Question #1 *Topic 1*

A retail company has a generative AI (GenAI) product recommendation application that uses Amazon Bedrock. The application suggests products to customers based on browsing history and demographics. The company needs to implement fairness evaluation across multiple demographic groups to detect and measure bias in recommendations between two prompt approaches. The company wants to collect and monitor fairness metrics in real time. The company must receive an alert if the fairness metrics show a discrepancy of more than 15% between demographic groups. The company must receive weekly reports that compare the performance of the two prompt approaches.
Which solution will meet these requirements with the LEAST custom development effort?

A. Configure an Amazon CloudWatch dashboard to display default metrics from Amazon Bedrock API calls. Create custom metrics based on model outputs. Set up Amazon EventBridge rules to invoke AWS lambda functions that perform post-processing analysis on model responses and publish custom fairness metrics.

B. Create the two prompt variants in Amazon Bedrock Prompt Management. Use Amazon Bedrock Flows to deploy the prompt variants with defined traffic allocation. Configure Amazon Bedrock guardrails that have content filters to monitor demographic fairness. Set up Amazon CloudWatch alarms on the GuardrailContentSource dimension that use InvocationsIntervened metrics to detect recommendation discrepancy threshold violations.

C. Set up Amazon SageMaker Clarify to analyze model outputs. Publish fairness metrics to Amazon CloudWatch. Create CloudWatch composite alarms that combine SageMaker Clarify bias metrics with Amazon Bedrock latency metrics to provide a comprehensive fairness evaluation dashboard.

D. Create an Amazon Bedrock model evaluation job to compare fairness between the two prompt variants. Enable model invocation logging in Amazon CloudWatch. Set up CloudWatch alarms for InvocationsIntervened metrics with a dimension for each demographic group.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #2 *Topic 1*

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? (Choose 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.

Suggested Answer: *ADF*

Currently there are no comments in this discussion, be the first to comment!

Question #3 *Topic 1*

A company has deployed an AI assistant as a React application that uses AWS Amplify, an AWS AppSync GraphQL API, and Amazon Bedrock Knowledge Bases. The application uses the GraphQL API to call the Amazon Bedrock RetrieveAndGenerate API for knowledge base interactions. The company configures an AWS Lambda resolver to use the RequestResponse invocation type.
Application users report frequent timeouts and slow response times. Users report these problems more frequently for complex questions that require longer processing.
The company needs a solution to fix these performance issues and enhance the user experience.
Which solution will meet these requirements?

A. Use AWS Amplify AI Kit to implement streaming responses from the GraphQL API and to optimize client-side rendering.

B. Increase the timeout value of the Lambda resolver. Implement retry logic with exponential backoff.

C. Update the application to send an API request to an Amazon SQS queue. Update the AWS AppSync resolver to poll and process the queue.

D. Change the RetrieveAndGenerate API to the InvokeModelWithResponseStream API. Update the application to use an Amazon API Gateway WebSocket API to support the streaming response.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #4 *Topic 1*

An ecommerce company operates a global product recommendation system that needs to switch between multiple foundation models (FM) in Amazon Bedrock based on regulations, cost optimization, and performance requirements. The company must apply custom controls based on proprietary business logic, including dynamic cost thresholds, AWS Region-specific compliance rules, and real-time A/B testing across multiple FMs. The system must be able to switch between FMs without deploying new code. The system must route user requests based on complex rules including user tier, transaction value, regulatory zone, and real-time cost metrics that change hourly and require immediate propagation across thousands of concurrent requests.
Which solution will meet these requirements?

A. Deploy an AWS Lambda function that uses environment variables to store routing rules and Amazon Bedrock FM IDs. Use the Lambda console to update the environment variables when business requirements change. Configure an Amazon API Gateway REST API to read request parameters to make routing decisions.

B. Deploy Amazon API Gateway REST API request transformation templates to implement routing logic based on request attributes. Store Amazon Bedrock FM endpoints as REST API stage variables. Update the variables when the system switches between models.

C. Configure an AWS Lambda function to fetch routing configurations from the AWS AppConfig Agent for each user request. Run business logic in the Lambda function to select the appropriate FM for each request. Expose the FM through a single Amazon API Gateway REST API endpoint.

D. Use AWS Lambda authorizers for an Amazon API Gateway REST API to evaluate routing rules that are stored in AWS AppConfig. Return authorization contexts based on business logic. Route requests to model-specific Lambda functions for each Amazon Bedrock FM.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #5 *Topic 1*

A company is developing an internal generative AI (GenAI) assistant that uses Amazon Bedrock to summarize corporate documents for multiple business units. The GenAI assistant must generate responses in a consistent format that includes a document summary, classification of business risks, and terms that are flagged for review. The GenAI assistant must adapt the tone of responses for each user's business unit, such as legal, human resources, or finance. The GenAI assistant must block hate speech, inappropriate topics, and sensitive information such as personal health information.

The company needs a solution to centrally manage prompt variants across business units and teams. The company wants to minimize ongoing orchestration efforts and maintenance for post-processing logic. The company also wants to have the ability to adjust content moderation criteria for the GenAI assistant over time.

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

A. Use Amazon Bedrock Prompt Management to configure reusable templates and business unit-specific prompt variants. Apply Amazon Bedrock guardrails that have category filters and sensitive term lists to block prohibited content.

B. Use Amazon Bedrock Prompt Management to define base templates. Enforce business unit-specific tone by using system prompt variables. Configure Amazon Bedrock guardrails to apply audience-based threshold tuning. Manage the guardrails by using an internal administration API.

C. Use Amazon Bedrock with business unit-based instruction injection in API calls. Store response formatting rules in Amazon DynamoDB. Use AWS Step functions to validate responses. Use Amazon Comprehend to apply content filters after the GenAI assistant generates responses.

