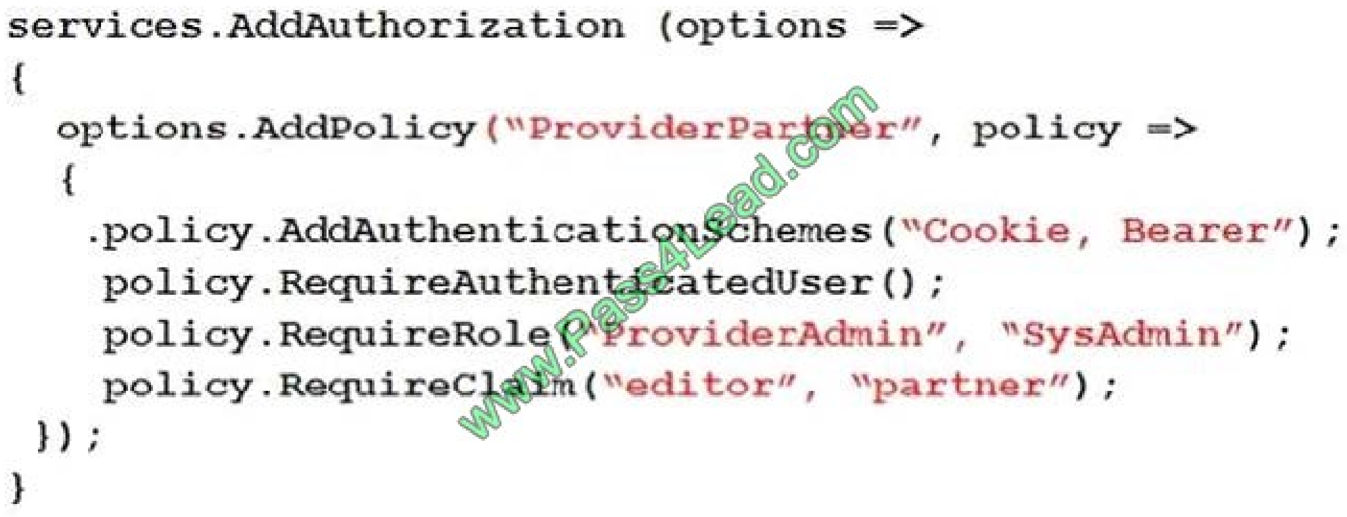
AZ-203$^{\text{Q\&As}}$

Developing Solutions for Microsoft Azure

QUESTION 1

DRAG DROP

You are developing a .NET Core model-view controller (MVC) application hosted on Azure for a health care system that allows providers access to their information. You develop the following code:



```
services.AddAuthorization (options =>
{
  options.AddPolicy("ProviderPartner", policy =>
  {
   .policy.AddAuthenticationSchemes("Cookie, Bearer");
    policy.RequireAuthenticatedUser();
    policy.RequireRole("ProviderAdmin", "SysAdmin");
    policy.RequireClaim("editor", "partner");
 });
}
```

You define a role named SysAdmin.

You need to ensure that the application meets the following authorization requirements:

Allow the ProviderAdmin and SysAdmin roles access to the Partner controller regardless of whether the user holds an editor claim of partner.

