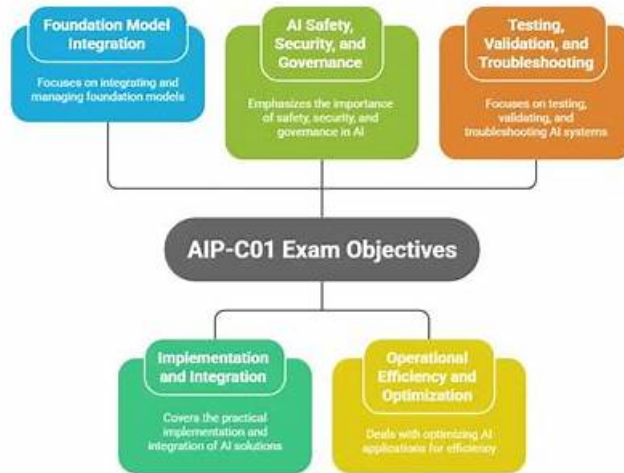


AIP-C01資格問題対応 & AIP-C01学習教材



さらに、Jpexam AIP-C01ダンプの一部が現在無料で提供されています：https://drive.google.com/open?id=1dPAVY3gVn-M9EB0I6Ir_Plv9Aw-3iSi

Amazon認定を取得したい場合は、行動し始めてみませんか？最初のステップは、AIP-C01試験に合格することです。時間は誰も待っていません。AIP-C01試験に合格した場合にのみ、より良いプロモーションを取得できます。そして、あなたがより効率的にそれを渡したいなら、私たちはあなたにとって最高のパートナーでなければなりません。私たちはプロのAIP-C01質問トレントプロバイダーであり、AIP-C01トレーニング資料は信頼に値します。AIP-C01ラーニングガイドに多大な努力を払っているため、10年以上にわたってこの分野でより良い成果を上げています。AIP-C01学習ガイドが最適です。

仕事に取り掛かって顧客とやり取りする前に厳密に訓練された責任ある忍耐強いスタッフ。AIP-C01試験の準備の質を実践し、経験すると、それらの保守性と有用性を思い出すでしょう。AIP-C01練習教材が試験受験者の98%以上が夢の証明書を取得するのに役立つ理由を説明しています。あなたもそれを手に入れることができると信じてください。

>> AIP-C01資格問題対応 <<

最新の Amazon AIP-C01資格問題対応 & 合格スムーズ AIP-C01学習教材 | 更新する AIP-C01関連合格問題

あなたはその他のAmazon AIP-C01「AWS Certified Generative AI Developer - Professional」認証試験に関するツールサイトでも見るかも知れませんが、弊社はIT業界の中で重要な地位があって、Jpexamの問題集は君に100%で合格させることと君のキャリアに変らせることだけでなく一年間中で無料でサービスを提供することもできます。

Amazon AIP-C01 認定試験の出題範囲:

トピック	出題範囲
トピック 1	<ul style="list-style-type: none">AIの安全性、セキュリティ、ガバナンス: この領域では、入出力の安全性制御、データセキュリティとプライバシー保護、コンプライアンスメカニズム、透明性と公平性を含む責任あるAI原則を扱います。

トピック 2	<ul style="list-style-type: none"> • 基盤モデルの統合、データ管理、およびコンプライアンス：この領域では、GenAIアーキテクチャの設計、基盤モデルの選択と構成、データパイプラインとベクトルストアの構築、検索メカニズムの実装、および迅速なエンジニアリングガバナンスの確立を扱います。
トピック 3	<ul style="list-style-type: none"> • テスト、検証、およびトラブルシューティング：この領域では、基盤モデルの出力の評価、品質保証プロセスの実装、およびプロンプト、統合、検索システムなどのGenAI固有の問題のトラブルシューティングを扱います。
トピック 4	<ul style="list-style-type: none"> • 実装と統合：この領域では、エージェント型AIシステムの構築、基盤モデルの展開、GenAIとエンタープライズシステムの統合、FM APIの実装、およびAWSツールを使用したアプリケーション開発に焦点を当てています。
トピック 5	<ul style="list-style-type: none"> • GenAIアプリケーションの運用効率と最適化：この分野は、コスト最適化戦略、レイテンシとスループットのパフォーマンスチューニング、およびGenAIアプリケーション向けの包括的な監視システムの導入を網羅しています。

Amazon AWS Certified Generative AI Developer - Professional 認定 AIP-C01 試験問題 (Q26-Q31):

質問 # 26

A company is using Amazon Bedrock to develop an AI-powered application that uses a foundation model that supports cross-Region inference and provisioned throughput. The application must serve users in Europe and North America with consistently low latency. The application must comply with data residency regulations that require European user data to remain within Europe-based AWS Regions.

