

AZ-203.VCEplus.premium.exam.46q

Number: AZ-203
Passing Score: 800
Time Limit: 120 min
File Version: 1.0



Website: <https://vceplus.com>

VCE to PDF Converter: <https://vceplus.com/vce-to-pdf/>

Facebook: <https://www.facebook.com/VCE.For.All.VN/>

Twitter : https://twitter.com/VCE_Plus



AZ-203

Developing Solutions for Microsoft Azure

Version 1.0



Question Set 1

QUESTION 1

HOTSPOT

You have an Azure Batch project that processes and converts files and stores the files in Azure storage. You are developing a function to start the batch job.

You add the following parameters to the function.

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

You must ensure that converted files are placed in the container referenced by the outputContainerSasUrl parameter. Files which fail to convert are placed in the container referenced by the failedContainerSasUrl parameter.

You need to ensure the files are correctly processed.

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 List<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
    string outputContainerSasUrl, string failedContainerSasUrl)
{
    BatchSharedKeyCredentials sharedKeyCredentials =
        new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
    List<CloudTask> tasks = new List<CloudTask>();
    using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
    {
        CloudJob = batchClient.JobOperations. 

|           |
|-----------|
| ▼         |
| GetJob    |
| GetTask   |
| EnableJob |
| CreateJob |

 ();

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

|                |
|----------------|
| ▼              |
| TaskSuccess    |
| TaskFailure    |
| TaskCompletion |

 ))) );

            outputFileList.Add(new OutputFile (fileTask.Output,
                new OutputFileDestination (failedContainer),
                new OutputFileUploadOptions (OutputFileUploadCondition, 

|                |
|----------------|
| ▼              |
| TaskSuccess    |
| TaskFailure    |
| TaskCompletion |

 ))) );

            task. 

|               |
|---------------|
| ▼             |
| OutputFiles   |
| FilesToStage  |
| ResourceFiles |
| StageFiles    |

 =outputFileList;

            task.Add(task);
        });
    }
    return tasks,
}

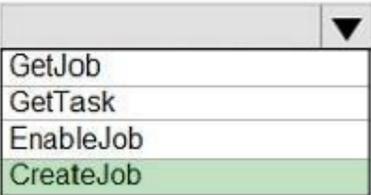
```

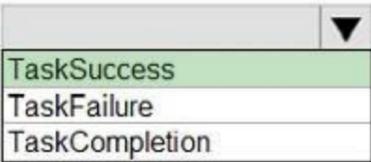
Correct Answer:



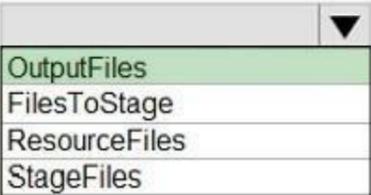
Answer Area

```

public List<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
    string outputContainerSasUrl, string failedContainerSasUrl)
{
    BatchSharedKeyCredentials sharedKeyCredentials =
        new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
    List<CloudTask> tasks = new List<CloudTask>();
    using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
    {
        CloudJob = batchClient.JobOperations.  ();

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

            outputFileList.Add(new OutputFile (fileTask.Output,
                new OutputFileDestination(failedContainer),
                new OutputFileUploadOptions (OutputFileUploadCondition,  ));

            task  =outputFileList;

