

AZ-204^{Q&As}

Developing Solutions for Microsoft Azure

Pass Microsoft AZ-204 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.leads4pass.com/az-204.html>

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



QUESTION 1

You are developing a web application that runs as an Azure Web App. The web application stores data in Azure SQL Database and stores files in an Azure Storage account. The web application makes HTTP requests to external services as part of normal operations.

The web application is instrumented with Application Insights. The external services are OpenTelemetry compliant.

You need to ensure that the customer ID of the signed in user is associated with all operations throughout the overall system.

What should you do?

- A. Add the customer ID for the signed in user to the CorrelationContext in the web application
- B. On the current SpanContext, set the Traceld to the customer ID for the signed in user
- C. Set the header Ocp-Apim-Trace to the customer ID for the signed in user
- D. Create a new SpanContext with the TraceFlags value set to the customer ID for the signed in user

Correct Answer: A

Reference: <https://docs.microsoft.com/en-us/azure/azure-monitor/app/correlation>

QUESTION 2**HOTSPOT**

You are developing an ASP.NET Core time sheet application that runs as an Azure Web App. Users of the application enter their time sheet information on the first day of every month.

The application uses a third-party web service to validate data.

The application encounters periodic server errors due to errors that result from calling a third-party web server. Each request to the third-party server has the same chance of failure.

You need to configure an Azure Monitor alert to detect server errors unrelated to the third-party service. You must minimize false-positive alerts.

How should you complete the Azure Resource Manager template? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

```
"type": "Microsoft.Insights/metricAlerts",
"properties": {
"criteria": {
  "odata.type": "...",
  "allOf": [
    {
      "criterionType": "
      DynamicThresholdCriterion
      SingleResourceMultipleMetricCriteria

      "metricName": "
      Http4xx
      Http5xx

      "alertSensitivity": "
      Low
      High

    ]
  ]
}
```

Correct Answer:

```
"type": "Microsoft.Insights/metricAlerts",
"properties": {
"criteria": {
  "odata.type": "...",
  "allOf": [
    {
      "criterionType": "
      DynamicThresholdCriterion
      SingleResourceMultipleMetricCriteria

      "metricName": "
      Http4xx
      Http5xx

      "alertSensitivity": "
      Low
      High

    ]
  ]
}
```

Box 1: DynamicThresholdCriterion

Box 2: Http5xx Server errors are in the 5xx range. Client errors are in the 4xx range

Box 3: Low

Reference: <https://docs.microsoft.com/en-us/azure/azure-monitor/alerts/alerts-dynamic-thresholds>

QUESTION 3

You are developing a mobile instant messaging app for a company.

The mobile app must meet the following requirements:

Support offline data sync.

Update the latest messages during normal sync cycles.

You need to implement Offline Data Sync.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Retrieve records from Offline Data Sync on every call to the PullAsync method.
- B. Retrieve records from Offline Data Sync using an Incremental Sync.
- C. Push records to Offline Data Sync using an Incremental Sync.
- D. Return the updatedAt column from the Mobile Service Backend and implement sorting by using the column.
- E. Return the updatedAt column from the Mobile Service Backend and implement sorting by the message id.

Correct Answer: BE

B: Incremental Sync: the first parameter to the pull operation is a query name that is used only on the client. If you use a non-null query name, the Azure Mobile SDK performs an incremental sync. Each time a pull operation returns a set of results, the latest updatedAt timestamp from that result set is stored in the SDK local system tables. Subsequent pull operations retrieve only records after that timestamp.

E (not D): To use incremental sync, your server must return meaningful updatedAt values and must also support sorting by this field. However, since the SDK adds its own sort on the updatedAt field, you cannot use a pull query that has its own orderBy clause.

References: <https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-offline-data-sync>

QUESTION 4

DRAG DROP

You develop an Azure solution that uses Cosmos DB.

The current Cosmos DB container must be replicated and must use a partition key that is optimized for queries.

You need to implement a change feed processor solution.

Which change feed processor components should you use? To answer, drag the appropriate components to the correct requirements. Each component may be used once, more than once, or not at all. You may need to drag the split bar

between panes or scroll to view the content.

NOTE: Each correct selection is worth one point.

Select and Place:

Components

Host

Delegate

Lease container

Monitored container

Requirement

Store the data from which the change feed is generated.

Coordinate processing of the change feed across multiple workers.

Use the change feed processor to listen for changes.

Handle each batch of changes.

Component

Component

Component

Component

Component

Correct Answer:

Components

Requirement

Component

Store the data from which the change feed is generated.

Monitored container

Coordinate processing of the change feed across multiple workers.

Lease container

Use the change feed processor to listen for changes.

Host

Handle each batch of changes.

Delegate

Box 1: The monitored container

The monitored container has the data from which the change feed is generated. Any inserts and updates to the monitored container are reflected in the change feed of the container.

Box 2: The lease container

The lease container acts as a state storage and coordinates processing the change feed across multiple workers. The lease container can be stored in the same account as the monitored container or in a separate account.

Box 3: The host: A host is an application instance that uses the change feed processor to listen for changes. Multiple instances with the same lease configuration can run in parallel, but each instance should have a different instance name.

Box 4: The delegate

The delegate is the code that defines what you, the developer, want to do with each batch of changes that the change feed processor reads.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/change-feed-processor>

QUESTION 5

You are developing a .Net web application that stores data in Azure Cosmos DB. The application must use the Core API and allow millions of reads and writes. The Azure Cosmos DB account has been created with multiple write regions enabled. The application has been deployed to the East US2 and Central US region.

You need to update the application to support multi-region writes.

What are two possible ways to achieve this goal? Each correct answer presents parts of the solutions.

NOTE: Each correct selection is worth one point.

- A. Update the ConnectionPolicy class for the Cosmos client and populate the PreferredLocations property based on the geo-proximity of the application.
- B. Update Azure Cosmos DB to use the Strong consistency level. Add indexed properties to the container to indicate region.
- C. Update the ConnectionPolicy class for the Cosmos client and set the UseMultipleWriteLocations property to true.
- D. Create and deploy a custom conflict resolution policy.
- E. Update Azure Cosmos DB to use the Session consistency level. Send the SessionToken property value from the FeedResponse object of the write action to the end-user by using a cookie.

Correct Answer: CD

C: The UseMultipleWriteLocations property of the ConnectionPolicy class gets or sets the flag to enable writes on any locations (regions) for geo-replicated database accounts in the Azure Cosmos DB service.

Note: Once an account has been created with multiple write regions enabled, you must make two changes in your application to the ConnectionPolicy for the Cosmos client to enable the multi-region writes in Azure Cosmos DB. Within the ConnectionPolicy, set UseMultipleWriteLocations to true and pass the name of the region where the application is deployed to ApplicationRegion. This will populate the PreferredLocations property based on the geo-proximity from location passed in. If a new region is later added to the account, the application does not have to be updated or redeployed, it will automatically detect the closer region and will auto-home on to it should a regional event occur.

Azure core API application " ConnectionPolicy class" cosmos db multiple write regions enabled

D: With multi-region writes, when multiple clients write to the same item, conflicts may occur. When a conflict occurs, you can resolve the conflict by using different conflict resolution policies.

Note: Conflict resolution policy can only be specified at container creation time and cannot be modified after container creation.

Reference: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.documents.client.connectionpolicy>
<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-multi-master> <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts>