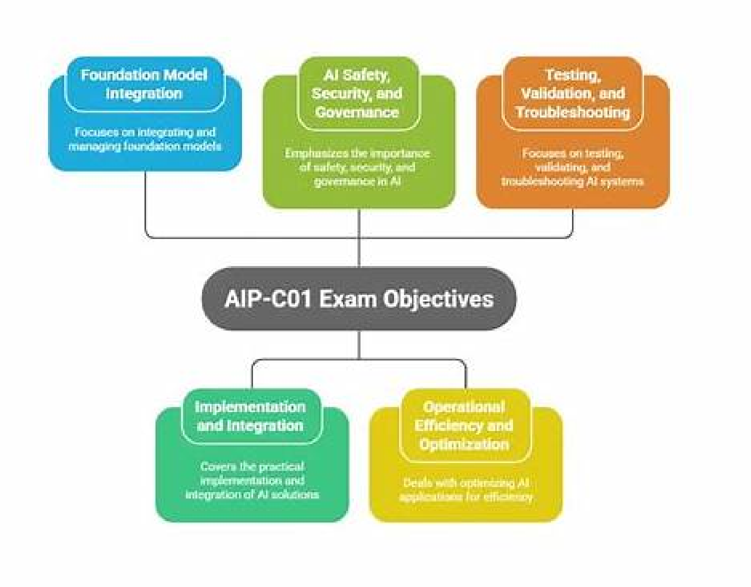
AIP-C01 Training Pdf Material & AIP-C01 Latest Study Material & AIP-C01 Test Practice Vce



P.S. Free & New AIP-C01 dumps are available on Google Drive shared by CramPDF: https://drive.google.com/open?id=11fmH0weyuJOAqdhEKwqHaGofgTXdiXGZ

Our company has successfully created ourselves famous brands in the past years, and more importantly, all of the AIP-C01 exam braindumps from our company have been authenticated by the international authoritative institutes and cater for the demands of all customers at the same time. We are attested that the quality of the AIP-C01 test prep from our company have won great faith and favor of customers. We persist in keeping close contact with international relative massive enterprise and have broad cooperation in order to create the best helpful and most suitable AIP-C01 study practice question for all customers. We can promise that our company will provide the authoritative study platform for all people who want to prepare for the exam. If you buy the AIP-C01 test prep from our company, we can assure to you that you will have the chance to enjoy the authoritative study platform provided by our company to improve your study efficiency.

Amazon AIP-C01 Exam Syllabus Topics:

Topic	Details
Topic 1	• Testing, Validation, and Troubleshooting: This domain covers evaluating foundation model outputs, implementing quality assurance processes, and troubleshooting GenAI-specific issues including prompts, integrations, and retrieval systems.
Topic 2	• AI Safety, Security, and Governance: This domain addresses input • output safety controls, data security and privacy protections, compliance mechanisms, and responsible AI principles including transparency and fairness.
Topic 3	• Foundation Model Integration, Data Management, and Compliance: This domain covers designing GenAI architectures, selecting and configuring foundation models, building data pipelines and vector stores, implementing retrieval mechanisms, and establishing prompt engineering governance.
Topic 4	• Implementation and Integration: This domain focuses on building agentic AI systems, deploying foundation models, integrating GenAI with enterprise systems, implementing FM APIs, and developing applications using AWS tools.
Topic 5	• Operational Efficiency and Optimization for GenAI Applications: This domain encompasses cost optimization strategies, performance tuning for latency and throughput, and implementing comprehensive monitoring systems for GenAI applications.

>> New AIP-C01 Braindumps Sheet <<

New AIP-C01 Exam Book, AIP-C01 Valid Exam Cost

These AWS Certified Generative AI Developer - Professional (AIP-C01) practice test questions also boost your confidence. If you have prepared well, tried all the Amazon AWS Certified Generative AI Developer - Professional Certification Exams, and understood each concept clearly, there is minimal or no chance of failure. Desktop Practice exam software and web-based AWS Certified Generative AI Developer - Professional (AIP-C01) practice test are available at CramPDF.

