



Microsoft

Exam Questions AI-200

Developing AI Cloud Solutions on Azure

About ExamBible

Your Partner of IT Exam

Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1

You are configuring sampling for a distributed application that sends traces to Azure Monitor. The solution must:

- Preserve upstream sampling decisions across distributed traces
- Capture all spans during local testing.
- Sample 10 percent of traces in production.

You need to apply the appropriate sampling configuration for each requirement.

What should you do? To answer, move the appropriate configurations to the correct requirements. You may use each configuration once, more than once or not at all. You may need to move the split bar Between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Configurations

AlwaysOnSampler

BatchSpanProcessor

ParentBasedSampler

AtzureMonitorTraceExporter

TraceIdRatioBasedSampler(0.1)

Selecting OpenTelemetry sampling strategies

Requirement

Maintain parent sampling decisions.

Record all spans during testing.

Sample 10 percent of traces in production.

Configuration

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Verified Answer: Preserve parent sampling decisions: ParentBasedSampler. Capture all spans locally: AlwaysOnSampler. Sample 10% in production: TraceIdRatioBasedSampler(0.1).

Detailed Explanation: Parent-based sampling propagates the sampling decision from the upstream parent, preventing broken sampling decisions across a distributed trace. AlwaysOnSampler records every span, which is suitable for local test environments where complete trace visibility is more important than telemetry volume. TraceIdRatioBasedSampler with 0.1 selects approximately ten percent of traces based on trace identifiers, providing deterministic fixed-ratio sampling for production. A batch span processor and an exporter control processing/export, not the sampling decision itself.

Study Guide Alignment: Security and operations: Key Vault, App Configuration, managed identity, OpenTelemetry, Azure Monitor, and KQL-based troubleshooting.

Official Microsoft Learn References: AI-200 Study Guide | OpenTelemetry sampling in Azure Monitor | Enable Azure Monitor OpenTelemetry

NEW QUESTION 2

You need to configure a connection string for the partner-facing service according to the technical requirements.

What should you use?

- A. Azure Key Vault references in App Service settings
B. GitHub secrets
C. Azure Container Registry Helm chart package
D. Dockerfile ENV instructions

Answer: A

Explanation:

Azure Key Vault references in App Service settings satisfy the requirement to keep secrets out of container images, source control, and directly stored application configuration. App Service resolves the referenced secret at runtime by using the app identity, so the application can consume the value as a normal setting without embedding credentials in the image. GitHub secrets are build/deployment secrets rather than a runtime App Service secret-delivery mechanism. Dockerfile ENV instructions would place secret material in the image configuration and violate the case requirements.

Study Guide Alignment: Security and operations: Key Vault, App Configuration, managed identity, OpenTelemetry, Azure Monitor, and KQL-based troubleshooting.

Official Microsoft Learn References: AI-200 Study Guide | Use Key Vault references for App Service and Functions | Managed identities for Azure resources

NEW QUESTION 3

You are developing a microservices-based application that uses Azure Container Apps. The application consists of several containerized services that handle tasks, such as processing orders, managing inventory, and generating reports. You deploy a new revision of the processing orders app.

Processing orders must be triggered by a web request and must always be available based on incoming web requests.

You need to validate that the replica is ready to handle incoming requests.

What should you implement?

- A. HTTP liveness probe
B. HTTP startup probe
C. TCP readiness probe
D. HTTP readiness probe
E. TCP liveness probe

Answer: D

Explanation:

Detailed Explanation: A readiness probe answers whether a running replica is currently ready to receive traffic. For a web-triggered order-processing service, an HTTP readiness probe can test the application endpoint and keep the replica out of the routing pool until it is able to handle requests. Liveness probes detect whether the process should be restarted, and startup probes protect slow-starting containers during initialization. A TCP readiness probe can test socket acceptance, but the HTTP readiness probe is the more application-aware choice for the HTTP service described.

Study Guide Alignment: Containerized Azure workloads: registry builds, App Service containers, Container Apps revision/scaling behavior, and AKS deployment choices.

Official Microsoft Learn References: AI-200 Study Guide | Health probes in Azure Container Apps

NEW QUESTION 4

You need to address the known issue resulting from vector similarity queries.
Which two actions should you perform? Each correct answer presents part of the solution. Choose two. NOTE: Each correct selection is worth one point.

- A. Add a composite index on the vector fields and metadata properties of the container.
- B. Reduce the numeric precision of the vector embeddings where supported.
- C. Set the account consistency level to Strong.
- D. Change the vector index type from flat to quantizedFlat or diskANN.

