

## DP-420<sup>Q&As</sup>

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.leads4pass.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

### HOTSPOT

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

You have the Azure virtual networks and subnets shown in the following table.

Subnet	Network	IP address range	Virtual machine
subnet1	vnet1	10.0.0.0/24	VM1
subnet2	vnet1	10.0.1.0/24	VM2
subnet3	vnet2	10.1.0.0/24	VM3

The vnet1 and vnet2 networks are connected by using a virtual network peer.

The Firewall and virtual network settings for account1 are configured as shown in the exhibit.

Allow access from

- All networks  Selected networks

Configure network security for your Azure Cosmos DB account. [Learn more.](#)

Virtual networks

Secure your Azure Cosmos DB account with virtual networks. [+ Add existing virtual network](#) [+Add new virtual network](#)

Virtual Network	Subnet	Address range	Endpoint Status
∨ vnet1	1	10.0.0.0/16	
	vnet1.subnet1	10.0.1.0/24	✓ Enabled

Firewall

Add IP ranges to allow access from the internet or your on-premises networks. [+Add my current IP](#) ⓘ

**IP(Single IPv4 or CIDR range)**

Exceptions

- Accept connections from within public Azure datacenters ⓘ  
 Allow access from Azure Portal ⓘ

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

## Answer Area

Statements	Yes	No
VM1 can access account 1	<input type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input type="radio"/>

Correct Answer:

## Answer Area

Statements	Yes	No
VM1 can access account 1	<input checked="" type="radio"/>	<input type="radio"/>
VM2 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>
VM3 can access account 1	<input type="radio"/>	<input checked="" type="radio"/>

Box 1: Yes

VM1 is on vnet1.subnet1 which has the Endpoint Status enabled.

Box 2: No

Only virtual network and their subnets added to Azure Cosmos account have access. Their peered VNets cannot access the account until the subnets within peered virtual networks are added to the account.

Box 3: No

Only virtual network and their subnets added to Azure Cosmos account have access.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-configure-vnet-service-endpoint>

## QUESTION 2

You have an application that queries an Azure Cosmos DB for NoSQL account.

You discover that the following two queries run frequently,

```
SELECT * FROM c WHERE c.name = @name ORDER BY c.name
DESC, c.timestamp DESC
```

```
SELECT * FROM c WHERE c.name = @name AND c.timestamp
ORDER BY c.name ASC, c.timestamp ASC
```

You need to minimize the request units (RUs) consumed by reads and writes. What should you create?

- A. a composite index for (name DESC, time stamp ASC)
- B. a composite index for (name ASC, time stamp DESC)
- C. a composite index for (name ASC time stamp ASC) and a composite index for (name, time stamp disc)
- D. a composite index for (name ASC, time stamp ASC)

Correct Answer: D

## QUESTION 3

### HOTSPOT

You have an Azure Cosmos DB Core (SQL) API account named storage1 that uses provisioned throughput capacity mode.

The storage1 account contains the databases shown in the following table.

Name	Throughput	Max request units per second (RU/s)	Geo-redundancy	Multi-region writes	Number of regions
db1	Autoscale	5,000	Disabled	Disabled	1
db2	Autoscale	8,000	Enabled	Enabled	3

The databases contain the containers shown in the following table.

Name	Database	Throughput
cn01	db1	Container - autoscale maximum RU/s of 10,000
cn02	db1	Database
cn03	db1	Database
cn04	db1	Database
cn05	db1	Database
cn11	db2	Database
cn12	db2	Database
cn13	db2	Database
cn14	db2	Database
cn15	db2	Database
cn16	db2	Database
cn17	db2	Database
cn18	db2	Database

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

## Answer Area

Statements	Yes	No
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input type="radio"/>
To db2, you can add a new container that uses database throughput	<input type="radio"/>	<input type="radio"/>

Correct Answer:

## Answer Area

Statements	Yes	No
At a minimum, you will be billed for 4,000 RU/s per hour for db1	<input type="radio"/>	<input checked="" type="radio"/>
The maximum throughput that can be consumed by cn11 is 400 RU/s	<input type="radio"/>	<input checked="" type="radio"/>
To db2, you can add a new container that uses database throughput	<input checked="" type="radio"/>	<input type="radio"/>

Box 1: No

Four containers with 1000 RU/s each.

Box 2: No

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container.

Box 3: Yes

Max 8000 RU/s for db2. 8 containers, so 1000 RU/s for each container. Can very well add an additional container.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/plan-manage-costs>

<https://azure.microsoft.com/en-us/pricing/details/cosmos-db/>

### 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

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 account 1 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 the function to have an Azure CosmosDB trigger.

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

Instead configure an Azure Monitor alert to trigger the function.

You can set up alerts from the Azure Cosmos DB pane or the Azure Monitor service in the Azure portal.

Reference:

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

[DP-420 Study Guide](#)

[DP-420 Exam Questions](#)

[DP-420 Braindumps](#)