Microsoft.AZ-204-KR.v2026-03-20.q232

□□□□:	AZ-204-KR
□□□□:	Developing Solutions for Microsoft Azure (AZ-204 Korean Version)
□□□:	Microsoft
□□ □□ □□□:	232
□□:	v2026-03-20
# □□ □:	225
# □□ □□□:	2320
https://www.krdump.com/Microsoft.AZ-204-KR.v2026-03-20.q232.html	

NEW QUESTION: 1

Azure Container Apps□ □□□□ □□□□ □□□□□□ □□□□□□□ □□□□□□□ □□ □□ □□□□.

□ □□□□□□□ □□□□□□□ □ □□□ □□□ □□ Dapr□ □□□□□.

OrderService□ PaymentService□□ □ □□ □□□□ □□□□□.

OrderService□ PaymentService□□ ProcessPayment□□ □□□□ □□□□ □□□.

□□ □□□□ □□□□ □□□.

□□ □□□□□ □□□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□□ □□□□□.

□□: □□ □□□ 1□□□□.

Code to invoke the microservice method

```
using System;
using System.Net.Http;
using System.Text;
using Dapr.Client;
using System.Threading.Tasks;
namespace OrderService
{
    class Program
    {
        static async Task Main(string[] args)
        {
            using var httpClient = new HttpClient();
            var client = DaprClient.CreateInvokeHttpClient();
            using var client = new DaprBuilder().build();
        }
    }
}
```

sync<object, string>("paymentservice","ProcessPayment", new { amount = 100.0 });

Answer:

Explanation:

NEW QUESTION: 2

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

□□□□□ Windows □□□ □□ □□□ Azure App Service API □□ □□□□ □□□□□.

Testing □ Production□□□ □□ □□ □□□ □□□□□. □□□□ □□ □□□□ □□ □□□ □□□□□□.

□□ □□□ □□□□ □□ □□□□□ □□□□ □□□□ □□ □□□□ □□□□ □□□.

□□□: applicationInitialization □□ □□□ □□□□□ web.config □□□ □□□□□□□□. □ □□□□ □□□ □□□ □□ □□□ □□□ □□□□□.

□□□□ □□□ □□□□□?

A. □

B. □□□

Answer: (SHOW ANSWER)

Specify custom warm-up.

Some apps might require custom warm-up actions before the swap. The applicationInitialization configuration element in web.config lets you specify custom initialization actions. The swap operation waits for this custom warm-up to finish before swapping with the target slot. Here's a sample web.config fragment.
<system.webServer>
<applicationInitialization>
<add initializationPage="/" hostName="[app hostname]" />
<add initializationPage="/Home/About" hostName="[app hostname]" />
</applicationInitialization>
</system.webServer>
Reference:
https://docs.microsoft.com/en-us/azure/app-service/deploy-staging-slots#troubleshoot-swaps

NEW QUESTION: 3
□□ □□□□□ □□ □□ □□□□ □□□□ □□□□□□□ □□□□. Azure API Management □ □□□□ API□ □□□□□.
□□ □□ □□□ □□□□□ API □□□ □□□□ □□□.
* □□ □□ □□□□□ □□□□□.
* □□□□ □□ □□□□ □□□□□.
* □□□ □□□□ □□ □□□□□ □□□□□.
□□ □□□ □□□ □□□□ □□□? □□□□□ □□ □□□ □□□ □□□□□ □□□□□□ □. □ □□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □ □□□ □□ □□□ □□□ □□□□ □□□ □□□□□ □ □ □□□□. □□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:

Explanation:

NEW QUESTION: 4

account1□□□ NoSQL API □□□ □□ Azure Cosmos DB□ □□□□□. □□ □□□ □□□□ account1□ □□□□□.

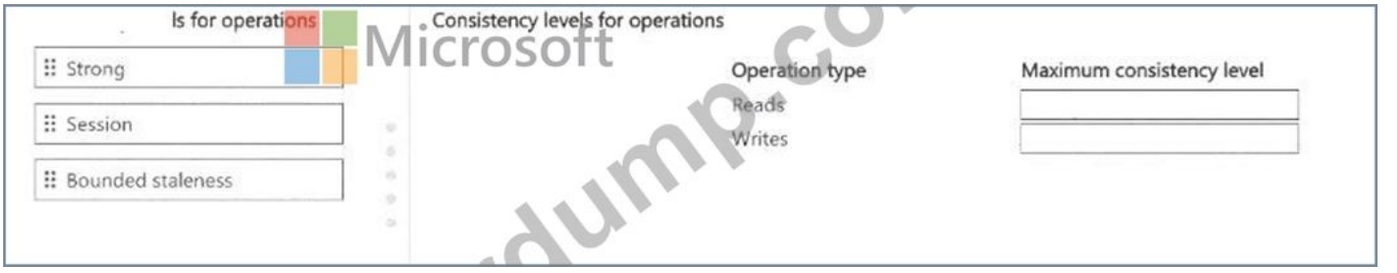
app1□□□ □□□□□□□ □□ □ □□ □□□ □□□□ □□ account1□ □□□□□ □□□□ □ □□□. app1□□ account1□□ □□□ □□ □□□ □□□□ □□□□ □□□.

Azure Cosmos DB SDK □□□□□□ □□□□ □□ □□□ □□□ □□□□□□ app1□ □□□ □□ □□□.

app1□ □□ □ □□ □□□ □□□ □□ □□□ □□□ □□□□ □□□.

□□ □□□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□ □□□ □□□ □□ □□□ □ □□□□. □ □□ □□□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□ □.

□□□□ □□□ □ □□□□ □□ □□□ □□□□□ □□□□□ □ □□ □□□□.



Answer:

Explanation:

NEW QUESTION: 5

account1□□□ NoSQL API □□□□ Azure Cosmos DB□ □□□□□. □ □□□□ db1□□□ □□□□□□□ □□, □ □□□□□□□□ container1□□□ □□□□□ □□□□. □□ □□□ □□□ □□□□ account!□ □□□□□.

□□□□ 1□ □□□□ Appl□□□ □□□□□□□ □□□□□ □□□. Appl□ □□ □□□□□ □□□ □□□ □□□□ □□□. Appl□ □□ □□□ □□□ □□□ □□□ □ □□□ □□□□ □ □□.

□□ □□ □□ □□□ □□□□□ □□□ □□□□ □□□.

□□ □□□ □□□□ □□□?

A. □□ □□

B. □□ □□

C. □□ □□

D. □□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 6

□□ □ □□□□ □□□□ □□□.

□□□□ □□□ □□□□ □□□?

Answer:

Explanation:

Answer Area

Azure	Configuration
Plan	Free Standard Premium Isolated
Service	App Service Web App App Service Static Web App Azure Function App Azure Blob Storage

NEW QUESTION: 7

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

□□□ □□□□□□□□ □□□□ □□□□ □ □□□□ □□□□ Azure □□□ □□□□□□□ □□□□ □□□□. □□□□ □□□□ □□□□ □□□□ □□ □ □□□□.

□□□ □□ □□ □□□ □□□□.

* □□□ □□□ 80□□□□□(GB)□□ □□ □ □□□□.
* □□□□ □□□□(FIFO) □□□ □□□□□.
* Azure □□□ □□□□□□.

□□□ □□□□ □□□□ □□□.

□□ □□: .Net API□ □□□□ □□□ □□□□□□□□ Azure Service Bus □□ □□□□ □□ □□□.
Azure Service Bus □□□ □□□□□ Azure Windows VM□ □□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Don't use a VM, instead create an Azure Function App that uses an Azure Service Bus Queue trigger.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function

NEW QUESTION: 8
□□ APIM(Azure API Management) □□□ API□ □□□□ □□□□.
□□□ □□□ □□□ □□□ □□□ □□□ □□□□ □ □□□□ □□ □□□ □□□□ API□ □ □□□ □□□.
□□ API□ □□□□ □□□.
□□□ □□ □□□?
A. □□□□ □□□ □□□ □ □□□ □□□ □□□□ □□□□□.
B. □□□□ □□□ □□□ □□□ □□□□ □□□ □□□□ □□□□□.
C. □□ □□□ □ □□□□□ □□□ □□□□ □□□□□.
D. □□□□ □□□ □□□ □ □□□□□ □□□ □□□□ □□□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 9
□□ □□ □ □□ □□ □□□□ □□□□ □□□ □□□□. □□□ □□□□ □□ □□□□ □□ □ □□□□□. Azure Search□ □□□□ □□ □□ □□□ □□□□ □□□□ □□□□ □□□.
* □□□□ □□, □□, □□ □ □□□ □□□□□ □□□ □ □□□ □□□.
* □□□□ □□, □□, □□ □ □□ □□□□ □□ □□ □□□ □□ □ □□□ □□□.
* □□□ □□ □□□ □□□ □□□□□ □□□.
□□□□ □□□□ □□□ □□□□ □□□.
□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
[SerializePropertyNameAsCamelCase]
 public class Restaurant
 {
  [Key, IsFilterable]
  public int RestaurantId { get; set; }
  [IsSearchable, IsFilterable, IsSortable]
  public string Name { get; set; }
```

- [IsSearchable.IsFilterable.IsSortable, IsFacetable]
- [IsFilterable.IsFacetable,Required]
- [IsSearchable]
- [IsSearchable, Required]

```
  public string location { get; set; }
  public string Phone { get; set; }
```

- [Required]
- [IsSearchable]
- [IsFilterable, IsFacetable, Required]
- [IsFilterable, IsFacetable, IsSortable]

```
  public string Description { get; set; }
```

- [IsFiltrable, IsSortable, IsSearchable]
- [IsFilterable, IsSortable, IsFacetable]
- [IsFiltrable, IsSortable, Key]
- [IsFilteranle, IsSortable, IsSearchable, Required]

```
  public double Rating { get; set; }
```

- [IsSearchable, IsFilterable, IsFacetable]
- [IsFilterbale, IsSortable, Key]
- [IsFilterable, IsSortable, IsSearchable]

[IsFilterable, IsSortable, Key, Required]

```
public List<string> Cuisines { get; set; }
```

- [IsFilterable, IsSortable, Key, Required]
- [IsSearchable, IsSortable, IsFacetable]
- [IsFilterable, IsSortable, Key, IsSearchable]
- [IsFilterable, IsFacetable]

```
public bool FamilyFriendly { get; set; }
```

Answer:

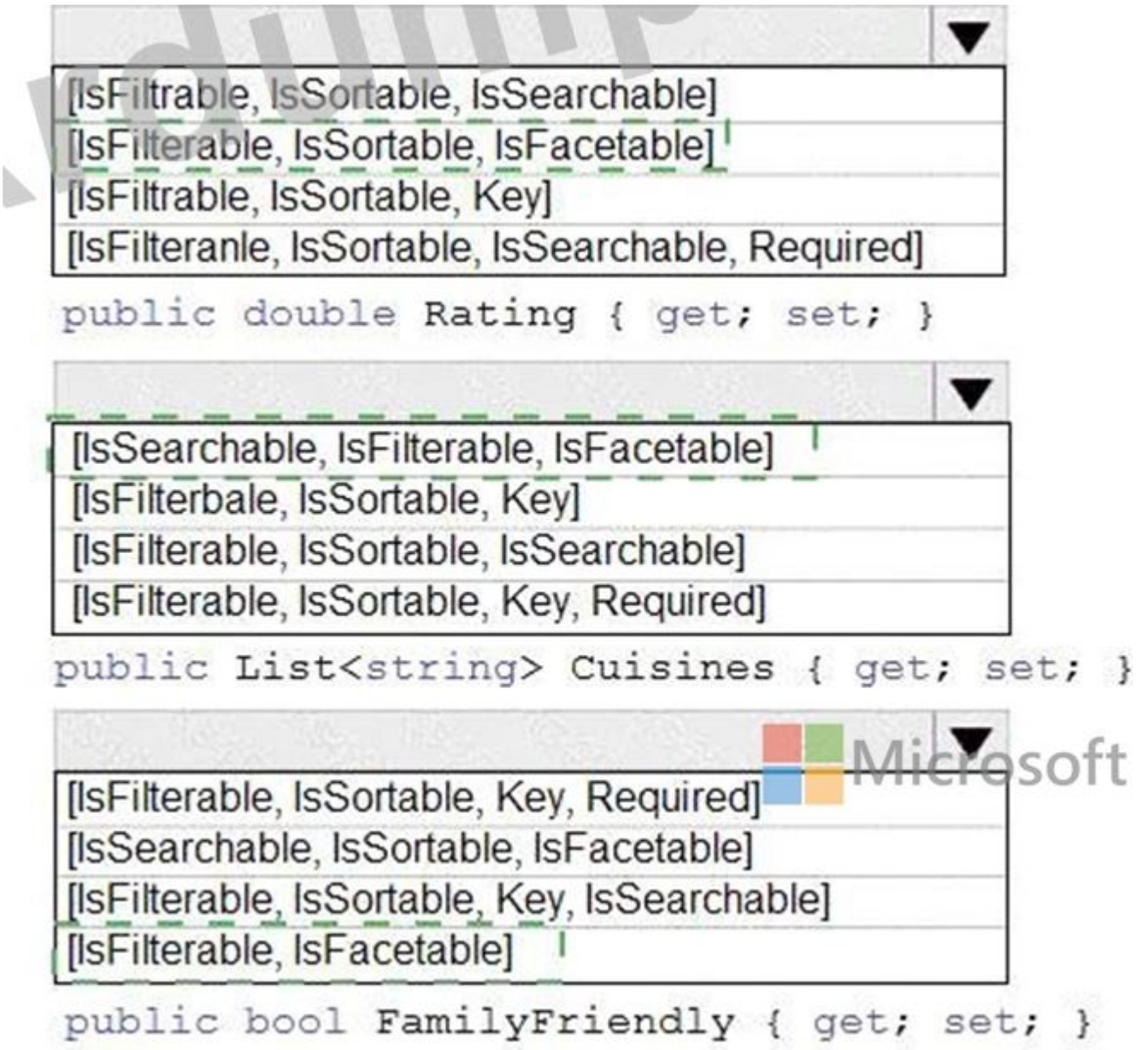
```
[SerializePropertyNameAsCamelCase]
public class Restaurant
{
 [Key, IsFilterable]
 public int RestaurantId { get; set; }
 [IsSearchable, IsFilterable, IsSortable]
 public string Name { get; set; }
```

- [IsSearchable,IsFilterable,IsSortable, IsFacetable]
- [IsFilterable,IsFacetable,Required]
- [IsSearchable]
- [IsSearchable, Required]

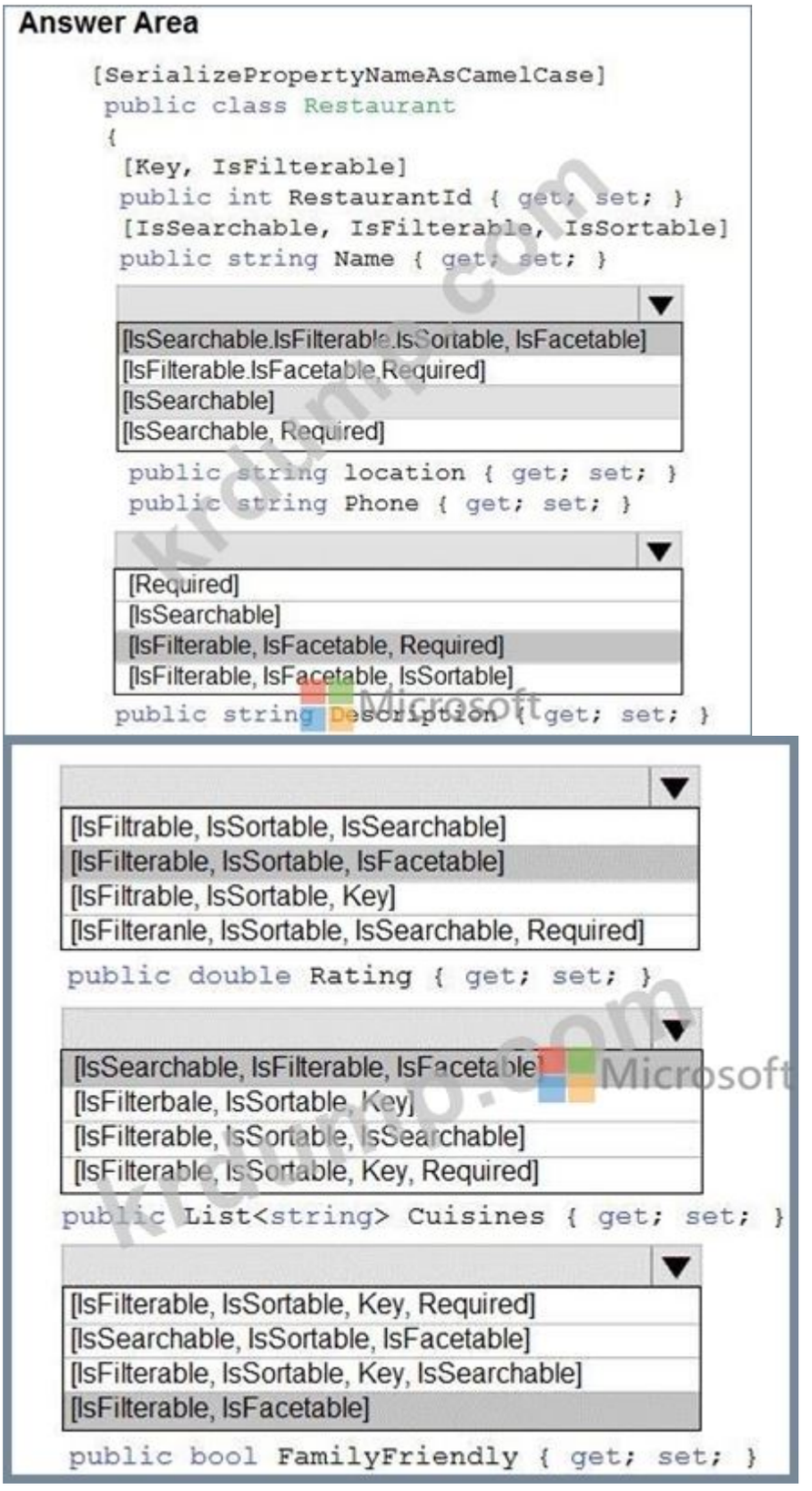
```
 public string location { get; set; }
 public string Phone { get; set; }
```

- [Required]
- [IsSearchable]
- [IsFilterable, IsFacetable, Required]
- [IsFilterable, IsFacetable, IsSortable]

```
public string Description { get; set; }
```



Explanation:



Box 1: [IsSearchable.IsFilterable.IsSortable,IsFacetable]

Location

Users must be able to search for restaurants by name, description, location, and cuisine.

Users must be able to narrow the results further by location, cuisine, rating, and family-friendliness.

Box 2: [IsSearchable.IsFilterable.IsSortable,Required]

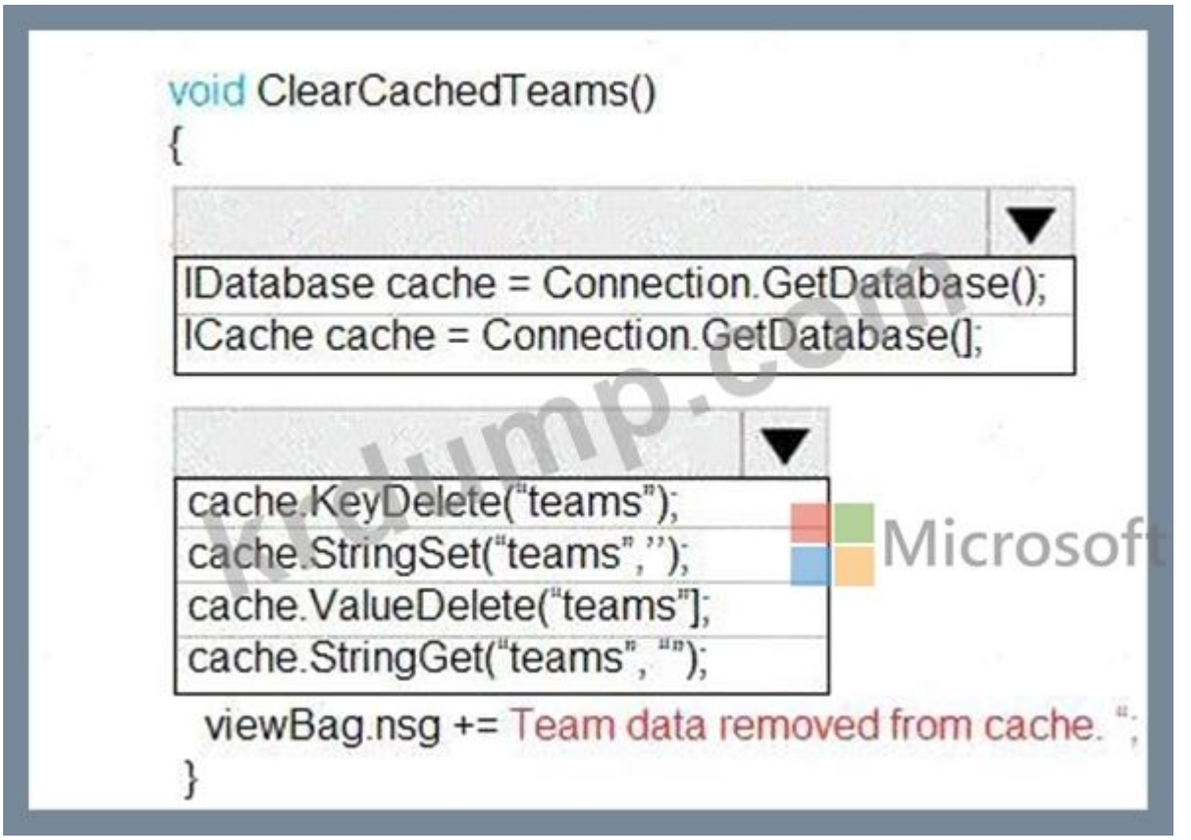
Description

Users must be able to search for restaurants by name, description, location, and cuisine.
All words in descriptions must be included in searches.
Box 3: [IsFilterable,IsSortable,IsFaceTable]
Rating
Users must be able to narrow the results further by location, cuisine, rating, and family-friendliness.
Box 4: [IsSearchable.IsFilterable,IsFacetable]
Cuisines
Users must be able to search for restaurants by name, description, location, and cuisine.
Users must be able to narrow the results further by location, cuisine, rating, and family-friendliness.
Box 5: [IsFilterable,IsFacetable]
FamilyFriendly
Users must be able to narrow the results further by location, cuisine, rating, and family-friendliness.
References:
https://www.henkboelman.com/azure-search-the-basics/

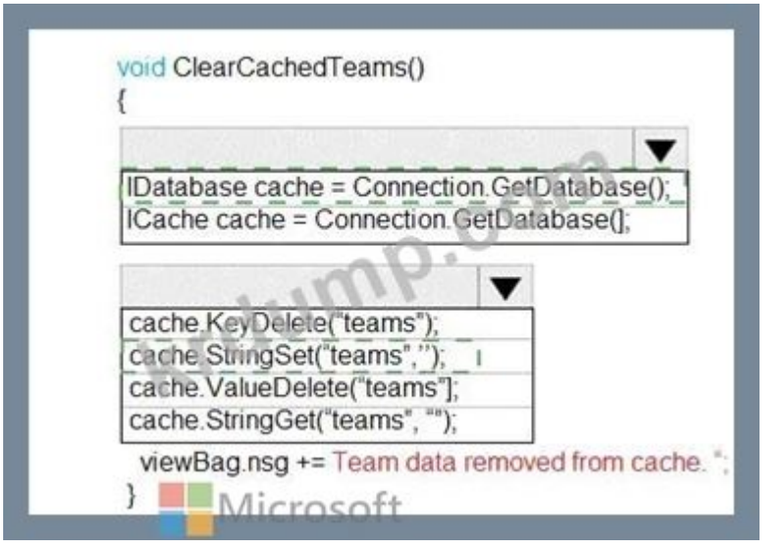
NEW QUESTION: 10
Azure SDK□ □□□□ □ □□□□□□□ □□□□ □□□□. □ □□□□□□□ □□□ ma □ □ □□ BlockBlobStorage □□□ □□□ □□□□□□. □□□□□□□ □□□□□ □□□□ □ □ □□ □□□□ □□□□□□ □□□□ □□□. □□□□ □□□ □□□□ □□□□ □ □ □□ □□□ □□ □□□ □□□□ □□□.
□□□□ □□□ □□□□ □□□.
□□ □□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□ □□□□□.
□□: □□□ □□ □□□ □□□ 1□□ □□□□□.
Answer:
See the Explanation below:
Explanation:
See the answer in below image.

NEW QUESTION: 11
□□□□ □□ □□□□ □□□□ □□□□. □□□□ □□ □□□□ □□□□□ □□□□□ □□ □□ □□□ □□□ □□□□ □ □ □□□□. □□□□□ Team□□□ □□□□ □□□□□. □□ □□□□ □□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□ Azure Redis Cache □ □□□□ □□□ □□□□□.

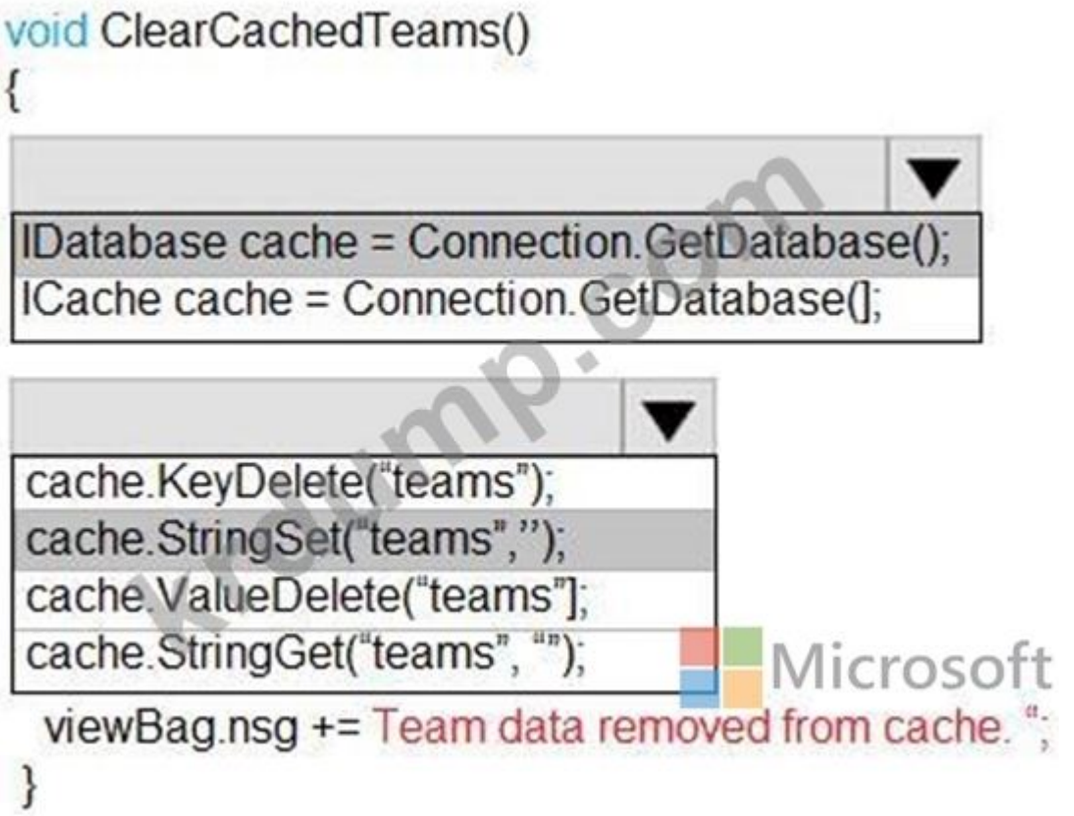
□ □□□□ □□□□ □□□ □□□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: IDatabase cache = connection.GetDatabase();

Connection refers to a previously configured ConnectionMultiplexer.

Box 2: cache.StringSet("teams","")

To specify the expiration of an item in the cache, use the TimeSpan parameter of StringSet.

cache.StringSet("key1", "value1", TimeSpan.FromMinutes(90));

References:

https://azure.microsoft.com/sv-se/blog/lap-around-azure-redis-cache-preview/

NEW QUESTION: 12

ContentUploadService □□□ □□□□ □□□.

□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

A. CS23 □□ □□ □□□□ □□□□□.

□□: □□□

B. CS24 □□ □□ □□□□ □□□□□.

os□□: Windows

C. CS24 □□ □□ □□□□ □□□□□.

os□□: □□□

D. □□ □□□□ CS23 □□ □□□□□.

□□: □□

Answer: A,C (LEAVE A REPLY)

Scenario: All Internal services must only be accessible from Internal Virtual Networks (VNets)

There are three Network Location types - Private, Public and Domain Reference:

https://devblogs.microsoft.com/powershell/setting-network-location-to-private/

NEW QUESTION: 13

Microsoft Entra □□□□ □□□□ □□□□□□□ □□□□ □□□□. □□□□ □□ □□□ □ □□ □□□□□. □□□ □□□ □□□ □□□□ □□□. □□ □□□ □□□□ □□□?

A. □□□ □□□ □□□ □□□ □□

B. □□□ □□□ □□□ □□□ □□

C. □□□ □□□ □□□ □ □□ □□□ □□

D. □□□ □□□ □□□ □ □□ □□□ □□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 14

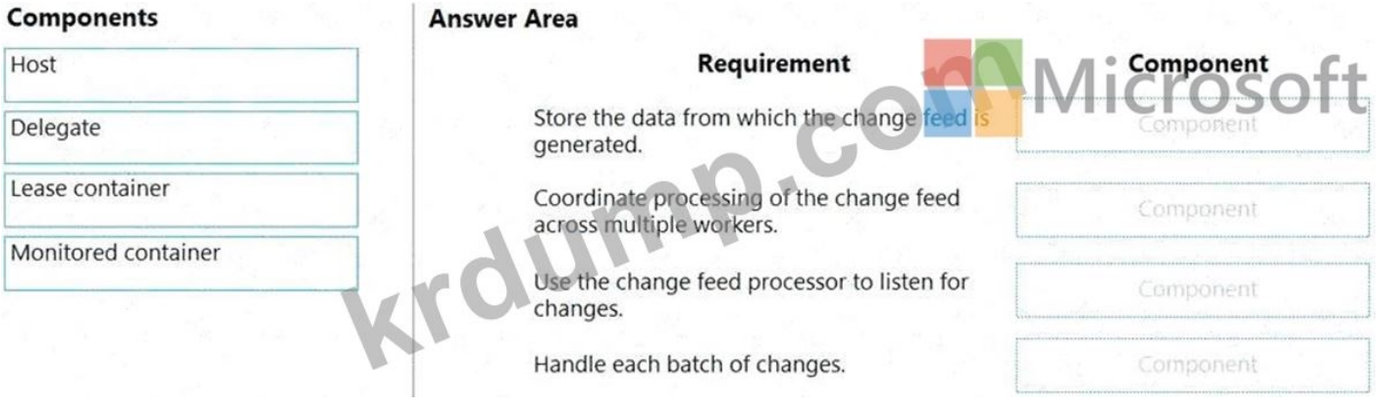
Cosmos DB□ □□□□ Azure □□□□ □□□□□.

□□ Cosmos DB □□□□□ □□□□□ □□ □□□ □□□□ □□□ □□ □□□□ □□□.

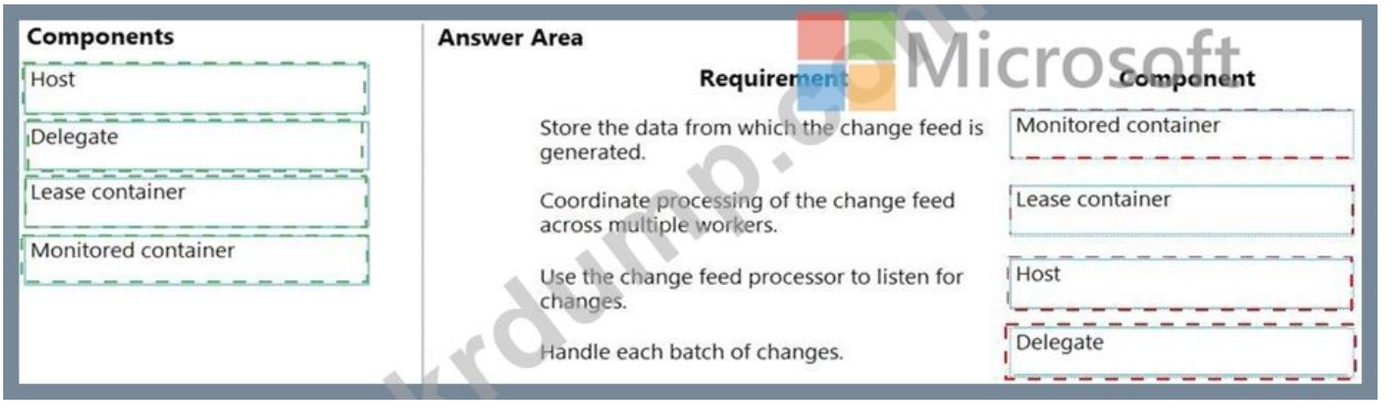
□□ □□ □□□□ □□□□ □□□□ □□□.

□□ □□ □□ □□□□ □□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□ □□□ □ □ □□□□ □□□□□□□. □ □□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

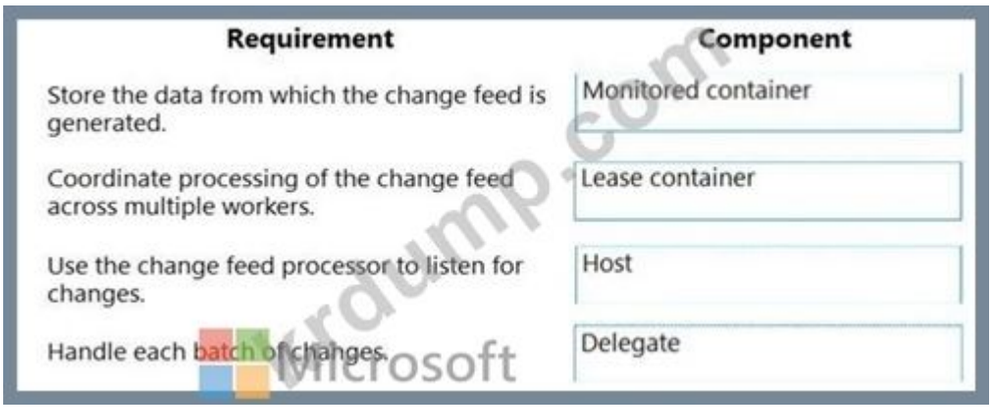
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: The monitored container

The monitored container has the data from which the change feed is generated. Any inserts and updates to the monitored container are reflected in the change feed of the container.

Box 2: The lease container

The lease container acts as a state storage and coordinates processing the change feed across multiple workers. The lease container can be stored in the same account as the monitored container or in a separate account.

Box 3: The host: A host is an application instance that uses the change feed processor to listen for changes.

Multiple instances with the same lease configuration can run in parallel, but each instance should have a different instance name.

Box 4: The delegate

The delegate is the code that defines what you, the developer, want to do with each batch of changes that the change feed processor reads.

Reference:

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

NEW QUESTION: 15

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□□ □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □ □ □□ □ □□□ □□ □□, □□ □□□□ □□□ □□ □□□ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□ □ □□□□. □□□ □□□ □□□ □□ □□□ □□□ □ □□□□.

Azure Event Grid□ □□□□ □□ □□□□ □□□ □□□ □□□□ □□□□ □□□□□□□ □ □□□ □□□□.

□□□ □□ □□ □□□ □□□□□.

* □□ □□□ □□□ □□□ □□□ □□□□ □□□□ □□ □□ □□□□ □□□ □□□.

* □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□□.

* □□ □ □□ □□□ Microsoft Entra ID□ □□□□ □□□□ □□□.

* □□□□ □□ □□□□□□ □□□□□ □□□.

Azure Event Grid□ □□□□ □□□.

□□□: □□□ □□□□ □□□□ □□□□□. □ □□□ □□ □□□ □□ □□□ □□□□.

□□□□ □□□ □□□□□?

A. □□□

B. □

Answer: A (LEAVE A REPLY)

NEW QUESTION: 16

ContosoApp□□□ ASP.NET Core □□□□□□□□ □□□□ Docker □□□□ □□ □□□□ □. setupScript.ps1□□□ □□ □□□□□ ContosoApp.dll□ □□□ □□□ □□□□□□ □□□ □□□□.

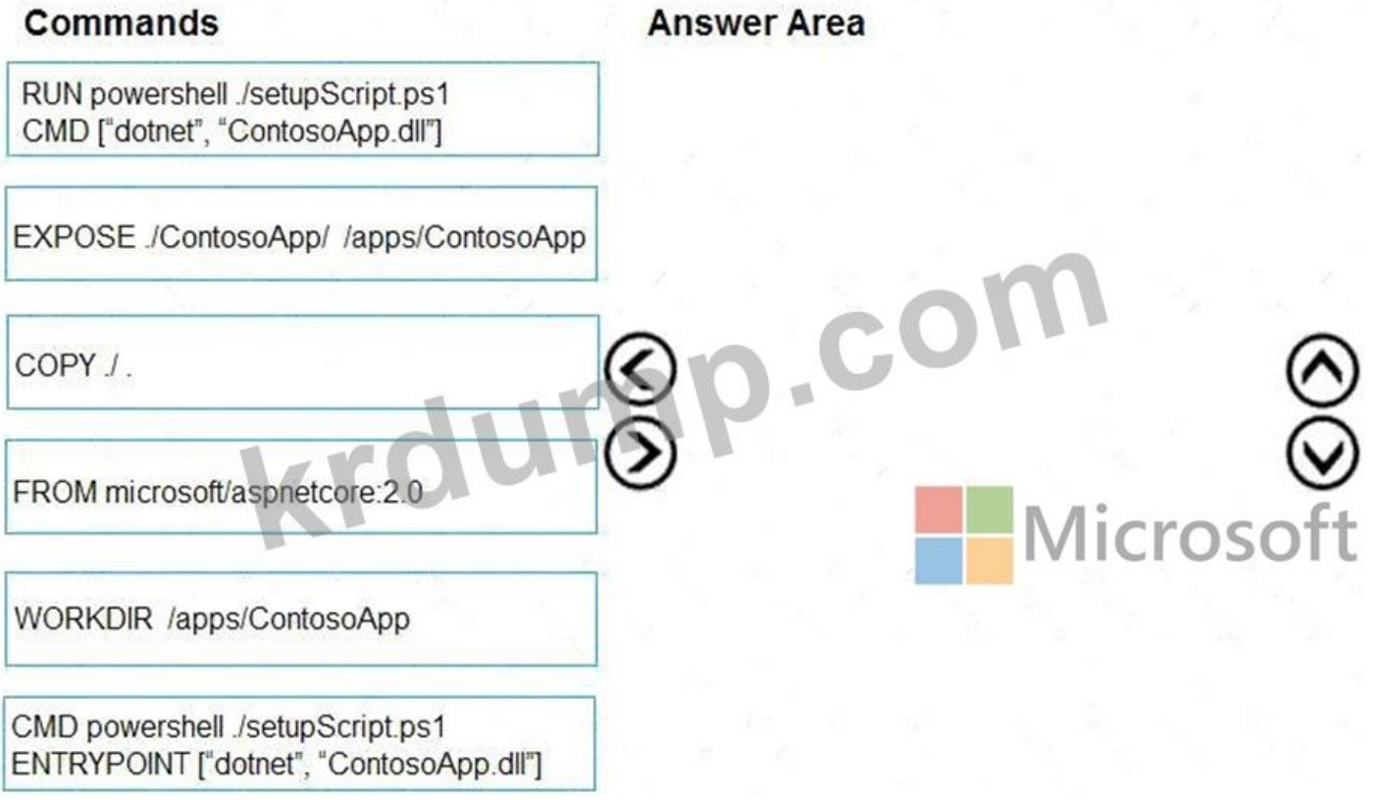
□□ □□ □□□ □□□□ Dockerfile □□□ □□□□ □□□.

*□□□□□ □□□□ setupScript.ps1□ □□□□□.

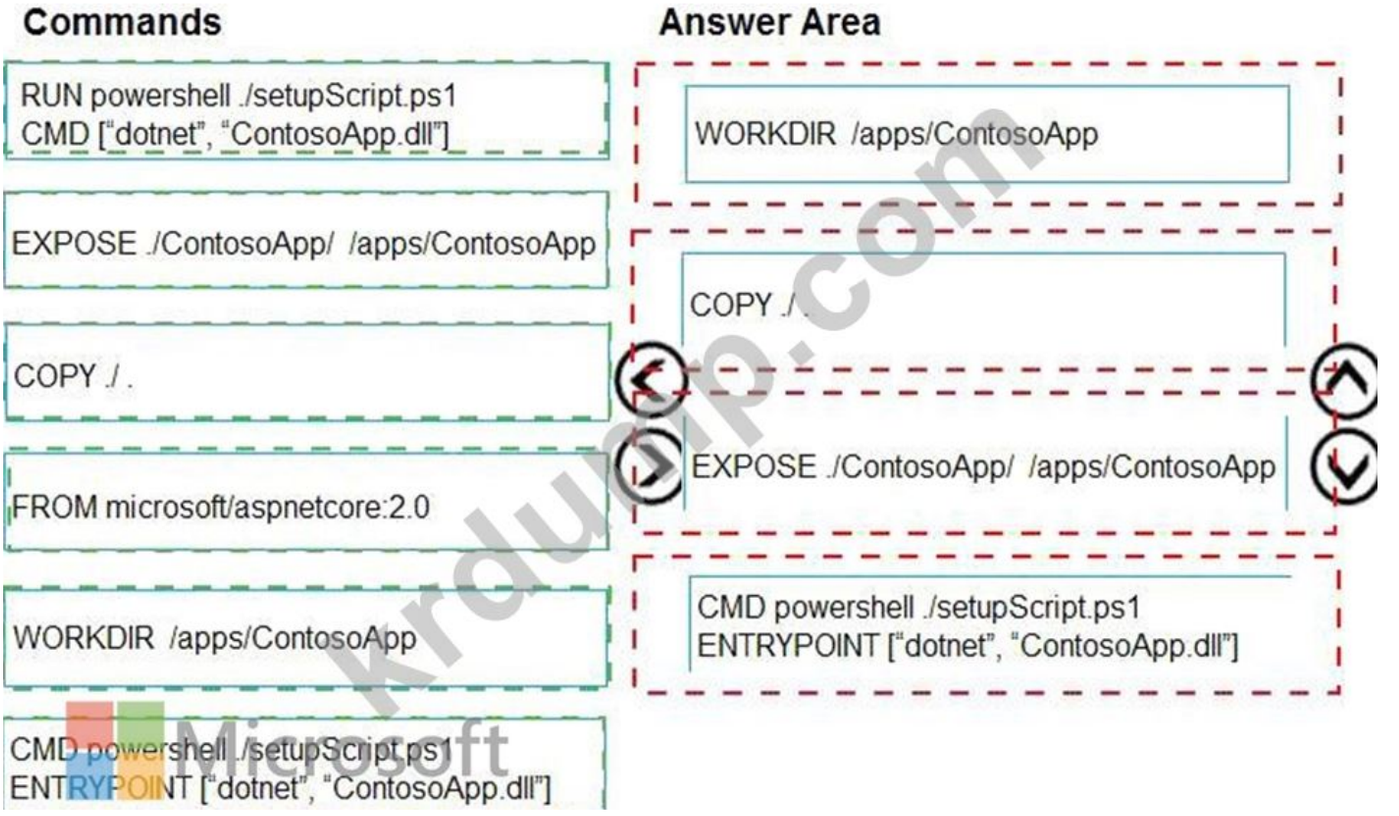
*□□□□□ □□□ □ ContosoApp.dll□ □□□□□.

Docker □□□ ContosoApp.dll □ setupScript.ps1□ □□□ □□□ □□□ □□□□□ □□□.

□□□□ □□□□ □ □□□□ □□ □ □□ □□□ □□□□□? □□□□□ □□ □□□□ □□ □□□ □□ □□□□ □□□□ □□□ □□□ □□□□□□.



Answer:



Explanation:

Step 1: WORKDIR /apps/ContosoApp

Step 2: COPY ./-

The Docker document must be created in the same folder where ContosoApp.dll and setupScript.ps1 are stored.

Step 3: EXPOSE ./ContosoApp/ /app/ContosoApp

Step 4: CMD powershell ./setupScript.ps1

ENTRYPOINT ["dotnet", "ContosoApp.dll"]

You need to create a Dockerfile document that meets the following requirements:

Call setupScript.ps1 when the container is built.

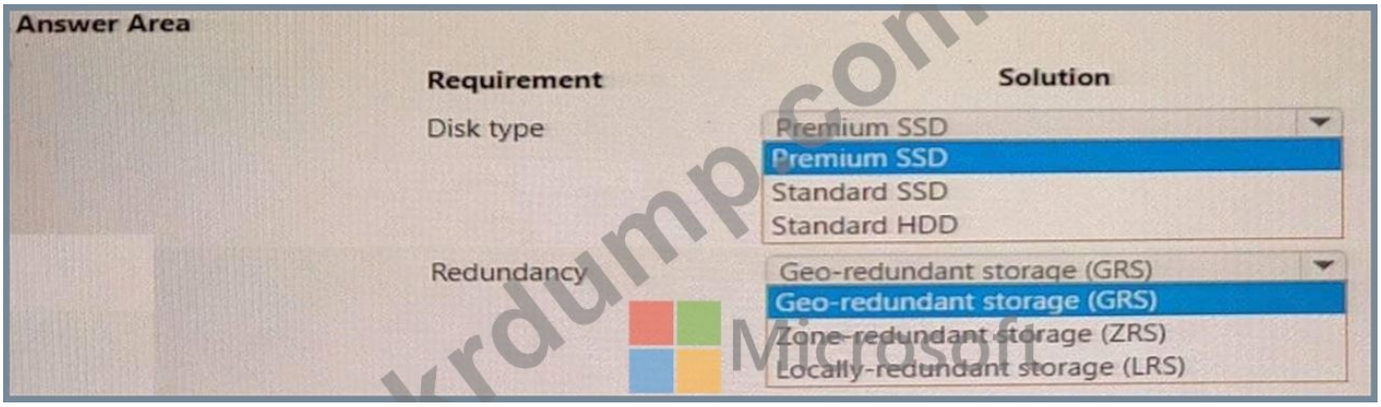
Run ContosoApp.dll when the container starts.

References:

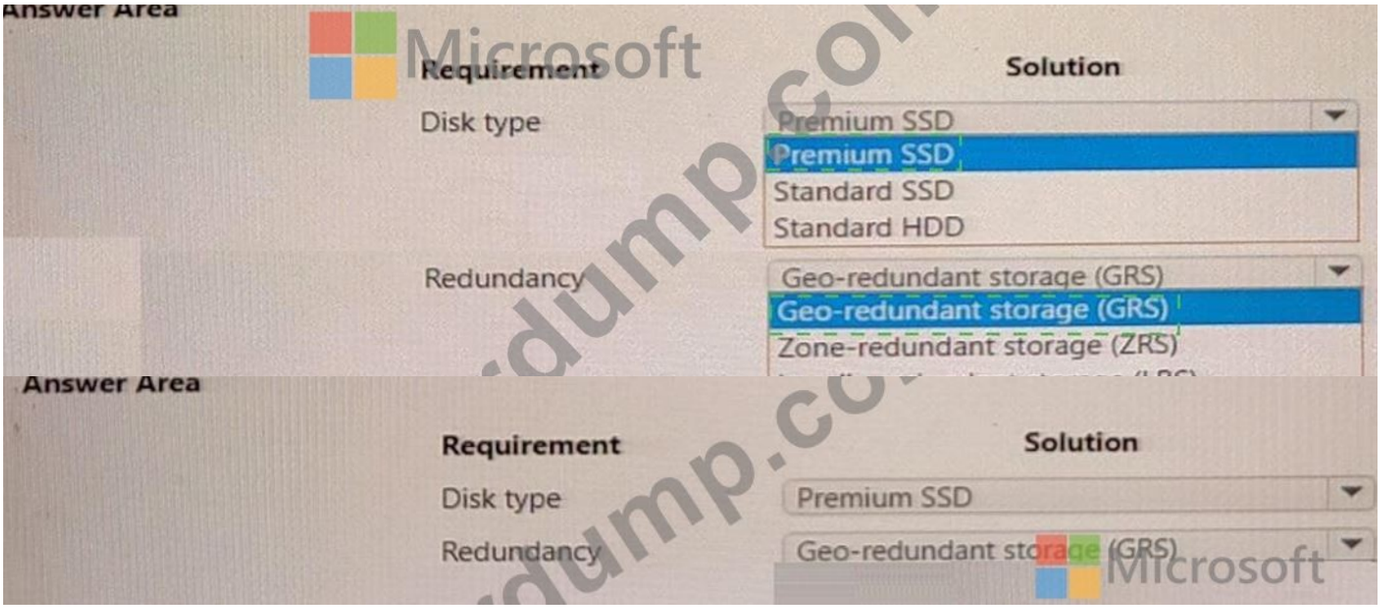
https://docs.microsoft.com/en-us/azure/app-service/containers/tutorial-custom-docker-image

NEW QUESTION: 17

□□□ □□□ Azure VM(□□ □□)□□ □□□ DNA □□□□ □□□□ SaaS(Software-as-a-Service) □□□□□□□ □□□□ □□□□. □□□□ VM□ □□□ □□ □□□□ □□□□□.
□□ □□□ VM□ □□□ □□□□ □□□ □□ □□□□□.
□□□ □□ □□ □□□ □□□□.
* □□ □□□□□ □□□ □□□ □□□□ □□□ □□□□ □□□ □□□ □ □□□ □□□.
* □□ □□□□□ Azure □□□ □□ □□ □ □□ □□ □□□ □□□□□ □□□.
□□ □□□ □□□□□ □□□ □□□□□□ VM□ □□ □□□□ □□□□□□□ □□□. □□ □□□ Azure Managed Disk□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



NEW QUESTION: 18

Azure Front Door Service□ □□□□ □□□□.

Brotli □□□ □□□□ □□□ □□□□ □□□ □□□□ □□□□. □□□□ XML □□□ □□□ □ □□□□ □□□□□□. □□ □□□ 9MB□□□.

□□□ □□ □□□ □□□□ □□□.

□□□□□ □□ □□□□ □□□ □□□ □□□□□□.

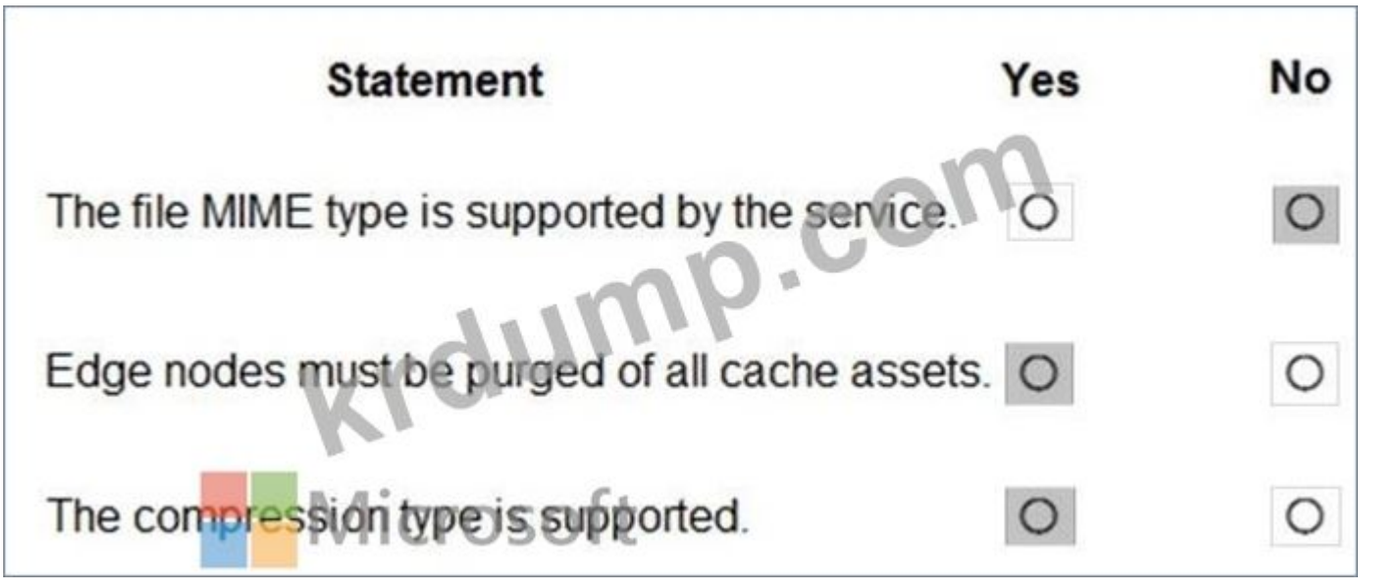
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Statement	Yes	No
The file MIME type is supported by the service.	○	○
Edge nodes must be purged of all cache assets.	○	○
The compression type is supported.	○	○

Answer:

Statement	Yes	No
The file MIME type is supported by the service.	○	○
Edge nodes must be purged of all cache assets.	○	○
The compression type is supported.	○	○

Explanation:



Box 1: No

Front Door can dynamically compress content on the edge, resulting in a smaller and faster response to your clients. All files are eligible for compression. However, a file must be of a MIME type that is eligible for compression list.

Box 2: No

Sometimes you may wish to purge cached content from all edge nodes and force them all to retrieve new updated assets. This might be due to updates to your web application, or to quickly update assets that contain incorrect information.

Box 3: Yes

These profiles support the following compression encodings: Gzip (GNU zip), Brotli Reference: https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching

NEW QUESTION: 19

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

Azure Web App□□ □□□ □ □□□□ □□□□ □□□□. □□□□ Azure AD(Azure Active Directory) □□ □□□ □□□□ □□□□□.

□ □□□□ □□ □□□, □□ □ □□ □□ □□ □ □□□ □□□□□ □□□ □□□□□. □□ □□□ □□□□□ □□□□ Azure AD □□ □□ □□□ □□□□ □□□. □□□ □□□□ □□ □.

□□□:

*□ Azure AD □□□□□□□ □□□□□□ □□□ groupMembershipClaims □□ □□ □□□ □□□□□.

*□ □□□□□ □□□□ □□□ □□□□ □□ JWI□ □□ □□□ □□ □□□□□.

□□□□ □□□ □□□□□?

A. □

B. □□□

Answer: (SHOW ANSWER)

To configure Manifest to include Group Claims in Auth Token

1. Go to Azure Active Directory to configure the Manifest. Click on Azure Active Directory, and go to App registrations to find your application:

2. Click on your application (or search for it if you have a lot of apps) and edit the Manifest by clicking on it.

3. Locate the "groupMembershipClaims" setting. Set its value to either "SecurityGroup" or "All". To help you decide which:

"SecurityGroup" - groups claim will contain the identifiers of all security groups of which the user is a member.

"All" - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member Now your application will include group claims in your manifest and you can use this fact in your code.

References:

https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

NEW QUESTION: 20

Azure App Configuration□□ □□□□ □□ □□□□ □□□□ ASP.NET Core □□ □□□□ □ □□□. □□□□□□ □□ □□□□ □□□□ AppFeatureFlagStore□□ Azure App Configuration □□□□ □□□□.

□□ □□ □□□ □□□□□ □□ □□□□□□ □□□.

* □□ □□ □□□ □□ □□ □□□ □□□□ □□□ □□□□□.
* □□□□ □□ □□□□ □□ □□□□ □□□□ □□ □□□□ □□□□□.
* □□□□ □□ □□□□ □□□□□□ □□□□□.

□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
if (env.IsDevelopment())
{
   app.UseDeveloperExceptionPage();
{
else
{
   app.UseExceptionHandler("/Error");
}

app. [          ] ();
app. [          ] ();
app. [          ] ();

app.UseEndpoint(endpoints =>
{
   endpoints.MapRazorPages();
});
}
```

- UseAuthentication
- UseStaticFiles
- UseSession
- UseCookiePolicy

- UseAuthorization
- UseHttpsRedirection
- UseSession
- UseCookiePolicy

- UseAzureAppConfiguration
- UseRequestLocalization
- UseCors
- UseStaticFiles

Answer:

Answer Area

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
if (env.IsDevelopment())
{
   app.UseDeveloperExceptionPage();
{
else
{
   app.UseExceptionHandler("/Error");
}

app.  [dropdown]  ();
```

- UseAuthentication
- UseStaticFiles
- UseSession
- UseCookiePolicy

```
app.  [dropdown]  ();
```

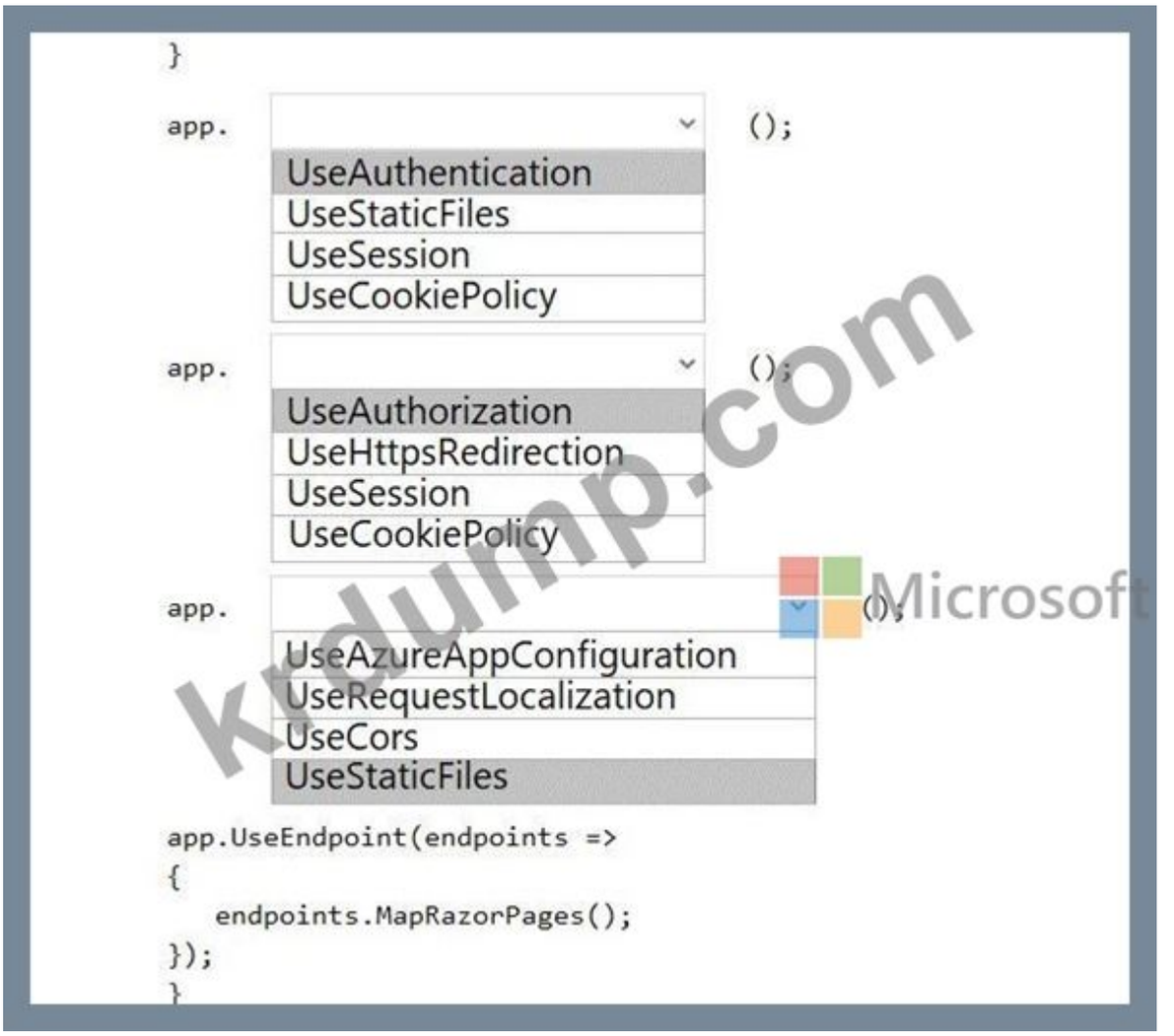
- UseAuthorization
- UseHttpsRedirection
- UseSession
- UseCookiePolicy

```
app.  [dropdown]  ();
```

- UseAzureAppConfiguration
- UseRequestLocalization
- UseCors
- UseStaticFiles

```
app.UseEndpoint(endpoints =>
{
   endpoints.MapRazorPages();
});
}
```


Explanation:



Box 1: UseAuthentication

Need to validate users before users are allowed access to secure resources.

UseAuthentication adds the AuthenticationMiddleware to the specified IApplicationBuilder, which enables authentication capabilities.

Box 2: UseAuthorization

Need to permit users to access secure resources.

UseAuthorization adds the AuthorizationMiddleware to the specified IApplicationBuilder, which enables authorization capabilities.

Box 3: UseStaticFiles

Need to use the Export feature in the app without requiring a restart of the app.

UseStaticFiles enables static file serving for the current request path Reference:

https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.iapplicationbuilder?view=aspnetcore-5.0

NEW QUESTION: 21

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□

□□□ □□□□ □□□□.
Azure Web App□□ □□□ □ □□□□ □□□□ □□□□. □□□□ Azure AD(Azure Active Directory) □□ □□□ □□□□ □□□□□.
□ □□□□ □□ □□□, □□ □ □□ □□ □□ □ □□□ □□□□□ □□□ □□□□□. □□ □□□ □□□□□ □□□□ Azure AD □□ □□ □□□ □□□□ □□□.
□□□ □□□□ □□□.
□□□:
* □ Azure AD □□□□□□□ □□□□. □□□□□□□ □□□□□□□ □□□□□□□ □□□ □□ □□□ □□□□ □□□□□□ □□□ □□□□□.
* □ □□□ □□□ Azure AD □□□ □□□□□. □ □□□□□ □□□□ □□□ □□□□ □□ JWT□ □□ □□□ □□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
To configure Manifest to include Group Claims in Auth Token
Go to Azure Active Directory to configure the Manifest. Click on Azure Active Directory, and go to App registrations to find your application:
Click on your application (or search for it if you have a lot of apps) and edit the Manifest by clicking on it.
Locate the "groupMembershipClaims" setting. Set its value to either "SecurityGroup" or "All". To help you decide which:
"SecurityGroup" - groups claim will contain the identifiers of all security groups of which the user is a member.
"All" - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member Now your application will include group claims in your manifest and you can use this fact in your code.
Reference:
https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

NEW QUESTION: 22
□□□ □□□ □□□□ □□□□ □□ □□□□ □□□□. □□□□□ □□, □□□ □□ □ □□ □□□ □□□ □□□□□. □□ □□□□□ □□□□□ □□ □ □□□□. □□□□□ □□□ □ □□□ □□ □□□□□ □ □□□□□.
□□ □□□□ Azure Table Storage□ □□□□ □□□.
□□ □□□□□ □□ □□□□ □□□□□ □□□ □□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

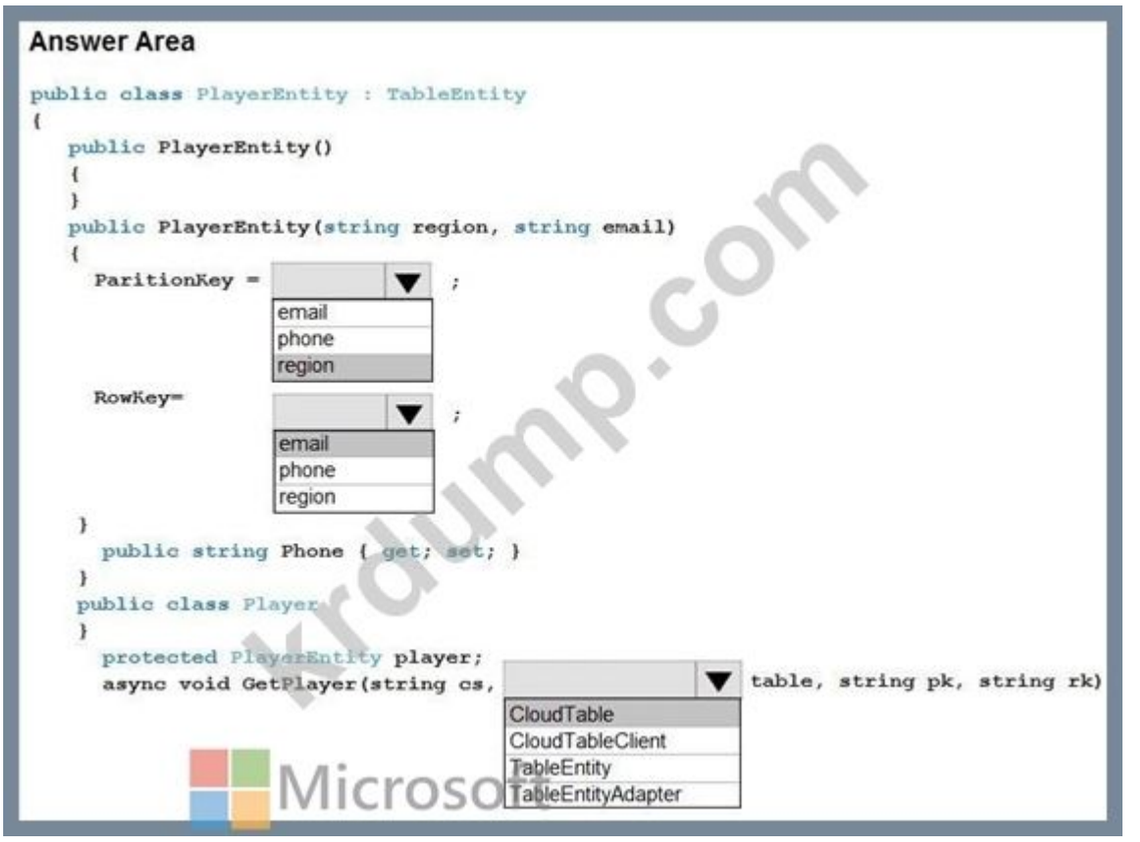
```
public class PlayerEntity : TableEntity
{
   public PlayerEntity()
   {
   }
   public PlayerEntity(string region, string email)
   {
     ParitionKey = [ ▼ ] ;
                   email
                   phone
                   region
     RowKey=       [ ▼ ] ;
                   email
                   phone
                   region
   }
     public string Phone { get; set; }
   }
   public class Player
   }

protected PlayerEntity player;
async void GetPlayer(string cs, [ ▼ ] table, string pk, string rk)
                                 CloudTable
                                 CloudTableClient
                                 TableEntity
                                 TableEntityAdapter
{
   [ ▼ ]
   TableEntity query =TableEntity.Retrieve<PlayerEntity>(pk, rk);
   TableOperation query =TableOperation.Retrieve<PlayerEntity>(pk,rk);
   TableResult query =TableQuery.Retrieve<PlayerEntity>(pk,rk);
   TableResultSegment query =TableResult.Retrieve<PlayerEntity>(pk, rk);

   [ ▼ ]
   TableEntity data =await table.ExecuteAsync(query);
   TableOperation data =await.table.ExeucteAsync(query);
   TableQuery data =await table.ExecuteAsync(query);
   TableResult data =await table.ExecuteAsync(query);
player=data.Result as PlayerEntity;
 }
}
```

Answer:

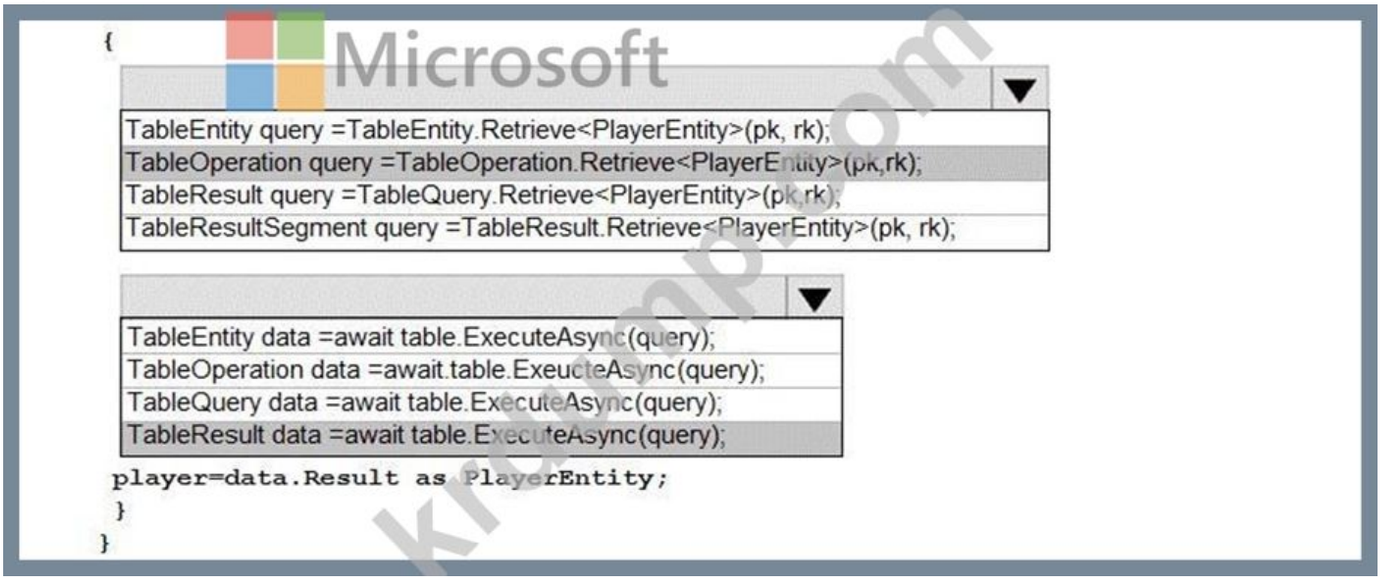
```
public class PlayerEntity : TableEntity
{
   public PlayerEntity()
   {
   }
   public PlayerEntity(string region, string email)
   {
     ParitionKey = [dropdown] ;
         email
         phone
         region

     RowKey= [dropdown] ;
         email
         phone
         region
   }
     public string Phone { get; set; }
   }
   public class Player
   }

protected PlayerEntity player;
async void GetPlayer(string cs, [dropdown] table, string pk, string rk)
         CloudTable
         CloudTableClient
         TableEntity
         TableEntityAdapter
{
  [dropdown]
     TableEntity query =TableEntity.Retrieve<PlayerEntity>(pk, rk);
     TableOperation query =TableOperation.Retrieve<PlayerEntity>(pk,rk);
     TableResult query =TableQuery.Retrieve<PlayerEntity>(pk,rk);
     TableResultSegment query =TableResult.Retrieve<PlayerEntity>(pk, rk);

  [dropdown]
     TableEntity data =await table.ExecuteAsync(query);
     TableOperation data =await.table.ExeucteAsync(query);
     TableQuery data =await table.ExecuteAsync(query);
     TableResult data =await table.ExecuteAsync(query);
 player=data.Result as PlayerEntity;
 }
}
```


Microsoft

Explanation:



Explanation:



Box 1: region

The player's region will be used to load-balance data.

Choosing the PartitionKey.

The core of any table's design is based on its scalability, the queries used to access it, and storage operation requirements. The PartitionKey values you choose will dictate how a table will be partitioned and the type of queries that can be used. Storage operations, in particular inserts, can also affect your choice of PartitionKey values.

Box 2: email

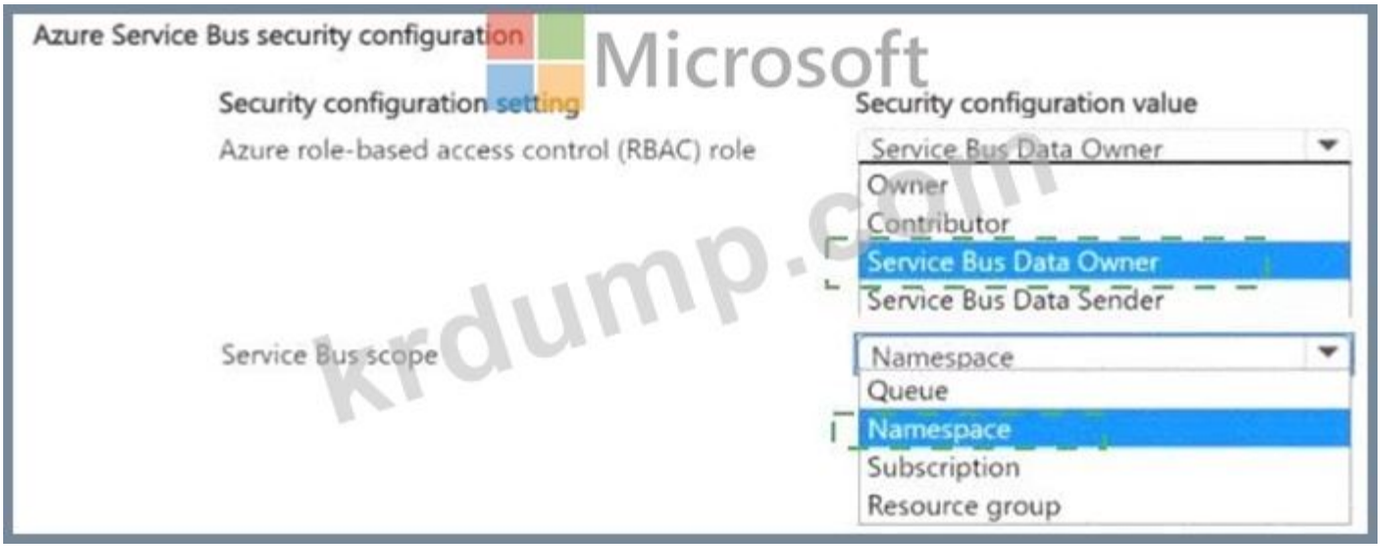
Not phone number some players may not have a phone number.

Box 3: CloudTable

Box 4 : TableOperation query =..
Box 5: TableResult
References:
https://docs.microsoft.com/en-us/rest/api/storageservices/designing-a-scalable-partitioning-strategy-for-azure- table-storage

NEW QUESTION: 23
□□□□ □□ □□□ □□□□ □□□.
□□ Service Bus □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□ □□.
□□□□: □□ □□□ 1□□□□.

Answer:



Explanation:

NEW QUESTION: 24
ASP.NET Core Web API □ □□□□ □□□□ □□□□. □ □□□□ □□ □□ □□ □ □□□

□□□ Azure Application Insights□ □□□□□. □ □□□□ Microsoft SQL Server □□□ □□ □□□□□□ □□□□ □□ □□□.
□□ □□□□□□□ □□ □□□ □□ □□□ □□□ □□□□□ □□□□ □□□.
□□□□□□□ □□□□ □□ □ □□ □□□ □□ □□ □□□ □□□□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. Telemetry.Context.Operation.Id
B. Tetemetry.Context.Cloud.RoleInstance
C. □□ □□.Id
D. Telemetry.ContextSession.Id
E. □□ □□.□□
Answer: A,C (LEAVE A REPLY)
References:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking
Explanation:
Example:
public async Task Enqueue(string payload)
{
// StartOperation is a helper method that initializes the telemetry item
// and allows correlation of this operation with its parent and children.
var operation = telemetryClient.StartOperation<DependencyTelemetry>("enqueue " +
queueName); operation.Telemetry.Type = "Azure Service Bus"; operation.Telemetry.Data =
"Enqueue " + queueName; var message = new BrokeredMessage(payload);
// Service Bus queue allows the property bag to pass along with the message.
// We will use them to pass our correlation identifiers (and other context)
// to the consumer.
message.Properties.Add("ParentId", operation.Telemetry.Id);
message.Properties.Add("RootId", operation.Telemetry.Context.Operation.Id); Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking

NEW QUESTION: 25
Windows □□□ □□ □ □□□ □□□ □□ □□□□□.
□□□□ □ □ □□ □ □□□ □□□ □□□□ □□□ □□□ □□□□ □□□.
□□ □□□ □□□□ □□□.
□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
                                   hub =
```

NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

```
                                       .
```

NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

GetInstallation
CreateClientFromConnectionString
CreateOrUpdateInstallation
PatchInstallation

```
 (notificationHubConnection, notificationHubName);
string windowsToastPayload =
 @"<toast><visual><binding template=""ToastText01""><text id=""1"">" +
 @"New item to view" + @"</text></binding></visual></toast>";
try
{
var result =
  await hub.                                   (windowsToastPayload);
  ...
}
catch (System.Exception ex)
{
 ...
}
...
```

SendWindowsNativeNotificationAsync
SubmitNotificationHubJobAsync
ScheduleNotificationAsync
SendAppleNativeNotificationAsync

Answer:

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
```

[▾] hub =

- NotificationHubClient
- NotificationHubClientSettings
- NotificationHubJob
- NotificationDetails

[▾] . [▾]

- NotificationHubClient
- NotificationHubClientSettings
- NotificationHubJob
- NotificationDetails

- GetInstallation
- CreateClientFromConnectionString
- CreateOrUpdateInstallation
- PatchInstallation

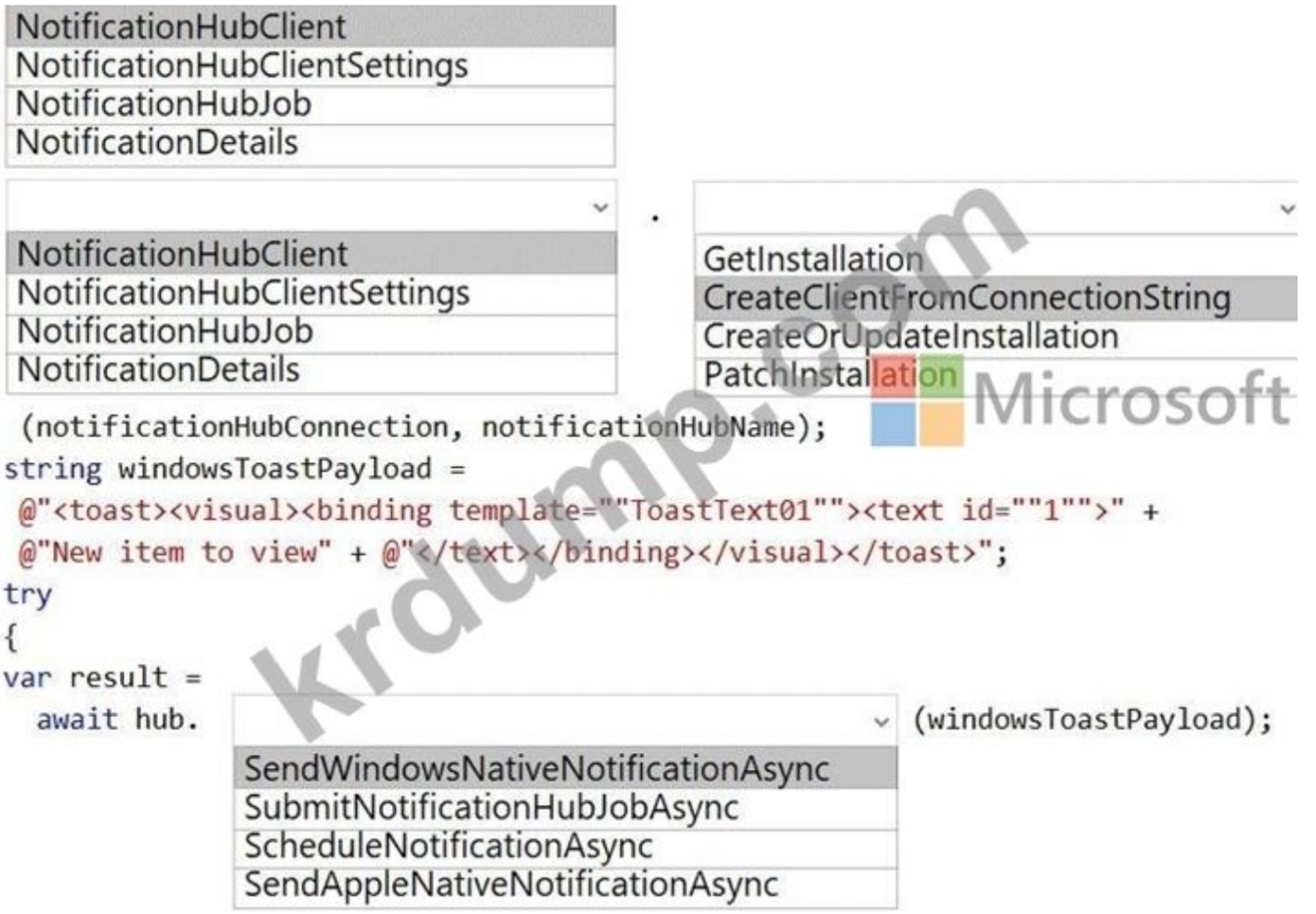
```
 (notificationHubConnection, notificationHubName);
string windowsToastPayload =
 @"<toast><visual><binding template=""ToastText01""><text id=""1"">" +
 @"New item to view" + @"</text></binding></visual></toast>";
try
{
var result =
  await hub.
```

[▾] (windowsToastPayload);

- SendWindowsNativeNotificationAsync
- SubmitNotificationHubJobAsync
- ScheduleNotificationAsync
- SendAppleNativeNotificationAsync

```
  ...
}
catch (System.Exception ex)
{
 ...
}
```

Explanation:



Box 1: NotificationHubClient

Box 2: NotificationHubClient

Box 3: CreateClientFromConnectionString

// Initialize the Notification Hub

NotificationHubClient hub =

NotificationHubClient.CreateClientFromConnectionString(listenConnString, hubName); Box 4: SendWindowsNativeNotificationAsync Send the push notification.

var result = await hub.SendWindowsNativeNotificationAsync(windowsToastPayload); Reference: https://docs.microsoft.com/en-us/azure/notification-hubs/notification-hubs-push-notification-registration- management

https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/app-service-mobile/app-service-mobile- windows-store-dotnet-get-started-push.md

NEW QUESTION: 26

API□ □□ □□□ □□□ □□□□ □□ API□ □□□□ Azure □□□ □□ □□□□. □□□ □ □□ mykeyvault□□ Azure Key Vault□ □□ token□□□ □□□ □□□□□.

Azure □□□ □□□ □□□□ □ □□□ □□□□ □□□. Azure □□ □ □□□ □□ □□ □□ □□ □□□?

A. App:Settings:Secret:mykeyvault:token

B.

