# AZ-203.20q

Number: AZ-203
Passing Score: 800
Time Limit: 120 min

**AZ-203**

**Developing Solutions for Microsoft Azure**

**Question Set 1**

**QUESTION 1**

You are writing code to create and run an Azure Batch job.

You have created a pool of compute nodes.

You need to choose the right class and its method to submit a batch job to the Batch service.

Which method should you use?

A. `JobOperations.EnableJobAsync(String, IEnumerable<BatchClientBehavior>,CancellationToken)`

B. `JobOperations.CreateJob()`

C. `CloudJob.Enable(IEnumerable<BatchClientBehavior>)`

D. `JobOperations.EnableJob(String,IEnumerable<BatchClientBehavior>)`

E. `CloudJob.CommitAsync(IEnumerable<BatchClientBehavior>, CancellationToken)`

**Correct Answer:** E

**Section: [none]**

**Explanation**

**Explanation/Reference:**

Explanation:

A Batch job is a logical grouping of one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. The app uses the BatchClient.JobOperations.CreateJob method to create a job on your pool.

The Commit method submits the job to the Batch service. Initially the job has no tasks.

```
{
    CloudJob job = batchClient.JobOperations.CreateJob();
job.Id = JobId;
    job.PoolInformation = new PoolInformation { PoolId = PoolId };

    job.Commit();
}
...
```

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

## Question Set 1

**QUESTION 1**
You develop a website. You plan to host the website in Azure. You expect the website to experience high traffic volumes after it is published.

You must ensure that the website remains available and responsive while minimizing cost.

You need to deploy the website.

What should you do?

A. Deploy the website to a virtual machine. Configure the virtual machine to automatically scale when the CPU load is high.
B. Deploy the website to an App Service that uses the Shared service tier. Configure the App service plan to automatically scale when the CPU load is high.
C. Deploy the website to an App Service that uses the Standard service tier. Configure the App service plan to automatically scale when the CPU load is high.
D. Deploy the website to a virtual machine. Configure a Scale Set to increase the virtual machine instance count when the CPU load is high.

**Correct Answer:** C
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Windows Azure Web Sites (WAWS) offers 3 modes: Standard, Free, and Shared.

Standard mode carries an enterprise-grade SLA (Service Level Agreement) of 99.9% monthly, even for sites with just one instance.
Standard mode runs on dedicated instances, making it different from the other ways to buy Windows Azure Web Sites.

Incorrect Answers:
B: Shared and Free modes do not offer the scaling flexibility of Standard, and they have some important limits.
Shared mode, just as the name states, also uses shared Compute resources, and also has a CPU limit. So, while neither Free nor Shared is likely to be the best choice for your production environment due to these limits.

**QUESTION 2**
You develop a serverless application using several Azure Functions. These functions connect to data from within the code.

You want to configure tracing for an Azure Function App project.

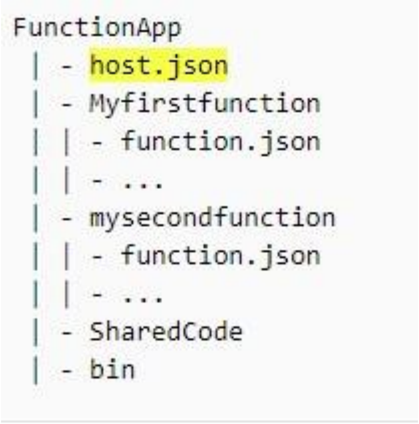
You need to change configuration settings in the host.json file.

Which tool should you use?

A. Visual Studio
B. Azure portal
C. Azure PowerShell
D. Azure Functions Core Tools (Azure CLI)

**Correct Answer:** B
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
The function editor built into the Azure portal lets you update the function.json file and the code file for a function. The host.json file, which contains some runtimespecific configurations, is in the root folder of the function app.



```
FunctionApp
 | - host.json
 | - Myfirstfunction
 | | - function.json
 | | - ...
 | - mysecondfunction
 | | - function.json
 | | - ...
 | - SharedCode
 | - bin
```

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

**QUESTION 3**
You are developing a mobile instant messaging app for a company.

The mobile app must meet the following requirements:

- Support offline data sync.
- Update the latest messages during normal sync cycles.

You need to implement Offline Data Sync.

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

**NOTE:** Each correct selection is worth one point.

A. Retrieve records from Offline Data Sync on every call to the PullAsync method.
B. Retrieve records from Offline Data Sync using an Incremental Sync.
C. Push records to Offline Data Sync using an Incremental Sync.
D. Return the updatedAt column from the Mobile Service Backend and implement sorting by using the column.
E. Return the updatedAt column from the Mobile Service Backend and implement sorting by the message id.

**Correct Answer:** BE
**Section: [none]**
**Explanation**

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

E (not D): To use incremental sync, your server must return meaningful updatedAt values and must also support sorting by this field. However, since the SDK adds its own sort on the updatedAt field, you cannot use a pull query that has its own orderBy clause.

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

**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.

**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:

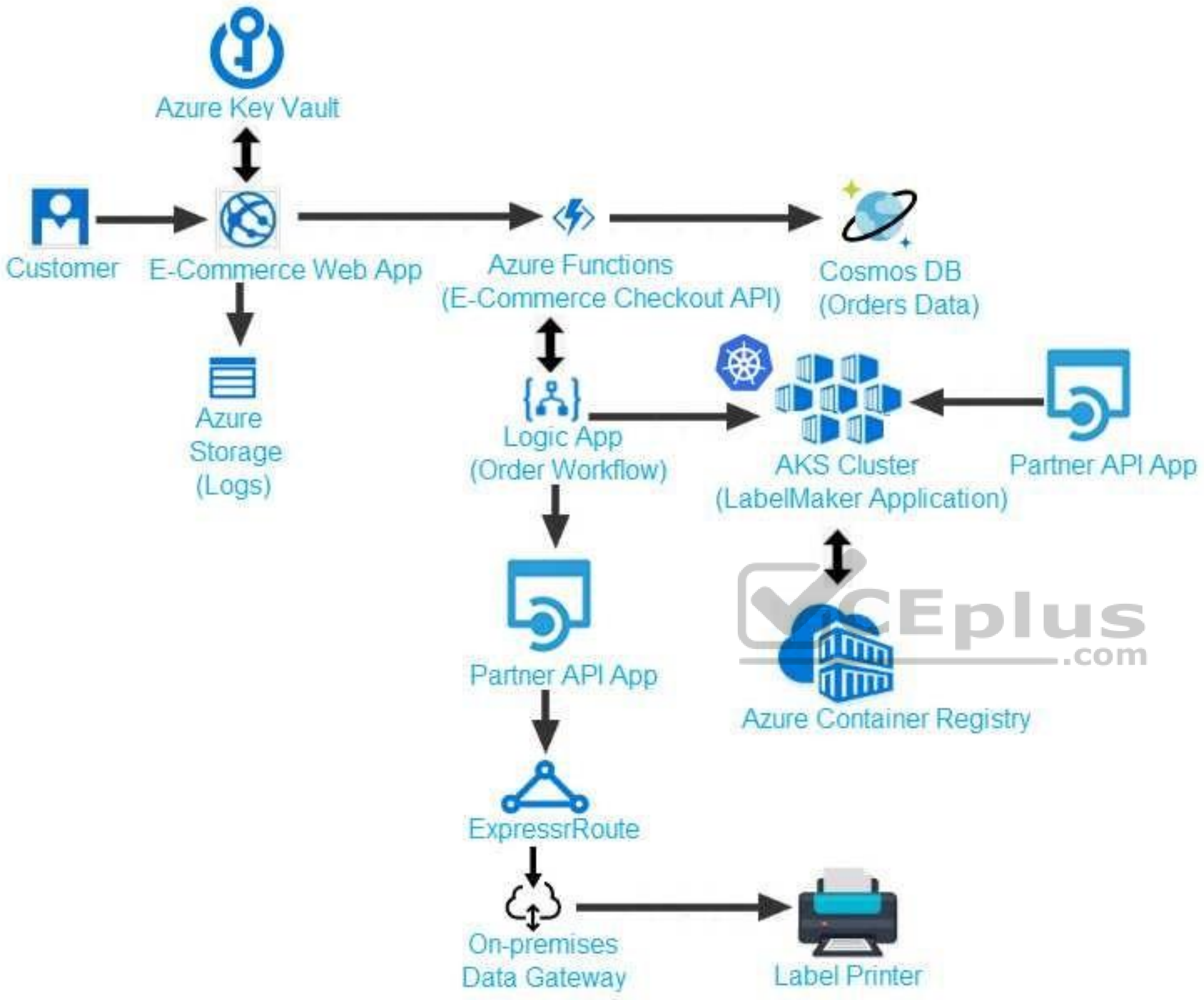
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization’s identity management solution.
- External partner logins must be audited monthly 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.

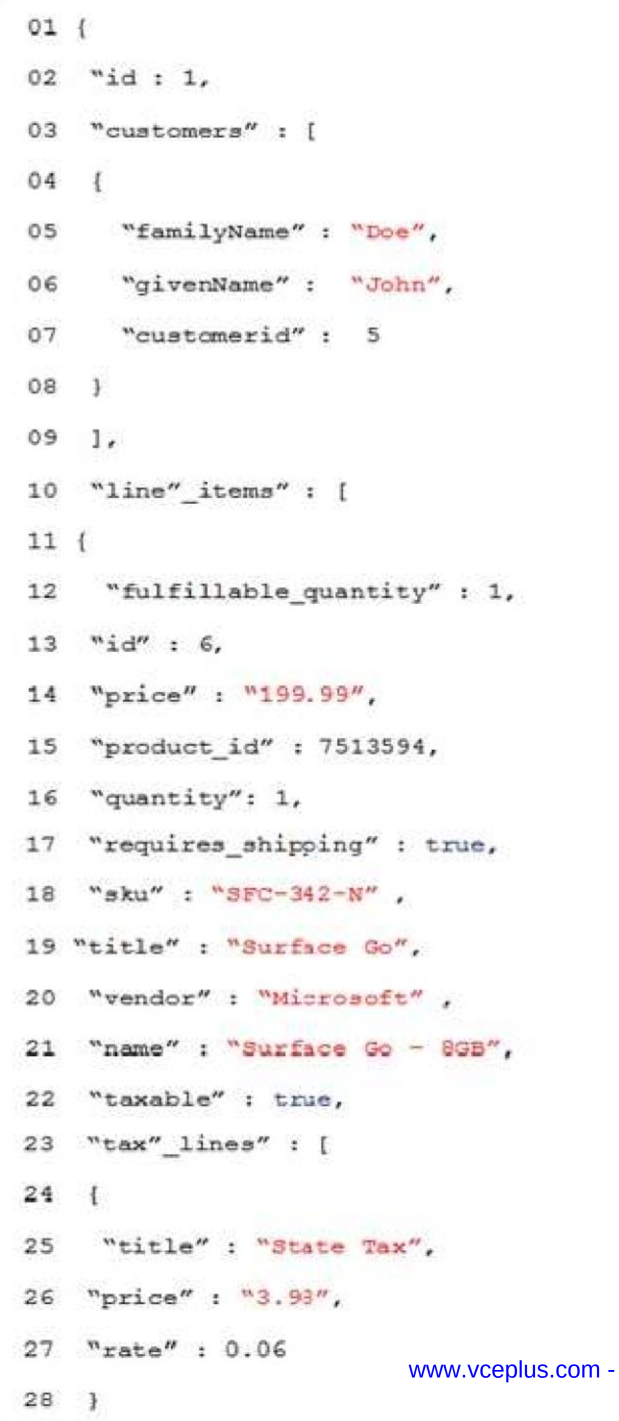


Calls to the Printer API App fail periodically due to printer communication timeouts.

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**



```
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,
23  “tax”_lines” : [
24   {
25    “title” : “State Tax”,
26  “price” : “3.98”,
27  “rate” : 0.06
28   }
```

**QUESTION 1**
You need to implement the e-commerce checkout API.

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

**NOTE:** Each correct selection is worth one point.

A. Set the function template's Mode property to **Webhook** and the Webhook type property to **Generic JSON**.
B. Create an Azure Function using the HTTP POST function template.
C. In the Azure Function App, enable Cross-Origin Resource Sharing (CORS) with all origins permitted.
D. In the Azure Function App, enable Managed Service Identity (MSI).
E. Set the function template's Mode property to **Webhook** and the Webhook type property to **GitHub**.
F. Create an Azure Function using the Generic webhook function template.

**Correct Answer:** ABD
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Scenario: E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).

D: A managed identity from Azure Active Directory allows your app to easily access other AAD-protected resources such as Azure Key Vault.

Incorrect Answers:
C: CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain.

References:
https://docs.microsoft.com/en-us/azure/app-service/overview-managed-identity

**Question Set 1**

**QUESTION 1**
You use Azure Table storage to store customer information for an application. The data contains customer details and is partitioned by last name.

You need to create a query that returns all customers with the last name Smith.

Which code segment should you use?

A. `TableQuery.GenerateFilterCondition("PartitionKey", Equals, "Smith")`

B. `TableQuery.GenerateFilterCondition("LastName", Equals, "Smith")`

C. `TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "Smith")`

D. `TableQuery.GenerateFilterCondition("LastName", QueryComparisons.Equal, "Smith")`

**Correct Answer:** C
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Retrieve all entities in a partition. The following code example specifies a filter for entities where 'Smith' is the partition key. This example prints the fields of each entity in the query results to the console.
Construct the query operation for all customer entities where PartitionKey="Smith".

TableQuery<CustomerEntity> query = new TableQuery<CustomerEntity>().Where(TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "Smith"));

References: https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet **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.

**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 principle of least privilege and provide privileges which are essential to perform the intended function.

- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI) ▪
Receipt data must always be encrypted at rest.
- All data must be protected in transit
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment, with the remaining parts obscured.
- In the case of a security breach access to all summary reports must be revoked without impacting other parts of the system.

## Issues

### Upload format issue

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

### Capacity issue

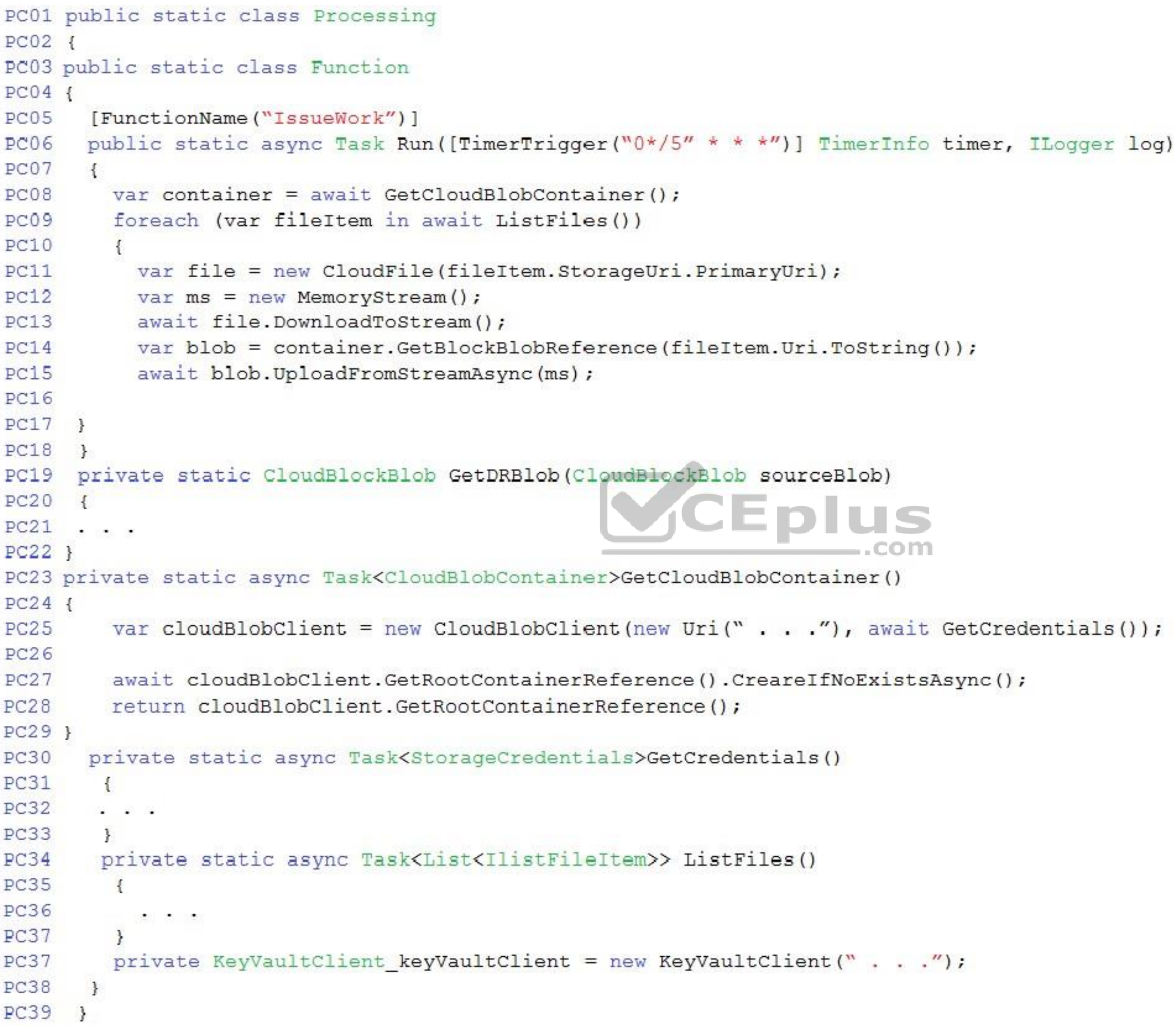
During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

