

Lead2Pass.com

First Test, First Pass!

<http://www.lead2pass.com>

Copyright © 1999-2021 Lead2pass.com, All Rights Reserved.





Vendor: Microsoft

Exam Code: AZ-204

Exam Name: Developing Solutions for Microsoft Azure

Version: 21.022

Important Notice

Product

Our Product Manager keeps an eye for Exam updates by Vendors. Free update is available within 150 days after your purchase.

You can login member center and download the latest product anytime. (Product downloaded from member center is always the latest.)

PS: Ensure you can pass the exam, please check the latest product in 2-3 days before the exam again.

Feedback

We devote to promote the product quality and the grade of service to ensure customers interest.

If you have any suggestions, please feel free to contact us at support@lead2pass.com

If you have any questions about our product, please provide Exam Number, Version, Page Number, Question Number, and your Login Account to us, please contact us at technology@lead2pass.com and our technical experts will provide support in 24 hours.

Copyright

The product of each order has its own encryption code, so you should use it independently.

If anyone who share the file we will disable the free update and account access.

Any unauthorized changes will be inflicted legal punishment. We will reserve the right of final explanation for this statement.

Order ID: ****

PayPal Name: ****

PayPal ID: ****

Exam A(112 Questions) + Exam B(225 Questions)**Exam A****QUESTION 1****Case Study 1****Current environment****Windows Server 2016 virtual machine**

The virtual machine (VM) runs BizTalk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and tracking stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

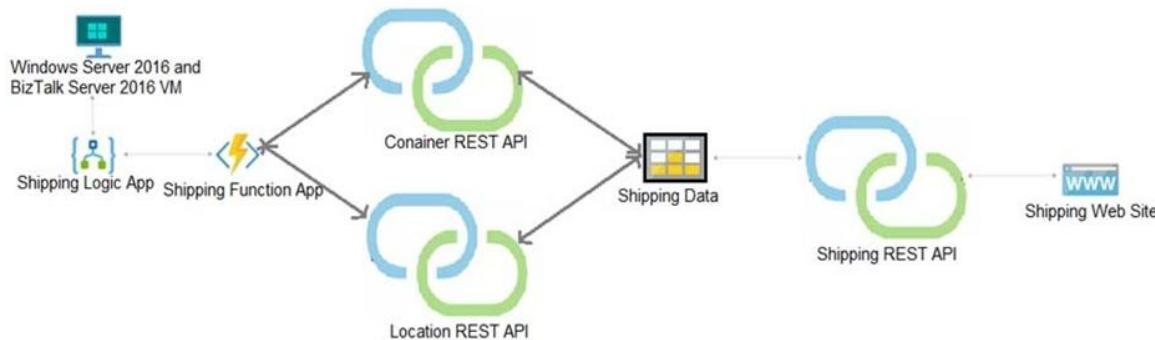
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com/>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Requirements****Shipping Logic app**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

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

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues**Windows Server 2016 VM**

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No
'Access-Control-Allow-Origin' header is present on the requested
resource. Origin 'http://test.wideworldimporters.com/' is therefore not
allowed access.
```

You need to secure the Shipping Logic App.

What should you use?

- Azure App Service Environment (ASE)
- Integration Service Environment (ISE)
- VNet service endpoint
- Azure AD B2B integration

Answer: B

Explanation:

Scenario: The Shipping Logic App requires secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.

You can access to Azure Virtual Network resources from Azure Logic Apps by using integration service environments (ISEs).

Sometimes, your logic apps and integration accounts need access to secured resources, such as virtual machines (VMs) and other systems or services, that are inside an Azure virtual network.

To set up this access, you can create an integration service environment (ISE) where you can run your logic apps and create your integration accounts.

Reference:

<https://docs.microsoft.com/en-us/azure/logic-apps/connect-virtual-network-vnet-isolated-environment-overview>

QUESTION 2

Case Study 1

Current environment

Windows Server 2016 virtual machine

The virtual machine (VM) runs BizTalk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and tracking stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

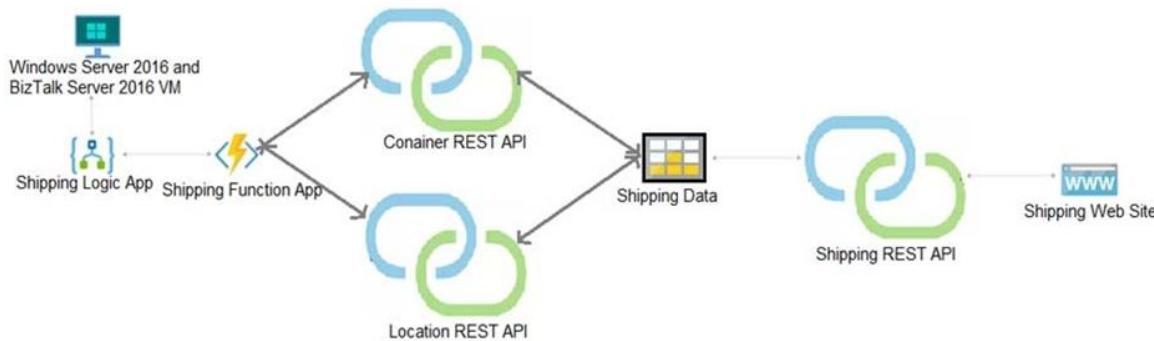
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com/>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Requirements

Shipping Logic app

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

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

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested
```

resource. Origin 'http://test.wideworldimporters.com/' is therefore not allowed access.

You need to support the requirements for the Shipping Logic App.

What should you use?

- A. Azure Active Directory Application Proxy
- B. Site-to-Site (S2S) VPN connection
- C. On-premises Data Gateway
- D. Point-to-Site (P2S) VPN connection

Answer: C

Explanation:

Before you can connect to on-premises data sources from Azure Logic Apps, download and install the on-premises data gateway on a local computer. The gateway works as a bridge that provides quick data transfer and encryption between data sources on premises (not in the cloud) and your logic apps.

The gateway supports BizTalk Server 2016.

Note: Microsoft have now fully incorporated the Azure BizTalk Services capabilities into Logic Apps and Azure App Service Hybrid Connections.

Logic Apps Enterprise Integration pack bring some of the enterprise B2B capabilities like AS2 and X12, EDI standards support Scenario: The Shipping Logic app must meet the following requirements:

Support the ocean transport and inland transport workflows by using a Logic App.

Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.

Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.

Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Reference:

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-gateway-install>

QUESTION 3

Case Study 1

Current environment

Windows Server 2016 virtual machine

The virtual machine (VM) runs BizTalk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and tracking stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

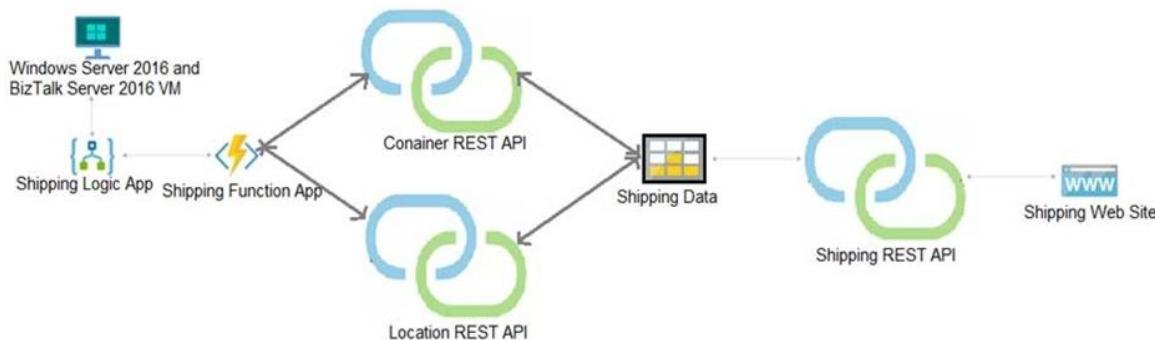
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com/>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Requirements

Shipping Logic app

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

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

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues**Windows Server 2016 VM**

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No
'Access-Control-Allow-Origin' header is present on the requested
resource. Origin 'http://test.wideworldimporters.com/' is therefore not
allowed access.
```

Hotspot Question

You need to configure Azure CDN for the Shipping web site.

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

NOTE: Each correct selection is worth one point.

Answer Area

| Option | Value |
|--------------|--|
| Tier | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #f0f0f0; border-bottom: 1px solid black; padding-bottom: 5px; margin-bottom: 5px;"></div><div>Standard</div><div>Premium</div></div> |
| Profile | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #f0f0f0; border-bottom: 1px solid black; padding-bottom: 5px; margin-bottom: 5px;"></div><div>Akamai</div><div>Microsoft</div></div> |
| Optimization | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #f0f0f0; border-bottom: 1px solid black; padding-bottom: 5px; margin-bottom: 5px;"></div><div>general web delivery</div><div>large file download</div><div>dynamic site acceleration</div><div>video-on-demand media streaming</div></div> |

Answer:

Answer Area

| Option | Value |
|--------------|---|
| Tier | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #e0e0e0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #a0ffa0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #fff; height: 1.2em;">Standard</div><div style="background-color: #fff; height: 1.2em;">Premium</div></div> |
| Profile | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #e0e0e0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #a0ffa0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #fff; height: 1.2em;">Akamai</div><div style="background-color: #fff; height: 1.2em;">Microsoft</div></div> |
| Optimization | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><div style="background-color: #e0e0e0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #e0e0e0; height: 1.2em; margin-bottom: 2px;"></div><div style="background-color: #a0ffa0; height: 1.2em; margin-bottom: 2px;">dynamic site acceleration</div><div style="background-color: #fff; height: 1.2em;">video-on-demand media streaming</div></div> |

Explanation:

Scenario: Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Tier: Standard

Profile: Akamai

Optimization: Dynamic site acceleration

Dynamic site acceleration (DSA) is available for Azure CDN Standard from Akamai, Azure CDN Standard from Verizon, and Azure CDN Premium from Verizon profiles.

DSA includes various techniques that benefit the latency and performance of dynamic content. Techniques include route and network optimization, TCP optimization, and more.

You can use this optimization to accelerate a web app that includes numerous responses that aren't cacheable. Examples are search results, checkout transactions, or real-time data. You can continue to use core Azure CDN caching capabilities for static data.

Reference:

<https://docs.microsoft.com/en-us/azure/cdn/cdn-optimization-overview>

QUESTION 4

Case Study 1

Current environment

Windows Server 2016 virtual machine

The virtual machine (VM) runs BizTalk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and tracking stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

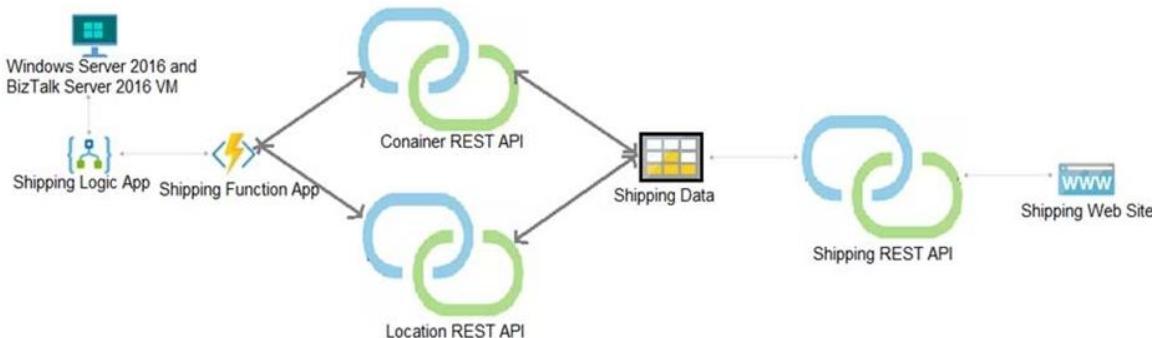
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com/>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Requirements

Shipping Logic app

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

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

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No
'Access-Control-Allow-Origin' header is present on the requested
resource. Origin 'http://test.wideworldimporters.com/' is therefore not
allowed access.
```

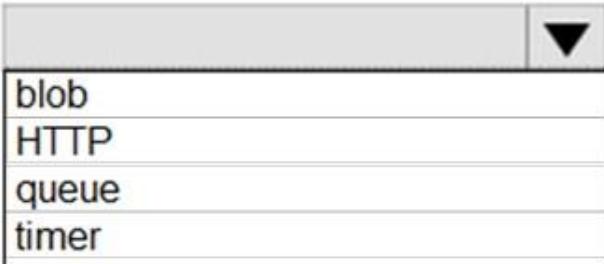
Hotspot Question

You need to secure the Shipping Function app.

