Download Free Amazon AIP-C01 Exam PDF | PrepBolt

Don't miss out! Download the latest free Amazon AWS Certified Generative AI Developer - Professional exam PDF questions. Access real AIP-C01 dumps with verified answers and boost your chances to pass your certification on the first try with PrepBolt AIP-C01 exam pdf questions and answers.

Thank you for Downloading AIP-C01 exam PDF Demo

https://prepbolt.com/AIP-C01.html

Question 1

Question Type: MultipleChoice

A healthcare company is using Amazon Bedrock to develop a real-time patient care AI assistant to respond to queries for separate departments that handle clinical inquiries, insurance verification, appointment scheduling, and insurance claims. The company wants to use a multi-agent architecture.

The company must ensure that the AI assistant is scalable and can onboard new features for patients. The AI assistant must be able to handle thousands of parallel patient interactions. The company must ensure that patients receive appropriate domain-specific responses to queries.

Which solution will meet these requirements?

Options:

A- Isolate data for each agent by using separate knowledge bases. Use IAM filtering to control access to each knowledge base. Deploy a supervisor agent to perform natural language intent classification on patient inquiries. Configure the supervisor agent to route queries to specialized collaborator agents to respond to department-specific queries. Configure each specialized collaborator agent to use Retrieval Augmented Generation (RAG) with the agent's department-specific knowledge base.
B- Create a separate supervisor agent for each department. Configure individual collaborator agents to perform natural language intent classification for each specialty domain within each department. Integrate each collaborator agent with department-specific knowledge bases only. Implement manual handoff processes between the supervisor agents.
C- Isolate data for each department in separate knowledge bases. Use IAM filtering to control access to each knowledge base. Deploy a single general-purpose agent. Configure multiple action groups within the general-purpose agent to perform specific department functions. Implement rule-based routing logic within the general-purpose agent instructions.
D- Implement multiple independent supervisor agents that run in parallel to respond to patient inquiries for each department. Configure multiple collaborator agents for each supervisor agent. Integrate all agents with the same knowledge base. Use external routing logic to merge responses from multiple supervisor agents.

Answer:

A

Explanation:

Option A is the most appropriate design because it provides scalable multi-agent orchestration, clear domain separation, and strong governance with minimal operational complexity. A supervisor-agent pattern is a standard AWS-recommended approach for multi-agent systems: one agent performs intent classification and routing, while specialized agents handle domain-specific tasks.

Isolating data with separate knowledge bases ensures that each specialized collaborator agent retrieves only the information relevant to its department. This improves response accuracy, reduces hallucinations, and supports privacy controls because clinical content, claims content, and scheduling content can have different access policies. IAM-based filtering ensures that each agent has permission only to the knowledge base it is authorized to use.

Routing patient inquiries through a supervisor agent supports high concurrency and extensibility. New departments or features can be added by introducing new collaborator agents and knowledge bases without redesigning the entire system. Because routing is handled centrally, changes in classification logic do not require updates across many independent supervisors.

Using RAG within each collaborator agent ensures that responses are grounded in department-approved information sources, which is critical in healthcare settings to reduce unsafe or incorrect guidance. This approach also improves performance because each retrieval scope is smaller and more relevant, supporting thousands of parallel interactions.

Option B introduces manual handoffs that do not scale. Option C relies on rule-based routing inside one general agent, which becomes brittle and difficult to govern as complexity grows. Option D mixes all departments into a single knowledge base and merges responses externally, increasing risk of incorrect domain answers and operational overhead.

Therefore, Option A best meets the scalability, correctness, and multi-agent onboarding requirements.

Question 2

Question Type: MultipleChoice

A company is designing an API for a generative AI (GenAI) application that uses a foundation model (FM) that is hosted on a managed model service. The API must stream responses to reduce latency, enforce token limits to manage compute resource usage, and implement retry logic to handle model timeouts and partial responses.

