

Microsoft

DP-420

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

- Up to Date products, reliable and verified.
- Questions and Answers in PDF Format.

Full Version Features:

- 90 Days Free Updates
- 30 Days Money Back Guarantee
- Instant Download Once Purchased
- 24 Hours Live Chat Support

For More Information:

<https://www.testsexpert.com/>

• Product Version

Visit us at <https://www.testsexpert.com/dp-420/>

Latest Version: 9.0

Question: 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>

Question: 2

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>

Question: 3

You need to identify which connectivity mode to use when implementing App2. The solution must support the planned changes and meet the business requirements.

Which connectivity mode should you identify?

- A. Direct mode over HTTPS
- B. Gateway mode (using HTTPS)
- C. Direct mode over TCP

Answer: C

Explanation:

Scenario: Develop an app named App2 that will run from the retail stores and query the data in account2. App2 must be limited to a single DNS endpoint when accessing account2.

By using Azure Private Link, you can connect to an Azure Cosmos account via a private endpoint. The private endpoint is a set of private IP addresses in a subnet within your virtual network.

When you're using Private Link with an Azure Cosmos account through a direct mode connection, you can use only the TCP protocol. The HTTP protocol is not currently supported.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-private-endpoints>

Question: 4

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 conproduct Vendor 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>

Question: 5

You need to provide a solution for the Azure Functions notifications following updates to con-product.

The solution must meet the business requirements and the product catalog requirements.

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

NOTE: Each correct selection is worth one point.

- A. Configure the trigger for each function to use a different leaseCollectionPrefix
- B. Configure the trigger for each function to use the same leaseCollectionName
- C. Configure the trigger for each function to use a different leaseCollectionName
- D. Configure the trigger for each function to use the same leaseCollectionPrefix

Answer: AC

Explanation:

leaseCollectionPrefix: when set, the value is added as a prefix to the leases created in the Lease collection for this Function. Using a prefix allows two separate Azure Functions to share the same Lease collection by using different prefixes.

Scenario: Use Azure Functions to send notifications about product updates to different recipients.

Trigger the execution of two Azure functions following every update to any document in the con-product container.

Reference:

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

For More Information – Visit link below:
<https://www.testsexpert.com/>

16\$ Discount Coupon: **9M2GK4NW**

Features:

■ Money Back Guarantee.....



■ 100% Course Coverage.....



■ 90 Days Free Updates.....



■ Instant Email Delivery after Order.....