Answer: BD

Explanation:

The objective is to reduce RU consumption from vector similarity queries. Microsoft guidance identifies vector numeric precision and vector-index selection as major cost and performance levers. Using a lower supported vector precision can reduce storage and processing cost, while quantizedFlat and DiskANN are designed to reduce latency and RU consumption compared with flat search for appropriate data sizes. Strong consistency would increase resource cost rather than solve vector-search efficiency. A regular composite index is not a substitute for the vector index. Option B should be read as reducing vector numeric precision, not changing a generic ??indexing precision?? setting.
Study Guide Alignment: AI data-management workloads: Cosmos DB, PostgreSQL, caching, vector storage, vector retrieval, consistency, and connection optimization.
Official Microsoft Learn References: AI-200 Study Guide | Vector search in Azure Cosmos DB | Optimize Cosmos DB vector search performance

NEW QUESTION 5

You need to implement trace correlation according to the business requirements.
Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.
NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Actions

⋮ Redeploy the instrumented services.

⋮ Configure a trace exporter in the OpenTelemetry SDK.

⋮ Instrument the application code by using the OpenTelemetry SDK.

⋮ Call TelemetryClient.TrackEvent() within service methods.

⋮ Implement a view in the OpenTelemetry SDK.

Answer Area

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Trace correlation requires the application to create OpenTelemetry spans and an exporter to send those spans to Azure Monitor. Instrumentation must therefore be present before the updated service is deployed. The previous Application Insights SDK instrumentation does not satisfy the case-study requirement that all tracing use OpenTelemetry. Configuring an OpenTelemetry view is unrelated to basic distributed-trace export, and calling TrackEvent is an Application Insights SDK pattern rather than the required OpenTelemetry approach.
Study Guide Alignment: Security and operations: Key Vault, App Configuration, managed identity, OpenTelemetry, Azure Monitor, and KQL-based troubleshooting.
Official Microsoft Learn References: AI-200 Study Guide | Enable Azure Monitor OpenTelemetry

NEW QUESTION 6

You are developing an AI application that retrieves database credentials from Key Vault by using the Azure SDK for Python.
The application must use managed identity for authentication.