```
@Microsoft.KeyVault(SecretUri=https://mykeyvault.vault.azure.net/secrets/token/)
```

C.
```
KeyVault:mykeyvault;Secret:token
```

D.
```
AZUREKVCONNSTR_ https://mykeyvault.vault.azure.net/secrets/token/
```

Answer: B (LEAVE A REPLY)

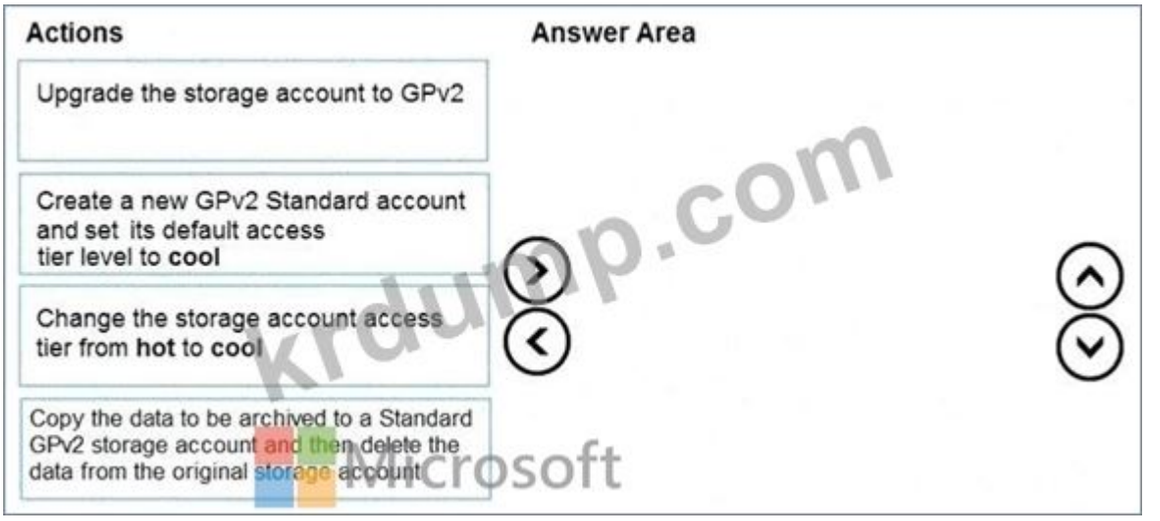
NEW QUESTION: 27

Azure Blob GPv1 Premium □□□□ □□□ □□□□ □□ □□□□□□□ □□□□ □□□□. 3 □□ □□ □ □□□□ □□ □□□□ □□□□.

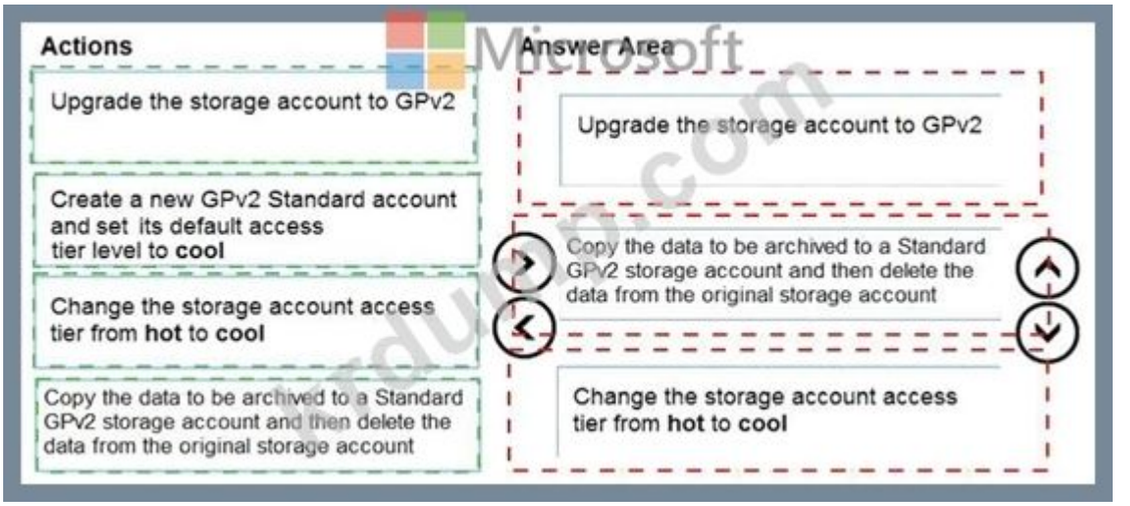
3□□□ □□ □□□□ □□ □□□ □ □□□ □□□. 1□ □□ □ □□□□ □□□□ □□□ □ □ □□□ □□□ □□□□.

□□□ □□□□ □□ □□□□ □□ Blob □□□□ □□ □□□□□ □□□□ □□ □□ □□ □ □□ □□□□□ □□□ □□□□ □□□.

□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□ □□□□□□.



Answer:



Explanation:

Step 1: Upgrade the storage account to GPv2
Object storage data tiering between hot, cool, and archive is supported in Blob Storage and General Purpose v2 (GPv2) accounts. General Purpose v1 (GPv1) accounts don't support tiering. You can easily convert your existing GPv1 or Blob Storage accounts to GPv2 accounts through the Azure portal.
Step 2: Copy the data to be archived to a Standard GPv2 storage account and then delete the data from the original storage account Step 3: Change the storage account access tier from hot to cool Note: Hot - Optimized for storing data that is accessed frequently.
Cool - Optimized for storing data that is infrequently accessed and stored for at least 30 days.
Archive - Optimized for storing data that is rarely accessed and stored for at least 180 days with flexible latency requirements, on the order of hours.
Only the hot and cool access tiers can be set at the account level. The archive access tier can only be set at the blob level.
Reference:
https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers

NEW QUESTION: 28
□□ staticwebapp.config.json □□□ □□□□ Azure Static Web App□ □□□□ □□□ □□□ app_location □□ □□□□□.

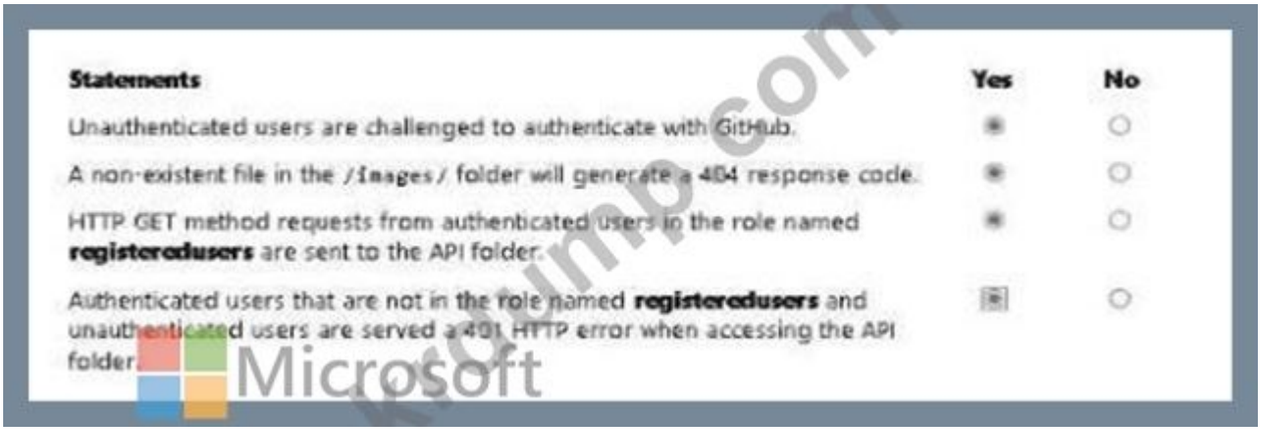
```
{
  "routes": [
    {
      "route": "/api/*",
      "methods": ["GET"],
      "allowedRoles": ["registeredusers"]
    },
    {
      "route": "/api/*",
      "methods": ["PUT", "POST", "PATCH", "DELETE"]
```

Statements	Yes	No
Unauthenticated users are challenged to authenticate with GitHub.	○	○
A non-existent file in the /images/ folder will generate a 404 response code.	○	○
HTTP GET method requests from authenticated users in the role named **registeredusers** are sent to the API folder.	○	○
Authenticated users that are not in the role named **registeredusers** and unauthenticated users are served a 401 HTTP error when accessing the API folder.	○	○

Answer:

Statements	Yes	No
Unauthenticated users are challenged to authenticate with GitHub.	○	○
A non-existent file in the /images/ folder will generate a 404 response code.	○	○
HTTP GET method requests from authenticated users in the role named **registeredusers** are sent to the API folder.	○	○
Authenticated users that are not in the role named **registeredusers** and unauthenticated users are served a 401 HTTP error when accessing the API folder.	○	○

Explanation:



Statements	Yes	No
Unauthenticated users are challenged to authenticate with GitHub.	●	○
A non-existent file in the /images/ folder will generate a 404 response code.	●	○
HTTP GET method requests from authenticated users in the role named **registeredusers** are sent to the API folder.	●	○
Authenticated users that are not in the role named **registeredusers** and unauthenticated users are served a 401 HTTP error when accessing the API folder.	●	○

NEW QUESTION: 29

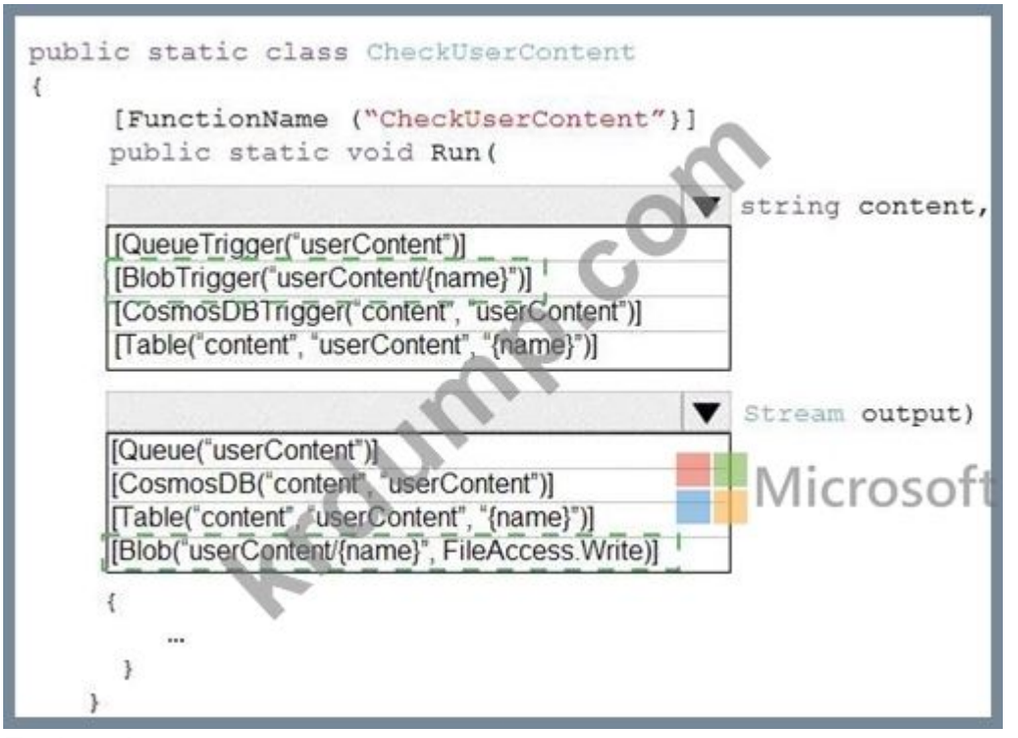
CheckUserContent □□□ □□ □□□□ □□□□ □□□.

□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

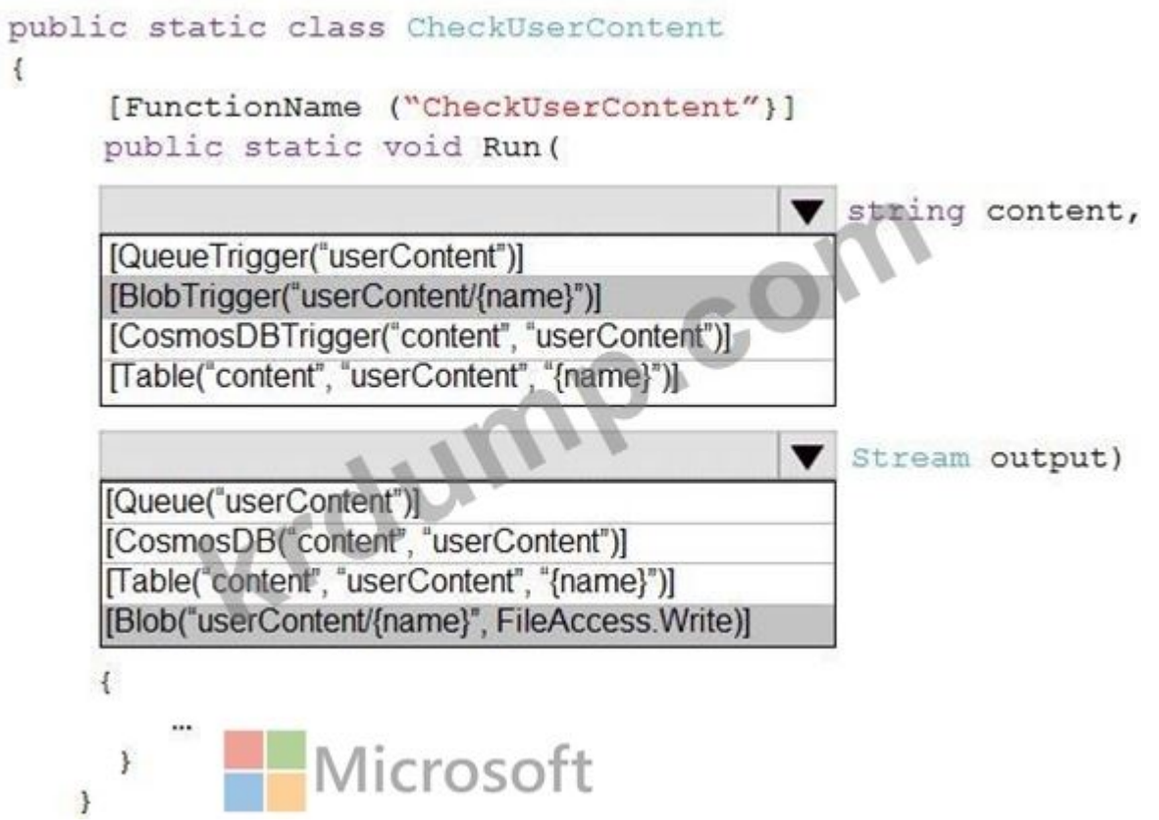
□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
public static class CheckUserContent
{
      [FunctionName ("CheckUserContent")]
      public static void Run(
            [QueueTrigger("userContent")]
            [BlobTrigger("userContent/{name}")]
            [CosmosDBTrigger("content", "userContent")]
            [Table("content", "userContent", "{name}")]
            string content,
            [Queue("userContent")]
            [CosmosDB("content", "userContent")]
            [Table("content", "userContent", "{name}")]
            [Blob("userContent/{name}", FileAccess.Write)]
            Stream output)
      {
            ...
      }
}
```

Answer:



Explanation:



Box 1: [BlobTrigger(..)]

Box 2: [Blob(..)]

Azure Blob storage output binding for Azure Functions. The output binding allows you to modify and delete blob storage data in an Azure Function.

The attribute's constructor takes the path to the blob and a FileAccess parameter indicating read or write, as shown in the following example:

```
[FunctionName("ResizeImage")]
public static void Run(
[BlobTrigger("sample-images/{name}")] Stream image,
[Blob("sample-images-md/{name}", FileAccess.Write)] Stream imageSmall)
{
}
```

Scenario: You must create an Azure Function named CheckUserContent to perform the content checks.

The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

Reference:

https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-output

NEW QUESTION: 30

□□ □□ □□□□ □□□□ □□□□. POS □□□□□□□ Azure Service Bus □□ □□□ □ □□ □□□□□. □□□ □□□ □□□□ □□ □□□□ □□□□□.

Property	Description
ShipLocation	the country/region where the order will be shipped
CorrelationId	a priority value for the order
Quantity	a user-defined field that stores the quantity of items in an order
AuditedAt	a user-defined field that records the date an order is audited

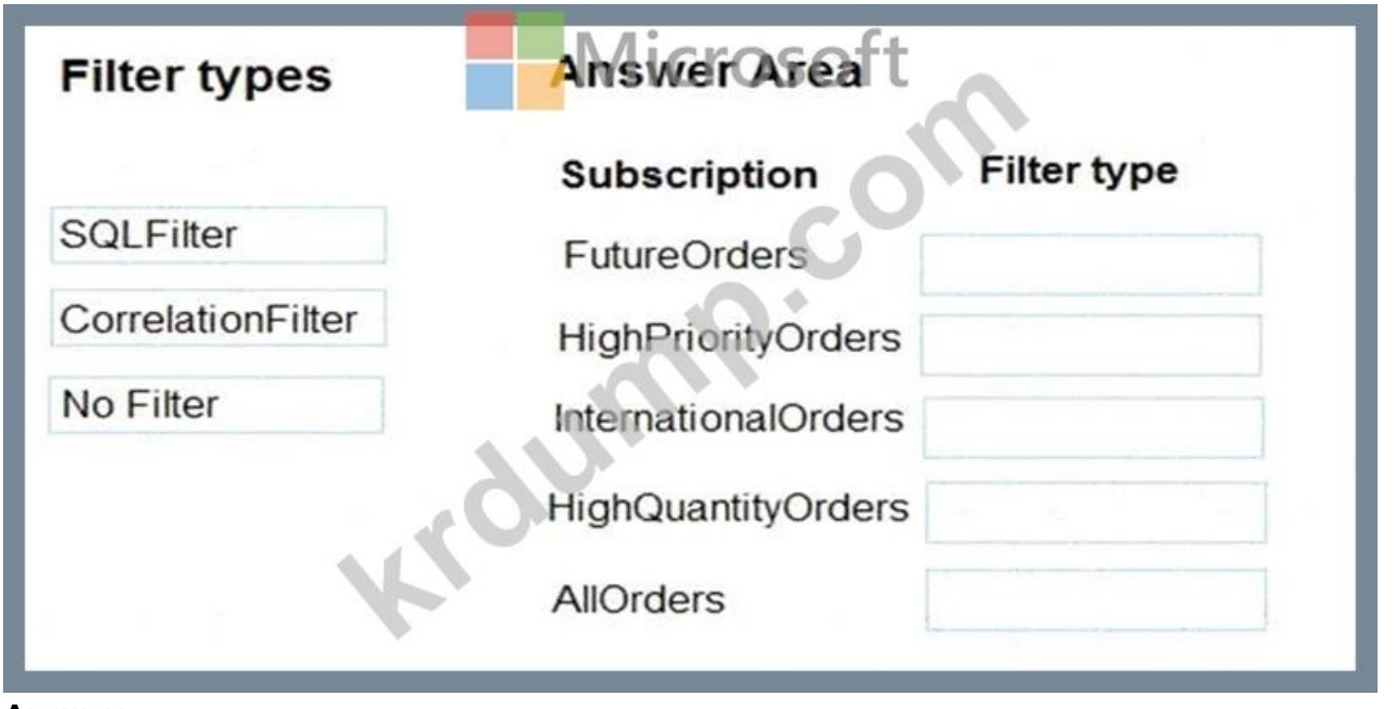
□□□□ □□□ □□ □□□ □□ □□ □□□ □□□□.

Subscription type	Comments
FutureOrders	This subscription is reserved for future use and must not receive any orders.
HighPriorityOrders	Handle all high priority orders and International orders.
InternationalOrders	Handle orders where the country/region is not United States.
HighQuantityOrders	Handle only orders with quantities greater than 100 units.
AllOrders	This subscription is used for auditing purposes. This subscription must receive every single order. AllOrders has an Action defined that updates the AuditedAt property to include the date and time it was received by the subscription.

□□□ □□□□ □□ □□□□ □□□□ □□□□ □□□□□ □□□.

□□ □□ □□□ □□□□ □□□? □□ □□□□ □□□ □□ □□□ □□ □□□□ □□□□□ □. □ □□ □□□ □ □, □□ □ □□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□ □ □□ □□□ □□□□□□ □□□□□ □ □□ □□□□.

□□: □□ □□□ 1□□□□.



Answer:

Explanation:

FutureOrders: SQLFilter
HighPriortyOrders: CorrelationFilter
CorrelationID only
InternationalOrders: SQLFilter
Country NOT USA requires an SQL Filter
HighQuantityOrders: SQLFilter
Need to use relational operators so an SQL Filter is needed.
AllOrders: No Filter
SQL Filter: SQL Filters - A SqlFilter holds a SQL-like conditional expression that is evaluated in the broker against the arriving messages' user-defined properties and system properties. All system properties must be prefixed with sys. in the conditional expression. The SQL-language subset for filter conditions tests for the existence of properties (EXISTS), as well as for null-values (IS NULL), logical NOT/AND/OR, relational operators, simple numeric arithmetic, and simple text pattern matching with LIKE.

Correlation Filters - A CorrelationFilter holds a set of conditions that are matched against one or more of an arriving message's user and system properties. A common use is to match against the CorrelationId property, but the application can also choose to match against ContentType, Label, MessageId, ReplyTo, ReplyToSessionId, SessionId, To, and any user-defined properties. A match exists when an arriving message's value for a property is equal to the value specified in the correlation filter. For string expressions, the comparison is case-sensitive. When specifying multiple match properties, the filter combines them as a logical AND condition, meaning for the filter to match, all conditions must match.
Boolean filters - The TrueFilter and FalseFilter either cause all arriving messages (true) or none of the arriving messages (false) to be selected for the subscription.
References:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/topic-filters

NEW QUESTION: 31
GitHub □□□□□□□ Azure Web App□ ASP.NET Core □ □□□□ □□□ □□□ □□ □□ □□. □□□□□□ □□□□□ □□□ □□ □□□□ □□□□ □□□□.
Azure Web App □□ □□ □□□ □□□ □□□□□.
□ □□□□□ □□□ □□□□ □□□□ □□ □□ □□ □□□□□ □□□□ □□□.
□ □□□ □□□□ □□ □□□ □ □□ □□□ □□□□□? □ □□□ □□□ □□□□ □□□□ □.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□ □□□□ □□□□ □ □□□□ □□□□ □□□□□ □□□□ □□□ □□ □ .deployment□□ □□□ □□□□.
B. □□ □□□ □□ □□□□□ □□□□ □□□□ csproj □□□□ □□□ PreBuild □□□ □□ □□□.
C. □□ □□□□ □□□□ □ □□□□ □□□□ □□□□□ □□□□ run.cmd□□ □□□ /run □□□ □□□□.
D. host.json □□□ WEBSITE_RUN_FROM_PACKAGE □□□ □□ □□□ □□ □□□ □□□ □□□□□.
Answer: (SHOW ANSWER)
A: To customize your deployment, include a .deployment file in the repository root.
You just need to add a file to the root of your repository with the name .deployment and the content:
[config]
command = YOUR COMMAND TO RUN FOR DEPLOYMENT
this command can be just running a script (batch file) that has all that is required for your deployment, like copying files from the repository to the web root directory for example.
D: In Azure, you can run your functions directly from a deployment package file in your function app. The other option is to deploy your files in the d:\home\site\wwwroot directory of your function app (see A above).
To enable your function app to run from a package, you just add a

WEBSITE_RUN_FROM_PACKAGE setting to your function app settings.
Note: The host.json metadata file contains global configuration options that affect all functions for a function app.
References:
https://github.com/projectkudu/kudu/wiki/Custom-Deployment-Script
https://docs.microsoft.com/bs-latn-ba/azure/azure-functions/run-functions-from-deployment-package

NEW QUESTION: 32
ASP.NET□ □□□□ □□□□ □□ □□□□ □□□□ □□□□. □ □□□□ □□□□ □□□ □ □□ □□□ □ □□ □□, □□, □ □ □□ □ □□□ □□□□□□.
□□□□□□□ □□□□ □□□□ □□□□□ □□□□ □□□ □ □□□ Azure Search□ □□ □□□.
Azure □□ □□□□□ □□□□ □□□ □ □□□□ □□□ □□□ □□□ □□□□ □□□.
□□ □ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□□□□
B. SearchIndexClient
C. □□ □□□ □□□□□
D. □□ □□ □□
Answer: (SHOW ANSWER)
The various client libraries define classes like Index, Field, and Document, as well as operations like Indexes.
Create and Documents.Search on the SearchServiceClient and SearchIndexClient classes.
Example:
The sample application we'll be exploring creates a new index named "hotels", populates it with a few documents, then executes some search queries. Here is the main program, showing the overall flow:
/ This sample shows how to delete, create, upload documents and query an index static void Main(string[] args)
{
IConfigurationBuilder builder = new ConfigurationBuilder().AddJsonFile("appsettings.json");
IConfigurationRoot configuration = builder.Build(); SearchServiceClient serviceClient =

CreateSearchServiceClient(configuration); Console.WriteLine("{0}", "Deleting index...\n"); DeleteHotelsIndexIfExists(serviceClient); Console.WriteLine("{0}", "Creating index...\n"); CreateHotelsIndex(serviceClient); ISearchIndexClient indexClient = serviceClient.Indexes.GetClient("hotels"); References: https://docs.microsoft.com/en-us/azure/search/search-howto-dotnet-sdk

NEW QUESTION: 33
□□□□ □□ □□□ □□□□ □□□□□□□ □□□□ □□□□. □□□□□□□ □□□□□ Azure Cosmos D8□ □□□□□.
□□ □□□ □□ □ □□□□□□□ □□□□□□ □□□□□ □□□. □□ □□□ □□ □ □□ □□□□□ Azure□ □□□□□ □□□.
□□□□□□□ □□ Azure Cosmos D8 API□ □□□□ □□□?
A. □□□
B. □□□
C. □□
D. □□□□
Answer: D (LEAVE A REPLY)

NEW QUESTION: 34
Azure□ □□□ Java □□□□□□□ □□□□ □□□□. □□□□□□□ Azure Cosmos DB□ □□□ □□□□ □□□□□. □□□□□□ □□□ □□□ □□□□ □□□□□□ Always Encrypted□ □□□□ □□□.
□□ □□□ □□ □□□?
A. Azure AD □□ ID□ □□□ □ Azure Key Vault □□□□□ ID□ □□□□□.
B. CMK(□□ □□□ □)□ □□□ □ Azure Key Vault □□□□□ □□ □□□□□.
C. □□□□ JSON □□□ □□ □□□ □□□ □□□ □ □□□□□ □□□□.
D. Azure Cosmos DB SDK□ □□□□ DEK(□□□ □□□ □)□ □□□ Azure Cosmos DB□ □ □ □□□□□.
Answer: B (LEAVE A REPLY)

NEW QUESTION: 35
Azure Storage □□ □□□□ □□□□□□□ □□□□ □□□□.
□□ □□□ □□□□.

```
CloudStorageAccount storageAccount = CloudStorageAccount.Parse
(CloudConfigurationManager.GetSetting("StorageConnectionString"));
CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient()

CloudQueue queue = queueClient.GetQueueReference("appqueue") ;
await queu.CreateIfNotExistsAsync() ;

CloudQueueMessage peekedMessage = await queue.PeekMessageAsync() ;
if (peekedMessage != null)
{
     Console.WriteLine("The peeked message is: {0}", peekedMessage.AsString);
}
CloudQueueMessage message = await queue.GetMessageAsync() ;
```

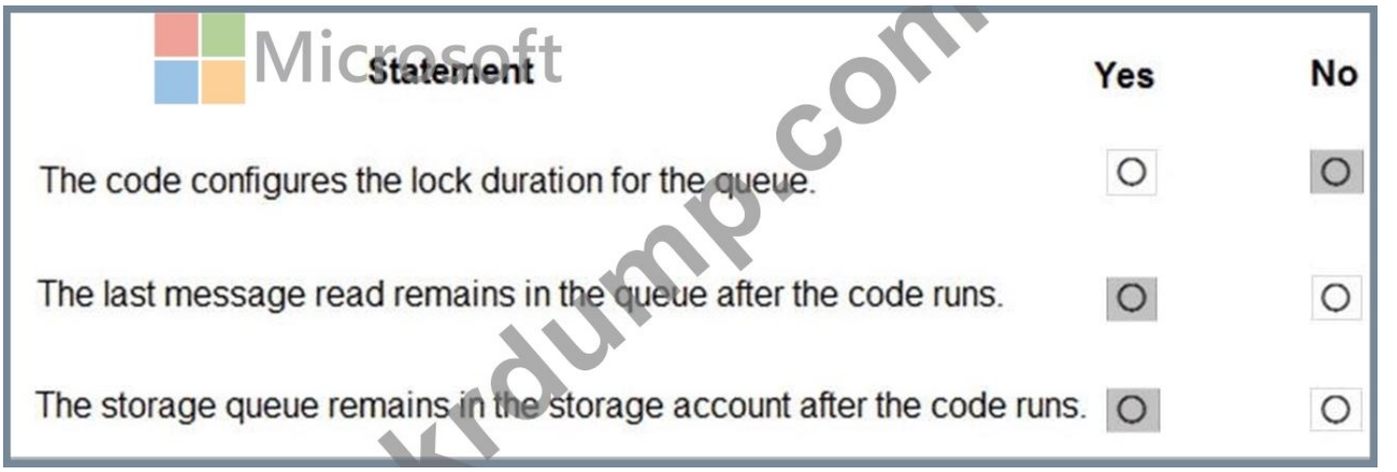
□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Statement	Yes	No
The code configures the lock duration for the queue.	○	○
The last message read remains in the queue after the code runs.	○	○
The storage queue remains in the storage account after the code runs.	○	○

Answer:

Statement	Yes	No
The code configures the lock duration for the queue.	○	○
The last message read remains in the queue after the code runs.	○	○
The storage queue remains in the storage account after the code runs.	○	○

Explanation:



Statement	Yes	No
The code configures the lock duration for the queue.	○	○
The last message read remains in the queue after the code runs.	○	○
The storage queue remains in the storage account after the code runs.	○	○

Box 1: No

The QueueDescription.LockDuration property gets or sets the duration of a peek lock; that is, the amount of time that the message is locked for other receivers. The maximum value for LockDuration is 5 minutes; the default value is 1 minute.

Box 2: Yes

You can peek at the message in the front of a queue without removing it from the queue by calling the PeekMessage method.

Box 3: Yes

Reference:

https://docs.microsoft.com/en-us/azure/storage/queues/storage-dotnet-how-to-use-queues

https://docs.microsoft.com/en-us/dotnet/api/microsoft.servicebus.messaging.queuedescription.lockduration

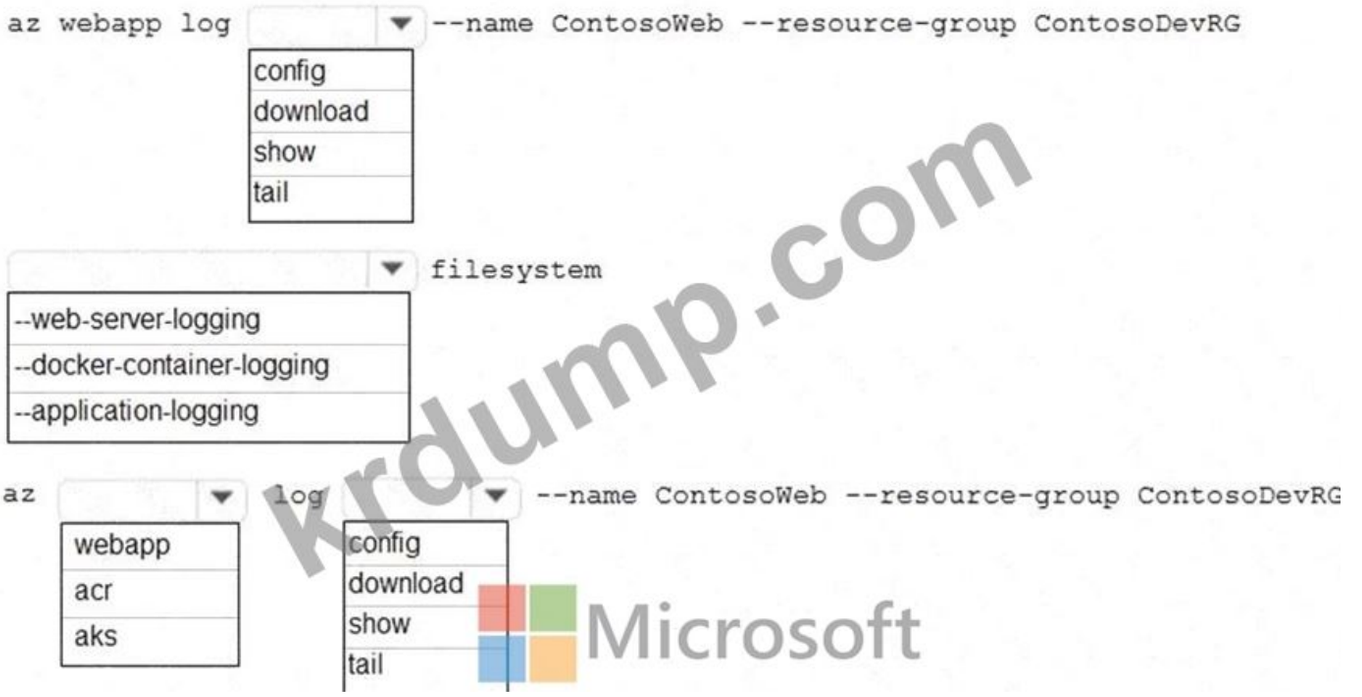
NEW QUESTION: 36

Linux□ App Service□ □□□ □□□ □□□□□. App Service □□□ □□□□. □□□ □□□ □□□ □□ Docker □□□□ □□□ Azure Container Registry□ □□□□□.

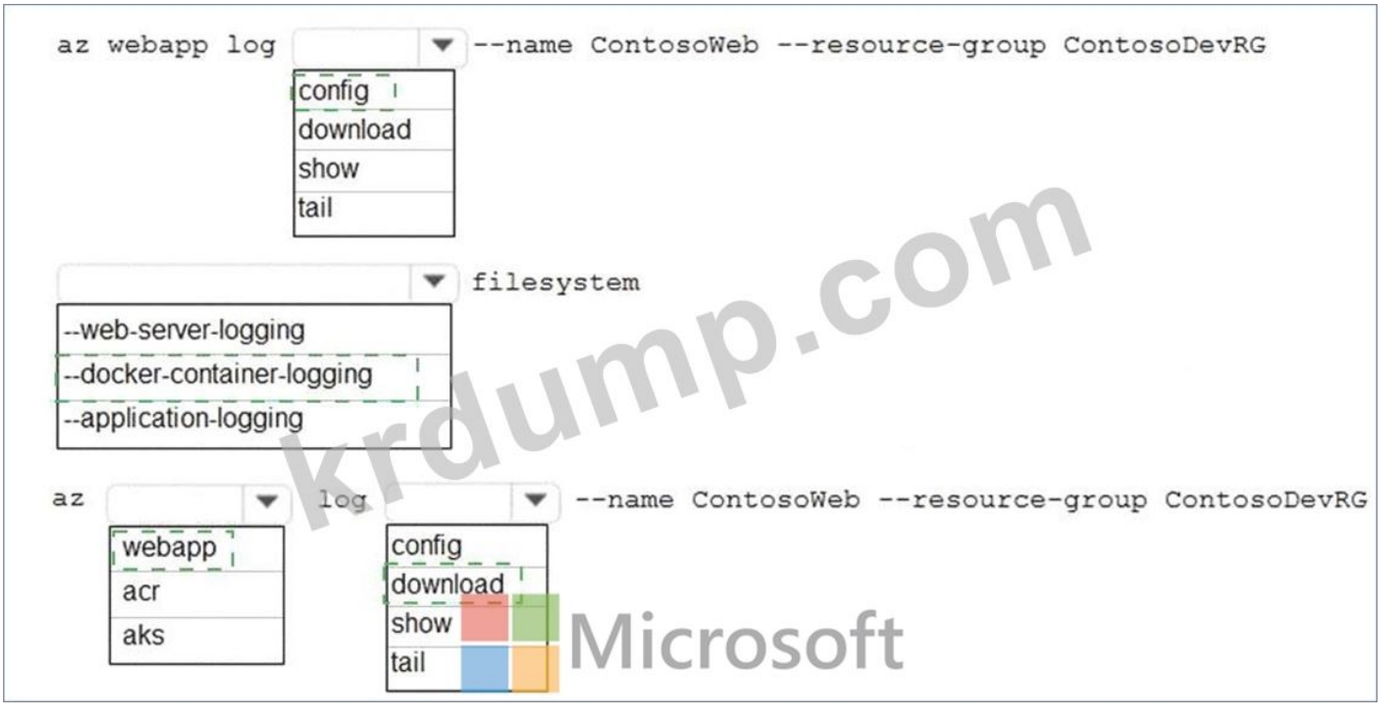
□□□□ □□□□ □□□□ □□ □□□ □□□□□ □□□□ □□□.

Azure CLI □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

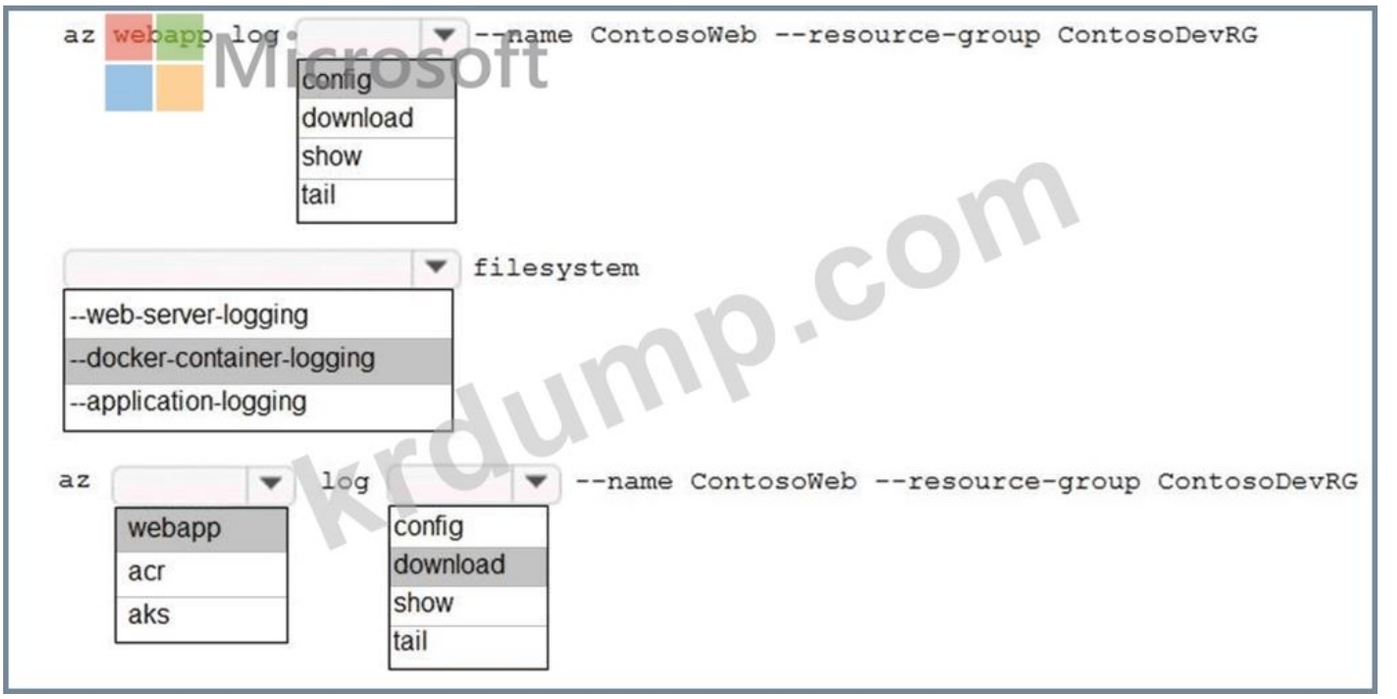
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: config

To Configure logging for a web app use the command:

az webapp log config

Box 2: --docker-container-logging

Syntax include:

az webapp log config [--docker-container-logging {filesystem, off}]

Box 3: webapp

To download a web app's log history as a zip file use the command:

az webapp log download

Box 4: download

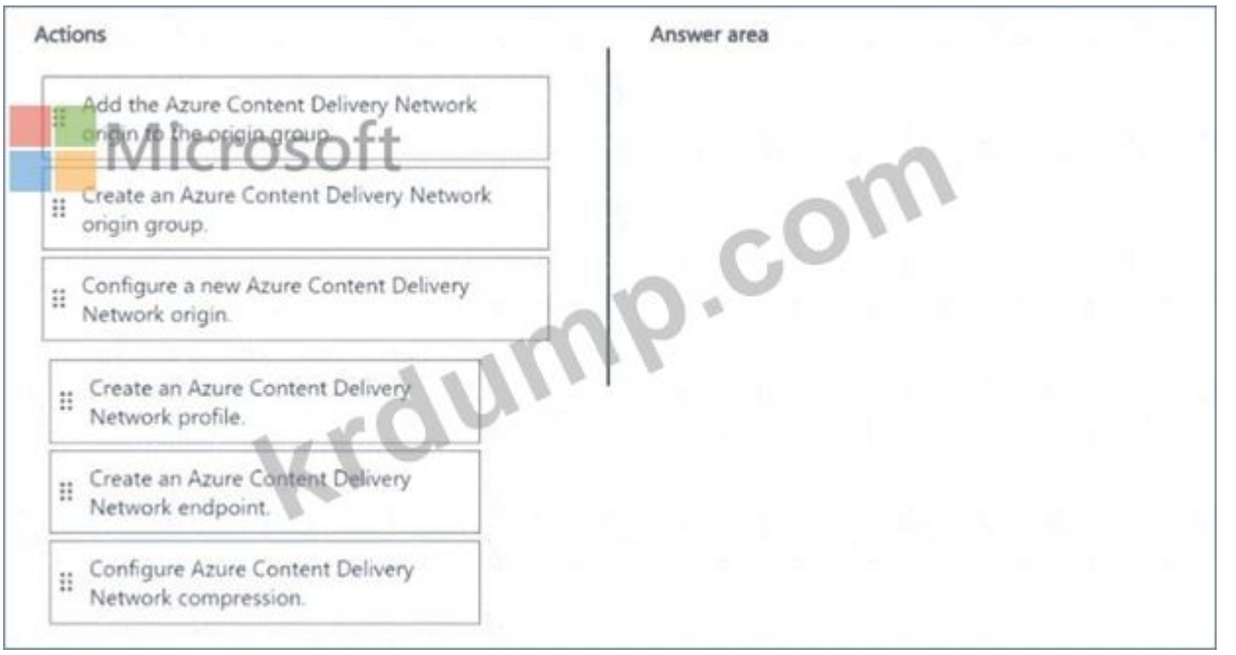
References:

https://docs.microsoft.com/en-us/cli/azure/webapp/log

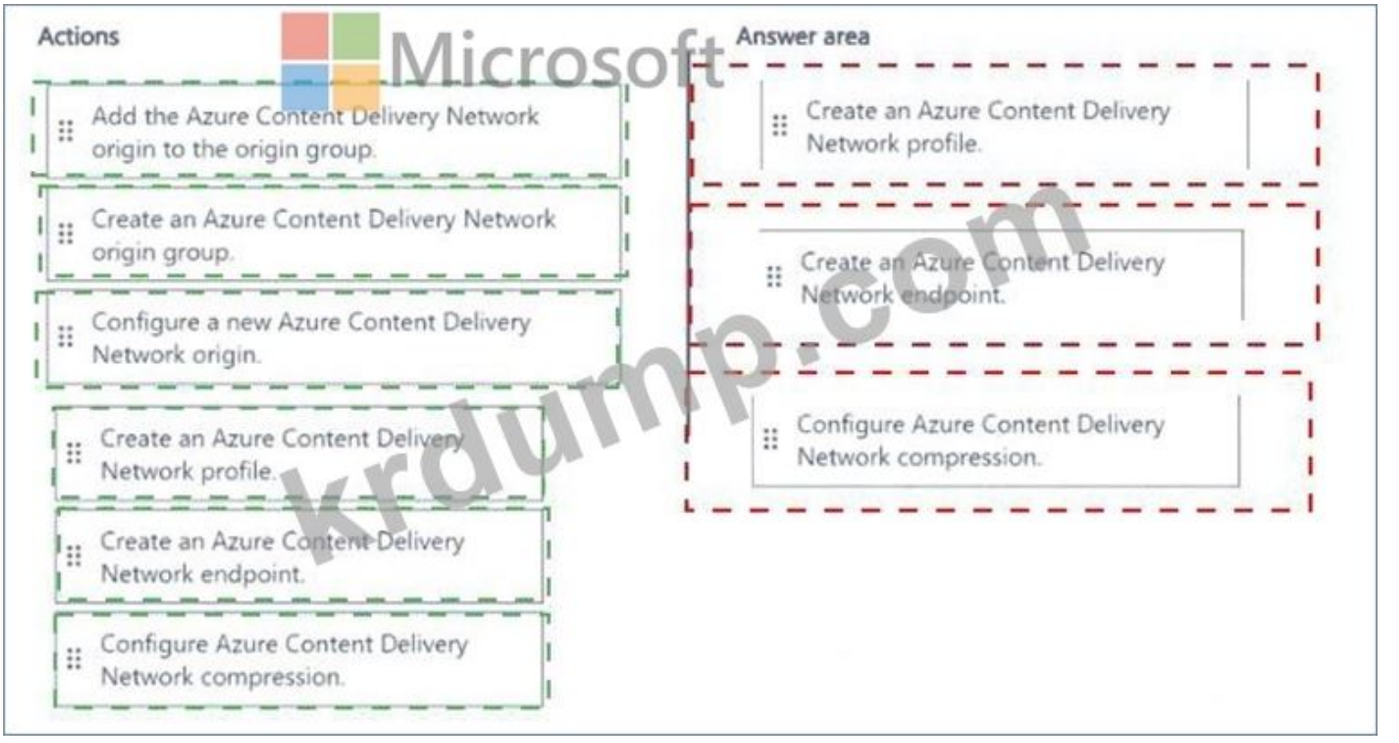
NEW QUESTION: 37

□□□□□ □□□ □□ □□ □□□ □□□□ □□□.

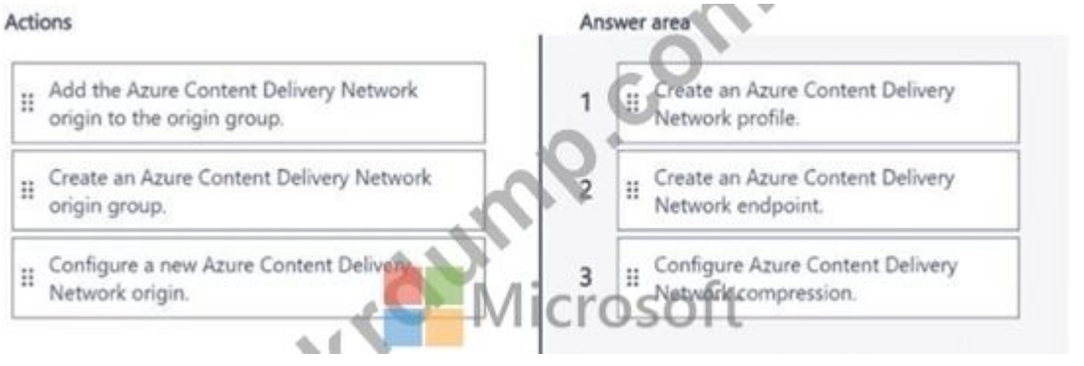
□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□ □ □□□ □□□ □□□ □□□ □□□□□.



Answer:



Explanation:



NEW QUESTION: 38

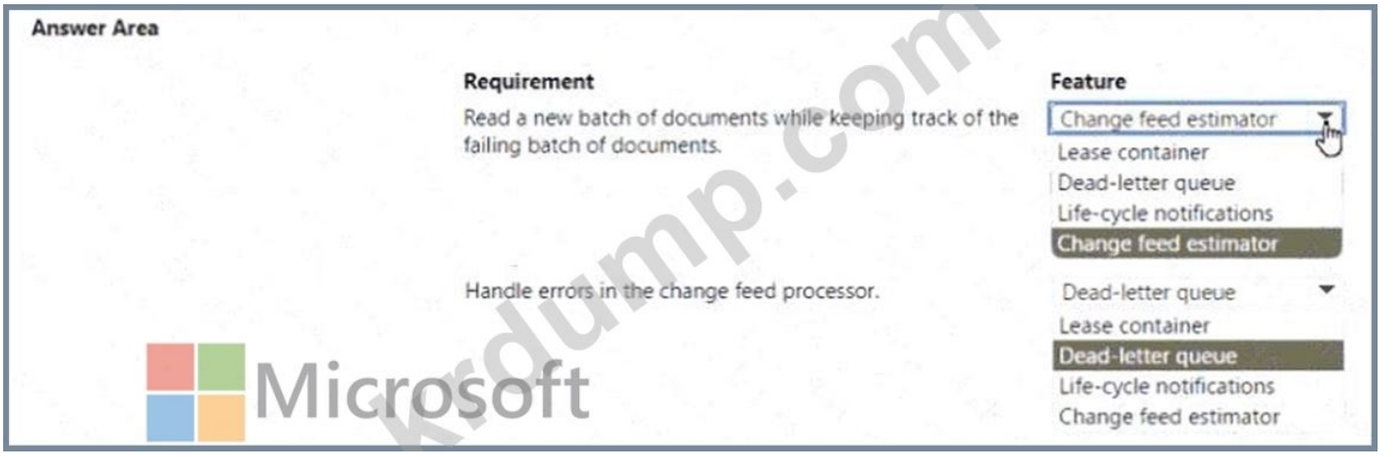
Azure Cosmos DB□ □□□□□ Azure □□□□ □□□□□. □□ Azure Cosmos DB SDK □ Azure Cosmos DB for NoSQL API. □□ Azure Cosmos DB□ Azure Functions □□□□ □□□□ □ □□□□ □□□□□□ □□ □□ □□□□□ □□□□□.

□□ □□ □ □□□ □□ □ □□ □□ □□ □□□ □□□□ □□□□□. □□□ □□ □□□ □ □□□ □□□ □□ □□□□□ □□□□□ □ □□ □□□ □□□ □□□.

□□□ □□□□ □□ □□ □□□□□ □□□□ □□□.

□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□ .

□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:

Explanation:

NEW QUESTION: 39

□ □□ □□ □□ □□□□ □□ □□□□ □□□□ Azure □□□□ □□□□ □□□□. □ □□ □□□ □□□□ □□□□ □□□ Azure Blob □□□□ □□□□ □□□□ □□□□□.

□□□□ □□ □□ □□□ □□□□ □□□.

* □□□□ Azure Blob □□□□ □□□□ □□□ □□□□□.

* □□ □□ □□□ □□□□ □□□□ □□□□□□.

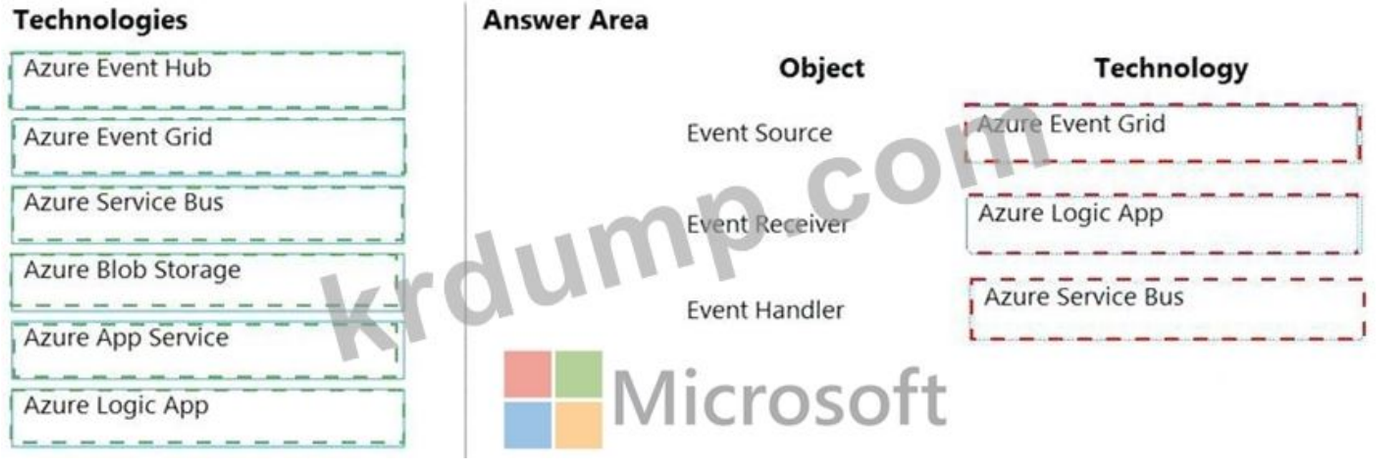
* Azure Cosmos DB□ □□□ □□□□ □□□□□ Azure Logic App□ □□□□□□.
* □□□□ □ □□□ □□□ □□□□□□.
* □□□□□ 24□□□ □□□□□.
* □□ □□□ □□□ □□□ □□□□□.

□□□□ □□□□ □□□.

□□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



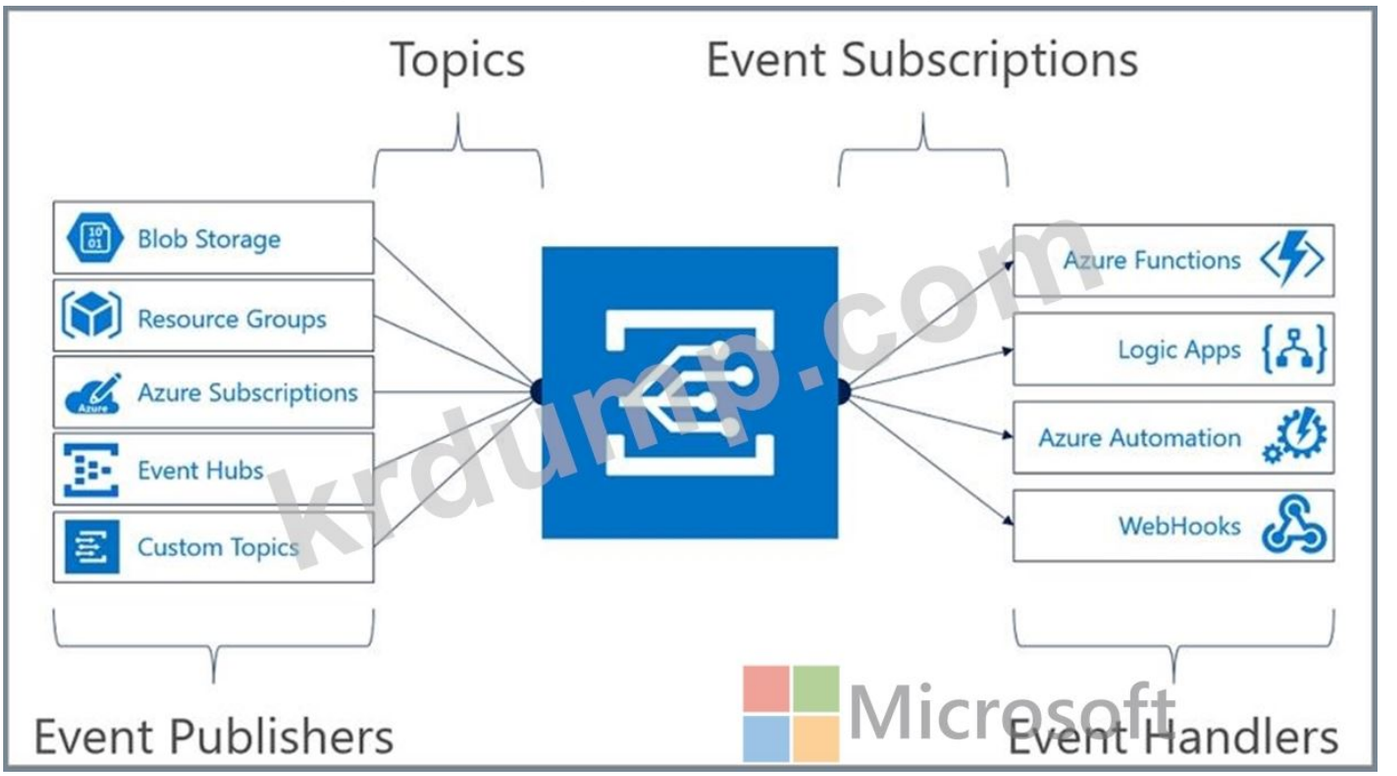
Explanation:

Box 1: Azure Event Grid

Blob storage events are pushed using Azure Event Grid to subscribers such as Azure Functions, Azure Logic Apps, or even to your own http listener. Event Grid provides reliable event delivery to your applications through rich retry policies and dead-lettering.

Box 2: Azure Logic App

Event Grid uses event subscriptions to route event messages to subscribers. This image illustrates the relationship between event publishers, event subscriptions, and event handlers.



Box 3: Azure Service Bus

The Event Grid service doesn't store events. Instead, events are stored in the Event Handlers, including ServiceBus, EventHubs, Storage Queue, WebHook endpoint, or many other supported Azure Services.

Reference:

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-event-overview

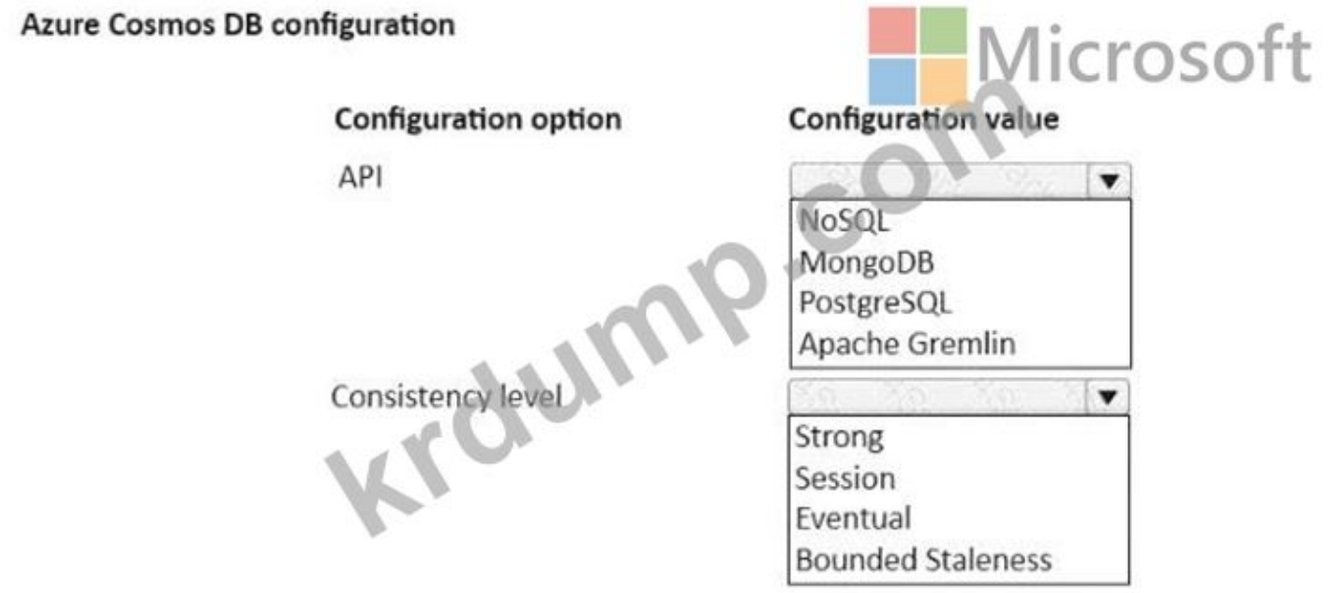
https://docs.microsoft.com/en-us/java/api/overview/azure/messaging-eventgrid-readme

NEW QUESTION: 40

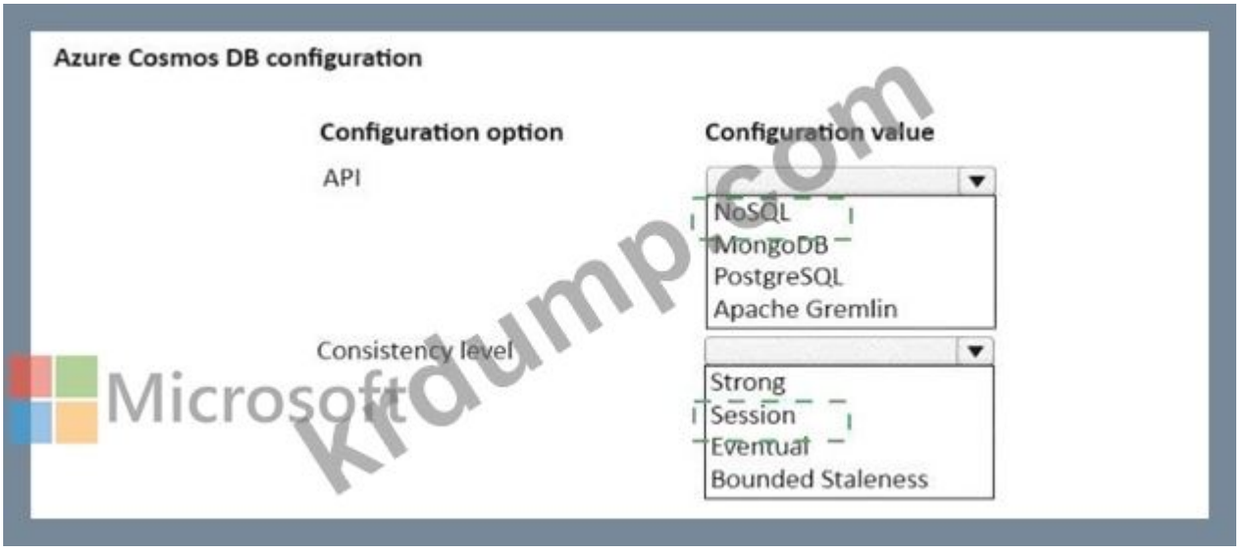
Azure Cosmos DB□ □□□ □□ □□□ □□□□ □□□.

□□ Azure Cosmos DB □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□ □□□.

□□□□: □□ □□□ 1□□□□.



Answer:



Explanation:

1. API:

NoSQL

Reasoning: Since the development must use a native API that stores data in a document format, the NoSQL API is the correct choice. Azure Cosmos DB's NoSQL API stores data in JSON documents and is the most commonly used API for document-based use cases, which aligns with the storage of customized items in the scenario.

2. Consistency Level:

Session

Reasoning: The scenario requires that customized items maximize throughput while ensuring that the data is accurate for the current user. Session consistency provides the best balance between performance and data accuracy for scenarios where multiple reads and writes are performed by the same user session. It ensures that a user reads their own writes, which is important for ensuring that customized items are accurate for the current user.

Final answer:

API: NoSQL

Consistency level: Session

NEW QUESTION: 41

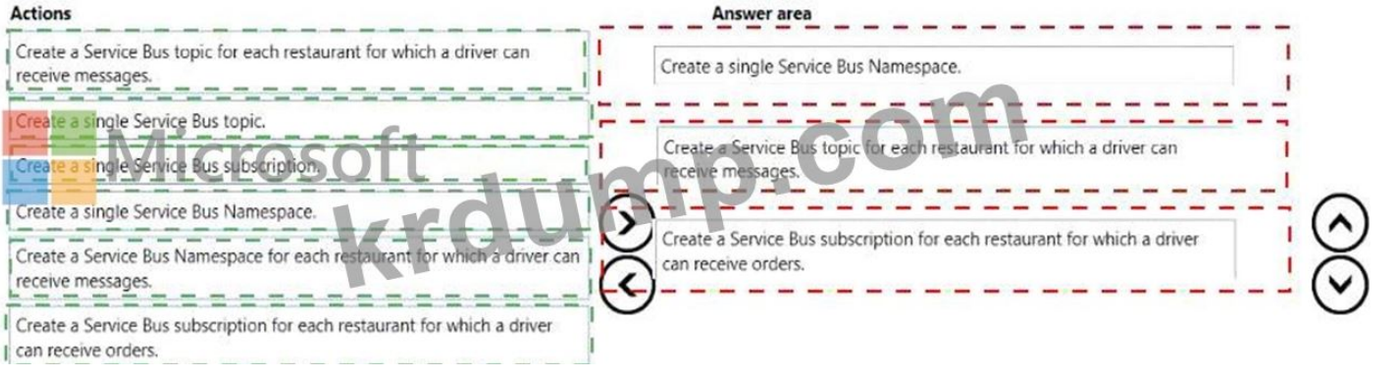
□□□ □□ □□□□ □□ □□□□□ □□□□ □□□□□. □□□□ □□ □□□ □□□□ □ □□□ □ □□□ □ □□ □□□ □□ □□□□ □□□□. □□ □□ □□□□□ □□□□□.

1. □□□□ □□□ □□□ □□□□□ □□□□□.
2. □□□ □□ □□□□ □□ □□□ □□ □□□□□ □□□□□.
3. □□□ □□□□□ □□□ □□□□□□ □□□□□.
4. □□□ □□□ □ □□ □□□□□ □□ □□□ □□ □□□□ □□ □□□ □□□□□.

Azure Service Bus □□□□ □□□□ □□□.

□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□ □□□□□□.

Answer:



Explanation:

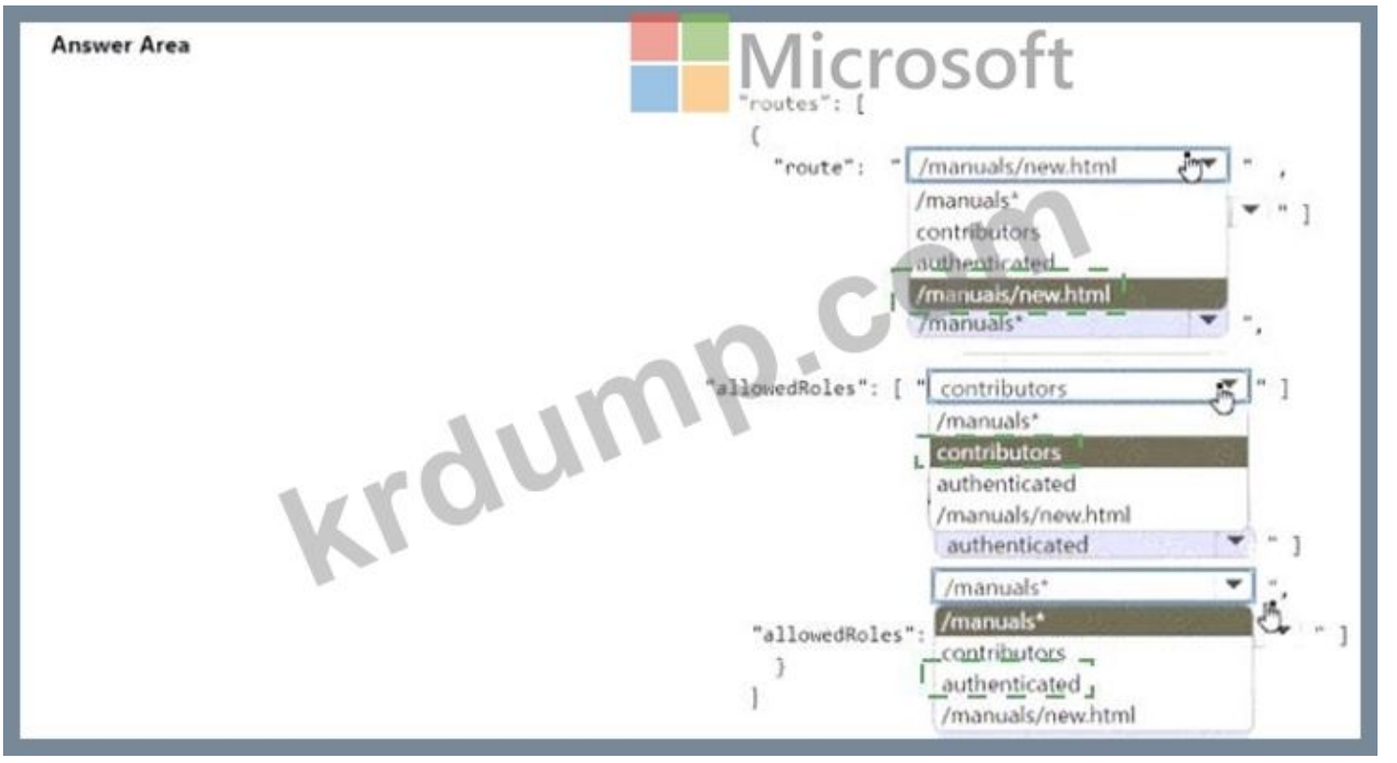
Box 1: Create a single Service Bus Namespace

To begin using Service Bus messaging entities in Azure, you must first create a namespace with a name that is unique across Azure. A namespace provides a scoping container for addressing

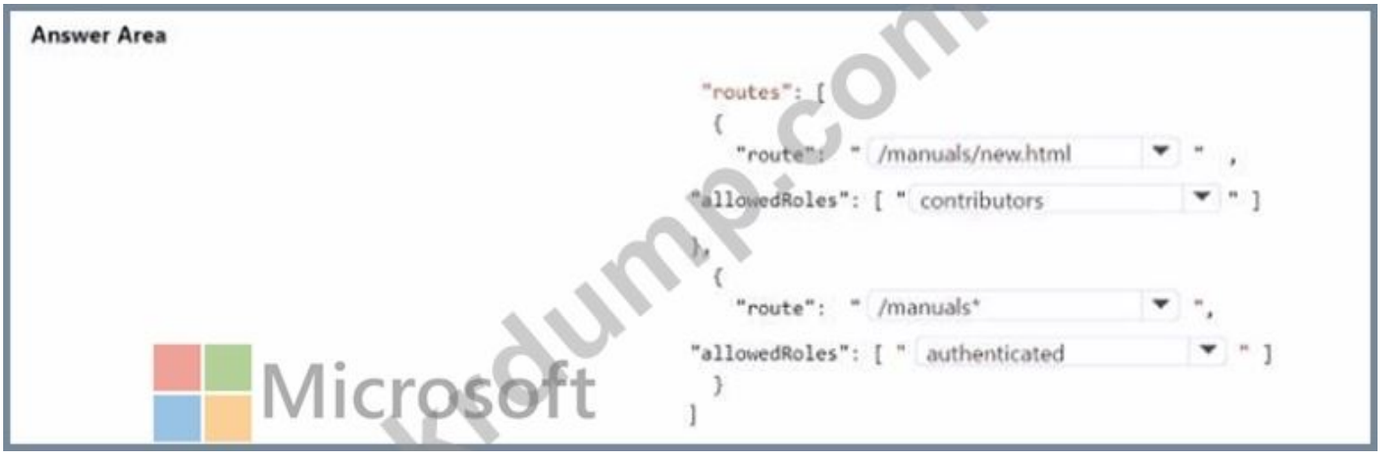
Service Bus resources within your application.
Box 2: Create a Service Bus Topic for each restaurant for which a driver can receive messages.
Create topics.
Box 3: Create a Service Bus subscription for each restaurant for which a driver can receive orders.
Topics can have multiple, independent subscriptions.
References:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview

NEW QUESTION: 42
□□ □□□□ □□□ □□ □□□□□□□ □□□□ □□□□. □□□□□□□ Azure Static Web □□□ □□□□□.
□□□ □□□□ /manuals □□□ □□□□ □ □ □□□ □□□□ /manuals/new html □□□□ □□□□ □ □□□□.
□□□ □□ □□□□ □□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:



NEW QUESTION: 43

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

Azure Storage Blob □□□□ □□□□ □□ HTTP □□□ Azure □□ □□ □□□□□. □□ Blob □ □□ □□□□ □□□□ □□□□□□.

□□ 4□ □□□ □□ □□ □□□□□. □□ Blob □□□□ □□□□ □□□.

□□ □□ □□□□ □□ Blob □□□□ □□□□□ □□□□ □□□.

□□ □□: host.json □□□□ □□□ functionTimeout □□□ 10□□□ □□□□□□□.

□□□□ □□□ □□□□□?

A. □

B. □□□

Answer: (SHOW ANSWER)

Instead pass the HTTP trigger payload into an Azure Service Bus queue to be processed by a queue trigger function and return an immediate HTTP success response.

Note: Large, long-running functions can cause unexpected timeout issues. General best practices include:

Whenever possible, refactor large functions into smaller function sets that work together and return responses fast. For example, a webhook or HTTP trigger function might require an acknowledgment response within a certain time limit; it's common for webhooks to require an immediate response. You can pass the HTTP trigger payload into a queue to be processed by a queue trigger function. This approach lets you defer the actual work and return an immediate response.

Reference:

https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices

NEW QUESTION: 44

□□□ Azure SQL Database□ □□□□ □□□ App1□□□ □□□□□□□ □□□□ □□□□.

App1□ □□ □□□□ □□□□ □□ □□□□□□.

App1□ □□□ □□□□ □□ □□□□ □□□□ □□□.

□□□□ □□□ □□□□ □□□?

A. Redis□ Azure □□

B. □□□□□ □□□

C. Azure HPC □□

D. CON □□□□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 45

Azure Blob □□□□□ □□□□ □□□□ □□□□ Azure □□ □□ □□□□ □□□□.

□□□□ □□□□ □ □□□ □□ □□□□□ □□ □□□□ □□ □□□ □□□□□ □□□. □ □ □□ □□□□ □ □□□□ □□□□ □□□ □□□□.

□□ □□ □□□□ □□□.

□□□ □□ □□□?

A. App Service □□□ □□□□□. Azure Blob Storage □□ □□□□ □□□□□ □□ □□ □□ □□□.

B. □□ □□□ □□□□□. Azure Blob Storage □□□□ □□□□□ □□ □□ □□□□□.

C. □□ □□□ □□□□□. □□□ □□□□ □□□□□ □□ □□ □□□□□.

D. App Service □□□ □□□□□. Azure Blob Storage □□□□ □□□□□ □□ □□ □□□□ □.

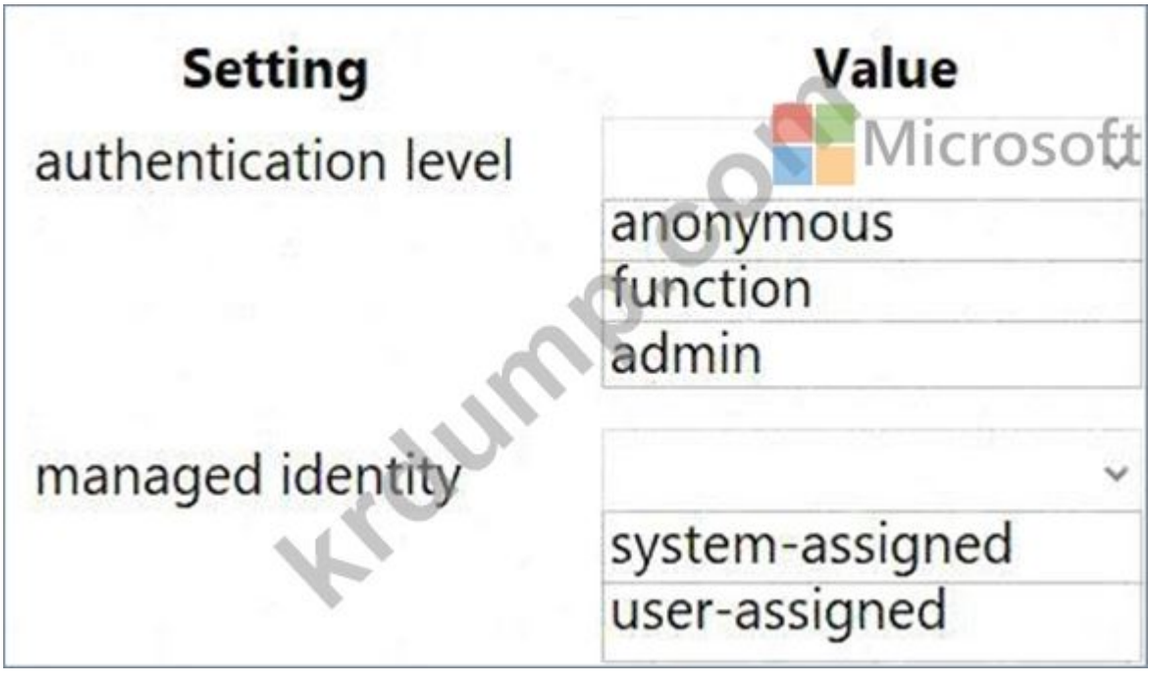
E. □□ □□□ □□□□□. Azure Blob Storage □□ □□□□ □□□□□ □□ □□ □□□□□.

Answer: B (LEAVE A REPLY)

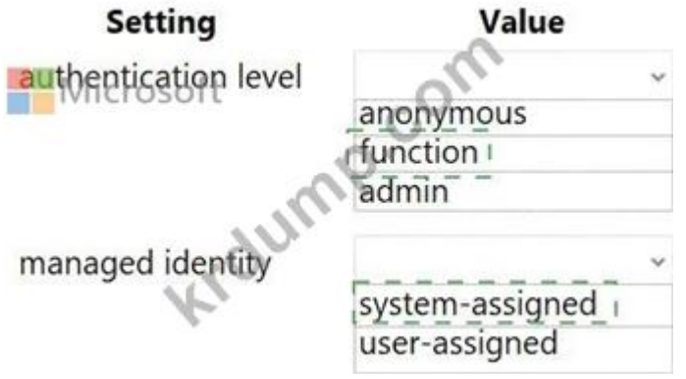
The Blob storage trigger starts a function when a new or updated blob is detected. The blob

contents are provided as input to the function.
The Consumption plan limits a function app on one virtual machine (VM) to 1.5 GB of memory.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger

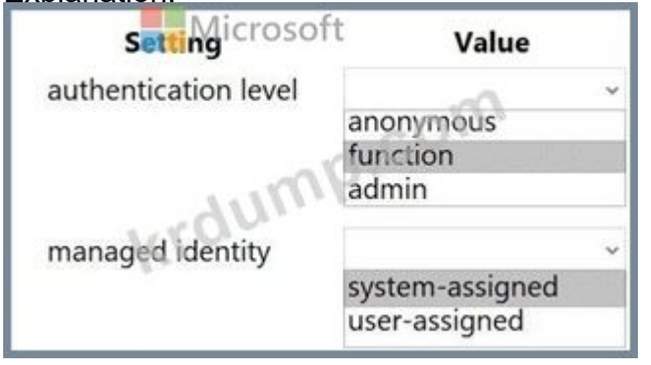
NEW QUESTION: 46
Azure Logic □ □□ □□□□ □□□□ □□□.
□□ □□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:

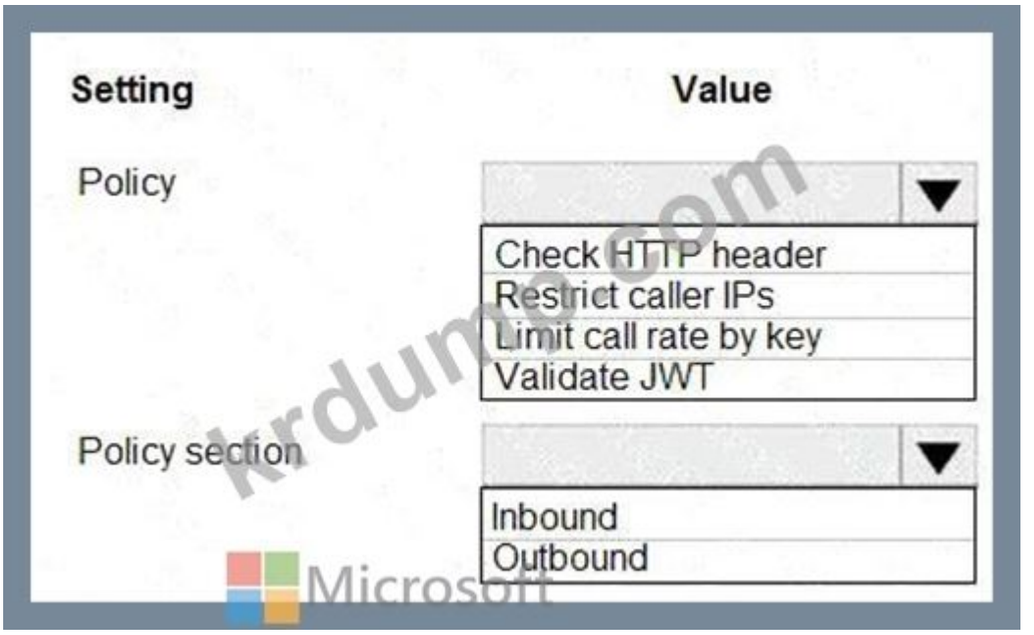


Explanation:

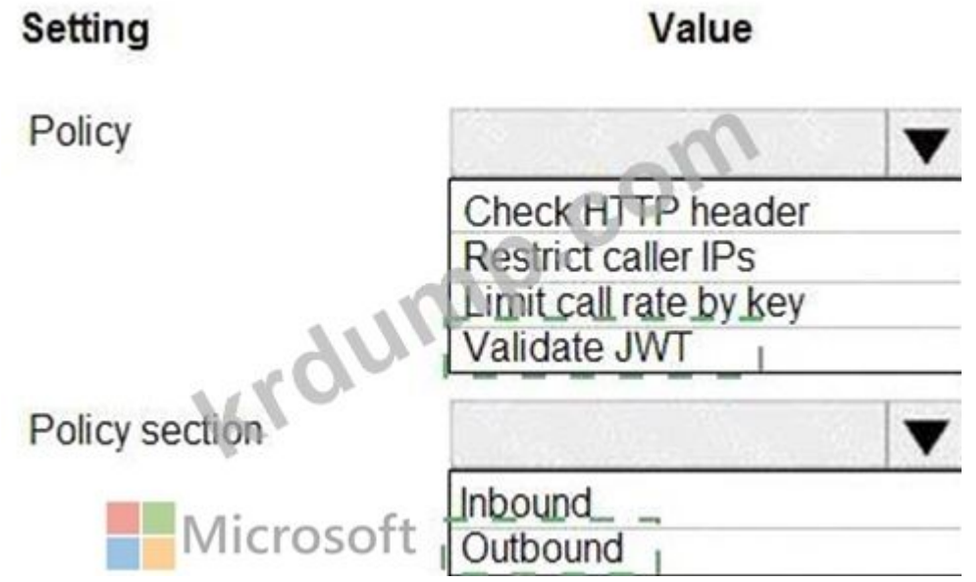


Scenario: You test the Logic app in a development environment. The following error message displays:
'400 Bad Request'
Troubleshooting of the error shows an HttpTrigger action to call the RequestUserApproval function.
Note: If the inbound call's request body doesn't match your schema, the trigger returns an HTTP 400 Bad Request error.
Box 1: function
If you have an Azure function where you want to use the system-assigned identity, first enable authentication for Azure functions.
Box 2: system-assigned
Your logic app or individual connections can use either the system-assigned identity or a single user-assigned identity, which you can share across a group of logic apps, but not both.
Reference:
https://docs.microsoft.com/en-us/azure/logic-apps/create-managed-service-identity

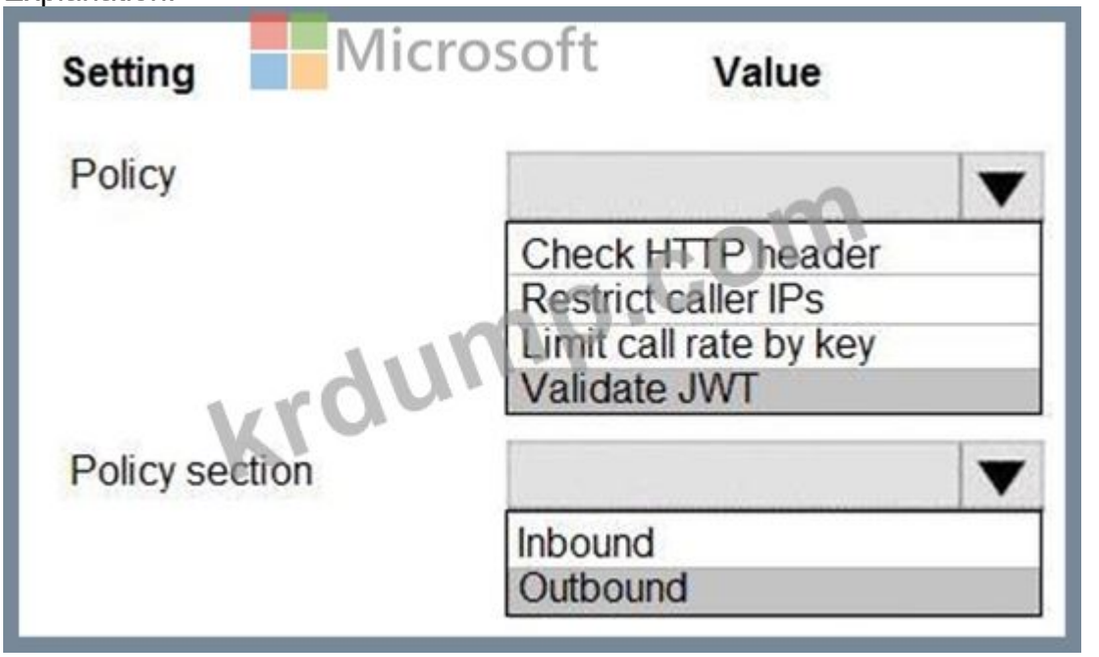
NEW QUESTION: 47
□□□ □□ API Management□ □□□□ □□□.
□□ □□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Validate JWT

The validate-jwt policy enforces existence and validity of a JWT extracted from either a specified HTTP Header or a specified query parameter.

Scenario: User authentication (see step 5 below)

The following steps detail the user authentication process:

The user selects Sign in in the website.

The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.

The user signs in.

Azure AD redirects the user's session back to the web application. The URL includes an access token.

The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.

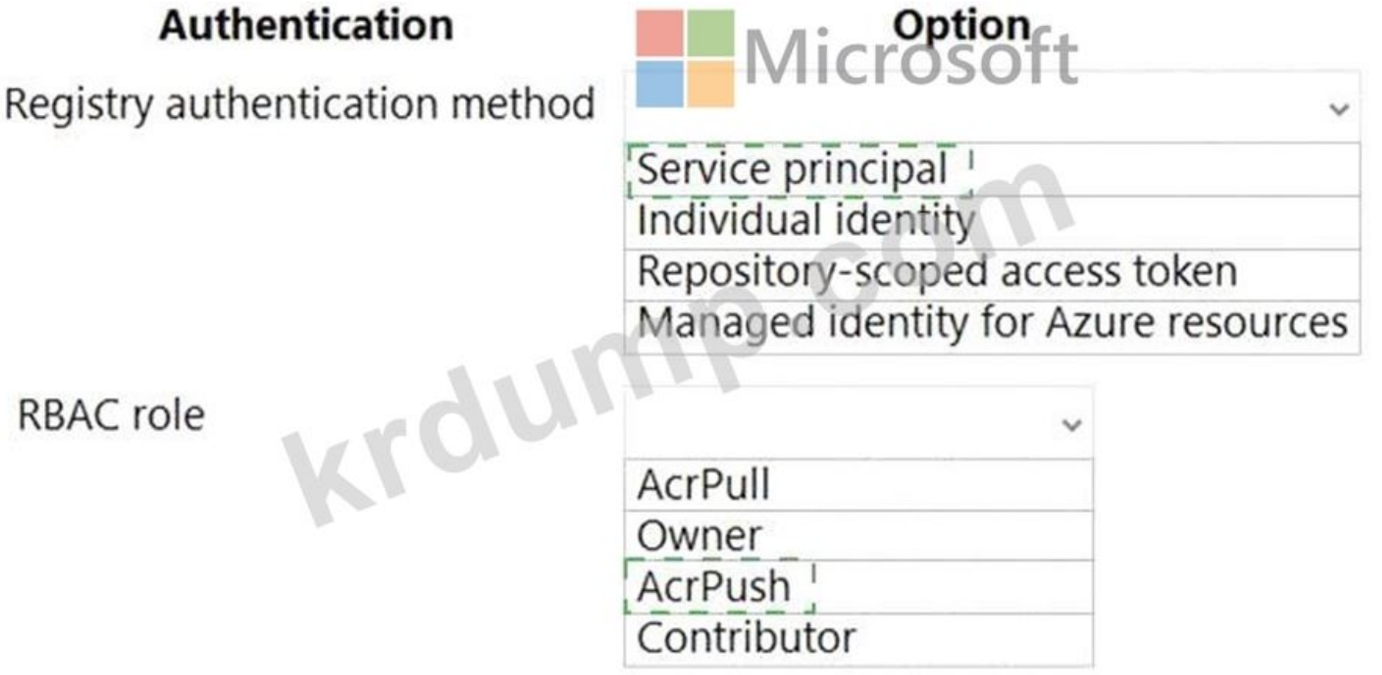
The back-end API validates the access token.

Box 2: Outbound
Reference:
https://docs.microsoft.com/en-us/azure/api-management/api-management-access-restriction-policies

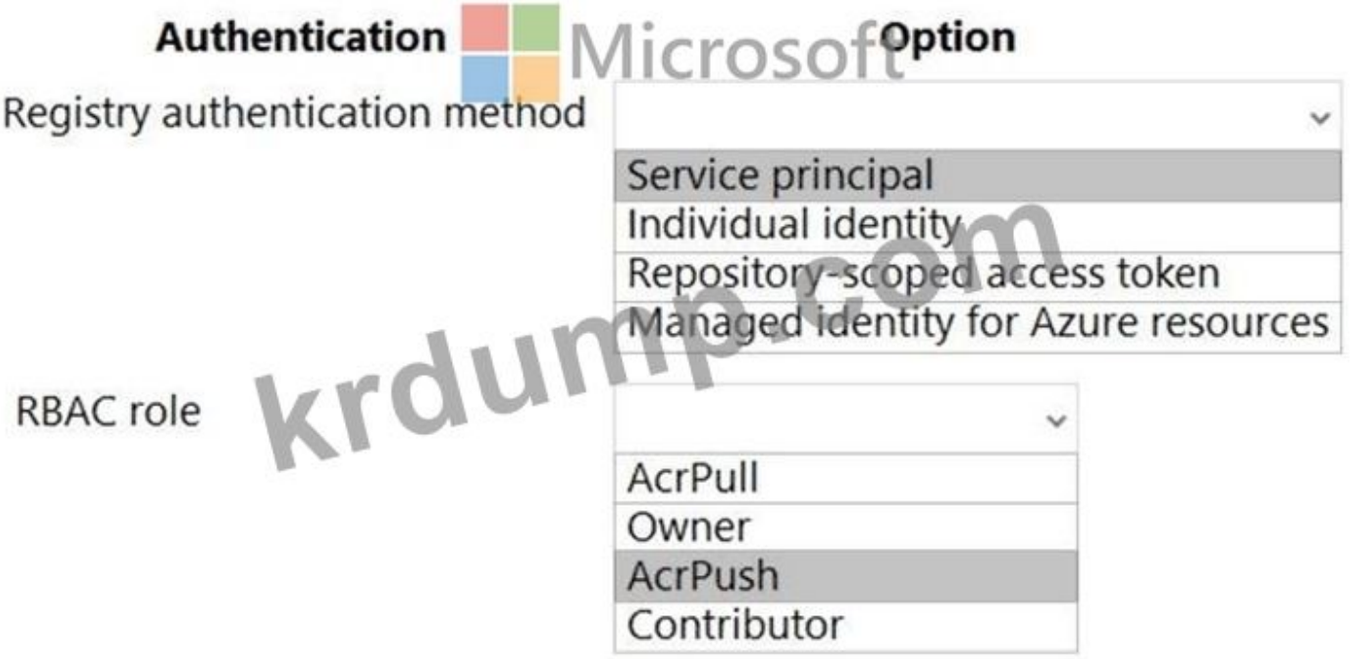
NEW QUESTION: 48
□□□□□□ □□□□□□□ □□□□□. □□ CI/CD(□□□□ □□ □ □□□□ □□□□) □□ □□□ □□□□ □ Azure □□□□ □□□□□ □□□□□□□ □□□ □□□□□.
□□□ □□□□□ □□ □□ □□□□□□ □□□ □□□□ □□□. □□ □□□□□ □□□□□ □□ □□□□ □□□□ □□□ □□ □□□ □□□. □□□ Azure AD(Azure Active Directory)□ □ □□□□ □□□. □□□□ □□ □□ □□□ □□□□ □□□.
□□ □□□□□ □□□□□□ □□□□ □ □□□ □□□□ □□□.
□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Authentication	Option
Registry authentication method	Service principal Individual identity Repository-scoped access token Managed identity for Azure resources
RBAC role	AcrPull Owner AcrPush Contributor

Answer:



Explanation:



Box 1: Service principal

Applications and container orchestrators can perform unattended, or "headless," authentication by using an Azure Active Directory (Azure AD) service principal.

Box 2: AcrPush

AcrPush provides pull/push permissions only and meets the principle of least privilege.

Reference:

https://docs.microsoft.com/en-us/azure/container-registry/container-registry-authentication?tabs=azure-cli

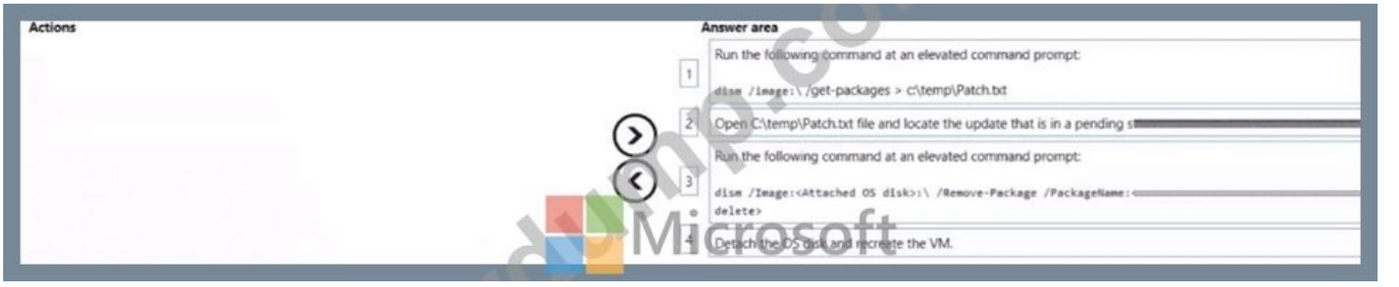
https://docs.microsoft.com/en-us/azure/container-registry/container-registry-roles?tabs=azure-cli

NEW QUESTION: 49

VM(□□ □□)□ □□ □□□□ □□□□□□□□.
□□□ VM□□ □□□□ □□□□.
VM□ Windows □□□□ □□□□□□ □□□□□□□. □□□ □□ VM□ OS □□□□ □□ VM□ □□□□□.
□□□ □□□□ □□□.
□□ □□□ □□□ □□□□ □□□' □□□□□ □□ □□□□ □□ □□□ □□ □□□□ □□ □□ □□□ □□□ □□□□□□.

