

DP-420^{Q&As}

Designing and Implementing Cloud-Native Applications Using Microsoft
Azure Cosmos DB

Pass Microsoft DP-420 Exam with 100% Guarantee

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

<https://www.leads4pass.com/dp-420.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 have an Azure Cosmos DB for NoSQL account configured for global distribution across four regions.

At connection time, the SQL SDK needs to identify the optimal endpoint for reading and writing.

Which two factors can influence the SDK? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. the consistency level in RequestOptions
- B. regional latency
- C. the default consistency level
- D. the PreferredLocations configuration
- E. a region being available

Correct Answer: DE

Explanation:

Connecting to a preferred region using the API for NoSQL

The SDKs accept an optional parameter PreferredLocations that is an ordered list of Azure regions.

The SDK will automatically send all writes to the current write region. All reads will be sent to the first available region in the preferred locations list. If the request fails, the client will fail down the list to the next region.

The SDK will only attempt to read from the regions specified in preferred locations. So, for example, if the Azure Cosmos DB account is available in four regions, but the client only specifies two read(non-write) regions within the

PreferredLocations, then no reads will be served out of the read region that is not specified in PreferredLocations. If the read regions specified in the PreferredLocations list are not available, reads will be served out of write region.

Reference: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/tutorial-global-distribution>

QUESTION 2

You have an Azure Cosmos DB for NoSQL account named account1 that supports an application named App1. App1 uses the consistent prefix consistency level.

You configure account1 to use a dedicated gateway and integrated cache.

You need to ensure that App1 can use the integrated cache.

Which two actions should you perform for APP1? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Change the connection mode to direct

- B. Change the account endpoint to <https://account1.sqlx.cosmos.azure.com>.
- C. Change the consistency level of requests to strong.
- D. Change the consistency level of requests to session.
- E. Change the account endpoint to <https://account1.documents.azure.com>

Correct Answer: BD

the Azure Cosmos DB integrated cache is an in-memory cache that is built-in to the Azure Cosmos DB dedicated gateway. The dedicated gateway is a front-end compute that stores cached data and routes requests to the backend database.

You can choose from a variety of dedicated gateway sizes based on the number of cores and memory needed for your workload¹. The integrated cache can reduce the RU consumption and latency of read operations by serving them from the cache instead of the backend containers².

For your scenario, to ensure that App1 can use the integrated cache, you should perform these two actions:

Change the account endpoint to <https://account1.sqlx.cosmos.azure.com>. This is the dedicated gateway endpoint that you need to use to connect to your Azure Cosmos DB account and leverage the integrated cache. The standard gateway

endpoint (<https://account1.documents.azure.com>) will not use the integrated cache².

Change the consistency level of requests to session. This is the highest consistency level that is supported by the integrated cache. If you use a higher consistency level (such as strong or bounded staleness), your requests will bypass the

integrated cache and go directly to the backend containers

QUESTION 3

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 in the review screen.

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to make the contents of container1 available as reference data for an Azure Stream Analytics job.

Solution: You create an Azure Synapse pipeline that uses Azure Cosmos DB Core (SQL) API as the input and Azure Blob Storage as the output.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: B

Instead create an Azure function that uses Azure Cosmos DB Core (SQL) API change feed as a trigger and Azure event hub as the output.

The Azure Cosmos DB change feed is a mechanism to get a continuous and incremental feed of records from an Azure Cosmos container as those records are being created or modified. Change feed support works by listening to container for any changes. It then outputs the sorted list of documents that were changed in the order in which they were modified.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/changefeed-ecommerce-solution>

QUESTION 4

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 in the review screen.

You have a database in an Azure Cosmos DB for NoSQL account that is configured for multi-region writes.

You need to use the Azure Cosmos DB SDK to implement the conflict resolution policy for a container. The solution must ensure that any conflicts are sent to the conflicts feed.

Solution: You set ConflictResolutionMode to LastWriterWins and you use the default settings for the policy.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: A

QUESTION 5

You have an Azure Cosmos DB Core (SQL) API account.

You run the following query against a container in the account.

```
SELECT  
  
IS_NUMBER("1234") AS A,  
  
IS_NUMBER(1234) AS B,  
  
IS_NUMBER({prop: 1234}) AS C
```

What is the output of the query?

- A. [{"A": false, "B": true, "C": false}]
- B. [{"A": true, "B": false, "C": true}]
- C. [{"A": true, "B": true, "C": false}]

D. [{"A": true, "B": true, "C": true}]

Correct Answer: A

IS_NUMBER returns a Boolean value indicating if the type of the specified expression is a number. "1234" is a string, not a number.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/sql-query-is-number>

QUESTION 6

DRAG DROP

You have an Azure Cosmos DB Core (SQL) API account that is configured for multi-region writes. The account contains a database that has two containers named container1 and container2.