How should you configure the app? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|---------------------|---|
| Authorization level |  |
| User claims |  |
| Trigger type |  |

Answer:

Answer Area

| Setting | Value |
|---------------------|---|
| Authorization level | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>Function</p><p>Anonymous</p><p>Admin</p></div> |
| User claims | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>JSON Web Token (JWT)</p><p>Shared Access Signature (SAS) token</p><p>API Key</p></div> |
| Trigger type | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>blob</p><p>HTTP</p><p>queue</p><p>timer</p></div> |

Explanation:

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

Box 1: Function

Box 2: JSON based Token (JWT)

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

Box 3: HTTP

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

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

Reference:

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

QUESTION 5
Case Study 1

Current environment**Windows Server 2016 virtual machine**

The virtual machine (VM) runs BizTalk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and tracking stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

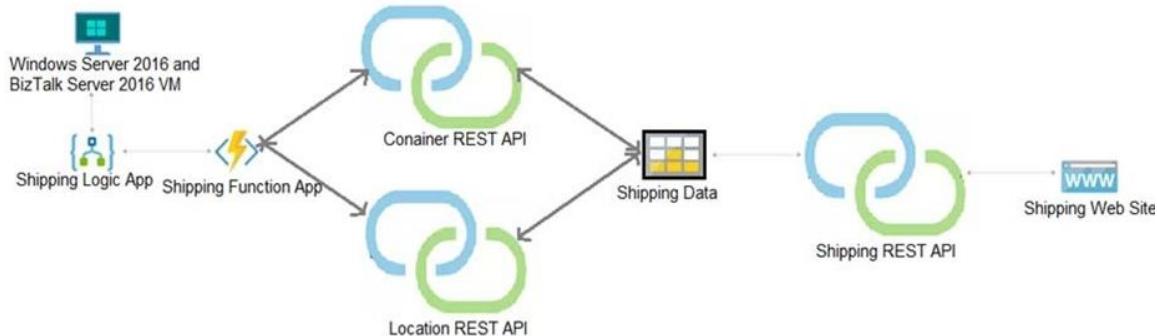
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com/>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Requirements****Shipping Logic app**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.

- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

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

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues**Windows Server 2016 VM**

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No
'Access-Control-Allow-Origin' header is present on the requested
resource. Origin 'http://test.wideworldimporters.com/' is therefore not
allowed access.
```

Drag and Drop Question

You need to support the message processing for the ocean transport workflow.

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

Actions**Answer Area**

Link the Logic App to the integration account.

Add partners, schemas, certificates, maps, and agreements.

Update the Logic App to use the partners, schemas, certificates, maps, and agreements.



Create a custom connector for the Logic App.

Link the custom connector to the Logic App.

Create an integration account in the Azure portal.

Answer:**Actions****Answer Area**

Create an integration account in the Azure portal.

Link the Logic App to the integration account.

Update the Logic App to use the partners, schemas, certificates, maps, and agreements.



Add partners, schemas, certificates, maps, and agreements.



Create a custom connector for the Logic App.



Link the custom connector to the Logic App.

Explanation:

Step 1: Create an integration account in the Azure portal

You can define custom metadata for artifacts in integration accounts and get that metadata during runtime for your logic app to use. For example, you can provide metadata for artifacts, such as partners, agreements, schemas, and maps - all store metadata using key-value pairs.

Step 2: Link the Logic App to the integration account

A logic app that's linked to the integration account and artifact metadata you want to use.

Step 3: Add partners, schemas, certificates, maps, and agreements

Step 4: Create a custom connector for the Logic App.

Reference:

<https://docs.microsoft.com/bs-latn-ba/azure/logic-apps/logic-apps-enterprise-integration-metadata>

QUESTION 6

Case Study 2

Requirements

ContentAnalysisService

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code

ContentUploadService

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10         ports:
CS11           - port: 80
CS12             protocol: TCP
CS13         resources:
CS14           requests:
CS15             cpw: 1.0
CS16             memoryInGB: 1.5
CS17
CS18 ipaddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {  
AM02     "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",  
AM03     "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",  
AM04  
AM05     "createdDateTime" : "2019-12-24T06:01:44Z",  
AM06     "logoUrl" : null,  
AM07     "logoutUrl" : null,  
AM08     "name" : "ContentAnalysisService",  
AM09  
AM10  
AM11     "orgRestrictions" : [],  
AM12     "parentalControlSettings" : {  
AM13         "countriesBlockedForMinors" : [],  
AM14         "legalAgeGroupRule" : "Allow"  
AM15     },  
AM16     "passwordCredentials" : []  
AM17 }
```

You need to configure the ContentUploadService deployment.

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

NOTE: Each correct selection is worth one point.

- A. Add the following markup to line CS23:
types: Private
- B. Add the following markup to line CS24:
osType: Windows
- C. Add the following markup to line CS24:
osType: Linux
- D. Add the following markup to line CS23:
types: Public

Answer: AC

Explanation:

Since all of the services must be available from a virtual network, we have to mention the types as Private

Since all of the services must be available from a virtual network, we have to mention the osType as Linux.

Currently this is the only OS that supports container instances to be available from a virtual network.

<https://docs.microsoft.com/en-us/azure/container-instances/container-instances-virtual-network-concepts#unsupported-networking-scenarios>

QUESTION 7

Case Study 2

Requirements

ContentAnalysisService

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14     "legalAgeGroupRule" : "Allow"
AM15   },
AM16   "passwordCredentials" : []
AM17 }
```

You need to store the user agreements.

Where should you store the agreement after it is completed?

- A. Azure Storage queue
- B. Azure Event Hub
- C. Azure Service Bus topic
- D. Azure Event Grid topic

Answer: B

Explanation:

Azure Event Hub is used for telemetry and distributed data streaming.

This service provides a single solution that enables rapid data retrieval for real-time processing as well as repeated replay of stored raw data. It can capture the streaming data into a file for processing and analysis.

It has the following characteristics:

low latency

capable of receiving and processing millions of events per second

at least once delivery

Reference:

<https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

QUESTION 8

Case Study 2

Requirements

ContentAnalysisService

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.

- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code

ContentUploadService

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {  
AM02     "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",  
AM03     "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",  
AM04  
AM05     "createdDateTime" : "2019-12-24T06:01:44Z",  
AM06     "logoUrl" : null,  
AM07     "logoutUrl" : null,  
AM08     "name" : "ContentAnalysisService",  
AM09  
AM10  
AM11     "orgRestrictions" : [],  
AM12     "parentalControlSettings" : {  
AM13         "countriesBlockedForMinors" : [],  
AM14         "legalAgeGroupRule" : "Allow"  
AM15     },  
AM16     "passwordCredentials" : []  
AM17 }
```

You need to monitor ContentUploadService according to the requirements.