D. Use Amazon Bedrock with custom prompt templates that are stored in Amazon DynamoDB. Create one AWS Lambda function to select business unit-specific prompts. Create a second Lambda function to call Amazon Comprehend to filter prohibited content from responses.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #6 *Topic 1*

A financial services company is building a customer support application that retrieves relevant financial regulation documents from a database based on semantic similarities to user queries. The application must integrate with Amazon Bedrock to generate responses. The application must be able to search documents that are in English, Spanish, and Portuguese. The application must filter documents by metadata such as publication date, regulatory agency, and document type.
The database stores approximately 10 million document embeddings. To minimize operational overhead, the company wants a solution that minimizes management and maintenance effort. The application must provide low-latency responses for real-time customer interactions.
Which solution will meet these requirements?

A. Use Amazon OpenSearch Serverless to provide vector search capabilities and metadata filtering. Connect to Amazon Bedrock Knowledge Bases to enable Retrieval Augmented Generation (RAG) capabilities that use an Anthropic Claude foundation model (FM).

B. Deploy an Amazon Aurora PostgreSQL database with the pgvector extension. Define tables to store embeddings and metadata. Use SQL queries to perform similarity searches. Send retrieved documents to Amazon Bedrock to generate responses.

C. Use Amazon S3 Vectors to configure a vector index and non-filterable metadata fields. Integrate S3 Vectors with Amazon Bedrock to enable Retrieval Augmented Generation (RAG) capabilities.

D. Set up an Amazon Neptune Analytics graph database. Configure a vector index that has appropriate dimensionality to store document embeddings. Use Amazon Bedrock to perform graph-based retrieval and to generate responses.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #7 *Topic 1*

A medical company is building a generative AI (GenAI) application that uses RAG to provide evidence-based medical information. The application uses Amazon OpenSearch Service to retrieve vector embeddings. Users report that searches frequently miss results that contain exact medical terms and acronyms and return too many semantically similar but irrelevant documents. The company needs to improve retrieval quality and maintain low end user latency, even as the document collection grows to millions of documents.
Which solution will meet these requirements with the LEAST operational overhead?

A. Configure hybrid search by combining vector similarity with keyword matching to improve semantic understanding and exact term and acronym matching.

B. Increase the dimensions of the vector embeddings from 384 to 1536. Use a post-processing AWS Lambda function to filter out irrelevant results after retrieval.

C. Replace OpenSearch Service with Amazon Kendra. Use query expansion to handle medical acronyms and terminology variants during pre-processing.

D. Implement a two-stage retrieval architecture in which initial vector search results are re-ranked by an ML model that is hosted on Amazon SageMaker AI.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #8 *Topic 1*

A company runs a generative AI (GenAI)-powered summarization application in an application AWS account that uses Amazon Bedrock. The application architecture includes an Amazon API Gateway REST API that forwards requests to AWS Lambda functions that are attached to private VPC subnets. The application summarizes sensitive customer records that the company stores in a governed data lake in a centralized data storage account. The company has enabled Amazon S3, Amazon Athena, and AWS Glue in the data storage account.
The company must ensure that calls that the application makes to Amazon Bedrock use only private connectivity between the company's application VPC and Amazon Bedrock. The company's data lake must provide fine-grained column-level access across the company's AWS accounts.
Which solution will meet these requirements?

A. In the application account, create interface VPC endpoints for Amazon Bedrock runtimes. Run Lambda functions in private subnets. Use IAM conditions on inference and data-plane policies to allow calls only to approved endpoints and roles. In the data storage account, use AWS Lake Formation LF-tag-based access control to create table and column-level cross-account grants.

B. Run Lambda functions in private subnets. Configure a NAT gateway to provide access to Amazon Bedrock and the data lake. Use S3 bucket policies and ACLs to manage permissions. Export AWS CloudTrail logs to Amazon S3 to perform weekly reviews.

C. Create a gateway endpoint only for Amazon S3 in the application account. Invoke Amazon Bedrock through public endpoints. Use database-level grants in AWS Lake Formation to manage data access. Stream AWS CloudTrail logs to Amazon CloudWatch Logs. Do not set up metric filters or alarms.

D. Use VPC endpoints to provide access to Amazon Bedrock and Amazon S3 in the application account. Use only IAM path-based policies to manage data lake access. Send AWS CloudTrail logs to Amazon CloudWatch Logs. Periodically create dashboards and allow public fallback for cross-Region reads to reduce setup time.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #9 *Topic 1*

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 to create dashboards that display prompt usage metrics. Store the approval status of content 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 IAM policies to control approval permissions. Create parameterized prompt templates by specifying variables.

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

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

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #10 *Topic 1*

A company is developing a customer support application that uses Amazon Bedrock foundation models (FMs) to provide real-time AI assistance to the company's employees. The application must display AI-generated responses character by character as the responses are generated. The application needs to support thousands of concurrent users with minimal latency. The responses typically take 15 to 45 seconds to finish.
Which solution will meet these requirements?

A. Configure an Amazon API Gateway WebSocket API with an AWS Lambda integration. Configure the WebSocket API to invoke the Amazon Bedrock InvokeModelWithResponseStream API and stream partial responses through WebSocket connections.

B. Configure an Amazon API Gateway REST API with an AWS Lambda integration. Configure the REST API to invoke the Amazon Bedrock standard InvokeModel API and implement frontend client-side polling every 100 ms for complete response chunks.

C. Implement direct frontend client connections to Amazon Bedrock by using IAM user credentials and the InvokeModelWithResponseStream API without any intermediate gateway or proxy layer.

D. Configure an Amazon API Gateway HTTP API with an AWS Lambda integration. Configure the HTTP API to cache complete responses in an Amazon DynamoDB table and serve the responses through multiple paginated GET requests to frontend clients.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #11 *Topic 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? (Choose 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.

Suggested Answer: *AD*

Currently there are no comments in this discussion, be the first to comment!