Answer:

Explanation:



Remove the update that causes the problem
Take a snapshot of the OS disk of the affected VM as a backup.
Attach the OS disk to a recovery VM.
Once the OS disk is attached on the recovery VM, run diskmgmt.msc to open Disk Management, and ensure the attached disk is ONLINE.
(Step 1) Open an elevated command prompt instance (Run as administrator). Run the following command to get the list of the update packages that are on the attached OS disk:
dism /image:<Attached OS disk>:\ /get-packages > c:\temp\Patch_level
(Step 2) Open the C:\temp\Patch_level.txt file, and then read it from the bottom up. Locate the update that's in Install Pending or Uninstall Pending state.
Remove the update that caused the problem:
dism /Image:<Attached OS disk>:\ /Remove-Package /PackageName:<PACK
(Step 4) Detach the OS disk and recreate the VM. Then check whether the issue is resolved.
Reference:
https://docs.microsoft.com/en-us/troubleshoot/azure/virtual-machines/troubleshoot-stuck-updating-boot-error

NEW QUESTION: 50

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□□ □□□□ □□□□. □□□□ □□□ □□□ □□□□□ □□□□□.
Azure App Service□ □□ ASP.NET □ □□□□□□□ □□ □ □□□□ □□□□. □□ □□ □ □□ HTML □□□ □□□ □□□□□.
□□ □□ □□□ □□□□ □□□□ □□□□□ □□□□ □□□.
* □□ ASP.NET □ □□ □□□□□□ □□ □□□ □□□□□.
* □□ □□□ □□ □□□□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□□□ □□□ □□.
* □□ □□□ □□ □□ HTTP □□□ □□□□□.
□□□ □□□□ □□□.
□□ □□: ARR(□□ □□□□ □□ □□□)□ □□□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Instead deploy and configure Azure Cache for Redis. Update the web applications.
Reference:
https://docs.microsoft.com/en-us/azure/architecture/best-practices/caching#managing-concurrency-in-a-cache

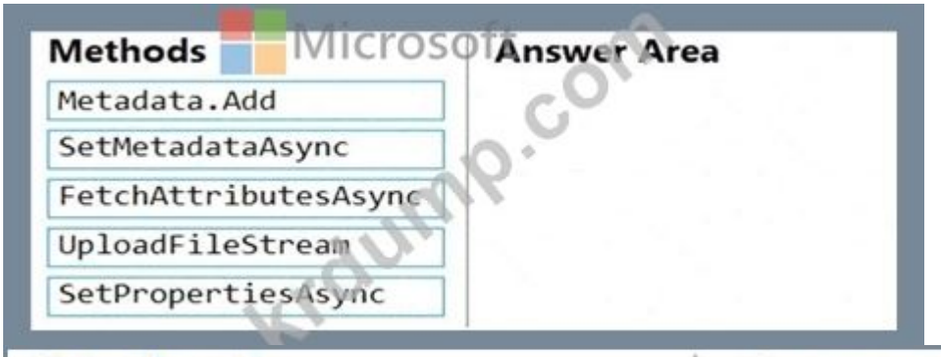
NEW QUESTION: 51
□□□□ □□□□□□ □□□ □□ □□ □ □□□□ □□ □□□□ □□□□ □□□.
□□ □ □□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. ID □□ □□
B. ID □□ □□□
C. HTTP □□ □□
D. Azure AD □□ URI
E. Azure AD □□□ ID
Answer: (SHOW ANSWER)
Claims in access tokens
JWTs (JSON Web Tokens) are split into three pieces:
* Header - Provides information about how to validate the token including information about the type of token and how it was signed.
* Payload - Contains all of the important data about the user or app that is attempting to call your service.
* Signature - Is the raw material used to validate the token.
Your client can get an access token from either the v1.0 endpoint or the v2.0 endpoint using a variety of protocols.
Scenario: User authentication (see step 5 below)
The following steps detail the user authentication process:

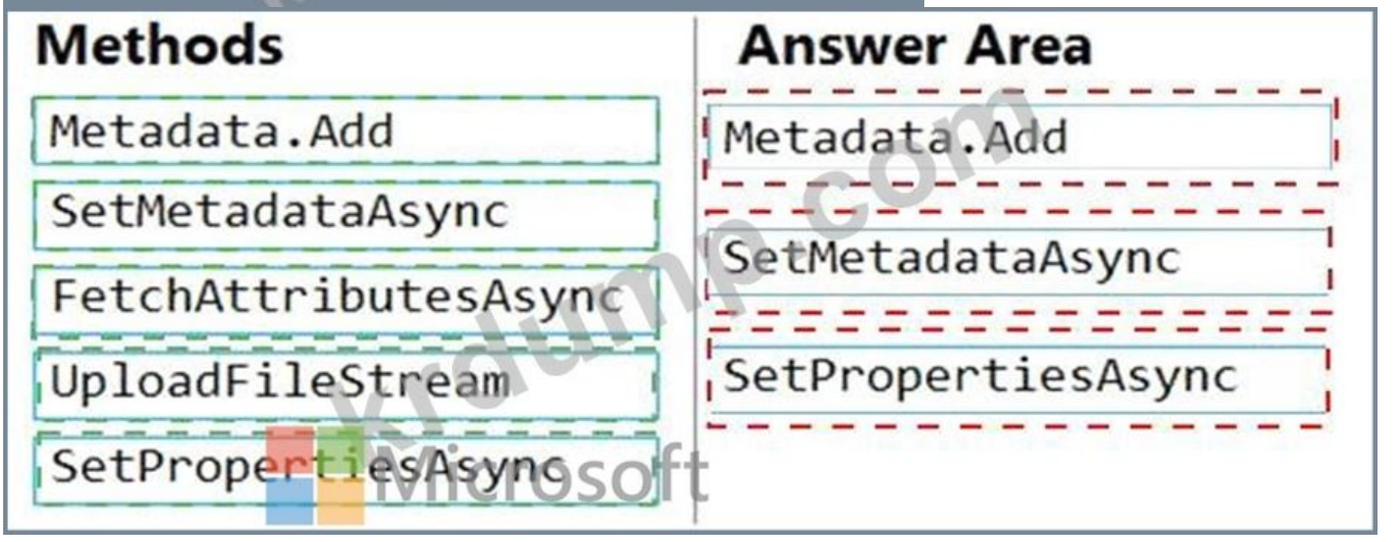
* The user selects Sign in in the website.
* The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
* The user signs in.
* Azure AD redirects the user's session back to the web application. The URL includes an access token.
* The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
* The back-end API validates the access token.
Reference:
https://docs.microsoft.com/en-us/azure/api-management/api-management-access-restriction-policies

NEW QUESTION: 52
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
Margie's Travel□ □□ □□ □ □□ □□ □□□□□□. □□□ □□□□ □□□□ □□□□ □□ □□. □□□□ □□□ □□□□□ □□ Azure Search□ □□□□ □□□.
Azure Search□□ □□□□ □□□□.
Azure Search .NET SDK□ □□□□ □□□□ □□□□ Azure Search □□□□ □□□□ □□□.
□□□:
1. □□ □□□ □□□ SearchServiceClient □□□ □□□□.
2. □□□□ □□ □□□ □□□ DataContainer□ □□□□□.
3. DataSource □□□□□ □□□ Container □□□ DataContainer□ □□□□□.
4. SearchServiceClient□ DataSources □□□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Use the following method:
1. - Create a SearchIndexClient object to connect to the search index
2. - Create an IndexBatch that contains the documents which must be added.
3. - Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.
References:
https://docs.microsoft.com/en-us/azure/search/search-howto-dotnet-sdk

NEW QUESTION: 53
Azure Blob □□□□□ □□□□ □□□□□□□ □□□□.
Blob□ □□□□□□ □□□□□□ □□□.

□□□ □□□ □□□□ □□ □ □□ □□□ □□□□□? □□□□□ □□ □□□□ □□□ □□ □ □□ □□□□ □□□□ □□□ □□□ □□□□□□.





Explanation:

Metadata.Add example:

// Add metadata to the dictionary by calling the Add method

metadata.Add("docType", "textDocuments");

SetMetadataAsync example:

// Set the blob's metadata.

await blob.SetMetadataAsync(metadata);

// Set the blob's properties.

await blob.SetPropertiesAsync();

Reference:

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-properties-metadata

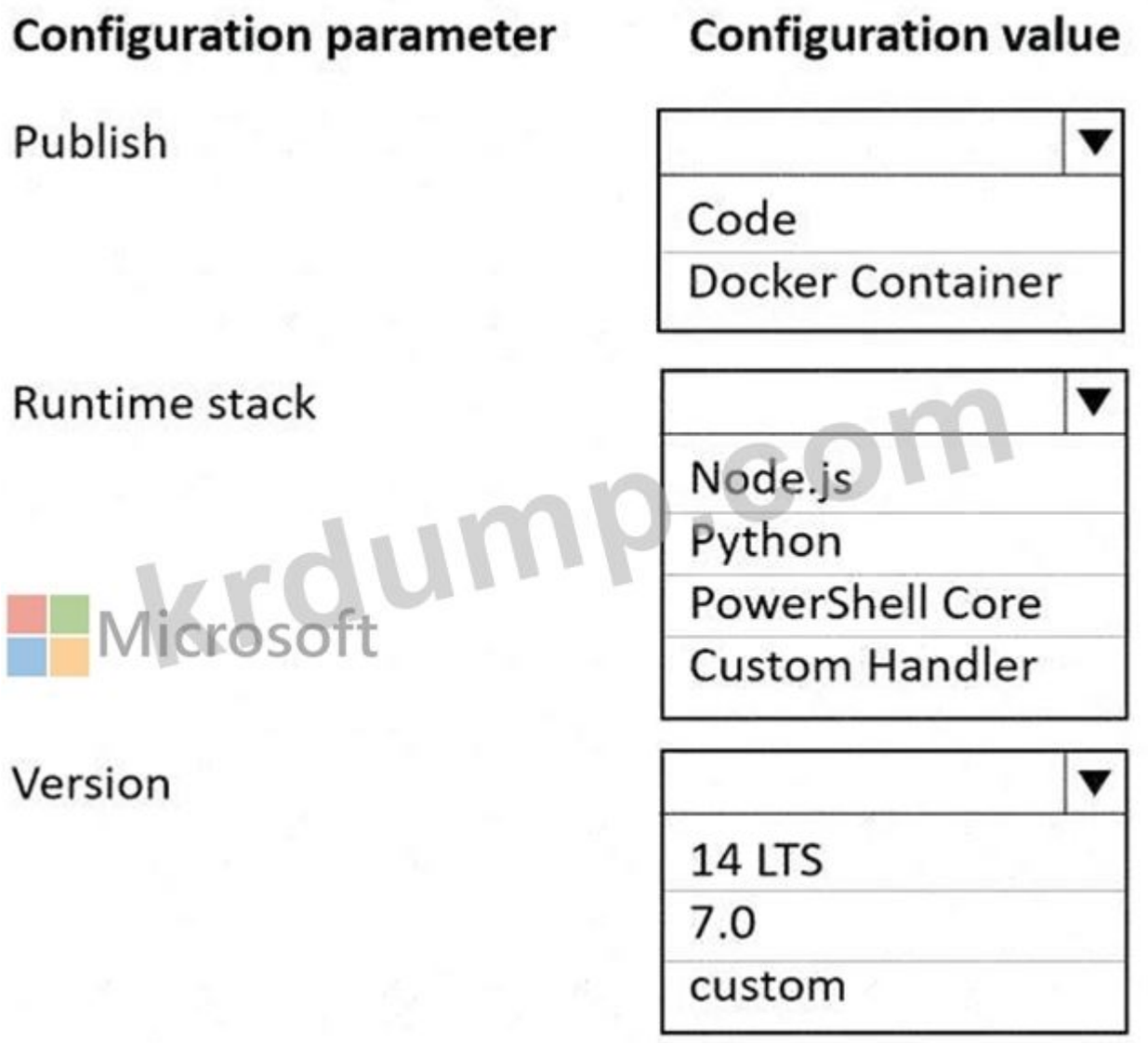
NEW QUESTION: 54

Azure □□ □□ □□□□ □□□□. Azure □□ □ □□□□□ □□□□ □□ □□□ □□□□ □ □□ □□□□□. □□ □□□ HTTP □□□□□□ □□□□□.

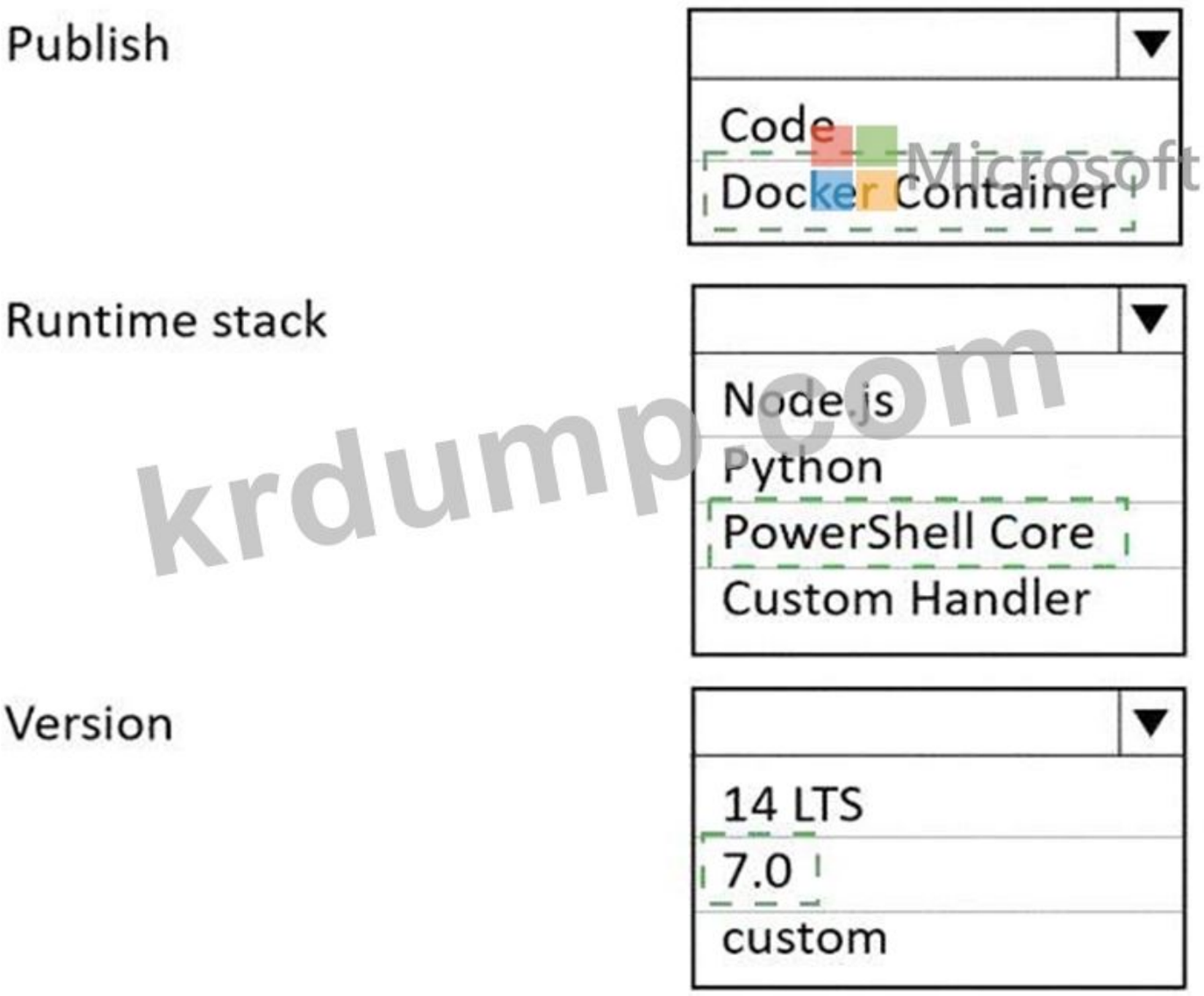
□□□□ Azure □□ □ □□□ □□□ □□□□ □□□.

□□□ □□ □□□□ □□□.

□□ □□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

Box 1: Docker container

A custom handler can be deployed to every Azure Functions hosting option. If your handler requires operating system or platform dependencies (such as a language runtime), you may need to use a custom container. You can create and deploy your code to Azure Functions as a custom Docker container.

Box 2: PowerShell core

When creating a function app in Azure for custom handlers, we recommend you select .NET Core as the stack. A "Custom" stack for custom handlers will be added in the future.

PowerShell Core (PSC) is based on the new .NET Core runtime.

Box 3: 7.0

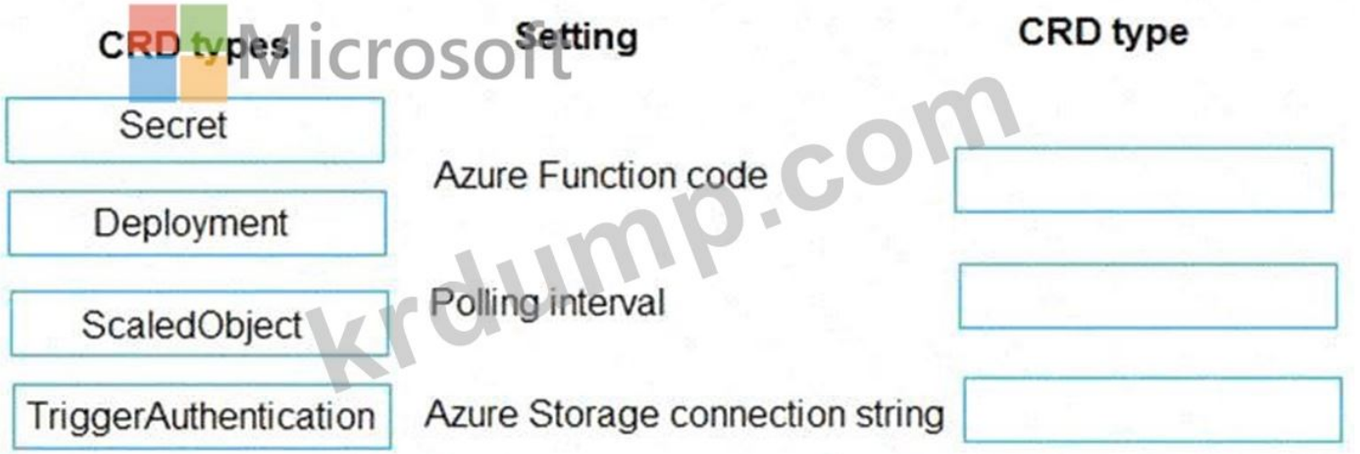
On Windows: The Azure Az PowerShell module is also supported for use with PowerShell 5.1 on Windows.

On Linux: PowerShell 7.0.6 LTS, PowerShell 7.1.3, or higher is the recommended version of

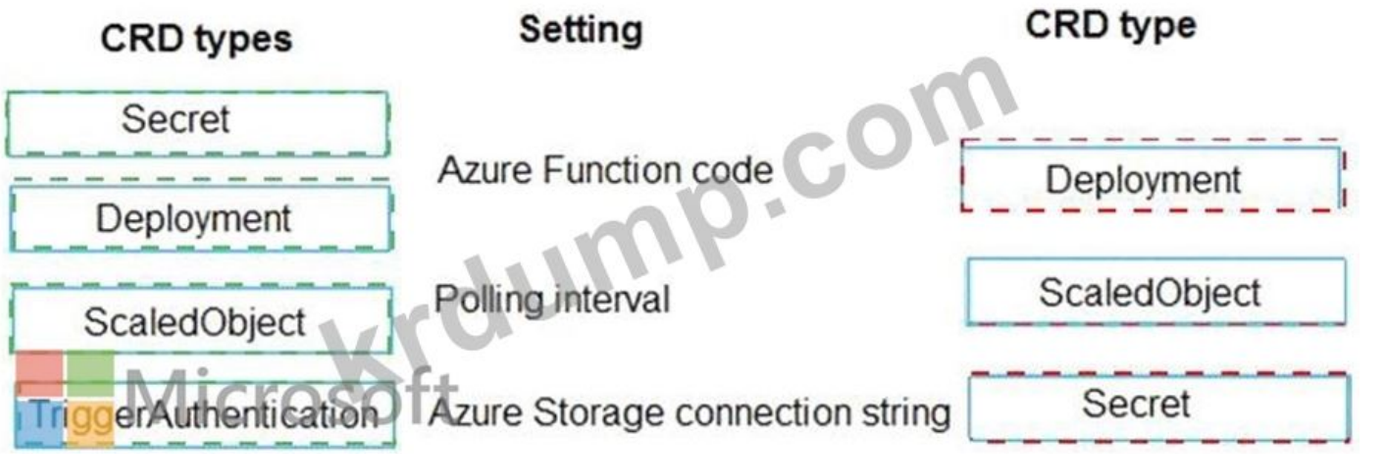
PowerShell for use with the Azure Az PowerShell module on all platforms.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-function-linux-custom-image
https://docs.microsoft.com/en-us/powershell/azure/install-az-ps?view=azps-7.1.0

NEW QUESTION: 55
□□□ □□□ □□ □□□ □□□ □□□□ □□□ □□□ □□□ □□□□□□ □□□□ □ □ □□□□□□ □□□□ □□□□. □□□ □□□ □□□ □□□ □□□ □□□□ □□ □□ □□ □ □□□. Azure Redo Cache□ □□□□ □□ □□□□ □□ □□□ □□□ □□□□.
Azure □□□ □□□ □□□□ □□ □□□□□ □□□ □□□□□ □□□.
Azure Redis □□ □□□□□ □□□□ □□□.
□□ □ □□ □□□ □□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□□□: □□ 1□□ 1□□□□.
A. □□ □□□ □□□□□ □□□□□.
B. □□□□ □□ □ □□ □□□ □□□ □□□□□.
C. RDS □□□□ □□□□ Azure Redis □□
D. □□□□ □□□□ Azure Redis □□
Answer: A,D (LEAVE A REPLY)

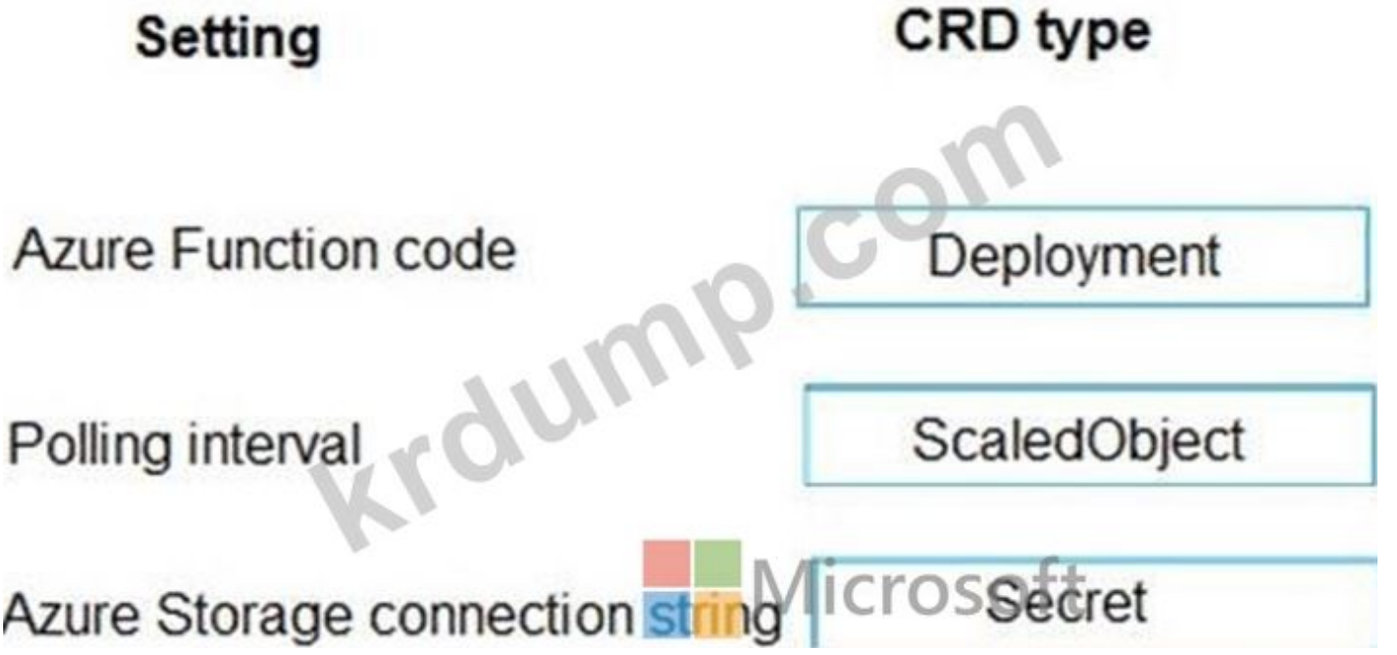
NEW QUESTION: 56
□□□ Azure Function□ □□□□ □□□ □□□□ SaaS(Software as a Service) □□□ □□□ □□□.
Azure Function□ □□ Azure Storage □□ □□ □□□□□ Azure Function □□□ □□□□□.
Kubernetes □□ KEDA(Event Driven Autoscaling)□ □□□□ Azure Function□ Kubernetes□ □ □□□□□□ □□□ □□ □□□□.
Azure □□□ □□ Kubernetes □□□ □□ □□□ □□(CRD)□ □□□□ □□□.
□□ CRD□ □□□□ □□□? □□□□□ □□□ CRD □□□ □□□ □□□ □□□□□□□. □ CRD □□□ □ □, □□ □ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□ □ □□ □□□ □□□ □□□□□ □ □ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Deployment

To deploy Azure Functions to Kubernetes use the func kubernetes deploy command has several attributes that directly control how our app scales, once it is deployed to Kubernetes.

Box 2: ScaledObject

With --polling-interval, we can control the interval used by KEDA to check Azure Service Bus Queue for messages.

Example of ScaledObject with polling interval

apiVersion: keda.k8s.io/v1alpha1

kind: ScaledObject

metadata:

name: transformer-fn

namespace: tt

labels:

deploymentName: transformer-fn

spec:

scaleTargetRef:
deploymentName: transformer-fn
pollingInterval: 5
minReplicaCount: 0
maxReplicaCount: 100
Box 3: Secret
Store connection strings in Kubernetes Secrets.
Example: to create the Secret in our demo Namespace:

```
# create the k8s demo namespace
kubectl create namespace tt
# grab connection string from Azure Service Bus
KEDA_SCALER_CONNECTION_STRING=$(az servicebus queue authorization-rule keys list \
-g $RG_NAME \
--namespace-name $SBN_NAME \
--queue-name inbound \
-n keda-scaler \
--query "primaryConnectionString" \
-o tsv)
# create the kubernetes secret
kubectl create secret generic tt-keda-auth \
--from-literal KedaScaler=$KEDA_SCALER_CONNECTION_STRING \
--namespace tt
```

Reference:
https://www.thinktecture.com/en/kubernetes/serverless-workloads-with-keda/

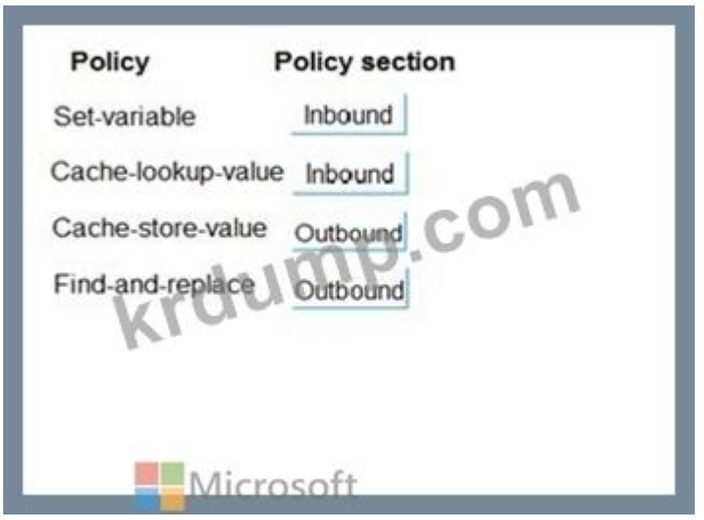
NEW QUESTION: 57
Azure Batch □□□ □□□ □□□□ □□□ □□□□ □□□□.
□□□ □□ □□ □□□□□□.
□□ □□□□ □□ □□□ □□□□□ □□□ □□□□ □□ □□□ □□□□ □□□.
□□ □□□ □□□□ □□□?
A. JobOperations.CreateJobO
B. CloudJob.Enable(IEnumerable<BatchClientBehavior>)
C. CloudJob.CommitAsync(IEnumerable<BatchClientBehavior>, CancellationToken)
D. JobOperations.EnableJob(String, IEnumerable<BatchClientBehavior>)
E.
JobOperations.EnableJobAsync(String.IEnumerable<BatchClientBehavior>.CancellationToken)
Answer: C (LEAVE A REPLY)
A Batch job is a logical grouping of one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. The app uses the BatchClient.JobOperations.CreateJob method to create a job on your pool.
The Commit method submits the job to the Batch service. Initially the job has no tasks.

{
CloudJob job = batchClient.JobOperations.CreateJob();
job.Id = JobId;
job.PoolInformation = new PoolInformation { PoolId = PoolId };
job.Commit();
}
References:
https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet

NEW QUESTION: 58
Contoso, Ltd.□ APIM(Azure API Management)□ □□□□ □□□□ API□ □□□□□. API□ JWT □□□□ □□□□□ □□□ □□□□□.
APIM □□□□□□ □□ □□ □□□ □□□□ □□□. □□ □□□□□ □□□ □□□ □□□□ □□□□□ □□□□□□ □□□ ID□ □□□□ □□ □□□ ID□ □□ □□□ □□□□ □□□. □□ □□□ □□ □□□ □□□□ □□□.
* □□□ □□□ □□□ □□□□ □□ set-variable □□
* □□ □□ □ □□
* □□ □□ □□ □□
* □□□ □□□ □□□ □□ □□□ □□□□□□ □□ □ □□□ □□ □□□ □□□□ □□ □ □ □□□ □□□□□? □□□□□ □□□ □□□ □□□ □□□□ □□□□□□□. □ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □ □□□ □□ □□□ □□□ □□□ □ □□□ □□□□□ □ □ □□□□. □□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:

Explanation:



Box 1: Inbound.

A set-variable policy to store the detected user identity.

Example:

<policies>

<inbound>

<!-- How you determine user identity is application dependent -->

<set-variable

name="enduserid"

value="@(context.Request.Headers.GetValueOrDefault("Authorization","").Split(' ')

[1].AsJwt()?.Subject)" /> Box 2: Inbound A cache-lookup-value policy Example:

<inbound>

<base />

<cache-lookup vary-by-developer="true | false" vary-by-developer-groups="true | false"

downstream-caching- type="none | private | public" must-revalidate="true | false">

<vary-by-query-parameter>parameter name</vary-by-query-parameter> <!-- optional, can

repeated several times -->

</cache-lookup>

</inbound>

Box 3: Outbound

A cache-store-value policy.

Example:

<outbound>

<base />

<cache-store duration="3600" />

</outbound>

Box 4: Outbound

A find-and-replace policy to update the response body with the user profile information.

Example:

<outbound>

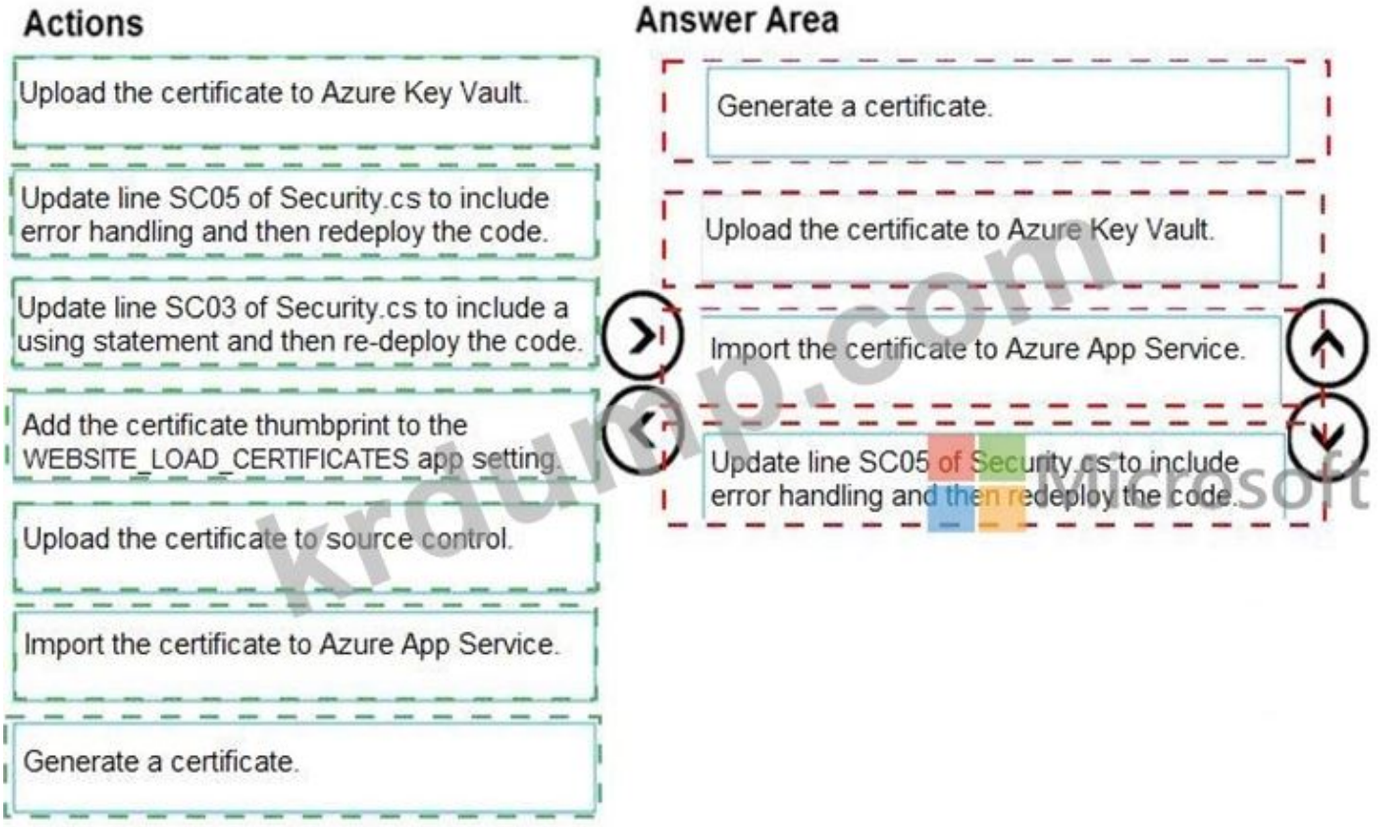
<!-- Update response body with user profile-->

<find-and-replace
from="'$userprofile$'"
to="@((string)context.Variables["userprofile"])" />
<base />
</outbound>
Reference:
https://docs.microsoft.com/en-us/azure/api-management/api-management-caching-policies
https://docs.microsoft.com/en-us/azure/api-management/api-management-sample-cache-by-key

NEW QUESTION: 59
Azure App Services □□□ □□□□. Azure SQL □□□□□□ □□□□. □□□ □□□ Azure Storage □□ □ Azure Redis Cache □□□□.
□□□□ □ □□ □□□ □□□ □ □□□ □□□. □□□□□ □ □□ □□ Contribute □□□ □ □□□ □□□. □□□ □□□□ □□□.
□□ □ □□ □□□ □□□ □ □□□□? □ □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. New-AzureRmRoleAssignment
B. az □□ □□ □□□
C. az □□ □□ □□
D. □□□ AzureRmRoleDefinition
Answer: A,B (LEAVE A REPLY)
References:
https://docs.microsoft.com/en-us/cli/azure/role/assignment?view=azure-cli-latest#az-role-assignment-create
https://docs.microsoft.com/en-us/powershell/module/azurerm.resources/new-azurermroleassignment?
view=azurermps-6.13.0

NEW QUESTION: 60
□□ □□□□ □□ □□□ □□□□□.
□□ 4□□ □□□ □□□□ □□□□□ □□□□ □□□? □□□□□ □□ □□□□ □□ □□□ □□ □□□□ □□□□ □□□ □□□ □□□□□□.

Answer:



Explanation:

Scenario: Corporate website
While testing the site, the following error message displays:
CryptographicException: The system cannot find the file specified.
Step 1: Generate a certificate
Step 2: Upload the certificate to Azure Key Vault
Scenario: All SSL certificates and credentials must be stored in Azure Key Vault.
Step 3: Import the certificate to Azure App Service
Step 4: Update line SCO5 of Security.cs to include error handling and then redeploy the code
Reference:
https://docs.microsoft.com/en-us/azure/app-service/configure-ssl-certificate

NEW QUESTION: 61
APSPlanl□□□ Azure App Service □□□□ □□ V2 □□ □□□□ □□□□□□. APSPlanl□ □ WebAppl□□□ App Service □□□ □□□□ □□□□. APSPlanl□ □□ □□ □□ □□ □□ □ □□□□ □□□□□. WebAppl □□ □□□ □□□□□ □□□. □□□: APSPlanl□ □□□□ V2 □□ □□□□ □□□□□.
□□□□ □□□ □□□□□?
A. □□□
B. □
Answer: (SHOW ANSWER)

NEW QUESTION: 62
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□□ □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □ □ □□ □ □□□ □□ □□, □□ □□□□ □□□ □□ □□□ □□□□.

□ □□□ □□□ □□□ □□□ □□ □□□ □ □□□□. □□□ □□□ □□□ □□ □□□ □□ □□ □□□□.
Azure Container Apps □□ □□□□ □□□□ □□□ □□□ □□□□□□□.
□□□□□ □□□□ □□ □□□□ □ □□□ □□□□□. □□□ □□ □□ 0□□ □□□□□ □ □□ □□ □□□□□□.
□□□□ □□ □□□ □□□□ □□□.
□□ □□: □□□ □□□□□ □□□□ □□ □□ □□□ □□ 1□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)

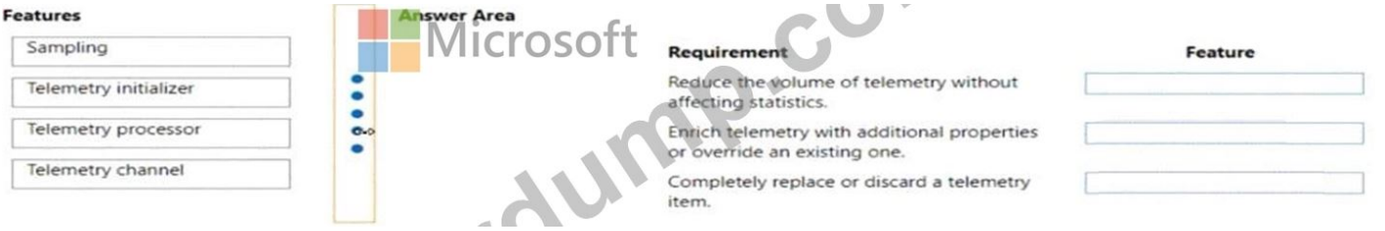
NEW QUESTION: 63
□□ □□ □□□□ □□ □□□□□ □□□□ □□□□ □□□□. □ □□□□ □□□ □□□ □ □□ Azure Batch □□□ □□□□ □□ □□□ □□ □□□□□ □□□ □□□□□□□□.
Azure Batch□□ □□□□ □□ □□□ □□□ □□□□ □□□.
□□□ □□ □□□?
A. Python□□ □□□ □□: TaskAddParameter
B. Python□□ □□□ □□: JobAddParameter
C. Azure Portal□□ Batch □□□ □□□□.
D. .NET □□□□□ BatchClient.PoolOperations.CreateJob □□□□ □□□□□.
Answer: (SHOW ANSWER)
A Batch job is a logical grouping of one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. The app uses the BatchClient.JobOperations.CreateJob method to create a job on your pool.
Note:
Step 1: Create a pool of compute nodes. When you create a pool, you specify the number of compute nodes for the pool, their size, and the operating system. When each task in your job runs, it's assigned to execute on one of the nodes in your pool.
Step 2 : Create a job. A job manages a collection of tasks. You associate each job to a specific pool where that job's tasks will run.
Step 3: Add tasks to the job. Each task runs the application or script that you uploaded to process the data files it downloads from your Storage account. As each task completes, it can upload its output to Azure Storage.
References:
https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet

NEW QUESTION: 64
Java □□□□□□□ □□□□ Azure□ □□□□□. □□□□□□□ Application Insights SDK□ □□□□ □□□□□□□.
Application Insights □□□□ □□□ □□ □□ □□ □□□□ □□□□ □□□□ □□□.

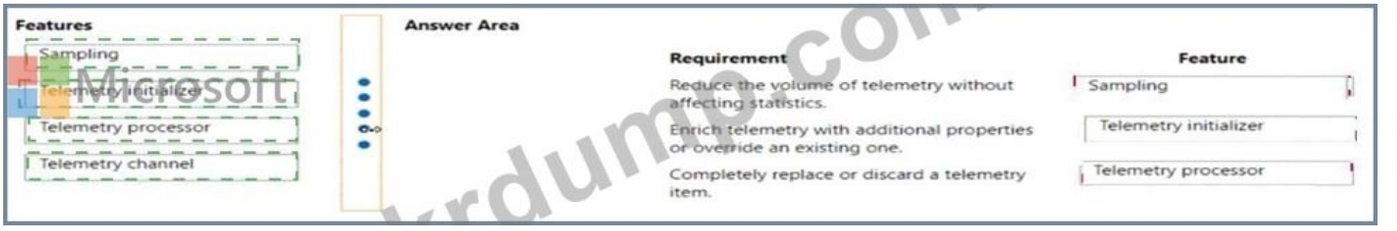
□□ □□ □□□□ □□□□ □□□.

□□ Application Insights SDK □□□ □□□□ □□□? □□□□□ □□□ □□□ □□□ □□ □□□□ □□□□□□□□. □ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□ □. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

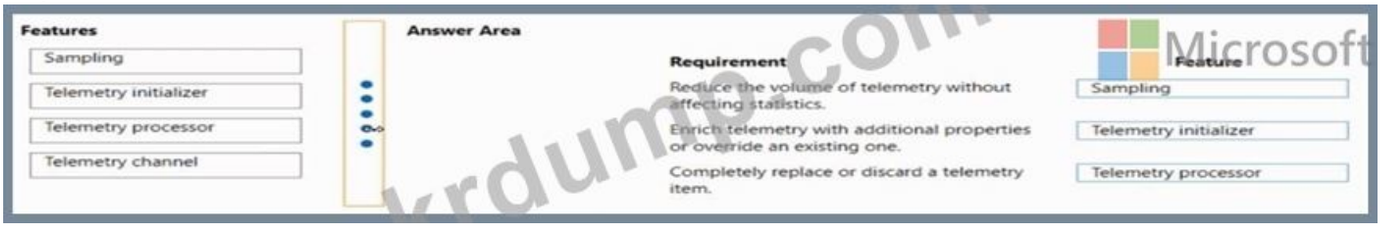
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

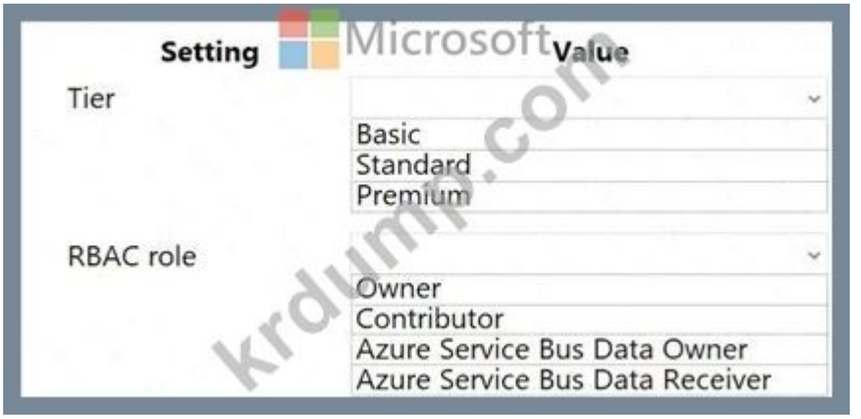


NEW QUESTION: 65

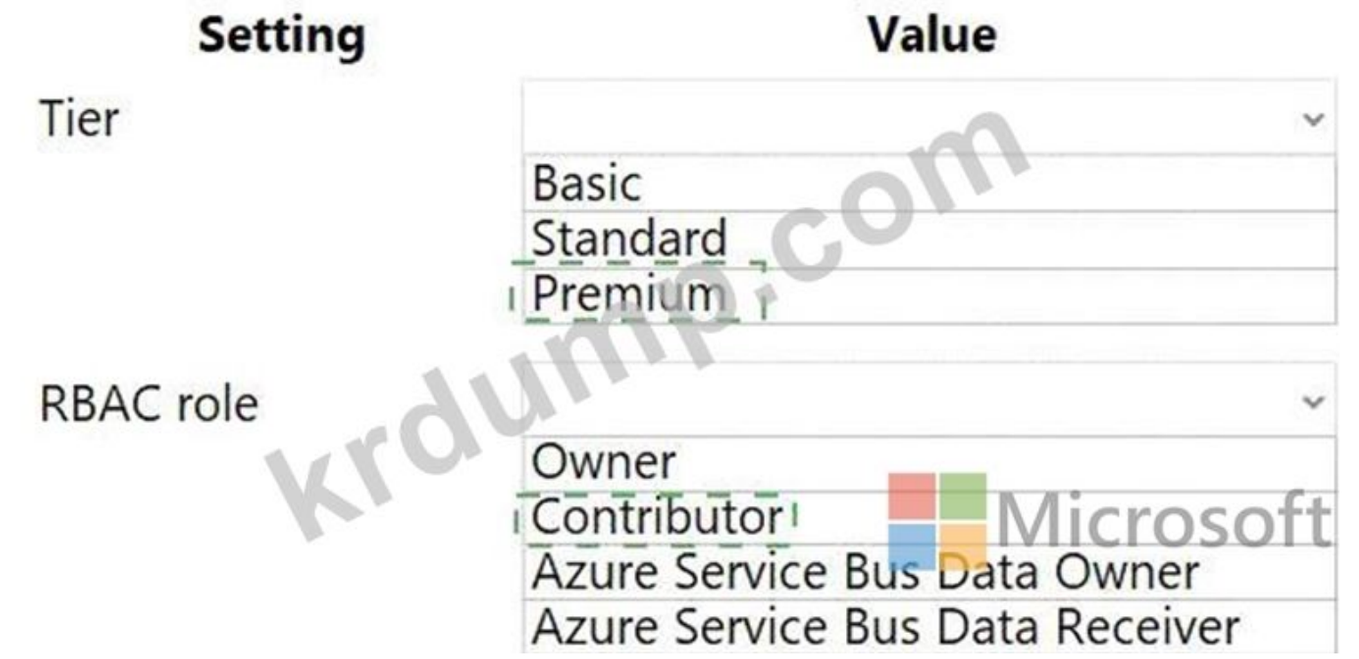
Event Grid □□□ Azure Service Bus□ □□□□ □□□.

□□ Azure Service Bus □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□ □□□□.

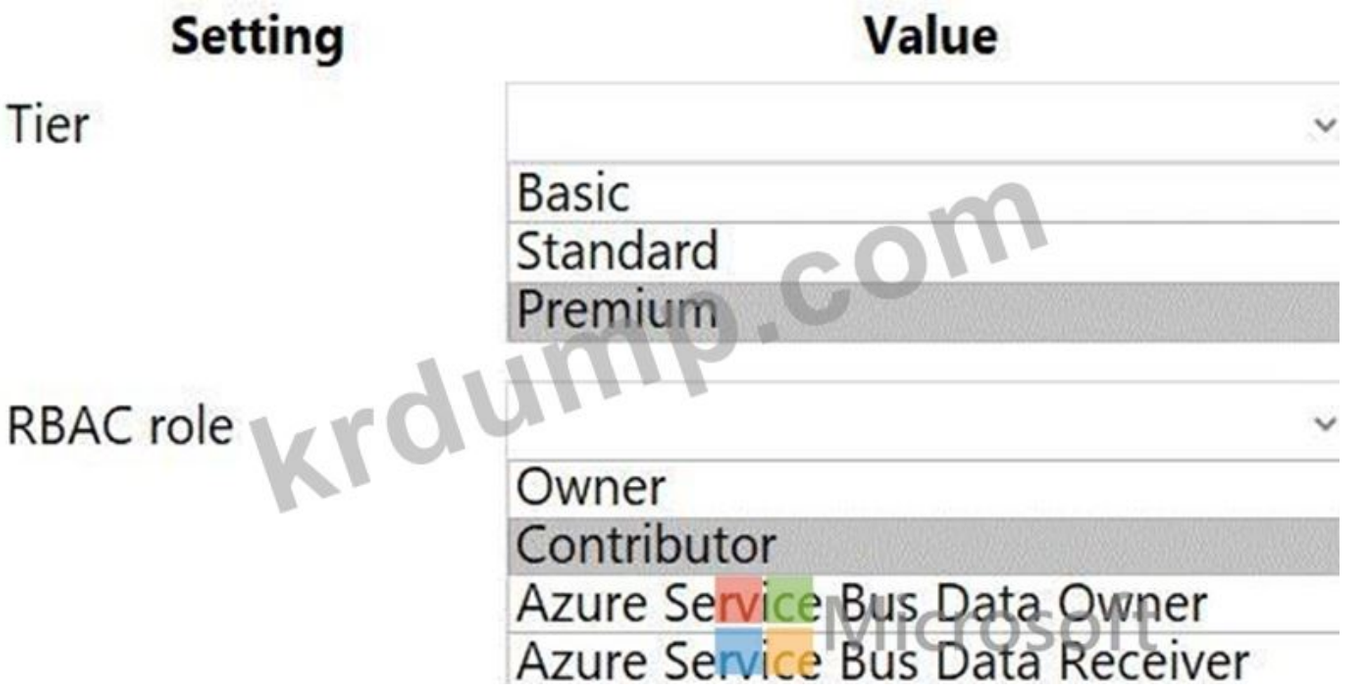
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Premium

Service Bus can now emit events to Event Grid when there are messages in a queue or a subscription when no receivers are present. You can create Event Grid subscriptions to your Service Bus namespaces, listen to these events, and then react to the events by starting a receiver. With this feature, you can use Service Bus in reactive programming models.

To enable the feature, you need the following items:

A Service Bus Premium namespace with at least one Service Bus queue or a Service Bus topic with at least one subscription.

Contributor access to the Service Bus namespace.

Box 2: Contributor

Reference:

https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-to-event-grid-integration-concept

NEW QUESTION: 66
APSPlan1□□□ Azure App Service □□□□ □□ B1 □□ □□□□ □□□□□□. APSPlan1□ □ App Service □□ □□□□ □□□□. APSPlan1□ □□ □□ □□ □□ □□□ □□□□ □□ □□□. □ □ □□ □□□ □□□□□ □□□.
□□ □□: ASPPlan1□ □□ □□ □□□□ □□□□□.
□□□□ □□□ □□□□□?
A. □□□
B. □
Answer: A (LEAVE A REPLY)

NEW QUESTION: 67
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
□ □□ 2,000□ □□□□ POS(Point-of-Sale) □□ □□□□ □□□□ Azure □□□□ □□□□ □□□□. □□ □□□ 24□□□□ 2MB□ □□□□ □□□ □ □□□□. □ □□ □□□□ □□□ □ □□□□ 1~5□□ □□□ □□□□.
Azure Blob Storage□ □□□□ □□□□ □□□□ □□□. □□ □□□□ □□ □□□□ □□□ □ □□ □□□□□ □□□. □□ □□ □□ □□ □□□□□.
□□ □□□□ □□□□ □□ □□□□ □□□□ □□□.
□□ □□: Azure □□ □□□ □□□□□□□□. □□□ □□ □□□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: (SHOW ANSWER)
Instead use an Azure Service Bus, which is used order processing and financial transactions.
Reference:
https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services

NEW QUESTION: 68
□□ □□□ □□□ □□□□ account1□□□ NoSQL□ Azure Cosmos DB □□□ □□□□□□ □□.
□□□□ □□□ □□□ □□□ □□□□□. account1□ □□ □□ □ □□ □□□ □□ □□□ □ □ □□□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□ □□ □□□ □□□ □□□□ □□□.
□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□□ □□□ □□ 1□□ □□□ □□□□.

Answer Area

Operation type	Resulting consistency level
Read operations	strong session consistent prefix
Write operations	strong session consistent prefix

Answer:

Answer Area

Operation type	Resulting consistency level
Read operations	strong session consistent prefix
Write operations	strong session consistent prefix

Explanation:

Consistent Prefix

Strong

NEW QUESTION: 69

Azure Cosmos DB SQL API□ □□□□ Azure Cosmos DB □□□□ □□□□ □□□□. □□□ □□ □□□ □□ □□□ □□□□□. □ □□□□ □□ □□ □□□ □□□ □ □□□□.
□□□ □□□□ □□□ □□ □□□ □□ □□□□ □□ □□□□. Azure Cosmos DB□ □□ □ □□ □□ □□□□□ □□ □□□□ □□□ □□□□ □□□□□□□ □□ □□ □□□ □□□□ □ □□□□□□□ □□ □□□□□ □□□□ □□□.
□□□ □□ □□□□ □□□.
□□□ □ □□ □ □□ □□□ □□ □□□□□? □ □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□□ □□□□ □□□ □□ □□ □□ □□
B. □□□ □□ □□□□ □□ □□ □□□
C. □□ □□ □□□ □□ □□□
D. □□□ □□□ □□□□ □
E. □□□ □□ □□□□ □□ □□ □
Answer: A,C (LEAVE A REPLY)
You can form a partition key by concatenating multiple property values into a single artificial partitionKey property. These keys are referred to as synthetic keys.
Another possible strategy to distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.
Note: It's the best practice to have a partition key with many distinct values, such as hundreds or thousands.
The goal is to distribute your data and workload evenly across the items associated with these partition key values. If such a property doesn't exist in your data, you can construct a synthetic partition key.
References:
https://docs.microsoft.com/en-us/azure/cosmos-db/synthetic-partition-keys

NEW QUESTION: 70
Azure □□□ □□ Azure □□ □□ □□□□ □□□□□□□ □□□□. □□ □□□ □ □□□□ □□□ □□□□□□ □□□ Azure Key Vault□ □□□□□.
□□□ □□□□□□ □□ □□□□□□ □□□ □□□ □□□□ □ □□□. Azure AD(Azure Active Directory)□ □□ □□□ □□□□□ □□□.
□□□□□□ □□□ □□□□ □□□ □□□□ □□□.
□□□ □□ □□□?
A. Key Vault□ □□□□ □ □□ □□□ □□ □□ □□□ □□ □□ ID□ □□□ □□ □□ ID□ □□□□□ □ App Service□ □□□□□.
B. Key Vault□ □□□□ □ □□ □□□ □□ □□ Azure AD □□□ □□□ □□□ App Services □□□ □□□□□ □□□ □□□□ Key Vault□ □□□□□□.
C. Key Vault□ □□□□ □ □□ □□□ □□ □ App Service□□ □□□ □□ □□ ID□ □□□ □.
D. □ App Service□ □□ Key Vault□ □□□□ □ □□ □□□ □□ Azure AD □□□ □□□ □

□□ App Services □□□ □□□□ □□□□ Key Vault□ □□□□□□.

Answer: C (LEAVE A REPLY)

Use Key Vault references for App Service and Azure Functions.

Key Vault references currently only support system-assigned managed identities. User-assigned identities cannot be used.

Reference:

https://docs.microsoft.com/en-us/azure/app-service/app-service-key-vault-references

NEW QUESTION: 71

□□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□□.

□ □□□□□ Azure Table Storage□ □□□□ □□□ □□□ □□□□ □□□ □□ □□□□□.

□□ □□□ □□ □□□ □□□ □ □□ □□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
 CloudConfigurationManager.GetSetting("StorageConnectionString"));
CloudTableClient tableClient = storageAccount.CreateCloudTableClient()
CloudTable table = tableClient.GetTableReference("clients");
Table.CreateIfNotExists();
```

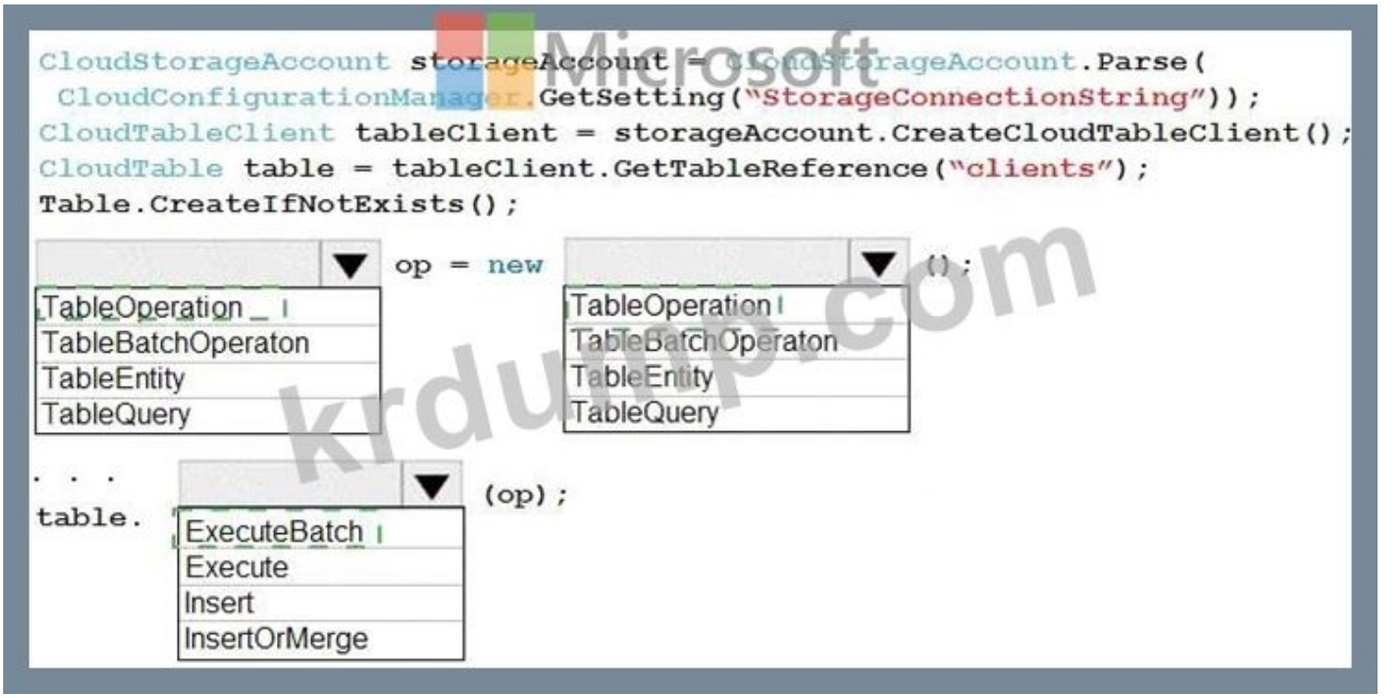
[▼] op = new [▼] ();

TableOperation	TableOperation
TableBatchOperaton	TableBatchOperaton
TableEntity	TableEntity
TableQuery	TableQuery

. .
table. [▼] (op);

ExecuteBatch
Execute
Insert
InsertOrMerge

Answer:



Explanation:

Answer Area

Box 1, Box 2: TableBatchOperation

Create the batch operation.

TableBatchOperation op = new TableBatchOperation();

Box 3: ExecuteBatch

/ Execute the batch operation.

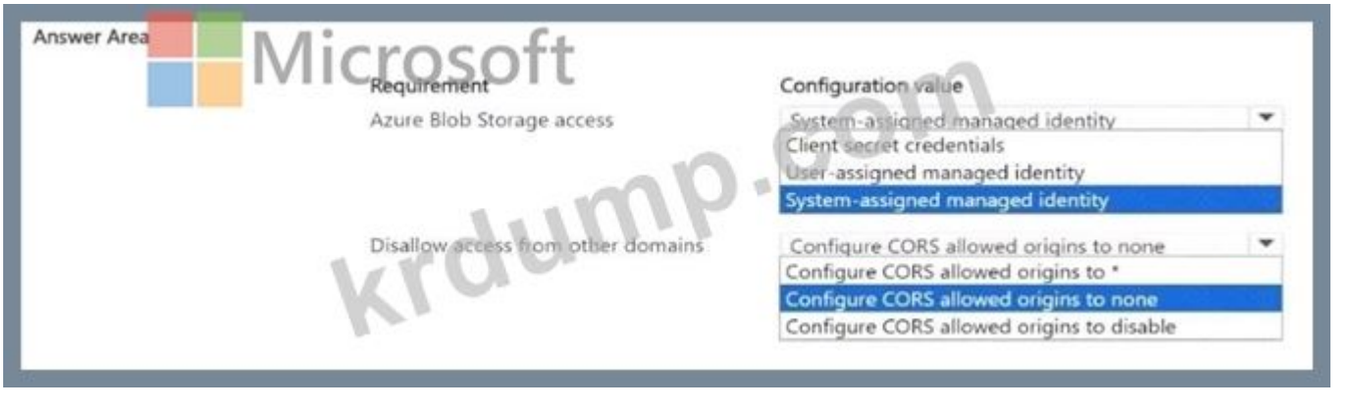
table.ExecuteBatch(op);

Note: You can insert a batch of entities into a table in one write operation. Some other notes on batch operations:

You can perform updates, deletes, and inserts in the same single batch operation.

A single batch operation can include up to 100 entities.
All entities in a single batch operation must have the same partition key.
While it is possible to perform a query as a batch operation, it must be the only operation in the batch.
References:
https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet

NEW QUESTION: 72
App1□□□ Azure Functions □□ □□□□ □□□□. □□ CORS(Cross-Origin Requests)□ □ □□ □□□□□.
□□□ □□ □□ □□□ □□□□□.
* App1 □□□ Azure Blob Storage □□□ □□□□ □□□□□ □□□.
* Azure Blob Storage □□□ □□□□□□ □□ □□□□□□□ □□□ □□□□ □□□□.
* □□ □□□□ □□□□□□ □□□□ JavaScript □□□ □□ □□□ □□ □□□ □ □□□ □ □□.
App1□ □□□□ □□□.
□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□□□: □□ □□□ 1□□□□.



Answer:

Explanation:

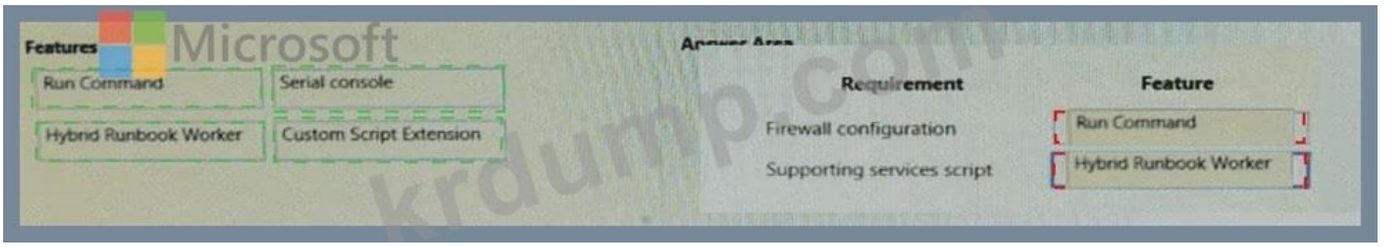
NEW QUESTION: 73

Azure VM(□□ □□) □□ □□□□□□ □□□ □□□□ □□□□. □□□□□□□ □□□□ VM□□ □□□ □□ □□ □□□ □□□□.

* VM□ □□□□□□□ □□□□ □□□□ □□□□□ Azure □□□□ □□□□ □ □□□□.
* □□ □□□□ Azure Storage□ □□□ Azure PowerShell □□□□□ □□□□ □□□□ □□ □. □□ □□□ □□□□□ □□□□ □□□.

□□ □□□ □□□□ □□□? □□□□□ □□□ □□□ □□□ □□ □□□□ □□□□□□□.

Answer:



Explanation:

Reference:

https://docs.microsoft.com/en-us/azure/automation/automation-hybrid-runbook-worker

https://docs.microsoft.com/en-us/azure/virtual-machines/windows/run-command

NEW QUESTION: 74

□□□□ □□□ □□□ □□□ □ □□□ □□□ □□ □□ □□□□□□. □ □□□ □□□ □□ □□□.

□□ □□□ □□□□ □□□□ □□□?

A. Azure □ □□□

B. Azure API □□

C. □□ □□

D. Azure □□

Answer: A (LEAVE A REPLY)

NEW QUESTION: 75

□□□□ Mob □□□ □□□ □□□□□. □□□□ □□□ □□□ □□ Blob □□□□ □□□ □ □ □□□□ □□□□.

□□ □□□ □□ Blob □□□ □□□ □□□□ □□□. Blob □□□ □□ □□□ □□□□ □ □□ □.

□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:

Explanation:

NEW QUESTION: 76

Azure Blob □□□□□ □□□ □□ □□□□ □□□□ □□□□□□□ □□□□ □□□□. □□ □□ companymedia□□ Azure Blob □□□□ □□□ □□□ companyimages□□ Azure Blob □ □□□ □□□□□ □□□□□□. □□□ □□□□ □□□□□ □□ Blob□ □□ □□ Blob □□ □ □□□ □□ □□□□□□.

□□□ □□□□□ □□ □□ □□□□ □□ Blob□ □□□ □□□. □□ □□□□ status□□ □ □□ □□□ final □□□ □□□□ □□□.

GET □□□ □□ URL□ □□□□ □□□.

URI□ □□□ □□□□ □□□? □□□□□ □□□ □□□□□ □□□ □□ URI □□□□ □□□ □□□□. □ □□□□□ □ □, □□ □ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □ □□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

Answer:

Explanation:

NEW QUESTION: 77

□□ □□□□□ □□ □□□ □□□□ □□□□ □□ Azure □□□ □□□ □□□□.

□□ □□□□ □□□ □□ □□□□ □ □□□□ □□□□ □□□□ □□□. □□ □□□□□ □ □ □□ □□□ □□□□ □□□.

* □□□ □□□ □□□□□□.
* □□□ □□□□ □ □□□ □□□ □□□ □□□□□□.
* □□□ □□ □□□□□ □□ □□□□ □□□□□□.

□□□ □□□□ □□□?

A. □□□□

B. Azure □□□ □□□

C. Azure □□

D. .NET □□□□ □□□□□ □□□□□

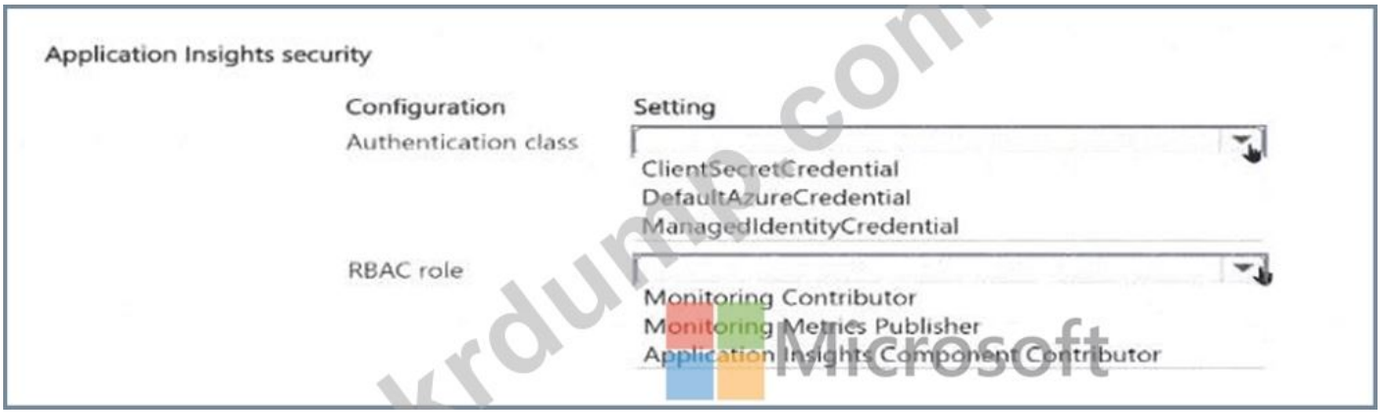
Answer: (SHOW ANSWER)

You can copy blobs, directories, and containers between storage accounts by using the AzCopy v10 command-line utility.

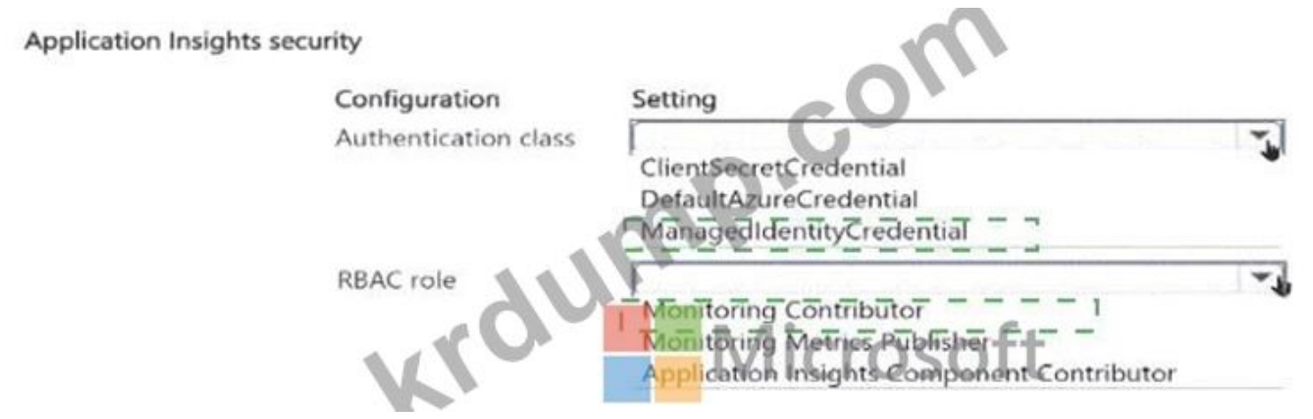
The copy operation is synchronous so when the command returns, that indicates that all files

have been copied.
Reference:
https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-blobs-copy

NEW QUESTION: 78
Workspace1□□□ Application Insights □□ □□□ □□□□□ Azure □□□ □□□□.
.Net □□ □□□ □□□□ App1□□□ App Service □□□ □□□ □□□□□. App1□□ □□□ □□ □□ □□□□ Workspace1□ □□□□□. Microsoft Entra ID □□ □ Azure □□ □□ □□ □ □□(RBAC)□ □□□□ □□ □□ □□ □□□□ Workspace1□ □□□□ □□ □□□ □□□ □□.
App1□ □□ Entra ID □□ □□ □□□□ RBAC □□□ □□□□ □□□.
□□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□: □□ □□□ 1□□□□.



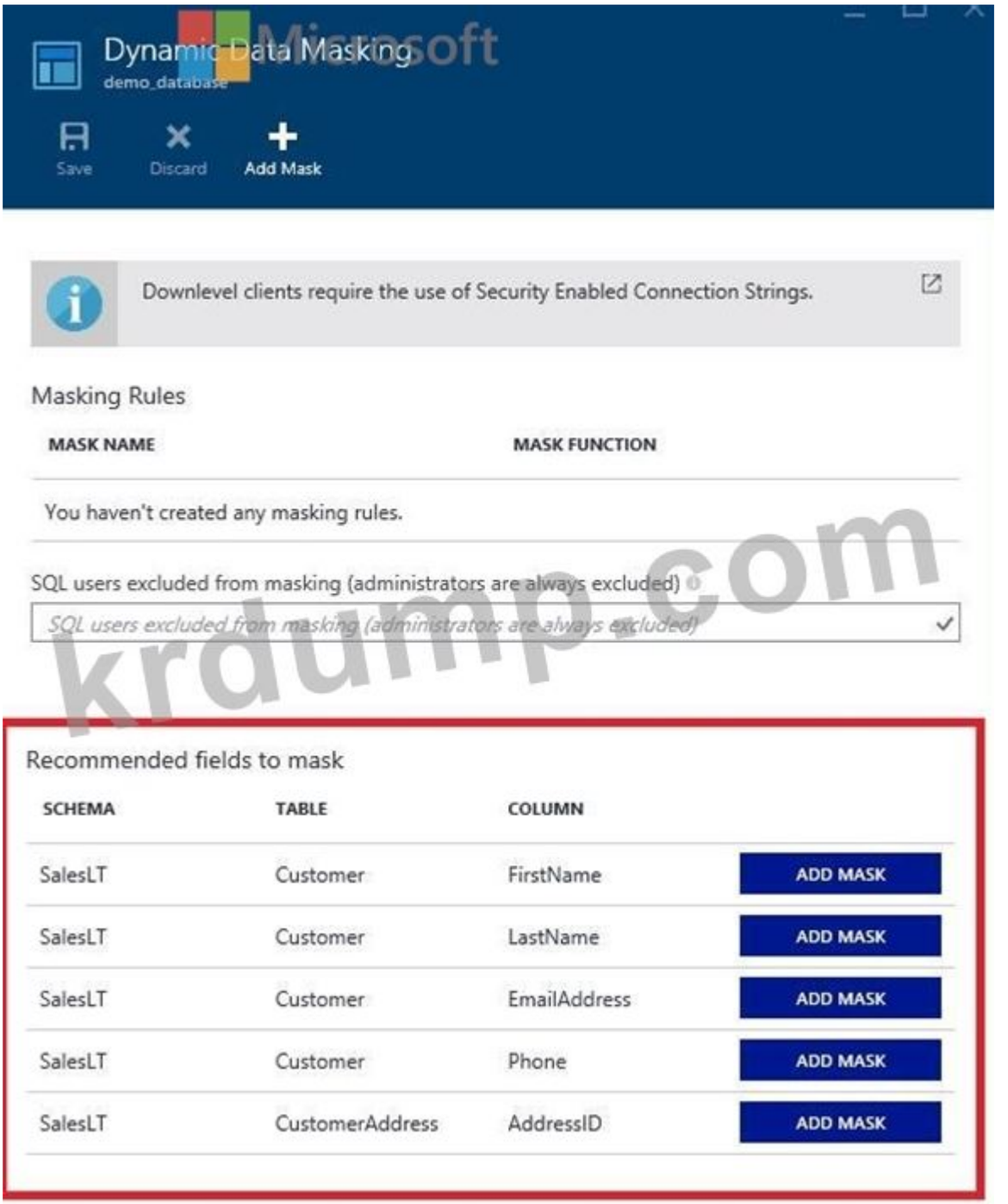
Answer:



Explanation:

NEW QUESTION: 79
Azure SQL □□□□□□□ □□□□ □□□ □□ □□ □□□ □□□ □□□□ □□□□ □□□ □□.
□□ □□□□ □□ □□□ □□□ □□□□□.

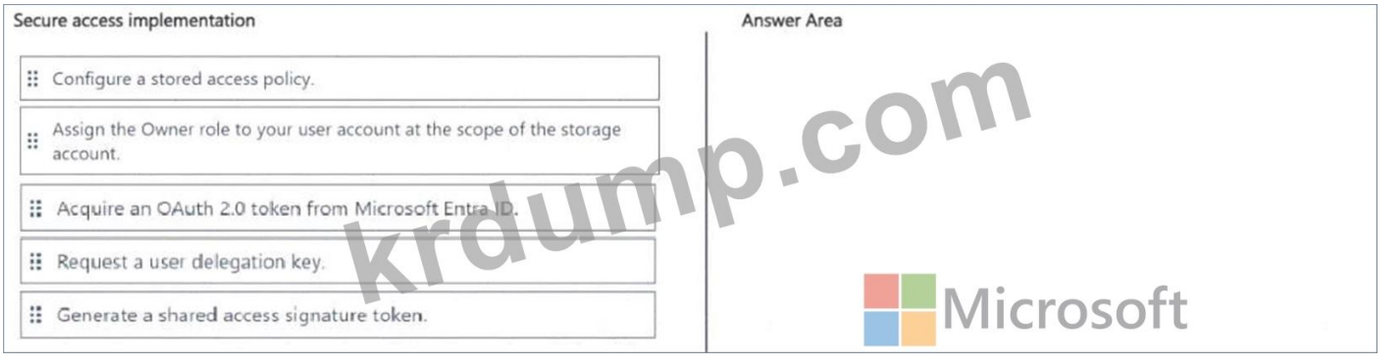
□□□ □□ □□□□ □□□□ □□□□□□ □□□ □□□ □□□ □□□.
□□ □□□ □□□□ □□□ □ □□□□ □□ □ □□ □□□ □□□□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□
B. □
C. □□□
D. □□□
E. □□□
Answer: A,B,E (LEAVE A REPLY)
In the Dynamic Data Masking configuration page, you may see some database columns that the recommendations engine has flagged for masking. In order to accept the recommendations, just click Add Mask for one or more columns and a mask is created based on the default type for this column. You can change the masking function by clicking on the masking rule and editing the masking field format to a different format of your choice.



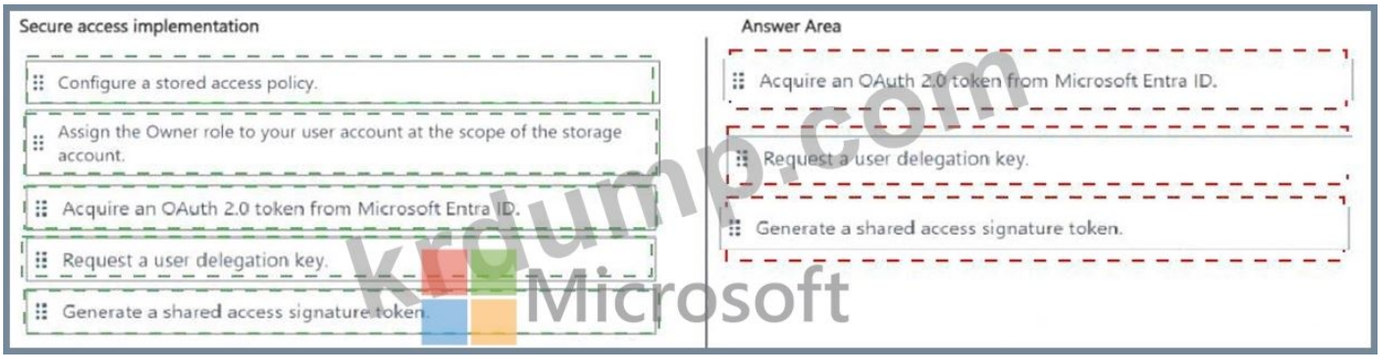
References:
https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dynamic-data-masking-get-started-portal

NEW QUESTION: 80
contoso.com□□□ Microsoft Entra □□□□ □□□ Azure □□□ □□□□□. □□□□ □□ □ □ □□ Contributor Azure □□ □□ □□□ □□(RBAC) □□□ □□ Azure BI □□□ □□□□ □.
□□□ 1□ □□□□□ BLOB□ □□ □□ □□□□ □□□ □□□□□. □□□□ □□ □□ □□ □ □□□□ □□□.
* □□□□ □□□□□□ □□ □□□ □□□□□. □□□□ Microsoft Entra □□ □□□ □□□□

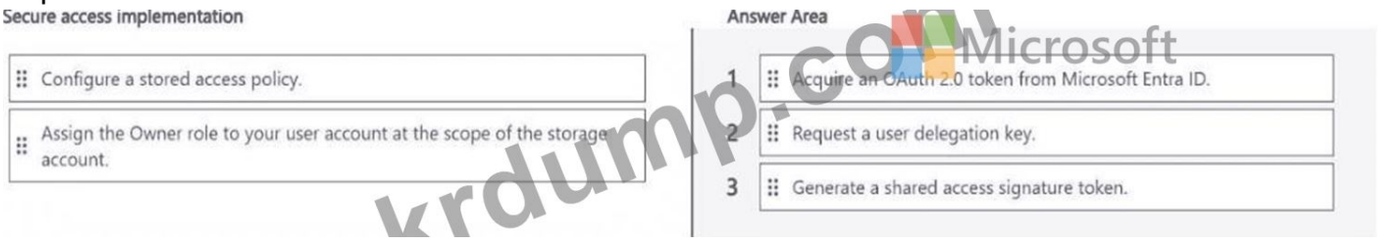
□□□□□ □□□.
* □□□1 □□□□ □□ □□□ □□□□ □□□ □□□ □ □□□ □□□ □□ □□ □□ □□□ □□□□□ □□□.
* □□ □□□ □□□ □□□□□ □□□.
□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□ □□ □□□ □□ □□□□ □□□ □□ □□□ □□ □□□ □□ □□□□ □□□ □□□ □□□ □□□□□.



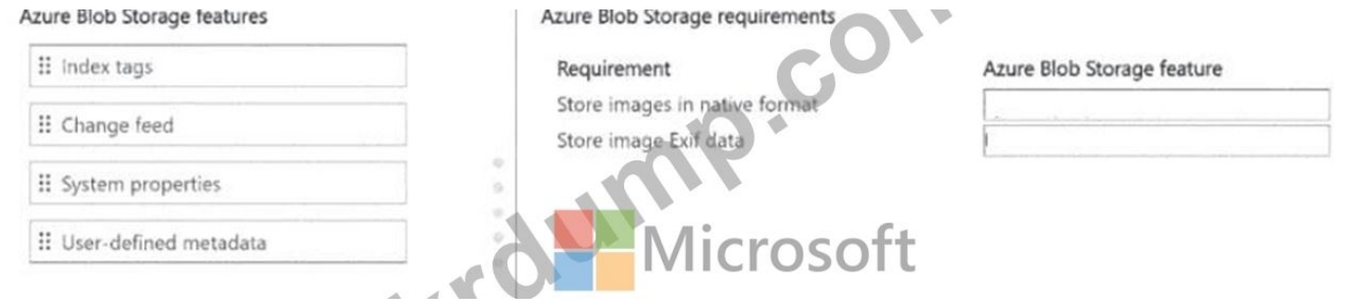
Answer:



Explanation:



NEW QUESTION: 81
□□□□ □□ □□□□ □□□□ □□□.
□□ Azure Blob Storage □□□ □□□□ □□□? □□□□□ □□□ Azure Blob Storage □□□ □□□ □□ □□□□ □□□□. □ Azure Blob Storage □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□ □□□ □ □□□ □□□□ □□□□□ □□□□ □□□□ □ □ □ □□□□. □□: □□□ □□□ □□ 1□□□□.



Answer:

Explanation:

NEW QUESTION: 82

□□□ □□□□ □□□□ □□ □□ □□□□ □□ □ □□ □□□□□□□ □□□□□□. □□ □□□□□ □□□, □□□□ □□ □ □□ □□ □□□□ □□ □□□ □□□□ □□□.

□□ □□□□□□□ □□□□ □□ □□□□ □□ □□□ 3□ □□□ □□□ □□□ □□□□ □ □ □ □□□□ □□ □□□□□ □□□ □ □□□ □□□□ □□□.

□□□□□□□ □□ Azure Cosmos D6 API□ □□□□ □□□?

A. □□

B. □□□

C. □□□□

D. □□□□

Answer: (SHOW ANSWER)

NEW QUESTION: 83

□□ □□□□ □□□ RESTful API □□□ □□□□ □□□□.

Azure API Management□ □□ API□ □□□□ □□□□ □□□. □□□ □□ □□ □□□ □□ □□.

* □□ □ □□□ □□ □□□□□□ API□ □□□□ □ □□□ □□□.

* API □□ □□ □□□□ □□□ □□□□ □□□ □□□ □□□□ □□□.

* □□□ □□□□ □□□□ □□ □□□□ □□□ □□□ □□□ □□ □□□ □□ □ □□□ □

□□□ □□□.
* API □□ □□, □□ □□, □□ □□□ □□ □□□□ □□□□ □□□.
API□ □□□□ □□□. □□□ □□ □□□?
A. □□□ □□ □□ □□□□□. □□ □□□ □□□□, □□□ □□ API □□□ □□□□□. □□ □□□□ Application Insights□ □□□□□.
B. □□ □ □□ □□□ OAuth 2.0□ □□□□□. □□ □□ □□□ □□□□□.
C. □□□ □□□□ □ □□□□□ □□□ □□□□□. □□□ □□□ □□ Azure Monitor□ □□ □□□□.
D. □□□□□□ □□□□□ □□□□ □□□□□ □□□□□. □□ □□ □□□ □□□□□. □ □□ □□□□□ □□ □ □□ □□□ □□□□□. HTTPS□ □□ □□ □□□ □□□□□. IP □ □□ □□□ □□□□□. □□□□□ □□□□□□□□ □□ □□□ □□□□□. Azure □□ □□ □ □□□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 84
□□□□ □ □ □□□ □□□□ SaaS □□□□□□□ □□ □□□□.
□□ □□□□□ □□ □□□ □□□ □ □□□ □□ □□□. □□ □□ □□□□□ □□□ □□□ □□ □□ □□ □□ □□□ □□□□.
□ □□ □□□□! Edition □□□ □□□ □□□ □□ □□□ □□ □□□ □□□ □ □□□ □□ □. Azure □□□ □□□□□ □□□.
□□□□□□□ □□ Azure Cosmos OB API□ □□□□ □□□?
A. □□
B. □□□ API
C. □□□□
D. □□□□
Answer: (SHOW ANSWER)

NEW QUESTION: 85
Azure Storage □□□ □□□ □□ □□□ □□ □□□□□□□ □□□□□.
□□□□ □□ □□□□ □□□□ □□□. □□□ □□ □□ □□□ □□□□.
* Azure □□□□ □□ □□ □□□□□□ □□□□ □□□□□.
* □□□□ □□ □□ □□ □□□□ □□□ □□ □□ □□□□ □□□ □ □□□ □□□□□.
* □□□ □□ □□□ □□ □□□ □□□□ □□□□ □□□□□.
□□ □□□ □□ □□□ □□(SAS)□ □□□□ □□□?
A. □□□ □□□ □□□ □□ □□□ SAS
B. □□ SAS
C. □□□ □□ SAS
D. □□ SAS□ □□ □□□ SAS
Answer: A (LEAVE A REPLY)
A service SAS is secured with the storage account key. A service SAS delegates access to a resource in only one of the Azure Storage services: Blob storage, Queue storage, Table storage,

or Azure Files.
Stored access policies give you the option to revoke permissions for a service SAS without having to regenerate the storage account keys.
Reference:
https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

NEW QUESTION: 86
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
□□□ □□□□ □□ SaaS(Software as a Service) □□□□ □□□□□. □□□□ □□□ □ □ □□□ □□□□ □□ Azure Storage Blob □□□□ □□□ □□□□□. □□□□ □□ □□□ □ □ V2□□□.
□□□ □□□□□ □□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□□□ □□□.
□□□ □□□ □□□ □□□□ □□□□ □□□□□ 1□ □□□ □□□□□ □□□.
□□ □□□ □□□□ □□□□□ □□□□ □□□.
□□□: Blob □□□□□ □□□□ Azure □□□ □□ □□□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: (SHOW ANSWER)
Azure Storage events allow applications to react to events. Common Blob storage event scenarios include image or video processing, search indexing, or any file-oriented workflow. Events are pushed using Azure Event Grid to subscribers such as Azure Functions, Azure Logic Apps, or even to your own http listener.
Note: Only storage accounts of kind StorageV2 (general purpose v2) and BlobStorage support event integration. Storage (general purpose v1) does not support integration with Event Grid.
Reference:
https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-event-overview

NEW QUESTION: 87
□ □□□ Azure □□□□ □□□ □□□□ □□□□□□□□ □□□□ □□□□. □□ HTTP □ □ □□□□ □□□□□ □□□□. □□□□□ □□□□□□□ □□□□□ □□ □□□□□ □□ □□ □□□ □□□□□. □□□□□□□□ □□□□□ □□□□ □□□. □□□ □□ □□□?
A. □□ □□□□□ □□ □□ □□□□ □□□□□□ □□ □□□□ □□□□□.
B. □ □□□□□ □□ □□□ □□□□□.
C. □□ □□ □□□ □□□□□□.
D. □□ □□□□□ □□ □□ □□□ □□□□□.
E. □□ □□ □□□ □□□□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 88

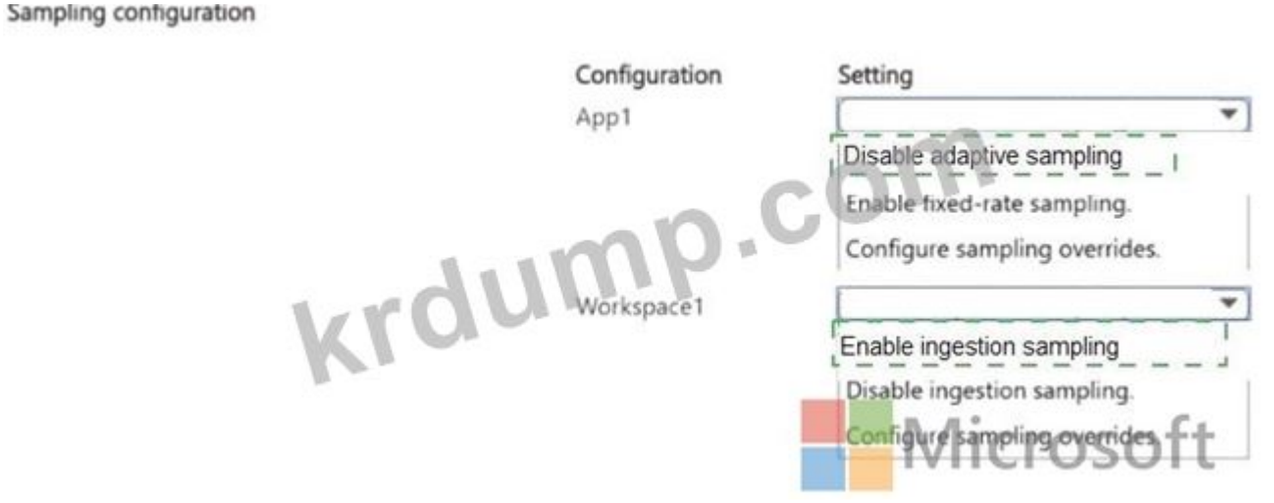
C#□□ □□□ App1□□□ Azure Function □□ Workspaces App1□□□ Application Insights □ □ □□□ □□□□. □ □□ □□□ Application Insights□ □□□□□ □□□□□□ □□ □□□ Workspaces□ □□□□ □□□□ □□□□. App1□ □□ □□ □□□ Workspacel□ □□ □□□ □ □□□□□ □□□□ □□ □□□ □ □□□□.

□□ □□ □□□ Workspacel□ □□ □□□ □□ □□□ □□□□ □□□.

□□□ □□ □□□? □□□□□ □□ □□□□ □□□ □□□□ □□□□□.

□□: □□ □□□ 1□□□□.

Answer:



Explanation:

NEW QUESTION: 89

Azure □□ □□□□ □□□□ □□ □□ □□□□□□□ □□□□ □□□□. □□ □□□ Windows Server□ □□□□□. □□□ □□ □□□ □□ 443□ □□ □□□□□ □□□□ □ □ □□□. □□ □□□ □□□□ □□ □□ □□□□ □ □□□□. □□ □□ □□□ □□□□ □□ □□□ □□□□ □□□□ □□□□ □□□□ □□□.

* □□ □□□□□ □□□□ □□ □□□ □□□ □ □□□□.

* □□□□□ □□ □□□ □□□ □□□□□□. □□ Azure □□□□ □□□□ □□□?

A. □□ □□

B. □□□ □□

C. □□□ □□□□□

D. Azure □□□□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 90

□□□ □□□ □□□□ □□□ □□ □□□□ □□□□.

□□□ □□ □□ □□ □□□ □□□□ □□□.

* □□□□ □□□ □□□□ □□□□□.

* □□□□ □□□ □□ □□ □□ □□□□ □□□□□□□.

□□□□ □□□ □□□□ □□□□ □□□.

□□ □ □□ □□□ □□□□ □□□? □ conn I □□□ □□□□ □□□ □□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

A. PullAsync □□□□ □□□ □□□ □□□□ □□□ □□□□□ □□□□ □□□□□.

B. □□ □□□□ □□□□ □□□□ □□□ □□□□□ □□□□ □□□□□.

C. □□ □□□□ □□□□ □□□□ □□□□ □□□ □□□□ □□□□□.

D. Mobile Service Backend□□ updatedAt □□□ □□□□ □□□ □□□□ □□□ □□□□□.

E. □□□ □□□ □□□□□ updatedAt □□ □□□□ □□□ ID□□ □□□ □□□□□.

Answer: (SHOW ANSWER)

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

NEW QUESTION: 91

Azure□ □□□□ □□□□ □□□□□ Azure Cosmos DB□ □□□□ □□□□□□□ □□□□ □□ □□□□. □□□□□□□ □□ □□□ □□□ □□□ □□ □□□ □□□□ □□□□□.

□□□□□□□□ □□ ACID □□□□ □□□ □□□ □□ □□ □□□ □□□□□□ □□□ □

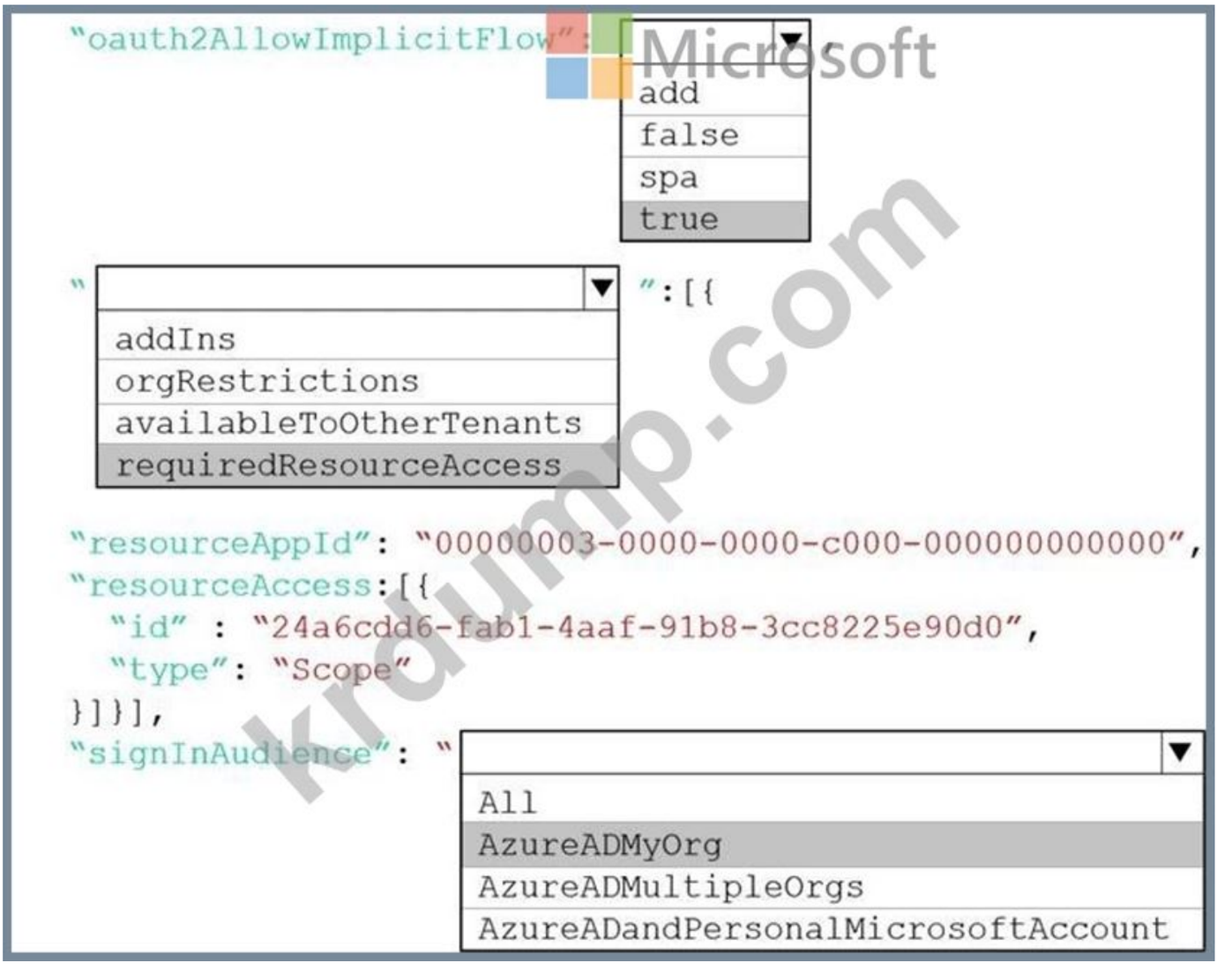
□□□□.
Azure Cosmos DB□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□.
B. Apache Gremlin API□ Azure Cosmos DB□ □□□□□ Azure Cosmos DB□ □□□□□.
C. □□□ □□□□ □□ □□□□ □□ □□□□ □□□□.
D. Azure Cosmos DB for MongoDB API□ □□□□□ Azure Cosmos DB□ □□□□□.
Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 92
□□ □□□ Azure AD(Azure Active Directory) □□□□□ Microsoft Graph□ □□□ □□□□ □□□□ □□□ □□□□ SPA(□□ □□□ □□□□□□) □ □□□□□□□ □□□□.
□□□□ □□□ Azure AD □□□□□ □□□□ Microsoft Graph□ □□□□ □□□□ □ □□□ □□□.
□ □□□ □□ □□□□□□ □□□□□□ □□□□ □□□.
□□□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.


{
"oauth2AllowImplicitFlow":
add
false
spa
true
"
":[{
addIns
orgRestrictions
availableToOtherTenants
requiredResourceAccess
"resourceAppId": "00000003-0000-0000-c000-000000000000",
"resourceAccess":[{
"id" : "24a6cdd6-fab1-4aaf-91b8-3cc8225e90d0",
"type": "Scope"
}]}],
"signInAudience": "
"
All
AzureADMyOrg
AzureADMultipleOrgs
AzureADandPersonalMicrosoftAccount
}

Answer:

Explanation:



Box 1: true

The oauth2AllowImplicitFlow attribute Specifies whether this web app can request OAuth2.0 implicit flow access tokens. The default is false. This flag is used for browser-based apps, like JavaScript single-page apps.

In implicit flow, the app receives tokens directly from the Azure Active Directory (Azure AD) authorize endpoint, without any server-to-server exchange. All authentication logic and session handling is done entirely in the JavaScript client with either a page redirect or a pop-up box.

Box 2: requiredResourceAccess

With dynamic consent, requiredResourceAccess drives the admin consent experience and the user consent experience for users who are using static consent. However, this parameter doesn't drive the user consent experience for the general case.

resourceAppId is the unique identifier for the resource that the app requires access to. This value should be equal to the appId declared on the target resource app.

resourceAccess is an array that lists the OAuth2.0 permission scopes and app roles that the app requires from the specified resource. Contains the id and type values of the specified resources.

Example:

"requiredResourceAccess": [

{

"resourceAppId": "00000002-0000-0000-c000-000000000000",
"resourceAccess": [
{
"id": "311a71cc-e848-46a1-bdf8-97ff7156d8e6",
"type": "Scope"
}
]
}
],
Box 3: AzureADMyOrg
The signInAudience attribute specifies what Microsoft accounts are supported for the current application.
Supported values are:
AzureADMyOrg - Users with a Microsoft work or school account in my organization's Azure AD tenant (for example, single tenant) AzureADMultipleOrgs - Users with a Microsoft work or school account in any organization's Azure AD tenant (for example, multi-tenant) AzureADandPersonalMicrosoftAccount - Users with a personal Microsoft account, or a work or school account in any organization's Azure AD tenant Reference:
https://docs.microsoft.com/en-us/azure/active-directory/develop/reference-app-manifest
https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-implicit-grant-flow

NEW QUESTION: 93
APSPlanl□□□ Azure App Service □□□□ □□ B1 □□ □□□□ □□□□□□. APSPlanl□ □ WebAppl□□□ App Service □□□ □□□□ □□□□. APSPlanl□ □□ □□ □□ □□ □□ □ □□□□ □□□□□. WebAppl □□ □□□ □□□□□ □□□.
□□□: APSPlan1□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: (SHOW ANSWER)

NEW QUESTION: 94
□□ □□□ □□□□ □□□.
□□□ □□ □□□?
A. Azure Function□ □□□□ Azure Blob □□□□ □□□□ □□
B. □□ □□□ □□□ □□□□□ □□□□ □□□□□□ □□□□□.
C. Azure Function□ □□ App Service □□□□ □□
D. □□□ □□□ □□□□□ PC09 □□□□ □□□□ □□□ □□□□□□□.
Answer: D (LEAVE A REPLY)
If you want to read the files in parallel, you cannot use forEach. Each of the async callback function calls does return a promise. You can await the array of promises that you'll get with

Promise.all.

Scenario: Capacity issue: During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

