EXAMTOPICS
- Expert Verified, Online, **Free**.

Custom View Settings

Question #32 *Topic 3*

DRAG DROP -

You are developing an Azure-hosted application that must use an on-premises hardware security module (HSM) key.

The key must be transferred to your existing Azure Key Vault by using the Bring Your Own Key (BYOK) process.

You need to securely transfer the key to Azure Key Vault.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:

Actions	Answer Area
Generate a key transfer blob file by using the HSM vendor-provided tool.	
Generate a Key Exchange Key (KEK).	
Create a custom policy definition in Azure Policy.	
Run the `az keyvault key import` command.	
Run the `az keyvault key restore` command.	
Retrieve the Key Exchange Key (KEK) public key.	

Correct Answer:

Actions	Answer Area
	Generate a Key Exchange Key (KEK).
	Retrieve the Key Exchange Key (KEK) public key.
Create a custom policy definition in Azure Policy.	Generate a key transfer blob file by using the HSM vendor-provided tool.
	Run the `az keyvault key import` command.
Run the `az keyvault key restore` command.	

To perform a key transfer, a user performs following steps:

☞ Generate KEK.

☞ Retrieve the public key of the KEK.

☞ Using HSM vendor provided BYOK tool - Import the KEK into the target HSM and exports the Target Key protected by the KEK.

☞ Import the protected Target Key to Azure Key Vault.

Step 1: Generate a Key Exchange Key (KEK).

Step 2: Retrieve the Key Exchange Key (KEK) public key.

Step 3: Generate a key transfer blob file by using the HSM vendor-provided tool.

Generate key transfer blob using HSM vendor provided BYOK tool

Step 4: Run the az keyvault key import command

Upload key transfer blob to import HSM-key.

Customer will transfer the Key Transfer Blob (".byok" file) to an online workstation and then run a az keyvault key import command to import this blob as a new

HSM-backed key into Key Vault.

To import an RSA key use this command:

az keyvault key import
Reference:
https://docs.microsoft.com/en-us/azure/key-vault/keys/byok-specification

Question #33 *Topic 3*

You develop and deploy an Azure Logic app that calls an Azure Function app. The Azure Function app includes an OpenAPI (Swagger) definition and uses an
Azure Blob storage account. All resources are secured by using Azure Active Directory (Azure AD).
The Azure Logic app must securely access the Azure Blob storage account. Azure AD resources must remain if the Azure Logic app is deleted.
You need to secure the Azure Logic app.
What should you do?

A. Create a user-assigned managed identity and assign role-based access controls.

B. Create an Azure AD custom role and assign the role to the Azure Blob storage account.

C. Create an Azure Key Vault and issue a client certificate.

D. Create a system-assigned managed identity and issue a client certificate.

E. Create an Azure AD custom role and assign role-based access controls.

Correct Answer: *A*
To give a managed identity access to an Azure resource, you need to add a role to the target resource for that identity.
Note: To easily authenticate access to other resources that are protected by Azure Active Directory (Azure AD) without having to sign in and provide credentials or secrets, your logic app can use a managed identity (formerly known as Managed Service Identity or MSI). Azure manages this identity for you and helps secure your credentials because you don't have to provide or rotate secrets.
If you set up your logic app to use the system-assigned identity or a manually created, user-assigned identity, the function in your logic app can also use that same identity for authentication.
Reference:
https://docs.microsoft.com/en-us/azure/logic-apps/create-managed-service-identity https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates-for-clients

HOTSPOT -

You are developing an application that uses a premium block blob storage account. You are optimizing costs by automating Azure Blob Storage access tiers.

You apply the following policy rules to the storage account. You must determine the implications of applying the rules to the data. (Line numbers are included for reference only.)

```
01 {
02   "rules":
03     {
04       "name": "agingDataRule",
05       "enabled": true,
06       "type": "Lifecycle",
07       "definition":
08         "filters": {
09           "blobTypes": [ "blockBlob" ],
10           "prefixMatch": [ "container1/salesorders", "container2/inventory" ]
11         },
12         "actions": {
13           "baseBlob": {
14             "tierToCool": { "daysAfterModificationGreaterThan": 60 },
15             "tierToArchive": { "daysAfterModificationGreaterThan": 120 }
16           }
17         }
18       }
19     },
20     {
21       "enabled": true,
22       "name": "lastAccessedDataRule",
23       "type": "Lifecycle",
24       "definition": {
25         "actions": {
26           "baseBlob": {
27             "enableAutoTierToHotFromCool": true,
28             "tierToCool": {
29               "daysAfterLastAccessTimeGreaterThan": 30
30             }
31           }
32         },
33         "filters": {
34           "blobTypes": [ "blockBlob" ]
35         }
36       }
37     },
38     {
39       "rules": [
40         {
41           "name": "expirationDataRule",
42           "enabled": true,
43           "type": "Lifecycle",
44           "definition": {
45             "filters": {
46               "blobTypes": [ "blockBlob" ]
47             },
48             "actions": {
49               "baseBlob": {
50                 "delete": { "daysAfterModificationGreaterThan": 730 }
51               }
52             }
53           }
54         }
55       ]
56     }
57   ]
58 }
```

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

	Yes	No
Block blobs prefixed with container1/salesorders or container2/inventory which have not been modified in over 60 days are moved to cool storage. Blobs that have not been modified in 120 days are moved to the archive tier.	○	○
Blobs are moved to cool storage if they have not been accessed for 30 days.	○	○
Blobs will automatically be tiered from cool back to hot if accessed again after being tiered to cool.	○	○
All block blobs older than 730 days will be deleted.	○	○

