AZ-202.VCEplus.premium.exam.61q

Number: AZ-202
Passing Score: 800
Time Limit: 120 min
File Version: 1.0

AZ-202

Microsoft Azure Developer Certification Transition (beta)

Version 1.0

Testlet 1

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. 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 manager their receipts and perform any additional tasks needed for reimbursement.

Receipt processing
Employees may upload receipts in two ways:

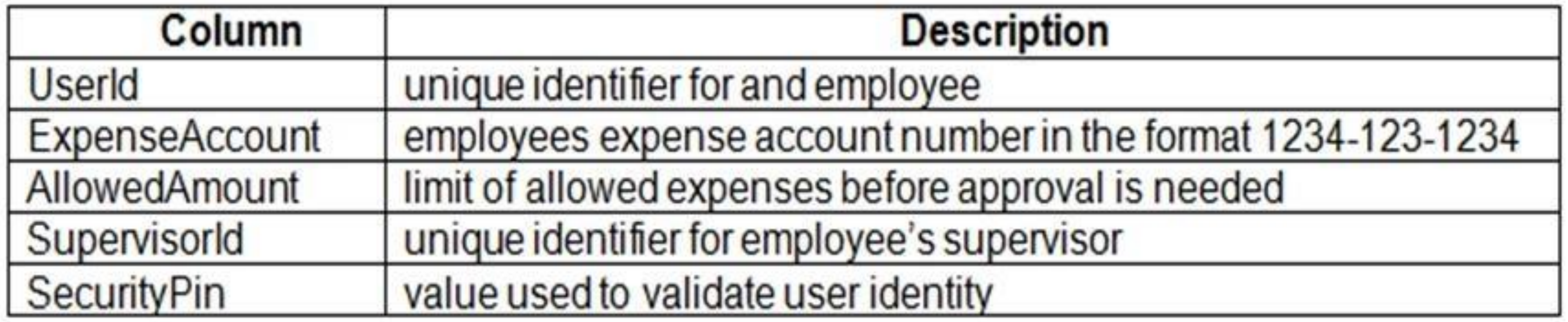
- Uploading using an Azure Files mounted folder
- Uploading using the web application

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.

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. **Requirements**

Receipt processing

Concurrent processing of a receipt must be prevented.

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.

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

▪ Users’ 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 Least Privilege Principal.
▪ 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.

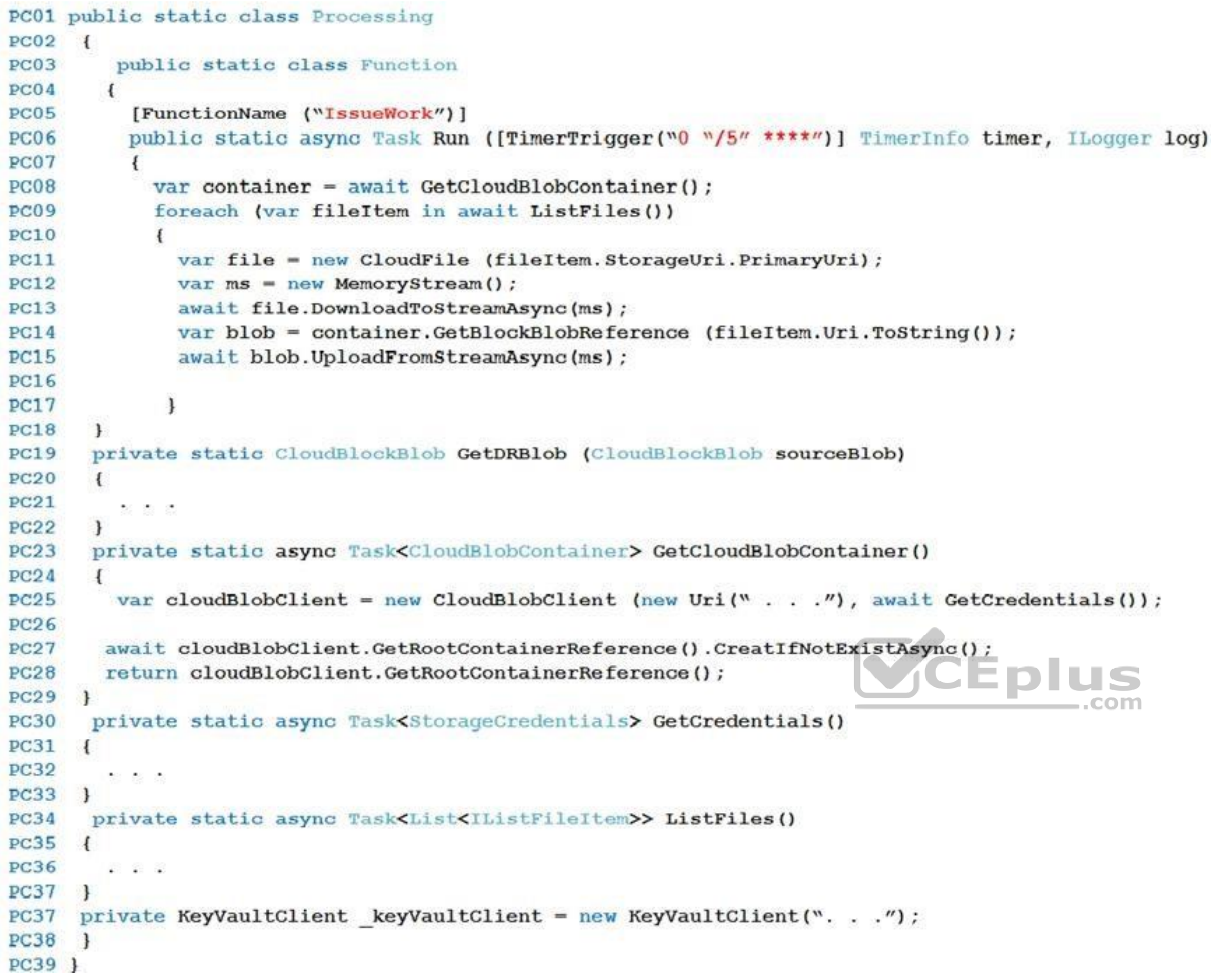
Capacity issue

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

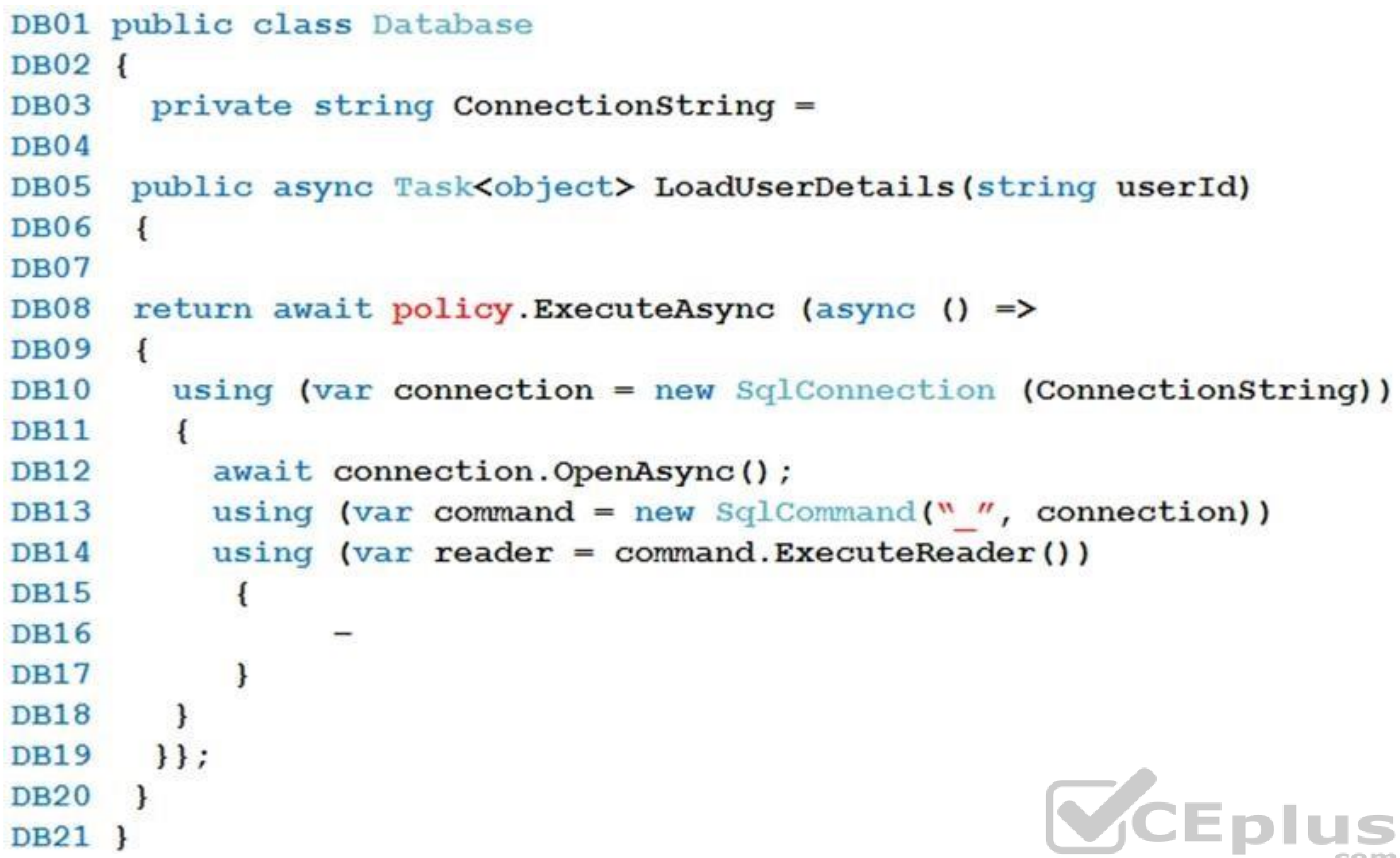
Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Processing.cs



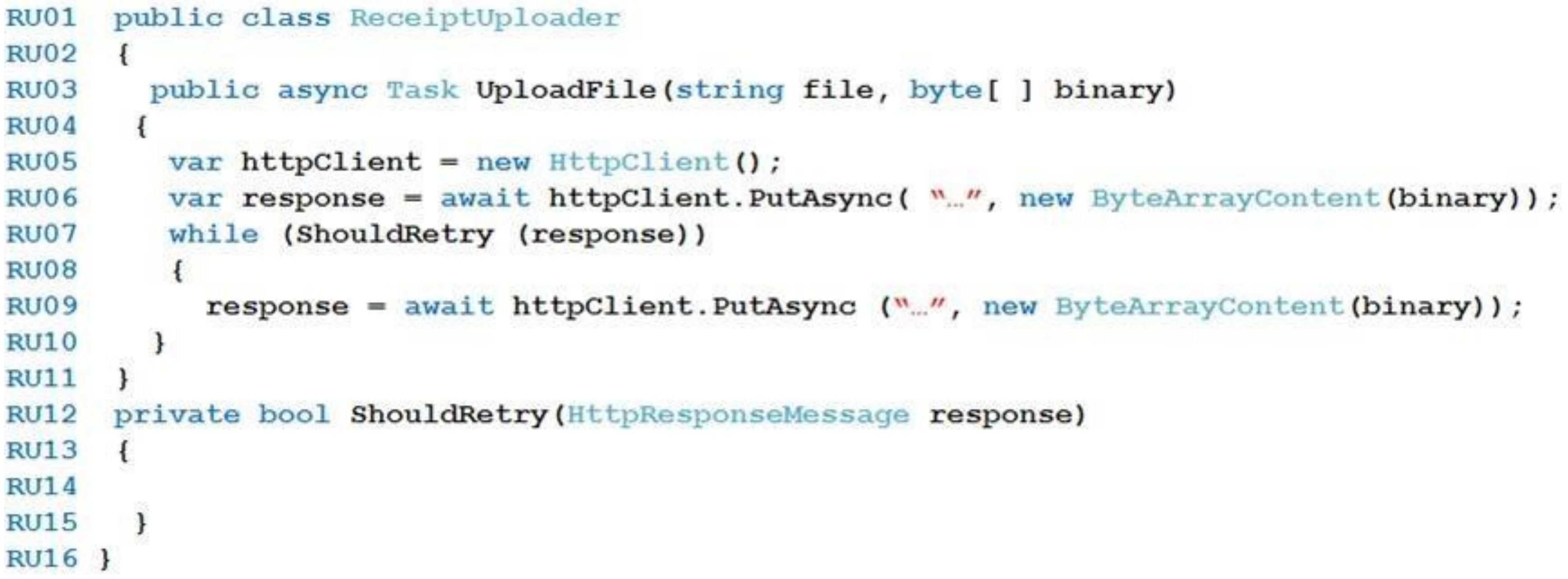
```
PC01 public static class Processing
PC02  {
PC03     public static class Function
PC04    {
PC05       [FunctionName (“IssueWork”)]
PC06      public static async Task Run ([TimerTrigger(“0 “/5” ****”)] TimerInfo timer, ILogger log)
PC07      {
PC08        var container = await GetCloudBlobContainer();
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().CreatIfNotExistAsync();
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



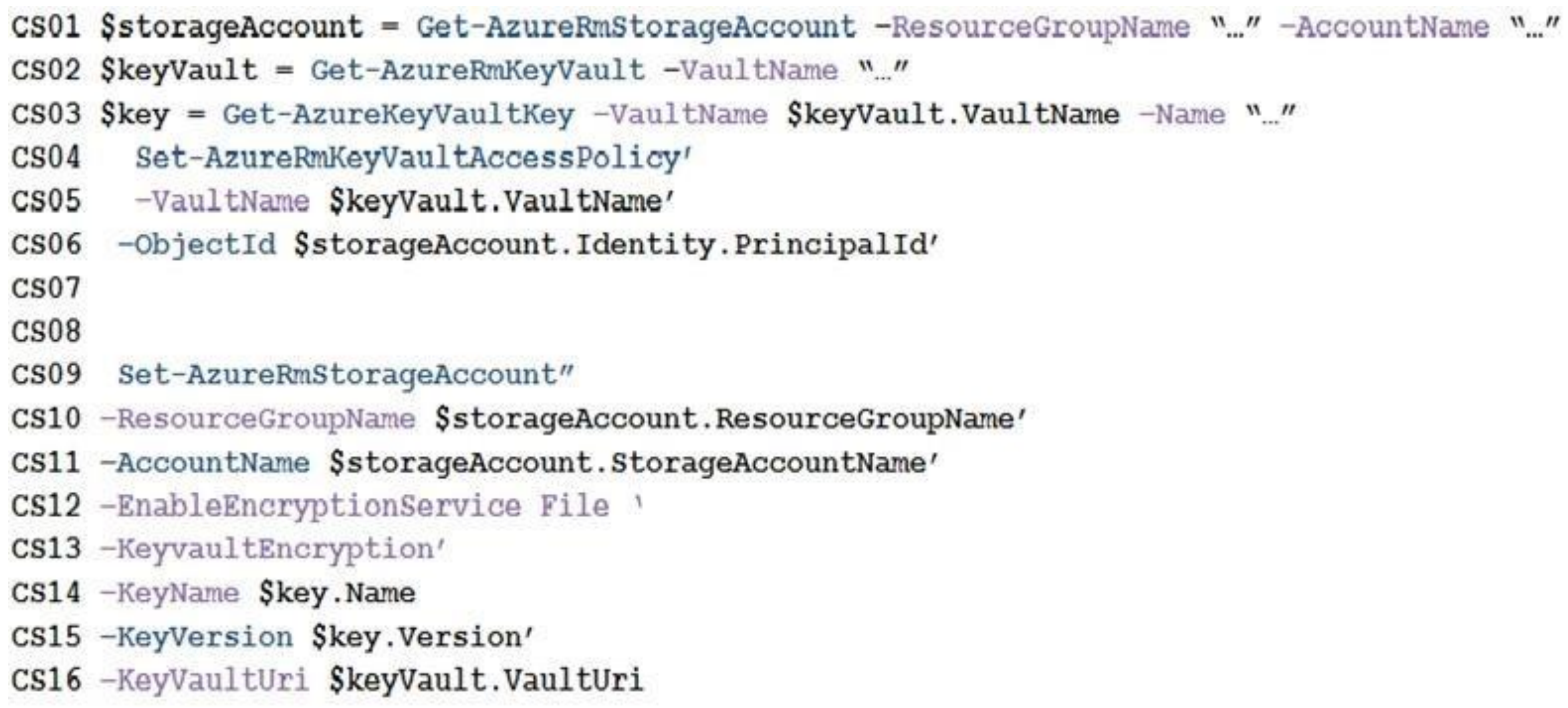
```
DB01 public class Database
DB02 {
DB03   private string ConnectionString =
DB04
DB05  public async Task<object> LoadUserDetails(string userId)
DB06  {
DB07
DB08  return await policy.ExecuteAsync (async () =>
DB09  {
DB10    using (var connection = new SqlConnection (ConnectionString))
DB11    {
DB12      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
```

QUESTION 1
DRAG DROP

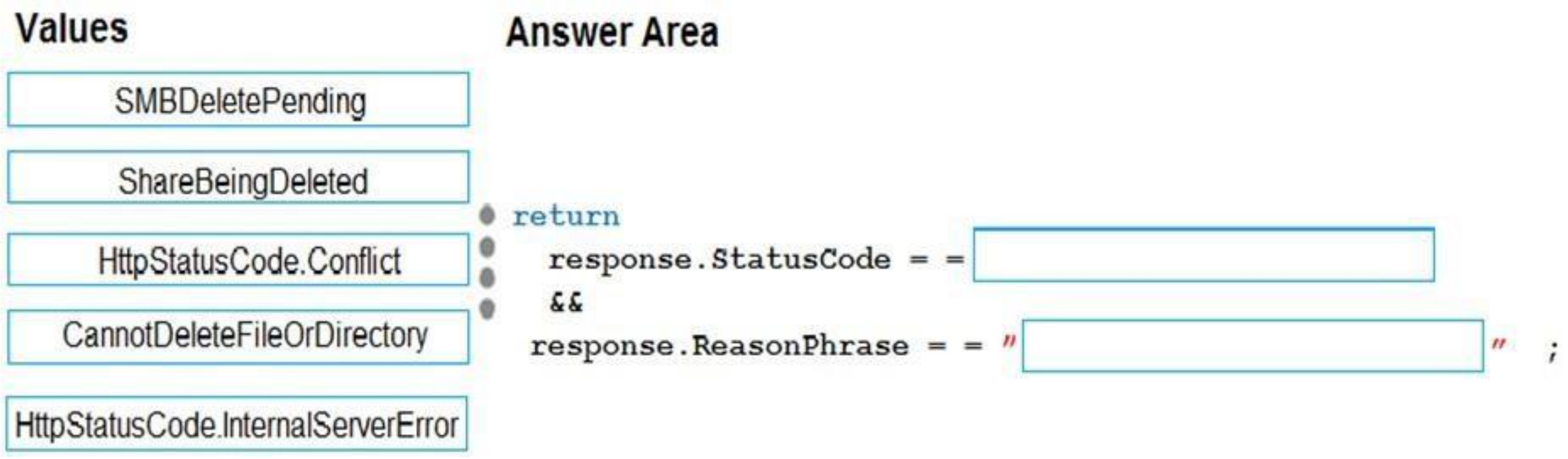
You need to ensure that the upload format issue is resolved.

What code should you add at line RU14?

To answer, drag the appropriate code fragments to the correct locations. Each code fragment 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:



Values

SMBDeletePending

ShareBeingDeleted

HttpStatusCode.Conflict

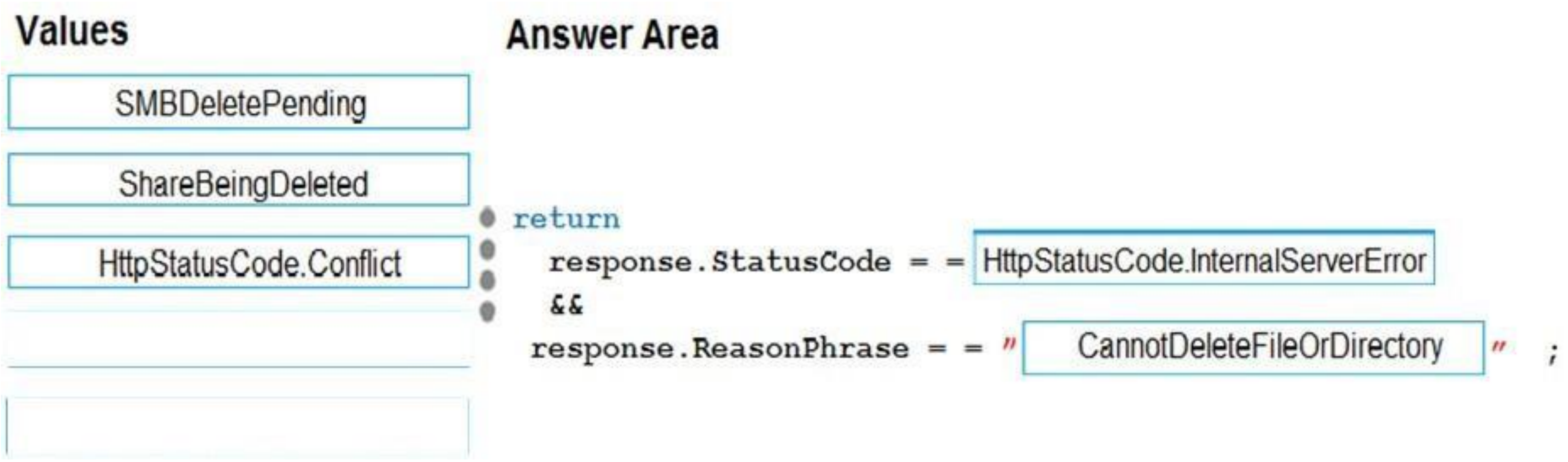
CannotDeleteFileOrDirectory

HttpStatusCode.InternalServerError

Answer Area

```
return
  response.StatusCode = = [          ]
  &&
 response.ReasonPhrase = = "[          ]"  ;
```

Correct Answer:



Values

SMBDeletePending

ShareBeingDeleted

HttpStatusCode.Conflict

Answer Area

```
return
  response.StatusCode = = HttpStatusCode.InternalServerError
  &&
 response.ReasonPhrase = = " CannotDeleteFileOrDirectory "  ;
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: HttpStatusCode.InternalServerError
HttpStatusCode.InternalServerError is equivalent to HTTP status 500. InternalServerError indicates that a generic error has occurred on the server.

Box 2: CannotDeleteFileOrDirectory
HttpResponseMessage.ReasonPhrase Property gets or sets the reason phrase which typically is sent by servers together with the status code.

Scenario: 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.

References: https://docs.microsoft.com/en-us/dotnet/api/system.net.httpstatuscode?redirectedfrom=MSDN&view=netframework-4.7.2

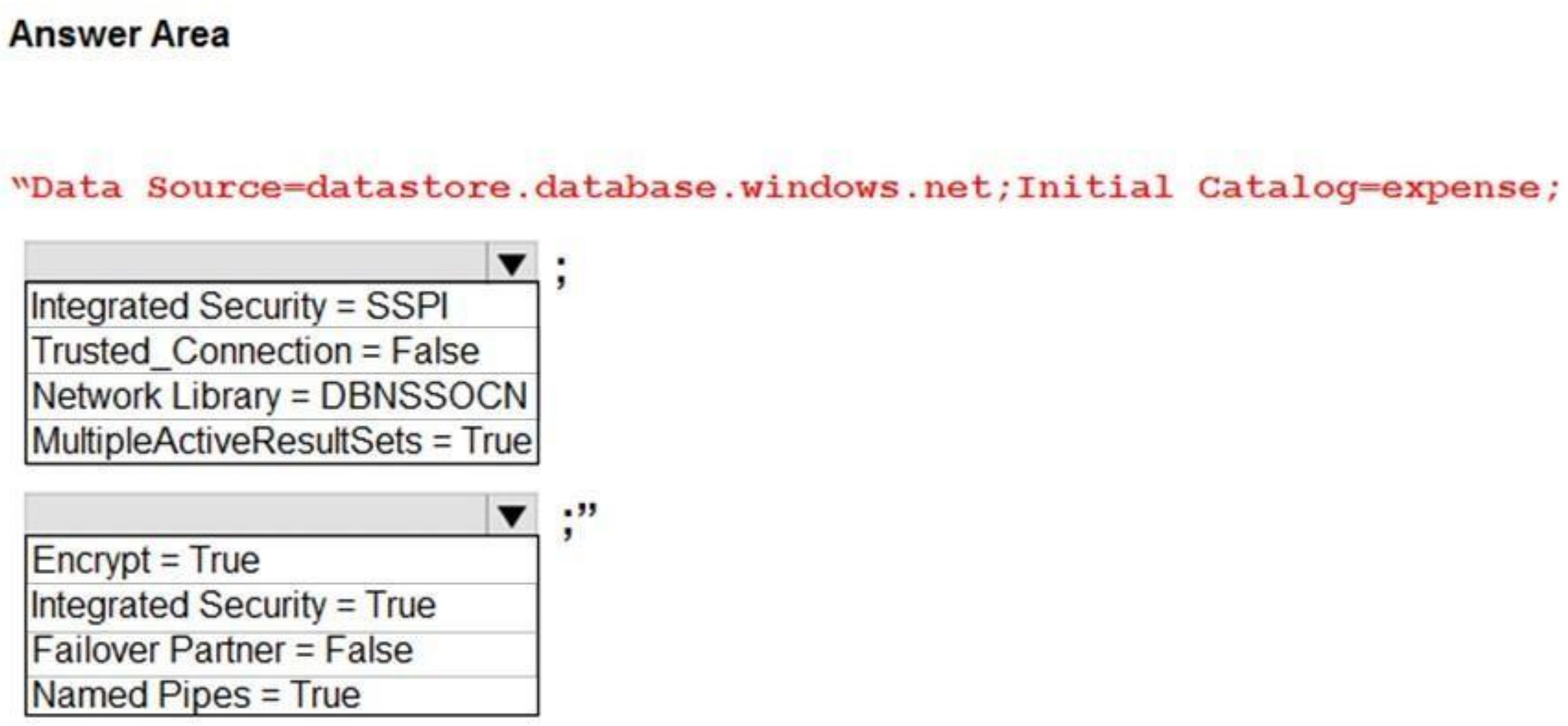
QUESTION 2
HOTSPOT

You need to ensure that security requirements are met.

What value should be used for the ConnectionString field on line DB03 in the Database class? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area

```
“Data Source=datastore.database.windows.net;Initial Catalog=expense;
```

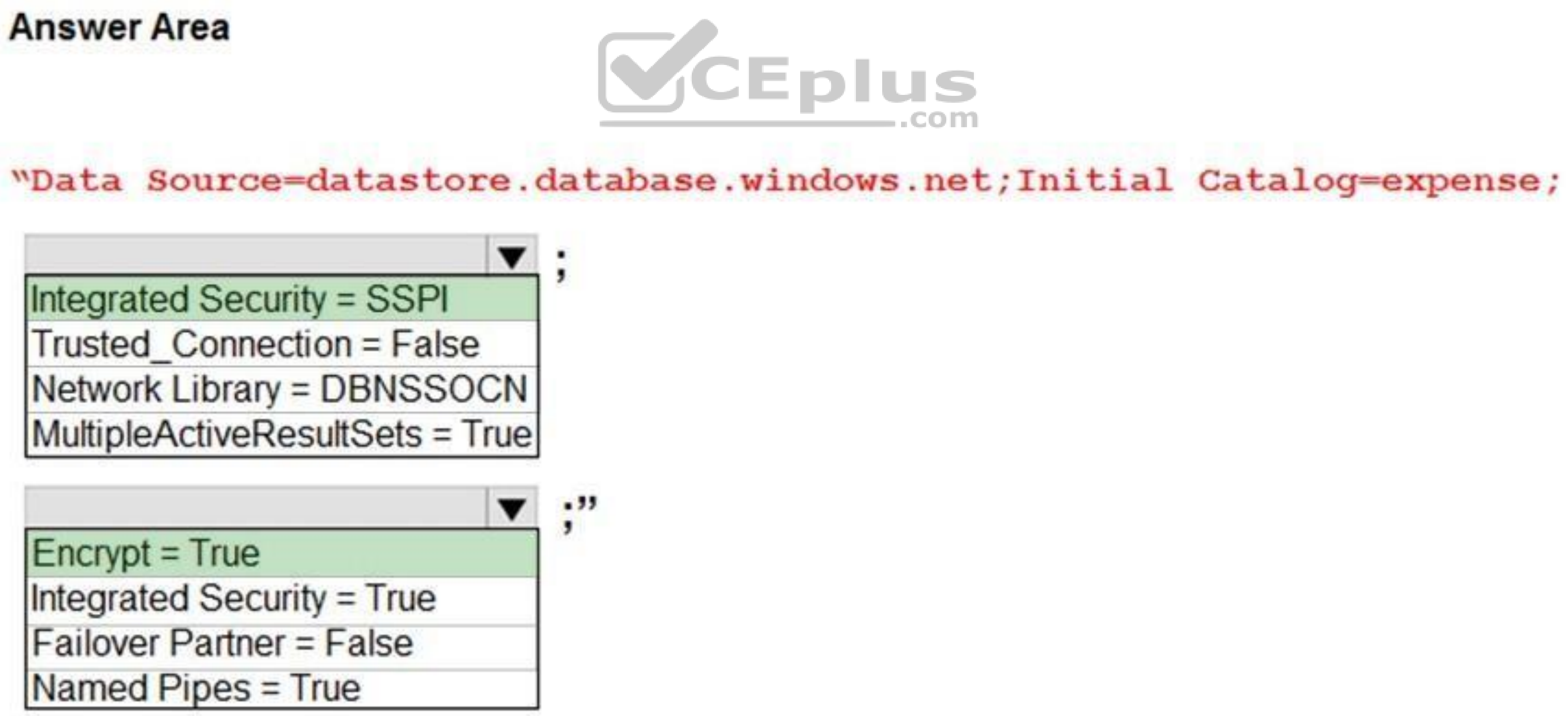
▼ ;

- Integrated Security = SSPI
- Trusted_Connection = False
- Network Library = DBNSSOCN
- MultipleActiveResultSets = True

▼ ;”

- Encrypt = True
- Integrated Security = True
- Failover Partner = False
- Named Pipes = True

Correct Answer:



Answer Area

```
“Data Source=datastore.database.windows.net;Initial Catalog=expense;
```

▼ ;

- [✓ Integrated Security = SSPI]
- Trusted_Connection = False
- Network Library = DBNSSOCN
- MultipleActiveResultSets = True

▼ ;”

- [✓ Encrypt = True]
- Integrated Security = True
- Failover Partner = False
- Named Pipes = True

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Integrated Security=SSPI

Integrated security: For all data source types, connect using the current user account.

For SqlClient you can use Integrated Security=true; or Integrated Security=SSPI;

Scenario: All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)

Box 2: Encrypt = True
Scenario: All data must be protected in transit.

References: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-syntax

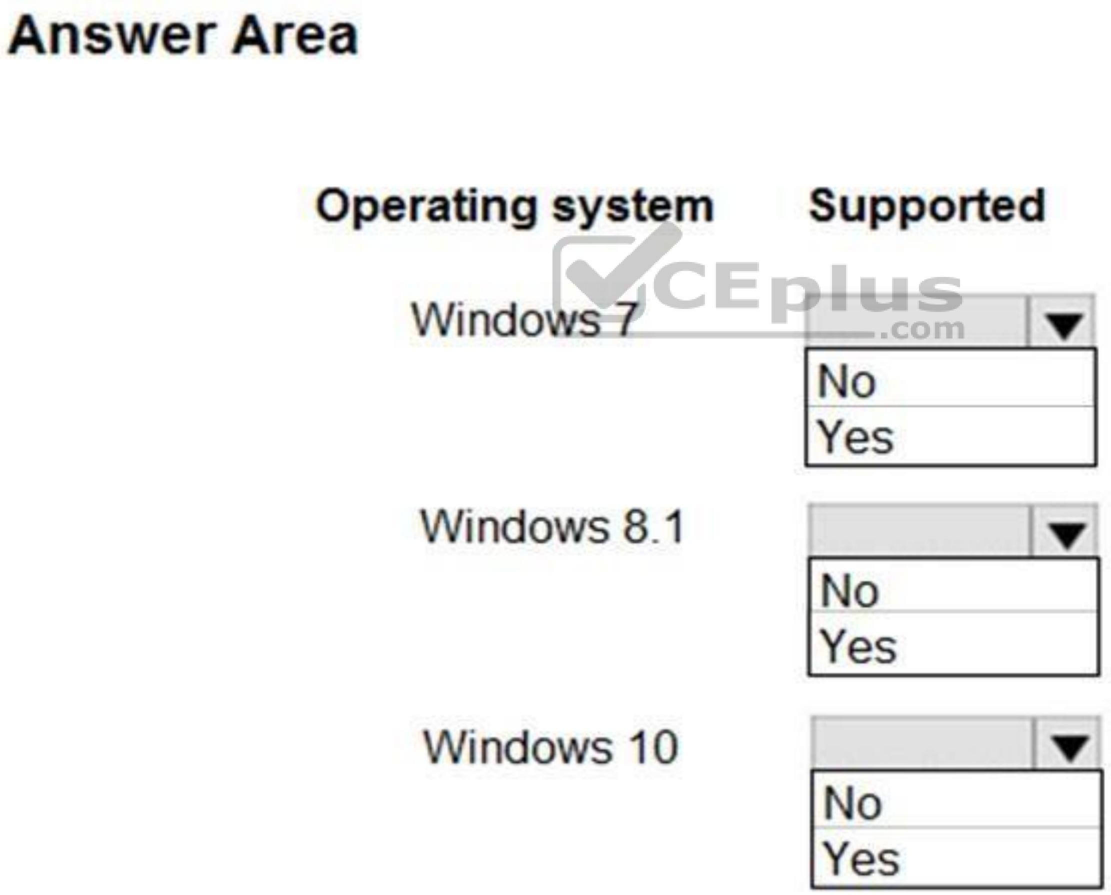
QUESTION 3
HOTSPOT

You need to add the Supporting Operating Systems section to the Getting Started document.

How should you complete the section? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

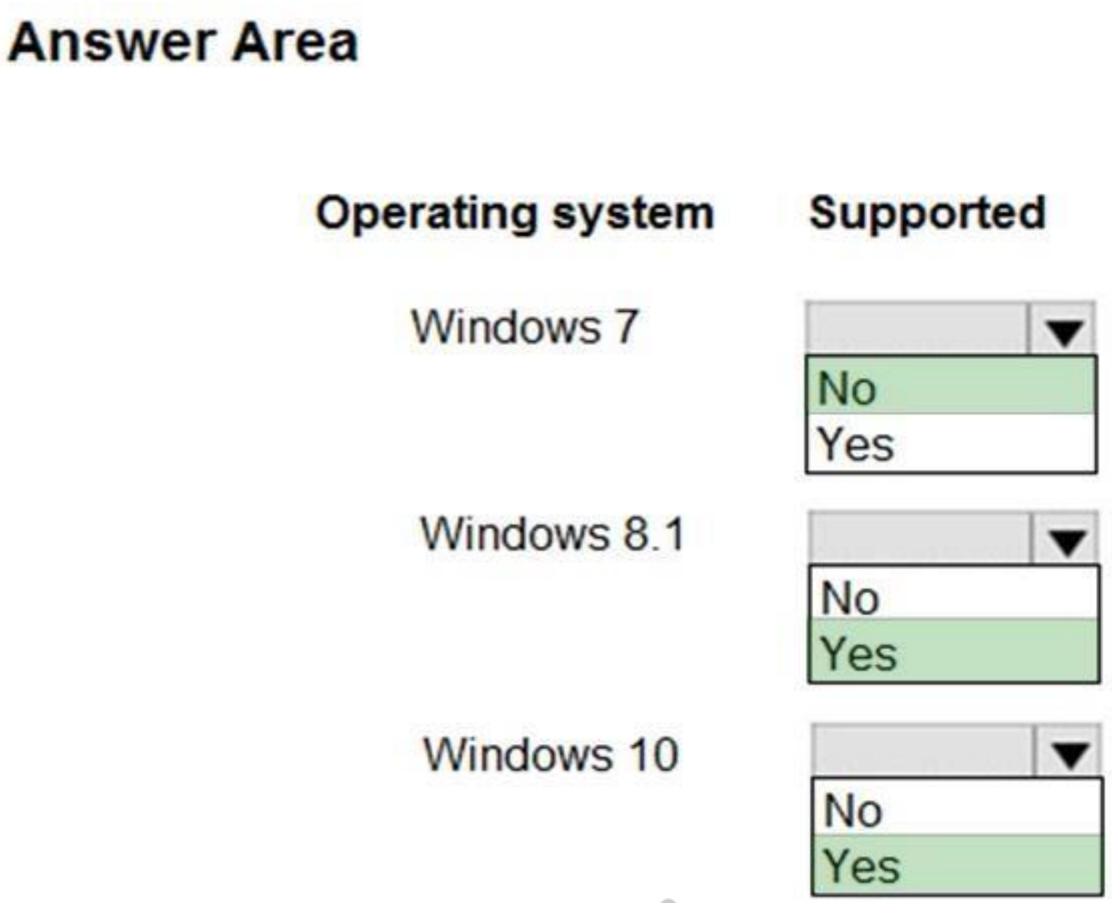
Hot Area:



Answer Area

Operating system	Supported
Windows 7	No Yes
Windows 8.1	No Yes
Windows 10	No Yes

Correct Answer:

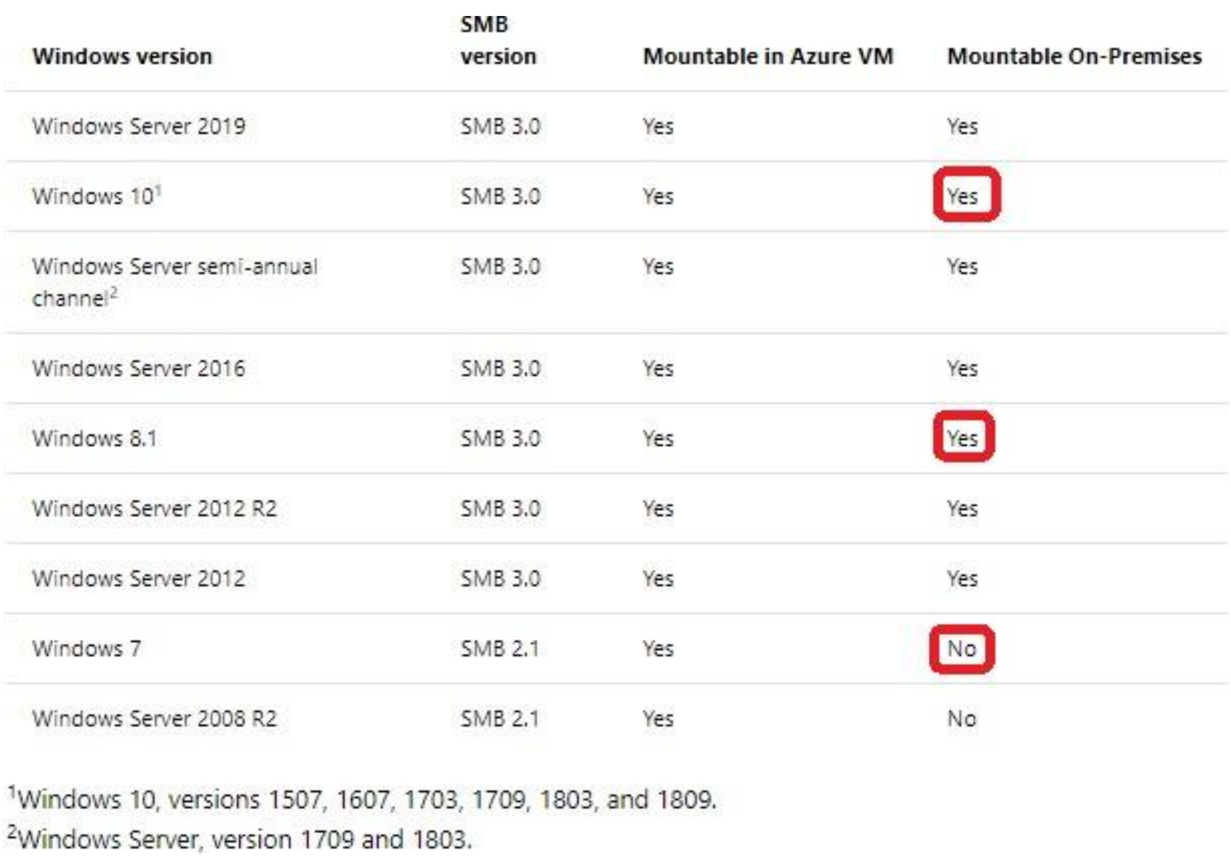


Answer Area

Operating system	Supported
Windows 7	[✓ No] Yes
Windows 8.1	No [✓ Yes]
Windows 10	No [✓ Yes]

Section: [none]
Explanation

Explanation/Reference:
Explanation:



Windows version	SMB version	Mountable in Azure VM	Mountable On-Premises
Windows Server 2019	SMB 3.0	Yes	Yes
Windows 10[1]	SMB 3.0	Yes	Yes
Windows Server semi-annual channel[2]	SMB 3.0	Yes	Yes
Windows Server 2016	SMB 3.0	Yes	Yes
Windows 8.1	SMB 3.0	Yes	Yes
Windows Server 2012 R2	SMB 3.0	Yes	Yes
Windows Server 2012	SMB 3.0	Yes	Yes
Windows 7	SMB 2.1	Yes	No
Windows Server 2008 R2	SMB 2.1	Yes	No

[1] Windows 10, versions 1507, 1607, 1703, 1709, 1803, and 1809.

[2] Windows Server, version 1709 and 1803.

Scenario: 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.

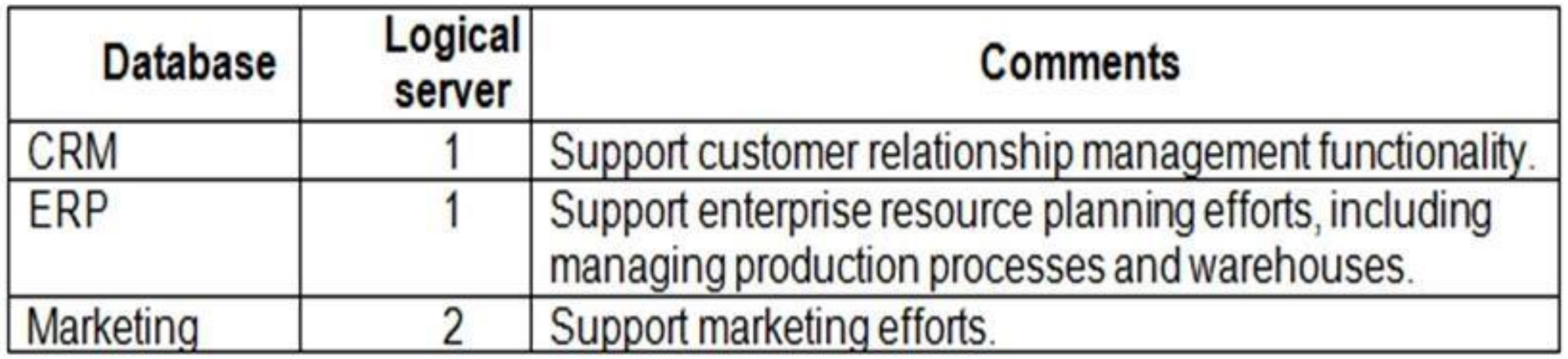
You can use Azure file shares on a Windows installation that is running either in an Azure VM or on-premises. The following table illustrates which OS versions support accessing file shares in which environment:

References:
https://docs.microsoft.com/en-us/azure/storage/files/storage-how-to-use-files-windows

Question Set 2

QUESTION 1
HOTSPOT

A company is developing a software as a service (SaaS) solution in Azure for other business to manage customers. The solution includes the following Azure SQL Database instances. Customer data exists in all databases.



Database	Logical server	Comments
CRM	1	Support customer relationship management functionality.
ERP	1	Support enterprise resource planning efforts, including managing production processes and warehouses.
Marketing	2	Support marketing efforts.

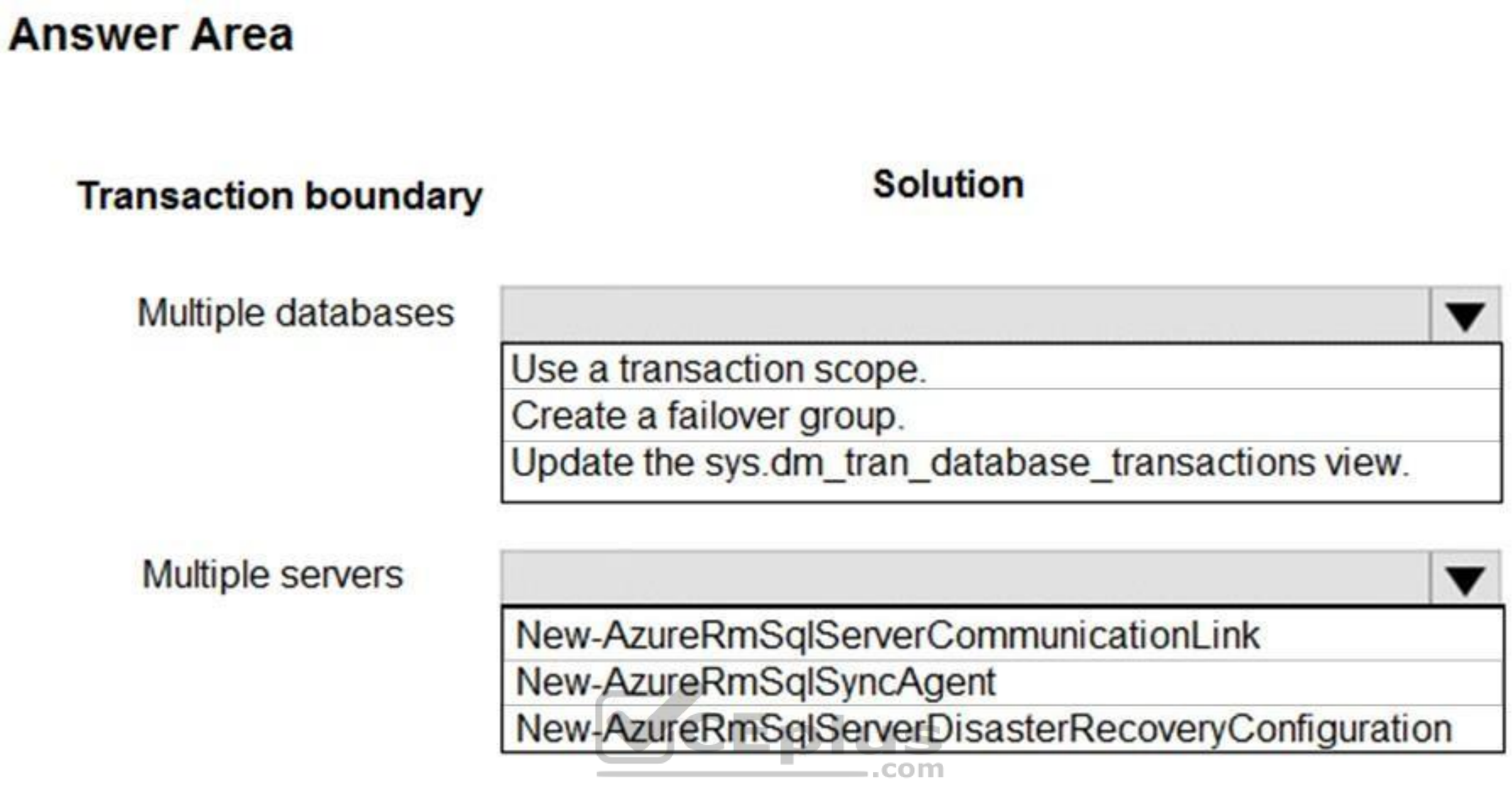
Customers may request that the company remove data that relates to the customer. If a customer makes a deletion request, all details from the customer must be removed from the solution. All deletions for a customer must occur in one distributed transaction that includes the CRM, ERP, and Marketing databases.