### Log capacity issue

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

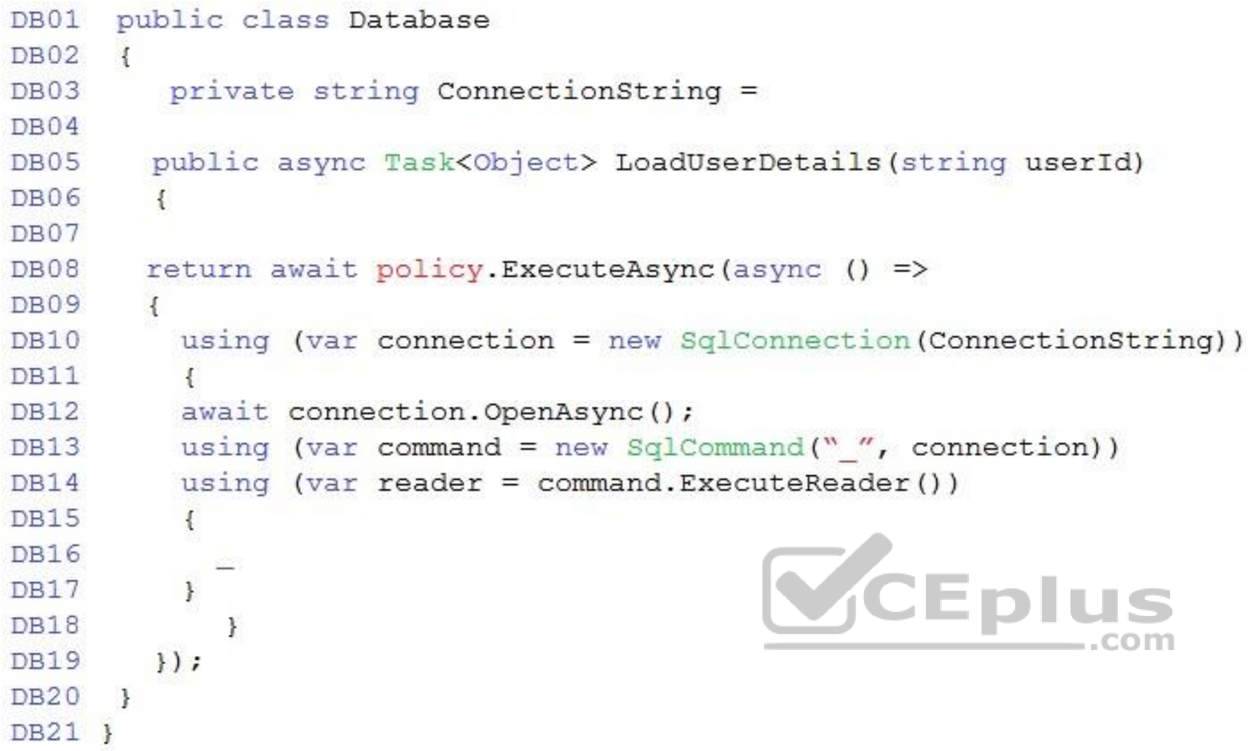
## Application code

### Processing.cs



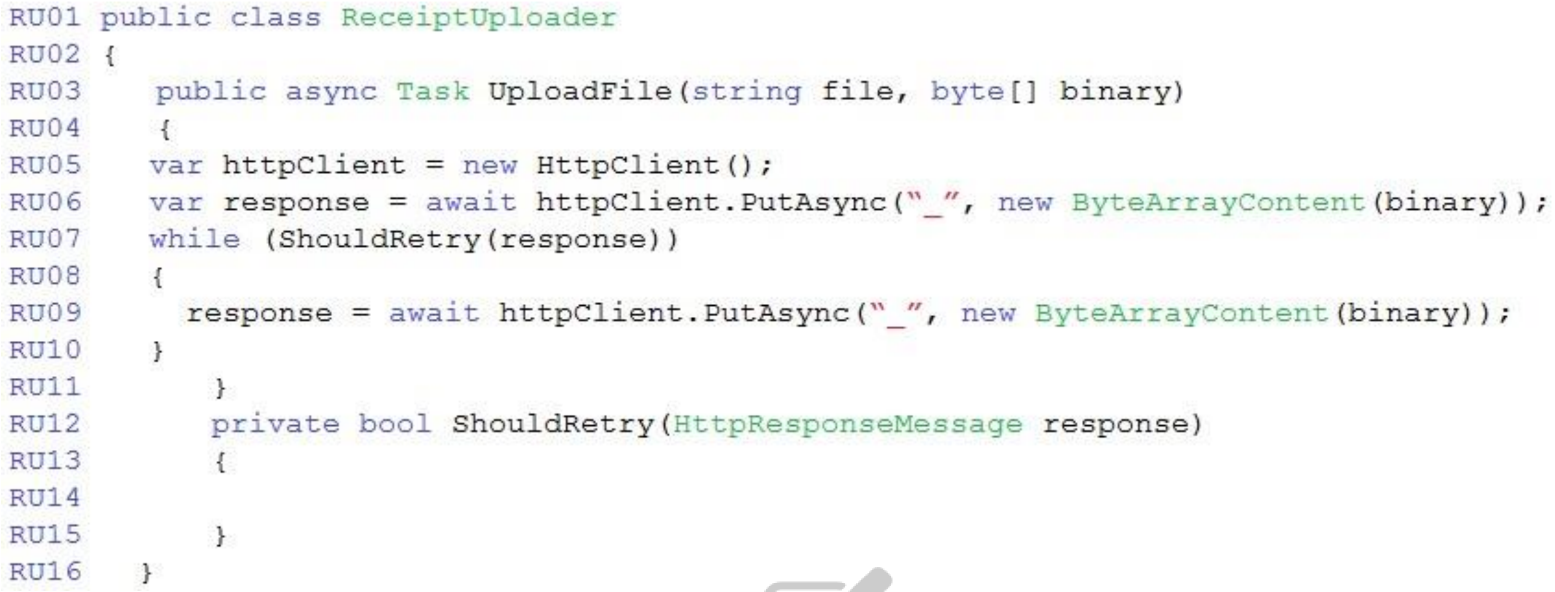
```
PC01 public static class Processing
PC02 {
PC03 public static class Function
PC04 {
PC05  [FunctionName(“IssueWork”)]
PC06  public static async Task Run([TimerTrigger(“0*/5” * * *”)] TimerInfo timer, ILogger log)
PC07  {
PC08   var container = await GetCloudBlobContainer();
PC09   foreach (var fileItem in await ListFiles())
PC10   {
PC11    var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12    var ms = new MemoryStream();
PC13    await file.DownloadToStream();
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().CreareIfNoExistsAsync();
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**



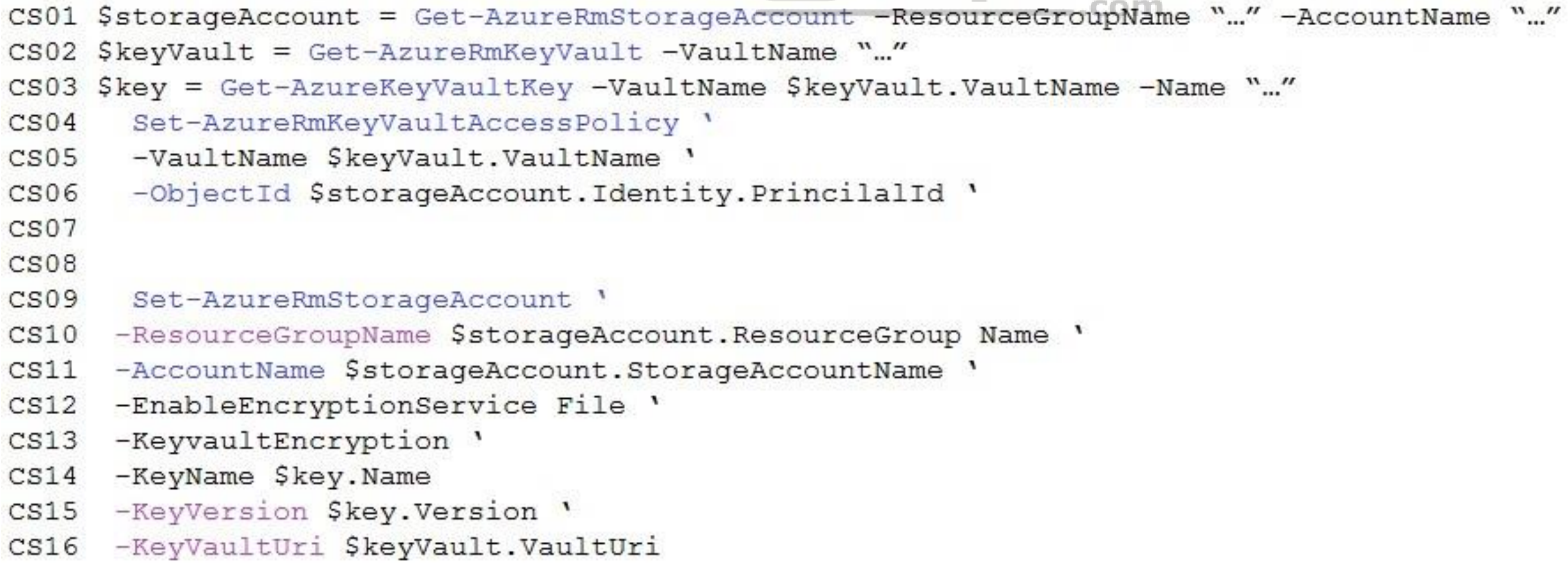
```
DB01  public class Database
DB02  {
DB03     private string ConnectionString =
DB04
DB05    public async Task<Object> LoadUserDetails(string userId)
DB06    {
DB07
DB08    return await policy.ExecuteAsync(async () =>
DB09    {
DB10      using (var connection = new SqlConnection(ConnectionString))
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.PrincilalId `
CS07
CS08
CS09   Set-AzureRmStorageAccount `
CS10  -ResourceGroupName $storageAccount.ResourceGroup Name `
CS11  -AccountName $storageAccount.StorageAccountName `
CS12  -EnableEncryptionService File `
CS13  -KeyvaultEncryption `
CS14  -KeyName $key.Name
CS15  -KeyVersion $key.Version `
CS16  -KeyVaultUri $keyVault.VaultUri
```

**QUESTION 1**

You need to construct the link to the summary report for the email that is sent to users.

What should you do?

A. Create a SharedAccessBlobPolicy and add it to the containers SharedAccessPolicies. Call GetSharedAccessSignature on the blob and use the resulting link.
B. Create a SharedAccessAccountPolicy and call GetSharedAccessSignature on storage account and use the resulting link.
C. Create a SharedAccessBlobPolicy and set the expiry time to two weeks from today. Call GetSharedAccessSignature on the blob and use the resulting link.
D. Create a SharedAccessBlobPolicy and set the expiry time to two weeks from today. Call GetSharedAccessSignature on the container and use the resulting link.

**Correct Answer:** D
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Scenario: 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.

Create a stored access policy to manage signatures on a container's resources, and then generate the shared access signature on the container, setting the constraints directly on the signature.

Code example: Add a method that generates the shared access signature for the container and returns the signature URI.

```
static string GetContainerSasUri(CloudBlobContainer container)
{
   //Set the expiry time and permissions for the container.
   //In this case no start time is specified, so the shared access signature becomes valid immediately.
   SharedAccessBlobPolicy sasConstraints = new SharedAccessBlobPolicy();
sasConstraints.SharedAccessExpiryTime = DateTimeOffset.UtcNow.AddHours(24);
   sasConstraints.Permissions = SharedAccessBlobPermissions.List | SharedAccessBlobPermissions.Write;

   //Generate the shared access signature on the container, setting the constraints directly on the signature.
   string sasContainerToken = container.GetSharedAccessSignature(sasConstraints);