The following is a sample of a document in container1:

```
{  
  "customerId": 1234,  
  "firstName": "John",  
  "lastName": "Smith",  
  "policyYear": 2021  
}
```

The following is a sample of a document in container2:

```
{  
  "gpsId": 1234,  
  "latitude": 38.8951,  
  "longitude": -77.0364  
}
```

You need to configure conflict resolution to meet the following requirements:

1.
For container1 you must resolve conflicts by using the highest value for policyYear.
2.
For container2 you must resolve conflicts by accepting the distance closest to latitude: 40.730610 and longitude: -73.935242.
- 3.

Administrative effort must be minimized to implement the solution.

What should you configure for each container? To answer, drag the appropriate configurations to the correct containers. Each configuration may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:

Configurations

- Last Write Wins (default) mode
- Merge Procedures (custom) mode
- An application that reads from the conflicts feed

Answer Area

Container1:

Container2:

Correct Answer:

Configurations

-
-
- An application that reads from the conflicts feed

Answer Area

Container1:

Container2:

Box 1: Last Write Wins (LWW) (default) mode

Last Write Wins (LWW): This resolution policy, by default, uses a system-defined timestamp property. It's based on the time-synchronization clock protocol.

Box 2: Merge Procedures (custom) mode

Custom: This resolution policy is designed for application-defined semantics for reconciliation of conflicts. When you set this policy on your Azure Cosmos container, you also need to register a merge stored procedure. This procedure is

automatically invoked when conflicts are detected under a database transaction on the server. The system provides exactly once guarantee for the execution of a merge procedure as part of the commitment protocol.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/conflict-resolution-policies>

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-manage-conflicts>

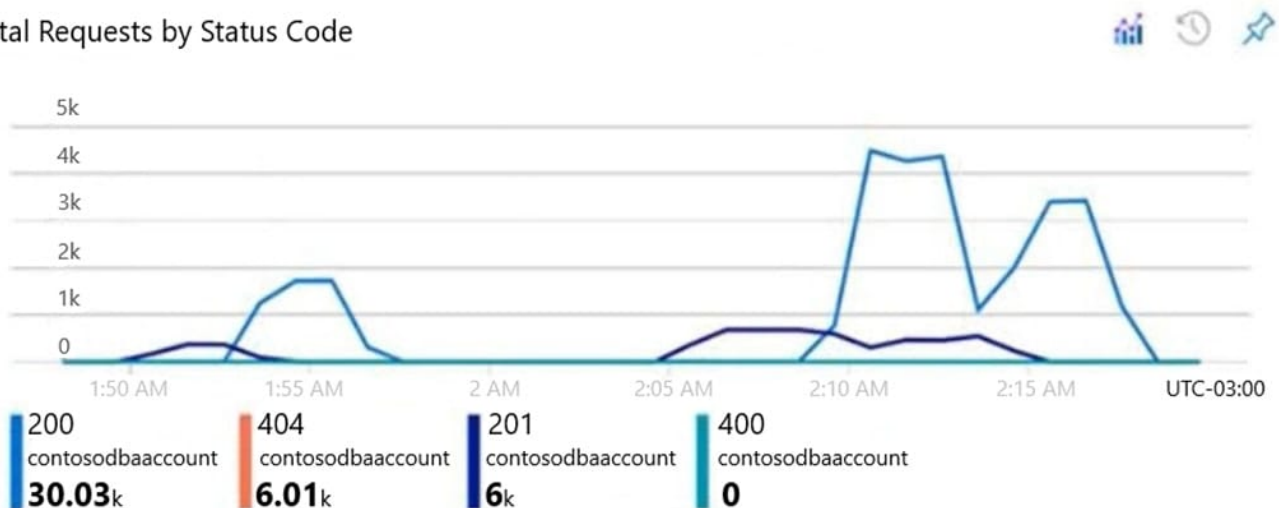
QUESTION 7

HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account used by an application named App1.

You open the Insights pane for the account and see the following chart.