You review the following code segment that retrieves a secret from Key Vault:

```
01 from azure.identity import DefaultAzureCredential
02 from azure.keyvault.secrets import SecretClient
03
04 credential = DefaultAzureCredential()
05 client = SecretClient(
06     vault_url="https://vault.vault.azure.net/",
07     credential=credential
08 )
09
10 secret = client.get_secret("dbPassword", version="123")
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Secret retrieval behavior

Statement	Yes	No
The code retrieves a specific version of the dbPassword secret.	☐	☐
The code authenticates to Key Vault without storing client secrets in the application when deployed to an Azure hosted environment that has a managed identity enabled.	☐	☐
If dbPassword is rotated and a new version is created, subsequent calls to this code will retrieve the new version.	☐	☐

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

The get_secret call supplies an explicit version value, so it retrieves that particular version of dbPassword. DefaultAzureCredential can authenticate to Key Vault by using the Azure-hosted application's managed identity without embedding a client secret in the application. Because the code requests a fixed version, later secret rotation creates a newer version but subsequent executions of this exact code continue to request the specified old version. To follow the latest version automatically, the version parameter must be omitted.

Study Guide Alignment: Security and operations: Key Vault, App Configuration, managed identity, OpenTelemetry, Azure Monitor, and KQL-based troubleshooting.

Official Microsoft Learn References: AI-200 Study Guide | Managed identities for Azure resources | Use Key Vault references for App Service and Functions

NEW QUESTION 7

A company is implementing a publish-subscribe (Pub/Sub) messaging component by using Azure Service Bus. You are developing the first subscription application.

In the Azure portal you see that messages are being sent to the subscription for each topic. You create and initialize a subscription client object by supplying the correct details, but the subscription application is still not consuming the messages.

You need to ensure that the subscription client processes all messages. Which code segment should you use?

- A. subscriptionClient = new SubscriptionClient(ServiceBusConnectionString, TopicName, SubscriptionName);
- B. subscriptionClient.RegisterMessageHandler(ProcessMessagesAsync, messageHandlerOptions);
- C. await subscriptionClient.AddRuleAsync(new RuleDescription(RuleDescription.DefaultRuleName, new TrueFilter()));
- D. await subscriptionClient.CloseAsync();

Answer: B

Explanation:

Creating a subscription client establishes the client object but does not start application-level message processing. The message handler must be registered so received subscription messages are delivered to the application callback. Adding a default TrueFilter is only necessary when subscription rules require adjustment and does not start the receiver; the default subscription rule already accepts messages in a normal subscription. Closing the client would stop processing rather than enable it.

Study Guide Alignment: Azure service integration: Service Bus, Event Grid, Azure Functions triggers /bindings, and event-driven processing.

Official Microsoft Learn References: AI-200 Study Guide | Service Bus message transfers, locks, and settlement

NEW QUESTION 8

You are designing a solution that will use two Azure Functions apps: App1 and App2. App1 is Windows based and will be deployed as code. App2 is Linux based and will be deployed as a container image. Estimates show that the duration of the request processing for both apps will range from 1 to 10 minutes.

You plan to implement App1 and App2 by using the hosting plan to satisfy the following requirements:

- Request processing can complete within the estimated time range.

• The autoscaling behavior is event driven.

• The upper scaling limit is maximized.

You need to create the hosting plan for the implementation.

Which hosting plan should you create? To answer, move the appropriate hosting plans to the correct apps. You may use each hosting plan once, more than once, or not at all. You may..

Hosting plans

Premium

Dedicated

Consumption

App hosting plans

App

App1

App2

Hosting plan

Hosting plans

Hosting plans

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Hosting plans

Premium

Dedicated

Consumption

App hosting plans

App

App1

App2

Hosting plan

Consumption

Premium

Explanation

pp1: Consumption. App2: Premium.

Detailed Explanation: For App1, the Windows code-based Consumption plan satisfies event-driven autoscaling, allows execution up to the stated ten-minute range, and has the highest scaling ceiling among the listed event-driven choices for this workload. App2 is deployed as a Linux container image; among Premium, Dedicated, and Consumption, Premium is the event-driven plan that supports Linux containerized Functions and permits long-running execution. Dedicated hosting is not the requested event-driven autoscale model. The original Premium/Premium answer therefore did not maximize App1??s upper scaling limit.

Study Guide Alignment: Azure service integration: Service Bus, Event Grid, Azure Functions triggers/bindings, and event-driven processing.

Official Microsoft Learn References: AI-200 Study Guide | Azure Functions scale and hosting

NEW QUESTION 9

You are reviewing the message processing code in a backend worker service.

You review the following Python code that initializes and starts a Service Bus processor