Which command should you use?

- A. az monitor metrics alert create -n alert -g ... - --scopes ... - --condition "avg Percentage CPU > 8"
- B. az monitor metrics alert create -n alert -g ... - --scopes ... - --condition "avg Percentage CPU > 80"
- C. az monitor metrics alert create -n alert -g ... - --scopes ... - --condition "CPU Usage > 800"
- D. az monitor metrics alert create -n alert -g ... - --scopes ... - --condition "CPU Usage > 8"

Answer: B

Explanation:

Scenario: An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores

Reference:

<https://docs.microsoft.com/en-us/cli/azure/monitor/metrics/alert>

QUESTION 9

Case Study 2

Requirements

ContentAnalysisService

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipaddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14       "legalAgeGroupRule" : "Allow"
AM15     },
AM16   "passwordCredentials" : []
AM17 }
```

Hotspot Question

You need to implement the bindings for the CheckUserContent function.

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.

Answer Area

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

Answer:

Answer Area

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

Explanation:

Box 1: [BlobTrigger(..)]

Box 2: [Blob(..)]

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

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

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

Scenario: You must create an Azure Function named CheckUserContent to perform the content checks. The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

Reference:

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

QUESTION 10**Case Study 2****Requirements****ContentAnalysisService**

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipaddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {  
AM02     "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",  
AM03     "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",  
AM04  
AM05     "createdDateTime" : "2019-12-24T06:01:44Z",  
AM06     "logoUrl" : null,  
AM07     "logoutUrl" : null,  
AM08     "name" : "ContentAnalysisService",  
AM09  
AM10  
AM11     "orgRestrictions" : [],  
AM12     "parentalControlSettings" : {  
AM13         "countriesBlockedForMinors" : [],  
AM14         "legalAgeGroupRule" : "Allow"  
AM15     },  
AM16     "passwordCredentials" : []  
AM17 }
```

Drag and Drop Question

You need to add markup at line AM04 to implement the ContentReview role.

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

NOTE: Each correct selection is worth one point.

| Json segments | Answer Area |
|---------------------|--|
| User | "appRoles" : [|
| value | { " [] " : [|
| role | " [] " " |
| Application |], |
| allowedMemberTypes | "displayName" : "ContentReviewer", |
| allowedAccountTypes | "id" : "e1c2ade8-98f8-45fd-aa4a-6d24b512c22a", |
| | "isEnabled" : true, |
| | " [] " " : "ContentReviewer" |
| | } |
| |], |

Answer:

Json segments

role

Application

allowedAccountTypes

Answer Area

```
"appRoles" : [
{
    "allowedMemberTypes" : [
        "User"
    ],
    "displayName": "ContentReviewer",
    "id": "e1c2ade8-98f8-45fd-aa4a-6d24b512c22a",
    "isEnabled" : true,
    "value" : "ContentReviewer"
},
],
```

Explanation:

Box 1: allowedMemberTypes

allowedMemberTypes specifies whether this app role definition can be assigned to users and groups by setting to "User", or to other applications (that are accessing this application in daemon service scenarios) by setting to "Application", or to both.

Note: The following example shows the appRoles that you can assign to users.

```
"appId": "8763f1c4-f988-489c-a51e-158e9ef97d6a",
"appRoles": [
{
    "allowedMemberTypes": [
        "User"
    ],
    "displayName": "Writer",
    "id": "d1c2ade8-98f8-45fd-aa4a-6d06b947c66f",
    "isEnabled": true,
    "description": "Writers Have the ability to create tasks.",
    "value": "Writer"
},
],
"availableToOtherTenants": false,
```

Box 2: User

Scenario: In order to review content a user must be part of a ContentReviewer role.

Box 3: value

value specifies the value which will be included in the roles claim in authentication and access tokens.

Reference:

<https://docs.microsoft.com/en-us/graph/api/resources/approle>

QUESTION 11**Case Study 2****Requirements****ContentAnalysisService**

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14     "legalAgeGroupRule" : "Allow"
AM15   },
AM16   "passwordCredentials" : []
AM17 }
```

Hotspot Question

You need to add code at line AM09 to ensure that users can review content using ContentAnalysisService.

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

NOTE: Each correct selection is worth one point.

Answer Area

| | |
|---------------------------------------|---|
| "allowPublicClient":true | , |
| "oauth2Permissions": ["login"] | , |
| "oauth2AllowUrlPathMatching":true | , |
| "oauth2AllowIdTokenImplicitFlow":true | |

| | |
|--|---|
| "oauth2AllowImplicitFlow": true | , |
| "oauth2RequiredPostResponse":true | , |
| "preAuthorizedApplications":["SPA"] | , |
| "knownClientApplications":["ContentAnalysisService"] | |

Answer:

Answer Area

```
"allowPublicClient":true  
"oauth2Permissions": ["login"]  
"oauth2AllowUrlPathMatching":true  
"oauth2AllowIdTokenImplicitFlow":true
```

```
"oauth2AllowImplicitFlow": true  
"oauth2RequiredPostResponse":true  
"preAuthorizedApplications":["SPA"]  
"knownClientApplications":["ContentAnalysisService"]
```

Explanation:

Box 1: "oauth2Permissions": ["login"]

oauth2Permissions specifies the collection of OAuth 2.0 permission scopes that the web API (resource) app exposes to client apps. These permission scopes may be granted to client apps during consent.

Box 2: "oauth2AllowImplicitFlow":true

For applications (Angular, Ember.js, React.js, and so on), Microsoft identity platform supports the OAuth 2.0 Implicit Grant flow.

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/reference-app-manifest>

QUESTION 12**Case Study 2****Requirements**

ContentAnalysisService

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

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

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU-cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from Internal Virtual Networks (VNets)
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso.Ltd to review content, store cookies on user devices and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

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

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location : westus
CS04 name : contentUploadService
CS05 properties :
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpw: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipaddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol : TCP
CS23
CS24
CS25 networkProfile
CS26 id :
/subscriptions/98..19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14       "legalAgeGroupRule" : "Allow"
AM15     },
AM16   "passwordCredentials" : []
AM17 }
```

Hotspot Question

You need to ensure that network security policies are met.