You need to activate support for transactions over multiple databases on the same logical server and activate support for transactions over multiple servers.

Which actions and Azure PowerShell command 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

Transaction boundary | Solution

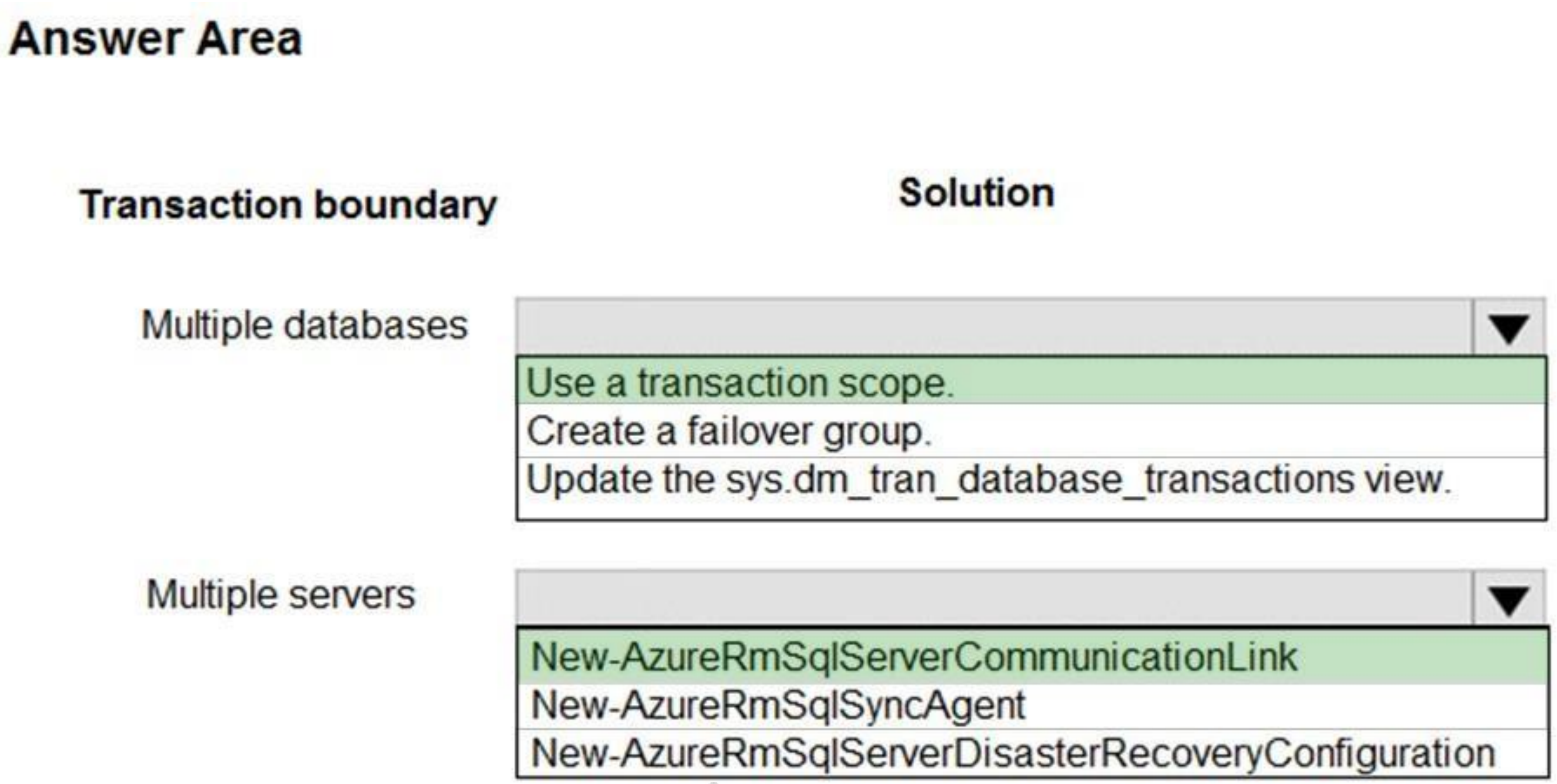
Multiple databases

- Use a transaction scope.
- Create a failover group.
- Update the sys.dm_tran_database_transactions view.

Multiple servers

- New-AzureRmSqlServerCommunicationLink
- New-AzureRmSqlSyncAgent
- New-AzureRmSqlServerDisasterRecoveryConfiguration

Correct Answer:



Answer Area

Transaction boundary	Solution
Multiple databases	Use a transaction scope. Create a failover group. Update the sys.dm_tran_database_transactions view.
Multiple servers	New-AzureRmSqlServerCommunicationLink New-AzureRmSqlSyncAgent New-AzureRmSqlServerDisasterRecoveryConfiguration

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Use a transaction scope
The TransactionScope class establishes an ambient transaction in .NET. (An “ambient transaction” is one that lives in the current thread.) All connections opened within the TransactionScope participate in the transaction. If different databases participate, the transaction is automatically elevated to a distributed transaction.

Box 2: New-AzureRmSqlServerCommunicationLink
New-AzureRmSqlServerCommunicationLink: Use this cmdlet to create a new communication relationship between two logical servers in Azure SQL DB. The relationship is symmetric which means both servers can initiate transactions with the other server.

References: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-transactions-overview

QUESTION 2
HOTSPOT

A construction company creates three-dimensional models from photographs and design diagrams of buildings. The company plans to store high-resolution photographs and blueprint files in Azure Blob Storage. The files are currently stored in the construction company’s office.

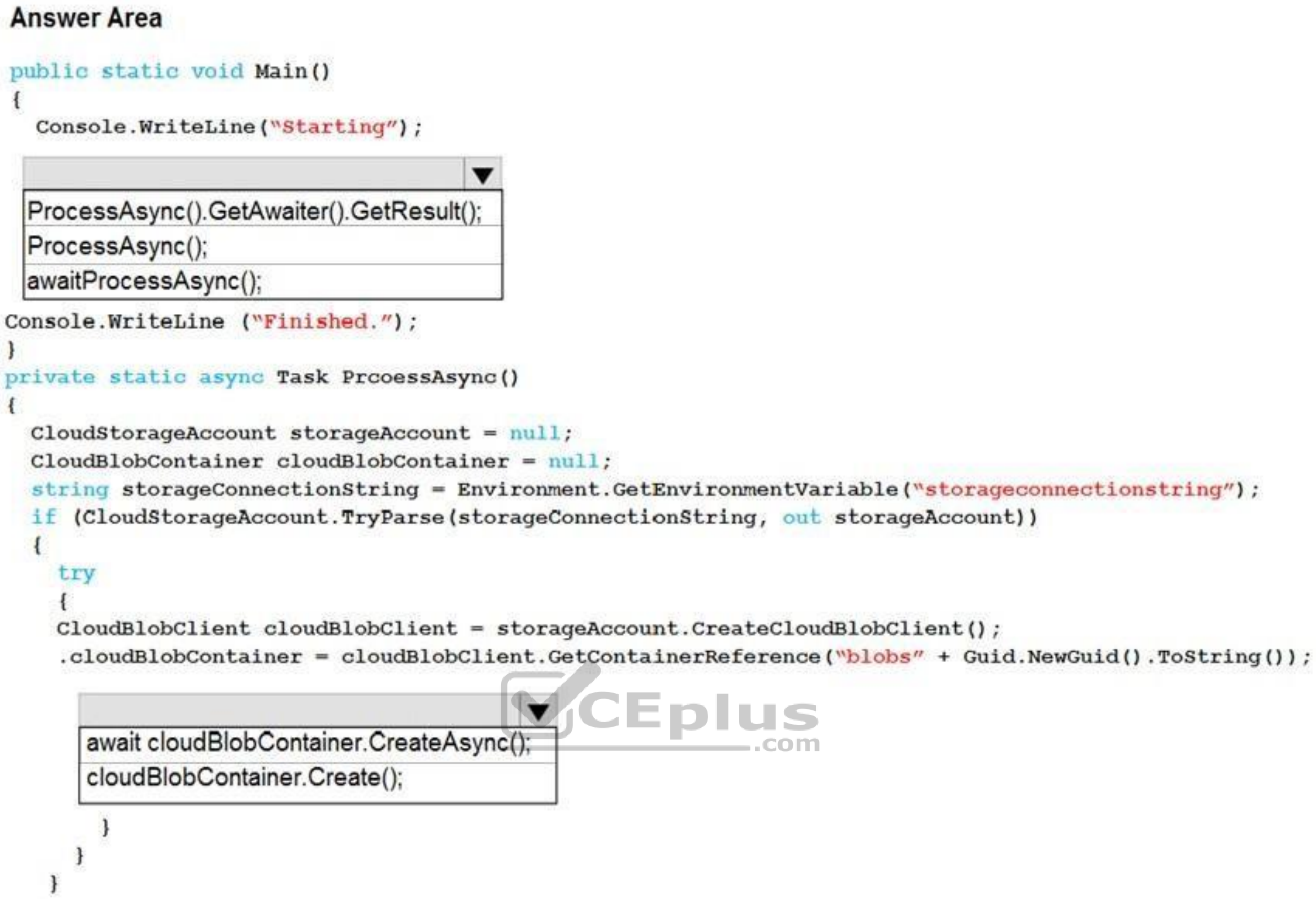
You are developing a tool to connect to Azure Storage, create container, and then upload the files. The tool must remain responsive to the end user while it is running and performing remote I/O operations. It must also wait for methods to complete before continuing.

You need to complete the configuration.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct solution is worth one point.

Hot Area:



Answer Area

```
public static void Main()
{
  Console.WriteLine("Starting");
```

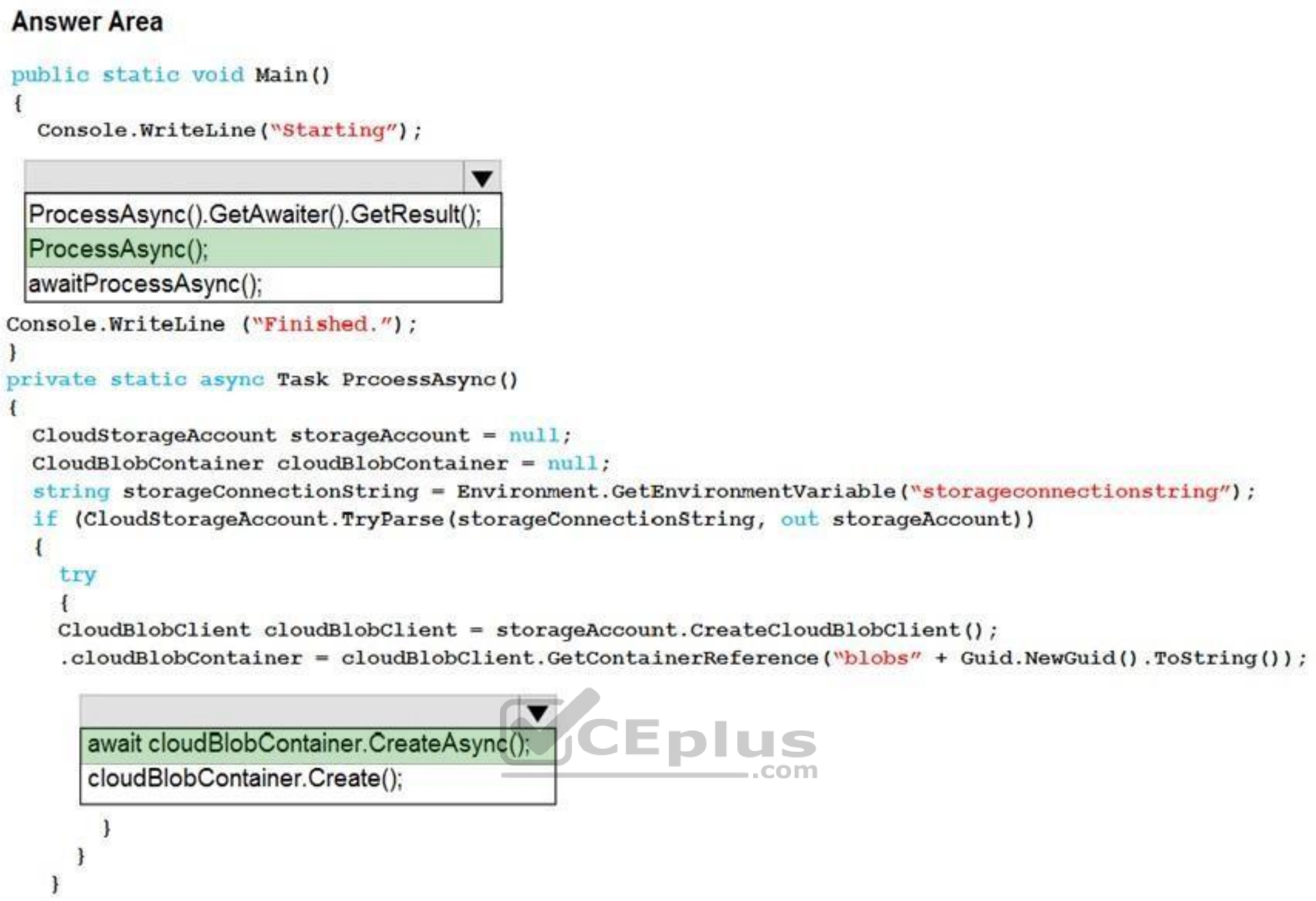
```
ProcessAsync().GetAwaiter().GetResult();
ProcessAsync();
awaitProcessAsync();
```

```
Console.WriteLine ("Finished.");
}
private static async Task PrcoessAsync()
{
  CloudStorageAccount storageAccount = null;
  CloudBlobContainer cloudBlobContainer = null;
  string storageConnectionString = Environment.GetEnvironmentVariable("storageconnectionstring");
  if (CloudStorageAccount.TryParse(storageConnectionString, out storageAccount))
  {
    try
    {
    CloudBlobClient cloudBlobClient = storageAccount.CreateCloudBlobClient();
    .cloudBlobContainer = cloudBlobClient.GetContainerReference("blobs" + Guid.NewGuid().ToString());
```

```
await cloudBlobContainer.CreateAsync();
cloudBlobContainer.Create();
```

```
      }
    }
  }
```

Correct Answer:



Answer Area

```
public static void Main()
{
  Console.WriteLine("Starting");
```

- ProcessAsync().GetAwaiter().GetResult();
- [✓ ProcessAsync();]
- awaitProcessAsync();

```
Console.WriteLine ("Finished.");
}
private static async Task PrcoessAsync()
{
  CloudStorageAccount storageAccount = null;
  CloudBlobContainer cloudBlobContainer = null;
  string storageConnectionString = Environment.GetEnvironmentVariable("storageconnectionstring");
  if (CloudStorageAccount.TryParse(storageConnectionString, out storageAccount))
  {
    try
    {
    CloudBlobClient cloudBlobClient = storageAccount.CreateCloudBlobClient();
    .cloudBlobContainer = cloudBlobClient.GetContainerReference("blobs" + Guid.NewGuid().ToString());
```

- [✓ await cloudBlobContainer.CreateAsync();]
- cloudBlobContainer.Create();

```
      }
    }
  }
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: ProcessAsync();

Box 2: await cloudBlobContainer.CreatAsync();
If you specify that a method is an async method by using the async modifier, you enable the following two capabilities.

The marked async method can use await to designate suspension points. The await operator tells the compiler that the async method can't continue past that point until the awaited asynchronous process is complete. In the meantime, control returns to the caller of the async method.

The suspension of an async method at an await expression doesn't constitute an exit from the method, and finally blocks don't run.

The marked async method can itself be awaited by methods that call it.

References: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/

QUESTION 3
DRAG DROP

You are creating a collaborative image hosting platform as an ASP.NET MVC web application. Users add, update, and modify images on the platform. Images are stored in Azure Blob storage.

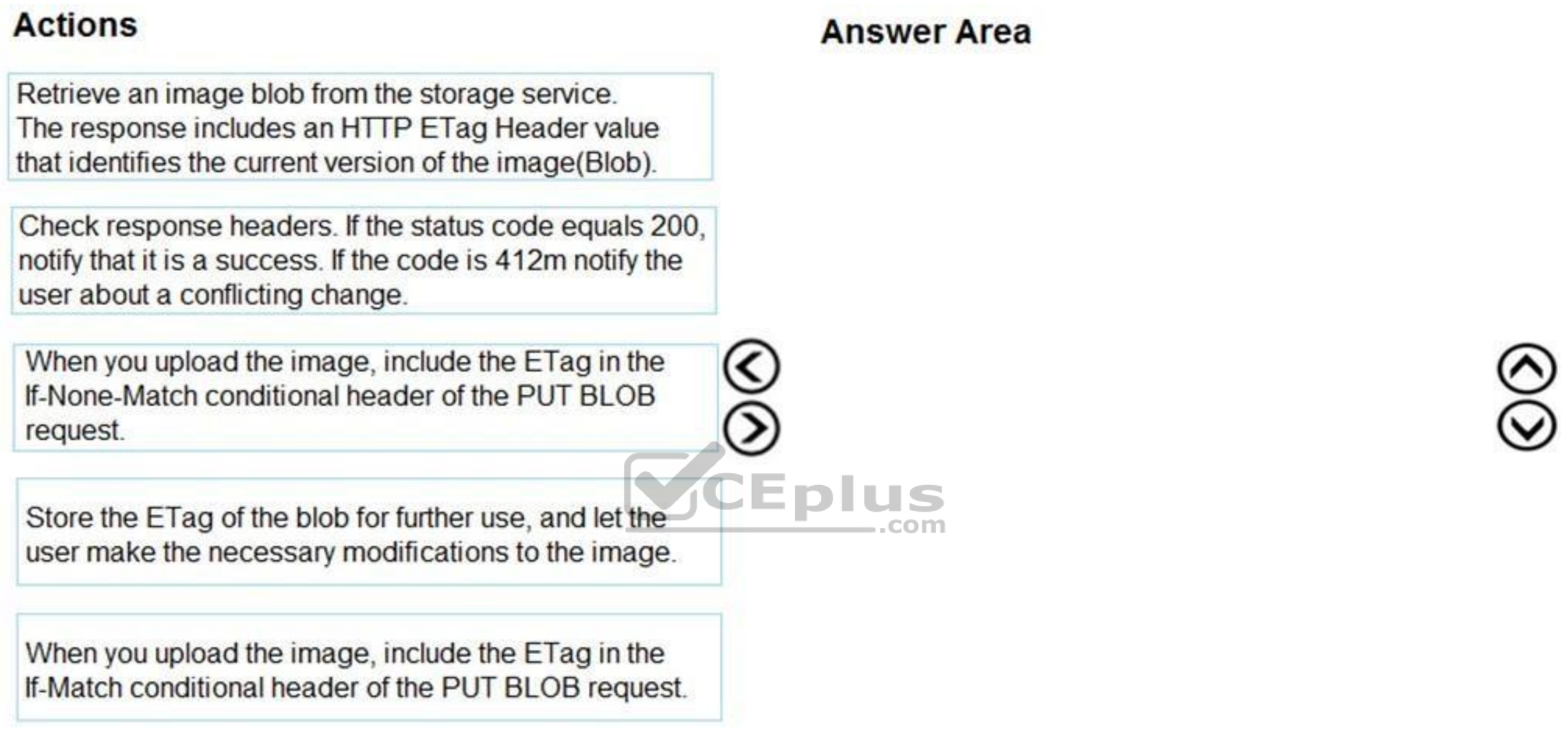
More than one user at a time must be able to modify the same image.

You need to implement optimistic concurrency for uploading images.

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

NOTE: Each correct solution is worth one point.

Select and Place:



Actions

Retrieve an image blob from the storage service. The response includes an HTTP ETag Header value that identifies the current version of the image(Blob).

Check response headers. If the status code equals 200, notify that it is a success. If the code is 412m notify the user about a conflicting change.

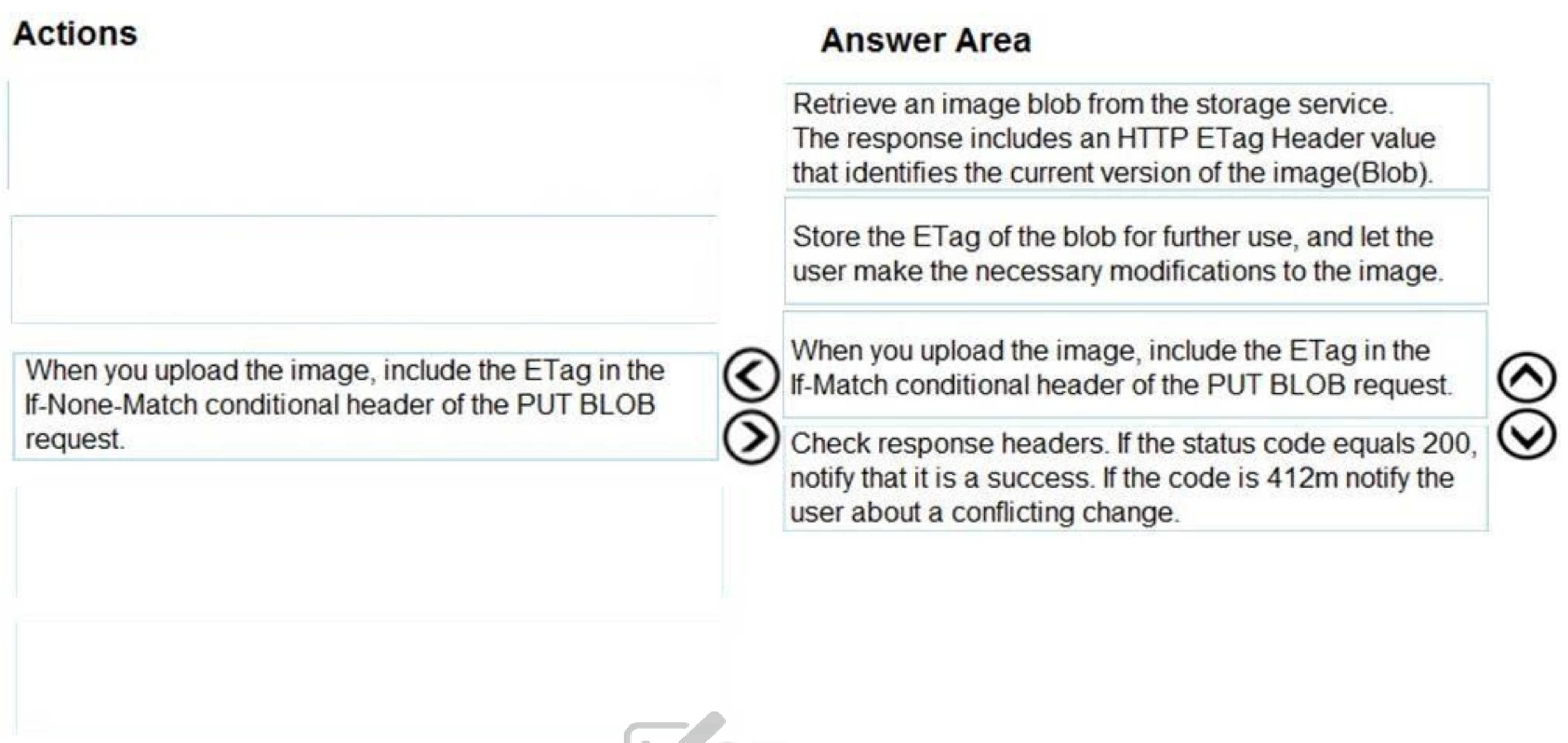
When you upload the image, include the ETag in the If-None-Match conditional header of the PUT BLOB request.

Store the ETag of the blob for further use, and let the user make the necessary modifications to the image.

When you upload the image, include the ETag in the If-Match conditional header of the PUT BLOB request.

Answer Area

Correct Answer:



Actions

When you upload the image, include the ETag in the If-None-Match conditional header of the PUT BLOB request.

Answer Area

Retrieve an image blob from the storage service. The response includes an HTTP ETag Header value that identifies the current version of the image(Blob).

Store the ETag of the blob for further use, and let the user make the necessary modifications to the image.

When you upload the image, include the ETag in the If-Match conditional header of the PUT BLOB request.

Check response headers. If the status code equals 200, notify that it is a success. If the code is 412m notify the user about a conflicting change.

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Optimistic concurrency means allowing concurrency conflicts to happen, and then reacting appropriately if they do.

For the PUT method, If-Match can be used to prevent the lost update problem. It can check if the modification of a resource that the user wants to upload will not override another change that has been done since the original resource was fetched. If the request cannot be fulfilled, the 412 (Precondition Failed) response is returned.

References: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/If-Match

QUESTION 4
Members of the finance department for a company review and make changes to a Microsoft Excel workbook that is hosted on OneDrive. The workbook contains projected costs and revenue for a project.

You need to develop an Azure Function that ingests data from the modified workbook and places it into a Microsoft Word document.

Which two objects should you implement? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. A group conversation subscription
B. An Excel table input binding
C. A group subscription
D. An Excel table output binding
E. An auth token input binding

Correct Answer: BC
Section: [none]

Explanation

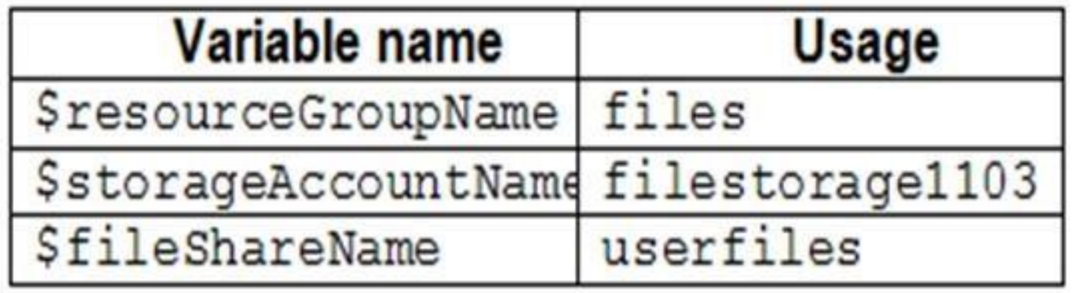
Explanation/Reference:
Explanation:
B: Azure Functions supports trigger, input, and output bindings for external files. These bindings create API connections to SaaS providers, or use existing API connections from your Function App's resource group.

References: https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-external-file

QUESTION 5
DRAG DROP

You have an Azure subscription.

You must create a file share with a quota of 2,048 GB. You create the following variables:



Variable name	Usage
$resourceGroupName	files
$storageAccountName	filestorage1103
$fileShareName	userfiles

In which order should you arrange the Azure CLI commands to develop the solution? To answer, move all the commands from the list of commands to the answer area and arrange them in the correct order.

Select and Place:

Correct Answer:

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Step 1:
Create a resource group
A resource group is a logical container in which Azure resources are deployed and managed. If you don't already have an Azure resource group, you can use the az group create command to create one.

The following example creates a resource group named myResourceGroup in the East US location: az

group create --name myResourceGroup --location eastus

Step 2:
Create a storage account
The following example creates a storage account named mystorageaccount<random number> by using the az storage account create command, and then puts the name of that storage account in the $STORAGEACCT variable.

```
STORAGEACCT=$(az storage account create \
 --resource-group "myResourceGroup" \
 --name "mystorageacct$RANDOM" \
 --location eastus \
 --sku Standard_LRS \  --
query "name" | tr -d '"')
```

Step 3:
Get the storage account key
Storage account keys control access to resources in a storage account. The keys are automatically created when you create a storage account. You can get the storage account keys for your storage account by using the az storage account keys list command:

```
STORAGEKEY=$(az storage account keys list \
 --resource-group "myResourceGroup" \
 --account-name $STORAGEACCT \
 --query "[0].value" | tr -d '"')
```

Step 4:
Now, you can create your Azure file share. Create file shares by using the az storage share create command. This example creates an Azure file share named myshare: az storage share create \
 --account-name $STORAGEACCT \
 --account-key $STORAGEKEY \
 --name "myshare"

References: https://docs.microsoft.com/en-us/azure/storage/files/storage-how-to-use-files-cli

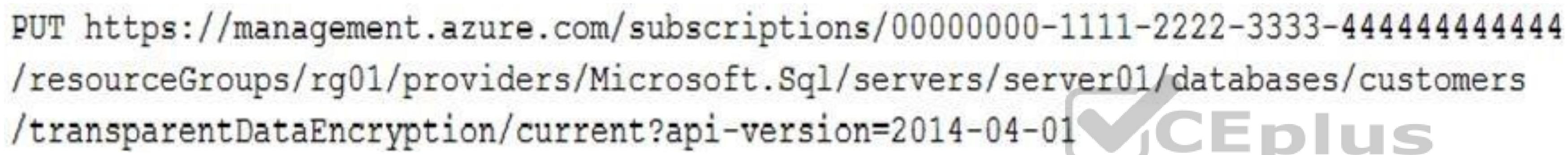
QUESTION 6
A company uses Azure SQL Database to store data for an app. The data includes sensitive information.

You need to implement measures that allow only members of the managers group to see sensitive information.

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

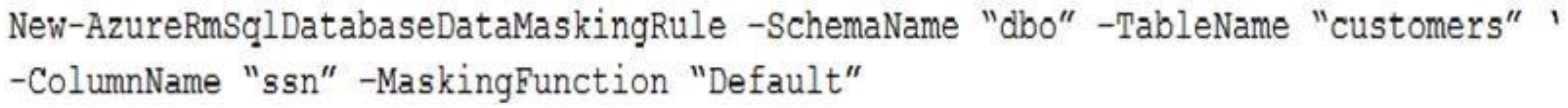
NOTE: Each correct selection is worth one point.

A. Navigate to the following URL:



```
PUT https://management.azure.com/subscriptions/00000000-1111-2222-3333-444444444444
/resourceGroups/rg01/providers/Microsoft.Sql/servers/server01/databases/customers
/transparentDataEncryption/current?api-version=2014-04-01
```

B. Exclude the managers group.
C. Include the managers group.
D. Run the following Azure PowerShell command:



```
New-AzureRmSqlDatabaseDataMaskingRule -SchemaName “dbo” -TableName “customers” `
-ColumnName “ssn” -MaskingFunction “Default”
```

E. Exclude the administrators group.

Correct Answer: BD
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Dynamic data masking helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal impact on the application layer.

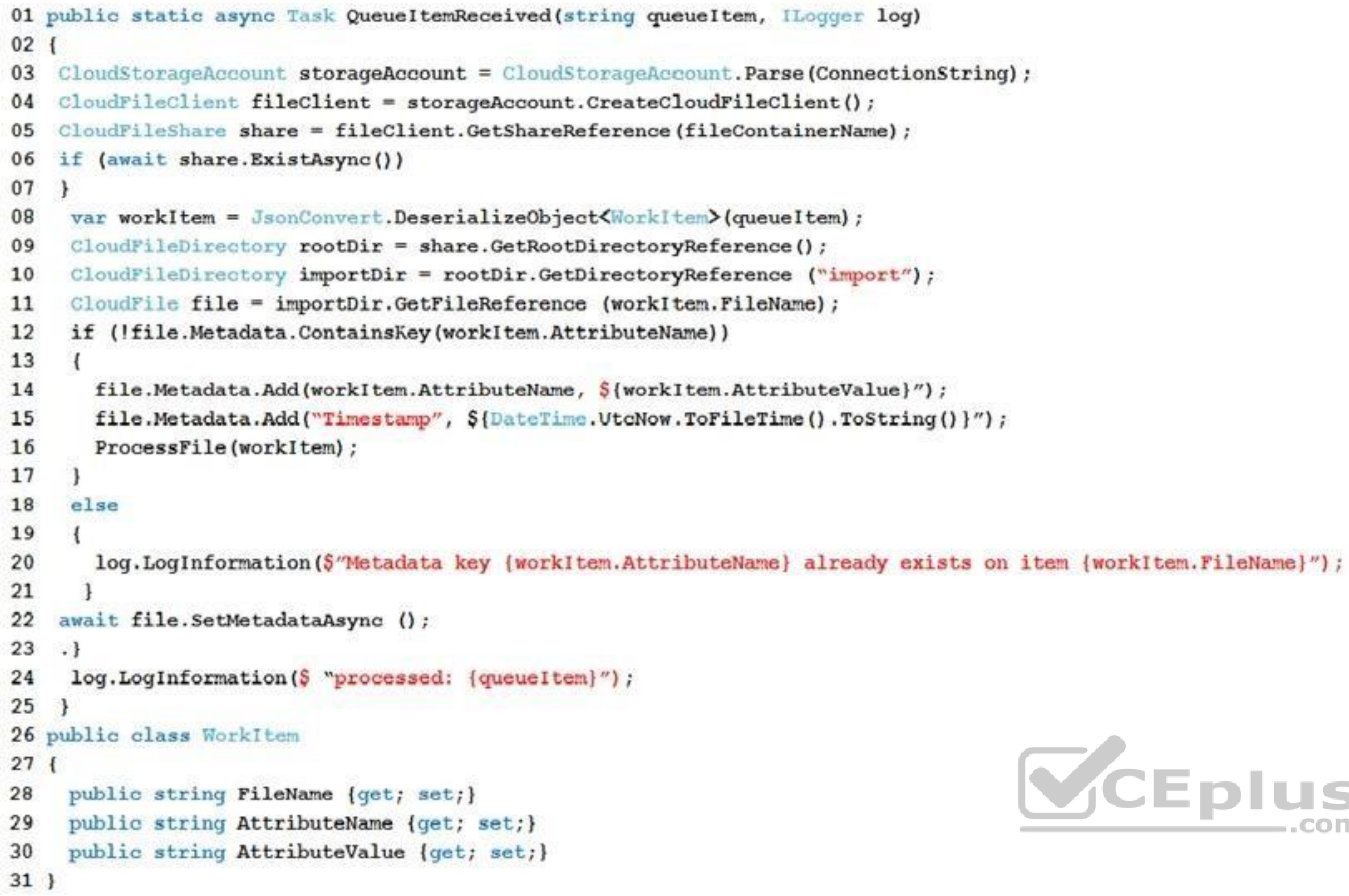
SQL users excluded from masking - A set of SQL users or AAD identities that get unmasked data in the SQL query results.

Note: The New-AzureRmSqlDatabaseDataMaskingRule cmdlet creates a data masking rule for an Azure SQL database.

References: https://docs.microsoft.com/en-us/powershell/module/azurerm.sql/new-azurermsqldatabasedatamaskingrule?view=azurermps-6.13.0

QUESTION 7
HOTSPOT

You are reviewing the following code for an Azure Function. The code is called each time an item is added to a queue. The queue item is a JSON string that deserializes into a class named WorkItem. (Line numbers are included for reference only.)



```
public static async Task QueueItemReceived(string queueItem, ILogger log)
{
 CloudStorageAccount storageAccount = CloudStorageAccount.Parse(ConnectionString);
 CloudFileClient fileClient = storageAccount.CreateCloudFileClient();
 CloudFileShare share = fileClient.GetShareReference(fileContainerName);
 if (await share.ExistAsync())
 }
  var workItem = JsonConvert.DeserializeObject<WorkItem>(queueItem);
  CloudFileDirectory rootDir = share.GetRootDirectoryReference();
  CloudFileDirectory importDir = rootDir.GetDirectoryReference ("import");
  CloudFile file = importDir.GetFileReference (workItem.FileName);
  if (!file.Metadata.ContainsKey(workItem.AttributeName))
  {
    file.Metadata.Add(workItem.AttributeName, ${workItem.AttributeValue}");
    file.Metadata.Add("Timestamp", ${DateTime.UtcNow.ToFileTime().ToString()}");
    ProcessFile(workItem);
  }
  else
  {
    log.LogInformation($"Metadata key {workItem.AttributeName} already exists on item {workItem.FileName}");
   }
 await file.SetMetadataAsync ();
 .}
  log.LogInformation($ "processed: {queueItem}");
 }
public class WorkItem
{
  public string FileName {get; set;}
  public string AttributeName {get; set;}
  public string AttributeValue {get; set;}
}
```

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

NOTE: Each correct selection is worth one point.

Hot Area:

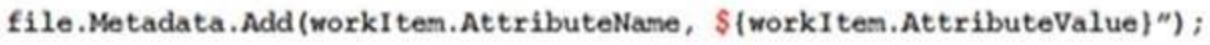
Correct Answer:

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Yes
The following code example sets metadata on a container. One value is set using the collection's Add method.

public static async Task AddContainerMetadataAsync(CloudBlobContainer container)
{
// Add some metadata to the container.
container.Metadata.Add("docType", "textDocuments");
container.Metadata["category"] = "guidance";

Scenario:



```
file.Metadata.Add(workItem.AttributeName, ${workItem.AttributeValue}");
```

Box 2: No
It would not load the metadata

Box 3: Yes

QUESTION 8
You develop a solution that uses an Azure SQL Database to store user information for a mobile app.

The app stores sensitive information about users.

You need to hide sensitive information from developers that query the data for the mobile app.

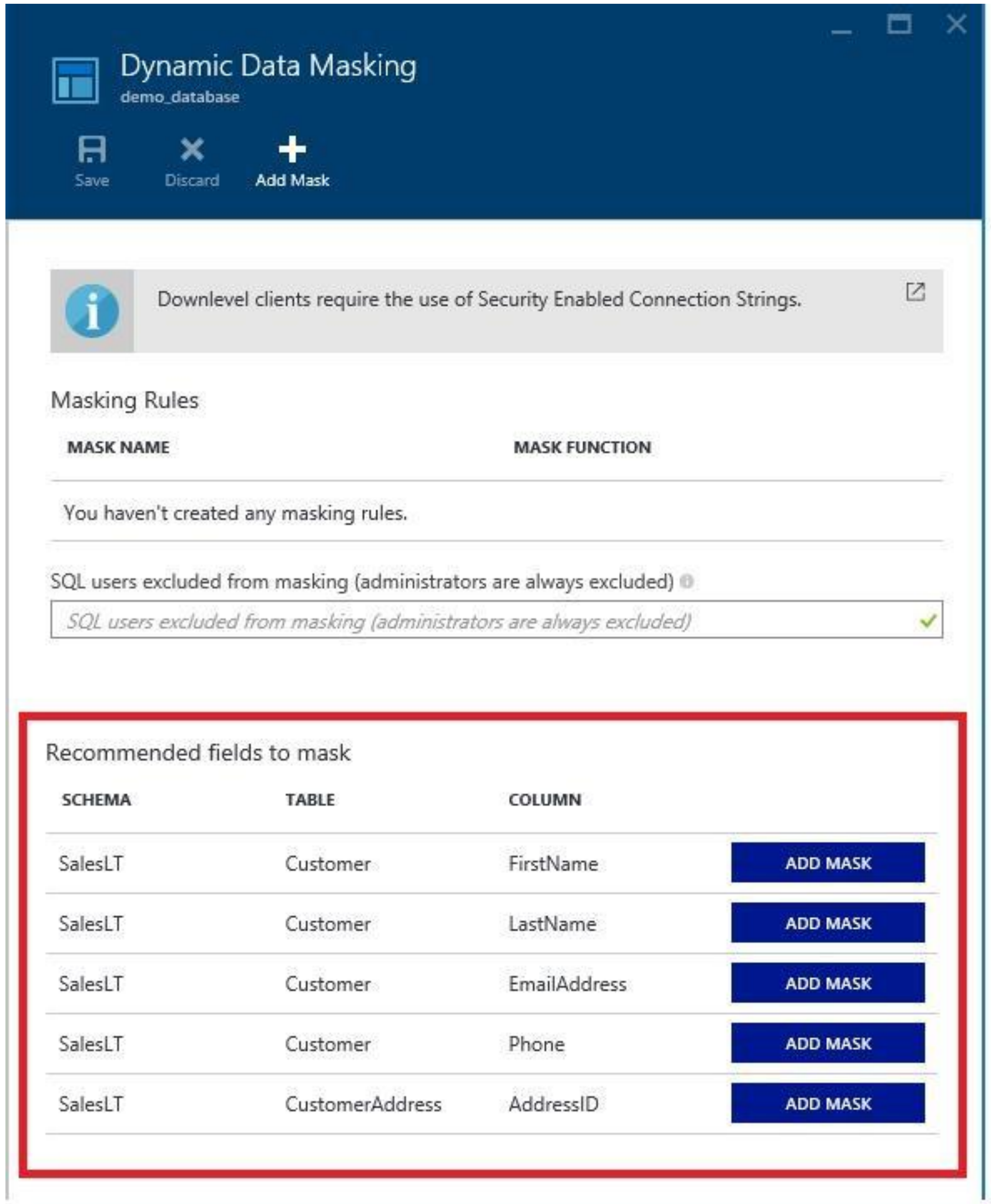
Which three items must you identify when configuring dynamic data masking? Each correct answer presents a part of the solution.

NOTE: Each correct selection is worth one point.

A. Schema
B. Table
C. Index
D. Column
E. Trigger

Correct Answer: ABD
Section: [none]
Explanation

Explanation/Reference:
Explanation:
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.



Dynamic Data Masking
demo_database

Downlevel clients require the use of Security Enabled Connection Strings.

Masking Rules

MASK NAME	MASK FUNCTION
You haven't created any masking rules.	

SQL users excluded from masking (administrators are always excluded)

SQL users excluded from masking (administrators are always excluded)

Recommended fields to mask

SCHEMA	TABLE	COLUMN	
SalesLT	Customer	FirstName	ADD MASK
SalesLT	Customer	LastName	ADD MASK
SalesLT	Customer	EmailAddress	ADD MASK
SalesLT	Customer	Phone	ADD MASK
SalesLT	CustomerAddress	AddressID	ADD MASK

References:
https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dynamic-data-masking-get-started-portal **Testlet 1**

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. 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 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.

Data

You identify the following requirements for data management and manipulation:
- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

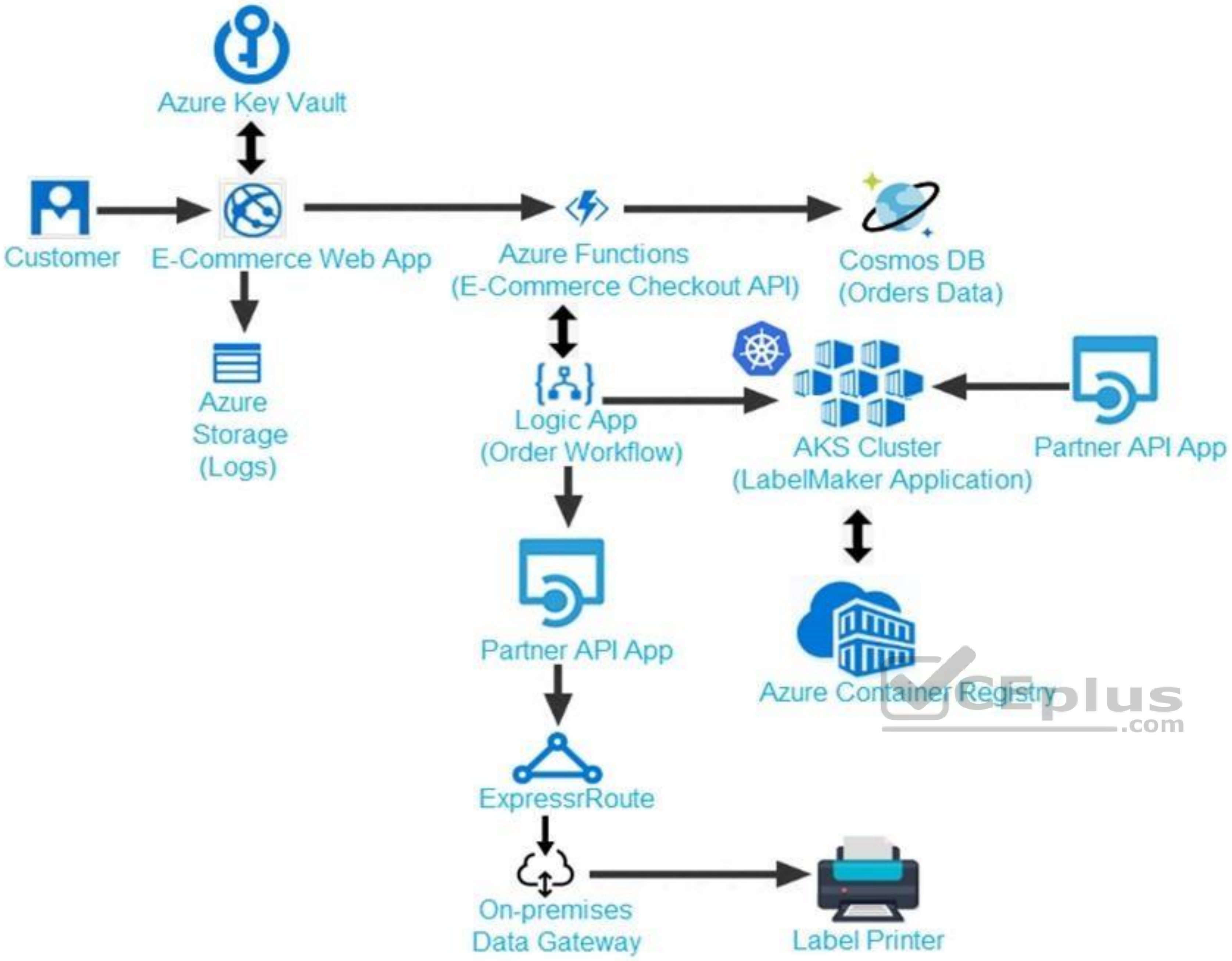
You have the following security requirements:

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

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.



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

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

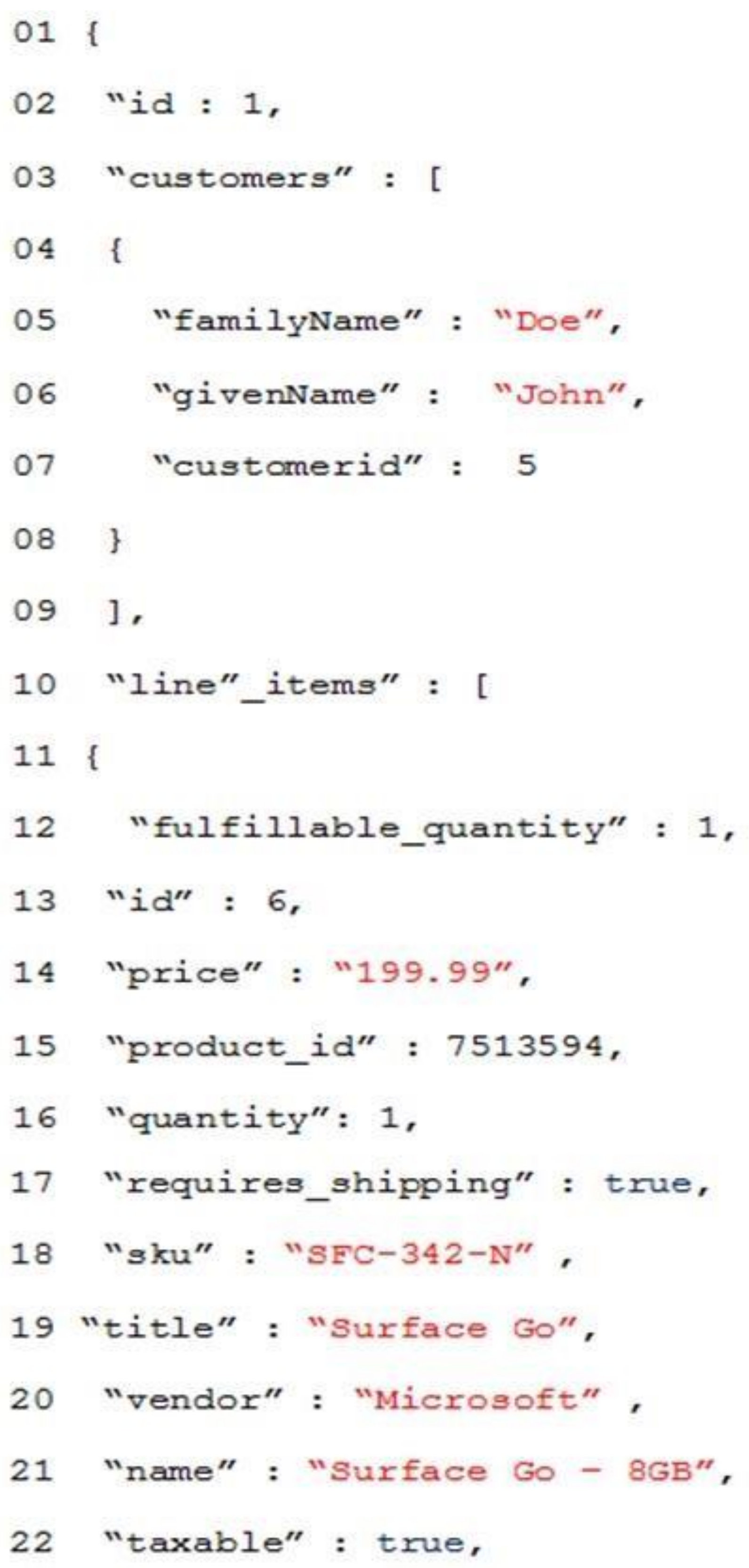
The order workflow fails to run upon initial deployment to Azure.

Order json.

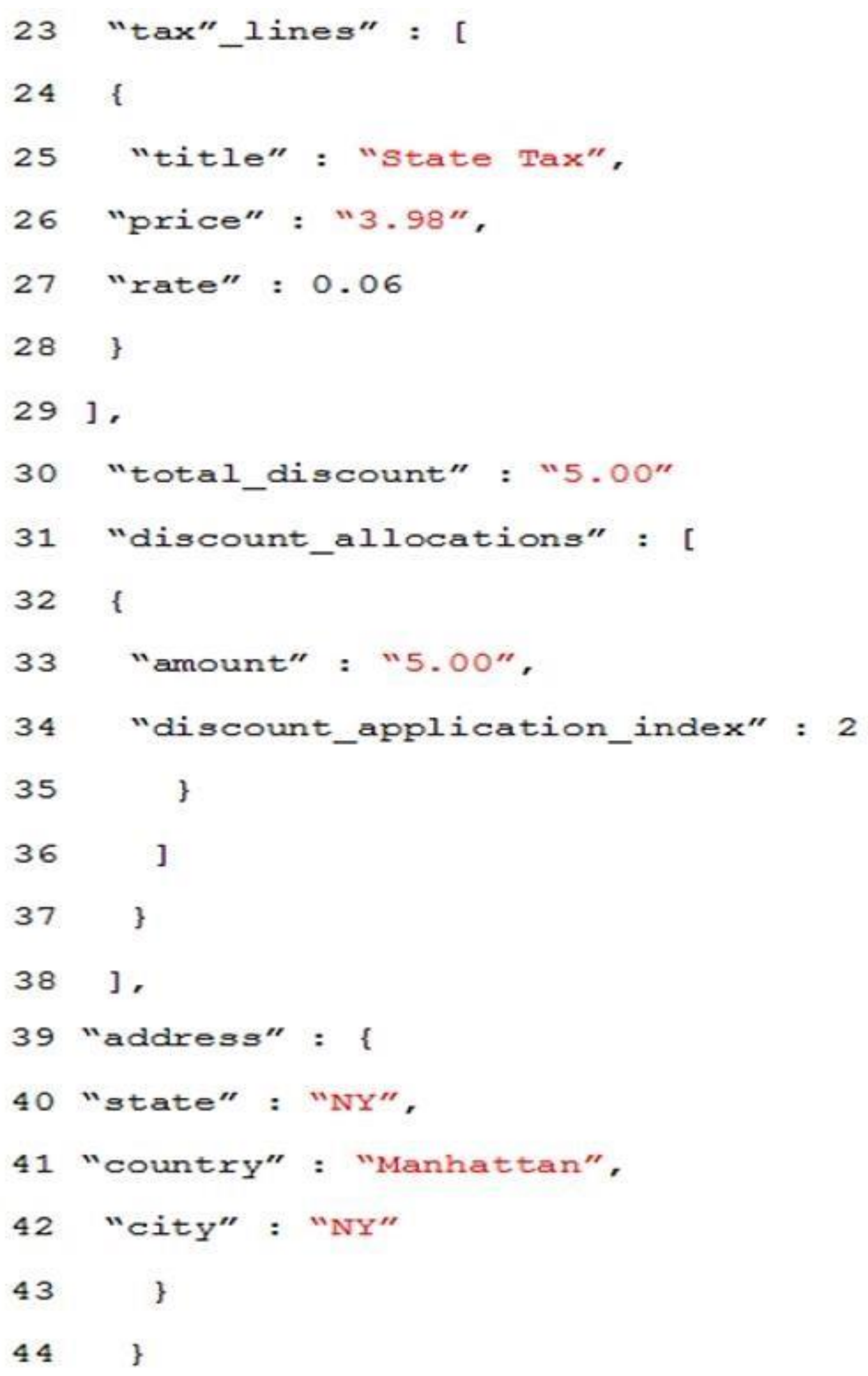
Relevant portions of the app files are shown below. Line numbers are included for reference only.

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

Order .json