```
PC08      var container = await GetCloudBlobContainer();
PC09      foreach (var fileItem in await ListFiles())
PC10      {
PC11        var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12        var ms = new MemoryStream();
PC13        await file.DownloadToStreamAsync(ms);
PC14        var blob = container.GetBlockBlobReference(fileItem.Uri.ToString());
PC15        await blob.UploadFromStreamAsync(ms);
PC16
PC17      }
```

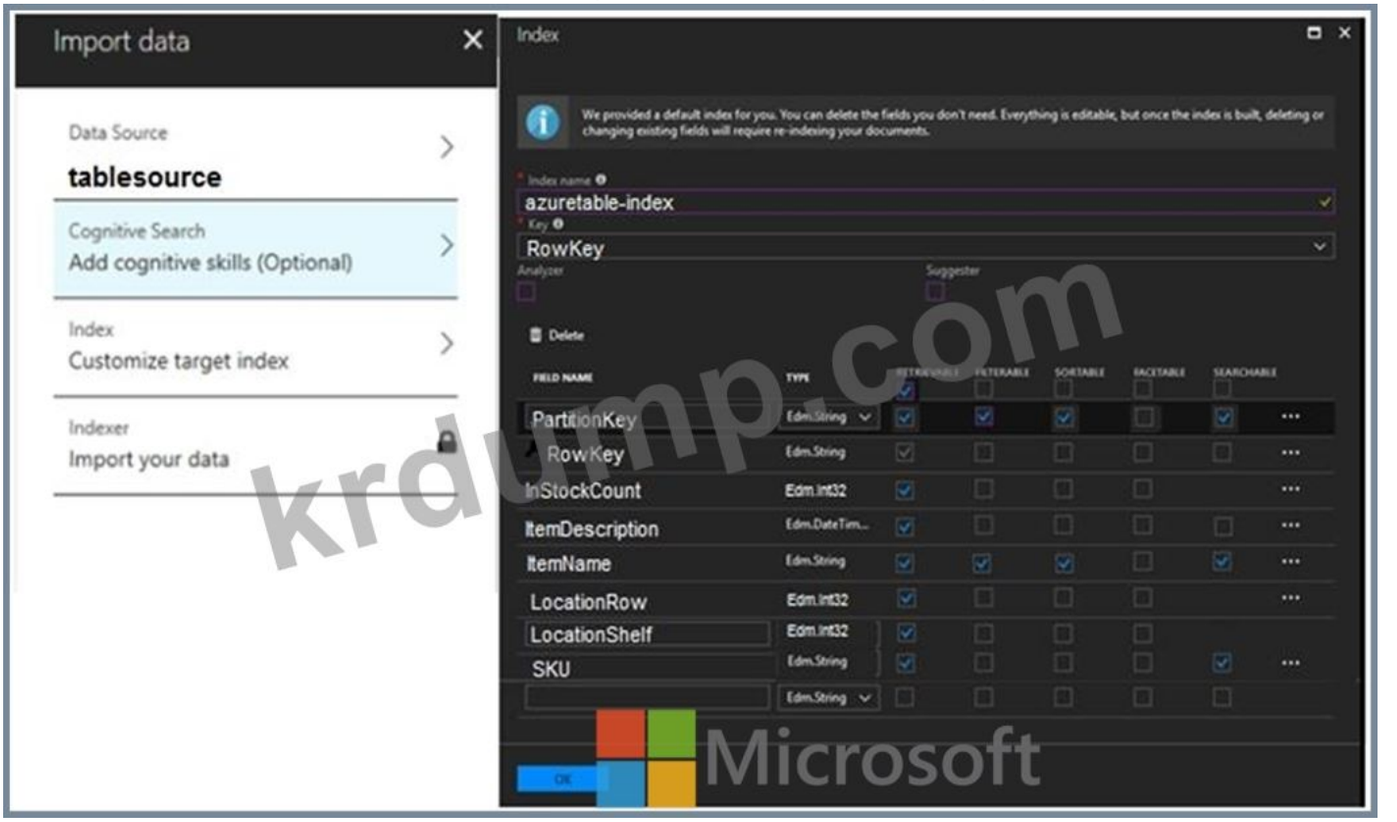
Reference:

https://stackoverflow.com/questions/37576685/using-async-await-with-a-foreach-loop

NEW QUESTION: 95

Azure □□ □□□ □□□ □□□□ □□□□ □□□□.

□□□ □□□□ □□□ □□□□ □□□□ □□□□ □□□□□□□. □□□ □□ □□ □□□□ □□□ □□ □□□ □□□□ □□□□□. (□□ □□ □□ □□□□□.)



□□□□ □□□ □□ □□□ □□□□ Azure □□□□ □□□□□. □□□□□ □□□□ □□□ □□□□ □□□ □□□□ □□ □□ □□□□ □□ □ □□ □□□□ □□□□ □□□□. □□□ □□□□ □□□□ □□□ □ □□□□□□. (□□□□ □□□ □□ □□□□□.) □□□□ □□ □□ □□□□ □ □□ □□□□ □□ □□□□□.

PartitionKey	RowKey	Timestamp	inStockCount	itemDescription	itemName	LocationRow	LocationShelf	SKU
Food	3	2018-08-25T15:47:29.535Z	32	A box of chocolate candy bars	Choco bar	5	3	123425
Hardware	2	2018-08-25T15:46:06.403Z	2	A bag of bolts	Bolts	1	4	678564
Hardware	1	2018-08-25T15:46:41.403Z	23	A box of nails	Nails	2	1	654365

□□□□ □□□□ □□□ □□□ □ □□ □□□□ □□□□ □□□□ □□□□. □□ □□□ □ □□□ □□□□ □□ □□□ □□□ □□□□□. □□□□□ □□□□ bag□□□ □□□ □□□ □□□ □□ □□□□ □□ □□□ □□□□□ □□□. (□□ □□□ □□ □□□□□.) □□□ □ □□□ □□□.

□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.

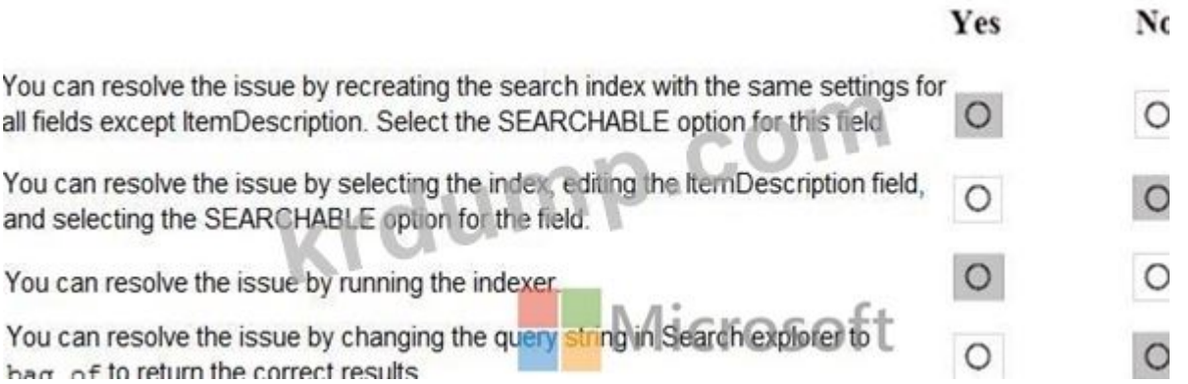
□□: □ □□□ □□□ 1□□ □□□ □□□□.

	Yes	No
You can resolve the issue by recreating the search index with the same settings for all fields except ItemDescription. Select the SEARCHABLE option for this field	○	○
You can resolve the issue by selecting the index, editing the ItemDescription field, and selecting the SEARCHABLE option for the field.	○	○
You can resolve the issue by running the indexer.	○	○
You can resolve the issue by changing the query string in Search explorer to `bag of` to return the correct results	○	○

Answer:

	Yes	No
You can resolve the issue by recreating the search index with the same settings for all fields except ItemDescription. Select the SEARCHABLE option for this field	○	○
You can resolve the issue by selecting the index, editing the ItemDescription field, and selecting the SEARCHABLE option for the field.	○	○
You can resolve the issue by running the indexer.	○	○
You can resolve the issue by changing the query string in Search explorer to `bag of` to return the correct results	○	○

Explanation:



Box 1: Yes

The ItemDescription field in not searchable.

Box 2: No

The ItemDescription field in not searchable, but we would need to recreate the index.

Box 3: Yes

An indexer in Azure Search is a crawler that extracts searchable data and metadata from an external Azure data source and populates an index based on field-to-field mappings between the index and your data source.

This approach is sometimes referred to as a 'pull model' because the service pulls data in without you having to write any code that adds data to an index.

Box 4: No

References:

https://docs.microsoft.com/en-us/azure/search/search-what-is-an-index

https://docs.microsoft.com/en-us/azure/search/search-indexer-overview

NEW QUESTION: 96

Azure API Management □□ □ □□□ lo □□□□□□ □□□□□. □□□ □ □□□□ HSTS(HTTP Strict Transport Security)□ □□□□□.

□□□ □□□□ □□ □□ □□□□ □□□ HTTP □□ □□□ □□□□□ □□□.

□□ □□□ □□□□ Azure API Management □□□□□ □□□□ □□□.

□□ □ □□ □□□ □□□ □ □□□□? □ □□□ □□□ □□□□ □□□□□. □□: □ □□□ 1□□ □□□ □□□□.

A. □□ □□

B. □□□□□ □□

C. OAuth □□□□□ □□ □□ □□

D. □□□ □□

Answer: (SHOW ANSWER)

NEW QUESTION: 97

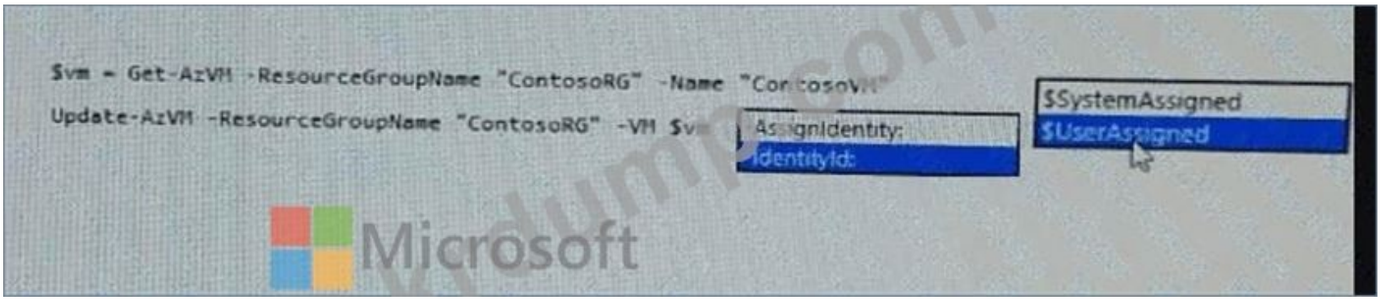
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□ □□□ □□□ □□□ □ □□ □□□ □□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□ □□ □ □□, □□ □□ □□□□ □□□ □□ □ □□□□.

□ □□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□ □□ □□□ □□ □□ □ □□□□ □□□□.
Azure Blob Storage, Azure Resource Manager □ □□ Azure □□□□ □□□□ □□ □□□□□ □□□□ □□ □□□□□□□ □□□□ □□□□.
□□□□ □□ □□ □□□ □□□□ □□□.
* □□ □□ □□ □□ □□ □□□□ □□□□□.
* □□□ □□□□ □□□ □□□ □□□□ □□□□□□□ □□□ □□ □□□□ □□□□□.
* □□ □□□□□□□ □□□□□ □□□ □ □□ □□□□ □□ □□□□ □□□ □□□ □□□.
* □□□□ □□□□ □□□ Azure Functions□ □□□□ □□□□ □□□□□.
* □□□ □□□ □ □□□ □□□ □□□ □□ □□□ □□ □□□□□□.
□□□ □□□□ □□□□ □□□.
□□ □□: Azure Logic Apps□ □□□□ Azure □□□□ □□ □□□ □□□□□ □□□□□.
Logic Apps □□ □□ □□□ □□□□ □□ □□□□ □□□□□□. Logic Apps□□ Azure Functions□ □□□□□ □□□□ □□□□ □□□□□.
□□□□ □□□ □□□□□?
A. □□□
B. □
Answer: (SHOW ANSWER)

NEW QUESTION: 98
□□ □□□ □□□□ □□□□ Azure □□ □□ □□ □□□ □□□□ □□□□□. □□□□ SSL □ □□□□ □□□□ □□ REST API □□□□□□ □□□□ □□□□. Azure □□ □□ □□□ □□□□ □□□□□.
□□ □□□ □□□ □ □□□ □□ □□□ □□□□ □□ □□□ □□□ □□□.
□□□ □□□□ □□□□ □□□□□□ □□□.
□□□ □□ □□□?
A. URL □ □□□□ □□□□□.
B. TrackAvailability()□ □□□□ □□□ Application Insights□ □□□ □□□ □□□ □□□ □□ □□.
C. GetMetric("□□ □□")□ □□□□ □□□ □□□ □□□ □□□ □□□ Application Insights□ □□□□.
D. Azure □□ □□ □ □□ □□□ □□□□□. FunctionAppLogs □ Log Analytics□ □□□ □□ □ □□□□□□.
Answer: B (LEAVE A REPLY)
You can create an Azure Function with TrackAvailability() that will run periodically according to the configuration given in TimerTrigger function with your own business logic. The results of this test will be sent to your Application Insights resource, where you will be able to query for and alert on the availability results data. This allows you to create customized tests similar to what you can do via Availability Monitoring in the portal.
Customized tests will allow you to write more complex availability tests than is possible using the portal UI, monitor an app inside of your Azure VNET, change the endpoint address, or create an

availability test even if this feature is not available in your region.
D18912E1457D5D1DDCBD40AB3BF70D5D
Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/availability-azure-functions

NEW QUESTION: 99
Azure VM(□□ □□)□ □□□□□ □□ □□□□□□□ □□□□ □□□□. □□□□□□□ □ □□ □□ □□□ VM □□□ □□□□□ □□□□□ □□□. VM□ □□ □□ ID□ □□□□□ □ □□□ □□□.
PowerShell □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□ □□□□.
□□ □ □□□ □□□ 1□□ □□□ □□□□.



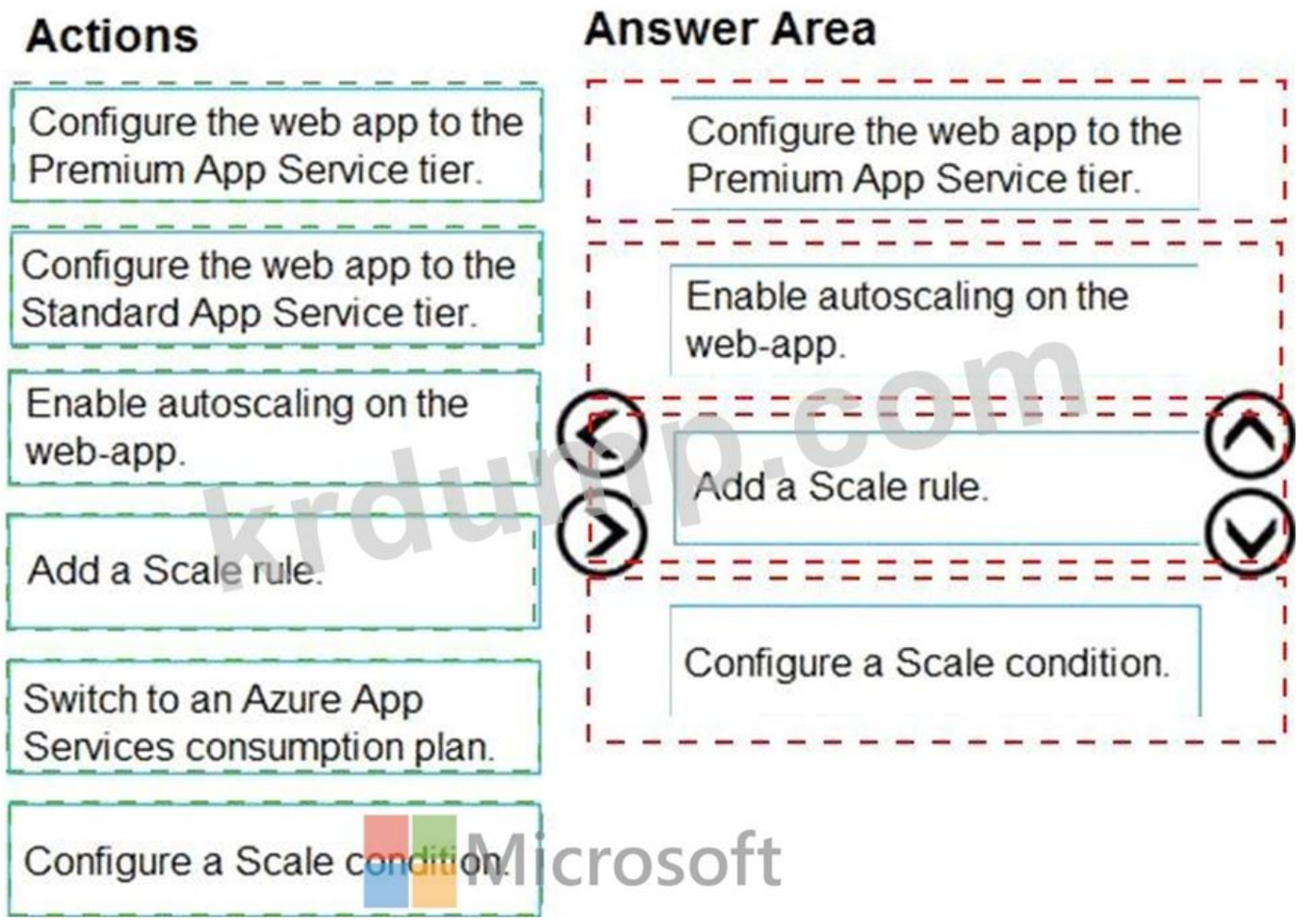
Answer:

Explanation:
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
Update-AzVM -ResourceGroupName myResourceGroup -VM $vm -AssignIdentity:
$SystemAssigned
https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/qs-configure- powershell-windows-vm

NEW QUESTION: 100
Microsoft Azure App Service□ Web Apps □□□ □□□□ □□ D1 □ □□□ □□□ □□□□ □□□ □□□□□.
□□□ □□□□ □□ □□□ □□ □□□ □□□□□□.
CPU □□□ □ 85%□ □ □ □□ □□□□ □□□□ □□□ □□□□□□ □□ □□□.
□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □□ □□□ □□ □□ □□□ □□□□□. □□□ □□□ □□□ □□ □□□□ □□ □□ □.


Actions
Answer Area
Configure the web app to the Premium App Service tier.
Configure the web app to the Standard App Service tier.
Enable autoscaling on the web-app.
Add a Scale rule.
Switch to an Azure App Services consumption plan.
Configure a Scale condition.
krdump.com
Microsoft

Answer:



Explanation:

Step 1: Configure the web app to the Standard App Service Tier

The Standard tier supports auto-scaling, and we should minimize the cost.

Step 2: Enable autoscaling on the web app

First enable autoscale

Step 3: Add a scale rule

Step 4: Add a Scale condition

Reference:

https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-autoscale-get-started

NEW QUESTION: 101

Azure Event Grid□ □□ □□□□ □□□□□ □□□□ □□□.

□□□ □□□□ □□□?

A. Azure Event Grid □□

B. Azure Service Bus □□

C. Azure Service Bus □

D. Azure □□□□ □

E. Azure Logic App □□□ □□ □□□

Answer: D (LEAVE A REPLY)

As a solution architect/developer, you should consider using Service Bus queues when:

* Your solution needs to receive messages without having to poll the queue. With Service Bus, you can achieve it by using a long-polling receive operation using the TCP-based protocols that Service Bus supports.

Reference:

https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-azure-and-service-bus-queues- compared-contrasted

NEW QUESTION: 102

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

Azure Storage Blob □□□□ □□□□ □□ HTTP □□□ Azure □□ □□ □□□□□. □□ Blob □ □□ □□□□ □□□□ □□□□□□.

□□ 4□ □□□ □□ □□ □□□□□. □□ Blob □□□□ □□□□ □□□.

□□ □□ □□□□ □□ Blob □□□□ □□□□□ □□□□ □□□.

□□ □□: HTTP □□□ □□□□□ Azure Service Bus □□ □□□□ □ □□□ □□□□ □□□ □ □□□□ HTTP □□ □□□ □□□□□.

□□□□ □□□ □□□□□?

A. □

B. □□□

Answer: A (LEAVE A REPLY)

Large, long-running functions can cause unexpected timeout issues. General best practices include:

Whenever possible, refactor large functions into smaller function sets that work together and return responses fast. For example, a webhook or HTTP trigger function might require an acknowledgment response within a certain time limit; it's common for webhooks to require an immediate response. You can pass the HTTP trigger payload into a queue to be processed by a queue trigger function. This approach lets you defer the actual work and return an immediate response.

Reference:

https://docs.microsoft.com/en-us/azure/azure-functions/functions-best-practices

NEW QUESTION: 103
Azure App Service □□□ □□□□ □□□□.
□□□ Azure Redis Cache□ □□ □□□ □□□□ □□□□ □□□.
□□□ Azure Redis Cache□ □□□□ □□□.
□□ □ □□ Azure Redis Cache □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□ □.
□ □□□ □□□ 1□□ □□□ □□□□.
A. SSL □□
B. □□ □□
C. □□
D. □□□ □□
E. □□□ □
F. □□ ID
Answer: A,D,E (LEAVE A REPLY)
https://learn.microsoft.com/en-us/azure/azure-cache-for-redis/cache-web-app-howto

NEW QUESTION: 104
□□□ □□□ □□□ □□ □□□ □□ □□ □□ □□□ □□□□□ Azure Mobile Apps SDK□ □□□□ Application Insights □□ □□□ □□□□ □□□.
Application Insights□ Usage Analytics □□□ □□□□ □ □□□ □□□□ □□□□ □□□.
□□ □ □□ □□□ □□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□. □□: □ □□ □ 1□□ □□□ □□□□.
A. □□
B. □□ ID
C. □□
D. □□□ ID
E. □□□
Answer: A,D,E (LEAVE A REPLY)
Application Insights is a service for monitoring the performance and usage of your apps. This module allows you to send telemetry of various kinds (events, traces, etc.) to the Application Insights service where your data can be visualized in the Azure Portal.
Application Insights manages the ID of a session for you.
References:
https://github.com/microsoft/ApplicationInsights-Android

NEW QUESTION: 105
□□ □□ □□□(□□ □□□)□ □□□□ Azure □□ □□ □□□□ □□□□. □□ □□□ □□ □□ 11□□ □□□□□. □□□□ □□ □ 1□□ □□ □□□□ □□□. □□□ □□□□ □□□ □□□□ □□□□□ □□□ □□□□□. □□ □□ □□ □□□ □□□(UTC - 8) □□□□□

Azure □□□ □□ □□□□ □□ □□□□□ □□□□□ Azure □□□ □□□ □□□ □□□ □ □□□□ □□□□ □□□.

□□□ □□ □□□?

A. □□□ □□□ □□ □□□□ WEBSITE_TIME_ZONE□□□ □ □□□ □□□□□.

B. Azure Function□ □□□□ □□□□ □□□□□□□.

C. □□ 7□□ □□□□□ □□□ □□□ □□

D. □□ □□ □□□□ □□□□□ Azure □□□ □□□□□.

Answer: D (LEAVE A REPLY)

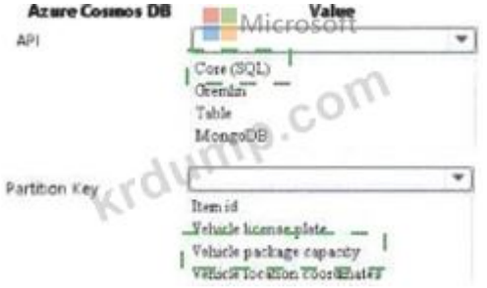
NEW QUESTION: 106

□□ □□□ □□ □□ □□□□ □□□□ □□□.

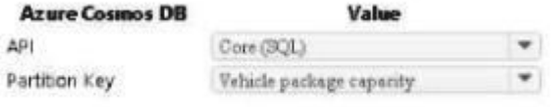
□□□□ □□□ □□□□ □□□?

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:



NEW QUESTION: 107

□□ □□□ □□ □□□□ □□□□□ □□□ EventGridController.cs□ EG15 □□ □□□ □□ □□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□□□ □□□ □□□ □□□□□ □□.

□ □□ □□□□□ □ □, □ □ □□ □□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Code segments

topic
status
eventType
Succeeded
operationName
resourceProvider

Answer Area

```
if {
  @event[ "data" ][ "  code segment  " ].ToString() == "  code segment  "
  &&
  @event[ "data" ][ "  code segment  " ].ToString() == "Microsoft.Web/sites/write"
)
```

Answer:

Code segments

topic
status
eventType
Succeeded
operationName
resourceProvider

Answer Area

```
if {
  @event[ "data" ][ "  status  " ].ToString() == "  Succeeded  "
  &&
  @event[ "data" ][ "  operationName  " ].ToString() == "Microsoft.Web/sites/write"
)
```

Explanation:

```
if {
  @event[ "data" ][ "  status  " ].ToString() == "  Succeeded  "
  &&
  @event[ "data" ][ "  operationName  " ].ToString() == "Microsoft.Web/sites/write"
)
```

Scenario, Log policy: All Azure App Service Web Apps must write logs to Azure Blob storage.

Box 1: Status
Box 2: Succeeded

Box 3: operationName

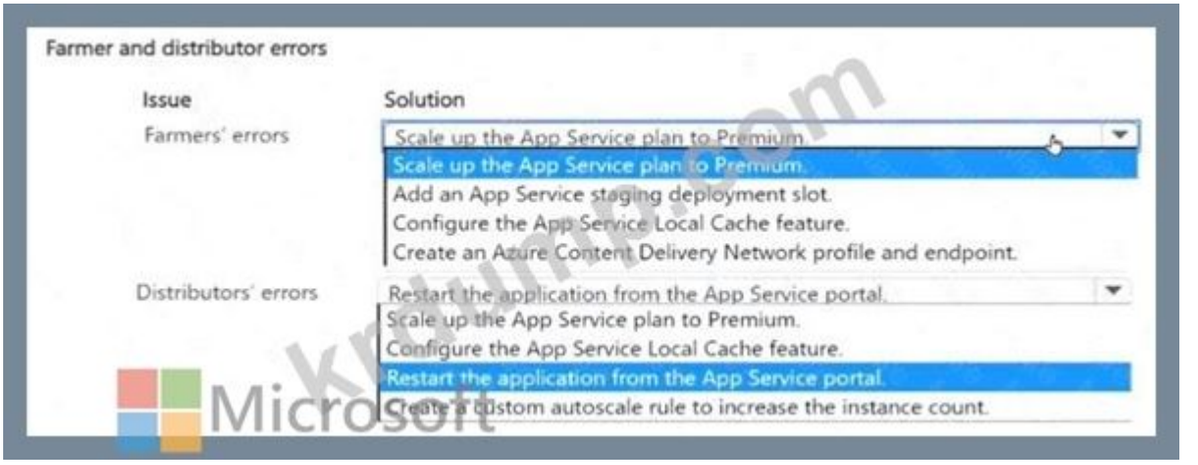
Microsoft.Web/sites/write is resource provider operation. It creates a new Web App or updates an existing one.

Reference:

https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations

NEW QUESTION: 108

□□□ □□□□□ □□□ □□□□□ □□□.
□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□. □□: □
□□□ 1□□□□.



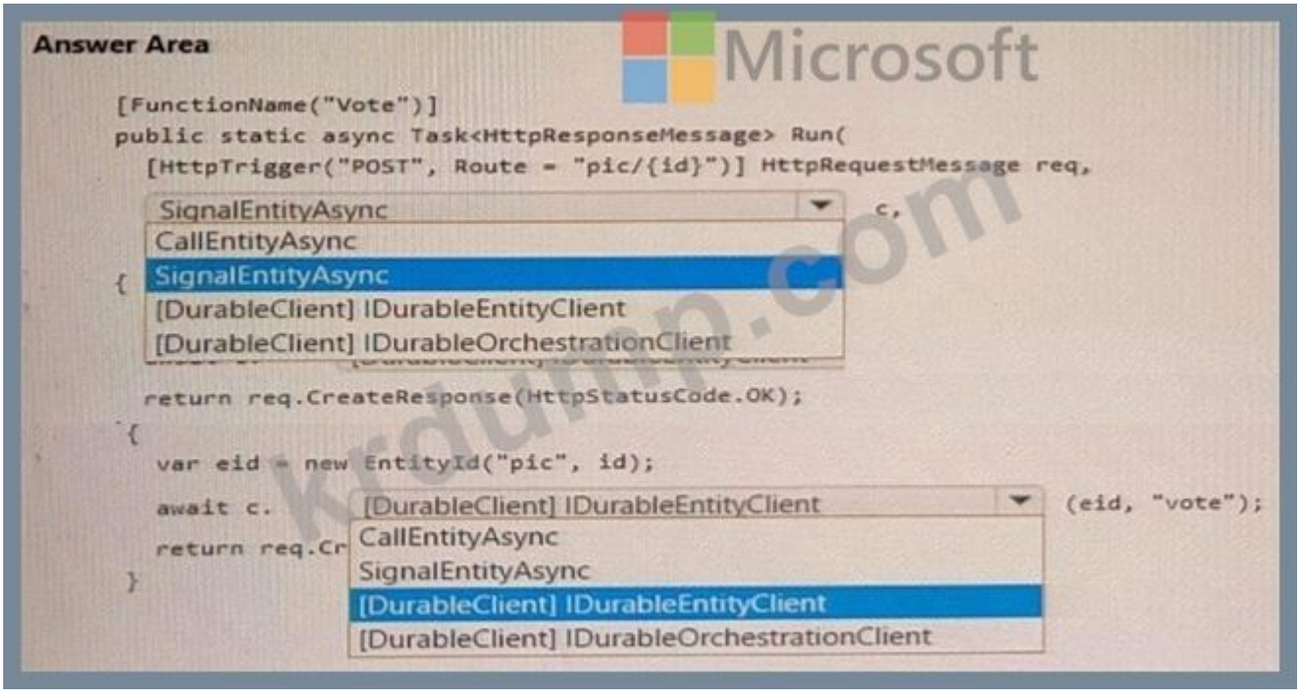
Answer:

Explanation:

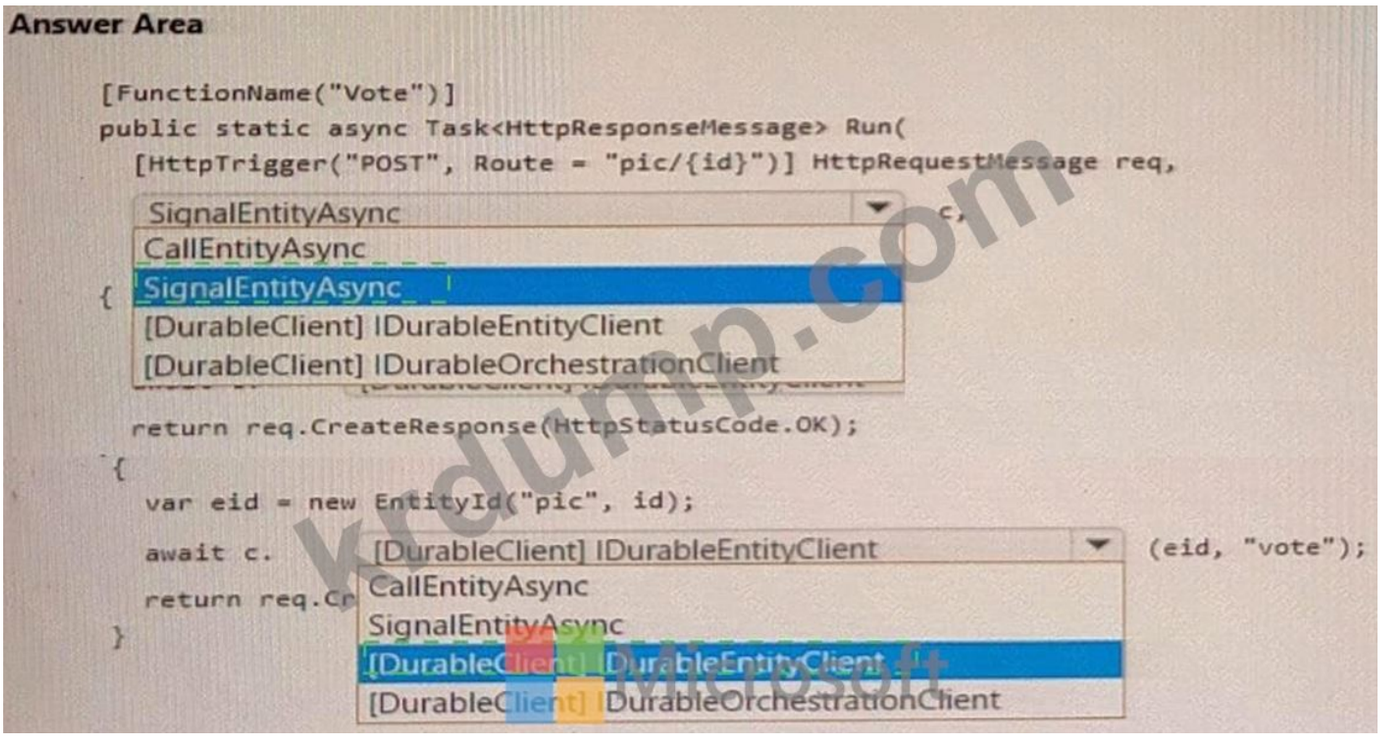
NEW QUESTION: 109
□□□ □ □□□ □□□ □□ □ □□□□□ □□□□□ □□ □□ □□□ □□ □□ □□ □□□ □ □□□□ □□□□.
Azure Service Bus□ □□□□ □□□ □□□□□. □ □□ □□□□□ Azure Service Bus□ □□ □□ □□□□□ □□□ □□ □□□ □□□□□. □□ □□□□ □□ □□□□ □□□□□ □□ □. □ □□ □□□□ □□□□ □□ □□□ □□ □□□ □□□□□ □□□.
□□ □□ □□ □□□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□ □□□ SessionID □□ □□ ReplyToSessionId □□□ □□□□□.

B. □□ □□□ MessageId □□□ □□ DevileryCount □□□ □□□□□.
C. □□ □□□ SessionID □□ □□ SequenceNumber □□□ □□□□□.
D. □□ □□□ MessageId □□□ □□ CorrelationId □□□ □□□□□.
E. □□ □□□ SequenceNumber □□ □□ DeliveryCount □□□ □□□□□.
F. □□ □□□ MessageId □□ □□ SequenceNumber □□□ □□□□□.
Answer: (SHOW ANSWER)
Reference:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messages-payloads

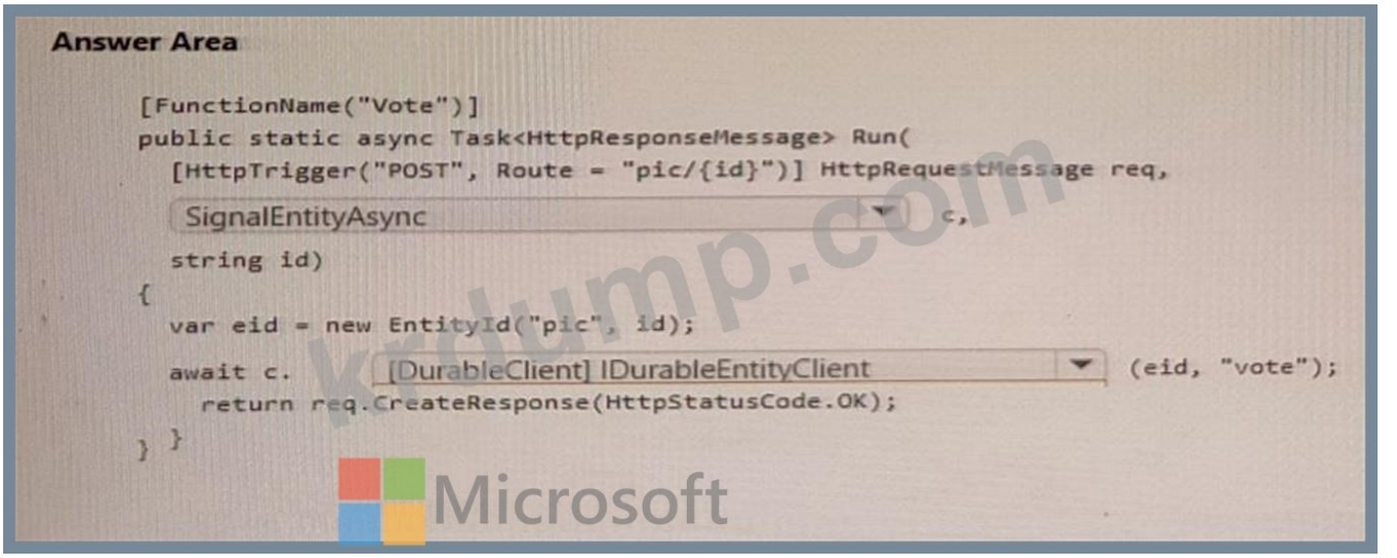
NEW QUESTION: 110
□□□ □□□□□ □□□ □□□□ □□ □□□□ □□□ □□□ □ □□ □□□ □□□ □□□ □ □□□□.
□ □□□ Azure Functions□ □□□□ □□□□□□ □□□ □□ □□□□ □□□□ □□ □□ □□□□□. □□ □□□ 30□□□□. □□ □□□ □□□□□ □□□.
□□□ □□ Azure Function□ □□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.



Answer:

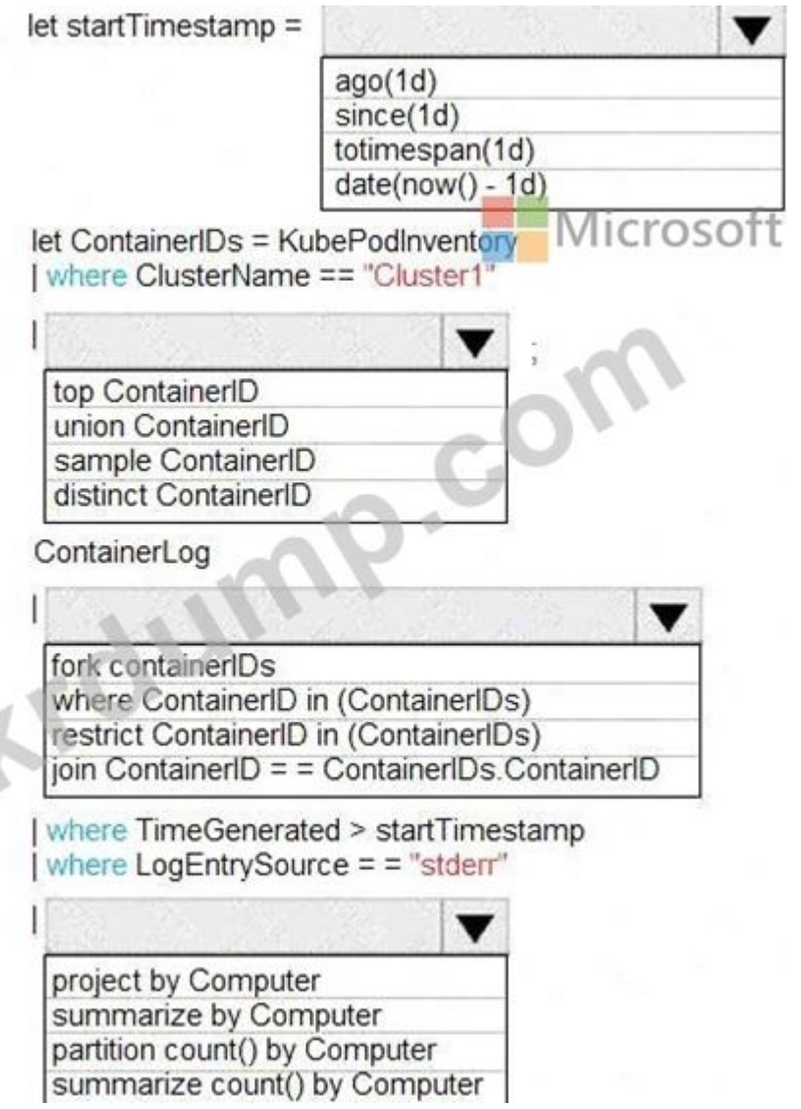


Explanation:

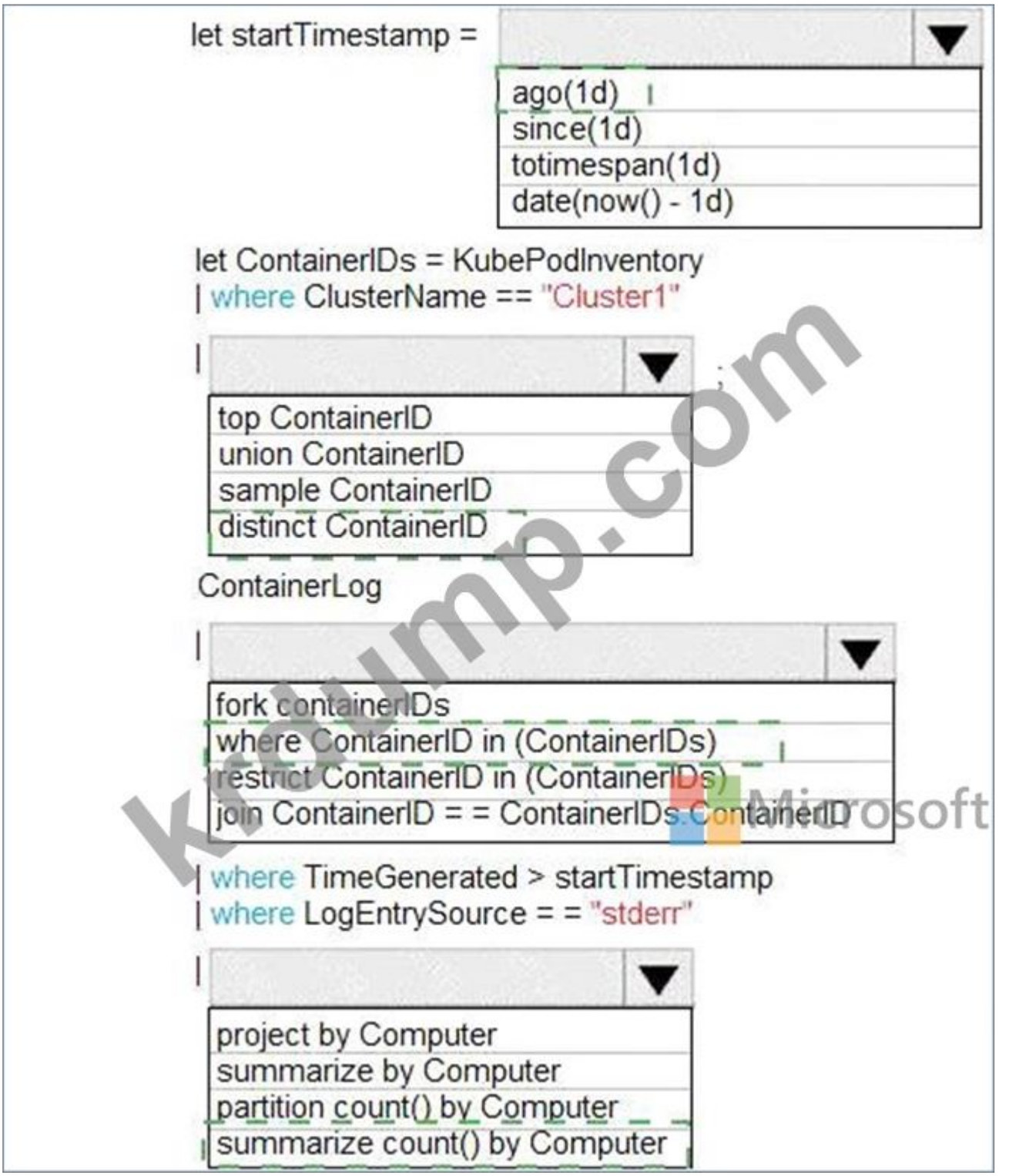


NEW QUESTION: 111

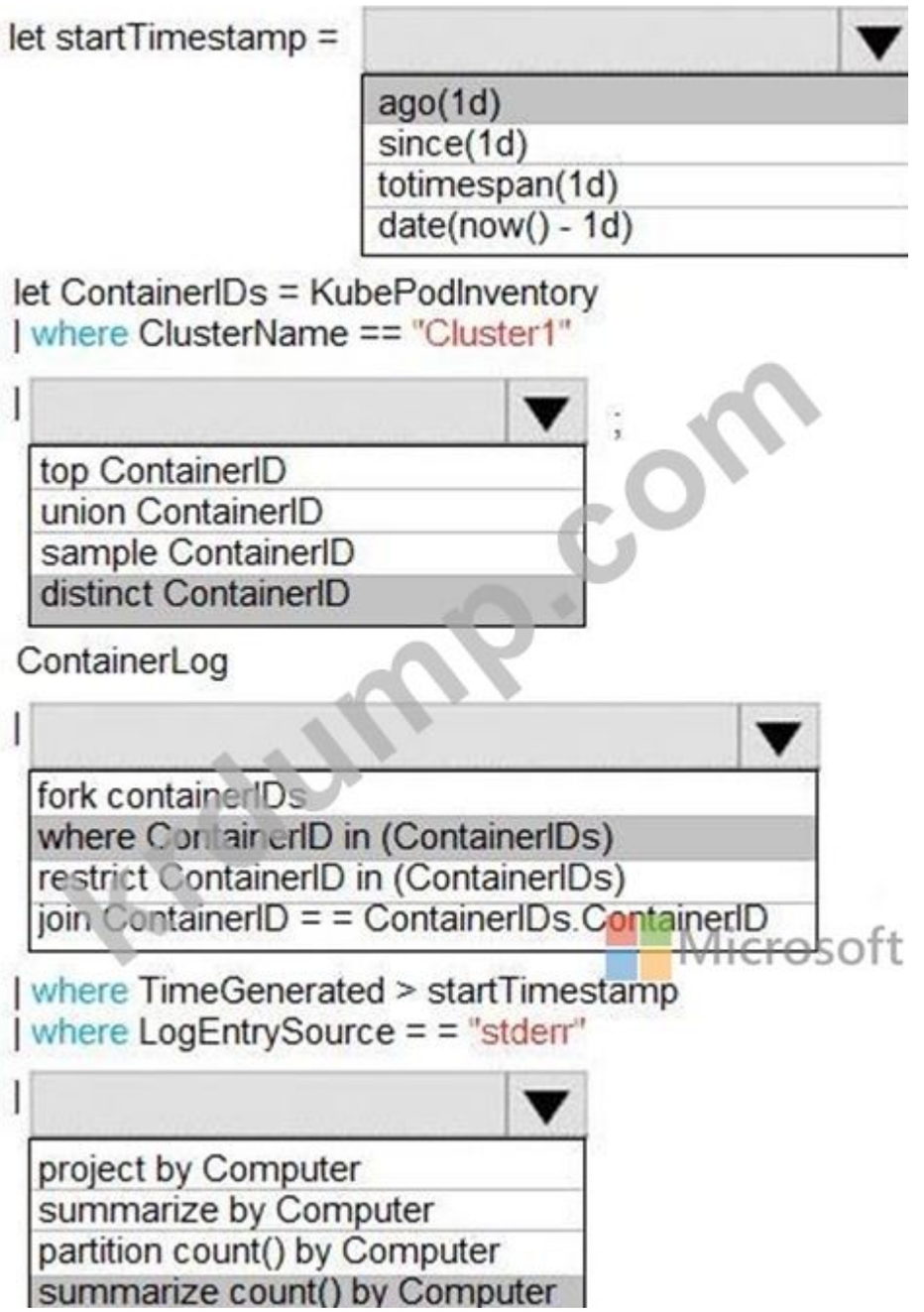
cluster1□□□ Azure Kubernetes □□□□□□ □□ □□ □□□□□□□ □□□□□ □□□□.
□□□□□ □□□□□ Azure Monitor□ □□□□ □□□□□ □□□□□□□.
□□□□□□□ □□□□ □□□□□□ □□□□ □□ □□□ □□□□.
□□ □□□ □□ 24□□ □□ □□□□□□□□ □□ □□ □□□ □□□□□.
□□□ □□□□ VM(□□ □□)□ □□□□ □□□.
Azure Monitor □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□ □□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.


let startTimestamp =
ago(1d)
since(1d)
totimespan(1d)
date(now() - 1d)
let ContainerIDs = KubePodInventory
| where ClusterName == "Cluster1"
|
top ContainerID
union ContainerID
sample ContainerID
distinct ContainerID
;
ContainerLog
|
fork containerIDs
where ContainerID in (ContainerIDs)
restrict ContainerID in (ContainerIDs)
join ContainerID = = ContainerIDs.ContainerID
| where TimeGenerated > startTimestamp
| where LogEntrySource = = "stderr"
|
project by Computer
summarize by Computer
partition count() by Computer
summarize count() by Computer

Answer:



Explanation:



Box 1: ago(1d)

Box 2: distinct containerID

Box 3: where ContainerID in (ContainerIDs)

Box 4: summarize Count by Computer

Summarize: aggregate groups of rows

Use summarize to identify groups of records, according to one or more columns, and apply aggregations to them. The most common use of summarize is count, which returns the number of results in each group.

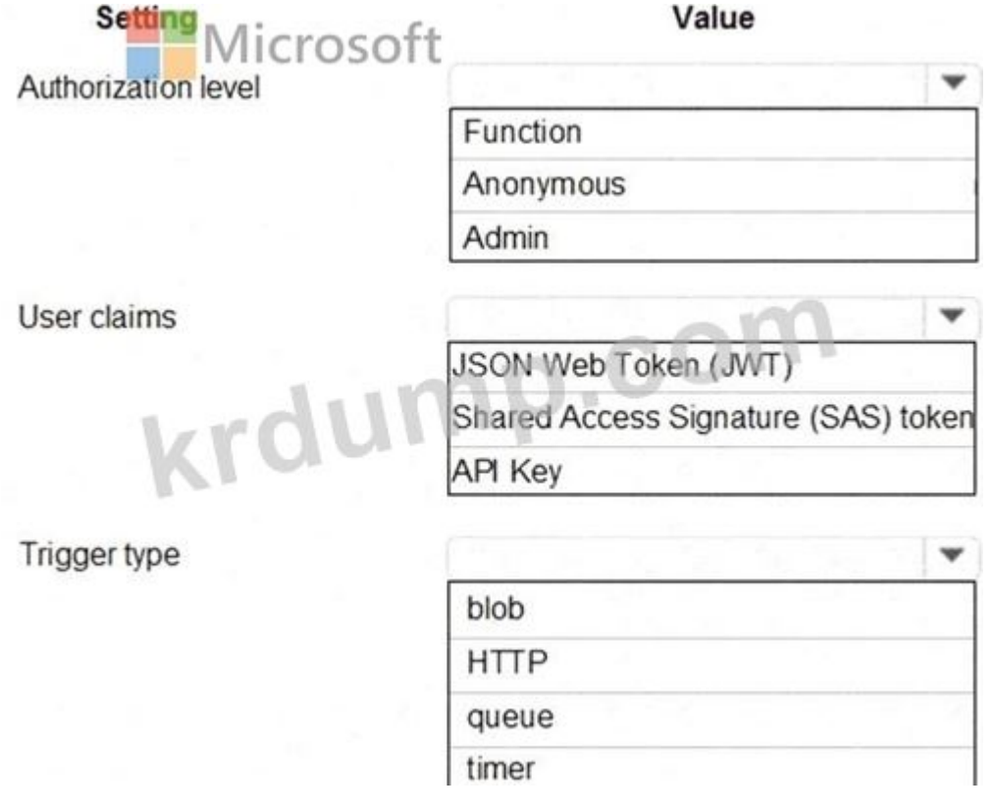
Reference:

https://docs.microsoft.com/en-us/azure/azure-monitor/log-query/get-started-queries

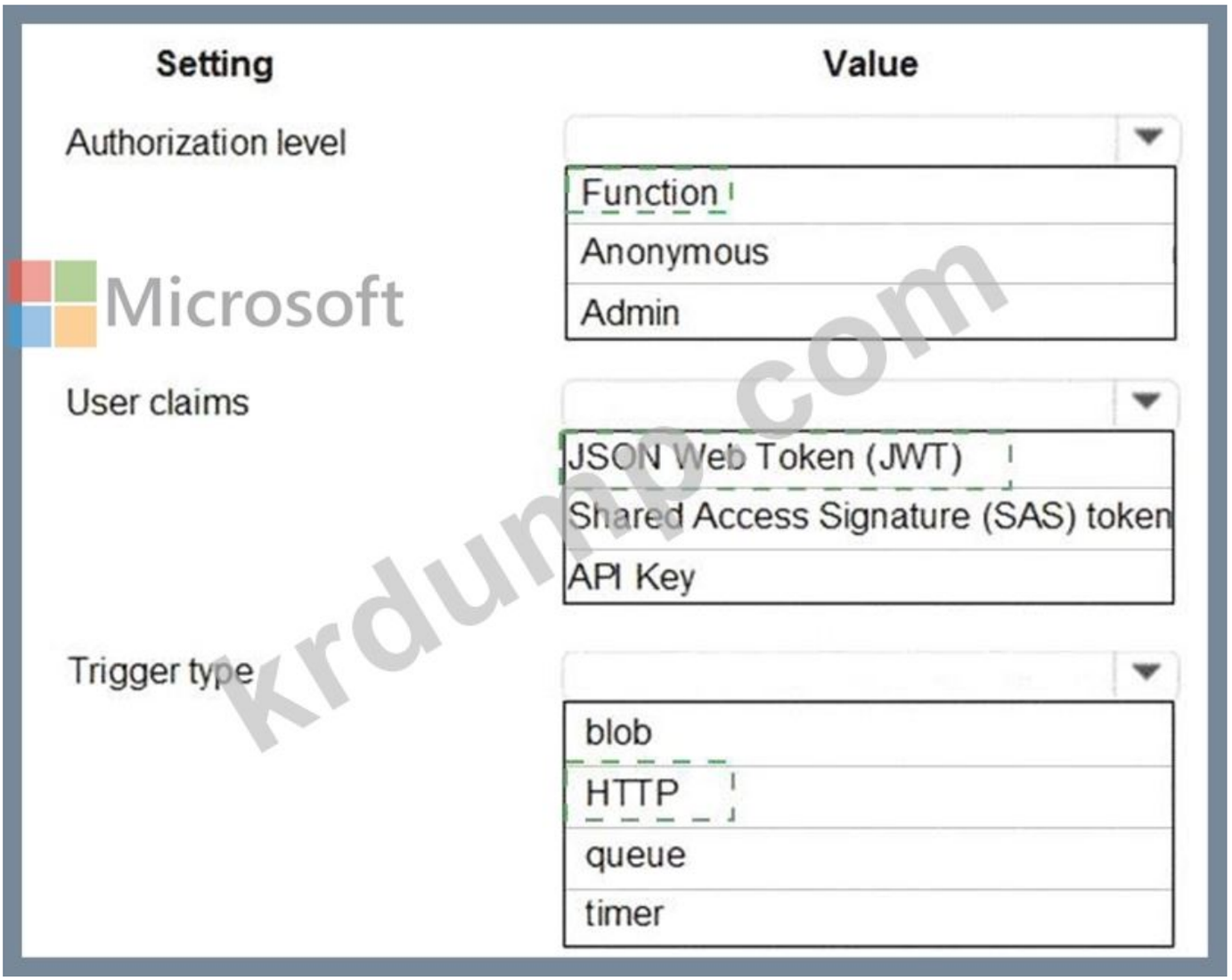
https://docs.microsoft.com/en-us/azure/azure-monitor/log-query/query-optimization

NEW QUESTION: 112

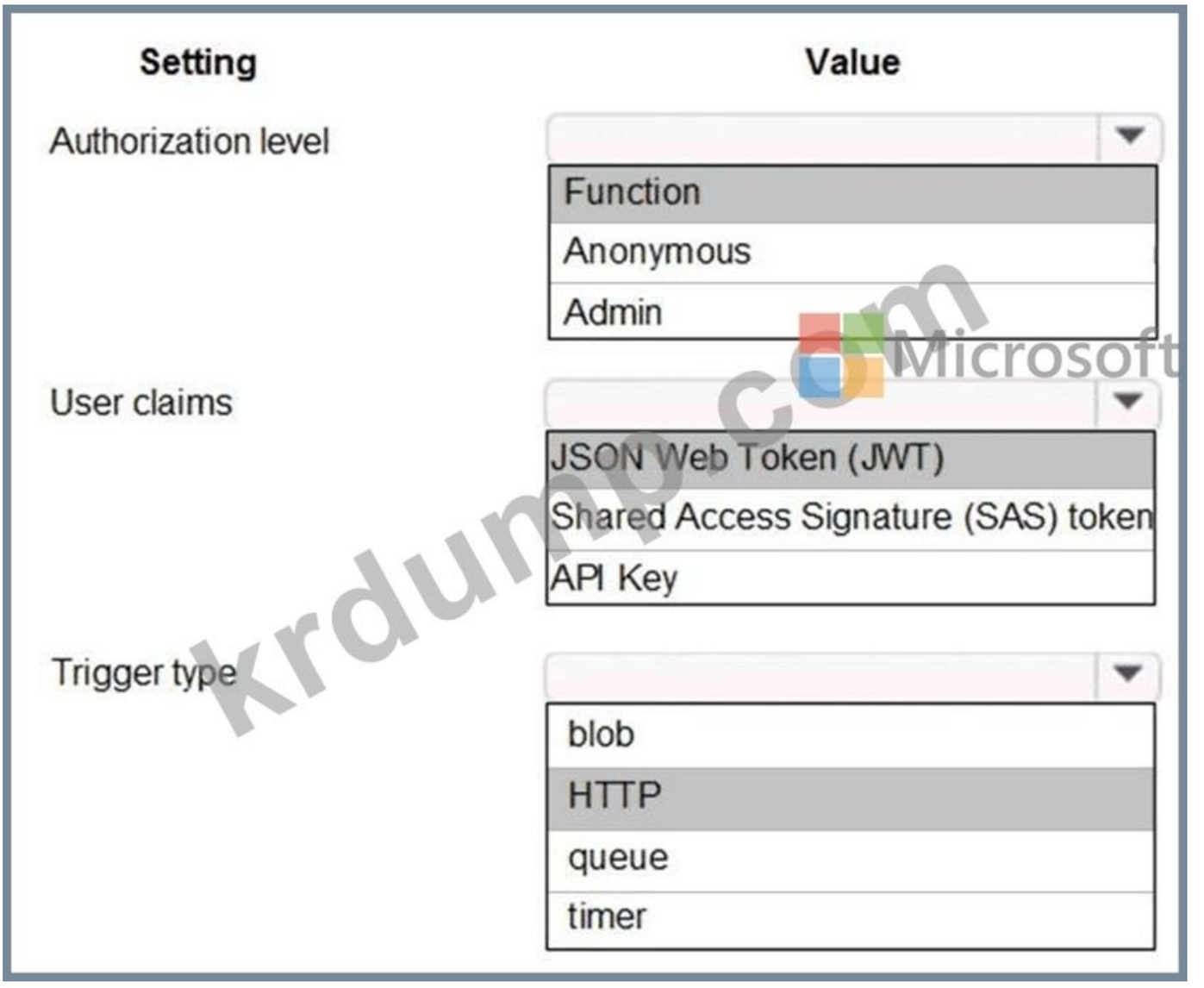
Shipping Function □□ □□□□ □□□.
□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Scenario: Shipping Function app: Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

Box 1: Function

Box 2: JSON based Token (JWT)

Azure AD uses JSON based tokens (JWTs) that contain claims

Box 3: HTTP

How a web app delegates sign-in to Azure AD and obtains a token

User authentication happens via the browser. The OpenID protocol uses standard HTTP protocol messages.

References:

https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-scenarios

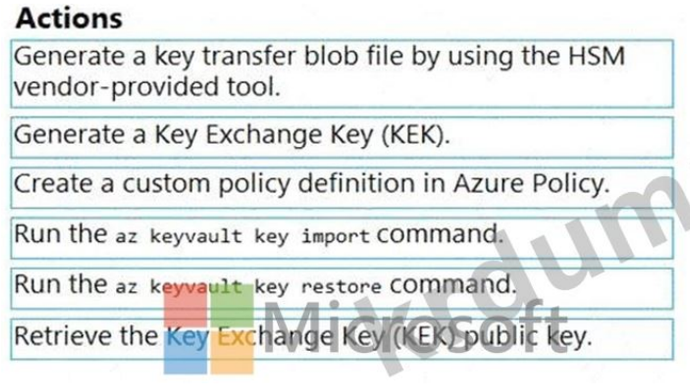
NEW QUESTION: 113

□+□□□□ HSM(□□□□ □□ □□) □□ □□□□ □□ Azure □□□ □□□□□□□ □□□ □ □□□□.

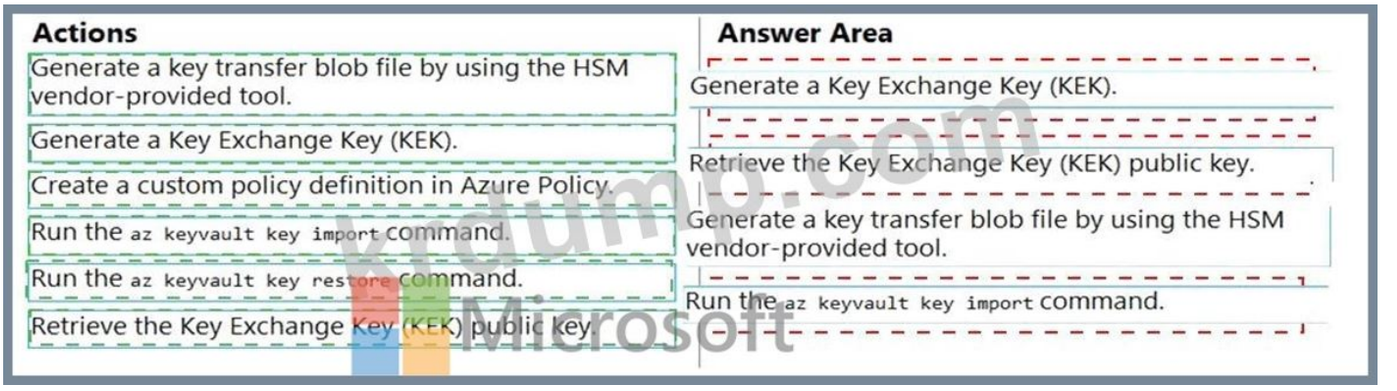
BYOK(Bring Your Own Key) □□□□□ □□□□ □□ □□ Azure Key Vault□ □□□□ □□□.

□□ Azure Key Vault□ □□□□ □□□□ □□□.

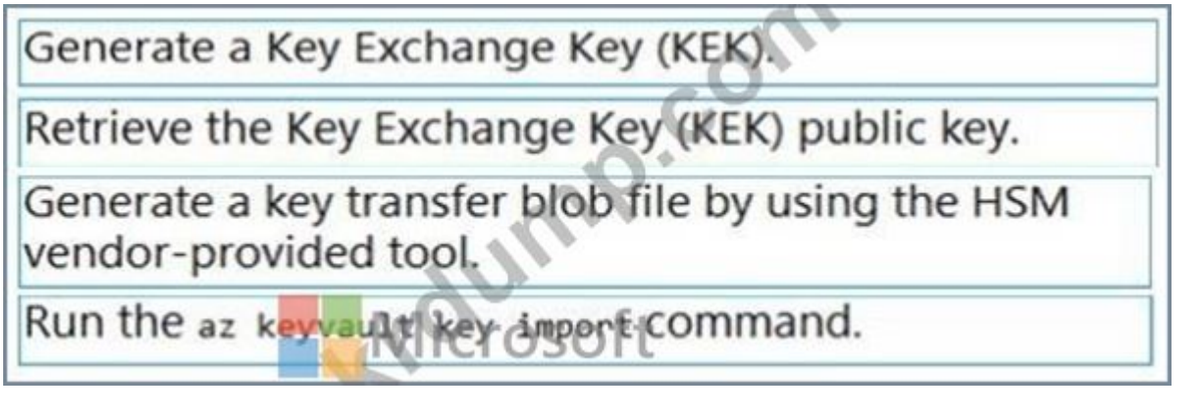
□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□ □□□□□□.



Answer:



Explanation:



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

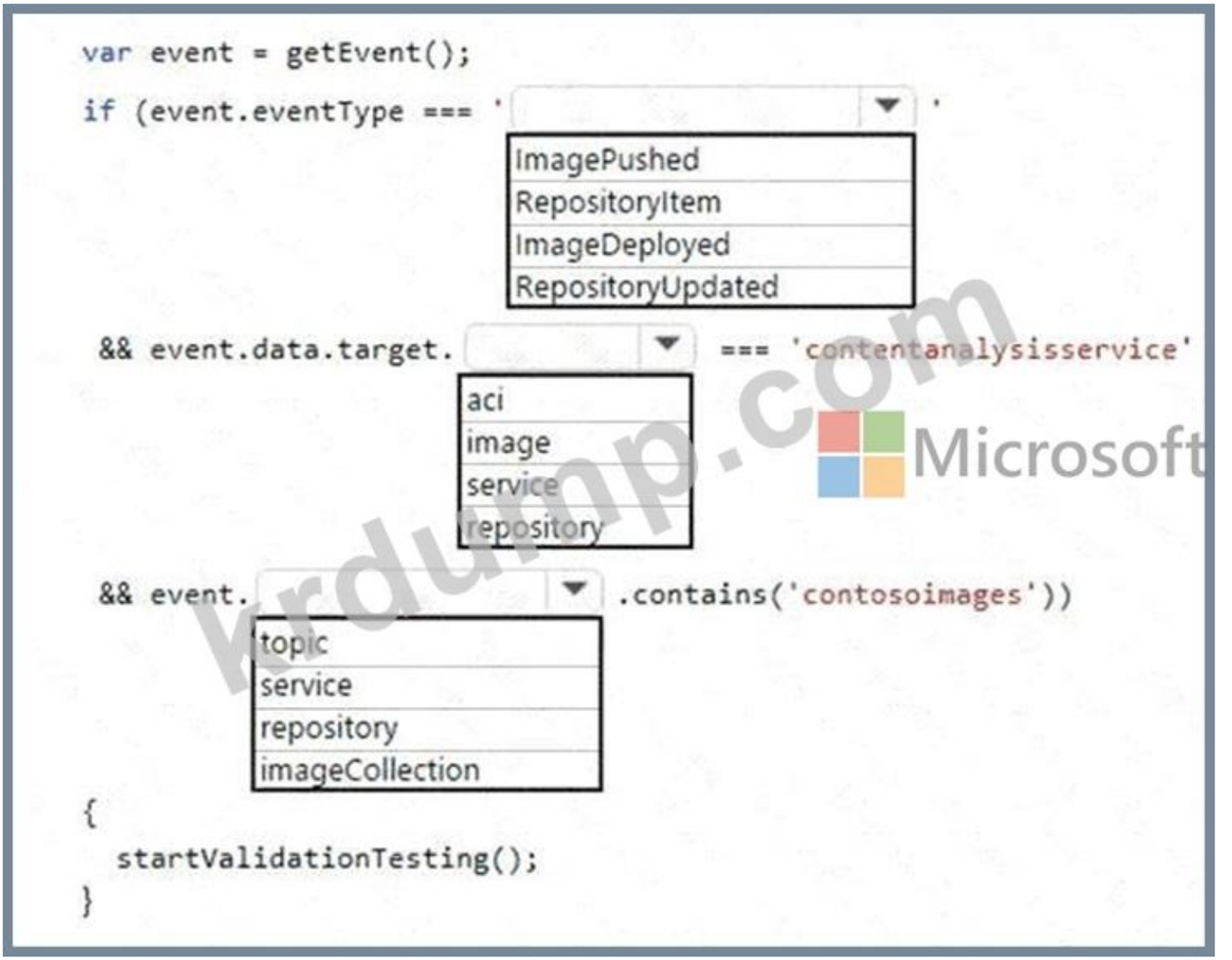
Step 4: Run the az keyvault key import command

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

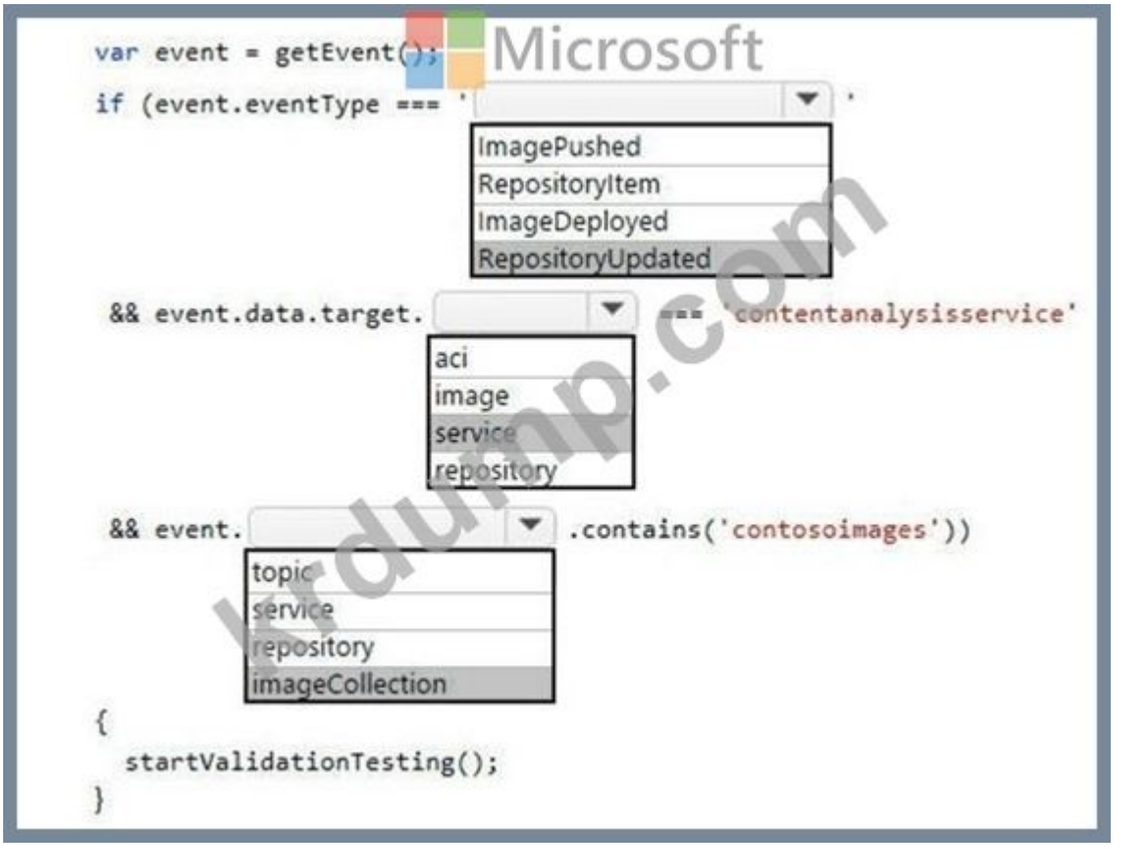
NEW QUESTION: 114
□□ □□□ □□ □□□ □□ □□□□ □□□□□□ □□□□ □□□.
□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:

```
var event = getEvent();
if (event.eventType === '                    ')
                          ImagePushed
                          RepositoryItem
                          ImageDeployed
                          RepositoryUpdated
 && event.data.target.                  === 'contentanalysisservice'
                       aci
                       image
                       service
                       repository
 && event.                    .contains('contosoimages'))
                topic
                service
                repository
                imageCollection
{
  startValidationTesting();
}
```

Explanation:



Box 1: RepositoryUpdated

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Box 2: service

Box 3: imageCollection

Reference:
https://docs.microsoft.com/en-us/azure/devops/notifications/oob-supported-event-types

NEW QUESTION: 115
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
□ □□ 2,000□ □□□□ POS(Point-of-Sale) □□ □□□□ □□□□ Azure □□□□ □□□□ □□□□. □□ □□□ 24□□□□ 2MB□ □□□□ □□□ □ □□□□. □ □□ □□□□ □□□ □ □□□□ 1~5□□ □□□ □□□□.
Azure Blob Storage□ □□□□ □□□□ □□□□ □□□. □□ □□□□ □□ □□□□ □□□ □ □□ □□□□□ □□□. □□ □□ □□ □□ □□□□□.
□□ □□□□ □□□□ □□ □□□□ □□□□ □□□.
□□ □□: Azure □□□ □□□ □□□□□□□□. □□□ □□□□ □□□ □□ □□□□ □□□ □□□□□□.
A. □
B. □□□
Answer: A (LEAVE A REPLY)
References:
https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-programming-guide

NEW QUESTION: 116
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
Azure Web App□□ □□□ □ □□□□ □□□□ □□□□. □□□□ Azure AD(Azure Active Directory) □□ □□□ □□□□ □□□□□.
□ □□□□ □□ □□□, □□ □ □□ □□ □□ □ □□□ □□□□□ □□□ □□□□□. □□ □□□ □□□□□ □□□□ Azure AD □□ □□ □□□ □□□□ □□□. □□□ □□□□ □□ □.
□□ □□: □□□ □□□ □□□□ Azure AD □□□□ □□□□□ □ □□□□ □□ Azure Web App□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: (SHOW ANSWER)
Instead in the Azure AD application's manifest, set value of the groupMembershipClaims option to

All.
References:
https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/

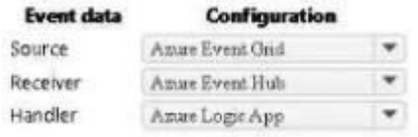
NEW QUESTION: 117
□+□□□□ □□ □□□□ Azure□ □□□□□□□□ □□□.
□□□ □□□□ □□□?
A. Azure Migrate
B. Azure Cosmos DB □□□ □□□□□□ □□(dt.exe)
C. □□□□
D. Azure □□□□□□ □□□□□□ □□□
Answer: (SHOW ANSWER)
Migrate from on-premises or cloud implementations of MongoDB to Azure Cosmos DB with minimal downtime by using Azure Database Migration Service. Perform resilient migrations of MongoDB data at scale and with high reliability.
Scenario: Data migration from on-premises to Azure must minimize costs and downtime.
The application uses MongoDB JSON document storage database for all container and transport information.
References:
https://azure.microsoft.com/en-us/updates/mongodb-to-azure-cosmos-db-online-and-offline-migrations-are-now-available/

NEW QUESTION: 118
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □ □□□□ □ □□□□ □□□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □ □ □□ □ □□□ □□ □□, □□ □□□□ □□□ □□ □□□ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□ □ □□□□. □□□ □□□ □□□ □□ □□□ □□□ □ □□□□.
Azure Container Apps □□ □□□□ □□□□ □□□ □□□ □□□□□□□.
□□□□□ □□□□ □□ □□□□ □ □□□ □□□□□. □□□ □□ □□ 0□□ □□□□□ □ □□ □□ □□□□□□.
□□□□ □□ □□□ □□□□ □□□.
□□ □□: □□□ □□□□□, TCP □□ □□□ □□□, □□ □□□ □□□□ □□ □□□□□.
□□□□ □□□ □□□□□?
A. □□□
B. □
Answer: A (LEAVE A REPLY)

NEW QUESTION: 119
□□□ □□ □□□□ □□ □□□ □□□□ □□□□ □□□.
□□ □□□ □□□□ □□□?

Answer:

Explanation:



NEW QUESTION: 120

Azure Blob Storage□ □□□ □□ □□□□ □□□□ □□□□□□□ □□□□ □□□□. □□□ □ □ □□□□ □□□ □□ □□□□□□ □□□.

□□□ □□ □□□ □□ □□□□ □□□□□ □ □□□.

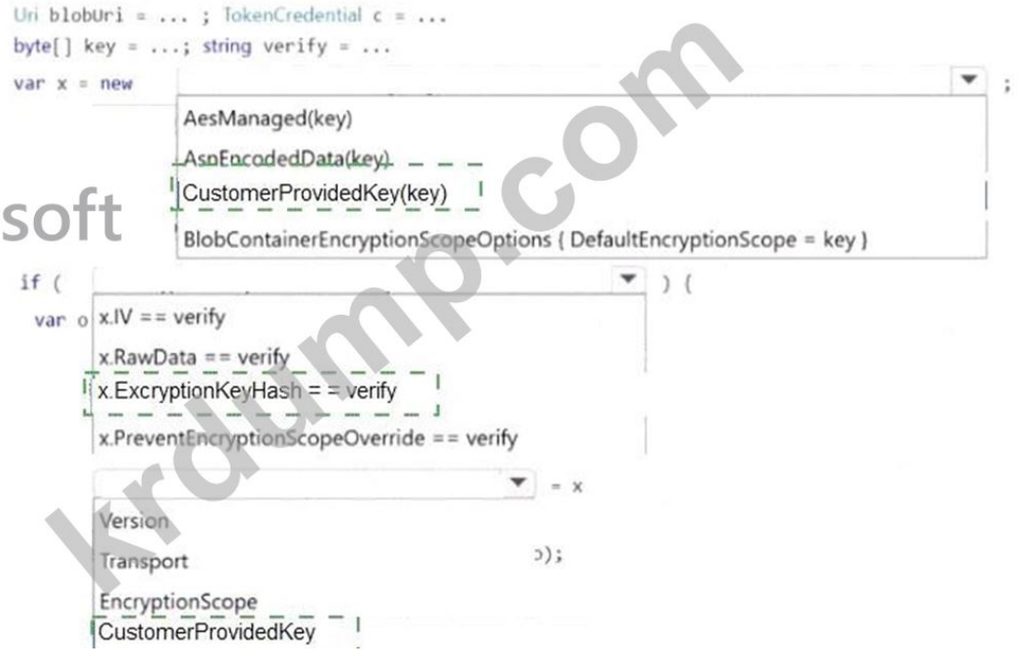
Blob□ □□□□□□□ □□□□ □□□.

□□ □□□□□ □□□ □□□□ □□□? □□□□ □□ □□□□ □□□ □□□□ □□□□□.

Answer Area

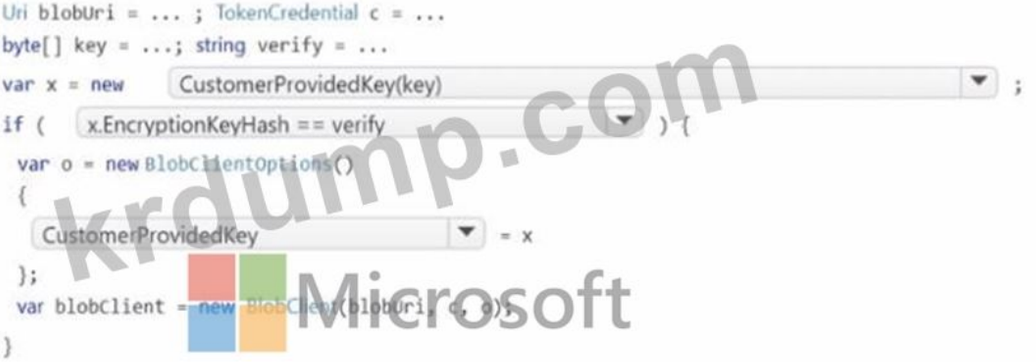
Answer:

Answer Area



Explanation:

Answer Area



NEW QUESTION: 121

.NET□ Azure Storage □□□□□ □□□□□□ □□□□ □□□□ □□□□ □□□□. □□ □ □□ □□□□. (□ □□□ □□□□□□ □□□□ □□□□.)

