EXAM AI-200

Developing AI Cloud Solutions on Azure Practice Question Bank

Every question, answer, and explanation from the exam simulator

109	8	2	700
QUESTIONS	QUESTION FORMATS	CASE STUDIES	PASS SCORE / 1000

Fact-checked. Every question and answer has been checked against current Microsoft Learn documentation, and every explanation links to the pages that support it.

Take the timed, scored, interactive version online: www.mightybs.com/certifications/ai-200/exam/

Study hub: www.mightybs.com/certifications/ai-200/

Updated September 29, 2026. Independent study material, not affiliated with or endorsed by Microsoft. All company names are fictitious.

Part 1: Questions

Answers are not shown here, so you can test yourself first.

- Interactive formats are printed as answer areas. For drop-down items choose one option per row; for drag and drop, match each answer choice to a row (a choice can be used more than once); for build lists, write the actions in order.
- Scenario series share one scenario: decide for each proposed solution whether it meets the goal.
- Case studies come last, each followed by the questions that refer to it.
- Check your answers in Part 2. Each question number links to its answer. On the real exam each correct selection is worth one point, and 700 of 1000 passes.

---

1 **Multiple response · Domain 1**

Your container source code is kept in a Git repository.

A new container image must be built and saved in the registry automatically each time a developer commits code to the Git repository.

You must rely as little as possible on external build infrastructure.

You need Azure Container Registry (ACR) to handle the build process automatically by using native capabilities.

Which two ACR capabilities should you use?

NOTE: Each correct selection is worth one point.

A. ACR webhook

B. ACR task that uses the Git repository as its build context

C. ACR Artifact Cache rule

D. Source code commit trigger on the ACR task

SCENARIO SERIES (4 QUESTIONS)

> **Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**
>
> **After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.**

You plan to run a container in an Azure App Service API app named api2. The source code for api2 is stored in a GitHub repository.

The container uses an API key when connecting to a backend service.

You need to keep the API key out of Git commit history.

2 **SOLUTION**

Save the API key as an application setting for api2.

Does the solution meet the goal? Yes ◯ No ◯

3 **SOLUTION**

Store the API key in Azure Key Vault and reference it from Azure App Service.

Does the solution meet the goal? Yes ◯ No ◯

4 **SOLUTION**

Place the API key in the Dockerfile.

Does the solution meet the goal? Yes ◯ No ◯

5 **SOLUTION**

Reference the API key by using Azure App Service settings.

Does the solution meet the goal? Yes ◯ No ◯

---

6 Multiple choice · Domain 1

You keep several versions of a container image in Azure Container Registry.

Production must always run one exact image build, even if image tags are changed later.

You need deployment image selection to be predictable and immutable.

What action should you take?

NOTE: Each correct selection is worth one point.

- (A) Deploy the image by referencing the latest tag.
- (B) Reference the image by its SHA digest.
- (C) Apply a production tag to the image and deploy that tag.
- (D) Run scheduled nightly rebuilds for the image.

7 Drop-down selection · Domain 1

You publish a Linux container image to App Service.

At runtime, the container needs the following environment variables:

- A non-sensitive setting named MODEL_VERSION
- A database password that must be protected

You need App Service to provide these environment variables when the container runs.

Which configuration should you use for each value?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Runtime value	App Service configuration
MODEL_VERSION value	Plain text app setting · Connection string setting · Dockerfile ENV directive
Database password value	Plain text app setting · Connection string that includes the password · Key Vault reference

Choose one option in each row.

8 Multiple response · Domain 1

You use ACR Tasks to automate container image builds.

Container images must be rebuilt for these events:

- Application updates occur.
- Content in dependent images changes.
- Scheduled rebuilds are needed.

You need ACR Tasks to perform the required automated image rebuilds.

Which three triggers should you configure?

NOTE: Each correct selection is worth one point.

A. Trigger on base image updates

B. Trigger from registry events

C. Scheduled timer trigger

D. Trigger on source code commits

E. Webhook notification trigger

9 Drag and drop · Domain 1

You are preparing a container image for production deployment.

The container image must have the following:

- A unique version tag
- Secure authentication to Azure Container Registry (ACR)
- Availability in ACR

You need the container image to satisfy the production requirements.

Which action should you perform for each requirement?

To answer, move the appropriate actions to the correct requirements. You may use each action 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 CHOICES

1. Sign in to ACR by using Microsoft Entra ID.
2. Build the image on the local host.
3. Upload the image to ACR.
4. Assign a unique tag to the image.

Need	Action to perform
Make sure the image has a version identifier.	
Permit a secure push operation to ACR.	
Make the image available in the registry for deployment.	

10 Drop-down selection · Domain 1

You deploy an App Service app from a private container image in Azure Container Registry (ACR).

App Service must authenticate to ACR before pulling the image.

The solution must not keep static registry credentials.

You need to configure secure authentication for pulling the image.

Which configurations should you use?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Configuration choice
Authenticate App Service when pulling from ACR.	Grant AcrPull to a managed identity. · Add a webhook on ACR. · Turn on the admin user.
Prevent storage of static registry credentials.	Put credentials in the Dockerfile. · Save the registry password in application settings. · Use a managed identity with a role assignment.

Choose one option in each row.

11 Drag and drop · Domain 1

You publish an API to Azure Container Apps.

The solution must provide these capabilities:

- Allow multiple application versions to be active at the same time.
- Send a specified share of incoming requests to a secondary version.

You need to define how revisions are handled.

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 CHOICES

1. Traffic splitting configuration
2. Traffic isolation
3. Single revision mode
4. Multiple revision mode

Capability	Revision configuration
Validate a new revision without changing production traffic.	
Send 20 percent of requests to a new revision.	

12 Drop-down selection · Domain 1

You plan to deploy an Azure Container Apps app named App1. App1 will use an access key to call a backend API.

The solution must keep the key outside the App1 environment and reduce maintenance effort.

You need to configure secure storage for the App1 key.

Which configurations should you use?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Configuration
Choose where to store the key for App1.	Azure Key Vault key object · Azure Key Vault secret object · Azure Container Apps secret value
Permit App1 to read the key store securely.	Service principal · Managed identity · Workload identity

Choose one option in each row.

13 Multiple response · Domain 1

You publish a new revision of an ACA app.

You need to move production traffic to the revision gradually while monitoring performance, and you must be able to roll back quickly.

Which two actions should you take?

NOTE: Each correct selection is worth one point.

- A Configure traffic splitting.
- B Restart the target revision.
- C Raise the replica count.
- D Switch to single revision mode.
- E Enable multiple revision mode.

14 Multiple choice · Domain 1

A container in an AKS cluster keeps restarting.

Pod events report probe failures, but node-level CPU and memory metrics are normal.

You need to find the cause of the repeated restarts.

What should you do first?

NOTE: Each correct selection is worth one point.

- A Review the pod events and container logs.
- B Reduce initialDelaySeconds for the container liveness probe.
- C Drain and restart the node that hosts the pod.
- D Increase the deployment replica count.

15 Multiple response · Domain 1

An ACA app reads messages from an Azure Storage queue.

The app must scale automatically from messages in one specified Azure Storage queue by using a Kubernetes Event-driven Autoscaler (KEDA) custom scale rule.

You need to set the required values for the scale rule.

Which two values should you configure?

NOTE: Each correct selection is worth one point.

- A Name of the queue
- B Scaler trigger type
- C Maximum replica count
- D Polling interval value

16 Drag and drop · Domain 1

You are deploying a web application to AKS.

The solution must:

- Scale out the application by adding pods when CPU usage peaks.
- Expose the application only inside the cluster.

You need to select 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 the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.

ANSWER CHOICES

1. Deployment resource
2. HorizontalPodAutoscaler resource
3. ClusterIP service
4. Ingress resource
5. VerticalPodAutoscaler resource

Requirement	Kubernetes resource
Add pods during high CPU demand.	
Make the application reachable only within the cluster.	

17 Multiple response · Domain 1

You develop a message-processing service that runs in Azure Container Apps and reads from an Azure Service Bus queue.

The solution must reduce costs by consuming no compute resources when the queue is empty.

You need to configure autoscaling for the service.

Which two actions should you perform?

NOTE: Each correct selection is worth one point.

A. Use HTTP ingress concurrency scaling.
B. Raise the scale rule's maximum replica limit.
C. Allow the scaling rule to terminate all active replicas.
D. Configure a Kubernetes Event-driven Autoscaler rule that monitors queue length.

18 Multiple choice · Domain 1

An AKS application depends on an internal API in the same cluster. Every pod is healthy and Ready, but calls between the services time out.

You need to decide whether the problem is Kubernetes service configuration or application code.

What should you do first?

NOTE: Each correct selection is worth one point.

- A Restart the deployment that is failing.
- B Examine the Service and Endpoints objects.
- C Add replicas to both services.
- D Check DNS resolution from your workstation.

19 Build list · Domain 1

You have an existing AKS cluster and a container image in Azure Container Registry.

You must release a new version without interrupting traffic.

You need to use a manifest file to perform a rolling update.

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.

ACTIONS

- A Check the rollout status.
- B Apply the revised manifest.
- C Scale the deployment to zero before the update.
- D Remove the existing deployment and create it again.
- E Change the image tag in the deployment manifest.

20 Drag and drop · Domain 2

You are building a Retrieval-Augmented Generation (RAG) solution with the Azure Cosmos DB for NoSQL API.

A container named Articles stores technical content, and each item has an embedding property.

The solution must support efficient similarity searches between a user query vector and saved articles.

You need to set up the database resources for semantic retrieval.

Which settings should you choose?

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 CHOICES

1. Set up a vector index.
2. Set up a composite index.
3. Save data as a numeric array.
4. Save data as a Base64 encoded string.

Need	Setting
Enable distance calculations between vectors.	
Choose the document representation for high-dimensional values.	

21 Multiple choice · Domain 2

You run several change feed processor instances to handle a high ingestion workload in Azure Cosmos DB for NoSQL.

Each processor instance must own and process a separate subset of partitions.

You need the processing workload distributed across all instances.

What should you configure?

NOTE: Each correct selection is worth one point.

A. lease container

B. vector indexing precision

C. Strong consistency

D. autoscale throughput

22 Build list · Domain 2

You are writing a Python app that uses the azure-cosmos SDK to read from Azure Cosmos DB for NoSQL.

You need to connect to an existing account and issue a SQL query against an existing container.

Which four actions should you perform in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- A Execute the query.
- B Create the CosmosClient.
- C Get the container reference.
- D Set the database name.
- E Create DatabaseProxy.

23 Drag and drop · Domain 2

You are building a .NET application that stores application data in Azure Cosmos DB for NoSQL.

The application communicates with the database account through the Azure Cosmos DB for NoSQL SDK.

The application must support these tasks:

- Open the connection with the account endpoint and key.
- Configure shared throughput.
- Run create, read, update, and delete (CRUD) item operations in a container.

You need to choose the SDK components that let the application access and manage Azure Cosmos DB for NoSQL data.

Which SDK components should you use?

To answer, move the appropriate components to the correct requirements. You may use each component 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 CHOICES

- 1 Container object
- 2 CosmosClient instance
- 3 Database object
- 4 Indexing policy

Task	SDK component
Open the connection with the account endpoint and key.	
Configure shared throughput.	
Run item CRUD operations.	

24 Multiple choice · Domain 2

You build an AI application that stores telemetry documents in Azure Cosmos DB for NoSQL.

Each document has 30 properties, but queries use only deviceId and timestamp as filters.

Write operations use more RUs than expected.

You need to lower write RU consumption without reducing query performance.

What should you do?

NOTE: Each correct selection is worth one point.

A. Add included paths for deviceId and timestamp and keep the default root path included.

B. Use strong consistency for every request.

C. Exclude unqueried properties from indexing.

D. Change to a different partition key.

25 Drop-down selection · Domain 2

You are creating an event-driven solution that responds when documents are inserted or updated in Azure Cosmos DB for NoSQL.

The app must use a change feed processor to handle new or changed documents and keep processor state between executions.

You need to set up the components for stateful change feed processing.

Which components should you configure?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Component
Read changes from the source container.	Lease container · Monitored container · Vector container
Persist continuation tokens and partition ownership.	Indexing container · Lease container · Monitored container
Handle new or changed items by using the default change feed mode.	Full scan mode · Latest version mode · Strong consistency mode

Choose one option in each row.

26 Multiple response · Domain 2

You deploy an AI application in several Azure regions.

The application must observe writes from every region within a predictable interval.

You need to choose the proper consistency level.

Which two consistency levels can meet the goal?

NOTE: Each correct selection is worth one point.

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

27 Multiple response · Domain 2

An Azure Cosmos DB for NoSQL container stores embeddings in a property named embedding.

You must return the five documents that are most similar to an input embedding.

You need to write the similarity query.

Which two query elements should you include?

Each correct answer presents part of the solution. Choose two.
NOTE: Each correct selection is worth one point.

- A Use OFFSET pagination.
- B Select DISTINCT embedding.
- C Use Strong consistency.
- D GROUP BY the embedding value.
- E Constrain the result count.
- F Use only an equality predicate.
- G Order results by vector distance.
- H Define a vector index policy on the container.

28 Drag and drop · Domain 2

A Python API uses a SQL statement to retrieve a document from Azure Database for PostgreSQL. The document ID comes from user input, and the current code inserts that value directly into the SQL text.

You need to reduce the chance of SQL injection when the statement runs.

How should you change the implementation?

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 CHOICES

1. Use a parameterized query.
2. Pass the ID as an argument.
3. Escape quotation marks in the ID.
4. Supply the parameter tuple to the SDK method.

Need	Configuration
Change the SQL statement structure.	
Bind the user-provided value.	
Run the statement.	

29 Drop-down selection · Domain 2

You are building an application whose Python API runs similarity queries against Azure Database for PostgreSQL. The app opens a new database connection for each request.

At peak traffic, the app sometimes cannot open new sessions, and logs show that the connection limit has been reached.

You need to reduce connection setup overhead and increase throughput.

How should you complete the configuration?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Need	Configuration
Connection approach	Enable PgBouncer. · Scale the compute tier. · Enable automatic vacuum. · Increase connection timeout.
Port for connections	5432 · 6432 · 8080 · 8443

Choose one option in each row.

30 Multiple response · Domain 2

You are designing an Azure Database for PostgreSQL table that will support semantic search. Queries often filter by the created_at column.

The table must support vector similarity search and dependable date filtering.

You need the table design to satisfy both requirements.

Which two actions should you perform?

NOTE: Each correct selection is worth one point.

- (A) Store created_at as free-form text.
- (B) Store embeddings in a pgvector vector column such as vector(n).
- (C) Store embeddings in a varchar column.
- (D) Use a typed timestamp column for created_at.

31 Build list · Domain 2

A recommendation system keeps 3 million embeddings in Azure Database for PostgreSQL.

During peak usage, P95 latency for similarity queries rises and the cache hit ratio drops sharply.

You think the vector index working set no longer fits in memory, causing more disk reads.

You need to scale server resources and confirm the effect.

Which four actions should you perform in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- (A) Run a fixed workload before peak hours to record baseline P95 latency and capture EXPLAIN ANALYZE for the vector query.
- (B) Check peak-hour memory pressure and cache hit ratio, then compare them with the baseline.
- (C) Move to a tier that provides more memory per vCore.
- (D) Run the same benchmark again and compare P95 and P99 latency.
- (E) Turn off autovacuum.
- (F) Increase the embedding dimensionality.

32 Drag and drop · Domain 2

A semantic search app queries Azure Database for PostgreSQL. A table stores document embeddings and metadata in these columns:

- embedding (pgvector)
- department
- created_at

The app must return the five closest documents for a query embedding, restricted to the finance department.

You need to build semantic retrieval with a metadata filter.

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.
NOTE: Each correct selection is worth one point.