```
01 {
02  "id : 1,
03  "customers" : [
04  {
05    "familyName" : "Doe",
06    "givenName" :  "John",
07    "customerid" :  5
08  }
09  ],
10  "line"_items" : [
11 {
12   "fulfillable_quantity" : 1,
13  "id" : 6,
14  "price" : "199.99",
15  "product_id" : 7513594,
16  "quantity": 1,
17  "requires_shipping" : true,
18  "sku" : "SFC-342-N" ,
19 "title" : "Surface Go",
20  "vendor" : "Microsoft" ,
21  "name" : "Surface Go - 8GB",
22  "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",
"country" : "Manhattan",
 "city" : "NY"
   }
  }
```

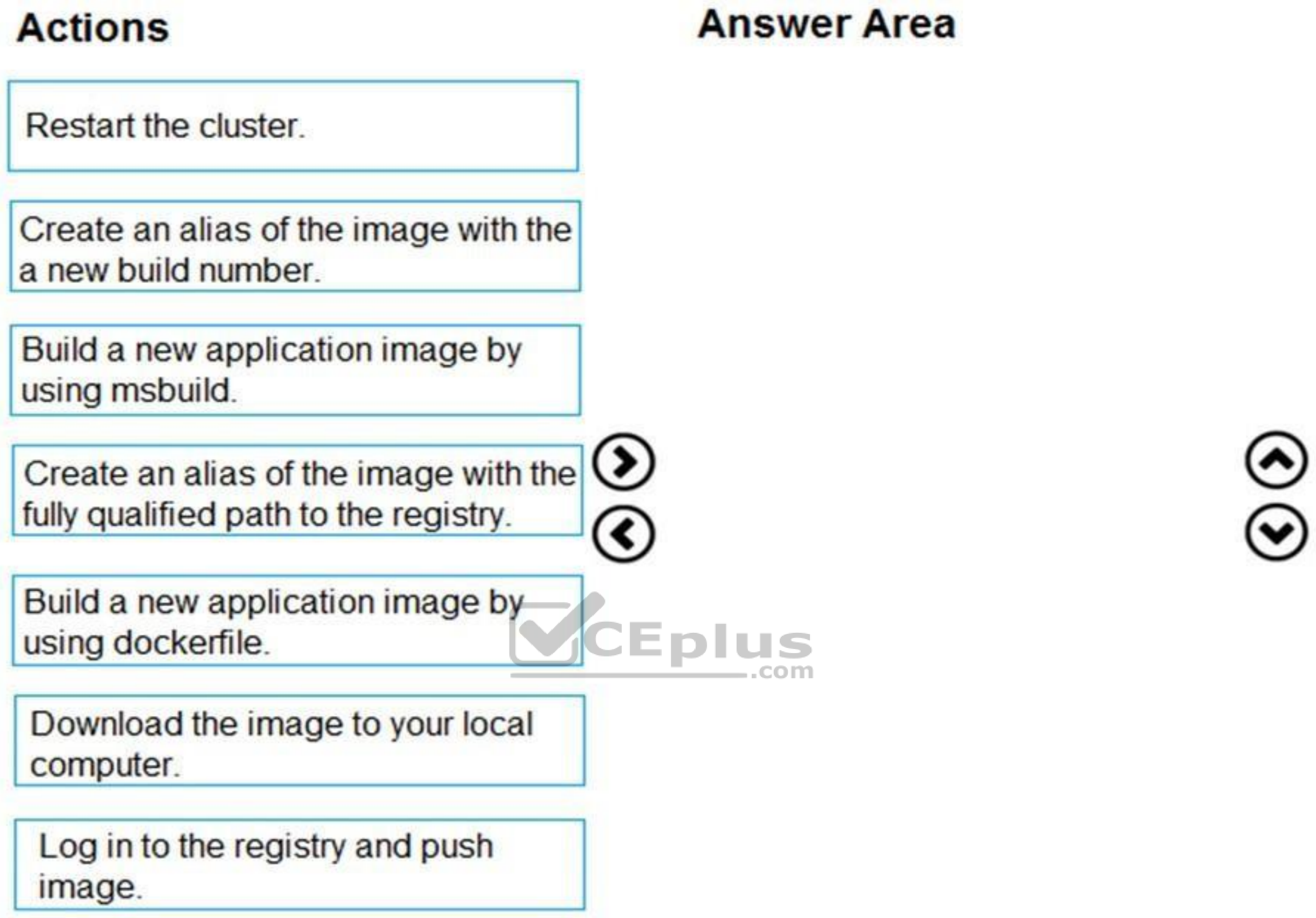
QUESTION 1
DRAG DROP

You need to deploy a new version of the LabelMaker application.

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

NOTE: Each correct selection is worth one point.

Select and Place:



Actions

Answer Area

Restart the cluster.

Create an alias of the image with the a new build number.

Build a new application image by using msbuild.

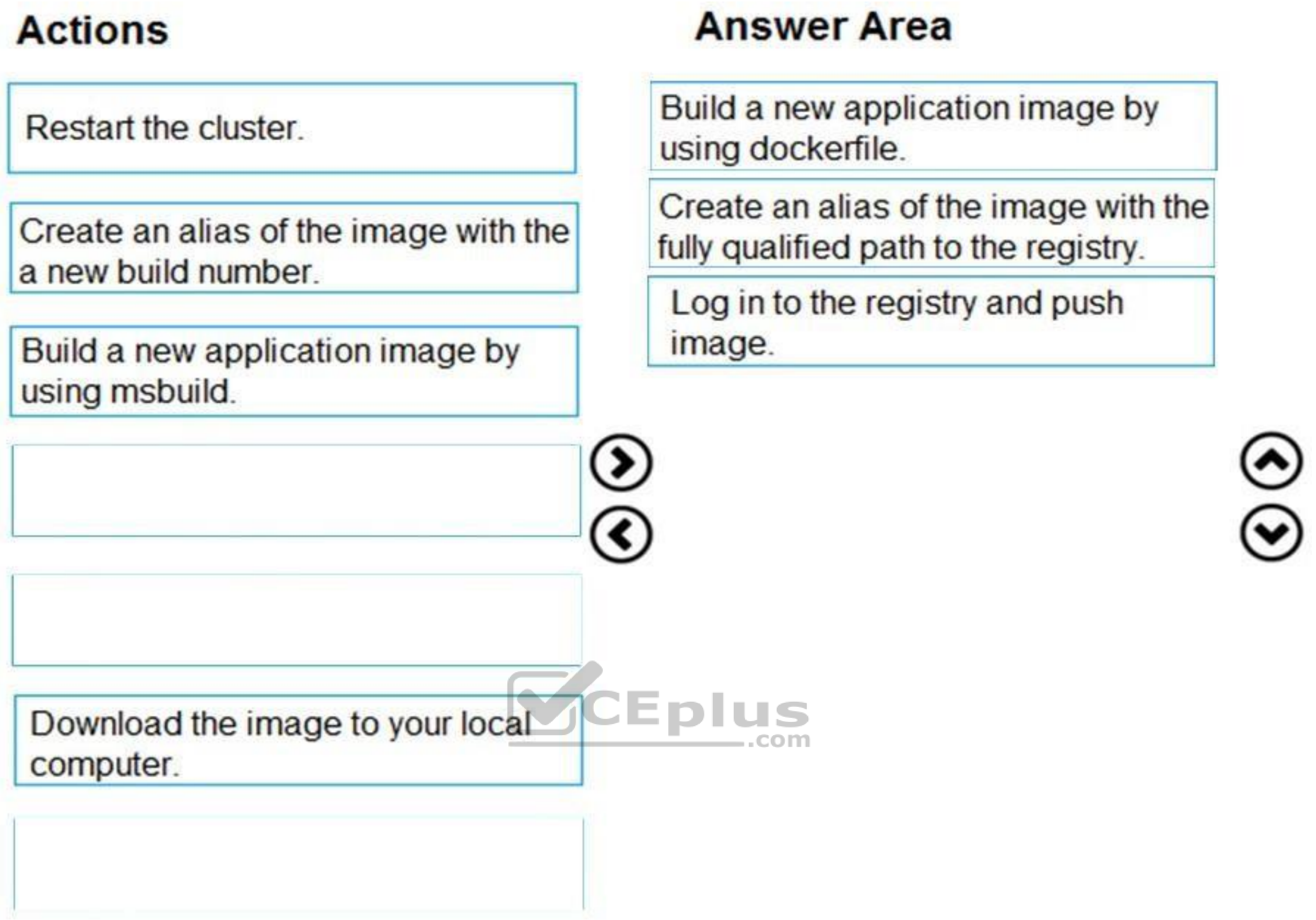
Create an alias of the image with the fully qualified path to the registry.

Build a new application image by using dockerfile.

Download the image to your local computer.

Log in to the registry and push image.

Correct Answer:



Actions

Restart the cluster.

Create an alias of the image with the a new build number.

Build a new application image by using msbuild.

Download the image to your local computer.

Answer Area

Build a new application image by using dockerfile.

Create an alias of the image with the fully qualified path to the registry.

Log in to the registry and push image.

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Step 1: Build a new application image by using dockerfile

Step 2: Create an alias if the image with the fully qualified path to the registry
Before you can push the image to a private registry, you've to ensure a proper image name. This can be achieved using the docker tag command. For demonstration purpose, we'll use Docker's hello world image, rename it and push it to ACR.

```
# pulls hello-world from the public docker hub
$ docker pull hello-world
# tag the image in order to be able to push it to a private registry
$ docker tag hello-word <REGISTRY_NAME>/hello-world
# push the image
$ docker push <REGISTRY_NAME>/hello-world
```

Step 3: Log in to the registry and push image

In order to push images to the newly created ACR instance, you need to login to ACR form the Docker CLI. Once logged in, you can push any existing docker image to your ACR instance.

Scenario:
Coho Winery plans to move the application to Azure and continue to support label creation. 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.

References: https://thorsten-hans.com/how-to-use-a-private-azure-container-registry-with-kubernetes-9b86e67b93b6 https://docs.microsoft.com/en-us/azure/container-registry/container-registry-tutorial-quick-task

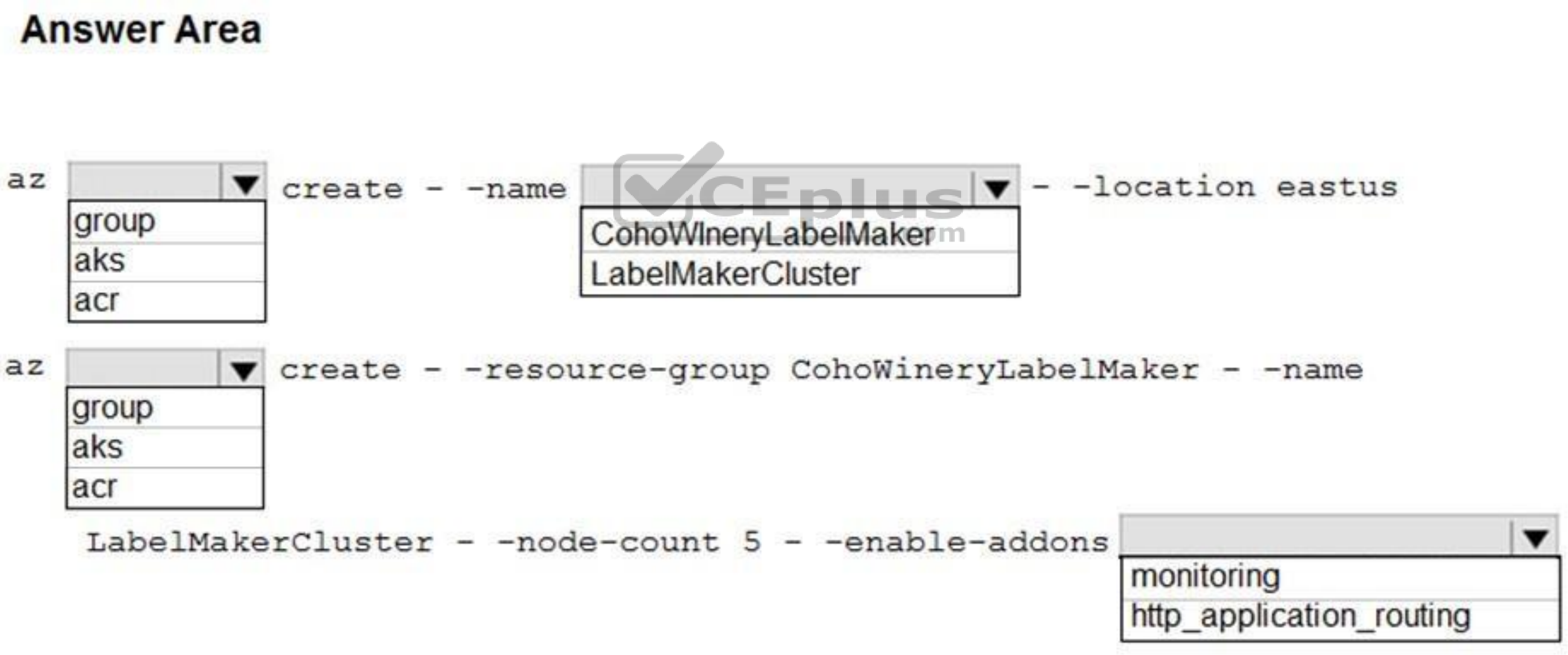
QUESTION 2
HOTSPOT

You need to ensure that you can deploy the LabelMaker application.

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

NOTE: Each correct selection is worth one point.

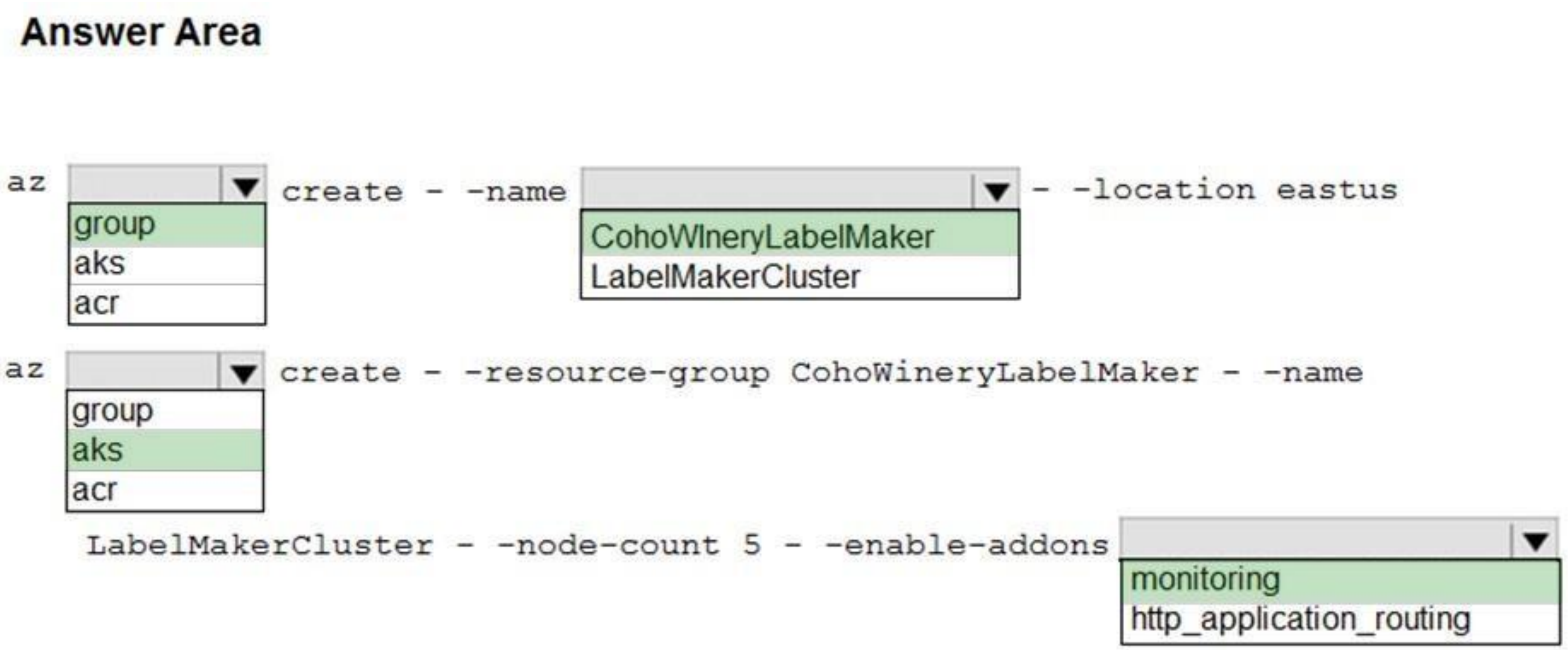
Hot Area:



Answer Area

```
az [▼] create - -name [▼] - -location eastus
az [▼] create - -resource-group CohoWineryLabelMaker - -name
LabelMakerCluster - -node-count 5 - -enable-addons [▼]
```

Correct Answer:



Answer Area

az ▼ create - -name ▼ - -location eastus

[✓ group]
aks
acr

[✓ CohoWIneryLabelMaker]
LabelMakerCluster

az ▼ create - -resource-group CohoWineryLabelMaker - -name LabelMakerCluster - -node-count 5 - -enable-addons ▼

group
[✓ aks]
acr

[✓ monitoring]
http_application_routing

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: group
Create a resource group with the az group create command. An Azure resource group is a logical group in which Azure resources are deployed and managed.

The following example creates a resource group named myResourceGroup in the westeurope location.

az group create --name myResourceGroup --location westeurope

Box 2: CohoWinterLabelMaker
Use the resource group named, which is used in the second command.

Box 3: aks
The command az aks create, is used to create a new managed Kubernetes cluster.

Box 4: monitoring
Scenario: 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.

QUESTION 3 You need to troubleshoot the
order workflow.

What should you do? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Review the run history.
B. Review the activity log.
C. Review the API connections.
D. Review the trigger history.

Correct Answer: BD
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The order workflow fails to run upon initial deployment to Azure.

Deployment errors arise from conditions that occur during the deployment process. They appear in the activity log.

References: https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-audit

Testlet 2

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware’s internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

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.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

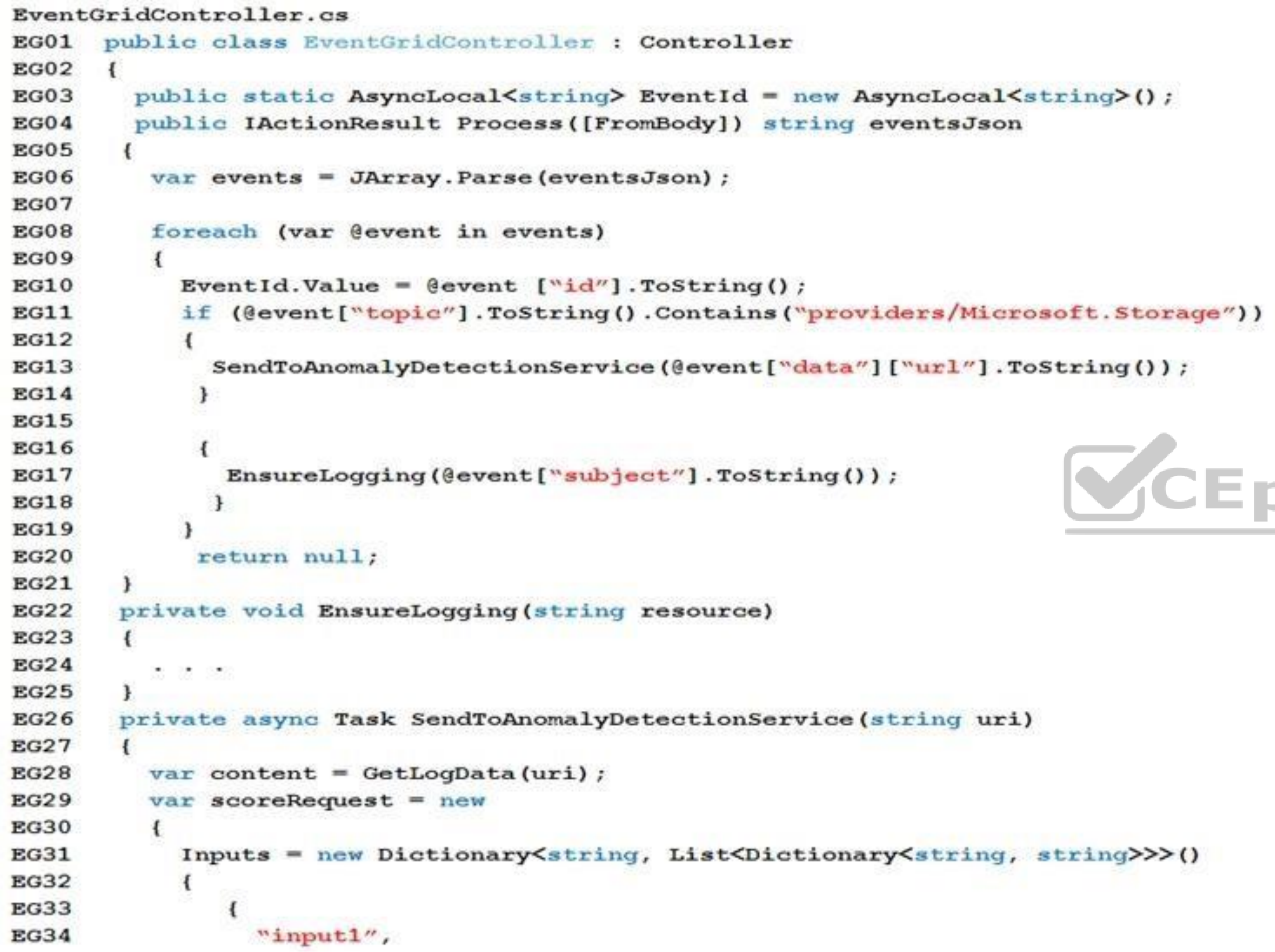
Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

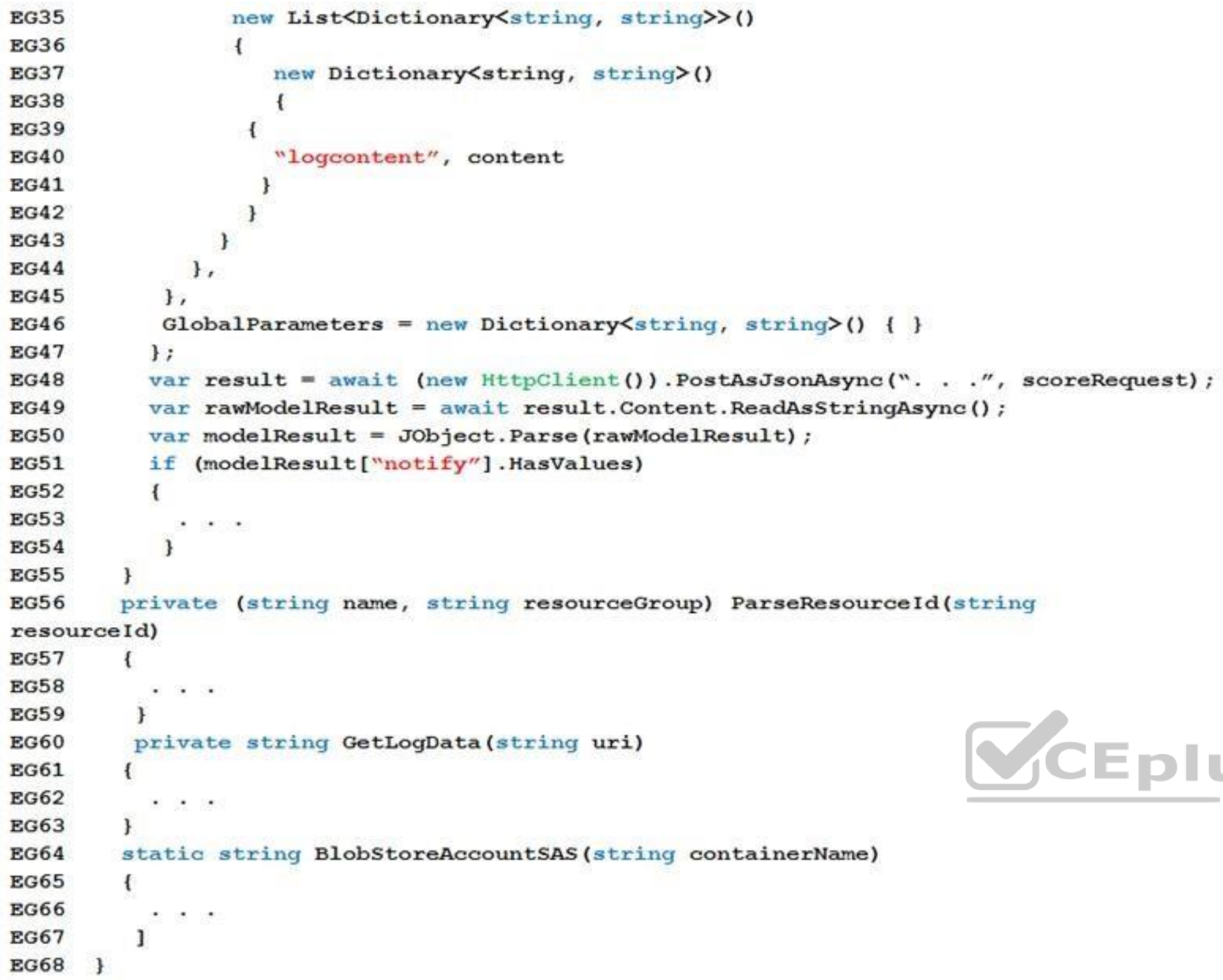
Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

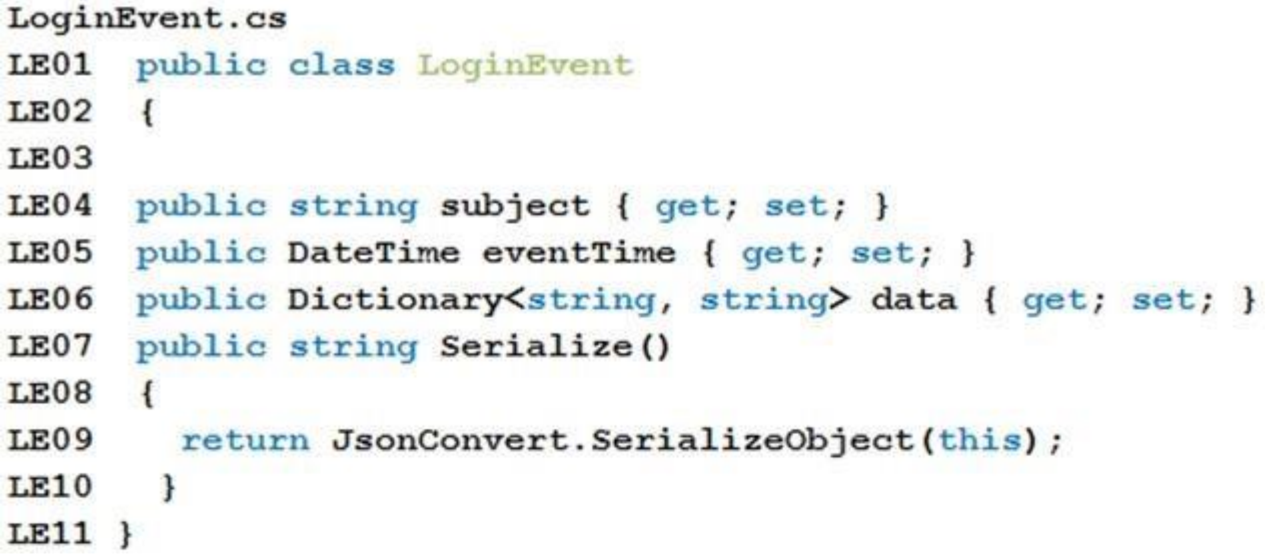


```
EventGridController.cs
EG01 public class EventGridController : Controller
EG02 {
EG03   public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04   public IActionResult Process([FromBody]) string eventsJson
EG05  {
EG06    var events = JArray.Parse(eventsJson);
EG07
EG08    foreach (var @event in events)
EG09    {
EG10      EventId.Value = @event [“id”].ToString();
EG11      if (@event[“topic”].ToString().Contains(“providers/Microsoft.Storage”))
EG12      {
EG13        SendToAnomalyDetectionService(@event[“data”][“url”].ToString());
EG14       }
EG15
EG16       {
EG17         EnsureLogging(@event[“subject”].ToString());
EG18        }
EG19      }
EG20     return null;
EG21  }
EG22  private void EnsureLogging(string resource)
EG23  {
EG24    . . .
EG25  }
EG26  private async Task SendToAnomalyDetectionService(string uri)
EG27  {
EG28    var content = GetLogData(uri);
EG29    var scoreRequest = new
EG30    {
EG31      Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32      {
EG33         {
EG34           “input1”,
```



```
EG35              new List<Dictionary<string, string>>()
EG36              {
EG37                 new Dictionary<string, string>()
EG38                 {
EG39               {
EG40                 "logcontent", content
EG41                }
EG42               }
EG43             }
EG44           },
EG45         },
EG46         GlobalParameters = new Dictionary<string, string>() { }
EG47        };
EG48        var result = await (new HttpClient()).PostAsJsonAsync(". . .", scoreRequest);
EG49        var rawModelResult = await result.Content.ReadAsStringAsync();
EG50        var modelResult = JObject.Parse(rawModelResult);
EG51        if (modelResult["notify"].HasValues)
EG52        {
EG53          . . .
EG54         }
EG55      }
EG56      private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57      {
EG58        . . .
EG59       }
EG60       private string GetLogData(string uri)
EG61      {
EG62        . . .
EG63      }
EG64      static string BlobStoreAccountSAS(string containerName)
EG65      {
EG66        . . .
EG67       ]
EG68    }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.



```
LoginEvent.cs
LE01  public class LoginEvent
LE02  {
LE03
LE04  public string subject { get; set; }
LE05  public DateTime eventTime { get; set; }
LE06  public Dictionary<string, string> data { get; set; }
LE07  public string Serialize()
LE08  {
LE09    return JsonConvert.SerializeObject(this);
LE10   }
LE11 }
```

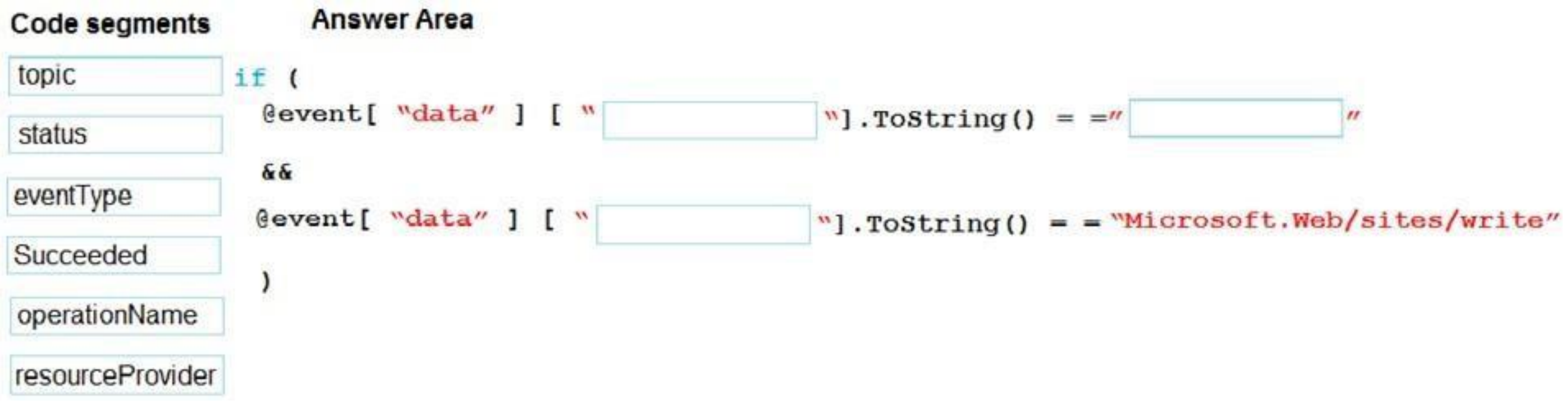
QUESTION 1

You need to add code at line EG15 in EventGridController.cs to ensure that the Log policy applies to all services.

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

topic

status

eventType

Succeeded

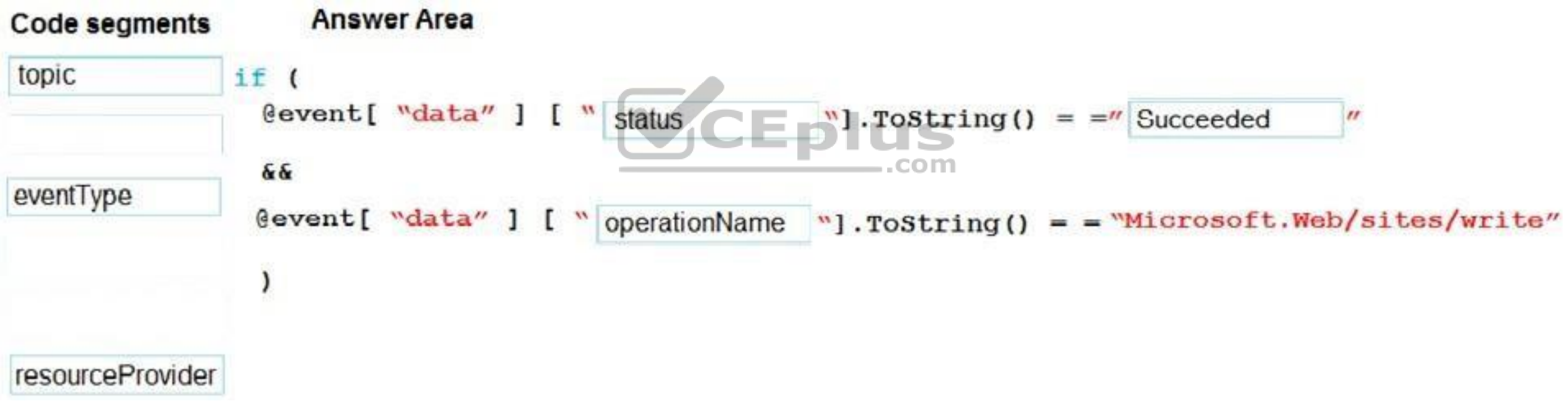
operationName

resourceProvider

Answer Area

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

Correct Answer:



Code segments

topic

eventType

resourceProvider

Answer Area

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

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Status

Box 2: Succeded

Box 3: operationName

Scenario: Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Question Set 3

QUESTION 1
HOTSPOT

You are developing a workflow solution using Azure technologies.

What should you implement to meet each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Correct Answer:

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Logic Apps only
You can manually trigger a logic app deployed in Azure from Visual Studio. On the Logic App Designer toolbar, choose Run Trigger.

To check the status and diagnose problems with logic app runs, you can review the details, such as inputs and outputs, for those runs in Visual Studio.

Box 2: Durable functions only

Box 3: Durable functions and Logic Apps

References: https://docs.microsoft.com/en-us/azure/logic-apps/manage-logic-apps-with-visual-studio https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-create-portal

QUESTION 2
You are developing a software solution for an autonomous transportation system. The solution uses large data sets and Azure Batch processing to simulate navigation sets for entire fleets of vehicles.

You need to create compute nodes for the solution on Azure Batch.

What should you do?

A. In Python, implement the class: TaskAddParameter B.
In Python, implement the class: JobAddParameter
C. In the Azure portal, create a Batch account.
D. In Azure CLI, run the command: az batch pool create

Correct Answer: D
Section: [none]
Explanation

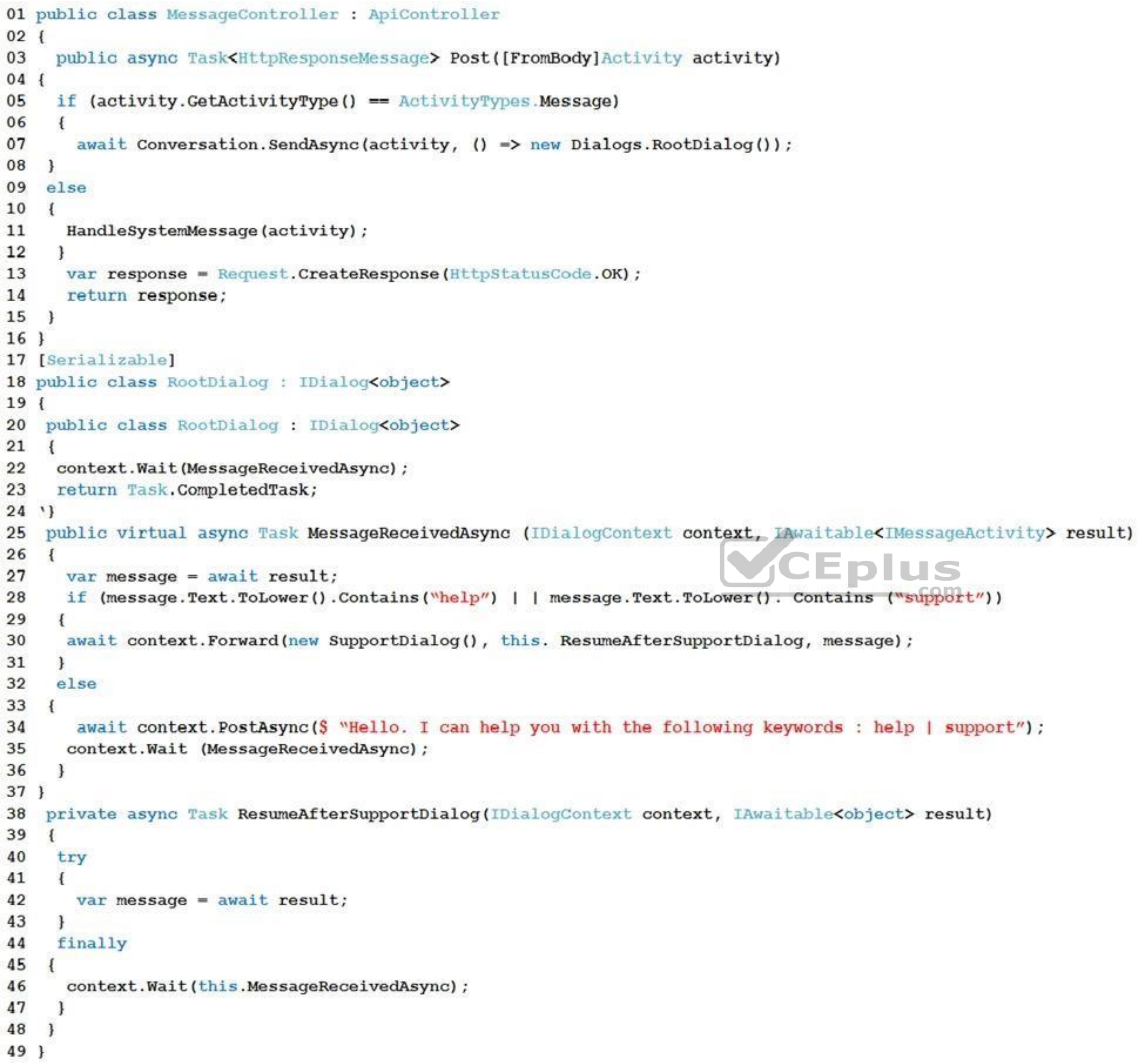
Explanation/Reference:
Explanation:
You can create a pool of compute nodes using the az batch pool create command. The following example creates a pool named mypool of 2 size Standard_A1_v2 nodes running Ubuntu 16.04 LTS. The suggested node size offers a good balance of performance versus cost for this quick example.

```
az batch pool create \
 --id mypool --vm-size Standard_A1_v2 \
 --target-dedicated-nodes 2 \
 --image canonical:ubuntuserver:16.04-LTS \  --node-agent-sku-
id "batch.node.ubuntu 16.04"
```

References:
https://docs.microsoft.com/en-us/azure/batch/quick-create-cli

QUESTION 3
HOTSPOT

You develop software solutions for a web services company. You have the following code. (Line numbers are for reference only.)



```
public class MessageController : ApiController
{
  public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
{
  if (activity.GetActivityType() == ActivityTypes.Message)
  {
    await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
 }
 else
 {
   HandleSystemMessage(activity);
  }
   var response = Request.CreateResponse(HttpStatusCode.OK);
   return response;
 }
}
[Serializable]
public class RootDialog : IDialog<object>
{
 public class RootDialog : IDialog<object>
 {
  context.Wait(MessageReceivedAsync);
  return Task.CompletedTask;
'}
 public virtual async Task MessageReceivedAsync (IDialogContext context, IAwaitable<IMessageActivity> result)
 {
   var message = await result;
   if (message.Text.ToLower().Contains("help") | | message.Text.ToLower(). Contains ("support"))
  {
   await context.Forward(new SupportDialog(), this. ResumeAfterSupportDialog, message);
  }
  else
 {
    await context.PostAsync($ "Hello. I can help you with the following keywords : help | support");
   context.Wait (MessageReceivedAsync);
  }
}
 private async Task ResumeAfterSupportDialog(IDialogContext context, IAwaitable<object> result)
 {
  try
  {
    var message = await result;
  }
  finally
 {
   context.Wait(this.MessageReceivedAsync);
  }
 }
}
```

You need to implement an immediate response customer support solution for the company’s website.

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

NOTE: Each correct selection is worth one point.

Hot Area:

Correct Answer:

Section: [none]
Explanation

Explanation/Reference:

QUESTION 4
DRAG DROP

A company backs up all manufacturing data to Azure Blob Storage. Admins move blobs from hot storage to archive tier storage every month.

You must automatically move blocks to Archive tier after they have not been accessed for **180** days. The path for any item that is not archived must be placed in an existing queue. This operation must be performed automatically once a month. You set the value of TierAgeInDays to 180.

How should you configure the Logic App? To answer, drag the appropriate triggers or action blocks to the correct trigger or action slots. Each trigger or action block 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:

Correct Answer:

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Recurrence

Box 2: Insert Entity Box

3 (if true): Tier Blob

Box 4: (if false):
Leave blank.

References: https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-perform-data-operations

QUESTION 5
HOTSPOT

You develop a microservice solution for a company.