   //Return the URI string for the container, including the SAS token.
   return container.Uri + sasContainerToken;
```

}

Incorrect Answers:
C: Call GetSharedAccessSignature on the container, not on the blob.

References: https://docs.microsoft.com/en-us/azure/storage/blobs/storage-dotnet-shared-access-signature-part-2

**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:

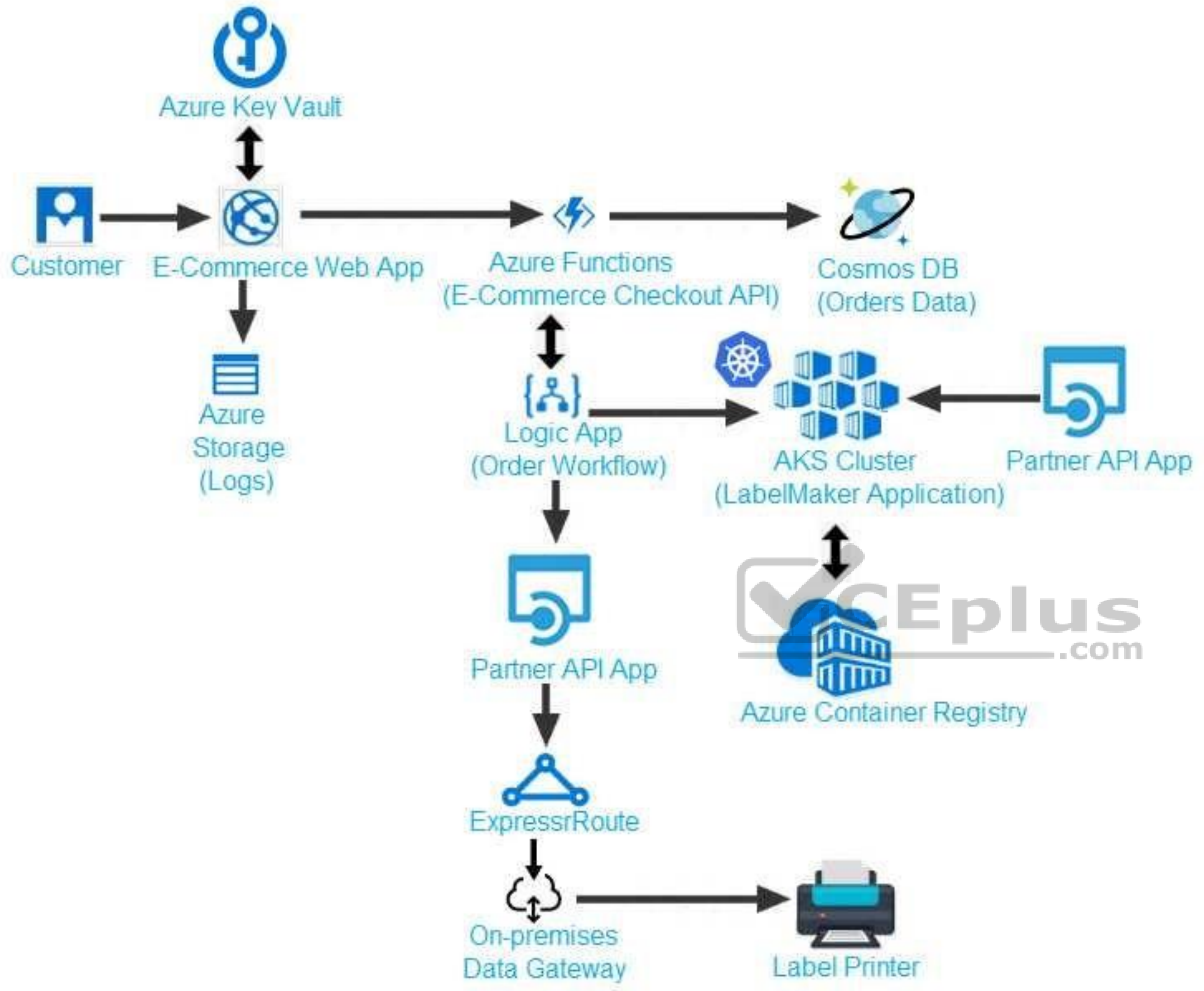
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization’s identity management solution.
- External partner logins must be audited monthly 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.

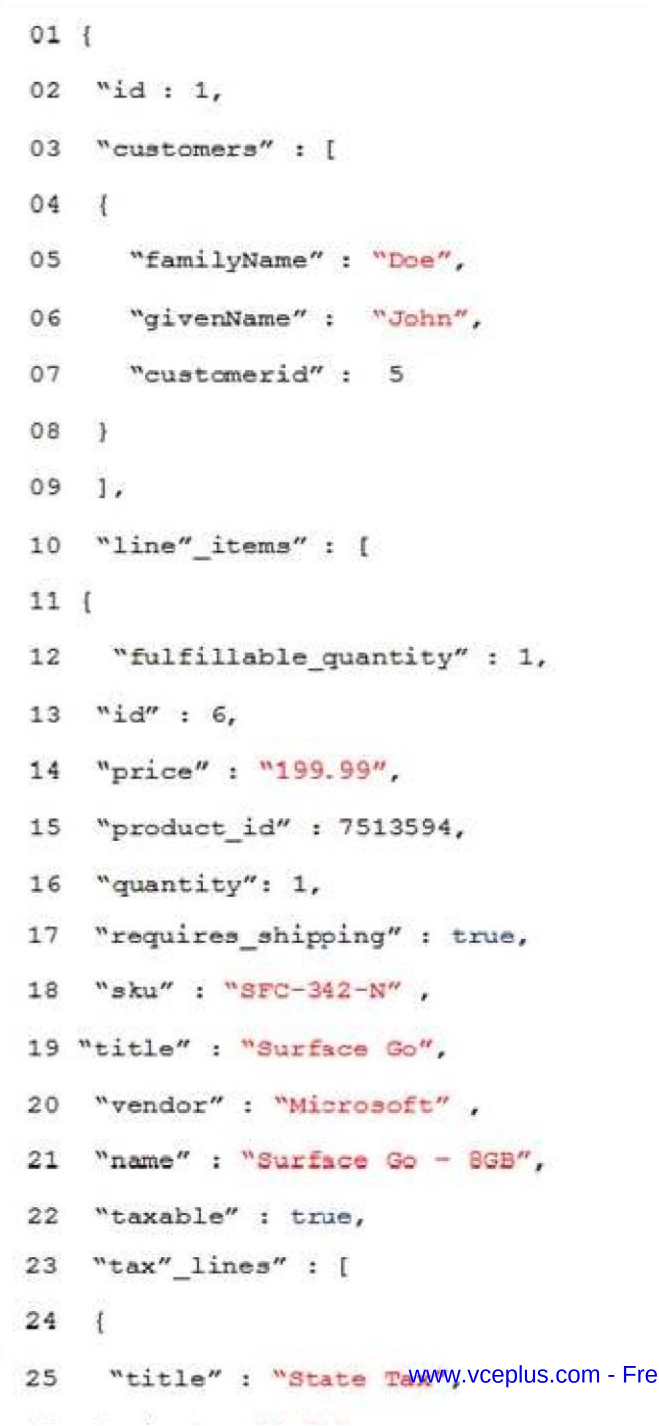


Calls to the Printer API App fail periodically due to printer communication timeouts.

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**



```
{
 "id : 1,
 "customers" : [
 {
   "familyName" : "Doe",
   "givenName" :  "John",
   "customerid" :  5
 }
 ],
 "line"_items" : [
{
  "fulfillable_quantity" : 1,
 "id" : 6,
 "price" : "199.99",
 "product_id" : 7513594,
 "quantity": 1,
 "requires_shipping" : true,
 "sku" : "SFC-342-N" ,
"title" : "Surface Go",
 "vendor" : "Microsoft" ,
 "name" : "Surface Go - 8GB",
 "taxable" : true,
 "tax"_lines" : [
 {
  "title" : "State Tax",
```

**QUESTION 1**

You need to meet the LabelMaker application security requirement.

What should you do?

A. Create a conditional access policy and assign it to the Azure Kubernetes Service cluster.
B. Place the Azure Active Directory account into an Azure AD group. Create a ClusterRoleBinding and assign it to the group.
C. Create a RoleBinding and assign it to the Azure AD account.
D. Create a Microsoft Azure Active Directory service principal and assign it to the Azure Kubernetes Service (AKS) cluster.

**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.

**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 principle of least privilege and provide privileges which are essential to perform the intended function.

- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI) ▪
Receipt data must always be encrypted at rest.
- All data must be protected in transit
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment, with the remaining parts obscured.
- In the case of a security breach access to all summary reports must be revoked without impacting other parts of the system.

**Issues**

**Upload format issue**

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

**Capacity issue**

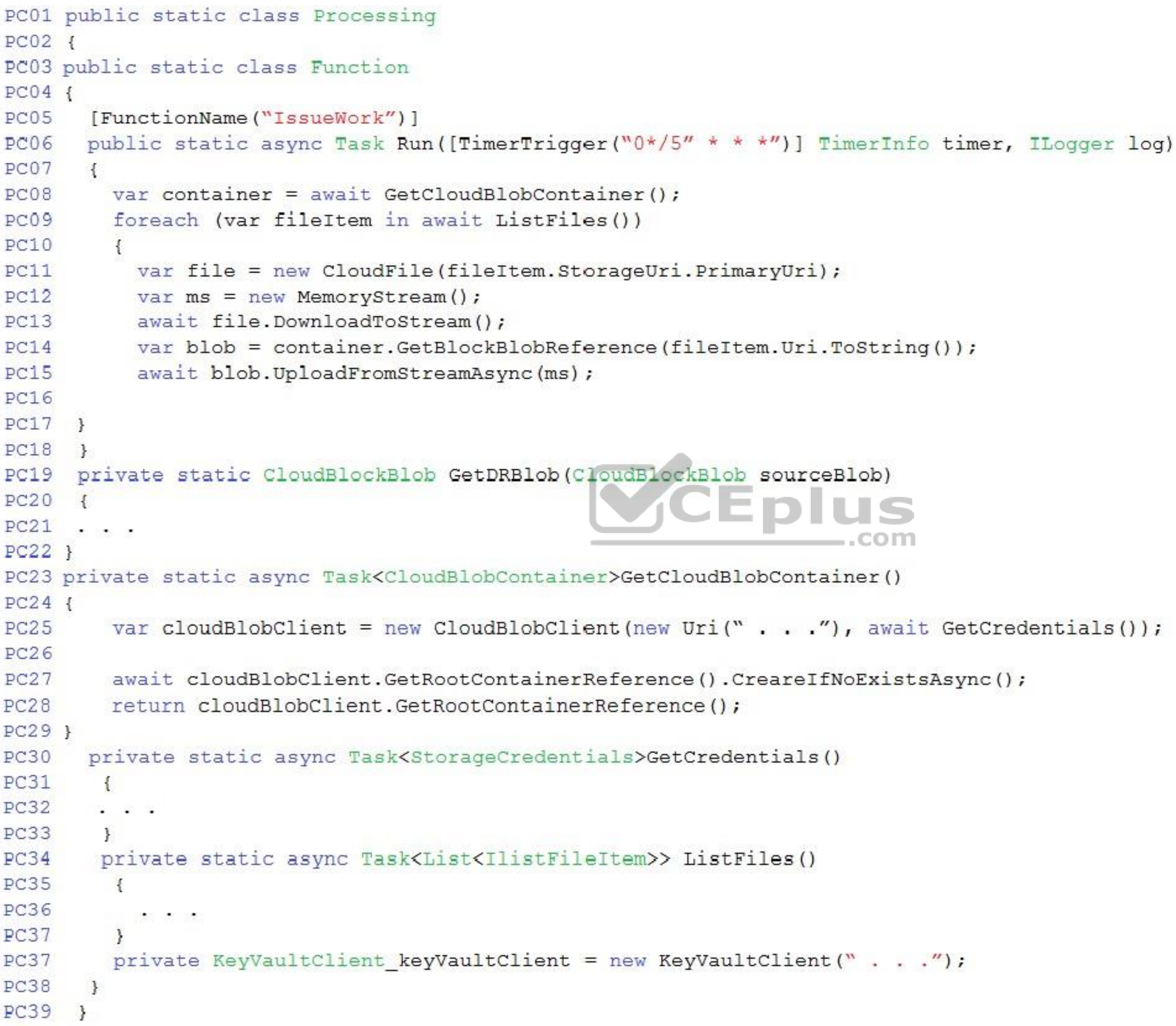
During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

### Log capacity issue

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

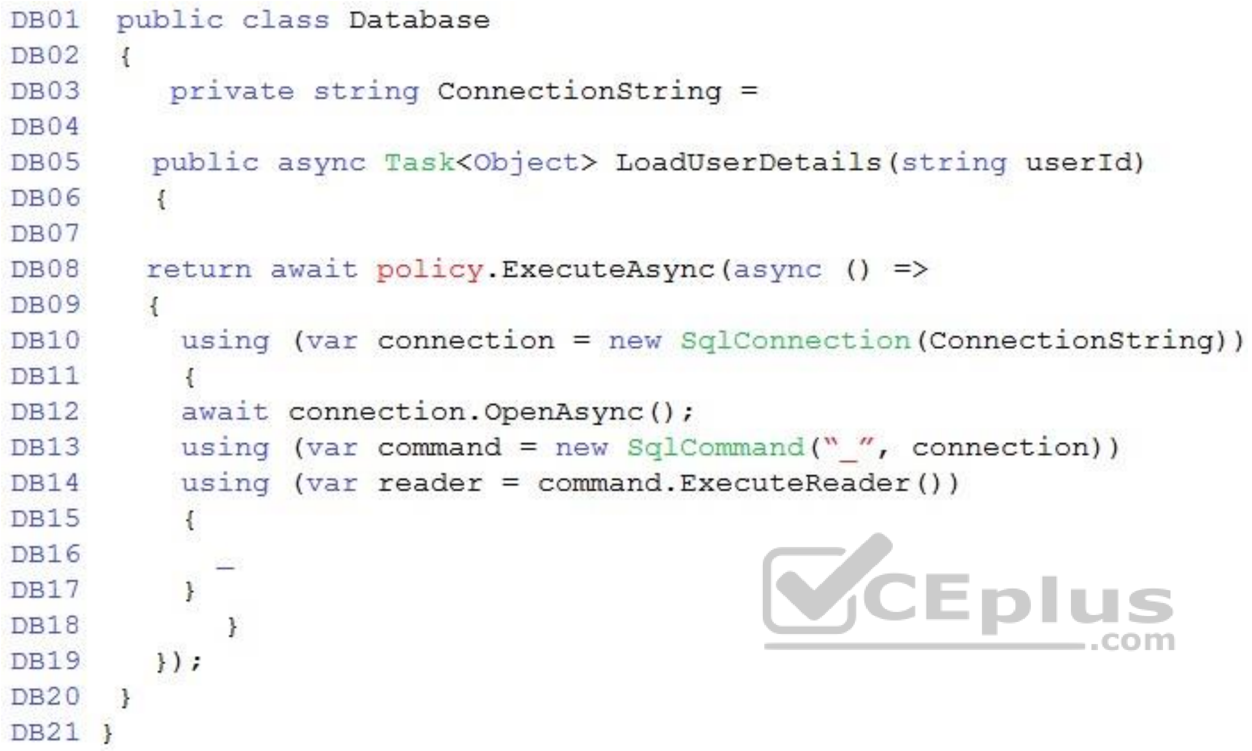
### Application code

### Processing.cs



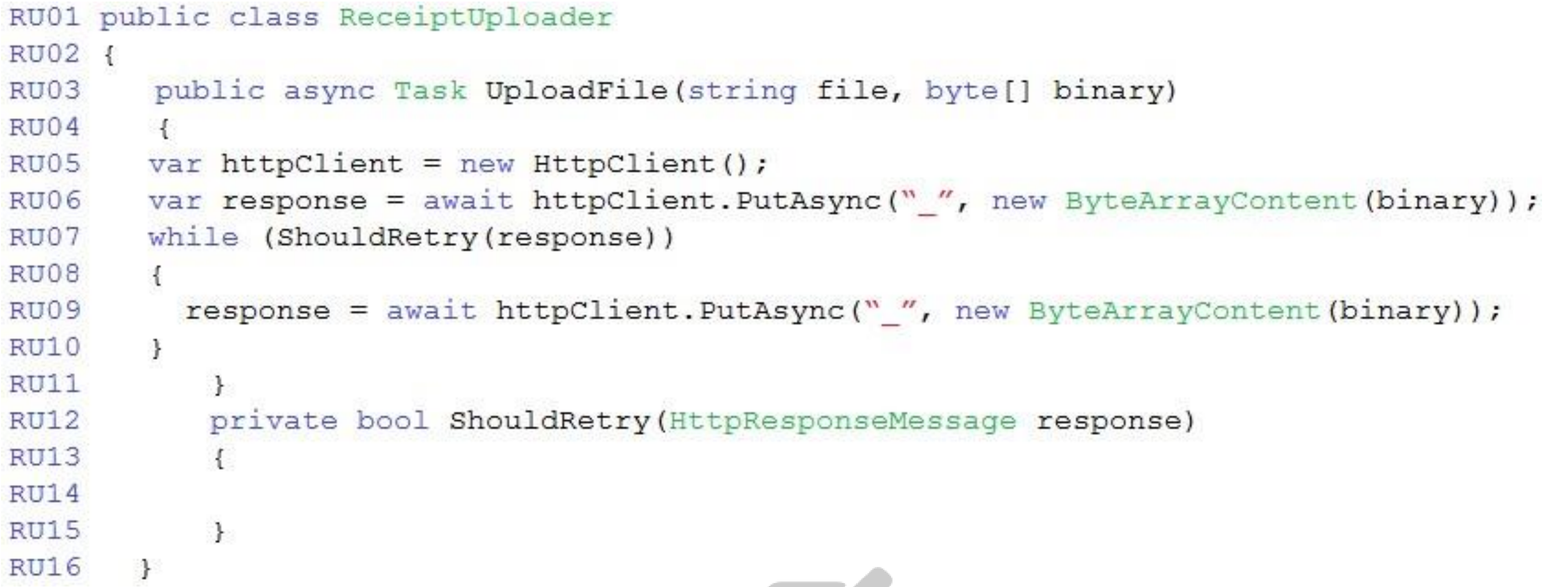
```
PC01 public static class Processing
PC02 {
PC03 public static class Function
PC04 {
PC05   [FunctionName(“IssueWork”)]
PC06   public static async Task Run([TimerTrigger(“0*/5” * * *”)] TimerInfo timer, ILogger log)
PC07   {
PC08     var container = await GetCloudBlobContainer();
PC09     foreach (var fileItem in await ListFiles())
PC10     {
PC11       var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12       var ms = new MemoryStream();
PC13       await file.DownloadToStream();
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().CreareIfNoExistsAsync();
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**