During testing, the application experiences service degradation when Regional traffic spikes reach service quotas. The company needs a solution that maintains application resilience and minimizes operational complexity.

Which solution will meet these requirements?

- A. Use Amazon Bedrock cross-Region inference profiles by specifying geographical codes in profile IDs when the application calls the InvokeModel API. Configure separate Amazon API Gateway HTTP APIs to direct European and North American users to the appropriate Regional endpoints.
- B. Configure provisioned throughput for Amazon Bedrock in multiple Regions. Implement failover logic in the application code to switch between Regions when throttling occurs. Use AWS Global Accelerator to route traffic to the appropriate endpoints based on user location.
- C. Deploy separate Amazon Bedrock instances in North American and European Regions. Use a custom routing layer that directs traffic based on user location. Configure Amazon CloudWatch alarms to monitor Regional service usage. Use Amazon SNS to send email alerts to the company when usage approaches specified thresholds.
- D. Deploy a multi-Region Amazon API Gateway HTTP API and AWS Lambda functions that implement retry logic to handle throttling. Configure the Lambda functions to call the foundation model in the nearest secondary Region when the application reaches service quotas in the primary Region. Use intelligent routing to ensure compliance with data residency requirements.

正解: A

解説:

Option B best meets the latency, resilience, and data residency requirements while keeping operational complexity low by using built-in Amazon Bedrock cross-Region inference behavior through inference profiles. Cross-Region inference profiles are designed to provide higher availability and better traffic absorption when a single Region experiences throttling, transient capacity constraints, or quota-related degradation. By selecting the appropriate geography-scoped inference profile (for example, a Europe-scoped profile for European users and a North America-scoped profile for North American users), the application can keep inference traffic within the required geographic boundary. This directly supports EU data residency needs because European requests can be served only by Europe-based Regions while still benefiting from multi-Region resilience inside Europe.

The question also highlights degradation when Regional traffic spikes hit quotas. Cross-Region inference profiles help mitigate these conditions by allowing Bedrock to serve requests from another Region within the same geography, improving continuity during spikes without requiring the company to implement custom retry-and-failover logic across Regions. This reduces development and operational burden compared to building and maintaining a bespoke routing and fallback system.

Using separate Amazon API Gateway HTTP APIs to direct European and North American users to the correct endpoints simplifies

request routing and provides a clean boundary for compliance controls, logging, and monitoring. It also allows each geography to scale independently and maintain consistently low latency by keeping users close to the entry point and the Bedrock geography they must use.

Option A requires custom routing and manual operational monitoring and does not inherently solve quota- driven degradation.

Option C adds significant complexity by embedding throttling retries and cross-Region selection logic in Lambda while still needing careful controls to prevent cross-border routing mistakes. Option D introduces the highest operational complexity and can inadvertently violate residency if failover crosses geographies unless additional safeguards are implemented.

質問 # 27

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 predefined monthly compute budget and must maintain auto scaling capabilities.

Which solution will meet these requirements?

- A. Deploy a large language model (LLM) on an Amazon SageMaker real-time endpoint that uses dedicated GPU instances.
- **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, complex reasoning model on Amazon Bedrock. Purchase provisioned throughput and optimize for batch processing.
- D. Deploy a mid-sized language model on an Amazon SageMaker serverless endpoint that is optimized for batch processing.

正解: B

解説:

Option B is the correct solution because it aligns with AWS guidance for building high-throughput, ultra-low- latency GenAI applications while maintaining predictable costs and automatic scaling. Amazon Bedrock provides access to foundation models that are specifically optimized for real-time inference use cases, including conversational and recommendation-style workloads that require responses within milliseconds.