ANSWER CHOICES

1. WHERE department ILIKE 'fin%'
2. ORDER BY created_at DESC LIMIT 5
3. WHERE department = 'Finance'
4. ORDER BY embedding <=> :query_embedding LIMIT 5

Need	Query part
Restrict rows to Finance.	
Sort by similarity and return five rows.	

33 Multiple choice · Domain 2

An application searches 5 million embeddings in Azure Database for PostgreSQL with pgvector. Queries frequently filter by department before ordering by cosine distance.

P95 latency for vector similarity queries is above the SLA, and monitoring shows sustained high CPU during execution.

You need to lower P95 latency for filtered vector similarity searches.

What should you do?

A. Add B-tree indexes on commonly filtered metadata columns.
B. Add a pgvector index on the embedding column.
C. Turn off autovacuum.
D. Increase the number of embedding dimensions.
E. Remove the similarity ORDER BY clause.
F. Replace parameterized queries with string concatenation.
G. Store embeddings as JSON.
H. Increase statement timeout.

34 Multiple response · Domain 2

You are creating an AI search API that stores semantic search results in Redis.

Results must stay cached for 10 minutes, and cached entries must not be served after the source data changes.

You need a cache-aside approach that preserves consistency.

Which two actions should you perform?

NOTE: Each correct selection is worth one point.

- A Set a 10-minute Time to Live on each key.
- B Use sliding expiration based on key access.
- C Enable cache notifications for keyspace events.
- D Delete related cache keys when source data changes.

35 Multiple choice · Domain 2

You are improving an AI inference API that relies on Redis caching.

You must limit the chance of returning stale data while keeping cache maintenance effort low.

You need to choose the caching strategy that meets the requirements.

What should you do?

- A Increase memory allocation.
- B Enable keyspace notifications.
- C Reset expiration after every read.
- D Use the allkeys-lru eviction policy.
- E Turn off expiration for cache keys.
- F Tune Time to Live values for cache keys.
- G Purge cache keys after every request.
- H Invalidate cache entries when source data changes.

36 Multiple choice · Domain 2

Redis stores embeddings by using keys in the form doc:{id}. Some embeddings are read often, while others are rarely accessed.

Memory usage stays well below the configured memory limit.

You need a caching approach that keeps frequently read embeddings in memory and removes embeddings that are rarely read.

What should you use?

NOTE: Each correct selection is worth one point.

- A allkeys-lru eviction
- B volatile-ttl eviction
- C EXPIRE command
- D fixed time-window expiration

37 Yes or No · Domain 2

You are adding semantic search to a chatbot, and document embeddings are stored in Redis.

Review this Python code that connects to Redis and stores an embedding value:

```
01  import numpy as np
02  from redis import Redis
03  embedding = np.array([0.12, 0.98, 0.44], dtype=np.float32).tobytes()
04  r = Redis(
05      host="myredis.eastus.redis.azure.net",
06      port=10000,
07      password="accesskey",
08      ssl=True
09  )
10  r.hset("doc:1", mapping={"embedding": embedding})
11  r.expire("doc:1", 600)
```

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

Statement	Yes	No
The embedding value is saved in a Redis hash field.	○	○
The code configures similarity search for the embedding field.	○	○
The key expires after 10 minutes.	○	○

38 Multiple choice · Domain 2

You are implementing the expiration policy for AI-generated summaries cached in Redis.

You need each AI summary removed five minutes after it is created, no matter how often it is read.

What should you do?

NOTE: Each correct selection is worth one point.

A. Set absolute expiration on each cache key.

B. Set sliding expiration on each cache key.

C. Run the PERSIST command after each successful read.

D. Use allkeys-lru as the maxmemory-policy.

---

39 Drop-down selection · Domain 2

You are developing a Retrieval-Augmented Generation (RAG) solution for an organization.

Redis caches AI responses and embedding vectors.

The solution must satisfy these requirements:

- AI responses must expire exactly 24 hours after being cached.
- Cached embeddings must stay synchronized with the current source data.

You need to configure Redis to satisfy the requirements.

Which actions should you perform?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Action
AI responses must expire exactly 24 hours after being cached.	Configure allkeys-lru eviction. · Reset expiration on every read. · Set a Time to Live (TTL) on each key.
Cached embeddings must stay synchronized with the current source data.	Configure volatile-lru eviction. · Delete related keys when the document changes. · Increase maximum memory allocation.

Choose one option in each row.

40 Drop-down selection · Domain 2

You are adding semantic retrieval by using Redis.

The solution must provide efficient vector similarity search for AI retrieval at scale.

You need to choose suitable vector schema settings.

Which configuration values should you select?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Schema setting	Value
Embedding field type	Numeric · Tag · Vector
Vector index setting	Flat index · HNSW index · Metadata index

Choose one option in each row.

41 Multiple choice · Domain 2

You are building semantic retrieval for a chatbot.

Embeddings already exist in Redis, but vector similarity queries return no matches.

You need to fix the vector similarity search problem.

What should you do?

- (A) Create an HNSW vector index for the embedding field.
- (B) Create a FLAT vector index for the embedding field
- (C) Configure Redis Database Backup (RDB) persistence.
- (D) Turn on geo-replication.
- (E) Set a 24-hour Time to Live (TTL) on embedding keys.
- (F) Configure an index that includes only metadata.
- (G) Increase maximum memory allocation.
- (H) Reset expiration after each read.

SCENARIO SERIES (4 QUESTIONS)

> **Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**
>
> **After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.**

You are getting an Azure Function app ready for production. The same app will be deployed to several environments.

Configuration must vary by environment, and secrets must not be committed to source control.

You need to build the implementation.

42 **SOLUTION**

Place connection strings in app settings.

Does the solution meet the goal? Yes ◯ No ◯

43 **SOLUTION**

Place production secrets in local.settings.json.

Does the solution meet the goal? Yes ◯ No ◯

44 **SOLUTION**

Reference Key Vault secrets from app settings.

Does the solution meet the goal? Yes ◯ No ◯

45 **SOLUTION**

Store environment-specific values in App Configuration and let the function app access them with managed identity.

Does the solution meet the goal? Yes ◯ No ◯

46 Build list · Domain 3

A new Azure subscription is available. You are setting up a custom Event Grid flow for AI inference notifications.

You need to create the Event Grid resources so high-confidence events are routed to a downstream processor.

Which three actions should you complete in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- A Create an Event Grid domain.
- B Register the Event Grid resource provider.
- C Create an Event Grid partner topic.
- D Create a custom Event Grid topic.
- E Create an event subscription.

47 Multiple choice · Domain 3

An Event Grid subscription invokes an Azure Function.

You need to make sure events are not lost when the endpoint returns HTTP 400.

What should you configure?

- A Set a retry policy.
- B Use optimistic batching.
- C Set a dead-letter destination.
- D Use asynchronous handshake validation.

48 Drop-down selection · Domain 3

You set up an Event Grid event subscription that sends AI file-upload events to an Azure Function endpoint.

Events must be delivered only when both conditions are true:

The event path begins with /uploads/ai/.

The data.fileType payload property equals "pdf."

If the subscriber endpoint cannot accept an event after repeated retries, undelivered events must be saved for later investigation and reprocessing.

You need to choose the filtering and reliability configuration.

Which settings should you use?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Requirement	Setting
Condition on the path prefix	Subject filtering · Advanced filtering · Service Bus session
Condition on the payload property	Subject filtering · Advanced filtering · Event type filtering
Handling events that are not delivered	Subject filtering · Duplicate detection · Dead-letter destination

Choose one option in each row.

49 Multiple response · Domain 3

You are planning a Service Bus-based messaging design for AI document processing.

A single published message must fan out to multiple independent consumers, and every consumer must get its own copy.

Which two Service Bus entities are required?

NOTE: Each correct selection is worth one point.

- A Service Bus dead-letter queue
- B Service Bus message session
- C Service Bus queue
- D Service Bus subscription
- E Service Bus topic

50 Drop-down selection · Domain 3

You are preparing an Event Grid subscription for AI inference events.

The solution must meet these needs:

Evaluate fields in the event payload.

Keep events that cannot be delivered.

Cap the number of delivery retries.

You need to set the Event Grid options.

Which configurations should you select?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Configuration
Evaluate payload fields.	Advanced filtering · Subject filtering · Service Bus session
Keep undelivered events.	Dead-letter destination · Endpoint validation · Duplicate detection
Cap retry attempts.	Maximum delivery attempts · Message sessions · Auto-complete

Choose one option in each row.

51 Build list · Domain 3

You are creating and configuring a Service Bus processor for AI batch workloads.

The processor must connect to an existing queue, attach message and error handling logic, and then start receiving messages.

You need to initialize the Service Bus processor for message and error handling.

Which four actions should you perform in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- A Start processing messages.
- B Create the Service Bus client.
- C Create a processor for the queue.
- D Register the message and error handlers.
- E Dead-letter messages that fail.

52 Multiple choice · Domain 3

An Azure Service Bus queue handles AI enrichment work items.

Some messages keep failing because their payloads are malformed.

You need to keep those repeatedly failing messages from blocking valid ones while making them available for separate review.

Which message action should you use?

NOTE: Each correct selection is worth one point.

A abandon the message

B complete the message

C dead-letter the message

D defer the message

53 Multiple choice · Domain 3

Your Azure Service Bus message processor depends on a call to an external API.

When the API is temporarily down, processing must be postponed without increasing the delivery count.

You need a message action that lets you handle the message later, after the API becomes available, while keeping it retrievable.

Which message action should you use?

NOTE: Each correct selection is worth one point.

A Abandon

B Complete

C Dead-letter

D Defer

54 Multiple choice · Domain 3

You send custom AI inference events to an Event Grid topic.

Downstream services should receive an event only when confidenceScore is greater than 0.80.

You need to evaluate a numeric value from the event payload.

Which filter type should you configure?

NOTE: Each correct selection is worth one point.

A subject filter

B advanced filter

C event type filter

55 Drag and drop · Domain 3

You are designing a back-end pipeline for AI inference requests.

The pipeline must support these behaviors:

Send one message to multiple subscribers.

Process messages in first-in, first-out (FIFO) order.

Separate messages that fail.

You need to choose the right Service Bus entities.

How should the Service Bus entities be configured?

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 CHOICES

1. Service Bus queue
2. Service Bus topic
3. dead-letter queue
4. subscription rule
5. subscription filter
6. auto-forward target

Need	Configuration
Send one message to multiple subscribers.	
Process messages in FIFO order.	
Separate failed messages.	

56 Yes or No · Domain 3

You are examining message-processing code in a back-end worker service.

Review this Service Bus processor code:

```
01  await using var client = new ServiceBusClient(connectionString);
02  ServiceBusProcessor processor = client.CreateProcessor("orders",
03      new ServiceBusProcessorOptions { AutoCompleteMessages = true });
04  processor.ProcessMessageAsync += MessageHandler;
05  processor.ProcessErrorAsync += ErrorHandler;
06  await processor.StartProcessingAsync();
```

Statement	Yes	No
The processor receives messages from the orders queue.	○	○
If MessageHandler throws an exception, the processor completes the message automatically.	○	○
Processing errors are sent to ErrorHandler.	○	○

57 Multiple choice · Domain 3

You deploy a production Azure Function app that uses an Azure SQL Database.

The solution must provide these capabilities:

Keep secrets out of source control.

Allow secret rotation without redeploying the function app.

Maintain availability while credentials are updated.

You need to set up secure, maintainable secret management.

What should you configure?

NOTE: Each correct selection is worth one point.

A. Parameter file committed to source control

B. Environment variables stored in local.settings.json

C. App settings that reference Key Vault

D. Connection string hard-coded in the startup class

58 Multiple response · Domain 3

You are designing an Azure Function app for large image uploads that users submit through an HTTP endpoint.

The solution must meet these requirements:

Decouple image processing from the upload request so clients do not time out.

Retry failed background processing automatically.

Scale image-processing workers independently from incoming HTTP upload traffic.

You need to design a reliable, scalable asynchronous processing pattern.

Which two actions should you implement?

NOTE: Each correct selection is worth one point.

A. Use the Queue Storage trigger built-in retry behavior.

B. Write upload metadata to Azure Storage.

C. Create a queue-triggered function for image processing.

D. Process the image in the HTTP-triggered function.

59 Multiple choice · Domain 3

An HTTP-triggered Azure Function needs to read a JSON configuration file from Azure Blob Storage.

You need to bring the configuration file contents into the function without writing Azure Storage SDK client code.

What should you use?

NOTE: Each correct selection is worth one point.

- A HTTP trigger
- B Input binding
- C Output binding
- D Timer trigger

60 Multiple choice · Domain 3

You are designing an Azure Function app that provides a public API.

The solution must meet these needs:

Return validation results synchronously.

Support Microsoft Entra ID authentication.

Scale automatically as load changes.

Prevent duplicate processing.

You need to choose a trigger.

Which trigger should you use?

NOTE: Each correct selection is worth one point.

- A Event Grid trigger
- B Azure Queue Storage trigger
- C HTTP trigger
- D Service Bus topic trigger

61 Multiple choice · Domain 3

A backend process must start when a blob is uploaded to Azure Storage.

The solution must meet these requirements:

Avoid polling.

Keep latency low.

Start automatically at upload time.

You need processing to begin immediately after a file upload.

Which trigger should you implement?

NOTE: Each correct selection is worth one point.

A. Event Grid trigger
B. HTTP trigger
C. Queue Storage trigger
D. Timer trigger

62 Multiple response · Domain 3

You are designing an Azure Function to process orders from a new Azure Service Bus queue.

You need to drop duplicate messages with the same MessageId during a detection window and preserve failed messages for analysis.

Which two actions should you implement?

NOTE: Each correct selection is worth one point.

A. Turn on Service Bus sessions.
B. Add dead-letter handling.
C. Use an HTTP trigger to process orders.
D. Turn on duplicate detection for the queue.

63 Yes or No · Domain 3

You are examining an Azure Function app that handles incoming order requests for a company.

The function must meet these requirements:

Accept order submissions from an external client app.

Require controlled access for security.

Return a response that contains the processed request payload.

Review this code segment:

```
01  [FunctionName("ProcessOrder")]
02  public static async Task<IActionResult> Run(
03    [HttpTrigger(AuthorizationLevel.Function, "post")] HttpRequest req,
04    ILogger log)
05  {
06    string result = await new StreamReader(req.Body).ReadToEndAsync();
07    return new OkObjectResult(result);
08  }
```

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

Statement	Yes	No
Invoking the function requires a function key.	◯	◯
The function accepts HTTP GET requests.	◯	◯
The response contains the request body.	◯	◯
The function checks the request body before returning.	◯	◯

64 Drag and drop · Domain 3

You are planning Azure Functions for three separate backend workloads.

Each workload needs one of these dispatch models:

Send an immediate response back to a client.

Handle background work from a queue.

Execute code on a fixed schedule.

You need to match each requirement to a trigger.

Which triggers should you choose?

To answer, move the appropriate triggers to the correct requirements. You may use each trigger 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 CHOICES

1. Event Grid trigger
2. HTTP trigger
3. Queue Storage trigger
4. Timer trigger
5. Blob Storage trigger

Requirement	Trigger type
Send an immediate response back to a client.	
Handle background work from a queue.	
Execute code on a fixed schedule.	

65 Build list · Domain 3

You are deploying an Azure Function app that uses managed identity to read secrets from Key Vault.

Infrastructure, identity, and secret references must be ready before the function code is deployed.

You need to perform a secure deployment of the function app.

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

A. Enable a managed identity for the function app and grant it access to Key Vault secrets.
B. Configure application settings with Key Vault references.
C. Create the storage account.
D. Create the function app.
E. Deploy the function code.

66 Multiple response · Domain 4

You are building an AI API that reads connection strings and API keys from Azure Key Vault.

You must configure the following security behavior:

- The API must authenticate to Key Vault without saving credentials in application configuration files.
- The API identity must receive only the permissions required to read secrets.
- The design must reduce the blast radius if an identity or credential is compromised.

You need to design secure Key Vault access for the API.

Which two actions should you take?