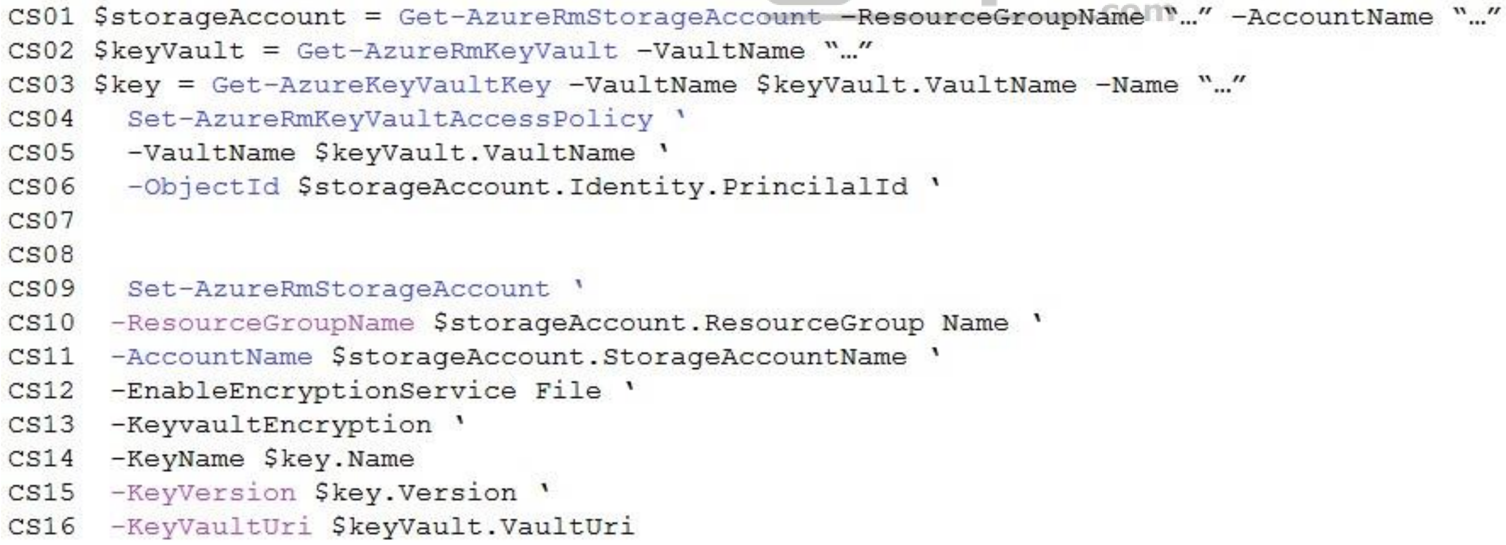
```
DB01  public class Database
DB02  {
DB03      private string ConnectionString =
DB04
DB05     public async Task<Object> LoadUserDetails(string userId)
DB06     {
DB07
DB08     return await policy.ExecuteAsync(async () =>
DB09     {
DB10       using (var connection = new SqlConnection(ConnectionString))
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.PrincilalId `
CS07
CS08
CS09   Set-AzureRmStorageAccount `
CS10  -ResourceGroupName $storageAccount.ResourceGroup Name `
CS11  -AccountName $storageAccount.StorageAccountName `
CS12  -EnableEncryptionService File `
CS13  -KeyvaultEncryption `
CS14  -KeyName $key.Name
CS15  -KeyVersion $key.Version `
CS16  -KeyVaultUri $keyVault.VaultUri
```

**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 that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**

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

You need to ensure that the SecurityPin security requirements are met.

Solution: Enable Always Encrypted for the SecurityPin column using a certificate based on a trusted certificate authority. Update the Getting Started document with instructions to ensure that the certificate is installed on user machines.

Does the solution meet the goal?

A. Yes
B. No

**Correct Answer:** B
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:

Enable Always Encrypted is correct, but only the WebAppIdentity service principal should be given access to the certificate.

Scenario: 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.

**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 that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**

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

You need to ensure that the SecurityPin security requirements are met.

Solution: Using the Azure Portal, add Data Masking to the SecurityPin column, and exclude the dbo user. Add a SQL security policy with a filter predicate based on the user identity.

Does the solution meet the goal?

A. Yes

B. No

**Correct Answer:** B
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:

Instead of DataMasing, enable Always Encrypted for the SecurityPin column.

Scenario: 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.

**QUESTION 3**
**Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**

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

You need to ensure that the SecurityPin security requirements are met.

Solution: Enable Always Encrypted for the SecurityPin column using a certificate contained in Azure Key Vault and grant the WebAppIdentity service principal access to the certificate.

Does the solution meet the goal?

A. Yes
B. No

**Correct Answer:** A
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Scenario: 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.

**QUESTION 4**
You need to ensure the security policies are met.

What code do you add at line CS07?

A. `-PermissionsToCertificates create, encrypt, decrypt`
B. `-PermissionsToKeys wrapkey, unwrapkey, get`
C. `-PermissionsToCertificates wrapkey, unwrapkey, get`
D. `-PermissionsToKeys create, encrypt, decrypt`

**Correct Answer:** A
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Scenario: All certificates and secrets used to secure data must be stored in Azure Key Vault.
You must adhere to the principle of least privilege and provide privileges which are essential to perform the intended function.

The Set-AzureRmKeyValutAccessPolicy parameter -PermissionsToKeys specifies an array of key operation permissions to grant to a user or service principal. The acceptable values for this parameter: decrypt, encrypt, unwrapKey, wrapKey, verify, sign, get, list, update, create, import, delete, backup, restore, recover, purge

Incorrect Answers:
A, C: The Set-AzureRmKeyValutAccessPolicy parameter -PermissionsToCertificates specifies an array of certificate permissions to grant to a user or service principal. The acceptable values for this parameter: get, list, delete, create, import, update, managecontacts, getissuers, listissuers, setissuers, deleteissuers, manageissuers, recover, purge, backup, restore

References: https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/set-azurermkeyvaultaccesspolicy **Question Set 3**

**QUESTION 1**
You are developing an internal website for employees to view sensitive data. The website uses Azure Active Directory (AAD) for authentication.

You need to implement multifactor authentication for the website.

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

**NOTE:** Each correct selection is worth one point.

A. Upgrade to Azure AD Premium.
B. In Azure AD conditional access, enable the baseline policy.
C. In Azure AD, create a new conditional access policy.