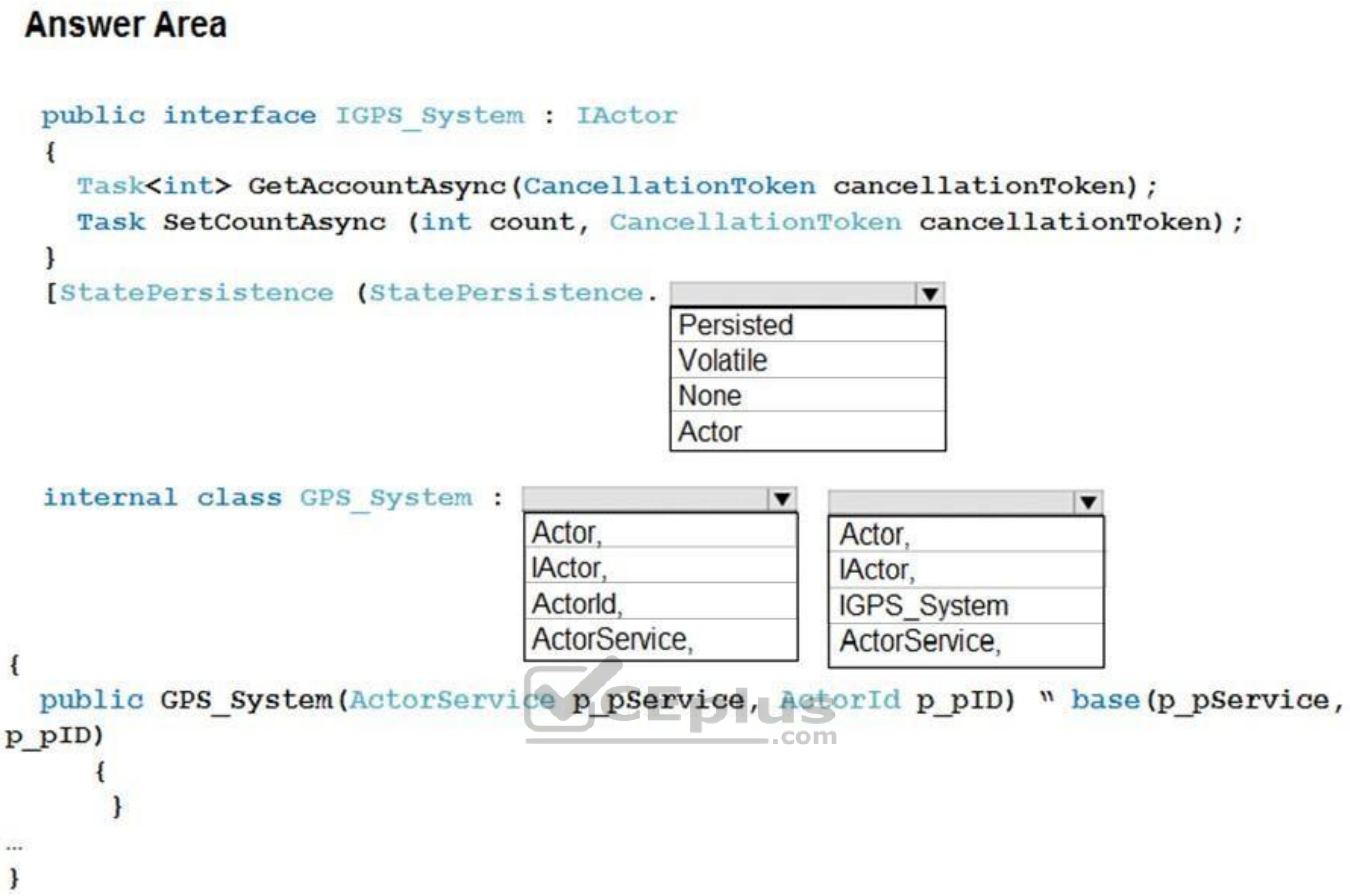
The solution uses the Actor design pattern for all services including mappings, GPS, and communication. The mapping and communication systems must persist state on disk. The GPS system must persist state in-memory. All Actors must have a service replica count of three.

You need to implement code for the GPS system.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

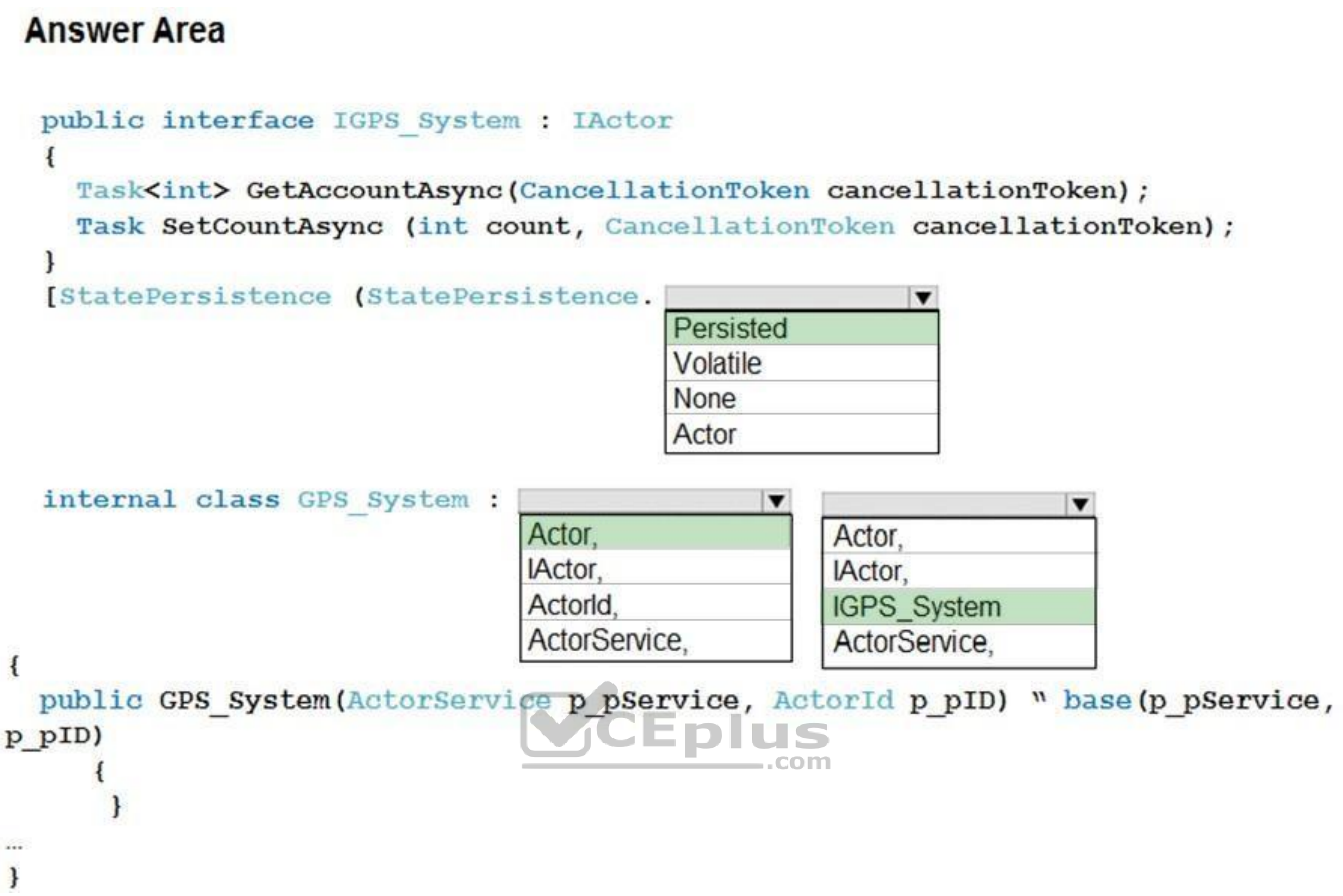


Answer Area

```
 public interface IGPS_System : IActor
 {
   Task<int> GetAccountAsync(CancellationToken cancellationToken);
   Task SetCountAsync (int count, CancellationToken cancellationToken);
 }
 [StatePersistence (StatePersistence. ▼
                                      Persisted
                                      Volatile
                                      None
                                      Actor

 internal class GPS_System : ▼              ▼
                             Actor,         Actor,
                             IActor,        IActor,
                             ActorId,       IGPS_System
                             ActorService,  ActorService,
{
  public GPS_System(ActorService p_pService, ActorId p_pID) " base(p_pService,
p_pID)
      {
       }
...
}
```

Correct Answer:



Answer Area

```
public interface IGPS_System : IActor
{
  Task<int> GetAccountAsync(CancellationToken cancellationToken);
  Task SetCountAsync (int count, CancellationToken cancellationToken);
}
[StatePersistence (StatePersistence.

internal class GPS_System :

{
  public GPS_System(ActorService p_pService, ActorId p_pID) " base(p_pService,
p_pID)
      {
       }
...
}
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:

Example:
The state manager retrieval methods return a reference to an object in local memory. Modifying this object in local memory alone does not cause it to be saved durably. When an object is retrieved from the state manager and modified, it must be reinserted into the state manager to be saved durably.

```
[StatePersistence(StatePersistence.Persisted)]
class MyActor : Actor, IMyActor
{
 public MyActor(ActorService actorService, ActorId actorId)
 : base(actorService, actorId)
 {
 }

 public Task SetCountAsync(int value)
 {
 return this.StateManager.SetStateAsync<int>("MyState", value);
 }
```

}

References: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-access-save-remove-state

QUESTION 6

You maintain an Azure Web App that runs in a container. The container is using a DockerFile that is copied to numerous places and consumes a large amount of storage.

You need to optimize the Dockerfile.

What should you do?

A. Minimize layers by concatenating all RUN instructions on one line.
B. Configure the CLI with a .dockerignore file.
C. Minimize layers by grouping actions in as few RUN instructions as possible.
D. Use multiple RUN instructions for cached operation.

Correct Answer: C
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Minimize the number of layers.
Prior to Docker 17.05, and even more, prior to Docker 1.10, it was important to minimize the number of layers in your image.C In Docker 1.10 and higher, only RUN, COPY, and ADD instructions create layers.

References: https://docs.docker.com/v17.09/engine/userguide/eng-image/dockerfile_best-practices

Testlet 1

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. 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 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.

Data

You identify the following requirements for data management and manipulation:

- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

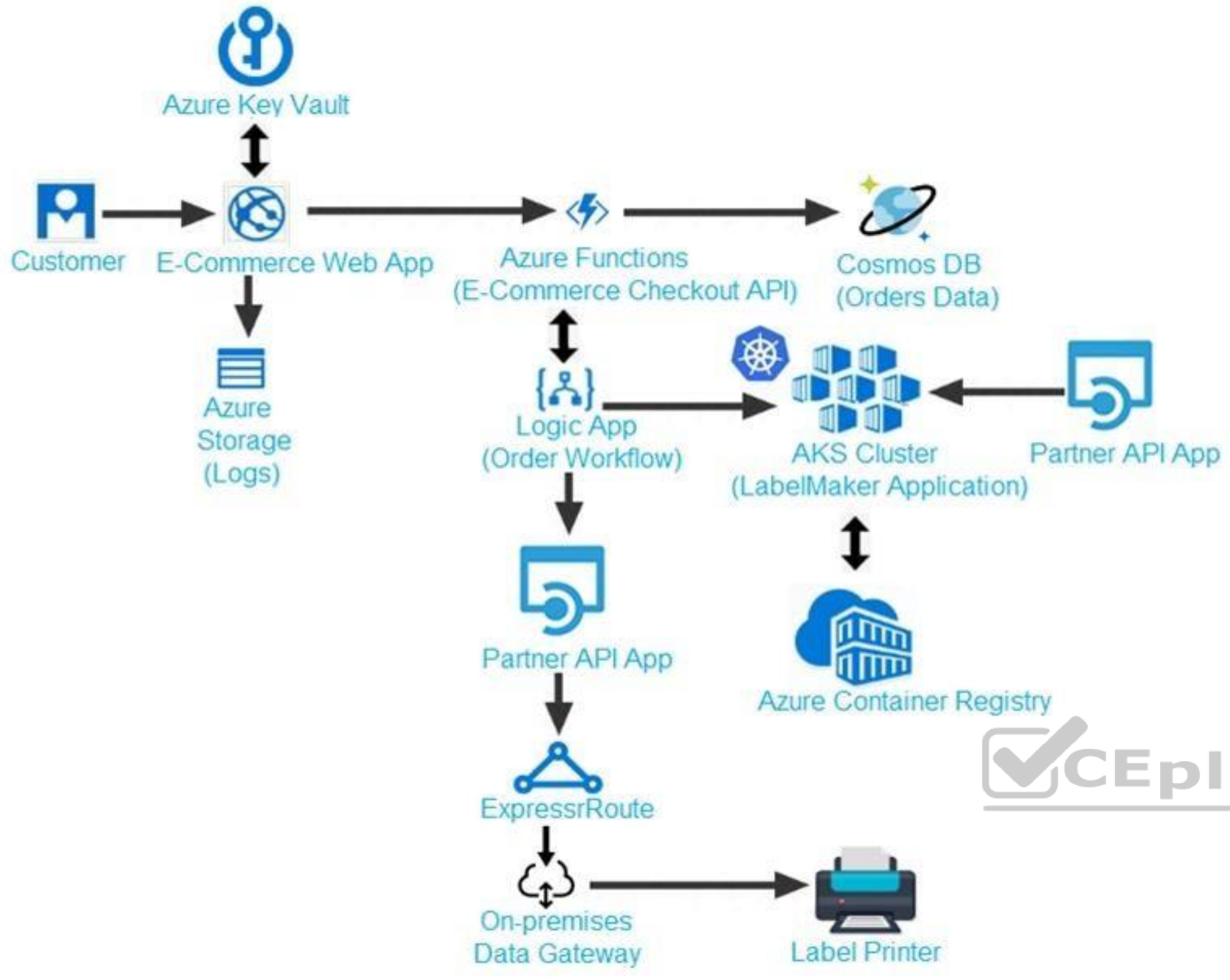
You have the following security requirements:

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

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.



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

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

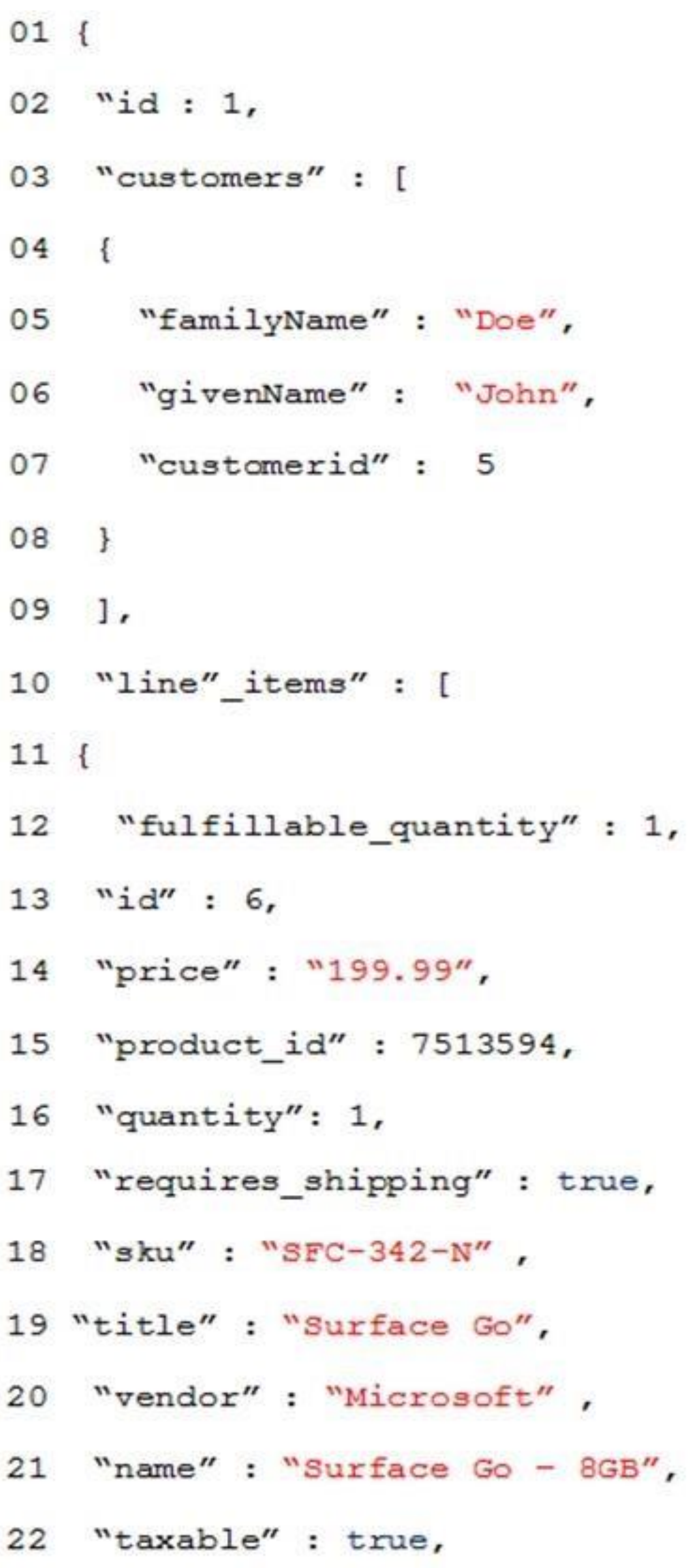
The order workflow fails to run upon initial deployment to Azure.

Order json.

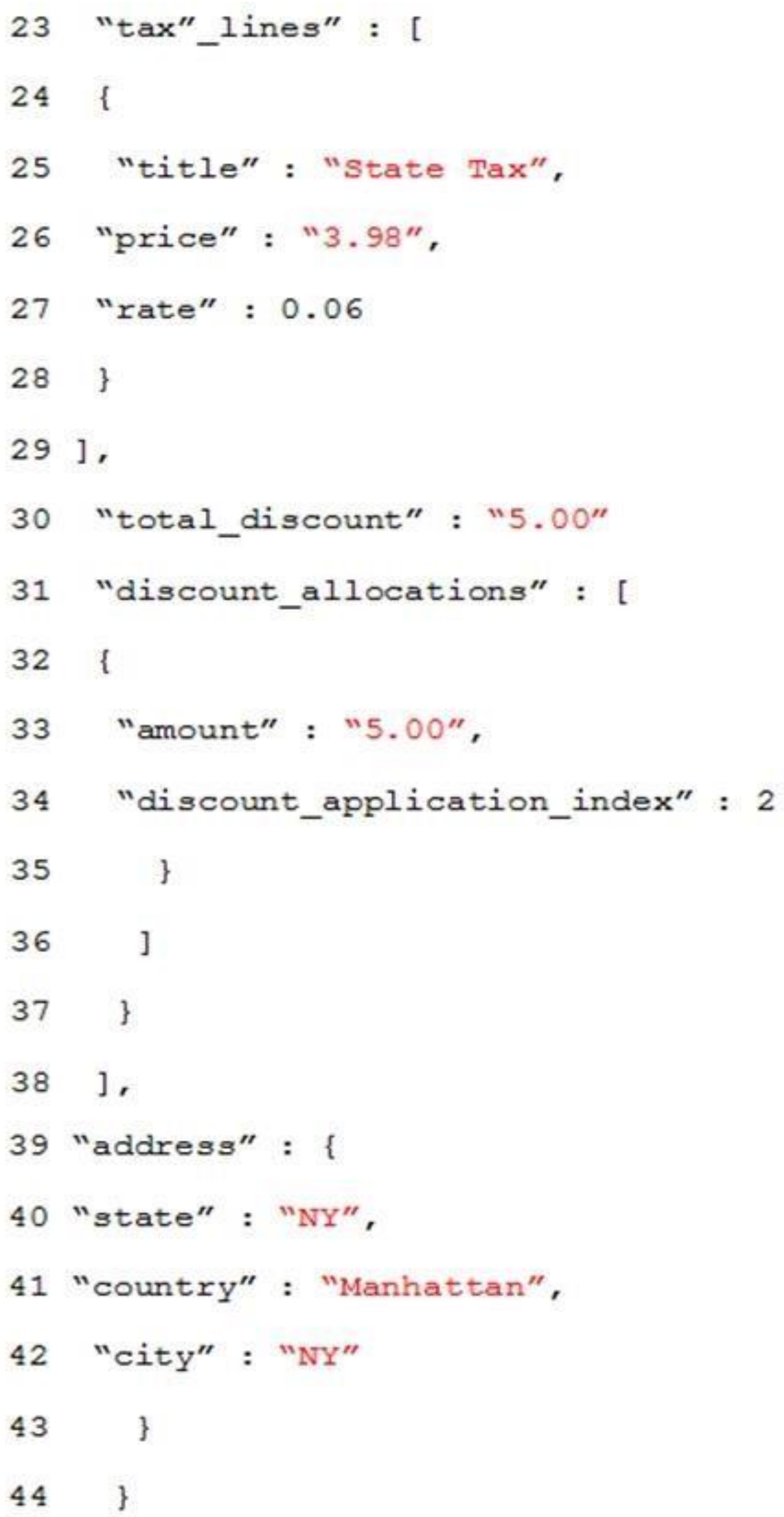
Relevant portions of the app files are shown below. Line numbers are included for reference only.

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

Order .json



```
01 {
02  “id : 1,
03  “customers” : [
04  {
05    “familyName” : “Doe”,
06    “givenName” :  “John”,
07    “customerid” :  5
08  }
09  ],
10  “line”_items” : [
11 {
12   “fulfillable_quantity” : 1,
13  “id” : 6,
14  “price” : “199.99”,
15  “product_id” : 7513594,
16  “quantity”: 1,
17  “requires_shipping” : true,
18  “sku” : “SFC-342-N” ,
19 “title” : “Surface Go”,
20  “vendor” : “Microsoft” ,
21  “name” : “Surface Go - 8GB”,
22  “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”,
“country” : “Manhattan”,
 “city” : “NY”
  }
 }
```

QUESTION 1
Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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: Place the Azure Active Directory account into an Azure AD group. Create a ClusterRoleBinding and assign it to the group.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

Permissions can be granted within a namespace with a RoleBinding, or cluster-wide with a ClusterRoleBinding.

References: https://kubernetes.io/docs/reference/access-authn-authz/rbac/

QUESTION 2
Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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 conditional access policy and assign it to the Azure Kubernetes Service cluster.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

Before an Azure Active Directory account can be used with the AKS cluster, a role binding or cluster role binding needs to be created.

References: https://docs.microsoft.com/en-us/azure/aks/aad-integration

QUESTION 3
Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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 RoleBinding and assign it to the Azure AD account.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

Permissions can be granted within a namespace with a RoleBinding, or cluster-wide with a ClusterRoleBinding.

References:
https://kubernetes.io/docs/reference/access-authn-authz/rbac/

Testlet 2

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

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.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

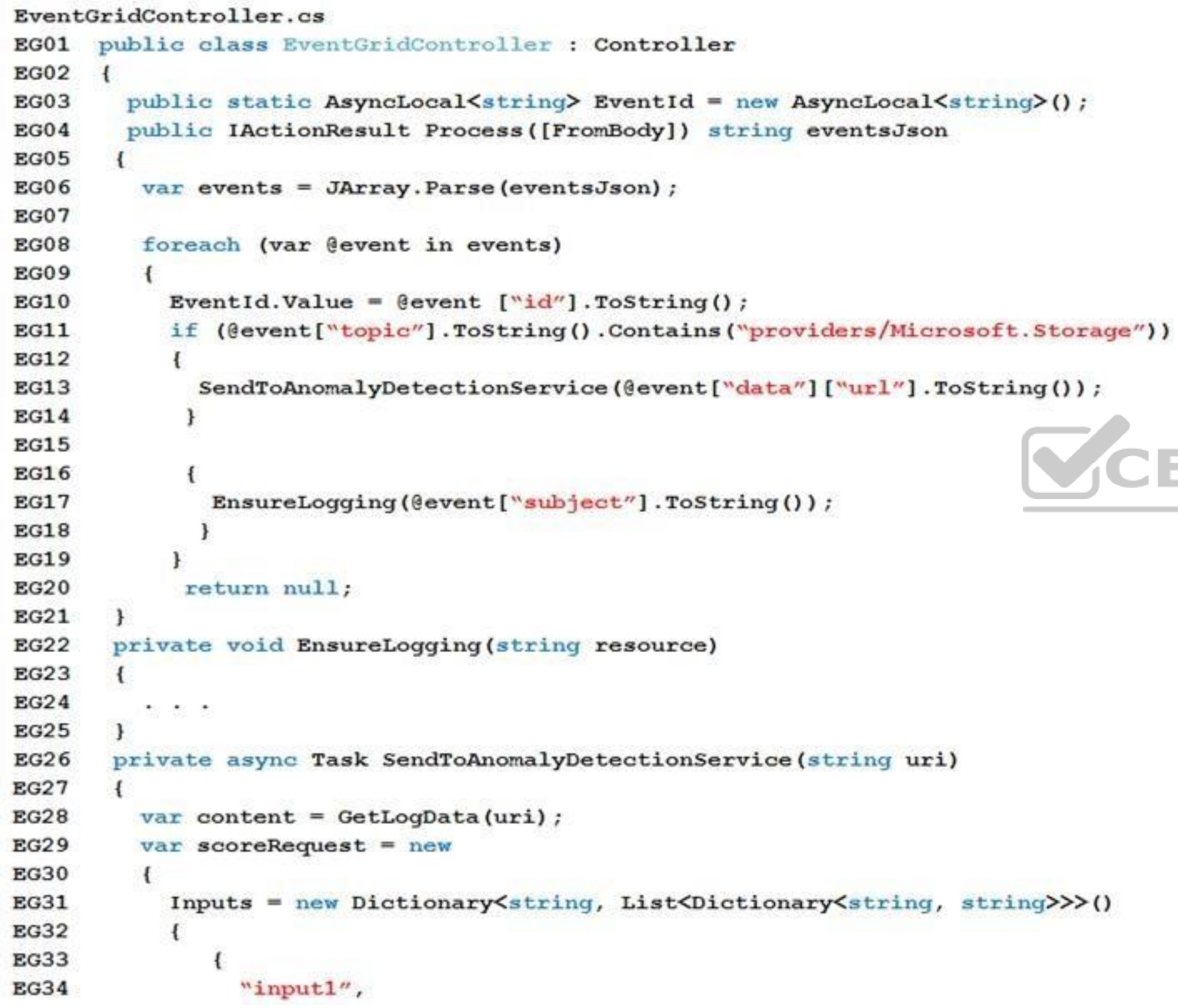
Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

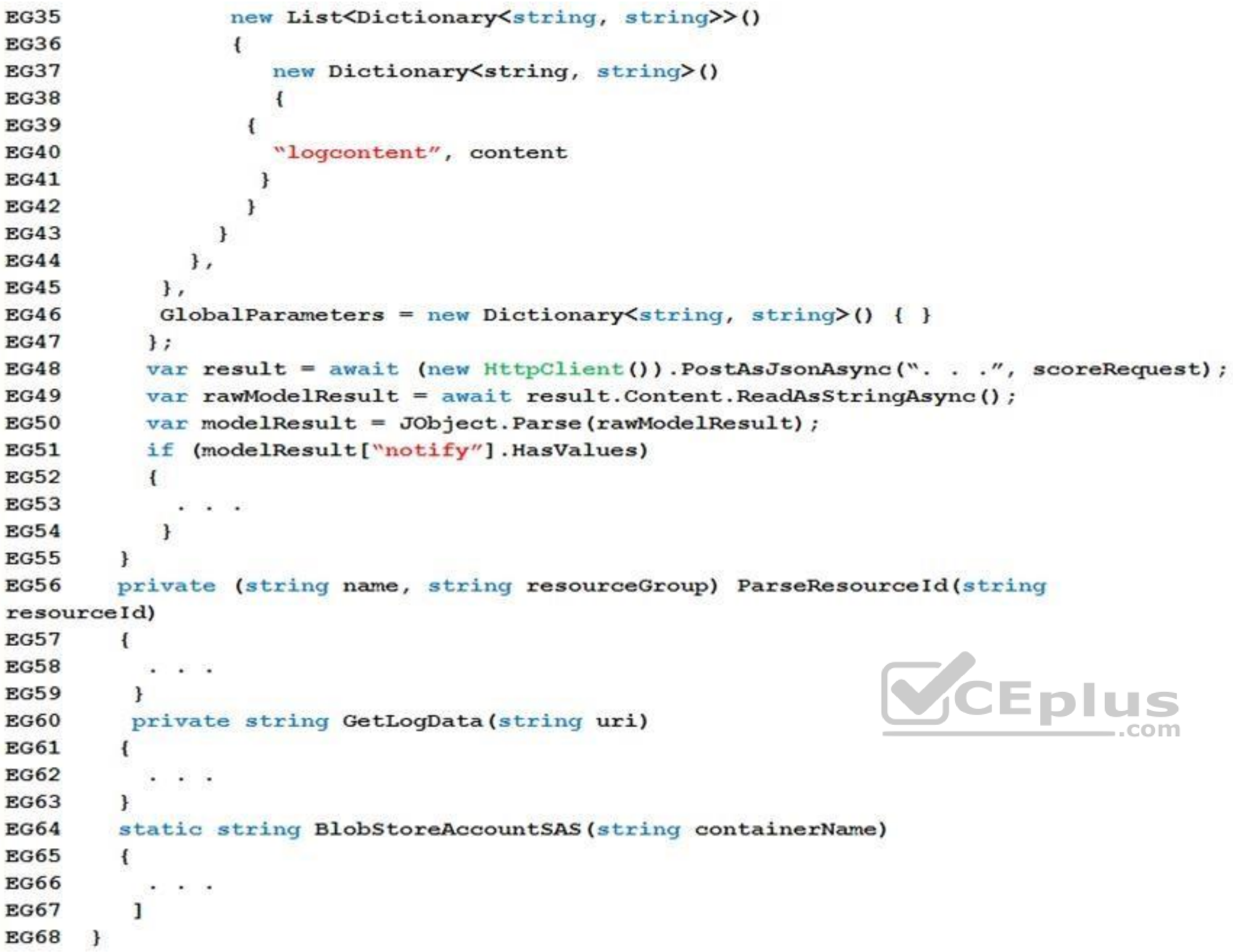
Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

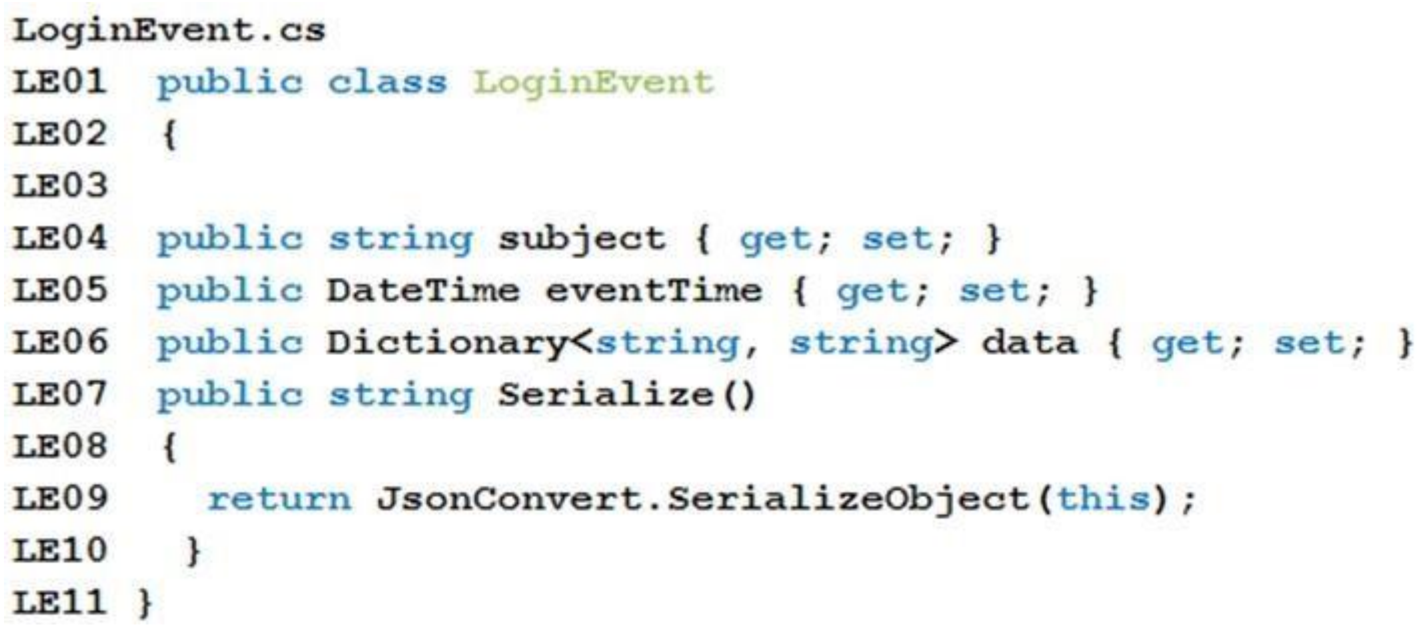


```
EventGridController.cs
EG01 public class EventGridController : Controller
EG02 {
EG03   public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04   public IActionResult Process([FromBody]) string eventsJson
EG05  {
EG06    var events = JArray.Parse(eventsJson);
EG07
EG08    foreach (var @event in events)
EG09    {
EG10      EventId.Value = @event [“id”].ToString();
EG11      if (@event[“topic”].ToString().Contains(“providers/Microsoft.Storage”))
EG12      {
EG13        SendToAnomalyDetectionService(@event[“data”][“url”].ToString());
EG14       }
EG15
EG16       {
EG17        EnsureLogging(@event[“subject”].ToString());
EG18        }
EG19      }
EG20       return null;
EG21  }
EG22  private void EnsureLogging(string resource)
EG23  {
EG24    . . .
EG25  }
EG26  private async Task SendToAnomalyDetectionService(string uri)
EG27  {
EG28    var content = GetLogData(uri);
EG29    var scoreRequest = new
EG30    {
EG31      Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32      {
EG33         {
EG34           “input1”,
```



```
EG35            new List<Dictionary<string, string>>()
EG36            {
EG37               new Dictionary<string, string>()
EG38               {
EG39             {
EG40               “logcontent”, content
EG41              }
EG42             }
EG43           }
EG44         },
EG45       },
EG46       GlobalParameters = new Dictionary<string, string>() { }
EG47      };
EG48      var result = await (new HttpClient()).PostAsJsonAsync(“. . .”, scoreRequest);
EG49      var rawModelResult = await result.Content.ReadAsStringAsync();
EG50      var modelResult = JObject.Parse(rawModelResult);
EG51      if (modelResult[“notify”].HasValues)
EG52      {
EG53        . . .
EG54       }
EG55    }
EG56    private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57    {
EG58      . . .
EG59     }
EG60     private string GetLogData(string uri)
EG61    {
EG62      . . .
EG63    }
EG64    static string BlobStoreAccountSAS(string containerName)
EG65    {
EG66      . . .
EG67     ]
EG68  }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.



```
LoginEvent.cs
LE01  public class LoginEvent
LE02  {
LE03
LE04  public string subject { get; set; }
LE05  public DateTime eventTime { get; set; }
LE06  public Dictionary<string, string> data { get; set; }
LE07  public string Serialize()
LE08  {
LE09    return JsonConvert.SerializeObject(this);
LE10    }
LE11 }
```

QUESTION 1

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 ensure that authentication events are triggered and processed according to the policy.

Solution: Create a new Azure Event Grid topic and add a subscription for the events.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Use a separate Azure Event Grid topics and subscriptions for sign-in and sign-out events.

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.

QUESTION 2

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 ensure that authentication events are triggered and processed according to the policy.

Solution: Create a new Azure Event Grid subscription for all authentication that delivers messages to an Azure Event Hub. Use the subscription to process signout events.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Use a separate Azure Event Grid topics and subscriptions for sign-in and sign-out events.

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.

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. Determine whether the solution meets the stated goals.

You need to ensure that authentication events are triggered and processed according to the policy.

Solution: Create separate Azure Event Grid topics and subscriptions for sign-in and sign-out events.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
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.

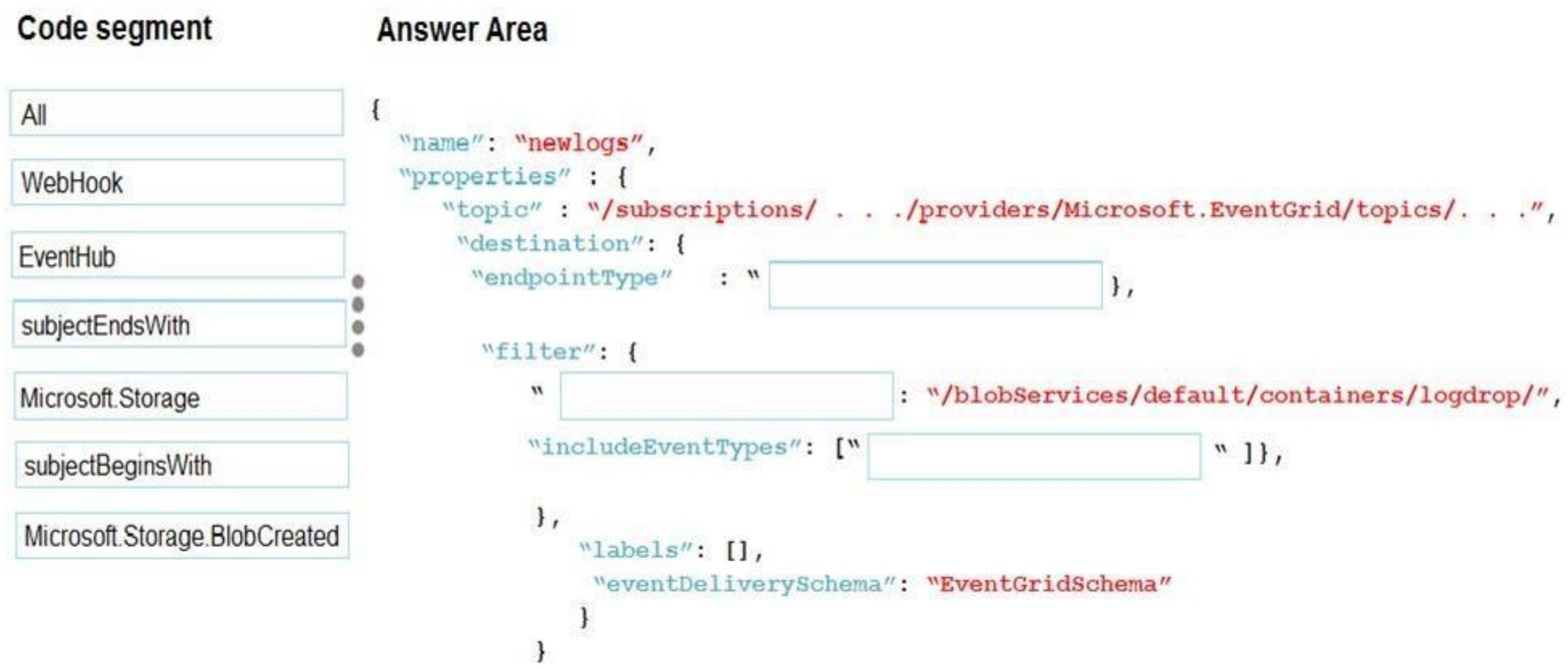
QUESTION 4
DRAG DROP

You need to implement the Log policy.

How should you complete the Azure Event Grid subscription? To answer, drag the appropriate JSON segments to the correct locations. Each JSON 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 segment

- All
- WebHook
- EventHub
- subjectEndsWith
- Microsoft.Storage
- subjectBeginsWith
- Microsoft.Storage.BlobCreated

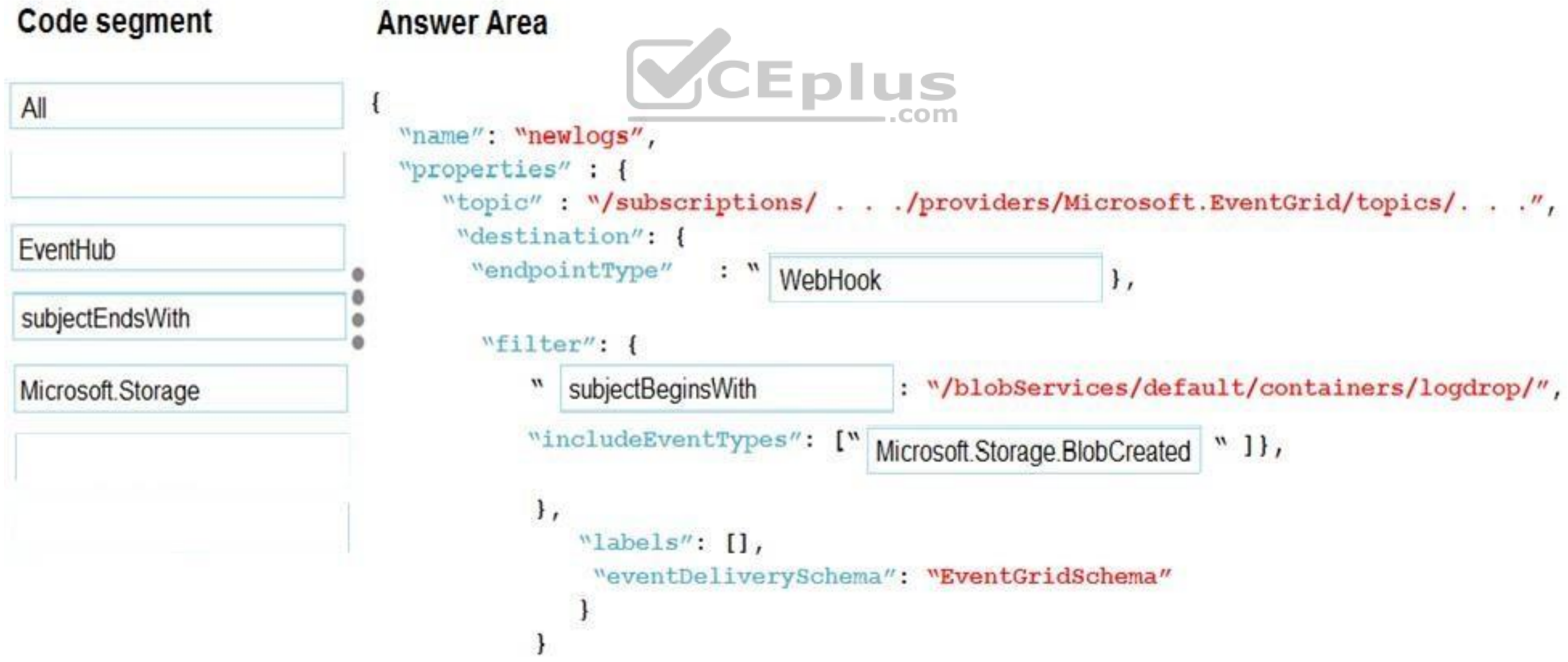
Answer Area

```
{
  "name": "newlogs",
  "properties" : {
     "topic" : "/subscriptions/ . . ./providers/Microsoft.EventGrid/topics/. . .",
      "destination": {
       "endpointType"   : "                    },

        "filter": {
            "                    : "/blobServices/default/containers/logdrop/",
            "includeEventTypes": ["                    " ]},

            },
              "labels": [],
               "eventDeliverySchema": "EventGridSchema"
              }
            }
```

Correct Answer:



Code segment

- All
- EventHub
- subjectEndsWith
- Microsoft.Storage

Answer Area

```
{
  "name": "newlogs",
  "properties" : {
     "topic" : "/subscriptions/ . . ./providers/Microsoft.EventGrid/topics/. . .",
      "destination": {
       "endpointType"   : " WebHook              },

        "filter": {
            " subjectBeginsWith   : "/blobServices/default/containers/logdrop/",
            "includeEventTypes": [" Microsoft.Storage.BlobCreated " ]},

            },
              "labels": [],
               "eventDeliverySchema": "EventGridSchema"
              }
            }
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1:WebHook

Scenario: If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

endpointType: The type of endpoint for the subscription (webhook/HTTP, Event Hub, or queue).

Box 2: SubjectBeginsWith

Box 3: Microsoft.Storage.BlobCreated

Scenario: Log Policy
All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named logdrop. Logs must remain in the container for 15 days.

Example subscription schema

```
{
  "properties": {
    "destination": {
      "endpointType": "webhook",
      "properties": {
          "endpointUrl": "https://example.azurewebsites.net/api/HttpTriggerCSharp1?code=VXbGWce53l48Mt8wuotr0GPmyJ/nDT4hgdFj9DpBiRt38qqnnm5OFg=="
      }
    },
    "filter": {
      "includedEventTypes": [ "Microsoft.Storage.BlobCreated", "Microsoft.Storage.BlobDeleted" ],
      "subjectBeginsWith": "blobServices/default/containers/mycontainer/log",
      "subjectEndsWith": ".jpg",
      "isSubjectCaseSensitive ": "true"
    }
  }

}
```

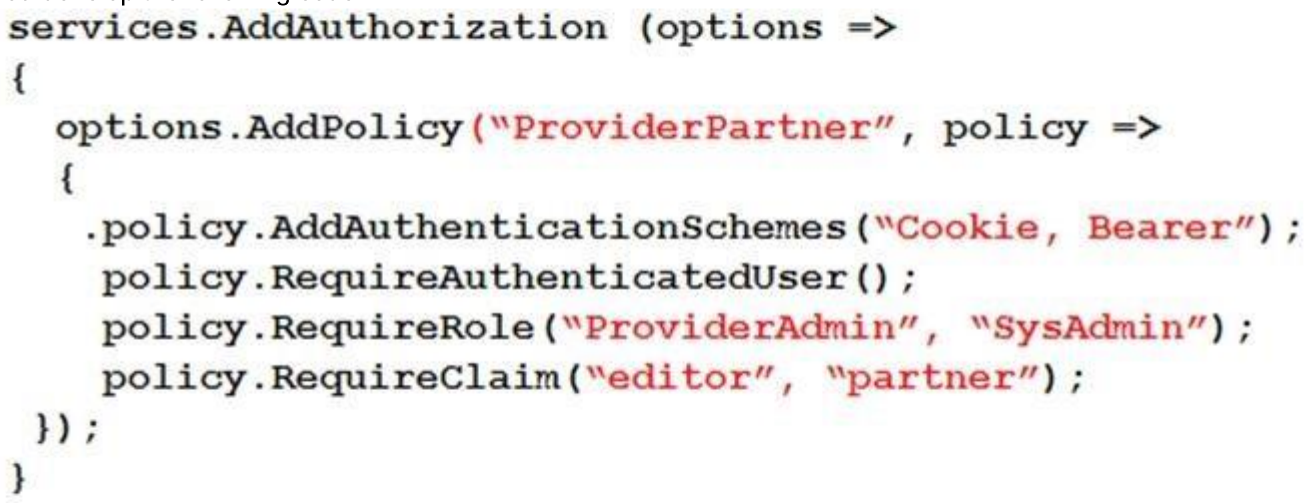
References: https://docs.microsoft.com/en-us/azure/event-grid/subscription-creation-schema

Question Set 3

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.

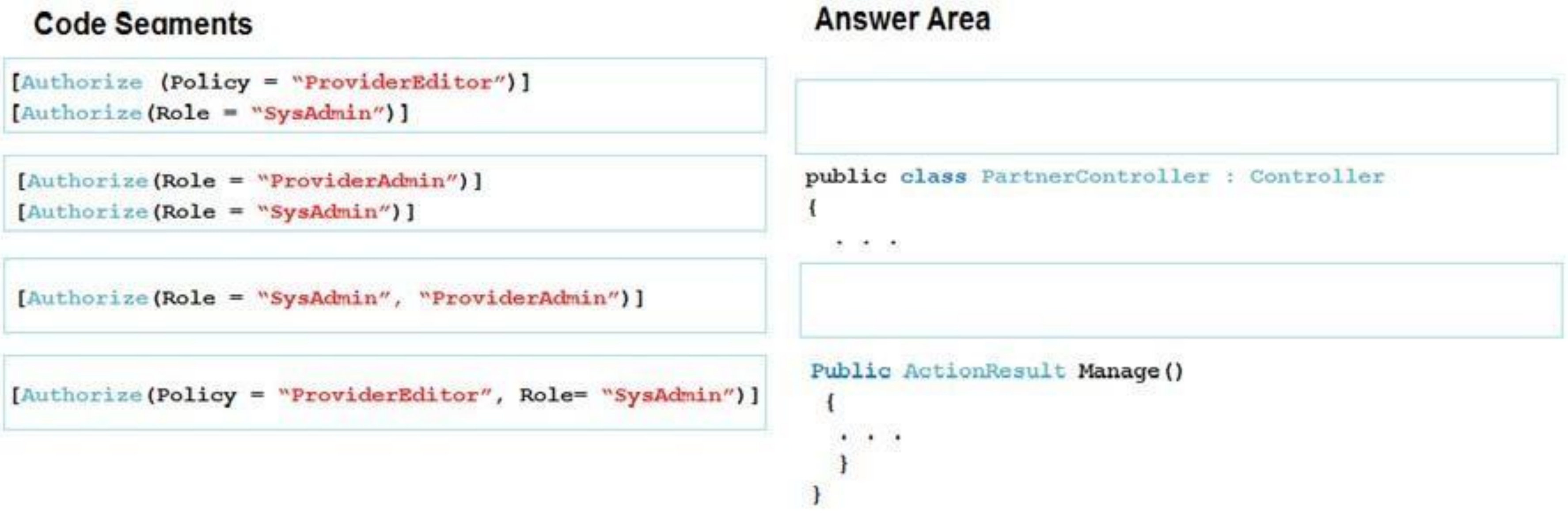
You need to ensure that the application meets the following authorization requirements:

▪ 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”)]
```