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

Options:

A- Integrate an Amazon API Gateway HTTP API with an AWS Lambda function to invoke Amazon Bedrock. Use Lambda response streaming to stream responses. Enforce token limits within the Lambda function. Implement retry logic for model timeouts by using Lambda and API Gateway timeout configurations.
B- Connect an Amazon API Gateway HTTP API directly to Amazon Bedrock. Simulate streaming by using client-side polling. Enforce token limits on the frontend. Configure retry behavior by using API Gateway integration settings.
C- Connect an Amazon API Gateway WebSocket API to an Amazon ECS service that hosts a

containerized inference server. Stream responses by using the WebSocket protocol. Enforce token limits within Amazon ECS. Handle model timeouts by using ECS task lifecycle hooks and restart policies.
D- Integrate an Amazon API Gateway REST API with an AWS Lambda function that invokes Amazon Bedrock. Use Lambda response streaming to stream responses. Enforce token limits within the Lambda function. Implement retry logic by using Lambda and API Gateway timeout configurations.

Answer:

A

Explanation:

Option A is the best solution because it satisfies streaming, token control, and retry requirements while keeping operational overhead low by using fully managed, serverless AWS services. Amazon API Gateway HTTP APIs provide a lightweight, cost-effective front door for APIs and integrate cleanly with AWS Lambda for request processing and security controls.

AWS Lambda response streaming allows the API to begin returning content to the client as soon as partial model output is available, reducing perceived latency and improving user experience for long responses. Using Lambda as the integration layer also provides a centralized place to enforce token-aware request handling, such as rejecting oversized requests, truncating optional context, or applying consistent limits across users and tenants to manage compute usage.

Retry logic is best handled in the client or integration layer for transient failures such as timeouts and throttling. Lambda can implement controlled retries with exponential backoff and jitter, while API Gateway timeouts help bound request lifetimes and prevent hung connections from consuming resources indefinitely. Because the model service is managed, the company avoids infrastructure management and focuses only on request shaping, safety, and resiliency behavior.

Option B is not suitable because client-side polling is not true streaming, front-end token enforcement is insecure and inconsistent, and API Gateway does not provide model-aware retry behavior on its own. Option C introduces container hosting and scaling complexity, which increases operational overhead compared to serverless. Option D can work, but REST APIs are generally heavier than HTTP APIs for this pattern and do not reduce overhead compared to Option A.

Therefore, Option A provides the required streaming and resiliency capabilities with the least infrastructure management effort.

Question 3

Question Type: MultipleChoice

A company uses Amazon Bedrock to build a Retrieval Augmented Generation (RAG) system. The RAG

system uses an Amazon Bedrock Knowledge Bases that is based on an Amazon S3 bucket as the data source for emergency news video content. The system retrieves transcripts, archived reports, and related documents from the S3 bucket.

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

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

Which solution will meet these requirements?

Options:

A- Enhance embeddings by using a domain-adapted model that is specifically trained on emergency news content for improved vector similarity.
B- Migrate to Amazon OpenSearch Service. Use vector fields and metadata filters to define the scope of results retrieval.
C- Enable metadata-aware filtering within the Amazon Bedrock knowledge base by indexing S3 object metadata.
D- Migrate to an Amazon Q Business index to perform structured metadata filtering and document categorization during retrieval.

Answer:

C

Explanation:

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

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

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

Option A requires retraining or adapting embedding models, which the company has already

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

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

Question 4

Question Type: MultipleChoice

A financial services company is developing a customer service AI assistant application that uses a foundation model (FM) in Amazon Bedrock. The application must provide transparent responses by documenting reasoning and by citing sources that are used for Retrieval Augmented Generation (RAG). The application must capture comprehensive audit trails for all responses to users. The application must be able to serve up to 10,000 concurrent users and must respond to each customer inquiry within 2 seconds.