            task.Add(task);
        });
    }
    return tasks,
}

```

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Box 1: CreateJob

Box 2: TaskSuccess

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

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

Box 3: TaskFailure

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

Box 4: OutputFiles

To specify output files for a task, create a collection of OutputFile objects and assign it to the CloudTask.OutputFiles property when you create the task.

References:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.batch.protocol.models.outputfileuploadcondition> <https://docs.microsoft.com/en-us/azure/batch/batch-task-output-files>

QUESTION 2 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>, Cancellation.Token)`
- B. `JobOperations.CreateJob()`
- C. `CloudJob.Enable(IEnumerable<BatchClientBehavior>)`
- D. `JobOperations.EnableJob(String, IEnumerable<BatchClientBehavior>)`
- E. `CloudJob.CommitAsync(IEnumerable<BatchClientBehavior>, Cancellation.Token)`

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

DRAG DROP



Fourth Coffee has an ASP.NET Core web app that runs in Docker. The app is mapped to the www.fourthcoffee.com domain.

Fourth Coffee is migrating this application to Azure.

You need to provision an App Service Web App to host this docker image and map the custom domain to the App Service web app.

A resource group named FourthCoffeePublicWebResourceGroup has been created in the WestUS region that contains an App Service Plan named AppServiceLinuxDockerPlan.

Which order should the CLI commands be used to develop the solution? To answer, move all of the Azure CLI command 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: #bin/bash

The appName is used when the webapp-name is created in step 2.

Step 2: az webapp config hostname add

The webapp-name is used when the webapp is created in step 3.

Step 3: az webapp create

Create a web app. In the Cloud Shell, create a web app in the myAppServicePlan App Service plan with the az webapp create command.

Step : az webapp config container set

In Create a web app, you specified an image on Docker Hub in the az webapp create command. This is good enough for a public image. To use a private image, you need to configure your Docker account ID and password in your Azure web app.

In the Cloud Shell, follow the az webapp create command with az webapp config container set.

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

QUESTION 3

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 runtime-specific 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 4 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.

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().CreateIfNoExistsAsync();
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:

Correct Answer:

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>

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.

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.

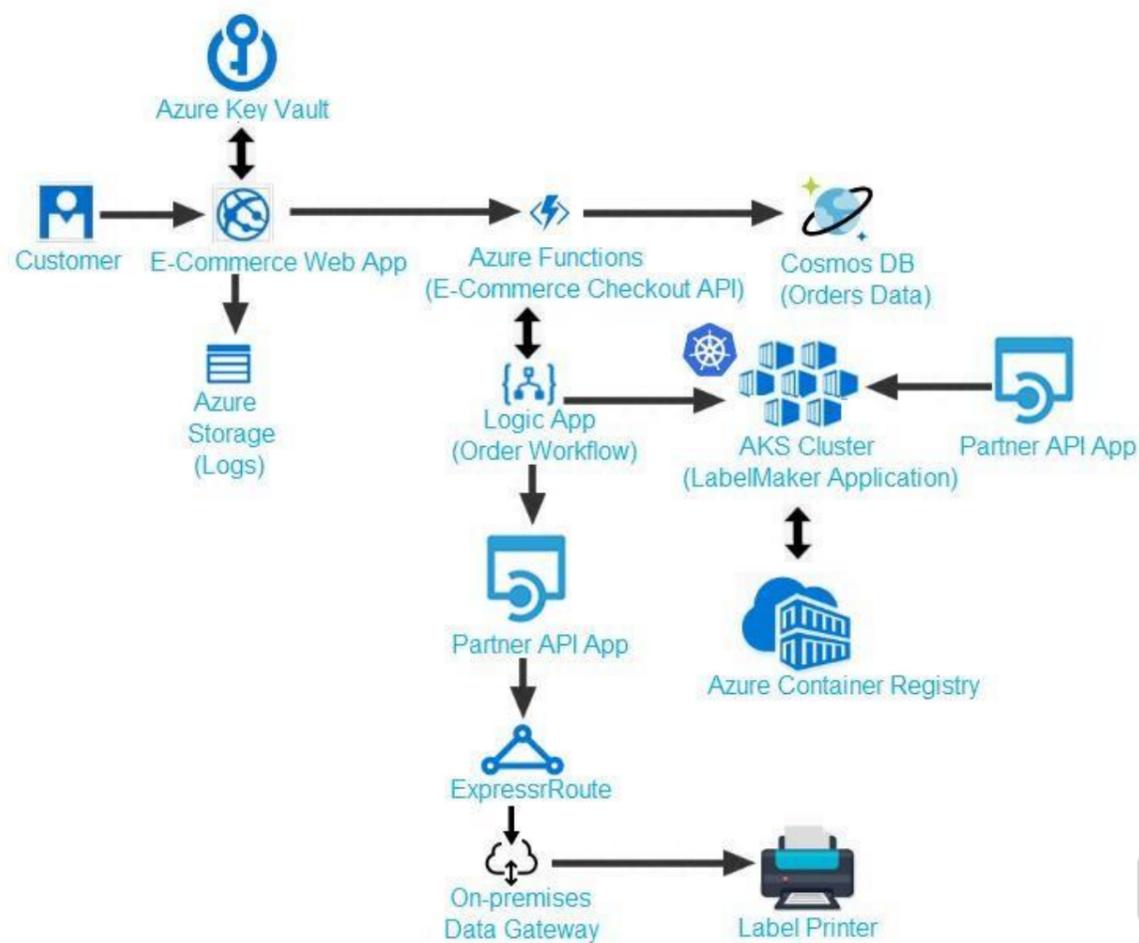
You have the following security requirements:

- 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" : "3.93",
27           "rate" : 0.06
28         }
29       ],
30       "total_discount" : "5.00"
31       "discount_allocations" : [
32         {
33           "amount" : "5.00",
34           "discount_application_index" : 2
35         }
36       ]
37     }
38   ],
39   "address" : {
40     "state" : "NY",
41     "country" : "Manhattan",
```



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

HOTSPOT

You have an app that stores player scores for an online game. The app stores data in Azure tables using a class named PlayerScore as the table entity. The table is populated with 100,000 records.

You are reviewing the following section of code that is intended to retrieve 20 records where the player score exceeds 15,000. (Line numbers are included for reference only.)

```

1 public void GetScore(string playerId, int score, string gameName)
2 {
3     Table Query<DynamicTableEntity> query = new TableQuery<DynamicTableEntity>().Select(new string[] { "Score" })
        .Where(TableQuery.GenerateFilterConditionForInt("Score", QueryComparisons.GreaterThanOrEqualTo, 15000)).Take
(20);
4     EntityResolver<KeyValuePair<string, int?>> resolver =
        (partitionKey, rowKey, ts, props, etag) => new KeyValuePair<string, int?>(rowKey, props["Score"].Int32Value);
5     foreach (var scoreItem in scoreTable.ExecuteQuery(query, resolver, null, null))
6     {
7         Console.WriteLine($"{scoreItem.Key} {scoreItem.Value}");
8     }
9
9 public class PlayerScore : TableEntity
10 {
11     public PlayerScore(string gameId, string playerId, int score, long timePlayed)
12     {
13         PartitionKey = gameId;
14         RowKey = playerId;
15         Score = score;
16         TimePlayed = timePlayed;
17     }
18     public int Score { get; set; }
19     public long TimePlayed { get; set; }
20 }

