



Microsoft

Exam Questions DP-420

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

About ExamBible

Your Partner of IT Exam

Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

NEW QUESTION 1

- (Exam Topic 1)

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.

Answer: C

Explanation:

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>

NEW QUESTION 2

- (Exam Topic 1)

You need to select the partition key for con-iot1. The solution must meet the IoT telemetry requirements. What should you select?

- A. the timestamp
- B. the humidity
- C. the temperature
- D. the device ID

Answer: D

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.

Scenario: The iotdb database will contain two containers named con-iot1 and con-iot2. Ensure that Azure Cosmos DB costs for IoT-related processing are predictable. Reference:

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

NEW QUESTION 3

- (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 4

- (Exam Topic 2)

You have an Azure Cosmos DB Core (SQL) API account named account1.

In account1, you run the following query in a container that contains 100GB of data. SELECT *

FROM c

WHERE LOWER(c.categoryid) = "hockey"

You view the following metrics while performing the query.

Retrieved Document Count	:	45,654
Retrieved Document Size	:	543,765,234 bytes
Output Document Count	:	12
Output Document Size	:	451 bytes
Index Utilization	:	0.00 %
Total Query Execution Time	:	2,400.34 milliseconds
Query Preparation Times		
Query Compilation Time	:	0.09 milliseconds
Logical Plan Build Time	:	0.04 milliseconds
Physical Plan Build Time	:	0.03 milliseconds
Query Optimization Time	:	0.01 milliseconds
Index Lookup Time	:	0.00 milliseconds
Document Load Time	:	3,167.26 milliseconds
Runtime Execution Times		
Query Engine Times	:	299.16 milliseconds
System Function Execution Time	:	79.34 milliseconds
User-defined Function Execution Time	:	0.00 milliseconds
Document Write Time	:	0.01 milliseconds
Client Side Metrics		
Retry Count	:	0
Request Charge	:	3,898.95 RUs

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
The query performs a cross-partition query	☐	☐
The query uses an index	☐	☐
Recreating the container with the partition key set to /categoryId will improve the performance of the query	☐	☐

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Box 1: No
Each physical partition should have its own index, but since no index is used, the query is not cross-partition.
Box 2: No
Index utilization is 0% and Index Look up time is also zero.
Box 3: Yes
A partition key index will be created, and the query will perform across the partitions. Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/how-to-query-container>

NEW QUESTION 5

- (Exam Topic 2)
You have a container named container1 in an Azure Cosmos DB Core (SQL) API account. The following is a sample of a document in container1.

```
{
  "studentId": "631282", "firstName": "James", "lastName": "Smith", "enrollmentYear": 1990, "isActivelyEnrolled": true, "address": {
    "street": "",
    "city": "",
    "stateProvince": "",
    "postal": ""
  }
}
```

The container1 container has the following indexing policy.

```
{
  "indexingMode": "consistent", "includePaths": [
    {
      "path": "/*"
    },
    {
      "path": "/address/city/?"
    }
  ],
  "excludePaths": [
    {
      "path": "/address/*"
    },
    {
      "path": "/firstName/?"
    }
  ]
}
```

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
The <code>/isActivelyEnrolled</code> property is included in the index	☐	☐
The <code>/firsrtname</code> property is included in the index	☐	☐
The <code>/address/city</code> property is included in the index	☐	☐

- A. Mastered
B. Not Mastered

Answer: A

Explanation:

Box 1: Yes

"path": "/"* is in includePaths.

Include the root path to selectively exclude paths that don't need to be indexed. This is the recommended approach as it lets Azure Cosmos DB proactively index any new property that may be added to your model.

Box 2: No

"path": "/firstName/?" is in excludePaths.

Box 3: Yes

"path": "/address/city/?" is in includePaths

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

NEW QUESTION 6

- (Exam Topic 2)

You maintain a relational database for a book publisher. The database contains the following tables.

Name	Column
Author	authorId (primary key)
	fullname
	address
	contactinfo
Book	bookId (primary key)
	isbn
	title
	genre
Bookauthorlnk	authorId (foreign key)
	bookId (foreign key)

The most common query lists the books for a given authorId.

You need to develop a non-relational data model for Azure Cosmos DB Core (SQL) API that will replace the relational database. The solution must minimize latency and read operation costs.

What should you include in the solution?

