

Microsoft

Exam Questions DP-420

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



NEW QUESTION 1

- (Exam Topic 1)

You configure multi-region writes for account1.

You need to ensure that App1 supports the new configuration for account1. The solution must meet the business requirements and the product catalog requirements.

What should you do?

- A. Set the default consistency level of account1 to bounded staleness.
- B. Create a private endpoint connection.
- C. Modify the connection policy of App1.
- D. Increase the number of request units per second (RU/s) allocated to the con-product and con-productVendor containers.

Answer: D

Explanation:

App1 queries the con-product and con-productVendor containers.

Note: Request unit is a performance currency abstracting the system resources such as CPU, IOPS, and memory that are required to perform the database operations supported by Azure Cosmos DB.

Scenario:

Develop an app named App1 that will run from all locations and query the data in account1.

Once multi-region writes are configured, maximize the performance of App1 queries against the data in account1.

Whenever there are multiple solutions for a requirement, select the solution that provides the best performance, as long as there are no additional costs associated.

Reference:

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

NEW QUESTION 2

- (Exam Topic 2)

You have a container in an Azure Cosmos DB Core (SQL) API account. The container stores telemetry data from IoT devices. The container uses telemetryId as the partition key and has a throughput of 1,000 request units per second (RU/s). Approximately 5,000 IoT devices submit data every five minutes by using the same telemetryId value.

You have an application that performs analytics on the data and frequently reads telemetry data for a single IoT device to perform trend analysis.

The following is a sample of a document in the container.

```
{
  "id" : "9ccf1906-2a30-4dc0-9644-2185f5dcbbd7",
  "deviceId" : "bba6fe24-6d97-4935-8d58-36baa4b8a0e1",
  "telemetryId" : "9d7816e6-f401-42ba-ad05-0e03de35c0b8",
  "date" : "2019-05-03",
  "time" : "13:05",
  "temp" : "21"
}
```

You need to reduce the amount of request units (RUs) consumed by the analytics application. What should you do?

- A. Decrease the offerThroughput value for the container.
- B. Increase the offerThroughput value for the container.
- C. Move the data to a new container that has a partition key of deviceId.
- D. Move the data to a new container that uses a partition key of date.

Answer: C

Explanation:

The partition key is what will determine how data is routed in the various partitions by Cosmos DB and needs to make sense in the context of your specific scenario. The IoT Device ID is generally the "natural" partition key for IoT applications.

Reference: <https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/iot-using-cosmos-db>

NEW QUESTION 3

- (Exam Topic 2)

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 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

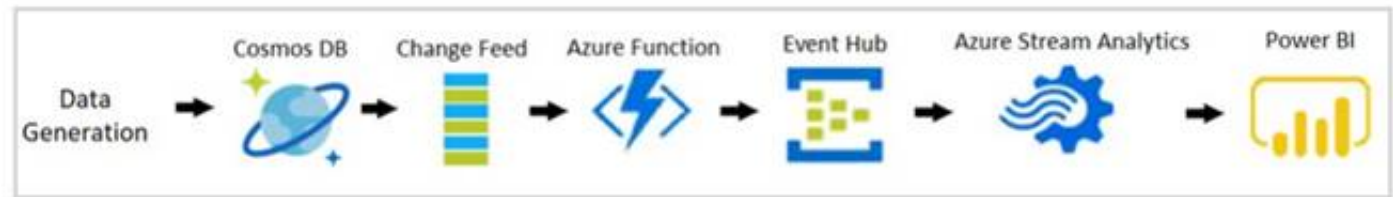
Answer: B

Explanation:

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.

The following diagram represents the data flow and components involved in the solution:



C:\Users\Admin\Desktop\Data\Odt

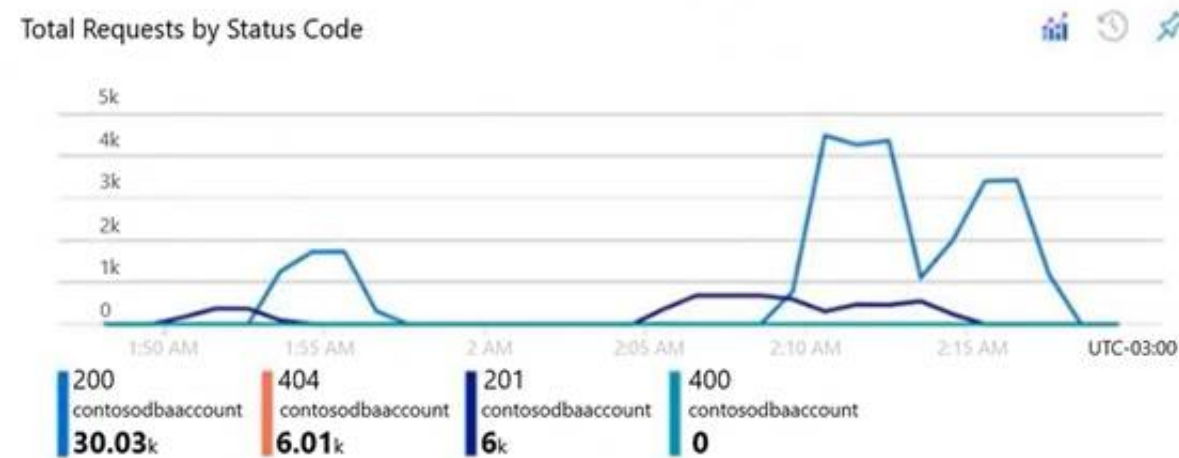
data\Untitled.jpg

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

NEW QUESTION 4

- (Exam Topic 2)

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.