```

You have the following code. (Line numbers are included for reference only.)

```

01 public void SaveScore(string gameId, string playerId, int score, long timePlayed)
02 {
03     CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connectionString);
04     CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
05     CloudTable table = tableClient.GetTableReference("scoreTable");
06     table.CreateIfNotExists();
07     var scoreRecord = new PlayerScore(gameId, playerId, score, timePlayed);
08     TableOperation insertOperation = TableOperation.Insert(scoreRecord);
09     table.Execute(insertOperation);
10 }
11 public class PlayerScore : TableEntity
12 {
13     public PlayerScore(string gameId, string playerId, int score, long timePlayed)
14     {
15         this.PartitionKey = gameId;
16         this.RowKey = playerId;
17         Score = score;
18         TimePlayed = timePlayed;
19     }
20     public int Score { get; set; }
21     public long TimePlayed { get; set; }
22 }

```

You store customer information in an Azure Cosmos database. The following data already exists in the database:

PartitionKey	RowKey	Email
Harp	Walter	wharp@contoso.com
Smith	Steve	ssmith@contoso.com
Smith	Jeff	jsmith@contoso.com

You develop the following code. (Line numbers are included for reference only.)

```
01 CloudTableClient tableClient = account.CreateCloudTableClient();
02 CloudTable table = tableClient.GetTableReference("people");
03 TableQuery<CustomerEntity> query = new TableQuery<CustomerEntity>()
04     .Where(TableQuery.CombineFilters(
05         TableQuery.Generate.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal, "Smith")
06         TableOperators.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal,
07         "ssmith@contoso.com")
08     ));
08 await table.ExecuteQuerySegmentedAsync<CustomerEntity>(query, null);
```

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: No

Box 2: Yes

The TableQuery.Take method defines the upper bound for the number of entities the query returns.

Example:

query.Take(10);

Box 3: Yes

Box 4: Yes

References: <https://www.vkinfotek.com/azureqa/how-do-i-query-azure-table-storage-using-tablequery-class.html>



QUESTION 2

HOTSPOT

A company develops a series of mobile games. All games use a single leaderboard service.

You have the following requirements:

- Code should be scalable and allow for growth.
 - Each record must consist of a playedId, gameId, score, and time played.
 - When users reach a new high score, the system will save the new score using the SaveScore function below. ▪
- Each game is assigned an Id based on the series title.

You have the following code. (Line numbers are included for reference only.)

```

01 public void SaveScore(string gameId, string playerId, int score, long timePlayed)
02 {
03     CloudStorageAccount storageAccount = CloudStorageAccount.Parse(connectionString);
04     CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
05     CloudTable table = tableClient.GetTableReference("scoreTable");
06     table.CreateIfNotExists();
07     var scoreRecord = new PlayerScore(gameId, playerId, score, timePlayed);
08     TableOperation insertOperation = TableOperation.Insert(scoreRecord);
09     table.Execute(insertOperation);
10 }
11 public class PlayerScore : TableEntity
12 {
13     public PlayerScore(string gameId, string playerId, int score, long timePlayed)
14     {
15         this.PartitionKey = gameId;
16         this.RowKey = playerId;
17         Score = score;
18         TimePlayed = timePlayed;
19     }
20     public int Score { get; set; }
21     public long TimePlayed { get; set; }
22 }

```

You store customer information in an Azure Cosmos database. The following data already exists in the database:

PartitionKey	RowKey	Email
Harp	Walter	wharp@contoso.com
Smith	Steve	ssmith@contoso.com
Smith	Jeff	jsmith@contoso.com



You develop the following code. (Line numbers are included for reference only.)

```

01 CloudTableClient tableClient = account.CreateCloudTableClient();
02 CloudTable table = tableClient.GetTableReference("people");
03 TableQuery<CustomerEntity> query = new TableQuery<CustomerEntity>()
04     .Where(TableQuery.CombineFilters(
05         TableQuery.Generate.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal, "Smith")
06         TableOperstors.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal,
07         "ssmith@contoso.com")
08         ));
08 await table.ExecuteQuerySegmentedAsync<CustomerEntity>(query, null);

```

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

Code for CosmosDB, example:

```
// Parse the connection string and return a reference to the storage account.  
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(  
CloudConfigurationManager.GetSetting("StorageConnectionString")); //  
Create the table client.  
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();  
// Retrieve a reference to the table.  
CloudTable table = tableClient.GetTableReference("people");  
// Create the TableOperation object that inserts the customer entity.  
TableOperation insertOperation = TableOperation.Insert(customer1);
```

Box 2: No
A new record will always be added as TableOperation.Insert is used, instead of TableOperation.InsertOrReplace.

Box 3: No
No partition key is used.