```
01 CloudBlockBlob src = null;
02 try
03 {
04    src = container.ListBlobs().OfType<CloudBlockBlob>().FirstOrDefault();
05    var id = await src.AcquireLeaseAsync(null);
06    var dst = container.GetBlockBlobReference(src.Name);
07    string cpid = await dst.StartCopyAsync(src);
08    await dst.FetchAttributeAsync();
09    return id;
10 }
11 catch (Exception e)
12 {
13    throw;
14 }
15 finally
16 {
17    if (src != null)
18    await src.FetchAttributesAsync();
19    if (src.Properties.LeaseState != LeaseState.Available)
20    await src.BreakLeaseAsync(new TimeSpan(0));
21 }
```

□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Statement	Yes	No
The code creates an infinite lease	○	○
The code at line 06 always creates a new blob	○	○
The finally block releases the lease	○	○

Answer:

Statement	Yes	No
The code creates an infinite lease	○	○
The code at line 06 always creates a new blob	○	○
The finally block releases the lease	○	○

Explanation:

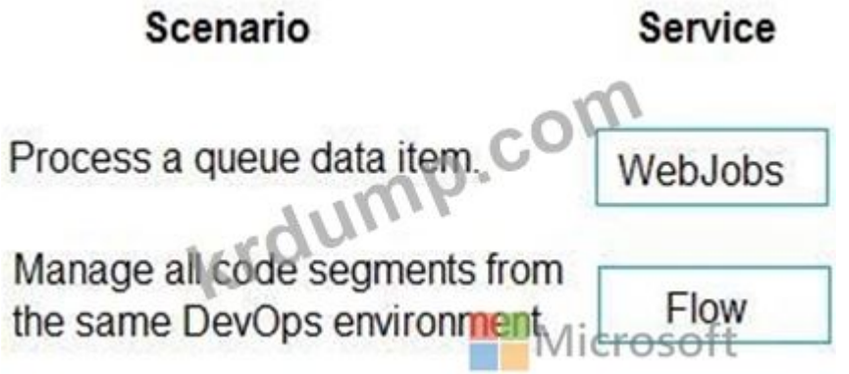
Box 1: Yes
AcquireLeaseAsync does not specify leaseTime.
leaseTime is a TimeSpan representing the span of time for which to acquire the lease, which will be rounded down to seconds. If null, an infinite lease will be acquired. If not null, this must be 15 to 60 seconds.
Box 2: No
The GetBlockBlobReference method just gets a reference to a block blob in this container.
Box 3: Yes
The BreakLeaseAsync method initiates an asynchronous operation that breaks the current lease on this container.
Reference:
https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.acquireleaseasync
https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.getblockblobreference
https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.breakleaseasync

NEW QUESTION: 122
MainApp□□□ □ □□ □□□□. WebJobs SDK□ □□□□ □□□□ App Service □□□□□ □□□ □□□□ □□□□. □ □□□ □ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□ □□□□□.
□□□□ □□□□ □□□.

□ □□□□□ □□ □□ □□□□ □□□□ □□□? □□□□□ □□□ □□□□ □□□ □□□ □□ □□□□□□□□. □ □□□□ □ □, □ □ □□ □□ □□ □□□□ □□ □ □□□□. □□□ □ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:

Explanation:



Box 1: WebJobs

A WebJob is a simple way to set up a background job, which can process continuously or on a schedule.

WebJobs differ from a cloud service as it gives you get less fine-grained control over your processing environment, making it a more true PaaS service.

Box 2: Flow

References:

https://code.msdn.microsoft.com/Processing-Service-Bus-84db27b4

NEW QUESTION: 123

□□ Azure□□ □□□□ C# .NET □ □□□□□□□ □□ □□ □□ Application Insights□ □□

□□□ □□□ □□□ □□□□□□. □□ □□□□□□□ □□□□ □□ □□□ □□□□.
□□□ □□□□□ Application Insights□ AI □□ □□ □□ □□□ □□□□ □□□.
□□□ □□ □□□?

A. □□ □□ □□□□ Azure Storage□ □□□□□ □□□□□ □□□□□.

B. □□□□□□ □□□ □□□□ □□□□□□.

C. □□□ □□□ □□□□□ □□□□□.

D. □□□□□□ □□ □□□□ □□□□ □□ □□□ □□□□ □□ □□ □□ □□ □□□ □□ □□□.

E. □□□ □□□□ □□□□□ □□□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 124

□□ □□□ □□□□□□ □□□□ □□□.
ConfigureSSE.ps1□ CS07 □□ □□ □□□ □□□□□?

A. -PermissionsToKeys □□, □□□, □□

B. -PermissionsToCertificates □□, □□□, □□

C. -PermissionsToCertificates wrapkey, unwrapkey, get

D. -PermissionsToKeys wrapkey, unwrapkey, □□□□

Answer: B (LEAVE A REPLY)

Scenario: All certificates and secrets used to secure data must be stored in Azure Key Vault.
You must adhere to the principle of least privilege and provide privileges which are essential to perform the intended function.
The Set-AzureRmKeyValutAccessPolicy parameter -PermissionsToKeys specifies an array of key operation permissions to grant to a user or service principal. The acceptable values for this parameter: decrypt, encrypt, unwrapKey, wrapKey, verify, sign, get, list, update, create, import, delete, backup, restore, recover, purge Reference:
https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/set-azurermkeyvaultaccesspolicy
Topic 6, Coho Winery
Case study
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. When you are ready to answer a question, click the Question button to return to the question.

LabelMaker app

Coho Winery produces, bottles, and distributes a variety of wines globally. You are a developer implementing highly scalable and resilient applications to support online order processing by using Azure solutions.

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs).

Coho Winery plans to move the application to Azure and continue to support label creation. External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Requirements. Data

You identify the following requirements for data management and manipulation:

Order data is stored as nonrelational JSON and must be queried using SQL.

Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

Requirements. Security

You have the following security requirements:

Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.

External partners must use their own credentials and authenticate with their organization's identity management solution.

External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.

Storage of e-commerce application settings must be maintained in Azure Key Vault.

E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).

Conditional access policies must be applied at the application level to protect company content.

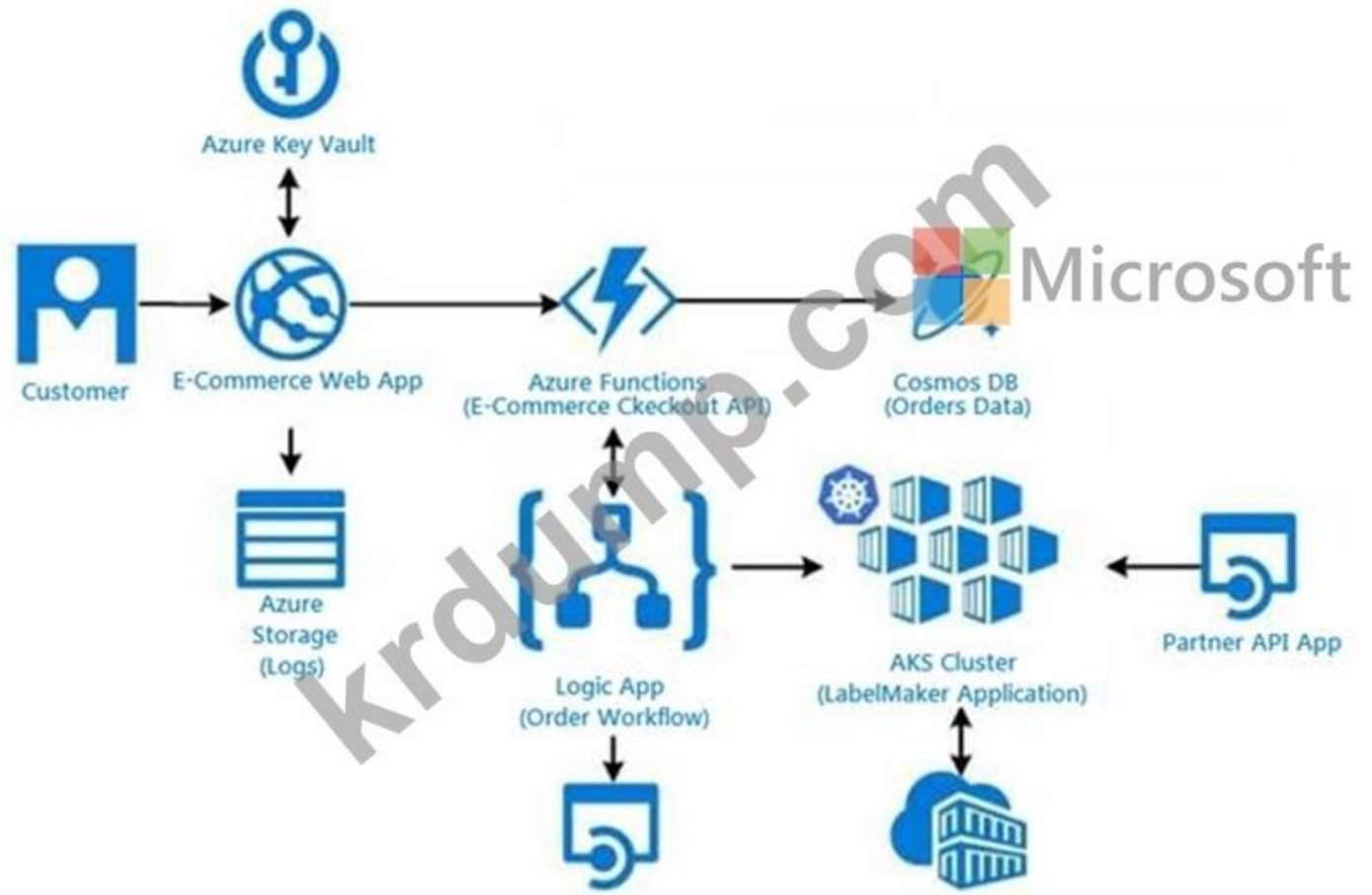
The LabelMaker application must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

Requirements. LabelMaker app

Azure Monitor Container Health must be used to monitor the performance of workloads that are deployed to Kubernetes environments and hosted on Azure Kubernetes Service (AKS).

You must use Azure Container Registry to publish images that support the AKS deployment.

Architecture



Issues

Calls to the Printer API App fail periodically due to printer communication timeouts.

Printer communication timeouts occur after 10 seconds. The label printer must only receive up to 5 attempts within one minute.

The order workflow fails to run upon initial deployment to Azure.

Order.json

Relevant portions of the app files are shown below. Line numbers are included for reference only.

This JSON file contains a representation of the data for an order that includes a single item.

Order.json

```
{
  “id” : 1,
  “customers” : [
  {
     “familyName” : “Doe”,
     “givenName” : “John”,
     “customerid” : 5
   }
  ],
  “line_items” : [
  {
     “fulfillable_quantity” : 1,
     “id” : 6,
     “price” : “199.99” ,
     “product_id” : 7513594,
     “quantity”: 1,
     “requires_shipping” : true ,
     “sku” : “SFC-342-N” ,
     “title”: “Surface Go” ,
     “vendor” : “Microsoft” ,
     “name” : “Surface Go - 8GB” ,
     “taxable” : true ,
     “tax_lines” : [
     {
       “title” : “State Tax” ,
       “price” : “3.98” ,
       “rate” : 0.06
      }
     ],
     “total_discount” : “5.00” ,
     “discount_allocations” : [
      {
        “amount” : “5.00” ,
        “discount_application_index” : 2
        }
     ]
   }
  ],
  “address” : {
  “state” : “NY” ,
  “state”: “Manhattan” ,
  “city” : “NY”
  }
}
```

NEW QUESTION: 125

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

□□□ □□□□□□□□ □□□□ □□□□ □ □□□□ □□□□ Azure □□□ □□□□□□□ □□□□ □□□□. □□□□ □□□□ □□□□ □□□□ □□ □ □□□□.

□□□ □□ □□ □□□ □□□□.

* □□□ □□□ 80□□□□□(GB)□□ □□ □ □□□□.

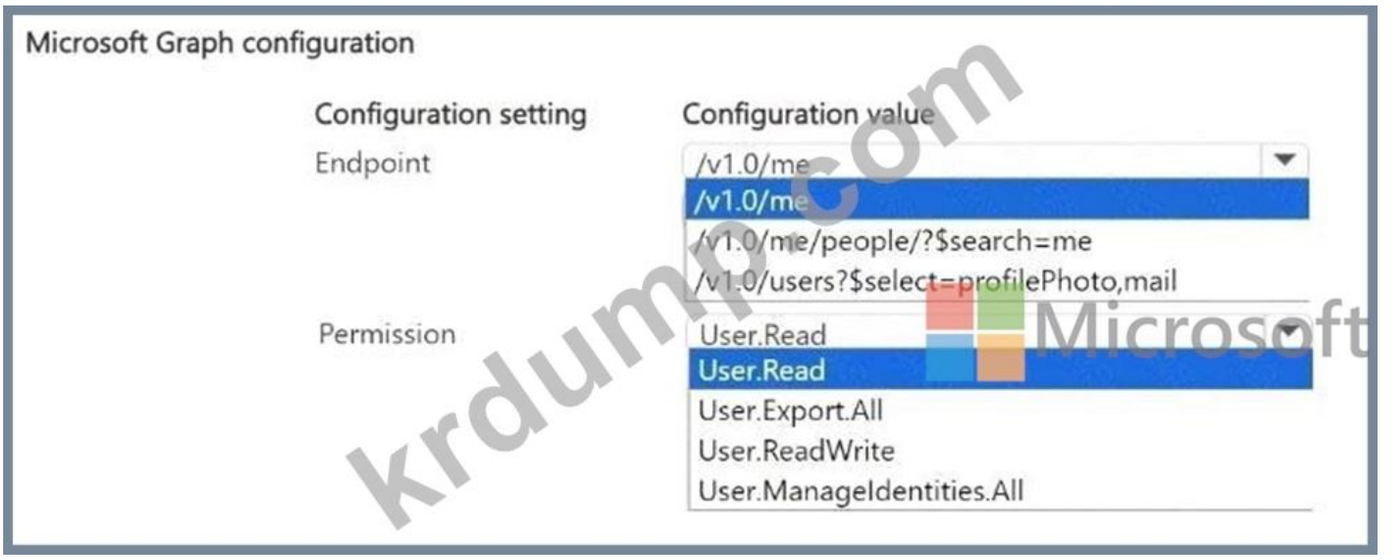
* □□□□ □□□□(FIFO) □□□ □□□□□.

* Azure □□□ □□□□□□.

□□□ □□□□ □□□□ □□□.
□□ □□: .Net API□ □□□□ □□□ □□□□□□□□ Azure Service Bus □□ □□□□ □□ □□□.
Azure Service Bus □ □□□□ □□□□ Azure □□ □□ □□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: A (LEAVE A REPLY)
You can create a function that is triggered when messages are submitted to an Azure Storage queue.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function

NEW QUESTION: 126
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
AKS(Azure Kubernetes Service) □□□□□ □□□ □□□□ □□□□ □□□□. □ □□□□□ □□□ □□ VNet, Azure Container Registry □□□ □ Azure Storage □□□ □□□□□.
□□□□ AKS □□□□ □□□ □□ Azure □□□□ □□ □□ □ □□□ □□□□ □□□.
Azure API□ □□ □□□ AKS □□□□□ □□□□ □□□.
□□□: Kubernetes□ Azure Policy □□ □□□ □□□□□ □□□□ Azure Policy □□□□ AKS □□□□□ GateKeeper □□ □□□□□ □□□□□. □□ □□ □□□ □□□□□ □□□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Instead create an AKS cluster that supports network policy. Create and apply a network to allow traffic only from within a defined namespace References:
https://docs.microsoft.com/en-us/azure/aks/use-network-policies

NEW QUESTION: 127
□□□□□ □□□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□□.
□□ Microsoft Graph □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□ □□.
□□: □□ □□□ 1□□□□.



Answer:

Explanation:

NEW QUESTION: 128

□□ Azure Functions□ □□□□ □□□□ □□□□□□□ □□□□□. □□□ □□□ □□ □□ □ □□□□ □□□□□.

Azure □□ □ □□□□□ □□ □□□ □□□□□ □□□.

hostjson □□□□ □□ □□□ □□□□ □□□.

□□ □□□ □□□□ □□□?

A. Azure □□

B. Azure PowerShell

C. Azure Functions □□ □□(Azure CLI)

D. □□□ □□□□

Answer: A (LEAVE A REPLY)

The function editor built into the Azure portal lets you update the function.json file and the code file for a function. The host.json file, which contains some runtime-specific configurations, is in the root folder of the function app.

References:

https://docs.microsoft.com/en-us/azure/azure-functions/functions-reference#fileupdate

NEW QUESTION: 129

□□□ □□ □ □□□□ □□□□ SaaS □□□ □□□□□□.

□□□ □□ □ □□□□ □□ □□ □□□ □□□□ □□□.

* API Management□ □□□□ □□□□ □□□

* □□□ □□ OpenID Connect □□

* □□ □□ □□

□□ □□ □□□□ □□ □□ □□ □ □□□□ □□□ □ □□ □□□ □□□□□□.

□□□□ □□ API Management □□□ □□□□□?

A. jsonp

B. □□-□□□

C. □□ □□

D. □□□ □□-jwt

Answer: D (LEAVE A REPLY)

Add the validate-jwt policy to validate the OAuth token for every incoming request.

Reference:

https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad

NEW QUESTION: 130

Visual Studio□ □□□□ Azure □□ □□ □□□□ □□□□. □□ Azure Web App□□ □□□ □□□ □□□□□. □□□ □□ □□□ Azure Queue Storage□ □□□□□.

□□ □□□ Azure □□ □ □□□ □□□□ □□□.

```
public static class OrderProcessor
{
  [FunctionName("ProcessOrders")]
  public static void ProcessOrders([QueueTrigger("incoming-orders")]CloudQueueMessage myQueueItem, [Table("Orders")]ICollector<Order> tableBindings, TraceWriter log)
  {
    log.Info($"Processing Order: {myQueueItem.Id}");
    log.Info($"Queue Insertion Time: {myQueueItem.InsertionTime}");
    log.Info($"Queue Expiration Time: {myQueueItem.ExpirationTime}");
    tableBindings.Add(JsonConvert.DeserializeObject<Order>(myQueueItem.AsString));
  }
  [FunctionName("ProcessOrders-Poison")]
  public static void ProcessFailedOrders([QueueTrigger("incoming-orders-poison")]CloudQueueMessage myQueueItem, TraceWriter log)
  {
    log.Error($"Failed to process order: {myQueueItem.AsString}");
    . . .
  }
}
```


□□: □ □□□ □□□ 1□□ □□□ □□□□.

	Yes	No
The code will log the time that the order was processed from the queue.	○	○
When the ProcessOrders function fails, the function will retry up to five times for a given order, including the first try.	○	○
When there are multiple orders in the queue, a batch of orders will be retrieved from the queue and the ProcessOrders function will run multiple instances concurrently to process the orders.	○	○
The ProcessOrders function will output the order to an Orders table in Azure Table Storage.	○	○

Answer:

	Yes	No
The code will log the time that the order was processed from the queue.	○	○
When the ProcessOrders function fails, the function will retry up to five times for a given order, including the first try.	○	○
When there are multiple orders in the queue, a batch of orders will be retrieved from the queue and the ProcessOrders function will run multiple instances concurrently to process the orders.	○	○
The ProcessOrders function will output the order to an Orders table in Azure Table Storage.	○	○

Explanation:

	Yes	No
he code will log the time that the order was processed from the queue.	○	○
Vhen the ProcessOrders function fails, the function will retry up to five imes for a given order, including the first try.	○	○
Vhen there are multiple orders in the queue, a batch of orders will be etrieved from the queue and the ProcessOrders function will run nultiple instances concurrently to process the orders.	○	○
The ProcessOrders function will output the order to an Orders table n Azure Table Storage	○	○

Box 1: No

ExpirationTime - The time that the message expires.

InsertionTime - The time that the message was added to the queue.

Box 2: Yes
maxDequeueCount - The number of times to try processing a message before moving it to the poison queue.
Default value is 5.
Box 3: Yes
When there are multiple queue messages waiting, the queue trigger retrieves a batch of messages and invokes function instances concurrently to process them. By default, the batch size is 16. When the number being processed gets down to 8, the runtime gets another batch and starts processing those messages. So the maximum number of concurrent messages being processed per function on one virtual machine (VM) is 24.
Box 4: Yes
References:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-queue

NEW QUESTION: 131
All□□□ Application Insights □□□□ Appl□□□ Azure App Service □□□ □□□ Azure □□ □ □□□□. AM□□ □□ □□□ □□□□ □□□ URL□ App1□ □□□□□.
□□□□ □□□□ □□ □□□□□ □□□ □□□ □□□□□ □□ □□□. □□□ □□ □□□?
A. □□□ □□□
B. □□□ □□□□□□.
C. □□ □□□ □□□□.
D. □□ □□□ □□□□.
Answer: D (LEAVE A REPLY)

NEW QUESTION: 132
Azure B2B □□□ □□□□ □□□ □□□□ B2B □ □□□□□□□ □□□□ □□□□. □□ □ □□ □□□□□□ □□□□ Azure B2B□ □□□□□. □ □□□□□□□ □□ □□□□ □□ □ □□ □□□ □□□□ □□□ □□□ □□□ □ □□□□. □□□□ □□ □□□□ □□□□ □□ □□ □□□ □□□□ □□□□□ □□□ □□□□ □□□□□□ □□□□ □□□□ □□□. □□ □□□□ □□□□ Azure Active Directory(Azure AD)□□ □□□□ □□□□□□ □□□. □□ Graph API □□□□□ □□□□ □□□ □□□□ □□□□□□□ □□□ □□□□□?
A. uscrFlowType
B. □□
C. □□□Ustr
D. □□□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 133
□□□ □□□□ Azure App Service□ □□□□□. Azure App Service□ □□□□ □□ □□□ □ □□□□.
□□□□ □ □□ □□□□ □□□ □□ □□□ □□ □□ □□ □□ □□□ □□□□ □□□ □□

□□□. □□ □□ □□□□ □ □□□□ □□ □□ □□□□□ □□□. □ □□□ □□□□□ □□ □ □□□ □□□ □□□□□ □□□.
□□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□. □□: □ □□□ □ □□ □□□ □□□□.
A. Application Insights □□□ □□□ □□□□□□.
B. □□ □□□□ □□□□□□.
C. Azure App Service □□□ Premium□□ □□□□□
D. □ □□□□ Always On □□□ □□□□□□.
E. □□□ □□□□ □□□□□□.
F. □□□□□□ □□□□□□.
G. App Service □□□ □□ □□ □□ □□□□□.
Answer: D,E,F (LEAVE A REPLY)

NEW QUESTION: 134
□□□ □□□ □□ □□□ □□□ □□□□ □□□ □□□ □□□ □□□□□□ □□□□ □ □ □□□□□□ □□□□ □□□□. □□□ □□□ □□□ □□□ □□□ □□□□ □□□ □□ □ □□. □□ □□□□ □□ □□□ □□□ □□□ Azure Redo Cache□ □□□ □□□□□.
Azure □□□ □□ □□ □ □□□□□ □□□ □□□□□ □□□□ □□□.
Azure Redis □□ □□□□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □□□ □□□ □□□ 1□□ □□□ □□□□.
A. □□□□ □□□□ Azure Redis □□
B. RDS □□□□ □□□□ Azure Redis □□
C. □□□□ □□ □ □□ □□□ □□ □□
D. □□ □□□ □□□□□ □□
Answer: (SHOW ANSWER)

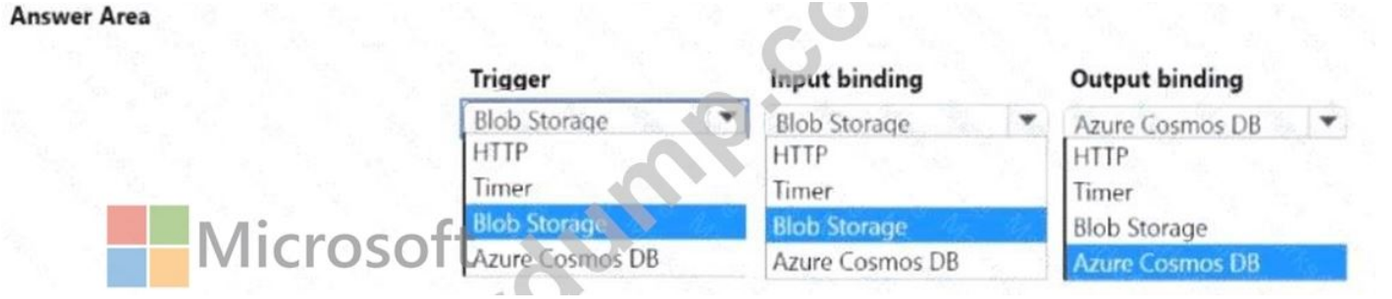
NEW QUESTION: 135
Azure □□ □□□□□ □□□□ □□ □□□□ □□□ □□□□ □□□. □□□□□ 30□□ □□ □□□ □□□□□. □□ □□□ □□□ 4□□ □□ IP □□ □□ □□□□ □□□ □□□□ □□ □□□□□ □□□ □□□□ □□ □□□ □□ □□□□ □□□ □□□□□.
□□□□□ □□□ □□□□□ □□□?
A. Azure □□□□ □□
B. Azure □□ WAN
C. NAT □□□□□
D. □□□ □□ □□
Answer: B (LEAVE A REPLY)
NEW QUESTION: 136
□□□ □□ □□□ □□ □□□ □□□□ □□□□□□□ □□□□ □□□□.
□□□□ □□ □□□□□ □□□□ □□, □□□ □□□□ □□□ □ □□□□ □□□ □□□□

□ □□□ □ □□□ □□□.
□□□□□□□ □□ Azure Cosmos DB API□ □□□□ □□□?
A. □□□□
B. □□
C. □□□
D. □□□ API
E. □□□□
Answer: E (LEAVE A REPLY)

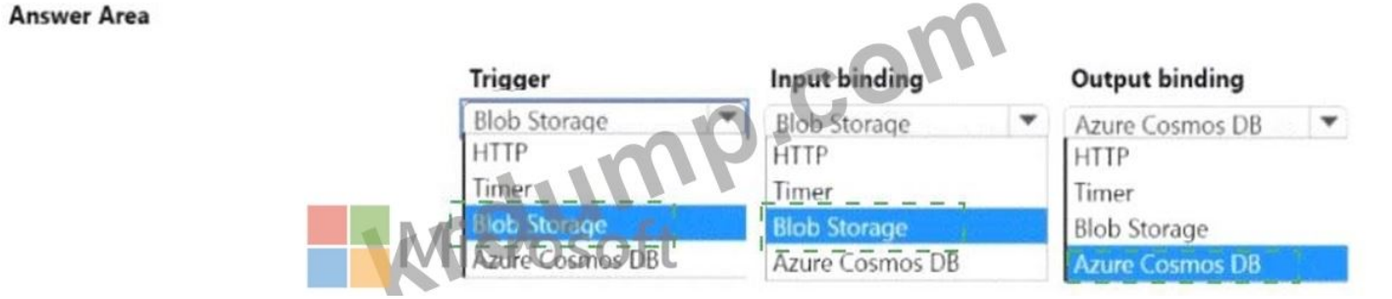
NEW QUESTION: 137
□□ APIM(Azure API Management) □□□ API□ □□ □□□□.
APIM□□ API□ □□ □□□ □□□□ □□□. REST □□□□□□ □□□□ APIM□ □□ □□ □ □□□□□ □□□. □□ □□□□ □□ □□□ □□□□□ □□□.
* API □□□□ □□□ APIM□ □□□ □□□ □□□□□.
* □□□□□ □□□ □□ □□□ □□ □□□ □□□□ □□
* □□ □□ □ □□□ □□ □ □□ □□□ □□□□ □□
* □□□□□□ □□ □□ □□ APIM□ □□□ □□□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □□□ □□□ □□ 1□□ □□□ □□□□.
A. API□ □□□□ □ □□□□ □□□ □□ □□ □□ □□□ □□□□□□.
B. □□ true□ □□□ API □□□ Ocp-Apim-Trace □□ □□ □□□□□.
C. API□ □□ □□□□ □□□□ □□□ □□ Ocp-Apim-Subscription-Key □□ □□ □□□□□.
D. □□□ □□ □□□ □□□□ □□□□□. API □□□ □□□□ □□□□□ □□ □□□ □□□ □□□□□.
E. □□□ □□ □□□ □□□□ □□□□□. □□ □□□ □□□□ □□□□ □□ □□□ □□□ □□□□□.
Answer: A,B,D (LEAVE A REPLY)
The correct answer is A, B, and C. To inspect request processing of the APIs in APIM, you need to do the following three actions:
Enable the Allow tracing setting for the subscription used to inspect the API. This setting allows you to trace request processing in APIM using the test console, a REST client, or a client app. You can enable this setting in the portal by selecting Subscriptions and then selecting the subscription you want to use for debugging1.

Add the Ocp-Apim-Trace header value to the API call with a value set to true. This header triggers tracing when making requests to APIM using a REST client or a client app. You also need to add the Ocp-Apim- Subscription-Key header value to the key for a subscription that allows access to the API1.
Add the Ocp-Apim-Subscription-Key header value to the key for a subscription that allows access to the API.
This header authenticates your request and grants you access to the API. You can find the key for your subscription in the portal by selecting Subscriptions and then selecting Show/hide keys1.
You do not need to create and configure a custom policy for tracing request processing. The trace policy is used to add a custom trace into the request tracing output, Application Insights telemetries, and/or resource logs2. It is not required for inspecting the APIs.

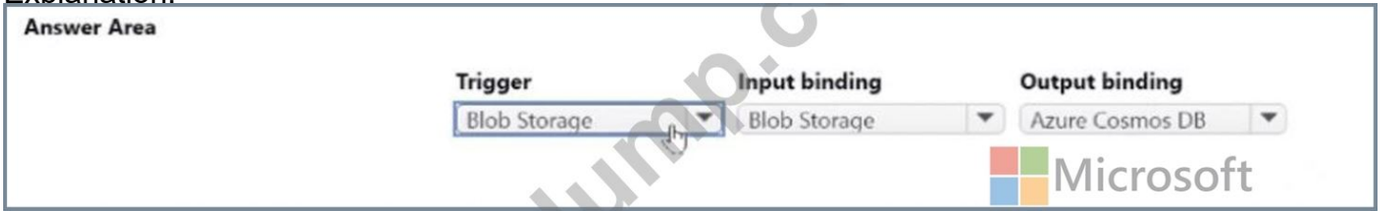
NEW QUESTION: 138
Azure □□ □□ □□ □□□□.
Azure □□ □□ WebHook□ □□□□□ Azure Blob Storage□□ □□□□ □□ □ Azure Cosmos DB □□□ □□□□ □□□.
Azure □□ □□ □□□□ □□□.
□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□: □□□ □□□ □□ 1□□ □□□ □□□□.



Answer:



Explanation:



NEW QUESTION: 139
Azure API Management□ □□ API□ □□□□ □□□□□.

APICounts□□ □□ □□ □□□ □□□□□.

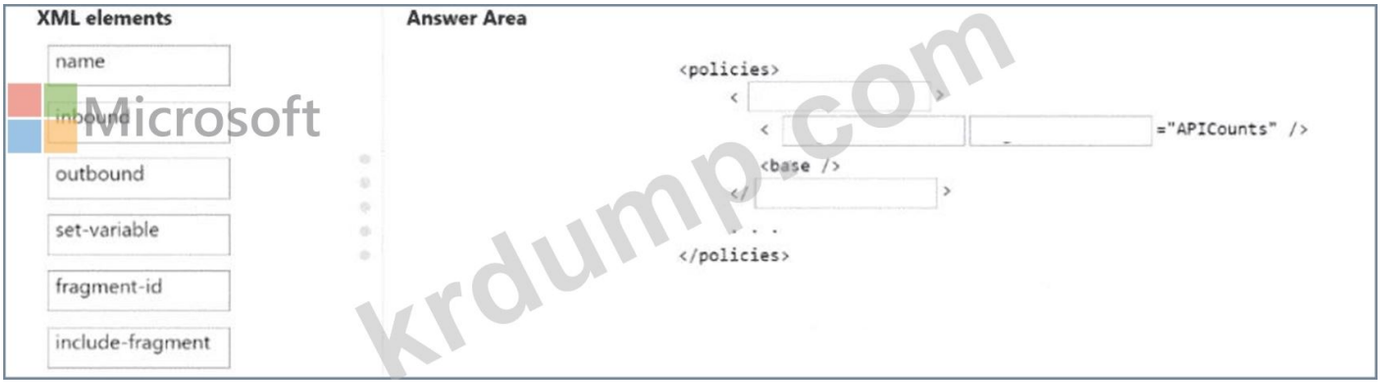
```
<fragment>
    <emit-metric value="1" namespace="custom-metrics">
        <dimension name="User ID" />
        <dimension name="Operation ID" />
        <dimension name="API ID" />
        <dimension name="Client IP" value="@(context.Request.IpAddress)" />
    </emit-metric>
</fragment>
```

□□ □□□ □□□ □□□ API□□ □□□□□□ □□□. □□ □□□ □□ API□ □□□□□ □ □ □□ □□□□ API□ □□□ □ □□□□□ □□□.

□□ □□□ □□□□ □□□.

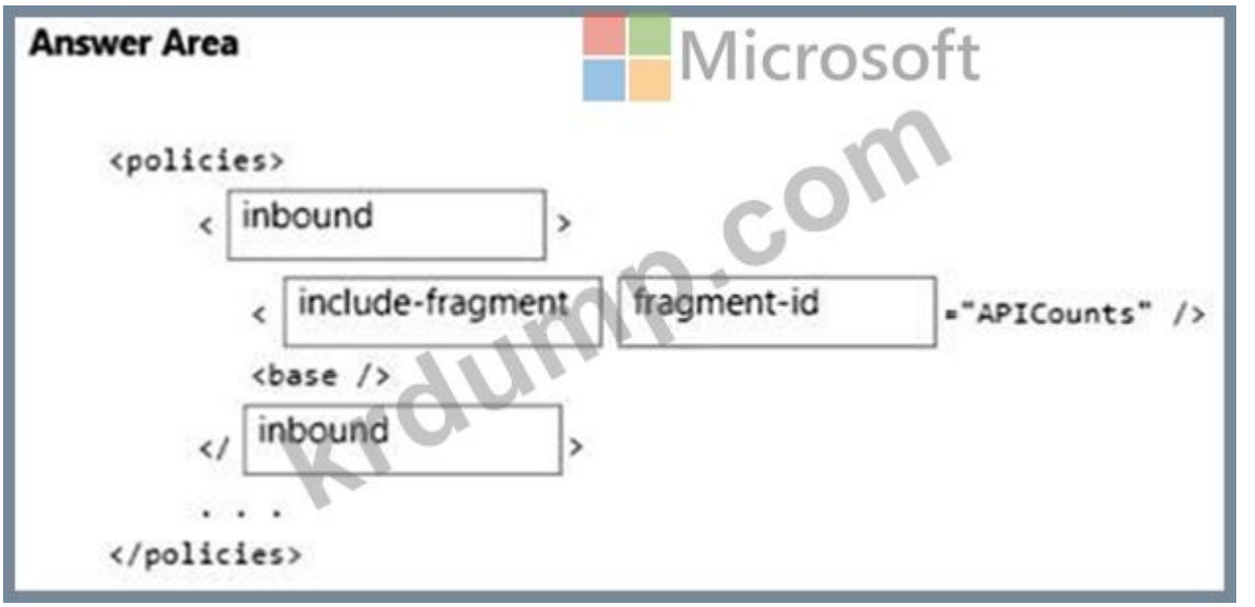
□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□□ XML □□□ □□□ □□□□ □□ □□□□. □ XML □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □□ □□□□.

□□: □□□ □□□ □□ 1□□ □□□ □□□□.



Answer:

Explanation:



https://learn.microsoft.com/en-us/azure/api-management/include-fragment-policy

NEW QUESTION: 140
□□ □□□□□ □□ □□□ □□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□□□: □□ □□□ 1□□□□.
A. Azure App Configuration □□□□ □□ □□ □□□ □□□□□□□.
B. Azure □ □□ □□□□ □□□□.
C. Azure Key Vault□ □□ □□ □□□ □□□□□□□.
D. Azure Key Vault□ □□□□.
E. □□□□ ID□ □□□□.
Answer: C,D,E (LEAVE A REPLY)

NEW QUESTION: 141
Application Insights SDK□ □□□□ □□□□ ASP.Net Core □□□□□□□ □□□□□.
□ □□□□□□□ □□ □□ □□□ □□ □□ □□□ □□ □□ □□□□ □□□□□. □□□□ □□□, □□ □ □□□ □□ □□□□ □□ □□ □□□ □□ □□□□ □□□□ □□ □□□ □ □□□□. □□ □□ □□□□ HTTP □□ □ □□ □□ □□□ □□□□ □□ □□□□□ □ □□ □□□ □□□□□ □□□.
□□□□□□ □□ □□ □□□□ □□ □□□□□ □□□ □□□ □□□□ □□□ □□ □□ □ □□, □□□ □□ □ □□ □□□ □□□ □□□.
□□□ □□ □□□?
A. Log Analytics □□ □□□□ □□□□□ □□ □ □□ □□□ □□□□□. □□□ □□□□ □ □□□□.
30□.
B. Log Analytics □□ □□□ □□ □□ □□□ □□□□□.
C. Log Analytics □□ □□□ □□ □□□ □□□□□. □□ □□ □□ □□□ □□□□.
D. □□ □□□□ □□□□□ □□□ □□□□□. □□ maxTelemetrylterns □□ □□□□□.

Answer: D (LEAVE A REPLY)

NEW QUESTION: 142

Azure□□ □□□□□ □□ □□□ □ □□□□□□□ □□□□ □□□□. □□□□□□□ Azure Cosmos DB□ □□□□ □□ □□□ □□□□□. □□ SDK□ □□□□ □□□□□□□ □□ □ □□ □□□□ □□□□.

□ Azure Cosmos DB □□□□□ □□□□. □□ □□□□□ appSettings.json □□□ □□□ □ □□□□□ □□□ □□ □□ □□ □□□□□.

□□ □□□□□□ □□□ □□□□ □□□□. (□ □□□ □□□□□□ □□□□ □□□□.)

```
01 using System;
02 using System.Threading.Tasks;
03 using Microsoft.Azure.Cosmos;
04 using Microsoft.Extensions.Configuration;
05 using Newtonsoft.Json;
06 namespace SalesOrders
07 {
08   public class SalesOrder
09   {
10     . . .
11   }
12   internal class ManageSalesOrders
13     {
14       private static async Task GenerateSalesOrders()
15         {
16           IConfigurationRoot configuration = new ConfigurationBuilder().AddJsonFile("appSettings.json").Build();
17           string endpoint = configuration["EndPointUrl"];
18           string authKey = configuration["AuthorizationKey"];
19           using CosmosClient client = new CosmosClient(endpoint, authKey);
20           Database database = null;
21           using (await client.GetDatabase("SalesOrders").DeleteStreamAsync()) { }
22           database = await client.CreateDatabaseIfNotExistsAsync("SalesOrders");
23           Container container1 = await database.CreateContainerAsync(id: "Container1", partitionKeyPath: "/AccountNumber");
24           Container container2 = await database.CreateContainerAsync(id: "Container2", partitionKeyPath: "/AccountNumber");
25           SalesOrder salesOrder1 = new SalesOrder() { AccountNumber = "123456" };
26           await container1.CreateItemAsync(salesOrder1, new PartitionKey(salesOrder1.AccountNumber));
27           SalesOrder salesOrder2 = new SalesOrder() { AccountNumber = "654321" };
28           await container1.CreateItemAsync(salesOrder2, new PartitionKey(salesOrder2.AccountNumber));
29           SalesOrder salesOrder3 = new SalesOrder() { AccountNumber = "109876" };
30           await container2.CreateItemAsync(salesOrder3, new PartitionKey(salesOrder3.AccountNumber));
31           _ = await database.CreateUserAsync("User1");
32           User user1 = database.GetUser("User1");
33           _ = await user1.ReadAsync();
34       }
35   }
36 }
```

□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Statements	Yes	No
A database named SalesOrders is created. The database will include two containers.	○	○
Container1 will contain two items.	○	○
Container2 will contain one item.	○	○

Answer:

Statements	Yes	No
A database named SalesOrders is created. The database will include two containers.	○	○
Container1 will contain two items.	○	○
Container2 will contain one item.	○	○

Explanation:

Statements	Yes	No
A database named SalesOrders is created. The database will include two containers.	○	○
Container1 will contain two items.	○	○
Container2 will contain one item.	○	○

Box 1: Yes
The createDatabaseIfNotExistsAsync method checks if a database exists, and if it doesn't, create it.
The Database.CreateContainerAsync method creates a container as an asynchronous operation in the Azure Cosmos service.
Box 2: Yes
The CosmosContainer.CreateItemAsync method creates an item as an asynchronous operation in the Azure Cosmos service.
Box 3: Yes
Reference:
https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.cosmosclient.createdatabaseifnotexistsasync
https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.database.createcontainerasync
https://docs.microsoft.com/en-us/dotnet/api/azure.cosmos.cosmoscontainer.createitemasync

NEW QUESTION: 143
□□□□□ □□□□ Python □ □□□□ Azure Web App□ □□□ □□□ □□ □□□□. □□□ □ □□□ □□□□ □□□□ □□ □□□□□ □□□□□. □□□□□ □□□□ Dockerfile□ □ □□ □□□□.

```
FROM python:3
ADD website.py
CMD [ "python", "./website.py"]
```

□□ □□□ □□□□ □□□□□ □□□□□. □□□□□ Azure Container Registry □□□□□ □□□□ □□□□□□□□.

```
docker build -t images.azurecr.io/website:v1.0.0
```

□□□□□□ □□□ □□□ □□□ admin□□□.

□ □□ □□ □□□ □□□□ □□ □□□ □□□ □ □□□□ □□□□ □□□.

□ □□□□ □□□□□ Azure Web App□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
az configure --defaults web=website
az configure --defaults group=website
az appservice plan create --name websitePlan [ ▼ ]
```

- --sku SHARED
- --tags container
- --sku B1 --hyper-v
- --sku B1 --is-linux

```
az webapp create --plan websitePlan [ ▼ ]
```

- --deployment-source-url images.azurecr.io/website:v1.0.0
- --deployment-source-url images.azurecr.io/website:latest
- --deployment-container-image-name images.azurecr.io/website:v1.0.0
- --deployment-container-image-name images.azurecr.io/website:latest

```
az webapp config [ ▼ ]
```

- set --python-version 2.7 --generic-configurations user=admin password=admin
- set –python-version 3.6 --generic-configurations user=admin password=admin
- container set --docker-registry-server-url https://images.azurecr.io –u admin –p admin
- container set --docker-registry-server-url https://images.azurecr.io/wsebsite -u admin -p admin

Answer:

```
az configure --defaults web=website
az configure --defaults group=website
az appservice plan create --name websitePlan [ ▼ ]
```

- --sku SHARED
- --tags container
- --sku B1 --hyper-v
- --sku B1 --is-linux

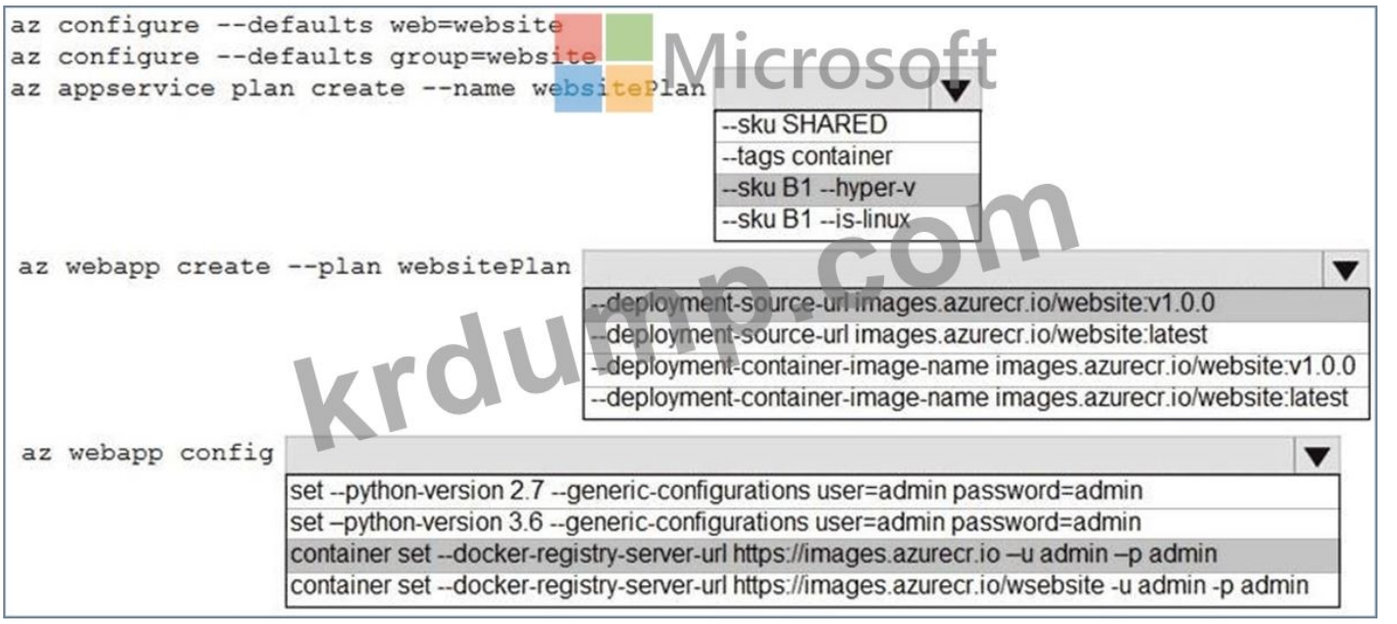
```
az webapp create --plan websitePlan [ ▼ ]
```

- --deployment-source-url images.azurecr.io/website:v1.0.0
- --deployment-source-url images.azurecr.io/website:latest
- --deployment-container-image-name images.azurecr.io/website:v1.0.0
- --deployment-container-image-name images.azurecr.io/website:latest

```
az webapp config [ ▼ ]
```

- set --python-version 2.7 --generic-configurations user=admin password=admin
- set –python-version 3.6 --generic-configurations user=admin password=admin
- container set --docker-registry-server-url https://images.azurecr.io –u admin –p admin
- container set --docker-registry-server-url https://images.azurecr.io/wsebsite -u admin -p admin

Explanation:



Box 1: --SKU B1 --hyper-v

--hyper-v

Host web app on Windows container.

Box 2: --deployment-source-url images.azurecr.io/website:v1.0.0

--deployment-source-url -u

Git repository URL to link with manual integration.

The Web App must always run the same version of the website regardless of future builds.

Incorrect:

--deployment-container-image-name -i

Linux only. Container image name from Docker Hub, e.g. publisher/image-name:tag.

Box 3: az webapp config container set -url https://images.azurecr.io -u admin -p admin az webapp config container set Set a web app container's settings.

Paremeter: --docker-registry-server-url -r

The container registry server url.

The Azure Container Registry instance named images is a private registry.

Example:

az webapp config container set --docker-registry-server-url https://{azure-container-registry-name}.azurecr.io Reference:

https://docs.microsoft.com/en-us/cli/azure/appservice/plan

NEW QUESTION: 144

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.

□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

□□□ □□□□□□□□ □□□□ □□□□ □ □□□□ □□□□ Azure □□□ □□□□□□□ □□□□ □□□□. □□□□ □□□□ □□□□ □□□□ □□ □ □□□□.

□□□ □□ □□ □□□ □□□□.
* □□□ □□□ 80□□□□□(GB)□□ □□ □ □□□□.
* □□□□ □□□□(FIFO) □□□ □□□□□.
* Azure □□□ □□□□□□.
□□□ □□□□ □□□□ □□□.
□□□: .Net API□ □□□□ □□□ □□□□□□□□ Azure Storage Queue□ □□□□ □□□ □□. Azure Storage Queue □□□□□ □□□□□ Azure VM□ □□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Don't use a VM, instead create an Azure Function App that uses an Azure Service Bus Queue trigger.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function

NEW QUESTION: 145
Azure Blob Storage□ □□□ □□ □□□□ □□□□ □□□□□□□ □□□□ □□□□.
□□□□□ □□□ □□ □□ □□□ □□□□.
* □□□□□□□ □□□□ □□□□ □, □□□□ Exif(□□ □□ □□□ □□ □□) □□□□ blob □□□□□□ □□□□□.
* □□□□ □□ □□□ □□□□□□ □□□□□ Exif □□□□ □□□□□.
* REST API□ □□□□□.
□□□□□□□□ □□□ Exif □□□□ Blob □□□□□□ □□□□ □□□.
□□ HTTP □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□□□: □□ □□□ 1□□□□.

Answer:

Explanation:

NEW QUESTION: 146

□□□□ □□□□ □□□ □□□□ Azure App Service□ □□□□□. □ □□ 4□□ □□□□□ □□□□ □□□□ □□□ □□□□ □□ □□□ □□□□□.

□□ □□ □□□ □□□□□ □□□□ □ □□ □□□□□□ □□□.

* □□ □□ □□□ □□ □□□□□□ □□□ □□ □□□□□ □ □ □□□□ □□□ 1□□ □□ □ □□ □□□ □□□□ □□□□□ □□□□ □□□□□.

* □ □□ □□, □□□□□□ □□, □□ □□ □ □□ □□ □□□□ Azure Storage Blob □□□□ □□□□.

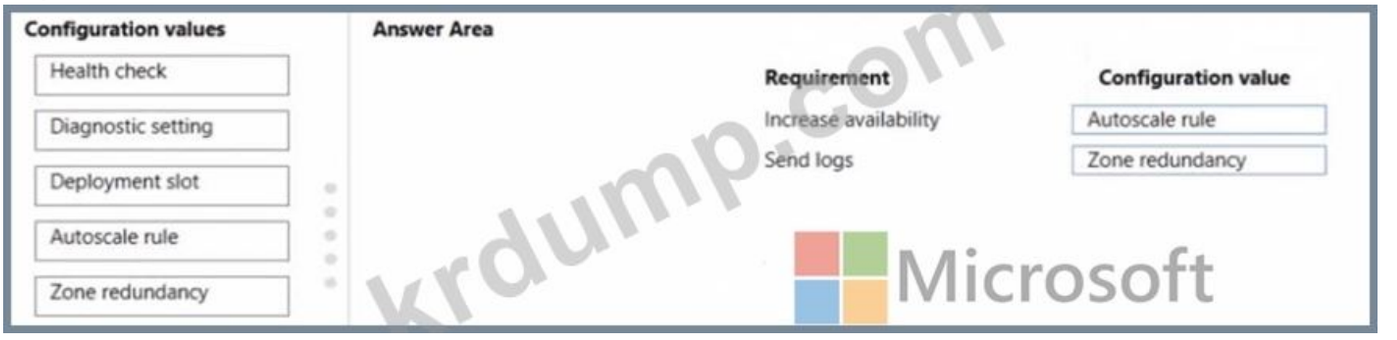
Azure App Service□ □□□□ □□□.

□□ □□ □□□□ □□□? □□□□□ □□□ □□ □□ □□□ □□ □□□□ □□□□□□□.

□ □□ □□ □ □ □□ □□□ □ □□□□....

Answer:

Explanation:



NEW QUESTION: 147

Azure Blob Storage □□□□□ □□□ □□□ □□□□ □ □□□ □ □□ ASP.NET Core □ □ □□□ □□□□ □□□□.

□ □□□ □□□□ Azure AD(Azure Active Directory) □□ □□□ □□□□ □□□□□.

□□□ □□□□ □□□□□ □□ □□ □□ □□□ □□(RBAC) □□ □□□ □□□□□.

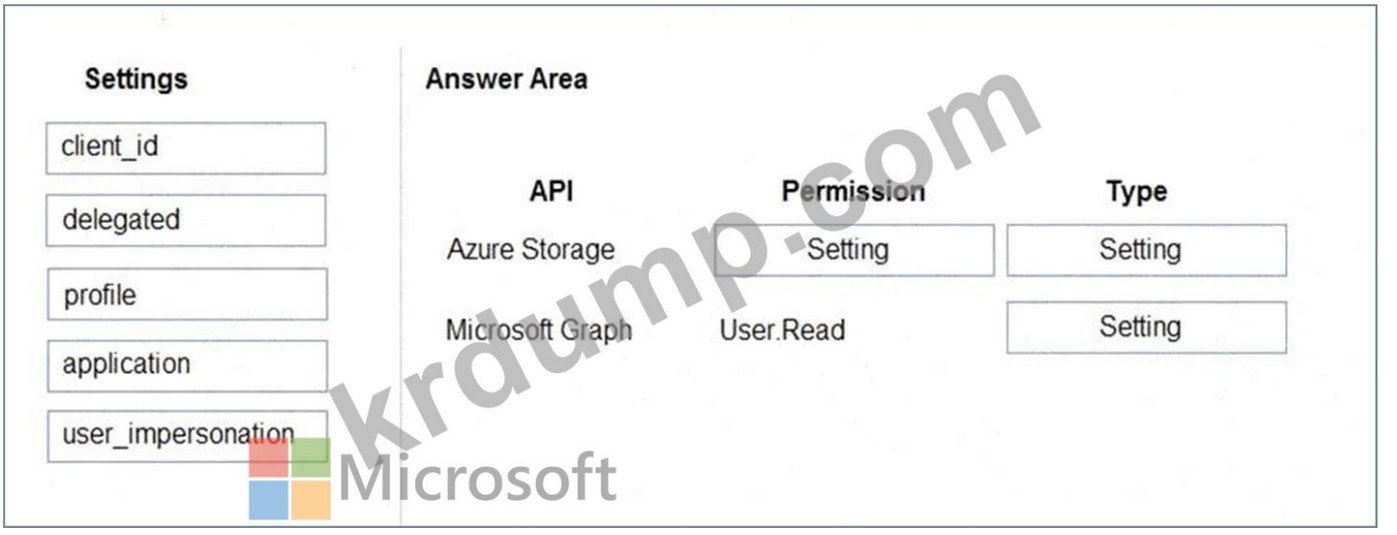
RBAC □□□ □□□□ □□□□□.

Azure Blob □□□□□□ □□□ □□□ □□□ □ □□□ □ □□□□ Azure AD □□□□□□□ □□□□ □□□.

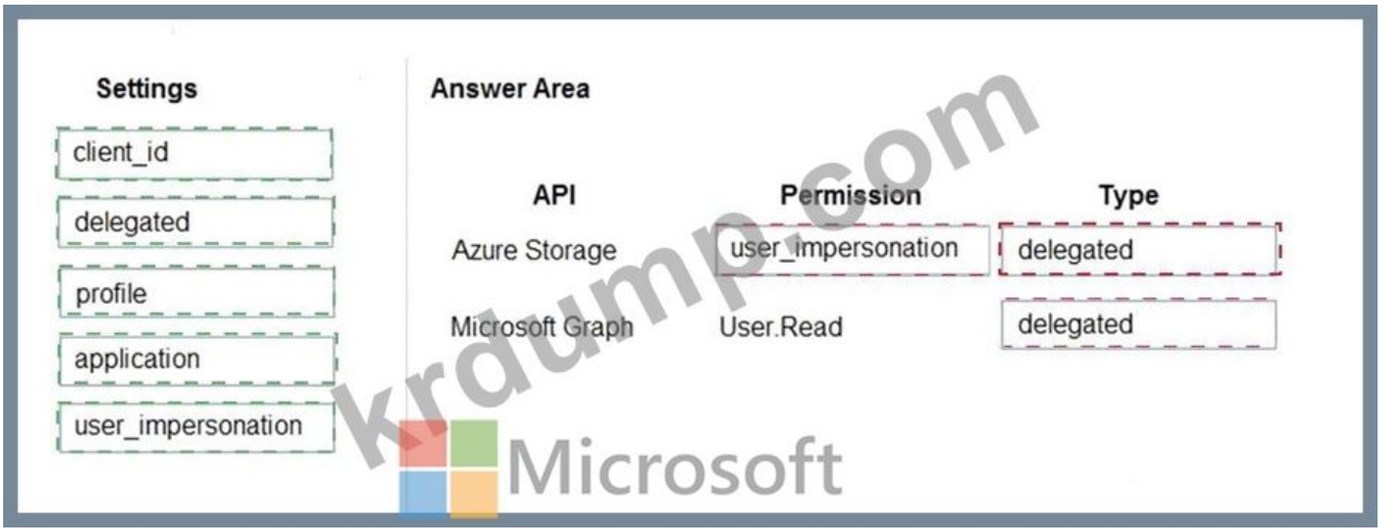
□□□□□□□ □□□ □□□□ □□□? □□□ □□□□ □□□ □□□ □□□ □□□ □□□ □□□.

□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□ □ □□ □□□ □□□ □□□□□ □ □ □□□□.

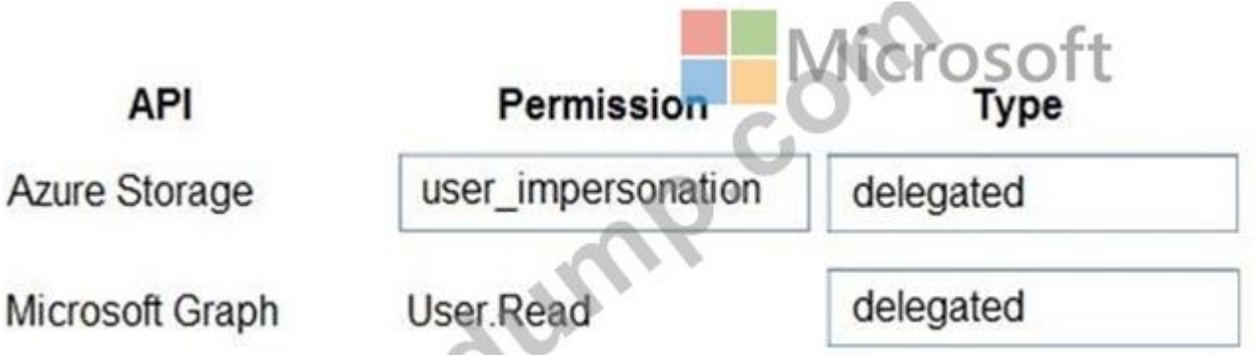
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: user_impersonation

Box 2: delegated

Example:

1.Select the API permissions section

2.Click the Add a permission button and then:

Ensure that the My APIs tab is selected

3.In the list of APIs, select the API TodoListService-aspnetcore.

4.In the Delegated permissions section, ensure that the right permissions are checked: user_impersonation.

5.Select the Add permissions button.

Box 3: delegated

Example

1.Select the API permissions section

2.Click the Add a permission button and then,

Ensure that the Microsoft APIs tab is selected

3.In the Commonly used Microsoft APIs section, click on Microsoft Graph

4.In the Delegated permissions section, ensure that the right permissions are checked: User.Read. Use the search box if necessary.

5.Select the Add permissions button

References:

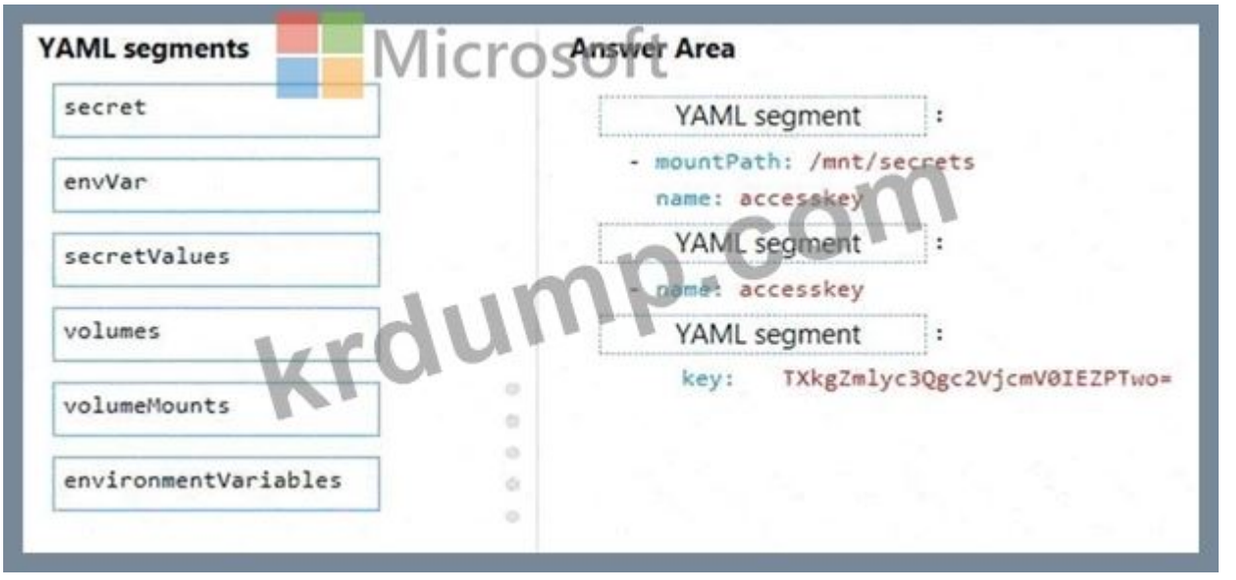
https://docs.microsoft.com/en-us/samples/azure-samples/active-directory-dotnet-webapp-webapi-openidconnect-aspnetcore/calling-a-web-api-in-an-aspnet-core-web-application-using-azure-ad/

NEW QUESTION: 148

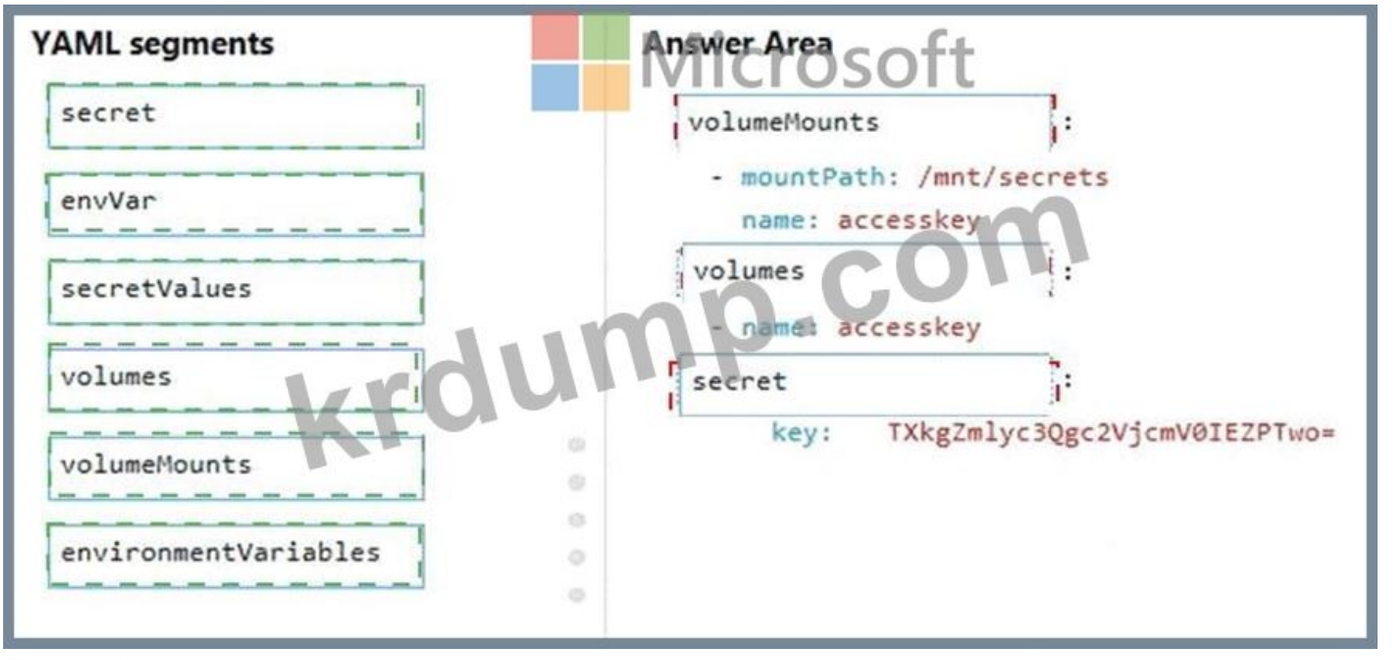
ContentUploadService□ Azure Storage □□□ □□ □□□□ □ □□□ CS17 □□ YAML □□ □ □□□□ □□□.

YAML □□□□ □□□ □□□□ □□□? □□□□□ □□□ YAML □□□□□ □□□ □□□ □ □□□□□□. □ YAML □□□□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□ □. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

Box 1: volumeMounts

Example:

volumeMounts:

- mountPath: /mnt/secrets

name: secretvolume1

volumes:

- name: secretvolume1

secret:

mysecret1: TXkgZmlyc3Qgc2VjcmV0IEZPTwo=

Box 2: volumes

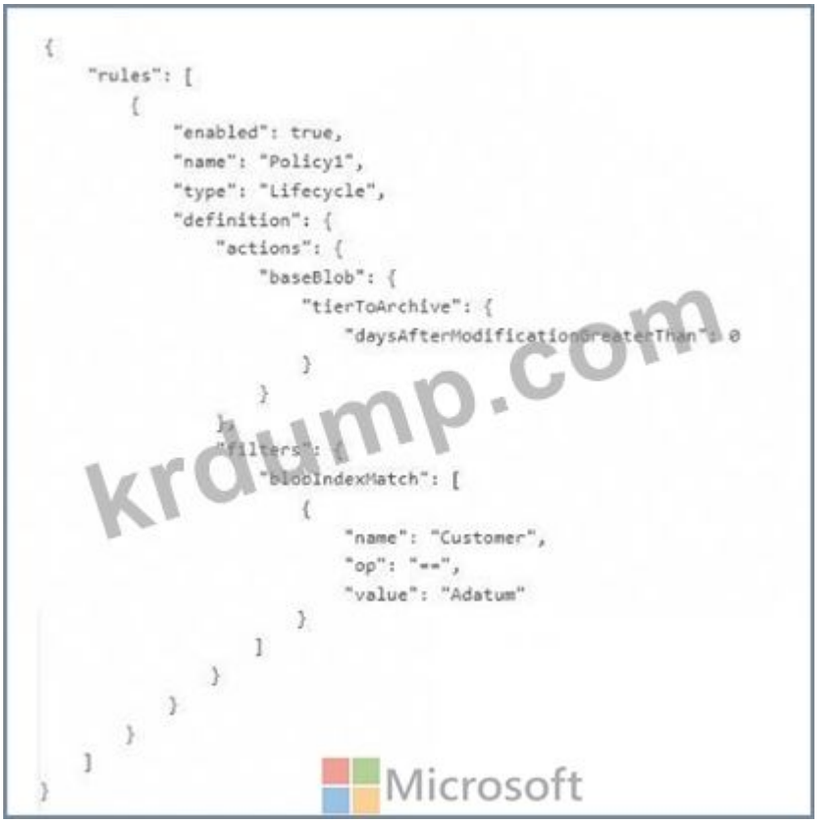
Box 3: secret

Reference:

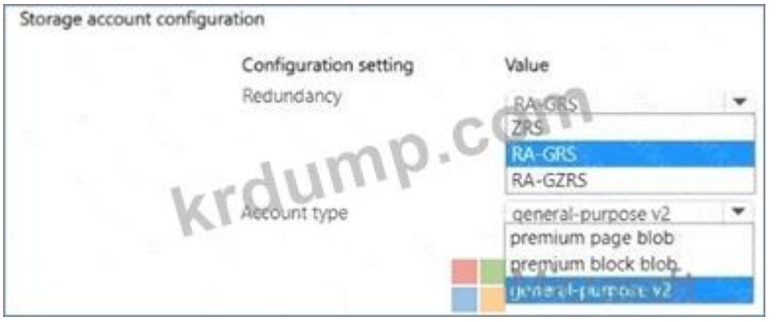
https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-secret

NEW QUESTION: 149

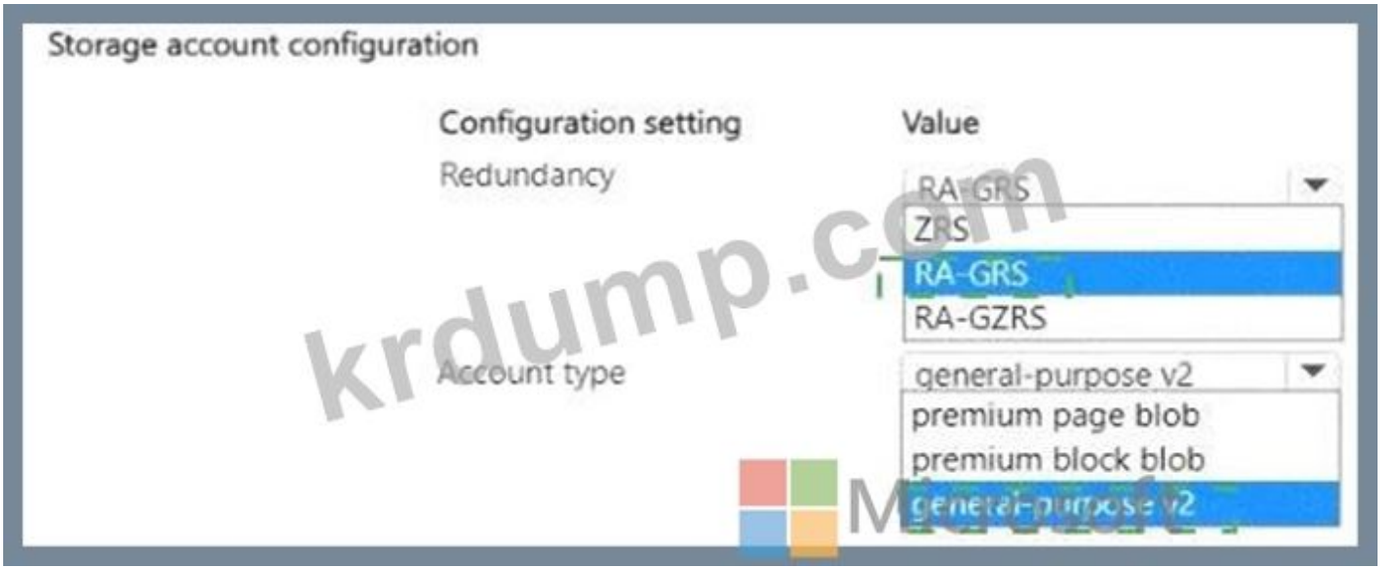
□□□ □□ □□□ □□ □□ □□ □□□ □□□□.



Azure Blob Storage □□□ □□□□ □□□ □□ 1□ □□□ □□□□□. □□□□ □□□□ □ □□ □□□□□ □□□.
□□□ □□□□□ □□□ □□□□ □□□.
□□ □□ □□□ □□□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□ □ □□□□□.
□□□□: □□ □□□ 1□□□□.



Answer:



Explanation:

NEW QUESTION: 150
□□□□ Azure Service Bus□ □□□□ □□□□(Pub/Sub) □□□ □□ □□□ □□□□ □□□ □. □ □□ □□ □□□□□□□ □□□□ □□□□.
Azure Portal□□ □□□□ □ □□□ □□ □□□□ □□□□ □□ □ □ □□□□. □□□ □□ □□□ □□□□ □□ □□□□□ □□□ □□□ □□□□□□ □□ □□□□□□□ □□□ □□ □□ □□□□ □□□□.
□□ □□□□□□ □□ □□□ □□□□ □□□.

□□□ □□ □□□?

A. await subscriptionClient.CloseAsync();

B. await subscriptionClient.AddRuleAsync(new RuleDescription(RuleDescription.DefaultRuleName, new TrueFilter()));

C. subscriptionClient.RegisterMessageHandler(ProcessMessagesAsync, messageHandlerOptions);

D. subscriptionClient = new SubscriptionClient(ServiceBusConnectionString, TopicName, SubscriptionName);

Answer: C (LEAVE A REPLY)

Using topic client, call RegisterMessageHandler which is used to receive messages continuously from the entity. It registers a message handler and begins a new thread to receive messages. This handler is waited on every time a new message is received by the receiver.

subscriptionClient.RegisterMessageHandler(ReceiveMessagesAsync, messageHandlerOptions);

References:

https://www.c-sharpcorner.com/article/azure-service-bus-topic-and-subscription-pub-sub/

NEW QUESTION: 151

□□□□ □□□ □□ □□□ □□ □□□ □□ □□□□ □□ □□□ □□□ □□□□ □□□. □ □□ □□□□ □□□?

A. Azure AD □□□ □□

B. Azure RBAC □□

C. □□ □□□ □□(SAS) □□

D. Azure AD ID □□

E. Azure AD □□ □□ □□

Answer: C (LEAVE A REPLY)

NEW QUESTION: 152

□□□ □□ □ □□□□ □□□ Azure □□□□ □□□□ Azure Batch □□□□□ □□□□. □ □ □□□ □□□□ □□□ □□ □□□□.

□□□ □□ □□□□□ □□□□□.

Parameter name	Description
fileTasks	a list of tasks to be run
jobId	the identifier that must be assigned to the job
outputContainerSasUrl	a storage SAS URL to store successfully converted files
failedContainerSasUrl	a storage SAS URL to store copies of files that failed to convert

□□□ □□□ outputContainerSasUrl □□□□□ □□□□ □□□□□ □□□□□□ □□□□ □□□. □□□ □□□ □□□ failedContainerSasUrl □□□□□ □□□□ □□□□□ □□□□ □.

□□□ □□□□ □□□□□□ □□□□ □□□.

□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

```
public list<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
 string outputContainerSasUrl, string failedContainerSasUrl)
{
  BatchSharedKeyCredentials sharedKeyCredentials =
   new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
 List<CloudTask> tasks = new List<CloudTask>();
 using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
{
   CloudJob = batchClient.JobOperations. [ ▼ ] ();
                                         GetJob
                                         GetTask
                                         EnableJob
                                         CreateJob
        job.Id = jobId,
        job.PoolInfromation = new PoolInformation { PoolId = poolId };
        job.Commit();
        fileTasks.ForEach((fileTask) =>
        {
          string taskId = $"Task{DateTime.Now.ToFileTimeUtc().ToString()}";
          CloudTask task = new CloudTask (taskId, fileTask.Command);
          List<OutputFile> outputFileList = new List<OutputFile>();
          OutputFileBlobContainerDestination outputContainer =
           new OutputFileBlobContainerDestination(outputContainerSasUrl);
        OutputFileBlobContainerDestination failedContainer =
     new OutputFileBlobContainerDestination (failedContainerSasUrl);
 outputFileList.Add(new OutputFile(fileTask.Output,
     new OutputFileDestination(outputContainer),
     new OutputFileUploadOptions(OutputFileUploadCondition. [ ▼ ] )));
                                                           TaskSuccess
                                                           TaskFailure
                                                           TaskCompletion
    outputFileList.Add(new OutputFile(fileTask.Output,
     new OutputFileDestination(failedContainer),
     new OutputFileUploadOptions(OutputFileUploadCondition, [ ▼ ] )));
                                                           TaskSuccess
                                                           TaskFailure
                                                           TaskCompletion
     task [ ▼ ] =outputFileList;
          OutputFiles
          FilesToStage
          ResourceFiles
          StageFiles
     task.Add(task);
     });
     ]
     return tasks,
 }
```

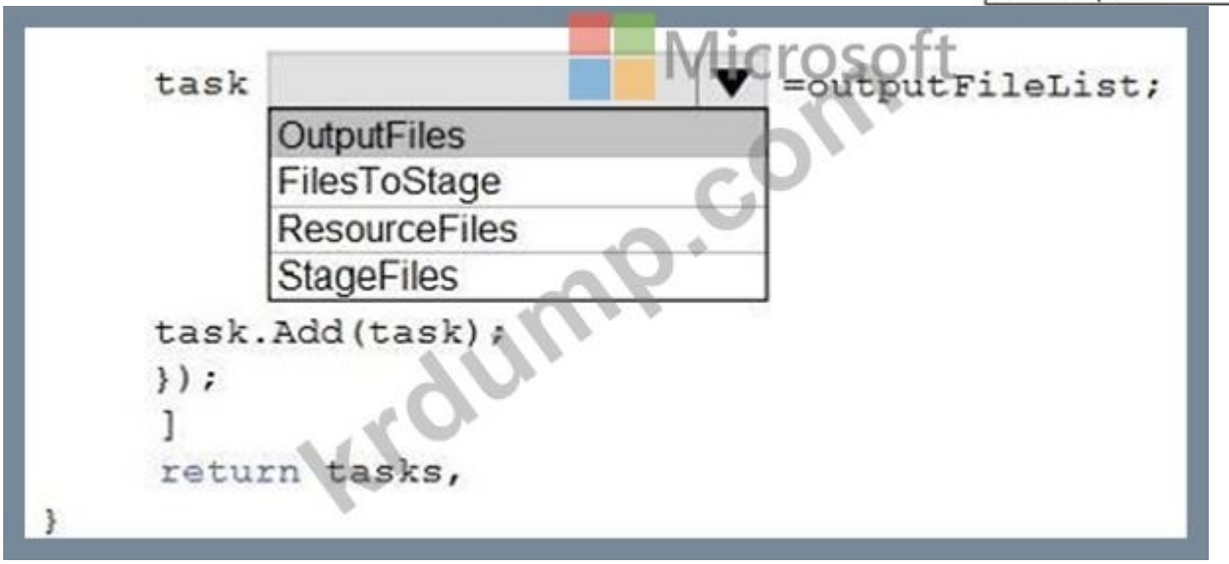
Answer:

Answer Area

```
public list<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
 string outputContainerSasUrl, string failedContainerSasUrl)
{
  BatchSharedKeyCredentials sharedKeyCredentials =
   new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
 List<CloudTask> tasks = new List<CloudTask>();
 using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
{
   CloudJob = batchClient.JobOperations. [▼] ();
                                          GetJob
                                          GetTask
                                          EnableJob
                                          CreateJob

        job.Id = jobId,
        job.PoolInfromation = new PoolInformation { PoolId = poolId };
        job.Commit();
        fileTasks.ForEach((fileTask) =>
        {
          string taskId = $"Task{DateTime.Now.ToFileTimeUtc().ToString()}";
          CloudTask task = new CloudTask (taskId, fileTask.Command);
          List<OutputFile> outputFileList = new List<OutputFile>();
          OutputFileBlobContainerDestination outputContainer =
           new OutputFileBlobContainerDestination(outputContainerSasUrl);
        OutputFileBlobContainerDestination failedContainer =
     new OutputFileBlobContainerDestination (failedContainerSasUrl);
 outputFileList.Add(new OutputFile(fileTask.Output,
    new OutputFileDestination(outputContainer),
    new OutputFileUploadOptions(OutputFileUploadCondition. [▼] )));
                                                            TaskSuccess
                                                            TaskFailure
                                                            TaskCompletion

    outputFileList.Add(new OutputFile(fileTask.Output,
     new OutputFileDestination(failedContainer),
     new OutputFileUploadOptions(OutputFileUploadCondition, [▼] )));
                                                            TaskSuccess
                                                            TaskFailure
                                                            TaskCompletion

     task [▼] =outputFileList;
          OutputFiles
          FilesToStage
          ResourceFiles
          StageFiles
     task.Add(task);
     });
     ]
     return tasks,
}
```


Explanation:

```
CloudJob = batchClient.JobOperations. [GetJob / GetTask / EnableJob / CreateJob] ();
    job.Id = jobId,
    job.PoolInfromation = new PoolInformation { PoolId = poolId };
    job.Commit();
    fileTasks.ForEach((fileTask) =>
    {
      string taskId = $"Task{DateTime.Now.ToFileTimeUtc().ToString()}";
      CloudTask task = new CloudTask (taskId, fileTask.Command);
      List<OutputFile> outputFileList = new List<OutputFile>();
      OutputFileBlobContainerDestination outputContainer =
       new OutputFileBlobContainerDestination(outputContainerSasUrl);
    OutputFileBlobContainerDestination failedContainer =
       new OutputFileBlobContainerDestination (failedContainerSasUrl);
    outputFileList.Add(new OutputFile(fileTask.Output,
       new OutputFileDestination(outputContainer),
       new OutputFileUploadOptions(OutputFileUploadCondition. [TaskSuccess / TaskFailure / TaskCompletion] )));

      outputFileList.Add(new OutputFile(fileTask.Output,
       new OutputFileDestination(failedContainer),
       new OutputFileUploadOptions(OutputFileUploadCondition, [TaskSuccess / TaskFailure / TaskCompletion] )));
```



Box 1: CreateJob

Box 2: TaskSuccess

TaskSuccess: Upload the file(s) only after the task process exits with an exit code of 0.

Incorrect: TaskCompletion: Upload the file(s) after the task process exits, no matter what the exit code was.

Box 3: TaskFailure

TaskFailure:Upload the file(s) only after the task process exits with a nonzero exit code.

Box 4: OutputFiles

To specify output files for a task, create a collection of OutputFile objects and assign it to the

CloudTask.

OutputFiles property when you create the task.

References:

https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.batch.protocol.models.outputfileuploadcondition

https://docs.microsoft.com/en-us/azure/batch/batch-task-output-files

NEW QUESTION: 153

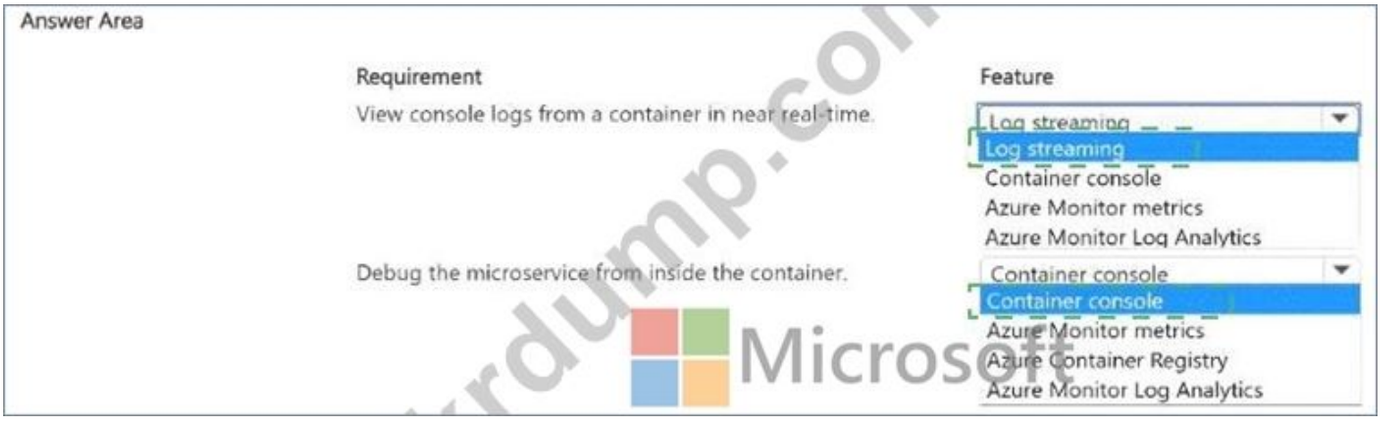
Azure Container Apps□□ □□□□ □□ □□ □□□□□□□□ □□□□ □□□□.

□□□□□□□□ □□□□□□ □□□□ □□□.

□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□□□: □□ □□□ 1□□□□.

Answer:



Explanation:

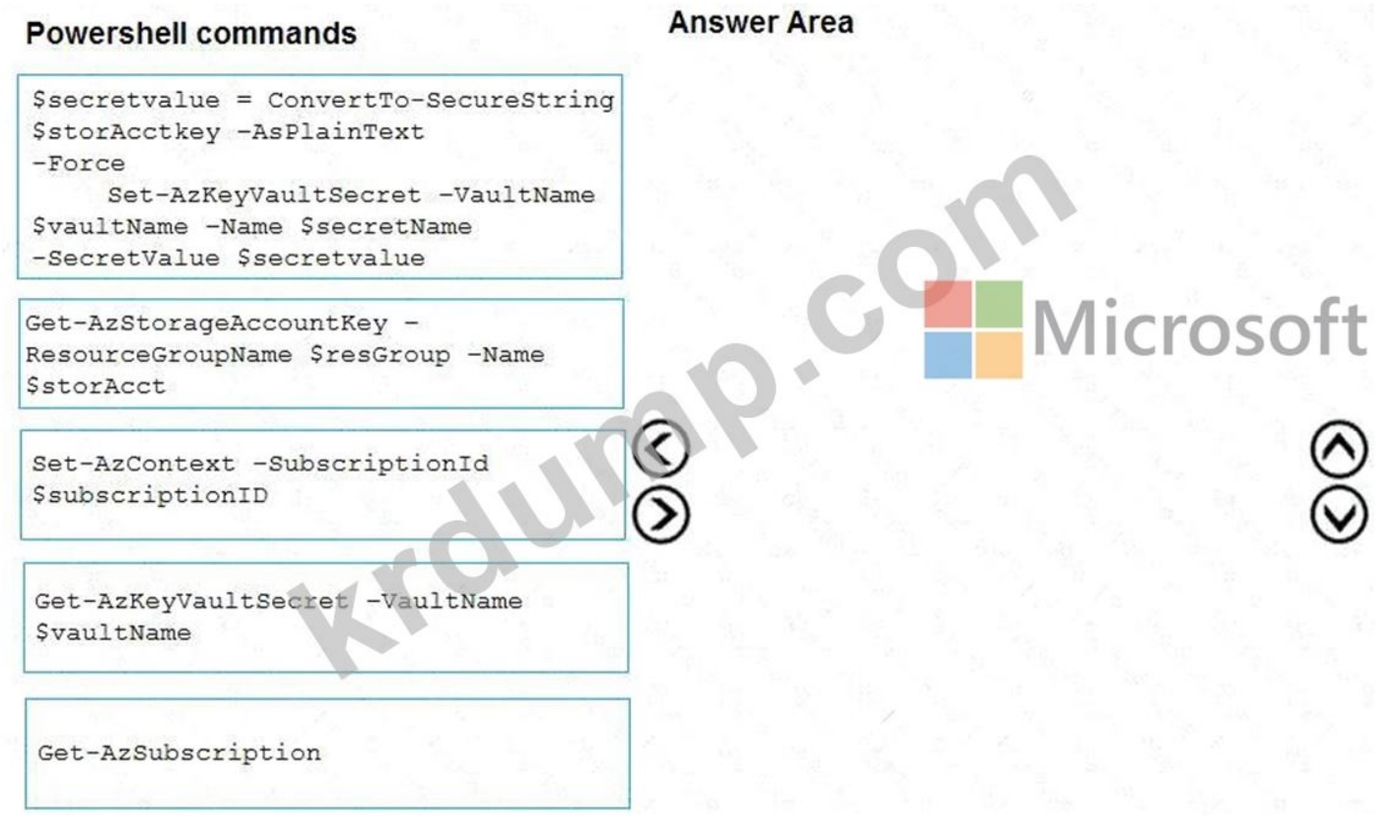
NEW QUESTION: 154

□□□□□□□ □□ □□□□. □ □□ □□□ □□□□ □ □□ Azure □□□ □□□ □□□□.

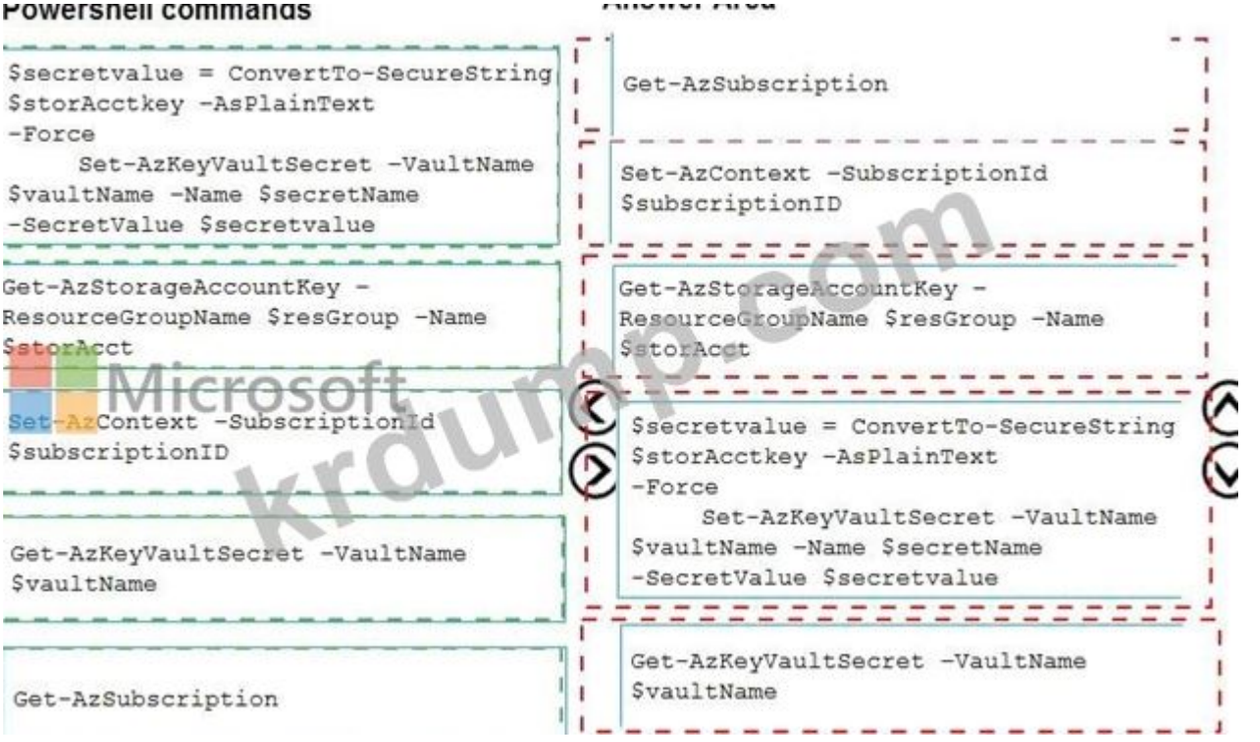
Azure Key Vault□□ □□□□ □□ □ □□□ □□□□ □□□.