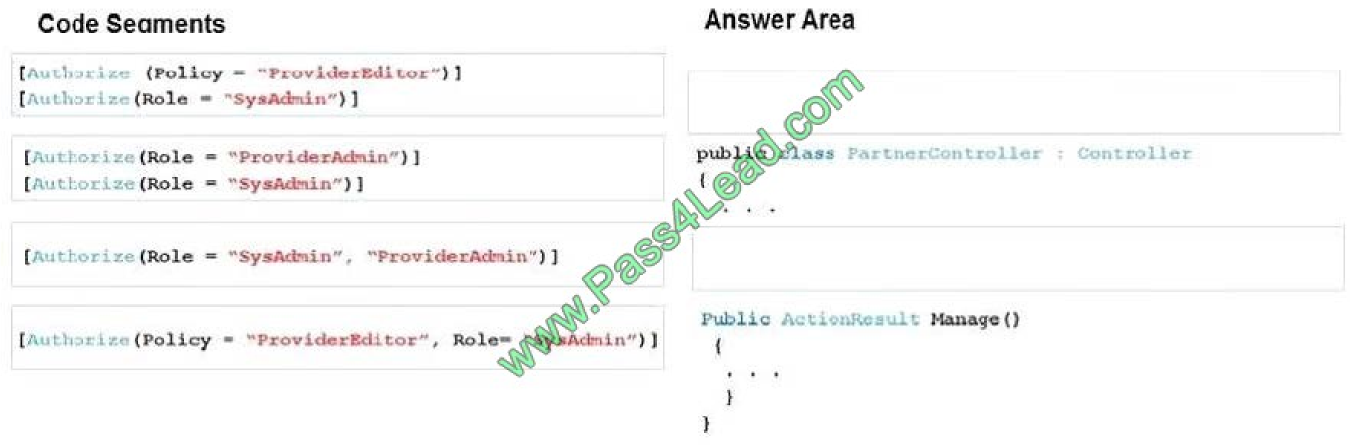
Limit access to the Manage action of the controller to users with an editor claim of partner who are also members of the SysAdmin role.

How should you complete the code? To answer, drag the appropriate code segments to the correct locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to

view content.

NOTE: Each correct selection is worth one point.

Select and Place:



Code Segments

```
[Authorize (Policy = "ProviderEditor")]
[Authorize(Role = "SysAdmin")]
```

```
[Authorize(Role = "ProviderAdmin")]
[Authorize(Role = "SysAdmin")]
```

```
[Authorize(Role = "SysAdmin", "ProviderAdmin")]
```

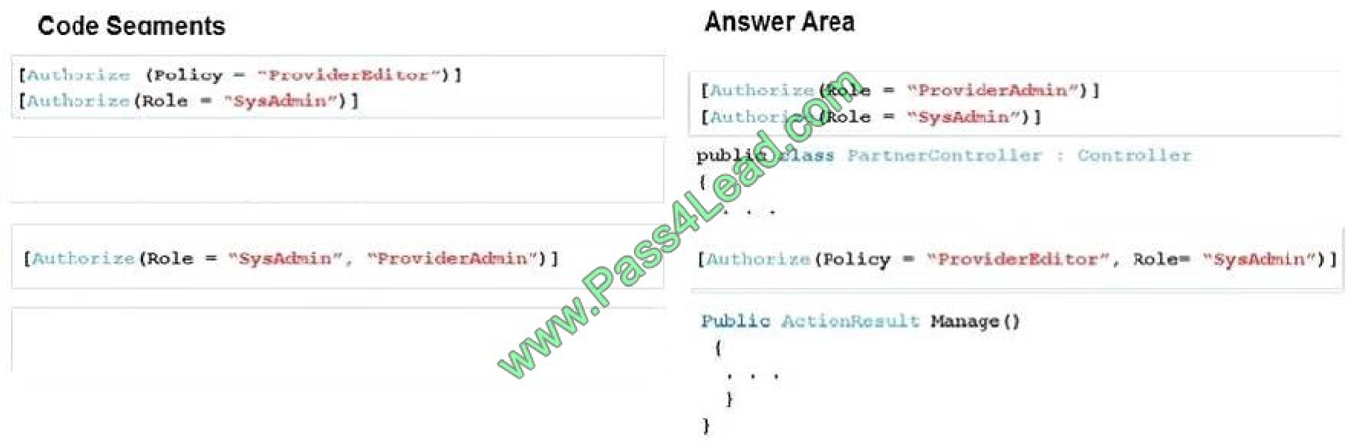
```
[Authorize(Policy = "ProviderEditor", Role= "SysAdmin")]
```

Answer Area

```
public class PartnerController : Controller
{
  . . .

 Public ActionResult Manage()
  {
    . . .
    }
}
```

Correct Answer:



Code Segments

```
[Authorize (Policy = "ProviderEditor")]
[Authorize(Role = "SysAdmin")]
```

```
[Authorize(Role = "SysAdmin", "ProviderAdmin")]
```

Answer Area

```
[Authorize(Role = "ProviderAdmin")]
[Authorize(Role = "SysAdmin")]
public class PartnerController : Controller
{
  . . .
[Authorize(Policy = "ProviderEditor", Role= "SysAdmin")]
 Public ActionResult Manage()
  {
    . . .
    }
}
```

Explanation:

Box 1:

Allow the ProviderAdmin and SysAdmin roles access to the Partner controller regardless of whether the user holds an editor claim of partner.