D. In Azure AD, enable application proxy.
E. Configure the website to use Azure AD B2C.

**Correct Answer:** AC
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:

A: Multi-Factor Authentication comes as part of the following offerings:
- Azure Active Directory Premium licenses - Full featured use of Azure Multi-Factor Authentication Service (Cloud) or Azure Multi-Factor Authentication Server (On-premises).
- Multi-Factor Authentication for Office 365
- Azure Active Directory Global Administrators

C: MFA Enabled by conditional access policy. It is the most flexible means to enable two-step verification for your users. Enabling using conditional access policy only works for Azure MFA in the cloud and is a premium feature of Azure AD.

References:
https://docs.microsoft.com/en-us/azure/active-directory/authentication/howto-mfa-getstarted

**Question Set 1**

**QUESTION 1**
You develop an Azure web app. You monitor performance of the web app by using Application Insights.

You need to ensure the cost for Application Insights does not exceed a preset budget.

What should you do?

A. Implement ingestions sampling using the Application Insights SDK.
B. Set a daily cap for the Application Insights instance.
C. Implement ingestion sampling using the Azure portal.
D. Implement adaptive sampling using the Azure portal.
E. Implement adaptive sampling using the Application Insights SDK.

**Correct Answer:** E
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Sampling is an effective way to reduce charges and stay within your monthly quota.

You can set sampling manually, either in the portal on the Usage and estimated costs page; or in the ASP.NET SDK in the .config file; or in the Java SDK in the ApplicationInsights.xml file, to also reduce the network traffic.

Adaptive sampling is the default for the ASP.NET SDK. Adaptive sampling automatically adjusts to the volume of telemetry that your app sends. It operates automatically in the SDK in your web app so that telemetry traffic on the network is reduced.

Incorrect Answers:
B: You can use the daily volume cap to limit the data collected.
To change the daily cap, in the Configure section of your Application Insights resource, in the Usage and estimated costs pane, select Daily Cap.

References: https://docs.microsoft.com/en-us/azure/azure-monitor/app/sampling

**QUESTION 2**
You are developing an ASP.NET Core Web API web service. The web service uses Azure Application Insights for all telemetry and dependency tracking. The web service reads and writes data to a database other than Microsoft SQL Server.

You need to ensure that dependency tracking works for calls to the third-party database.

Which two Dependency Telemetry properties should you store in the database? Each correct answer presents part of the **solution**.

**NOTE:** Each correct selection is worth one point.

A. Telemetry.Context.Operation.Id
B. Telemetry.Name
C. Telemetry.Context.Cloud.RoleInstance
D. Telemetry.Context.Session.Id
E. Telemetry.Id

**Correct Answer:** AE
**Section: [none]**
**Explanation**

**Explanation/Reference:**
References:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking

## 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.

### 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.

### Other

### Anomaly detection service

You have an anomaly detection service that analyzes log infrmation 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.

### Issues

### 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.

## 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

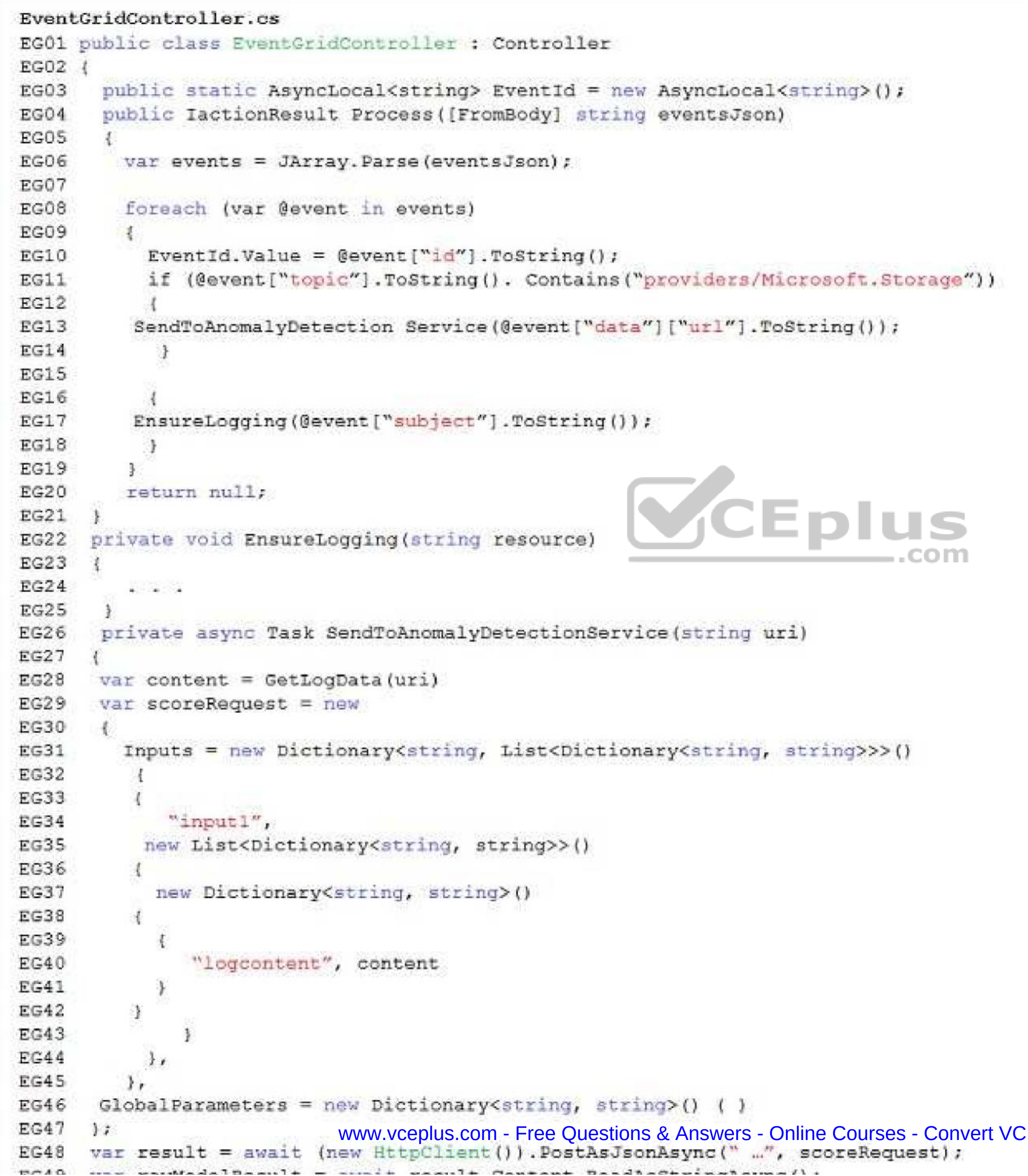
Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

### 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:

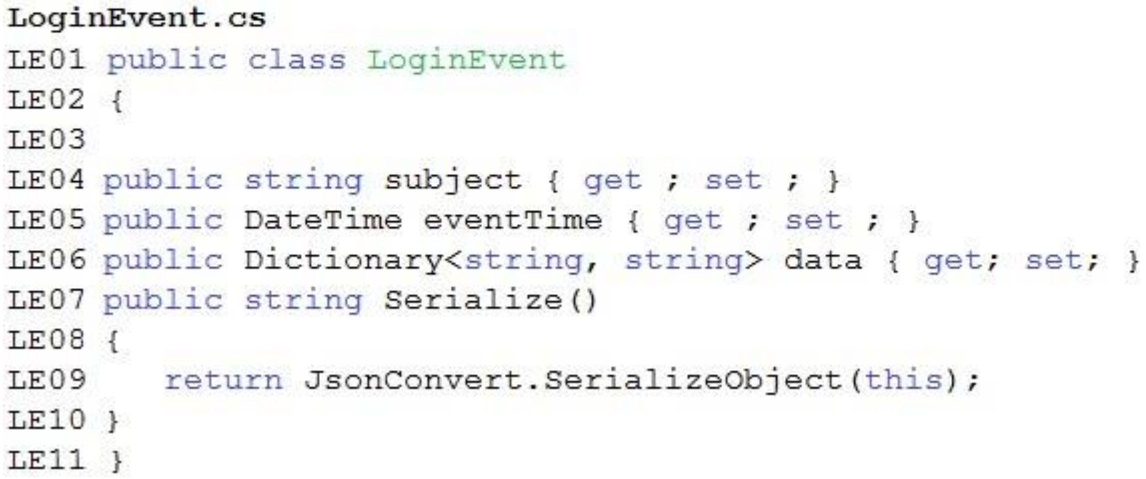
- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself.
- Ensure that scaling actions do not disrupt application usage.