Question #12 *Topic 1*

A healthcare company is using Amazon Bedrock to build a Retrieval Augmented Generation (RAG) application that helps practitioners make clinical decisions. The application must achieve high accuracy for patient information retrievals, identify hallucinations in generated content, and reduce human review costs.
Which solution will meet these requirements?

A. Use Amazon Comprehend to analyze and classify RAG responses and to extract medical entities and relationships. Use AWS Step Functions to orchestrate automated evaluations. Configure Amazon CloudWatch metrics to track entity recognition confidence scores. Configure CloudWatch to send an alert when accuracy falls below specified thresholds.

B. Implement automated large language model (LLM)-based evaluations that use a specialized model that is fine-tuned for medical content to assess all responses. Deploy AWS Lambda functions to parallelize evaluations. Publish results to Amazon CloudWatch metrics that track relevance and factual accuracy.

C. Configure Amazon CloudWatch Synthetics to generate test queries that have known answers on a regular schedule, and track model success rates. Set up dashboards that compare synthetic test results against expected outcomes.

D. Deploy a hybrid evaluation system that uses an automated LLM-as-a-judge evaluation to initially screen responses and targeted human reviews for edge cases. Use Amazon SageMaker Feature Store to maintain evaluation datasets. Use a built-in Amazon Bedrock evaluation to track retrieval precision and hallucination rates.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #13 *Topic 1*

Company configures a landing zone in AWS Control Tower. The company handles sensitive data that must remain within the European Union. The company must use only the eu-central-1 Region. The company uses SCPs to enforce data residency policies. GenAI developers at the company are assigned IAM roles that have full permissions for Amazon Bedrock.

The company must ensure that GenAI developers can use the Amazon Nova Pro model through Amazon Bedrock only by using cross-Region inference (CRI) and only in eu-central-1. The company enables model access for the GenAI developer IAM roles in Amazon Bedrock. However, when a GenAI developer attempts to invoke the model through the Amazon Bedrock Chat/Text playground, the GenAI developer receives the following error.

User: arn:aws:sts::123456789012:assumed-role/AssumedDevRole/DevUserName

Action: bedrock:InvokeModelWithResponseStream

On resource(s): arn:aws:bedrock:eu-west-3::foundation-model/amazon.nova-pro-v1:0

Context: a service control policy explicitly denies the action

The company needs a solution to resolve the error. The solution must retain the company's existing governance controls and must provide precise access control. The solution must comply with the company's existing data residency policies.

Which combination of solutions will meet these requirements? (Choose two.)

A. Add an AdministratorAccess policy to the GenAI developer IAM role.

B. Extend the existing SCPs to enable CRI for the eu.amazon.nova-pro-v1:0 inference profile.

C. Enable Amazon Bedrock model access for Amazon Nova Pro in the eu-west-3 Region.

D. Validate that the GenAI developer IAM roles have permissions to invoke Amazon Nova Pro through the eu.amazon.nova-pro.v1:0 inference profile on all European Union AWS Regions that can serve the model.

E. Extend the existing SCP to enable CRI for the eu.* inference profile.

Suggested Answer: *BD*

Currently there are no comments in this discussion, be the first to comment!

Question #14 *Topic 1*

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 together sequentially. Use conditional logic based on content classification.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #15 *Topic 1*

An ecommerce company is developing a generative AI (GenAI) solution that uses Amazon Bedrock with Anthropic Claude to recommend products to customers. Customers report that some of the recommended products are not available for sale on the website or are not relevant to the customer. Customers also report that the solutions takes a long time to generate some recommendations.
The company investigates the issues and finds that most interactions between customers and the product recommendation solution are unique. The company confirms that the solutions recommends products that are not in the company's product catalog. The company must resolve these issues.
Which solution will meet this requirement?

A. Increase grounding within Amazon Bedrock Guardrails. Enable Automated Reasoning checks. Set up provisioned throughput.

B. Use prompt engineering to restrict the model responses to relevant products. Use streaming techniques such as the InvokeModelWithResponseStream action to reduce perceived latency for the customers.

C. Create an Amazon Bedrock knowledge base. Implement Retrieval Augmented Generation (RAG). Set the PerformanceConfigLatency parameter to optimized.

D. Store product catalog data in Amazon OpenSearch Service. Validate the model's product recommendations against the product catalog. Use Amazon DynamoDB to implement response caching.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #16 *Topic 1*

A company is using AWS Lambda and REST APIs to build a reasoning agent to automate support workflows. The system must preserve memory across interactions, share the relevant agent state, and support event-driven invocation and synchronous invocation. The system must also enforce access control and session-based permissions.
Which combination of steps provides the MOST scalable solution? (Choose two.)

A. Use Amazon Bedrock AgentCore to manage memory and session-aware reasoning. Deploy the agent with built-in identity support, event handling, and observability.

B. Register the Lambda functions and the REST APIs as actions by using Amazon API Gateway and Amazon EventBridge. Enable Amazon Bedrock AgentCore to invoke the Lambda functions and the REST APIs without custom orchestration code.

C. Use Amazon Bedrock Agents for reasoning and conversation management. Use AWS Step Functions and Amazon SQS queues for orchestration. Store the agent state in Amazon DynamoDB to maintain memory between steps.

D. Deploy the reasoning logic as a container on Amazon ECS behind Amazon API Gateway. Use Amazon Aurora to store memory data and identity data.

E. Build a custom RAG pipeline by using Amazon Kendra and Amazon Bedrock. Use AWS Lambda to orchestrate tool invocations. Store the agent state in Amazon S3.

Suggested Answer: *AB*

Currently there are no comments in this discussion, be the first to comment!

Question #17 *Topic 1*

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. The analysts must be able to examine the relationships holistically to provide accurate investment guidance. The application must deliver comprehensive answers that capture indirect relationships between financial entities. The application must produce responses in less than 3 seconds.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon Bedrock Knowledge Bases with Graph RAG and Amazon Neptune Analytics to store the financial data. Analyze the 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 functions to query multiple vector embeddings in sequence.