Box 2:

Limit access to the Manage action of the controller to users with an editor claim of partner who are also members of the SysAdmin role.

QUESTION 2

HOTSPOT

You are developing an Azure App Service hosted ASP.NET Core web app to deliver video on-demand streaming media. You enable an Azure Content Delivery Network (CDN) Standard for the web endpoint. Customer videos are downloaded

from the web app by using the following example URL:http//www.contoso.com/content.p4?quality=1

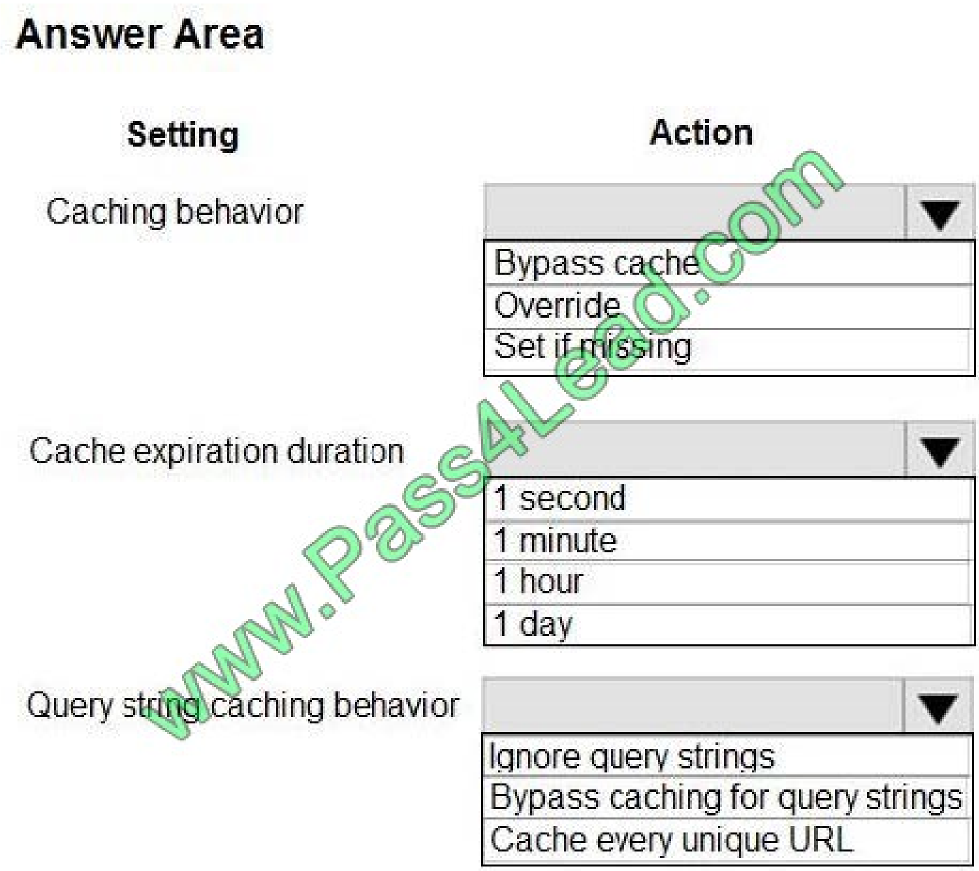
All media content must expire from the cache after one hour. Customer videos with varying quality must be delivered to the closest regional point of presence (POP) node.

You need to configure Azure CDN caching rules.