Box 4: Yes

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

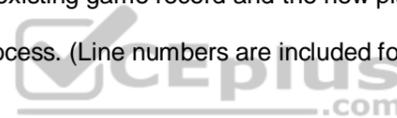
QUESTION 3 HOTSPOT

You are working for a company that designs mobile applications. They maintain a server where player records are assigned to their different games. The tracking system is new and in development.

The application uses Entity Framework to connect to an Azure Database. The database holds a Player table and Game table.

When adding a player, the code should insert a new player record, and add a relationship between an existing game record and the new player record.

The application will call CreatePlayerWithGame with the correct gameId and the playerId to start the process. (Line numbers are included for reference only.)



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

```



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:

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

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<PostTag>()
        .HasKey(t => new { t.PostId, t.TagId });

    modelBuilder.Entity<PostTag>()
        .HasOne(pt => pt.Post)
            .WithMany(p => p.PostTags)
            .HasForeignKey(pt => pt.PostId);

    modelBuilder.Entity<PostTag>()
        .HasOne(pt => pt.Tag)
            .WithMany(t => t.PostTags)
            .HasForeignKey(pt => pt.TagId);
}
```

QUESTION 4

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().CreateIfNoExistsAsync();
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.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

DRAG DROP

You need to ensure disaster recovery requirements are met.

What code should you add at line PC16?

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

-
-
-
-
-
-

Answer Area

```

var copyOptions = new CopyOptions {};
var context= new  (source,destination)=>Task.FromResult(true);
context.  (source, destination) => Task.FromResult(true);

await TransferManager.CopyAsync(blob, GetDRBlob(blob), isServiceCopy: 
, context:context, options: copyOptions); copyOptions, context);

```

Correct Answer:

Values

-
-
-
-
-
-

Answer Area

```
var copyOptions = new CopyOptions {};
var context= new  (source,destination)=>Task.FromResult(true);
context.  (source, destination) => Task.FromResult(true);
await TransferManager.CopyAsync(blob, GetDRBlob(blob), isServiceCopy: 
, context:context, options: copyOptions); copyOptions, context);
```

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Scenario, 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.

Box 1: DirectoryTransferContext
We transfer all files in the directory.

Note: The TransferContext object comes in two forms: SingleTransferContext and DirectoryTransferContext. The former is for transferring a single file and the latter is for transferring a directory of files.

Box 2: ShouldTransferCallbackAsync
The DirectoryTransferContext.ShouldTransferCallbackAsync delegate callback is invoked to tell whether a transfer should be done.

Box 3: False
If you want to use the retry policy in Copy, and want the copy can be resume if break in the middle, you can use SyncCopy (isServiceCopy = false).

Note that if you choose to use service side copy ('isServiceCopy' set to true), Azure (currently) doesn't provide SLA for that. Setting 'isServiceCopy' to false will download the source blob loca

References: <https://docs.microsoft.com/en-us/azure/storage/common/storage-use-data-movement-library> <https://docs.microsoft.com/en-us/dotnet/api/microsoft.windowsazure.storage.datamovement.directorytransfercontext.shouldtransfercallbackasync?view=azure-dotnet>

QUESTION 2

DRAG DROP

You need to add code at line PC32 in Processing.cs to implement the GetCredentials method in the Processing class.

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	Answer Area
MSITokenProvider("...", null)	var tp=new
tp.GetAccessTokenAsync("...")	var t=new TokenCredential(await
AzureServiceTokenProvider()	
StringTokenProvider("storage", "msi")	return new StorageCredentials(t);
tp.GetAuthenticationHeaderAsync(CancellationTokens.None)	

Correct Answer:

Code segments	Answer Area
MSITokenProvider("...", null)	var tp=new AzureServiceTokenProvider()
tp.GetAccessTokenAsync("...")	var t=new TokenCredential(await tp.GetAccessTokenAsync("..."))
AzureServiceTokenProvider()	
StringTokenProvider("storage", "msi")	return new StorageCredentials(t);
tp.GetAuthenticationHeaderAsync(CancellationTokens.None)	



Section: [none]

Explanation

Explanation/Reference:

Explanation:

Acquiring an access token is then quite easy. Example code:

```
private async Task<string> GetAccessTokenAsync()
{
    var tokenProvider = new AzureServiceTokenProvider();
    return await tokenProvider.GetAccessTokenAsync("https://storage.azure.com/");
}
```

References: <https://joonasw.net/view/azure-ad-authentication-with-azure-storage-and-managed-service-identity>

QUESTION 3 You need to construct the link to the summary report for the email that is sent to users.

What should you do?

- Create a SharedAccessBlobPolicy and add it to the containers SharedAccessPolicies.Call GetSharedAccessSignature on the blob and use the resulting link.
- 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.