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

D. Use Amazon DynamoDB to store financial data in a custom indexing system. Use an AWS Lambda function to query relevant records based on input questions. Use Amazon SageMaker AI to generate responses.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #18 *Topic 1*

A healthcare company uses Amazon Bedrock to deploy an application that generates summaries of clinical documents. The application experiences inconsistent response quality with occasional factual hallucinations. Monthly costs exceed the company's projections by 40%. A GenAI developer must implement a near real-time monitoring solution to detect hallucinations, identify abnormal token consumption, and provide early warnings of cost anomalies. The solution must require minimal custom development work and maintenance overhead.
Which solution will meet these requirements?

A. Configure Amazon CloudWatch alarms to monitor InputTokenCount and OutputTokenCount metrics to detect anomalies. Store model invocation logs in an Amazon S3 bucket. Use AWS Glue and Amazon Athena to identify potential hallucinations.

B. Run Amazon Bedrock evaluation jobs that use LLM-based judgments to detect hallucinations. Configure Amazon CloudWatch to track token usage. Create an AWS Lambda function to process CloudWatch metrics. Configure the Lambda function to send usage pattern notifications.

C. Configure Amazon Bedrock to store model invocation logs in an Amazon S3 bucket. Enable text output logging. Configure Amazon Bedrock guardrails to run contextual grounding checks to detect hallucinations. Create Amazon CloudWatch anomaly detection alarms for token usage metrics.

D. Use AWS CloudTrail to log all Amazon Bedrock API calls. Create a custom dashboard in Amazon QuickSight to visualize token usage patterns. Use Amazon SageMaker Model Monitor to detect quality drift in generated summaries.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #19 *Topic 1*

A company is building a generative AI (GenAI) application that produces content based on a variety of internal and external data sources. The company wants to ensure that the generated output is fully traceable. The application must support data source registration and enable metadata tagging to attribute content to its original source. The application must also maintain audit logs of data access and usage throughout the pipeline. Which solution will meet these requirements?

A. Use AWS Lake Formation to catalog data sources and control access. Apply metadata tags directly in Amazon S3. Use AWS CloudTrail to monitor API activity.

B. Use AWS Glue Data Catalog to register and tag data sources. Use Amazon CloudWatch Logs to monitor access patterns and application behavior.

C. Store data in Amazon S3 and use object tagging for attribution. Use AWS Glue Data Catalog to manage schema information. Use AWS CloudTrail to log access to S3 buckets.

D. Use AWS Glue Data Catalog to register all data sources. Apply metadata tags to attribute data sources. Use AWS CloudTrail to log access and activity across services.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #20 *Topic 1*

A financial services company needs to build a document analysis system that uses Amazon Bedrock to process quarterly reports. The system must analyze financial data, perform sentiment analysis, and validate compliance across batches of reports. Each batch contains 5 reports. Each report requires multiple foundation model (FM) calls. The solution must finish the analysis within 10 seconds for each batch. Current sequential processing takes 45 seconds for each batch.
Which solution will meet these requirements?

A. Use AWS Lambda functions with provisioned concurrency to process each analysis type sequentially. Configure the Lambda function timeouts to 10 seconds. Configure automatic retries with exponential backoff.

B. Use AWS Step Functions with a Parallel state to invoke separate AWS Lambda functions for each analysis type simultaneously. Configure Amazon Bedrock client timeouts. Use Amazon CloudWatch metrics to track execution time and model inference latency.

C. Create an Amazon SQS queue to buffer analysis requests. Deploy multiple AWS Lambda functions with reserved concurrency. Configure each Lambda function to process different aspects of each report sequentially and then combine the results.

D. Deploy an Amazon ECS cluster that runs containers that process each report sequentially. Use a load balancer to distribute batch workloads. Configure an auto-scaling policy based on CPU utilization to handle demand fluctuations.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #21 *Topic 1*

A company is using Amazon Bedrock to build a customer-facing AI assistant to handle sensitive customer inquiries. The company must use defense-in-depth safety controls to block sophisticated prompt injection attacks. The company must keep audit logs of all safety interventions. The AI assistant must have cross-Region failover capabilities.
Which solution will meet these requirements?

A. Configure Amazon Bedrock guardrails to use content filters to protect against prompt injection attacks. Set the content filters to high. Use a guardrail profile to implement cross-Region guardrail inference. Use Amazon CloudWatch Logs with custom metrics to capture detailed guardrail intervention events.

B. Configure Amazon Bedrock guardrails to use content filters to protect against prompt injection attacks. Set the content filters to high. Use AWS WAF to block suspicious inputs. Use AWS CloudTrail to log API calls for audits.

C. Deploy Amazon Comprehend custom classification to detect prompt injection attacks. Use Amazon API Gateway to validate requests. Use Amazon CloudWatch Logs with custom metrics to capture detailed intervention events.

D. Configure Amazon Bedrock guardrails to use custom content filters to protect against harmful content. Set the content filters to high. Use word filters to protect against known attack patterns. Configure cross-Region guardrail replication to provide failover capabilities. Store logs in AWS CloudTrail for compliance auditing.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #22 *Topic 1*

A company is designing a canary deployment strategy for a payment processing API. The system must support automated gradual traffic shifting between multiple Amazon Bedrock models based on real-time inference metrics, historical traffic patterns, and service health. The solution must be able to gradually increase traffic to new model versions. The system must increase traffic if metrics remain healthy and decrease traffic if the performance degrades below acceptable thresholds.
The company needs to comprehensively monitor inference latency and error rates during the deployment phase. The company must also be able to halt deployments and revert to a previous model version without any manual intervention.
Which solution will meet these requirements?

A. Use Amazon Bedrock with provisioned throughput to host the versions of the model. Configure an Amazon EventBridge rule to invoke an AWS Step Functions workflow when a new model version is released. Configure the workflow to shift traffic in stages, wait for a specified time period, and invoke an AWS Lambda function to check Amazon CloudWatch performance metrics. Configure the workflow to increase traffic if the metrics meet thresholds and to trigger a traffic rollback if performance metrics fall below thresholds.