```
from azure.servicebus import ServiceBusClient, ServiceBusReceiveMode

01 import json
02 from azure.servicebus import ServiceBusClient, ServiceBusReceiveMode
03 from azure.identity import DefaultAzureCredential
04 credential = DefaultAzureCredential()
05 with client.get_queue_receiver(
06     queue_name="inference-requests",
```

Service Bus processor behavior and SDK defaults

Statement	Yes	No
Messages are removed from the queue as soon as they are received by the processor.	☐	☐
If message processing fails due to a transient service error, the message can be retried by another consumer.	☐	☐
Messages with invalid JSON payloads are retried until the maximum delivery count is reached.	☐	☐

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

The default Service Bus receive behavior is PeekLock rather than receive-and-delete, so merely receiving a message does not remove it from the queue; it must be settled successfully. If processing fails and the message remains unsettled or the lock is lost, it can become available for another delivery. Invalid JSON, however, is an application-level content problem: Service Bus does not inspect the payload as JSON and automatically retry it to MaxDeliveryCount solely because deserialization fails. The application??s settlement/error path determines what happens.

Study Guide Alignment: Azure service integration: Service Bus, Event Grid, Azure Functions triggers/bindings, and event-driven processing.

Official Microsoft Learn References: AI-200 Study Guide | Service Bus message transfers, locks, and settlement | Service Bus messages, routing, and correlation

NEW QUESTION 10

You are developing a Java application to be deployed in Azure. The application stores sensitive data in Azure Cosmos DB.

You need to configure Always Encrypted to encrypt the sensitive data inside the application. What should you do first?

- A. Create a Microsoft Entra ID managed identity and assign the identity to a new Azure Key Vault instance.
- B. Create a new container to include an encryption policy with the JSON properties to be encrypted.
- C. Create a customer-managed key (CMK) and store the key in a new Azure Key Vault instance.
- D. Create a data encryption key (DEK) by using the Azure Cosmos DB SDK and store the key in Azure Cosmos DB.

Answer: C

Explanation:

For Cosmos DB Always Encrypted, the key hierarchy starts with a customer-managed key held in Azure Key Vault. The data encryption key used by the client is protected by that customer- managed key. Microsoft’s setup sequence therefore begins by preparing Key Vault and the CMK before creating DEKs or an encryption policy. Creating a managed identity, a container policy, or a DEK can be necessary later, but none precedes establishing the CMK that protects the client-side encryption key material.

Study Guide Alignment: AI data-management workloads: Cosmos DB, PostgreSQL, caching, vector storage, vector retrieval, consistency, and connection optimization.

Official Microsoft Learn References: AI-200 Study Guide | Always Encrypted for Azure Cosmos DB

NEW QUESTION 10

You are developing a solution that uses several Azure Service Bus queues. You create an Azure Event Grid subscription for the Azure Service Bus namespace. You use Azure Functions as subscribers to process the messages.

You need to ensure that events can be sent to Azure Event Grid from the queues. The solution must use the principle of least privilege and minimize costs. Which Azure Service Bus role and tier level should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Configuration	Value
Tier level	Basic Standard Premium
Role	Azure Service Bus Data Receiver Azure Service Bus Data Sender Azure Service Bus Data Owner

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Microsoft documents Event Grid integration for Service Bus queues and topics as requiring a Service Bus Premium namespace. The Azure Function subscriber must then receive messages from the queue; the least-privilege built-in data-plane role for that purpose is Azure Service Bus Data Receiver. Data Sender does not permit receiving, and Data Owner is broader than necessary. Premium therefore satisfies the feature prerequisite, while Data Receiver satisfies the stated least-privilege requirement.

Study Guide Alignment: Azure service integration: Service Bus, Event Grid, Azure Functions triggers /bindings, and event-driven processing.

Official Microsoft Learn References: AI-200 Study Guide | Service Bus to Event Grid integration | Authenticate Service Bus with Microsoft Entra ID

NEW QUESTION 12

You are developing several microservices to run on Azure Container Apps. External HTTP ingress traffic has been enabled for the Microservices. A deployed microservice must be updated to allow users to test new features. You have the following requirements:

- * Enable and maintain a single URL for the updated microservice to provide to test users.
- * Update the microservice that corresponds to the current microservice version. You need to configure Azure Container Apps.

Which features should you configure? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Anwer Area

Requirement:	Feature
Single URL for test users	Revision Label Revision mode Container image Container registry
Current microservice activation	Revision Label Revision mode Container image Container registry

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Azure Container Apps revision labels provide a stable label-specific URL that can be reassigned to a different revision, which is appropriate for a consistent test endpoint. Revision mode controls

whether one or multiple revisions can be active at the same time and therefore governs how the current and updated application versions are activated. The container image and registry identify what is deployed, but they do not provide the stable test URL or the application-level revision activation behavior requested. Study Guide Alignment: Containerized Azure workloads: registry builds, App Service containers, Container Apps revision/scaling behavior, and AKS deployment choices.

Official Microsoft Learn References: AI-200 Study Guide | Azure Container Apps revisions | Azure Container Apps revision labels

NEW QUESTION 15

You deploy a production Azure Function app that connects to an Azure SQL Database. The solution must provide the following functionality:

- Prevent secrets from being exposed in source control.
- Support secret rotation without redeploying the function app.
- Avoid downtime during credential updates.

You need to configure secure and maintainable secret management. What should you configure?

- A. Application settings with Key Vault references
- B. Environment variables in local.settings.json
- C. Parameter file stored in source control
- D. Hard coded connection string in the startup class

Answer: A

Explanation:

Key Vault references in Function/App Service application settings keep the connection secret outside source control while allowing the platform to resolve it at runtime by identity. When a referenced secret version is not fixed, rotation can be picked up without rebuilding or redeploying the function code.

`local.settings.json` is intended for local development and must not be used as a production secret store. A source-controlled parameter file or hard-coded connection string directly violates the security requirement.

Study Guide Alignment: Security and operations: Key Vault, App Configuration, managed identity, OpenTelemetry, Azure Monitor, and KQL-based troubleshooting.

Official Microsoft Learn References: AI-200 Study Guide | Use Key Vault references for App Service and Functions

NEW QUESTION 17

An ACA app processes messages from an Azure Storage queue.

The app must scale automatically based on messages in a specific Azure Storage queue by using a Kubernetes Event-driven Autoscaler (KEDA) custom scale rule. You need to configure the required scale rule values.

Which two values should you configure? Each correct answer presents part of the solution. Choose two. NOTE: Each correct selection is worth one point.

- A. Polling interval
- B. Trigger type
- C. Maximum replicas
- D. Queue name

Answer: BD

Explanation:

A KEDA custom scale rule must identify what scaler to use and which resource within that scaler should be observed. For Azure Storage queues, the trigger type selects the Storage Queue scaler and the queue name identifies the specific queue whose message count drives scaling. Polling interval and maximum replicas are general scaling controls, but they do not define the required queue-specific trigger. Therefore trigger type and queue name are the essential rule values tested here.

Study Guide Alignment: Containerized Azure workloads: registry builds, App Service containers, Container Apps revision/scaling behavior, and AKS deployment choices.

Official Microsoft Learn References: AI-200 Study Guide | Set scaling rules in Azure Container Apps

NEW QUESTION 20

You have an Azure web app that uses Azure Cosmos DB as a data store. You create a Cosmos DB container by running the following PowerShell script:

```
$resourceGroupName = "testResourceGroup"
$accountName = "testCosmosAccount"
$databaseName = "testDatabase"
$containerName = "testContainer"
$partitionKeyPath = "/EmployeeId"
$autoscaleMaxThroughput = 5000
```

```
New-AzCosmosDBSqlContainer `
    -ResourceGroupName $resourceGroupName `
    -AccountName $accountName `
    -DatabaseName $databaseName `
    -Name $containerName `
    -PartitionKeyKind Hash `
    -PartitionKeyPath $partitionKeyPath `
    -AutoscaleMaxThroughput $autoscaleMaxThroughput
```

You create the following queries that target the container:

```
SELECT * FROM c WHERE c.EmployeeId > '12345'
SELECT * FROM c WHERE c.UserId = '12345'
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
The minimum throughput for the container is 400 RU/s	☐	☐
The first query statement is an in-partition query.	☐	☐
The second query statement is a cross-partition query.	☐	☐

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

The question maps directly to the AI-200 objective "Develop AI solutions by using Azure Cosmos DB for NoSQL," which includes running queries and optimizing query performance and Request Unit (RU) consumption.

Statement 1: No — The minimum throughput is not 400 RU/s.

The PowerShell command provisions the container with:

-AutoscaleMaxThroughput 5000

Azure Cosmos DB autoscale operates between approximately 10% and 100% of the configured maximum throughput . Microsoft documentation specifically gives the example of an autoscale container provisioned with 5,000 RU/s scaling between 500 RU/s and 5,000 RU/s . Therefore, for this container, the minimum operating autoscale throughput is 500 RU/s , not 400 RU/s.

Therefore:
??The minimum throughput for the container is 400 RU/s.?? # No Statement 2: No — The first query is not an in-partition query. The container uses: /EmployeeId as its partition key. The first query is:
SELECT * FROM c WHERE c.EmployeeId > ' 12345 '
Although the query references the partition key, it uses a range predicate (>) , not an equality predicate. Microsoft explicitly states that a range filter on a partition key is not scoped to a single physical partition . To qualify as an in-partition query, the filter must identify the applicable partition, typically through an equality predicate such as:
WHERE c.EmployeeId = ' 12345 '
Microsoft ' s documentation provides essentially the same example: a query using DeviceId > ... against a container partitioned by DeviceId is not an in-partition query .
Therefore:
??The first query statement is an in-partition query.?? # No Statement 3: Yes — The second query is a cross-partition query. The second query is:
SELECT * FROM c WHERE c.UserId = ' 12345 '
The container ' s partition key is /EmployeeId, not /UserId. Because the query contains no filter on the partition key , Azure Cosmos DB cannot route it to one logical partition. It must fan out the query across the applicable physical partitions and combine the results.
Microsoft describes this behavior directly: when a query does not contain a filter on the partition key, it must execute across the partitions.
Therefore:
??The second query statement is a cross-partition query.?? # Yes

NEW QUESTION 24

A semantic search application queries Azure Database for PostgreSQL and stores document embeddings and metadata in a table with the following columns:

- embedding (pgvector)
- department
- created_at

The application must return the top five most similar documents for a given query embedding only from the finance department. You need to implement semantic retrieval with metadata filtering.

Which query components should you select? To answer, move the appropriate query components to the correct requirements. You may use each query component once, more than once, or not at all. You may need to move the split bar between panes or scroll to view content.

Query components

WHERE department = 'Finance'

WHERE department LIKE 'dep%'

ORDER BY created_at DESC LIMIT 5

ORDER BY embedding <=> query_embedding LIMIT 5

Vector similarity search with metadata filtering

Requirement

Filter rows to finance.