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.

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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>

NEW QUESTION 5

- (Exam Topic 2)

You need to configure an Apache Kafka instance to ingest data from an Azure Cosmos DB Core (SQL) API account. The data from a container named telemetry must be added to a Kafka topic named iot. The solution must store the data in a compact binary format.

Which three configuration items should you include in the solution? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. "connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSourceConnector"
- B. "key.converter": "org.apache.kafka.connect.json.JsonConverter"
- C. "key.converter": "io.confluent.connect.avro.AvroConverter"
- D. "connect.cosmos.containers.topicmap": "iot#telemetry"
- E. "connect.cosmos.containers.topicmap": "iot"
- F. "connector.class": "com.azure.cosmos.kafka.connect.source.CosmosDBSinkConnector"

Answer: CDF

Explanation:

C: Avro is binary format, while JSON is text.
F: Kafka Connect for Azure Cosmos DB is a connector to read from and write data to Azure Cosmos DB. The Azure Cosmos DB sink connector allows you to export data from Apache Kafka topics to an Azure Cosmos DB database. The connector polls data from Kafka to write to containers in the database based on the topics subscription.
D: Create the Azure Cosmos DB sink connector in Kafka Connect. The following JSON body defines config for the sink connector.
Extract:
"connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector", "key.converter": "org.apache.kafka.connect.json.AvroConverter"
"connect.cosmos.containers.topicmap": "hotels#kafka"
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/sql/kafka-connector-sink https://www.confluent.io/blog/kafka-connect-deep-dive-converters-serialization-explained/

NEW QUESTION 6

- (Exam Topic 2)
You have the indexing policy shown in the following exhibit.

SQL API

ItemsSettings

Test

Scale

families

Items

Settings

Stored Procedures

User Defined Functions

Triggers

Settings

Indexing Policy

1 {

2 "indexingMode": "consistent",

3 "automatic": true,

4 "includedPaths": [

5 {

6 "path": "/surname/?"

7 }

8 },

9 "excludedPaths": [

10 {

11 "path": "/*"

12 }

13 },

14 "compositeIndexes": [

15 [

16 {

17 "path": "/name"

18 },

19 {

20 "path": "/age"

21 }

22]

23]

24 }

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.

Answer Area

When creating a query, which ORDER BY statement will execute successfully?

ORDER BY c.age ASC, c.name ASC

ORDER BY c.age DESC, c.name DESC

ORDER BY c.name ASC, c.age DESC

ORDER BY c.name DESC, c.age ASC

ORDER BY c.name DESC, c.age DESC

During the creation of an item, when will the index update?

Never

At a scheduled interval

At the same time as the item creation

After the item appears in the change feed

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Box 1: ORDER BY c.name DESC, c.age DESC
Queries that have an ORDER BY clause with two or more properties require a composite index.
The following considerations are used when using composite indexes for queries with an ORDER BY clause with two or more properties:
If the composite index paths do not match the sequence of the properties in the ORDER BY clause, then the composite index can't support the query.
The order of composite index paths (ascending or descending) should also match the order in the ORDER BY clause.
The composite index also supports an ORDER BY clause with the opposite order on all paths. Box 2: At the same time as the item creation
Azure Cosmos DB supports two indexing modes:
Consistent: The index is updated synchronously as you create, update or delete items. This means that the consistency of your read queries will be the

consistency configured for the account.
 None: Indexing is disabled on the container.
 Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/index-policy>

NEW QUESTION 7

- (Exam Topic 2)

You have an Azure Cosmos DB Core (SQL) API account named account1 that has the disableKeyBasedMetadataWriteAccess property enabled. You are developing an app named App1 that will be used by a user named DevUser1 to create containers in account1. DevUser1 has a non-privileged user account in the Azure Active Directory (Azure AD) tenant. You need to ensure that DevUser1 can use App1 to create containers in account1. What should you do? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area

Grant permissions to create containers by using:

<input type="checkbox"/>	Account keys
<input type="checkbox"/>	Resource tokens
<input type="checkbox"/>	Role-based access control (RBAC)

Create containers by using the:

<input type="checkbox"/>	Azure AD Graph API
<input type="checkbox"/>	Azure Resource Manager API
<input type="checkbox"/>	SQL (Core) API

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Box 1: Resource tokens
 Resource tokens provide access to the application resources within a database. Resource tokens: Provide access to specific containers, partition keys, documents, attachments, stored procedures, triggers, and UDFs.
 Box 2: Azure Resource Manager API
 You can use Azure Resource Manager to help deploy and manage your Azure Cosmos DB accounts, databases, and containers.
 Reference:
<https://docs.microsoft.com/en-us/azure/cosmos-db/secure-access-to-data> <https://docs.microsoft.com/en-us/rest/api/resources/>

NEW QUESTION 8

- (Exam Topic 2)

You need to implement a trigger in Azure Cosmos DB Core (SQL) API that will run before an item is inserted into a container. Which two actions should you perform to ensure that the trigger runs? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- A. Append pre to the name of the JavaScript function trigger.
- B. For each create request, set the access condition in RequestOptions.
- C. Register the trigger as a pre-trigger.
- D. For each create request, set the consistency level to session in RequestOptions.
- E. For each create request, set the trigger name in RequestOptions.

