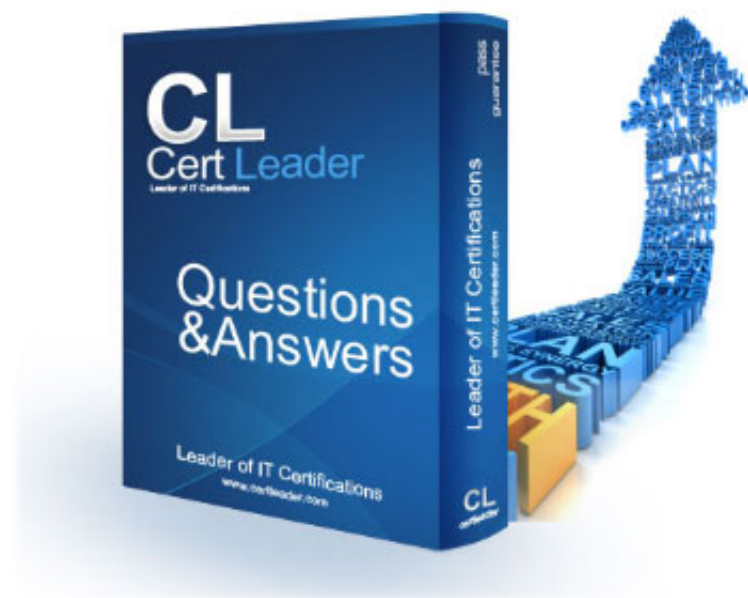


AI-200 Dumps

Developing AI Cloud Solutions on Azure

<https://www.certleader.com/AI-200-dumps.html>



NEW QUESTION 1

You are investigating high latency in an AI search application that processes millions of requests daily. Telemetry is stored in Azure Monitor Logs. You must create a KQL query that correlates information from the AppRequests table and the AppDependencies table. The query must meet the following requirements:

- Include only data from the last 24 hours.
- Filter for failed requests only.
- Calculate the average duration of dependencies, grouped by operation.

The query must follow best practice to optimize the performance by minimizing the initial data scan. You need to create the query.

In which order should perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Actions

⋮ Summarize average dependency duration by operation.

⋮ Join the dependencies table.

⋮ Select the requests table.

⋮ Filter the failed requests.

⋮ Apply a time filter.

Answer Area

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Verified Answer:1) Select the requests table; 2) apply the 24-hour time filter; 3) filter failed requests; 4) join the dependencies table; 5) summarize average dependency duration by operation.

Detailed Explanation:KQL performs best when high-selectivity filters are applied as early as possible, especially the time predicate that limits the amount of data scanned. Starting from AppRequests, restricting to the last 24 hours, and then filtering failures reduces the left-side dataset before the join. The dependency table is then correlated with those requests, and aggregation is performed last to calculate average dependency duration by operation. Joining or summarizing before the time and failure filters would process more data than necessary.

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 | Optimize log queries in Azure Monitor

NEW QUESTION 2

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

⋮ AtureMonitorTraceExporter

⋮ 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 3

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 4

You are developing a .NET application that uses Azure Cosmos DB for NoSQL to store application data The application uses the Azure Cosmos DB for NoSQL SDK to interact with the database account. The application must perform the following tasks

- Initialize the connection by using the account endpoint and key.
- Define shared throughput.
- Perform create, read, update, and delete (CRUD) operations on items stored in a container

Components

Database

CosmosClient

Container

Indexing policy

Answer Area

Azure Cosmos DB for NoSQL SDK object hierarchy

Requirement	Component
Initialize the connection by using the account endpoint and key.	
Define shared throughput.	
Perform item CRUD operations.	

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Verified Answer:Initialize connection: CosmosClient. Define shared throughput: Database. Perform item CRUD: Container.

Detailed Explanation:The Cosmos DB SDK follows the service resource hierarchy. CosmosClient represents the account connection and is initialized with the account endpoint and credential. Shared throughput is configured at the database level when multiple containers share database throughput. Item create, read, update, and delete operations are performed through a container object because containers hold items and define partitioning/indexing behavior. An indexing policy is configuration attached to a container, not the SDK object used for CRUD.

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 | Cosmos DB databases, containers, and items

NEW QUESTION 5

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 6

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 7

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. Choose two

NOTE: Each correct selection is worth one point.

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

Answer: AD

Explanation:

The cache-aside design needs both bounded lifetime and explicit invalidation. A ten-minute TTL ensures cached search results do not persist indefinitely, while deleting related keys when source data changes prevents known-stale entries from being returned during that ten-minute window. Sliding expiration would extend lifetime based on access and can preserve stale data. Keyspace notifications report Redis key events; they do not replace application logic that invalidates entries when the authoritative source changes.

