Textract with AWS Step Functions workflows to identify and redact PII from generated reports.

D. Configure Amazon Athena to query data sources to analyze and report on data lineage. Use Amazon CloudWatch custom metrics to monitor PII exposure in Amazon Bedrock responses and establish AWS X-Ray tracing to generate an audit trail.

Use an Amazon Rekognition Custom Labels model to detect sensitive information in the data that Amazon Bedrock processes.

ANSWER: A

Explanation:

Use AWS Glue Data Catalog to register all data sources and track lineage. Use Amazon Bedrock Guardrails PII filters. Enable AWS CloudTrail logging for all Amazon Bedrock API calls with Amazon S3 integration. Use Amazon Macie to scan stored data for sensitive information and publish findings to Amazon CloudWatch Logs. Create CloudWatch dashboards to visualize the findings and generate automated compliance reports. This combines controls that operate at the appropriate stages of the GenAI data lifecycle. AWS Glue Data Catalog centralizes technical metadata for datasets from the system's sources and supports lineage visibility for data processed through AWS Glue, allowing the company to establish traceability from ingestion through downstream processing. Amazon Bedrock Guardrails can apply sensitive-information filters during model inference, including detecting and masking or blocking configured PII types before generated content is returned. This provides the required real-time enforcement rather than relying solely on a later scan of stored results.

CloudTrail records Amazon Bedrock API activity and can deliver logs to an S3 bucket for durable, queryable audit retention. Macie complements these inline protections by discovering and classifying sensitive data in supported S3 storage locations. Macie findings can be routed through Amazon EventBridge to CloudWatch Logs and dashboards, where the organization can monitor findings and drive reporting workflows. Together, the controls provide data traceability, runtime PII safeguards, and retained evidence for compliance reporting. See Amazon Bedrock sensitive information filters and AWS CloudTrail logging.

QUESTION NO: 7

A financial services company wants to develop an Amazon Bedrock application that gives analysts the ability to query quarterly earnings reports and financial statements. The financial docu ments are typically 5–100 pages long and contain both tabular data and text. The application must provide contextually accurate responses that preserve the relationship between financial metrics and their explanatory text. To support accurate and scalable retrieval, the application must incorporate document segmentation and context management strategies.

Which solution will meet these requirements?

A. Use a direct model invocation approach that uses Anthropic Claude to process each financial document as a single input. Use fine-tuned prompts that instruct the model to parse tables and text separately.

B. Use Amazon Bedrock Knowledge Bases to create a Retrieval Augmented Generation (RAG) application that retrieves relevant information from contextually chunked sections of financial documents. Segment documents based on their structural layout. Include citations that reference the original source materials.

C. Deploy an Amazon Bedrock agent that has an action group that calls custom AWS Lambda functions to analyze financial documents. Configure the Lambda functions to perform fixed-size chunking when a user submits a query about financial metrics.

D. Create one specialized Amazon Bedrock application that is optimized for structured data. Create a second application that is optimized for unstructured data. Configure each application to use a tailored chunking strategy that is suited to the application ' s content type. Implement logic to link queries to the appropriate sources.

ANSWER: B

Explanation:

Use Amazon Bedrock Knowledge Bases to create a Retrieval Augmented Generation (RAG) application that retrieves relevant information from contextually chunked sections of financial documents. Segment documents based on their structural layout. Include citations that reference the original source materials. This approach is appropriate because a knowledge base provides a managed RAG workflow: it ingests source documents, creates embeddings, stores and retrieves relevant chunks, and supplies the retrieved context to a foundation model for a grounded response. For lengthy quarterly reports, retrieval avoids placing an entire document into the model prompt and instead selects the material most relevant to the analyst's question.

Chunking based on document structure is especially valuable for financial reports, where tables, headings, notes, and adjacent narrative collectively establish the meaning of a metric. Preserving useful context in each retrievable segment helps ensure that an answer includes the numeric figure together with qualifying language, time period, units, and related explanation. Amazon Bedrock Knowledge Bases supports configurable chunking approaches and custom processing during ingestion, allowing the retrieval design to be adapted to the document format and context requirements. Knowledge Bases also supports returning source references, enabling citations that let analysts verify generated answers against the original financial materials. See the Amazon Bedrock Knowledge Bases User Guide and chunking configuration documentation.

QUESTION NO: 8

A bank is building a generative AI (GenAI) application that uses Amazon Bedrock to assess loan applications by using scanned financial documents. The application must extract structured data from the documents. The application must redact personally identifiable information (PII) before inference. The application must use foundation models (FMs) to generate approvals. The application must route low-confidence document extraction results to human reviewers who are within the same AWS Region as the loan applicant.