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

Options:

A- Enable tracing for Amazon Bedrock Agents. Configure structured prompts that direct the FM to provide evidence presentations. Integrate Amazon Bedrock Knowledge Bases with data sources to enable RAG. Configure the application to reference and cite authoritative content. Deploy the application in a Multi-AZ architecture. Use Amazon API Gateway and AWS Lambda functions to scale the application. Use Amazon CloudFront to provide low-latency delivery.
B- Enable tracing for Amazon Bedrock agents. Integrate a custom RAG pipeline with Amazon OpenSearch Service to retrieve and cite sources. Configure structured prompts to present retrieved evidence. Deploy the application behind an Amazon API Gateway REST API. Use AWS Lambda functions and Amazon CloudFront to scale the application and to provide low latency. Store logs in Amazon S3 and use AWS CloudTrail to capture audit trails.
C- Use Amazon CloudWatch to monitor latency and error rates. Embed model prompts directly in the application backend to cite sources. Store application interactions with users in Amazon RDS for audits.
D- Store generated responses and supporting evidence in an Amazon S3 bucket. Enable versioning on the bucket for audits. Use AWS Glue to catalog retrieved documents. Process the retrieved documents in Amazon Athena to generate periodic compliance reports.

Answer:

A

Explanation:

Option A is the correct solution because it relies on native Amazon Bedrock capabilities to deliver transparency, auditability, scalability, and low latency with minimal operational overhead. Amazon Bedrock Knowledge Bases provide a fully managed Retrieval Augmented Generation (RAG) implementation that automatically handles document ingestion, embedding, retrieval, and source attribution, enabling the application to cite authoritative content without building custom pipelines.

Enabling tracing for Amazon Bedrock Agents provides end-to-end visibility into agent reasoning steps, tool usage, and model interactions. This satisfies the requirement for comprehensive audit trails and supports regulatory review in financial services environments. Structured prompts further ensure that responses explicitly present reasoning and supporting evidence in a controlled, auditable format.

Using Amazon API Gateway and AWS Lambda allows the application to scale automatically to thousands of concurrent users without capacity planning. These services are designed for bursty workloads and can easily support the stated requirement of up to 10,000 concurrent users. Amazon CloudFront reduces latency by caching and accelerating content delivery, helping the application meet the strict 2-second response-time requirement.

Option B introduces a custom RAG pipeline with OpenSearch, increasing operational complexity and maintenance effort. Option C lacks native RAG integration and does not provide transparent reasoning or citation management. Option D focuses on offline compliance reporting rather than real-time transparency and low-latency responses.

Therefore, Option A best meets all requirements while minimizing infrastructure and operational overhead.

Question 5

Question Type: MultipleChoice

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.)

Options:

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:

The correct combination is B and C because together they address both workload diversity and hybrid deployment requirements with minimal custom engineering.

Option B provides consistent, high-throughput access by configuring provisioned throughput in Amazon Bedrock. Provisioned throughput guarantees predictable capacity and performance, which is essential for batch processing workloads that require sustained inference rates. This eliminates cold starts and throttling concerns that can occur with purely on-demand usage, making it well suited for high-volume enterprise workloads.

Option C enables hybrid deployment across cloud and on-premises environments by deploying foundation models to Amazon SageMaker AI endpoints and using Amazon SageMaker Neo for edge and on-premises optimization. SageMaker Neo compiles models for target hardware, allowing inference to run efficiently outside the AWS cloud while still using AWS-managed tooling. Orchestrating these deployments with AWS Lambda allows consistent invocation patterns across environments.

Option A uses asynchronous endpoints, which are not suitable for real-time, low-latency inference. Option D addresses scaling but does not support on-premises or hybrid deployment. Option E simplifies model onboarding but does not address hybrid execution or guaranteed throughput.

Therefore, Options B and C together provide real-time and batch support, predictable performance, and true hybrid deployment while minimizing operational overhead.

Thank You for trying AIP-C01 PDF Demo

To try our AIP-C01 practice exam software visit link below

https://prepbolt.com/AIP-C01.html

Start Your AIP-C01 Preparation

Use Coupon “SAVE50” for extra 50% discount on the purchase of Practice Test Software. Test your AIP-C01 preparation with actual exam questions.