KillTest

Mejor calidad. Mejor servicio

Preguntas de práctica

https://www.killtest.es

Actualizaciones gratuitas durante un año

Exam : **AI-200**

Title : Developing AI Cloud Solutions on Azure (beta)

Version : DEMO

1. 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.

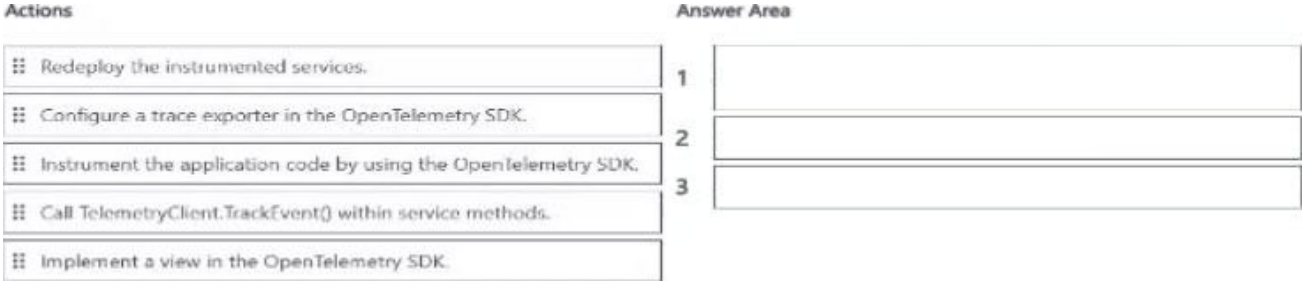
Business Requirements
Embeddings for new or updated Azure Cosmos DB for NoSQL–hosted documents must be automatically generated.
Backend API services must scale automatically during business hours.
Cold start delay of backend APIs must be minimized.
Secrets must be stored outside of container images.
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.

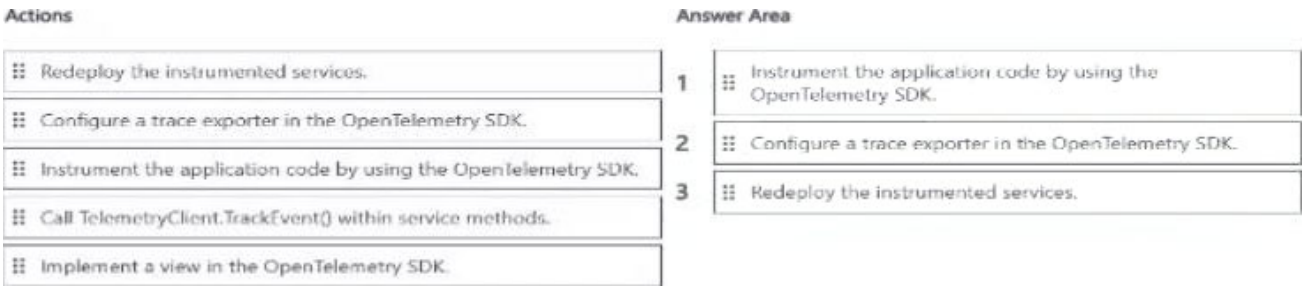
Technical Requirements
Container images must be built automatically and validated before code updates are merged into the main branch.
Image build automation must run inside the Azure Container Registry, eliminating dependency on local developer machines and external build services.
Dependency of image 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.

DRAG DROP
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.



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

2.You need to deploy Azure function resources and apps by using an automated, version-controlled CI/CD pipeline that supports declarative infrastructure deployment.

What should you use?

A. Local Git deployment

B. GitHub Actions

C. Azure Functions Core Tools

D. Azure CLI

Answer: B

Explanation:

Detailed Explanation

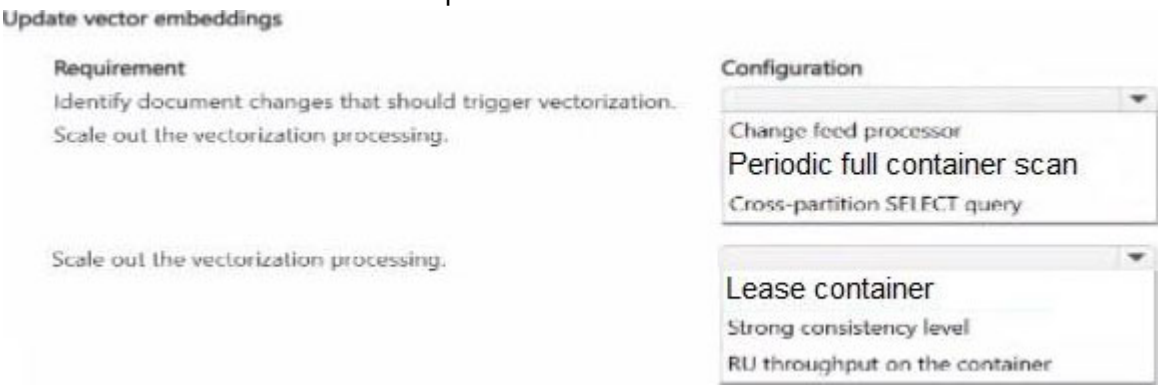
GitHub Actions is the correct orchestration mechanism among the options because the case requires an automated, version-controlled pipeline and Bicep-based declarative deployment. Microsoft documents GitHub Actions as a supported way to deploy Bicep and automate Azure resource deployment. Local Git, Azure Functions Core Tools, and Azure CLI can participate in development or scripted deployment, but they do not by themselves satisfy the stated requirement for a repository-driven, repeatable CI/CD pipeline.

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 | Deploy Bicep with GitHub Actions | Automate Function app resource deployment

3.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: B, D
Explanation:
Detailed 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

4.HOTSPOT
You need to configure vector embedding updates according to the business and technical requirements.
Which information should you use? To answer, select the appropriate option in the answer area. NOTE: Each correct selection is worth one point.



Answer:

Explanation:
Verified Answer
Identify document changes: Change feed processor. Scale out vectorization processing: Lease

container.
Detailed Explanation
The change feed processor is designed to react to inserts and updates in a monitored Cosmos DB container, which directly matches the requirement to generate embeddings for new or changed documents. Its lease container stores processing state and coordinates work across multiple workers, allowing the processing workload to scale out without duplicating ownership of change-feed ranges. A periodic full scan would consume unnecessary RUs, and neither strong consistency nor container RU throughput is the coordination mechanism for distributed change-feed workers.
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 Cosmos DB change feed processor

5.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:
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