How should you configure network security? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Technology | Value |
|-----------------|---|
| SSL certificate | <input type="button" value="▼"/> Valid root certificate Self-signed certificate |
| Proxy type | <input type="button" value="▼"/> nginx Azure Application Gateway |

Answer:

Answer Area

| Technology | Value |
|-----------------|---|
| SSL certificate | <input type="button" value="▼"/> Valid root certificate Self-signed certificate |
| Proxy type | <input type="button" value="▼"/> nginx Azure Application Gateway |

Explanation:

Box 1: Valid root certificate

Scenario: All websites and services must use SSL from a valid root certificate authority.

Box 2: Azure Application Gateway**Scenario:**

Any web service accessible over the Internet must be protected from cross site scripting attacks.

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

All parts of the system must support inbound and outbound traffic restrictions.

Azure Web Application Firewall (WAF) on Azure Application Gateway provides centralized protection of your web applications from common exploits and vulnerabilities. Web applications are increasingly targeted by malicious attacks that exploit commonly known vulnerabilities. SQL injection and cross-site scripting are among the most common attacks.

Application Gateway supports autoscaling, SSL offloading, and end-to-end SSL, a web application firewall (WAF), cookie-based session affinity, URL path-based routing, multisite hosting, redirection, rewrite HTTP headers and other features.

Note: Both Nginx and Azure Application Gateway act as a reverse proxy with Layer 7 load-balancing features plus a WAF to ensure strong protection against common web vulnerabilities and exploits.

You can modify Nginx web server configuration/SSL for X-XSS protection. This helps to prevent cross-site scripting exploits by forcing the injection of HTTP headers with X-XSS protection.

Reference:

<https://docs.microsoft.com/en-us/azure/web-application-firewall/ag/ag-overview>

<https://www.upguard.com/articles/10-tips-for-securing-your-nginx-deployment>

QUESTION 13

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 develop a software as a service (SaaS) offering to manage photographs. Users upload photos to a web service which then stores the photos in Azure Storage Blob storage. The storage account type is General-purpose V2.

When photos are uploaded, they must be processed to produce and save a mobile-friendly version of the image. The process to produce a mobile-friendly version of the image must start in less than one minute.

You need to design the process that starts the photo processing.

Solution: Convert the Azure Storage account to a BlockBlobStorage storage account.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Not necessary to convert the account, instead move photo processing to an Azure Function triggered from the blob upload..

Azure Storage events allow applications to react to events. Common Blob storage event scenarios include image or video processing, search indexing, or any file-oriented workflow.

Note: Only storage accounts of kind StorageV2 (general purpose v2) and BlobStorage support event integration. Storage (general purpose v1) does not support integration with Event Grid.

Reference:

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

QUESTION 14

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 develop a software as a service (SaaS) offering to manage photographs. Users upload photos to a web service which then stores the photos in Azure Storage Blob storage. The storage account type is General-purpose V2.

When photos are uploaded, they must be processed to produce and save a mobile-friendly version of the image. The process to produce a mobile-friendly version of the image must start in less than one minute.

You need to design the process that starts the photo processing.

Solution: Move photo processing to an Azure Function triggered from the blob upload.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

Azure Storage events allow applications to react to events. Common Blob storage event scenarios include image or video processing, search indexing, or any file-oriented workflow. Events are pushed using Azure Event Grid to subscribers such as Azure Functions, Azure Logic Apps, or even to your own http listener.

Note: Only storage accounts of kind StorageV2 (general purpose v2) and BlobStorage support event integration. Storage (general purpose v1) does not support integration with Event Grid.

Reference:

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

QUESTION 15

You are developing an application that uses Azure Blob storage.

The application must read the transaction logs of all the changes that occur to the blobs and the blob metadata in the storage account for auditing purposes. The changes must be in the order in which they occurred, include only create, update, delete, and copy operations and be retained for compliance reasons.

You need to process the transaction logs asynchronously.

What should you do?

- A. Process all Azure Blob storage events by using Azure Event Grid with a subscriber Azure Function app.
- B. Enable the change feed on the storage account and process all changes for available events.
- C. Process all Azure Storage Analytics logs for successful blob events.
- D. Use the Azure Monitor HTTP Data Collector API and scan the request body for successful blob events.

Answer: B

Explanation:

Change feed support in Azure Blob Storage

The purpose of the change feed is to provide transaction logs of all the changes that occur to the blobs and the blob metadata in your storage account. The change feed provides ordered, guaranteed, durable, immutable, read-only log of these changes. Client applications can read these logs at any time, either in streaming or in batch mode. The change feed enables you to build efficient and scalable solutions that process change events that occur in your Blob Storage account at a low cost.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-change-feed>

QUESTION 16

You are developing a web app that is protected by Azure Web Application Firewall (WAF). All traffic to the web app is routed through an Azure Application Gateway instance that is used by multiple web apps. The web app address is contoso.azurewebsites.net.

All traffic must be secured with SSL. The Azure Application Gateway instance is used by multiple web apps.

You need to configure the Azure Application Gateway for the app.

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

NOTE: Each correct selection is worth one point.

- A. In the Azure Application Gateway's HTTP setting, enable the Use for App service setting.
- B. Convert the web app to run in an Azure App service environment (ASE).
- C. Add an authentication certificate for contoso.azurewebsites.net to the Azure Application gateway.
- D. In the Azure Application Gateway's HTTP setting, set the value of the Override backend path option to contoso22.azurewebsites.net.