Total Requests by Status Code



Use the drop-down menus to select the answer choice that answers each question based on the information presented in the graphic.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

The HTTP 404 status code is caused by [answer choice]

	▼
incorrect connection URLs	
an intermittent firewall issue	
incorrectly formatted partition keys	
requesting resources that do not exist	

There are [answer choice] successful resource creations in the account during the time period of the chart

	▼
zero	
6 thousand	
6.01 thousand	
30.03 thousand	
36.03 thousand	

Correct Answer:

Answer Area

The HTTP 404 status code is caused by [answer choice]

	▼
incorrect connection URLs	
an intermittent firewall issue	
incorrectly formatted partition keys	
requesting resources that do not exist	

There are [answer choice] successful resource creations in the account during the time period of the chart

	▼
zero	
6 thousand	
6.01 thousand	
30.03 thousand	
36.03 thousand	

Box 1: incorrect connection URLs

400 Bad Request: Returned when there is an error in the request URI, headers, or body. The response body will contain an error message explaining what the specific problem is.

The HyperText Transfer Protocol (HTTP) 400 Bad Request response status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (for example, malformed request syntax, invalid request message framing, or deceptive request routing).

Box 2: 6 thousand

201 Created: Success on PUT or POST. Object created or updated successfully.

Note:

200 OK: Success on GET, PUT, or POST. Returned for a successful response.

404 Not Found: Returned when a resource does not exist on the server. If you are managing or querying an index, check the syntax and verify the index name is specified correctly.

Reference:

<https://docs.microsoft.com/en-us/rest/api/searchservice/http-status-codes>

QUESTION 8

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account.

You need to provide a user named User1 with the ability to insert items into container1 by using role-based access control (RBAC). The solution must use the principle of least privilege.

Which roles should you assign to User1?

- A. CosmosDB Operator only
- B. DocumentDB Account Contributor and Cosmos DB Built-in Data Contributor
- C. DocumentDB Account Contributor only
- D. Cosmos DB Built-in Data Contributor only

Correct Answer: A

Cosmos DB Operator: Can provision Azure Cosmos accounts, databases, and containers. Cannot access any data or use Data Explorer. Incorrect Answers:

B: DocumentDB Account Contributor can manage Azure Cosmos DB accounts. Azure Cosmos DB is formerly known as DocumentDB.

C: DocumentDB Account Contributor: Can manage Azure Cosmos DB accounts.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/role-based-access-control>

QUESTION 9

HOTSPOT

You have an Azure Cosmos DB Core (SQL) account that has a single write region in West Europe.

You run the following Azure CLI script.

```
az cosmosdb update -n $accountName -g $resourceGroupName \  
  --locations regionName='West Europe' failoverPriority=0 isZoneRedundant=False \  
  --locations regionName='North Europe' failoverPriority=1 isZoneRedundant=False  
  
az cosmosdb failover-priority-change -n $accountName -g $resourceGroupName \  
  --failover-policies 'North Europe=0' 'West Europe=1'
```

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

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input type="radio"/>	<input type="radio"/>

Correct Answer:

Answer Area

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	<input checked="" type="radio"/>	<input type="radio"/>
After running the script, the Azure Cosmos DB instance in West Europe will be writable	<input type="radio"/>	<input checked="" type="radio"/>
The cost of the Azure Cosmos DB account is unaffected by running the script	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: Yes

The Automatic failover option allows Azure Cosmos DB to failover to the region with the highest failover priority with no user action should a region become unavailable.

Box 2: No

West Europe is used for failover. Only North Europe is writable.

To Configure multi-region set UseMultipleWriteLocations to true.

Box 3: Yes

Provisioned throughput with single write region costs \$0.008/hour per 100 RU/s and provisioned throughput with

multiple writable regions costs \$0.016/per hour per 100 RU/s.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-multi-master> <https://docs.microsoft.com/en-us/azure/cosmos-db/optimize-cost-regions>

QUESTION 10

You have an Azure Cosmos DB for NoSQL account1 that is configured for automatic failover. The account1 account has a single read-write region in West US and a and a read region in East US.

You run the following PowerShell command.

```
Update-AzCosmosDBAccountFailoverPriority -ResourceGroupName `rg1` -Name `account1` -FailoverPolicy @(`East US`, `West US`)
```

What is the effect of running the command?

- A. A manual failover will occur.
- B. The account will be unavailable to writes during the change.
- C. The provisioned throughput for account1 will increase.
- D. The account will be configured for multi-region writes.

Correct Answer: D

You can use the Set-AzCosmosDBAccountRegion cmdlet to update the regions that an Azure Cosmos DB account uses. You can use this cmdlet to add a region or change the region failover order. The cmdlet requires a resource group name, an Azure Cosmos DB account name, and a list of regions in desired failover order¹. For your scenario, based on the PowerShell command, you are using the Set- AzCosmosDBAccountRegion cmdlet to update the regions for an Azure Cosmos DB account named account1 that is configured for automatic failover. The command specifies two regions: West US and East US. The effect of running the command is that the account will be configured for multi-region writes.