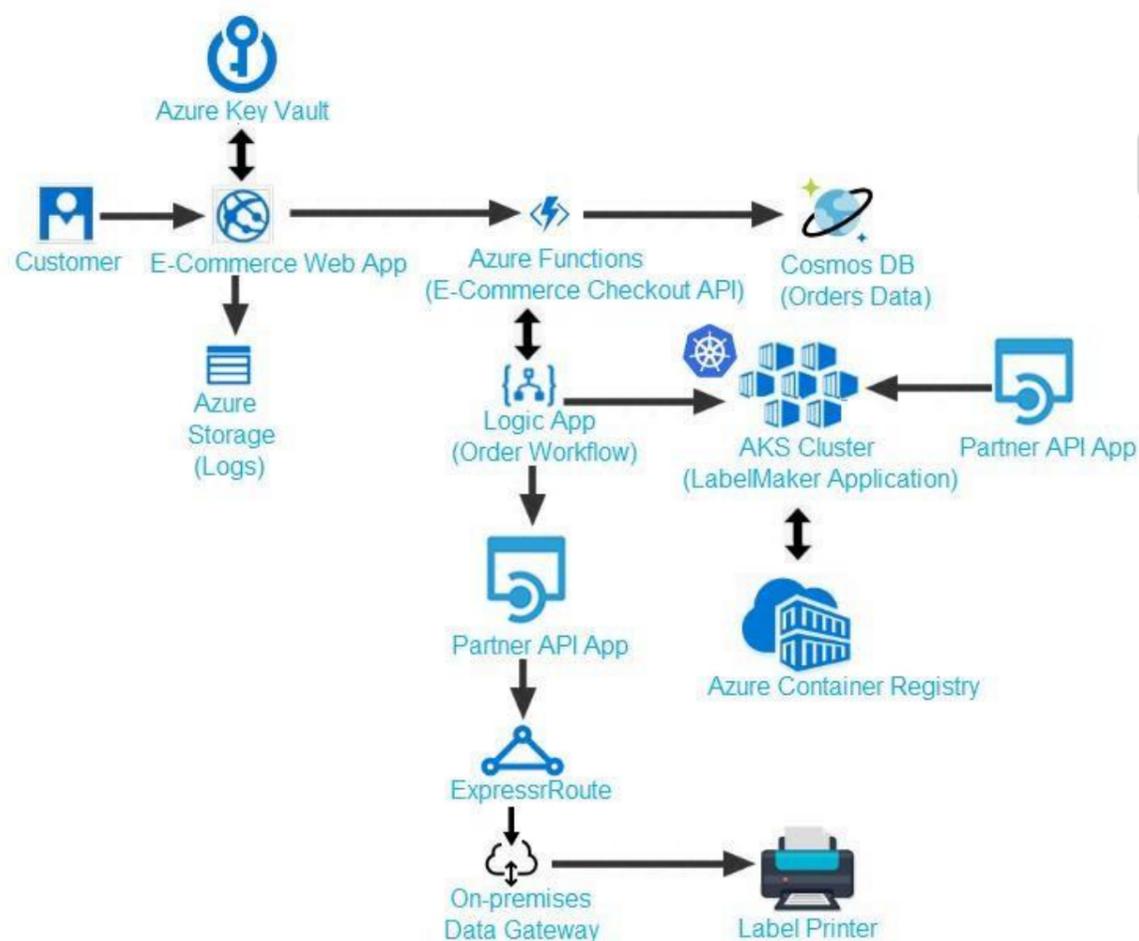
You have the following security requirements:

- 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



```
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" : "3.93",
27           "rate" : 0.06
28         }
29       ],
30       "total_discount" : "5.00"
31       "discount_allocations" : [
32         {
33           "amount" : "5.00",
34           "discount_application_index" : 2
35         }
36       ]
37     }
38   ]
39 }
```



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/>

QUESTION 2

HOTSPOT

You need to meet the security requirements for external partners.

Which Azure Active Directory features should you use?

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: B2B

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

Azure Active Directory (Azure AD) business-to-business (B2B) collaboration lets you securely share your company's applications and services with guest users from any other organization, while maintaining control over your own corporate data. Work safely and securely with external partners, large or small, even if they don't have Azure AD or an IT department. A simple invitation and redemption process lets partners use their own credentials to access your company's resources. Developers can use Azure AD business-to-business APIs to customize the invitation process or write applications like self-service sign-up portals.

Box 2: Access Review

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

Azure Active Directory (Azure AD) Access Reviews enable organizations to efficiently manage group memberships, access to enterprise applications, and role assignments.

Administrators can use Azure Active Directory (Azure AD) to create an access review for group members or users assigned to an application. Azure AD automatically sends reviewers an email that prompts them to review access.

References:

<https://docs.microsoft.com/en-us/azure/active-directory/b2b/what-is-b2b>

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().CreateIfNoExistsAsync();
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

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-AzureRmKeyVaultAccessPolicy 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-AzureRmKeyVaultAccessPolicy 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/azurermskeyvault/set-azurermskeyvaultaccesspolicy>



Question Set 3

QUESTION 1

HOTSPOT

Your company is migrating applications to Azure. The IT department must allow internal developers to communicate with Microsoft support.

The service agents of the IT department must only have view resources and create support ticket permissions to all subscriptions. A new custom role must be created by reusing a default role definition and changing the permissions.

You need to create the custom role.