Low-latency models in Amazon Bedrock are designed to handle very high request rates with minimal per- request overhead. Purchasing provisioned throughput ensures that sufficient model capacity is reserved to handle peak loads, eliminating cold starts and reducing request queuing during traffic surges. This is critical when supporting up to 500,000 concurrent calls with strict latency requirements.

Automatic scaling policies allow the application to dynamically adjust capacity based on demand, ensuring cost efficiency during off-peak hours while maintaining performance during peak usage. This directly supports the requirement to stay within a predefined monthly compute budget.

Option A fails because batch processing and complex reasoning models introduce higher latency and are not suitable for real-time suggestions. Option C introduces significantly higher operational and cost overhead due to dedicated GPU instances and manual scaling responsibilities. Option D is optimized for batch workloads and cannot meet the sub-200 ms latency requirement.

Therefore, Option B provides the best balance of performance, scalability, cost control, and operational simplicity using AWS-native GenAI services.

質問 # 28

A company has a generative AI (GenAI) application that uses Amazon Bedrock to provide real-time responses to customer queries. The company has noticed intermittent failures with API calls to foundation models (FMs) during peak traffic periods.

The company needs a solution to handle transient errors and provide detailed observability into FM performance. The solution must prevent cascading failures during throttling events and provide distributed tracing across service boundaries to identify latency contributors. The solution must also enable correlation of performance issues with specific FM characteristics.

Which solution will meet these requirements?

- A. Implement a custom retry mechanism with a fixed delay of 1 second between retries. Configure Amazon CloudWatch alarms to monitor the application's error rates and latency metrics.
- B. Implement client-side caching of all FM responses. Add custom logging statements in the application code to record API call durations.
- **C. Configure the AWS SDK with standard retry mode and exponential backoff with jitter. Use AWS X-Ray tracing with annotations to identify and filter service components.**
- D. Configure the AWS SDK with adaptive retry mode. Use AWS CloudTrail distributed tracing to monitor throttling events.

正解: C

解説:

Option B best meets the combined resiliency and observability requirements because it applies AWS- recommended retry behavior for transient throttling and enables true distributed tracing across service boundaries. During peak traffic, intermittent failures are commonly caused by throttling and other transient conditions. The AWS SDK standard retry mode provides exponential backoff with jitter, which reduces synchronized retry storms, prevents cascading failures, and improves overall system stability. Jitter is important because it spreads retry attempts over time, reducing load amplification during throttling events.

For observability, AWS X-Ray provides distributed tracing that follows a request across components such as API Gateway or load balancers, application services, and downstream calls to Amazon Bedrock. X-Ray can identify where latency is being introduced and which downstream call is contributing most to end-to-end response time. This is required to "identify latency contributors" and isolate performance issues under load.

The requirement also states that the company must correlate performance issues with specific FM characteristics. X-Ray annotations are designed for this purpose: the application can annotate traces with the model ID, inference parameters, region, or inference profile used. This enables filtering and analysis (for example, comparing latency or error patterns by model, parameter set, or endpoint configuration) without building a separate telemetry system.

Option A's fixed-delay retries increase synchronized retry behavior and do not provide distributed tracing.

Option C does not prevent cascading failures and cannot provide cross-service tracing. Option D is incorrect because CloudTrail is an audit logging service and does not provide distributed tracing for request latency analysis.

Therefore, Option B provides the correct combination of resilient retries and deep, model-correlated distributed observability for Amazon Bedrock workloads.

質問 # 29

A financial services company needs to pre-process unstructured data such as customer transcripts, financial reports, and documentation. The company stores the unstructured data in Amazon S3 to support an Amazon Bedrock application.

The company must validate data quality, create auditable metadata, monitor data metrics, and customize text chunking to optimize foundation model (FM) performance.

Which solution will meet these requirements with the LEAST development effort?

- A. Set up an AWS Glue crawler to catalog data sources. Create AWS Glue ETL jobs to run custom transformation scripts. Use AWS Glue Data Quality to validate and monitor data quality. Load processed data into Amazon Bedrock.
- B. Create an AWS Step Functions workflow to orchestrate data pre-processing tasks. Run custom code on Amazon EC2 instances. Use Amazon SageMaker Model Monitor to monitor data quality. Load processed data into Amazon Bedrock.
- C. Use Amazon SageMaker Data Wrangler to create a data flow. Configure Amazon CloudWatch metrics and alarms to monitor data quality. Use a custom AWS Lambda function to pre-process the data. Load processed data into Amazon Bedrock.
- D. Use Amazon Comprehend to extract entities. Create an AWS Lambda function to chunk text. Run Amazon Athena to query and validate data quality. Load processed data into Amazon Bedrock.