B. Use AWS Lambda functions to invoke various Amazon Bedrock model versions. Use an Amazon API Gateway HTTP API with stage variables and weighted routing to shift traffic gradually to new model versions. Use Amazon CloudWatch to monitor performance metrics. Use external logic to adjust traffic between model versions and to roll back if performance falls below thresholds.

C. Use Amazon SageMaker AI endpoint variants to represent multiple Amazon Bedrock model versions. Use variant weights to shift traffic. Use Amazon CloudWatch to monitor performance metrics. Use SageMaker Model Monitor to trigger AWS Lambda functions to roll back a model deployment if performance drops below a specified threshold. Configure an Amazon EventBridge rule to roll back model deployments if an anomaly is detected.

D. Use Amazon OpenSearch Service to track inference logs. Configure OpenSearch Service to invoke an AWS Systems Manager Automation runbook to update Amazon Bedrock model endpoints to shift traffic based on the inference logs.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #23 *Topic 1*

A financial services company uses an AI application to process financial documents by using Amazon Bedrock. During business hours, the application handles approximately 10,000 requests each hour, which requires consistent throughput.
The company uses the CreateProvisionedModelThroughput API to purchase provisioned throughput. Amazon CloudWatch metrics show that the provisioned capacity is unused while on-demand requests are being throttled. The company finds the following code in the application: python response = bedrock_runtime.invoke_model(modelId="anthropic.claude-v2", body=json.dumps(payload))
The company needs the application to use the provisioned throughput and to resolve the throttling issues.
Which solution will meet these requirements?

A. Increase the number of model units (MUs) in the provisioned throughput configuration.

B. Replace the model ID parameter with the ARN of the provisioned model that the CreateProvisionedModelThroughput API returns.

C. Add exponential backoff retry logic to handle throttling exceptions during peak hours.

D. Modify the application to use the InvokeModelWithResponseStream API instead of the InvokeModel API.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #24 *Topic 1*

A company is building an AI advisory application by using Amazon Bedrock. The application will provide recommendations to customers. The company needs the application to explain its reasoning process and cite specific sources for data. The application must retrieve information from company data sources and show step-by-step reasoning for recommendations. The application must also link data claims to source documents and maintain response latency under 3 seconds.
Which solution will meet these requirements with the LEAST operational overhead?

A. Use Amazon Bedrock Knowledge Bases with source attribution enabled. Use the Anthropic Claude Messages API with RAG to set high-relevance thresholds for source documents. Store reasoning and citations in Amazon S3 for auditing purposes.

B. Use Amazon Bedrock with Anthropic Claude models and extended thinking. Configure a 4,000-token thinking budget. Store reasoning traces and citations in Amazon DynamoDB for auditing purposes.

C. Configure Amazon SageMaker AI with a custom Anthropic Claude model. Use the model's reasoning parameter and AWS Lambda to process responses. Add source citations from a separate Amazon RDS database.

D. Use Amazon Bedrock with Anthropic Claude models and chain-of-thought reasoning. Configure custom retrieval tracking with the Amazon Bedrock Knowledge Bases API. Use Amazon CloudWatch to monitor response latency metrics.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #25 *Topic 1*

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. 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.

B. 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.

C. 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.

D. 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.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!

Question #26 *Topic 1*

A retail company is developing a customer service application that must process 10,000 daily queries about products, orders, and warranties. The application must be able to respond to queries about 50,000 product documents that are updated every day. The application must integrate with an order management API to check the status of orders and to help process returns. The application must maintain context throughout multi-turn interactions with customers. The company must collect complete audit trails for application responses.
Which solution will meet these requirements with the LEAST operational overhead?

A. Deploy a fine-tuned Amazon Bedrock Anthropic Claude model for each product category. Create AWS Lambda functions to connect each model to the order management API. Store conversation history in Amazon DynamoDB.

B. Create a custom model that uses continued pre-training on Amazon Bedrock to handle all product documentation. Set up an Amazon API Gateway REST API that uses AWS Lambda functions to connect the model to the order management API.

C. Use Amazon SageMaker AI with containers to deploy models. Use Amazon Kendra to search product documents. Use AWS Step Functions to orchestrate calls to the order management API.

D. Use an Amazon Bedrock agent with action groups to integrate with the order management API. Associate an Amazon Bedrock knowledge base with the agent to search product documentation by using Retrieval Augmentation Generation (RAG). Enable trace events to capture audit trails.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #27 *Topic 1*

An ecommerce company is using Amazon Bedrock to build a generative AI (GenAI) application. The application uses AWS Step Functions to orchestrate a multi-agent workflow to produce detailed product descriptions. The workflow consists of three sequential states: a description generator, a technical specifications validator, and a brand voice consistency checker. Each state produces intermediate reasoning traces and outputs that are passed to the next state. The application uses an Amazon S3 bucket for process storage and to store outputs.
During testing, the company discovers that outputs between Step Functions states frequently exceed the 256 KB quota and cause workflow failures.
A GenAI Developer needs to revise the application architecture to efficiently handle the Step Functions 256 KB quota and maintain workflow observability. The revised architecture must preserve the existing multi-agent reasoning and acting (ReAct) pattern.
Which solution will meet these requirements with the LEAST operational overhead?

A. Store intermediate outputs in Amazon DynamoDB. Pass only references between states. Create a Map state that retrieves the complete data from DynamoDB when required for each agent's processing step.

B. Configure an Amazon Bedrock integration to use the S3 bucket URI in the input parameter for large outputs. Use the ResultPath field and the ResultSelector field to route S3 references between the agent steps while maintaining the sequential validation workflow.

C. Use AWS Lambda functions to compress outputs to less than 256 KB before each agent state. Configure each agent task to decompress the outputs before processing and to compress results before passing them to the next state.

