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.passapply.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 implement a solution to meet the product catalog requirements. What should you do to implement the conflict resolution policy.

A. Remove frequently changed field from the index policy of the con-product container.

B. Disable indexing on all fields in the index policy of the con-product container.

C. Set the default consistency level for account1 to eventual.

D. Create a new container and migrate the product catalog data to the new container.

Correct Answer: D

QUESTION 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"

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 3

You have an Azure Cosmos DB account named account1.

You have several apps that connect to account1 by using the account\\'s secondary key.

You then configure the apps to authenticate by using service principals.

You need to ensure that account1 will only allow apps to connect by using an Azure AD identity.

Which account property should you modify?

A. disableKeyBasedMetadataWriteAccess ,

B. disableLocalAuth

C. userAssignedIdentatxe

D. allowedOrxgins

Correct Answer: B

The disableLocalAuth property is a boolean flag that indicates whether local authentication methods such as primary/secondary keys are disabled for the Azure Cosmos DB account. Setting this property to true improves security by ensuring that Azure Cosmos DB accounts exclusively require Azure Active Directory identities for authentication1.

QUESTION 4

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 5

You have a container m an Azure Cosmos DB for NoSQL account.

Data update volumes are unpredictable.

You need to process the change teed of the container by using a web app that has multiple instances. The change feed will be processed by using the change feed processor from the Azure Cosmos DB SDK. The multiple instances must share the workload.

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

NOTE: Each correct selection is worth one point.

A. Configure the same processor name for all the instances.

B. Configure a different processor name for each instance.

C. Configure a different lease container configuration for each instance.

D. Configure the same instance name for all the instances. 13

E. Configure a different instance name for each instance.

F. Configure the same lease container configuration for all the instances.

Correct Answer: AEF