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: Set-AzureRmRoleDefinition Input-File C:\SupportRole.json

The Set-AzureRmRoleDefinition cmdlet updates an existing custom role in Azure Role-Based Access Control. Provide the updated role definition as an input to the command as a JSON file or a PSRoleDefinition object. The role definition for the updated custom role MUST contain the Id and all other required properties of the role even if they are not updated: DisplayName, Description, Actions, AssignableScope

Box 2: "*/read*.* Microsoft.Support/*"

Microsoft.Support/* Create and manage support tickets

"Microsoft.Support" role definition azure



Incorrect Answers:

Get-AzureRmRoleDefinition. The Get-AzureRmRoleDefinition command does not have an action section.

First, use the Get-AzureRmRoleDefinition command to retrieve the custom role that you wish to modify. Then, modify the properties that you wish to change. Finally, save the role definition using the Set-AzureRmRoleDefinition command.

References: <https://docs.microsoft.com/en-us/azure/role-based-access-control/custom-roles-powershell>

QUESTION 2

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 3

DRAG DROP

Contoso, Ltd. provides an API to customers by using Azure API Management (APIM). The API authorizes users with a JWT token.

You must implement response caching for the APIM gateway. The caching mechanism must detect the user ID of the client that accesses data for a given location and cache the response for that user ID.

You need to add the following policies to the policies file:

- a set-variable policy to store the detected user identity
- a cache-lookup-value policy
- a cache-store-value policy
- a find-and-replace policy to update the response body with the user profile information

To which policy section should you add the policies? To answer, drag the appropriate sections to the correct policies. Each section 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: Inbound.

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

```
<policies>
  <inbound>
    <!-- How you determine user identity is application dependent -->
    <set-variable
name="enduserid"
value="@context.Request.Headers.GetValueOrDefault("Authorization","").Split(' ')[1].AsJwt()?.Subject" /> Etc.
```

Box 2: Inbound

A cache-lookup-value policy

Example:

```
<inbound>
  <base />
  <cache-lookup vary-by-developer="true | false" vary-by-developer-groups="true | false" downstream-caching-type="none | private | public" must-revalidate="true | false">
    <vary-by-query-parameter>parameter name</vary-by-query-parameter> <!-- optional, can repeated several times --> </cache-lookup>
</inbound>
```

Box 3: Outbound A cache-store-value policy.

```
Example:  
<outbound>  
<base />  
  <cache-store duration="3600" />  
</outbound>
```

Box 4: Outbound

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

```
<outbound>  
  <!-- Update response body with user profile-->  
  <find-and-replace  
from="$userprofile$"  
  to="@((string)context.Variables["userprofile"])" />  
  <base />  
</outbound>
```

References: <https://docs.microsoft.com/en-us/azure/api-management/api-management-caching-policies> <https://docs.microsoft.com/en-us/azure/api-management/api-management-sample-cache-by-key>

Question Set 1

QUESTION 1

HOTSPOT

A company is developing a gaming platform. Users can join teams to play online and see leaderboards that include player statistics. The solution includes an entity named Team.

You plan to implement an Azure Redis Cache instance to improve the efficiency of data operations for entities that rarely change.

You need to invalidate the cache when team data is changed.

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:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Box 1: IDatabase cache = connection.GetDatabase();
Connection refers to a previously configured ConnectionMultiplexer.

Box 2: cache.SetString("teams", "
To specify the expiration of an item in the cache, use the TimeSpan parameter of StringSet. cache.SetString("key1",
"value1", TimeSpan.FromMinutes(90));

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



QUESTION 2

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 3

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 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.

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 = JObject.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();
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 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>

QUESTION 2

DRAG DROP

You need to implement telemetry for non-user actions.

How should you complete the Filter class? 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:

Correct Answer:

Section: [none]

Explanation



Explanation/Reference:

Explanation:

Scenario: Exclude non-user actions from Application Insights telemetry.

Box 1: ITelemetryProcessor

To create a filter, implement ITelemetryProcessor. This technique gives you more direct control over what is included or excluded from the telemetry stream.

Box 2: ITelemetryProcessor

Box 3: ITelemetryProcessor

Box 4: RequestTelemetry

Box 5: /health

To filter out an item, just terminate the chain.

References: <https://docs.microsoft.com/en-us/azure/azure-monitor/app/api-filtering-sampling>

QUESTION 3

DRAG DROP

You need to ensure that PolicyLib requirements are met.

How should you complete the code segment? 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

- Process
- Initialize
- telemetry.Sequence
- ITelemetryProcessor
- ITelemetryInitializer
- Telemetry.Context
- EventGridController.EventId.Value
- [(EventTelemetry) telemetry.Properties["EventId"]

Answer Area

```

public class IncludeEventId:
{
    public void
    (ITelemetry telemetry)
    {
        .Properties["EventId"] =
    }
}
    
```

Correct Answer:

Code segments

Process

Initialize

telemetry.Sequence

ITelemetryProcessor

ITelemetryInitializer

Telemetry.Context

EventGridController.EventId.Value

[(EventTelemetry) telemetry.Properties["EventId"]]

Answer Area

```
public class IncludeEventId: ITelemetryInitializer
{
    public void Initialize
    (ITelemetry telemetry)
    {
        Telemetry.Context.Properties["EventId"] =
        [(EventTelemetry) telemetry.Properties["EventId"]]
    }
}
```

Section: [none]

Explanation

Explanation/Reference:

Explanation:

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:

- 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.