正解: A

解説:

Option B is the most appropriate solution because it uses AWS-native, purpose-built data engineering and governance services to address data quality validation, metadata creation, monitoring, and transformation with minimal custom development. AWS Glue is designed specifically for large-scale data preparation and integrates seamlessly with Amazon S3, making it ideal for preprocessing unstructured datasets for downstream GenAI applications.

AWS Glue crawlers automatically infer schemas and populate the AWS Glue Data Catalog, creating auditable, queryable metadata for all datasets. This satisfies the requirement for traceability and governance, which is especially critical in financial services environments. Glue ETL jobs allow teams to implement customizable transformation logic, including text normalization and chunking strategies optimized for foundation model context windows.

AWS Glue Data Quality provides built-in rulesets for validating completeness, accuracy, and consistency. It also publishes quality metrics that can be monitored over time, meeting the requirement for ongoing data quality monitoring without building custom validation frameworks.

Because AWS Glue is fully managed, it eliminates the need to manage infrastructure, scaling, or orchestration. This significantly reduces development and operational effort compared to custom Lambda pipelines or EC2-based processing. The processed and validated data can then be safely ingested into Amazon Bedrock workflows or knowledge bases.

Option A and C require custom logic for validation, monitoring, and chunking, increasing development complexity. Option D introduces unnecessary infrastructure management and services not optimized for data preprocessing.

Therefore, Option B best meets the requirements while minimizing development effort and aligning with AWS Generative AI data preparation best practices.

質問 # 30

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. 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.**
- C. 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.
- 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.

正解: B

解説:

Option C best satisfies the requirements because it combines application-aware observability, metric baselining, anomaly detection, and correlated alerting using fully managed AWS services with minimal operational overhead. Amazon CloudWatch Application Insights is designed to automatically monitor application health by analyzing metrics, logs, and events across EC2-based workloads. This aligns directly with the need to detect intermittent performance issues and deviations from expected behavior.

By publishing custom metrics using the CloudWatch embedded metric format, the application can track generative AI-specific signals such as recommendation quality indicators, token usage, request volume, and response latency from Amazon Bedrock foundation model calls. Adding dimensions such as request type or user segment enables fine-grained visibility into which workloads or customer groups are impacted when recommendation quality degrades.

A critical requirement is detecting degradation compared to established baselines and generating alerts within

10 minutes. CloudWatch anomaly detection automatically builds statistical models of normal behavior for time-series metrics and flags deviations without requiring manually tuned thresholds. This capability is well suited for monitoring foundation model behavior, which can vary subtly over time. When anomalies are detected, CloudWatch alarms can trigger notifications with contextual metric data quickly, meeting the alerting requirement.

CloudWatch Logs Insights complements the metric-based view by enabling log pattern analysis and correlation. Engineers can query application logs and model response logs to identify recurring error patterns or shifts in output behavior that explain why recommendations no longer align with user preferences.

Application Insights further correlates metrics and logs to surface probable root causes, reducing mean time to resolution.

The other options lack one or more critical elements. Option A focuses on infrastructure-level metrics without baseline anomaly detection. Option B emphasizes tracing and auditing but does not provide automated performance deviation analysis. Option D offers flexibility but requires significantly more development and operational effort than a native CloudWatch-based solution.

質問 # 31

.....

私たちに知られているように、当社は、すべての受験者向けのAIP-C01試験資料を編集するために設立された専門ブランドです。当社のAIP-C01ガイドファイルは、この分野の当社の多くの専門家や教授によって設計されています。当社のAIP-C01認定準備資料は、教材資料市場で絶対的な権限を持っていると約束できます。弊社が設計した教材はあなたに最適な選択になると信じています。試験の準備をするときは、当社のAIP-C01ガイドファイルに完全に依存できます。

AIP-C01学習教材: https://www.jpexam.com/AIP-C01_exam.html