NOTE: Each correct selection is worth one point.

A. Use a system-assigned managed identity.

B. Store a secret value in Azure App Configuration.

C. Grant the Key Vault Secrets User role at the vault scope.

D. Assign the Key Vault Administrator role at the subscription scope.

---

67 Drop-down selection · Domain 4

You use the Python SDK to build a containerized AI app that reads a runtime setting from an Azure App Configuration store. In Azure, the app runs under a managed identity.

The same code must also run in a local development environment.

You need to complete the code that reads the setting.

Which choices complete the code?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

```
var1 = [A]()
var2 = AzureAppConfigurationClient(base_url=endpoint, credential=var1)
var3 = var2.[B](key="FeatureX", label="production")
```

[A] ClientSecretCredential · DefaultAzureCredential · ManagedIdentityCredential

[B] list_labels · get_configuration_setting · list_configuration_settings

68 Multiple choice · Domain 4

You are creating an AI application whose behavior depends on a dynamically retrieved value for a specific key in Azure App Configuration.

The application will be deployed to test, staging, and production.

You need to assign a different value for the same key in each environment.

Which Azure App Configuration feature should you use?

- A Labels
- B Key prefixes
- C Content types
- D Resource tags

69 Drag and drop · Domain 4

You are setting up an AI application hosted on Azure.

The application must keep secrets separate from configuration values that are not sensitive.

You need to choose the Azure service for each configuration item.

Where should each item be stored?

To answer, move the appropriate Azure services to the correct configuration items. You may use each Azure service 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 CHOICES

1. Azure App Configuration
2. Azure Key Vault
3. Azure Storage

Item to store	Service to use
REST API endpoint URL	
Beta model feature flag	
Database password value	
Storage account access key	

70 Yes or No · Domain 4

You are building an AI application that uses the Azure SDK for Python to read database credentials from Key Vault.

The application must support secret rotation and authenticate securely.

Review this Key Vault secret retrieval code:

```
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.

Statement	Yes	No
The code reads the current latest version of dbPassword.	○	○
When the app runs on an Azure host with managed identity enabled, the code can authenticate to Key Vault without storing client secrets in the application.	○	○
If dbPassword is rotated and a new version is created, later calls made by this code retrieve the new version automatically.	○	○

71 Multiple response · Domain 4

An AI platform uses Azure App Configuration to manage feature flags and endpoint routing.

The platform currently stores secrets with configuration data and does not support dynamic refresh.

The new design must refresh configuration dynamically and must not store secrets in App Configuration.

You need to provide secure dynamic configuration management for the platform.

Which three actions should you take?

NOTE: Each correct selection is worth one point.

- (A) Enable configuration refresh with a polling interval.
- (B) Put configuration values in environment variables.
- (C) Store API keys in Key Vault.
- (D) Use labels to separate environments.
- (E) Use managed identity for App Configuration access.

72 Multiple choice · Domain 4

An AI application reads values from Azure App Configuration and currently uses static configuration.

You need to add dynamic configuration updates while keeping request latency low.

What should you do?

NOTE: Each correct selection is worth one point.

- A Cache configuration values and refresh them on an interval.
- B Set configuration values as environment variables.
- C Read configuration values from App Configuration for every request.
- D Store configuration values in Key Vault.

73 Drag and drop · Domain 4

You are evaluating how an AI application retrieves credentials from Key Vault.

You need to assess the security result of each implementation approach.

Which result applies to each approach?

To answer, move the appropriate outcomes to the correct implementations. You may use each outcome 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 CHOICES

1. Enables automatic rotation
2. Avoids storing credentials
3. Needs a manual change after rotation
4. Creates credential exposure

Approach	Result
Request a secret without a version.	
Request a secret by using a version identifier.	
Authenticate by using managed identity.	
Use a client secret saved in environment variables.	

74 Multiple response · Domain 4

You are developing an AI API that runs in ACA and needs database credentials stored in Key Vault. The vault uses Azure RBAC for access control.

You need a secure secret access design that prevents credential exposure and uses rotated secrets without redeploying the container.

Which three actions should you take?

NOTE: Each correct selection is worth one point.

- (A) Enable a system-assigned managed identity.
- (B) Create a Key Vault RBAC role assignment.
- (C) Create a Key Vault access policy.
- (D) Read the secret at runtime by using the SDK.
- (E) Export the secret as part of deployment.

---

75 Build list · Domain 4

An AI service is being deployed to ACA.

The service must use managed identity to read secrets securely from Key Vault.

You need to configure the secure access path.

Which three actions should you perform in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- (A) Enable a system-assigned managed identity.
- (B) Create an access policy that includes the recover permission.
- (C) Grant Key Vault Secrets User to the managed identity.
- (D) Read the secret by using the SDK.

76 Multiple response · Domain 4

An AI application connects to a database whose credential rotates every 30 days.

A manual application update is currently required whenever the credential rotates.

You need the application to use the newest secret version without manual updates.

Which two actions should you take?

NOTE: Each correct selection is worth one point.

A. Configure secret autorotation by using Event Grid and Azure Functions.

B. Retrieve secrets with a version identifier.

C. Retrieve secrets without a version identifier.

D. Cache the secret for 30 days.

77 Multiple response · Domain 4

You are diagnosing intermittent connectivity issues in a production Azure OpenAI service. Telemetry is written to a Log Analytics workspace.

You must write a KQL query for the AppRequests table. The query must include requests from the last 30 minutes that have a failed status.

The result must show the total failure count for each unique operation.

You need to choose the KQL operators that filter the time window and aggregate failure counts.

Which two operators should you use?

NOTE: Each correct selection is worth one point.

A. distinct

B. project

C. summarize

D. where

78 Build list · Domain 4

You are investigating high latency in an AI search application that handles millions of requests each day. Telemetry is stored in Azure Monitor Logs.

You must build a KQL query that correlates data from the AppRequests table and the AppDependencies table. The query must:

- Include only data from the last 24 hours.
- Include only failed requests.
- Calculate average dependency duration grouped by operation.

The query must improve performance by reducing the initial data scan.

You need to build the query.

Which five actions should you perform in order?

To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

A Apply the time filter.

B Filter to failed requests.

C Start with the requests table.

D Join to the dependencies table.

E Summarize average dependency duration by operation.

79 Drop-down selection · Domain 4

A company uses Azure Monitor Application Insights to collect telemetry for incoming requests and dependency calls.

You must find failed requests from the past hour and compute the average duration of their dependency calls grouped by operation name.

You need to analyze the telemetry in Application Insights.

Which operators should complete the query?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

```
requests
| [A] timestamp > ago(1h) and success == false
| join kind=inner (
    dependencies
    | [B] timestamp > ago(1h)
    | project operation_Id, depDuration = duration
) on operation_Id
| [C] avgDepDuration = avg(depDuration) by operation_Name
```

[A] extend · project · where

[B] distinct · render · where

[C] extend · project · summarize

80 Drag and drop · Domain 4

A Python API running in ACA must export distributed traces to Azure Monitor.

The API creates spans, but no traces are visible in Azure Monitor.

You need to configure the OpenTelemetry SDK pipeline so traces are sent to Azure Monitor.

What should you configure?

To answer, move the appropriate actions to the correct requirements. You may use each action 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 CHOICES

1. Turn on log sampling.
2. Call tracer.start_as_current_span().
3. Set up the application's TracerProvider for tracing.
4. Add a span processor that sends spans to the exporter.
5. Instantiate the Azure Monitor trace exporter.

Need	Action
Register the global tracer provider.	
Send traces to Azure Monitor.	
Attach the exporter to the provider.	
Create spans in application code.	

81 Drag and drop · Domain 4

You are setting sampling behavior for a distributed application that exports traces to Azure Monitor.

The solution must:

- Honor upstream sampling decisions across distributed traces.
- Record every span during local testing.
- Sample 10 percent of traces in production.

You need to select the correct sampling configuration for each requirement.

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 CHOICES

1. AlwaysOnSampler
2. ParentBased
3. TraceIdRatioBased(0.1)
4. BatchSpanProcessor
5. AzureMonitorTraceExporter

Requirement	Configuration
Honor parent sampling choices.	
Capture every span in local tests.	
Sample 10 percent of production traces.	

82 Drop-down selection · Domain 4

You are using the requests table in Azure Monitor to troubleshoot latency.

You need to compute average request duration by cloud role name and restrict the results to the last 30 minutes.

Which code elements should you choose?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

```
requests
| [A]   timestamp > ago(30m)
| [B] avg(duration) by cloud_RoleName
```

[A] distinct · render · where

[B] extend · project · summarize

---

83 Yes or No · Domain 4

You are examining Python tracing setup for an application that must export distributed traces to Azure Monitor.

Review this OpenTelemetry tracing configuration:

```
01 from opentelemetry import trace
02 from opentelemetry.sdk.trace import TracerProvider
03 from opentelemetry.sdk.trace.export import BatchSpanProcessor
04 from azure.monitor.opentelemetry.exporter import AzureMonitorTraceExporter
05
06 trace.set_tracer_provider(TracerProvider())
07 tracer = trace.get_tracer(__name__)
08
09 exporter = AzureMonitorTraceExporter(connection_string="<conn>")
10 span_processor = BatchSpanProcessor(exporter)
11 trace.get_tracer_provider().add_span_processor(span_processor)
```

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

Statement	Yes	No
The tracer provider is configured before any spans are created.	○	○
The setup exports traces synchronously.	○	○
The setup enables trace export to Azure Monitor.	○	○

84 Multiple response · Domain 4

You use Azure Monitor Application Insights to capture dependency telemetry.

You must:

- Correlate failed requests with dependency calls from the last hour.
- Compute average dependency duration for each operation.

You need to build the KQL query with the fewest required statements.

Which three operators should you use?

NOTE: Each correct selection is worth one point.

A extend

B join

C distinct

D summarize

E where

SCENARIO SERIES (4 QUESTIONS)

> **Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**
>
> **After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.**

You use Azure Monitor Application Insights to collect request telemetry for a production API.

A teammate suggests the following query to find high-latency requests:

```
requests
| where timestamp > ago(30m)
| where duration > 2000
| summarize avg(duration) by operation_Name
```

You need to find requests from the past 30 minutes whose duration is greater than two seconds.

85 **SOLUTION**

The query limits results to the past 30 minutes.

Does the solution meet the goal? Yes ◯ No ◯

86 **SOLUTION**

The query includes only requests with duration greater than two seconds.

Does the solution meet the goal? Yes ◯ No ◯

87 **SOLUTION**

The query returns each individual slow request row.

Does the solution meet the goal? Yes ◯ No ◯

88 **SOLUTION**

The query returns average request duration by operation.

Does the solution meet the goal? Yes ◯ No ◯

89 Yes or No · Domain 4

A Python web API uses OpenTelemetry tracing. The call_downstream_service function sends an outbound HTTP request by using the requests library.

This is the only OpenTelemetry configuration in the application:

```
01  from opentelemetry import trace
02  tracer = trace.get_tracer(__name__)
03  def process_request():
04    with tracer.start_as_current_span("process_request"):
05        call_downstream_service()
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Statement	Yes	No
The code starts a span when each request is processed.	◯	◯
The code automatically sends trace context to downstream HTTP services.	◯	◯
Additional instrumentation is required for end-to-end trace correlation.	◯	◯

Case study: Northwind Traders

Questions 90 to 99 refer to this case study.

Background

Northwind Traders helps specialty grocers and regional food retailers turn sales signals into AI-based demand forecasts and product recommendations. To modernize the offering, the company is placing the full solution on Microsoft Azure.

After transaction feeds arrive, the platform produces embeddings for semantic retrieval, evaluates vector similarity, and serves recommendations from microservices that run in containers. Python and Azure SDKs are used by developers, while operations staff owns orchestration, elastic scale, observability, and security for the container environment.

Strict targets for security, scalability, and performance apply to the solution.

Current environment

Solution components

- A containerized Python application exposes the Recommendation engine through an API.
- pgvector on Azure Database for PostgreSQL is used to hold the embeddings.
- Metadata filters are applied with vector similarity search during semantic retrieval.
- The caching tier is implemented with Azure Managed Redis.
- Azure Container Apps (ACA) runs both the front-end tier and the API tier.
- Azure Kubernetes Service (AKS) hosts the workloads that perform batch model retraining.

Build and release

- Azure Container Registry (ACR) is the repository for container images.
- When code is committed, ACR Tasks runs image builds as part of CI/CD.
- Revision management is available in the ACA environments.
- Kubernetes manifest files kept in Git are used to deploy AKS workloads.

Observability

- Azure Monitor is the destination for collected log data.
- Troubleshooting includes reviewing Kubernetes events together with container logs.
- To understand latency surges, developers analyze the data with KQL queries.

Business requirements

- **Customer experience:** Even when seasonal traffic rises unpredictably, users must receive recommendations through a smooth experience with low latency.
- **Operational cost efficiency:** Compute spending must be controlled by releasing idle resources and preventing scaling from creating excessive costs.
- **Data integrity and freshness:** The solution must ensure that recommendations use current pricing and catalog metadata at all times to avoid dissatisfied customers.
- **Security and compliance:** A Zero Trust model must be followed by removing long-lived credentials and managing every sensitive secret from a central location.
- **Worldwide scale:** Millions of new product embeddings must be ingested quickly each day without harming query performance for retailers already using the platform

Technical requirements

- **Latency objective:** During peak load, semantic search must complete with latency below 200 milliseconds.

- **Vector workload design:** Embeddings must use pgvector, and compute overhead must be reduced through metadata filtering. For vector workloads, compute and memory must be configured so high-dimensional indexes stay in RAM and mathematical operations remain efficient. Products must satisfy mandatory metadata constraints before vector similarity calculations are performed against them.
- **Connection handling:** High concurrency with minimal latency is required for database connections, so connection optimization must be implemented.
- **Ingestion sequencing:** Secondary indexes must not be applied until after the bulk embedding load is complete, which ensures maximum ingestion throughput.
- **Cache lifecycle:** A 10-minute automatic expiration is required for Redis cache entries. Cache entries must also be invalidated reactively when metadata is updated.
- **Identity model:** Every service-to-service and service-to-database authentication flow must use managed identities. Configuration files must never contain plain-text credentials.
- **Secret governance:** A central location must store all secrets. Automatic rotation must occur by using a centralized lifecycle policy.
- **Scaling model:** Event-driven scaling must be implemented with Kubernetes event-driven autoscaling (KEDA). HTTP traffic must drive scaling for the Recommendation API, while queue length must drive batch job scaling and those jobs must support scale-to-zero.
- **CI/CD controls:** Azure Container Registry must store every image. Source code commits must trigger automated image builds through ACR Tasks.
- **Monitoring workflow:** Performance telemetry and microservice connectivity failures must be investigated with KQL. Logs and events must be inspected when AKS and ACA issues are being troubleshot.

90 Multiple response · Domain 1

CASE STUDY: NORTHWIND TRADERS

You need to build the container image for the product recommendation engine.

Which two actions are required?

Each correct answer presents part of the solution. Choose two.
NOTE: Each correct selection is worth one point.

A. Push the revised code to the source repository.
B. Set up a base image update trigger for the ACR Task.
C. Release the container image to Azure Kubernetes Service.
D. Use a GitHub Action to rebuild the application after each push.
E. Set up an Azure DevOps pipeline that uses a source repository context.
F. Create and configure an ACR Task that uses a source repository context.

91 Drop-down selection · Domain 1

CASE STUDY: NORTHWIND TRADERS

You need to deploy a workload that performs batch retraining.

How should you complete the scaling settings?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Scaling requirement	Setting to use
Scale from the number of queued messages.	CPU utilization scale rule · Azure Service Bus scale rule · HTTP concurrency scale rule
Remove workload instances when the queue has no messages.	Minimum replicas set to 0 · Minimum replicas set to 1
Limit burst scaling to a controlled value.	Maximum replicas set to 10 · Minimum replicas set to 10

Choose one option in each row.