□□□□ □□□□ □□ PowerShell □□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□

□ □□ □□□ □□ □□□□ □□□□ □□□ □□□ □□□□□□.



Answer:



Explanation:

Step 1: Get-AzSubscription

If you have multiple subscriptions, you might have to specify the one that was used to create your key vault.

Enter the following to see the subscriptions for your account:

Get-AzSubscription

Step 2: Set-AzContext -SubscriptionId

To specify the subscription that's associated with the key vault you'll be logging, enter:

Set-AzContext -SubscriptionId <subscriptionID>

Step 3: Get-AzStorageAccountKey

You must get that storage account key.

Step 4: $secretvalue = ConvertTo-SecureString <storageAccountKey> -AsPlainText -Force Set-AzKeyVaultSecret -VaultName <vaultName> -Name <secretName> -SecretValue $secretvalue

After retrieving your secret (in this case, your storage account key), you must convert that key to a secure string, and then create a secret with that value in your key vault.

Step 5: Get-AzKeyVaultSecret

Next, get the URI for the secret you created. You'll need this URI in a later step to call the key vault and retrieve your secret. Run the following PowerShell command and make note of the ID value, which is the secret's URI:

Get-AzKeyVaultSecret -VaultName <vaultName>

Reference:

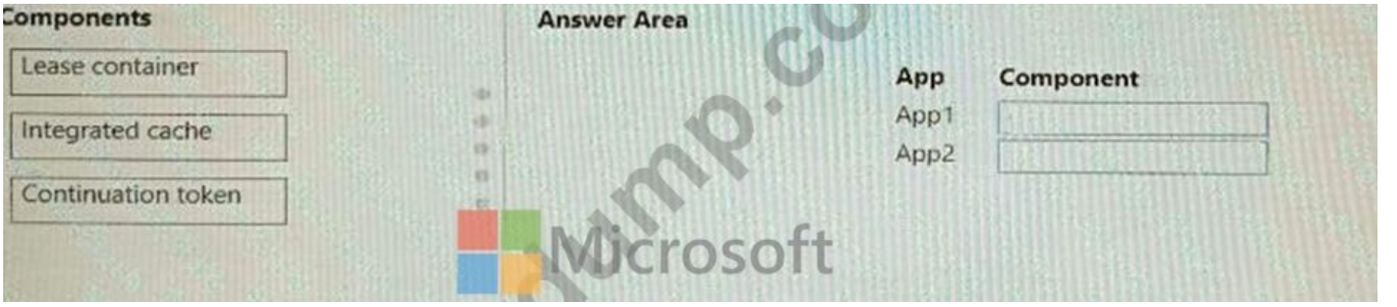
https://docs.microsoft.com/bs-latn-ba/Azure/key-vault/key-vault-key-rotation-log-monitoring

NEW QUESTION: 155

NoSQL□ Azure Cosmos DB □□□ □□□□.

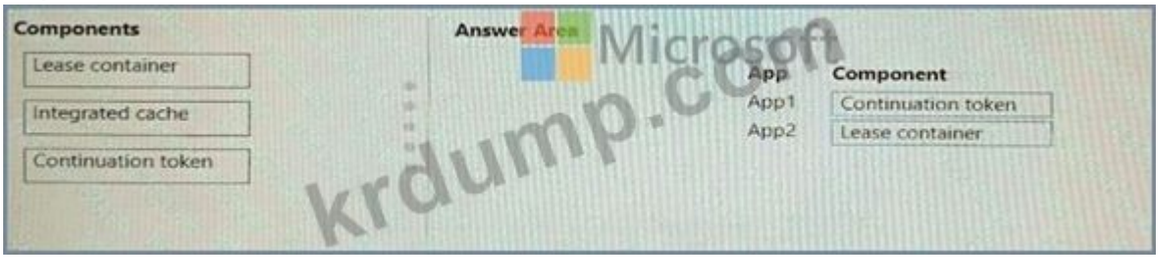
□□□□□ □□ □□ □□□ □□□□ □□ □□ □□ □□□ □□□ App1 □ App2□□ □ □□ □□ □□□ □□□□□.

App1□ □ □□□ □□□□ App2□ □□ □□□ □□□□□.
App1□ App2□□ □□ □□□ □□□ □□ □□□ □□□□ □□□ □□□□ □□□.
□□ □□ □□□ □□□□ □□□? □□□□□ □□□ □□□□□ □□□ □□□ □□□□□□.
□ □□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

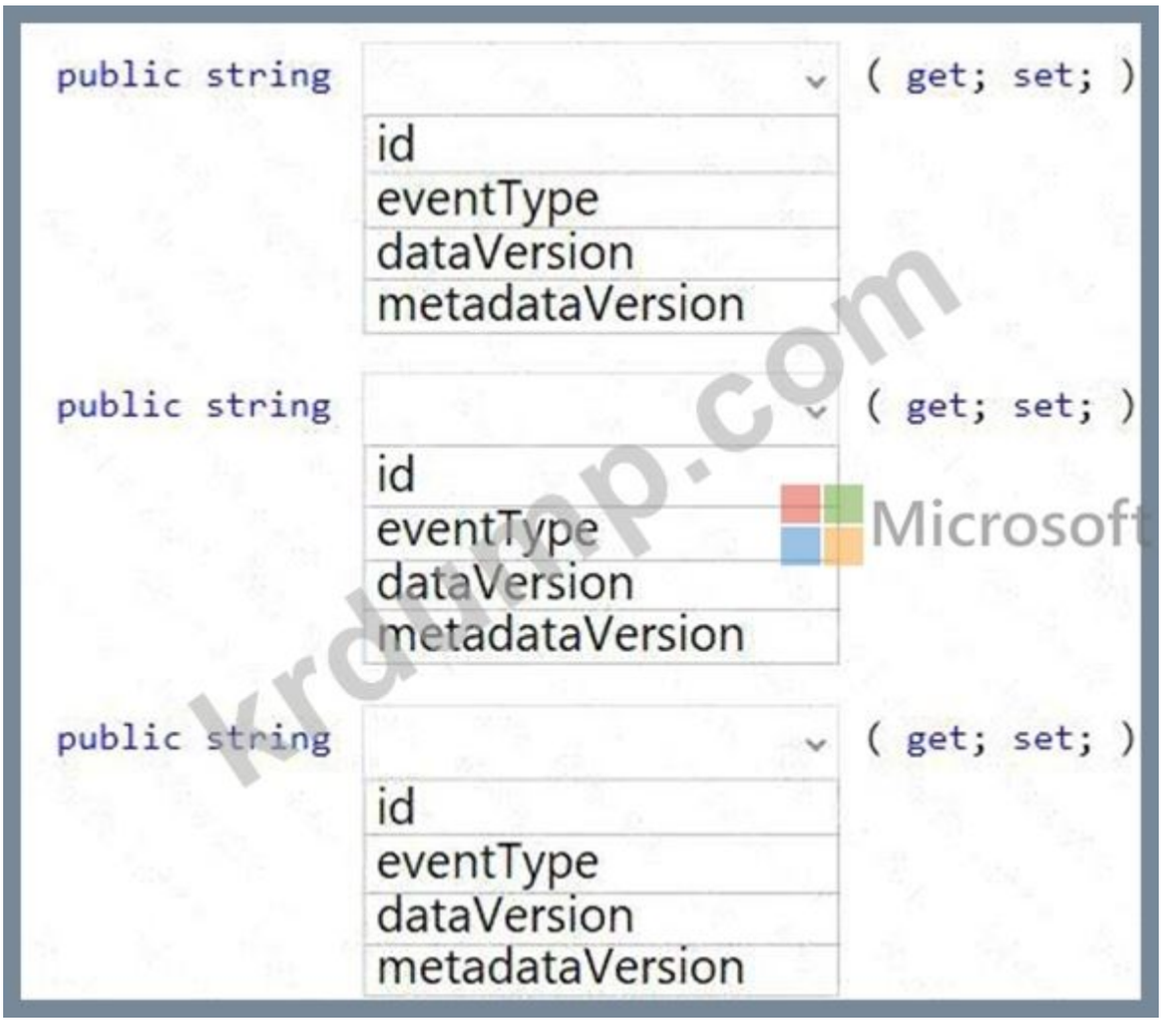


Answer:

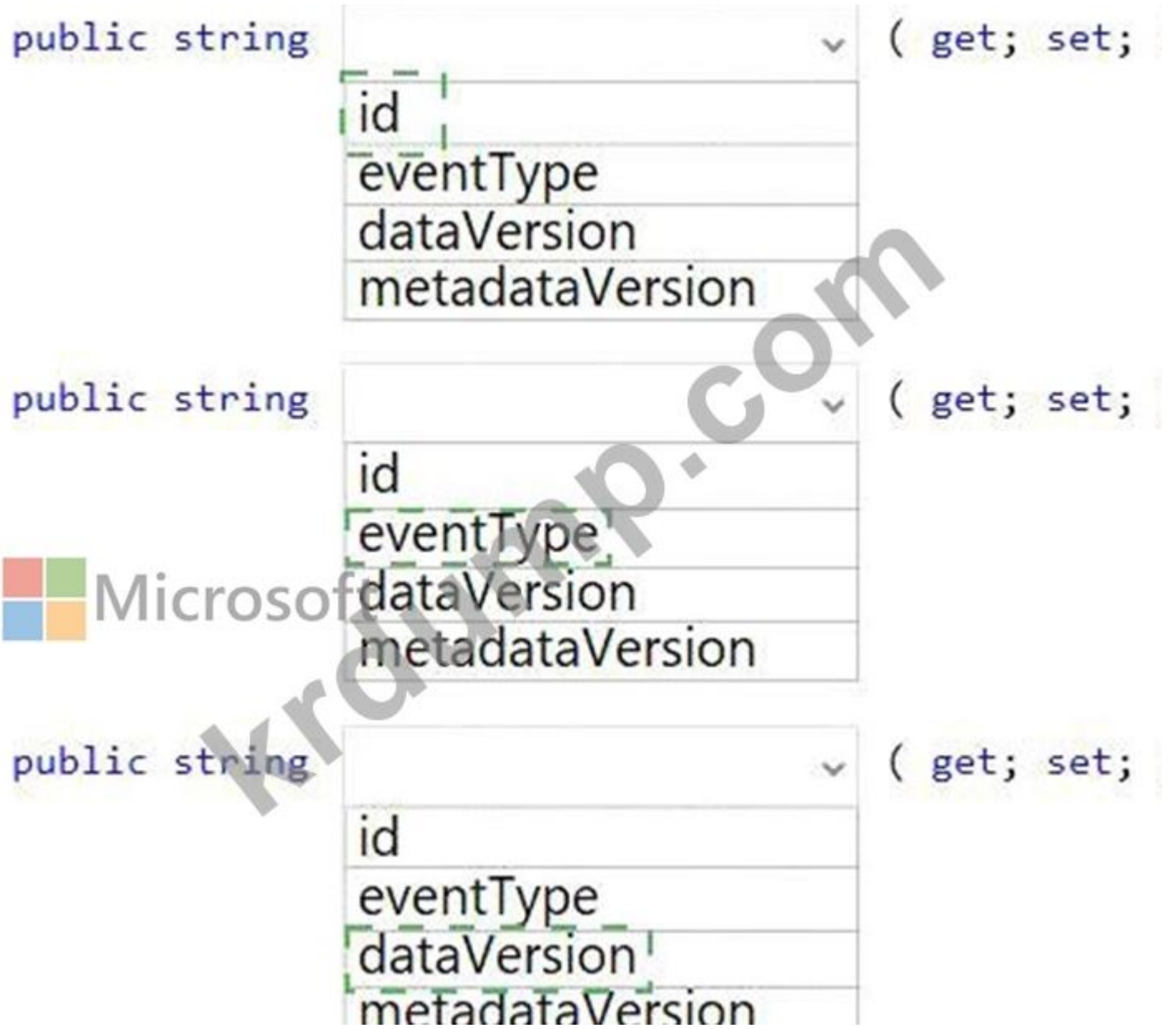
Explanation:



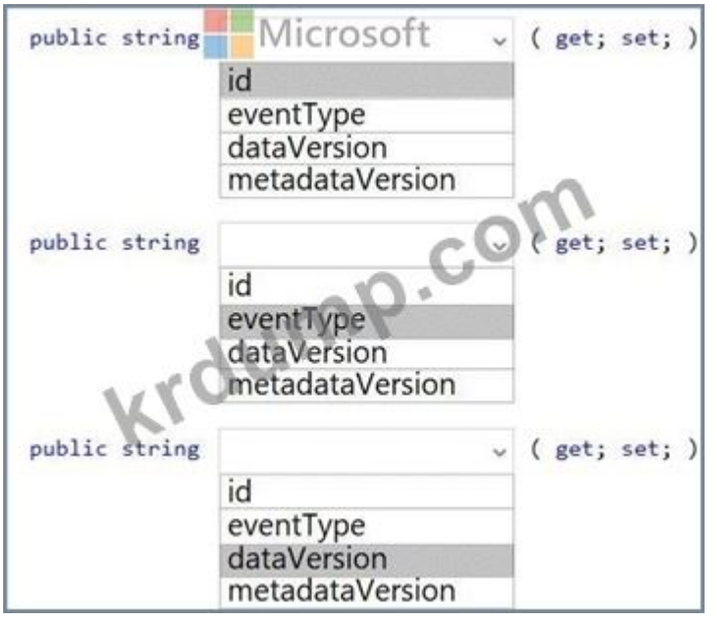
NEW QUESTION: 156
□□ □□ □□□□ □□□□ □□□□□ □□□ LoginEvent.cs□ LE03 □□ □□□ □□□□ □ □□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.


public string
(get; set;)
id
eventType
dataVersion
metadataVersion
public string
(get; set;)
id
eventType
dataVersion
metadataVersion
public string
(get; set;)
id
eventType
dataVersion
metadataVersion

Answer:



Explanation:



Box 1: id

id is a unique identifier for the event.

Box 2: eventType
eventType is one of the registered event types for this event source.
Box 3: dataVersion
dataVersion is the schema version of the data object. The publisher defines the schema version.
Scenario: Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.
The following example shows the properties that are used by all event publishers:
[
{
"topic": string,
"subject": string,
"id": string,
"eventType": string,
"eventTime": string,
"data":{
object-unique-to-each-publisher
},
"dataVersion": string,
"metadataVersion": string
}
]
Reference:
https://docs.microsoft.com/en-us/azure/event-grid/event-schema

NEW QUESTION: 157
□□□ □□□□□□ □□□□ API!□□ Azure API Management □□□□□ □□□□.
□□ □□□□ □□□□ API1 □□□□□□ □□□□□ □□□ □□ □□□ □□□□□ □□□□ □□ □□□ □□□□ □□□ □□□ □□□□□. □□ □□□□ □□ □□□ □□□□ □□□□.

```
<policies>
    <inbound>
        TARGET1
        <set-header name="x-request-context-data" exists-action="override">
            <value>@(TARGET2.Deployment.Region)</value>
      </set-header>
    </inbound>
</policies>
```

□□ □□□ □□ □□ □□□ □□□ □□□□.
* □□□ API1 □□□□□□ □□□□□ □□□ □□ □□□ □□□□ □□□ □□□□□.
* □□□ □□□ □□□ □□ □□□ □□□ □□□ □□□□□ □□□.
□□ □□□ □□□□ □□□.
□□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

Answer:

Explanation:

NEW QUESTION: 158

Redis□ Azure Cache□ □□□□ □ □□□□□□□ □□□□ □□□□. □□□ □□ □□□□ □□ □□□□ □ □□□ □□□□□.

□□□ □□ □□□ □□ □□□ □□□□ Azure Cache for Redis□ □□□□ □□□. □□□ □□ □□ □□□ □□□ □□□□ □□ □ □□ □□□□□□.

□□□ □□ □□□ □□ □□□ □□□□□□ Azure Cache for Redis□ □□□□ □□□. □□□ □□□ □ □□ □□ □□□ □□□□□?

□□: □ □□□ □□□ 1□□ □□□ □□□□.

A. □□ □□

B. allkeys-lru

C. □□□-lru

D. □□ □-□□□

E. □□□-ttl

F. □□□-□□□

Answer: (SHOW ANSWER)

B: The allkeys-lru policy evict keys by trying to remove the less recently used (LRU) keys first, in order to make space for the new data added. Use the allkeys-lru policy when you expect a power-law distribution in the popularity of your requests, that is, you expect that a subset of elements will

be accessed far more often than the rest.
C: volatile-lru: evict keys by trying to remove the less recently used (LRU) keys first, but only among keys that have an expire set, in order to make space for the new data added.
Note: The allkeys-lru policy is more memory efficient since there is no need to set an expire for the key to be evicted under memory pressure.
Reference:
https://redis.io/topics/lru-cache

NEW QUESTION: 159
□□ □□□□□□ □□ □□□□ □□□□ □□□.
□ □□□ □□□ □ □□ □ □□ □□□ □□□□□? □ □□□ □□□ □□□□ □□□□□.
□□: □□ □□□ 1□□□□.
A. App Service □□□□□ □□□□ □□□ □□ □□□□ □□□□ □□□□□□ □□□□□.
B. Azure App Service □□ □□ □□□ □□□□ □ □□ WEBSITE_LOCAL_CACHE_SIZEINMB □□ □□□□□.
C. □□ □□ □□□□ Azure Blob Storage□ □□□□□. □□□ □□□ Azure Content Delivery Network□ □□□□□□.
D. □□□ Azure Traffic Manager □□□□ □□□□. □□ □□□□ □□ □□□ □□□ □□□□, chtld □□□□ □□ □□ □□□ □□□ □□□□ □□□□□.
E. Azure Traffic Manager □□□ □□□ □□ □□□ □□□□□□□.
Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 160
Windows □□□ □□ □ □□□ □□□ □□ □□ □□□□□.
□□□□ □ □ □□ □ □□□ □□□ □□□□ □□□ □□□ □□□□ □□□.
□□ □□□ □□□□ □□□.
□□ □□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

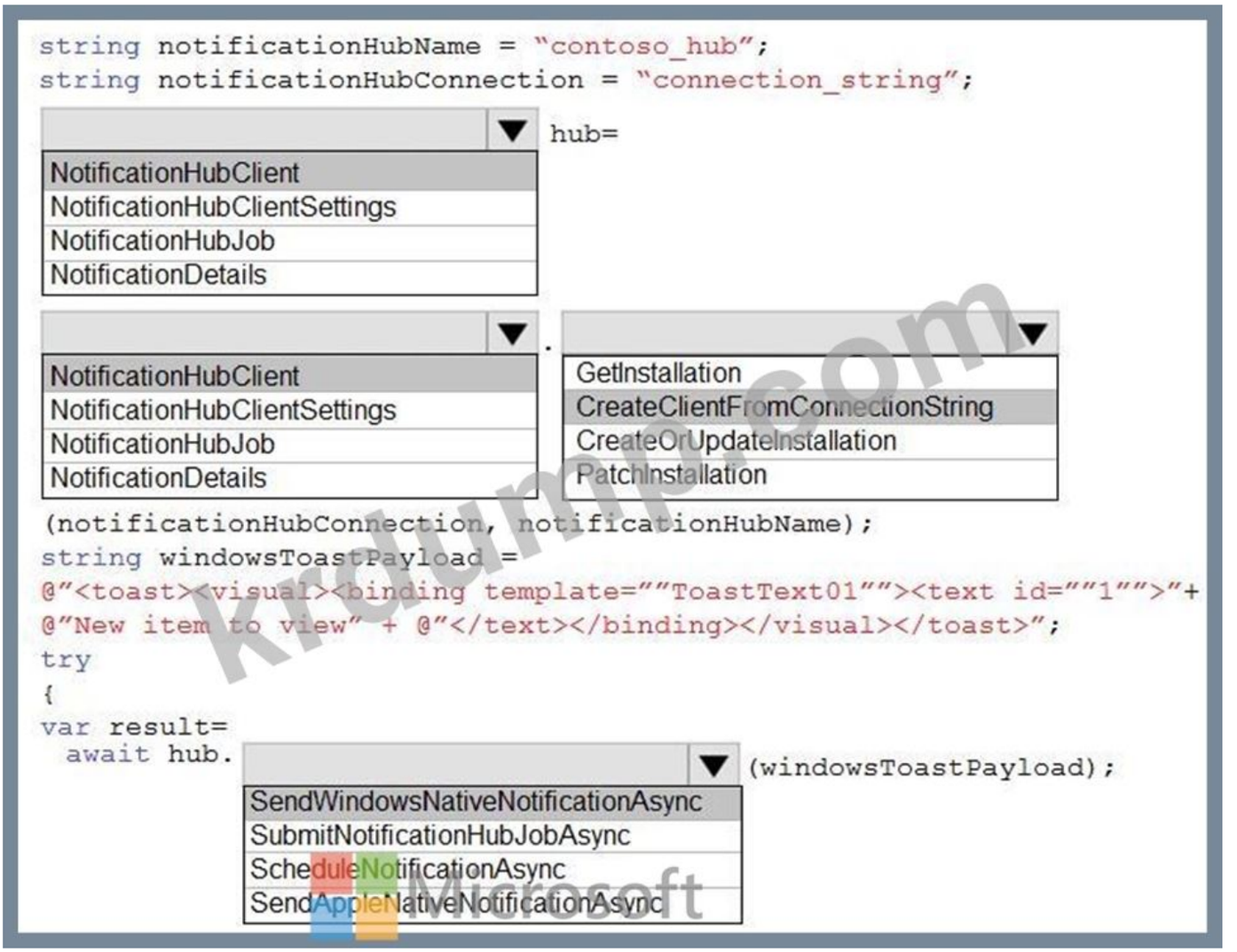
```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";

[ ▼ ] hub=
    NotificationHubClient
    NotificationHubClientSettings
    NotificationHubJob
    NotificationDetails

[ ▼ ] . [ ▼ ]
    NotificationHubClient                 GetInstallation
    NotificationHubClientSettings         CreateClientFromConnectionString
    NotificationHubJob                    CreateOrUpdateInstallation
    NotificationDetails                   PatchInstallation

(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"<toast><visual><binding template=""ToastText01""><text id=""1"">"+
@"New item to view" + @"</text></binding></visual></toast>";
try
{
var result=
  await hub.[ ▼ ](windowsToastPayload);
        SendWindowsNativeNotificationAsync
        SubmitNotificationHubJobAsync
        ScheduleNotificationAsync
        SendAppleNativeNotificationAsync
. . .
}
catch (System.Exception ex)
{
  . . .
}
  . . .
```

Answer:

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";

[ ▼ ] hub=
    NotificationHubClient
    NotificationHubClientSettings
    NotificationHubJob
    NotificationDetails

[ ▼ ] .
    NotificationHubClient
    NotificationHubClientSettings
    NotificationHubJob
    NotificationDetails

[ ▼ ]
    GetInstallation
    CreateClientFromConnectionString
    CreateOrUpdateInstallation
    PatchInstallation

(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"<toast><visual><binding template=""ToastText01""><text id=""1"">"+
@"New item to view" + @"</text></binding></visual></toast>";
try
{
var result=
  await hub.[ ▼ ](windowsToastPayload);
    SendWindowsNativeNotificationAsync
    SubmitNotificationHubJobAsync
    ScheduleNotificationAsync
    SendAppleNativeNotificationAsync
. . .
}
catch (System.Exception ex)
{
  . . .
}
 . . .
```

Explanation:



Box 1: NotificationHubClient

Box 2: NotificationHubClient

Box 3: CreateClientFromConnectionString

// Initialize the Notification Hub

NotificationHubClient hub =

NotificationHubClient.CreateClientFromConnectionString(listenConnString, hubName); Box 4: SendWindowsNativeNotificationAsync Send the push notification.

var result = await hub.SendWindowsNativeNotificationAsync(windowsToastPayload);

References:

https://docs.microsoft.com/en-us/azure/notification-hubs/notification-hubs-push-notification-registration- management

https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/app-service-mobile/app-service-mobile- windows-store-dotnet-get-started-push.md

NEW QUESTION: 161

Azure Batch □□□ □□□ □□□□□ □□□□ □□□□□ □□□ □□□□. □□□□ □□□□ □□ □□ □ □□□ □□□ □□□□.

□□□ □□ □□□□□ □□□□.

Parameter name	Description
$script	the script that will run across the batch pool
$image	the image that pool worker processes will use
$sku	the node agent SKU Id
$numberOfJobs	the number of jobs to run

□□, □□□ □ □□ □□□□ Azure CLI □□□□□ □□□□ □□□.

□□□□ □□□□ □□ □□□□ □□ □□□ □□□□ □□□? □□□□ □□□ □□□□ □□ □□ □□ □□□□ □□ □□□□ □□ □□□ □□□□ □□□□□.

Command segments

```
az batch pool create
 --id mypool --vm-size Standard_A1_v2
 --target-dedicated-nodes 2
 --image $image
 --node-agent-sku-id $sku
```

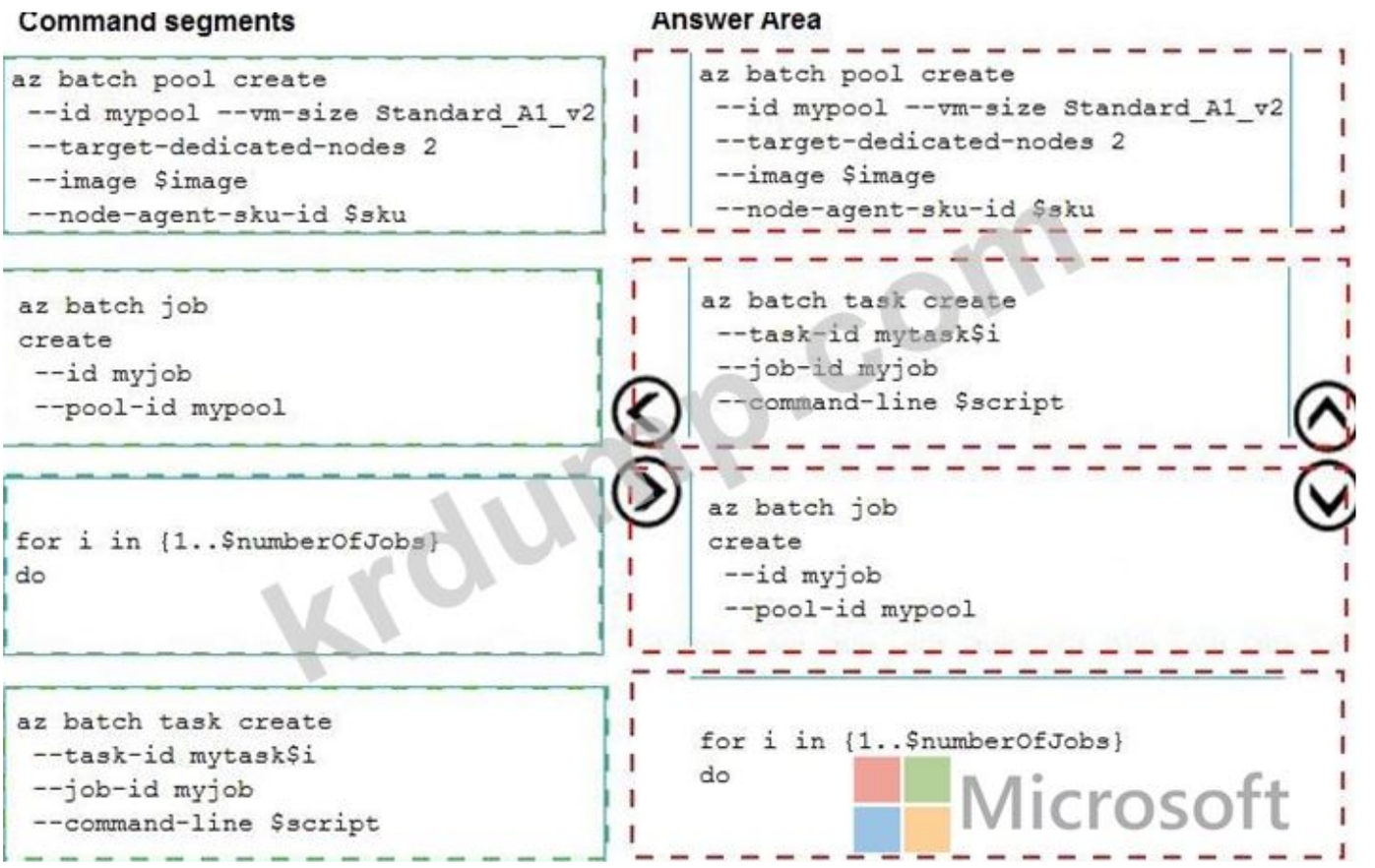
```
az batch job
create
 --id myjob
 --pool-id mypool
```

```
for i in {1..$numberOfJobs}
do
```

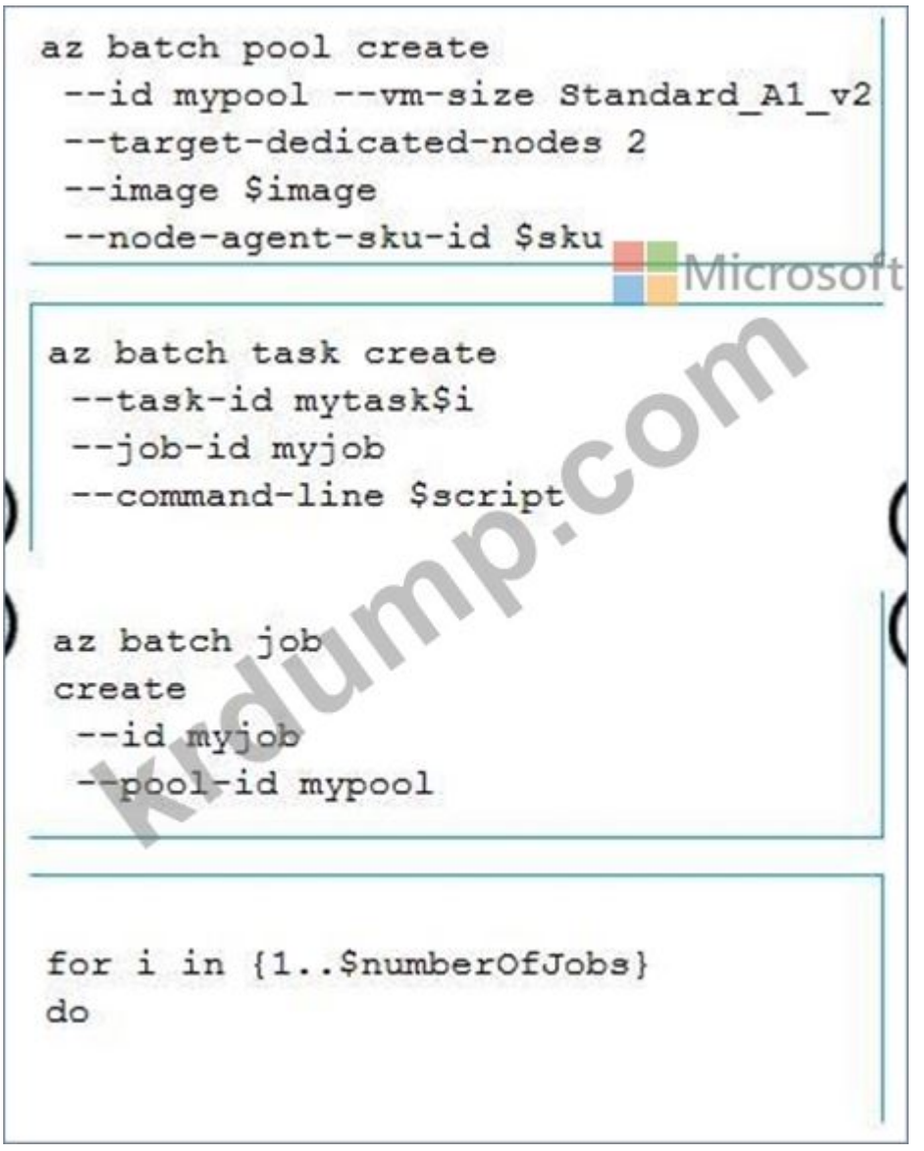
```
az batch task create
 --task-id mytask$i
 --job-id myjob
 --command-line $script
```

Answer Area

Answer:



Explanation:



Step 1: az batch pool create

Create a new Linux pool with a virtual machine configuration.

az batch pool create \

--id mypool \

--vm-size Standard_A1 \

--target-dedicated 2 \

--image canonical:ubuntuserver:16.04-LTS \

--node-agent-sku-id "batch.node.ubuntu 16.04"

Step 2: az batch job create

Create a new job to encapsulate the tasks that are added.

az batch job create \

--id myjob \

--pool-id mypool

Step 3: az batch task create

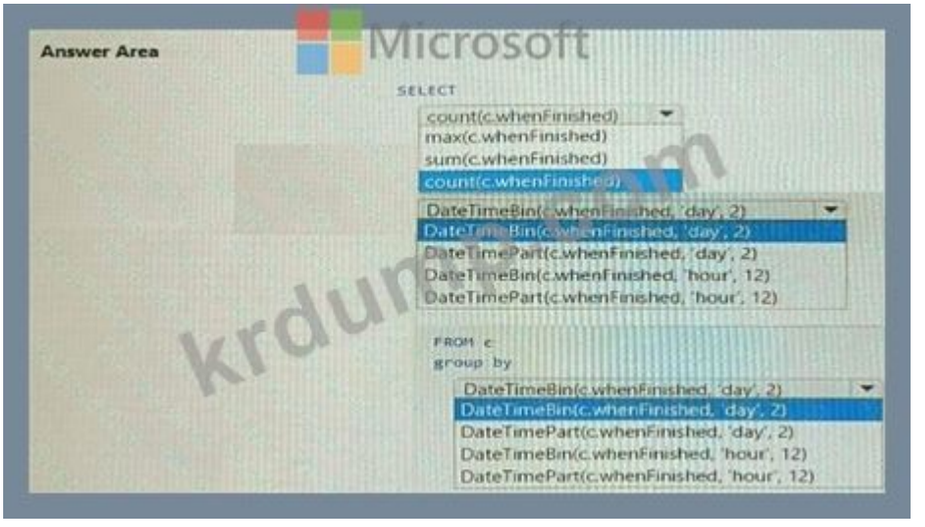
Add tasks to the job. Here the task is a basic shell command.

az batch task create \

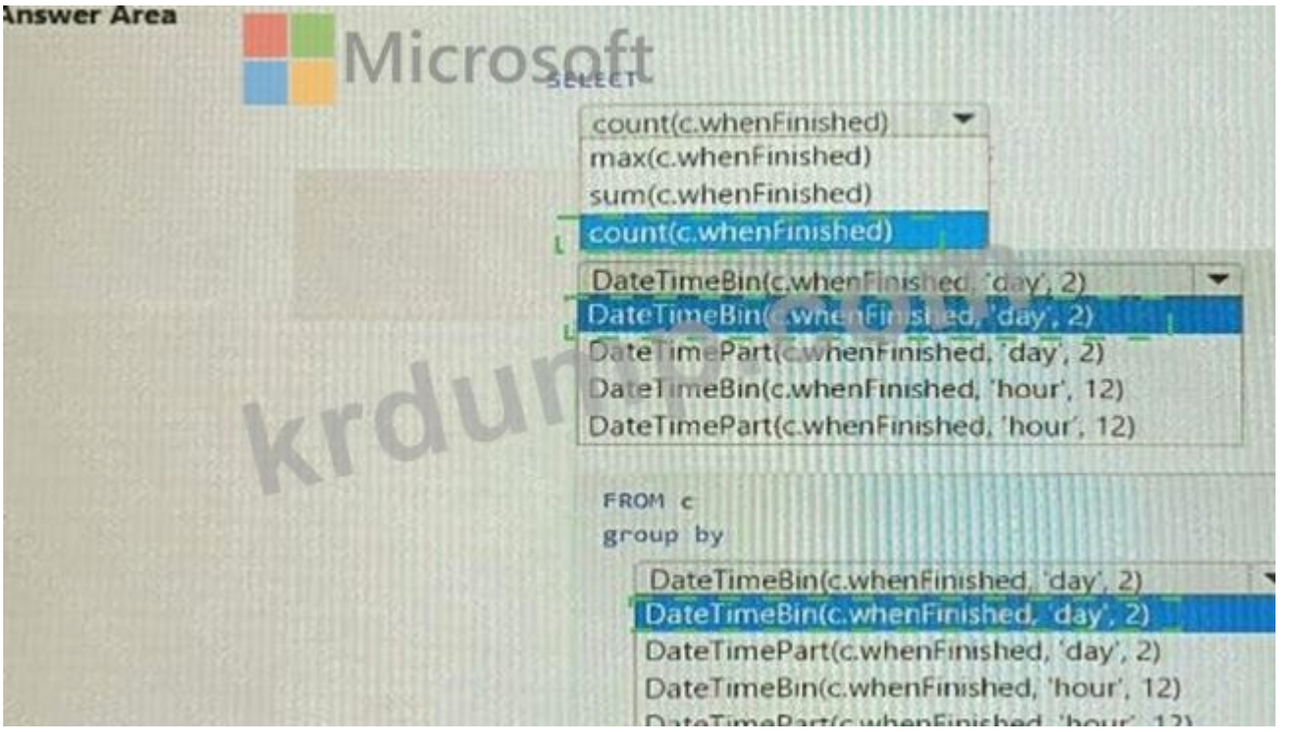
--job-id myjob \

--task-id task1 \
--command-line "/bin/bash -c 'printenv AZ_BATCH_TASK_WORKING_DIR'"
Step 4: for i in {1..$numberOfJobs} do
References:
https://docs.microsoft.com/bs-latn-ba/azure/batch/scripts/batch-cli-sample-run-job

NEW QUESTION: 162
□□□ □□□ □□□□ AI □□ □□□□ □□□□ □□□□□□□ □□□□□. □□□ □□ □□ □□ □□□ □□□ □□□□ □□□ □□□□ □□□□□□.
□□ □□□□ 'whenFinished'□□ □□□ □□□□ □ □□ □□□ □□ Azure Cosmos DB□ □ □□□□.
□□□ □□□□ □ □□ □□□ □□ □□ □□□□ □□□ □□□□□.
□□□ □□ □□□ □□□□ □□□.
□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

NEW QUESTION: 163

□□□ Azure □□□ □□□□ □□ □Ł□□□ □□ □□□ Azure□ □□□□□□□ □□□□ □. □□□ □□□(□□ Azure □□ □□ □□□)□ □□□□ □□□. □□□ □□□ □□□□ □□ □?

A. □□ □□□ □□

B. Azure □□□ □□ □

C. Azure □□ □

D. Azure □□ □□□□□ □□

Answer: (SHOW ANSWER)

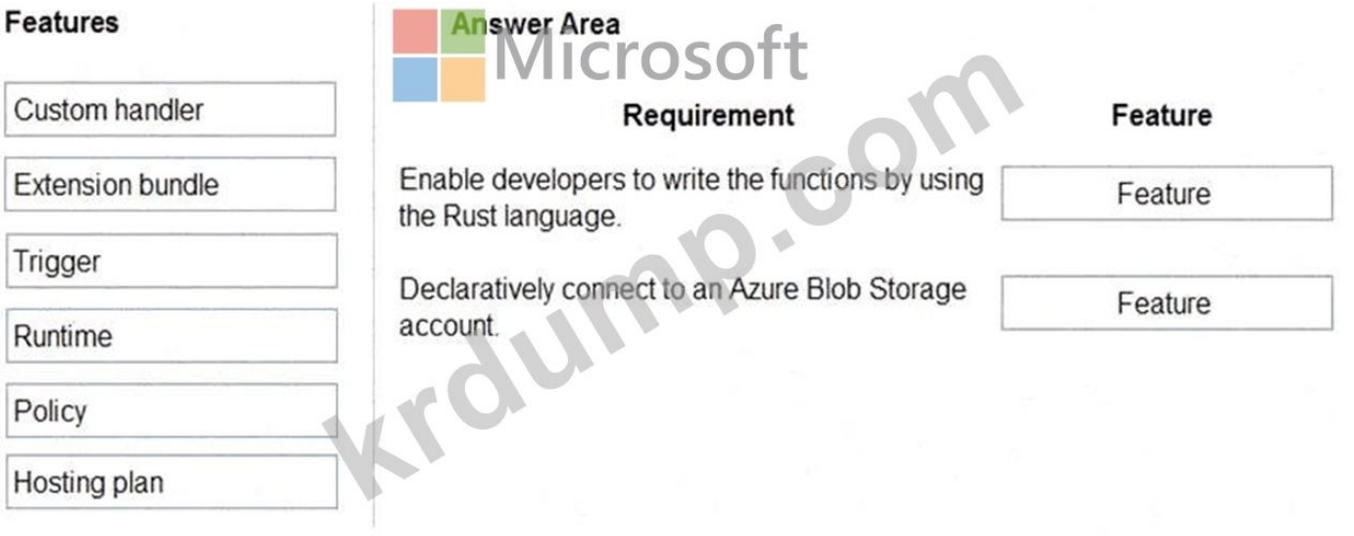
NEW QUESTION: 164

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□□ □□□□ □□□□. □□□□ □□□ □□□ □□□□□ □□□□□.

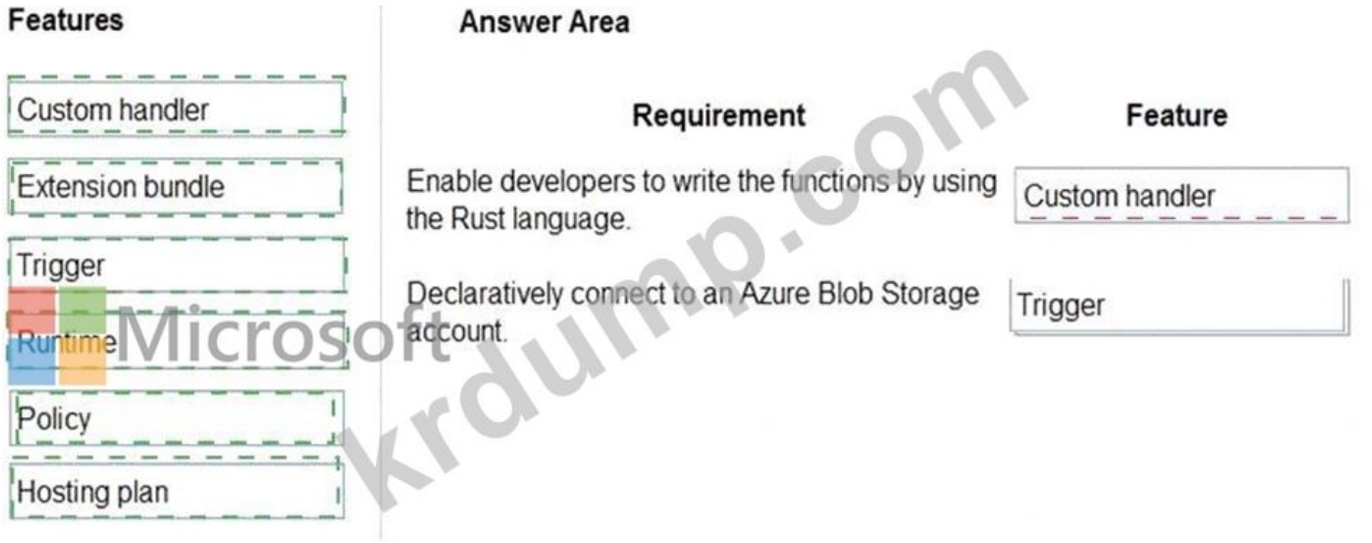
□□ ASP.Net □ □□□□□□□ □□□□ Azure App Service□ □□□□ □□□□. □□ □□ □ □□ HTML □□□ □□□ □□□□□. □□ □□ □□□ □□□□ □□□□ □□□□□ □□□□ □□□.

*□□ ASP.NET □ □□ □□□□□□ □□ □□ □□
*□□ □□□ □ □□ □□□□ □□ □□□ □□ □□ □□□□ □□ □□□ □□ □□□□ □□ □□□.
*□□ □□□ □□ □□ HTTP □□ □□
□□□ □□□□ □□□.
□□ □□□: Azure Database for PostgreSQL□ □□□□ □□□□□. □ □□□□□□□ □□□ □□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: B (LEAVE A REPLY)
Instead deploy and configure Azure Cache for Redis. Update the web applications.
Reference:
https://docs.microsoft.com/en-us/azure/architecture/best-practices/caching#managing-concurrency-in-a-cache

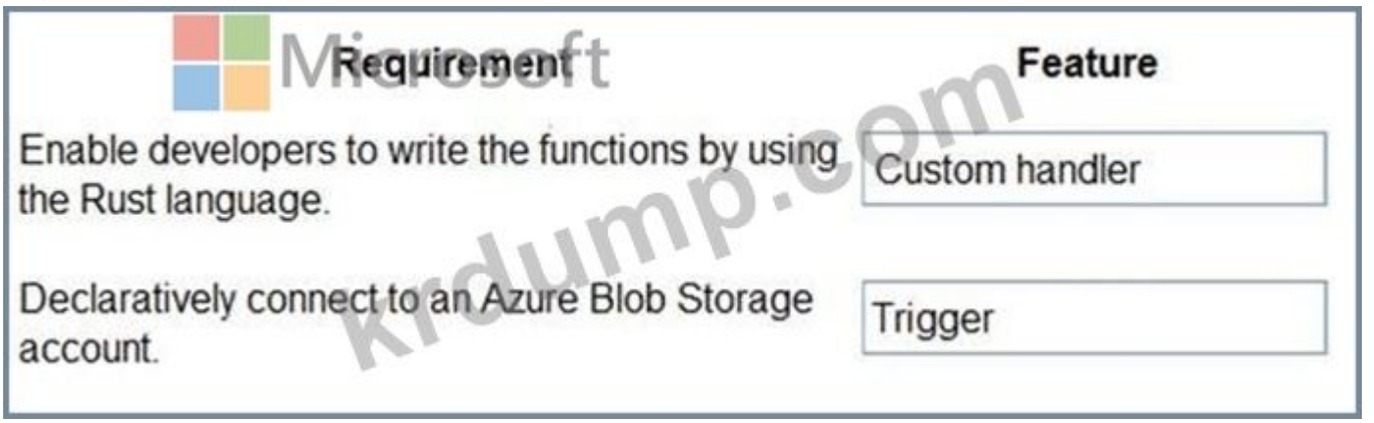
NEW QUESTION: 165
Azure □□ □□ □□□□ □□□□.
□□ □□ □□ □□□ □□□□ □□□.
* □□□□ Rust □□□ □□□□ □□□ □□□ □ □□□□.
* Azure Blob Storage □□□ □□□□□ □□□□□.
□□ □□□□ □□□.
□□ Azure □□ □ □□□ □□□□ □□□? □□□□□ □□□ □□□ □□□ □□ □□□□ □ □□□□□□. □ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Custom handler

Custom handlers can be used to create functions in any language or runtime by running an HTTP server process, for example Go or Rust.

Box 2: Trigger

Functions are invoked by a trigger and can have exactly one. In addition to invoking the function, certain triggers also serve as bindings. You may also define multiple bindings in addition to the trigger. Bindings provide a declarative way to connect data to your code.

Reference:

https://docs.microsoft.com/en-us/azure/azure-functions/create-first-function-vs-code-other

https://docs.microsoft.com/en-us/dotnet/architecture/serverless/azure-functions

NEW QUESTION: 166

□□□□ Java □ □□ □□ □□□□. □□ □□□ □□ □□□ □□ GitHub □□□□□□□ □□ □□□□.

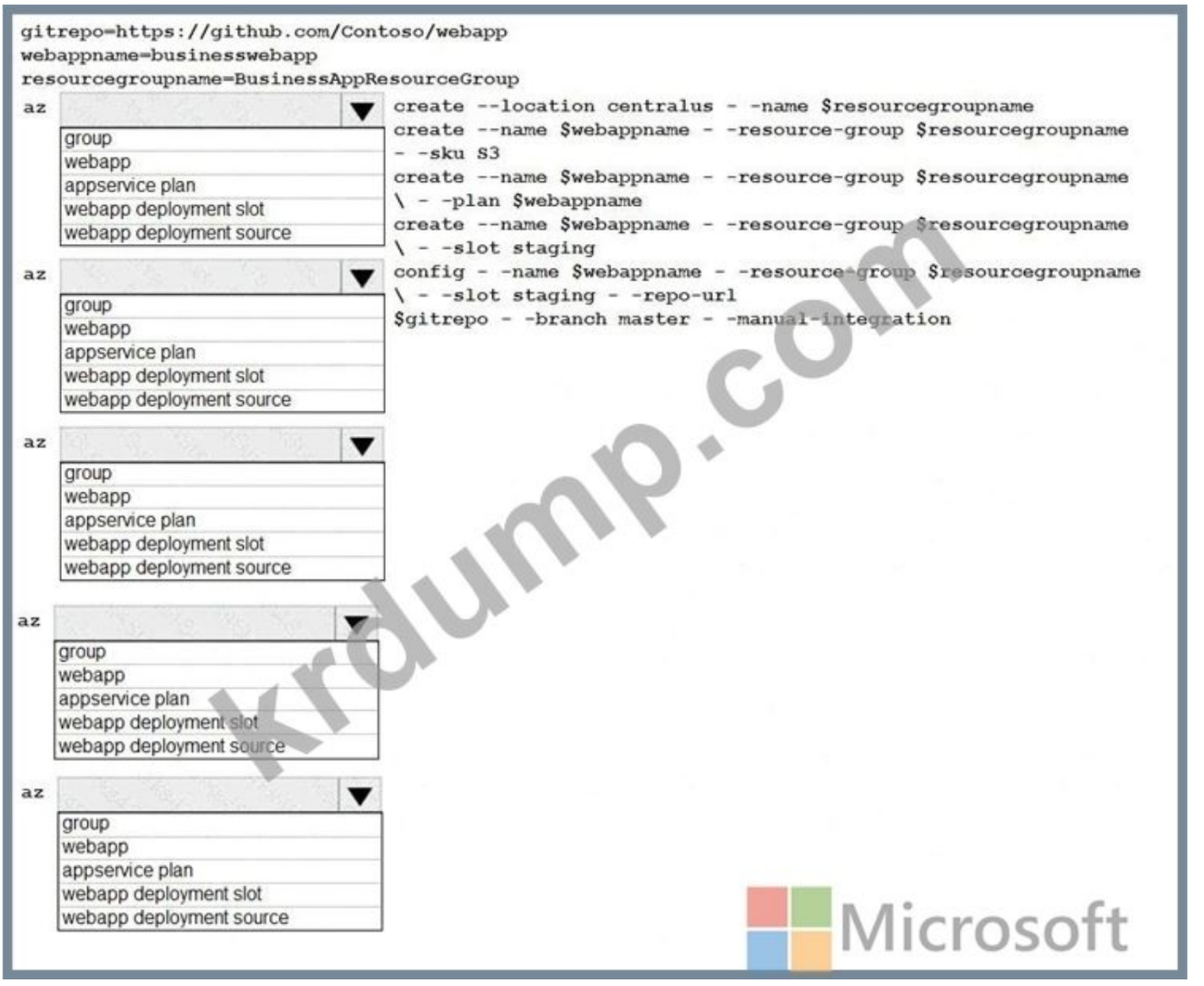
https://github.com/Contoso/webapp.

□□□□□□ □□□□ □□ □□□ □□□□ □□□. staging□□□ □□ □□□ □□ □□ □□ □□ □□□□ □□□.

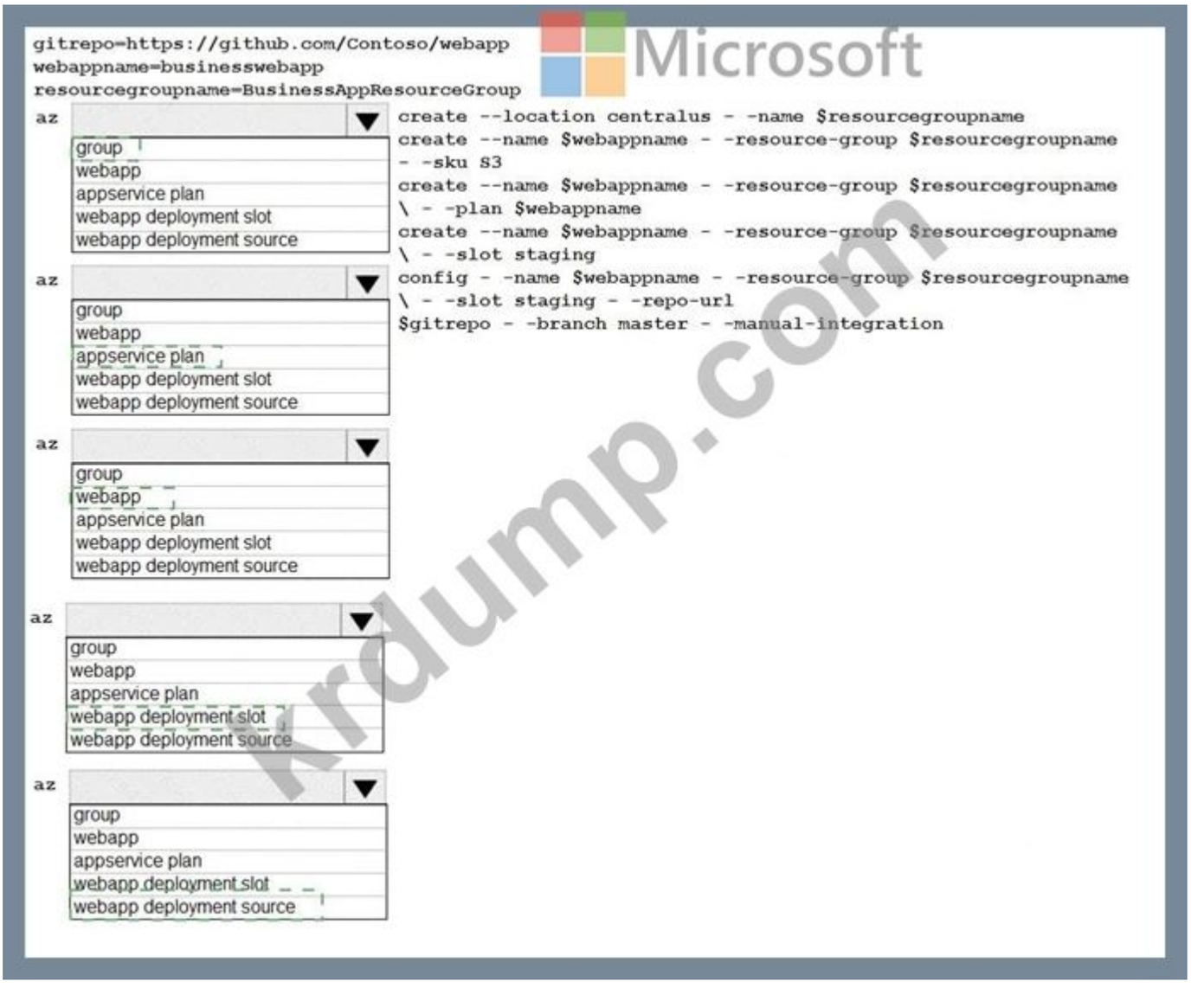
□□□ □□□ □□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

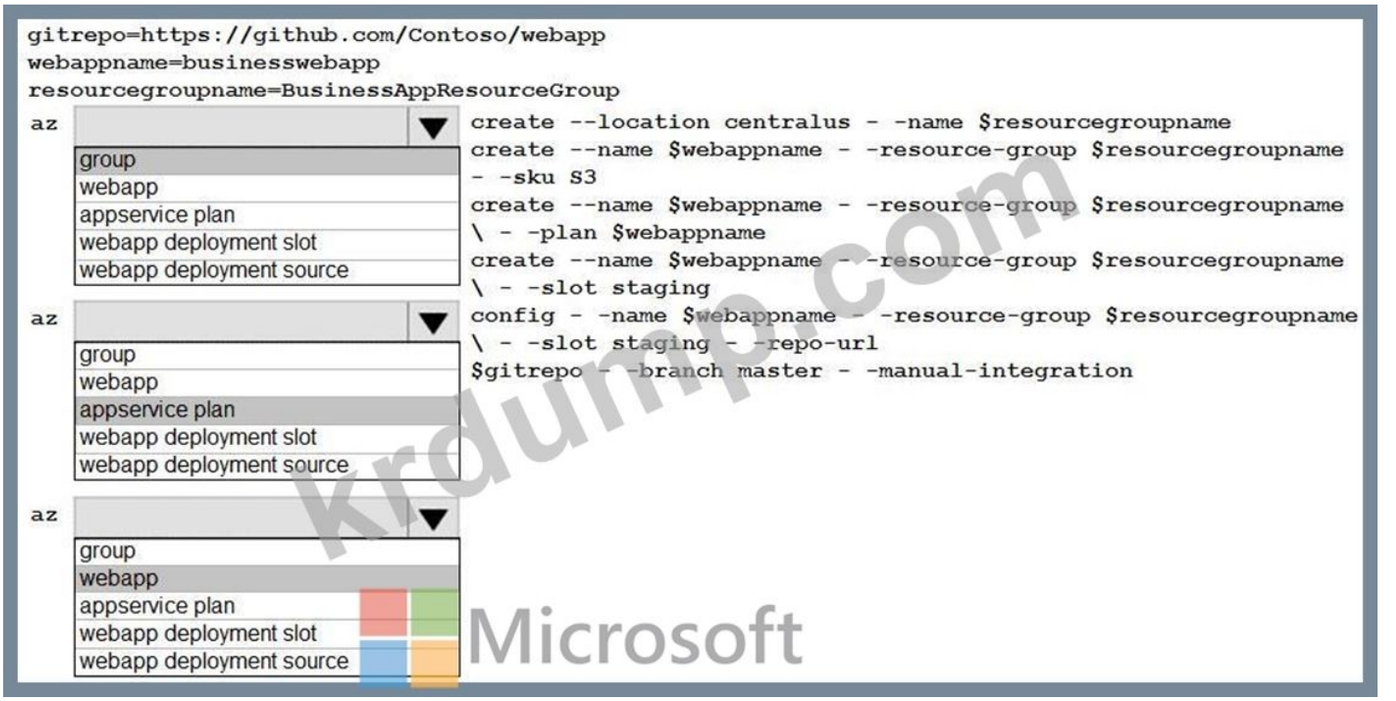
□□: □ □□□ □□□ 1□□ □□□ □□□□.

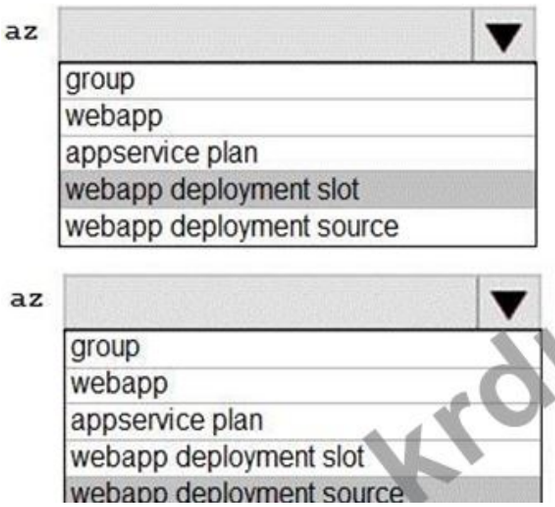


Answer:



Explanation:





Box 1: group

Create a resource group.

az group create --location westeurope --name myResourceGroup

Box 2: appservice plan

Create an App Service plan in STANDARD tier (minimum required by deployment slots).

az appservice plan create --name $webappname --resource-group myResourceGroup --sku S1

Box 3: webapp

Create a web app.

az webapp create --name $webappname --resource-group myResourceGroup \

--plan $webappname

Box 4: webapp deployment slot

#Create a deployment slot with the name "staging".

az webapp deployment slot create --name $webappname --resource-group myResourceGroup \

--slot staging

Box 5: webapp deployment source

Deploy sample code to "staging" slot from GitHub.

az webapp deployment source config --name $webappname --resource-group myResourceGroup \

--slot staging --repo-url $gitrepo --branch master --manual-integration References:

https://docs.microsoft.com/en-us/azure/app-service/scripts/cli-deploy-staging-environment

NEW QUESTION: 167

100□□ Azure App Service □□□ □□□ Azure □□□ □□□□□. □ □□□ □□ Application Insights □□□□□ □□□□□.
□ □□□ □□□ □□ Application Insights □□□□□□ □□□ □□□ □□□□ □□□ □□□ □□.
□□□ □□ PowerShell □□ □□□□.
Get-AzApplicationInsifihtsrtebTest | Where-Object {$condition}
Scondition □□□ □□ □□□□ □□□.
□□ □□ □□□□ □□□?

A.

```
$_.WebTestKind -eq "ping"
```

B.

```
$_.Type -eq "ping"
```

C.

```
$_.WebTestKind -eq "standard"
```

D.

```
$_.Type -eq "standard"
```

Answer: C (LEAVE A REPLY)

NEW QUESTION: 168
□□□ □□ □□□□□ □□□□ □□ Azure Durable Function□ □□□□ □□□□.
□□□□□ □□ □□ □□□ □□□□ □□ □□ API□ □□□□ □□□.
Azure Durable Function□ □□□□ □□□.
□□ Azure Durable Function □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□. □ □: □ □□ □□□ □□□ □□□□.
A. □□□□□□□
B. □□□
C. □□
D. □□□□□
Answer: (SHOW ANSWER)
https://learn.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-types-features-overview

NEW QUESTION: 169
□□ □□□ □□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□□□: □□ □□□ 1□□□□.
A. □□□ □□□ □□□□.
B. □□ □□□ □□□ □□□□□.
C. Microsoft Entra ID□ □□ □□□□□.
D. □□ □□□ □□(SAS) □□□ □□□□□.
E. □□ □□ □□□ □□(SAS) □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 170

Azure Storage□ □□□□ Azure □□ □□□□ □□□ □ □□□□ □□□□ □□□□. □□ ID □ □□□□□ □□ □□ □□□ □□□□□.

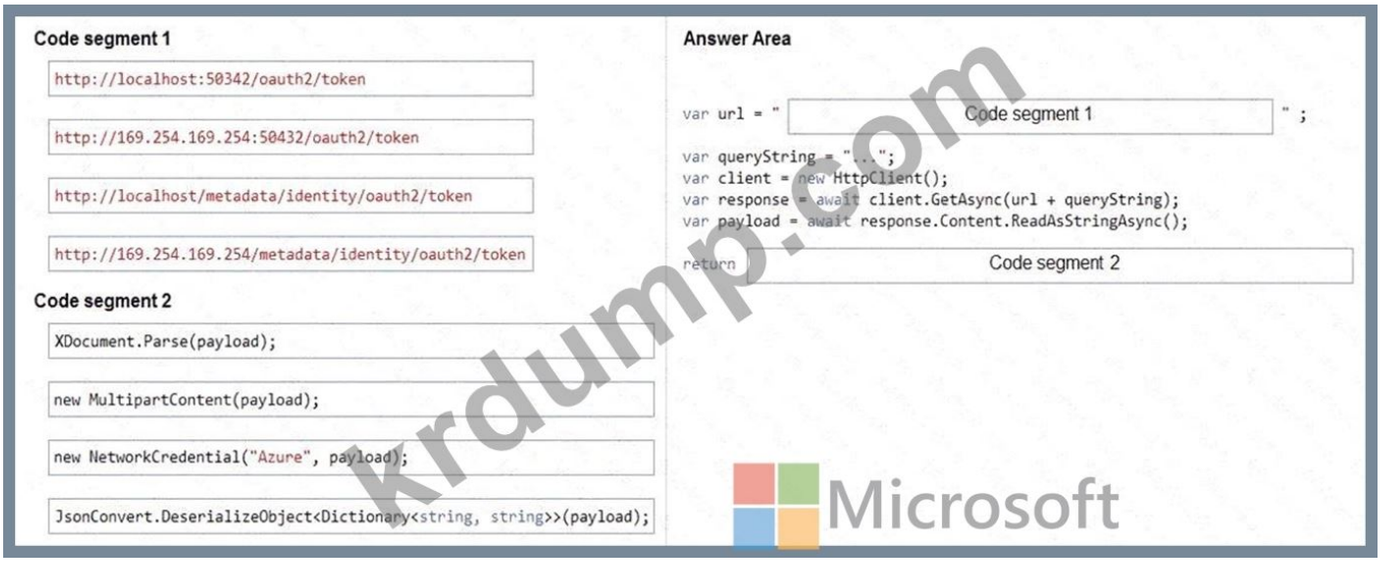
□□□ □□ □□ □□□ □□□□.

* □□ □□ □□ □□□□□ Azure Storage □□□ □□□□ □ □□□□.

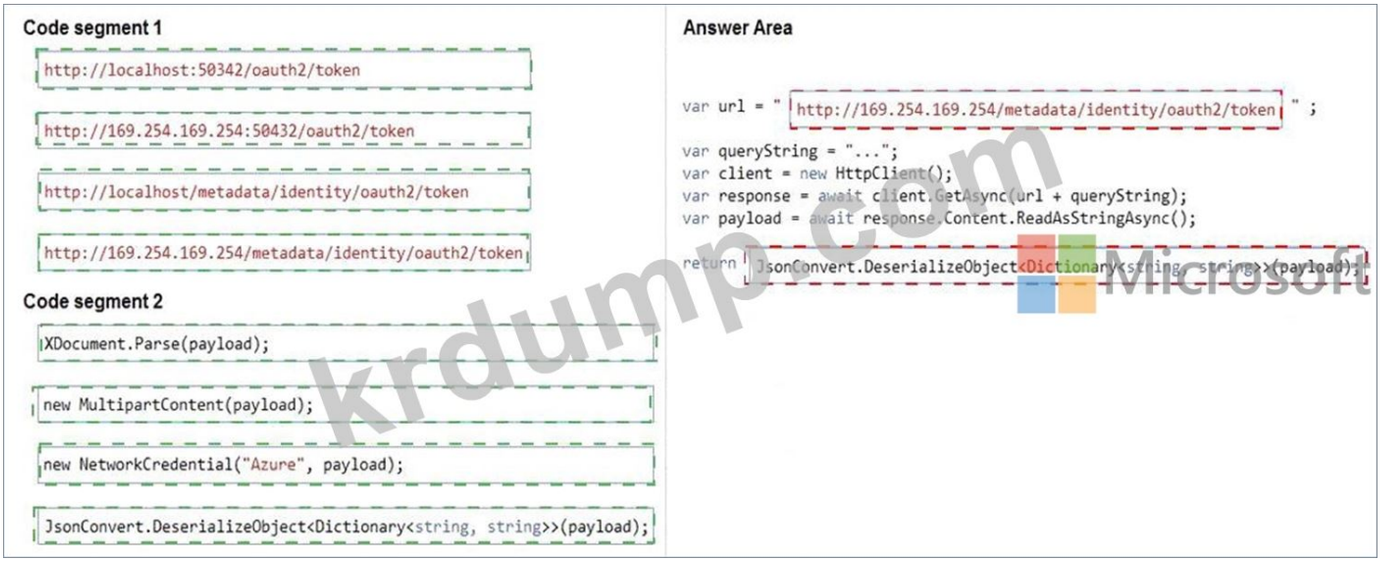
* Azure Instance Metadata Service □□□□□□ □□□□ □□□.

Azure □□□□ □□□□□□ □□□ □□□ □□□□ □□□ □□□□ □□□. □□□□□ □□ □ □□ □□□□□ □□□ □□□ □□□□□□□. □ □□ □□□□□ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□ □□.

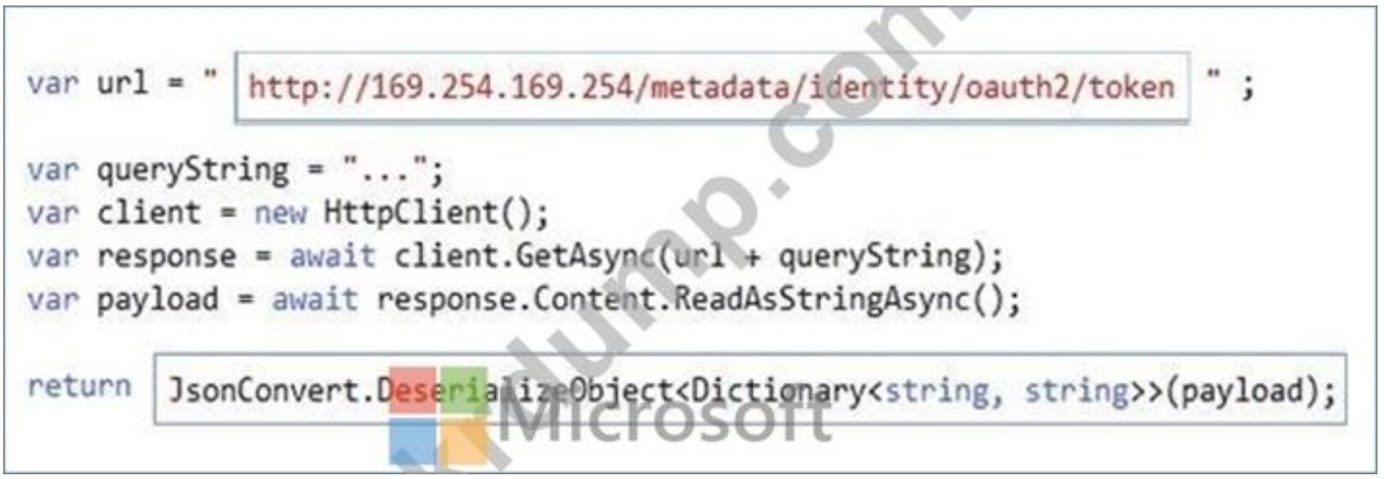
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Azure Instance Metadata Service endpoints "/oauth2/token"

Box

1: http://169.254.169.254/metadata/identity/oauth2/token

Sample request using the Azure Instance Metadata Service (IMDS) endpoint (recommended):

GET 'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01 &resource=https://management.azure.com/' HTTP/1.1 Metadata: true

Box 2: JsonConvert.DeserializeObject<Dictionary<string,string>>(payload); Deserialized token response; returning access code.

Reference:

https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/how-to-use-vm- token

https://docs.microsoft.com/en-us/azure/service-fabric/how-to-managed-identity-service-fabric-app-code

NEW QUESTION: 171

□□ □□ □□□ □□□□ □□ □□□ □□□□ □□□□ □□□□.

*□□ □□□ □□ □□ □□□□ □□ □□□ □□□□□ □□□□ □□ □□ □□ □□ □□□ □□□□ □□□.

*□□□ □□ □□ □□□□ □□□□ □□ □□□□□ □□ □□□□□ □□□.

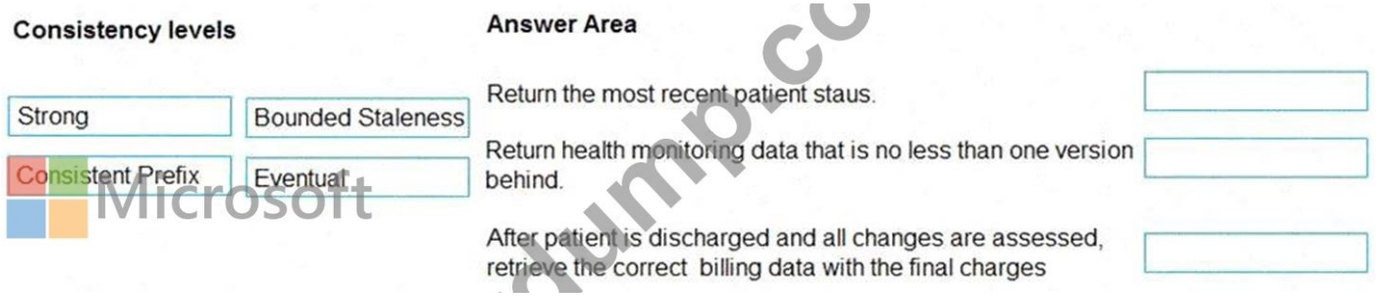
*□□□ □□□□ □□ □□□ □□□ □ □□ □□ □□□ □□ □□□ □□□□□.

Cosmos DB NoSQL □□□□□□□ □□□□□□□ □□□□□□ □□□ □□ □□□ □□□ Strong□□ □□□□□. □□□ □□□ □□ □□□□□ □□□□□.

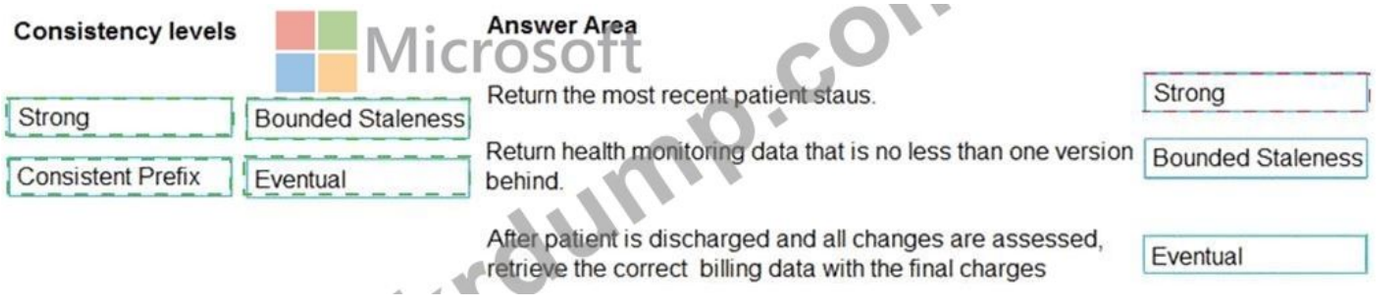
□□ □□□ □□□ □□□□ □□□ □□□ □□□□□ □□□. □□□□□ □□□ □□□ □□ □ □□□□□ □□ □□□□ □□ □□□ □□□ □□□□□ □□□.

□□ □□□ □□□ □□□□ □□□? □□□□□ □□□ □□□ □□□ □□□ □□ □□□□ □ □□□□□□. □ □□□ □□□ □ □, □□ □ □□□□□ □□ □□□□ □□ □ □□□□. □□ □□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

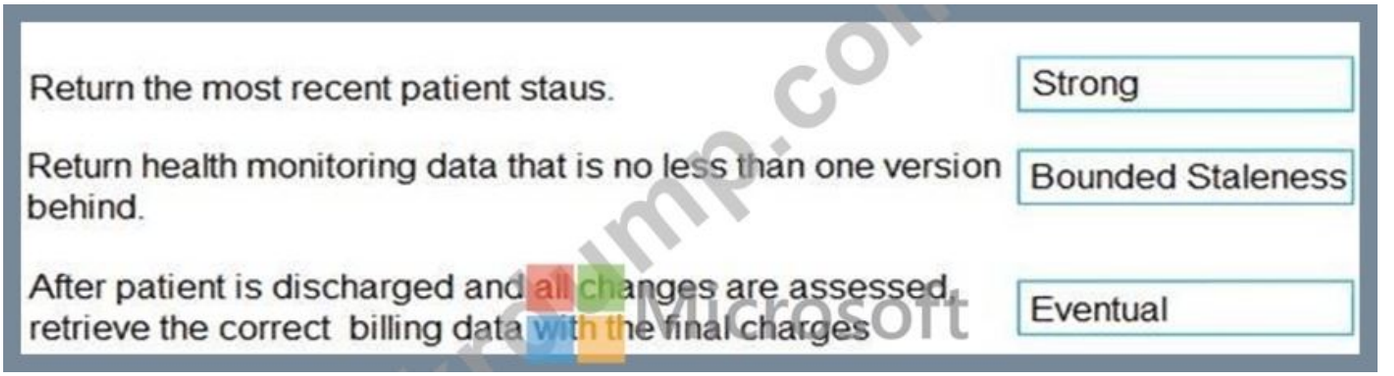
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Strong

Strong: Strong consistency offers a linearizability guarantee. 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: Bounded staleness

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. 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

Box 3: Eventual

Eventual: There's no ordering guarantee for reads. In the absence of any further writes, the replicas eventually converge.

References:

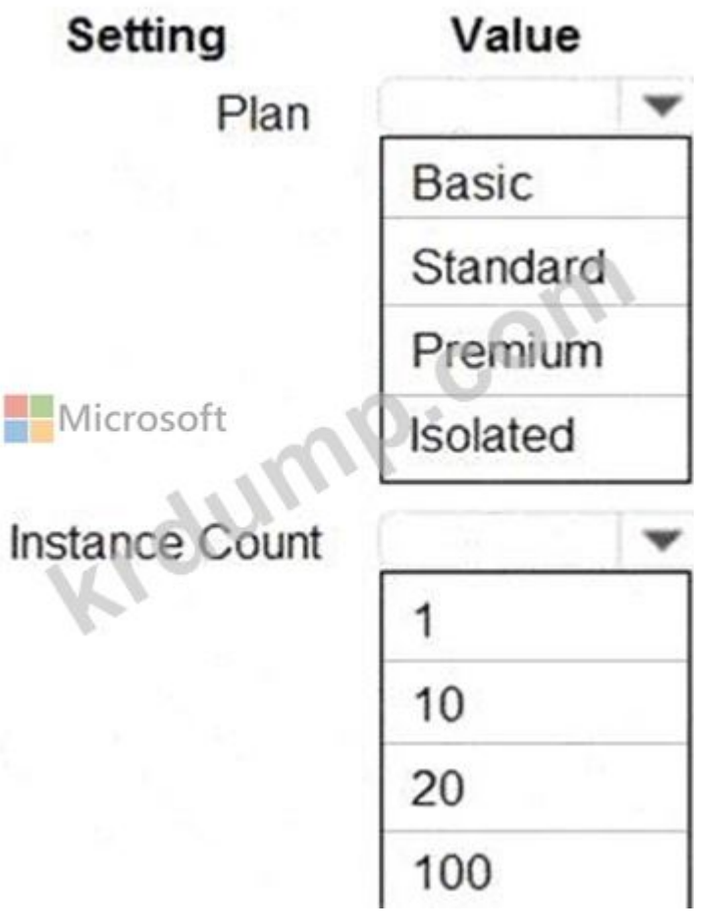
https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

NEW QUESTION: 172

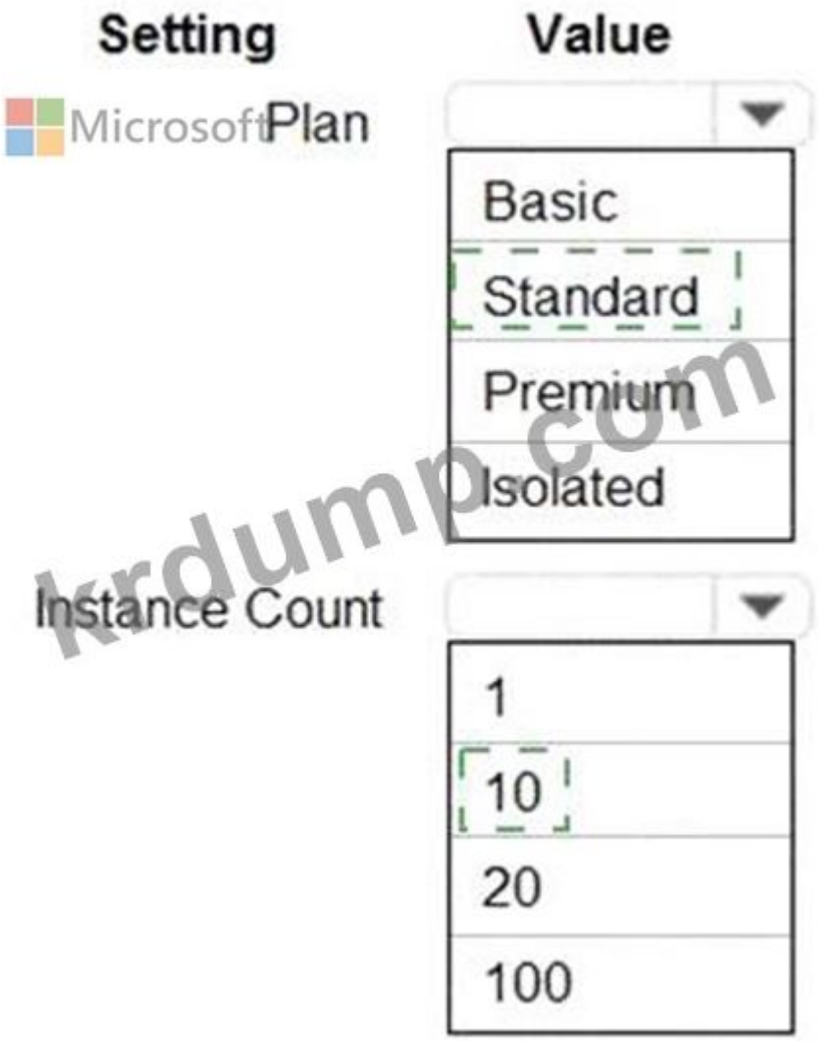
REST API □□ □□□ □□□□□ Azure App Service□ □□□□ □□□.

□□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

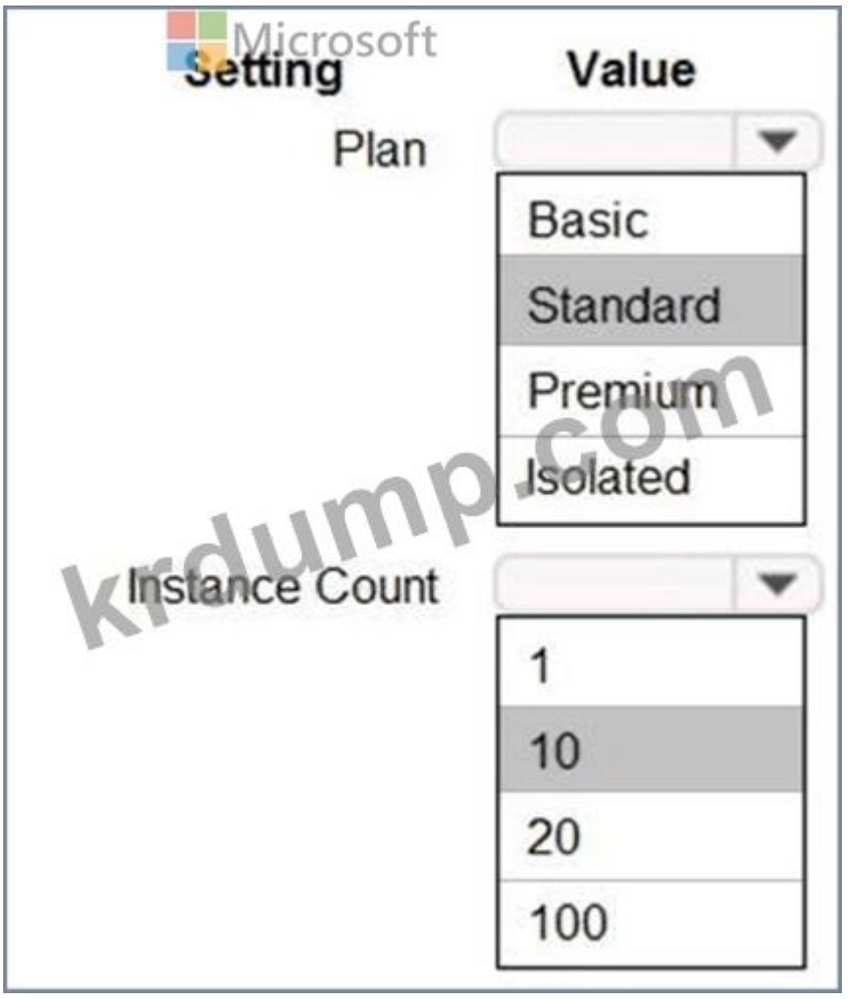
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



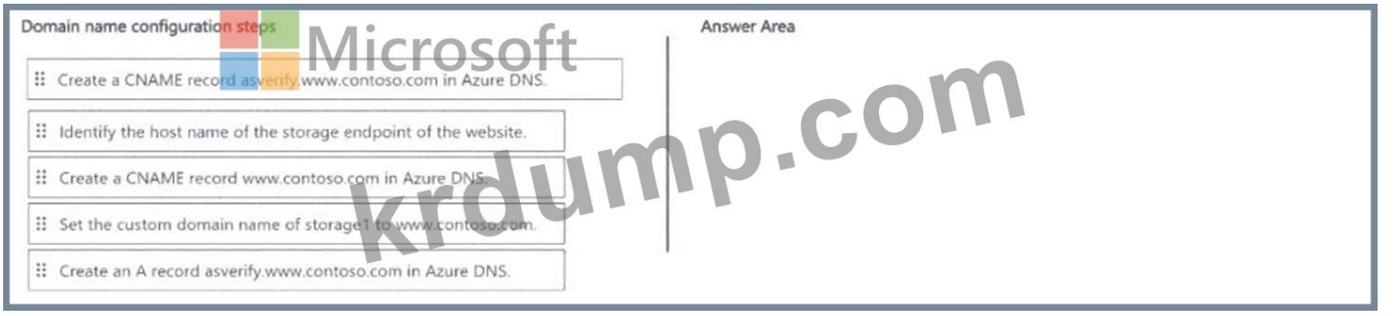
Explanation:



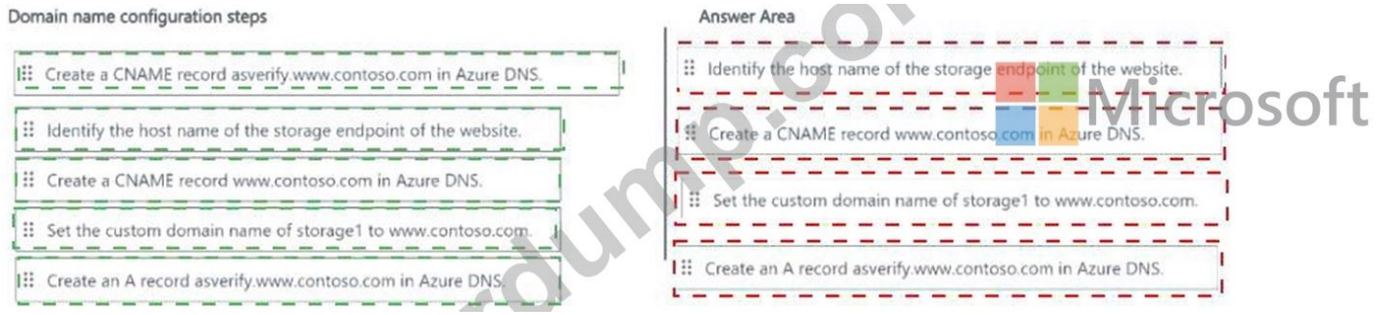
Plan: Standard
Standard support auto-scaling
Instance Count: 10
Max instances for standard is 10.
Scenario:
The REST API's that support the solution must meet the following requirements:
Allow deployment to a testing location within Azure while not incurring additional costs.
Automatically scale to double capacity during peak shipping times while not causing application downtime.
Minimize costs when selecting an Azure payment model.
References:
https://azure.microsoft.com/en-us/pricing/details/app-service/plans/

NEW QUESTION: 173
Azure Storage □□□ storage1□ □□□□ □□ □□□□□ □□□□. web.core.windows.net □ □□□ □□□ URL□ □□□□ □□□□□ □□□□□□.
URL www.contoso.com□ □□ □□□□□ □□□□ □ □□□ □□□ □□□□□. □□□□□ □ □ □□ □□□□ □ □□□ □□□.
contoso.com □□□ Azure DNS□ □□□□□□.
□□□□ □□□ □□□□ □□□.

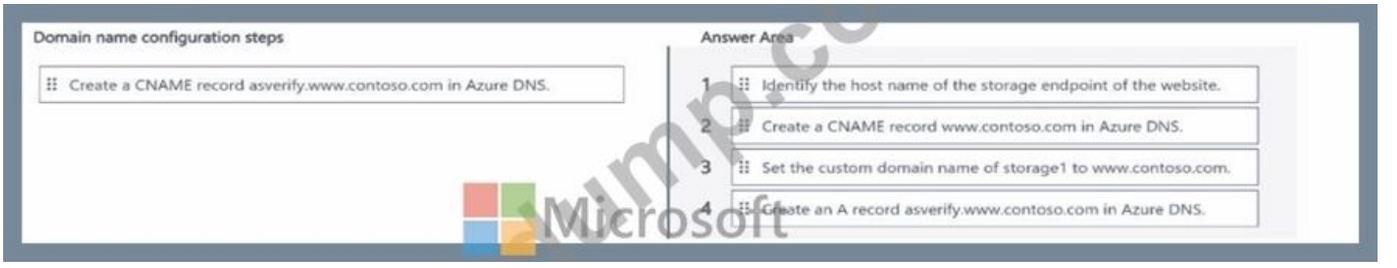
□□ □ □□ □□□ □□□□ □□□□ □□□? □□□□□ □□□ □□ □□ □□ □□□□ □□ □□□ □□ □□ □□□ □□ □□□□ □□□ □□□ □□□ □□□□□.



Answer:



Explanation:



NEW QUESTION: 174

□□□ □□□ □□□□□□□ □□□□□ □□□□ □□□ □□□□. □□□ □□□□ □□□ □ □ □□□ □□□□ □□□ □□□□□. □□ □□□□ □□□ □□ □□□□.

□□□□□□□ Entity Framework□ □□□□ Azure □□□□□□□ □□□□□. □□□□□□ □□ Player □□□□ Game □□□□ □□□□.

□□□□□ □□□ □ □□□ □ □□□□ □□□□ □□□□ □□ □□ □□□□ □ □□□□ □ □□ □□ □□□ □□□□ □□□.

□□□□□□□ □□□□□ □□□□ □□ □□□ gameId □ playerId□ CreatePlayerWithGame □ □□□□□.

(□ □□□ □□□□□□ □□□□ □□□□.)