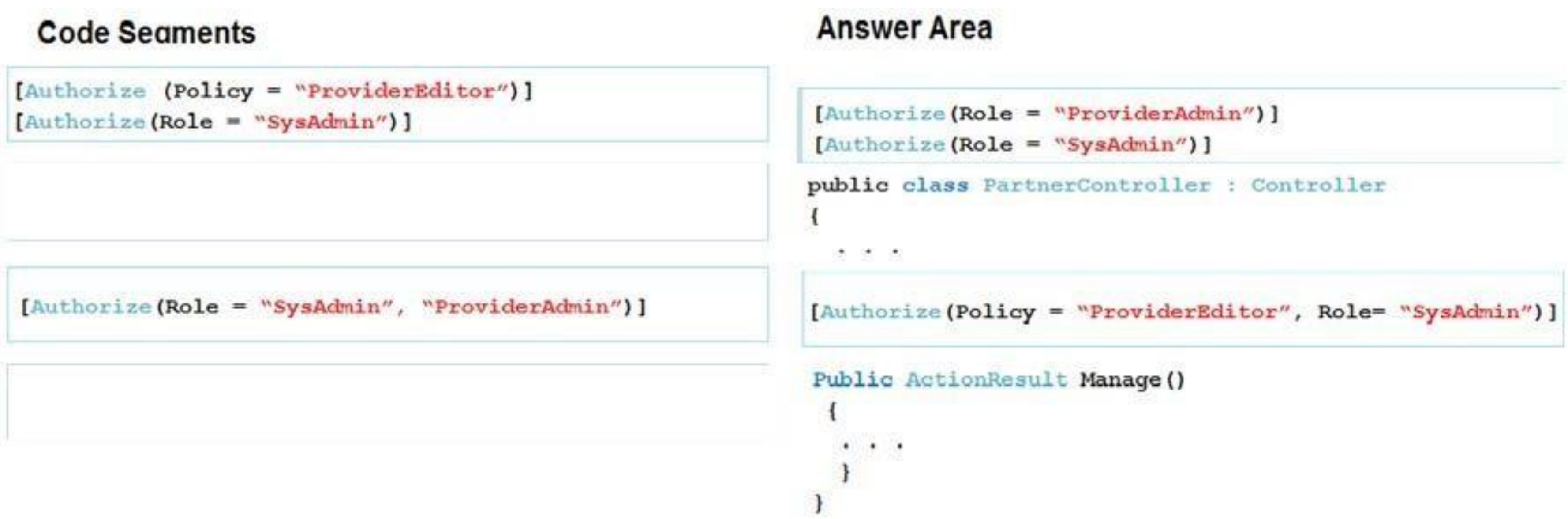
```
[Authorize(Policy = “ProviderEditor”, Role= “SysAdmin”)]
```

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()
 {
  . . .
 }
}
```

Section: [none]
Explanation

Explanation/Reference:
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
You host an on-premises ASP.NET Web API at the company headquarters. The Web API is consumed by applications running at the company’s branch offices using the Azure Relay service. All the users of the applications are on the same Azure Active Directory (Azure AD).

You need to ensure that the applications can consume the Web API.

What should you do?

A. Create a Shared Access policy for Send permissions and another for Receive permissions.
B. Create separate Azure AD groups named **Senders** and **Receivers**. In Access Control (IAM) for the Relay namespace, assign **Senders** the Reader role and assign **Receivers** the Reader role.
C. Create dedicated Azure AD identities named **Sender** and **Receiver**. Assign **Sender** the Azure AD Identity Reader role. Assign **Receiver** the Azure AD Identity Reader role. Configure applications to use the respective identities.
D. Create a Shared Access policy for the namespace. Use a connection string in Web API and use a different connection string in consumer applications.

Correct Answer: D
Section: [none]
Explanation

Explanation/Reference:
Explanation:
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.

When you publish an application through Azure Active Directory Application Proxy, you create an external URL for your users to go to when they're working remotely. This URL gets the default domain yourtenant.msappproxy.net.

References: https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy-configure-custom-domain
Testlet 1

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

Best for You Organics Company is a global restaurant franchise that has multiple locations. The company wants to enhance user experiences and vendor integrations. The company plans to implement automated mobile ordering and delivery services.

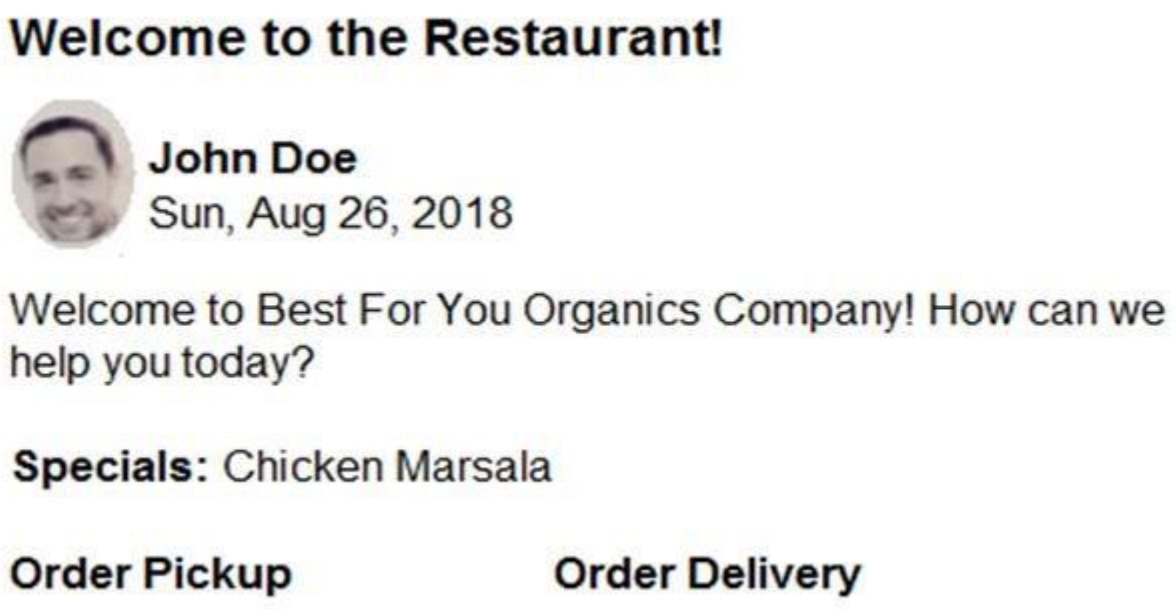
Best For You Organics hosts an Azure web app at the URL https://www.bestforyouorganics.com.
Users can use the web app to browse restaurant locations, menu items, nutritional, information, and company information. The company developed and deployed a cross-platform mobile app.

Requirements

You must develop a chartbot by using the Bot Builder SDK and Language Understanding Intelligence Service (LUIS). The chatbot must allow users to order food for pickup or delivery.

The chatbot must meet the following requirements:

▪ Ensure that chatbot endpoint can be accessed only by the Bot Framework connector.
▪ Use natural language processing and speech recognition so that users can interact with the chatbot by using text and voice. Processing must be server-based. ▪
Alert users about promotions at local restaurants.
▪ Enable users to place an order for delivery or pickup by using their voice.
▪ Greet the user upon sigh-in by displaying a graphical interface that contains action buttons. ▪
The chatbot greeting interface must match the formatting of the following example:



Welcome to the Restaurant!

John Doe
Sun, Aug 26, 2018

Welcome to Best For You Organics Company! How can we help you today?

Specials: Chicken Marsala

Order Pickup **Order Delivery**

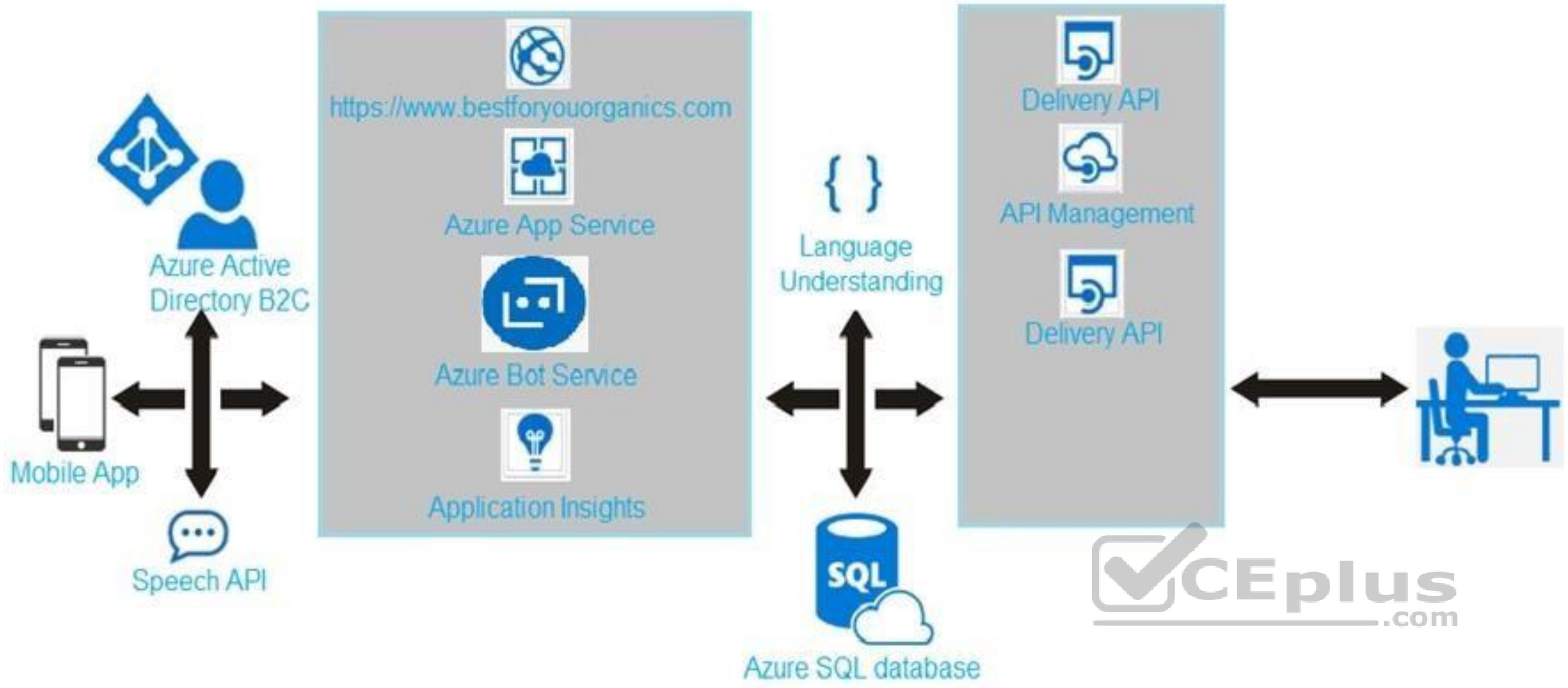
Vendor API
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

▪ API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.

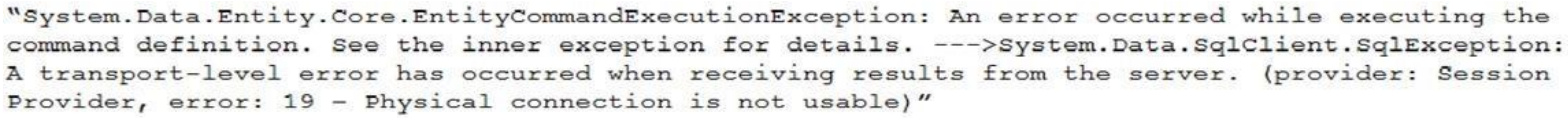
- If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.
- APIs must prevent API usage spikes on a per-subscription basis by limiting the call rate to 100 calls per minute.
- The Inventory API must be written by using ASP.NET Core and Node.js.
- The API must be updated to provide an interface to Azure SQL Database. Database objects must be managed by using code.
- The Delivery API must be protected by using the OAuth 2.0 protocol with Azure Active Directory (Azure AD) when called from the Azure web app. You register the Delivery API and web app in Azure AD. You enable OAuth 2.0 in the web app.
- The delivery API must update the Products table, the Vendor transactions table, and the Billing table in a single transaction.

The Best For You Organics Company architecture team has created the following diagram depicting the expected deployments into Azure:



Delivery API

The Delivery API intermittently throws the following exception:



```
"System.Data.Entity.Core.EntityCommandExecutionException: An error occurred while executing the
command definition. See the inner exception for details. --->System.Data.SqlClient.SqlException:
A transport-level error has occurred when receiving results from the server. (provider: Session
Provider, error: 19 - Physical connection is not usable)"
```

Chatbot greeting

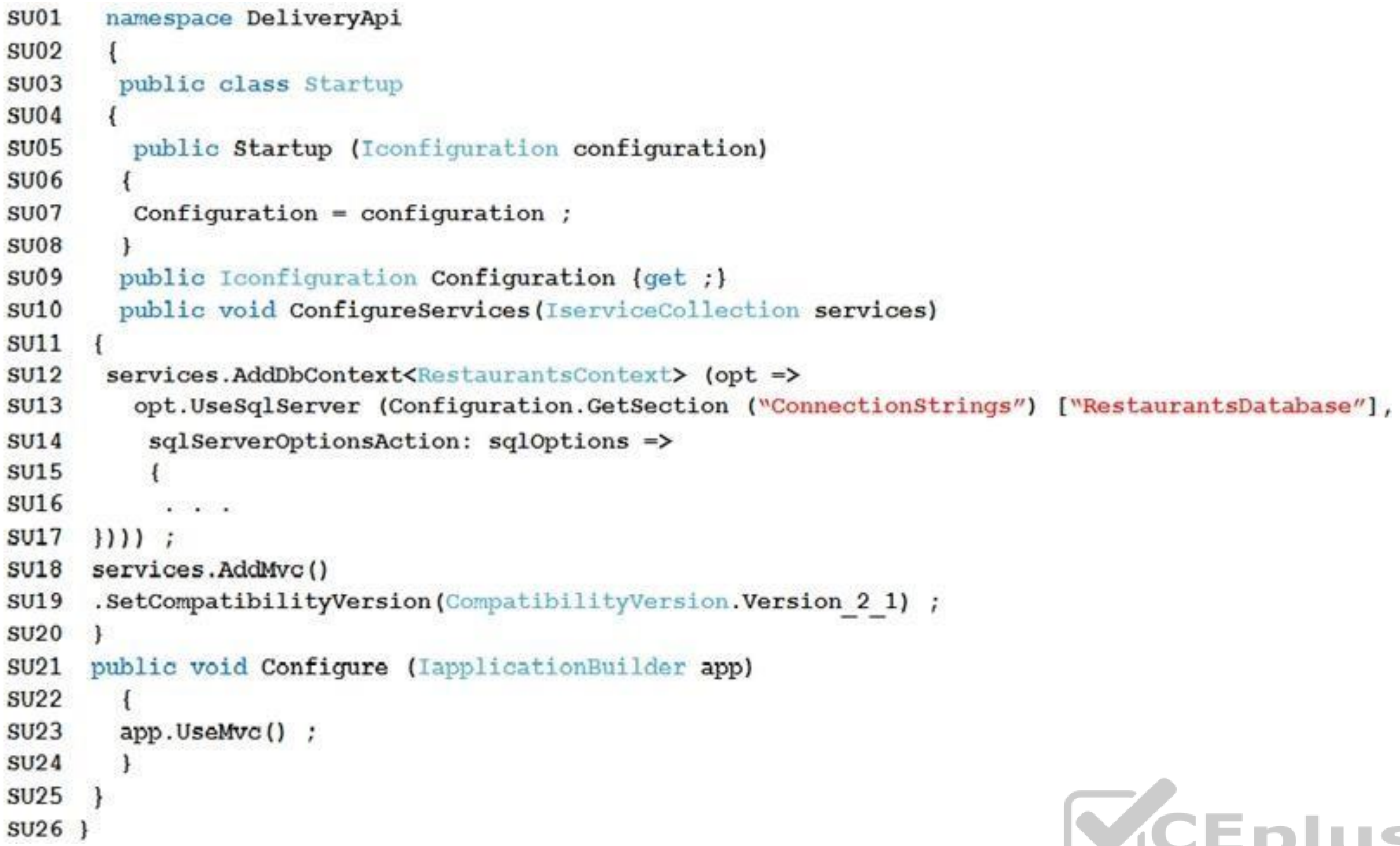
The chatbot's greeting does not show the user's name. You need to debug the chatbot locally.

Language processing

Users report that the bot fails to understand when a customer attempts to order dishes that use Italian names.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

Startup.cs



```
SU01  namespace DeliveryApi
SU02  {
SU03   public class Startup
SU04  {
SU05    public Startup (Iconfiguration configuration)
SU06    {
SU07    Configuration = configuration ;
SU08    }
SU09   public Iconfiguration Configuration {get ;}
SU10   public void ConfigureServices(IserviceCollection services)
SU11 {
SU12  services.AddDbContext<RestaurantsContext> (opt =>
SU13    opt.UseSqlServer (Configuration.GetSection ("ConnectionStrings") ["RestaurantsDatabase"],
SU14     sqlServerOptionsAction: sqlOptions =>
SU15     {
SU16      . . .
SU17 }))) ;
SU18 services.AddMvc()
SU19 .SetCompatibilityVersion(CompatibilityVersion.Version_2_1) ;
SU20 }
SU21 public void Configure (IapplicationBuilder app)
SU22   {
SU23   app.UseMvc() ;
SU24   }
SU25 }
SU26 }
```

QUESTION 1

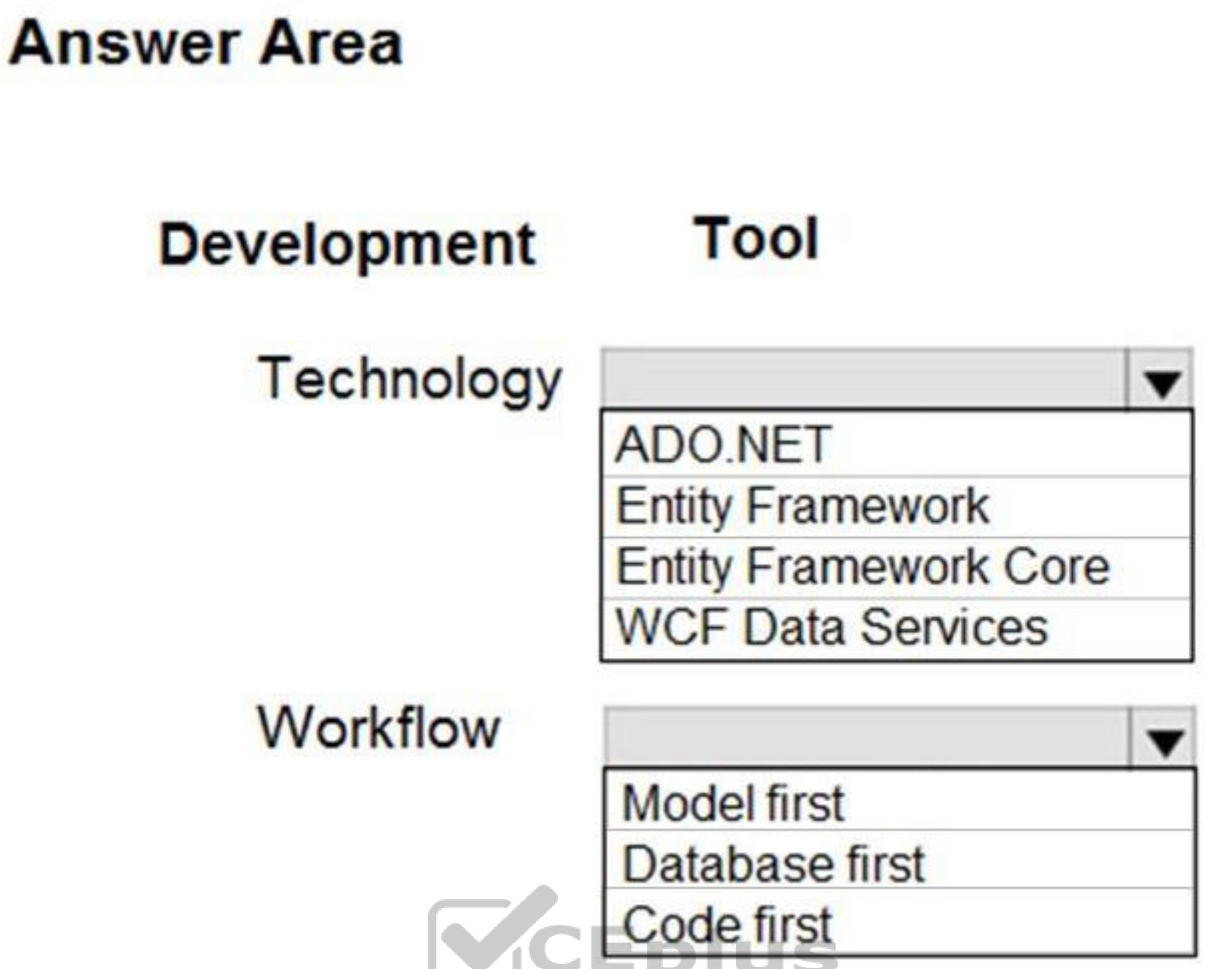
HOTSPOT

You need to update the Inventory API.

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

NOTE: Each correct selection is worth one point.

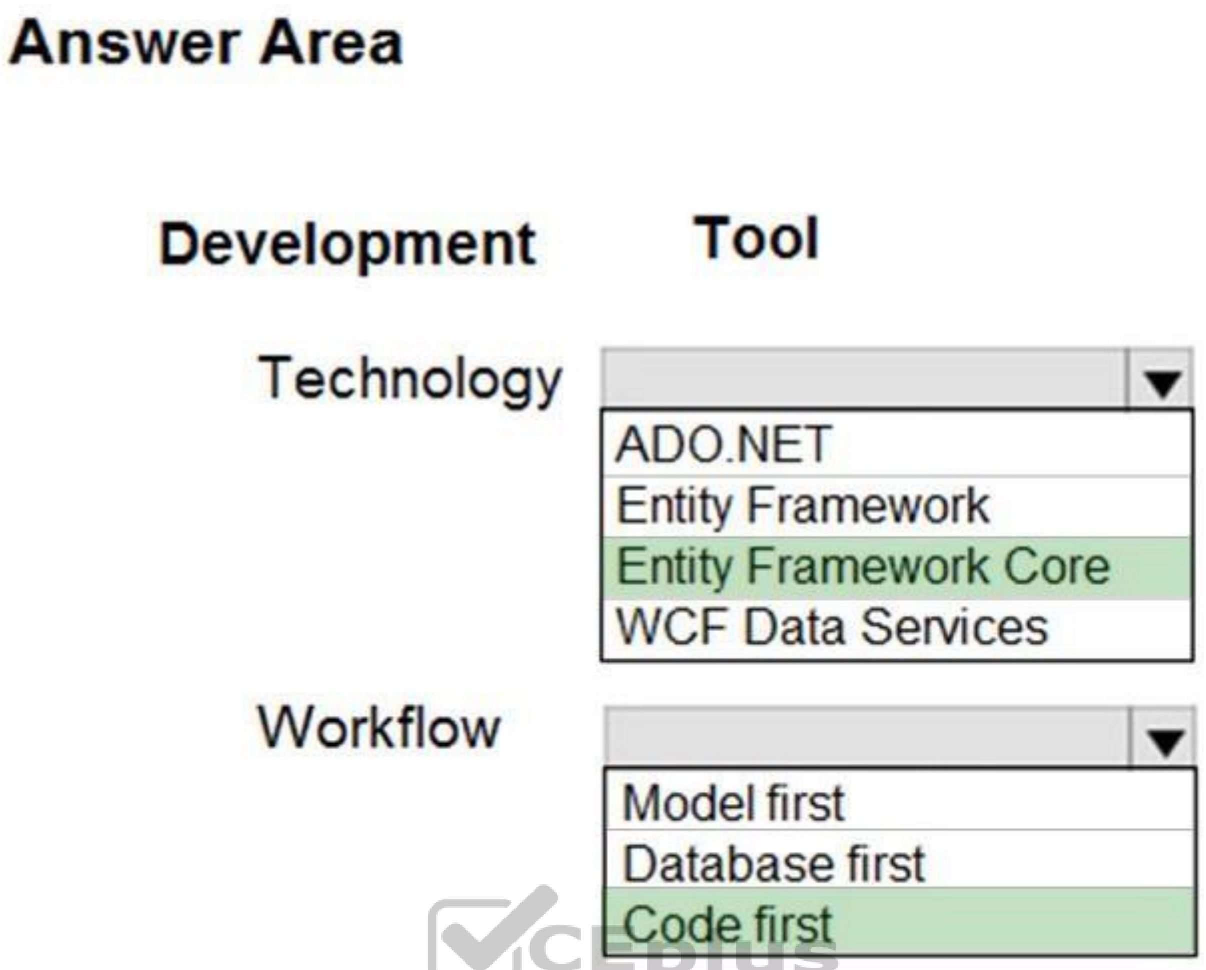
Hot Area:



Answer Area

Development	Tool
Technology	▼ ADO.NET Entity Framework Entity Framework Core WCF Data Services
Workflow	▼ Model first Database first Code first

Correct Answer:



Answer Area

Development	Tool
Technology	ADO.NET Entity Framework [✓ Entity Framework Core] WCF Data Services
Workflow	Model first Database first [✓ Code first]

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The Inventory API must be written by using ASP.NET Core and Node.js.

Box 1: Entity Framework Core

Box 2: Code first

References:
https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application **Testlet 2**

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. 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 manager their receipts and perform any additional tasks needed for reimbursement.

Receipt processing
Employees may upload receipts in two ways:

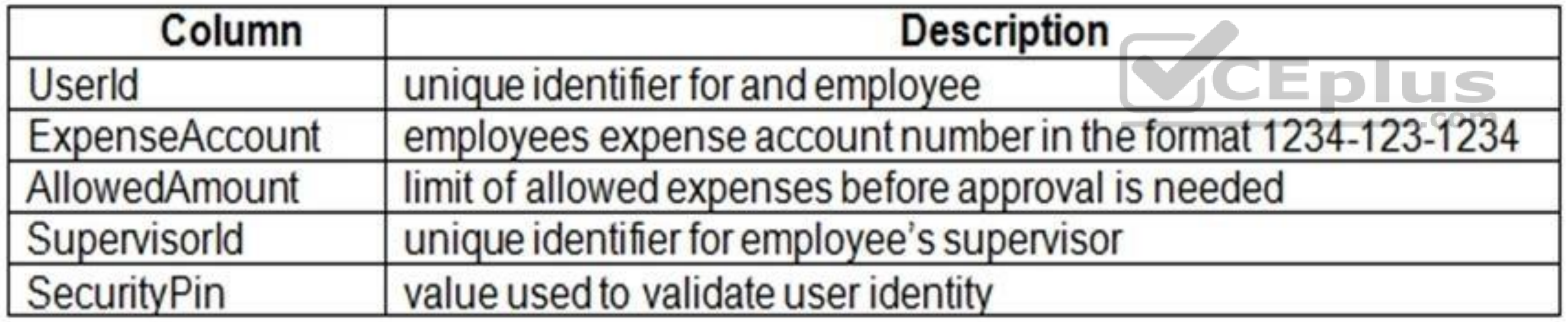
- Uploading using an Azure Files mounted folder
- Uploading using the web application

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.

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. **Requirements**

Receipt processing

Concurrent processing of a receipt must be prevented.

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.

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

▪ Users’ 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 Least Privilege Principal.
▪ 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.

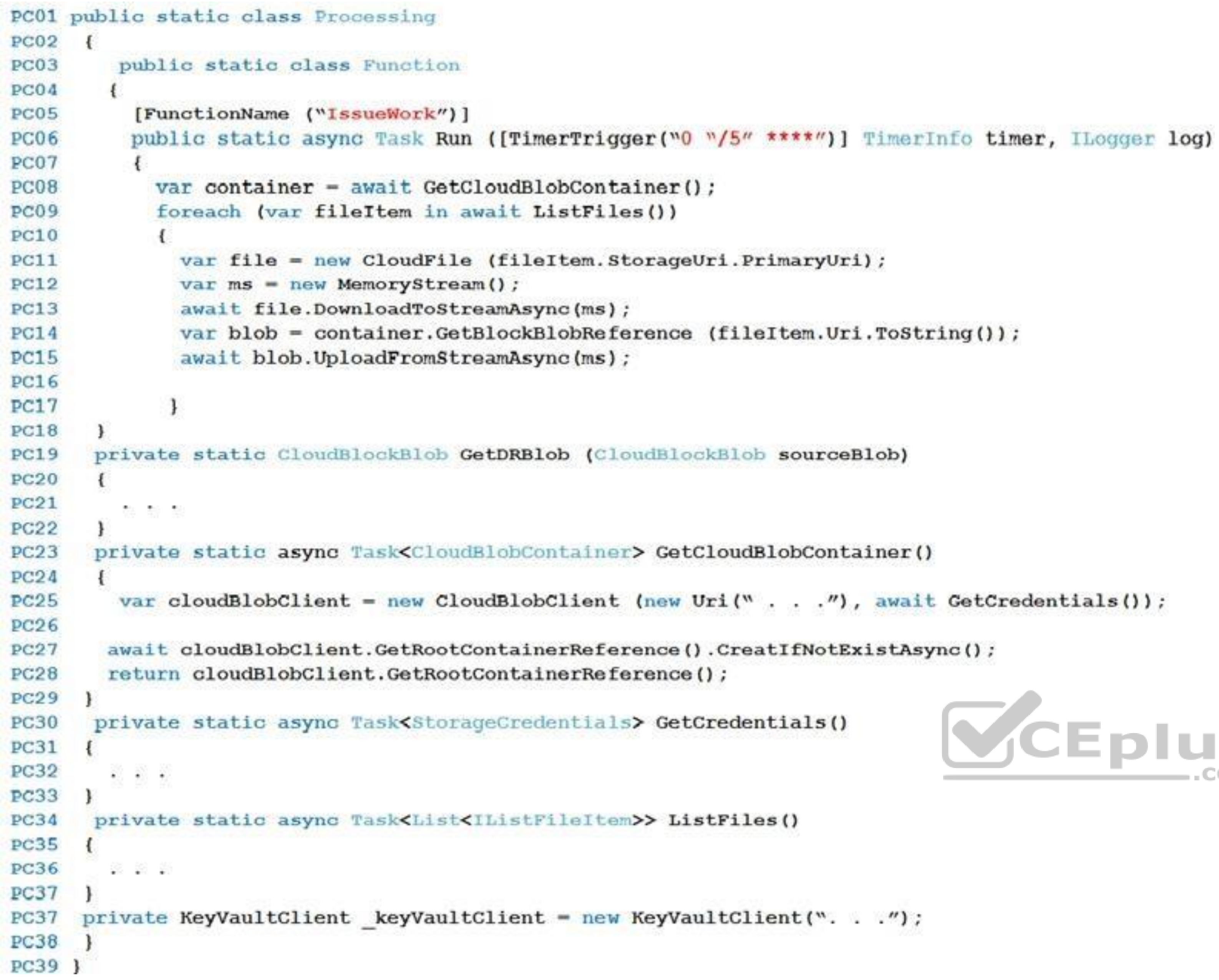
Capacity issue

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

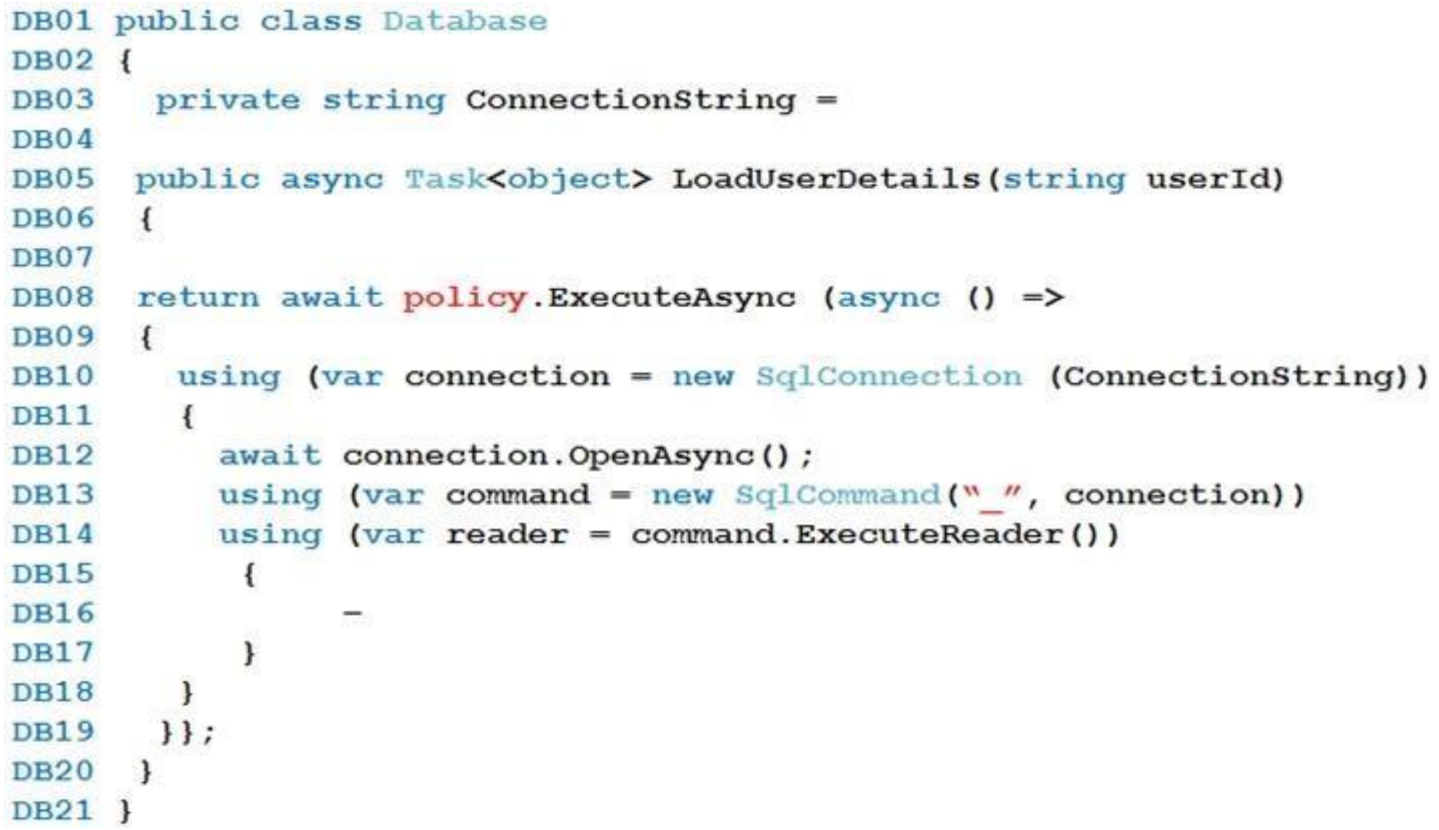
Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Processing.cs



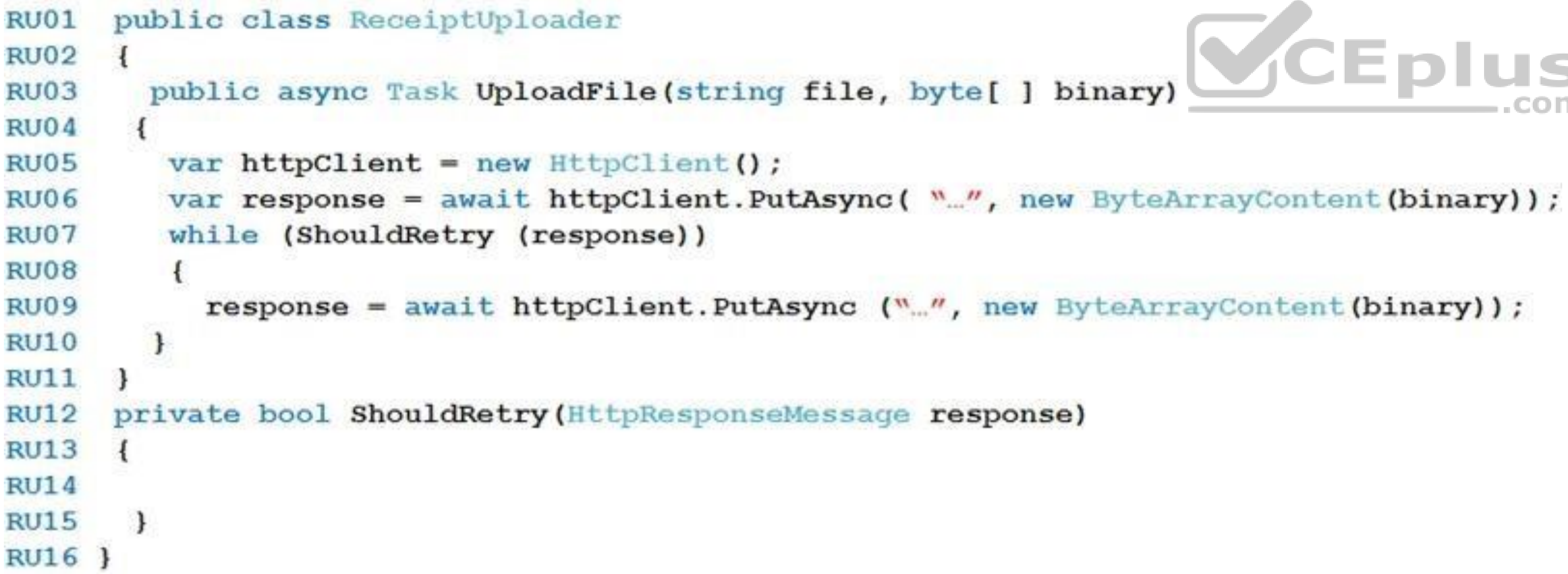
```
PC01 public static class Processing
PC02  {
PC03     public static class Function
PC04    {
PC05      [FunctionName (“IssueWork”)]
PC06      public static async Task Run ([TimerTrigger(“0 “/5” ****”)] TimerInfo timer, ILogger log)
PC07      {
PC08        var container = await GetCloudBlobContainer();
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().CreatIfNotExistAsync();
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



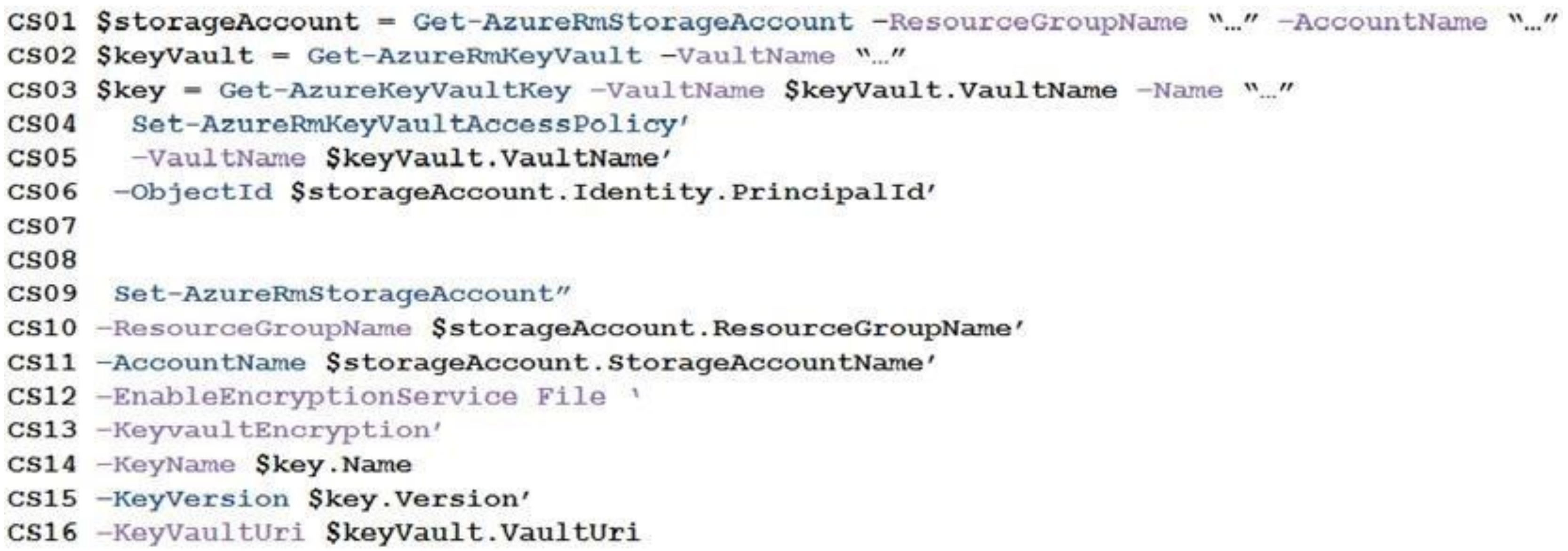
```
DB01 public class Database
DB02 {
DB03   private string ConnectionString =
DB04
DB05  public async Task<object> LoadUserDetails(string userId)
DB06  {
DB07
DB08  return await policy.ExecuteAsync (async () =>
DB09  {
DB10    using (var connection = new SqlConnection (ConnectionString))
DB11    {
DB12      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
```

QUESTION 1
HOTSPOT

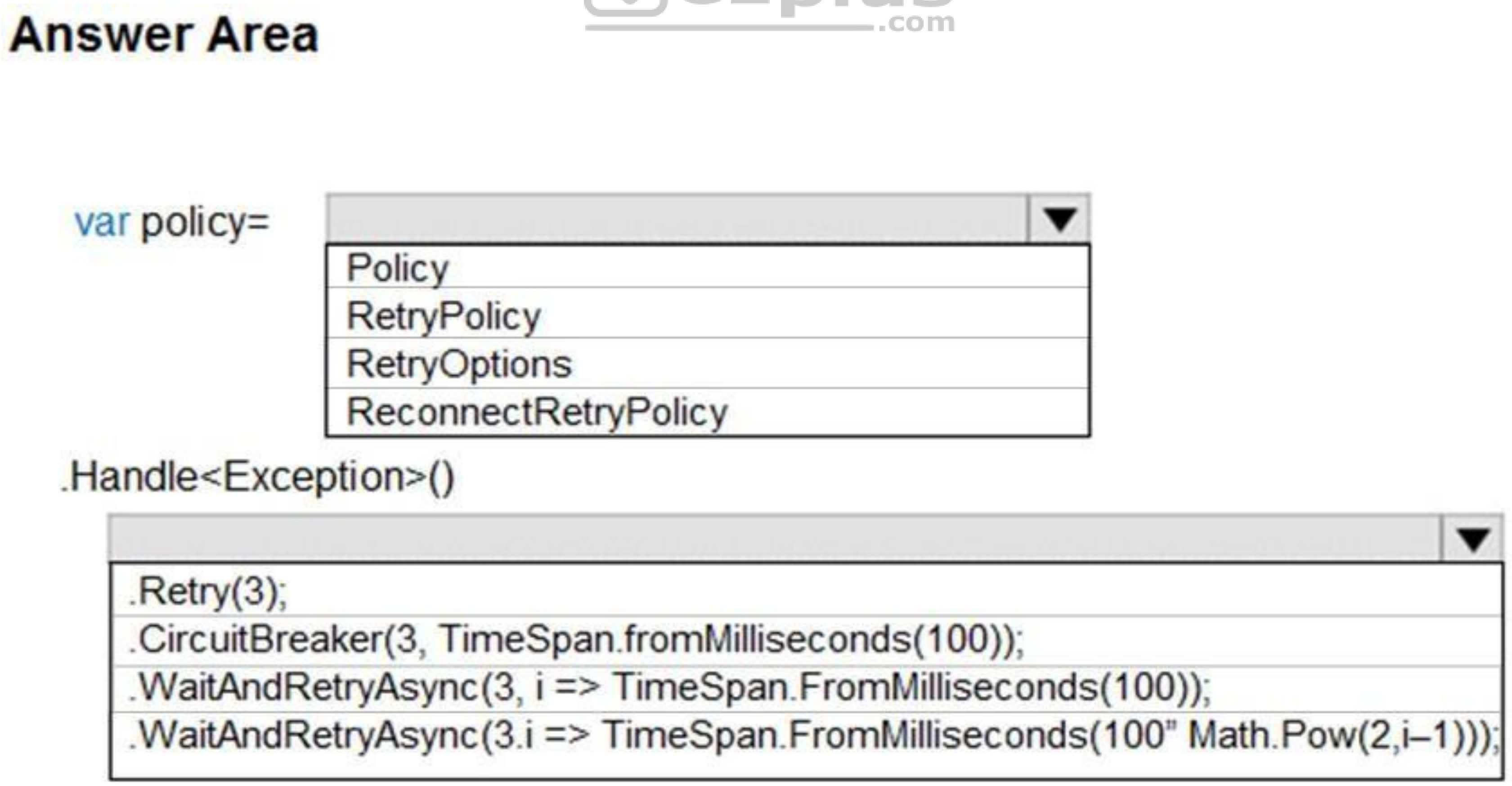
You need to configure retries in the LoadUserDetails function in the Database class without impacting user experience.

What code, should you insert on line DB07?

To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area

var policy= ▼

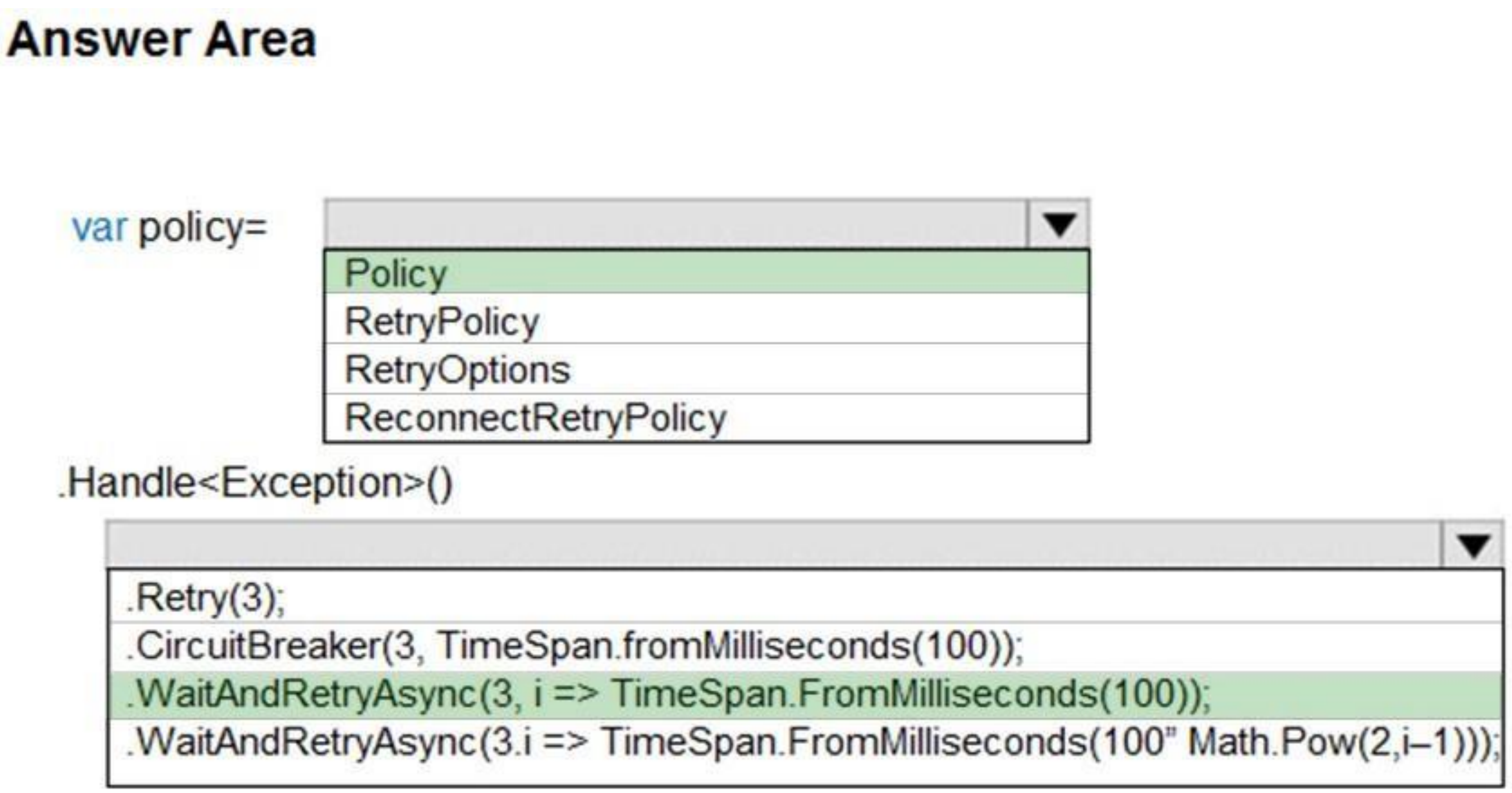
- Policy
- RetryPolicy
- RetryOptions
- ReconnectRetryPolicy

.Handle<Exception>()

▼

- .Retry(3);
- .CircuitBreaker(3, TimeSpan.fromMilliseconds(100));
- .WaitAndRetryAsync(3, i => TimeSpan.FromMilliseconds(100));
- .WaitAndRetryAsync(3.i => TimeSpan.FromMilliseconds(100" Math.Pow(2,i–1)));

Correct Answer:



Answer Area

```
var policy=
```

- Policy
- RetryPolicy
- RetryOptions
- ReconnectRetryPolicy

```
.Handle<Exception>()
```

- .Retry(3);
- .CircuitBreaker(3, TimeSpan.fromMilliseconds(100));
- .WaitAndRetryAsync(3, i => TimeSpan.FromMilliseconds(100));
- .WaitAndRetryAsync(3.i => TimeSpan.FromMilliseconds(100" Math.Pow(2,i–1)));

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: Policy

```
RetryPolicy retry = Policy
  .Handle<HttpRequestException>()
  .Retry(3);
```

The above example will create a retry policy which will retry up to three times if an action fails with an exception handled by the Policy.