Which options should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

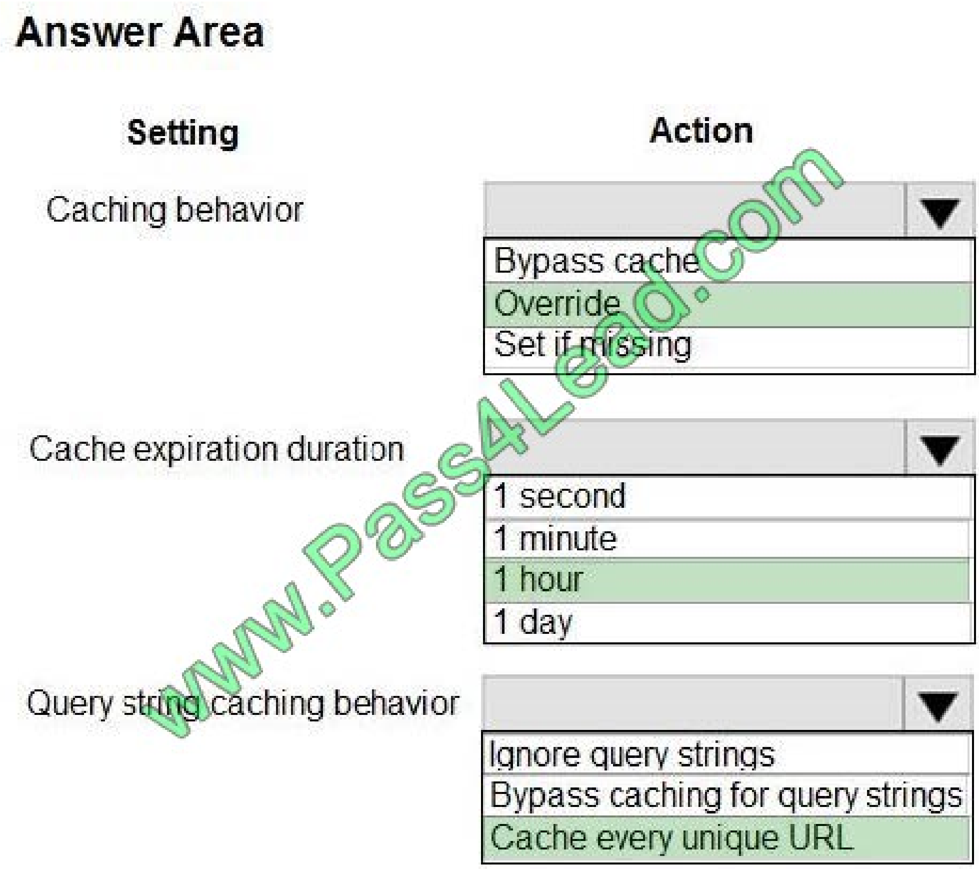
Hot Area:



Answer Area

Setting	Action
Caching behavior	Bypass cache Override Set if missing
Cache expiration duration	1 second 1 minute 1 hour 1 day
Query string caching behavior	Ignore query strings Bypass caching for query strings Cache every unique URL

Correct Answer:



Answer Area

Setting	Action
Caching behavior	Bypass cache [✓ Override] Set if missing
Cache expiration duration	1 second 1 minute [✓ 1 hour] 1 day
Query string caching behavior	Ignore query strings Bypass caching for query strings [✓ Cache every unique URL]

Box 1: Override

Override: Ignore origin-provided cache duration; use the provided cache duration instead. This will not override cache-control: no-cache.

Set if missing: Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration.

Incorrect:

Bypass cache: Do not cache and ignore origin-provided cache-directive headers.

Box 2: 1 hour

All media content must expire from the cache after one hour.

Box 3: Cache every unique URL

Cache every unique URL: In this mode, each request with a unique URL, including the query string, is treated as a unique asset with its own cache. For example, the response from the origin server for a request for example.ashx?q=test1 is

cached at the POP node and returned for subsequent caches with the same query string. A request for example.ashx?q=test2 is cached as a separate asset with its own time-to-live setting.