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      SendToAnomalyDetection Service(@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();
```

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 meet the scaling requirements for Policy Service.

What should you store in Azure Redis Cache?

A. TempData
B. HttpContext.Items
C. ViewState
D. Session state

**Correct Answer:** D
**Section: [none]**
**Explanation**

**Explanation/Reference:**

Explanation:
Azure Cache for Redis provides a session state provider that you can use to store your session state in-memory with Azure Cache for Redis instead of a SQL Server database.

Scenario: You have a shared library named PolicyLib that contains functionality common to all ASP.NET Core web services and applications. The PolicyLib library must:
Ensure that scaling actions do not disrupt application usage.

References:
https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-aspnet-session-state-provider

## Testlet 3

### 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.

### 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 principle of least privilege and provide privileges which are essential to perform the intended function.

- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI) ▪
Receipt data must always be encrypted at rest.
- All data must be protected in transit
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment, with the remaining parts obscured.
- In the case of a security breach access to all summary reports must be revoked without impacting other parts of the system.

**Issues**

**Upload format issue**

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

**Capacity issue**

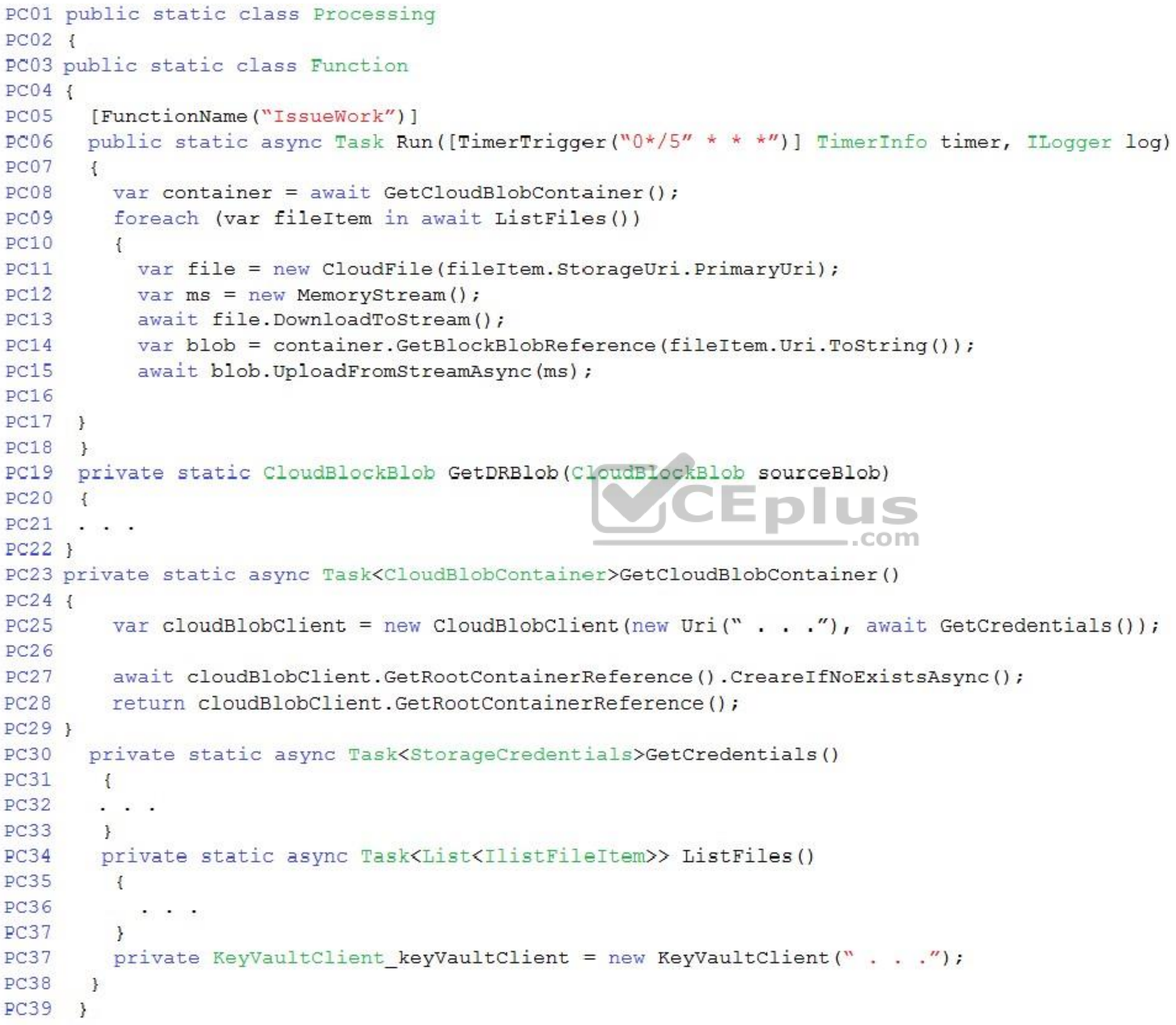
During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

**Log capacity issue**

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

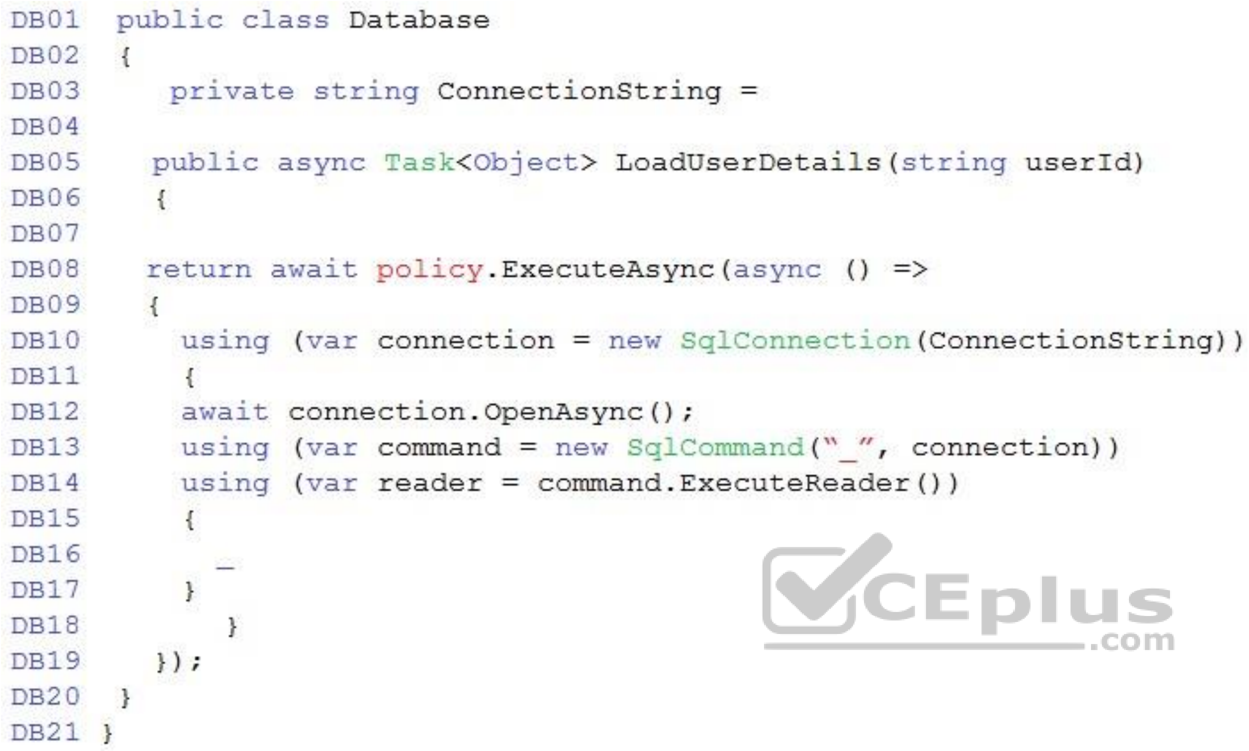
**Application code**

**Processing.cs**



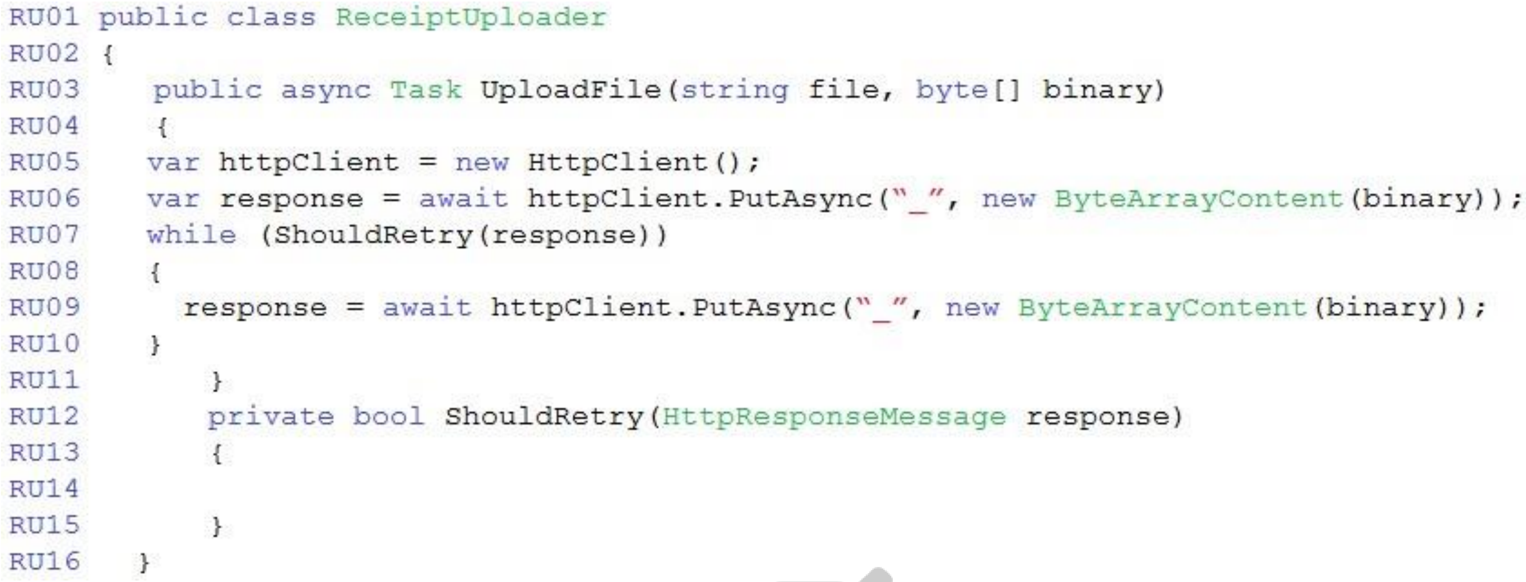
```
PC01 public static class Processing
PC02 {
PC03 public static class Function
PC04 {
PC05   [FunctionName("IssueWork")]
PC06   public static async Task Run([TimerTrigger("0*/5" * * *")] TimerInfo timer, ILogger log)
PC07   {
PC08     var container = await GetCloudBlobContainer();
PC09     foreach (var fileItem in await ListFiles())
PC10     {
PC11       var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12       var ms = new MemoryStream();
PC13       await file.DownloadToStream();
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().CreareIfNoExistsAsync();
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**



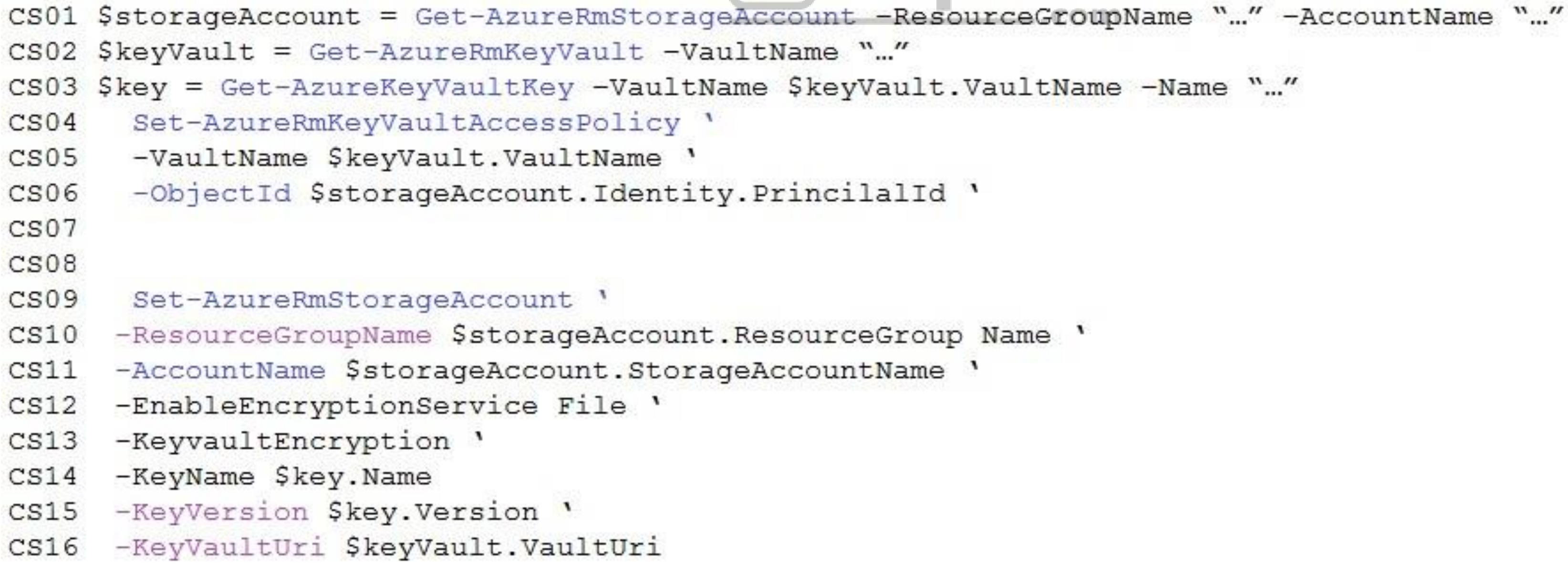
```
DB01  public class Database
DB02  {
DB03     private string ConnectionString =
DB04
DB05    public async Task<Object> LoadUserDetails(string userId)
DB06    {
DB07
DB08    return await policy.ExecuteAsync(async () =>
DB09    {
DB10      using (var connection = new SqlConnection(ConnectionString))
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.PrincilalId `
CS07
CS08
CS09   Set-AzureRmStorageAccount `
CS10  -ResourceGroupName $storageAccount.ResourceGroup Name `
CS11  -AccountName $storageAccount.StorageAccountName `
CS12  -EnableEncryptionService File `
CS13  -KeyvaultEncryption `
CS14  -KeyName $key.Name
CS15  -KeyVersion $key.Version `
CS16  -KeyVaultUri $keyVault.VaultUri
```

**QUESTION 1**

You need to resolve the log capacity issue.