Correct Answer:

Answer Area

	Yes	No
Block blobs prefixed with container1/salesorders or container2/inventory which have not been modified in over 60 days are moved to cool storage. Blobs that have not been modified in 120 days are moved to the archive tier.	○	○
Blobs are moved to cool storage if they have not been accessed for 30 days.	○	○
Blobs will automatically be tiered from cool back to hot if accessed again after being tiered to cool.	○	○
All block blobs older than 730 days will be deleted.	○	○

Box 1: Yes -

```
"rules":
  {
    "name": "agingDataRule",
    "enabled": true,
    "type": "Lifecycle",
    "definition":
      "filters": {
        "blobTypes": [ "blockBlob" ],
        "prefixMatch": [ "container1/salesorders", "container2/inventory" ]
      },
      "actions": {
        "baseBlob": {
          "tierToCool": { "daysAfterModificationGreaterThan": 60 },
          "tierToArchive": { "daysAfterModificationGreaterThan": 120 }
```

Box 2: Yes -

```
"enabled": true,
"name": "lastAccessedDataRule",
"type": "Lifecycle",
"definition": {
  "actions": {
    "baseBlob": {
      "enableAutoTierToHotFromCool": true,
      "tierToCool": {
        "daysAfterLastAccessTimeGreaterThan": 30
      }
    }
  },
```

Box 3: Yes -

Box 4: Yes -

```
"rules": [
  {
    "name": "expirationDataRule",
    "enabled": true,
    "type": "Lifecycle",
    "definition": {
      "filters": {
        "blobTypes": [ "blockBlob" ]
      },
      "actions": {
        "baseBlob": {
          "delete": { "daysAfterModificationGreaterThan": 730 }
```

Question #35 *Topic 3*

You are developing a solution that will use a multi-partitioned Azure Cosmos DB database. You plan to use the latest Azure Cosmos DB SDK for development.

The solution must meet the following requirements:

☞ Send insert and update operations to an Azure Blob storage account.

☞ Process changes to all partitions immediately.

☞ Allow parallelization of change processing.

You need to process the Azure Cosmos DB operations.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Create an Azure App Service API and implement the change feed estimator of the SDK. Scale the API by using multiple Azure App Service instances.

B. Create a background job in an Azure Kubernetes Service and implement the change feed feature of the SDK.

C. Create an Azure Function to use a trigger for Azure Cosmos DB. Configure the trigger to connect to the container.

D. Create an Azure Function that uses a FeedIterator object that processes the change feed by using the pull model on the container. Use a FeedRange objext to parallelize the processing of the change feed across multiple functions.

Correct Answer: *C*

Azure Functions is the simplest option if you are just getting started using the change feed. Due to its simplicity, it is also the recommended option for most change feed use cases. When you create an Azure Functions trigger for Azure Cosmos DB, you select the container to connect, and the Azure Function gets triggered whenever there is a change in the container. Because Azure Functions uses the change feed processor behind the scenes, it automatically parallelizes change processing across your container's partitions.

Note: You can work with change feed using the following options:

☞ Using change feed with Azure Functions

☞ Using change feed with change feed processor

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/read-change-feed

HOTSPOT -
You have an Azure Web app that uses Cosmos DB as a data store. You create a CosmosDB container by running the following PowerShell script:

```
$resourceGroupName = "testResourceGroup"
$accountName = "testCosmosAccount"
$databaseName = "testDatabase"
$containerName = "testContainer"
$partitionKeyPath = "/EmployeeId"
$autoscaleMaxThroughput = 5000

New-AzCosmosDBSqlContainer -
-ResourceGroupName $resourceGroupName
-AccountName $accountName
-DatabaseName $databaseName
-Name $containerName
-PartitionKeyKind Hash
-PartitionKeyPath $partitionKeyPath
-AutoscaleMaxThroughput $autoscaleMaxThroughput
```

You create the following queries that target the container:
SELECT * FROM c WHERE c.EmployeeId > '12345'
SELECT * FROM c WHERE c.UserID = '12345'
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

	Yes	No
The minimum throughput for the container is 400 R/Us.	○	○
The first query statement is an in-partition query.	○	○
The second query statement is a cross-partition query.	○	○

Correct Answer:

Answer Area

	Yes	No
The minimum throughput for the container is 400 R/Us.	○	○
The first query statement is an in-partition query.	○	○
The second query statement is a cross-partition query.	○	○

Box 1: No -
You set the highest, or maximum RU/s Tmax you don't want the system to exceed. The system automatically scales the throughput T such that 0.1* Tmax <= T <=
Tmax.
In this example we have autoscaleMaxThroughput = 5000, so the minimum throughput for the container is 500 R/Us.

Box 2: No -
First query: SELECT * FROM c WHERE c.EmployeeId > '12345'
Here's a query that has a range filter on the partition key and won't be scoped to a single physical partition. In order to be an in-partition query, the query must have an equality filter that includes the partition key:
SELECT * FROM c WHERE c.DeviceId > 'XMS-0001'

Box 3: Yes -
Example of In-partition query:

Consider the below query with an equality filter on DeviceId. If we run this query on a container partitioned on DeviceId, this query will filter to a single physical partition.
SELECT * FROM c WHERE c.DeviceId = 'XMS-0001'
Reference:
https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-choose-offer https://docs.microsoft.com/en-us/azure/cosmos-db/how-to-query-container

Previous Questions

Next Questions