Incorrect Answers:

Bypass caching for query strings: In this mode, requests with query strings are not cached at the CDN POP node. The POP node retrieves the asset directly from the origin server and passes it to the requestor with each request.

Ignore query strings: Default mode. In this mode, the CDN point-of-presence (POP) node passes the query strings from the requestor to the origin server on the first request and caches the asset. All subsequent requests for the asset that are

served from the POP ignore the query strings until the cached asset expires.

References:

https://docs.microsoft.com/en-us/azure/cdn/cdn-query-string

QUESTION 3

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this question, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

Margie\\'s Travel is an international travel and bookings management service. The company is expanding into restaurant bookings. You are tasked with implementing Azure Search for the restaurants listed in their solution.

You create the index in Azure Search.

You need to import the restaurant data into the Azure Search service by using the Azure Search .NET SDK.

Solution:

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.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

1.

The index needs to be populated. To do this, we will need a SearchIndexClient. There are two ways to obtain one: by constructing it, or by calling Indexes.GetClient on the SearchServiceClient. Here we will use the first method.

2.

Create the indexBatch with the documents

Something like:

```
var hotels = new Hotel[];

{

new Hotel()

{

HotelId = "3",

BaseRate = 129.99,

Description = "Close to town hall and the river"

}

};

...

var batch = IndexBatch.Upload(hotels);
```

3.

The next step is to populate the newly-created index

Example:

```
var batch = IndexBatch.Upload(hotels);

try { indexClient.Documents.Index(batch); }
```

References: https://docs.microsoft.com/en-us/azure/search/search-howto-dotnet-sdk

---

QUESTION 4

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You need to meet the LabelMaker application security requirement.

Solution: Create a Microsoft Azure Active Directory service principal and assign it to the Azure Kubernetes Service (AKS) cluster.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

QUESTION 5

HOTSPOT

You are developing a ticket reservation system for an airline.

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

Ensure at least 99/99% availability and provide low latency.

Accept reservations even in network outages or other unforseen failures

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

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

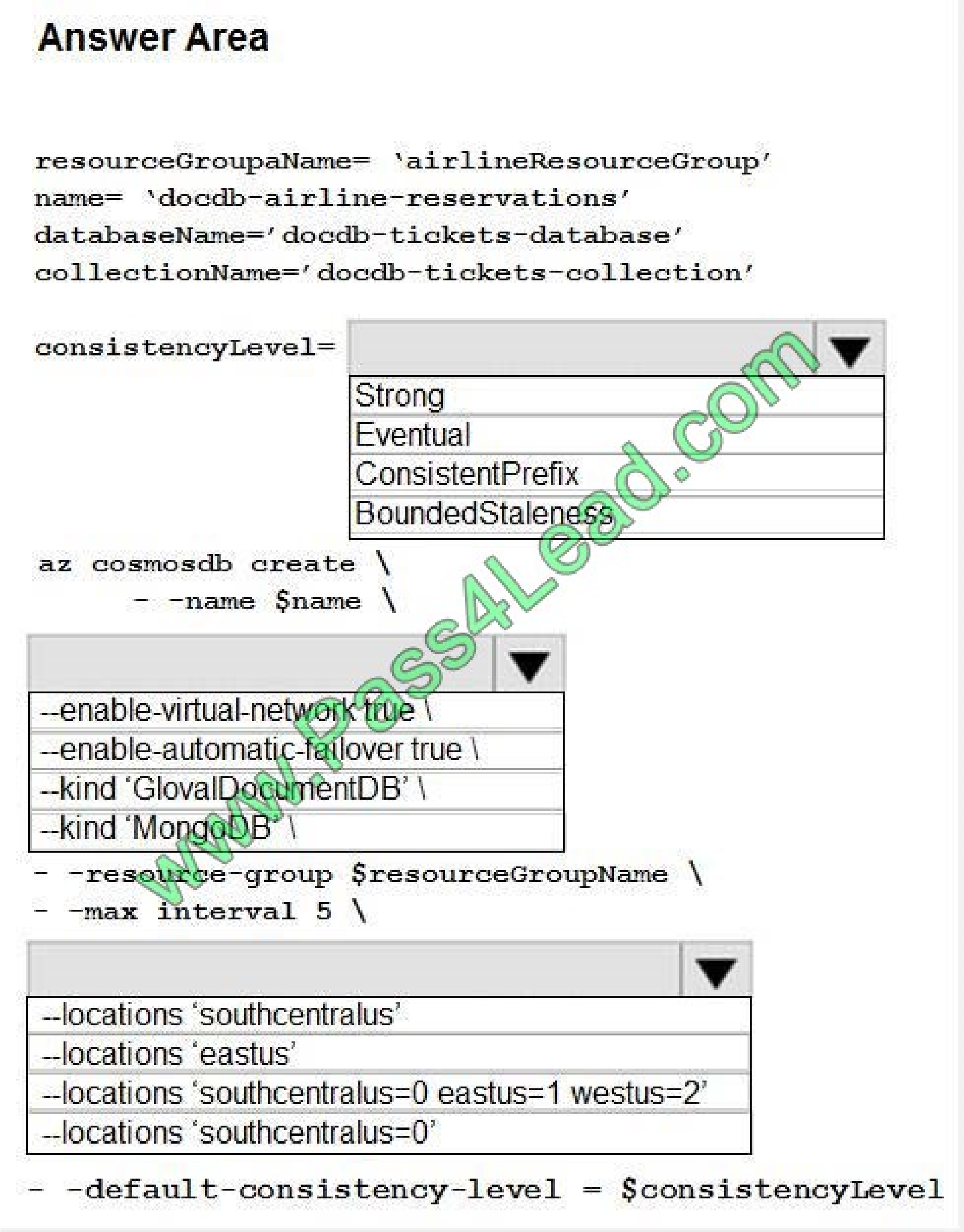
You provision a resource group named airlineResourceGroup in the Azure South-Central US region.