92 Drag and drop · Domain 1

CASE STUDY: NORTHWIND TRADERS

You need to troubleshoot communication failures between microservices running in AKS.

Which troubleshooting action should you use for each issue?

To answer, move the appropriate action to the correct troubleshooting scenario. You may use each action 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 CHOICES

1. Review container logs.
2. Review Pod descriptions.
3. Review Kubernetes events.
4. Review Kubernetes service endpoints.

Troubleshooting case	Recommended action
Confirm whether a Service's selector matches any running Pods.	
Find the application error that makes a container crash and restart.	
Check the readiness probe settings (path, port, and thresholds) of a Pod that never becomes ready.	

93 Multiple choice · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to increase throughput for concurrent requests from the application to PostgreSQL.

What should you implement?

NOTE: Each correct selection is worth one point.

A. Configure read replicas.
B. Raise shared_buffers.
C. Raise max_connections.
D. Add connection pooling.

94 Drop-down selection · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to configure resources for the Azure Database for PostgreSQL instance.

How should you complete the setup to satisfy the business and technical requirements?

To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Need	Configuration
Satisfy the 200-ms semantic search latency target.	Scale compute vCores. · Raise max_connections. · Allocate more memory.
Keep high-dimensional pgvector indexes resident in memory.	Enable a read replica. · Raise max_connections. · Allocate more memory.
Handle ongoing ingestion of transaction-derived embeddings.	Enable storage autoscale. · Scale compute vCores. · Increase backup retention.

Choose one option in each row.

95 Drop-down selection · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to optimize secure database access from the containerized Recommendation API.

How should you configure the app?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Need	Configuration
Meet the authentication policy for database access.	Use a connection pooling library. · Use managed identity authentication. · Store credentials in environment variables.
Serve high-concurrency requests with low latency.	Use an unlimited pool size. · Use a connection pooling library. · Increase the database max_connections parameter.
Maintain database stability during traffic spikes.	Set a maximum pool size. · Enable asynchronous query execution. · Use a retry policy with exponential backoff.

Choose one option in each row.

96 Multiple choice · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to tune vector search queries according to the technical requirements.

What should you do?

NOTE: Each correct selection is worth one point.

- **A** Raise shared_buffers.
- **B** Raise max_connections.
- **C** Create a B-tree index on metadata filter columns.
- **D** Create an IVFFlat index on the embedding column.

97 Build list · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to build the semantic retrieval workflow for the recommendation engine so Northwind Traders meets its technical and performance requirements.

Which four actions should you perform in order?

To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

ACTIONS

- A Create a B-tree index on the metadata filter columns.
- B Define a table schema that includes vector and metadata columns.
- C Increase Redis memory for the caching layer.
- D Load embedding vectors with the related product metadata.
- E Run a similarity search with a WHERE clause and the <=> operator.
- F Configure a Hierarchical Navigable Small World (HNSW) index on the metadata columns.

98 Drag and drop · Domain 2

CASE STUDY: NORTHWIND TRADERS

You need to set up Redis integration for the Recommendation API.

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 CHOICES

- 1 Turn on Redis RDB persistence.
- 2 Set a Time to Live on every cache key.
- 3 Publish invalidation events to a Redis channel.
- 4 Use the cache-aside pattern with lazy loading.

Need	Configuration
Meet the expiration policy for recommendation results.	
Preserve catalog freshness and data integrity after updates.	
Meet the latency target for repeated product requests.	

99 Multiple choice · Domain 4

CASE STUDY: NORTHWIND TRADERS

You need to implement credential management for the Recommendation API that meets the security and identity requirements of Northwind Traders Inc.

Which service should you choose?

NOTE: Each correct selection is worth one point.

A. Azure Key Vault

B. Kubernetes secrets

C. Azure App Configuration

D. Container environment variables

Case study: Trey Research

Questions 100 to 109 refer to this case study.

Background

Trey Research builds AI research-knowledge platforms for pharmaceutical and life-science firms that need to discover and reuse scientific content. The company is updating the platform so it can provide semantic search, intelligent document retrieval, and real-time integrations with partners.

Python and Azure SDKs are the engineering team's primary development tools. The redesigned architecture must accommodate serverless backend processing, vector search workloads, and microservices packaged in containers.

Planned application architecture

- Docker is used to package each microservice as a container.
- Local development is used for containerized microservices and Azure Function apps, and the resulting code is stored in GitHub.
- Azure Container Registry (ACR) serves as the store for custom images used by the microservice containers.
- The source for base images is Docker Hub.
- The document repository, metadata store, and vector store all use Azure Cosmos DB for NoSQL.
- For documents hosted in Azure Cosmos DB for NoSQL, Azure Functions generates vector embeddings and then sends Service Bus messages that trigger search index updates.
- Backend API services hosted as Azure Container Apps (ACA) apps provide semantic search over documents in Azure Cosmos DB for NoSQL. These API services consume Service Bus messages and refresh search indexes.
- When the embedding model is changed, Azure Kubernetes Service (AKS) handles batch vector embedding regeneration for documents that already exist in Azure Cosmos DB for NoSQL.
- Documents submitted by business partners enter internal AI workflows for semantic search and retrieval through a containerized webhook that is exposed on the extranet.

Monitoring

- Azure resource telemetry is delivered to Azure Monitor.
- Logs from ACA apps, AKS containers, and Azure Functions apps are centralized in a Log Analytics workspace.
- The current Azure Functions monitoring approach relies on instrumentation from the Azure Application Insights SDK.

Business requirements

- For documents hosted in Azure Cosmos DB for NoSQL that are new or changed, embedding generation must happen automatically.
- The solution must ensure automatic scaling of backend API services during business hours.
- Minimizing cold start delay is required for the backend APIs.
- Container images must not contain secrets.
- Across apps and hosts in Azure Functions, developers must be able to connect related telemetry.
- OpenTelemetry SDK instrumentation must be used to implement all tracing.
- The solution must minimize development effort.

Technical requirements

- Before code updates can be merged into the main branch, container images must be built automatically and validated.

- Local developer machines must not be required for image builds.
- The length of a Service Bus queue must control event-driven scaling in ACA.
- The solution must minimize RU consumption in Azure Cosmos DB for NoSQL.
- Embeddings stored in Azure Cosmos DB for NoSQL must be used for vector similarity search.
- Azure App Service must host the containerized webhook service used by partners.
- Runtime secret delivery must be provided by Azure App Service without dependence on external services.
- Version-controlled Bicep templates must be used to deploy resources and workloads.

Known issues

- Vector similarity queries cause sharp increases in RU consumption.

100 Drop-down selection · Domain 1

CASE STUDY: TREY RESEARCH

You need to implement the required container image build workflow.

Which settings should be configured?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Build area	Selected setting
Trigger used for container image builds	Commit trigger · Pull request trigger · Base image update trigger
Image build implementation	ACR Task execution · GitHub workflow · Docker Compose setup

Choose one option in each row.

101 Multiple choice · Domain 1

CASE STUDY: TREY RESEARCH

You need to supply a connection string to the partner-facing service as required.

Which option should you use?

NOTE: Each correct selection is worth one point.

A. Secrets stored in GitHub

B. ENV instructions in the Dockerfile

C. Environment variables configured in App Service

D. A Helm chart package in Azure Container Registry

102 Multiple choice · Domain 1

CASE STUDY: TREY RESEARCH

You need to deploy the batch embedding workload in accordance with the technical requirements.

What should you use?

NOTE: Each correct selection is worth one point.

- A Azure CLI az aks commands
- B XML files
- C YAML files
- D kubectl run and kubectl create commands

103 Drag and drop · Domain 1

CASE STUDY: TREY RESEARCH

You need to configure backend API scaling to satisfy the business and technical requirements.

Which setting should you use for each element?

To answer, move the appropriate settings to the correct elements. You may use each setting 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 CHOICES

1. 0
2. 1
3. queueLength
4. messageCount
5. Azure Service Bus
6. Azure Container Apps

Scaling element	Value
Scale rule source	
Scale trigger metadata	
Minimum replica count	

104 Multiple response · Domain 2

CASE STUDY: TREY RESEARCH

You need to mitigate the known performance issue caused by vector similarity searches.

Which two actions should you take?

NOTE: Each correct selection is worth one point.

- A Revise the container indexing policy.
- B Use Strong consistency for the account.
- C Use lower-precision indexing for the vector fields.
- D Create a composite index that includes the vector fields and container metadata.

105 Drop-down selection · Domain 2

CASE STUDY: TREY RESEARCH

You need to set up vector embedding refreshes that satisfy the business and technical requirements.

Which settings should you choose?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Requirement	Setting
Detect document updates that require vectorization.	Change feed processor · Scheduled full container scan · Cross-partition SELECT query
Expand vectorization processing across workers.	Lease container · Strong consistency level · Container RU throughput

Choose one option in each row.

106 Multiple choice · Domain 3

CASE STUDY: TREY RESEARCH

You need to configure Azure Functions to create vector embeddings for the planned application architecture.

Which trigger type should you choose?

NOTE: Each correct selection is worth one point.

- (A) HTTP trigger
- (B) Event Grid trigger
- (C) Service Bus trigger
- (D) Azure Cosmos DB trigger

107 Multiple choice · Domain 3

CASE STUDY: TREY RESEARCH

Choose how to release both the Azure Function infrastructure and the function app while satisfying the business and technical requirements.

Which deployment option should you select?

NOTE: Each correct selection is worth one point.

- (A) Azure CLI deployment
- (B) GitHub Actions pipeline
- (C) Local Git publishing
- (D) Azure Functions Core Tools publish

108 Build list · Domain 4

CASE STUDY: TREY RESEARCH

You need to implement trace correlation as required by the business requirements.

Which three actions should you perform in order?

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

- A Redeploy the services after instrumentation.
- B Add an OpenTelemetry SDK view.
- C Configure an OpenTelemetry SDK trace exporter.
- D Invoke TelemetryClient.TrackEvent() in service methods.
- E Instrument application code with the OpenTelemetry SDK.

109 Multiple choice · Domain 4

CASE STUDY: TREY RESEARCH

According to the business requirements, you need an ad hoc investigation of log data from AKS, ACA, and Azure Function apps.

Which option should you select?

NOTE: Each correct selection is worth one point.

- A Kusto Query Language (KQL) query
- B Function written in Python
- C Azure activity log data
- D Custom Azure Monitor workbook

Part 2: Answers and explanations

A quick answer key first ("see" means a table answer, shown in full below), then the reasoning for every question.

1	BD	2	Y	3	Y	4	N	5	Y	6	B	7	see	8	ACD
9	see	10	see	11	see	12	see	13	AE	14	A	15	AB	16	see
17	CD	18	B	19	EBA	20	see	21	A	22	BDCA	23	see	24	C
25	see	26	AD	27	EG	28	see	29	see	30	BD	31	ABCD	32	see
33	A/B	34	AD	35	F/H	36	C	37	see	38	A	39	see	40	see
41	A/B	42	Y	43	N	44	Y	45	Y	46	BDE	47	C	48	see
49	DE	50	see	51	BCDA	52	C	53	D	54	B	55	see	56	see
57	C	58	AC	59	B	60	C	61	A	62	BD	63	see	64	see
65	CDABE	66	AC	67	see	68	A	69	see	70	see	71	ACE	72	A
73	see	74	ABD	75	ACD	76	AC	77	CD	78	CABDE	79	see	80	see
81	see	82	see	83	see	84	BDE	85	Y	86	Y	87	N	88	Y
89	see	90	AF	91	see	92	see	93	D	94	see	95	see	96	C
97	BDAE	98	see	99	A	100	see	101	C	102	C	103	see	104	AC
105	see	106	D	107	B	108	CEA	109	A						

---

1 Multiple response · Develop containerized solutions on Azure

ANSWERS: B, D

- **B** ACR task that uses the Git repository as its build context
- **D** Source code commit trigger on the ACR task

An ACR task runs the build in Azure by using the repository as its context, so no local Docker engine or separate build server is needed. A source code commit trigger starts the task automatically whenever a developer commits to the repository, which meets the automatic build requirement.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Webhooks are native to ACR, but they usually notify downstream services after registry events instead of building images from source code.
- **C.** An Artifact Cache rule caches images from external registries such as Docker Hub. It does not build new images from source code.

Microsoft Learn: Store manage containers azure container registry: build run acr tasks · Container registry: container registry tasks overview

2 Yes or No · Develop containerized solutions on Azure

ANSWER: YES

Solution: Save the API key as an application setting for api2.

An App Service application setting is stored in service configuration and exposed to the container as an environment variable at runtime. The key is not committed to the Git repository.

Microsoft Learn: Deploy containers azure app service: configure settings · App service: app service key vault references

3 Yes or No · Develop containerized solutions on Azure

ANSWER: YES

Solution: Store the API key in Azure Key Vault and reference it from Azure App Service.

Keeping the key in Azure Key Vault and referencing it from App Service prevents the value from being recorded in Git commit history.

Microsoft Learn: Deploy containers azure app service: configure settings · App service: app service key vault references

4 Yes or No · Develop containerized solutions on Azure

ANSWER: NO

Solution: Place the API key in the Dockerfile.

Putting the key in the Dockerfile would add it to the source repository and therefore to Git commit history.

Microsoft Learn: Deploy containers azure app service: configure settings · App service: app service key vault references

5 Yes or No · Develop containerized solutions on Azure

ANSWER: YES

Solution: Reference the API key by using Azure App Service settings.

App Service settings can provide the key to the app without adding the value to source control or commit history.

Microsoft Learn: Deploy containers azure app service: configure settings · App service: app service key vault references

6 Multiple choice · Develop containerized solutions on Azure

ANSWER: B

B Reference the image by its SHA digest.

A SHA digest uniquely identifies a specific immutable image version.

WHY THE OTHER CHOICES ARE WRONG

- **A.** The latest tag can be moved to a different build, so it is not immutable.
- **C.** A tag such as production can be reassigned, so it cannot guarantee the same build forever.
- **D.** Rebuilding on a schedule does not ensure that deployment points to one immutable image version.

Microsoft Learn: Container registry: container registry image tag version

7 Drop-down selection · Develop containerized solutions on Azure

ANSWER

MODEL_VERSION value	**Plain text app setting**
Database password value	**Key Vault reference**

MODEL_VERSION value: App Service application settings are appropriate for non-sensitive values and are injected as environment variables. Database password value: A Key Vault reference lets App Service retrieve the password securely at runtime.

Microsoft Learn: Deploy containers azure app service: configure settings · App service: app service key vault references

8 Multiple response · Develop containerized solutions on Azure

ANSWERS: A, C, D

- **A** Trigger on base image updates
- **C** Scheduled timer trigger
- **D** Trigger on source code commits

A base image update trigger monitors the base image layer and queues a rebuild when that image changes, keeping dependent images current. A timer trigger uses a cron expression to run the task on a defined schedule, which supports recurring maintenance rebuilds. A source code commit trigger starts a rebuild automatically when the application source changes.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Registry events such as image push or delete can be sent to services like Event Grid, but they are not ACR Tasks build trigger types.
- **E.** A webhook notification can inform other systems about events, but it does not initiate ACR Tasks builds.

Microsoft Learn: Container registry: container registry tasks overview · Store manage containers azure container registry: build run acr tasks

9 Drag and drop · Develop containerized solutions on Azure

ANSWER

Make sure the image has a version identifier.	**Assign a unique tag to the image.**
Permit a secure push operation to ACR.	**Sign in to ACR by using Microsoft Entra ID.**
Make the image available in the registry for deployment.	**Upload the image to ACR.**

Make sure the image has a version identifier.: A unique tag identifies the specific image version intended for deployment. Permit a secure push operation to ACR.: You must authenticate to ACR before the client can push an image. Make the image available in the registry for deployment.: Docker push or az acr build uploads the image layers and manifest to ACR so deployments can use the image.