Answer: AD

Explanation:

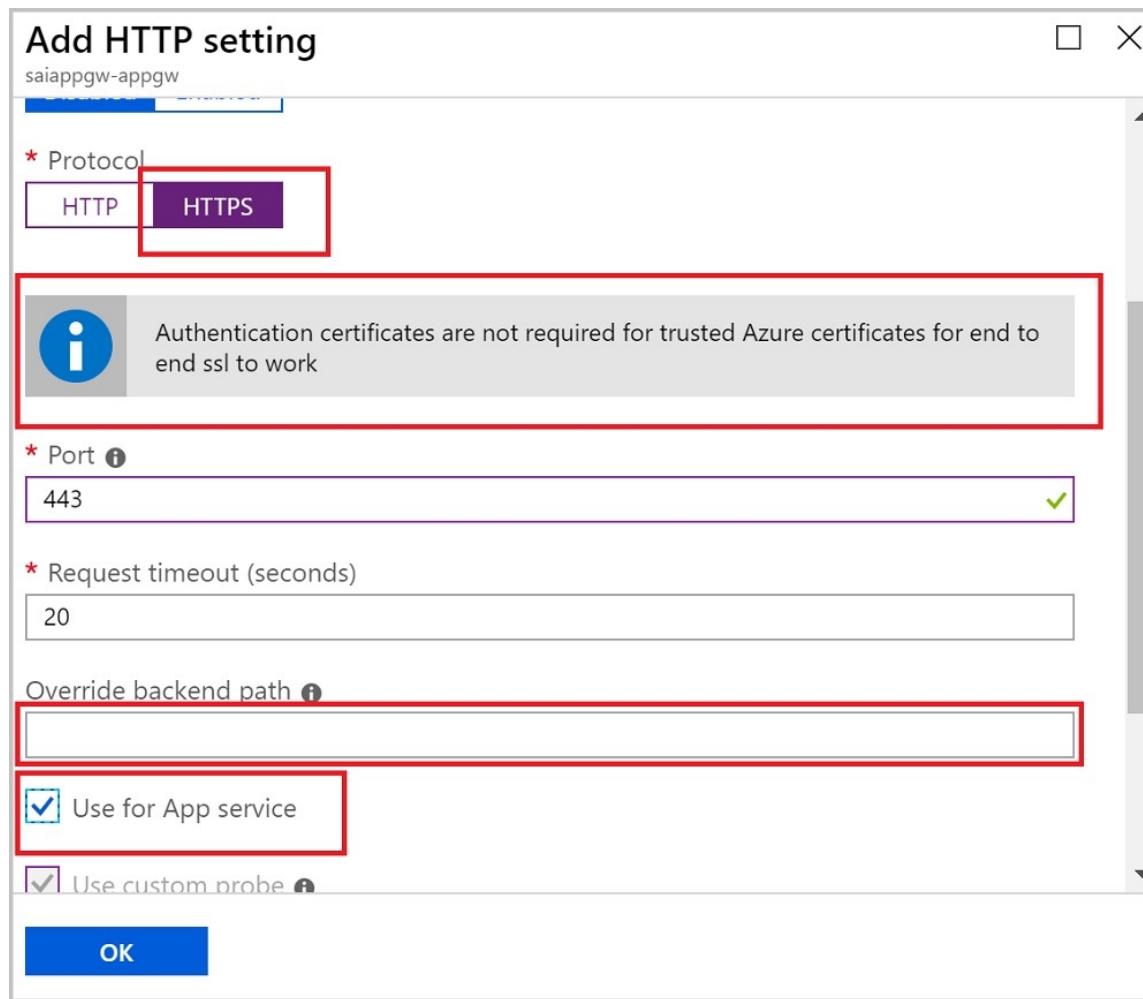
D: The ability to specify a host override is defined in the HTTP settings and can be applied to any back-end pool during rule creation.

The ability to derive the host name from the IP or FQDN of the back-end pool members. HTTP settings also provide an option to dynamically pick the host name from a back-end pool member's FQDN if configured with the option to derive host name from an individual back-end pool member.

A (not C): SSL termination and end to end SSL with multi-tenant services.

In case of end to end SSL, trusted Azure services such as Azure App service web apps do not

require whitelisting the backends in the application gateway. Therefore, there is no need to add any authentication certificates.

**Reference:**

<https://docs.microsoft.com/en-us/azure/application-gateway/application-gateway-web-app-overview>

QUESTION 17

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 develop a software as a service (SaaS) offering to manage photographs. Users upload photos to a web service which then stores the photos in Azure Storage Blob storage. The storage account type is General-purpose V2.

When photos are uploaded, they must be processed to produce and save a mobile-friendly

version of the image. The process to produce a mobile-friendly version of the image must start in less than one minute.

You need to design the process that starts the photo processing.

Solution: Trigger the photo processing from Blob storage events.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

You need to catch the triggered event, so move the photo processing to an Azure Function triggered from the blob upload Note: Azure Storage events allow applications to react to events. Common Blob storage event scenarios include image or video processing, search indexing, or any file-oriented workflow.

Events are pushed using Azure Event Grid to subscribers such as Azure Functions, Azure Logic Apps, or even to your own http listener.

Note: Only storage accounts of kind StorageV2 (general purpose v2) and BlobStorage support event integration. Storage (general purpose v1) does not support integration with Event Grid.

Reference:

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

QUESTION 18

You develop Azure solutions.

You must connect to a No-SQL globally-distributed database by using the .NET API.

You need to create an object to configure and execute requests in the database.

Which code segment should you use?

- A. new Container(EndpointUri, PrimaryKey);
- B. new Database(Endpoint, PrimaryKey);
- C. new CosmosClient(EndpointUri, PrimaryKey);

Answer: C

Explanation:

Example:

```
// Create a new instance of the Cosmos Client  
this.cosmosClient = new CosmosClient(EndpointUri, PrimaryKey)
```

//ADD THIS PART TO YOUR CODE

```
await this.CreateDatabaseAsync();
```

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/sql-api-get-started>

QUESTION 19

Your company is developing an Azure API.

You need to implement authentication for the Azure API. You have the following requirements:

- All API calls must be secure.
- Callers to the API must not send credentials to the API.

Which authentication mechanism should you use?

- A. Basic
- B. Anonymous
- C. Managed identity
- D. Client certificate

Answer: C

Explanation:

Use the authentication-managed-identity policy to authenticate with a backend service using the managed identity of the API Management service. This policy essentially uses the managed identity to obtain an access token from Azure Active Directory for accessing the specified resource. After successfully obtaining the token, the policy will set the value of the token in the Authorization header using the Bearer scheme.