What should you do?

A. Set a LogCategoryFilter during startup.
B. Create an Application Insights Telemetry Filter.
C. Change the minimum log level in the host.json file for the function.
D. Implement Application Insights Sampling.

**Correct Answer:** D
**Section: [none]**
**Explanation**

**Explanation/Reference:**

Explanation:
Scenario, the log capacity issue: Developers report that the number of log message in the trace output for the processor is too high, resulting in lost log messages.

Sampling is a feature in Azure Application Insights. It is the recommended way to reduce telemetry traffic and storage, while preserving a statistically correct analysis of application data. The filter selects items that are related, so that you can navigate between items when you are doing diagnostic investigations. When metric counts are presented to you in the portal, they are renormalized to take account of the sampling, to minimize any effect on the statistics.

Sampling reduces traffic and data costs, and helps you avoid throttling.

References:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/sampling

## Testlet 4

### 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:

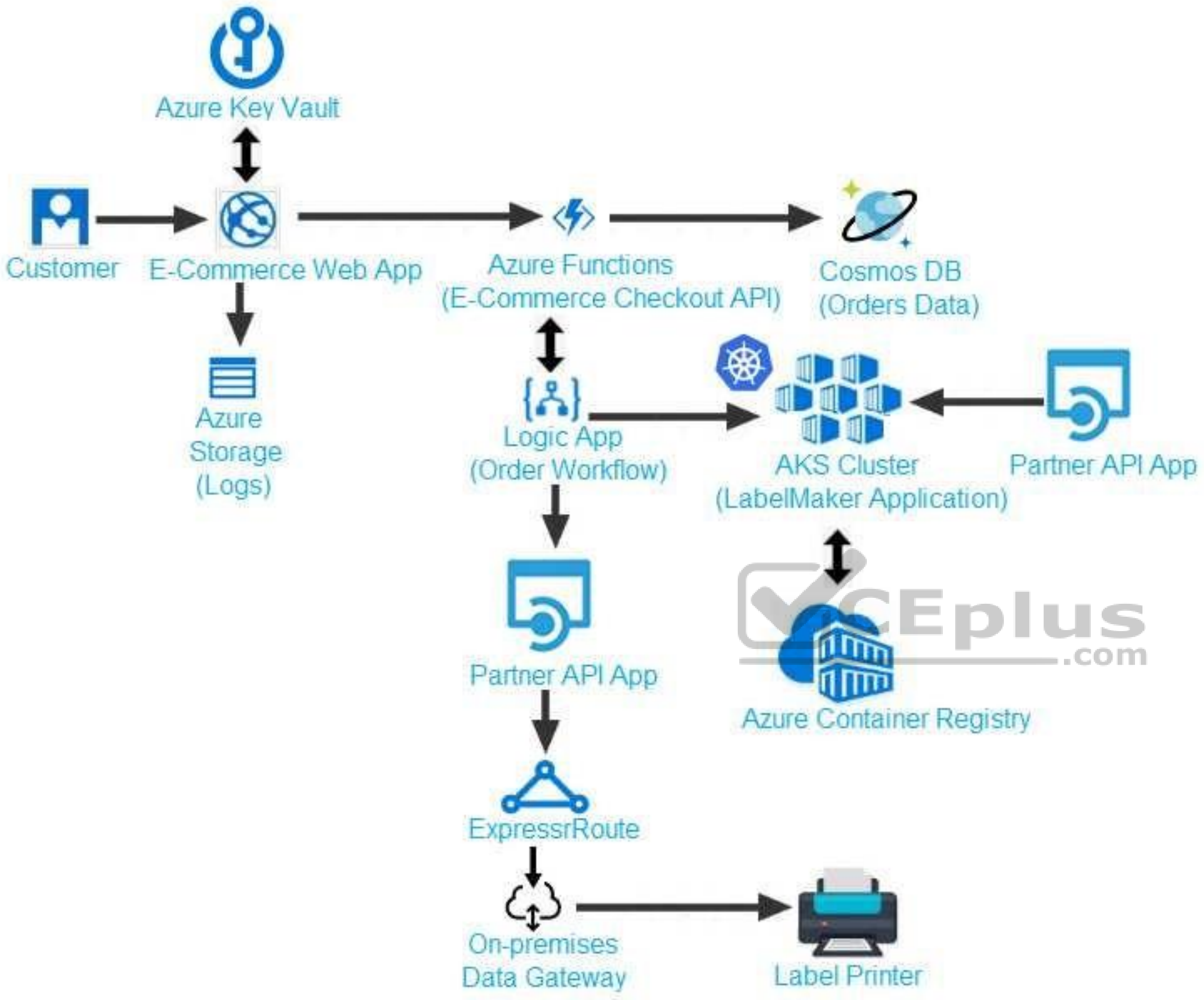
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization’s identity management solution.
- External partner logins must be audited monthly 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.

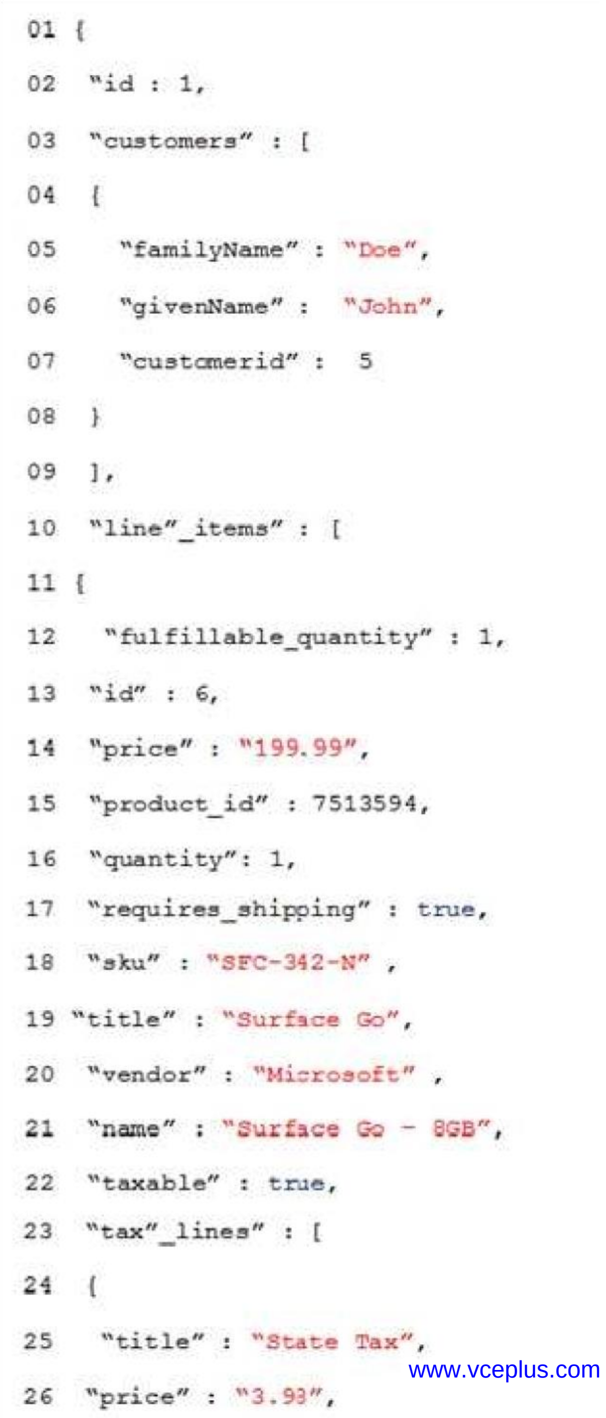


Calls to the Printer API App fail periodically due to printer communication timeouts.

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**



```
{
 “id : 1,
 “customers” : [
  {
   “familyName” : “Doe”,
   “givenName” :  “John”,
   “customerid” :  5
  }
 ],
 “line”_items” : [
{
  “fulfillable_quantity” : 1,
 “id” : 6,
 “price” : “199.99”,
 “product_id” : 7513594,
 “quantity”: 1,
 “requires_shipping” : true,
 “sku” : “SFC-342-N” ,
“title” : “Surface Go”,
 “vendor” : “Microsoft” ,
 “name” : “Surface Go - 8GB”,
 “taxable” : true,
 “tax”_lines” : [
  {
  “title” : “State Tax”,
 “price” : “3.98”,
```

**QUESTION 1**

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

**Correct Answer:** AD
**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

**Question Set 1**

**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 that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.**

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

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

You create the index in Azure Search.

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

Solution:

1. Create a SearchServiceClient object to connect to the search index.
2. Create a DataContainer that contains the documents which must be added.
3. Create a DataSource instance and set its Container property to the DataContainer.
4. Set the DataSources property of the SearchServiceClient.

Does the solution meet the goal?

A. Yes
B. No

**Correct Answer:** B
**Section: [none]**
**Explanation**

**Explanation/Reference:**
Explanation:
Use the following method:
1. Create a SearchIndexClient object to connect to the search index
2. Create an IndexBatch that contains the documents which must be added.
3. Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.

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

**QUESTION 2**
A company is implementing a publish-subscribe (Pub/Sub) messaging component by using Azure Service Bus. You are developing the first subscription application.

In the Azure portal you see that messages are being sent to the subscription for each topic. You create and initialize a subscription client object by supplying the correct details, but the subscription application is still not consuming the messages.

You need to complete the source code of the subscription client

What should you do?

A. `await subscriptionClient.CloseAsync();`

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

C. `subscriptionClient.RegisterMessageHandler(ProcessMessagesAsync, messageHandlerOptions);`

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

**Correct Answer:** C
**Section: [none]**
**Explanation**

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

subscriptionClient.RegisterMessageHandler(ReceiveMessagesAsync, messageHandlerOptions);

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