AZ-204 Q&As

Developing Solutions for Microsoft Azure

Pass Microsoft AZ-204 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/az-204.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 are creating an Azure key vault using PowerShell. Objects deleted from the key vault must be kept for a set period of 90 days. Which two of the following parameters must be used in conjunction to meet the requirement? (Choose two.)

A. EnabledForDeployment

B. EnablePurgeProtection

C. EnabledForTemplateDeployment

D. EnableSoftDelete

Correct Answer: BD

Reference: https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/new-azurermkeyvault

https://docs.microsoft.com/en-us/azure/key-vault/key-vault-ovw-soft-delete

QUESTION 2

You are developing a mobile instant messaging app for a company.

The mobile app must meet the following requirements:

Support offline data sync.

Update the latest messages during normal sync cycles.

You need to implement Offline Data Sync.

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

NOTE: Each correct selection is worth one point.

A. Retrieve records from Offline Data Sync on every call to the PullAsync method.

B. Retrieve records from Offline Data Sync using an Incremental Sync.

C. Push records to Offline Data Sync using an Incremental Sync.

D. Return the updatedAt column from the Mobile Service Backend and implement sorting by using the column.

E. Return the updatedAt column from the Mobile Service Backend and implement sorting by the message id.

Correct Answer: BE

B: Incremental Sync: the first parameter to the pull operation is a query name that is used only on the client. If you use a non-null query name, the Azure Mobile SDK performs an incremental sync. Each time a pull operation returns a set of results, the latest updatedAt timestamp from that result set is stored in the SDK local system tables. Subsequent pull operations retrieve only records after that timestamp.

E (not D): To use incremental sync, your server must return meaningful updatedAt values and must also support sorting

by this field. However, since the SDK adds its own sort on the updatedAt field, you cannot use a pull query that has its own orderBy clause.

References: https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-offline-data-sync

QUESTION 3

HOTSPOT

You are developing a ticket reservation system for an airline.

The storage solution for the application must meet the following requirements:

1.

Ensure at least 99.99% availability and provide low latency.

2.

Accept reservations event when localized network outages or other unforeseen failures occur.

3.

Process reservations in the exact sequence as reservations are submitted to minimize overbooking or selling the same seat to multiple travelers.

4.

Allow simultaneous and out-of-order reservations with a maximum five-second tolerance window.

You provision a resource group named airlineResourceGroup in the Azure South-Central US region.

You need to provision a SQL SPI Cosmos DB account to support the app.

How should you complete the Azure CLI commands? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

```
resourceGroupName- +airlineResourceGroup'
name- +docdb-airline-reservations'
databaseName- 'docdb-tickets-database'
collectionName- 'docdb-tickets-collection'
consistencyLevel-
```

- Strong
- Eventual
- ConsistentPrefix
- BoundedStaleness

```
az cosmosdb create \
--name $name \
```

- --enable-virtual-network true\
- --enable-automatic-failover true\
- --kind 'GlobalDocumentDB' \
- --kind 'MongoDB'\

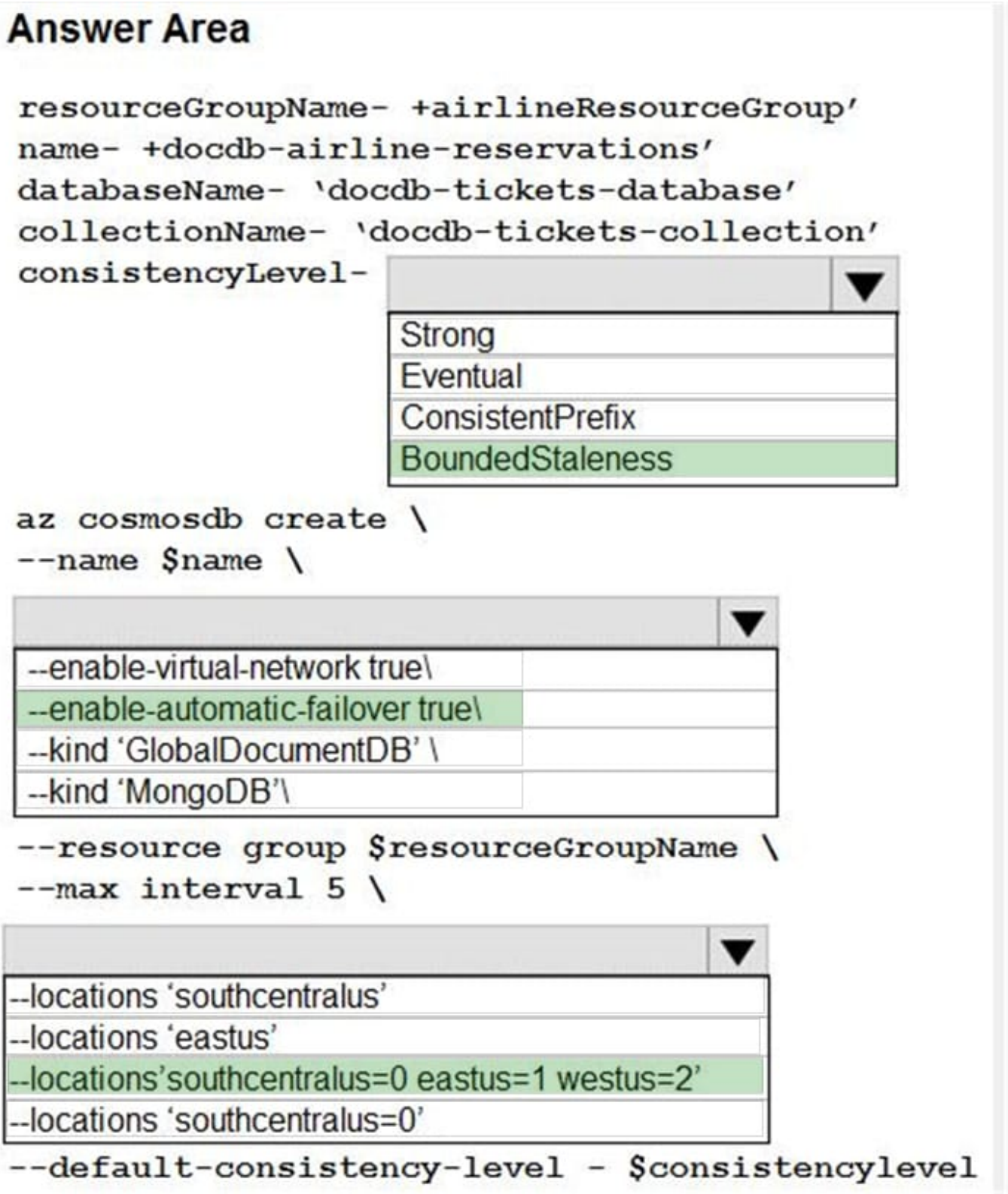
```
--resource group $resourceGroupName \
--max interval 5 \
```