Reference:

<https://docs.microsoft.com/bs-cyrillic/azure/api-management/api-management-authentication-policies>

QUESTION 20

You are a developer for a SaaS company that offers many web services.

All web services for the company must meet the following requirements:

- Use API Management to access the services
- Use OpenID Connect for authentication
- Prevent anonymous usage

A recent security audit found that several web services can be called without any authentication.

Which API Management policy should you implement?

- A. jsonp
- B. authentication-certificate
- C. check-header
- D. validate-jwt

Answer: D

Explanation:

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

Incorrect Answers:

A: The jsonp policy adds JSON with padding (JSONP) support to an operation or an API to allow cross-domain calls from JavaScript browser-based clients.

JSONP is a method used in JavaScript programs to request data from a server in a different domain. JSONP bypasses the limitation enforced by most web browsers where access to web pages must be in the same domain.

JSONP - Adds JSON with padding (JSONP) support to an operation or an API to allow cross-domain calls from JavaScript browser-based clients.

Reference:

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

QUESTION 21

You have a new Azure subscription. You are developing an internal website for employees to view sensitive data. The website uses Azure Active Directory (Azure AD) for authentication.

You need to implement multifactor authentication for the website.

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

NOTE: Each correct selection is worth one point.

- A. Configure the website to use Azure AD B2C.
- B. In Azure AD, create a new conditional access policy.
- C. Upgrade to Azure AD Premium.
- D. In Azure AD, enable application proxy.
- E. In Azure AD conditional access, enable the baseline policy.

Answer: BC

Explanation:

B: 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.

C: 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

Reference:

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

QUESTION 22

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 use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

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

Answer: BD

Explanation:

Example:

```
public async Task Enqueue(string payload)
{
    // StartOperation is a helper method that initializes the telemetry item
```

```
// and allows correlation of this operation with its parent and children.  
var operation = telemetryClient.StartOperation<DependencyTelemetry>("enqueue " +  
queueName); operation.Telemetry.Type = "Azure Service Bus";  
operation.Telemetry.Data = "Enqueue " + queueName;  
var message = new BrokeredMessage(payload);  
// Service Bus queue allows the property bag to pass along with the message.  
// We will use them to pass our correlation identifiers (and other context)  
// to the consumer.  
message.Properties.Add("ParentId", operation.Telemetry.Id);  
message.Properties.Add("RootId", operation.Telemetry.Context.Operation.Id);  
Reference:  
https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking
```

QUESTION 23

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 are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

- Queue size must not grow larger than 80 gigabytes (GB).
- Use first-in-first-out (FIFO) ordering of messages.
- Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Service Bus Queue from the mobile application. Create an Azure Function App that uses an Azure Service Bus Queue trigger.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

You can create a function that is triggered when messages are submitted to an Azure Storage queue.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

QUESTION 24

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device data in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Notification Hub. Register all devices with the hub.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead use an Azure Service Bus, which is used for order processing and financial transactions.

Reference:

<https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

QUESTION 25

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device data in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Service Bus. Configure a topic to receive the device data by using a correlation filter.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

A message is raw data produced by a service to be consumed or stored elsewhere. The Service Bus is for high-value enterprise messaging, and is used for order processing and financial transactions.

Reference:

<https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

QUESTION 26

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device data in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Event Grid. Configure event filtering to evaluate the device identifier.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead use an Azure Service Bus, which is used for order processing and financial transactions.

Note: An event is a lightweight notification of a condition or a state change. Event hubs are usually used for reacting to status changes.

Reference:

<https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

QUESTION 27

A company is developing a solution that allows smart refrigerators to send temperature information to a central location. You have an existing Service Bus.

The solution must receive and store messages until they can be processed. You create an Azure Service Bus Instance by providing a name, pricing tier, subscription, resource group, and location.

You need to complete the configuration.

Which Azure CLI or PowerShell command should you run?

- A. `az servicebus queue create
--resource-group fridge-rg
--namespace-name fridge-ns
--name fridge-q`
- B. `New-AzureRmResourceGroup
-Name fridge-rg
-Location fridge-loc`
- C. `New-AzureRmServiceBusNamespace
-ResourceGroupName fridge-rg
-NamespaceName fridge-loc
-Location fridge-loc`
- D. `connectionString=$(az servicebus namespace authorization-rule keys list
--resource-group fridge-rg
--namespace-name fridge-ns
--query primaryConnectionString --output tsv)`

Answer: A

Explanation:

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

Note:

Steps:

Step 1: # Create a resource group
`resourceGroupName="myResourceGroup"`

`az group create --name $resourceGroupName --location eastus`

Step 2: # Create a Service Bus messaging namespace with a unique name

`namespaceName=myNameSpace$RANDOM`

`az servicebus namespace create --resource-group $resourceGroupName --name
$namespaceName --location eastus`

Step 3: # Create a Service Bus queue

`az servicebus queue create --resource-group $resourceGroupName --namespace-name
$namespaceName --name BasicQueue`

Step 4: # Get the connection string for the namespace

`connectionString=$(az servicebus namespace authorization-rule keys list --resource-group
$resourceGroupName --namespace-name $namespaceName --name
RootManageSharedAccessKey --query primaryConnectionString --output tsv)`

Reference:

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

QUESTION 28

You are developing a solution that will use Azure messaging services.

You need to ensure that the solution uses a publish-subscribe model and eliminates the need for constant polling.

What are two possible ways to achieve the goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Service Bus
- B. Event Hub
- C. Event Grid
- D. Queue

Answer: AC

Explanation:

It is strongly recommended to use available messaging products and services that support a publish-subscribe model, rather than building your own. In Azure, consider using Service Bus or Event Grid. Other technologies that can be used for pub/sub messaging include Redis, RabbitMQ, and Apache Kafka.

Reference:

<https://docs.microsoft.com/en-us/azure/architecture/patterns/publisher-subscriber>

QUESTION 29

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 ensure that the subscription client processes all messages.

Which code segment should you use?