Answer: C

Explanation:

C: When triggers are registered, you can specify the operations that it can run with.
 F: When executing, pre-triggers are passed in the RequestOptions object by specifying PreTriggerInclude and then passing the name of the trigger in a List object.
 Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-use-stored-procedures-triggers-udfs>

NEW QUESTION 9

- (Exam Topic 2)

You are implementing an Azure Data Factory data flow that will use an Azure Cosmos DB (SQL API) sink to write a dataset. The data flow will use 2,000 Apache Spark partitions. You need to ensure that the ingestion from each Spark partition is balanced to optimize throughput. Which sink setting should you configure?

- A. Throughput
- B. Write throughput budget
- C. Batch size
- D. Collection action

Answer: C

Explanation:

Batch size: An integer that represents how many objects are being written to Cosmos DB collection in each batch. Usually, starting with the default batch size is sufficient. To further tune this value, note: Cosmos DB limits single request's size to 2MB. The formula is "Request Size = Single Document Size * Batch Size". If you hit error saying "Request size is too large", reduce the batch size value. The larger the batch size, the better throughput the service can achieve, while make sure you allocate enough RUs to empower your workload.
 Reference: <https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db>

NEW QUESTION 10

- (Exam Topic 2)

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.

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"/>

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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>

NEW QUESTION 10

- (Exam Topic 2)

You have a database in an Azure Cosmos DB Core (SQL) API account. The database is backed up every two hours. You need to implement a solution that supports point-in-time restore. What should you do first?

- A. Enable Continuous Backup for the account.
- B. Configure the Backup & Restore settings for the account.
- C. Create a new account that has a periodic backup policy.
- D. Configure the Point In Time Restore settings for the account.

Answer: A

Explanation:

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/provision-account-continuous-backup>

NEW QUESTION 15

- (Exam Topic 2)

You are creating a database in an Azure Cosmos DB Core (SQL) API account. The database will be used by an application that will provide users with the ability to share online posts. Users will also be able to submit comments on other users' posts. You need to store the data shown in the following table.

Type	Description
Users	Information about a user who will use the application
Posts	Text of up to 1,000 characters that a user will share with other users
Comments	Text of up to 280 characters that users will submit as a comment on a post
Interests	Information about a user's interests

The application has the following characteristics: Users can submit an unlimited number of posts. The average number of posts submitted by a user will be more than 1,000. Posts can have an unlimited number of comments from different users. The average number of comments per post will be 100, but many posts will exceed 1,000 comments. Users will be limited to having a maximum of 20 interests. For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth one point.

Answer Area

Statements	Yes	No
If you embed the posts data into the users data instead of creating a separate document for each post, you will increase the write operation costs for new posts	<input type="radio"/>	<input type="radio"/>
If you embed the comments data into the posts data instead of creating a separate document for each comment you will increase the write operation costs for new comments	<input type="radio"/>	<input type="radio"/>
If you embed the interests data into the users data instead of creating a separate document for each interest, you will increase the read operation costs for displaying the users and their associated interests	<input type="radio"/>	<input type="radio"/>

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Box 1: Yes
Non-relational data increases write costs, but can decrease read costs.
Box 2: Yes
Non-relational data increases write costs, but can decrease read costs.
Box 3: No
Non-relational data increases write costs, but can decrease read costs.

NEW QUESTION 17

- (Exam Topic 2)

You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. Upserts of items in container1 occur every three seconds.

You have an Azure Functions app named function1 that is supposed to run whenever items are inserted or replaced in container1.

You discover that function1 runs, but not on every upsert.

You need to ensure that function1 processes each upsert within one second of the upsert. Which property should you change in the Function.json file of function1?

- A. checkpointInterval
B. leaseCollectionsThroughput
C. maxItemsPerInvocation
D. feedPollDelay

Answer: D

Explanation:

With an upsert operation we can either insert or update an existing record at the same time.

FeedPollDelay: The time (in milliseconds) for the delay between polling a partition for new changes on the feed, after all current changes are drained. Default is 5,000 milliseconds, or 5 seconds.

Reference: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-cosmosdb-v2-trigger>

NEW QUESTION 19

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

DP-420 Practice Exam Features:

- * DP-420 Questions and Answers Updated Frequently
- * DP-420 Practice Questions Verified by Expert Senior Certified Staff
- * DP-420 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * DP-420 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year

100% Actual & Verified — Instant Download, Please Click
[Order The DP-420 Practice Test Here](#)