Microsoft.AI-200.v2026-10-07.q23

Exam Code:	AI-200
Exam Name:	Developing AI Cloud Solutions on Azure
Certification Provider:	Microsoft
Free Question Number:	23
Version:	v2026-10-07
# of views:	112
# of Questions views:	241
https://www.freecram.net/torrent/Microsoft.AI-200.v2026-10-07.q23.html	

NEW QUESTION: 1
You optimize an AI inference API that uses Redis caching.
You must reduce the risk of serving outdated data while minimizing cache management overhead. You need to implement the caching strategy that satisfies the requirements. What should you do?
A. Reset expiration on each read.
B. Disable expiration for cache keys.
C. Set eviction policy to allkeys-lru.
D. Trigger invalidation when source data changes
Answer: (SHOW ANSWER)
Detailed Explanation: If the priority is to minimize stale results, source-driven invalidation is the direct control: when the underlying record changes, delete the corresponding cache entry so the next request repopulates it from current data. Sliding expiration can keep a frequently accessed stale item alive, disabling expiration is worse, and an allkeys-LRU policy evicts according to memory pressure rather than data freshness. Invalidation therefore best addresses correctness without requiring constant cache-wide maintenance.
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: 2
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:



Explanation:

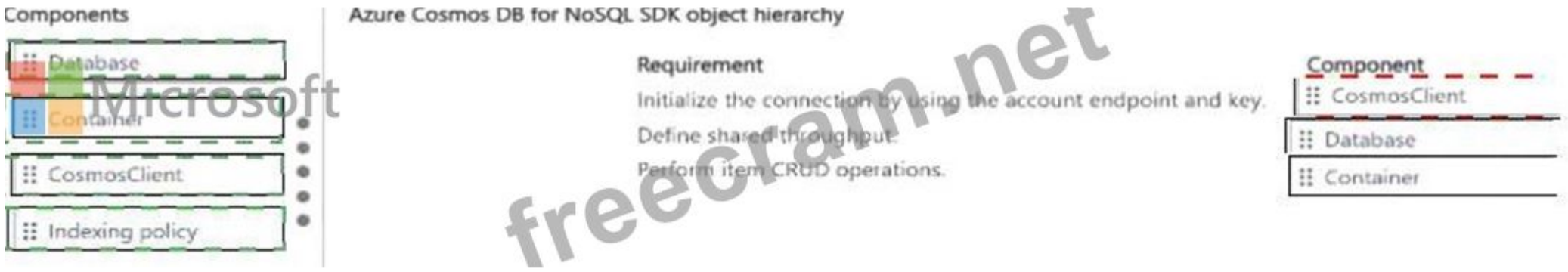


Verified answer: Tier: Premium. Role: Azure Service Bus Data Receiver.
Detailed Explanation: Microsoft documents Event Grid integration for Service Bus queues and topics as requiring a Service Bus Premium namespace. The Azure Function subscriber must then receive messages from the queue; the least-privilege 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: 3
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



Answer:



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.

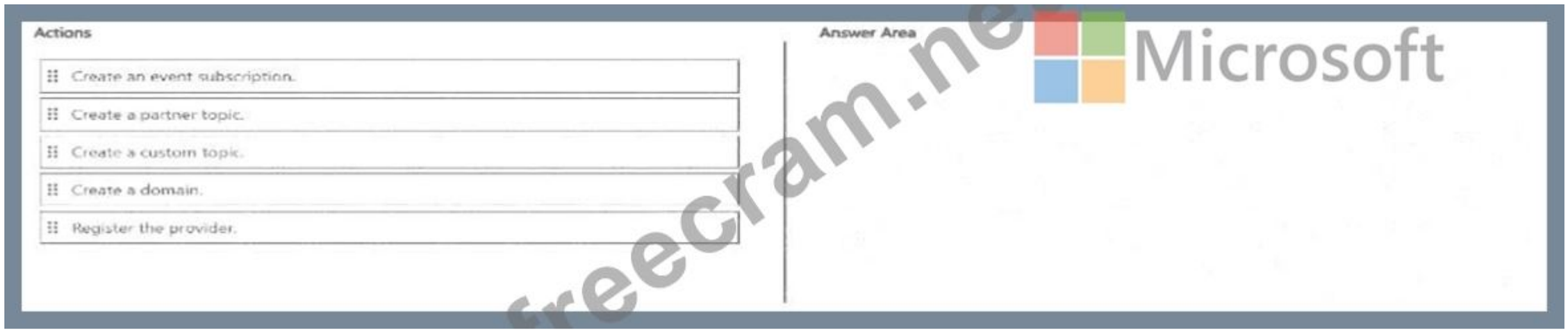
Official Microsoft Learn References: AI-200 Study Guide | Cosmos DB databases, containers, and items