The company must ensure that the application complies with strict Regional data residency and auditability requirements. The application must be able to scale to handle 25,000 applications each day and provide 99.9% availability.

Which combination of solutions will meet these requirements? (Select THREE.)

A. Deploy Amazon Textract and Amazon Augmented AI within the same Region to extract relevant data from the scanned documents. Route low-confidence pages to human reviewers.

B. Use AWS Lambda functions with Amazon Comprehend to detect and redact PII from extracted document text before inference. Apply Amazon Bedrock Guardrails to protect model interactions, and use IAM policies scoped to Regional resources to control access.

C. Use Amazon Kendra and Amazon OpenSearch Service to extract field-level values semantically from the uploaded documents before inference.

D. Store uploaded documents in Amazon S3 buckets in the same Region as each applicant. Apply object metadata and tags for audit context, and enable AWS CloudTrail S3 data events for object-level audit records.

E. Use AWS Glue Data Quality to validate the structured document data. Use AWS Step Functions to orchestrate a review workflow that includes a prompt engineering step that transforms validated data into optimized prompts before invoking Amazon Bedrock to assess loan applications.

F. Use Amazon SageMaker Clarify to generate fairness and bias reports based on model scoring decisions that Amazon Bedrock makes.

ANSWER: A B D

Explanation:

Deploying Amazon Textract and Amazon Augmented AI within the same Region provides the managed document-processing and human-in-the-loop capabilities required for this workload. Textract extracts structured content, including forms and tables, from scanned financial documents and returns confidence information. An Augmented AI flow can use confidence thresholds to send results that require review to a human workforce configured in the applicant's applicable Region. This keeps the extraction and review process aligned with Regional residency controls. See the Amazon Textract and Amazon Augmented AI documentation.

Using Lambda to orchestrate Amazon Comprehend PII detection and redaction ensures that extracted text is sanitized before it is included in a prompt to Amazon Bedrock. Bedrock Guardrails can apply sensitive-information controls to model interactions, while IAM policies scoped to Regional resource ARNs and requested Regions restrict access and invocation paths. Amazon Comprehend supports identifying PII entities in text, as described in the Amazon Comprehend PII documentation.

Storing each applicant's documents in the appropriate Regional Amazon S3 bucket supports residency requirements. S3 object tags and metadata can associate records with residency and case attributes, and CloudTrail S3 data events provide an auditable record of object activity. These managed, serverless services can scale for the stated daily volume and are designed for highly available architectures.

QUESTION NO: 9

A company uses an application to process customer support tickets. The company wants to integrate AI-powered sentiment analysis and auto-response generation into the application by using Amazon Bedrock. The company wants to prioritize urgent issues and reduce initial response times by 40% compared to manual responses. The solution must process 100 concurrent webhook requests with response times under 500 ms. The solution must maintain 99.9% availability across multiple AWS Regions and authenticate all incoming requests. The company must avoid any authentication failures. The company does not want to modify the existing application infrastructure, which includes several ticketing systems that use multiple webhook authentication methods. The solution must support scaling to handle occasional spikes up to 250,000 daily tickets during peak periods. Which solution will meet these requirements?

A. Use an Amazon API Gateway REST API with a Regional endpoint to receive webhook requests and invoke AWS Lambda functions. Configure Lambda authorizers to validate all the webhook authentication methods. Configure the Lambda functions to call Amazon Bedrock to perform sentiment analysis and generate responses. Store results in Amazon DynamoDB global tables to provide multi-Region availability.

B. Create AWS Lambda function URLs for each ticketing system. Configure the function URLs with the NONE authentication type. Configure separate Lambda functions to verify webhook signatures by using Hash-based Message Authentication Code (HMAC) validation in the function code. Deploy the functions to multiple Regions and use AWS Global Accelerator to route traffic. Use Amazon Bedrock to perform sentiment analysis and generate responses. Return

C. Set up an Amazon SQS queue in each Region to receive webhook messages. Use the SQS queue to invoke AWS Lambda functions that call Amazon Comprehend to perform sentiment analysis and Amazon Lex to generate responses. Use Amazon EventBridge to retry message delivery to the application API.

D. Deploy an AWS AppSync GraphQL API to multiple Regions. Configure API tokens to authenticate incoming requests. Create GraphQL mutation resolvers that publish events to Amazon EventBridge. Configure EventBridge rules to invoke AWS Lambda functions that use Amazon Bedrock to perform sentiment analysis and generate responses. Use Amazon CloudFront to reduce latency.