D. Configure a separate Step Functions state machine to handle each agent's processing. Use Amazon EventBridge to coordinate the execution flow between state machines. Use S3 references for the outputs as event data.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #28 *Topic 1*

A company provides a service that helps users from around the world discover new restaurants. The service has 50 million monthly active users. The company wants to implement a semantic search solution across a database that contains 20 million restaurants and 200 million reviews. The company currently stores the data in a PostgresQL database.
The solution must support complex natural language queries and return results for at least 95% of queries within 500 ms. The solution must maintain data freshness for restaurant details that update hourly. The solution must also scale cost-effectively during peak usage periods.
Which solution will meet these requirements with the LEAST development effort?

A. Migrate the restaurant data to Amazon OpenSearch Service. Implement keyword-based search rules that use custom analyzers and relevance tuning to find restaurants based on attributes such as cuisine type, feature, and location. Create Amazon API Gateway HTTP API endpoints to transform user queries into structured search parameters.

B. Migrate the restaurant data to Amazon OpenSearch Service. Use a foundation model (FM) in Amazon Bedrock to generate vector embeddings from restaurant descriptions, reviews, and menu items. When users submit natural language queries, convert the queries to embeddings by using the same FM. Perform k-nearest neighbors (k-NN) searches to find semantically similar results.

C. Keep the restaurant data in PostgresQL and implement a pgvector extension. Use a foundation model (FM) in Amazon Bedrock to generate vector embeddings from restaurant data. Store the vector embeddings directly in PostgreSQL. Create an AWS Lambda function to convert natural language queries to vector representations by using the same FM. Configure the Lambda function to perform similarity searches within the database.

D. Migrate restaurant data to an Amazon Bedrock knowledge base by using a custom ingestion pipeline. Configure the knowledge base to automatically generate embeddings from restaurant information. Use the Amazon Bedrock Retrieve API with built-in vector search capabilities to query the knowledge base directly by using natural language input.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #29 *Topic 1*

A medical company uses Amazon Bedrock to power a clinical documentation summarization system. The system produces inconsistent summaries when handling complex clinical documents. The system performed well on simple clinical documents.
The company needs a solution that diagnoses inconsistencies, compares prompt performance against established metrics, and maintains historical records of prompt versions.
Which solution will meet these requirements?

A. Create multiple prompt variants by using Prompt management in Amazon Bedrock. Manually test the prompts with simple clinical documents. Deploy the highest performing version by using the Amazon Bedrock console.

B. Implement version control for prompts in a code repository with a test suite that contains complex clinical documents and quantifiable evaluation metrics. Use an automated testing framework to compare prompt versions and document performance patterns.

C. Deploy each new prompt version to separate Amazon Bedrock API endpoints. Split production traffic between the endpoints. Configure Amazon CloudWatch to capture response metrics and user feedback for automatic version selection.

D. Create a custom prompt evaluation flow in Amazon Bedrock Flows that applies the same clinical document inputs to different prompt variants. Use Amazon Comprehend Medical to analyze and score the factual accuracy of each version.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #30 *Topic 1*

A company uses Amazon Bedrock to generate technical content for customers. The company has recently experienced a surge in hallucination outputs when the company's model generates summaries of long technical documents. The model outputs include inaccurate or fabricated details. The company's current solution uses a large foundation model (FM) with a basic one-shot prompt that includes the full document in a single input.

The company needs a solution that will reduce hallucinations and meet factual accuracy goals. The solution must process more than 1,000 documents each hour and deliver summaries within 3 seconds for each document.

Which combination of solutions will meet these requirements? (Choose two.)

A. Implement zero-shot chain-of-thought (CoT) instructions that require step-by-step reasoning with explicit fact verification before the model generates each summary.

B. Use Retrieval Augmented Generation (RAG) with an Amazon Bedrock knowledge base. Apply semantic chunking and tuned embeddings to ground summaries in source content.

C. Configure Amazon Bedrock guardrails to block any generated output that matches patterns that are associated with hallucinated content.

D. Increase the temperature parameter in Amazon Bedrock.

E. Prompt the Amazon Bedrock model to summarize each full document in one pass.

Suggested Answer: *BC*

Currently there are no comments in this discussion, be the first to comment!

Question #31 *Topic 1*

A company has a recommendation system. The system's applications run on Amazon EC2 instances. The applications make API calls to Amazon Bedrock foundation models (FMs) to analyze customer behavior and generate personalized product recommendations.
The system is experiencing intermittent issues. Some recommendations do not match customer preferences. The company needs an observability solution to monitor operational metrics and detect patterns of operational performance degradation compared to established baselines. The solution must also generate alerts with correlation data within 10 minutes when FM behavior deviates from expected patterns.
Which solution will meet these requirements?

A. Configure Amazon CloudWatch Container Insights for the application infrastructure. Set up CloudWatch alarms for latency thresholds. Add custom metrics for token counts by using the CloudWatch embedded metric format. Create CloudWatch dashboards to visualize the data.

B. Implement AWS X-Ray to trace requests through the application components. Enable CloudWatch Logs Insights for error pattern detection. Set up AWS CloudTrail to monitor all API calls to Amazon Bedrock. Create custom dashboards in Amazon QuickSight.

C. Enable Amazon CloudWatch Application Insights for the application resources. Create custom metrics for recommendation quality, token usage, and response latency by using the CloudWatch embedded metric format with dimensions for request types and user segments. Configure CloudWatch anomaly detection on the model metrics. Establish log pattern analysis by using CloudWatch Logs Insights.

D. Use Amazon OpenSearch Service with the Observability plugin. Ingest model metrics and logs by using Amazon Kinesis. Create custom Piped Processing Language (PPL) queries to analyze model behavior patterns. Establish operational dashboards to visualize anomalies in real time.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #32 *Topic 1*

An enterprise application uses an Amazon Bedrock foundation model (FM) to process and analyze 50 to 200 pages of technical documents. Users are experiencing inconsistent responses and receiving truncated outputs when processing documents that exceed the FM's context window limits.
Which solution will resolve this problem?