NEW QUESTION: 4

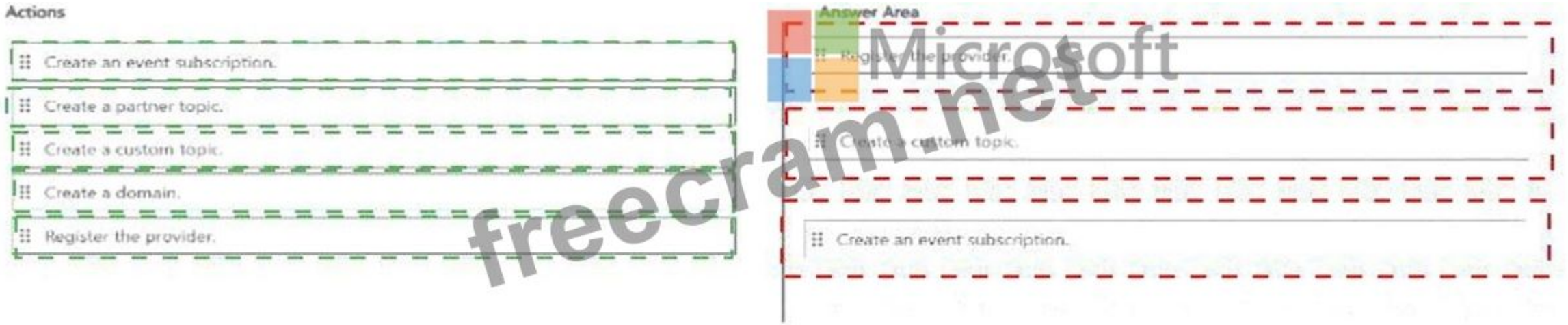
You have a newly provisioned Azure subscription. You are designing a custom Event Grid workflow for AI inference events.

You need to implement the Event Grid components to support routing of high-confidence events to a downstream processor.

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.



Answer:



Explanation:



Verified answer: 1) Register the Event Grid resource provider; 2) create a custom topic; 3) create an event subscription.

Detailed Explanation: A newly provisioned subscription must have the Event Grid resource provider available before Event Grid resources can be created. The publisher needs a custom topic as the event-ingress resource, and routing to a downstream processor is then defined by an event subscription on that topic. The event subscription can include filters such as event type or data fields so that only high-confidence events reach the processor. Creating a partner topic or domain is unnecessary for the stated custom workflow.

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 | Create an Event Grid custom topic or domain

NEW QUESTION: 5

You are developing a Retrieval-Augmented Generation (RAG) solution for a company. AI responses and embedding vectors are cached in Redis. The solution must meet the following requirements.

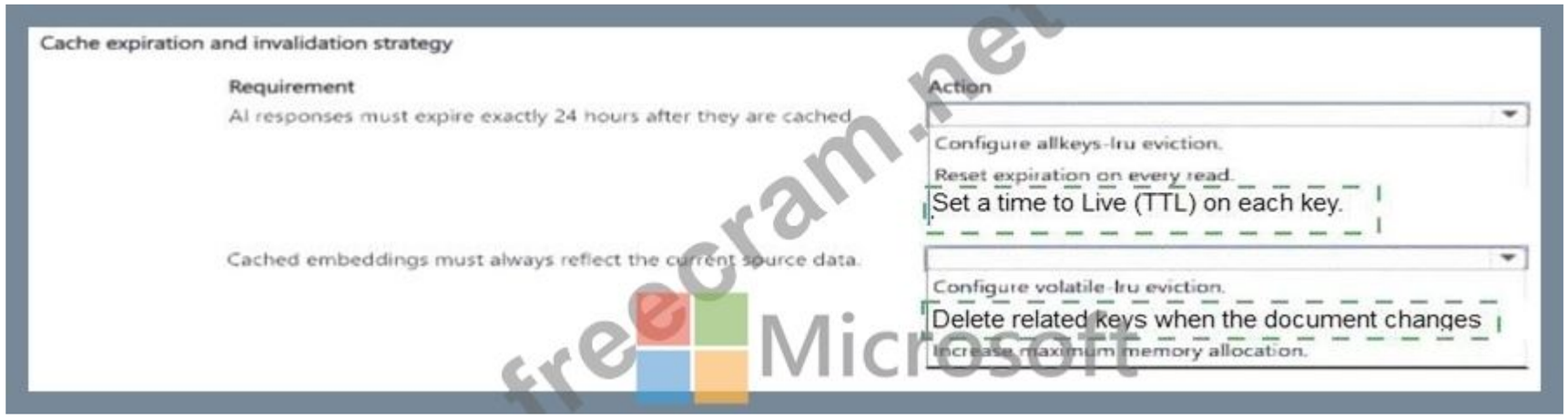
* AI responses must expire exactly 24 hours after they are cached

* Cached embeddings must always reflect the current source data.

You need to configure Redis to meet the requirements.



Answer:



Explanation:



I response expiry: set a 24-hour TTL on each key. Embedding freshness: delete related cache keys when the source document changes.