- A. Create a container for Author and a container for Boo
B. In each Author document, embed bookId for each book by the autho
C. In each Book document embed authorId of each author.
D. Create Author, Book, and Bookauthorlnk documents in the same container.
E. Create a container that contains a document for each Author and a document for each Boo
F. In each Book document, embed authorId.
G. Create a container for Author and a container for Boo
H. In each Author document and Book document embed the data from Bookauthorlnk.

Answer: A

Explanation:

Store multiple entity types in the same container.

NEW QUESTION 7

- (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.

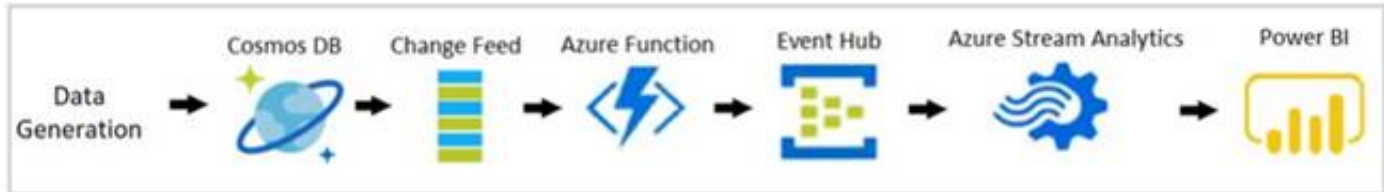
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

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 8
- (Exam Topic 2)

You have three containers in an Azure Cosmos DB Core (SQL) API account as shown in the following table.

Name	Database	Time to Live
cn1	db1	On (no default)
cn2	db1	Off
cn3	db1	On (no default)

You have the following Azure functions:
A function named Fn1 that reads the change feed of cn1
A function named Fn2 that reads the change feed of cn2
A function named Fn3 that reads the change feed of cn3
You perform the following actions: Delete an item named item1 from cn1. Update an item named item2 in cn2. For an item named item3 in cn3, update the item time to live to 3,600 seconds.
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
Fn1 will receive item1 from the change feed	☐	☐
Fn2 can check the _etag of item2 to see whether the item is an update or an insert	☐	☐
Fn3 will receive item3 from the change feed	☐	☐

A. Mastered
B. Not Mastered

Answer: A

Explanation:
Box 1: No
Azure Cosmos DB's change feed is a great choice as a central data store in event sourcing architectures where all data ingestion is modeled as writes (no updates or deletes).
Note: The change feed does not capture deletes. If you delete an item from your container, it is also removed from the change feed. The most common method of handling this is adding a soft marker on the items that are being deleted. You can add a property called "deleted" and set it to "true" at the time of deletion. This document update will show up in the change feed. You can set a TTL on this item so that it can be automatically deleted later.
Box 2: No
The _etag format is internal and you should not take dependency on it, because it can change anytime.
Box 3: Yes
Change feed support in Azure Cosmos DB works by listening to an Azure Cosmos container for any changes. Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/change-feed-design-patterns> <https://docs.microsoft.com/en-us/azure/cosmos-db/change-feed>

NEW QUESTION 9
- (Exam Topic 2)

You have a container in an Azure Cosmos DB Core (SQL) API account. You need to use the Azure Cosmos DB SDK to replace a document by using optimistic concurrency. What should you include in the code? To answer, select the appropriate options in the answer area. NOTE: Each correct selection is worth one point.

Answer Area

RequestOptions property to set:

	▼
AccessCondition	
ConsistencyLevel	
SessionToken	

Document property that will be compared:

	▼
_etag	
_id	
_rid	

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Box 1: ConsistencyLevel

The ItemRequestOptions Class ConsistencyLevel property gets or sets the consistency level required for the request in the Azure Cosmos DB service.

Azure Cosmos DB offers 5 different consistency levels. Strong, Bounded Staleness, Session, Consistent Prefix and Eventual - in order of strongest to weakest consistency.

Box 2: _etag

The ItemRequestOptions class helped us implement optimistic concurrency by specifying that we wanted the SDK to use the If-Match header to allow the server to decide whether a resource should be updated. The If-Match value is the ETag value to be checked against. If the ETag value matches the server ETag value, the resource is updated.

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.itemrequestoptions> <https://cosmosdb.github.io/labs/dotnet/labs/10-concurrency-control.html>

NEW QUESTION 10

- (Exam Topic 2)

You have the following query.

```
SELECT * FROM
```

```
WHERE c.sensor = "TEMP1"
```

```
AND c.value < 22
```

```
AND c.timestamp >= 1619146031231
```

You need to recommend a composite index strategy that will minimize the request units (RUs) consumed by the query.