Microsoft Learn: Container registry: container registry image tag version · Container registry: container registry authentication · Container registry: container registry get started docker cli

10 Drop-down selection · Develop containerized solutions on Azure

ANSWER

Authenticate App Service when pulling from ACR.	**Grant AcrPull to a managed identity.**
Prevent storage of static registry credentials.	**Use a managed identity with a role assignment.**

Authenticate App Service when pulling from ACR.: App Service can use a managed identity with Microsoft Entra ID to access ACR. The AcrPull role grants image pull permissions without static credentials. Prevent storage of static registry credentials.: Managed identity uses runtime Microsoft Entra ID tokens, so no static registry secret is stored in code or configuration.

Microsoft Learn: App service: configure custom container

11 Drag and drop · Develop containerized solutions on Azure

ANSWER

Validate a new revision without changing production traffic.	**Multiple revision mode**
Send 20 percent of requests to a new revision.	**Traffic splitting configuration**

Validate a new revision without changing production traffic.: Multiple revision mode permits more than one active revision, enabling revision-specific testing without affecting the main production endpoint. Send 20 percent of requests to a new revision.: Traffic splitting assigns percentage weights so ingress traffic can be distributed across active revisions.

Microsoft Learn: Container apps: revisions · Container apps: traffic splitting

12 Drop-down selection · Develop containerized solutions on Azure

ANSWER

Choose where to store the key for App1.	**Azure Key Vault secret object**
Permit App1 to read the key store securely.	**Managed identity**

Choose where to store the key for App1.: An Azure Key Vault secret can store an API access key outside the Container Apps environment. Permit App1 to read the key store securely.: A managed identity lets App1 authenticate to Key Vault and retrieve the secret with minimal maintenance.

Microsoft Learn: Container apps: manage secrets · Container apps: managed identity

13 Multiple response · Develop containerized solutions on Azure

ANSWERS: A, E

A Configure traffic splitting.

E Enable multiple revision mode.

Traffic splitting supports a controlled rollout by assigning traffic percentages between revisions. Multiple revision mode is required so traffic splitting can route requests among active revisions.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Restarting a revision does not create a staged release or rollback path.
- **C.** Changing replica count adjusts capacity but does not control rollout between revisions.
- **D.** Single revision mode allows only one active revision, so traffic cannot be shared for gradual rollout.

Microsoft Learn: Container apps: traffic splitting · Container apps: revisions

---

14 Multiple choice · Develop containerized solutions on Azure

ANSWER: A

A Review the pod events and container logs.

Reviewing events and logs is the first diagnostic step for separating application-level failures from cluster resource issues.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Changing initialDelaySeconds before reviewing logs may hide the issue and does not reveal whether the app needs more startup time or is failing permanently.
- **C.** Draining or restarting the node might move the pod, but it will not correct probe configuration or application health behavior.
- **D.** Scaling out can add capacity, but it does not identify the configuration or application error that causes restarts.

Microsoft Learn: Monitor apps azure kubernetes service: troubleshoot pods services · Deploy apps azure kubernetes service: deploy verify results

---

15 Multiple response · Develop containerized solutions on Azure

ANSWERS: A, B

A Name of the queue

B Scaler trigger type

The KEDA Azure Storage queue scaler requires the queue name so it knows which queue to monitor for message count. The trigger type identifies the KEDA scaler, such as an Azure queue scaler, and determines the event source and scaling logic.

WHY THE OTHER CHOICES ARE WRONG

- **C.** Maximum replicas limits scale-out, but it is not a required value for the Azure queue trigger itself.
- **D.** Polling interval is an optional tuning setting rather than a required trigger value.

Microsoft Learn: Scale containers azure container apps: event driven scaling keda · Scale containers azure container apps: keda scalers custom workloads · Container apps: scale app

16 Drag and drop · Develop containerized solutions on Azure

ANSWER

Add pods during high CPU demand.	**HorizontalPodAutoscaler resource**
Make the application reachable only within the cluster.	**ClusterIP service**

Add pods during high CPU demand.: A HorizontalPodAutoscaler adjusts the replica count automatically by using CPU or custom metrics. Make the application reachable only within the cluster.: A ClusterIP service exposes pods on the cluster network without creating external access.

Microsoft Learn: Aks: concepts scale · Aks: concepts network services · Aks: tutorial kubernetes scale

17 Multiple response · Develop containerized solutions on Azure

ANSWERS: C, D

C Allow the scaling rule to terminate all active replicas.

D Configure a Kubernetes Event-driven Autoscaler rule that monitors queue length.

Scale to zero requires the minimum replica count to allow zero active replicas when no messages are available. A KEDA rule can monitor Service Bus queue depth and start replicas only when queued work exists.

WHY THE OTHER CHOICES ARE WRONG

- **A.** HTTP concurrency scaling reacts to HTTP requests and does not monitor the Service Bus queue.
- **B.** Increasing the maximum replica count only raises the scale-out ceiling and does not enable event-driven scale to zero.

Microsoft Learn: Scale containers azure container apps: event driven scaling keda · Container apps: scale app

18 Multiple choice · Develop containerized solutions on Azure

ANSWER: B

B Examine the Service and Endpoints objects.

Checking Service selectors and Endpoints verifies whether traffic is mapped to the intended pods.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Restarting does not reveal whether selectors or endpoints are misconfigured.
- **C.** Adding replicas does not fix a selector or service endpoint problem.
- **D.** Workstation DNS testing should come after confirming that cluster endpoints exist.

Microsoft Learn: Monitor apps azure kubernetes service: troubleshoot pods services · Aks: concepts network services

19 Build list · Develop containerized solutions on Azure

CORRECT ORDER: E → B → A

1. Change the image tag in the deployment manifest.
2. Apply the revised manifest.
3. Check the rollout status.

Changing the image tag in the manifest specifies the new container version. Applying the revised manifest starts the Kubernetes rolling update. Checking rollout status verifies that the new version deployed successfully.

WHY THE OTHER CHOICES ARE WRONG

- **C.** Scaling to zero would stop serving traffic, which violates the zero-downtime requirement.
- **D.** Deleting and recreating the deployment causes downtime.

Microsoft Learn: Deploy apps azure kubernetes service: create deployment manifests · Deploy apps azure kubernetes service: deploy verify results

---

20 Drag and drop · Develop AI solutions by using Azure data management services

ANSWER

Enable distance calculations between vectors.	**Set up a vector index.**
Choose the document representation for high-dimensional values.	**Save data as a numeric array.**

Enable distance calculations between vectors.: A vector index lets VectorDistance searches evaluate nearest neighbors efficiently at scale. It helps the engine avoid expensive full scans for high-dimensional comparisons. Choose the document representation for high-dimensional values.: Embeddings must be represented as arrays of numbers so vector search functions can evaluate each dimension.

Microsoft Learn: Implement vector search azure cosmos db: store retrieve embeddings · Cosmos db: vector search · Query: vectordistance

---

21 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: A

A lease container

The lease container records ownership and checkpoints so multiple processor instances can coordinate partition processing and balance work.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Index precision does not control how change feed partitions are assigned to workers.
- **C.** Consistency level affects read guarantees, not partition ownership for processors.
- **D.** Autoscale throughput adjusts available RUs, but it does not coordinate which instance processes each partition.

Microsoft Learn: Cosmos db: change feed processor

22 Build list · Develop AI solutions by using Azure data management services

CORRECT ORDER: B → D → C → A

1. Create the CosmosClient.
2. Set the database name.
3. Get the container reference.
4. Execute the query.

The client is created first so the application can access Cosmos DB resources. The target database must be identified before retrieving a container from it. A container reference is needed before the SDK can run a query against that container. The query is executed only after the client, database, and container context are available.

WHY THE OTHER CHOICES ARE WRONG

- **E.** DatabaseProxy is not created directly and does not supply the needed container reference.

Microsoft Learn: Use azure cosmos db sql api sdk: connect to online account · Azure cosmos: azure.cosmos.containerproxy

---

23 Drag and drop · Develop AI solutions by using Azure data management services

ANSWER

Open the connection with the account endpoint and key.	**CosmosClient instance**
Configure shared throughput.	**Database object**
Run item CRUD operations.	**Container object**

Open the connection with the account endpoint and key.: A CosmosClient instance establishes access to the Azure Cosmos DB for NoSQL account endpoint and manages account-level operations. Configure shared throughput.: A Database object provides scoped access to a database, including shared throughput configuration. Run item CRUD operations.: Item create, read, update, and delete operations are performed through a container.

Microsoft Learn: Use azure cosmos db sql api sdk: connect to online account · Cosmos db: set throughput · Api: microsoft.azure.cosmos.container

---

24 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: C

C Exclude unqueried properties from indexing.

Adding excluded paths for properties that queries do not use prevents Cosmos DB from maintaining those index entries on each write while preserving indexes for deviceId and timestamp.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Adding included paths while keeping the root path included still indexes the other properties, so write RU costs remain high.
- **B.** Consistency settings do not lower the write cost created by indexing unused properties.
- **D.** A different partition key does not remove the indexing work performed for unused fields.

Microsoft Learn: Cosmos db: index policy · Cosmos db: how to manage indexing policy

25 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Read changes from the source container.	**Monitored container**
Persist continuation tokens and partition ownership.	**Lease container**
Handle new or changed items by using the default change feed mode.	**Latest version mode**

Read changes from the source container.: The monitored container supplies the change feed, so inserts and updates in that container are processed. Persist continuation tokens and partition ownership.: The lease container stores continuation tokens and ownership state for the processor. Handle new or changed items by using the default change feed mode.: Latest version mode processes creates and updates from the change feed and is the default consumption mode.

Microsoft Learn: Cosmos db: change feed processor · Cosmos db: change feed modes

26 Multiple response · Develop AI solutions by using Azure data management services

ANSWERS: A, D

A Bounded staleness

D Strong

Bounded staleness provides a defined staleness window for cross-region visibility. Strong consistency provides global linearizability across regions.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Eventual consistency does not provide a predictable bound for visibility or ordering.
- **C.** Session consistency preserves guarantees for a session, but it does not guarantee cross-region visibility before acknowledgment.

Microsoft Learn: Use consistency models azure cosmos db sql api: understand · Cosmos db: consistency levels

27 Multiple response · Develop AI solutions by using Azure data management services

ANSWERS: E, G

E Constrain the result count.

G Order results by vector distance.

A result limit is needed so the query returns only the top five matches. The query must sort by vector distance so the closest embeddings appear first.

WHY THE OTHER CHOICES ARE WRONG

- **A.** OFFSET pagination can skip rows, but it does not specify similarity ranking.
- **B.** DISTINCT removes duplicates and does not rank documents by similarity.
- **C.** Consistency level does not define the similarity ordering logic.
- **D.** Grouping aggregates rows instead of ranking them by similarity.
- **F.** Equality predicates compare exact values and do not perform vector similarity search.
- **H.** A vector index policy can improve performance, but it is a container configuration rather than a query element.

Microsoft Learn: Cosmos db: vector search · Query: top · Query: vectordistance

28 Drag and drop · Develop AI solutions by using Azure data management services

ANSWER

Change the SQL statement structure.	**Use a parameterized query.**
Bind the user-provided value.	**Pass the ID as an argument.**
Run the statement.	**Supply the parameter tuple to the SDK method.**

Change the SQL statement structure.: A parameterized query separates user input from SQL code so the input is not interpreted as part of the statement. Bind the user-provided value.: Passing the ID as an argument binds the user input to the statement parameter. Run the statement.: Supplying the parameter tuple to the SDK execution method runs the statement with the bound value.

Microsoft Learn: Build query azure database postgresql: integrate sdks apps

29 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Connection approach	**Enable PgBouncer.**
Port for connections	**6432**

Connection approach: PgBouncer pools and reuses database connections, reducing the overhead of creating sessions. Port for connections: Port 6432 is used for PgBouncer connection pooling in Azure Database for PostgreSQL.

Microsoft Learn: Connectivity: concepts pgbouncer · Optimize vector search azure database postgresql: connection optimization

30 Multiple response · Develop AI solutions by using Azure data management services

ANSWERS: B, D

B Store embeddings in a pgvector vector column such as vector(n).

D Use a typed timestamp column for created_at.

A pgvector vector column enables similarity operators and vector indexes. A timestamp data type supports reliable filtering and indexing by date.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Text storage makes date comparison and indexing unreliable.
- **C.** A varchar column does not support pgvector similarity operators.

Microsoft Learn: Extensions: how to use pgvector · Build query azure database postgresql: create manage schemas

31 Build list · Develop AI solutions by using Azure data management services

CORRECT ORDER: A → B → C → D

1. Run a fixed workload before peak hours to record baseline P95 latency and capture EXPLAIN ANALYZE for the vector query.
2. Check peak-hour memory pressure and cache hit ratio, then compare them with the baseline.
3. Move to a tier that provides more memory per vCore.
4. Run the same benchmark again and compare P95 and P99 latency.

A baseline workload and execution plan are needed before changing the server configuration. Comparing memory pressure and cache hit ratio with the baseline verifies whether memory is the bottleneck. A memory-optimized tier directly addresses pressure caused by the vector index working set. Repeating the same benchmark confirms whether the scaling change produced measurable latency improvement.

WHY THE OTHER CHOICES ARE WRONG

- **E.** Turning off autovacuum can cause table bloat and does not solve memory pressure.
- **F.** Larger embeddings require more compute and memory, which would worsen the pressure.

Microsoft Learn: Optimize vector search azure database postgresql: tune postgresql pgvector · Optimize vector search azure database postgresql: scale high volume workloads

32 Drag and drop · Develop AI solutions by using Azure data management services

ANSWER

Restrict rows to Finance.	**WHERE department = 'Finance'**
Sort by similarity and return five rows.	**ORDER BY embedding <=> :query_embedding LIMIT 5**

Restrict rows to Finance.: This predicate limits the candidate rows to the Finance department before similarity ranking. Sort by similarity and return five rows.: This clause orders by vector distance and limits the output to the five nearest neighbors.

Microsoft Learn: Implement vector search azure database postgresql: run vector similarity search semantic retrieval · Extensions: how to use pgvector

33 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: A OR B (EITHER IS CORRECT ON ITS OWN)

A Add B-tree indexes on commonly filtered metadata columns.

B Add a pgvector index on the embedding column.

Either of these is correct on its own: Metadata indexes reduce the number of rows that must be evaluated by the vector operator. A pgvector index reduces distance computations during nearest-neighbor retrieval.

WHY THE OTHER CHOICES ARE WRONG

- **C.** Disabling autovacuum can create table bloat and does not lower vector compute cost.
- **D.** More dimensions increase the work required for each distance calculation.
- **E.** Nearest-neighbor results require ordering by distance.
- **F.** String concatenation does not reduce vector compute and creates an injection risk.
- **G.** JSON storage prevents use of pgvector operators and vector indexes.
- **H.** A longer timeout allows queries to run longer but does not reduce compute overhead.

Microsoft Learn: Implement vector search azure database postgresql: run vector similarity search semantic retrieval · Optimize vector search azure database postgresql: choose configure vector indexes · Extensions: how to optimize performance pgvector

34 Multiple response · Develop AI solutions by using Azure data management services

ANSWERS: A, D

A Set a 10-minute Time to Live on each key.

D Delete related cache keys when source data changes.

A Time to Live enforces the fixed 10-minute lifetime by letting Redis remove the key after that duration. Deleting affected keys on source updates prevents stale results from being returned and forces the next request to reload fresh data.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Sliding expiration extends a key whenever it is read, so popular results could remain longer than 10 minutes.
- **C.** Keyspace notifications report Redis events, but they do not synchronize entries with changes in the external source data.

Microsoft Learn: Implement data operations azure managed redis: implement data operations

35 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: F OR H (EITHER IS CORRECT ON ITS OWN)

F Tune Time to Live values for cache keys.

H Invalidate cache entries when source data changes.

Either of these is correct on its own: Time to Live values let keys expire automatically, reducing stale responses without constant manual cleanup. Invalidating entries only when source data changes keeps responses current without unnecessary cache operations.