Detailed Explanation: A fixed TTL encodes the requirement that an AI response must expire after a defined period rather than survive indefinitely or have its lifetime extended on every read. Embedding freshness is a separate consistency requirement: when the authoritative document changes, any cached representation derived from the old document must be invalidated immediately. Memory eviction policies are capacity- management mechanisms and cannot guarantee a precise 24-hour lifetime or source-data consistency.

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: 6

You process Azure Service Bus messages that require a dependent external API call.

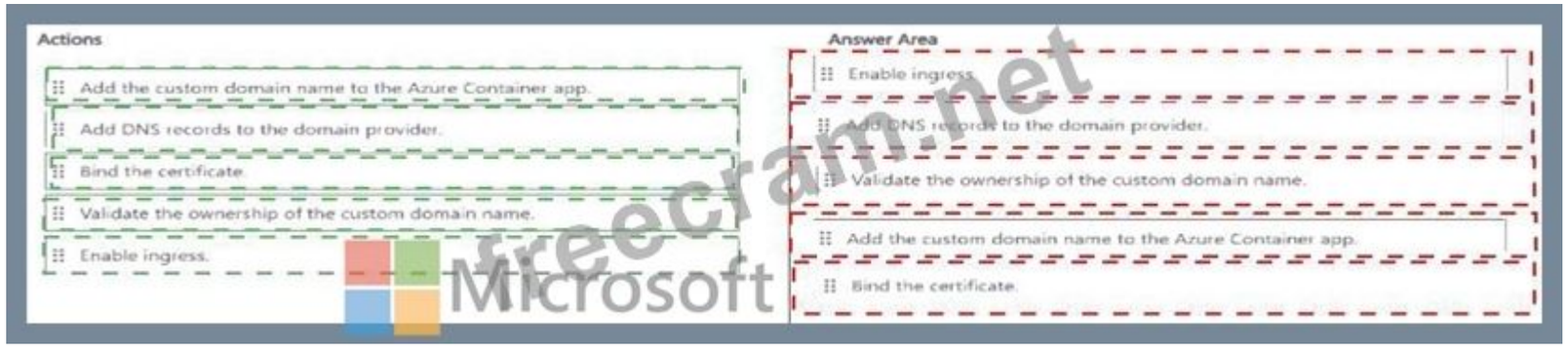
If the API is temporarily unavailable, you must delay processing of the message without incrementing the delivery count. You need to find a way to process the message when the API is available while keeping the message accessible. Which message action should you perform?

A. Complete
B. Abandon
C. Defer
D. Dead-letter
Answer: (SHOW ANSWER)

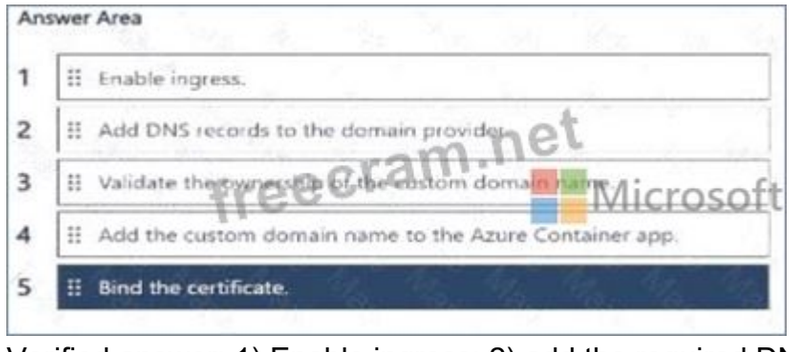
NEW QUESTION: 7
You are developing several microservices to run on Azure Container Apps.
The microservices must allow HTTPS access by using a custom domain.
You need to configure the custom domain in Azure Container Apps.
In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.



Answer:



Explanation:



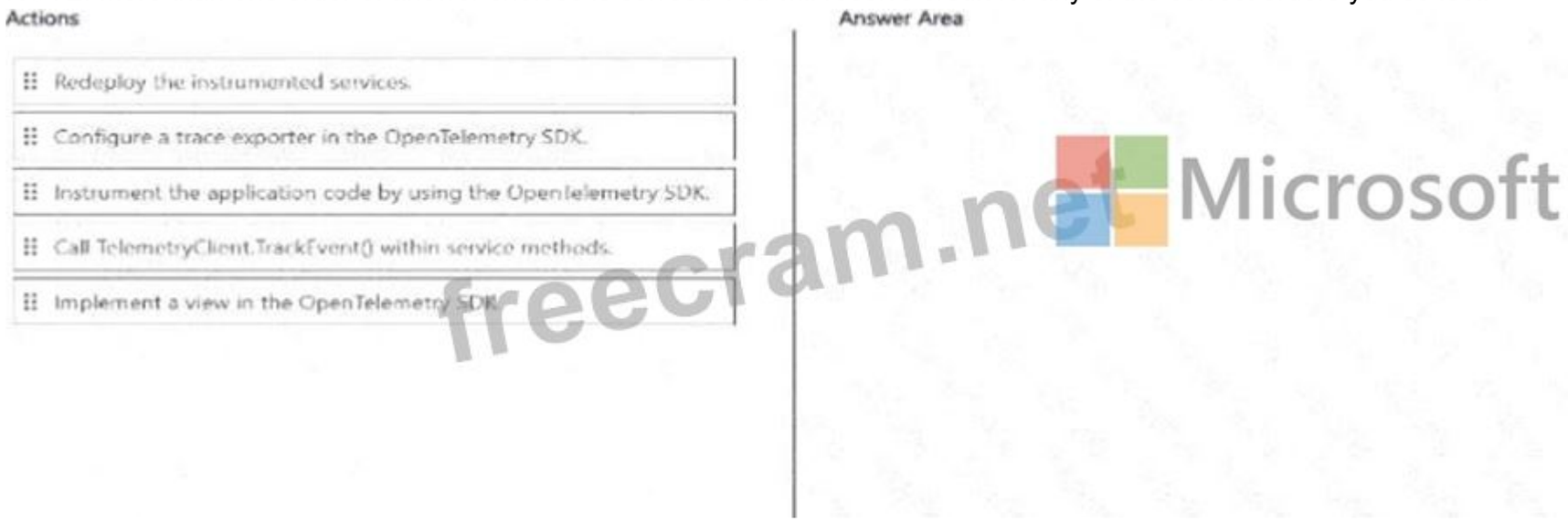
Verified answer: 1) Enable ingress; 2) add the required DNS records; 3) validate ownership of the custom domain; 4) add the custom domain to the Container App; 5) bind the certificate.
Detailed Explanation: A custom HTTPS hostname requires an ingress endpoint, DNS proof that the requester controls the hostname, the hostname association itself, and a TLS certificate bound to that hostname.

DNS records must exist before ownership validation can succeed. Once ownership is validated, the domain can be associated with the Container App and the certificate can be bound to provide HTTPS. Skipping ingress or DNS validation would prevent the hostname from being correctly routed and secured.
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 | Custom domains and certificates in Container Apps
Topic 1, Proseware Inc. Case Study
Background
Proseware Inc. develops AI-powered knowledge management solutions for enterprise customers. The company is modernizing its platform to support semantic search, intelligent document retrieval, and real-time partner integrations.
The engineering team uses Python and Azure SDKs. The architecture is being redesigned to support containerized microservices, vector search workloads, and serverless backend processing.
Planned Application Architecture
Microservices are containerized by using Docker.
Code for containerized microservices and Azure Function apps is developed locally but stored in a GitHub repository.
Custom images for containerized microservices are stored in Azure Container Registry (ACR).
Base images are stored in Docker Hub. Custom images must be rebuilt automatically whenever their base images are updated.
Azure Cosmos DB for NoSQL stores documents, metadata, and vector embeddings.
Azure Functions generate vector embeddings of Azure Cosmos DB for NoSQL-hosted documents and send messages to Service Bus to trigger search index updates.
Azure Container Apps (ACA) apps host backend API services that provide semantic search across Azure Cosmos DB for NoSQL documents. API services process Service Bus messages and update search indexes.
Azure Kubernetes Service (AKS) processes batch vector embedding regeneration for existing Azure Cosmos DB for NoSQL documents (whenever the embedding model is changed).
An extranet-facing containerized webhook allows business partners to submit documents to be processed by internal AI workflows for semantic search and retrieval.
Monitoring:
* Telemetry generated by Azure resources is sent to Azure Monitor.
* A Log Analytics workspace is used to collect ACA apps logs, AKS container logs, and Azure Functions apps logs.
* Monitoring of Azure Functions is currently implemented by using Azure Application Insights SDK instrumentation.
Business Requirements
Embeddings for new or updated Azure Cosmos DB for NoSQL-hosted documents must be automatically generated.
Backend API services must scale automatically during business hours.
Cold start delay of backend APIs must be minimized.
Secrets must be stored outside of container images.
Developers must be able to correlate telemetry across Azure Functions hosts and apps.
All tracing must be implemented by using OpenTelemetry SDK instrumentation.
Development efforts must be minimized.
Technical Requirements
Container images must be built automatically and validated before code updates are merged into the main branch.
Image build automation must run inside the Azure Container Registry, eliminating dependency on local developer machines and external build services.
Dependency of image builds on local developer machines must be eliminated.
Event-driven scaling in ACA must occur based on the number of pending messages in the Azure Service Bus queue.
Azure Cosmos DB for NoSQL RU consumption must be minimized.
Vector similarity search must use embeddings stored in Azure Cosmos DB for NoSQL.

The partner-facing containerized webhook service must run on Azure App Service.
Secrets must NOT be stored in container images, source control, or application configuration directly. They must be accessed securely at runtime.
All secrets must be stored centrally in Azure Key Vault and accessed at runtime through a managed identity.
Azure App Service must supply secrets at runtime without relying on external services.
Resources and workloads must be deployed by using Bicep templates through an automated, version- controlled pipeline. Local and command-line deployments must be eliminated to ensure repeatable, auditable deployments.
Known issues
RU consumption spikes during vector similarity queries.