Box 1: ITelemetryInitializer

Use telemetry initializers to define global properties that are sent with all telemetry; and to override selected behavior of the standard telemetry modules.

Box 2: Initialize

Box 3: Telemetry.Context

Box 4: [(EventTelemetry)telemetry.Properties("EventID")]

References:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/api-filtering-sampling>

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().CreateIfNoExistsAsync();
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 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:

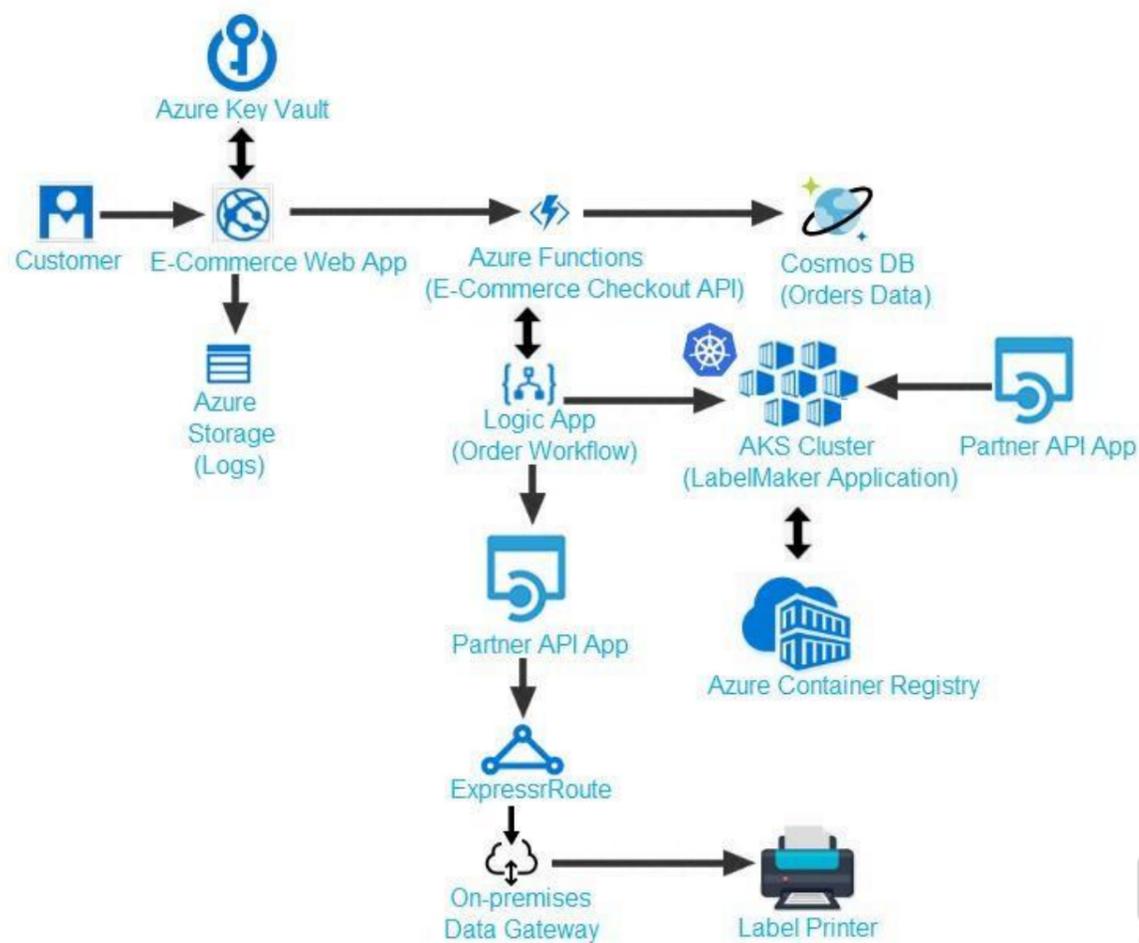
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly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

```
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" : "3.93",
27           "rate" : 0.06
28         }
29       ],
30       "total_discount" : "5.00"
31       "discount_allocations" : [
32         {
33           "amount" : "5.00",
34           "discount_application_index" : 2
35         }
36       ]
37     }
38   ],
```



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

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. Call the Documents.Suggest method of the SearchIndexClient and pass the DataSource.

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 3

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

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

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

You create the index in Azure Search.

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

Solution:

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



Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

Section: [none]

Explanation

Explanation/Reference:

Explanation:

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

2. Create the indexBatch with the documentsSomething like:

```
var hotels = new Hotel[];  
{  
    new Hotel()  
    {  
        HotelId = "3",  
        BaseRate = 129.99,  
        Description = "Close to town hall and the river"  
    }  
};  
...
```

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

3. The next step is to populate the newly-created index

Example:

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

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

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

QUESTION 4

DRAG DROP

You develop software solutions for a mobile delivery service. You are developing a mobile app that users can use to order from a restaurant in their area. The app uses the following workflow:

1. A driver selects the restaurants for which they will deliver orders.
2. Orders are sent to all available drivers in an area.
3. Only orders for the selected restaurants will appear for the driver.
4. The first driver to accept an order removes it from the list of available orders.