Multi-region writes is a feature of Azure Cosmos DB that allows you to write data to any region in your account and have it automatically replicated to all other regions. This feature provides high availability and low latency for write operations across multiple regions. To enable multi-region writes, you need to specify at least two regions in your account and set them as write regions². In your command, you are setting both West US and East US as write regions by using the -IsZoneRedundant parameter with a value of \$true for both regions.

QUESTION 11

You are troubleshooting the current issues caused by the application updates.

Which action can address the application updates issue without affecting the functionality of the application?

- A. Enable time to live for the con-product container.
- B. Set the default consistency level of account1 to strong.
- C. Set the default consistency level of account1 to bounded staleness.

D. Add a custom indexing policy to the con-product container.

Correct Answer: C

Bounded staleness is frequently chosen by globally distributed applications that expect low write latencies but require total global order guarantee. Bounded staleness is great for applications featuring group collaboration and sharing, stock ticker, publish-subscribe/queueing etc.

Scenario: Application updates in con-product frequently cause HTTP status code 429 "Too many requests". You discover that the 429 status code relates to excessive request unit (RU) consumption during the updates.

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

QUESTION 12

You have an Azure Cosmos DB for NoSQL account.

You need to create an Azure Monitor query that lists recent modifications to the regional failover policy.

Which Azure Monitor table should you query?

- A. CDBPartitionKeyStatistics
- B. CDBQueryRunTimeStatistics
- C. CDBDataPlaneRequests
- D. CDBControlPlaneRequests

Correct Answer: D

QUESTION 13

You have a database in an Azure Cosmos DB for NoSQL account that is configured for multi-region writes.

You need to use the Azure Cosmos DB SDK to implement the conflict resolution policy for a container. The solution must ensure that any conflict sent to the conflict feed.

Solution: You set ConflictResolutionMode to Custom and you use the default settings for the policy.

Does this meet the goal?

- A. Yes
- B. No

Correct Answer: B

Setting ConflictResolutionMode to Custom and using the default settings for the policy will not ensure that conflicts are sent to the conflict feed. You need to define a custom stored procedure using the "conflictingItems" parameter to handle conflicts properly.

QUESTION 14

You have a container named container1 in an Azure Cosmos DB for NoSQL account named account1 that is set to the session default consistency level. The average size of an item in container1 is 20 KB.

You have an application named App1 that uses the Azure Cosmos DB SDK and performs a point read on the same set of items in container1 every minute.

You need to minimize the consumption of the request units (RUs) associated to the reads by App1.

What should you do?

- A. In account1, change the default consistency level to bounded staleness.
- B. In App1, change the consistency level of read requests to consistent prefix.
- C. In account1, provision a dedicated gateway and integrated cache
- D. In App1, modify the connection policy settings.

Correct Answer: B

The cost of a point read for a 1 KB item is 1 RU. The cost of other operations depends on factors such as item size, indexing policy, consistency level, and query complexity¹. To minimize the consumption of RUs, you can optimize these

factors according to your application needs.

For your scenario, one possible way to minimize the consumption of RUs associated to the reads by App1 is to change the consistency level of read requests to consistent prefix. Consistent prefix is a lower consistency level than session,

which is the default consistency level for Azure Cosmos DB. Lower consistency levels consume fewer RUs than higher consistency levels². Consistent prefix guarantees that reads never see out-of-order writes and that monotonic reads are

preserved¹. This may be suitable for your application if you can tolerate some eventual consistency.

QUESTION 15

You have a container in an Azure Cosmos DB for NoSQL account. You need to create an alert based on a custom Log Analytics query. Which signal type should you use?

- A. Log
- B. Metrics
- C. Activity Log
- D. Resource Health

Correct Answer: A

Explanation:

You can receive an alert based on the metrics, activity log events, or Log Analytics logs on your Azure Cosmos DB

account:

*-> Log Analytics logs– This alert triggers when the value of a specified property in the results of a Log Analytics query crosses a threshold you assign. For example, you can write a Log Analytics query to monitor if the storage for a logical partition key is reaching the 20 GB logical partition key storage limit in Azure Cosmos DB.

Incorrect:

Metrics - The alert triggers when the value of a specified metric crosses a threshold you assign. For example, when the total request units consumed exceed 1000 RU/s. This alert is triggered both when the condition is first met and then afterwards when that condition is no longer being met.

Activity log events – This alert triggers when a certain event occurs. For example, when the keys of your Azure Cosmos DB account are accessed or refreshed.

Reference:

<https://learn.microsoft.com/en-us/azure/cosmos-db/create-alerts>

[Latest DP-420 Dumps](#)

[DP-420 Exam Questions](#)

[DP-420 Braindumps](#)