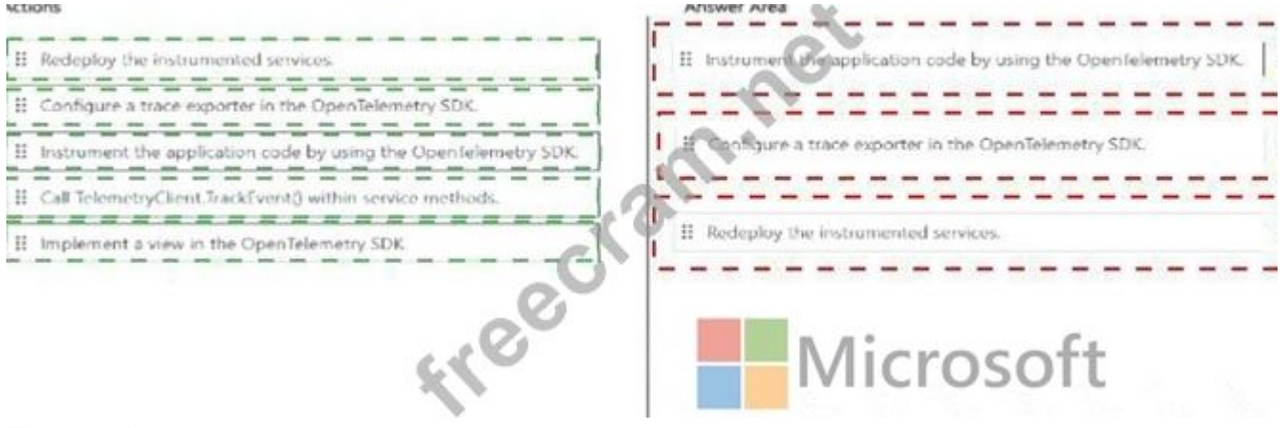
NEW QUESTION: 8
You need to configure a connection string for the partner-facing service according to the technical requirements.
What should you use?
A. Azure Key Vault references in App Service settings
B. GitHub secrets
C. Azure Container Registry Helm chart package
D. Dockerfile ENV instructions
Answer: (SHOW ANSWER)
Detailed 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: 9
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 se lect.



freecram.net Microsoft

Answer:



Explanation:



Verified answer: 1) Instrument the application code with the OpenTelemetry SDK; 2) configure an Azure Monitor trace exporter in the OpenTelemetry SDK; 3) redeploy the instrumented services.

Detailed 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: 10

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.



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:



Explanation:



Verified answer: Yes; Yes; No.

Detailed 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: 11

You plan to deploy a web application to AKS.

The solution must:

* Scale out the application by adding more pods during peak CPU usage
* Expose the application internally within the cluster only

You need to configure a Kubernetes resource for each requirement

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



Answer:



Explanation:



Verified answer: Scale out the application: HorizontalPodAutoscaler. Expose internally only: ClusterIP service.

Detailed Explanation: The Kubernetes HorizontalPodAutoscaler increases or decreases pod replicas in response to CPU, memory, or other metrics, so it is the resource that implements CPU-driven pod scale-out. A ClusterIP service creates an internal service IP that is reachable within the Kubernetes cluster and is the normal choice for an internal-only service. A Deployment defines the workload but does not itself implement metric-driven autoscaling, while Ingress is intended to route inbound network traffic.

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 | AKS scaling overview | Kubernetes services in AKS

NEW QUESTION: 12

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. Parameter file stored in source control

B. Environment variables in local.settings.json

C. Application settings with Key Vault references
D. Hard coded connection string in the startup class
Answer: (SHOW ANSWER)

NEW QUESTION: 13
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:



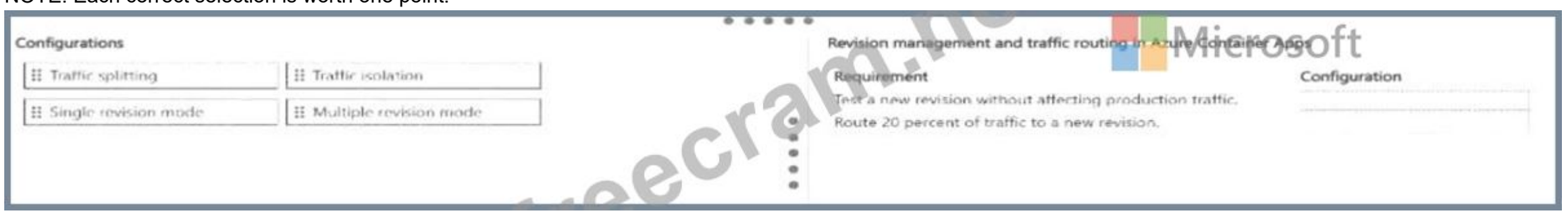
Explanation:



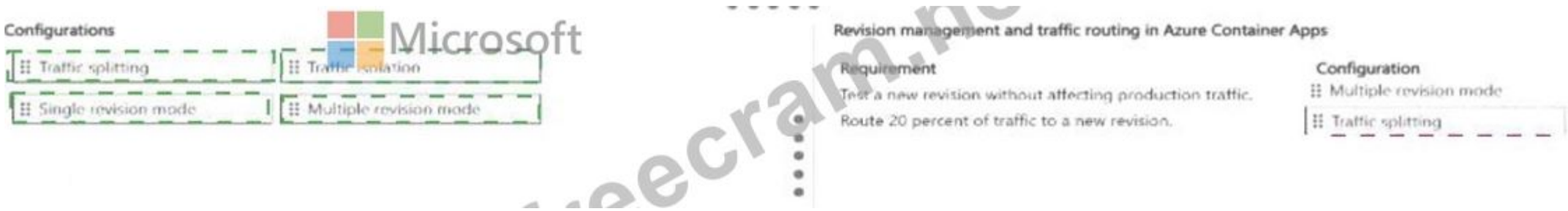
Verified answer: Single URL for test users: Revision label. Current microservice activation behavior:
Revision mode.
Detailed Explanation: Azure Container Apps revision labels provide a stable label-specific URL that can be reassigned to a different revision, which is appropriate for a consistent test 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

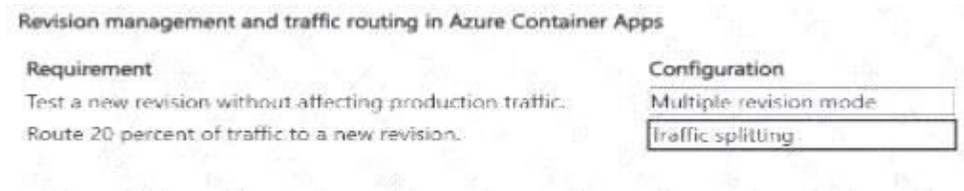
NEW QUESTION: 14
You deploy an API to Azure Container Apps.
The solution must provide the following functionality:
* Support the concurrent activation of multiple application versions.
* Allocate a specific percentage of incoming requests to a secondary version You need to configure revision behavior.
Which configurations should you use? 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.



Answer:



Explanation:



oncurrent active versions: Multiple revision mode. Route a percentage to the secondary version: Traffic splitting.
Detailed Explanation: Azure Container Apps supports side-by-side application versions through multiple revision mode. Once multiple revisions can be active, ingress traffic can be distributed by percentage across those revisions. This is the required combination for canary or controlled rollout scenarios. Single revision mode deactivates the previous revision as a new one becomes active, while traffic splitting is the specific feature that assigns a percentage of incoming requests to each active revision.
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

NEW QUESTION: 15

You are using the Python SDK to develop a containerized AI application that retrieves the value of a runtime setting stored in an Azure App Configuration resource. The application will run in Azure in the security context of a managed identity.
The application must work in a local development environment without any code changes.
You need to complete the code that implements the retrieval.
How should you complete the code? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:



Verified answer: var1 = DefaultAzureCredential(); retrieval call = get_configuration_setting(...).
Detailed Explanation: DefaultAzureCredential is designed to support the same application code across local development and Azure-hosted environments. Locally it can use developer credentials, while in Azure it can use the workload's managed identity. The App Configuration client's get_configuration_setting method retrieves a specific key/label pair, which matches the code shown. Using ManagedIdentityCredential directly would not provide the same local-development credential chain without code or environment-specific handling.
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 | Azure App Configuration Python client | Managed identities for Azure resources

NEW QUESTION: 16
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: (SHOW ANSWER)
Detailed 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

Valid AI-200 Dumps shared by EduDump.com for Helping Passing AI-200 Exam! EduDump.com now offer the **newest AI-200 exam dumps**, the EduDump.com AI-200 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com AI-200 dumps with Test Engine here: https://www.edudump.com/exams/Microsoft/AI-200/premium/ (**144** Q&As Dumps, **35%OFF Special Discount Code: freecram**)

NEW QUESTION: 17
You process Azure Service Bus messages that require a dependent external API call.
If the API is temporarily unavailable, you must delay processing of the message without incrementing the delivery count. You need to find a way to process the message when the API is available while keeping the message accessible. Which message action should you perform?
A. Defer
B. Complete
C. Dead-letter
D. Abandon
Answer: (SHOW ANSWER)
Detailed Explanation: Deferral is intended for a message that cannot be processed now because a particular dependency is temporarily unavailable, while the application wants to keep the message in the main queue for later retrieval by sequence number. Complete would permanently remove it, dead-letter would move it aside as a failed message, and abandon immediately releases the lock for redelivery. Deferral therefore best matches the requirement to postpone the individual message while keeping it available for deliberate later processing.
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 deferral | Service Bus message transfers, locks, and settlement

NEW QUESTION: 18
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

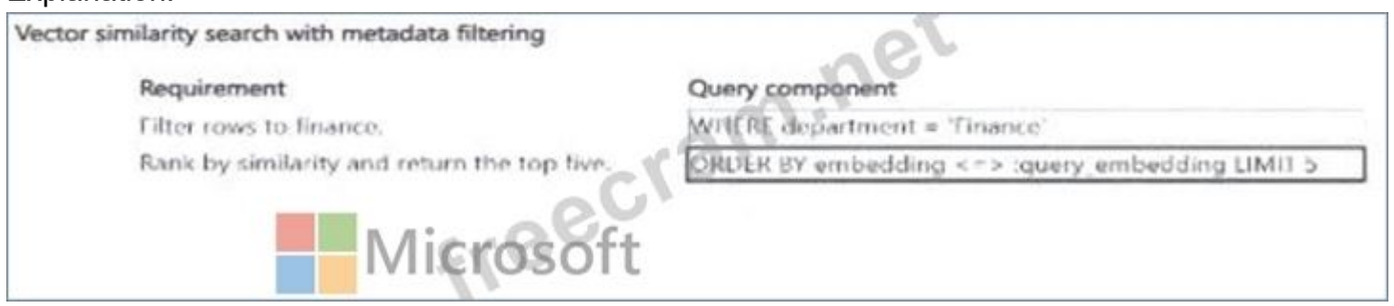
The application must return the top five most similar documents for a given query embedding only from the finance department. You need to implement semantic retrieval with metadata filtering.
Which query components should you select? To answer, move the appropriate query components to the correct requirements. 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.



Answer:



Explanation:



Verified answer: Filter: `WHERE department = ' finance ' `. Rank/top five: `ORDER BY embedding < = > query_embedding LIMIT 5`.
Detailed 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: 19
You provisioned an Azure Cosmos DB for NoSQL account named account1 with the default consistency level.
You plan to configure the consistency level on a per request basis. You plan to request Consistent Prefix consistency on a per-request basis.
You need to identify the resulting consistency level for read and write operations.
Which levels result from this configuration? To answer, select the appropriate options in the answer a rea.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:



Verified answer: Read operations: Consistent Prefix. Write operations: Session (the account default remains effective for writes).

Detailed Explanation: Azure Cosmos DB allows a client or request to override consistency for reads.

Microsoft explicitly notes that such an override applies only to reads; it does not change how writes are committed and replicated under the account's configured consistency. Because a new account uses Session consistency by default, requesting Consistent Prefix affects the reads while the account continues to use its Session consistency behavior for writes. This distinction is the key point tested by the hotspot.

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 | Manage Cosmos DB consistency levels | Cosmos DB consistency level choices

NEW QUESTION: 20

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.

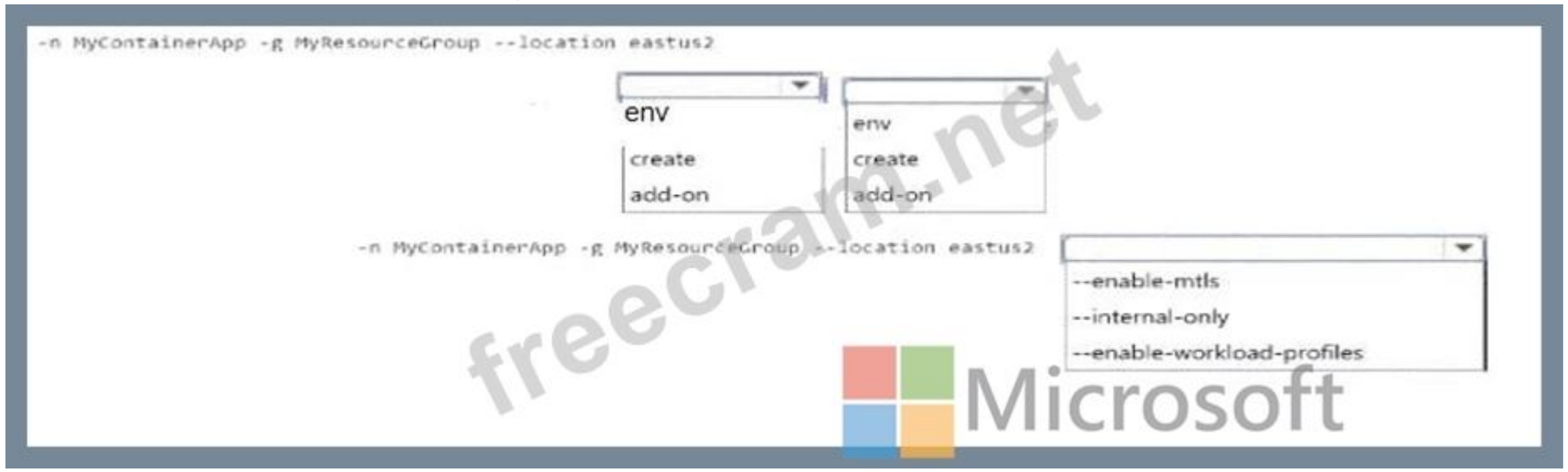
Freecram.net

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: (SHOW ANSWER)
Detailed 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