E. Deploy Amazon API Gateway Regional REST APIs and AWS Lambda functions in at least two AWS Regions. Use Lambda authorizers to validate each existing webhook authentication method, and use Amazon Route 53 health checks and failover routing to send requests to a healthy Region. Have Lambda invoke Amazon Bedrock for sentiment analysis and response generation, and store results in Amazon DynamoDB global tables.

ANSWER: E

Explanation:

Deploying Regional Amazon API Gateway REST APIs and Lambda functions in at least two Regions, with Amazon Route 53 health-check-based routing, provides an end-to-end multi-Region architecture rather than only replicating the data layer. Route 53 can direct webhook traffic to a healthy Regional API endpoint and fail over when an endpoint is unhealthy, supporting the availability objective without requiring changes to the existing ticketing systems. API Gateway Lambda authorizers provide a centralized extensibility point for validating the different existing webhook authentication schemes, such as tokens and HMAC signatures, before requests reach the ticket-processing Lambda function. This preserves the existing webhook integrations while ensuring every request is authenticated.

Lambda scales automatically to concurrent request demand and can invoke Amazon Bedrock for sentiment classification and generated initial responses. DynamoDB global tables replicate ticket results across the participating Regions, so either Regional processing stack can access current data with low latency. API Gateway, Lambda, and direct synchronous Bedrock invocation avoid inserting an asynchronous queue into the immediate webhook response path, helping the architecture meet the response-time target when an appropriately low-latency Bedrock model and prompt are selected. See the API Gateway Lambda authorizer documentation and the Route 53 DNS failover documentation.

QUESTION NO: 10

An ecommerce company is using an Anthropic Claude Sonnet model in Amazon Bedrock to generate product recommendations. An AWS Lambda function retrieves customer purchase data from Amazon DynamoDB, product reviews from Amazon S3, and customer profile information from Amazon RDS. Then the function sends the data directly to the Amazon Bedrock model through API calls. Recently, customers who have extensive purchase histories have begun to receive incomplete recommendations.

Amazon CloudWatch logs for the Lambda function show execution timeouts. CloudWatch logs for Amazon Bedrock API calls show intermittent errors. The company reviews the logs and finds that some requests are failing with context-length-exceeded errors. Other requests finish but appear to ignore portions of the input data.

The company wants the recommendation system to consider all customer data when the system generates recommendations. The company wants to use Amazon Bedrock Knowledge Bases to improve data organization and retrieval.

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

A. Implement a chunking strategy that divides the customer data into smaller segments. Configure the model to process each segment separately. Invoke the model a final time to synthesize the individual responses into comprehensive recommendations.

B. Modify the prompt structure to place the most critical information at the beginning and end of the context window. Implement token-counting logic to truncate less important data when the interaction approaches the model's maximum context length.

C. Replace Claude Sonnet with a model that has a larger context-window capacity. Increase the Lambda function timeout to accommodate longer processing times for larger inputs.

D. Configure the recommendation system to use the Converse API. Modify the additionalModelRequestFields parameter to increase the maximum token limit beyond the model's default context-window size.

E. Implement RAG by using a knowledge base to index the customer data with vector embeddings. Retrieve only the most semantically relevant information for each recommendation request based on the current customer context.

ANSWER: A E

Explanation:

Implementing a chunking strategy that divides the customer data into smaller segments, processes each segment separately, and then invokes the model for final synthesis addresses the context-window constraint without discarding historical information. Each segment can produce an intermediate analysis of purchases, profile attributes, or review signals. A final synthesis step then combines those intermediate results into a complete recommendation, allowing the workflow to account for the full set of customer data while keeping every model invocation within supported input limits. This also makes Lambda processing more predictable because oversized one-shot prompts are avoided.

Implementing RAG by using a knowledge base to index the customer data with vector embeddings provides the required organization and retrieval layer. Amazon Bedrock Knowledge Bases ingests source content, creates embeddings, and retrieves context that is semantically relevant to the current request. For recommendation generation, the application can use the customer's current context to retrieve the most pertinent purchase-history, profile, and review information, rather than repeatedly placing unstructured raw records into the prompt. Knowledge Bases supports configurable chunking during ingestion, which complements staged processing and helps ensure retrieved content is manageable and grounded in indexed data. See the Amazon Bedrock Knowledge Bases documentation and chunking configuration guidance.

QUESTION NO: 11

A company is designing a solution that uses foundation models (FMs) to support multiple AI workloads. Some FMs must be invoked on demand and in real time. Other FMs require consistent high-throughput access for batch processing.

The solution must support hybrid deployment patterns and run workloads across cloud infrastructure and on-premises infrastructure to comply with data residency and compliance requirements.