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

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

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area

```
resourceGroupaName= ‘airlineResourceGroup’
name= ‘docdb-airline-reservations’
databaseName=’docdb-tickets-database’
collectionName=’docdb-tickets-collection’

consistencyLevel=
```

- Strong
- Eventual
- ConsistentPrefix
- BoundedStaleness

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

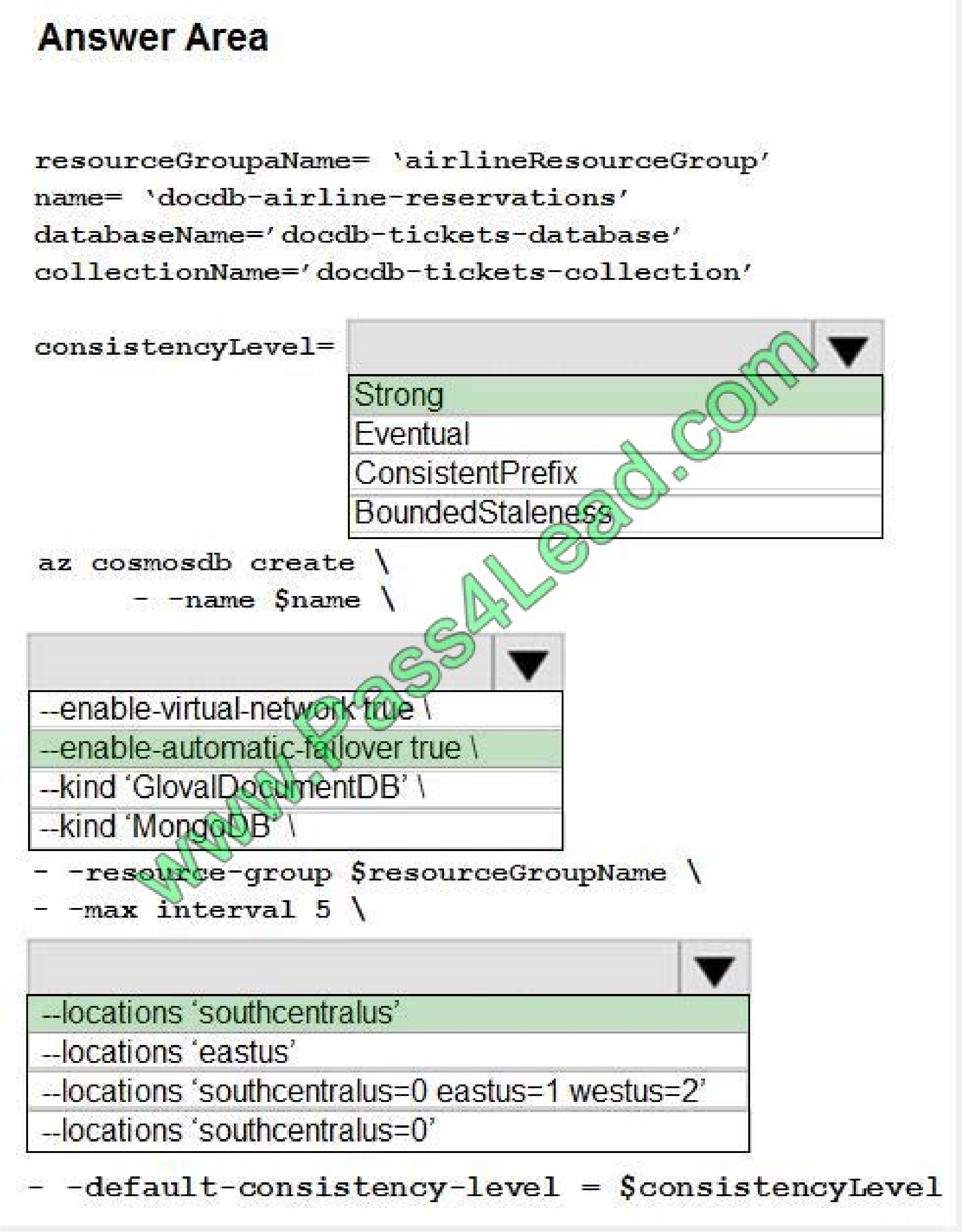
- --enable-virtual-network true \
- --enable-automatic-failover true \
- --kind ‘GlovalDocumentDB’ \
- --kind ‘MongoDB’ \

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

- --locations ‘southcentralus’
- --locations ‘eastus’
- --locations ‘southcentralus=0 eastus=1 westus=2’
- --locations ‘southcentralus=0’

```
- -default-consistency-level = $consistencyLevel
```

Correct Answer:



Answer Area

```
resourceGroupaName= 'airlineResourceGroup'
name= 'docdb-airline-reservations'
databaseName='docdb-tickets-database'
collectionName='docdb-tickets-collection'