Rank by similarity and return the top five.

Query component

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

The metadata predicate must restrict the candidate rows to the Finance department, so the `WHERE department = ' finance ' ` component supplies the required filter. The pgvector cosine-distance operator ` < = > ` orders rows by vector distance to the supplied query embedding, and `LIMIT 5` keeps only the five nearest matches. Ordering by creation date would rank recency rather than semantic similarity, and a wildcard department filter would not satisfy the Finance-only requirement.

Study Guide Alignment: AI data-management workloads: Cosmos DB, PostgreSQL, caching, vector storage, vector retrieval, consistency, and connection optimization.

Official Microsoft Learn References: AI-200 Study Guide | Vector similarity search with Azure PostgreSQL

NEW QUESTION 28

You are reviewing an Azure Function app that processes incoming order requests for a company. The function must:

- Accept order submissions from an external client application.
- Require controlled access for security
- Return a response containing the processed request payload.

You review the following code segment:

```
01 import azure.functions as func
02 app = func.FunctionApp(http_auth_level=func.AuthLevel.FUNCTION)
03 @app.route(route="processorder", methods=["POST"])
04 def processorder(req: func.HttpRequest) -> func.HttpResponse:
05     result = req.get_body().decode("utf 8")
06     return func.HttpResponse(result)
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.
NOTE: Each correct selection is worth one point.

Code-level analysis of HTTP trigger configuration			
	Statement	Yes	No
	The function requires a function key for invocation.	☐	☐
	The function supports HTTP GET requests.	☐	☐
	The response returns the request body.	☐	☐
	The function validates the request body before returning a response.	☐	☐

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

`auth_level=FUNCTION`` means invocation requires a function key unless a stronger platform authentication layer is configured, so the first statement is true. The route explicitly permits POST, not GET. The function reads ``req.get_body()`` and returns that value in the `HttpResponse`, so the response contains the request body. No schema, type, required-field, or other payload validation is performed before the response is constructed, making the final statement false.
Study Guide Alignment: Azure service integration: Service Bus, Event Grid, Azure Functions triggers /bindings, and event-driven processing.
Official Microsoft Learn References: AI-200 Study Guide | Azure Functions HTTP trigger

NEW QUESTION 31

You have an Event Grid subscription that triggers an Azure Function.
You need to prevent loss of events in case the endpoint returns an HTTP 400 response.
Which action should you perform?

- A. Implement optimistic batchin
B. Implement asynchronous handshake validatio
C. Configure a dead-letter destinatio
D. Configure a retry policy.

Answer: C

Explanation:

Configuring a dead-letter destination is exactly the right step to take.
By default, when an endpoint returns an HTTP 400 (Bad Request) or HTTP 413 (Payload Too Large) response, Azure Event Grid immediately stops delivery attempts and drops the message. It treats these specific errors as non-transient, client-side issues, meaning it skips its standard 24-hour retry policy. Configuring a dead-letter destination ensures that these dropped events are safely preserved for future troubleshooting and reprocessing.
Reference:
<https://turbo360.com/blog/azure-event-grid-dead-letter-monitoring>

NEW QUESTION 36

You are developing an AI search API that caches semantic search results in Redis.
Search results must remain cached for 10 minutes. If the underlying data changes, cached entries must NOT be returned.
You need to implement a cache-aside strategy to ensure data consistency.
Which two actions should you perform? Each correct answer presents part of the solution.
NOTE: Each correct selection is worth one point.

- A. Configure a cache notification for key space event
B. Delete related cache keys when the source data change
C. Implement sliding expiration based on key acces
D. Configure a 10-minute Time to Live on each key.

Answer: BD

Explanation:

Delete keys on mutation. Implement an event-driven or database-hook system to purge related Redis keys the moment the source data updates.
Use absolute expiration. Set a strict, un-extendable Time-To-Live (TTL) of 10 minutes when writing to the cache, ensuring the data naturally expires even if a deletion event is missed.
Incorrect:
[Not C]
Implementing a sliding expiration is not a good step for this specific architecture because it violates the requirement that entries must not be returned if the underlying data changes.
Reference:
<https://medium.com/real-world-net/net-caching-strategies-compared-redis-vs-memory-vs-hybrid-b8b8be1e708d>

NEW QUESTION 40

.....

Relate Links

100% Pass Your AI-200 Exam with Exambible Prep Materials

<https://www.exambible.com/AI-200-exam/>

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>