Amazon AWS Certified Generative AI Developer - Professional Sample Questions (Q105-Q110):

NEW QUESTION # 105
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. Store prompts in Amazon S3. Use AWS Step Functions to orchestrate the model interactions and service integrations.
- B. Use Amazon Bedrock Prompt Management to create versioned prompts. Include parameterized variables for each use case.
- C. Configure Amazon Bedrock intelligent prompt routing.
- D. Use Amazon Bedrock Flows to create workflows that combine FMs and AWS services.
- E. Configure AWS Config to record prompt changes. Use AWS CloudTrail to track prompt usage.

Answer: B,D

Explanation:
Amazon Bedrock Prompt Management is the purpose-built service for prompt lifecycle management. it provides native version control , allowing developers to test and compare variants side-by-side. Use of parameterized variables ensures that a single prompt structure can be consistently reused across different sub-applications (assistants vs. graders) while still being tailored to the specific context. To " integrate into the main application " with minimal overhead, Amazon Bedrock Flows provide a managed orchestration layer.
Flows allow developers to link managed prompts with AWS services (like knowledge bases or Lambda functions) without writing complex state-machine logic in Step Functions (Option B). This combination ensures consistent, auditable, and easily deployable prompt assets across the university ' s diverse use cases.

NEW QUESTION # 106
A financial services company is creating a Retrieval Augmented Generation (RAG) application that uses Amazon Bedrock to generate summaries of market activities. The application relies on a vector database that stores a small proprietary dataset with a low index count. The application must perform similarity searches.
The Amazon Bedrock model's responses must maximize accuracy and maintain high performance.
The company needs to configure the vector database and integrate it with the application.
Which solution will meet these requirements?

- A. Launch an Amazon Aurora PostgreSQL cluster and configure the index by using the Inverted File with Flat Compression (IVFFlat) algorithm. Configure the instance class to scale to a larger size when the load increases.
- B. Launch an Amazon DocumentDB cluster that has an IVFFlat index and a high probe value. Configure connections to the cluster as a replica set. Distribute reads to replica instances.
- C. Launch an Amazon MemoryDB cluster and configure the index by using the Hierarchical Navigable Small World (HNSW) algorithm. Configure a vertical scaling policy based on performance metrics.
- D. Launch an Amazon MemoryDB cluster and configure the index by using the Flat algorithm. Configure a horizontal scaling policy based on performance metrics.

Answer: C

Explanation:
Option B is the optimal solution because it maximizes similarity search accuracy and performance for a small, proprietary dataset while maintaining low operational complexity. Amazon MemoryDB is a fully managed, in- memory database that provides microsecond-level latency, making it ideal for real-time RAG workloads that require fast vector similarity searches.
For small datasets with low index counts, the Hierarchical Navigable Small World (HNSW) algorithm is recommended by AWS for its high recall and accuracy. Unlike approximate methods optimized for massive datasets, HNSW excels at returning the most semantically relevant vectors with minimal loss of precision, which directly improves the quality of responses generated by the Amazon Bedrock foundation model.
Vertical scaling in MemoryDB is sufficient for this use case because the dataset size is limited. Scaling up instance size provides increased memory and compute capacity without the complexity of managing distributed indexes or sharding strategies. This simplifies operations while maintaining predictable performance.
Option A's Flat algorithm is computationally expensive and inefficient at scale, even for moderate query volumes. Option C introduces higher latency and operational overhead by using a relational database not optimized for in-memory vector search. Option D is unsuitable because Amazon DocumentDB is not designed for high-performance vector similarity workloads and introduces unnecessary replica management complexity.
Therefore, Option B best meets the requirements for accuracy, performance, and efficient integration with an Amazon Bedrock-based RAG application.