You need to implement an Azure Service Bus solution.

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.

Select and Place:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Box 1: Create a single Service Bus Namespace

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

Box 2: Create a Service Bus Topic for each restaurant for which a driver can receive messages. Create topics.

Box 3: Create a Service Bus subscription for each restaurant for which a driver can receive orders. Topics can have multiple, independent subscriptions.

References: <https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview>

QUESTION 5

HOTSPOT

A company runs an international travel and bookings management service. The company plans to begin offering restaurant bookings.

You must develop a solution that uses Azure Search and meets the following requirements:

- Users must be able to search for restaurants by name, description, location, and cuisine.
- Users must be able to narrow the results further by location, cuisine, rating, and family-friendliness. ▪

All words in descriptions must be included in searches.

You need to add annotations to the restaurant class.

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:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

Explanation:

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

Location

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

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

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

Description

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

All words in descriptions must be included in searches.

Box 3: [IsFilterable,IsSortable,IsFaceTable]

Rating

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

Box 4: [IsSearchable.IsFilterable,IsFacetable]

Cuisines

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

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

Box 5: [IsFilterable,IsFacetable] FamilyFriendly

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

References: <https://www.henkboelman.com/azure-search-the-basics/>

QUESTION 6

DRAG DROP

You develop a gateway solution for a public facing news API.

The news API back end is implemented as a RESTful service and hosted in an Azure App Service instance.

You need to configure back-end authentication for the API Management service instance.

Which target and gateway credential type should you use? To answer, drag the appropriate values to the correct parameters. Each value 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: Azure Resource

Box 2: Client cert

API Management allows to secure access to the back-end service of an API using client certificates.

References: <https://docs.microsoft.com/en-us/rest/api/apimanagement/apimanagementrest/azure-api-management-rest-api-backend-entity>

QUESTION 7

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/>

QUESTION 8

DRAG DROP

You have an application that provides weather forecasting data to external partners. You use Azure API Management to publish APIs.

You must change the behavior of the API to meet the following requirements:

- Support alternative input parameters.
- Remove formatting text from responses.
- Provide additional context to back-end services.

Which types of policies should you implement? To answer, drag the policy types to the correct scenarios. Each policy 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:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

QUESTION 9

HOTSPOT

You are developing a .NET Core MVC application for customers to research hotels. The application will use Azure Search. The application will search the index by using various criteria to locate documents related to hotels. The index will include search fields for rate, a list of amenities, and distance to the nearest airport.

The application must support the following scenarios for specifying search criteria and organizing results:

- Search the index by using regular expressions.
- Organize results by counts for name-value pairs.
- List hotels within a specified distance to an airport and that fall within a specific price range.

You need to configure the SearchParameters class.

Which properties should you configure? 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: QueryType

The SearchParameters.QueryType Property gets or sets a value that specifies the syntax of the search query. The default is 'simple'. Use 'full' if your query uses the Lucene query syntax.

You can write queries against Azure Search based on the rich Lucene Query Parser syntax for specialized query forms: wildcard, fuzzy search, proximity search, regular expressions are a few examples.

Box 2: Facets

The facets property gets or sets the list of facet expressions to apply to the search query. Each facet expression contains a field name, optionally followed by a comma-separated list of name:value pairs.

Box 3: Filter

The Filter property gets or sets the OData \$filter expression to apply to the search query.

References:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.search.models.searchparameters> <https://docs.microsoft.com/en-us/azure/search/query-lucene-syntax>

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.search.models.searchparameters.querytype>

QUESTION 10

HOTSPOT

You develop a news and blog content delivery app for Windows devices.

A notification must arrive on a user's device when there is a new article available for them to view.

You need to implement push notifications.

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:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Box 1: NotificationHubClient

Box 2: NotificationHubClient

Box 3: CreateClientFromConnectionString

// Initialize the Notification Hub

```
NotificationHubClient hub = NotificationHubClient.CreateClientFromConnectionString(listenConnString, hubName);
```

Box 4: SendWindowsNativeNotificationAsync Send the push notification. var result =
await hub.SendWindowsNativeNotificationAsync(windowsToastPayload);

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

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



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 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.

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 = JObject.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();
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 {
```



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 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 section, 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 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:

Instead ensure that signout events have a subject prefix. Create an Azure Event Grid subscription that uses the subjectBeginsWith filter.

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 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 section, 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 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: B

Section: [none]

Explanation

Explanation/Reference:

Explanation:

Instead ensure that signout events have a subject prefix. Create an Azure Event Grid subscription that uses the subjectBeginsWith filter.

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 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 section, 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 authentication events are triggered and processed according to the policy.

Solution: Ensure that signout events have a subject prefix. Create an Azure Event Grid subscription that uses the subjectBeginsWith filter.

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.

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

QUESTION 4

DRAG DROP

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:

Correct Answer:

Section: [none]

Explanation

Explanation/Reference:

Explanation:

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

Box 1: Status

Box 2: Succeeded

Box 3: operationName

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

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