consistencyLevel=
```

[✓ Strong]
Eventual
ConsistentPrefix
BoundedStaleness

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

--enable-virtual-network true \
[✓ --enable-automatic-failover true \]
--kind 'GlovalDocumentDB' \
--kind 'MongoDB' \

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

[✓ --locations 'southcentralus']
--locations 'eastus'
--locations 'southcentralus=0 eastus=1 westus=2'
--locations 'southcentralus=0'

```
- -default-consistency-level = $consistencyLevel
```

Box 1: 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.

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

Box 2: Enable-automatic-failover

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

" Accept reservations even in network outages or other unforseen failures"

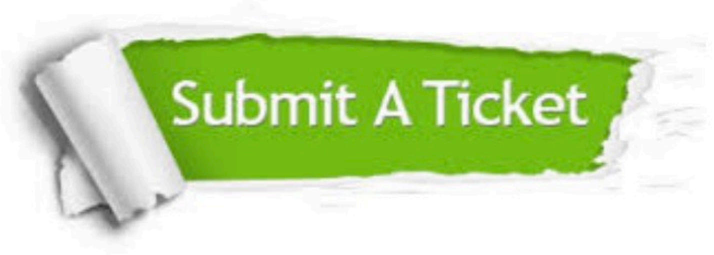
Box 3: southcentralus

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

References:

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

https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/cosmos-db/high-availability.md


Submit A Ticket