Box 2: WaitAndRetryAsync(3,i => TimeSpan.FromMilliseconds(100* Math.Pow(2,i-1)));
A common retry strategy is exponential backoff: this allows for retries to be made initially quickly, but then at progressively longer intervals, to avoid hitting a subsystem with repeated frequent calls if the subsystem may be struggling.

Example:

```
Policy
  .Handle<SomeExceptionType>()
  .WaitAndRetry(3, retryAttempt =>
    TimeSpan.FromSeconds(Math.Pow(2, retryAttempt))
);
```

References:
https://github.com/App-vNext/Polly/wiki/Retry

Question Set 3

QUESTION 1
DRAG DROP

You develop a web app that uses the tier D1 app service plan by using the Web Apps feature of Microsoft Azure App Service.

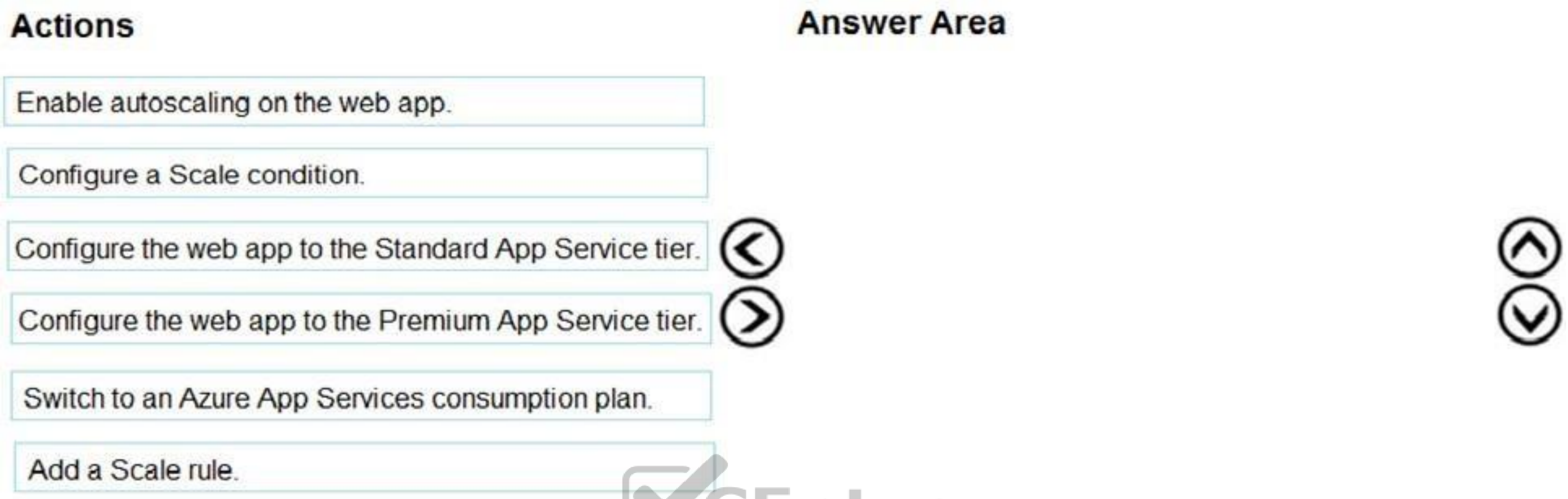
Spikes in traffic have caused increases in page load times.

You need to ensure that the web app automatically scales when CPU load is about 85 percent and minimize costs.

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

NOTE: Each correct selection is worth one point.

Select and Place:

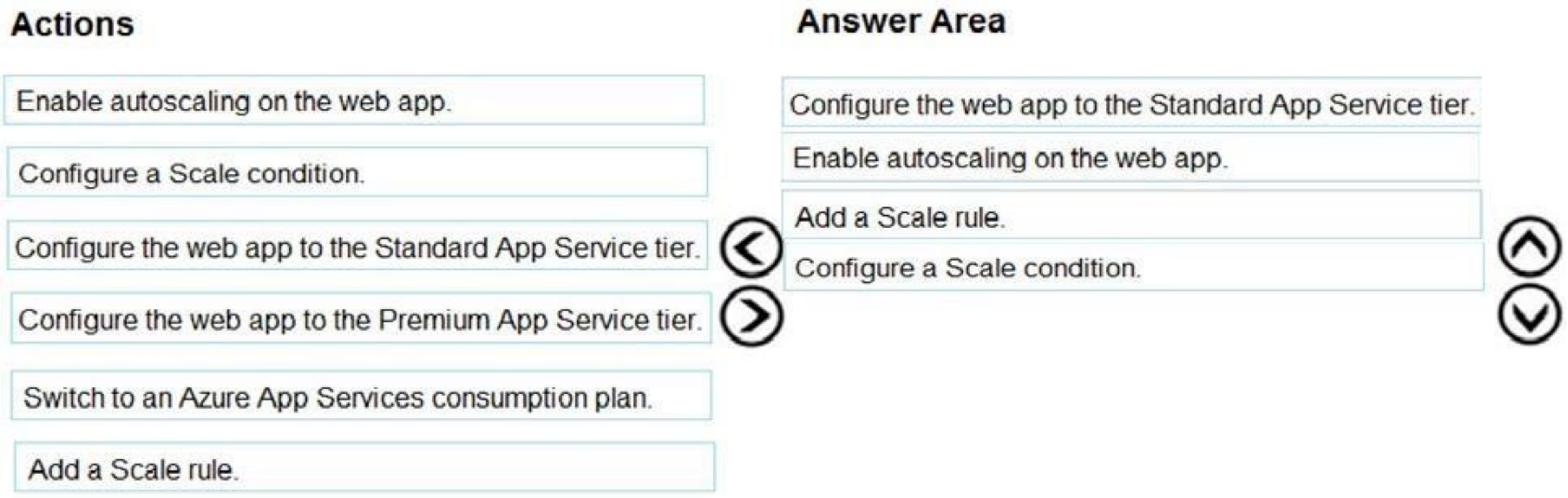


Actions

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

Answer Area

Correct Answer:



Actions

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

Answer Area

- Configure the web app to the Standard App Service tier.
- Enable autoscaling on the web app.
- Add a Scale rule.
- Configure a Scale condition.

Section: [none]
Explanation
Explanation/Reference:
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 condidation

References: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-autoscale-get-started https://azure.microsoft.com/en-us/pricing/details/app-service/plans/

QUESTION 2
DRAG DROP

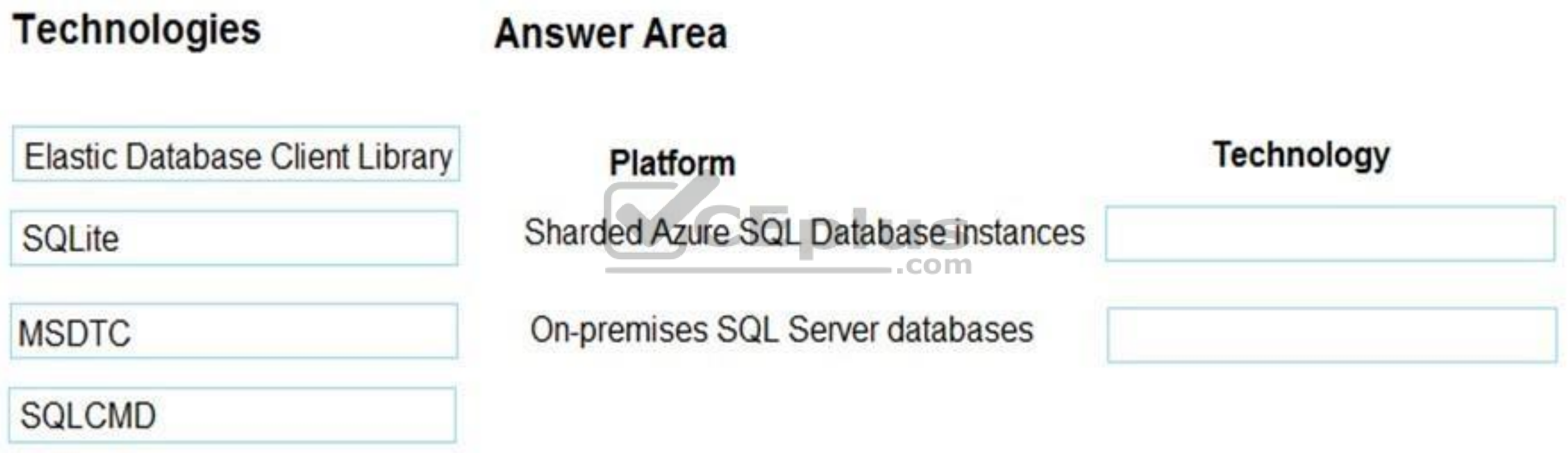
You are developing a multi-tenant ASP.NET Core application that will be hosted on Azure. The application will support multiple database platforms, including Azure SQL and on-premises SQL Server instances.

You need to ensure the application supports distributed transactions.

Which technologies should you use? To answer, drag the appropriate technologies to the correct scenarios. Each technology 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:



Technologies

Elastic Database Client Library

SQLite

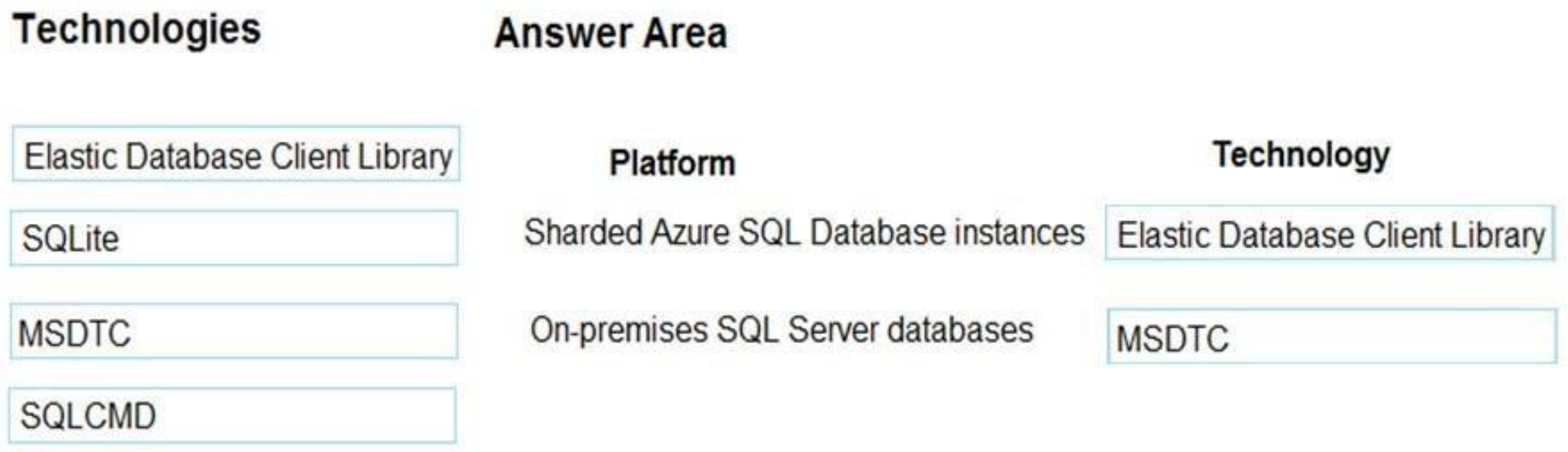
MSDTC

SQLCMD

Answer Area

Platform	Technology
Sharded Azure SQL Database instances	
On-premises SQL Server databases	

Correct Answer:



Technologies

Elastic Database Client Library

SQLite

MSDTC

SQLCMD

Answer Area

Platform	Technology
Sharded Azure SQL Database instances	Elastic Database Client Library
On-premises SQL Server databases	MSDTC

Section: [none]
Explanation

Explanation/Reference:

Explanation:

Box 1: Elastic Database Client Library

Sharded database applications in Azure: With this scenario, the data tier uses the Elastic Database client library or self-sharding to horizontally partition the data across many databases in SQL DB.

Box 2: MSDTC

Eelastic database transactions cannot stretch across on premises SQL Server and Azure SQL Database. For distributed transactions on premises, continue to use MSDTC.

References: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-transactions-overview

QUESTION 3

You have implemented code that uses elastic transactions spanning across three different Azure SQL Database logical servers.

Database administration report that some transactions take longer to complete than expected.

You need to use the correct tool to monitor all the transactions originating from the elastic transaction implementation.

Which tool should you use?

A. Use the dependencies section of Azure Applications Insights.
B. Run the sys.dm_tran_active_transactions dynamic management view.
C. Run the sys.dm_tran_current_snapshot dynamic management view.
D. Run the sys.dm_tran_active_snapshot_database_transactions dynamic management view.

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:

Explanation:

Use Dynamic Management Views (DMVs) in SQL DB to monitor status and progress of your ongoing elastic database transactions.

These DMVs are particularly useful:

sys.dm_tran_active_transactions: Lists currently active transactions and their status. The UOW (Unit Of Work) column can identify the different child transactions that belong to the same distributed transaction. All transactions within the same distributed transaction carry the same UOW value. sys.dm_tran_database_transactions: Provides additional information about transactions, such as placement of the transaction in the log. sys.dm_tran_locks: Provides information about the locks that are currently held by ongoing transactions

References: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-transactions-overview **QUESTION 4** A company is migrating an existing on-premises third-party website to Azure. The website is stateless.

The company does not have access to the source code for the website. They do have the original installer.

The number of visitors at the website varies throughout the year. The on-premises infrastructure was resized to accommodate peaks but the extra capacity was not used.

You need to implement a virtual machine scale set instance.

What should you do?

A. Use an autoscale setting to scale instances vertically.
B. Use an autoscale setting with unlimited maximum number of instances.
C. Scale out by one instance when the average CPU usage of one of the instance is over 80 percent.
D. Use Azure Monitor to create autoscale settings using custom metrics.

Correct Answer: D
Section: [none]
Explanation

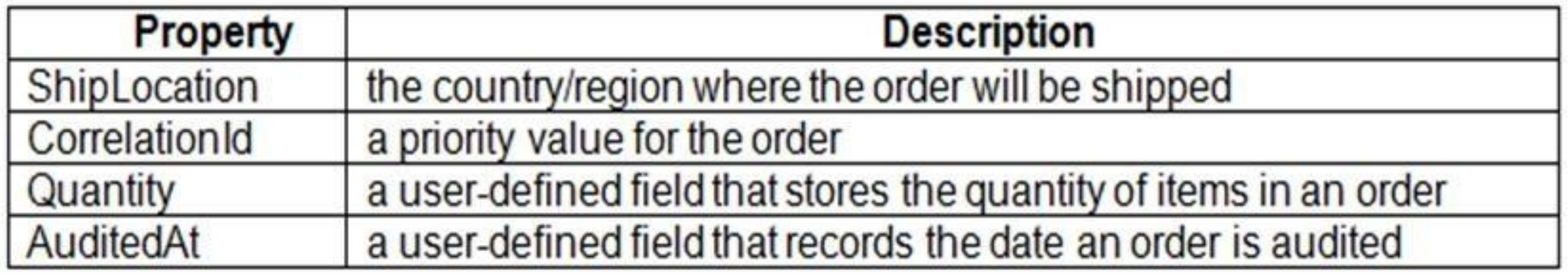
Explanation/Reference:

Explanation:
Azure Monitor autoscaling allows you to scale the number of running instances up or down, based on telemetry data (metrics).

References: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-autoscale-common-metrics

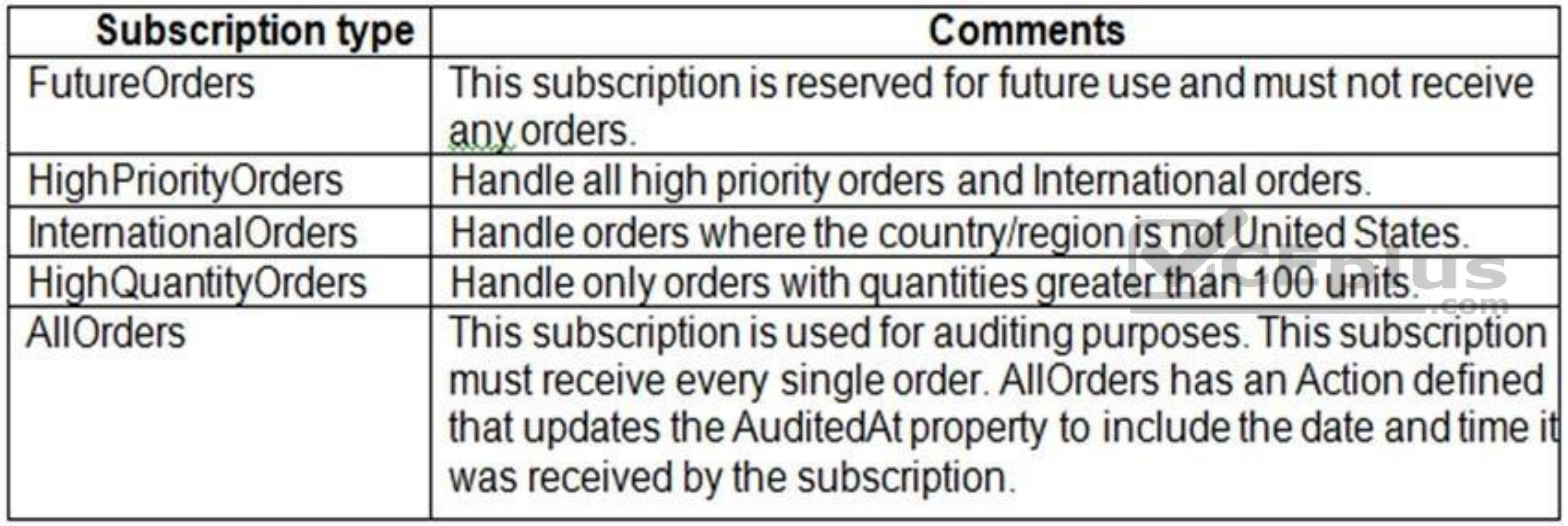
QUESTION 5
DRAG DROP

You are implementing an order processing system. A point of sale application publishes orders to topics in an Azure Service Bus queue. The label property for the topic includes the following data:



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

The system has the following requirements for subscriptions:



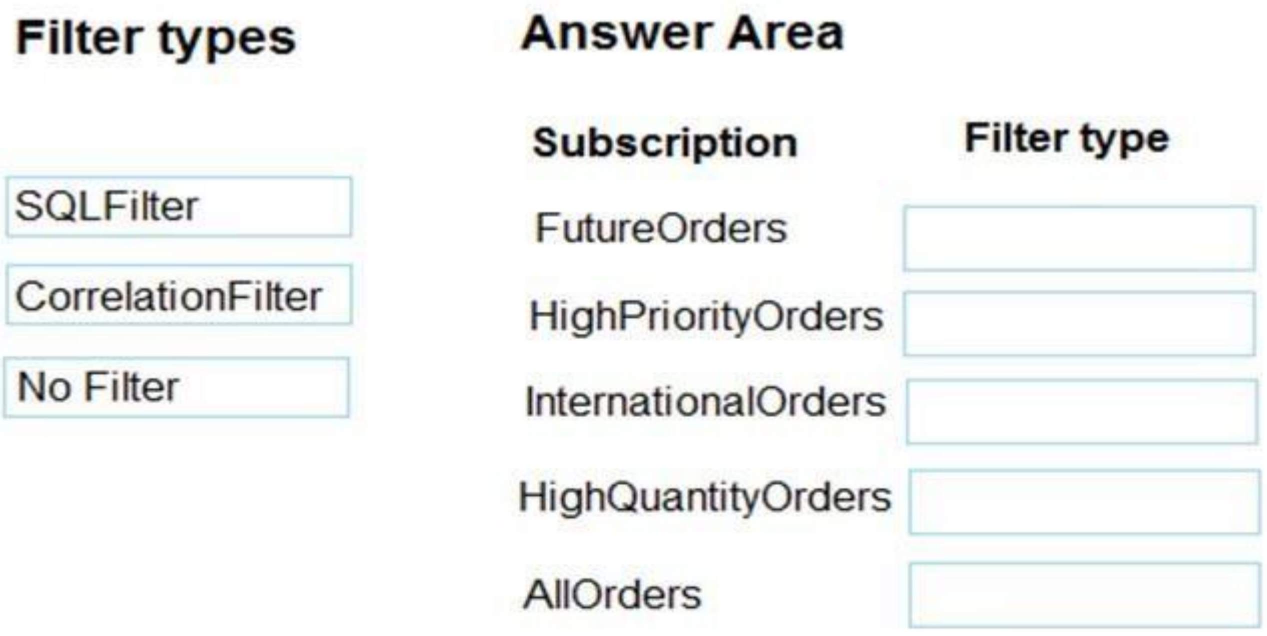
Subscription type	Comments
FutureOrders	This subscription is reserved for future use and must not receive any orders.
HighPriorityOrders	Handle all high priority orders and International orders.
InternationalOrders	Handle orders where the country/region is not United States.
HighQuantityOrders	Handle 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.

You need to implement filtering and maximize throughput while evaluating filters.

Which filter types should you implement? To answer, drag the appropriate filter types to the correct subscriptions. Each filter type 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:



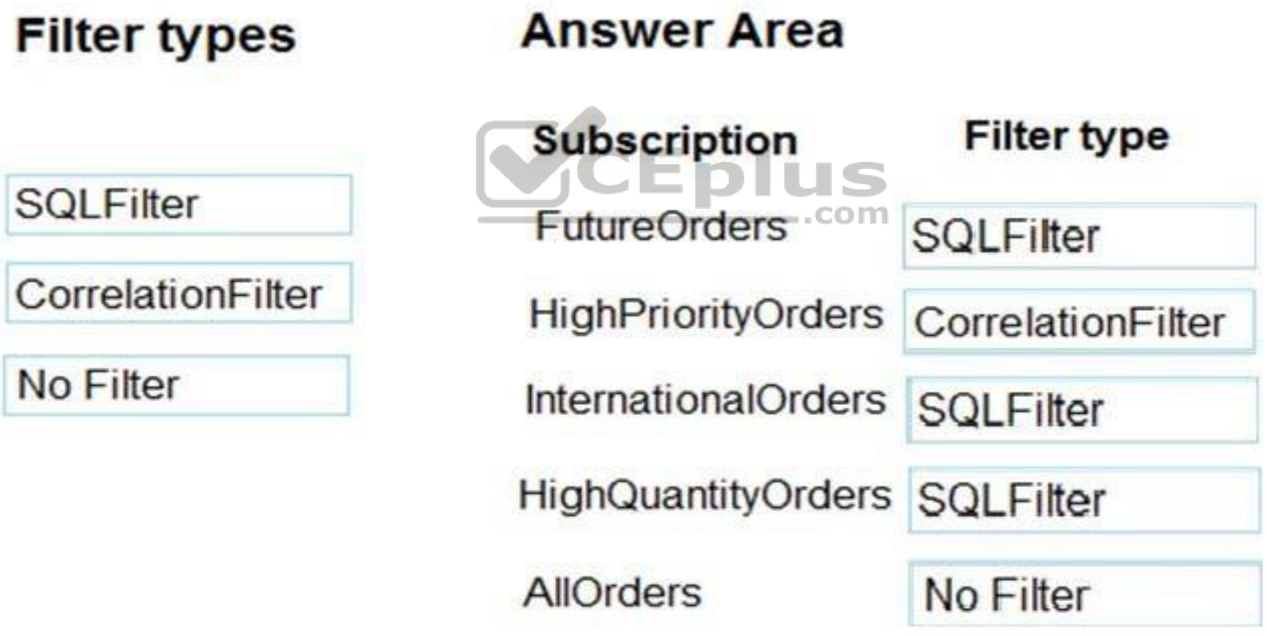
Filter types

- SQLFilter
- CorrelationFilter
- No Filter

Answer Area

Subscription	Filter type
FutureOrders	
HighPriorityOrders	
InternationalOrders	
HighQuantityOrders	
AllOrders	

Correct Answer:



Filter types

- SQLFilter
- CorrelationFilter
- No Filter

Answer Area

Subscription	Filter type
FutureOrders	SQLFilter
HighPriorityOrders	CorrelationFilter
InternationalOrders	SQLFilter
HighQuantityOrders	SQLFilter
AllOrders	No Filter

Section: [none]
Explanation

Explanation/Reference:
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

QUESTION 6
You are developing an Azure Batch solution to perform CPU intensive calculations. The calculations occur at a specific time each week and last for approximately one hour.

Before any changes are made, a timer must be created to measure the initial duration. The timer must start before the first calculation is queued to run on the computer node.

You need to implement the timer.

Before which line should the timer be created?

A. CloudJob = batchClient.JobOperations.CreateJob();
B. batchClient.JobOperations.AddTask(JobId, tasks);
C. batchClient.Utilities.CreateTaskStateMonitor().WaitAll(addedTasks, TaskState.Completed, timeout);
D. using (BatchClient batchClient = BatchClient.Open(cred))

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
References:
https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet

Testlet 1

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

Best for You Organics Company is a global restaurant franchise that has multiple locations. The company wants to enhance user experiences and vendor integrations. The company plans to implement automated mobile ordering and delivery services.

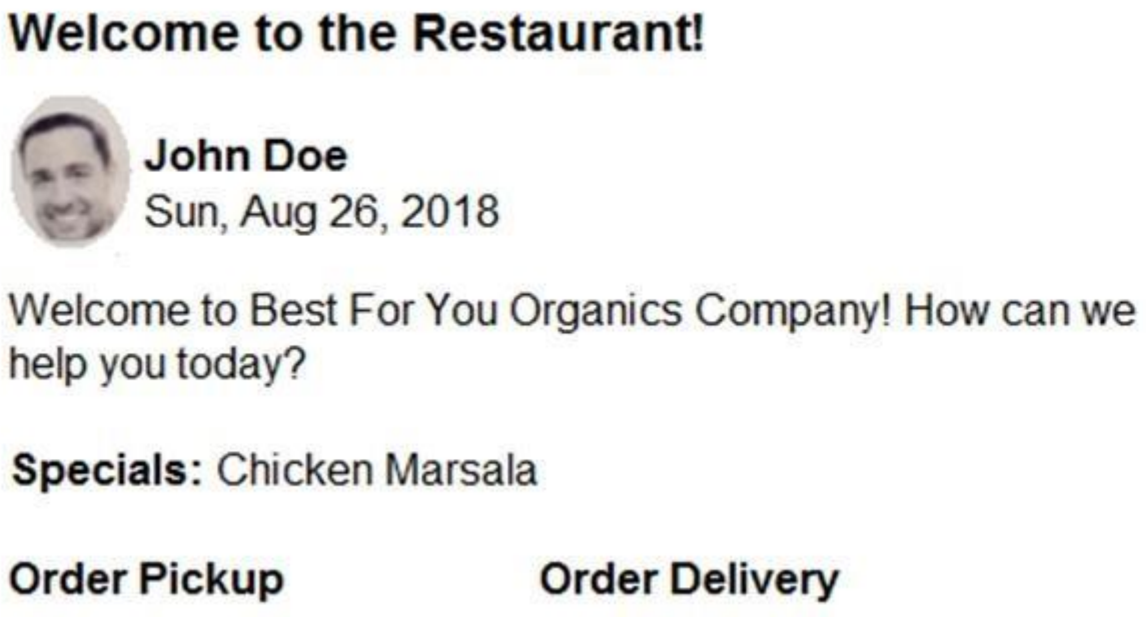
Best For You Organics hosts an Azure web app at the URL https://www.bestforyouorganics.com.
Users can use the web app to browse restaurant locations, menu items, nutritional, information, and company information. The company developed and deployed a cross-platform mobile app.

Requirements

You must develop a chartbot by using the Bot Builder SDK and Language Understanding Intelligence Service (LUIS). The chatbot must allow users to order food for pickup or delivery.

The chatbot must meet the following requirements:

▪ Ensure that chatbot endpoint can be accessed only by the Bot Framework connector.
▪ Use natural language processing and speech recognition so that users can interact with the chatbot by using text and voice. Processing must be server-based. ▪
Alert users about promotions at local restaurants.
▪ Enable users to place an order for delivery or pickup by using their voice.
▪ Greet the user upon sign-in by displaying a graphical interface that contains action buttons. ▪
The chatbot greeting interface must match the formatting of the following example:



Welcome to the Restaurant!

John Doe
Sun, Aug 26, 2018

Welcome to Best For You Organics Company! How can we help you today?

Specials: Chicken Marsala

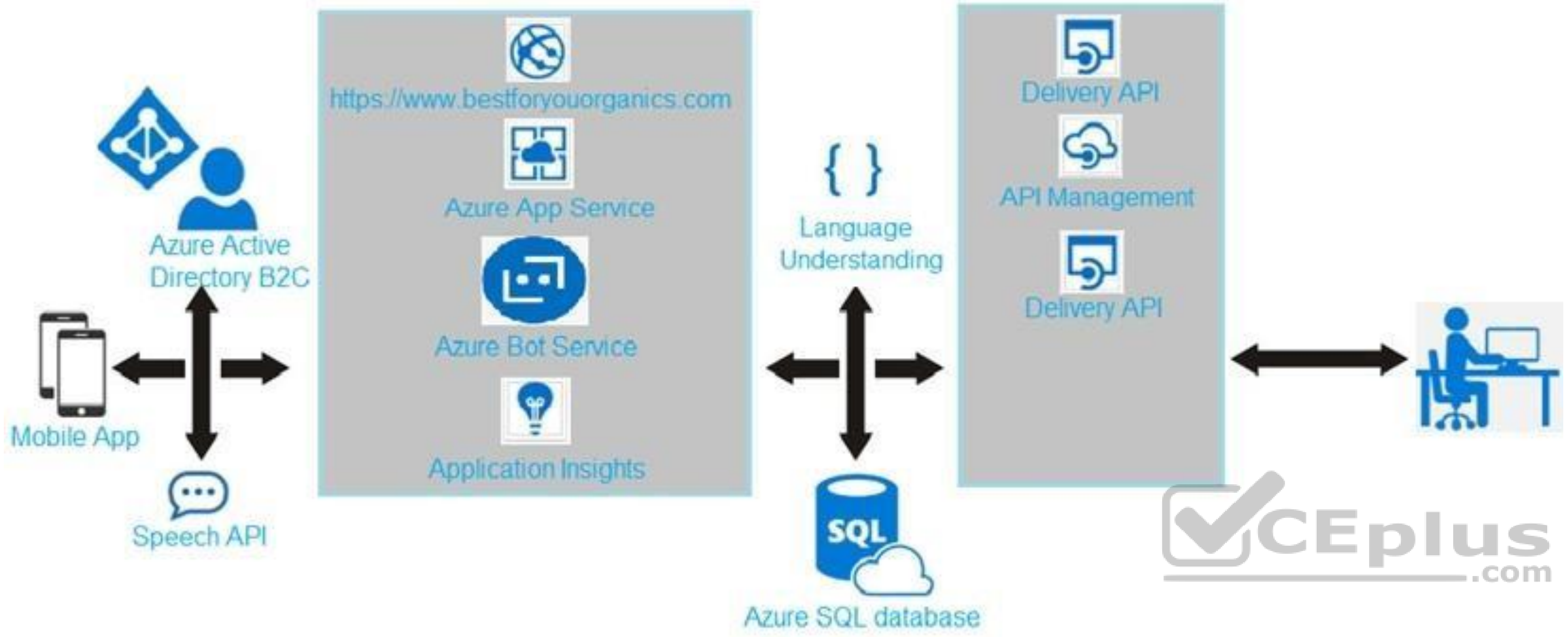
Order Pickup **Order Delivery**

Vendor API
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

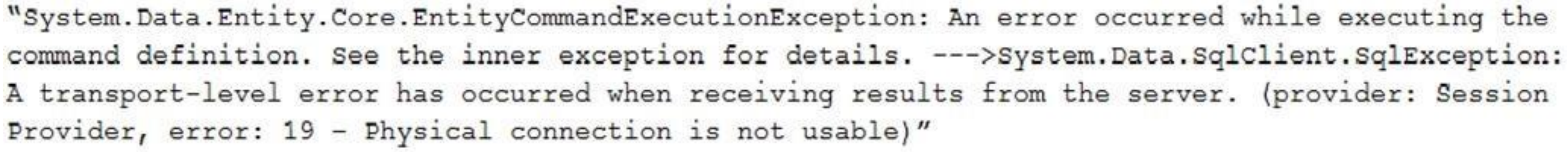
- API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.
- If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.
- APIs must prevent API usage spikes on a per-subscription basis by limiting the call rate to 100 calls per minute.
- The Inventory API must be written by using ASP.NET Core and Node.js.
- The API must be updated to provide an interface to Azure SQL Database. Database objects must be managed by using code.
- The Delivery API must be protected by using the OAuth 2.0 protocol with Azure Active Directory (Azure AD) when called from the Azure web app. You register the Delivery API and web app in Azure AD. You enable OAuth 2.0 in the web app.
- The delivery API must update the Products table, the Vendor transactions table, and the Billing table in a single transaction.

The Best For You Organics Company architecture team has created the following diagram depicting the expected deployments into Azure:



Delivery API

The Delivery API intermittently throws the following exception:



```
"System.Data.Entity.Core.EntityCommandExecutionException: An error occurred while executing the
command definition. See the inner exception for details. --->System.Data.SqlClient.SqlException:
A transport-level error has occurred when receiving results from the server. (provider: Session
Provider, error: 19 - Physical connection is not usable)"
```

Chatbot greeting

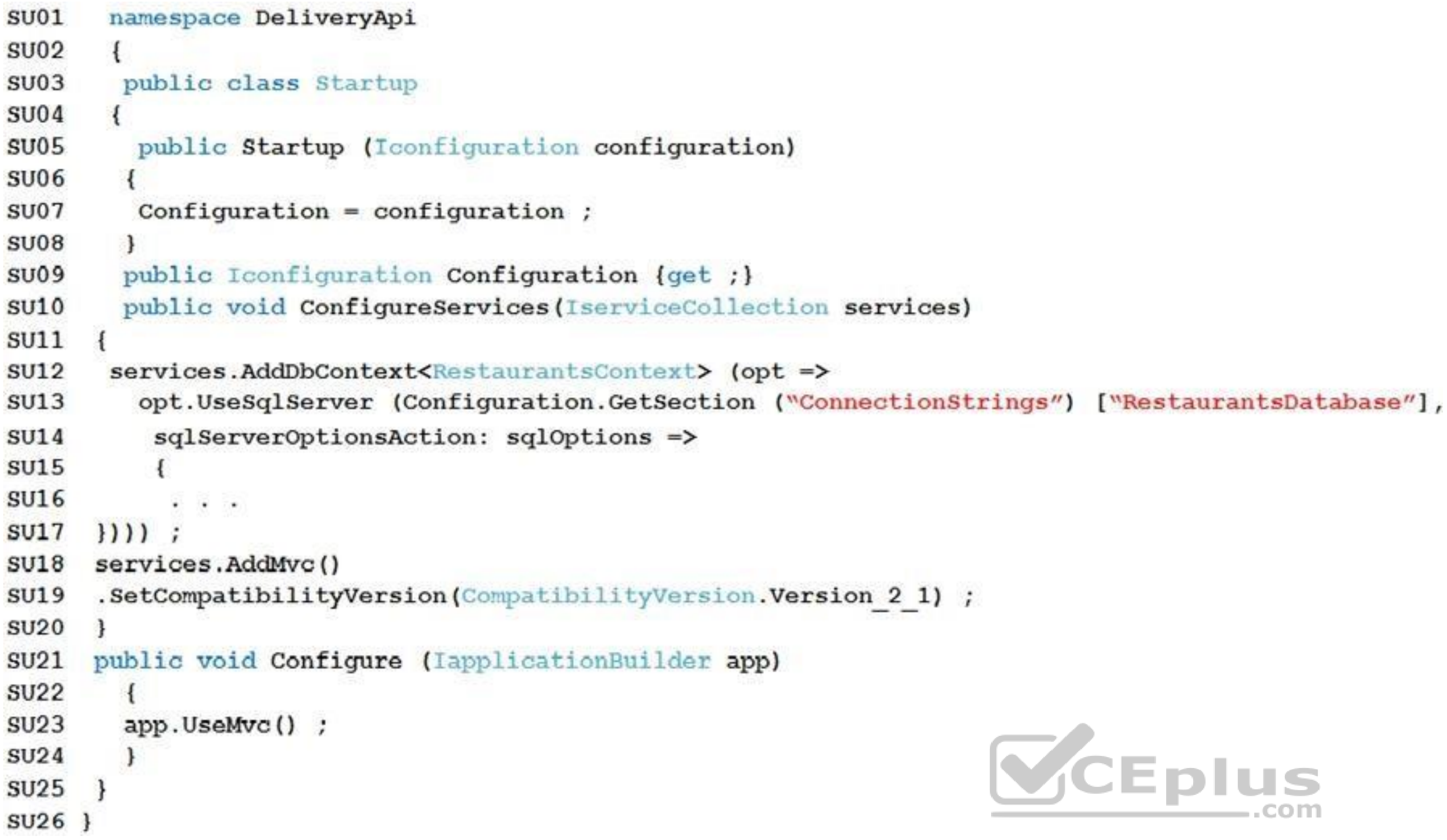
The chatbot's greeting does not show the user's name. You need to debug the chatbot locally.

Language processing

Users report that the bot fails to understand when a customer attempts to order dishes that use Italian names.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

Startup.cs



```
namespace DeliveryApi
{
 public class Startup
{
   public Startup (Iconfiguration configuration)
  {
   Configuration = configuration ;
  }
 public Iconfiguration Configuration {get ;}
 public void ConfigureServices(IserviceCollection services)
{
services.AddDbContext<RestaurantsContext> (opt =>
  opt.UseSqlServer (Configuration.GetSection ("ConnectionStrings") ["RestaurantsDatabase"],
    sqlServerOptionsAction: sqlOptions =>
    {
      . . .
})));
services.AddMvc()
.SetCompatibilityVersion(CompatibilityVersion.Version_2_1) ;
}
public void Configure (IapplicationBuilder app)
  {
  app.UseMvc() ;
  }
}
}
```

QUESTION 1

Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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 vendor notification requirement.

Solution: Create and apply a custom outbound Azure API Management policy.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario:
If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.

(API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.)

In Azure API Management (APIM), policies are a powerful capability of the system that allow the publisher to change the behavior of the API through configuration. Policies are a collection of Statements that are executed sequentially on the request or response of an API. Popular Statements include format conversion from XML to JSON and call rate limiting to restrict the amount of incoming calls from a developer. Many more policies are available out of the box.

References: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-policies

QUESTION 2
Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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 vendor notification requirement.

Solution: Update the Delivery API to send emails by using a Microsoft Office 365 SMTP server.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Use a custom outbound Azure API Management policy.

Scenario:
If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor. (API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.)

References: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-policies

QUESTION 3
Note: In this section you will see one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem, and you must determine whether the solution meets the stated goals. More than one solution might solve the problem. It is also possible that none of the solutions solve the problem.

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

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 vendor notification requirement.

Solution: Configure notifications in the Azure API Management instance.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation
Explanation/Reference:

Explanation:
Use a custom outbound Azure API Management policy.

Scenario:
If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor. (API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.)

References: https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-policies

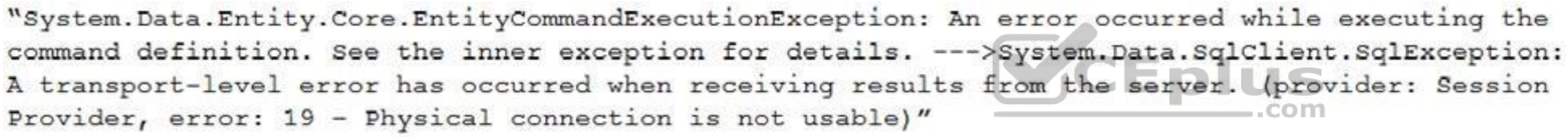
QUESTION 4 You need to resolve the delivery API error.

What should you do?

A. Implement simple retry by using the EnableRetryOnFailure feature of Entity Framework.
B. Implement exponential backoff by using the EnableRetryOnFailure feature of Entity Framework.
C. Implement the Circuit Breaker pattern by using the EnbaleRetryOnFailure feature of Entity Framework.
D. Invoke accustom execution strategy in Entity Framework.

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The Delivery API intermittently throws the following exception:



```
"System.Data.Entity.Core.EntityCommandExecutionException: An error occurred while executing the
command definition. See the inner exception for details. --->System.Data.SqlClient.SqlException:
A transport-level error has occurred when receiving results from the server. (provider: Session
Provider, error: 19 - Physical connection is not usable)"
```

A useful method to get rid of this error is to use RETRY LOGIC of Entity Framework 1.1.0

```
services.AddDbContext<DbContext>(options => options.UseSqlServer('yourconnectionstring',
…sqlServerOptionsAction: sqlOptions =>
…{
……sqlOptions.EnableRetryOnFailure(
………maxRetryCount: 5,
………maxRetryDelay: TimeSpan.FromSeconds(30),
………errorNumbersToAdd: new List<int>() { 19 });
…}));
```

In Retry logic, error 19 is not included. So you have to pass the error code 19 to set retry logic for error code 19.

