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.geekcert.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 have an Azure Cosmos DB Core (SQL) API account.

You configure the diagnostic settings to send all log information to a Log Analytics workspace.

You need to identify when the provisioned request units per second (RU/s) for resources within the account were modified.

You write the following query.

AzureDiagnostics

| where Category == "ControlPlaneRequests"

What should you include in the query?

A. | where OperationName startswith "AccountUpdateStart"

B. | where OperationName startswith "SqlContainersDelete"

C. | where OperationName startswith "MongoCollectionsThroughputUpdate"

D. | where OperationName startswith "SqlContainersThroughputUpdate"

Correct Answer: A

The following are the operation names in diagnostic logs for different operations:

1.

RegionAddStart, RegionAddComplete

2.

RegionRemoveStart, RegionRemoveComplete

3.

AccountDeleteStart, AccountDeleteComplete

4.

RegionFailoverStart, RegionFailoverComplete

5.

AccountCreateStart, AccountCreateComplete

6.

AccountUpdateStart, AccountUpdateComplete

7.

VirtualNetworkDeleteStart, VirtualNetworkDeleteComplete

8.

DiagnosticLogUpdateStart, DiagnosticLogUpdateComplete

Reference: https://docs.microsoft.com/en-us/azure/cosmos-db/audit-control-plane-logs

---

QUESTION 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 and 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.

Correct Answer: A

When creating a new Azure Cosmos DB account, in the Backup policy tab, choose continuous mode to enable the point in time restore functionality for the new account. With the point-in-time restore, data is restored to a new account, currently you can\\'t restore to an existing account.

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

---

QUESTION 3

You develop an application that uses Azure Cosmos DB Core (SQL) API.

You create an Azure pipeline to build and deploy the application.

You need to change the pipeline to run integration tests that you wrote for the application. The solution must execute entirely in the pipeline.

What should you add to the pipeline?

A. a deployment group named Cosmos DB testing

B. an Azure Cosmos DB Emulator task

C. a NuGet service connection that uses an Azure Cosmos DB API key

D. a secret variable that has a connection string to an Azure Cosmos DB database

Correct Answer: B

Explanation:

Set up a CI/CD pipeline with the Azure Cosmos DB Emulator build task in Azure DevOps

The Azure Cosmos DB Emulator provides a local environment that emulates the Azure Cosmos DB service for development purposes. The emulator allows you to develop and test your application locally, without creating an Azure

subscription or incurring any costs.

Reference:

https://learn.microsoft.com/en-us/azure/cosmos-db/tutorial-setup-ci-cd

QUESTION 4

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 5

You have a container named container1 in an Azure Cosmos DB for NoSQL account named account1 that is set to the session default consistency level. The average size of an item in container1 is 20 KB.

You have an application named App1 that uses the Azure Cosmos DB SDK and performs a point read on the same set of items in container1 every minute.

You need to minimize the consumption of the request units (RUs) associated to the reads by App1.

What should you do?

A. In account1, change the default consistency level to bounded staleness.

B. In App1, change the consistency level of read requests to consistent prefix.

C. In account1, provision a dedicated gateway and integrated cache

D. In App1, modify the connection policy settings.

Correct Answer: B

The cost of a point read for a 1 KB item is 1 RU. The cost of other operations depends on factors such as item size, indexing policy, consistency level, and query complexity1. To minimize the consumption of RUs, you can optimize these

factors according to your application needs.

For your scenario, one possible way to minimize the consumption of RUs associated to the reads by App1 is to change the consistency level of read requests to consistent prefix. Consistent prefix is a lower consistency level than session,

which is the default consistency level for Azure Cosmos DB. Lower consistency levels consume fewer RUs than higher consistency levels2. Consistent prefix guarantees that reads never see out-of-order writes and that monotonic reads are

preserved1. This may be suitable for your application if you can tolerate some eventual consistency.