- --locations 'southcentralus'
- --locations 'eastus'
- --locations'southcentralus=0 eastus=1 westus=2'
- --locations 'southcentralus=0'

```
--default-consistency-level - $consistencylevel
```

Correct Answer:

Answer Area



Box 1: BoundedStaleness

Bounded staleness: The reads are guaranteed to honor the consistent-prefix guarantee. The reads might lag behind writes by at most "K" versions (that is, "updates") of an item or by "T" time interval. In other words, when you choose

bounded staleness, the "staleness" can be configured in two ways:

The number of versions (K) of the item

The time interval (T) by which the reads might lag behind the writes

Incorrect Answers:

Strong

Strong consistency offers a linearizability guarantee. Linearizability refers to serving requests concurrently. The reads are guaranteed to return the most recent committed version of an item. A client never sees an uncommitted or partial write.

Users are always guaranteed to read the latest committed write.

Box 2: --enable-automatic-failover true\

For multi-region Cosmos accounts that are configured with a single-write region, enable automatic-failover by using Azure CLI or Azure portal. After you enable automatic failover, whenever there is a regional disaster, Cosmos DB will

automatically failover your account.

QUESTION 4

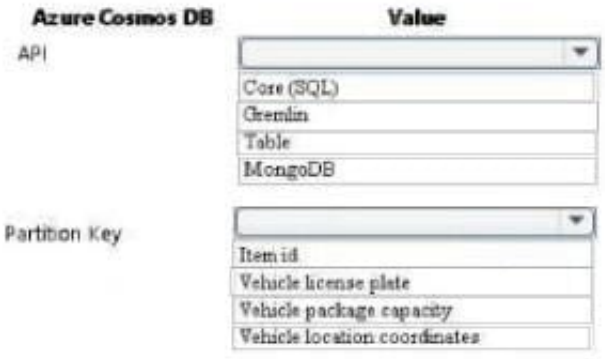
HOTSPOT

You need to implement the delivery service telemetry data.

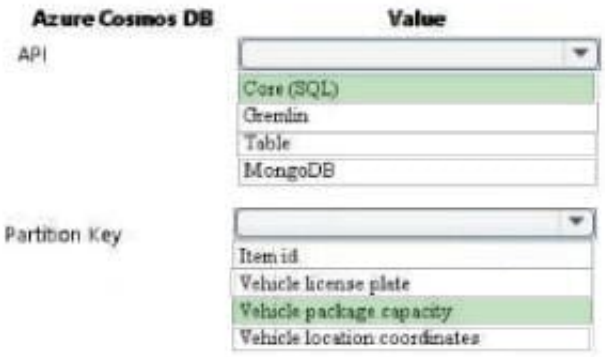
How should you configure the solution?

NOTE: Each correct selection is worth one point.

Hot Area:



Correct Answer:



QUESTION 5

Your company is developing an Azure API.

You need to implement authentication for the Azure API. You have the following requirements:

1.

All API calls must be secure.

2.

Callers to the API must not send credentials to the API. Which authentication mechanism should you use?

A. Basic

B. Anonymous

C. Managed identity

D. Client certificate

Correct Answer: C

Use the authentication-managed-identity policy to authenticate with a backend service using the managed identity of the API Management service. This policy essentially uses the managed identity to obtain an access token from Azure Active Directory for accessing the specified resource. After successfully obtaining the token, the policy will set the value of the token in the Authorization header using the Bearer scheme.

Reference: https://docs.microsoft.com/bs-cyrl-ba/azure/api-management/api-management-authentication-policies