References: https://stackoverflow.com/questions/47558062/error-19-physical-connection-error/47559967 **Testlet 2**

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

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.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

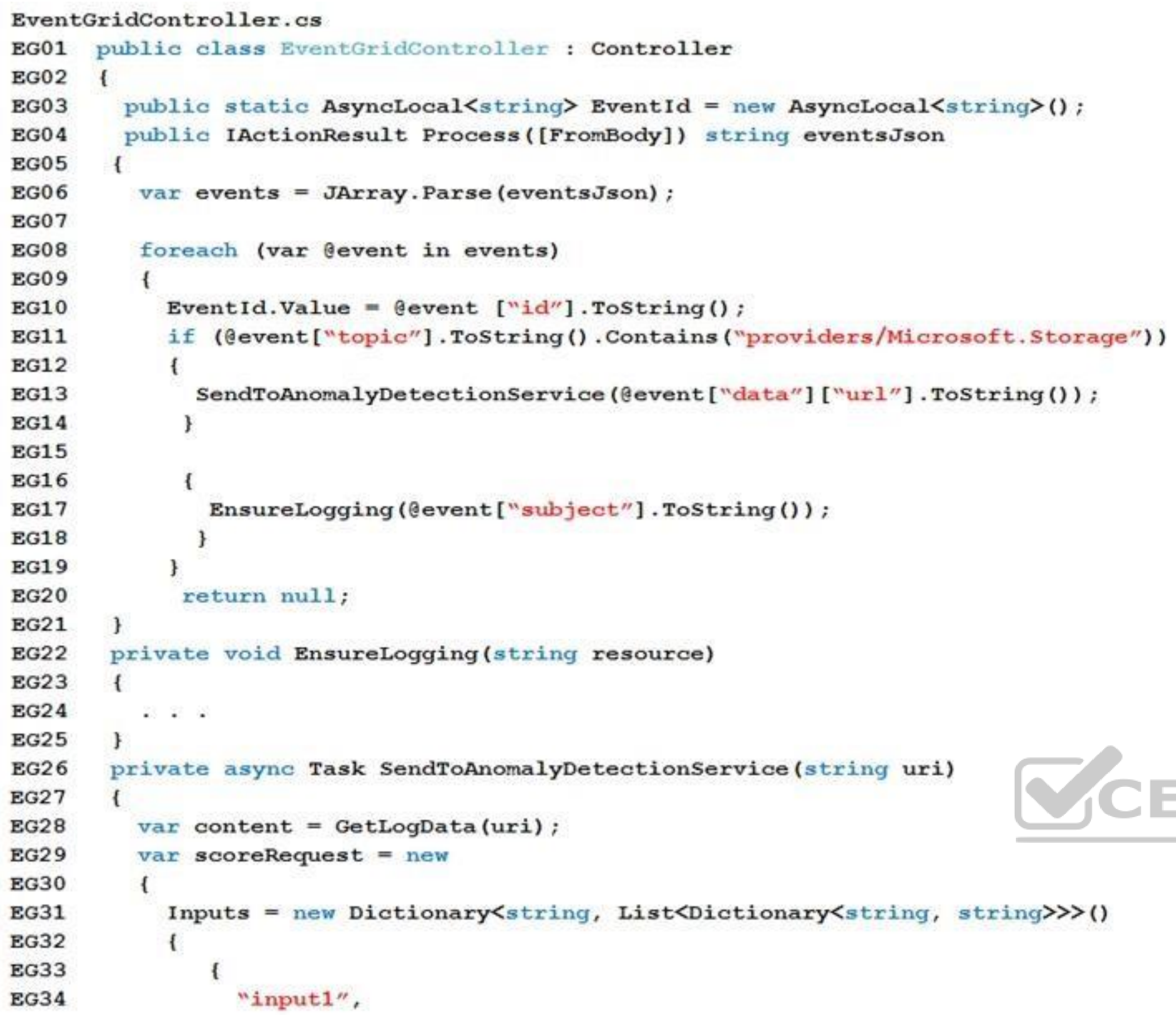
Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

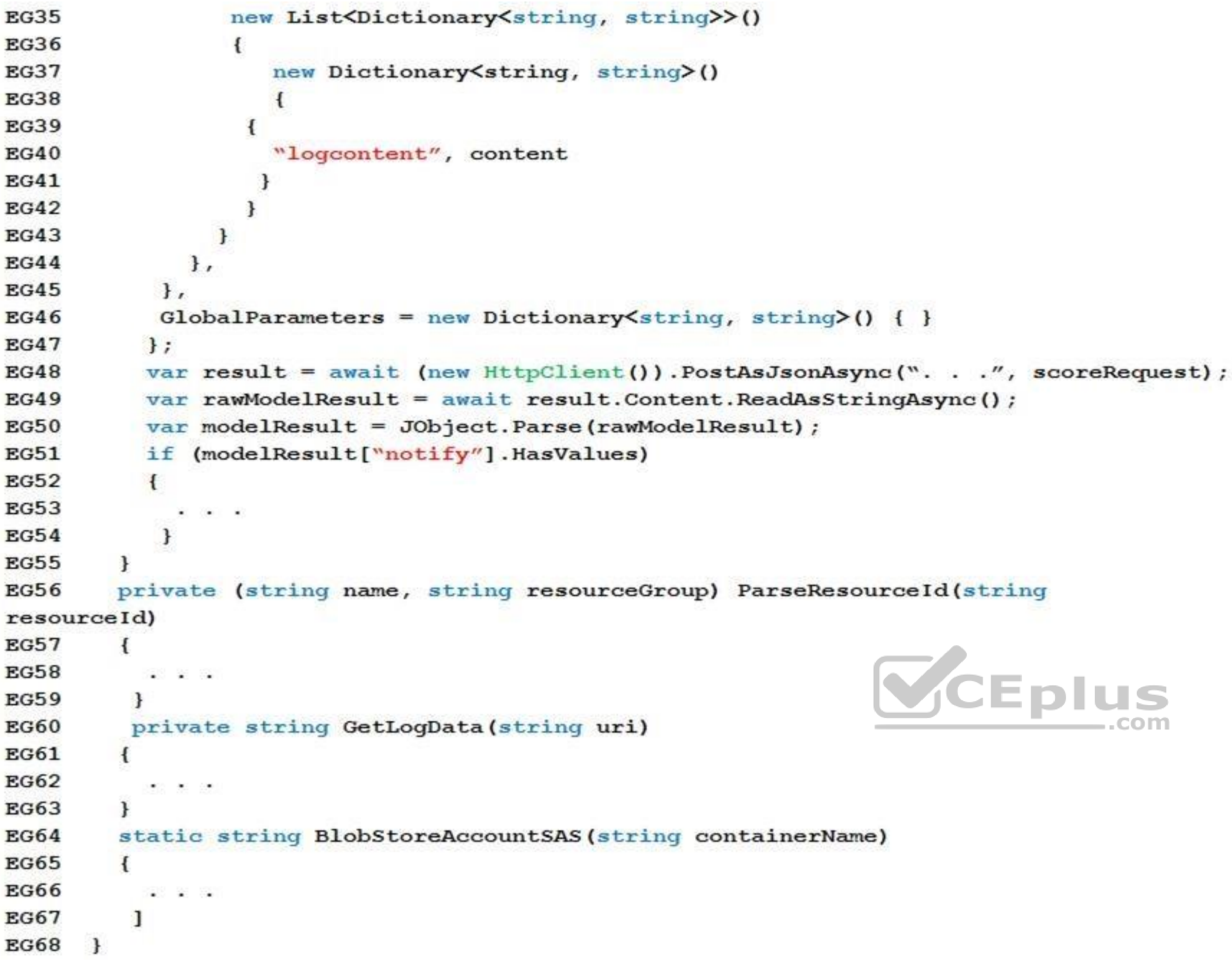
Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

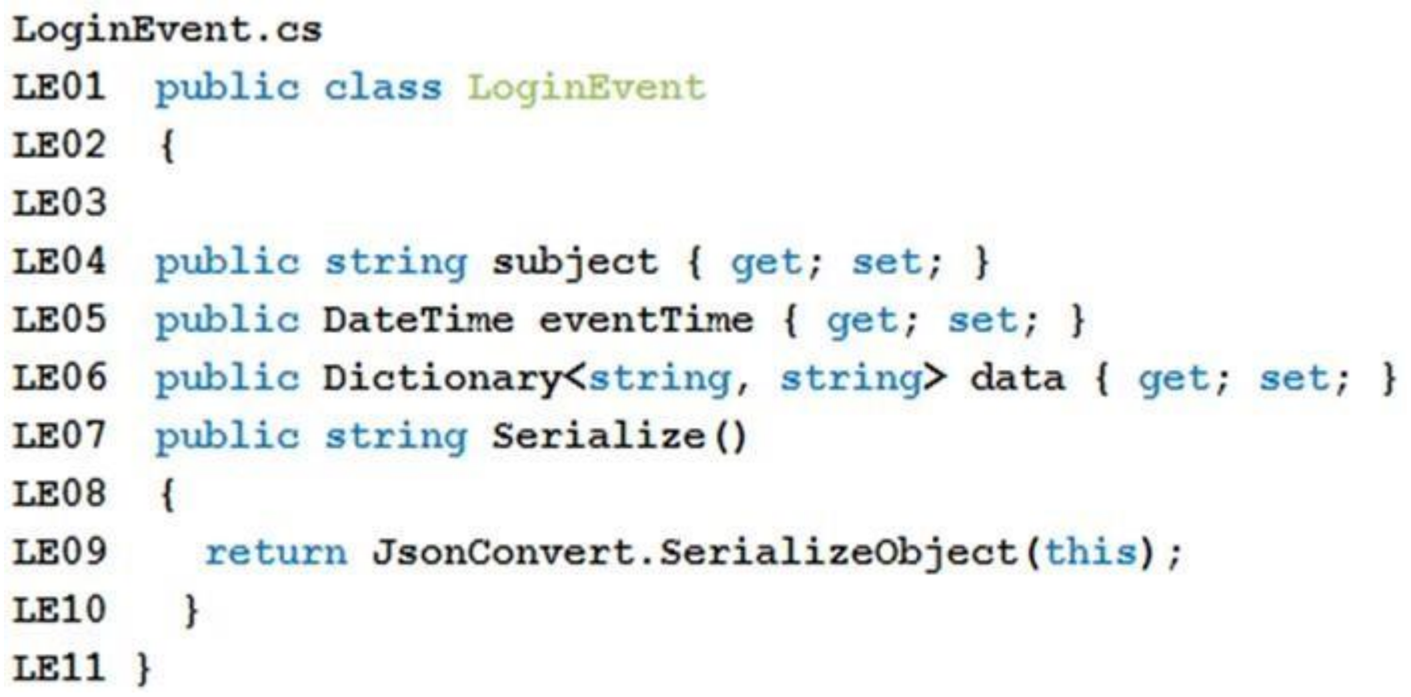


```
EventGridController.cs
EG01 public class EventGridController : Controller
EG02 {
EG03   public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04   public IActionResult Process([FromBody]) string eventsJson
EG05  {
EG06    var events = JArray.Parse(eventsJson);
EG07
EG08    foreach (var @event in events)
EG09    {
EG10      EventId.Value = @event ["id"].ToString();
EG11      if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12      {
EG13        SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14       }
EG15
EG16       {
EG17         EnsureLogging(@event["subject"].ToString());
EG18        }
EG19      }
EG20       return null;
EG21  }
EG22  private void EnsureLogging(string resource)
EG23  {
EG24    . . .
EG25  }
EG26  private async Task SendToAnomalyDetectionService(string uri)
EG27  {
EG28    var content = GetLogData(uri);
EG29    var scoreRequest = new
EG30    {
EG31      Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32      {
EG33          {
EG34            "input1",
```



```
EG35           new List<Dictionary<string, string>>()
EG36           {
EG37              new Dictionary<string, string>()
EG38              {
EG39            {
EG40              “logcontent”, content
EG41             }
EG42            }
EG43          }
EG44        },
EG45      },
EG46      GlobalParameters = new Dictionary<string, string>() { }
EG47     };
EG48     var result = await (new HttpClient()).PostAsJsonAsync(“. . .”, scoreRequest);
EG49     var rawModelResult = await result.Content.ReadAsStringAsync();
EG50     var modelResult = JObject.Parse(rawModelResult);
EG51     if (modelResult[“notify”].HasValues)
EG52     {
EG53       . . .
EG54      }
EG55   }
EG56   private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57   {
EG58     . . .
EG59    }
EG60    private string GetLogData(string uri)
EG61   {
EG62     . . .
EG63   }
EG64   static string BlobStoreAccountSAS(string containerName)
EG65   {
EG66     . . .
EG67    ]
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.



```
LoginEvent.cs
LE01  public class LoginEvent
LE02  {
LE03
LE04  public string subject { get; set; }
LE05  public DateTime eventTime { get; set; }
LE06  public Dictionary<string, string> data { get; set; }
LE07  public string Serialize()
LE08  {
LE09    return JsonConvert.SerializeObject(this);
LE10    }
LE11 }
```

QUESTION 1 You need to resolve a notification latency issue.

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

NOTE: Each correct selection is worth one point.

A. Ensure that the Azure Function is set to use a consumption plan.
B. Ensure that the Azure Function is using an App Service plan.
C. Set Always On to **false**.
D. Set Always On to **true**.

Correct Answer: BD
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Azure Functions can run on either a Consumption Plan or a dedicated App Service Plan. If you run in a dedicated mode, you need to turn on the Always On setting for your Function App to run properly. The Function runtime will go idle after a few minutes of inactivity, so only HTTP triggers will actually "wake up" your functions. This is similar to how WebJobs must have Always On enabled.

Scenario: Notification latency: Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Anomaly detection service: You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

References: https://github.com/Azure/Azure-Functions/wiki/Enable-Always-On-when-running-on-dedicated-App-Service-Plan **Question Set 3**

QUESTION 1
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 develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

- Routing Service – Routes a request to the appropriate service and must not persist data across sessions
- Account Service – Stores and manages all account information and authentication and requires data to persist across sessions.
- User Service – Stores and manages all user information and requires data to persist across sessions.
- Housing Network Service – Stores and manages the current real-estate economy and requires data to persist across sessions. - Trade Service – Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Create a Service Fabric Cluster with a stateless Reliable Service for Routing Service. Create stateful Reliable Services for all other components.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
We should use stateful services when we want data to persist, and stateless service when data must not persist.

Note:
For stateful services, the Reliable Services programming model allows you to consistently and reliably store your state right inside your service by using Reliable Collections.

A stateless service is one where there is no state maintained within the service across calls. Any state that is present is entirely disposable and doesn't require synchronization, replication, persistence, or high availability.

References: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction

QUESTION 2
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 develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

- Routing Service – Routes a request to the appropriate service and must not persist data across sessions
- Account Service – Stores and manages all account information and authentication and requires data to persist across sessions.
- User Service – Stores and manages all user information and requires data to persist across sessions.
- Housing Network Service – Stores and manages the current real-estate economy and requires data to persist across sessions. - Trade Service – Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Create a Service Fabric Cluster with a stateful Reliable Service for Routing Service. Deploy a Guest Executable to Service Fabric for each component.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
We should use stateful services when we want data to persist, and stateless service when data must not persist.

References: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction

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. Determine whether the solution meets the stated goals.

You develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

- Routing Service – Routes a request to the appropriate service and must not persist data across sessions
- Account Service – Stores and manages all account information and authentication and requires data to persist across sessions.
- User Service – Stores and manages all user information and requires data to persist across sessions.
- Housing Network Service – Stores and manages the current real-estate economy and requires data to persist across sessions. - Trade Service – Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Deploy a Windows container to Azure Service Fabric for each component.

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
We should use stateful services when we want data to persist, and stateless service when data must not persist.

References: https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction

QUESTION 4
HOTSPOT

You are developing an SMS-based testing solution. The solution sends users a question by using SMS. Early responders may qualify for prizes.

Users must respond with an answer choice within 90 seconds. You must be able to track how long it takes each user to respond.

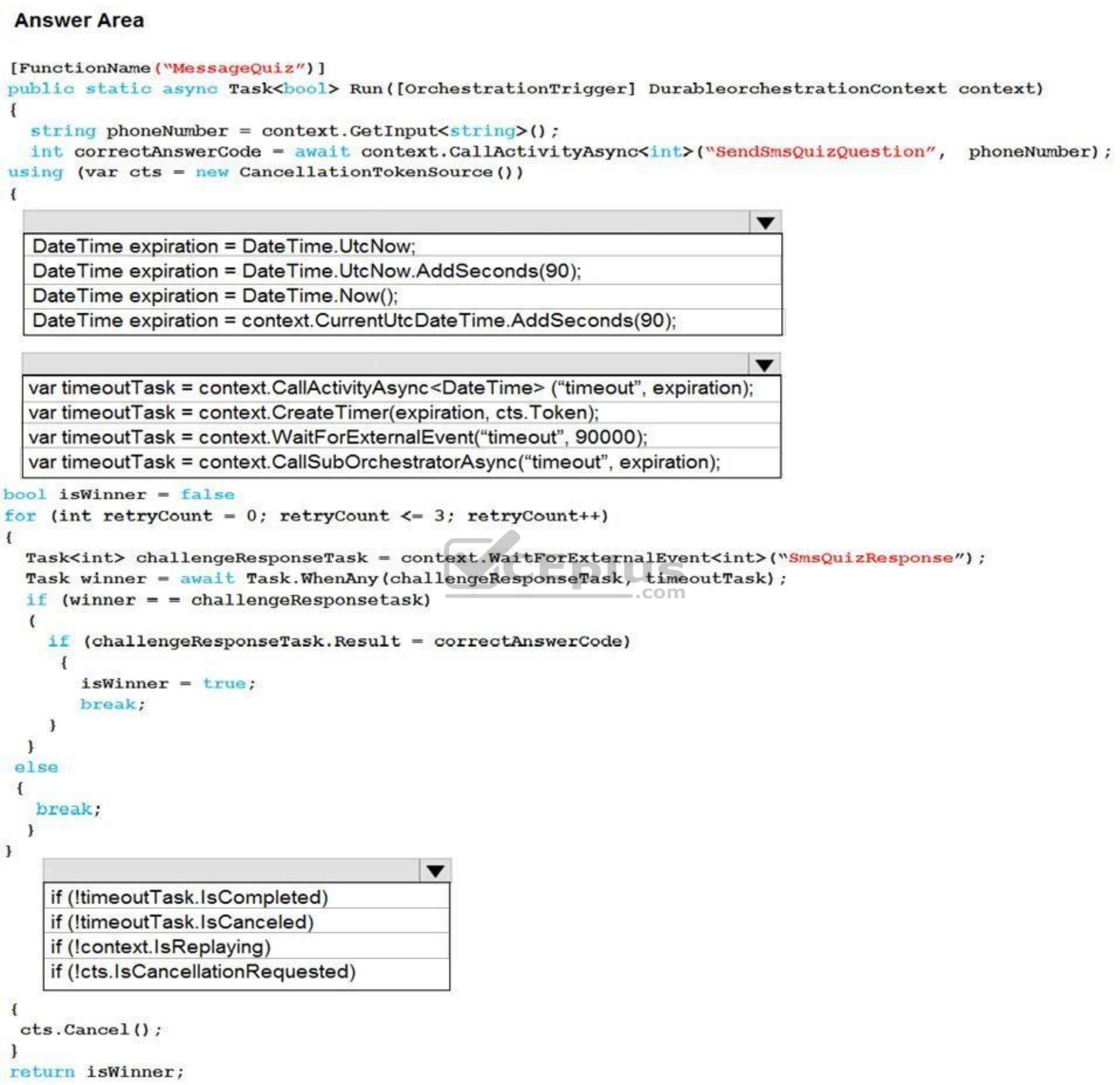
You create a durable Azure Function named SendSmsQuizQuestion that uses Twilio to send messages.

You need to write the code for MessageQuiz.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:



Answer Area

```
[FunctionName("MessageQuiz")]
public static async Task<bool> Run([OrchestrationTrigger] DurableorchestrationContext context)
{
  string phoneNumber = context.GetInput<string>();
  int correctAnswerCode = await context.CallActivityAsync<int>("SendSmsQuizQuestion",  phoneNumber);
using (var cts = new CancellationTokenSource())
{
```

- DateTime expiration = DateTime.UtcNow;
- DateTime expiration = DateTime.UtcNow.AddSeconds(90);
- DateTime expiration = DateTime.Now();
- DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);

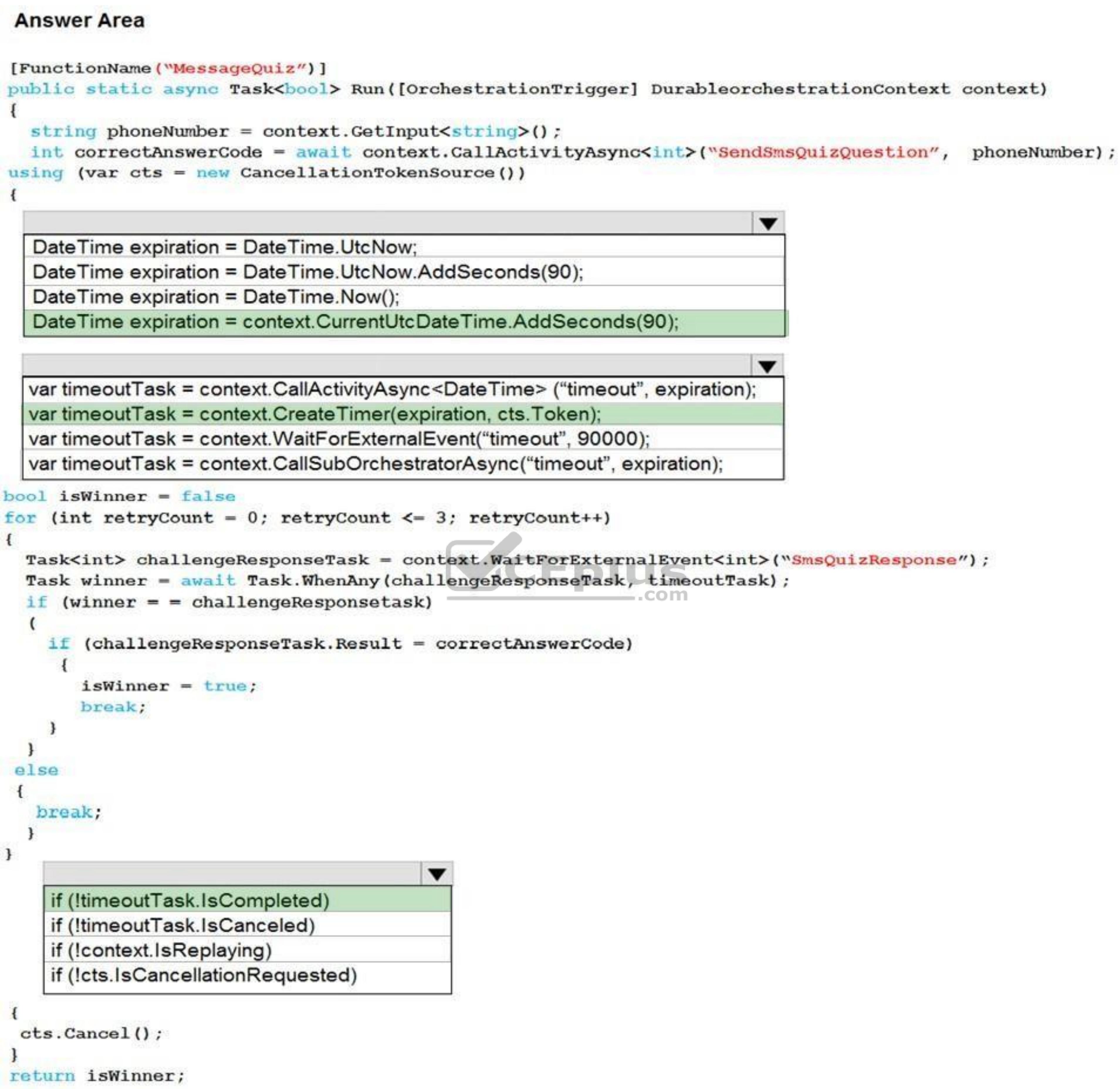
- var timeoutTask = context.CallActivityAsync<DateTime> ("timeout", expiration);
- var timeoutTask = context.CreateTimer(expiration, cts.Token);
- var timeoutTask = context.WaitForExternalEvent("timeout", 90000);
- var timeoutTask = context.CallSubOrchestratorAsync("timeout", expiration);

```
bool isWinner = false
for (int retryCount = 0; retryCount <= 3; retryCount++)
{
  Task<int> challengeResponseTask = context.WaitForExternalEvent<int>("SmsQuizResponse");
  Task winner = await Task.WhenAny(challengeResponseTask, timeoutTask);
  if (winner = = challengeResponsetask)
  (
    if (challengeResponseTask.Result = correctAnswerCode)
     {
       isWinner = true;
       break;
     }
  }
 else
 {
   break;
  }
}
```

- if (!timeoutTask.IsCompleted)
- if (!timeoutTask.IsCanceled)
- if (!context.IsReplaying)
- if (!cts.IsCancellationRequested)

```
{
 cts.Cancel();
}
return isWinner;
```

Correct Answer:



Answer Area

```
[FunctionName(“MessageQuiz”)]
public static async Task<bool> Run([OrchestrationTrigger] DurableorchestrationContext context)
{
  string phoneNumber = context.GetInput<string>();
  int correctAnswerCode = await context.CallActivityAsync<int>(“SendSmsQuizQuestion”,  phoneNumber);
using (var cts = new CancellationTokenSource())
{
```

- DateTime expiration = DateTime.UtcNow;
- DateTime expiration = DateTime.UtcNow.AddSeconds(90);
- DateTime expiration = DateTime.Now();
- [✓ DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);]

- var timeoutTask = context.CallActivityAsync<DateTime> (“timeout”, expiration);
- [✓ var timeoutTask = context.CreateTimer(expiration, cts.Token);]
- var timeoutTask = context.WaitForExternalEvent(“timeout”, 90000);
- var timeoutTask = context.CallSubOrchestratorAsync(“timeout”, expiration);

```
bool isWinner = false
for (int retryCount = 0; retryCount <= 3; retryCount++)
{
  Task<int> challengeResponseTask = context.WaitForExternalEvent<int>(“SmsQuizResponse”);
  Task winner = await Task.WhenAny(challengeResponseTask, timeoutTask);
  if (winner = = challengeResponsetask)
  (
    if (challengeResponseTask.Result = correctAnswerCode)
     {
      isWinner = true;
      break;
     }
  }
 else
 {
   break;
  }
}
```

- [✓ if (!timeoutTask.IsCompleted)]
- if (!timeoutTask.IsCanceled)
- if (!context.IsReplaying)
- if (!cts.IsCancellationRequested)

```
{
 cts.Cancel();
}
return isWinner;
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);
The user has 90 seconds to respond with the code they received in the SMS message.

Box 2:var timeoutTask = context.CreateTimer(expiration, cts.Token); Create a timer.

Box 3: if(!timeoutTask.IsCompleted)
All pending timers must be complete or canceled before the function exits.

References: https://github.com/Azure/azure-functions-durable-extension/blob/master/samples/precompiled/PhoneVerification.cs

QUESTION 5 You are developing an Azure Durable Function instance. You need to add a delay by using a durable timer.

What type of function should you use?

A. Activity
B. Orchestrator
C. Client
D. Webhook

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Durable Functions provides durable timers for use in orchestrator functions to implement delays or to set up timeouts on async actions.

References: https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview

QUESTION 6
HOTSPOT

You are creating an app that uses Event Grid to connect with other services. Your app's event data will be sent to a serverless function that checks compliance. This function is maintained by your company.

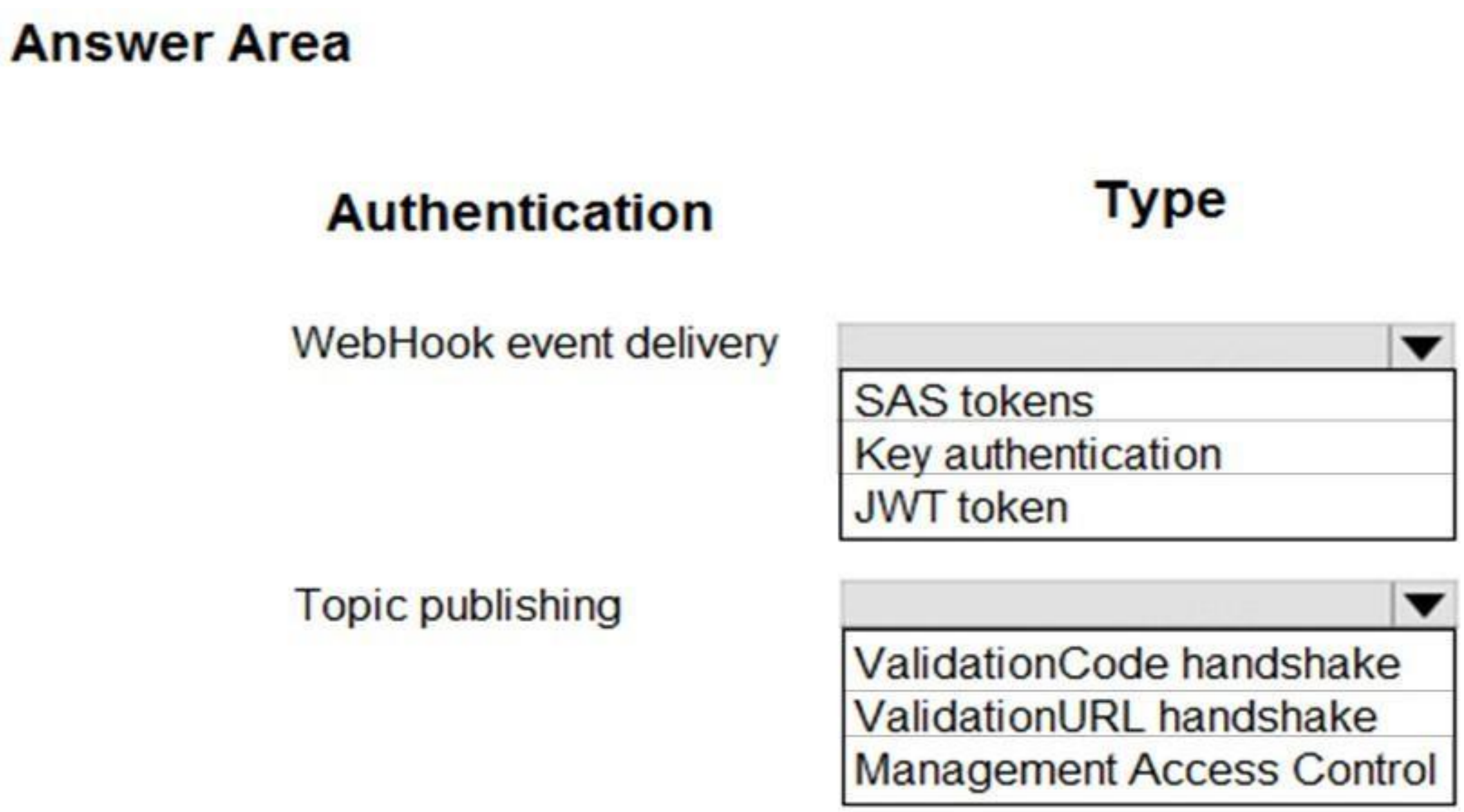
You write a new event subscription at the scope of your resource. The event must be invalidated after a specific period of time.

You need to configure Event Grid to ensure security.

What should you implement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

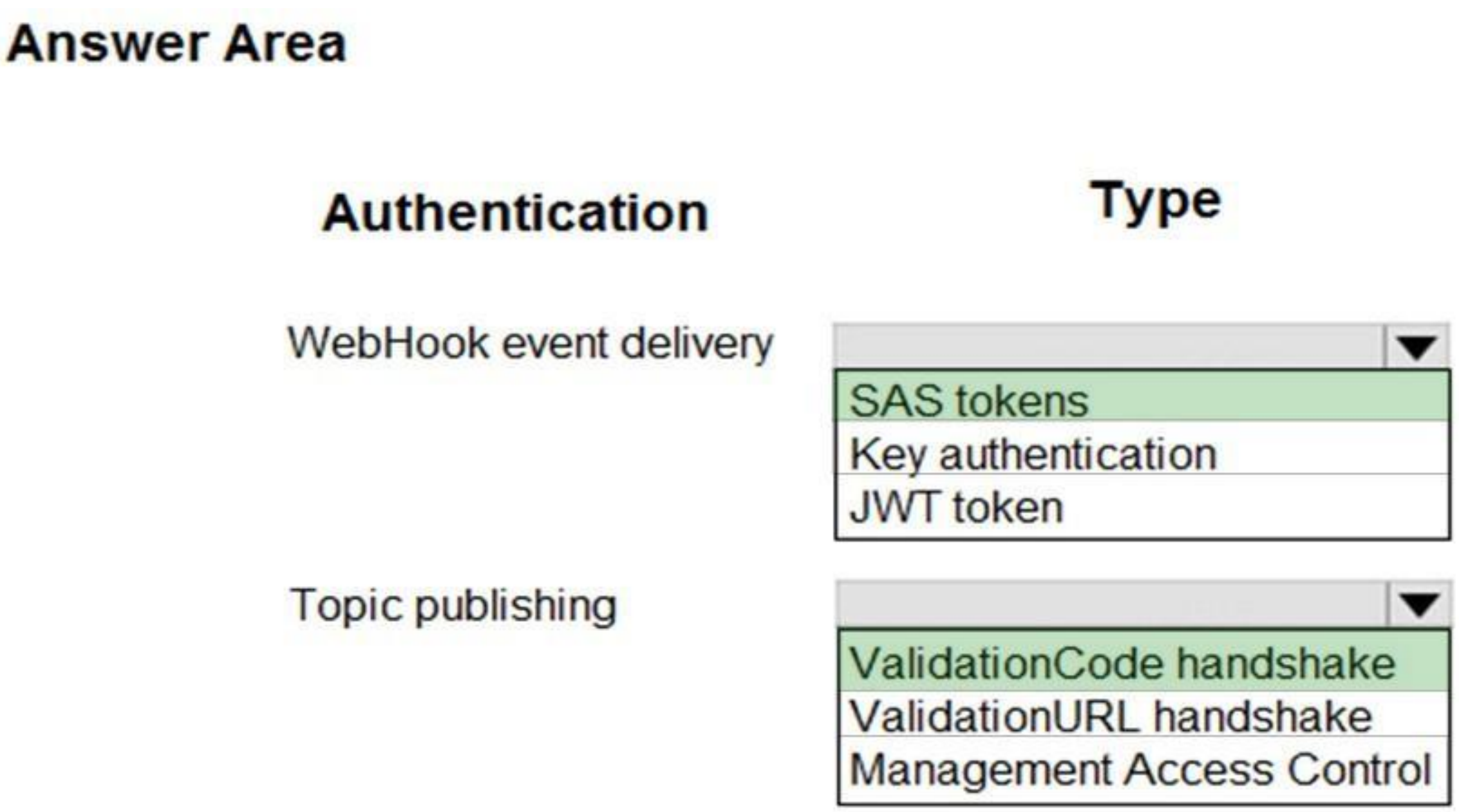
Hot Area:



Answer Area

Authentication	Type
WebHook event delivery	▼ SAS tokens Key authentication JWT token
Topic publishing	▼ ValidationCode handshake ValidationURL handshake Management Access Control

Correct Answer:



Answer Area

Authentication	Type
WebHook event delivery	SAS tokens Key authentication JWT token
Topic publishing	ValidationCode handshake ValidationURL handshake Management Access Control

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: SAS tokens
Custom topics use either Shared Access Signature (SAS) or key authentication. Microsoft recommends SAS, but key authentication provides simple programming, and is compatible with many existing webhook publishers. In this case we need the expiration time provided by SAS tokens.

Box 2: ValidationCode handshake
Event Grid supports two ways of validating the subscription: ValidationCode handshake (programmatic) and ValidationURL handshake (manual). If you control the source code for your endpoint, this method is recommended.

Incorrect Answers:
ValidationURL handshake (manual): In certain cases, you can't access the source code of the endpoint to implement the ValidationCode handshake. For example, if you use a third-party service (like Zapier or IFTTT), you can't programmatically respond with the validation code.

References: https://docs.microsoft.com/en-us/azure/event-grid/security-authentication

QUESTION 7
A company sells products worldwide and provides customer service in many languages. The company has a customer service email address for customer requests.

The language of the email is written in needs to be recognized and routed to the appropriate local language department.

You need to use the appropriate cognitive service to detect the language of the email.

How should you initiate language detection?

A. Use the Spark natural language processing functionality on Azure HDInsight.

B. Implement the DetectLanguageAsync method of the TextAnalyticsClient object.
C. Use the RecognizeAsync method of the SpeechRecognizer object.
D. Pass the content as a query parameter to https://api.cognitive.microsoft.com/bing/v7.0/search.

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
The DetectLanguageAsync(String) method detects the language of a text.

Incorrect Answers:
A: the Spark natural language processing functionality on Azure HDInsight Doesn't support language detection.
C: The RecognizeAsync method of the SpeechRecognizer object is used for speech recognition.
D: The request URL to https://api.cognitive.microsoft.com/bing/v7.0/search is used for customized searches.

References:
https://docs.microsoft.com/en-us/dotnet/api/microsoft.toolkit.services.microsofttranslator.translatorservice.detectlanguageasync?view=win-comm-toolkit-dotnet-stable

Testlet 1

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 question on 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 sections 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 on 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. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background

Best for You Organics Company is a global restaurant franchise that has multiple locations. The company wants to enhance user experiences and vendor integrations. The company plans to implement automated mobile ordering and delivery services.

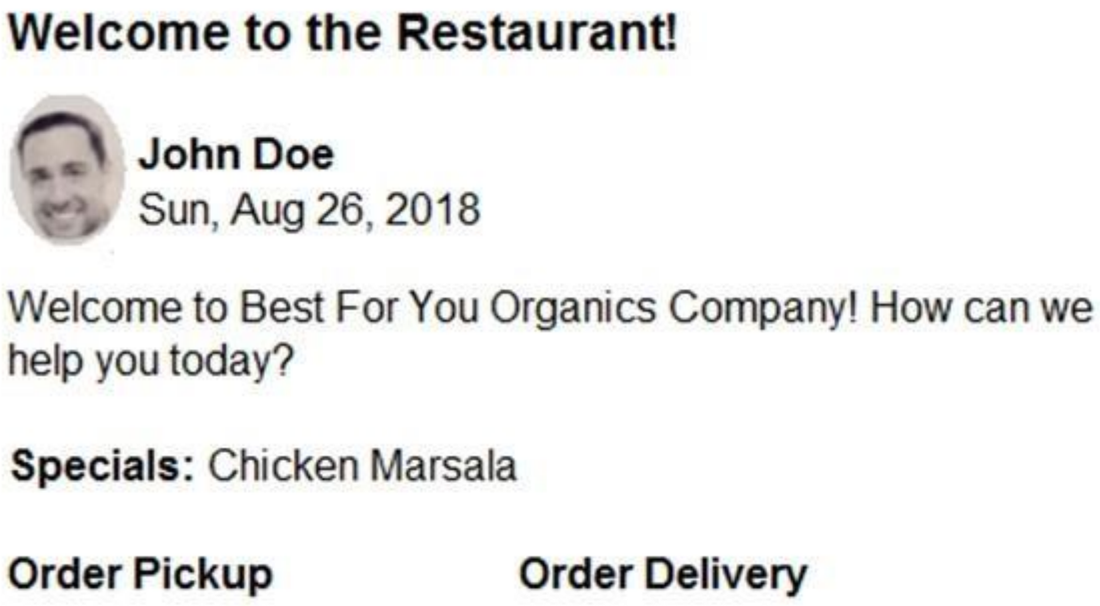
Best For You Organics hosts an Azure web app at the URL https://www.bestforyouorganics.com.
Users can use the web app to browse restaurant locations, menu items, nutritional, information, and company information. The company developed and deployed a cross-platform mobile app.

Requirements

You must develop a chartbot by using the Bot Builder SDK and Language Understanding Intelligence Service (LUIS). The chatbot must allow users to order food for pickup or delivery.

The chatbot must meet the following requirements:

▪ Ensure that chatbot endpoint can be accessed only by the Bot Framework connector.
▪ Use natural language processing and speech recognition so that users can interact with the chatbot by using text and voice. Processing must be server-based. ▪
Alert users about promotions at local restaurants.
▪ Enable users to place an order for delivery or pickup by using their voice.
▪ Greet the user upon sigh-in by displaying a graphical interface that contains action buttons. ▪
The chatbot greeting interface must match the formatting of the following example:



Welcome to the Restaurant!

John Doe
Sun, Aug 26, 2018

Welcome to Best For You Organics Company! How can we help you today?

Specials: Chicken Marsala

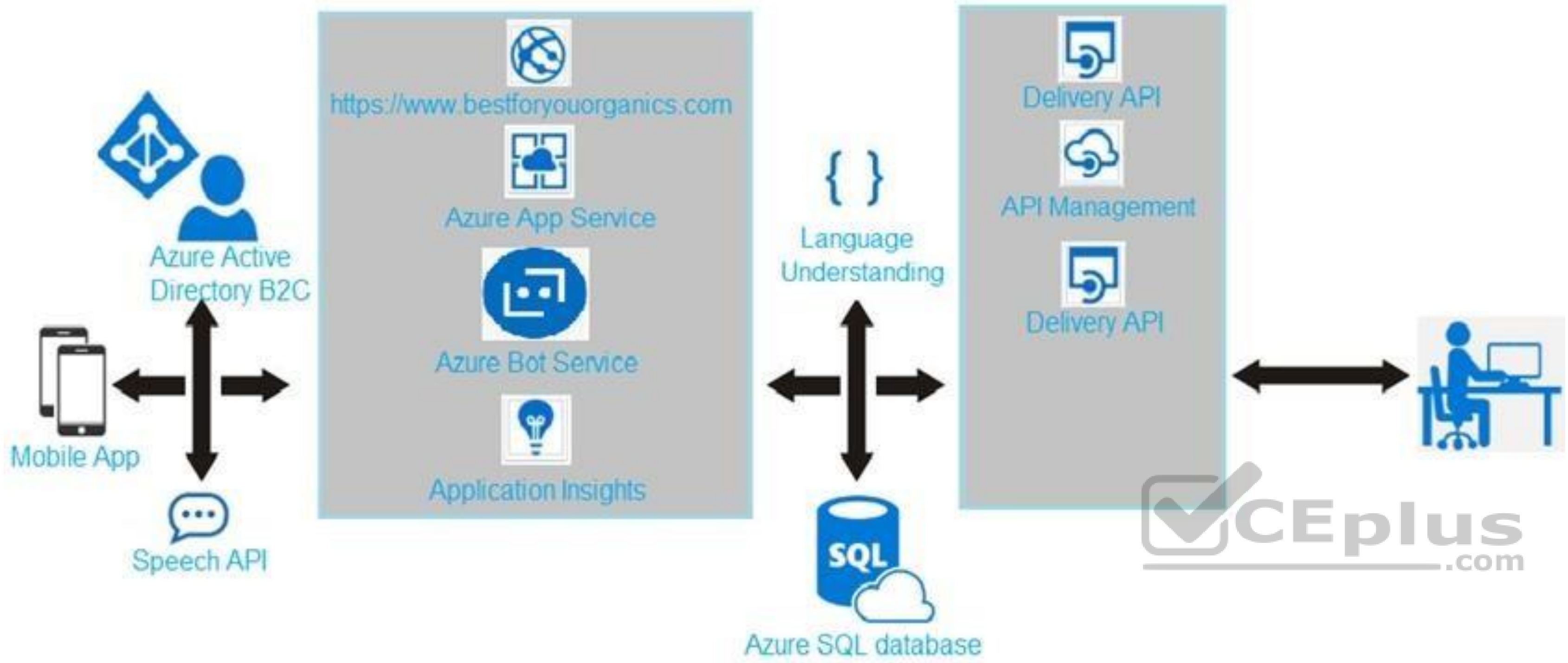
Order Pickup **Order Delivery**

Vendor API
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

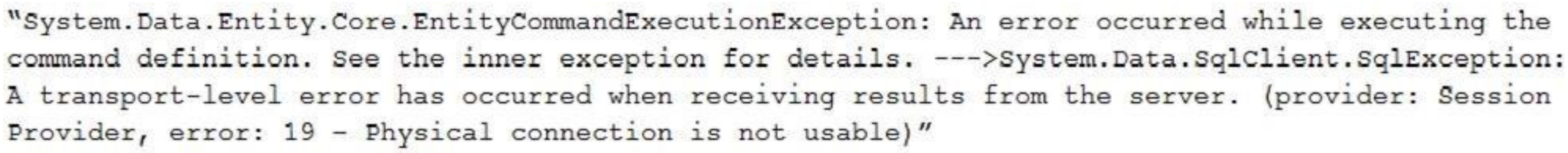
- API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.
- If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.
- APIs must prevent API usage spikes on a per-subscription basis by limiting the call rate to 100 calls per minute.
- The Inventory API must be written by using ASP.NET Core and Node.js.
- The API must be updated to provide an interface to Azure SQL Database. Database objects must be managed by using code.
- The Delivery API must be protected by using the OAuth 2.0 protocol with Azure Active Directory (Azure AD) when called from the Azure web app. You register the Delivery API and web app in Azure AD. You enable OAuth 2.0 in the web app.
- The delivery API must update the Products table, the Vendor transactions table, and the Billing table in a single transaction.

The Best For You Organics Company architecture team has created the following diagram depicting the expected deployments into Azure:



Delivery API

The Delivery API intermittently throws the following exception:



```
"System.Data.Entity.Core.EntityCommandExecutionException: An error occurred while executing the
command definition. See the inner exception for details. --->System.Data.SqlClient.SqlException:
A transport-level error has occurred when receiving results from the server. (provider: Session
Provider, error: 19 - Physical connection is not usable)"
```

Chatbot greeting

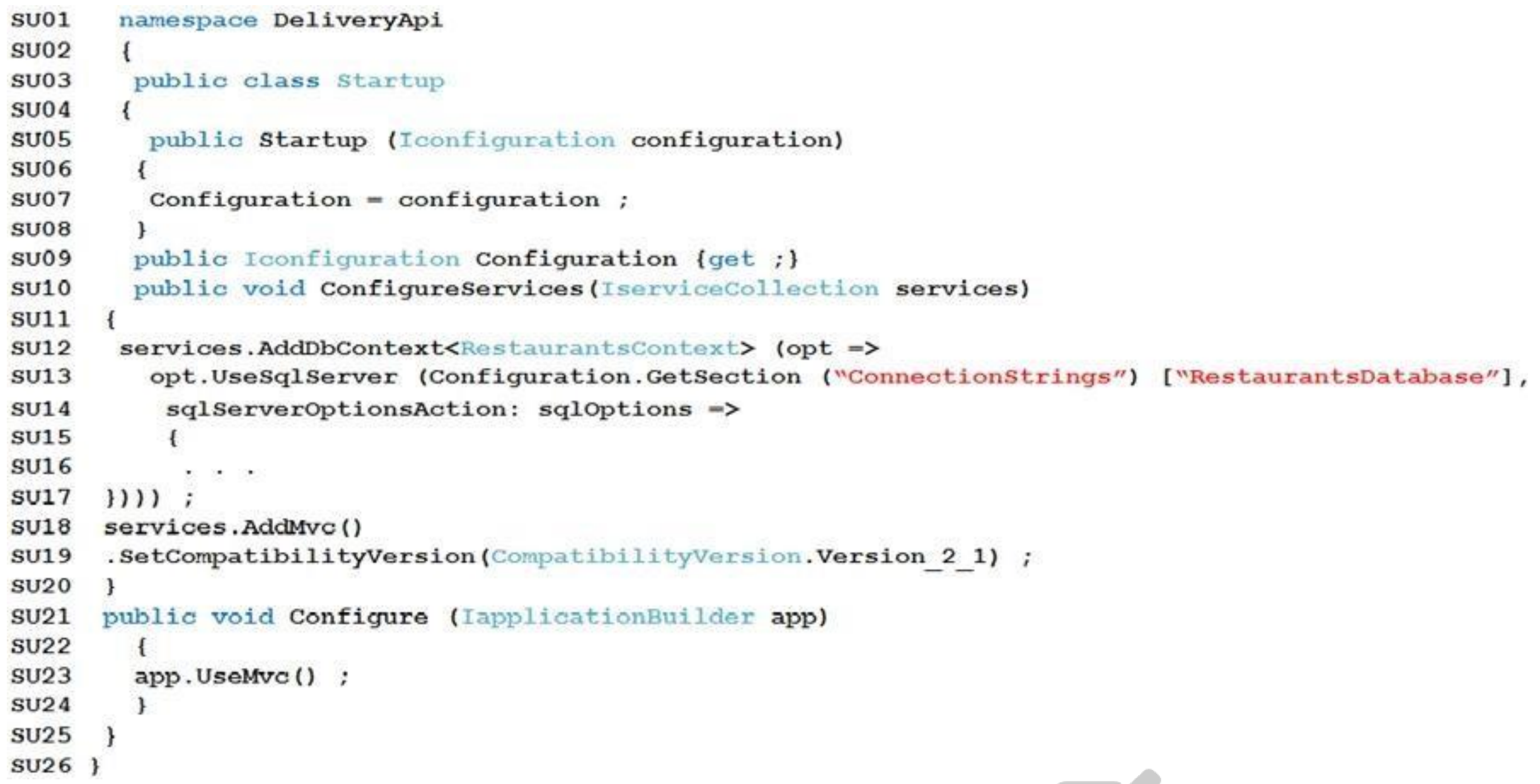
The chatbot’s greeting does not show the user’s name. You need to debug the chatbot locally.

Language processing

Users report that the bot fails to understand when a customer attempts to order dishes that use Italian names.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

Startup.cs



```
SU01   namespace DeliveryApi
SU02   {
SU03    public class Startup
SU04   {
SU05     public Startup (Iconfiguration configuration)
SU06     {
SU07     Configuration = configuration ;
SU08     }
SU09    public Iconfiguration Configuration {get ;}
SU10    public void ConfigureServices(IserviceCollection services)
SU11  {
SU12   services.AddDbContext<RestaurantsContext> (opt =>
SU13     opt.UseSqlServer (Configuration.GetSection ("ConnectionStrings") ["RestaurantsDatabase"],
SU14      sqlServerOptionsAction: sqlOptions =>
SU15      {
SU16       . . .
SU17  }))) ;
SU18  services.AddMvc()
SU19  .SetCompatibilityVersion(CompatibilityVersion.Version_2_1) ;
SU20  }
SU21  public void Configure (IapplicationBuilder app)
SU22     {
SU23    app.UseMvc() ;
SU24     }
SU25  }
SU26 }
```

QUESTION 1 You need to implement the purchase requirement.

What should you do?

A. Use the Bot Framework REST API attachment operations to send the user’s voice and the Speech Service API to recognize intents.
B. Use the Direct Line REST API to send the user’s voice and the Speech Service API to recognize intents.
C. Use the Speech Service API to send the user’s voice and the Bot Framework REST API conversation operations to recognize intents.
D. Use the Speech Service API to send the user’s voice and the Direct Line REST API to recognize intents.

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: Enable users to place an order for delivery or pickup by using their voice.
You must develop a chatbot by using the Bot Builder SDK and Language Understanding Intelligence Service (LUIS). The chatbot must allow users to order food for pickup or delivery.

The Bot Framework REST APIs enable you to build bots that exchange messages with channels configured in the Bot Framework Portal, store and retrieve state data, and connect your own client applications to your bots. All Bot Framework services use industry-standard REST and JSON over HTTPS.

The Speech Service API is used to recognize intents.

References: https://docs.microsoft.com/en-us/azure/bot-service/rest-api/bot-framework-rest-connector-concepts?view=azure-bot-service-4.0
https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/how-to-recognize-intents-from-speech-cpp **QUESTION 2** You need to debug the user greeting issue.

What should you use?

A. Azure Compute Emulator
B. Bot Framework Emulator
C. Azure Application Insights
D. Bot Framework Channel Inspector
E. Bot Connector service

Correct Answer: B
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The chatbot's greeting does not show the user's name. You need to debug the chatbot locally.

Debug your bot using an integrated development environment (IDE) such as Visual Studio or Visual Studio Code and the Bot Framework Emulator. You can use these methods to debug any bot locally.

References: https://docs.microsoft.com/en-us/azure/bot-service/bot-service-debug-bot?view=azure-bot-service-4.0

QUESTION 3 You need to update the chatbot to greet the user when they sign in.

