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.pass2lead.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 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"

Correct Answer: CDF

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"

Incorrect Answers:

B: JSON is plain text.

Note, full example:

{ "name": "cosmosdb-sink-connector", "config": {

"connector.class": "com.azure.cosmos.kafka.connect.sink.CosmosDBSinkConnector",

"tasks.max": "1",

"topics": [

"hotels"

],

"value.converter": "org.apache.kafka.connect.json.AvroConverter",

"value.converter.schemas.enable": "false",

"key.converter": "org.apache.kafka.connect.json.AvroConverter",

"key.converter.schemas.enable": "false",

"connect.cosmos.connection.endpoint": "https://.documents.azure.com:443/",

"connect.cosmos.master.key": "",

"connect.cosmos.databasename": "kafkaconnect",

"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/

QUESTION 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

Correct Answer: C

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.

Incorrect Answers:

A: Throughput: Set an optional value for the number of RUs you\\'d like to apply to your CosmosDB collection for each execution of this data flow. Minimum is 400.

B: Write throughput budget: An integer that represents the RUs you want to allocate for this Data Flow write operation, out of the total throughput allocated to the collection.

D: Collection action: Determines whether to recreate the destination collection prior to writing.

None: No action will be done to the collection. Recreate: The collection will get dropped and recreated

Reference: https://docs.microsoft.com/en-us/azure/data-factory/connector-azure-cosmos-db

QUESTION 3

You have an Azure Cosmos DB for NoSQL account that has multiple write regions.

You need to receive an alert when requests that target the database exceed the available request units per second (RU/s).

Which Azure Monitor signal should you use?

A. Region Removed

B. Document Quota

C. Metadata Requests

D. Data Usage

Correct Answer: B

Azure Monitor is a service that provides comprehensive monitoring for Azure resources, including Azure Cosmos DB. You can use Azure Monitor to collect, analyze, and alert on metrics and logs from your Azure Cosmos DB account. You can create alerts for Azure Cosmos DB using Azure Monitor based on the metrics, activity log events, or Log Analytics logs on your account1. For your scenario, if you want to receive an alert when requests that target the database exceed the available request units per second (RU/s), you should use the Document Quota metric. This metric measures the percentage of RU/s consumed by your account or container. You can create an alert rule on this metric from the Azure portal by following these steps2: In the Azure portal, select the Azure Cosmos DB account you want to monitor. Under the Monitoring section of the sidebar, select Alerts, and then select New alert rule. In the Create alert rule pane, fill out the Scope section by selecting your subscription name and resource type (Azure Cosmos DB accounts). In the Condition section, select Add condition and choose Document Quota from the list of signals. In the Configure signal logic pane, specify the threshold value and operator for your alert condition. For example, you can choose Greater than or equal to 90 as the threshold value and operator to receive an alert when your RU/s consumption reaches 90% or more of your provisioned throughput. In the Alert rule details section, specify a name and description for your alert rule. In the Actions section, select Add action group and choose how you want to receive notifications for your alert. For example, you can choose Email/SMS/Push/Voice as an action type and enter your email address or phone number as a receiver. Review your alert rule settings and select Create alert rule to save it.

QUESTION 4

You have the following query.

SELECT * FROM ? WHERE c.sensor = "TEMP1" AND c.value = 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)

Correct Answer: A

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 >,