A. Configure fixed-size chunking at 4,000 tokens for each chunk with 20% overlap. Use application-level logic to link multiple chunks sequentially until the FM's maximum context window of 200,000 tokens is reached before making inference calls.

B. Use hierarchical chunking with parent chunks of 8,000 tokens and child chunks of 2,000 tokens. Use Amazon Bedrock Knowledge Bases built-in retrieval to automatically select relevant parent chunks based on query context. Configure overlap tokens to maintain semantic continuity.

C. Use semantic chunking with a breakpoint percentile threshold of 95% and a buffer size of 3 sentences. Use the Amazon Bedrock RetrieveAndGenerate API call to dynamically select the most relevant chunks based on embedding similarity scores.

D. Create a pre-processing AWS Lambda function that analyzes document token count by using the FM's tokenizer. Configure the lambda function to split documents into equal segments that fit within 80% of the context window. Configure the Lambda function to process each segment independently before aggregating the results.

Suggested Answer: *C*

Currently there are no comments in this discussion, be the first to comment!

Question #33 *Topic 1*

A company is developing a generative AI (GenAI) application that analyzes customer service calls in real-time and generates suggested responses for human customer service agents. The application must process 500,000 concurrent calls during peak hours with less than 200 ms end-to-end latency for each suggestion. The company uses existing architecture to transcribe customer call audio streams. The application must not exceed a pre-defined monthly compute budget and must maintain auto scaling capabilities.
Which solution will meet these requirements?

A. Deploy a large, complex reasoning model on Amazon Bedrock. Purchase provisioned throughput and optimize for batch processing.

B. Deploy a low-latency, real-time optimized model on Amazon Bedrock. Purchase provisioned throughput and set up automatic scaling policies.

C. Deploy a large language model (LLM) on an Amazon SageMaker AI real-time endpoint that uses dedicated GPU instances.

D. Deploy a mid-sized language model on an Amazon SageMaker AI serverless endpoint that is optimized for batch processing.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #34 *Topic 1*

An ecommerce company is building an internal platform to develop generative AI applications by using Amazon Bedrock foundation models (FMs). Developers need to select models based on evaluations that are aligned to ecommerce use cases. The platform must display accuracy metrics for text generation and summarization in dashboards. The company has custom ecommerce datasets to use as standardized evaluation inputs.
Which combination of steps will meet these requirements with the LEAST operational overhead? (Choose two.)

A. Import the datasets to an Amazon S3 bucket. Provide appropriate IAM permissions and cross-origin resource sharing (CORS) permissions to give the evaluation jobs access to the datasets.

B. Import the datasets to an Amazon S3 bucket. Provide appropriate IAM permissions and a VPC endpoint configuration to give the evaluation jobs access to the datasets.

C. Configure an AWS Lambda function to create model evaluation jobs on a schedule in the Amazon Bedrock console. Provide the URI of the S3 bucket that contains the datasets as an input. Configure the evaluation jobs to measure the real world knowledge (RWK) score for text generation and BERT Score for summarization. Configure a second Lambda function to check the status of the jobs and publish custom logs to Amazon CloudWatch. Create a custom Amazon CloudWatch Logs Insights dashboard.

D. Use Amazon SageMaker Clarify on a schedule to create model evaluation jobs. Use open source frameworks to create and run standardized evaluations. Publish results to Amazon CloudWatch namespaces. Use the word error rate score for text generation and toxicity for summarization as metrics for accuracy. Configure an AWS Lambda function to check the status of the jobs and publish custom logs to CloudWatch. Create a custom Amazon CloudWatch Logs Insights dashboard.

E. Run an Amazon SageMaker AI notebook job on a schedule by using the fmevals or ragas framework to run evaluations that use the datasets in the S3 bucket. Write Python code in the notebook that makes direct InvokeModel API calls to the FMs and processes their responses for evaluation. Publish job status and results to Amazon CloudWatch Logs to measure the real world knowledge (RWK) score for text generation and toxicity for summarization as metrics for accuracy. Create a custom CloudWatch Logs Insights dashboard.

Suggested Answer: *AC*

Currently there are no comments in this discussion, be the first to comment!

Question #35 *Topic 1*

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 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.

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 that has the approval decision. Store all review decisions in Amazon DynamoDB.

C. 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 DynamoDB.

D. 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.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #36 *Topic 1*

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? (Choose three.)

A. Deploy Amazon Textract and Amazon Augmented AI (Amazon A2I) within the same Region to extract relevant data from the scanned documents. Route low-confidence pages to human reviewers.

B. Use AWS Lambda functions to detect and redact PII from submitted documents before inference. Apply Amazon Bedrock guardrails to prevent inappropriate or unauthorized content in model outputs. Configure Region-specific IAM roles to enforce data residency requirements and to control access to the extracted data.

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 and apply object metadata. Configure IAM policies to store original documents within the same Region as each applicant. Enable object tagging for future audits.

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.

Suggested Answer: *ABD*

Currently there are no comments in this discussion, be the first to comment!

Question #37 *Topic 1*

A software company is using Amazon Q Business to build an AI assistant that allows employees to access company information and personal information by using natural language prompts. The company stores this information in an Amazon S3 bucket.
Each department in the company has a dedicated prefix in the S3 bucket. Each object name includes the S3 prefix of the department that it belongs to. Each department can belong to only a single group in AWS IAM Identity Center. Each employee belongs to a single department.
The company configures Amazon Q Business to access data stored in an S3 bucket as a data source. The company needs to ensure that the AI assistant respects access controls based on the user's IAM Identity Center group membership.
Which solution will meet this requirement with the LEAST operational overhead?