NEW QUESTION # 107
A company uses AWS Lambda functions to build an AI agent solution. A GenAI developer must set up a Model Context Protocol (MCP) server that accesses user information. The GenAI developer must also configure the AI agent to use the new MCP server. The GenAI developer must ensure that only authorized users can access the MCP server.
Which solution will meet these requirements?

- A. Use a Lambda function to host the MCP server. Create an Amazon API Gateway HTTP API that proxies requests to the Lambda function. Configure the AI agent solution to use the Streamable HTTP transport to make requests through the HTTP API. Use Amazon Cognito to enforce OAuth 2.1.
- B. Use a Lambda layer to host the MCP server. Add the Lambda layer to the AI agent Lambda functions.Configure the agentic AI solution to use the STDIO transport to send requests to the MCP server. In the AI agent's MCP configuration, specify the Lambda layer ARN as the command. Specify the user credentials as environment variables.
- C. Use a Lambda function to host the MCP server. Grant the AI agent Lambda functions permission to invoke the Lambda function that hosts the MCP server. Configure the AI agent's MCP client to invoke the MCP server asynchronously.
- D. Use a Lambda function to host the MCP server. Grant the AI agent Lambda functions permission to invoke the Lambda function that hosts the MCP server. Configure the AI agent to use the STDIO transport with the MCP server.

Answer: A

Explanation:
Option C is the correct solution because it provides a secure, scalable, and standards-compliant way to expose an MCP server to an AI agent while enforcing strong user authorization. The Model Context Protocol supports HTTP-based transports for remote MCP servers, making Streamable HTTP the appropriate choice when the server is hosted as a managed service rather than a local process.
Hosting the MCP server in AWS Lambda enables automatic scaling and cost-efficient execution. By placing Amazon API Gateway in front of the Lambda function, the company creates a secure, managed HTTP endpoint that the AI agent can invoke reliably. This architecture cleanly separates transport, authentication, and business logic, which aligns with AWS serverless best practices.
Using Amazon Cognito to enforce OAuth 2.1 ensures that only authenticated and authorized users can access the MCP server. This satisfies security and compliance requirements when the MCP server handles sensitive user information. Cognito integrates natively with API Gateway, removing the need for custom authentication logic and reducing operational overhead.
Option A lacks user-level authorization controls. Option B and Option D rely on STDIO transport, which is intended for local or tightly coupled processes and is not suitable for distributed, serverless architectures.
Option D also introduces security risks by handling credentials through environment variables.
Therefore, Option C best meets the requirements for secure access control, scalability, and correct MCP integration in an AWS-based AI agent architecture.

NEW QUESTION # 108
A company is building a multicloud generative AI (GenAI)-powered secret resolution application that uses Amazon Bedrock and

Agent Squad. The application resolves secrets from multiple sources, including key stores and hardware security modules (HSMs). The application uses AWS Lambda functions to retrieve secrets from the sources. The application uses AWS AppConfig to implement dynamic feature gating. The application supports secret chaining and detects secret drift. The application handles short-lived and expiring secrets. The application also supports prompt flows for templated instructions. The application uses AWS Step Functions to orchestrate agents to resolve the secrets and to manage secret validation and drift detection.
The company finds multiple issues during application testing. The application does not refresh expired secrets in time for agents to use. The application sends alerts for secret drift, but agents still use stale data. Prompt flows within the application reuse outdated templates, which cause cascading failures. The company must resolve the performance issues.
Which solution will meet this requirement?

- A. Use Step Functions Map states to run agent workflows in parallel. Pass updated secret metadata through Lambda function outputs. Use AWS AppConfig to version all prompt flows to gate and roll back faulty templates.
- B. Use Amazon Bedrock Agents only. Configure Amazon Bedrock guardrails to restrict prompt variation.
Use an inline JSON schema for a single agent's workflow definition to chain tool calls.
- C. Use Amazon EventBridge Pipes to invoke resolvers based on Amazon CloudWatch log patterns. Store response metadata in DynamoDB with TTL and versioned writes. Use Amazon Q Developer to dynamically generate fallback prompts.
- D. Use a centralized Amazon EventBridge pipeline to invoke each agent. Store intermediate prompts in Amazon DynamoDB. Resolve agent ordering by using TTL-based backoff and retries.