What should you recommend?

- A. a composite index for (sensor ASC, value ASC) and a composite index for (sensor ASC, timestamp ASC)
- B. a composite index for (sensor ASC, value ASC, timestamp ASC) and a composite index for (sensor DESC, value DESC, timestamp DESC)
- C. a composite index for (value ASC, sensor ASC) and a composite index for (timestamp ASC, sensor ASC)
- D. a composite index for (sensor ASC, value ASC, timestamp ASC)

Answer: A

Explanation:

If a query has a filter with two or more properties, adding a composite index will improve performance. Consider the following query:

```
SELECT * FROM c WHERE c.name = "Tim" and c.age > 18
```

In the absence of a composite index on (name ASC, and age ASC), we will utilize a range index for this query. We can improve the efficiency of this query by creating a composite index for name and age.

Queries with multiple equality filters and a maximum of one range filter (such as >, <, <=, >=, !=) will utilize the composite index.

Reference:

<https://azure.microsoft.com/en-us/blog/three-ways-to-leverage-composite-indexes-in-azure-cosmos-db/>

NEW QUESTION 10

- (Exam Topic 2)

You are developing an application that will use an Azure Cosmos DB Core (SQL) API account as a data source.

You need to create a report that displays the top five most ordered fruits as shown in the following table.

Name	Type	Orders
apple	fruit	1,000
orange	fruit	600
banana	fruit, exotic	400
plum	fruit.	300
mango	fruit, exotic	200

A collection that contains aggregated data already exists. The following is a sample document:

```
{
  "name": "apple",
  "type": ["fruit", "exotic"], "orders": 10000
}
```

```
}
```

Which two queries can you use to retrieve data for the report? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A)

```
SELECT TOP i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.orders,i.types
```

B)

```
SELECT TOP i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.orders DESC
```

C)

```
SELECT TOP i.name, i.types, i.orders
FROM items i
WHERE EXISTS(SELECT VALUE t FROM t IN i.types WHERE t.name = 'fruit')
ORDER BY i.types DESC
```

D)

```
SELECT TOP i.name, i.types, i.orders
FROM items i
WHERE ARRAY_CONTAINS(i.types, {name: 'fruit'})
ORDER BY i.orders DESC
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: BD

Explanation:

ARRAY_CONTAINS returns a Boolean indicating whether the array contains the specified value. You can check for a partial or full match of an object by using a boolean expression within the command.

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

NEW QUESTION 12

- (Exam Topic 2)

You have an Azure Cosmos DB Core (SQL) API account that uses a custom conflict resolution policy. The account has a registered merge procedure that throws a runtime exception.

The runtime exception prevents conflicts from being resolved. You need to use an Azure function to resolve the conflicts. What should you use?

- A. a function that pulls items from the conflicts feed and is triggered by a timer trigger
- B. a function that receives items pushed from the change feed and is triggered by an Azure Cosmos DB trigger
- C. a function that pulls items from the change feed and is triggered by a timer trigger
- D. a function that receives items pushed from the conflicts feed and is triggered by an Azure Cosmos DB trigger

Answer: D

Explanation:

The Azure Cosmos DB Trigger uses the Azure Cosmos DB Change Feed to listen for inserts and updates across partitions. The change feed publishes inserts and updates, not deletions.

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

NEW QUESTION 16

- (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 20

- (Exam Topic 2)

You are designing an Azure Cosmos DB Core (SQL) API solution to store data from IoT devices. Writes from the devices will be occur every second.

The following is a sample of the data.