WHY THE OTHER CHOICES ARE WRONG

- **A.** More memory increases capacity but does not improve data freshness.
- **B.** Keyspace notifications report events in Redis and do not enforce freshness for source data.
- **C.** Extending expiration on reads can keep stale entries alive longer.
- **D.** LRU eviction removes keys under memory pressure and does not control freshness.
- **E.** Without expiration, stale cache entries can remain indefinitely.
- **G.** Purging after each request creates unnecessary management overhead and prevents cache reuse.

Microsoft Learn: Redis: best practices memory management · Best practices: caching

36 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: C

C EXPIRE command

Resetting the TTL with EXPIRE on each read implements sliding expiration: frequently read embeddings keep extending their lifetime, and rarely read ones expire. Eviction policies act only when memory reaches the configured limit, which this cache does not reach.

WHY THE OTHER CHOICES ARE WRONG

- **A.** allkeys-lru acts only when memory reaches the configured limit. With usage well below the limit it never runs, so rarely read embeddings are never removed.
- **B.** volatile-ttl also acts only at the memory limit, and it removes the keys with the shortest remaining TTL rather than extending the lifetime of keys that are read.
- **D.** A fixed time window removes keys on a schedule regardless of how often they are read, so frequently read embeddings expire too.

Microsoft Learn: Implement data operations azure managed redis: implement data operations · Best practices: caching · Redis: best practices memory management

37 Yes or No · Develop AI solutions by using Azure data management services

ANSWERS

The embedding value is saved in a Redis hash field.	**Yes**
The code configures similarity search for the embedding field.	**No**
The key expires after 10 minutes.	**Yes**

The embedding value is saved in a Redis hash field. Yes. hset stores the embedding value as a field in a Redis hash. The code configures similarity search for the embedding field. No. The code does not create a vector index or define a search schema. The key expires after 10 minutes. Yes. The expiration value is 600 seconds, which equals 10 minutes.

Microsoft Learn: Implement vector storage azure managed redis: index query vector data · Implement data operations azure managed redis: implement data operations · Redis: quickstart create managed redis

38 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: A

A Set absolute expiration on each cache key.

Absolute expiration uses a fixed timestamp or duration from key creation and does not extend when the key is read.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Sliding expiration resets the TTL on access, so frequently read summaries could live beyond five minutes.
- **C.** PERSIST removes the TTL from a key, preventing automatic expiration.
- **D.** allkeys-lru is a memory-pressure eviction policy and does not enforce a specific five-minute lifetime.

Microsoft Learn: Implement data operations azure managed redis: implement data operations

39 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

AI responses must expire exactly 24 hours after being cached.	**Set a Time to Live (TTL) on each key.**
Cached embeddings must stay synchronized with the current source data.	**Delete related keys when the document changes.**

AI responses must expire exactly 24 hours after being cached.: A fixed TTL removes the key automatically 24 hours after creation regardless of reads. Cached embeddings must stay synchronized with the current source data.: Explicitly deleting related keys when the document changes removes stale embeddings so fresh data is loaded next time.

Microsoft Learn: Best practices: caching · Implement data operations azure managed redis: implement data operations

40 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Embedding field type	**Vector**
Vector index setting	**HNSW index**

Embedding field type: A Vector field stores multi-dimensional embeddings and enables KNN searches with metrics such as COSINE or L2. Vector index setting: HNSW provides approximate nearest-neighbor search that is efficient at scale for large embedding sets.

Microsoft Learn: Implement vector storage azure managed redis: index query vector data · Redis: overview vector similarity

41 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: A OR B (EITHER IS CORRECT ON ITS OWN)

A Create an HNSW vector index for the embedding field.

B Create a FLAT vector index for the embedding field

Either of these is correct on its own: An HNSW vector index enables approximate nearest-neighbor similarity search on the embedding field in Azure Managed Redis. A FLAT vector index enables exhaustive similarity search on the embedding field in Azure Managed Redis.

WHY THE OTHER CHOICES ARE WRONG

- **C.** Persistence protects data but does not enable search queries.
- **D.** Replication copies data and does not configure indexing.
- **E.** TTL controls key lifetime and does not enable vector search.
- **F.** Metadata-only indexes cannot perform vector similarity queries.
- **G.** More memory does not create a searchable vector index.
- **H.** Expiration behavior does not determine whether vectors are indexed.

Microsoft Learn: Redis: overview vector similarity · Implement vector storage azure managed redis: index query vector data

42 Yes or No · Connect to and consume Azure services

ANSWER: YES

Solution: Place connection strings in app settings.

App settings keep configuration outside the codebase and can hold different values for each environment.

Microsoft Learn: Build backends azure functions: manage secrets configuration · App service: app service key vault references

43 Yes or No · Connect to and consume Azure services

ANSWER: NO

Solution: Place production secrets in local.settings.json.

local.settings.json is meant for local development, not secure storage for production secrets.

Microsoft Learn: Build backends azure functions: manage secrets configuration · App service: app service key vault references

44 Yes or No · Connect to and consume Azure services

ANSWER: YES

Solution: Reference Key Vault secrets from app settings.

Key Vault references keep secrets in Key Vault and allow rotation without changing code.

Microsoft Learn: Build backends azure functions: manage secrets configuration · App service: app service key vault references

45 Yes or No · Connect to and consume Azure services

ANSWER: YES

Solution: Store environment-specific values in App Configuration and let the function app access them with managed identity.

App Configuration keeps environment settings outside source control, and managed identity removes the need to store credentials in code or repositories.

Microsoft Learn: Build backends azure functions: manage secrets configuration · App service: app service key vault references

46 Build list · Connect to and consume Azure services

CORRECT ORDER: B → D → E

1. Register the Event Grid resource provider.
2. Create a custom Event Grid topic.
3. Create an event subscription.

The Event Grid provider must be registered before Event Grid resources can be created in the subscription. A custom topic gives the application an endpoint for publishing its own events after the provider is registered. The event subscription defines delivery to the downstream processor. It is created after the topic exists because the subscription attaches to that topic.

WHY THE OTHER CHOICES ARE WRONG

- **A.** An Event Grid domain is used for scenarios with many related custom topics or more advanced routing needs, not for this basic custom workflow.
- **C.** Partner topics are for events published by third-party providers. A custom AI inference workflow uses a custom topic instead.

Microsoft Learn: Event grid: custom event quickstart

47 Multiple choice · Connect to and consume Azure services

ANSWER: C

C Set a dead-letter destination.

A dead-letter destination saves events that cannot be delivered, including events rejected with HTTP 400.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Event Grid does not retry deliveries that fail with HTTP 400 because the response is treated as a bad request.
- **B.** Batching can reduce request overhead, but it does not change delivery handling for HTTP 400 failures.
- **D.** Handshake validation confirms that an endpoint can receive events during setup. It does not preserve failed deliveries after an HTTP 400 response.

Microsoft Learn: Event grid: delivery and retry

48 Drop-down selection · Connect to and consume Azure services

ANSWER

Condition on the path prefix	**Subject filtering**
Condition on the payload property	**Advanced filtering**
Handling events that are not delivered	**Dead-letter destination**

Condition on the path prefix: Subject filters match begins-with and ends-with patterns against the event subject. Condition on the payload property: Advanced filters can evaluate properties inside event data. Handling events that are not delivered: A dead-letter destination keeps undelivered Event Grid events for later review.

Microsoft Learn: Event grid: event filtering · Event grid: manage event delivery

49 Multiple response · Connect to and consume Azure services

ANSWERS: D, E

- **D** Service Bus subscription
- **E** Service Bus topic

Each subscription on a topic represents an independent consumer that receives its own copy. A topic is the Service Bus entity that publishes messages to multiple subscriptions.

WHY THE OTHER CHOICES ARE WRONG

- **A.** A dead-letter queue holds failed messages and is not used to fan out a publication to consumers.
- **B.** Message sessions preserve ordered, related message handling but are not needed for publish-subscribe fan-out.
- **C.** A queue is a point-to-point entity in which one consumer receives each message.

Microsoft Learn: Queue process operations service bus: choose queues topics subscriptions

50 Drop-down selection · Connect to and consume Azure services

ANSWER

Evaluate payload fields.	**Advanced filtering**
Keep undelivered events.	**Dead-letter destination**
Cap retry attempts.	**Maximum delivery attempts**

Evaluate payload fields.: Advanced filters inspect event data fields such as data.priority. Keep undelivered events.: Event Grid dead-lettering stores events that delivery cannot complete. Cap retry attempts.: The Event Grid retry policy includes a maximum delivery attempts setting.

Microsoft Learn: Event grid: event filtering · Event grid: manage event delivery

51 Build list · Connect to and consume Azure services

CORRECT ORDER: B → C → D → A

1. Create the Service Bus client.
2. Create a processor for the queue.
3. Register the message and error handlers.
4. Start processing messages.

ServiceBusClient establishes the namespace connection that is needed before a queue processor can be created. A processor for the target queue is required before handlers can be attached or processing can start. ProcessMessageAsync and ProcessErrorAsync handlers must be registered so the processor has logic for messages and errors before it starts. StartProcessingAsync() begins receiving from the queue and should run only after the processor and its handlers are configured.

WHY THE OTHER CHOICES ARE WRONG

- **E.** Dead-lettering is part of runtime message handling, not the initialization sequence.

Microsoft Learn: Api: azure.messaging.servicebus.servicebusprocessor · Api: azure.messaging.servicebus.servicebusprocessor.startprocessingasync

---

52 Multiple choice · Connect to and consume Azure services

ANSWER: C

C dead-letter the message

Dead-lettering moves the failed message to the dead-letter queue for isolation and investigation.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Abandoning makes the message available for another delivery and does not isolate poison messages.
- **B.** Completing removes the message permanently, so it cannot be inspected later.
- **D.** Deferring hides the message until it is retrieved by sequence number, but it does not place it in the dead-letter queue.

Microsoft Learn: Queue process operations service bus: process messages reliably

---

53 Multiple choice · Connect to and consume Azure services

ANSWER: D

D Defer

Defer postpones delivery and requires later retrieval by sequence number without increasing the delivery count.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Abandon makes the message immediately available again and increments its delivery count.
- **B.** Complete removes the message from the queue permanently.
- **C.** Dead-letter moves the message to the dead-letter queue for failure isolation.

Microsoft Learn: Service bus messaging: message deferral · Queue process operations service bus: process messages reliably

54 Multiple choice · Connect to and consume Azure services

ANSWER: B

B advanced filter

Advanced filters can inspect event data properties and apply numeric comparison operators.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Subject filters match only the subject string with begins-with or ends-with logic and cannot compare numeric payload values.
- **C.** An event type filter compares only the eventType value in the event envelope.

Microsoft Learn: Event grid: event filtering

55 Drag and drop · Connect to and consume Azure services

ANSWER

Send one message to multiple subscribers.	**Service Bus topic**
Process messages in FIFO order.	**Service Bus queue**
Separate failed messages.	**dead-letter queue**

Send one message to multiple subscribers.: A topic can have multiple subscriptions, allowing independent consumers to receive a copy of the message. Process messages in FIFO order.: A queue supports ordered processing for a single-consumer pattern. Separate failed messages.: Failed messages can be moved to the dead-letter queue for review and recovery.

Microsoft Learn: Service bus messaging: service bus queues topics subscriptions · Queue process operations service bus: process messages reliably

56 Yes or No · Connect to and consume Azure services

ANSWERS

The processor receives messages from the orders queue.	**Yes**
If MessageHandler throws an exception, the processor completes the message automatically.	**No**
Processing errors are sent to ErrorHandler.	**Yes**

The processor receives messages from the orders queue. Yes. CreateProcessor("orders") creates a processor for the queue named orders. If MessageHandler throws an exception, the processor completes the message automatically. No. With AutoCompleteMessages set to true, an unhandled exception causes the message to be abandoned instead of completed. Processing errors are sent to ErrorHandler. Yes. Assigning ProcessErrorAsync registers ErrorHandler as the error callback.

Microsoft Learn: Api: azure.messaging.servicebus.servicebusprocessor · Api: azure.messaging.servicebus.servicebusprocessoroptions.autocompletemessages

57 Multiple choice · Connect to and consume Azure services

ANSWER: C

C App settings that reference Key Vault

Key Vault references in app settings keep secrets outside the repository, let the app resolve them at runtime, and support rotation without code changes or redeployment.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Committing a parameter file that contains secrets to source control exposes sensitive values and violates secret management guidance.
- **B.** local.settings.json is for local development and is not used as a secure production configuration store by Azure Functions.
- **D.** A hard-coded connection string exposes the secret and requires a code change and redeployment when credentials rotate.

Microsoft Learn: App service: app service key vault references

58 Multiple response · Connect to and consume Azure services

ANSWERS: A, C

A Use the Queue Storage trigger built-in retry behavior.

C Create a queue-triggered function for image processing.

A Queue Storage trigger retries failed processing by built-in behavior: messages are retried up to five attempts, the delay between attempts is controlled by visibilityTimeout in host.json, and repeated failures go to the poison queue. A queue-triggered function lets the HTTP function enqueue work and return quickly, while background processing scales according to queue depth.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Persisting metadata can help with tracking, but it does not provide automatic retry or independent scaling for image processing.
- **D.** Processing the image inside the HTTP request path increases timeout risk and ties processing capacity to request execution.

Microsoft Learn: Azure functions: functions bindings storage queue trigger · Azure functions: functions bindings error pages

59 Multiple choice · Connect to and consume Azure services

ANSWER: B

B Input binding

An input binding supplies external data, such as a blob, to the function without manual SDK code.

WHY THE OTHER CHOICES ARE WRONG

- **A.** A trigger starts the function and does not retrieve blob contents.
- **C.** An output binding writes data from the function instead of reading existing data.
- **D.** A timer trigger runs on a schedule and does not retrieve the blob file.

Microsoft Learn: Azure functions: functions bindings storage blob · Azure functions: functions triggers bindings

60 Multiple choice · Connect to and consume Azure services

ANSWER: C

C HTTP trigger

HTTP triggers expose an endpoint that returns a synchronous response, can be protected with Microsoft Entra ID, and scale with incoming request volume.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Event Grid triggers run from event subscriptions asynchronously and do not expose a request-response API endpoint.
- **B.** Queue triggers process messages in the background and cannot return an immediate response to the calling client.
- **D.** Service Bus topic triggers consume subscription messages asynchronously and are not client-facing API endpoints.

Microsoft Learn: Azure functions: functions bindings http webhook trigger · App service: configure authentication provider aad · Azure functions: event driven scaling

61 Multiple choice · Connect to and consume Azure services

ANSWER: A

A Event Grid trigger

An Event Grid trigger subscribes to Azure Storage events such as Microsoft.Storage.BlobCreated and invokes the function through push delivery with minimal latency.

WHY THE OTHER CHOICES ARE WRONG

- **B.** An HTTP trigger requires a client to call the function.
- **C.** A Queue Storage trigger requires a message to be written to a queue.
- **D.** A Timer trigger runs on a schedule and does not react directly to uploads.

Microsoft Learn: Azure functions: functions event grid blob trigger

62 Multiple response · Connect to and consume Azure services

ANSWERS: B, D

B Add dead-letter handling.

D Turn on duplicate detection for the queue.

Dead-letter handling keeps failed messages available for inspection and possible replay. Duplicate detection on the queue rejects messages with the same MessageId within the configured detection window.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Sessions group related messages for ordered processing, but they do not prevent duplicate MessageId values from being accepted.
- **C.** An HTTP trigger removes the durable queue processing model and does not address duplicate detection or failed-message retention.