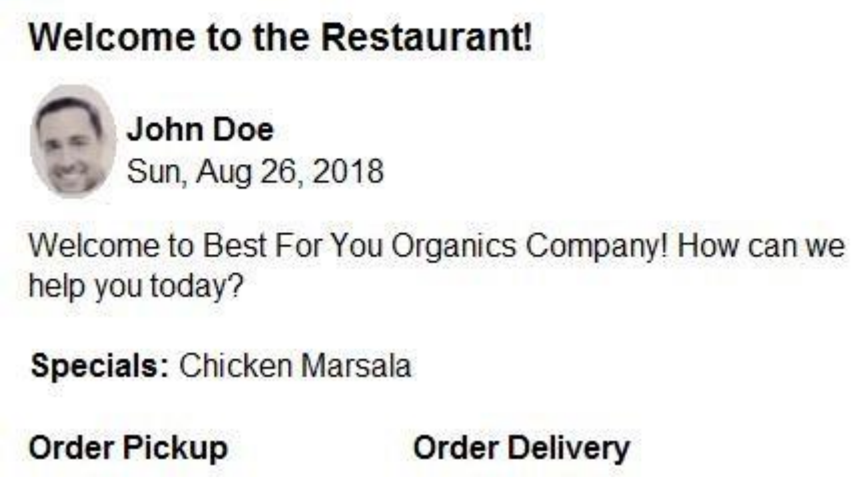
Which two rich card formats can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Sign-in
B. Animation
C. Hero
D. Thumbnail
E. Adaptive

Correct Answer: AD
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Scenario: The chatbot greeting interface must match the formatting of the following example:



Welcome to the Restaurant!

John Doe
Sun, Aug 26, 2018

Welcome to Best For You Organics Company! How can we help you today?

Specials: Chicken Marsala

Order Pickup **Order Delivery**

A message exchange between user and bot can contain one or more rich cards rendered as a list or carousel. The Attachments property of the Activity object contains an array of Attachment objects that represent the rich cards and media attachments within the message.

The Bot Framework currently supports eight types of rich cards:

- Thumbnail Card. A card that typically contains a single thumbnail image, one or more buttons, and text.
- SignIn Card. A card that enables a bot to request that a user sign-in. It typically contains text and one or more buttons that the user can click to initiate the sign-in process.

Incorrect Answers:
B: Animation Card. A card that can play animated GIFs or short videos.
C Hero Card. A card that typically contains a single large image, one or more buttons, and text.
E: Adaptive Card. A customizable card that can contain any combination of text, speech, images, buttons, and input fields.

Note:
- Receipt Card. A card that enables a bot to provide a receipt to the user. It typically contains the list of items to include on the receipt, tax and total information, and other text.
- Video Card. A card that can play videos.

References: https://docs.microsoft.com/en-us/azure/bot-service/dotnet/bot-builder-dotnet-add-rich-card-attachments?view=azure-bot-service-3.0

Question Set 2

QUESTION 1
DRAG DROP

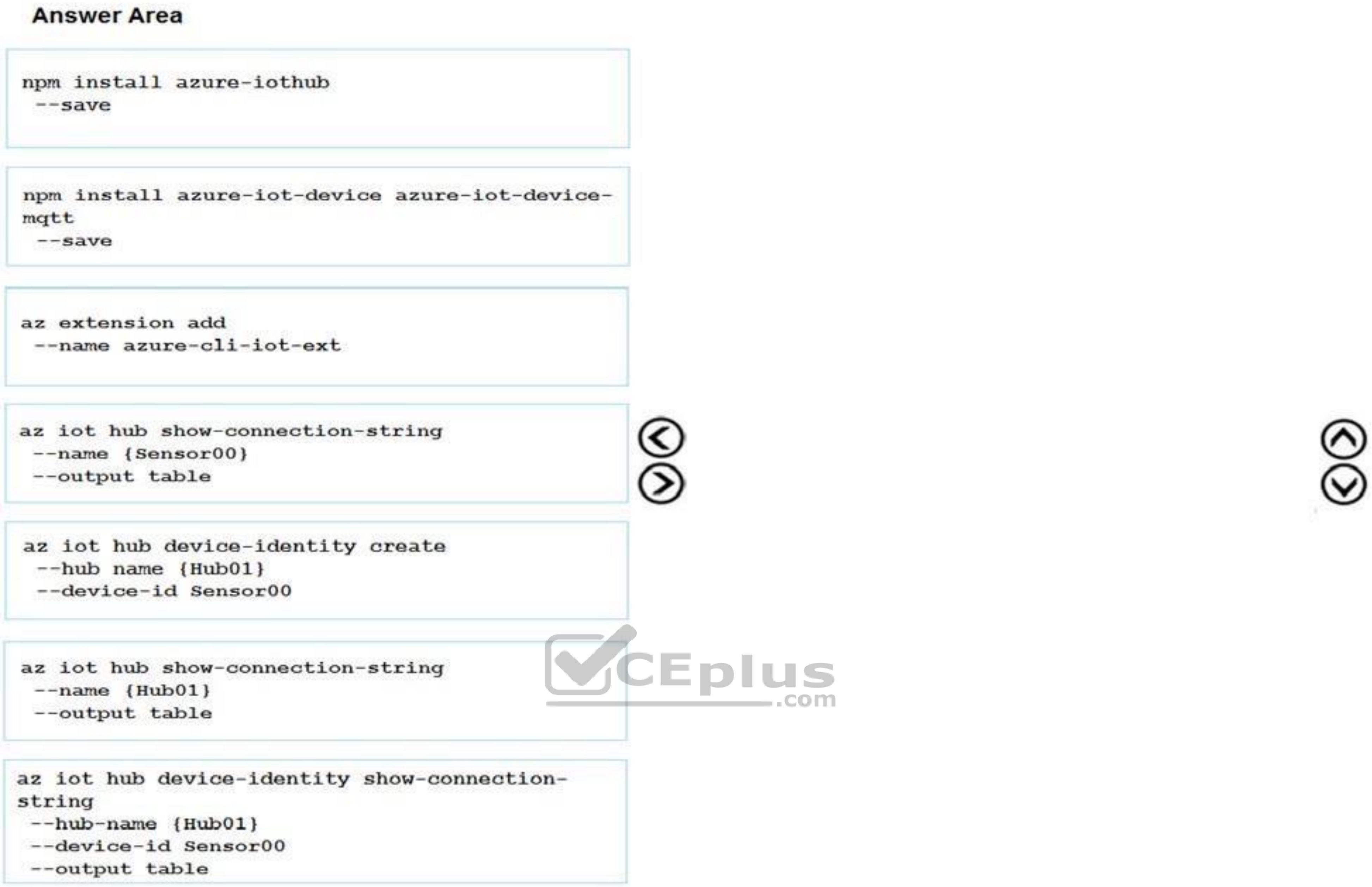
You develop an IoT solution by using Node.js. The solution is ready to deploy to the production environment.

You must implement the device twin capabilities of Azure IoT Hub. You must register a sensor named Sensor00. The IoT Hub name is Hub01.

You need to register the endpoint with ContosoHub01 so that you can configure them from your solution.

Which four commands should you use to develop the solution? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

Select and Place:



Answer Area

```
npm install azure-iothub
 --save
```

```
npm install azure-iot-device azure-iot-device-
mqtt
 --save
```

```
az extension add
 --name azure-cli-iot-ext
```

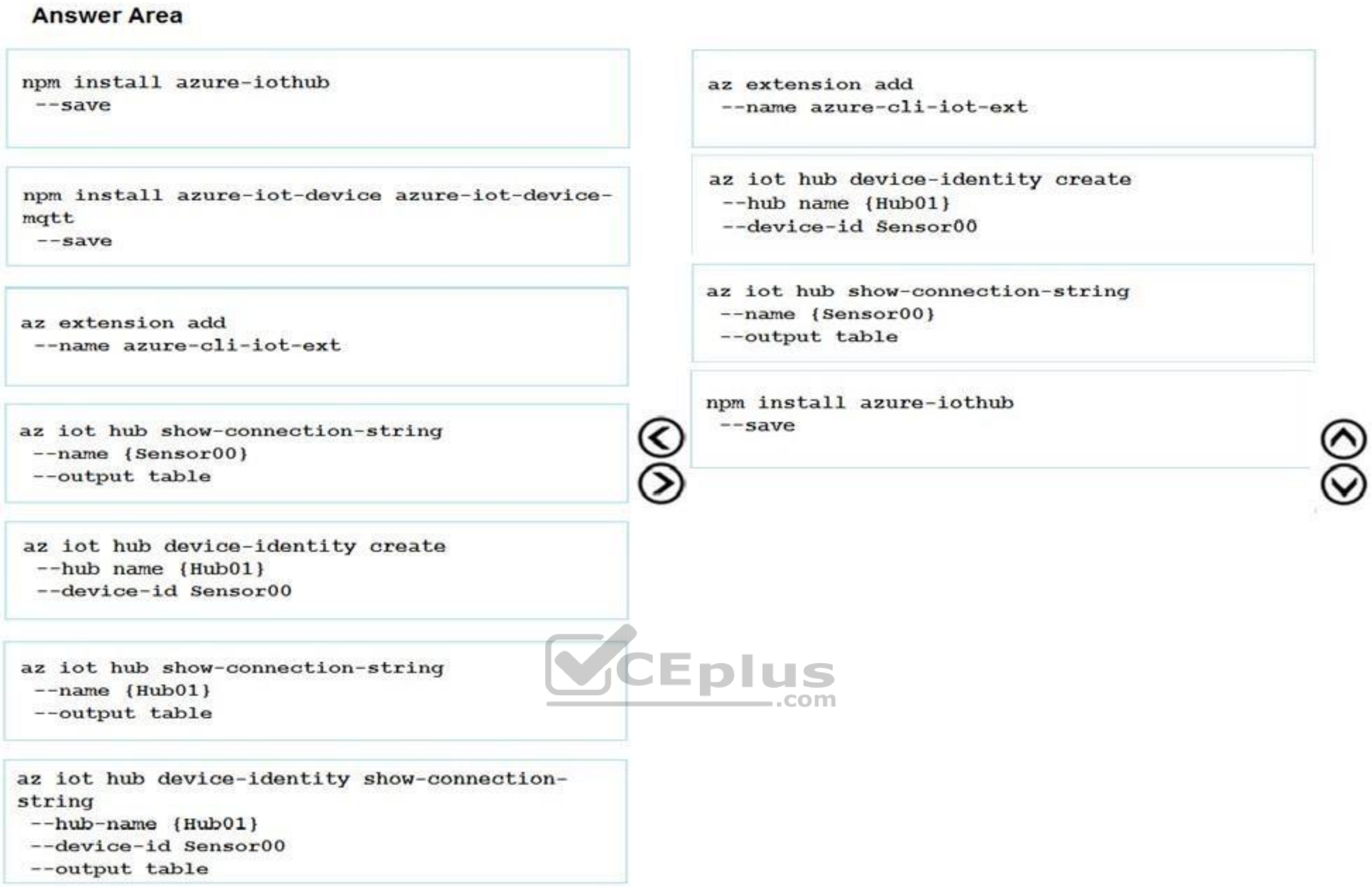
```
az iot hub show-connection-string
 --name {Sensor00}
 --output table
```

```
az iot hub device-identity create
 --hub name {Hub01}
 --device-id Sensor00
```

```
az iot hub show-connection-string
 --name {Hub01}
 --output table
```

```
az iot hub device-identity show-connection-
string
 --hub-name {Hub01}
 --device-id Sensor00
 --output table
```

Correct Answer:



Answer Area

```
npm install azure-iothub
 --save
```

```
npm install azure-iot-device azure-iot-device-
mqtt
 --save
```

```
az extension add
 --name azure-cli-iot-ext
```

```
az iot hub show-connection-string
 --name {Sensor00}
 --output table
```

```
az iot hub device-identity create
 --hub name {Hub01}
 --device-id Sensor00
```

```
az iot hub show-connection-string
 --name {Hub01}
 --output table
```

```
az iot hub device-identity show-connection-
string
 --hub-name {Hub01}
 --device-id Sensor00
 --output table
```

```
az extension add
 --name azure-cli-iot-ext
```

```
az iot hub device-identity create
 --hub name {Hub01}
 --device-id Sensor00
```

```
az iot hub show-connection-string
 --name {Sensor00}
 --output table
```

```
npm install azure-iothub
 --save
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Step 1: az extension add --name azure-cli-iot-ext
Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension: az extension add --name azure-cli-iot-ext

Step 2: az iot hub device-identity create
--hub-name {Hub01}
--device-id Sensor00

Create a new device identity called myDeviceId and retrieve the device connection string with these commands: az iot hub device-identity create --device-id myDeviceId --hub-name {Your IoT Hub name}
az iot hub device-identity show-connection-string --device-id myDeviceId --hub-name {Your IoT Hub name} -o table

Step 3: az iot hub device-identity show-connection-string

--hub-name {Hub01}

--device-id Sensor00 --
output table

Step 4:
Create the service app
In this section, you create a Node.js console app that adds location metadata to the device twin. npm

install azure-iothub --save

References: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-twin-getstarted

QUESTION 2 You are developing a speech-enabled home automation control bot.

The bot interprets some spoken words incorrectly.

You need to improve the spoken word recognition for the bot.

What should you implement?

A. The Skype for Business Channel and use scorable dialogs for improving conversation flow
B. The Skype Channel and Speech priming using a LUIS app
C. The Cortana Channel and Speech priming using a LUIS app
D. The Web Chat Channel and use scorable dialogs for improving conversation flow

Correct Answer: C
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Speech priming improves the recognition of spoken words and phrases that are commonly used in your bot. For speech-enabled bots that use the Web Chat and Cortana channels, speech priming uses examples specified in Language Understanding (LUIS) apps to improve speech recognition accuracy for important words.

References: https://docs.microsoft.com/en-us/azure/bot-service/bot-service-manage-speech-priming?view=azure-bot-service-4.0

QUESTION 3
HOTSPOT

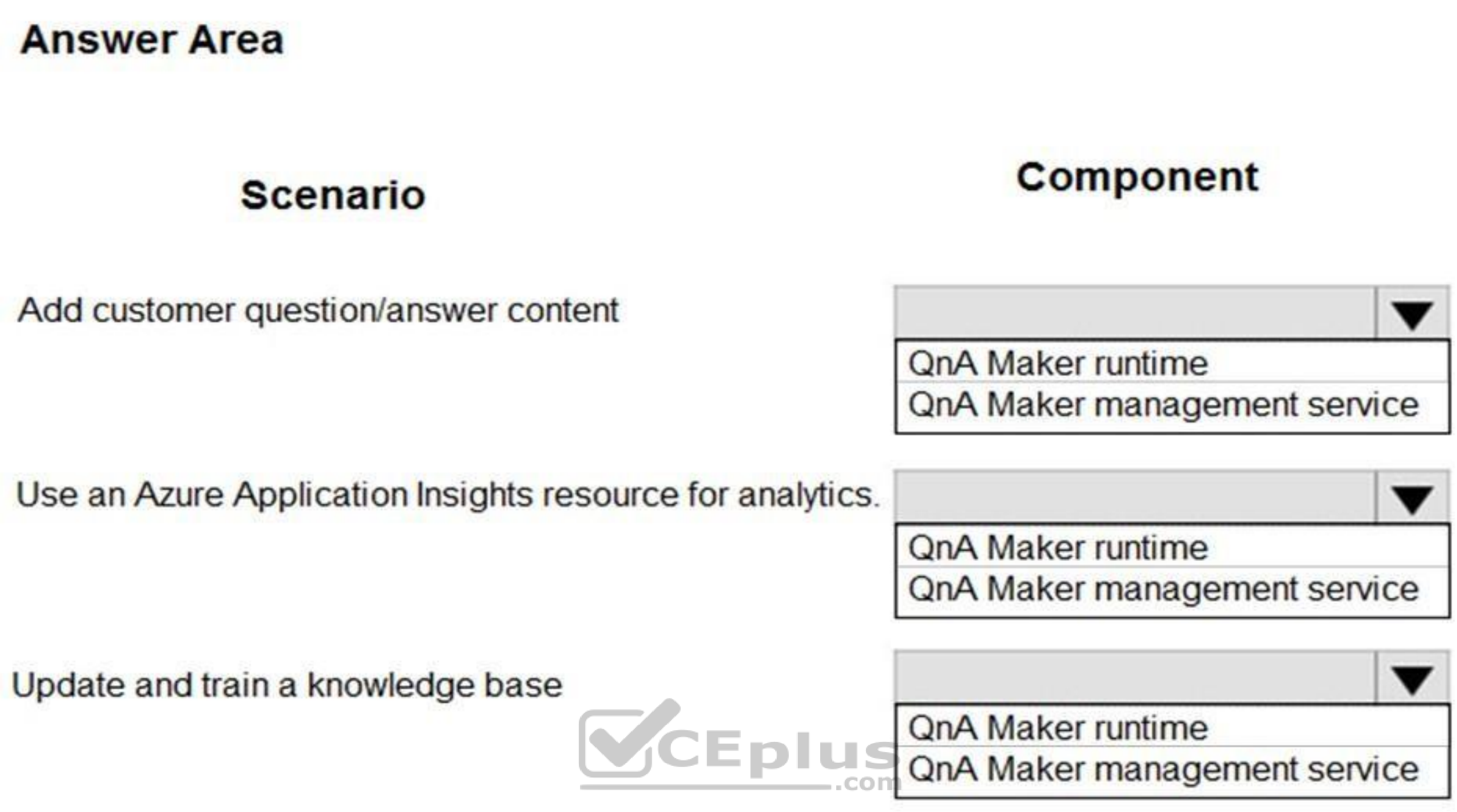
You are creating a bot for a company by using QnA Maker.

You need to ensure that the company can update the bot without third-party assistance.

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

NOTE: Each correct selection is worth one point.

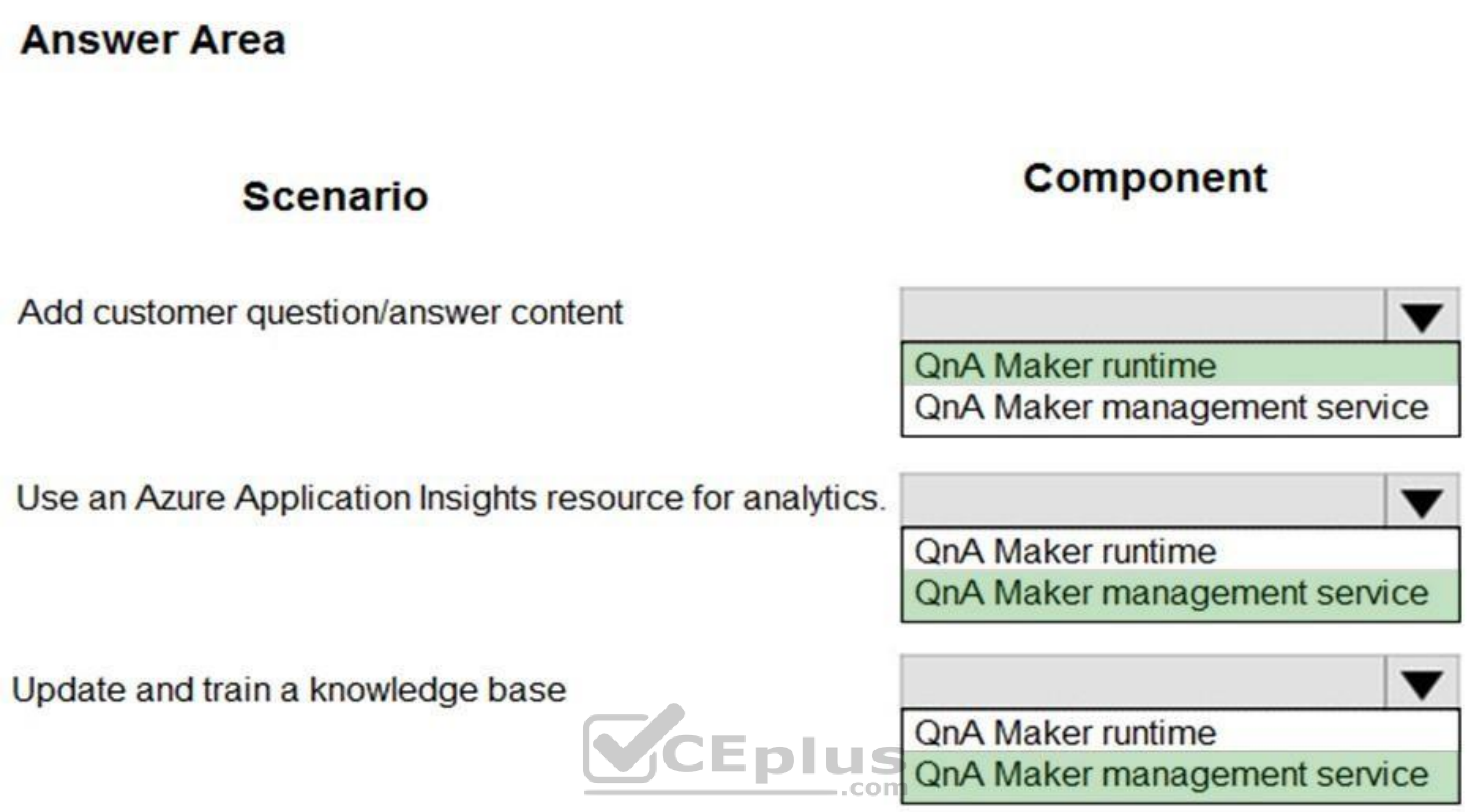
Hot Area:



Answer Area

Scenario	Component
Add customer question/answer content	▼ QnA Maker runtime QnA Maker management service
Use an Azure Application Insights resource for analytics.	▼ QnA Maker runtime QnA Maker management service
Update and train a knowledge base	▼ QnA Maker runtime QnA Maker management service

Correct Answer:



Answer Area

Scenario	Component
Add customer question/answer content	[✓ QnA Maker runtime] QnA Maker management service
Use an Azure Application Insights resource for analytics.	QnA Maker runtime [✓ QnA Maker management service]
Update and train a knowledge base	QnA Maker runtime [✓ QnA Maker management service]

Section: [none]
Explanation

Explanation/Reference:

QUESTION 4
DRAG DROP

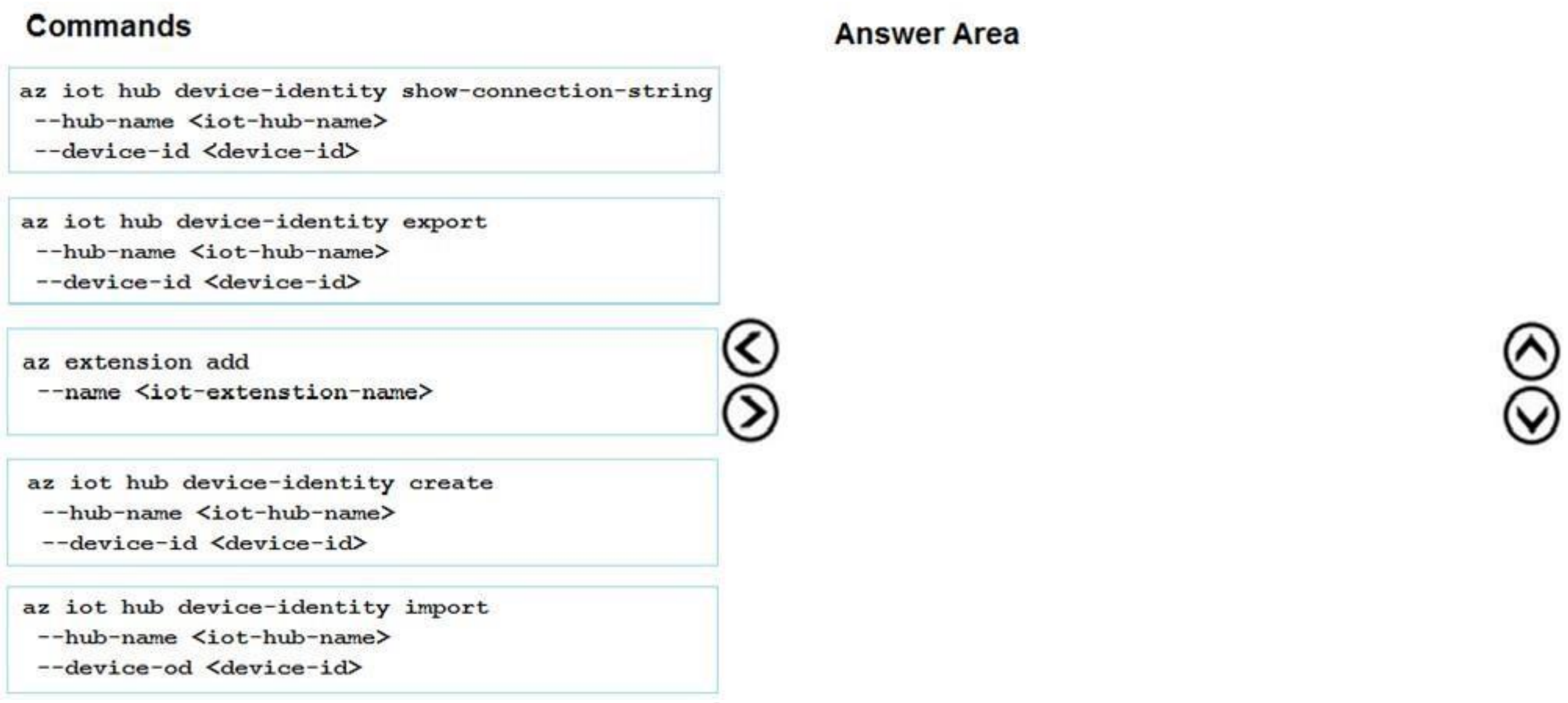
You are developing an IoT solution. The solution requires bidirectional communication between a .NET application and Azure IoT Hub.

You need to obtain connection information for a single test device.

Which three commands should you use to develop the solution? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

NOTE: Each correct selection is worth one point.

Select and Place:



Commands

```
az iot hub device-identity show-connection-string
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az iot hub device-identity export
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

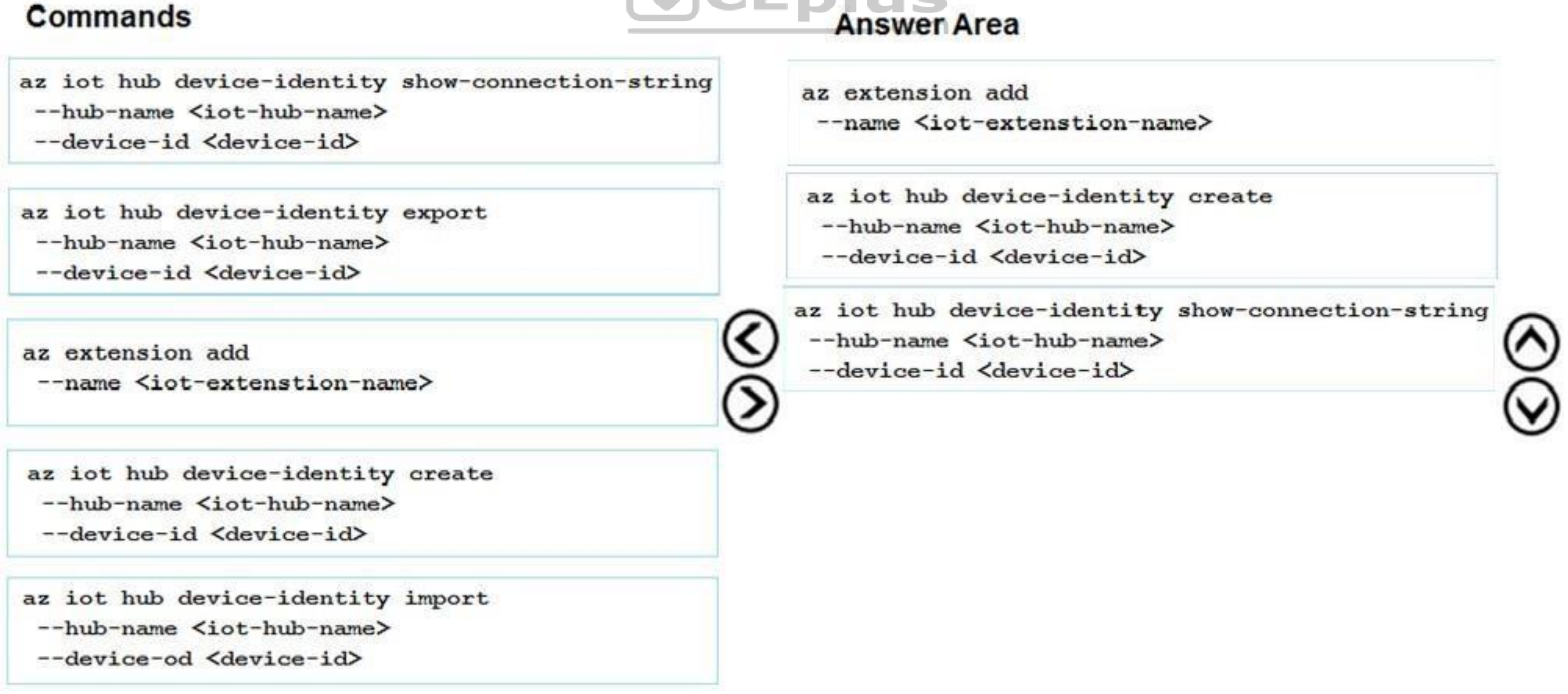
```
az extension add
 --name <iot-extenstion-name>
```

```
az iot hub device-identity create
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az iot hub device-identity import
 --hub-name <iot-hub-name>
 --device-od <device-id>
```

Answer Area

Correct Answer:



Commands

```
az iot hub device-identity show-connection-string
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az iot hub device-identity export
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az extension add
 --name <iot-extenstion-name>
```

```
az iot hub device-identity create
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az iot hub device-identity import
 --hub-name <iot-hub-name>
 --device-od <device-id>
```

Answer Area

```
az extension add
 --name <iot-extenstion-name>
```

```
az iot hub device-identity create
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

```
az iot hub device-identity show-connection-string
 --hub-name <iot-hub-name>
 --device-id <device-id>
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension:

Step 1: az extension add
--name <iot-extension-name>

Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension:

```
az extension add --name azure-cli-iot-ext
```

Step 2: az iot hub device-identity create
--hub-name <iot-hub-name>
--device-id <device-id>

Create a new device in the IoT Hub “DemoHub”
Command: az iot hub device-identity create --hub-name DemoHub --device-id testDevice

Step 3: az iot hub device-identity show-connection-string
--hub-name { iot-hub-name } --device-
id <device-id>

References: https://github.com/MicrosoftDocs/azure-docs/blob/master/includes/iot-hub-get-started-create-device-identity.md

QUESTION 5
HOTSPOT

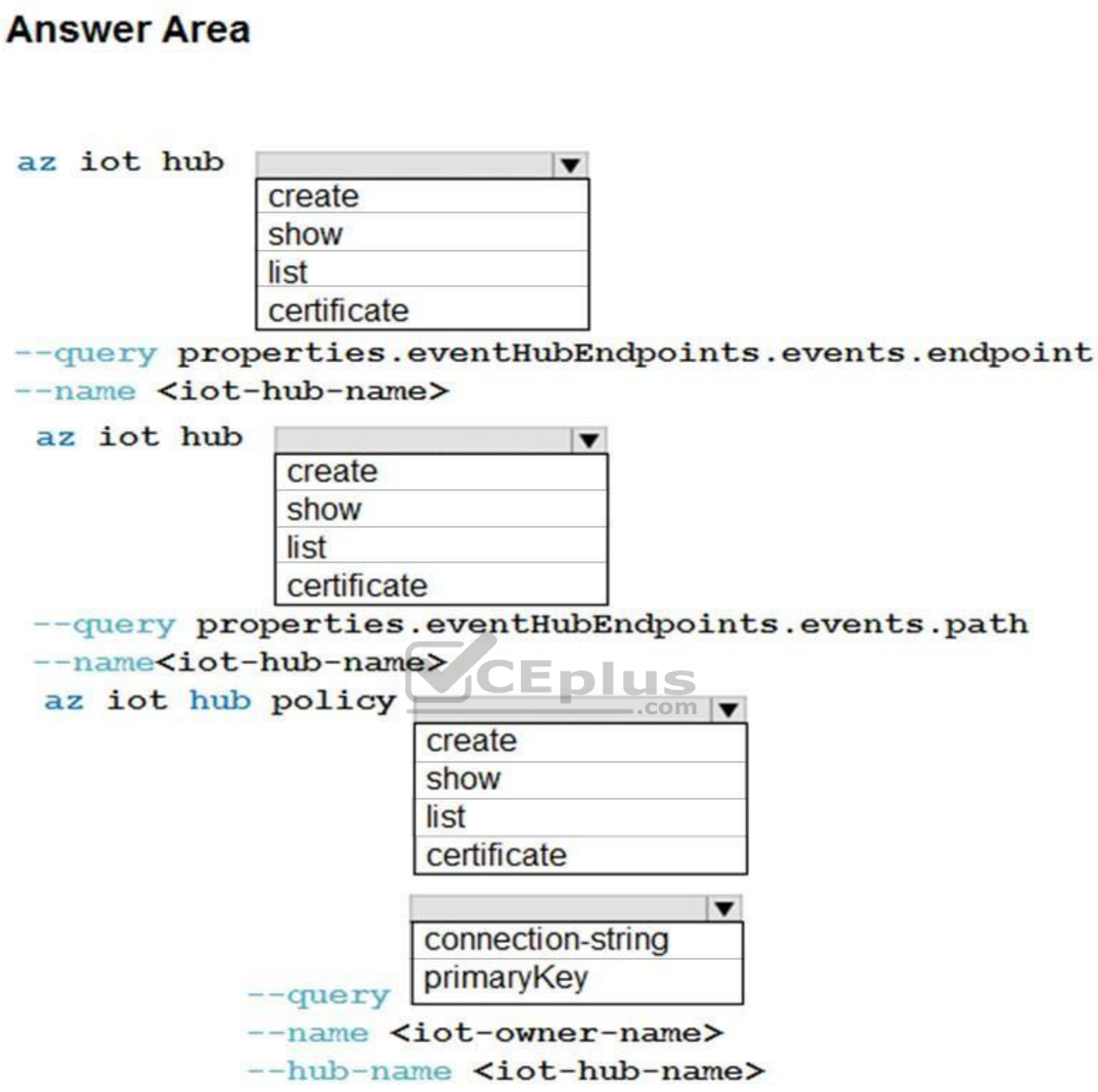
You are developing an IoT solution.

The solution requires bidirectional communication between a client .NET application and Azure IoT hub. A .NET back-end application will connect to the IoT Hub to process information.

You need to collect the values required for the back-end application to connect with the newly created IoT Hub.

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

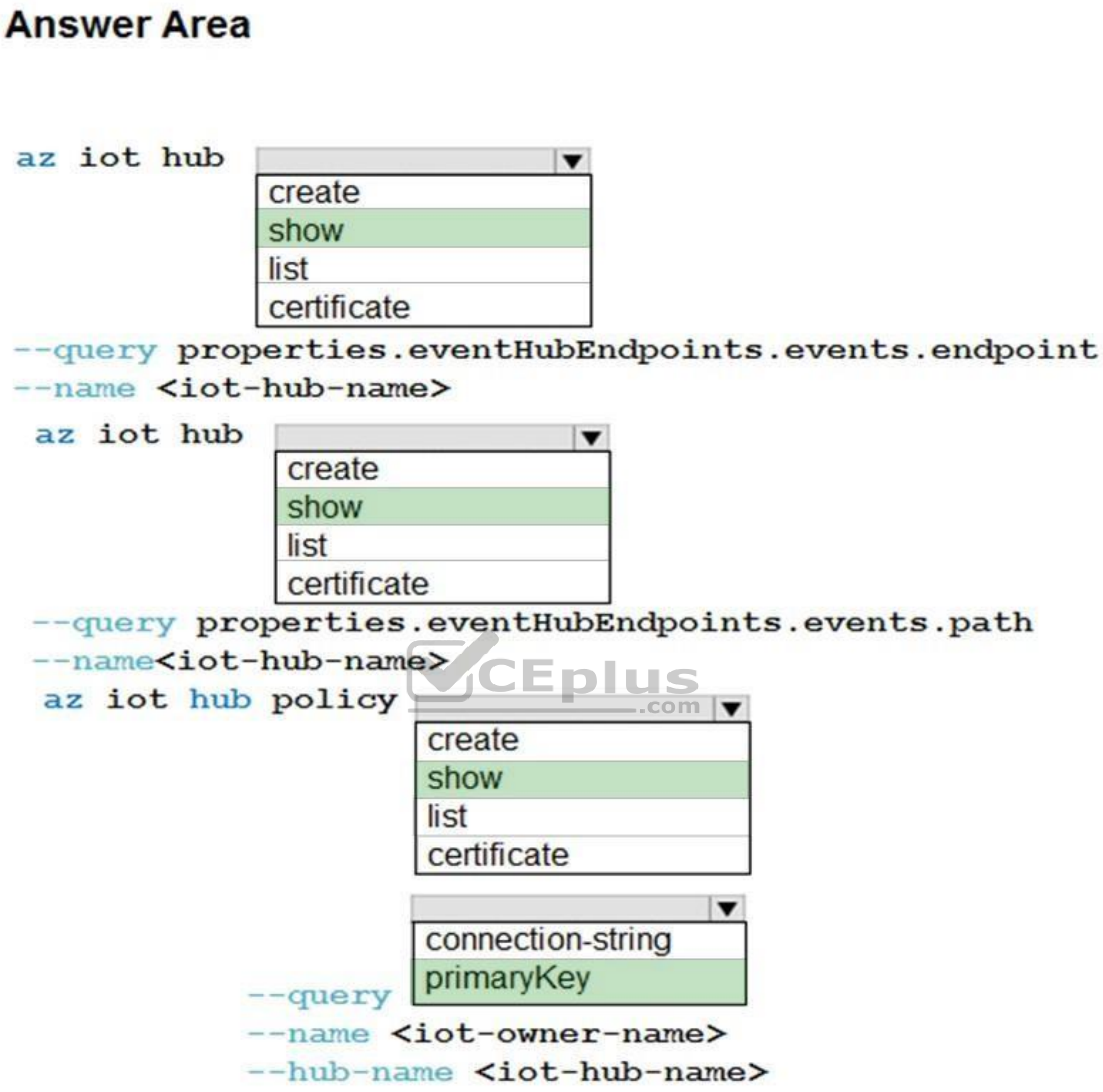
Hot Area:



Answer Area

```
az iot hub [ create | show | list | certificate ]
--query properties.eventHubEndpoints.events.endpoint
--name <iot-hub-name>
 az iot hub [ create | show | list | certificate ]
 --query properties.eventHubEndpoints.events.path
 --name<iot-hub-name>
 az iot hub policy [ create | show | list | certificate ]
                   [ connection-string | primaryKey ]
           --query
           --name <iot-owner-name>
           --hub-name <iot-hub-name>
```

Correct Answer:



Answer Area

```
az iot hub
```

- create
- [✓ show]
- list
- certificate

```
--query properties.eventHubEndpoints.events.endpoint
--name <iot-hub-name>
az iot hub
```

- create
- [✓ show]
- list
- certificate

```
--query properties.eventHubEndpoints.events.path
--name<iot-hub-name>
az iot hub policy
```

- create
- [✓ show]
- list
- certificate

- connection-string
- [✓ primaryKey]

```
--query
--name <iot-owner-name>
--hub-name <iot-hub-name>
```

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Box 1: show Event Hub-compatible endpoint
az iot hub show --query properties.eventHubEndpoints.events.endpoint --name {your IoT Hub name}

Box 2: show

Event Hub-compatible name
az iot hub show --query properties.eventHubEndpoints.events.path --name {your IoT Hub name}

Box 3: show
az iot hub policy show --name iothubowner --query primaryKey --hub-name {your IoT Hub name}
Box 4: primaryKey

References: https://github.com/Azure-Samples/azure-iot-samples-csharp/blob/master/iot-hub/Quickstarts/read-d2c-messages/ReadDeviceToCloudMessages.cs

QUESTION 6
A company is creating an IoT solution for connecting to smart refrigerators. You plan to use the Azure IoT Hub Device Provisioning Service for this process.

You need to provision the devices automatically.

Which feature of Device Provisioning Service should you use?

A. Device registration and configuration
B. Template registration
C. Error policy
D. Device simulation

Correct Answer: A
Section: [none]
Explanation

Explanation/Reference:
Explanation:
Azure IoT auto-provisioning can be broken into three phases:

- Service configuration - a one-time configuration of the Azure IoT Hub and IoT Hub Device Provisioning Service instances, establishing them and creating linkage between them.

- Device enrollment - the process of making the Device Provisioning Service instance aware of the devices that will attempt to register in the future.Note: The Device Provisioning Service is a helper service that enables just-in-timeprovisioning of devices to an IoT hub, without requiring human intervention. After successful provisioning, devices connect directly with their designated IoT Hub. This process is referred to as auto-provisioning, and provides an out-of-thebox registration and initial configuration experience for devices.

- Device registration and configuration - initiated upon boot up by registration software, which is built using a Device Provisioning Service client SDK appropriate for the device and attestation mechanism. The software establishes aconnection to the provisioning service for authentication of the device, and subsequent registration in the IoT Hub. Upon successful registration, the device is provided with its IoT Hub unique device ID and connection information, allowing it to pull its initial configuration and begin the telemetry process. In production environments, this phase can occur weeks or months after the previous two phases.

References: https://docs.microsoft.com/en-us/azure/iot-dps/concepts-auto-provisioning

QUESTION 7 You are creating an IoT solution using Azure Time Series Insights.

You configure the environment to ensure that all data for the current year is available.

What should you do?

A. Create a reference data set.
B. Add a disaster recovery (DR) strategy.
C. Change the pricing tier.
D. Set a value for the Data Retention time setting.

Correct Answer: D
Section: [none]
Explanation

Explanation/Reference:

Explanation:

The data is retained in Time Series Insights based on the selected retention days or maximum limits.

Retention is configurable in the Azure portal. The longest allowable retention period is a rolling year of 12 months + 1 month, which is defined as 400 days.

References:

https://azure.microsoft.com/en-us/pricing/details/time-series-insights/

QUESTION 8

DRAG DROP

You develop a bot by using Language Understanding Intelligence Service (LUIS) and the .NET Bot framework. You use LUIS in the Azure portal to optimize the bot.

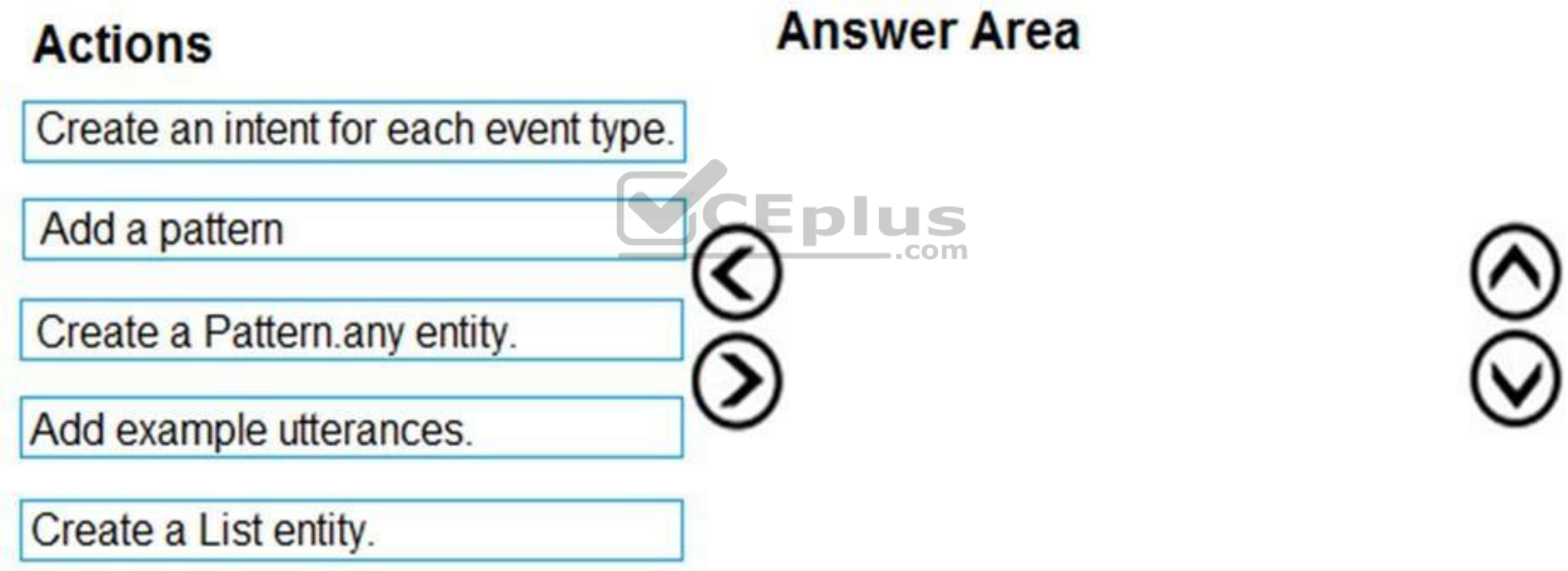
You review the utterances and determine that users are requesting time and venue information for events.

You need to improve the prediction efficiency of the bot.

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

NOTE: Each correct selection is worth one point.

Select and Place:



Actions

Create an intent for each event type.

Add a pattern

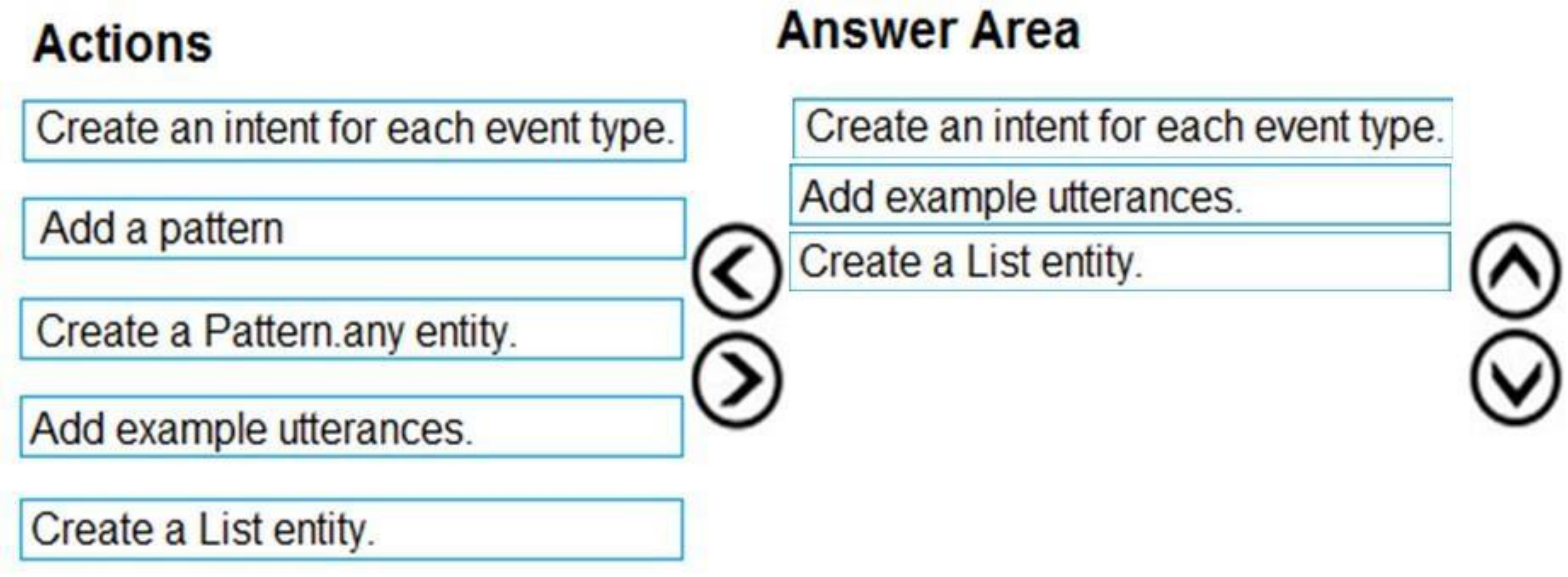
Create a Pattern.any entity.

Add example utterances.

Create a List entity.

Answer Area

Correct Answer:



Actions

- Create an intent for each event type.
- Add a pattern
- Create a Pattern.any entity.
- Add example utterances.
- Create a List entity.

Answer Area

- Create an intent for each event type.
- Add example utterances.
- Create a List entity.

Section: [none]
Explanation

Explanation/Reference:
Explanation:
Step 1: Create an intent for each event type
Identify your intents

Step 2: Add example utterances
Create example utterances for each intent

Step 3: Create a List Entity
Identify your entities
A list entity is an explicitly specified list of values. Each value consists of one or more synonyms. In a travel app, you might choose to create a list entity to represent airport names.

References:
https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-plan-your-app