```
{
  "id" : "03c1ca5a-dbl8-4231-908f-09a9bc7a7c3e",
  "deviceManufacturer" : "Contoso, Ltd",
  "deviceId" : "f460df85-799f-4d58-b051-67561b4993c6",
  "timestamp" : "2021-09-19T13:47:45",
  "sensor1Value" : true,
  "sensor2Value" : "75",
  "sensor3Value" : "4554",
  "sensor4Value" : "454",
  "sensor5Value" : "42128"
}
```

You need to select a partition key that meets the following requirements for writes: Minimizes the partition skew
Avoids capacity limits Avoids hot partitions What should you do?

- A. Use timestamp as the partition key.
- B. Create a new synthetic key that contains deviceId and sensor1Value.
- C. Create a new synthetic key that contains deviceId and deviceManufacturer.
- D. Create a new synthetic key that contains deviceId and a random number.

Answer: D

Explanation:

Use a partition key with a random suffix. Distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/sql/synthetic-partition-keys>

NEW QUESTION 24

- (Exam Topic 2)

You have an application named App1 that reads the data in an Azure Cosmos DB Core (SQL) API account. App1 runs the same read queries every minute. The default consistency level for the account is set to eventual.

You discover that every query consumes request units (RUs) instead of using the cache.

You verify the IntegratedCacheItemHitRate metric and the IntegratedCacheQueryHitRate metric. Both metrics have values of 0.

You verify that the dedicated gateway cluster is provisioned and used in the connection string. You need to ensure that App1 uses the Azure Cosmos DB integrated cache.

What should you configure?

- A. the indexing policy of the Azure Cosmos DB container
- B. the consistency level of the requests from App1
- C. the connectivity mode of the App1 CosmosClient
- D. the default consistency level of the Azure Cosmos DB account

Answer: C

Explanation:

Because the integrated cache is specific to your Azure Cosmos DB account and requires significant CPU and memory, it requires a dedicated gateway node. Connect to Azure Cosmos DB using gateway mode.

Reference: <https://docs.microsoft.com/en-us/azure/cosmos-db/integrated-cache-faq>

NEW QUESTION 29

- (Exam Topic 2)

You plan to deploy two Azure Cosmos DB Core (SQL) API accounts that will each contain a single database. The accounts will be configured as shown in the following table.

Name	Description
development	- Supports the development of new application features - Used intermittently as needed during development
shipments	- Captures over 100,000 updates per second generated at unpredictable rimes throughout the business day - Used with Azure Synapse Link for analytics

How should you provision the containers within each account to minimize costs? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

development:

▼

Serverless capacity mode

Provisioned throughput capacity mode and manual throughput

Provisioned throughput capacity mode and autoscale throughput

shipments:

▼

Serverless capacity mode

Provisioned throughput capacity mode and manual throughput

Provisioned throughput capacity mode and autoscale throughput

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Box 1: Serverless capacity mode

Azure Cosmos DB serverless best fits scenarios where you expect intermittent and unpredictable traffic with long idle times. Because provisioning capacity in such situations isn't required and may be cost-prohibitive, Azure Cosmos DB serverless should be considered in the following use-cases:

Getting started with Azure Cosmos DB

Running applications with bursty, intermittent traffic that is hard to forecast, or low (<10%) average-to-peak traffic ratio

Developing, testing, prototyping and running in production new applications where the traffic pattern is unknown

Integrating with serverless compute services like Azure Functions

Box 2: Provisioned throughput capacity mode and autoscale throughput The use cases of autoscale include:

Variable or unpredictable workloads: When your workloads have variable or unpredictable spikes in usage, autoscale helps by automatically scaling up and down based on usage. Examples include retail websites that have different traffic patterns depending on seasonality; IOT workloads that have spikes at various times during the day; line of business applications that see peak usage a few times a month or year, and more. With autoscale, you no longer need to manually provision for peak or average capacity.

Reference:

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

<https://docs.microsoft.com/en-us/azure/cosmos-db/provision-throughput-autoscale#use-cases-of-autoscale>

NEW QUESTION 32

- (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.

Answer Area

Statements	Yes	No
After running the script, there will be an instance of Azure Cosmos DB in North Europe that is writable	☐	☐
After running the script, the Azure Cosmos DB instance in West Europe will be writable	☐	☐
The cost of the Azure Cosmos DB account is unaffected by running the script	☐	☐

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 35

- (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 39

- (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 an Azure Cosmos DB Core (SQL) API account named account1 that uses autoscale throughput. You need to run an Azure function when the normalized request units per second for a container in account1 exceeds a specific value.

Solution: You configure an Azure Monitor alert to trigger the function. Does this meet the goal?

A. Yes

B. No

Answer: A

Explanation:

You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal. Note: Alerts are used to set up recurring tests to monitor the availability and responsiveness of your Azure

Cosmos DB resources. Alerts can send you a notification in the form of an email, or execute an Azure Function when one of your metrics reaches the threshold or if a specific event is logged in the activity log.

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

NEW QUESTION 43

.....

Relate Links

100% Pass Your DP-420 Exam with ExamBible Prep Materials

<https://www.exambible.com/DP-420-exam/>

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>