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

A. Use AWS Lambda to orchestrate low-latency FM inference by invoking FMs hosted on Amazon SageMaker AI asynchronous endpoints.

B. Configure provisioned throughput in Amazon Bedrock to ensure consistent performance for high-volume workloads.

C. Deploy FMs to Amazon SageMaker AI endpoints with support for edge deployment by using Amazon SageMaker Neo. Orchestrate the FMs by using AWS Lambda to support hybrid deployment.

D. Use Amazon Bedrock with auto-scaling to handle unpredictable traffic surges.

E. Use Amazon SageMaker JumpStart to host and invoke the FMs.

ANSWER: B C

Explanation:

“Configure provisioned throughput in Amazon Bedrock to ensure consistent performance for high-volume workloads” meets the need for dependable capacity for sustained batch inference. Amazon Bedrock Provisioned Throughput reserves model inference capacity for a selected foundation model, providing the predictable throughput needed when a workload cannot depend solely on shared, on-demand capacity. The same Bedrock solution can use on-demand model invocation for the workloads that require immediate, real-time responses, while provisioned capacity serves the high-volume processing requirement. AWS describes the available commitment and capacity choices in its Amazon Bedrock Provisioned Throughput documentation.

“Deploy FMs to Amazon SageMaker AI endpoints with support for edge deployment by using Amazon SageMaker Neo. Orchestrate the FMs by using AWS Lambda to support hybrid deployment” provides the hybrid component. SageMaker AI endpoints can host models in AWS for cloud-based inference, while SageMaker Neo can compile supported models for target hardware so inference artifacts can be deployed efficiently on edge or on-premises systems. Lambda can coordinate invocation flows and routing between the cloud-hosted and locally deployed workloads. This pattern allows workloads subject to residency or compliance constraints to execute outside the AWS Region while retaining AWS-based model-development and deployment tooling. See the Amazon SageMaker Neo Developer Guide for Neo compilation and edge deployment support.

QUESTION NO: 12

A company is building an Amazon Bedrock agent that helps procurement employees check the status of purchase orders. Purchase order status is stored in an internal system that is accessible only through an existing REST API. The agent must obtain an order number from the user, invoke the API, and use the returned status to answer the user.

The company wants to use managed agent capabilities and does not want to place purchase order data into a knowledge base.

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

A. Create an Amazon Bedrock agent action group that uses an OpenAPI schema for the purchase order REST API.

B. Configure an AWS Lambda function as the action group's executor to invoke the internal REST API.

C. Ingest purchase order records into an Amazon Bedrock knowledge base and synchronize the data source every 24 hours.

D. Store the OpenAPI schema as a prompt variable in Amazon Bedrock Prompt Management.

E. Configure Amazon Bedrock Guardrails custom word filters to retrieve the purchase order status.

ANSWER: A B

Explanation:

Create an Amazon Bedrock agent action group that uses an OpenAPI schema for the purchase order REST API. The schema defines the API operations, request parameters, and response structure that the agent can use when it decides to retrieve an order status. Required parameters in the schema allow the agent to request an order number when the user has not supplied one.

Configure an AWS Lambda function as the action group's executor to call the internal REST API and return the API response to the agent. The Lambda function can apply the required authentication and network integration while the agent uses the returned status as grounded tool output. A knowledge base is suited to retrieving indexed reference content, not to making real-time calls to an operational purchase order system.

QUESTION NO: 13

A finance company is developing an AI assistant to help clients plan investments and manage their portfolios. The company identifies several high-risk conversation patterns such as requests for specific stock recommendations or guaranteed returns. High-risk conversation patterns could lead to regulatory violations if the company cannot implement appropriate controls.

The company must ensure that the AI assistant does not provide inappropriate financial advice, generate content about competitors, or make claims that are not factually grounded in the company ' s approved financial guidance. The company wants to use Amazon Bedrock Guardrails to implement a solution.

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

A. Add the high-risk conversation patterns to a denied topics guardrail.

B. Configure a content filter guardrail to filter prompts that contain the high-risk conversation patterns.

C. Configure a content filter guardrail to filter prompts that contain competitor names.

D. Add the names of competitors as custom word filters. Set the input and output actions to block.

E. Set a low grounding score threshold.

F. Set a high grounding score threshold.

ANSWER: A D F

Explanation:

Add the high-risk conversation patterns to a denied topics guardrail is appropriate because denied topics use natural-language topic definitions and examples to identify and block subject areas that the organization does not want the model to address. The company can define requests for individualized stock recommendations, guaranteed investment performance, and related regulated-advice scenarios as prohibited topics. This applies a policy-level control aligned to the identified compliance risk.