A. Create a JSON file named acl.json in each department folder. In each file, create access control entries that specify the IAM Identity Center group that should have access to that department's data. Indicate the location of the JSON file in the Access Control section of the data source settings.

B. Create a single JSON file named acl.json at the top level of the S3 bucket. Add access control entries that map each department's S3 prefix to its corresponding IAM Identity Center group. Indicate the location of the JSON file in the Access Control section of the data source settings.

C. For each IAM Identity Center group, create a separate permissions set that denies access to all prefixes in the S3 bucket. Add a StringNotEquals condition key to the permissions set for each group that specifies the department each group is associated with. Attach the permissions sets to the Identity Center groups.

D. Create a metadata file named metadata.json at the top level of the S3 bucket. Add an AccessControlList object to the file that specifies the S3 path of each department's prefix. Specify the IAM Identity Center group that should have access to each department's prefix. Reference the file location in the data source metadata settings.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #38 *Topic 1*

A healthcare company is using Amazon Bedrock to build a system to help practitioners make clinical decisions. The system must provide treatment recommendations to physicians based only on approved medical documentation and must cite specific sources. The system must not hallucinate or produce factually incorrect information.
Which solution will meet these requirements with the LEAST operational overhead?

A. Integrate Amazon Bedrock with Amazon Kendra to retrieve approved documents. Implement custom post-processing to compare generated responses against source documents and to include citations.

B. Deploy an Amazon Bedrock knowledge base and connect it to approved clinical source documents. Use the Amazon Bedrock RetrieveAndGenerate API to return citations from the knowledge base.

C. Use Amazon Bedrock and Amazon Comprehend Medical to extract medical entities. Implement verification logic against a medical terminology database.

D. Use an Amazon Bedrock knowledge base with Retrieve API calls and InvokeModel API calls to retrieve approved clinical source documents. Implement verification logic to compare against retrieved sources and to cite sources.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #39 *Topic 1*

A financial services company is developing a real-time generative AI (GenAI) assistant to support human call center agents. The GenAI assistant must transcribe live customer speech, analyze context, and provide incremental suggestions to call center agents while a customer is still speaking. To preserve responsiveness, the GenAI assistant must maintain end-to-end latency under 1 second from speech to initial response display. The architecture must use only managed AWS services and must support bidirectional streaming to ensure that call center agents receive updates in real time.
Which solution will meet these requirements?

A. Use the Amazon Transcribe streaming API to transcribe calls. Pass the text to Amazon Comprehend to perform sentiment analysis. Feed the results to Anthropic Claude on Amazon Bedrock by using the InvokeModel API. Store results in Amazon DynamoDB. Use a WebSocket API to display the results.

B. Use Amazon Transcribe streaming with partial results enabled to deliver fragments of transcribed text before customers finish speaking. Forward text fragments to Amazon Bedrock by using the InvokeModelWithResponseStream API. Stream responses to call center agents through an Amazon API Gateway WebSocket API.

C. Use Amazon Transcribe batch processing to convert calls to text. Pass complete transcripts to Anthropic Claude on Amazon Bedrock by using the ConverseStream API. Return responses through an Amazon Lex chatbot interface that call center agents can access from their work computers.

D. Use the Amazon Transcribe streaming API with an AWS Lambda function to transcribe each audio segment. Configure the Lambda function to call the Amazon Titan Embeddings model on Amazon Bedrock by using the InvokeModel API. Configure the Lambda function to publish results to an Amazon SNS topic. Subscribe the call center agents to the SNS topic.

Suggested Answer: *B*

Currently there are no comments in this discussion, be the first to comment!

Question #40 *Topic 1*

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.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #41 *Topic 1*

A company has set up Amazon Q Developer Pro licenses for all developers at the company. The company maintains a list of approved resources that developers must use when developing applications. The approved resources include internal libraries, proprietary algorithmic techniques, and sample code with approved styling.
A new team of developers is using Amazon Q Developer to develop a new Java-based application. The company must ensure that the new developer team uses the company's approved resources. The company does not want to make project-level modifications.
Which solution will meet these requirements?

A. Create a Git repository that contains all of the approved internal libraries, algorithms, and code samples. Include this Git repository in the application project locally as part of the workspace. Ensure that the developers use the @workspace context to retrieve suggestions from the Git repository.

B. In the project root folder, create a folder named .amazonq/rules. Add the approved internal libraries, algorithms, and code samples to the folder.

C. Create a folder in the application project named rules. Store the guidelines and code in the folder for Amazon Q Developer to reference product code suggestions.

D. Create an Amazon Q Developer customization that includes the approved data sources. Ensure that the developers use the customization to develop the application.

Suggested Answer: *D*

Currently there are no comments in this discussion, be the first to comment!

Question #42 *Topic 1*

An ecommerce company is using Amazon Bedrock to build a customer service AI assistant. The AI assistant needs to process over 50,000 customer inquiries every day. The AI assistant occasionally experiences traffic spikes of up to 150,000 inquiries every day during promotional events. Analysis shows that 40% of inquiries follow similar patterns that share the same context.
A GenAI developer must design a solution that will ensure low latency and consistent performance for the AI assistant during traffic spikes.
Which solution will meet these requirements MOST cost-effectively?

A. Configure latency-optimized inference by setting the latency parameter to optimized in the performance configuration of the request to Amazon Bedrock. Use prompt caching to handle the repetitive inquiries.

B. Purchase provisioned throughput and model units (MUs) that are sized to handle peak traffic loads. Use Amazon ElastiCache (Redis OSS) to cache repetitive inquiries.

C. Use Amazon Bedrock Agents and custom knowledge bases to pre-process customer inquiries. Configure cross-Region inference to distribute traffic.

D. Use AWS Lambda functions to pre-process requests by using a custom prompt routing mechanism. Use Amazon DynamoDB as a caching layer to handle frequently asked questions.

Suggested Answer: *A*

Currently there are no comments in this discussion, be the first to comment!