```
01. namespace ContosoCradt
02. {
03.    public class PlayerDbContext : DbContext
04.    {
05.  public PlayerDbContext() : base ("name-dBConnString") { ]
06.  public DbSet<Player> Players { get ; set ; }
07.  public DbSet<Game> Games { get ; set ]
08.  protected override void DmModelCreating{DBModelBuilder modelBuilder}
09.  {
10.   modelBuilder.Entity<Player>().MesMany(x => x.Games). WithMany (x => x Players);
11.  }
12.  }
13.   internal series class dbConfiguration : DbMigrationConfiguration<PlayerDbContext>
14.  {
15.  public dbConfiguration[] . {AutomaticMigrationsEnabled = true ; }
16.  {
17.    public class mp
18.    {
19.  public void CreatePlayerWithGame(int playerId, int gameId) => AddPlayer(playerId, GetGame[gameId));
20.  public game GetGame(int gameId)
21.  {
22.    using (var db = new PlayerDbContext())
23.  {
24.    return db.Games.FirstOrDefault(x => x.GameId == gameId);
25.  }
26.   }
27.   public Player AddPlayer (int playerId, Game game)
28.   {
29.  using (var db = new PlayerDbContext())
30.  {
31.    var player = new Player
32.     {
33.      PlayerId = playerId,
34.      Games = new List <Game> {game },
35.  };
36.  db.Players.Add(player);
37.  db.SaveChanges();
38.  return player;
39.    }
40. }
41. public class Player
42. {
43.  public int PlayerId { get ; set; }
44.  public string PlayerName { get ; set; }
45.  public virtual List<Game> Games { get ; set; }
46.  }
47.  public class Game
48.  {
49.  public int GameIs { get ; set }
50.  public string Title { get ; set; }
51.  public string Platform { get ; set; ]
52.  public virtual List<Player> Players { get ; set; }
53.     }
54.  }
```


□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:

Explanation:

Many-to-many relationships without an entity class to represent the join table are not yet supported. However, you can represent a many-to-many relationship by including an entity class for the join table and mapping two separate one-to-many relationships.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
modelBuilder.Entity<PostTag>()
HasKey(t => new { t.PostId, t.TagId });
modelBuilder.Entity<PostTag>()
HasOne(pt => pt.Post)
WithMany(p => p.PostTags)
HasForeignKey(pt => pt.PostId);
modelBuilder.Entity<PostTag>()
HasOne(pt => pt.Tag)
```

```
WithMany(t => t.PostTags)
HasForeignKey(pt => pt.TagId);
}
}
```

NEW QUESTION: 175

Azure Functions □□ □□□ □□□□□.

Azure Functions □□□ □□□ □□ □□ □□□ □□□□.

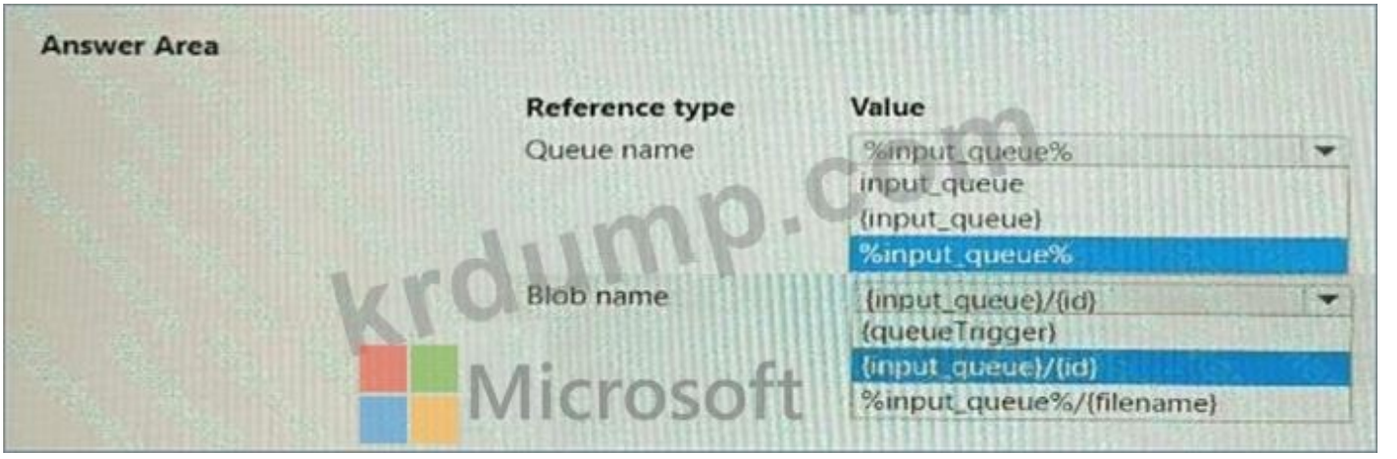
* Azure Storage □□ □□□ □□□□ □□ □□□□□□ □□□.

* input-queue□□ □ □□□□ □□□ □□□ □□□ □□□□ □□□.

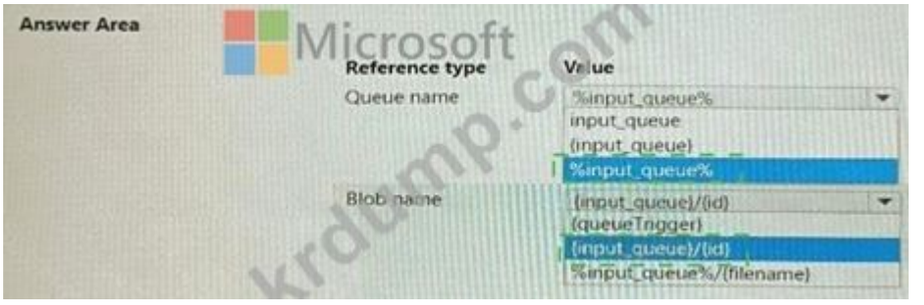
* □□□ □□□ □□□ □□□ Azure Blob Storage□ □□□□ □□□.

□□□□ □□□ □ Blob □□□ □□□□ □□□ □□□□ □□□. Azure Functions □□ □□□ □□.

□□□ □□□ □□□□ □□□? □□□□□ □□□□ □□□ □□ □□□□□□. □□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

NEW QUESTION: 176

Azure Blob Storage□□ □□□□ □□□□ □□□□ □□□□□□□ □□□□ □□□□. □□□ □□□□ □□□□□ □□ □□(VM)□□ □□□□□□. VM□ □□□ □ VPN □□□□□ □□□ □□□□ Azure□ □□□□□. □□□□□□□ Azure AD(Azure Active Directory) □□ □□□ □ □□□ □□□□□.

□□□□□□□□ □□ □□, □□ □□ □ □□ □□□ □□ Azure Blob □□□□ □□□ □□ □ □□ □□□ □□□□□ □□□. Azure Blob □□□□ □□ □□□□ □□□□□□□ Azure AD □ □ □□□ □□□□ □□□ □□□□ □□□□ □□□. □□□□□ □□□□□□ □□□ □□□ □□ □□□ □□□□ □□□ □ □□□ □□□.

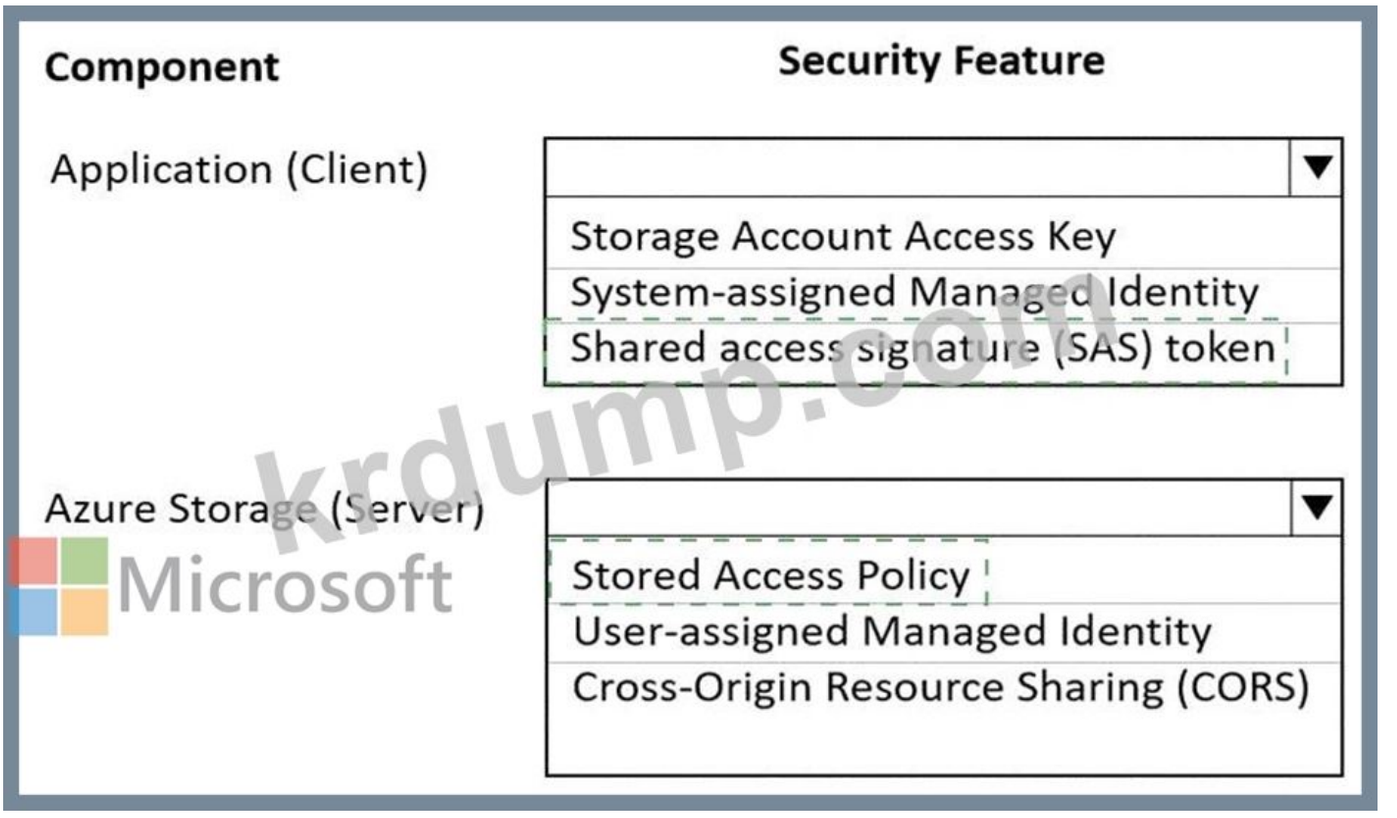
Azure Blob □□□□□ □□ □□□□□□ □□□□ □□□□ □□□.

□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

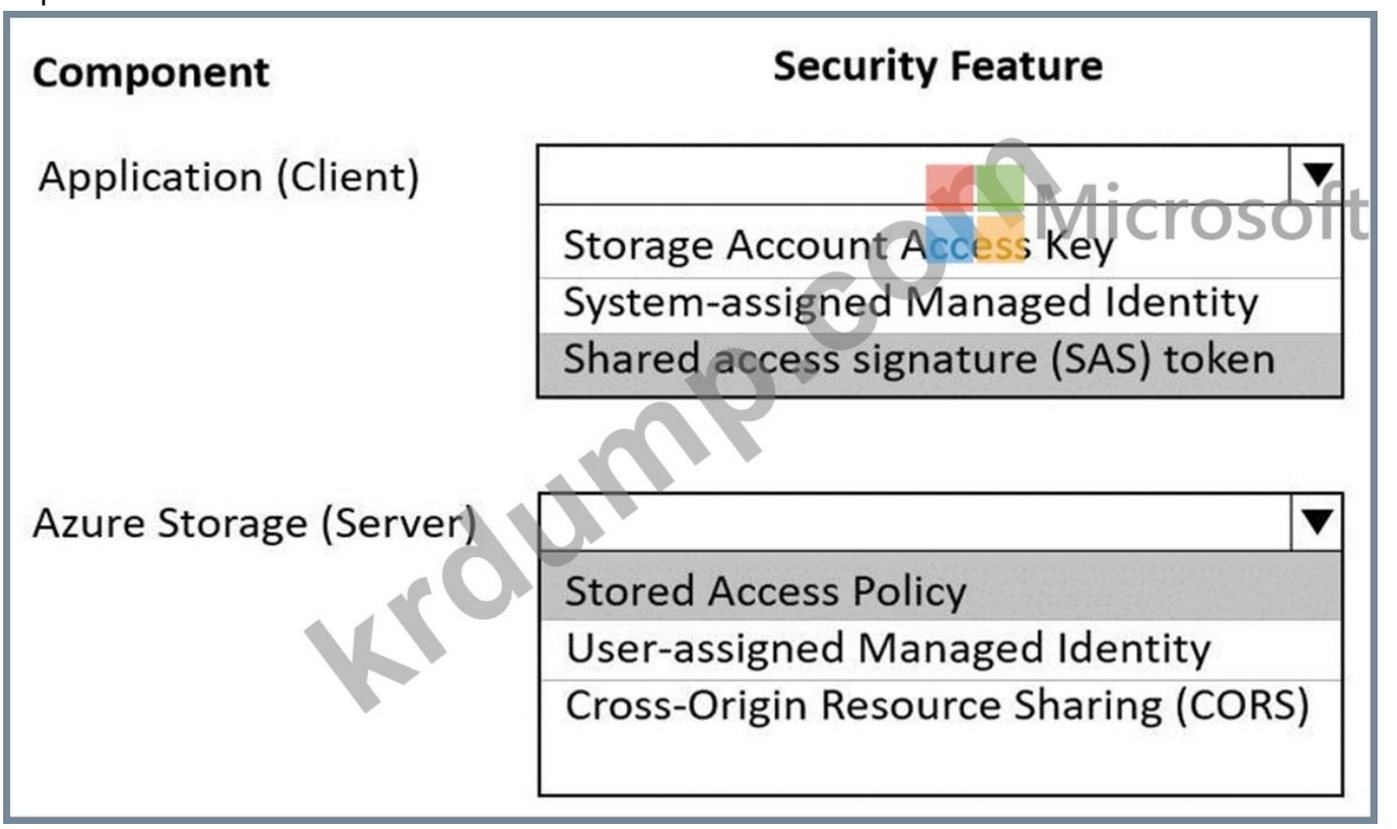
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Component	Security Feature
Application (Client)	Storage Account Access Key System-assigned Managed Identity Shared access signature (SAS) token
Azure Storage (Server)	Stored Access Policy User-assigned Managed Identity Cross-Origin Resource Sharing (CORS)

Answer:



Explanation:



Box 1: Shared access signature (SAS) token

When your application design requires shared access signatures for access to Blob storage, use Azure AD credentials to create a user delegation SAS when possible for superior security.

Box 2: Stored access policy

Stored access policies give you the option to revoke permissions for a service SAS without having to regenerate the storage account keys.

A shared access signature can take one of the following two forms:
Service SAS with stored access policy. A stored access policy is defined on a resource container, which can be a blob container, table, queue, or file share. The stored access policy can be used to manage constraints for one or more service shared access signatures. When you associate a service SAS with a stored access policy, the SAS inherits the constraints - the start time, expiry time, and permissions - defined for the stored access policy.
Ad hoc SAS.
Reference:
https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

NEW QUESTION: 177
Azure App Service□□ □□□□□ ASP.NET Core □□ □□□□ □□□□.
□□ □□□ □□□ □□ Microsoft Entra ID□□ □□□ □□ □□□□ □□□□ □□□. □ □□ □ □□□□ □□□□ □□□□ □□□. □□□ □□□ □□□ □□□ □□ □□□□ □□□□ □ □□. □□□ □□ □□□?
A. □□ □□ https://graph.microsoft.eom/.default □□□ □□□□□.
B. □ □□□□□□ appRoles □□□ □□□ □□□□□.
C. □ □□□□□□ groupMembershipClaims □□□ □□□ □□□□□.
D. Microsoft Entra ID□□ □□ □□□ □□□□ □□ □□□ □□ □□□□□ □□□□□.
E. OAuth 2.0 □□ □□ □□□ □□□□□ □□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 178
□ □□□□ □□□ □□□□ □□ □□□ □□ □□□ □□ □ □□ □□□□ □□□□ □□□□.
□□ Service Bus□ □□□□.
□□□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□ □□□. □□, □□ □□ □□, □□, □□□ □□ □ □□□ □□□□ Azure Service Bus □□□□□ □□□□.
□□□ □□□□ □□□.
□□ Azure CLI □□ PowerShell □□□ □□□□ □□□?

```
az servicebus namespace create
    - -resource-group fridge-rg
    - -name fridge-ns
    - -location fridge-loc

az servicebus queue create
 --resource-group fridge-rg
 --namespace-name fridge-ns
 --name fridge-q

connectionString-$(az servicebus namespace authorization-rule keys list
    --resource-group fridge-rg
    --fridge-ns fridge-ns
    --name RootManageSharedAccessKey
    --query primaryConnectionString --output tsv)

 az group create
    --name fridge-rg
    --location fridge-log
```


A. □□ A
B. □□ B
C. □□ C
D. □□ D

Answer: B (LEAVE A REPLY)

A service bus instance has already been created (Step 2 below). Next is step 3, Create a Service Bus queue.

Note:

Steps:

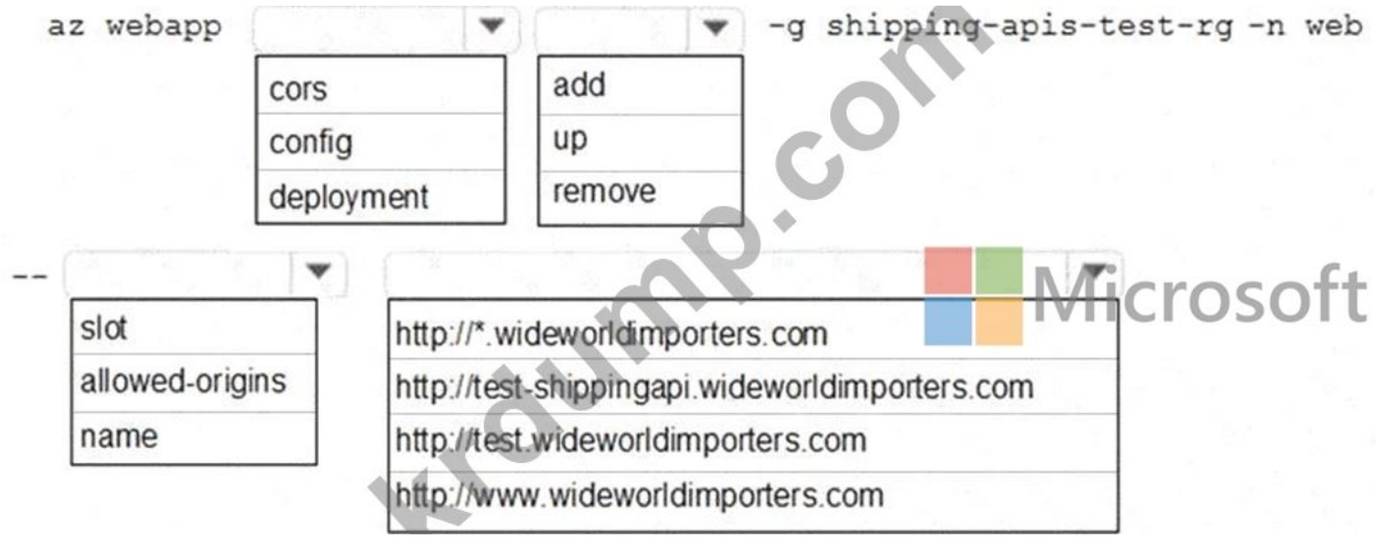
Step 1: # Create a resource group
resourceGroupName="myResourceGroup"
az group create --name $resourceGroupName --location eastus
Step 2: # Create a Service Bus messaging namespace with a unique name
namespaceName=myNameSpace$RANDOM
az servicebus namespace create --resource-group $resourceGroupName --name $namespaceName --location eastus Step 3: # Create a Service Bus queue az servicebus queue create --resource-group $resourceGroupName --namespace-name $namespaceName -- name BasicQueue Step 4: # Get the connection string for the namespace connectionString=$(az servicebus namespace authorization-rule keys list --resource-group $resourceGroupName --namespace-name $namespaceName --name RootManageSharedAccessKey --query primaryConnectionString --output tsv) References: https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-quickstart-cli

NEW QUESTION: 179

□□□ □□□ □□□□□ API□ □□□□□□ □□□.
Azure CLI □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□

□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:

Explanation:

Enable Cross-Origin Resource Sharing (CORS) on your Azure App Service Web App.

Enter the full URL of the site you want to allow to access your WEB API or * to allow all domains.

Box 1: cors

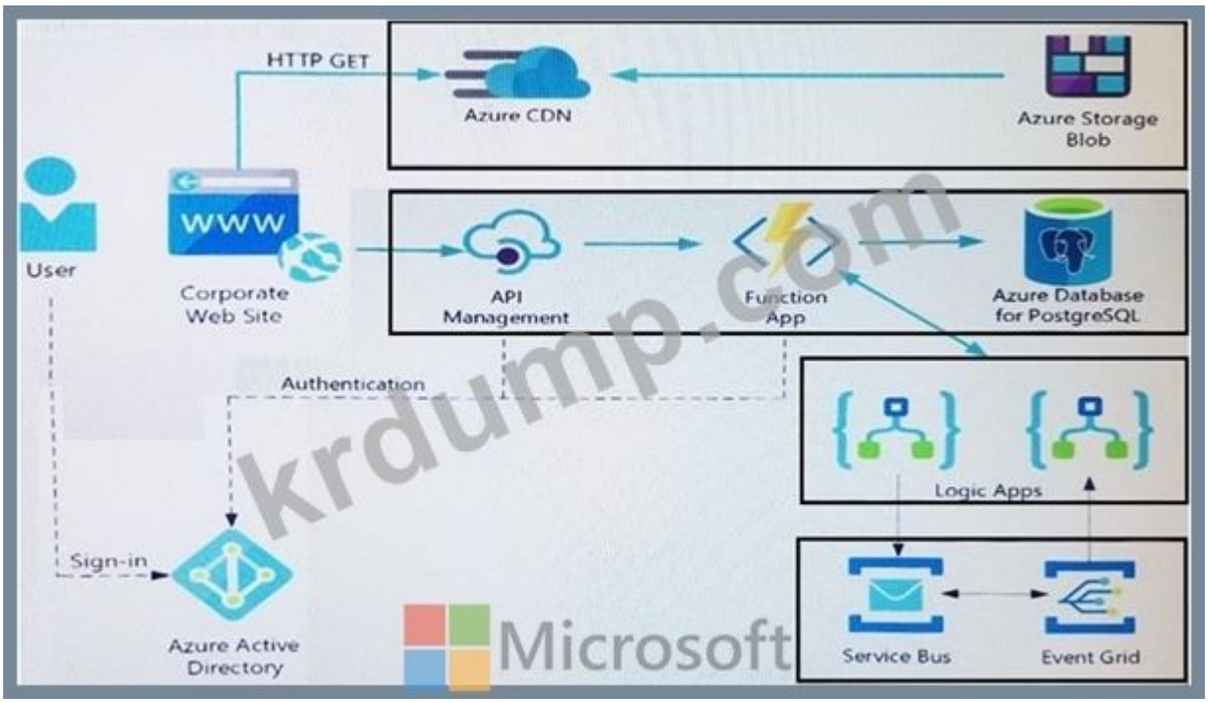
Box 2: add

Box 3: allowed-origins
Box
4: http://testwideworldimporters.com/
References:
http://donovanbrown.com/post/How-to-clear-No-Access-Control-Allow-Origin-header-error-with-Azure-App- Service

NEW QUESTION: 180
□□□ API □□□ □□ □□ □□ □□ □□□ □□□□ □□□.
□□ Application Insights API □□□ □□□□ □□□?
A. □□ □□
B. □□ □□□
C. □□□□□
D. □□ □□□
E. □□ □□
Answer: C (LEAVE A REPLY)

NEW QUESTION: 181
Shipping Logic App□ □□□□ □□□.
□□□ □□□□ □□□?
A. □□ □ □□□ □□(ASE)
B. Azure AD B2B □□
C. □□ □□□ □□(ISE)
D. VNet □□□ □□
Answer: (SHOW ANSWER)
Scenario: The Shipping Logic App requires secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
You can access to Azure Virtual Network resources from Azure Logic Apps by using integration service environments (ISEs).
Sometimes, your logic apps and integration accounts need access to secured resources, such as virtual machines (VMs) and other systems or services, that are inside an Azure virtual network.
To set up this access, you can create an integration service environment (ISE) where you can run your logic apps and create your integration accounts.
References:
https://docs.microsoft.com/en-us/azure/logic-apps/connect-virtual-network-vnet-isolated-environment- overview
Topic 3, City Power & Light
Case study
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all

questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.
To start the case study
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. When you are ready to answer a question, click the Question button to return to the question.
Background
City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.
Current environment
Architecture overview
The
company has a public website located at http://www.cpandl.com/. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.
API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.
The solution uses Application Insights, Azure Monitor, and Azure Key Vault.
Architecture diagram
The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

The user selects Sign in in the website.

The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.

The user signs in.

Azure AD redirects the user's session back to the web application. The URL includes an access token.

The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.

The back-end API validates the access token.

Requirements

Corporate website

Communications and content must be secured by using SSL.

Communications must use HTTPS.

Data must be replicated to a secondary region and three availability zones.

Data storage costs must be minimized.

Azure Database for PostgreSQL

The database connection string is stored in Azure Key Vault with the following attributes:

Azure Key Vault name: cpandlkeyvault

Secret name: PostgreSQLConn

Id: 80df3e46ffcd4f1cb187f79905e9a1e8

The connection information is updated frequently. The application must always use the latest information to connect to the database.

Azure Service Bus and Azure Event Grid

Azure Event Grid must use Azure Service Bus for queue-based load leveling.

Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.

Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

All SSL certificates and credentials must be stored in Azure Key Vault.

File access must restrict access by IP, protocol, and Azure AD rights.

All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

Auditing of the file updates and transfers must be enabled to comply with General Data Protection Regulation (GDPR). The file updates must be read-only, stored in the order in which they occurred, include only create, update, delete, and copy operations, and be retained for compliance reasons.

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

You perform local testing for the RequestUserApproval function. The following error message displays:

'Timeout value of 00:10:00 exceeded by function: RequestUserApproval'

The same error message displays when you test the function in an Azure development environment when you run the following Kusto query:

FunctionAppLogs

| where FunctionName = = "RequestUserApproval"

Logic app

You test the Logic app in a development environment. The following error message displays:

'400 Bad Request'

Troubleshooting of the error shows an HttpTrigger action to call the RequestUserApproval function.

Code

Corporate website

Security.cs:

```
SC01 public class Security
SC02 {
SC03 var bytes = System.IO.File.ReadAllBytes("~/var/ssl/private");
SC04 var cert = new System.Security.Cryptography.X509Certificate2(bytes);
SC05 var certName = cert.FriendlyName;
SC06 }
```

Function app

RequestUserApproval.cs:

```
RA01 public static class RequestUserApproval
RA02 {
RA03 [FunctionName("RequestUserApproval")]
RA04 public static async Task<IActionResult> Run(
RA05 [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req.
ILogger log)
RA06 {
RA07    log.LogInformation("RequestUserApproval function processed a request.");
RA08 ...
RA09 return ProcessRequest(req)
RA10    ? (ActionResult)new OkObjectResult($"User approval processed")
RA11    : new BadRequestObjectResult("Failed to process user approval");
RA12 }
RA13 private static bool ProcessRequest(HttpRequest req)
RA14 {
RA15    ...
RA16 }
RA17 }
```


NEW QUESTION: 182

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□□ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□ □□ □ □□□ □□ □□ □□□□ □□□ □□ □□ □□□□.

□ □□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □ □ □□□ □□□□ □□□□.

Azure Event Grid□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□□□□□ □□□□ □ □□□.

□□□ □□ □□ □□□ □□□□.

* □□ □□□ □□□ □□□ □□□ □□□□ □□□□ □□ □□ □□□□ □□□ □□□.
* □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□□.
* □□ □ □□□ Microsoft Entra ID□ □□□□ □□□□ □□□.
* □□□□ □□ □□□□□□ □□□□□ □□□.

Azure Event Gnd□ □□□□ □□□.

□□ □□: □□□ □□□ □□□□ □□□□□. □ □□□ □□ □□□ □□□ □□□□.

□□□□ □□□ □□□□□?

A. □

B. □□□

Answer: B (LEAVE A REPLY)

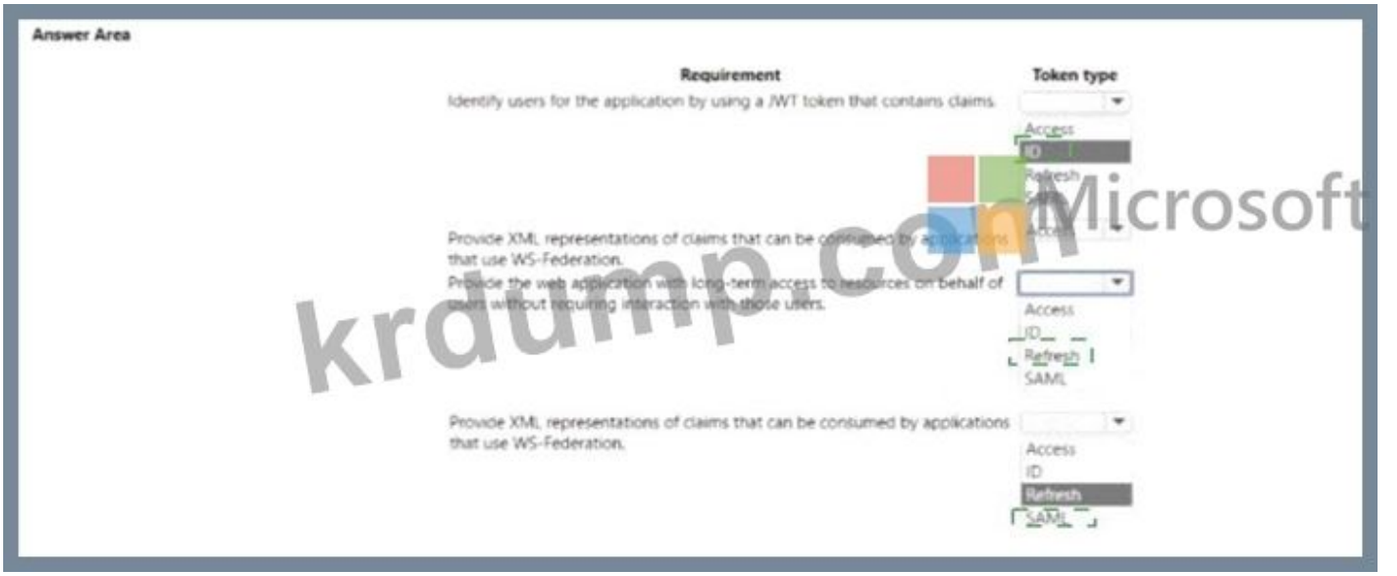
NEW QUESTION: 183

□□□ □ □□□ □□□ □□ Microsoft □□ □□□□ □□□□ □ □□□□□□□ □□□□ □ □□□. □ □□□□□□□ □□ REST API□ □□□□□.
□ □□□□□□□ □□ □□□ □□ □ □□ □□ □□□ □□□□ □□□□.
□□ □□□ □□□□ □□□□ □□□.
□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

Requirement	Token type
Identify users for the application by using a JWT token that contains claims.	Access / ID / Refresh / SAML
Provide XML representations of claims that can be consumed by applications that use WS-Federation.	Access / ID / Refresh / SAML
Provide the web application with long-term access to resources on behalf of users without requiring interaction with those users.	Access / ID / Refresh / SAML
Provide XML representations of claims that can be consumed by applications that use WS-Federation.	Access / ID / Refresh / SAML

Answer:



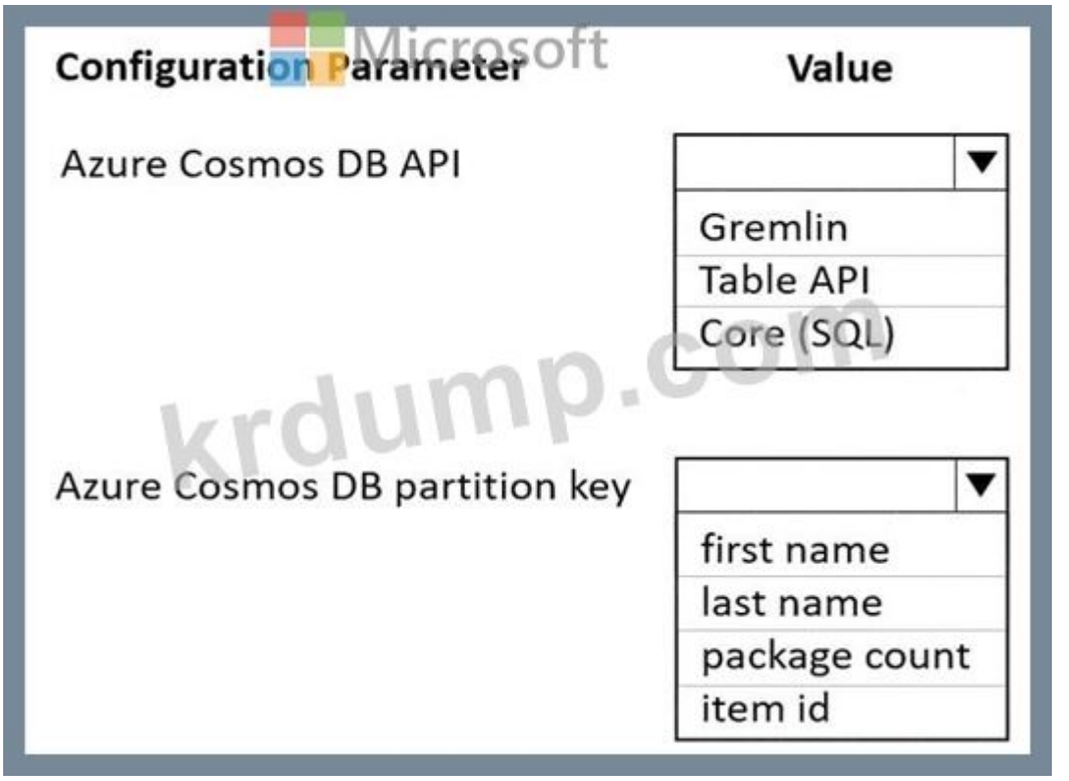
Explanation:

Answer Area

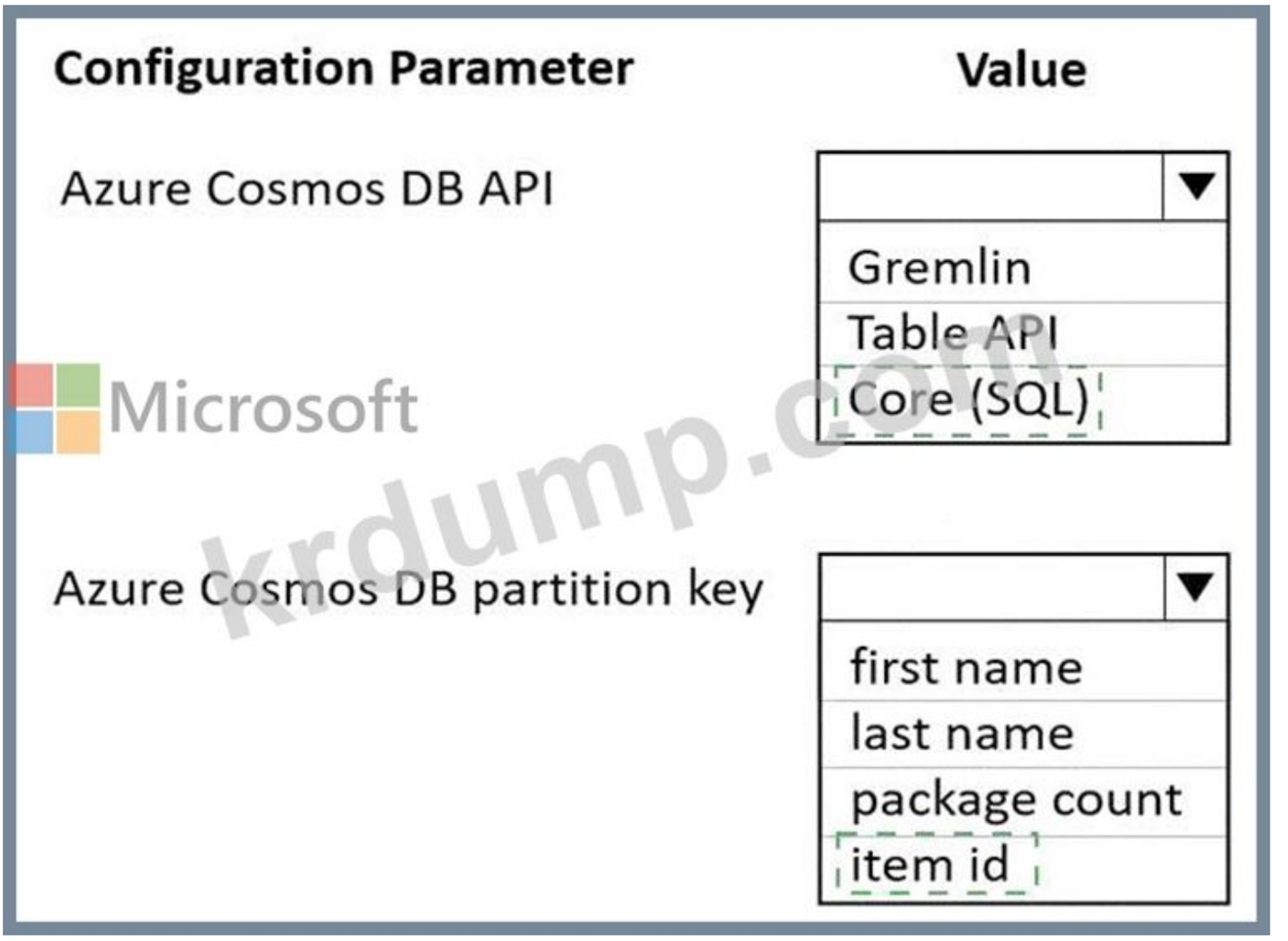
Requirement	Token type
Identify users for the application by using a JWT token that contains claims.	ID
Identify the permissions granted to APIs by using a JWT token that contains claims.	Access
Provide the web application with long-term access to resources on behalf of users without requiring interaction with those users.	Refresh
Provide XML representations of claims that can be consumed by applications that use WS-Federation.	SAML

NEW QUESTION: 184

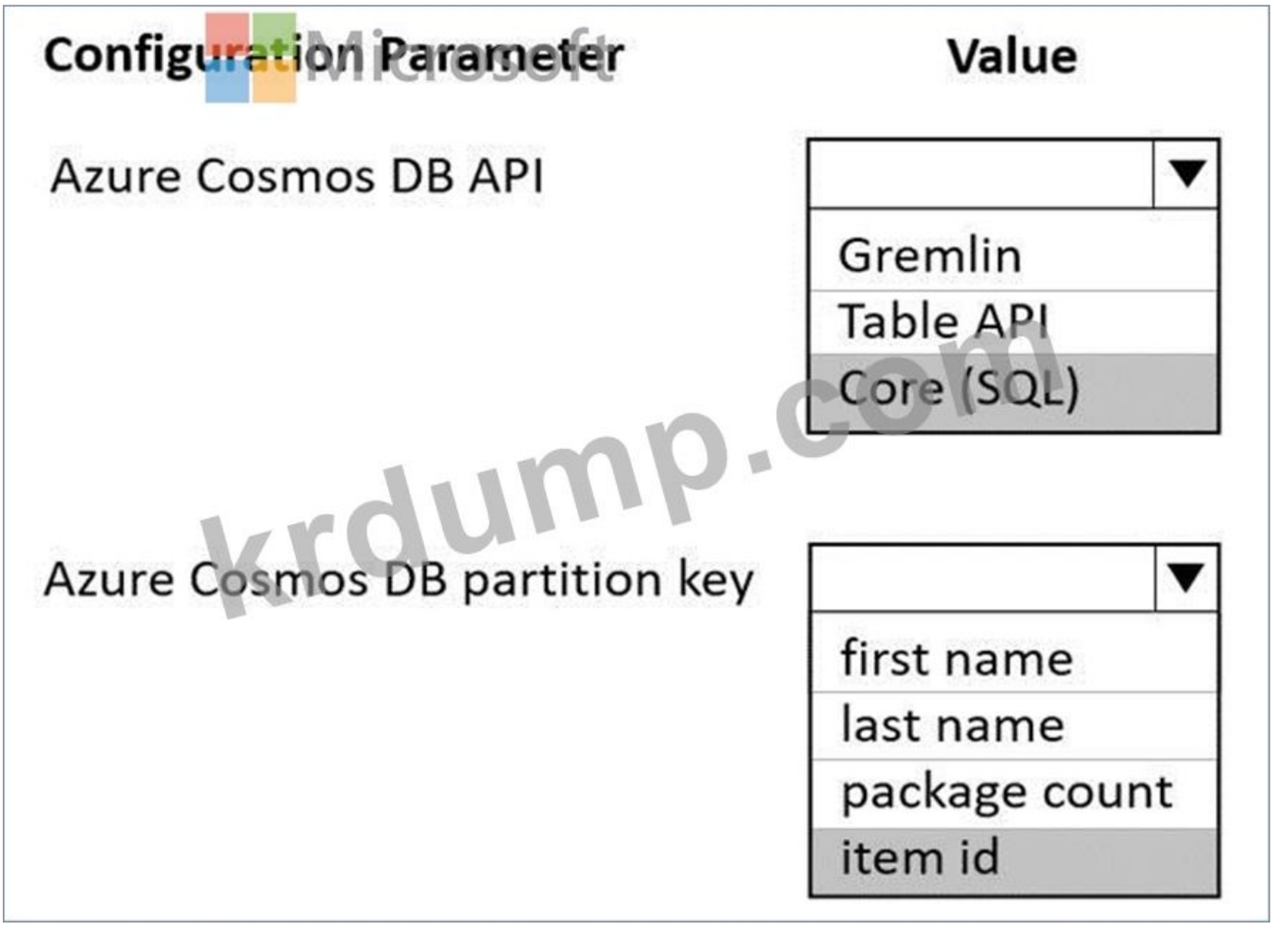
□□ □□□□□ □□ □□ □□ □□□□ □□, □, □□□ □, □□ ID □ □□ □□ □□□ □□ □□ □□□□□□□ □□□□ □□□□. □□ Azure Cosmos DB□ □□□□ □□□□□.
□□□□ □□□□□ Azure Cosmos DB□ □□□□ □□□.
□□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Core (SQL)

Core(SQL) API stores data in document format. It offers the best end-to-end experience as we have full control over the interface, service, and the SDK client libraries. SQL API supports analytics and offers performance isolation between operational and analytical workloads.

Box 2: item id

item id is a unique identifier and is suitable for the partition key.

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/choose-api

https://docs.microsoft.com/en-us/azure/cosmos-db/partitioning-overview

NEW QUESTION: 185

□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□. □ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.

□ □□ 2,000□ □□□□ POS(Point-of-Sale) □□ □□□□ □□□□ Azure □□□□ □□□□ □□□□. □□ □□□ 24□□□□ 2MB□ □□□□ □□□ □ □□□□. □ □□ □□□□ □□□ □ □□□□ 1~5□□ □□□ □□□□.

Azure Blob Storage□ □□□□ □□□□ □□□□ □□□. □□ □□□□ □□ □□□□ □□□

□ □□ □□□□□ □□□. □□ □□ □□ □□ □□□□□.
□□ □□□□ □□□□ □□ □□□□ □□□□ □□□.
□□□: Azure Event Grid□ □□□□□□□□. □□□ □□□□ □□□ □□ □□□□ □□□ □ □□□□□.
□□□□ □□□ □□□□□?

A. □

B. □□□

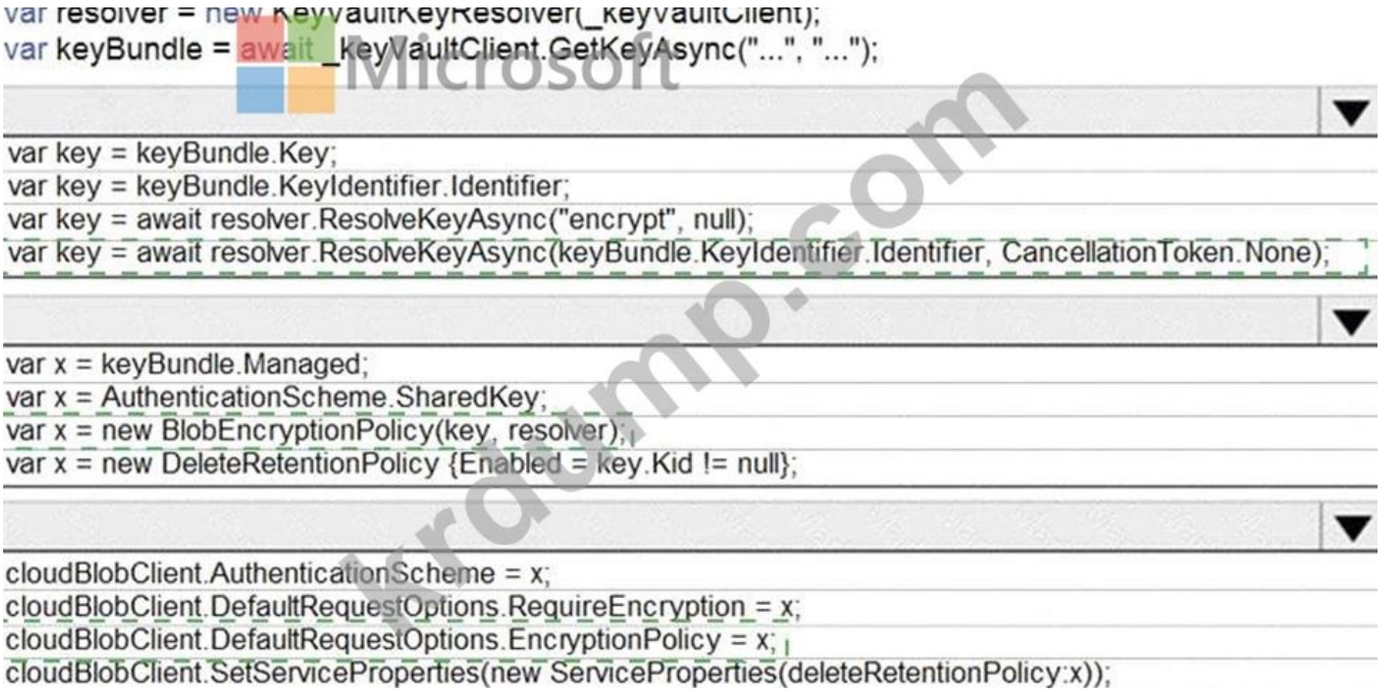
Answer: A (LEAVE A REPLY)

Reference:
https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services

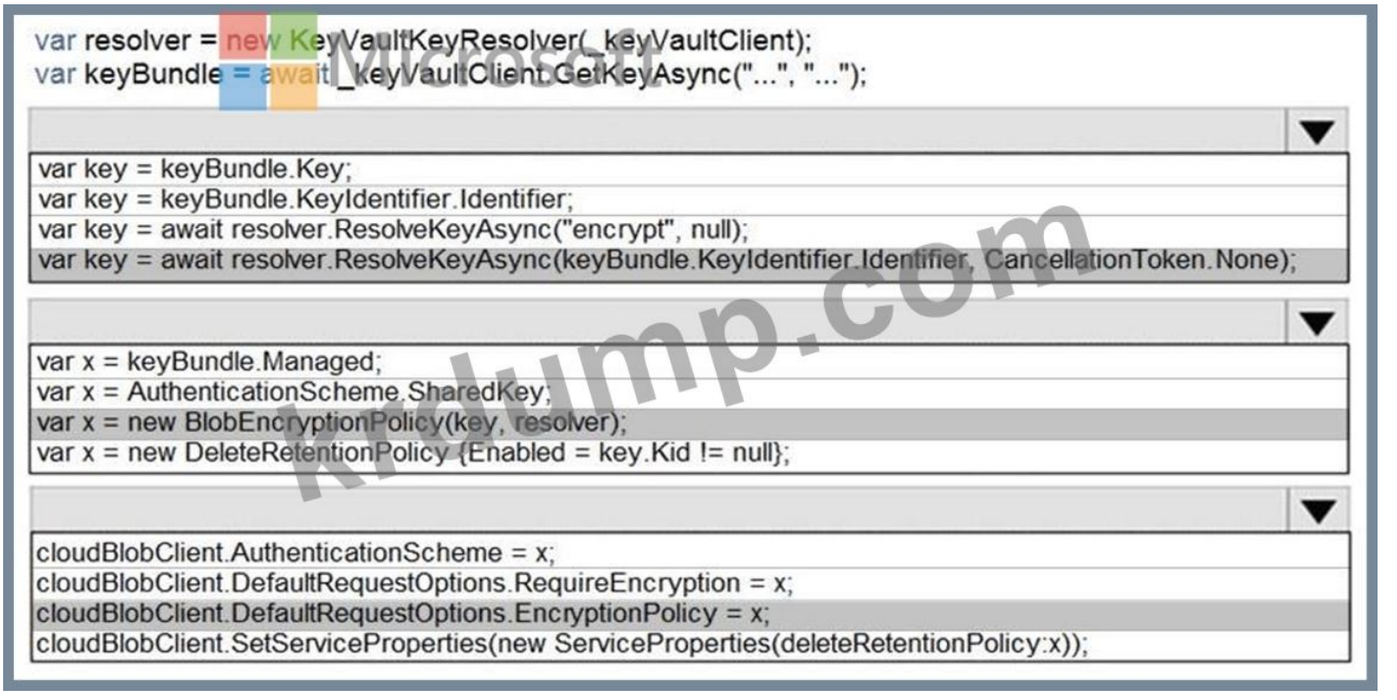
NEW QUESTION: 186
□□ □□□ □□□□□ □□□ Processing.cs□ PC26 □□ □□□ □□□□ □□□.
PC26 □□ □□□ □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
var resolver = new KeyVaultKeyResolver(_keyVaultClient);
var keyBundle = await _keyVaultClient.GetKeyAsync("...", "...");

[ ▼ ]
var key = keyBundle.Key;
var key = keyBundle.KeyIdentifier.Identifier;
var key = await resolver.ResolveKeyAsync("encrypt", null);
var key = await resolver.ResolveKeyAsync(keyBundle.KeyIdentifier.Identifier, CancellationToken.None);

[ ▼ ]
var x = keyBundle.Managed;
var x = AuthenticationScheme.SharedKey;
var x = new BlobEncryptionPolicy(key, resolver);
var x = new DeleteRetentionPolicy {Enabled = key.Kid != null};

[ ▼ ]
cloudBlobClient.AuthenticationScheme = x;
cloudBlobClient.DefaultRequestOptions.RequireEncryption = x;
cloudBlobClient.DefaultRequestOptions.EncryptionPolicy = x;
cloudBlobClient.SetServiceProperties(new ServiceProperties(deleteRetentionPolicy:x));
```


Answer:



Explanation:



Box 1: var key = await Resolver.ResolveKeyAsyn(keyBundle,KeyIdentifier.CancellationToken.None); Box 2: var x = new BlobEncryptionPolicy(key,resolver); Example:

// We begin with cloudKey1, and a resolver capable of resolving and caching Key Vault secrets.
BlobEncryptionPolicy encryptionPolicy = new BlobEncryptionPolicy(cloudKey1, cachingResolver);
client.DefaultRequestOptions.EncryptionPolicy = encryptionPolicy; Box 3: cloudblobClient.
DefaultRequestOptions.EncryptionPolicy = x; Reference:
https://github.com/Azure/azure-storage-net/blob/master/Samples/GettingStarted/EncryptionSamples/KeyRotation/Program.cs

NEW QUESTION: 187

Azure Blob Storage □□□□ □□ Azure Functions □□ □□□ □□□□□. □□ □□□ □□□ □ □□ □□ □□□ □□□□□. □□ □□ □□ □□□ □□□□ □□□.

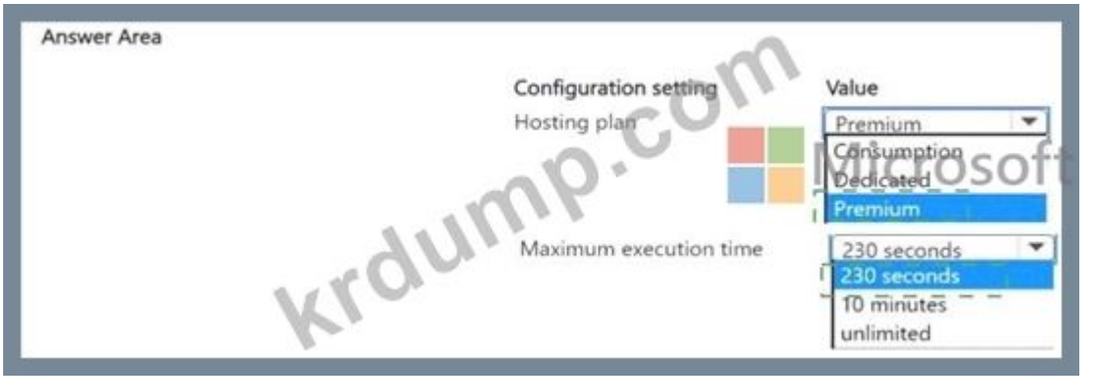
* □□□ □□ □□□□
* □□ □□ □□
* □□ □□□

□□ □□□ □ □□□ □□□ □□ □□□ □□□□ □□□.

□□ □□ □□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□ □□□□□.

□□□□: □□ □□□ 1□□□□.

Answer:



Explanation:

NEW QUESTION: 188

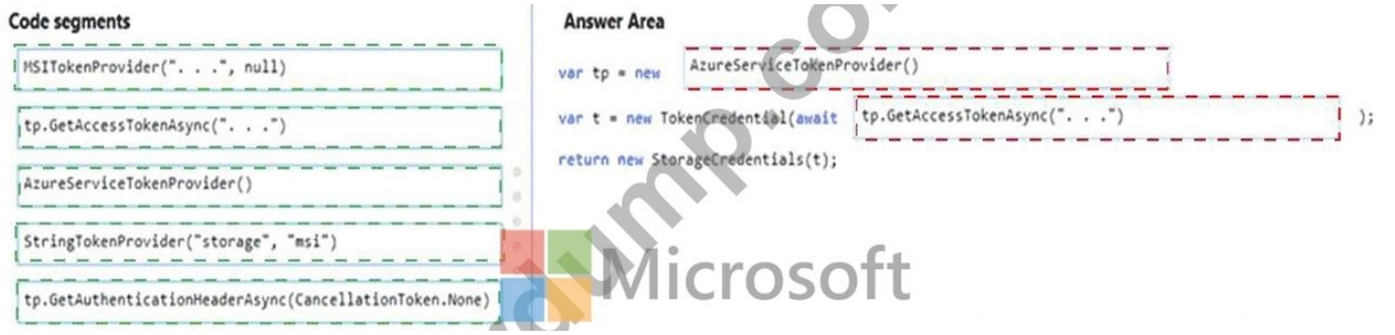
Processing □□□□□ GetCredentials □□□□ □□□□□ Processing.cs□ PC32 □□ □□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□□□ □□□ □□□ □□□□□ □□.

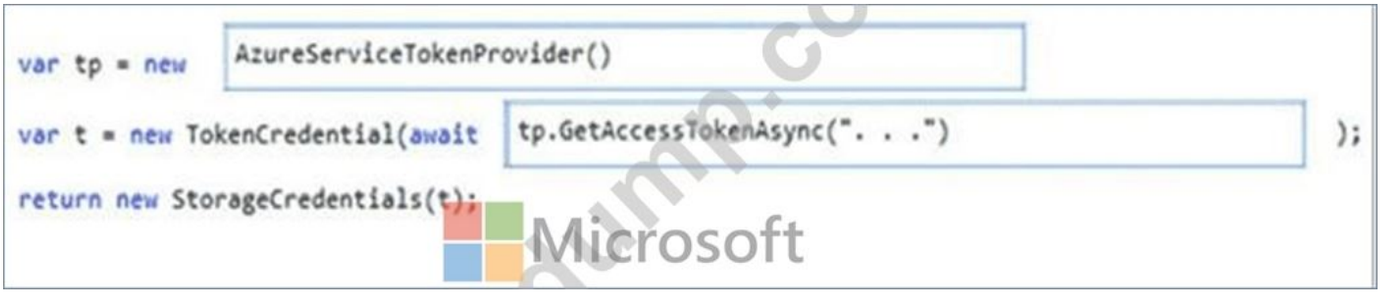
□ □□ □□□□□ □ □, □ □ □□ □□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:



Box 1: AzureServiceTokenProvider()

Box 2: tp.GetAccessTokenAsync("..")

Acquiring an access token is then quite easy. Example code:

private async Task<string> GetAccessTokenAsync()

{

var tokenProvider = new AzureServiceTokenProvider();

return await tokenProvider.GetAccessTokenAsync("https://storage.azure.com/");

}

Reference:

https://joonasw.net/view/azure-ad-authentication-with-azure-storage-and-managed-service-identity

NEW QUESTION: 189

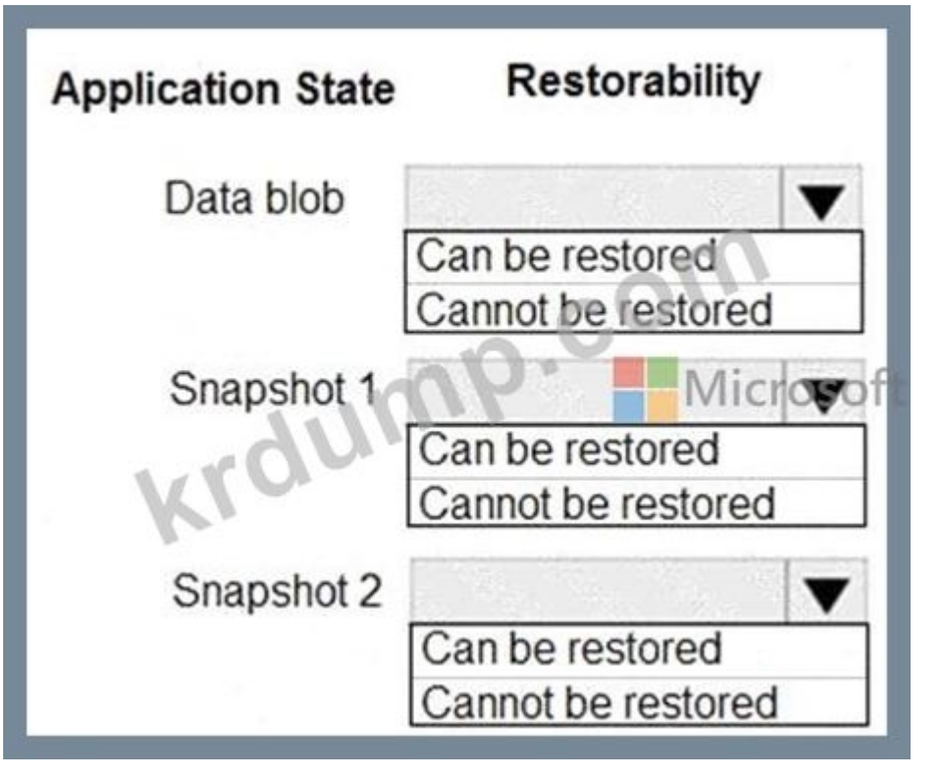
□□□□□ □□□ Azure Blob□ □□□□ □□□□□□ □□□□ □□□□ □□□□□□□ □□ □□ □□□□. □□□□□□□ □□□□□□ □□□ □□ □□□ □□□ □ □□□ Blob □□□ □ □□□□. Azure Storage □□□ □□ □□□ □□□□□ □□□□.

□□□□ □□ □□□ □□□□ □□□□□.

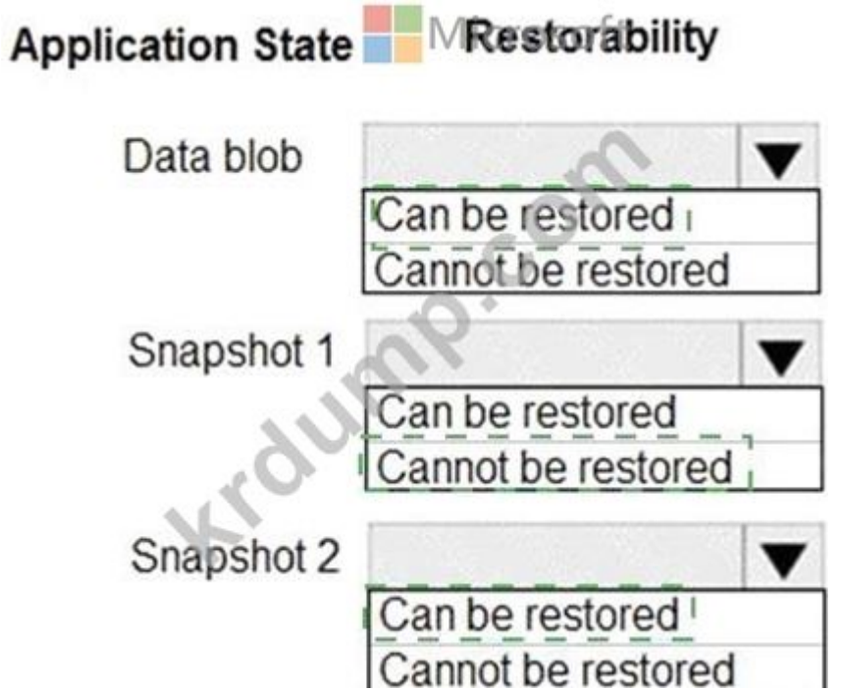
*Blob□ □□□□□□□□□.

*□□□ 1□ □□□□□.

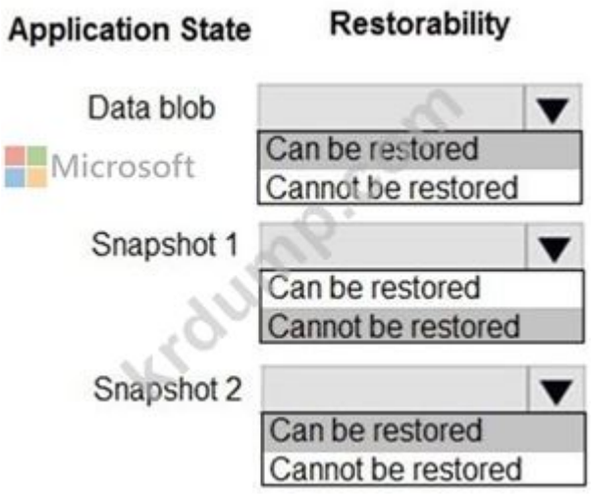
*□□□ 2□ □□□□□.
*□□□ 1□ □□□□□□□.
□□□ □□□ □□□ □□□□ □□□ Blob□ □□ □□□□ □□□□□.
□□□ □ □□ □□□□□□ □□□ □□□□ □□□.
□□□□□□ □□□□ □□ □□□□ □□□□□? □□□□□ □□ □□□□ □□□ □□□ □ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:



Box 1: Can be restored

When enabled, soft delete enables you to save and recover your data when blobs or blob snapshots are deleted. This protection extends to blob data that is erased as the result of an overwrite.

Box 2: Cannot be restored

It has been deleted.

Box 3: Can be restored

It has not been deleted.

References:

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-soft-delete

NEW QUESTION: 190

Java □□□□□□□ □□ □ □□ □□□ □□□□ □□□□.

□□□□ VMSS(Virtual Machine Scale Set), □□ □□□□ □□ □ □□□□ □□ □□□ □□□ □□.

□□□□ □□□ □□□□□ □□□□ □□□ □□ □□ □□ □ □□ □□□□□ □□□ □□□ VMSS□ □□□□□ □ □□□.

Azure Resource Manager □□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer Area

```
{
. . .
"resources": [
  {                                                    pyIn
    "apiVersion": "2016-01-01",
    "type": "Microsoft.Storage/storageAccounts",
    "name": "[concat(                                    (), 'storage', uniqueString(resourceGroup().id))]",
                        copy
                        copyIndex
                        priority
                        dependsOn
     "location": "[resourceGroup().location]",
     . . .
       "sku": {
       "name": "Standard_LRS"
     },
     "kind": "Storage",
     "properties": {},
     "                                  ": {
          copy
          copyIndex
          priority
          dependsOn
       "name": "storagesetup",
       "count": 3
     }
  },
  {
    "apiVersion": "2015-06-15",
    "type": "Microsoft.Compute/virtualMachines",
    "name": "[concat('VM', uniqueString(resourceGroup().id))]",
    "                                  ": [
          copy
          copyIndex
          priority
          dependsOn
       "[variables('loadBalancerName')]",
       "[variables('virtualNetworkName')]",
       "storagesetup",
       ],
     . . .
   }
],
"outputs": {}
}
```


Answer:

Answer Area

```
{
. . .
"resources": [
  {                                                    pyln
    "apiVersion": "2016-01-01",
    "type": "Microsoft.Storage/storageAccounts",
    "name": "[concat(                                    (), 'storage', uniqueString(resourceGroup().id))]",
                        copy
                        copyIndex
                        priority
                        dependsOn
     "location": "[resourceGroup().location]",
     . . .
       "sku": {
       "name": "Standard_LRS"
     },
     "kind": "Storage",
     "properties": {},
     "                                    ": {
        copy
        copyIndex
        priority
        dependsOn
       "name": "storagesetup",
       "count": 3
     }
  },
  {
     "apiVersion": "2015-06-15",
     "type": "Microsoft.Compute/virtualMachines",
     "name": "[concat('VM', uniqueString(resourceGroup().id))]",
     "                                    ": [
        copy
        copyIndex
        priority
        dependsOn
       "[variables('loadBalancerName')]",
       "[variables('virtualNetworkName')]",
       "storagesetup",
       ],
     . . .
   }
],
"outputs": {}
```

Explanation:

```
{
. . .
"resources": [
  {                                                    pyln
    "apiVersion": "2016-01-01",
    "type": "Microsoft.Storage/storageAccounts",
    "name": "[concat(                                    (), 'storage', uniqueString(resourceGroup().id))]",
                        copy
                        copyIndex
                        priority
                        dependsOn
     "location": "[resourceGroup().location]",
     . . .
       "sku": {
       "name": "Standard_LRS"
```

Box 1: copyIndex

Notice that the name of each resource includes the copyIndex() function, which returns the current iteration in the loop. copyIndex() is zero-based.

Box 2: copy

By adding the copy element to the resources section of your template, you can dynamically set the number of resources to deploy.

Box 3: dependsOn

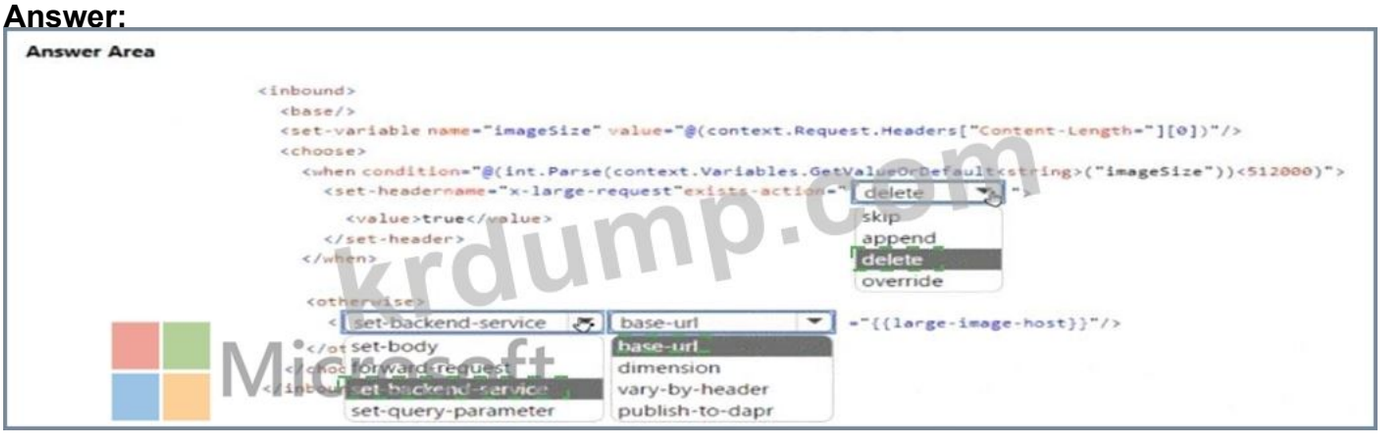
Example:

"type": "Microsoft.Compute/virtualMachineScaleSets",
"apiVersion": "2020-06-01",
"name": "[variables('namingInfix')]",
"location": "[parameters('location')]",
"sku": {
"name": "[parameters('vmSku')]",
"tier": "Standard",
"capacity": "[parameters('instanceCount')]"
},
"dependsOn": [
"[resourceId('Microsoft.Network/loadBalancers', variables('loadBalancerName'))]",
"[resourceId('Microsoft.Network/virtualNetworks', variables('virtualNetworkName'))]"

],
Reference:
https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/copy-resources
https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/quick-create-template-windows

NEW QUESTION: 191
Azure API Management□ □□□□ □□□□ □□□ □□□ □□□□ □□□□□. □□□□ □□ □ □ □□□□ □□ □□□ □□□□□.
500KB□ □□ □□□□ □□□ □ □□□ □□ □□ □□□□ □□□□ □□ □□□□□ □□□ □□. □□ □□ □□□□ x-lsrSe-request□□ □□□ □□□□□. 500KB□ □□ □□□□ □ □ □ □□□□ □□□□□ □□□□□ □ □□, □□ □ □□ □□□ □□□□□.
□□□□ □□□□ □□□□□ API □□ □□□ □□□□ □□□.
API □□ □□□□ □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □ □□□□.
□□: □□ □□□ 1□□□□.

Answer:



Explanation:

NEW QUESTION: 192

□□□□ Azure Service Bus□ □□□□ □□-□□(Pub/Sub) □□□ □□ □□□ □□□□ □□□ □. □ □□ □□ □□□□□□□ □□□□ □□□□.

Azure Portal□□ □□□□ □ □□□ □□ □□□□ □□□□ □□ □ □ □□□□. □□□ □□ □□□ □□□□ □□ □□□□□ □□□ □□□ □□□□□□ □□ □□□□□□□ □□□ □□ □□ □□□□ □□□□.

□□ □□□□□□ □□ □□□□ □□□□□ □□□□ □□□.

□□ □□ □□□□□ □□□□ □□□?

A. await subscriptionClient.AddRuleAsync(□ RuleDescription (RuleDescription.DefaultRuleName, new TrueFilter()));

B. subscriptionClient = new SubscriptionClient(ServiceBusConnectionString, TopicName, SubscriptionName); D18912E1457D5D1DDCBD40AB3BF70D5D

C. await subscriptionClient.CloseAsync();

D. subscriptionClient.RegisterMessageHandler(ProcessMessagesAsync,messageHandlerOptions);

Answer: D (LEAVE A REPLY)

Using topic client, call RegisterMessageHandler which is used to receive messages continuously from the entity. It registers a message handler and begins a new thread to receive messages. This handler is waited on every time a new message is received by the receiver.

subscriptionClient.RegisterMessageHandler(ReceiveMessagesAsync, messageHandlerOptions);

Reference:

https://www.c-sharpcorner.com/article/azure-service-bus-topic-and-subscription-pub-sub/

NEW QUESTION: 193

□□□□ □□□ □□ □□ □□□ □□□□ □□□.

□□ □□□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□□□ □□□ □□□ □□□ □□□□. □ □□ □□□□□ □ □, □ □ □□ □□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Code segments

- /health
- /status
- RequestTelemetry
- PageViewTelemetry
- ITelementryProcessor
- ITelementryInitializer

Answer Area

```
public class Filter : [ code segment ]
{
    private readonly [ code segment ] _next;
    public (Filter [ code segment ] next)
    {
    _next = next;
    }
    public void Process(ITelemetry item)
    {
      var x = item as [ code segment ];
      if (x?.Url.AbsolutePath == " [ code segment ] " )
      {
        return;
      }
      _next.Process(item);
    }
}
```


Answer:

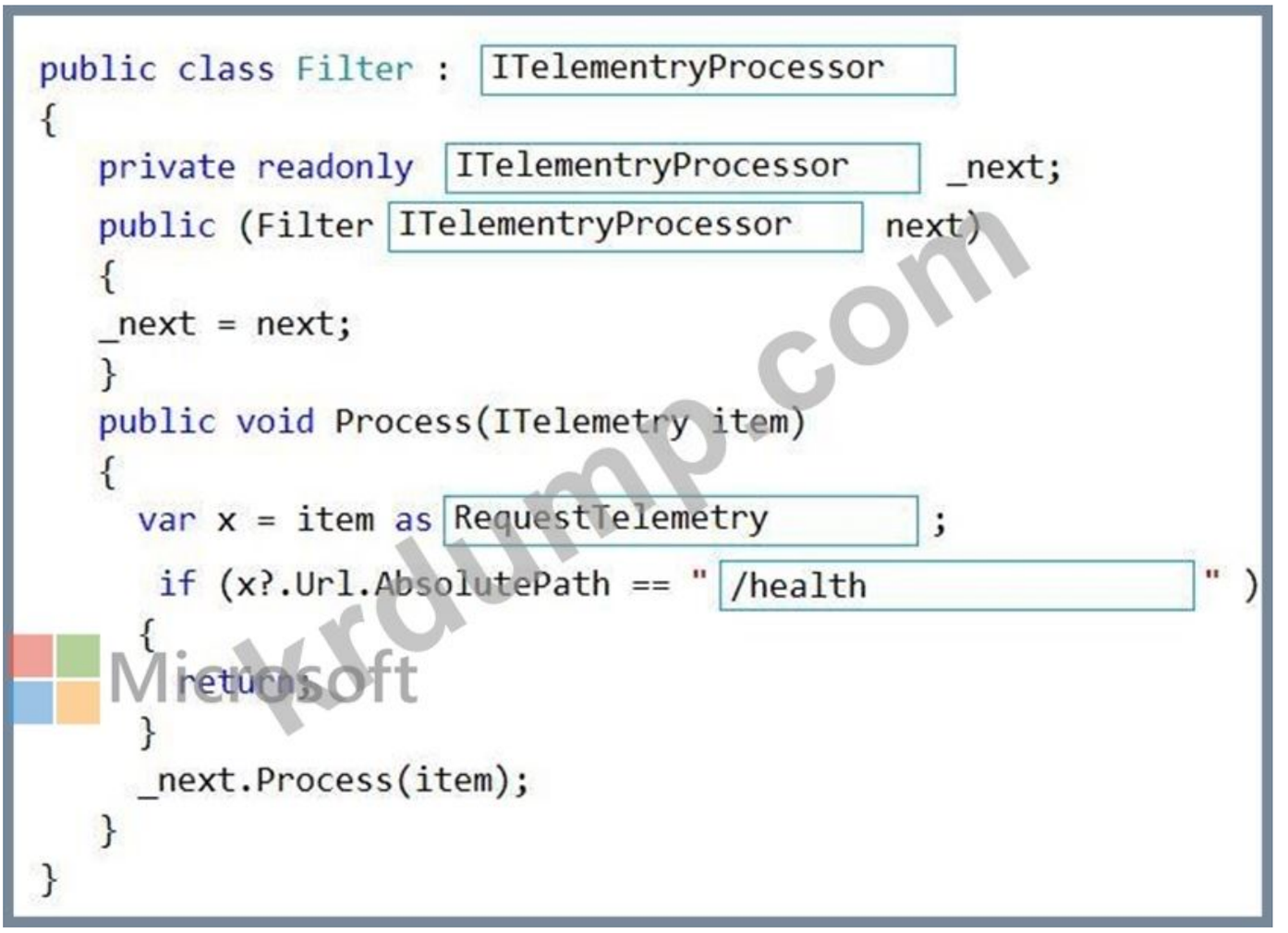
Code segments

- /health
- /status
- RequestTelemetry
- PageViewTelemetry
- ITelementryProcessor
- ITelementryInitializer

Answer Area

```
public class Filter : ITelementryProcessor
{
    private readonly ITelementryProcessor _next;
    public (Filter ITelementryProcessor next)
    {
    _next = next;
    }
    public void Process(ITelemetry item)
    {
      var x = item as RequestTelemetry;
      if (x?.Url.AbsolutePath == " /health " )
      {
        return;
      }
      _next.Process(item);
    }
}
```


Explanation:



Scenario: Exclude non-user actions from Application Insights telemetry.

Box 1: ITelemetryProcessor

To create a filter, implement ITelemetryProcessor. This technique gives you more direct control over what is included or excluded from the telemetry stream.

Box 2: ITelemetryProcessor

Box 3: ITelemetryProcessor

Box 4: RequestTelemetry

Box 5: /health

To filter out an item, just terminate the chain.

Reference:

https://docs.microsoft.com/en-us/azure/azure-monitor/app/api-filtering-sampling

Topic 5, Litware Inc

Case study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is

independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.
To start the case study
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. When you are ready to answer a question, click the Question button to return to the question.
Background
You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.
Overall architecture
Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.
Receipt processing
Employees may upload receipts in two ways:
Uploading using an Azure Files mounted folder
Uploading using the web application
Data Storage
Receipt and employee information is stored in an Azure SQL database.
Documentation
Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.
Solution details
Users table

Column	Description
UserId	unique identifier for and employee
ExpenseAccount	employees expense account number in the format 1234-123-1234
AllowedAmount	limit of allowed expenses before approval is needed
SupervisorId	unique identifier for employee's supervisor
SecurityPin	value used to validate user identity

Web Application
You enable MSI for the Web App and configure the Web App to use the security principal name WebAppIdentity.
Processing
Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL

database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application.

The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Requirements

Receipt processing

Concurrent processing of a receipt must be prevented.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

User's SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.

All certificates and secrets used to secure data must be stored in Azure Key Vault.

You must adhere to the principle of least privilege and provide privileges which are essential to perform the intended function.

All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI).

Receipt data must always be encrypted at rest.

All data must be protected in transit.

User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment, with the remaining parts obscured.

In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues

Upload format issue

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

Capacity issue

During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

Log capacity issue

Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Application code

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03   public static class Function
PC04   {
PC05     [FunctionName("IssueWork")]
PC06     public static async Task Run([TimerTrigger("0 */5 * * * *")] TimerInfo timer, ILogger
log)
PC07     {
PC08       var container = await GetCloudBlobContainer();
PC09       foreach (var fileItem in await ListFiles())
PC10       {
PC11         var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12         var ms = new MemoryStream();
PC13         await file.DownloadToStreamAsync(ms);
PC14         var blob = container.GetBlockBlobReference(fileItem.Uri.ToString());
PC15         await blob.UploadFromStreamAsync(ms);
PC16
PC17       }
PC18     }
PC19     private static CloudBlockBlob GetDRBlob(CloudBlockBlob sourceBlob)
PC20     {
PC21       . . .
PC22     }
PC23     private static async Task<CloudBlobContainer> GetCloudBlobContainer()
PC24     {
PC25       var cloudBlobClient = new CloudBlobClient(new Uri(". . ."), await GetCredentials());
PC26
PC27       await cloudBlobClient.GetRootContainerReference().CreateIfNotExistsAsync();
PC28       return cloudBlobClient.GetRootContainerReference();
PC29     }
PC30     private static async Task<StorageCredentials> GetCredentials()
PC31     {
PC32       . . .
PC33     }
PC34     private static async Task<List<IListFileItem>> ListFiles()
PC35     {
PC36       . . .
PC37     }
PC37     private KeyVaultClient _keyVaultClient = new KeyVaultClient(". . .");
PC38   }

PC39 }
```

Database.cs

```
DB01 public class Database
DB02 {
DB03   private string ConnectionString =
DB04
DB05   public async Task<object> LoadUserDetails(string userId)
DB06   {
DB07
DB08     return await policy.ExecuteAsync(async () =>
DB09     {
DB10       using (var connection = new SqlConnection(ConnectionString))
DB11       {
DB12         await connection.OpenAsync();
DB13         using (var command = new SqlCommand("…", connection))
DB14         using (var reader = command.ExecuteReader())
DB15         {
DB16           …
DB17         }
DB18       }
DB19     });
DB20   }

DB21 }
```

ReceiptUploader.cs

```
RU01 public class ReceiptUploader
RU02 {
RU03   public async Task UploadFile(string file, byte[] binary)
RU04   {
RU05     var httpClient = new HttpClient();
RU06     var response = await httpClient.PutAsync("…", new ByteArrayContent(binary));
RU07     while (ShouldRetry(response))
RU08     {
RU09       response = await httpClient.PutAsync("…", new ByteArrayContent(binary));
RU10     }
RU11   }
RU12   private bool ShouldRetry(HttpResponseMessage response)
RU13   {
RU14
RU15   }
RU16 }
```

ConfigureSSE.ps1

```
CS01 $storageAccount = Get-AzureRmStorageAccount -ResourceGroupName "..." -AccountName "..."
CS02 $keyVault = Get-AzureRmKeyVault -VaultName "..."
CS03 $key = Get-AzureKeyVaultKey -VaultName $keyVault.VaultName -Name "..."
CS04  Set-AzureRmKeyVaultAccessPolicy `
CS05   -VaultName $keyVault.VaultName `
CS06   -ObjectId $storageAccount.Identity.PrincipalId `
CS07
CS08
CS09  Set-AzureRmStorageAccount `
CS10   -ResourceGroupName $storageAccount.ResourceGroupName `
CS11   -AccountName $storageAccount.StorageAccountName `
CS12   -EnableEncryptionService File `
CS13   -KeyvaultEncryption `
CS14   -KeyName $key.Name
CS15   -KeyVersion $key.Version `
CS16   -KeyVaultUri $keyVault.VaultUri
```


NEW QUESTION: 194

Azure Cache for Radis□ □□ □□ □□□□□ redis1□ □□ □□□□ □□□□ □□□□.
□□ □□□ □□□ □ □□ □□ □□ □□□□ Maxmemory □□□ □□□□ □□□.
Maxmemory □□□ □□□ □□□□ □□□?

A. maxmemory-reserved □□ □□□□.

B. maxmemory-reserved □□ □□□□.

C. Maxmemory □□□ volatile-lru□ □□□□□.

D. Maxmemory □□□ noeviction□□ □□□□□.

Answer: B (LEAVE A REPLY)

NEW QUESTION: 195

Azure Container Apps□□ □□□□ □□ □□ □□□□□□□□ □□□□ □□□□. □□□□□ □□□ □□ □□ HTTP □□ □□□□ □□□□□□□□. □□□ □□□□□□□□ □□□□□ □ □□□□ □□□ □□□ □□□□ □ □□□ □□ □□□. □□□ □□ □□ □□□ □□□□.

* □□□□□ □□□□□□□□ □□ □□ URL□ □□□□□ □□□□ □□□ □□□□□ □□□ □□.

* □□ □□□□□□□ □□□ □□□□ □□□□□□□□ □□□□□□□. Azure Container Apps□ □□□□ □□□.

□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

Answer:

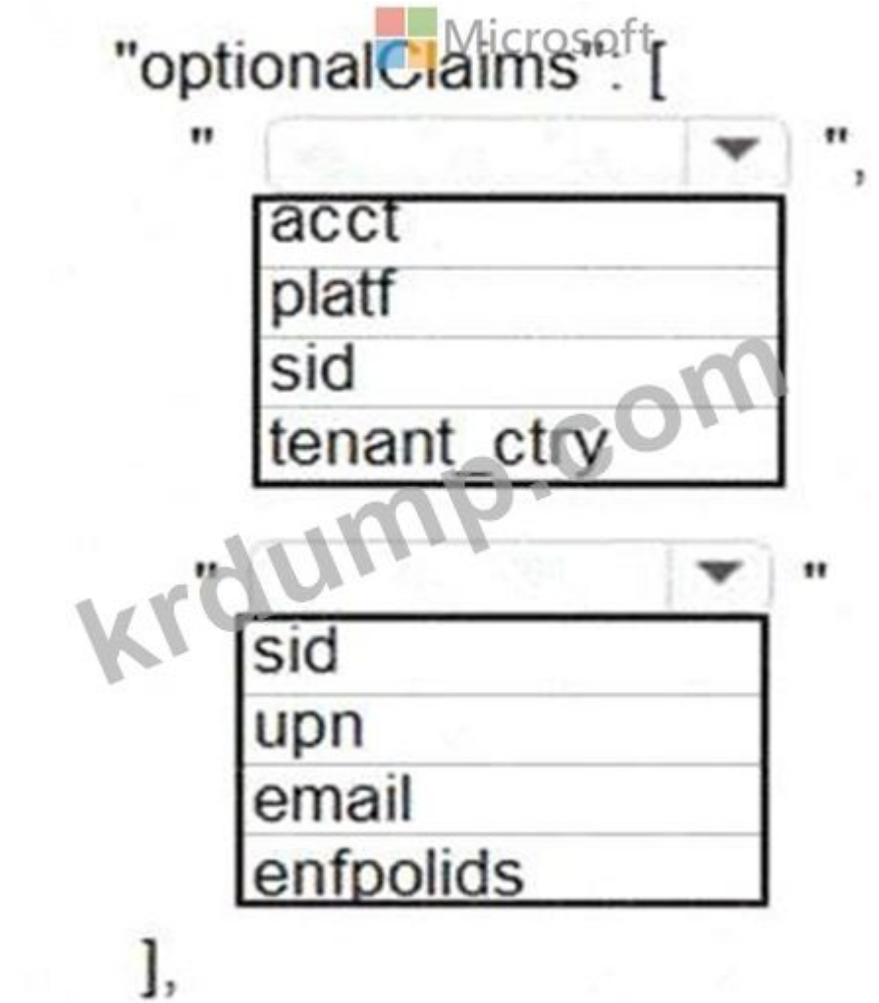
Explanation:

NEW QUESTION: 196

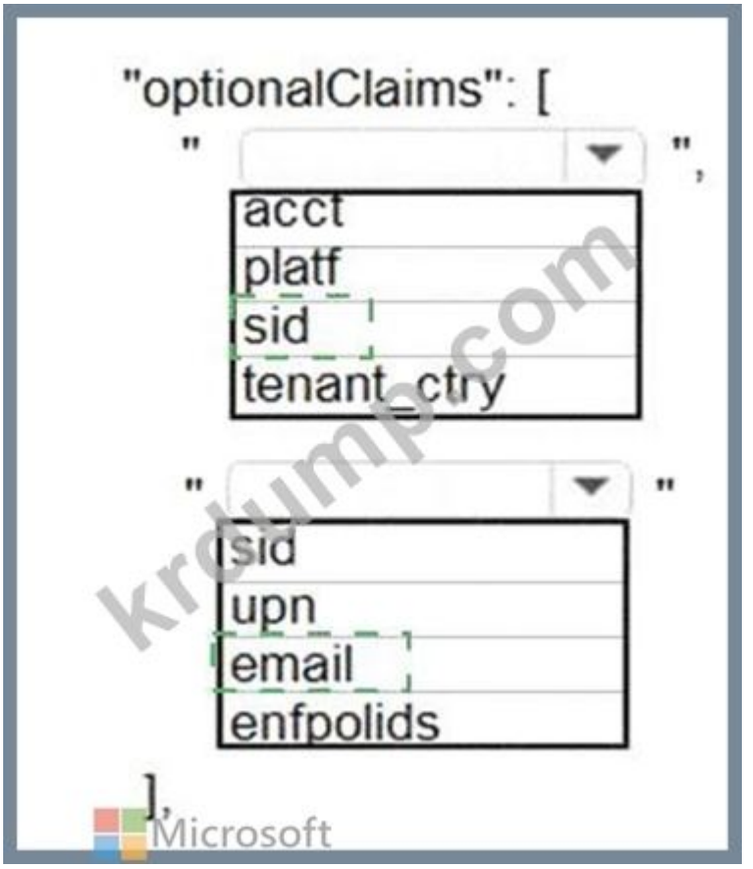
□□□□ □□□□ □□□□ □□ □□ □□□ □□□ □ □□□ □□□□□□ □□□□□□ AM10 □□ □□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

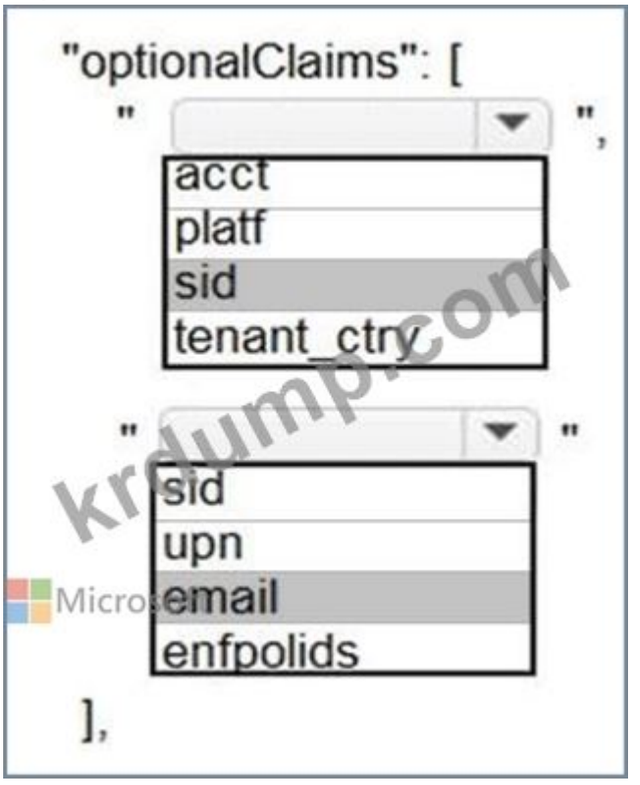
□□: □ □□□ □□□ 1□□ □□□ □□□□.