Microsoft Learn: Service bus messaging: duplicate detection · Queue process operations service bus: process messages reliably

63 Yes or No · Connect to and consume Azure services

ANSWERS

Invoking the function requires a function key.	**Yes**
The function accepts HTTP GET requests.	**No**
The response contains the request body.	**Yes**
The function checks the request body before returning.	**No**

Invoking the function requires a function key. Yes. AuthorizationLevel.Function requires callers to provide a function key for the HTTP-triggered function. The function accepts HTTP GET requests. No. The HttpTrigger attribute lists only "post", so GET is not an allowed method. The response contains the request body. Yes. The function reads req.Body into result and returns that value in OkObjectResult. The function checks the request body before returning. No. The code reads the body and returns it directly without validation or deserialization.

Microsoft Learn: Build backends azure functions: configure identity access · Azure functions: functions bindings http webhook trigger

64 Drag and drop · Connect to and consume Azure services

ANSWER

Send an immediate response back to a client.	**HTTP trigger**
Handle background work from a queue.	**Queue Storage trigger**
Execute code on a fixed schedule.	**Timer trigger**

Send an immediate response back to a client.: An HTTP trigger supports synchronous request-response interaction. Handle background work from a queue.: A Queue Storage trigger processes queued messages in the background. Execute code on a fixed schedule.: A Timer trigger starts functions according to a schedule.

Microsoft Learn: Build backends azure functions: manage secrets configuration · Azure functions: functions bindings storage queue trigger · Azure functions: functions triggers bindings

65 Build list · Connect to and consume Azure services

CORRECT ORDER: C → D → A → B → E

1. Create the storage account.
2. Create the function app.
3. Enable a managed identity for the function app and grant it access to Key Vault secrets.
4. Configure application settings with Key Vault references.
5. Deploy the function code.

A storage account is required before the function app can be created. The function app must exist before you can assign its identity. The function app needs a managed identity with permission before it can resolve Key Vault secrets. Key Vault references should be added after identity and access are configured. Code deployment should happen after the app settings and secret access are configured.

Microsoft Learn: Azure functions: storage considerations · App service: app service key vault references · Build backends azure functions: manage secrets configuration

---

66 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: A, C

A Use a system-assigned managed identity.

C Grant the Key Vault Secrets User role at the vault scope.

A managed identity removes the need to store service principal credentials such as client IDs and client secrets. This helps prevent credential leakage from code or configuration. A vault-scoped Secrets User assignment grants the minimum read access needed at the narrowest relevant scope. This follows least privilege and limits the impact of compromise.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Secrets should remain in Key Vault. Azure App Configuration is for non-sensitive settings, although it can point to Key Vault references.
- **D.** Subscription-level Key Vault Administrator rights are broader than required. This violates least privilege and expands the blast radius.

Microsoft Learn: Managed identities azure resources: overview · General: rbac guide · Identity platform: secure least privileged access

---

67 Drop-down selection · Secure, monitor, and troubleshoot Azure solutions

ANSWER

[A]	**DefaultAzureCredential**
[B]	**get_configuration_setting**

DefaultAzureCredential uses managed identity in Azure and can fall back to a developer credential locally, so the code does not need to change. get_configuration_setting returns the specific setting identified by the key and label.

Microsoft Learn: Azure: appconfiguration readme · Azure appconfiguration: azure.appconfiguration.azureappconfigurationclient · Azure identity: azure.identity.defaultazurecredential

68 Multiple choice · Secure, monitor, and troubleshoot Azure solutions

ANSWER: A

A Labels

Labels let one key have separate values for environments such as test, staging, and production. The application can request the value that matches the target environment label.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Key prefixes organize groups of keys into a hierarchy. They do not provide multiple environment-specific values for a single key.
- **C.** Content types describe the format of a configuration value, such as JSON. They do not select different values by environment.
- **D.** Resource tags are metadata on the Azure resource. They are not labels on individual configuration settings and cannot provide environment-specific values.

Microsoft Learn: Manage app settings app config: organize labels feature flags · Azure app configuration: howto best practices

69 Drag and drop · Secure, monitor, and troubleshoot Azure solutions

ANSWER

REST API endpoint URL	**Azure App Configuration**
Beta model feature flag	**Azure App Configuration**
Database password value	**Azure Key Vault**
Storage account access key	**Azure Key Vault**

REST API endpoint URL: Endpoint URLs are ordinary configuration values and fit Azure App Configuration. Beta model feature flag: Feature flags are application configuration data and are supported by Azure App Configuration. Database password value: A database password is a secret and should be stored in Azure Key Vault. Storage account access key: Storage account keys are secrets and belong in Azure Key Vault.

Microsoft Learn: Manage app settings app config: decide app config vs key vault · Azure app configuration: use key vault references dotnet core

70 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWERS

The code reads the current latest version of dbPassword.	**No**
When the app runs on an Azure host with managed identity enabled, the code can authenticate to Key Vault without storing client secrets in the application.	**Yes**
If dbPassword is rotated and a new version is created, later calls made by this code retrieve the new version automatically.	**No**

The code reads the current latest version of dbPassword. No. The get_secret call includes version="123", so Key Vault returns that specific version instead of resolving the latest version. When the app runs on an Azure host with managed identity enabled, the code can authenticate to Key Vault without storing client secrets in the application. Yes. DefaultAzureCredential includes ManagedIdentityCredential in its credential chain, so the app can obtain tokens without a client secret when managed identity is available. If dbPassword is rotated and a new version is created, later calls made by this code retrieve the new version automatically. No. The version argument pins retrieval to version 123. The application will not read a rotated version unless the version is removed or changed.

Microsoft Learn: Azure keyvault secrets: azure.keyvault.secrets.secretclient · Azure identity: azure.identity.defaultazurecredential · General: authentication

71 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: A, C, E

- **A** Enable configuration refresh with a polling interval.
- **C** Store API keys in Key Vault.
- **E** Use managed identity for App Configuration access.

Azure App Configuration can refresh cached settings on a polling interval, commonly with a sentinel key. This supports updates without restarting the application. API keys are secrets and should be stored in Key Vault. App Configuration can reference Key Vault secrets while keeping the secret values separate. Managed identity avoids storing App Configuration connection strings or access keys in the application.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Environment variables are static for a running application and do not provide centralized dynamic refresh.
- **D.** Labels are useful for environment separation, but this scenario requires dynamic refresh and secret separation, not environment isolation.

Microsoft Learn: Azure app configuration: howto best practices · Azure app configuration: use key vault references dotnet core · Azure app configuration: concept enable rbac

72 Multiple choice · Secure, monitor, and troubleshoot Azure solutions

ANSWER: A

A Cache configuration values and refresh them on an interval.

The Azure App Configuration SDK can cache settings locally and refresh them on a configured interval, often by checking a sentinel key. This supports dynamic updates without adding a remote call to every request.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Environment variables are static until the app is restarted or redeployed. They do not support dynamic configuration updates.
- **C.** Reading App Configuration on every request adds network latency and hurts performance.
- **D.** Azure Key Vault is for secrets, not general application configuration management.

Microsoft Learn: Azure app configuration: enable dynamic configuration aspnet core · Azure app configuration: howto best practices

73 Drag and drop · Secure, monitor, and troubleshoot Azure solutions

ANSWER

Request a secret without a version.	**Enables automatic rotation**
Request a secret by using a version identifier.	**Needs a manual change after rotation**
Authenticate by using managed identity.	**Avoids storing credentials**
Use a client secret saved in environment variables.	**Creates credential exposure**

Request a secret without a version.: When no version is supplied, Key Vault resolves the current version, so rotated versions can be picked up automatically. Request a secret by using a version identifier.: After rotation, the application must be updated to reference the new version. Authenticate by using managed identity.: Managed identity avoids placing a client secret or password in code or configuration. Use a client secret saved in environment variables.: Environment variables can expose the client secret through application or container configuration.

Microsoft Learn: General: about keys secrets certificates · Managed identities azure resources: overview · Azure keyvault secrets: azure.keyvault.secrets.secretclient

74 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: A, B, D

- **A** Enable a system-assigned managed identity.
- **B** Create a Key Vault RBAC role assignment.
- **D** Read the secret at runtime by using the SDK.

A managed identity lets the container authenticate without stored credentials. Because the vault uses Azure RBAC, an RBAC assignment authorizes the managed identity to read secrets. Runtime SDK retrieval allows the application to obtain the current secret value after rotation.

WHY THE OTHER CHOICES ARE WRONG

- **C.** A vault configured for Azure RBAC does not use Key Vault access policies for authorization.
- **E.** Exporting the secret during deployment bakes in a value and prevents automatic use of later rotations.

Microsoft Learn: General: authentication · General: rbac guide · General: about keys secrets certificates

---

75 Build list · Secure, monitor, and troubleshoot Azure solutions

CORRECT ORDER: A → C → D

1. Enable a system-assigned managed identity.
2. Grant Key Vault Secrets User to the managed identity.
3. Read the secret by using the SDK.

The application needs an identity before permissions can be assigned to it. The managed identity must receive least-privilege secret read access before the app can retrieve the secret. SDK retrieval should occur only after identity and authorization are configured.

WHY THE OTHER CHOICES ARE WRONG

- **B.** The recover permission restores a soft-deleted secret. It does not grant normal read access to a secret value.

Microsoft Learn: Managed identities azure resources: overview · General: rbac guide · Secrets: quick create python

---

76 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: A, C

- **A** Configure secret autorotation by using Event Grid and Azure Functions.
- **C** Retrieve secrets without a version identifier.

Event Grid can trigger an Azure Function to run the secret rotation workflow and create or update the secret automatically. This is an autorotation implementation, not a Key Vault rotation policy. When no version is specified, Key Vault returns the latest version of the secret.

WHY THE OTHER CHOICES ARE WRONG

- **B.** A version identifier pins the app to one version and prevents automatic use of later versions.
- **D.** Caching for 30 days can keep the app using an old value after the credential rotates.

Microsoft Learn: General: autorotation · Secrets: tutorial rotation · Azure keyvault secrets: azure.keyvault.secrets.secretclient

77 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: C, D

C summarize

D where

summarize aggregates rows. With count() and a by clause, it can produce failure totals by operation name. where filters rows, including conditions such as the timestamp range and the failed success status.

WHY THE OTHER CHOICES ARE WRONG

- **A.** distinct removes duplicate values but does not filter failed rows or count them by operation.
- **B.** project selects, renames, or removes columns. It does not filter rows or compute counts.

Microsoft Learn: Query: where operator · Query: summarize operator · App: data model complete

78 Build list · Secure, monitor, and troubleshoot Azure solutions

CORRECT ORDER: C → A → B → D → E

1. Start with the requests table.
2. Apply the time filter.
3. Filter to failed requests.
4. Join to the dependencies table.
5. Summarize average dependency duration by operation.

A KQL pipeline must begin with the primary source table, which is the requests data in this sequence. The time filter should be applied immediately after selecting the table so Kusto can reduce the scanned data before later operations. Filtering to failed requests before the join limits the rows that participate in the correlation step. After the request rows are narrowed, joining adds the dependency records needed for correlation before aggregation. summarize must run last because it aggregates the joined data into average dependency duration by operation.

Microsoft Learn: Query: best practices · Query: join operator · Query: summarize operator

79 Drop-down selection · Secure, monitor, and troubleshoot Azure solutions

ANSWER

[A]	**where**
[B]	**where**
[C]	**summarize**

where restricts the rows to failed requests for the required time range. where applies the timestamp condition to keep only rows from the last hour. summarize calculates avg(depDuration) for each operation_Name group.

Microsoft Learn: Query: where operator · Query: summarize operator · Analyze telemetry logs metrics: write basic kql queries

80 Drag and drop · Secure, monitor, and troubleshoot Azure solutions

ANSWER

Register the global tracer provider.	**Set up the application's TracerProvider for tracing.**
Send traces to Azure Monitor.	**Instantiate the Azure Monitor trace exporter.**
Attach the exporter to the provider.	**Add a span processor that sends spans to the exporter.**
Create spans in application code.	**Call tracer.start_as_current_span().**

Register the global tracer provider.: TracerProvider defines the global provider used by OpenTelemetry tracing. Send traces to Azure Monitor.: AzureMonitorTraceExporter sends trace data to Azure Monitor. Attach the exporter to the provider.: The span processor connects the exporter to the provider and handles batching. Create spans in application code.: tracer.start_as_current_span() creates spans for application operations.

Microsoft Learn: Azure: monitor opentelemetry exporter readme · Instrument app opentelemetry: configure spans traces

81 Drag and drop · Secure, monitor, and troubleshoot Azure solutions

ANSWER

Honor parent sampling choices.	**ParentBased**
Capture every span in local tests.	**AlwaysOnSampler**
Sample 10 percent of production traces.	**TraceIdRatioBased(0.1)**

Honor parent sampling choices.: ParentBased uses the parent span sampling decision for child spans. Capture every span in local tests.: AlwaysOnSampler records all spans, which satisfies local testing requirements. Sample 10 percent of production traces.: TraceIdRatioBased(0.1) samples 10 percent of traces based on the trace ID hash, keeping the decision consistent across services in the trace.

Microsoft Learn: App: opentelemetry configuration · App: opentelemetry sampling

82 Drop-down selection · Secure, monitor, and troubleshoot Azure solutions

ANSWER

[A]	**where**
[B]	**summarize**

where applies the timestamp condition that keeps rows from the last 30 minutes. summarize performs the average calculation grouped by cloud_RoleName.

Microsoft Learn: Query: where operator · Query: ago function · Query: summarize operator

83 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWERS

The tracer provider is configured before any spans are created.	**Yes**
The setup exports traces synchronously.	**No**
The setup enables trace export to Azure Monitor.	**Yes**

The tracer provider is configured before any spans are created. Yes. trace.set_tracer_provider is called before code creates spans, and the processor is registered before trace data is exported. The setup exports traces synchronously. No. BatchSpanProcessor exports spans asynchronously in batches. SimpleSpanProcessor is used for synchronous export behavior. The setup enables trace export to Azure Monitor. Yes. AzureMonitorTraceExporter is added through the span processor, allowing completed spans to be sent to Application Insights.

Microsoft Learn: Azure: monitor opentelemetry exporter readme · Azure monitor opentelemetry exporter: azure.monitor.opentelemetry.exporter.azuremonitortraceexporter

84 Multiple response · Secure, monitor, and troubleshoot Azure solutions

ANSWERS: B, D, E

- **B** join
- **D** summarize
- **E** where

join correlates request rows with dependency rows. summarize performs the aggregation needed for average duration. where filters the telemetry to failed requests and the required time range.

WHY THE OTHER CHOICES ARE WRONG

- **A.** extend only adds calculated columns and does not filter, correlate, or aggregate the telemetry.
- **C.** distinct returns unique column values and cannot perform the needed filtering, correlation, or duration aggregation.

Microsoft Learn: Query: join operator · Query: summarize operator · Query: where operator

85 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWER: YES

Solution: The query limits results to the past 30 minutes.

timestamp > ago(30m) restricts the request rows to the last 30 minutes.

Microsoft Learn: Query: where operator · Query: summarize operator · Analyze telemetry logs metrics: write basic kql queries

86 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWER: YES

Solution: The query includes only requests with duration greater than two seconds.

duration > 2000 filters out requests that are two seconds or shorter.

Microsoft Learn: Query: where operator · Query: summarize operator · Analyze telemetry logs metrics: write basic kql queries

87 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWER: NO

Solution: The query returns each individual slow request row.

summarize aggregates rows, so individual request records are not returned.

Microsoft Learn: Query: where operator · Query: summarize operator · Analyze telemetry logs metrics: write basic kql queries

88 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWER: YES

Solution: The query returns average request duration by operation.

summarize avg(duration) by operation_Name calculates the average duration for each operation_Name.

Microsoft Learn: Query: where operator · Query: summarize operator · Analyze telemetry logs metrics: write basic kql queries

89 Yes or No · Secure, monitor, and troubleshoot Azure solutions

ANSWERS

The code starts a span when each request is processed.	**Yes**
The code automatically sends trace context to downstream HTTP services.	**No**
Additional instrumentation is required for end-to-end trace correlation.	**Yes**

The code starts a span when each request is processed. Yes. tracer.start_as_current_span creates a span for the process_request operation. The code automatically sends trace context to downstream HTTP services. No. The snippet does not show HTTP instrumentation or propagation configuration for outbound requests. Additional instrumentation is required for end-to-end trace correlation. Yes. Distributed tracing needs inbound and outbound instrumentation so context is created and propagated across services.