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 | Azure Managed Redis documentation

NEW QUESTION 8

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.

Answer 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 9

You are designing an Azure Database for PostgreSQL table for semantic search. Queries frequently filter on the created_at column. The schema must support vector similarity search and reliable date filtering. You need to ensure that the table meets the requirements.

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. Store embeddings in a pgvector column.
- B. Store embeddings in a varchar column.
- C. Use a typed timestamp column for created_at.
- D. Store created_at as a free-form string.

Answer: AC

Explanation:

A semantic-search schema should use the pgvector vector type for embeddings so PostgreSQL can apply vector distance operators and vector indexes. The creation timestamp should use an actual timestamp data type so comparisons, range predicates, ordering, and indexes use date semantics rather than string ordering. Storing vectors or dates as free-form character data would sacrifice type validation and make similarity or date filtering less efficient and less reliable. 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 10

DRAG DROP - You are developing several microservices to run on Azure Container Apps. The microservices must allow HTTPS access by using a custom domain. You need to configure the custom domain in Azure Container Apps. In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Actions

- Add the custom domain name.
- Bind the certificate.
- Enable ingress.
- Validate the custom domain name.
- Add DNS records to the domain provider.

Answer area

- 1.
- 2.
- 3.
- 4.
- 5.

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Answer area

- 1.
- 2.
- 3.
- 4.
- 5.

Enable ingress.

Add DNS records to the domain provider.

Validate the custom domain name.

Add the custom domain name.

Bind the certificate.

NEW QUESTION 10

Drag and Drop

You are developing a new page for a website that uses Azure Cosmos DB for data storage. The feature uses documents that have the following format:

```
{
  "name": "John",
  "city": "Seattle"
}
```

You must display data for the new page in a specific order. You create the following query for the page:

```
SELECT *
FROM People p
ORDER BY p.name, p.city DESC
```

You need to configure an Azure Cosmos DB policy to support the query.

How should you configure the policy? To answer, drag the appropriate JSON segments to the correct locations. Each JSON segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

JSON segments	Answer Area
orderBy	<pre>{ "automatic": true, "ngMode": "Consistent", "includedPaths": [{ "path": "/" }], "excludedPaths": [], "": [[{ "path": "/name", "order": "descending" }, { "path": "/city", "order": " }]] }</pre>
sortOrder	
ascending	
descending	
compositeIndexes	

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Box 1: compositeIndexes

Azure Cosmos DB requires a defined composite index for any query utilizing an ORDER BY clause with multiple properties.

Box 2: descending

In an ORDER BY clause, properties without an explicit direction default to ascending order. Your query ORDER BY p.name, p.city DESC translates to an ascending sort on /name and a descending sort on /city.

Reference:

<https://docs.azure.cn/en-us/cosmos-db/index-policy>

NEW QUESTION 13

You deploy multiple instances of a change feed processor to handle a high ingestion rate within Azure Cosmos DB for NoSQL.

Each processor instance must process a different subset of partitions.

You need to ensure the workload is load-balanced across all processor instances.

What should you configure?

- A. indexing precision
- B. strong consistency
- C. lease container
- D. autoscale throughput

Answer: C

Explanation:

To handle a high ingestion rate in Azure Cosmos DB for NoSQL, you must deploy multiple instances of the change feed processor using the same workflow name and same lease container.

Reference:

<https://github.com/Azure/azure-cosmos-dotnet-v3/wiki/Comprehensive-Test-Suite-for-Azure-Cosmos-DB-Change-Feed-Processor:-All-Versions-and-Deletes>

NEW QUESTION 18

You deploy an AI application across multiple Azure regions.

The application must be able to view writes across all regions within a predictable time window.

You need to determine the appropriate consistency level.

What are two consistency levels you can use to achieve the goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Session
- B. Strong
- C. Bounded staleness
- D. Eventual

Answer: BC

Explanation:

The two appropriate consistency levels for a globally distributed AI application needing to view writes within a predictable time window are Bounded Staleness and Strong.

Bounded Staleness: This provides a guarantee that reads will lag behind writes by a user-specified "staleness window" (either in time or number of operations). It delivers high availability and predictable read consistency across all regions, balancing performance with strict read-order guarantees.

Strong: This provides linearizable and absolute consistency, ensuring that writes are seen synchronously across all regions. It features an exact zero-second predictable time window, though it comes at the expense of higher latency for multi-region operations due to the time required to replicate globally.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/consistency-levels>

NEW QUESTION 20

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your AI-200 Exam with Our Prep Materials Via below:

<https://www.certleader.com/AI-200-dumps.html>