Answer: A

Explanation:
Option A is the correct solution because it directly addresses all identified failure modes while preserving the existing Step Functions-based orchestration architecture with minimal redesign.
Using Step Functions Map states enables parallel execution of secret resolution workflows, which improves refresh latency for short-lived and expiring secrets. This ensures that secrets are refreshed in time before downstream agents require them. Passing updated secret metadata through Lambda outputs guarantees that subsequent steps always consume the latest resolved values, preventing agents from using stale data even after drift alerts are generated.
Versioning prompt flows in AWS AppConfig is critical to resolving cascading failures caused by outdated templates. AppConfig natively supports version control, validation, staged rollout, and rollback of configuration artifacts. By gating prompt flows through AppConfig, the company can immediately roll back faulty templates and prevent agents from reusing outdated instructions.
This solution maintains clear separation of concerns: Step Functions handle orchestration and parallelism, Lambda handles secret retrieval and metadata propagation, and AppConfig governs prompt lifecycle management. No additional event pipelines or custom retry coordination layers are required.
Option B oversimplifies the architecture and does not address secret lifecycle or drift. Option C introduces event-driven ordering complexity without solving prompt versioning. Option D introduces unnecessary tooling and dynamic prompt generation risk.
Therefore, Option A best resolves performance, correctness, and stability issues while minimizing operational overhead.

NEW QUESTION # 109
A financial services company is developing an AI-powered search assistant application to help investment advisors quickly retrieve investment data. The application runs as an AWS Lambda function. The company is using Amazon Bedrock to develop the application by using an Amazon Bedrock knowledge base that uses Amazon OpenSearch Serverless as its data source. The application agent must manage collections at scale by automatically assigning access permissions to collections and indexes that match a specific pattern. The company uses Amazon Bedrock tools to test the knowledge base. The knowledge base sync process finishes successfully. However, the test reveals a 400 Bad Authorization error from the BedrockAgentRuntime API and a 403 Forbidden error when the test attempts to access OpenSearch Serverless. The company must resolve the permissions issues. Which combination of solutions will meet this requirement? (Select TWO.)

- A. Update the Lambda function execution role to include the bedrock:InvokeAgent permission. Add the aoss:APIAccessAll permission to the Lambda execution role.
- B. Configure a VPC endpoint policy for OpenSearch Serverless. Add the endpoint to the Lambda function
' s VPC configuration.
- C. Enable IAM authentication for the OpenSearch Serverless domain. Add the es:ESHttp* permission to the Lambda function execution role.
- D. Configure AWS Secrets Manager to store OpenSearch Serverless credentials. Grant the Lambda function access to retrieve the credentials.
- E. Create an OpenSearch Serverless data access policy that includes pattern-based resource rules.

Answer: A,E

Explanation:
The errors described indicate missing permissions at both the application orchestration and data access levels.
The 400 Bad Authorization from BedrockAgentRuntime indicates the Lambda execution role lacks the identity permission to invoke the agent; adding bedrock:InvokeAgent and aoss:APIAccessAll (which allows the principal to interact with OpenSearch Serverless APIs) is necessary. The 403 Forbidden error from OpenSearch Serverless specifically relates to data-plane permissions. Unlike traditional OpenSearch, Serverless uses data access policies . To " manage collections at scale " automatically, a policy must be created that uses pattern-based resource rules (e.g., matching a prefix), ensuring that as new collections or indexes are created, the required principals (the Lambda role and the Bedrock service role) are granted the necessary access without manual policy updates for every new resource.

NEW QUESTION # 110
......