Add the names of competitors as custom word filters. Set the input and output actions to block provides an explicit control for competitor references. Amazon Bedrock Guardrails custom word filters support matching specified words and phrases, and the filters can be configured to evaluate both user input and model output. Blocking in both directions prevents requests designed to elicit competitor content and prevents the assistant from producing that content.

Set a high grounding score threshold enforces a stricter standard for outputs evaluated by the contextual grounding policy. When the assistant is supplied with the company’s approved financial-guidance content as grounding material, a higher threshold requires a stronger relationship between the response and that reference material before the response is allowed. This helps ensure factual claims remain supported by the approved source. See the Amazon Bedrock Guardrails User Guide and contextual grounding policy documentation.

QUESTION NO: 14

A university is building an AI-powered application that includes several sub-applications. The sub-applications include AI assistants, assignment graders, and internal analytics applications. The university is defining and testing multiple prompts by using various foundation models (FMs). The university wants to compare variants of each prompt and choose the variant that yield outputs that are best-suited for specified use cases. The university requires a version control solution for the prompts. The university must be able to test prompt variations and collect audit trails for prompt changes and usage. The solution must also maintain consistency while allowing the prompts to integrate into the main application. Which combination of solutions will meet these requirements with the LEAST operational overhead? (Select TWO.)

A. Use Amazon Bedrock Prompt Management to create versioned prompts. Include parameterized variables for each use case.

B. Store prompts in Amazon S3. Use AWS Step Functions to orchestrate the model interactions and service integrations.

C. Use Amazon Bedrock Flows to create workflows that combine FMs and AWS services.

D. Configure AWS Config to record prompt changes. Use AWS CloudTrail to track prompt usage.

E. Configure Amazon Bedrock intelligent prompt routing.

ANSWER: A C

Explanation:

Use Amazon Bedrock Prompt Management to create versioned prompts. Include parameterized variables for each use case. Amazon Bedrock Prompt Management is a managed prompt-lifecycle capability that supports creating prompts with multiple variants, evaluating and comparing those variants, and creating immutable prompt versions for use in production. Prompt versions provide a controlled, repeatable prompt artifact, while variables let the university retain a common prompt template and inject use-case-specific values for assistants, grading, and analytics. This promotes consistent behavior without maintaining separate copies of nearly identical prompts. Bedrock API activity, including prompt-management actions and runtime usage, can be recorded through AWS CloudTrail, supporting the required auditability with AWS-managed service integration. See the Amazon Bedrock Prompt Management documentation.

Use Amazon Bedrock Flows to create workflows that combine FMs and AWS services. Flows provide a managed visual orchestration layer for connecting foundation model invocations, prompt assets, knowledge bases, Lambda functions, and other flow nodes. A flow can invoke the approved, versioned prompt as part of the university's main application workflow, which makes prompt use more consistent across sub-applications and reduces the custom orchestration code and operational management otherwise required. Flow definitions can also be versioned and prepared for deployment, supporting controlled integration of the tested prompt assets. See the Amazon Bedrock Flows documentation.

QUESTION NO: 15

A legal services company is building an Amazon Bedrock knowledge base for internal research. The knowledge base contains documents for multiple practice groups. Some documents are restricted to a single practice group, while other documents are shared across groups.

The company has a trusted application that authenticates users and determines each user’s practice group before calling the Amazon Bedrock RetrieveAndGenerate API. The company needs to prevent retrieval of restricted documents for unauthorized users while retaining one shared knowledge base.

Which solution will meet these requirements?

A. Attach practice-group metadata to each document and include a retrieval metadata filter that matches the authenticated user’s practice group.

B. Add the authenticated user’s practice group to the system prompt and instruct the model not to mention unauthorized documents.

C. Create a separate foundation model for each practice group and invoke the appropriate model for each user.

D. Use Amazon Bedrock Guardrails denied topics to block references to the names of restricted practice groups.

ANSWER: A

Explanation:

Associate document-level metadata that identifies the authorized practice group with each source document, and include a metadata filter in every retrieval request that matches the authenticated user’s practice group. Amazon Bedrock Knowledge Bases supports metadata-based filtering during retrieval. The filter limits the candidate documents before the generated response is composed, which prevents the retrieval workflow from supplying unauthorized restricted content to the model.

Adding the practice group to the prompt is not an authorization control because the model could still receive restricted retrieved content. Creating a separate foundation model for each group does not enforce document-level retrieval permissions and adds unnecessary operational complexity. The application must derive the filter from trusted authorization data rather than from an unverified user-provided value.