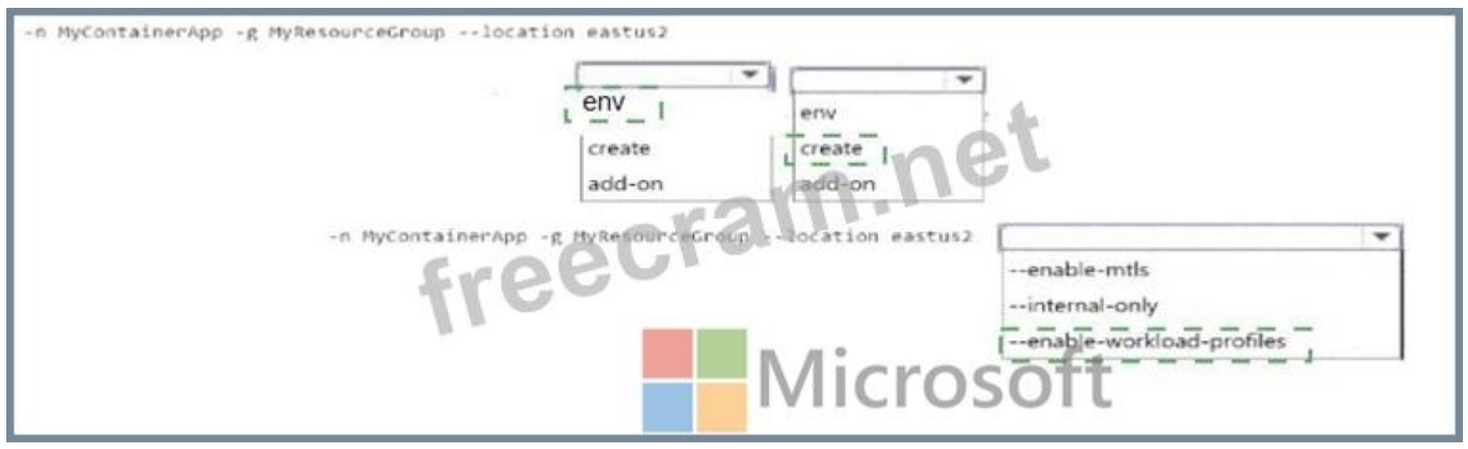
NEW QUESTION: 21
An ACA app processes messages from an Azure Storage queue.
The app must scale automatically based on messages in a specific Azure Storage queue by using a Kubernetes Event-driven Autoscaler (KEDA) custom scale rule. You need to configure the required scale rule values.
Which two values 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: (SHOW ANSWER)
Detailed 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: 22
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 must secure the container apps. All apps must reside in the same virtual network, share the same Dapr configuration, and share the same logging location.
Apps must support the configuration of the amount of memory and compute resources available to containers.
You need to configure the Azure Container App

How should you complete the CLI command? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:



Verified answer: `az containerapp env create ... --enable-workload-profiles`.
Detailed Explanation: The shared boundary for Container Apps is the managed environment. Apps in one environment can share the environment's virtual network integration, observability destination, and Dapr- related environment configuration. Enabling workload profiles provides selectable compute profiles and resource sizing options for workloads in that environment. The CLI must therefore create a Container Apps environment and enable workload profiles. Creating an individual container app alone would not establish the common environment-level network, logging, and resource-profile boundary required by the scenario.
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 environments

NEW QUESTION: 23
You develop a message-processing service deployed to Azure Container Apps. The service reads messages from an Azure Service Bus queue.
The solution must minimize costs by ensuring NO compute resources are consumed when the queue is empty.

You need to configure scaling for the service.
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 Kubernetes Event-driven Autoscaler rule that monitors queue length.
B. Configure the scaling rule to allow for the termination of all active replicas.
C. Increase the scaling rule to allow for the maximum running replica count.
D. Enable HTTP ingress concurrency scaling
Answer: (SHOW ANSWER)
Detailed Explanation: Azure Container Apps uses KEDA-based scale rules for event-driven workloads. A Service Bus scaler can monitor queue depth and create replicas only when messages are pending. To meet the explicit cost requirement, the scale configuration must permit a minimum replica count of zero so all active replicas can terminate when the queue is empty. Increasing the maximum replica count changes peak capacity, and HTTP ingress concurrency is not the correct scaling signal for a background Service Bus consumer.
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

Valid AI-200 Dumps shared by EduDump.com for Helping Passing AI-200 Exam! EduDump.com now offer the **newest AI-200 exam dumps**, the EduDump.com AI-200 exam **questions have been updated** and **answers have been corrected** get the **newest** EduDump.com AI-200 dumps with Test Engine here:
https://www.edudump.com/exams/Microsoft/AI-200/premium/ (**144** Q&As Dumps, **35%OFF** **Special Discount Code:** **freecram**)