Microsoft Learn: Instrument app opentelemetry: configure spans traces · Instrument app opentelemetry: add opentelemetry sdk · Azure: monitor opentelemetry exporter readme

90 Multiple response · Develop containerized solutions on Azure

ANSWERS: A, F

- **A** Push the revised code to the source repository.
- **F** Create and configure an ACR Task that uses a source repository context.

Committing the updated source code is the event that starts the automated build. The case study requires builds to start from source code commits. An ACR Task with the repository context defines the build and monitors the Git source for changes. That satisfies the requirement to use ACR Tasks.

WHY THE OTHER CHOICES ARE WRONG

- **B.** A base image update trigger can rebuild when a parent image changes, but the requirement is for builds that start from source code commits.
- **C.** Deploying to AKS occurs after the image is built. The question is about the CI image build workflow, not a CD deployment step.
- **D.** GitHub Actions can run CI/CD workflows, but the case study specifies ACR Tasks for automated builds.
- **E.** Azure DevOps could build containers, but it introduces a separate pipeline tool instead of using the required ACR Tasks approach.

Microsoft Learn: Store manage containers azure container registry: build run acr tasks · Container registry: container registry tasks overview

91 Drop-down selection · Develop containerized solutions on Azure

ANSWER

Scale from the number of queued messages.	**Azure Service Bus scale rule**
Remove workload instances when the queue has no messages.	**Minimum replicas set to 0**
Limit burst scaling to a controlled value.	**Maximum replicas set to 10**

Scale from the number of queued messages.: KEDA can query Azure Service Bus for activeMessageCount and scale directly from queue length, which matches the case study requirement. Remove workload instances when the queue has no messages.: A minimum replica count of 0 lets the event-driven workload scale down completely when there are no messages. Limit burst scaling to a controlled value.: A fixed maximum replica count caps scale-out so cost and resource usage remain predictable during queue spikes. A minimum replica count sets only the floor.

Microsoft Learn: Aks: keda about · Scale containers azure container apps: event driven scaling keda · Container apps: scale app

92 Drag and drop · Develop containerized solutions on Azure

ANSWER

Confirm whether a Service's selector matches any running Pods.	**Review Kubernetes service endpoints.**
Find the application error that makes a container crash and restart.	**Review container logs.**
Check the readiness probe settings (path, port, and thresholds) of a Pod that never becomes ready.	**Review Pod descriptions.**

Confirm whether a Service's selector matches any running Pods.: Checking service endpoints confirms whether the Service selector is matching healthy pods and routing traffic to them. Find the application error that makes a container crash and restart.: Container logs, especially from the previous instance, show runtime exceptions or missing settings that commonly cause restarts. Check the readiness probe settings (path, port, and thresholds) of a Pod that never becomes ready.: A Pod description lists the configured readiness probe (path, port, timing, and thresholds) along with its recent failures.

Microsoft Learn: Monitor apps azure kubernetes service: troubleshoot pods services · Deploy apps azure kubernetes service: deploy verify results

93 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: D

D Add connection pooling.

Connection pooling reuses established database connections, reducing connection creation overhead and improving concurrent throughput.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Read replicas can scale read volume, but they do not reduce the overhead of repeatedly opening database connections.
- **B.** shared_buffers improves data caching and query execution, not the cost of creating and managing connections.
- **C.** A higher max_connections value permits more sessions but can add contention and does not make connection setup cheaper.

Microsoft Learn: Connectivity: concepts connection pooling best practices · Connectivity: concepts pgbouncer

94 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Satisfy the 200-ms semantic search latency target.	**Scale compute vCores.**
Keep high-dimensional pgvector indexes resident in memory.	**Allocate more memory.**
Handle ongoing ingestion of transaction-derived embeddings.	**Enable storage autoscale.**

Satisfy the 200-ms semantic search latency target.: Vector similarity comparisons depend heavily on CPU. More vCores provide additional compute capacity for high-dimensional distance calculations. Keep high-dimensional pgvector indexes resident in memory.: More memory helps keep pgvector indexes and active result sets cached, reducing the need for disk-based temporary work. Handle ongoing ingestion of transaction-derived embeddings.: Storage autoscale expands capacity as embedding data grows and can increase available IOPS as storage increases.

Microsoft Learn: Optimize vector search azure database postgresql: scale high volume workloads · Optimize vector search azure database postgresql: tune postgresql pgvector · Scale: how to auto grow storage

95 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Meet the authentication policy for database access.	**Use managed identity authentication.**
Serve high-concurrency requests with low latency.	**Use a connection pooling library.**
Maintain database stability during traffic spikes.	**Set a maximum pool size.**

Meet the authentication policy for database access.: Managed identity authentication allows the Azure resource to authenticate with Microsoft Entra ID and avoids storing database credentials. Serve high-concurrency requests with low latency.: A connection pooling library keeps warm connections available and reduces repeated handshake latency. Maintain database stability during traffic spikes.: A capped pool limits concurrent database connections so spikes do not exhaust database CPU and memory.

Microsoft Learn: Security: security connect with managed identity · Optimize vector search azure database postgresql: connection optimization · Connectivity: concepts pgbouncer

96 Multiple choice · Develop AI solutions by using Azure data management services

ANSWER: C

C Create a B-tree index on metadata filter columns.

A B-tree index on filtered metadata lets PostgreSQL narrow the row set before running vector comparisons, reducing compute and latency.

WHY THE OTHER CHOICES ARE WRONG

- **A.** More shared_buffers can improve caching, but it does not reduce how many vector comparisons pgvector must compute.
- **B.** More connections permit additional sessions, but they do not optimize the plan or reduce CPU cycles for vector distance calculations.
- **D.** IVFFlat helps approximate nearest-neighbor search on embeddings, but it does not optimize the metadata filtering stage in this scenario.

Microsoft Learn: Implement vector search azure database postgresql: run vector similarity search semantic retrieval · Extensions: how to optimize performance pgvector

97 Build list · Develop AI solutions by using Azure data management services

CORRECT ORDER: B → D → A → E

1. Define a table schema that includes vector and metadata columns.
2. Load embedding vectors with the related product metadata.
3. Create a B-tree index on the metadata filter columns.
4. Run a similarity search with a WHERE clause and the <=> operator.

The table schema, including the vector type and metadata columns, must exist before data can be inserted or indexed. Embeddings and metadata are loaded after the schema is created and before indexes are applied for best ingestion throughput. After the data is loaded, a B-tree index on metadata columns supports the latency and compute requirements for filtered searches. The query is the final step, using the metadata filter first and the <=> operator for similarity ranking.

WHY THE OTHER CHOICES ARE WRONG

- **C.** Redis memory affects the caching layer, not the core PostgreSQL vector search and metadata filtering workflow.
- **F.** HNSW is a vector index type for embeddings, not an index for metadata columns.

Microsoft Learn: Extensions: how to optimize performance pgvector · Implement vector search azure database postgresql: run vector similarity search semantic retrieval · Azure ai: generative ai semantic search

98 Drag and drop · Develop AI solutions by using Azure data management services

ANSWER

Meet the expiration policy for recommendation results.	**Set a Time to Live on every cache key.**
Preserve catalog freshness and data integrity after updates.	**Publish invalidation events to a Redis channel.**
Meet the latency target for repeated product requests.	**Use the cache-aside pattern with lazy loading.**

Meet the expiration policy for recommendation results.: A TTL automatically removes each key after the configured duration, satisfying the 10-minute expiration requirement. Preserve catalog freshness and data integrity after updates.: Publishing invalidation events lets the application remove stale cache entries as soon as catalog data changes. Meet the latency target for repeated product requests.: Cache-aside stores results in Redis after the first retrieval so repeated requests can be served from cache instead of PostgreSQL.

Microsoft Learn: Implement data operations azure managed redis: implement data operations · Best practices: caching

99 Multiple choice · Secure, monitor, and troubleshoot Azure solutions

ANSWER: A

A Azure Key Vault

Azure Key Vault centralizes secret storage and supports secret lifecycle management and rotation automation. A managed identity lets the Recommendation API read secrets without placing credentials in code or configuration, which meets the Northwind Traders requirements.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Kubernetes secrets keep sensitive values inside a cluster, but they do not provide centralized lifecycle management or automatic rotation across the full platform, including ACA. They also do not address the requirement to use managed identities where possible.
- **C.** Azure App Configuration is intended for application settings and feature flags. By itself, it does not provide the secret protection or automated rotation workflow required for database credentials.
- **D.** Container environment variables expose secrets through process and deployment configuration. They provide no central secret store, audit trail, or automatic rotation mechanism.

Microsoft Learn: Secrets: about secrets · General: secure key vault · General: autorotation

100 Drop-down selection · Develop containerized solutions on Azure

ANSWER

Trigger used for container image builds	**Pull request trigger**
Image build implementation	**ACR Task execution**

Trigger used for container image builds: A pull request trigger runs for a pull request, so changes can be validated before they are merged into the main branch. Image build implementation: Using ACR Tasks runs the build in Azure, so developers do not depend on local machines for image creation.

Microsoft Learn: Store manage containers azure container registry: build run acr tasks

101 Multiple choice · Develop containerized solutions on Azure

ANSWER: C

C Environment variables configured in App Service

App Service configuration can inject environment variables and secrets at runtime without requiring another external service.

WHY THE OTHER CHOICES ARE WRONG

- **A.** GitHub secrets are for pipeline and repository workflows, not for providing runtime secrets directly to the service.
- **B.** Dockerfile ENV instructions bake the secret into the container image, which violates the requirement to keep secrets outside images.
- **D.** Helm charts describe Kubernetes resources and are not the App Service mechanism for securely supplying runtime secrets.

Microsoft Learn: Deploy containers azure app service: configure settings

102 Multiple choice · Develop containerized solutions on Azure

ANSWER: C

C YAML files

YAML is the supported format for declarative, version-controlled Kubernetes deployment manifests.

WHY THE OTHER CHOICES ARE WRONG

- **A.** Azure CLI az aks commands are imperative and do not provide the requested declarative deployment definition.
- **B.** Kubernetes declarative deployment manifests are not authored in XML.
- **D.** kubectl run creates resources directly from command-line arguments, and kubectl create is a command that submits a definition rather than being the definition itself. The requirement calls for declarative definitions kept in version control, which are YAML manifest files.

Microsoft Learn: Deploy apps azure kubernetes service: create deployment manifests · Aks: tutorial kubernetes deploy application

103 Drag and drop · Develop containerized solutions on Azure

ANSWER

Scale rule source	**Azure Service Bus**
Scale trigger metadata	**messageCount**
Minimum replica count	**1**

Scale rule source: Azure Service Bus is the KEDA scaler that drives scaling from the queue events. Scale trigger metadata: messageCount is the Azure Service Bus trigger metadata used to scale from the queue message count. Minimum replica count: Setting min-replicas to 1 keeps one replica running and avoids cold start delays.

Microsoft Learn: Scale containers azure container apps: event driven scaling keda · Container apps: scale app

104 Multiple response · Develop AI solutions by using Azure data management services

ANSWERS: A, C

A Revise the container indexing policy.

C Use lower-precision indexing for the vector fields.

Optimizing the indexing policy so only required properties are indexed reduces RU consumption during similarity queries and helps improve performance. Half-precision indexing or quantization compresses embeddings, which lowers storage, improves query performance, and reduces RU usage.

WHY THE OTHER CHOICES ARE WRONG

- **B.** Strong consistency generally raises latency and RU usage, so it would make the performance problem worse.
- **D.** Composite indexes optimize scalar predicates and ordering, not vector similarity policies. Adding one can increase indexing overhead without helping vector search.

Microsoft Learn: Cosmos db: vector search · Implement vector search azure cosmos db: store retrieve embeddings

105 Drop-down selection · Develop AI solutions by using Azure data management services

ANSWER

Detect document updates that require vectorization.	**Change feed processor**
Expand vectorization processing across workers.	**Lease container**

Detect document updates that require vectorization.: The change feed processor responds to inserts and updates and is intended for event-driven processing. Expand vectorization processing across workers.: A lease container coordinates partition ownership for processor instances and supports scalable distributed processing.

Microsoft Learn: Implement vector search azure cosmos db: change feed trigger embedding refresh · Cosmos db: change feed processor

106 Multiple choice · Connect to and consume Azure services

ANSWER: D

D Azure Cosmos DB trigger

The Azure Cosmos DB trigger invokes the function when items are created or updated in an Azure Cosmos DB for NoSQL container.

WHY THE OTHER CHOICES ARE WRONG

- **A.** An HTTP trigger starts from an incoming HTTP request rather than Azure Cosmos DB for NoSQL item changes.
- **B.** An Event Grid trigger responds to Event Grid events and is not the native trigger for Azure Cosmos DB for NoSQL document updates.
- **C.** A Service Bus trigger runs when a Service Bus message is available, not directly when an Azure Cosmos DB for NoSQL item changes.

Microsoft Learn: Azure functions: functions bindings cosmosdb v2 trigger · Azure functions: scenario database changes azure cosmosdb

107 Multiple choice · Connect to and consume Azure services

ANSWER: B

B GitHub Actions pipeline

A GitHub Actions pipeline can run from a YAML workflow in the repository and orchestrate ARM or Bicep templates together with function app deployment.

WHY THE OTHER CHOICES ARE WRONG

- **A.** An Azure CLI deployment is imperative and does not by itself keep the infrastructure definition declarative and version controlled.
- **C.** Local Git publishing deploys application code but does not declaratively provision or govern the Azure resources for the solution.
- **D.** Azure Functions Core Tools publish commands can deploy code, but they do not provide declarative infrastructure management.

Microsoft Learn: Azure functions: functions how to github actions · Bicep: deploy github actions

---

108 Build list · Secure, monitor, and troubleshoot Azure solutions

CORRECT ORDER (EITHER ORDER IS ACCEPTED): C → E → A OR E → C → A

1. Configure an OpenTelemetry SDK trace exporter.
2. Instrument application code with the OpenTelemetry SDK.
3. Redeploy the services after instrumentation.

or

1. Instrument application code with the OpenTelemetry SDK.
2. Configure an OpenTelemetry SDK trace exporter.
3. Redeploy the services after instrumentation.

A trace exporter provides the destination for emitted spans so correlation can be observed. The application code must use OpenTelemetry instrumentation to create spans and propagate context across Azure Functions, ACA, and AKS workloads. The tracing changes must be deployed before they can affect running services.

WHY THE OTHER CHOICES ARE WRONG

- **B.** An OpenTelemetry view customizes metric aggregation, such as histogram buckets. It does not create or connect distributed traces.
- **D.** TelemetryClient.TrackEvent() writes custom events, not OpenTelemetry spans, and does not provide context propagation across services.

Microsoft Learn: Instrument app opentelemetry: add opentelemetry sdk · Instrument app opentelemetry: configure spans traces · Azure: monitor opentelemetry exporter readme

109 Multiple choice · Secure, monitor, and troubleshoot Azure solutions

ANSWER: A

A Kusto Query Language (KQL) query

Kusto Query Language (KQL) can query and analyze log data in Azure Log Analytics for ad hoc investigation across AKS, Azure Container Apps, and Azure Functions.

WHY THE OTHER CHOICES ARE WRONG

- **B.** A Python function runs application logic. It is not the tool used to query telemetry logs for failed requests.
- **C.** The Azure activity log records resource-level operations. It does not contain the application telemetry needed to find failures in Azure Container Apps or Azure Functions.
- **D.** Azure Monitor workbooks can visualize and analyze telemetry, but building a custom workbook is more work than issuing a KQL query for a one-time analysis.

Microsoft Learn: Analyze telemetry logs metrics: write basic kql queries · Logs: get started queries