"optionalClaims": [
"
",
acct
platf
sid
tenant_ctry
"
"
sid
upn
email
enfpolids
],

Answer:


"optionalClaims": [
"
",
acct
platf
sid
tenant_ctry
"
"
sid
upn
email
enfpolids
],

Explanation:



Box 1: sid

Sid: Session ID, used for per-session user sign-out. Personal and Azure AD accounts.

Scenario: Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role.

Box 2: email

Scenario: All completed reviews must include the reviewer's email address for auditing purposes.

NEW QUESTION: 197

Azure FrontDoor□ □□□□ ASP.NET Core □ □□□□ □□□□ □□□□. □ □□□□□ □□ □□ □□ □□□ □□ □□□ □□□ □□□□ □ □□□□□. □□□ □□□ □□□□ CSV(□□ □ □□□ □) □□□ □□□□□□□. □□□□ 10□□□□ □□ □□□□□.

□□ □□ □□ □□ FrontDoor □□□□ □□ □□□ □□□□ □□□.

Front Door □□□□ □□ □□□ □□□□ □□□.

□□ □□□ □□ □□□ □□□□ □□□?

A. □□ □□

B. □□□□□

C. □□ □□□

Answer: (SHOW ANSWER)

These formats are supported in the lists of paths to purge:

Single path purge: Purge individual assets by specifying the full path of the asset (without the protocol and domain), with the file extension, for example, /pictures/strasbourg.png; Wildcard purge: Asterisk (*) may be used as a wildcard. Purge all folders, subfolders, and files under an endpoint with /* in the path or purge all subfolders and files under a specific folder by specifying the folder followed by /*, for example, /pictures/*.

Root domain purge: Purge the root of the endpoint with "/" in the path.

Reference:

https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching

NEW QUESTION: 198

□□□ □□□□□□□ □□□□ □□□□. Azure App Services□□ □□□□□□□ □□□□ □ □□□□.

□□□ □□□ □□ □□ □□□ □□□□.

* 5□□□ □ □□□□ □□□□□ □□□□□.

* □□□□□ □□□ □□ □□□ □□ □□□□□ □□□□□. □□□ □ JavaScript □□□ □□ □□ □□□ □□□ □□□□□ □□□.

* □□□□□ □□□ □□ □□ □□□ □□□□□.

* □□□□□ □□□□ □□□ □□□□□ □□□□ □ □ □ □□□□□□ □□□.

□□□□ □□□□ □ □□□□□ □□□□ □□□.

□□□ □□ □□□?

D18912E1457D5D1DDCBD40AB3BF70D5D

A. Selenium □ □□□□ □□□ □□□ □□□□ □□□□□□□□ □□□□□ □□□□□.

B. URL □ □□□□ □□□□ □□□□□ □□□□□.

C. □ □□□□ □□□□ Azure □□□ □□□□.

D. □□□ □ □□□□ □□□ □□□□□ □□□□□.

E. □ □□□□ □□□□ □□ □□□ □□ □□ □□□ □□□□ □□□□□□.

Answer: D (LEAVE A REPLY)

You can monitor a recorded sequence of URLs and interactions with a website via multi-step web tests.

Reference:

https://docs.microsoft.com/en-us/azure/azure-monitor/app/availability-multistep

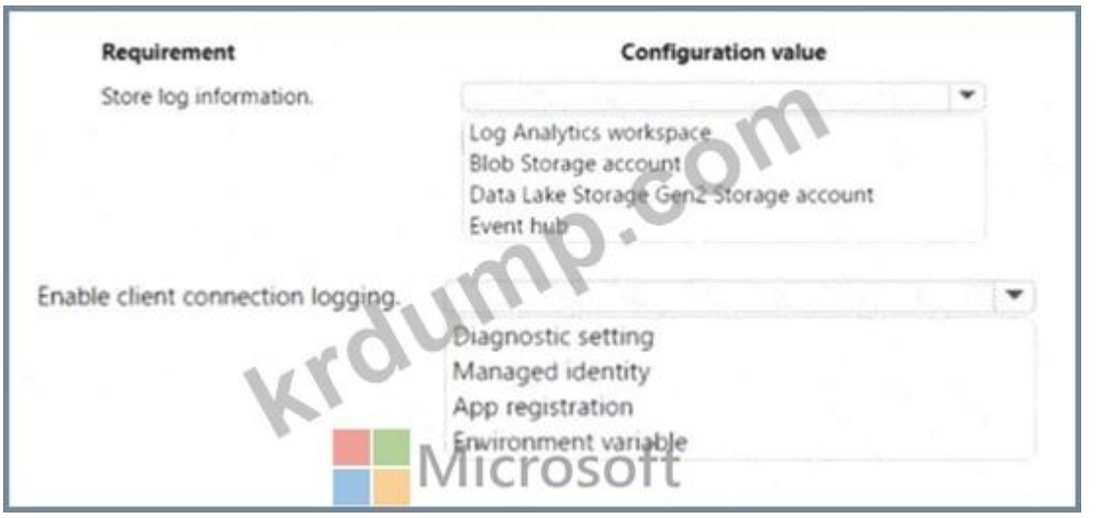
NEW QUESTION: 199

Azure □□□ □□□□□ □□□ □□ □□□□ □□□□ □□□□ □□□□. □□□□ □□□□ Azure Cosmos DB □ Azure Blob □□□□ □□□ □□□□ □□□□□. □□□□ Aue Key Vault

□ □□□ □□ □□ □□ □□□□ □□□□□.
□□ Key Vault □□ □□ □ □□□ □□□□□ □□ □ □□□ □□□□ □□□. □□ 3□□□□ □□□□ □□□. □□ □□ □□ □□□ □ □□ □□ □□□ □□□.
□ □□□ □□□□ □ □□ □□□ □□□□□ □□□.
□□ □ □□ □□□ □□□□ □□□? □ □□□ □□□□ □□□ □□□□□.
□□: □□□ □□□ □□□ □□□ □□□□.
A. □ Azure Event Grid □□□□□ □□□ □□□□□.
B. □ □□ □ □ □□ □□ □□ □□
C. Azure Key Vault □□□ □□ □ □□
D. Azure Key Vault □□
Answer: A,C (LEAVE A REPLY)
https://learn.microsoft.com/en-us/azure/key-vault/keys/how-to-configure-key-rotation

NEW QUESTION: 200
□□□ □□□ □□□ □□ Azure Web Apps□ □□□□□. □□□ □□□ □□ □□□□ □ □□ □ □□□ □□□ □ 15□□□ □□ □□□□ □□□□ □□□.
□□□□ □□□□ □□□ □ □□□□ □□ □□□□ □□□□ □□□□□ Azure Web Apps□ □□□□ □□□.
□□ □□ □□□ □□□□ □□□.
□ □□□ □□□□ □□ □□□ □ □□ □□ □□ □□□ □□□□□? □ □□□ □□□ □□□ □ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.
A. □□ □□□□
B. □□ □□ □□□
C. □□ □□ □□
D. CPU □□□ □□ □□□□□□
Answer: B,C (LEAVE A REPLY)

NEW QUESTION: 201
Azure Cache for Redis□ □□□ □□□ □□□□ Azure App Service □□□ □□□□ □□□□ □. □□ □□ 2 □□□ □□□□ □□□□□□□.
□□ □□ Azure Cache for Redis□□ □□□ □□□ □□□.
□□□ IP □□□ Redis □□□□□ □□ □□□□.
□□□ □□□□ Redis □□□ □□□□□□□.
Azure Cache for Redis □□□□□ □□□□ □□(□□)□□□.
□□ □□□ □□ □□ □□□ □□□ □□ □ □□□□□□□□ □□ □ □□□□ □□□. □□ □ □□□□ □□ □□□ □□ □□□ □□□ □□□□ □□□.
□□ □□ □□ □□□□ □□□?



Answer:

NEW QUESTION: 202

ContentUploadService □□□ □□□□□ http □□ □□ □□□ □□□□ □□□.

□□ □□□ □□ □□□□ □□□?

A. az □□ □□

B. az ams □□□ □□

C. az □□□ □□ □□

D. az □□□□ □□

Answer: C (LEAVE A REPLY)

Scenario: Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

"502 bad gateway" and "503 service unavailable" are common errors in your app hosted in Azure App Service.

Microsoft Azure publicizes each time there is a service interruption or performance degradation.

The az monitor activity-log command manages activity logs.

Note: Troubleshooting can be divided into three distinct tasks, in sequential order:

Observe and monitor application behavior

Collect data

Mitigate the issue

Reference:
https://docs.microsoft.com/en-us/cli/azure/monitor/activity-log
Topic 1, Contoso, Ltd
Case study
This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.
To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.
At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.
To start the case study
To display the first question in this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. When you are ready to answer a question, click the Question button to return to the question.
Background
Overview
You are a developer for Contoso, Ltd. The company has a social networking website that is developed as a Single Page Application (SPA). The main web application for the social networking website loads user uploaded content from blob storage.
You are developing a solution to monitor uploaded data for inappropriate content. The following process occurs when users upload content by using the SPA:
* Messages are sent to ContentUploadService.
* Content is processed by ContentAnalysisService.
* After processing is complete, the content is posted to the social network or a rejection message is posted in its place.
The ContentAnalysisService is deployed with Azure Container Instances from a private Azure Container Registry named contosoimages.
The solution will use eight CPU cores.
Azure Active Directory
Contoso, Ltd. uses Azure Active Directory (Azure AD) for both internal and guest accounts.
Requirements
ContentAnalysisService
The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a

specific threshold must be reviewed by an employee of Contoso, Ltd.
You must create an Azure Function named CheckUserContent to perform the content checks.
Costs
You must minimize costs for all Azure services.
Manual review
To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.
High availability
All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.
Monitoring
An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU cores.
Security
You have the following security requirements:
Any web service accessible over the Internet must be protected from cross site scripting attacks.
All websites and services must use SSL from a valid root certificate authority.
Azure Storage access keys must only be stored in memory and must be available only to the service.
All Internal services must only be accessible from internal Virtual Networks (VNets).
All parts of the system must support inbound and outbound traffic restrictions.
All service calls must be authenticated by using Azure AD.
User agreements
When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso, Ltd. to review content, store cookies on user devices, and track user's IP addresses.
Information regarding agreements is used by multiple divisions within Contoso, Ltd.
User responses must not be lost and must be available to all parties regardless of individual service uptime.
The volume of agreements is expected to be in the millions per hour.
Validation testing
When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.
Issues
Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.
Code

ContentUploadService

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location: westus
CS04 name: contentUploadService
CS05 properties:
CS06     containers:
CS07     - name: service
CS08         properties:
CS09         image: contoso/contentUploadService:latest
CS10         ports:
CS11         - port: 80
CS12           protocol: TCP
CS13         resources:
CS14           requests:
CS15             cpu: 1.0
CS16             memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19     ip: 10.23.121.112
CS20     ports:
CS21       - port: 80
CS22         protocol: TCP
CS23
CS24
CS25 networkProfile:
CS26
id: /subscriptions/98...19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subne
```

```
AM01 {
AM02      "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03      "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05      "createdDateTime" : "2019-12-24T06:01:44Z",
AM06      "logoUrl" : null,
AM07      "logoutUrl" : null,
AM08      "name" : "ContentAnalysisService",
AM09
AM10
AM11      "orgRestrictions" : [],
AM12      "parentalControlSettings" : {
AM13        "countriesBlockedForMinors" : [],
AM14        "legalAgeGroupRule" : "Allow"
AM15      },
AM16      "passwordCredentials" : []
AM17 }
```

NEW QUESTION: 203

Contoso, Ltd.□□ □□□ □□□□.

□□ XML □□□□ □□□□ API □□ □□□ □□□□□.

```
<set-variable name= "bodySize" value="@(context.Request.Headers["Content-Length"] [0])"/>
<choose>
 <when condition= "@(int.Parse(context.Variables.GetValueOrDefault<string> ("bodySize"))<512000)">
</when>
<otherwise>
     <rewrite-uri template= "/put"/>
     <set-backend-service base-url= "http://contoso.com/api/9.1/"/>
</otherwise>
</choose>
```

□□ □ □□□ □□ □□□ □□□ □□ □□□□□□. □□□ □□□ □□□□ □□□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Statement	Yes	No
The XML segment belongs in the <inbound> section of the policy.	○	○
If the body size is >256k, an error will occur.	○	○
If the request is http://contoso.com/api/9.2/, the policy will retain the higher version.	○	○

Answer:

Statement	Yes	No
The XML segment belongs in the <inbound> section of the policy.	○	○
If the body size is >256k, an error will occur.	○	○
If the request is http://contoso.com/api/9.2/, the policy will retain the higher version.	○	○

Explanation:

Statement	Yes	No
The XML segment belongs in the <inbound> section of the policy.	○	○
If the body size is >256k, an error will occur.	○	○
If the request is http://contoso.com/api/9.2/, the policy will retain the higher version.	○	○

Box 1: Yes

Use the set-backend-service policy to redirect an incoming request to a different backend than the one specified in the API settings for that operation. Syntax: <set-backend-service base-url="base URL of the backend service" /> Box 2: No The condition is on 512k, not on 256k.

Box 3: No

The set-backend-service policy changes the backend service base URL of the incoming request to the one specified in the policy.
Reference:
https://docs.microsoft.com/en-us/azure/api-management/api-management-transformation-policies

NEW QUESTION: 204
□□□□□ □□ □□□ □□□□ □□ □□□□□ □□□□ □□□□□□□ □□□□ □□□□.
□□□□ Azure Web PubSub □□□□ □□□□ □□□□□□□□ □□□ □□□□□. □□□□ Azure Functions □□ □□ □□□□□. □□□□ CDN(□□□ □□ □□□□)□ □□ □□□□□.
Azure □□□ □□ □□□□□ □□□□ □□ □□□□□ □□□□□ □□□.
CDN□ Azure □□ □□□ □□□□□ □□□□ □□□.
□□ □□□□ □□ HTTP □□□ □□□ □□□?
A. WebHook-Reauest-Callbackuirements
B. □□-□□-□□
C. □□□
D. □□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 205
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□ □ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□ □□□ □□□ □□□□ □□ □ □□□ □□ □□ □□□□ □□□ □□□□ □□ □ □□□□.
□ □□□ □□□ □□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □□ □□□ □□□□ □□□□.
□□□ □□□□ □□ SaaS(Software as a Service) □□□□ □□□□□. □□□□ □□□ □ □ □□□ □□□□ □□ Azure Storage Blob □□□□ □□□ □□□□□. □□□□ □□ □□□ □ □ V2□□□.
□□□ □□□□□ □□□ □□□ □□□ □□□□ □□□□ □□□□ □□ □□□□□ □□□.
□□□ □□□ □□□ □□□□ □□□□ □□□□□ 1□ □□□ □□□□□ □□□.
□□ □□□ □□□□ □□□□□ □□□□ □□□.
□□□: □□ □□□ □□□ □□□□ Blob □□□□□ □□□□□ Azure □□ □□ □□□□.
□□□□ □□□ □□□□□?
A. □
B. □□□
Answer: A (LEAVE A REPLY)
In the Consumption hosting plan, resources are added dynamically as required by your functions.
Reference:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-blob-triggered-function

NEW QUESTION: 206

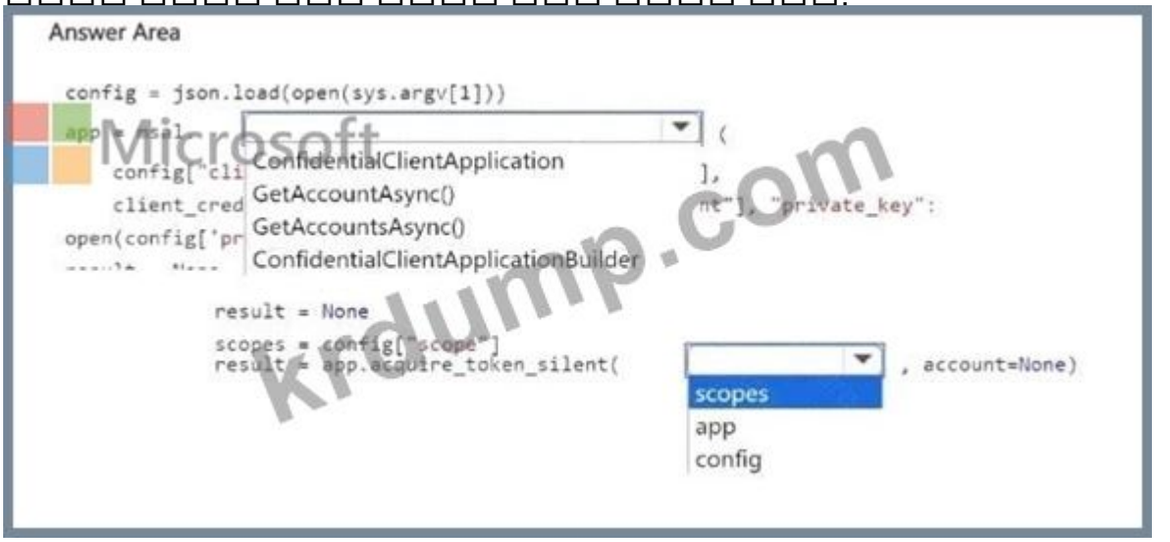
□□ □□ □□□ □□□□ □□□.
□□□ □□ □□□?
A. Application Insights □□ □□ □□ □□□
B. □□□ □□ host.json □□□ □□ □□ □□ □□
C. Application Insights □□□ □□
D. □□□□ □□ LogCategoryFilter □□
Answer: C (LEAVE A REPLY)
Scenario, the log capacity issue: Developers report that the number of log message in the trace output for the processor is too high, resulting in lost log messages.
Sampling is a feature in Azure Application Insights. It is the recommended way to reduce telemetry traffic and storage, while preserving a statistically correct analysis of application data. The filter selects items that are related, so that you can navigate between items when you are doing diagnostic investigations. When metric counts are presented to you in the portal, they are renormalized to take account of the sampling, to minimize any effect on the statistics.
Sampling reduces traffic and data costs, and helps you avoid throttling.
Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/sampling

NEW QUESTION: 207
Azure Web App□□ □□□□ □ □□□□□□□ □□□□ □□□□. □ □□□□□□□ Azure SQL Database□ □□□□ □□□□ Azure Storage □□□ □□□ □□□□□. □ □□□□□□ □ □□□□ □□□ □□□ □□ □□□□ □□ HTTP □□□ □□□□.
□ □□□□□□□ Application Insights□ □□□□□. □□ □□□□ OpenTelemetry□ □□□□ □.
□□□□ □□□□ □□ ID□ □□ □□□□ □□ □□□ □□□□ □□□ □□□□ □□□.
□□□ □□ □□□?
A. □□□□ □□□□ □□ ID□ □□□ TraceRags □□□ □ SpanContext□ □□□□.
B. □□ SpanContext□□ TraceId□ □□□□ □□□□ □□ ID□ □□□□□.
C. □□□□ □□□□ □□ ID□ □ □□□□□□□ CorrelationContext□ □□□□□.
D. Ocp-Apim-Trace □□□ □□□□ □□□□ □□ ID□ □□□□□.
Answer: C (LEAVE A REPLY)
Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/correlation

NEW QUESTION: 208
□□ □□ API□ □□□□□ □□□□ □□□□□. □□ API □□□□ RESTful □□□□ □□□ □ OpenAPI □□□ □□□□□.
Azure API Management □□□ □□□□□ □□□□ □□ API□ □□□□ □ □□□ □□□□ □ □□.
□□ Azure PowerShell □□□ □□□□ □□□?
A. □□□□-AzureRmApiManagementApi -Context $ApiMgmtContext -SpecificationFormat

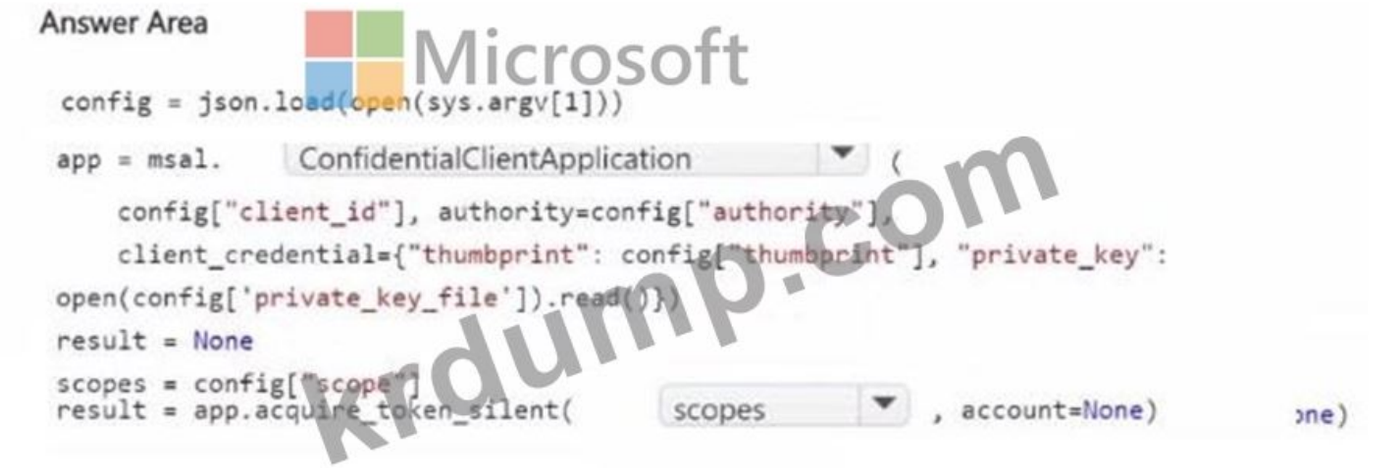
"Swagger" -SpecificationPath $SwaggerPath -□□ $Path
B. New-AzureRmApiManagementBackend -Context $ApiMgmtContext -Url $Url -Protocol http
C. New-AzureRmApiManagement -ResourceGroupName $ResourceGroup -Name $Name -Location $Location -Organization $Org -AdminEmail $AdminEmail
D. New-AzureRmApiManagementBackendProxy -Url $ApiUrl
Answer: A (LEAVE A REPLY)
New-AzureRmApiManagementBackendProxy creates a new Backend Proxy Object which can be piped when creating a new Backend entity.
Example: Create a Backend Proxy In-Memory Object
PS C:\>$secpassword = ConvertTo-SecureString "PlainTextPassword" -AsPlainText -Force PS C:\>$proxyCreds = New-Object System.Management.Automation.PSCredential ("foo", $secpassword) PS C:\>$credential = New-AzureRmApiManagementBackendProxy -Url "http://12.168.1.1:8080" - ProxyCredential $proxyCreds PS C:\>$apimContext = New-AzureRmApiManagementContext -ResourceGroupName "Api-Default- WestUS" -ServiceName "contoso" PS C:\>$backend = New-AzureRmApiManagementBackend -Context $apimContext -BackendId 123 -Url
'https://contoso.com/awesomeapi' -Protocol http -Title "first backend" -SkipCertificateChainValidation $true - Proxy $credential -Description "backend with proxy server"
Creates a Backend Proxy Object and sets up Backend References:
https://docs.microsoft.com/en-us/powershell/module/azurerm.apimanagement/new-azurermapimanagementbackendproxy?view=azurermps-6.13.0

NEW QUESTION: 209
Microsoft Graph API□ □□□□ □ □□□□□□□ □□□□ □□□□. Azure Portal□ □□□□ □□□ □□□□ □□□ X509 □□□□ □□□□□□.
□□□ □□, □□□□□□□ □□□□□ □□□, Microsoft Entra ID□ □□□ □□□□ □□□□ appsettings.json □□□ □□□□. X509 □□□□ □□□□ □□□□ ReadCertificate□□ □□□ □ □□□□.
□□□□ □□□□ □□□ □□□□ □□□ □□□□ □□□.



Answer:

Explanation:



NEW QUESTION: 210

Azure App Service □□□ □□□□ □□□□□. □□ □□ □□□ □□□□ Azure Traffic Manager□ □□□□□. □□ □□ Application Insights□ □□□□□ □□□□.

□□ □ □□ □□□ □□□□ □□□.

□□ □ □□ □□□□ □□□ □□□□□? □ □□□ □□□ □□□□ □□□□□. □□: □ □□ □ 1□□ □□□ □□□□.

A. Application Insights □□

B. Application Insights □ □□□

C. Azure Monitor □□

D. Azure Monitor □□□

Answer: A,C (LEAVE A REPLY)

Reference:

https://azure.microsoft.com/en-us/blog/creating-a-web-test-alert-programmatically-with-application-insights/

NEW QUESTION: 211

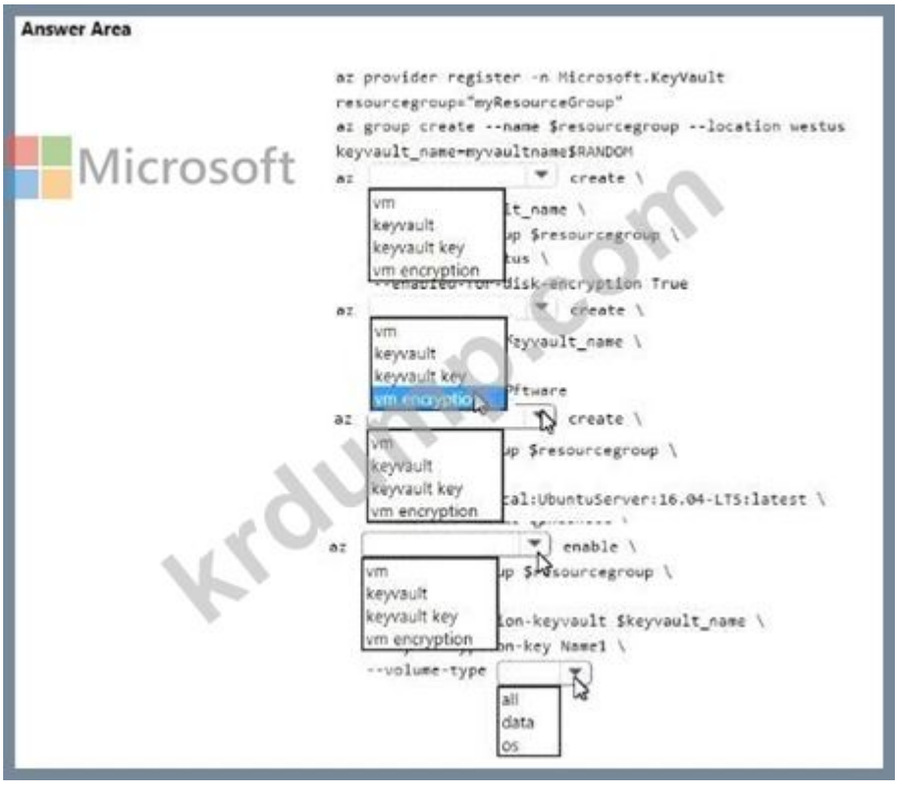
Azure□□ □□□□□ Linux VM(□□ □□)□ □ □□□□□□□ □□□ □□□□□.

□□ VM□ □□□ □□ □ □□ □□ □□ □□□ □□□□ □□ □□ □□ □□□ □□□ □□□ □ □□ □□□□ □□□□□ □□□.

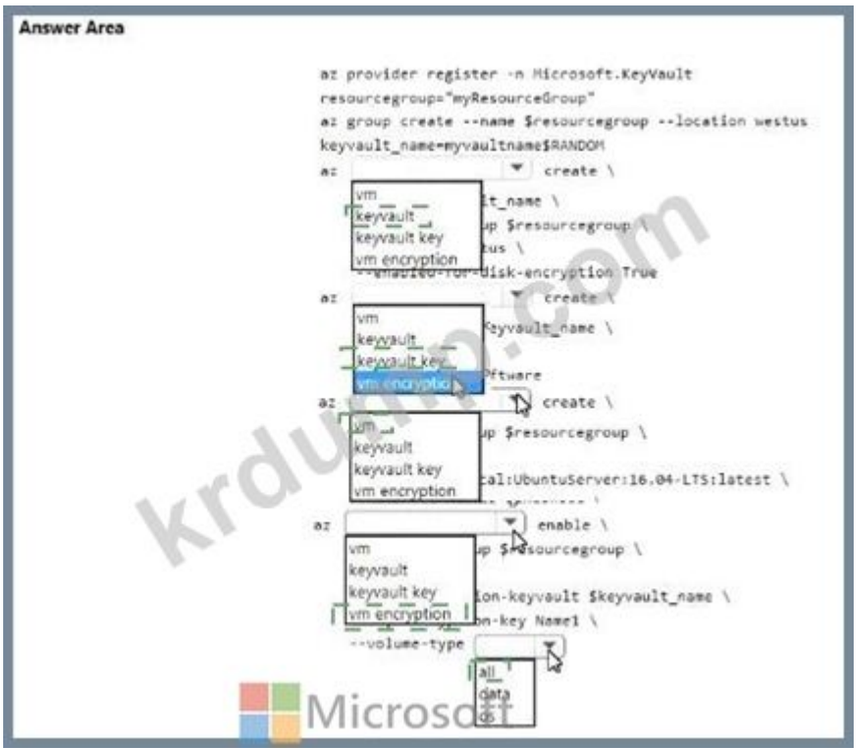
VM□ □□ Azure Disk Encryption□ □□□□ □□□.

Azure Cli □□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□ □.

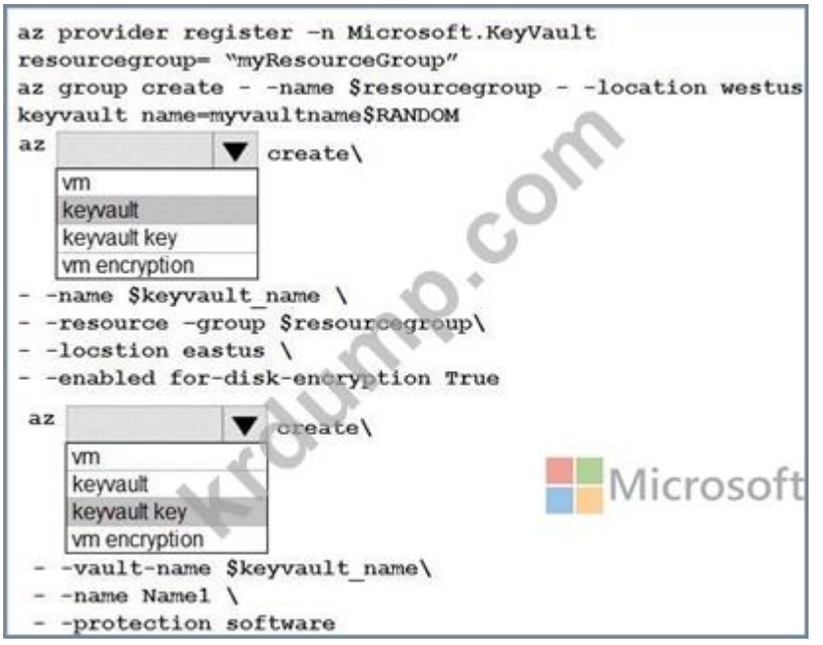
□□: □ □□□ □□□ 1□□ □□□ □□□□.

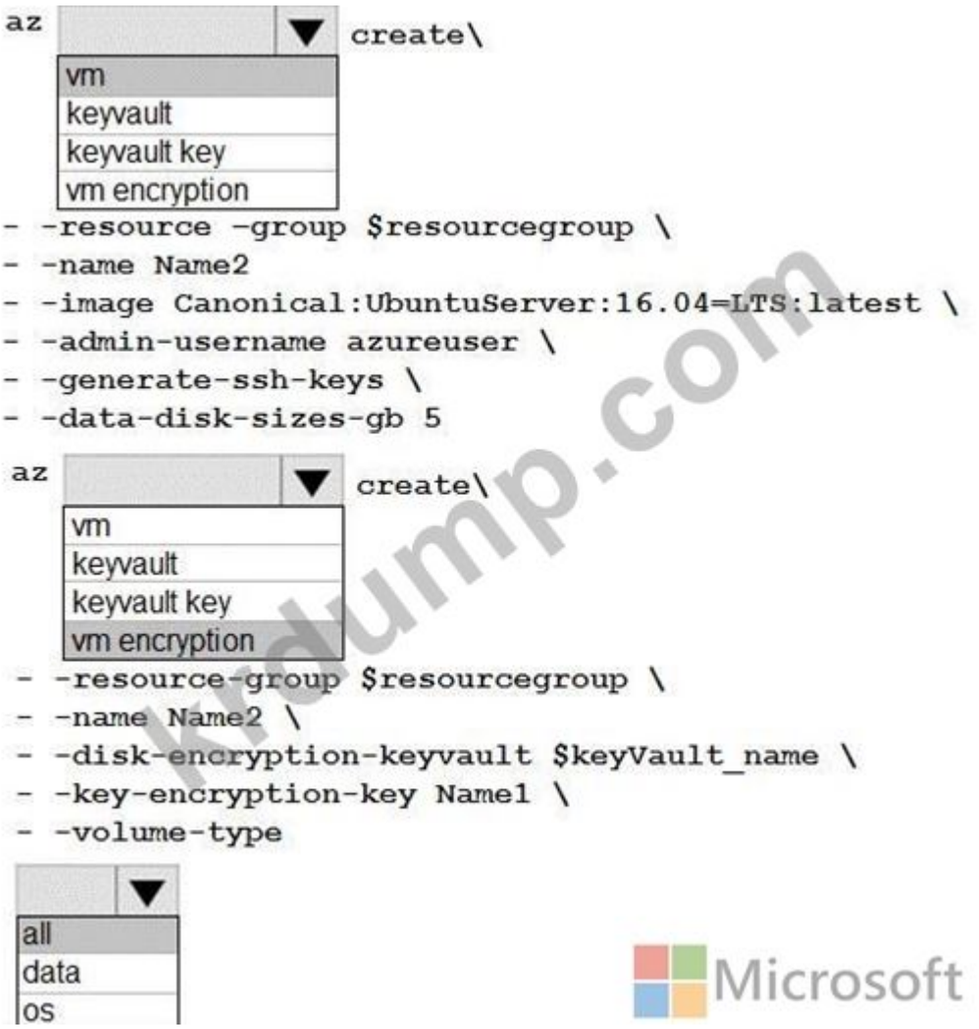


Answer:



Explanation:





Box 1: keyvault

Create an Azure Key Vault with az keyvault create and enable the Key Vault for use with disk encryption.

Specify a unique Key Vault name for keyvault_name as follows:

keyvault_name=myvaultname$RANDOM

az keyvault create \

--name $keyvault_name \

--resource-group $resourcegroup \

--location eastus \

--enabled-for-disk-encryption True

Box 2: keyvault key

The Azure platform needs to be granted access to request the cryptographic keys when the VM boots to decrypt the virtual disks. Create a cryptographic key in your Key Vault with az keyvault key create. The following example creates a key named myKey:

az keyvault key create \

--vault-name $keyvault_name \

--name myKey \

--protection software

Box 3: vm

Create a VM with az vm create. Only certain marketplace images support disk encryption. The following example creates a VM named myVM using an Ubuntu 16.04 LTS image:
az vm create \
--resource-group $resourcegroup \
--name myVM \
--image Canonical:UbuntuServer:16.04-LTS:latest \
--admin-username azureuser \
--generate-ssh-keys \
Box 4: vm encryption
Encrypt your VM with az vm encryption enable:
az vm encryption enable \
--resource-group $resourcegroup \
--name myVM \
--disk-encryption-keyvault $keyvault_name \
--key-encryption-key myKey \
--volume-type all
Note: seems to an error in the question. Should have enable instead of create.
Box 5: all
Encrypt both data and operating system.
References:
https://docs.microsoft.com/bs-latn-ba/azure/virtual-machines/linux/encrypt-disks

NEW QUESTION: 212
□□□□ Azure Cosmos DB□ □□□□□.
□□□ □□, □□□□ □□ □□□ □ □□□□ □□□□□□□ □□□□ □□□.
□□□ □□ □□□?
A. □□□ □□□ Lazy□ □□□□□.
B. □□□ □□□ □□ □□ □□ False□ □□□□□.
C. □□□ □□□ □□□□□ □□□□□.
D. EnableScanInQuery □□ □□ True□ □□□□□.
Answer: (SHOW ANSWER)

NEW QUESTION: 213

Azure Blob □□□□□ □□□□ □□□□□□□ □□□□ □□□□.
□□□□□□□ □□ □□□□ □□□ □□□ Blob □ Blob □□□□□□ □□□□ □□ □□ □ □□ □□□□ □□□ □□□ □□□. □□ □□□ □□□ □□□□ □□, □□□□, □□ □ □□ □□□ □□□□ □□ □□□ □□ □□□□ □□□.
□□□□ □□□ □□□□□□ □□□□ □□□.
□□□ □□ □□□?
A. □□□ Azure □□ □□ □□ Azure Event Grid□ □□□□ □□ Azure Blob □□□□ □□□□ □□□□□.
B. □□□□ □□□□ □□ □□□ □□□□□ □□ □□□ □□□□ □□ □□ □□ □□□ □□ □□□.
C. □□□□ Blob □□□□ □□ □□ Azure Storage □□ □□□ □□□□□.
D. Azure Monitor HTTP □□□ □□□ API□ □□□□ □□□□ Blob □□□□ □□ □□ □□□ □□□□□.
Answer: B (LEAVE A REPLY)
Change feed support in Azure Blob Storage
The purpose of the change feed is to provide transaction logs of all the changes that occur to the blobs and the blob metadata in your storage account. The change feed provides ordered, guaranteed, durable, immutable, read-only log of these changes. Client applications can read these logs at any time, either in streaming or in batch mode. The change feed enables you to build efficient and scalable solutions that process change events that occur in your Blob Storage account at a low cost.
Reference:
https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-change-feed

NEW QUESTION: 214
□ □□□ □□ □□□ □□□ □□□□□. □□ □□□ □□ □□□□ □□□□ □□□□□.
□□□ □□ □□ □□□ □□□□.
*□□□ □□ □□□□ □□ □□□ □□□□ □□□.
*□ □□□ playedId, gameId, score, time play□ □□□□□ □□□.
*□□□□ □□□ □□ □□□ □□□□ □□□□ □□□ SaveScore □□□ □□□□ □□□ □ □□ □□□□□.
*□ □□□ □□□ □□□ □□ □□□□ ID□ □□□□□.
□□ □□□ □□□□. (□ □□□ □□□□□□ □□□□ □□□□.)

```
01 public void SaveScore(string gameId, string playerId, int score, long timePlayed)
02 {
03   CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connectionString)
04   CloudTableClient tableClient = storageAccount.CreateCloudTableClinet();
05   CloudTable table = tableClient.GetTableReference("scoreTable");
06   table.CreateIfNotExists();
07   var scoreRecord = new PlayerScore(gameId, playerId, score, timePlayed);
08   TableOperation insertOperation = TableOperation.Insert(scoreRecord);
09   table.Execute(insertOperation);
10  }
11  public class PlayerScore : TableEntity
12 {
13   public PlayerScore(string gameId, string playerId, int score, long timePlayed)
14   {
15     this.PartitionKey = gameId;
16     this.RowKey = playerId;
17   Score = score;
18   TimePlayed = timePlayed;
19   }
20   public int Score { get; set; }
21   public long TimePlayed { get; set; }
22 }
```

□□ □□□ Azure Cosmos □□□□□□□ □□□□□. □□ □□□□ □□□□□□□ □□ □ □□□.

PartitionKey	RowKey	Email
Harp	Walter	wharp@contoso.com
Smith	Steve	ssmith@contoso.com
Smith	Jeff	jsmith@contoso.com

```
01   CloudTableClient tableClient = account.CreateCloudTableClient();
02   CloudTable table = tableClient.GetTableReference("people");
03   TableQuery<CustomerEntity> query = new TableQuery<CustomerEntity>()
04    .Where(TableQuery.CombineFilters(
05   TableQuery.Generate.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal, "Smith")
06   TableOperstors.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal,
"ssmith@contoso.com")
07   ));
08   await table.ExecuteQuerySegmentedAsync<CustomerEntity>(query, null);
```

□□ □ □□□ □□, □□□ □□□□□ '□'□ □□□□□. □□□ □□□ '□□□'□ □□□□□.

□□: □□ □□□ 1□□□□.

	Yes	No
The code will work with Cosmos DB.	○	○
The save score function will update and replace a record if one already exists with the same playerId and gameId.	○	○
The data for the game will be automatically partitioned.	○	○
This code will store the values for the gameId and playerId parameters in the database.	○	○

Answer:

Explanation:

Box 1: Yes

Code for CosmosDB, example:

```
// Parse the connection string and return a reference to the storage account.
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
CloudConfigurationManager.GetSetting("StorageConnectionString"));
// Create the table client.
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
// Retrieve a reference to the table.
CloudTable table = tableClient.GetTableReference("people");
// Create the TableOperation object that inserts the customer entity.
TableOperation insertOperation = TableOperation.Insert(customer1);
```

Box 2: No

A new record will always be added as TableOperation.Insert is used, instead of TableOperation.InsertOrReplace.

Box 3: No

No partition key is used.

Box 4: Yes

References:

https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet

NEW QUESTION: 215

API Management□ API□ □□□□□.

□□□□□ □□□□ □□□□ API□ □□ □□ □□□ □□□□ □□□.

□□□□□ □□□□ □□□□ API□ □□□ □□□□ □□ □□□□ □□□□ □□□.

□□ □ □□ □□ □□□ □□□ □ □□□□?

A. Azure AD □□

B. □□ DES(3DES) □□

C. □□ □□(CA) □□□

D. □□ □□□ □□□

E. □□ □

Answer: C,D (LEAVE A REPLY)

NEW QUESTION: 216

□□□□ □□□□ □□ □ □□ Python □□□□□ □□□□□.

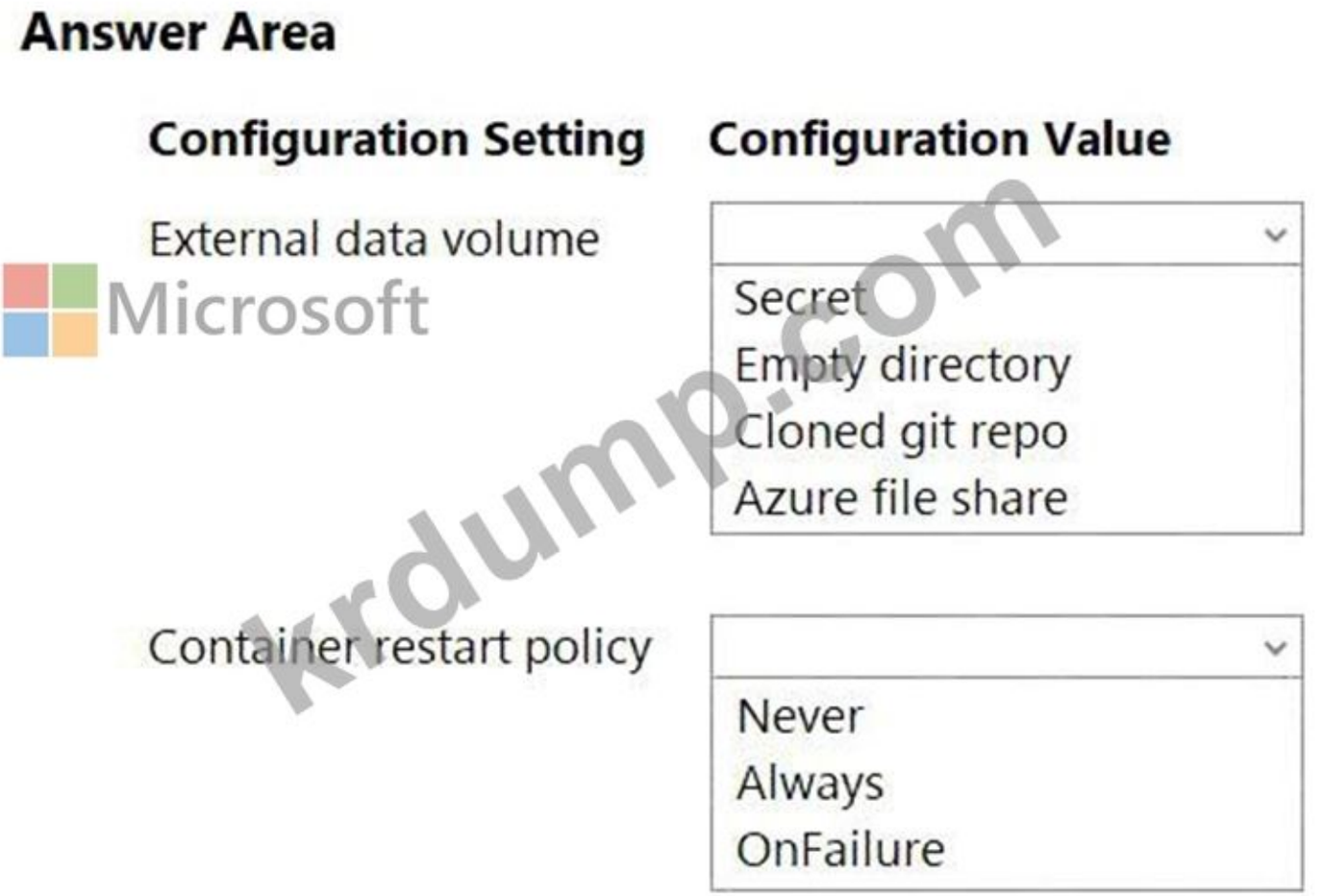
Python □□□□□ Azure Container Instance □□□□ □□□□ □□□□ □ □□ □□ Linux □ □□□□ □□□□□ □□□. □□□□□ SMB(□□ □□□ □□) □□□□□ □□□□ □□ □□ □□ □□□□□ □□□.

□□□□ □□□ □□□□□ □ □□ □□□□□ □□□.

Azure Container Instance□ □□□□ □□□.

□□ □□ □□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.

□□: □□□ □□□ □□ 1□□ □□□ □□□□.

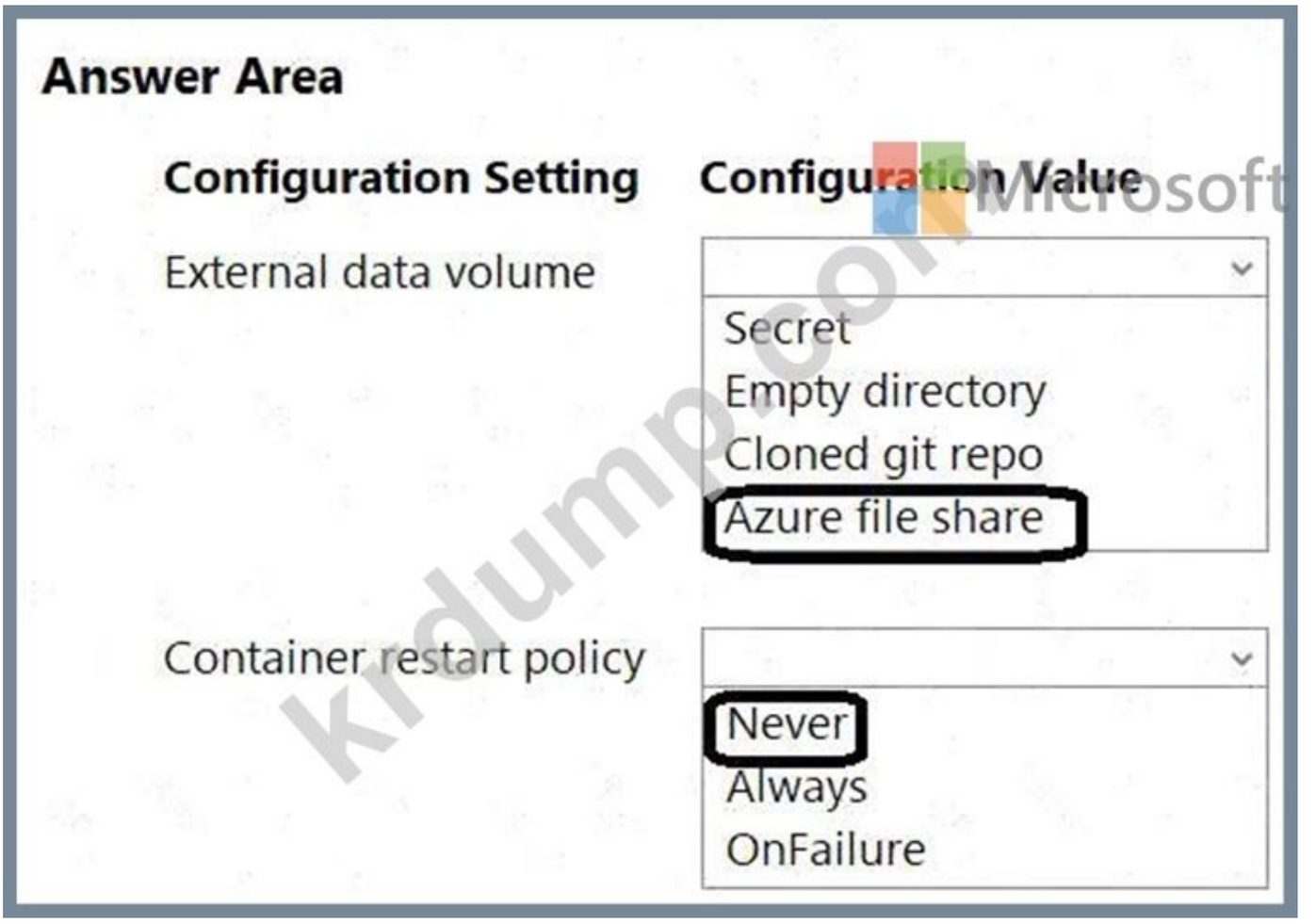


Answer:

Answer Area

Configuration Setting	Configuration Value
External data volume	Secret Empty directory Cloned git repo Azure file share
Container restart policy	Never Always OnFailure

Explanation:



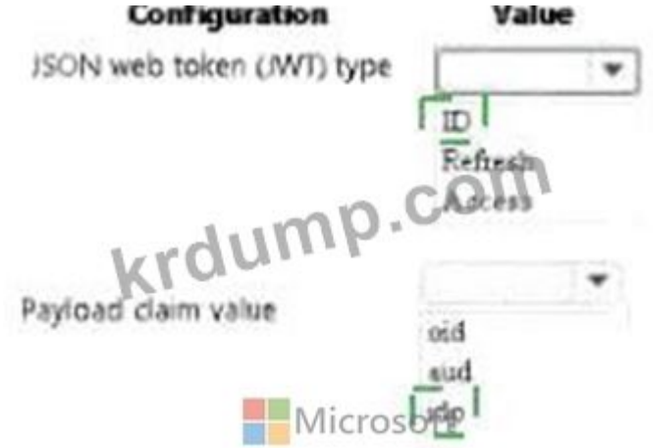
NEW QUESTION: 217

□□ □□ □□□ □□□ □□□□□ □□□□ □□□.

□□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□ □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:

Configuration	Value
JSON web token (JWT) type	ID
Payload claim value	idp

NEW QUESTION: 218

□□□□ Node.js □ □□ □□ □□□□. □□ □□□ □□ □□□ □□ GitHub □□□□□□□ □□□□□□.

https://github.com/TailSpinToys/weapp.

□□□□□□ □□□□ □□ □□□ □□□□ □□□. □□□□ □□ □□□ □□ □□ □□□□ □□□□ □□□.

□□□ □□□ □□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

```
$gitrepo="https://github.com/TailSpinToys/webapp"
$webappname="TailSpinToysWeb"
$location="WestUS2"

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name myResourceGroup -Location $location

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -ResourceGroupName myResourceGroup -Tier Standard

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -AppServicePlan $webappname -ResourceGroupName myResourceGrou

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -ResourceGroupName myResourceGroup -Slot review

$PropertiesObject = @{repoUrl = "$gitrepo";branch = "master";}
Set-AzResource -PropertyObject $PropertiesObject -ResourceGroupName myResourceGroup -ResourceType
Microsoft.Web/sites/slots/sourcecontrols -ResourceName $webappname/review/web -ApiVersion 2015-08-01 -Force
Switch-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup
-SourceSlotName review -DestinationSlotName production
```

Answer:

```
$gitrepo="https://github.com/TailSpinToys/webapp"
$webappname="TailSpinToysWeb"
$location="WestUS2"

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name myResourceGroup -Location $location

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -ResourceGroupName myResourceGroup -Tier Standard

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -AppServicePlan $webappname -ResourceGroupName myResourceGroup

New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -ResourceGroupName myResourceGroup -Slot review

$PropertiesObject = @{repoUrl = "$gitrepo";branch = "master";}
Set-AzResource -PropertyObject $PropertiesObject -ResourceGroupName myResourceGroup -ResourceType
Microsoft.Web/sites/slots/sourcecontrols -ResourceName $webappname/review/web -ApiVersion 2015-08-01 -Force
Switch-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup
-SourceSlotName review -DestinationSlotName production
```

Explanation:

The New-AzResourceGroup cmdlet creates an Azure resource group.

The New-AzAppServicePlan cmdlet creates an Azure App Service plan in a given location The New-AzWebApp cmdlet creates an Azure Web App in a given a resource group The New-AzWebAppSlot cmdlet creates an Azure Web App slot.

References:

https://docs.microsoft.com/en-us/powershell/module/az.resources/new-azresourcegroup?view=azps-2.3.2

https://docs.microsoft.com/en-us/powershell/module/az.websites/new-azappserviceplan?view=azps-2.3.2

https://docs.microsoft.com/en-us/powershell/module/az.websites/new-azwebapp?view=azps-2.3.2

https://docs.microsoft.com/en-us/powershell/module/az.websites/new-azwebappslot?view=azps-2.3.2

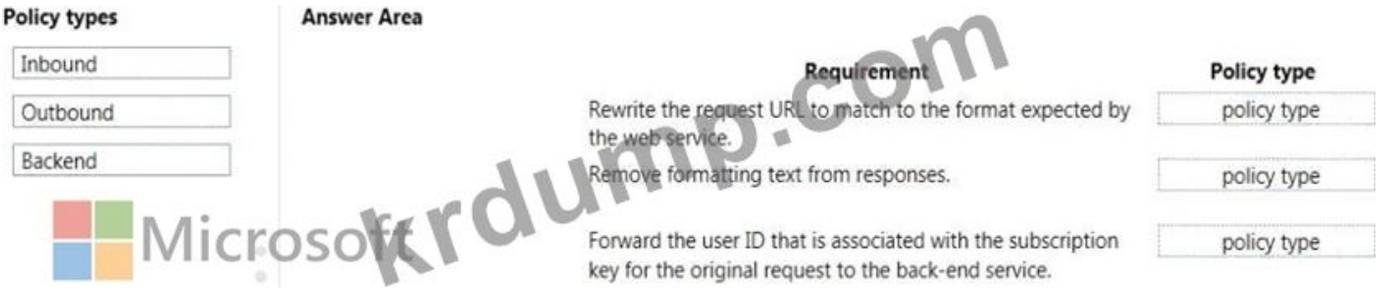
NEW QUESTION: 219

□□ □□□□□ □□ □□ □□□□ □□□□ □□□□□□□ □□□□. Azure API Management □ □□□□ API□ □□□□□.

□□ □□ □□□ □□□□□ API □□□ □□□□ □□□.

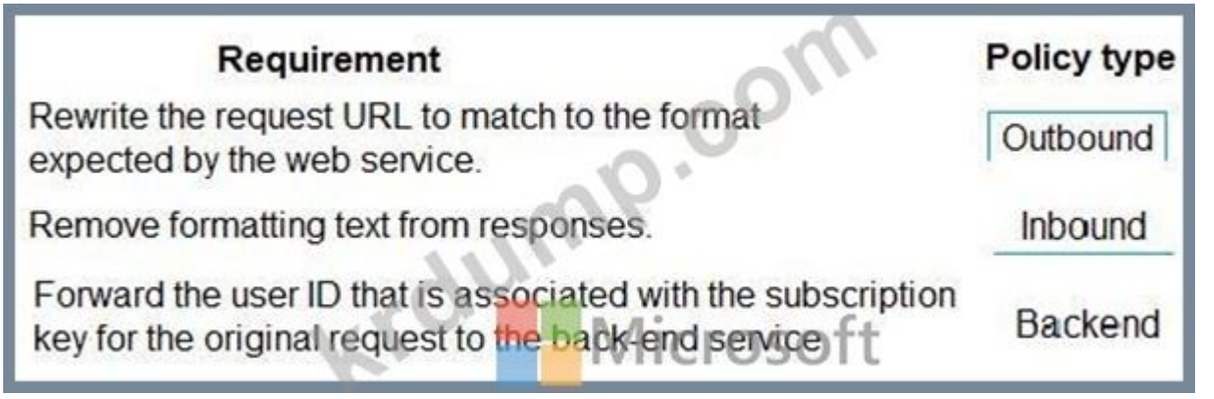
* □□ □□ □□□□□ □□□□□.
* □□□□ □□ □□□□ □□□□□.
* □□□ □□□□ □□ □□□□□ □□□□□.

□□ □□□ □□□ □□□□ □□□? □□□□□ □□ □□□ □□□ □□□□□ □□□□□□ □. □ □□ □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□. □ □□□ □□ □□□ □□□ □□□□ □□□ □□□□□ □ □ □□□□. □□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:

Explanation:



NEW QUESTION: 220

REST □ □□□□ □□ □□□□. □□□ Azure API Management □□□□□ □□□□ □□□□ □□□□□□.

□ □□□□ □□□ □□□□ □□□□ □□□□. □□□□ HTTP □□ □□ 409□ □□□□ □□ □□ □□ 500□ □□□□□. □□ □□□□ □□□□ □□□□□ □□□ □□□□□.

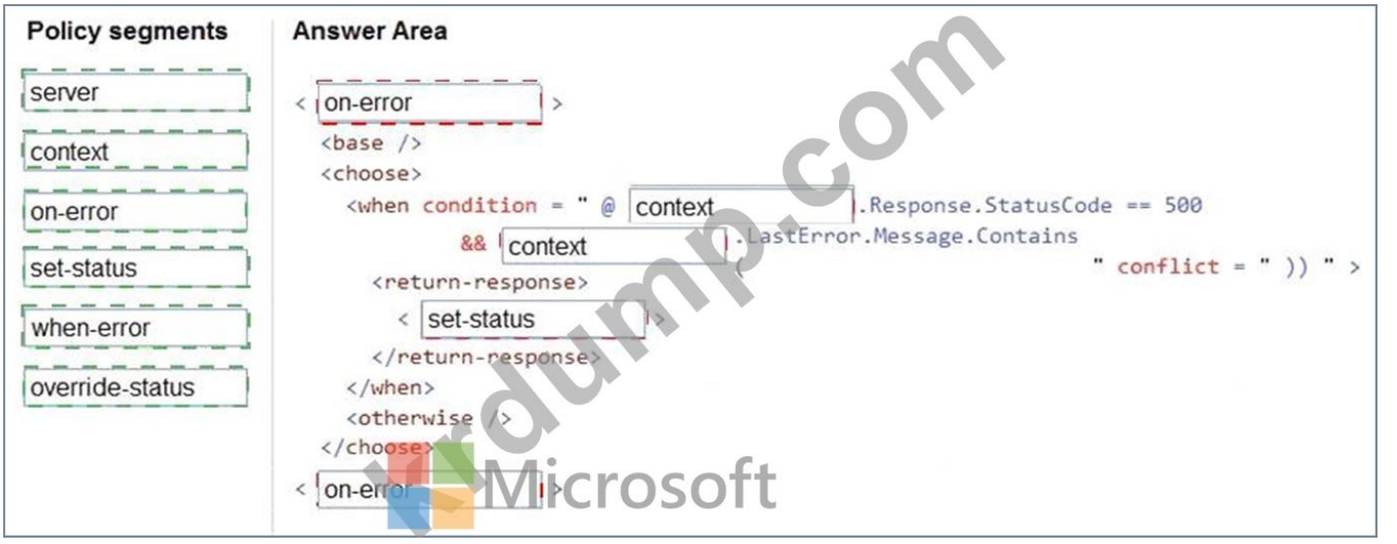
□□□ □□□ □□□ □□□□□ □□□□ □□□.

□□□ □□□ □□□□ □□□? □□□□□ □□□ □□ □□□□□ □□□ □□□ □□□□□ □□.

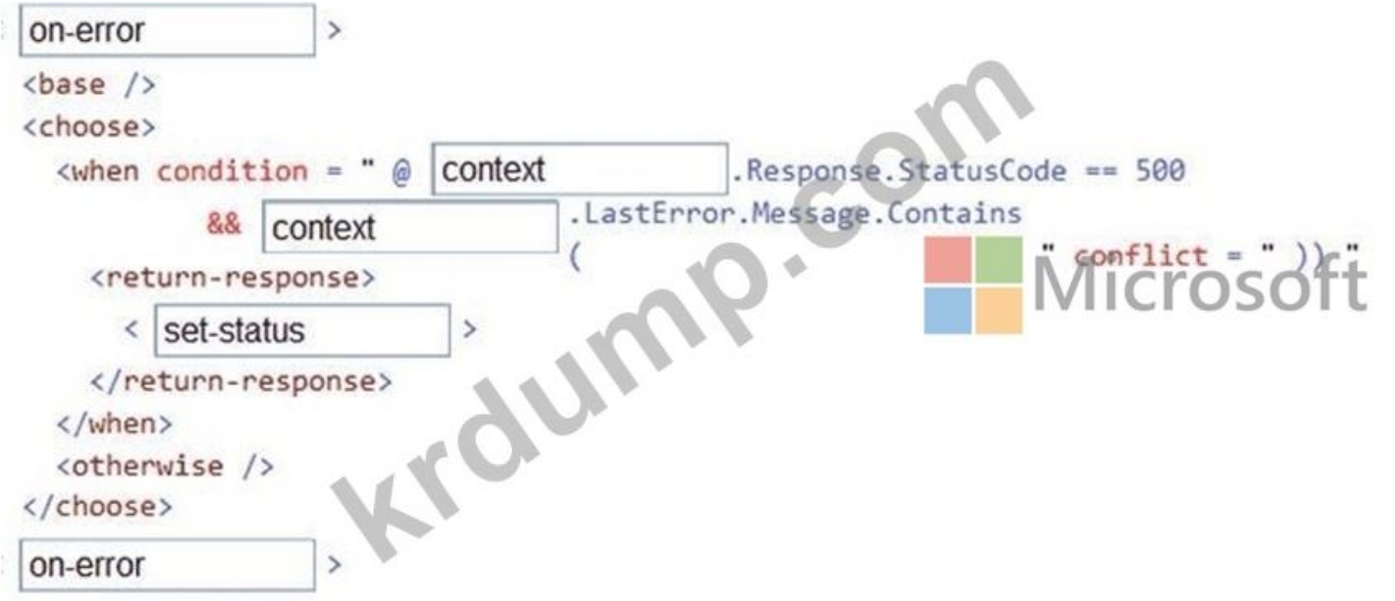
□ □□ □□□□□ □ □, □ □ □□ □□ □□ □□□□ □□ □ □□□□. □□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.

□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:



Box 1: on-error

Policies in Azure API Management are divided into inbound, backend, outbound, and on-error.
If there is no on-error section, callers will receive 400 or 500 HTTP response messages if an error condition occurs.

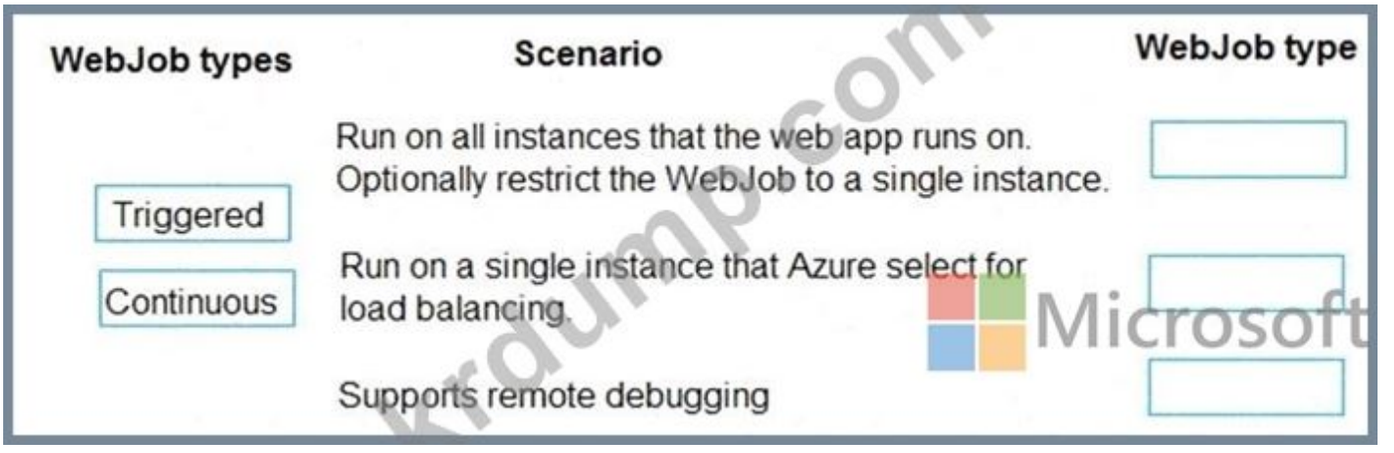
Box 2: context
Box 3: context
Box 4: set-status
The return-response policy aborts pipeline execution and returns either a default or custom response to the caller. Default response is 200 OK with no body.
Custom response can be specified via a context variable or policy statements.
Syntax:
<return-response response-variable-name="existing context variable">
<set-header/>
<set-body/>
<set-status/>
</return-response>
Box 5: on-error
Reference:
https://docs.microsoft.com/en-us/azure/api-management/api-management-error-handling-policies
https://docs.microsoft.com/en-us/azure/api-management/api-management-transformation-policies

NEW QUESTION: 221
Azure □□ □□ □□□□ Azure □□ □□ □□□□ □□□□□. Azure □□ □□ OpenAPI(Swagger) □□□ □□□□ Azure Blob □□□ □□□ □□□□□. □□ □□□□ Azure AD(Azure Active Directory)□ □□□□ □□□□□.
Azure □□ □□ Azure Blob □□□ □□□ □□□□ □□□□□ □□□. Azure Logic □□ □□□ □□ Azure AD □□□□ □□ □□□ □□□.
Azure □□ □□ □□□□ □□□.
□□□ □□ □□□?
A. Azure AD □□□ □□ □□□ □□□ □□ □□ □□□ □□□ □□□□□.
B. Azure AD □□□ □□ □□□ □□□ Azure Blob □□□ □□□ □□□ □□□□□.
C. Azure Key Vault□ □□□ □□□□□ □□□□ □□□□□.
D. □□□ □□ □□ ID□ □□□ □□ □□ □□□ □□□ □□□□□.
E. □□□ □□ □□ ID□ □□□ □□□□□ □□□□ □□□□□.
Answer: D (LEAVE A REPLY)
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:

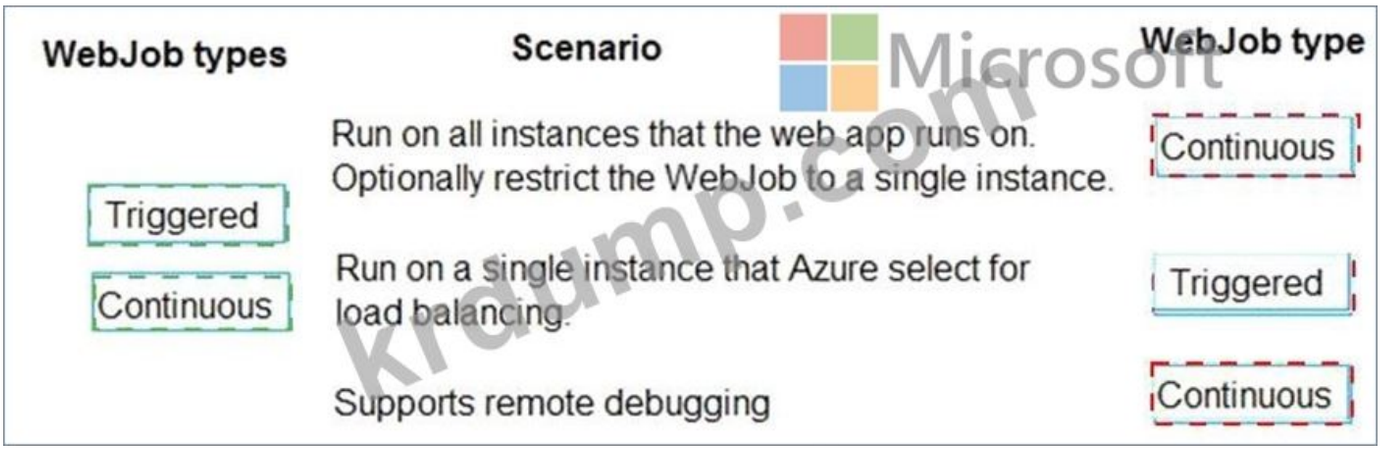
https://docs.microsoft.com/en-us/azure/logic-apps/create-managed-service-identity
https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates-for-clients

NEW QUESTION: 222
□□□ □□□□□□ □□□□ APIM1□□□ Azure API Management(APIM) □□ □□ □□□□ □ □□□□.
APIM1□ □□□□ □□ □□□ □□ □□□ □□□□ □□□ □□□□□□□ □□□□ API1□□ □ API□ □□□ □□□□□. □□ □□□ □□ IP □□□□ □□□ □□□□□□□ □□ □□ □ □ □□□□ □□□ □□□ □□□□ □□□□□ AP11□ □□ □□□ □□□□□.
□□ □□□ □□□□ API1 □□□ □□□□ □□□.
□□ □□□ □□□□ □□□?
A. ip-fitter
B. □□ □□ □□
C. □□ □□□
D. □□ □□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 223
Azure WebJobs□ □□ □□□□.
□ □□□□□ □□ WebJob □□□ □□□□ □□□.
□□ WebJob □□□ □□□□ □□□? □□□□□ □□□ WebJob □□□ □□□ □□□□□ □ □□□□□□. □ WebJob □□□ □ □, □ □ □□ □□□□□ □□ □□□□ □□ □ □□□□.
□□□□ □□□ □ □□□ □□ □□□ □□□ □□□□□ □ □ □□□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.



Answer:



Explanation:

Box 1: Continuous

Continuous runs on all instances that the web app runs on. You can optionally restrict the WebJob to a single instance.

Box 2: Triggered

Triggered runs on a single instance that Azure selects for load balancing.

Box 3: Continuous

Continuous supports remote debugging.

Note:

The following table describes the differences between continuous and triggered WebJobs.

Continuous	Triggered
Starts immediately when the WebJob is created. To keep the job from ending, the program or script typically does its work inside an endless loop. If the job does end, you can restart it.	Starts only when triggered manually or on a schedule.
Runs on all instances that the web app runs on. You can optionally restrict the WebJob to a single instance.	Runs on a single instance that Azure selects for load balancing.
Supports remote debugging.	Doesn't support remote debugging.

References:

https://docs.microsoft.com/en-us/azure/app-service/web-sites-create-web-jobs

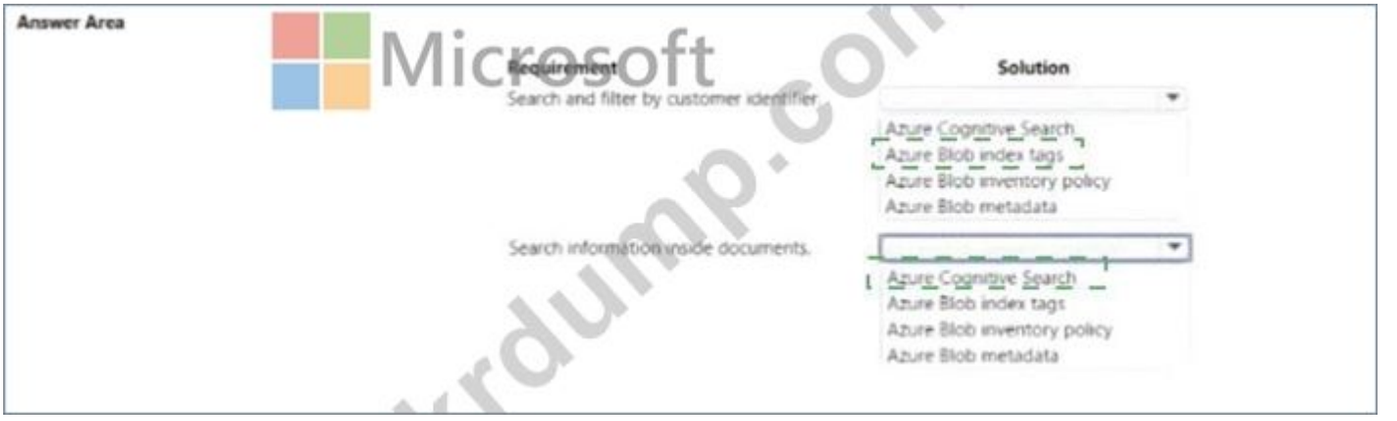
NEW QUESTION: 224

Azure Blob □□□□ □□□ □□□□ □□□□ □□□□ □□□□. □□□ □□ □□□□□ □□ □ □□□□□□. □□□ PDF, CSV, Microsoft Office □□ □ □□ □□□ □□□ □□□□□.
□□□□ □□ □□ □□□□□□ □□□ □□ □□□ □□□□ □□□. □□□□ □□ □□ □□ □ □□□□ □□□.

* □□□ □□□□ □□□ □□□□ □ □□ □□□□ □□□□□ □□□.
* □□ □□□□ □□□□ □□□□□.
* □□□ □□□ □□□ □□□ □ □□□□.
* □□□ □□□□□□.

□□□□ □□□□ □□ □□ □□ v2 □□□ □□□ □□□ □□□□□□.
□□□□ □□□□ □□□.
□□: □ □□□ □□□ 1□□ □□□ □□□□.

Answer:



Explanation:

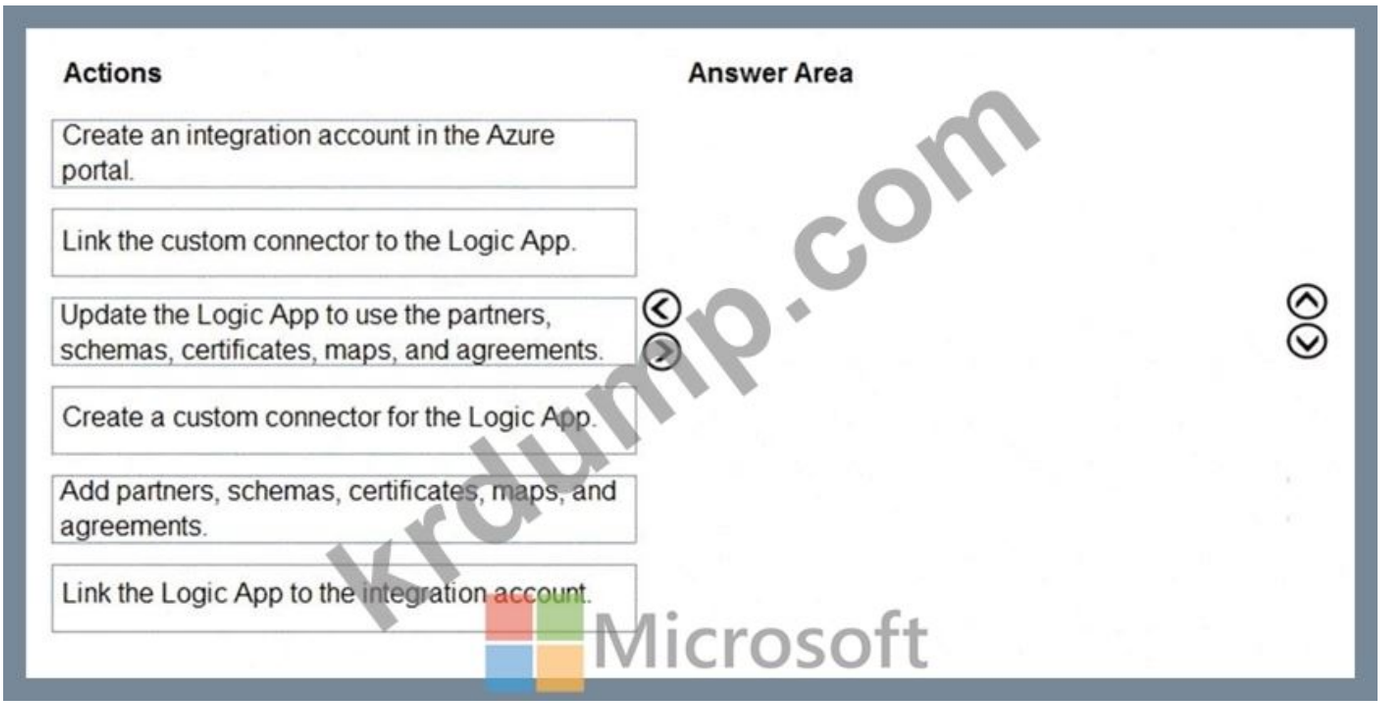
Azure Blob Index tags: https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-index-how-to?

tabs=azure-portal

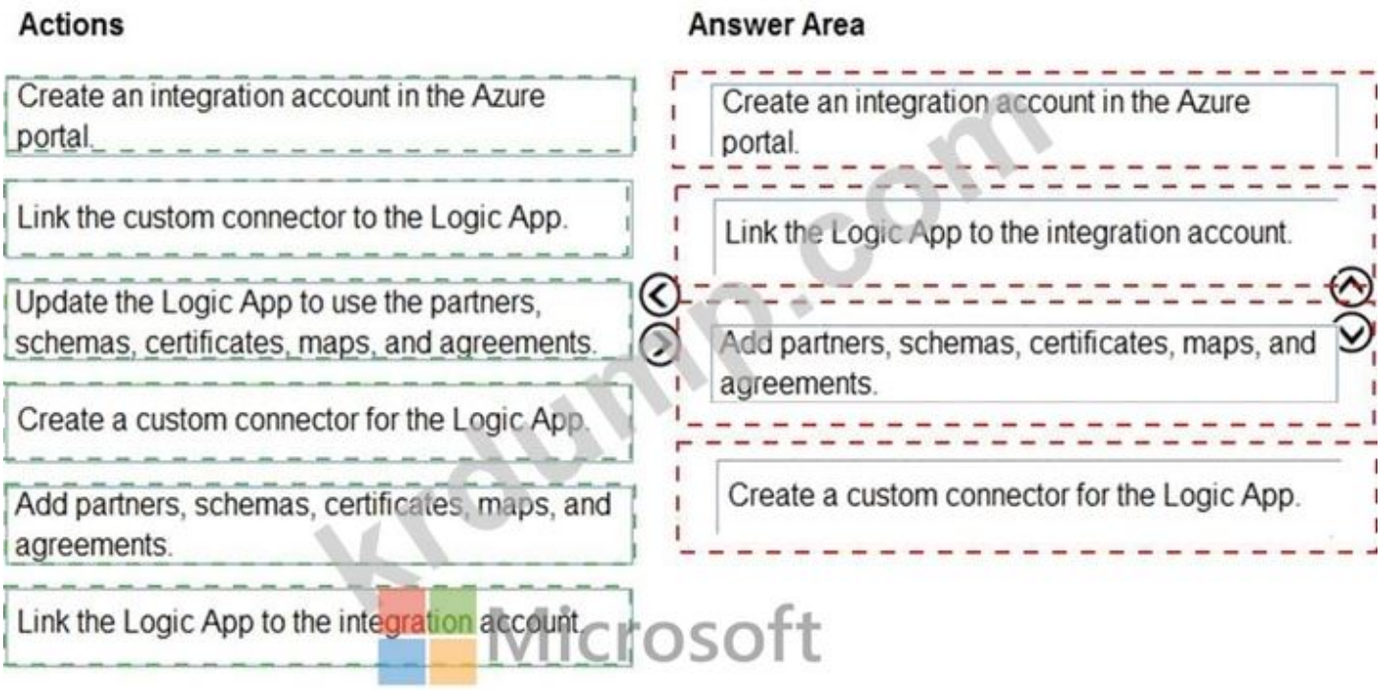
Azure Cognitive Search: Search inside documents

NEW QUESTION: 225

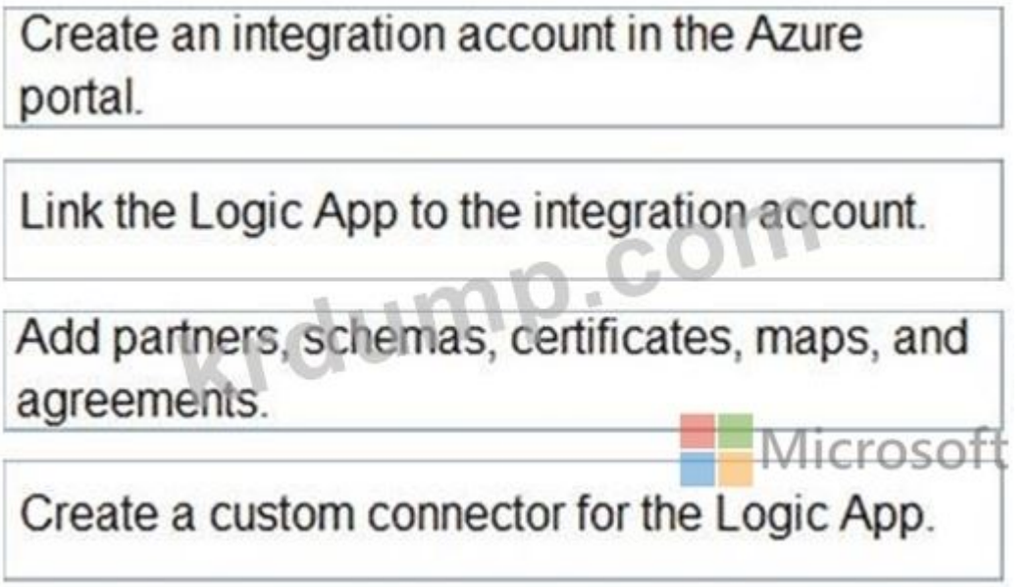
□□ □□ □□□□□□□□ □□□ □□□□ □□□ □□□? □□□□□□ □□ □□□□ □□ □□□ □□ □□ □□ □□□□ □□□ □□□ □□□□□□□.



Answer:



Explanation:



Step 1: Create an integration account in the Azure portal
You can define custom metadata for artifacts in integration accounts and get that metadata during runtime for your logic app to use. For example, you can provide metadata for artifacts, such as partners, agreements, schemas, and maps - all store metadata using key-value pairs.
Step 2: Link the Logic App to the integration account
A logic app that's linked to the integration account and artifact metadata you want to use.
Step 3: Add partners, schemas, certificates, maps, and agreements
Step 4: Create a custom connector for the Logic App.
References:
https://docs.microsoft.com/bs-latn-ba/azure/logic-apps/logic-apps-enterprise-integration-metadata

NEW QUESTION: 226
□□□□ □□ Blob Storage □□□ □□□□ □□□□□□□ □□□□ □□□□. Azure Blob Storage □□□ □□□ □□□□□ □□□ □□□□□ □□□□.
□□□□ □□□ □□ □□ □□□ □□□□□. □□□ □□□□ □□□□ □□□ □□□□ □□ □. (□ □□□ □□□□□□ □□□□ □□□□.)

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.	○	○

Answer:

Explanation:

1. Yes
2. Yes
3. Yes
4. No

https://docs.microsoft.com/en-us/azure/storage/blobs/lifecycle-management-overview?tabs=azure- portal#move-aging-data-to-a-cooler-tier

NEW QUESTION: 227

All□□□ Azure Application Insights □□□□ □□□□. AM□ Appl□□□ Azure App Service □ □□ □□□□□□□.

□□ □□□□ □□ Appl □□ □□□ □□□□ □□□□□ □□□ □□□□□. □ □□ □□□ □ □ Appl □□□□ □ □□ 5□ □□ □□□□□ □□□□ □□□□□. Azure Portal□□ "□□"□ □ □ □□ □□ □□□ □□□□ □□ □ □□□□ □□□□ □ □□□ □□□.

□□□ □□□□ □□□ AI1□ □□□□ □□□.

□□ □□□ □□ □□□ □□□□ □□□? □□□□□ □□ □□□□ □□□ □□□ □□□□□.
□□: □□ □□□ 1□□□□.

Answer:

Explanation:

NEW QUESTION: 228

Azure □□ □ □□□□□□□ □□□□ □□□□. □□□□□□□ □□□□□ □□□□ □□□ □□ □□□□ □□□ □□□ □□□□□. □□□□□□□ □□□□ □□□ □□ □□□ Azure Monitor □□□ □□□□ □□ □□ □□□□ □□□□□□.

□□□□□□□ □□ □□□□ □□□□□□□ □□□□□ □ □□ □□□□ □□□.

□□ □□ □□□□ □□□□ □□ □□ □□□□□ □□□ □□ □□□.

□□□ □□ □□□?

A. Azure Monitor □□□ □□□ □□□□.

B. □□□ □□□□ Azure Monitor □□ □□ □□□ □□□□□.

C. □□□□ □□ □□ Azure Monitor □□□ □□ □□□ □□□□ □□□ □□□□□.

D. □□□ □□□□ Azure Monitor □□ □□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 229

□□ □□□ □□□□ □□ Azure Durable Functions□ □□□□□.
□□ □□□□□□ □□□ □□□ □□□□ □□ □□ □□□ □□□□□. □□ □ □□□□ □□ □□ □ □□□ □□ □ □□ □□ □□ □□ □□□□□ □□□□□.
□□ □□□□□ □□ Azure Durable Functions□ □□□□ □□□.
□□ Azure Durable Functions □□□ □□□□ □□□?
A. □□
B. □□□□□□□
C. □□□
D. □□□□□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 230
□□□□ □□ □□ □□ □□□□ □□□□ □□□. □□ □□□ □□□□ □□□?
A. Azurite
B. Azure □□□ □□□
C. SQL Server □□ □□□□(SSMS)
D. Azure Storage □□□□□
Answer: B (LEAVE A REPLY)

NEW QUESTION: 231
□□: □ □□□ □□□ □□□□□ □□□□ □□□ □□□ □□□□□. □□□□ □ □□□□ □ □□ □□□ □□□ □ □□ □□□ □□□□ □□□□ □□□□. □□ □□ □□□□ □□□ □ □ □□ □□ □ □□□ □□ □□ □□□□ □□□ □□ □□ □□□□.
□ □□□ □□□ □□□ □□□ □□ □□□□ □□□ □ □□□□. □□□□□ □□□ □□□ □ □ □□□ □□□□ □□□□.
Azure Event Grid□ □□□□ □□ □□□ □□□ □□□□ □□□□ □□□□□□□ □□□□ □ □□□.
□□□ □□ □□ □□□ □□□□.
* □□ □□□ □□□ □□□ □□□ □□□□ □□□□ □□ □□ □□□□ □□□ □□□.
* □□□□ □□□□ □□ □□□ □□□□ □□□□□ □□□.
* □□ □ □□□ Microsoft Entra ID□ □□□□ □□□□ □□□.
* □□□□ □□ □□□□□□ □□□□□ □□□.
Azure Event Gnd□ □□□□ □□□.
□□ □□: □□□ □□ □□□ □□□□ □□□□□. □ □□□ □□ □□□ □□□ □□□□.
□□□□ □□□ □□□□□?
A. □□□
B. □
Answer: A (LEAVE A REPLY)
NEW QUESTION: 232
Azure □□□ □□□□ □□□□ □□□□.
□□ □□ □□□ □□□□ □□□□□ □□□□ □□□.

* □□ □□ □□
* □□ □□ □□□ □□□□□.
* □□□□ □□□ □□ □□ □□□□□.
□□ □ □□ □□□ □□ □□□ □□□□□? □□□ □□ □□□ □□□□ □□□□□. □□: □ □ □□□ 1□□□□.
A. Azure □□□ □□ □
B. Azure □□□ □
C. Azure □□□ □□ □□
D Azure □□□ □□
Answer: A,C (LEAVE A REPLY)
The Azure Service Bus Queue and Topic has duplicate detection.
Enabling duplicate detection helps keep track of the application-controlled MessageId of all messages sent into a queue or topic during a specified time window.
Reference:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/duplicate-detection