- A. await subscriptionClient.AddRuleAsync(new RuleDescription(RuleDescription.DefaultRuleName, new TrueFilter()));
- B. subscriptionClient = new SubscriptionClient(ServiceBusConnectionString, TopicName, SubscriptionName);
- C. await subscriptionClient.CloseAsync();
- D. subscriptionClient.RegisterMessageHandler(ProcessMessageAsync, messageHandlerOptions);

Answer: D

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);

Reference:

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

QUESTION 30

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 are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

- Queue size must not grow larger than 80 gigabytes (GB) .
- Use first-in-first-out (FIFO) ordering of messages.
- Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Storage Queue from the mobile application. Create an Azure VM that is triggered from Azure Storage Queue events.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Don't use a VM, instead create an Azure Function App that uses an Azure Service Bus Queue trigger.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

QUESTION 31

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 are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

- Queue size must not grow larger than 80 gigabytes (GB) .
- Use first-in-first-out (FIFO) ordering of messages.
- Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Service Bus Queue from the mobile application. Create an Azure Windows VM that is triggered from Azure Service Bus Queue.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Don't use a VM, instead create an Azure Function App that uses an Azure Service Bus Queue trigger.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

QUESTION 32

In the merged camera company, which technology would you use for the lens quality control procedure?

- A. Microsoft Power Automate
- B. Azure Logic Apps
- C. Azure Functions
- D. Azure App Service WebJobs

Answer: B

QUESTION 33

How many triggers must a function have?

- A. Zero
- B. One
- C. Two

Answer: B

QUESTION 34

A CRON expression is a string that consists of six fields that represent a set of times. The order of the six fields in Azure is: {second} {minute} {hour} {day} {month} {day of the week}. Suppose you needed a CRON expression that meant "every day", what special character would you put in the {day of the week} position?

- A. /
- B. *
- C. ,
- D. -

Answer: B

QUESTION 35

What is Durable Functions?

- A. Durable Functions is an extension of Azure Functions, that allow you to simplify complex stateful executions in a serverless-environment

- B. Durable Functions is a logical container for a single workflow that you define using triggers and actions.
- C. Durable Functions is a serverless compute service that enables you to run code on-demand without having to explicitly provision or manage infrastructure.

Answer: A

QUESTION 36

Suppose your Azure Function has a blob trigger associated with it and you want it to execute only when images are uploaded. Which of the following blob trigger Path values should you use?

- A. samples-workitems/{name}
- B. samples-workitems/{name}/png
- C. samples-workitems/{name}?png
- D. samples-workitems/{name}.png

Answer: D

QUESTION 37

Suppose you're sending a message with Azure Service Bus and you want multiple components to receive it. Which Azure Service Bus exchange feature should you use?

- A. Queue
- B. Topic
- C. Relay

Answer: B

QUESTION 38

In the merged camera company, which technology would you use for the ordering and dispatch procedure?

- A. Microsoft Power Automate
- B. Azure Logic Apps
- C. Azure Functions
- D. Azure App Service WebJobs

Answer: B

QUESTION 39

We secured our function against unknown HTTP callers by requiring a function-specific API key be passed with each call. Which of the following fields is the name header in the HTTP requests that needs to contain this key?

- A. x-functions-key
- B. x-requested-with
- C. x-csrf-token

Answer: A

QUESTION 40

Which of the following best describes the role of the Orchestrator function in a workflow?

- A. It's used as the basic unit of work (actions and tasks) in a durable function orchestration.
- B. It's the entry point for creating an instance of a Durable Functions orchestration.
- C. It's used for describing how actions are executed and the order in which actions are executed.

Answer: C

QUESTION 41

Which of the following is an advantage of using bindings in your Azure Functions to access data sources and data sinks?

- A. They provide access to more data sources than are available using code.
- B. They let you connect to Azure resources without authentication.
- C. They simplify the connection process; you don't need to code specific connection logic.

Answer: C

QUESTION 42

A company is building a traffic monitoring system. The system would be monitoring the traffic along 4 highways. The system would be responsible for producing a time series-based analysis report for each highway.

The traffic sensors on each highway have been configured to send its data to Azure Event Hubs. The data from Event Hubs is then consumed by three departments. Each department makes use of an Azure Web App to display the data.

You have to implement the Azure Event Hub instance. You need to implement a solution which ensures data throughput is maximized and latency is minimized.

Which of the following would you use as the partition key?

- A. Highway
- B. Department
- C. Timestamp
- D. Datestamp

Answer: A

Explanation:

Since the data would come in for each highway, the highway represented by probably a highway number would be ideal for the partition key.

The other options are incorrect since they would not provide ideal values for the distribution of data across the partitions.

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features#partitions>

QUESTION 43

You have to deploy a microservice based application to Azure. The application needs to be deployed to an Azure Kubernetes cluster. The solution has the following requirements:

- Reverse proxy capabilities
- Ability to configure traffic routing
- Termination of TLS with a custom certificate

Which of the following would you use to implement a single public IP endpoint to route traffic to multiple microservices?

- A. Helm
- B. Brigade
- C. Kubectl
- D. Ingress Controller
- E. Virtual Kubelet

Answer: D

Explanation:

You can use the Ingress controller to route traffic at the application layer.

An ingress controller is a piece of software that provides reverse proxy, configurable traffic routing, and TLS termination for Kubernetes services. Kubernetes ingress resources are used to configure the ingress rules and routes for individual Kubernetes services. Using an ingress controller and ingress rules, a single IP address can be used to route traffic to multiple services in a Kubernetes cluster.

<https://docs.microsoft.com/en-us/azure/aks/ingress-basic>

QUESTION 44

You have to develop an ASP.Net Core application. The application is used to work with blobs in an Azure storage account. The application authenticates via Azure AD credentials.

Role based access has been implemented on the containers that contain the blobs. These roles have been assigned to the users.

You have to configure the application so that the user's permissions can be used with the Azure Blob containers.

Which of the following would you use as the Permission for the Microsoft Graph API?

- A. User.Read
- B. User.Write
- C. client_id
- D. user_impersonation

Answer: A

Explanation:

For the Microsoft Graph API, we need to use the User.Read permission.

The **API permissions** pane now shows that your registered Azure AD application has access to both Microsoft Graph and the Azure Storage. Permissions are granted to Microsoft Graph automatically when you first register your app with Azure AD.

| API / PERMISSIONS NAME | TYPE | DESCRIPTION | ADMIN CONSENT REQUIRED |
|------------------------|-----------|-------------------------------|------------------------|
| user_impersonation | Delegated | Access Azure Storage | - |
| User.Read | Delegated | Sign in and read user profile | - |

These are the permissions that this application requests statically. You may also request user consent-able permissions dynamically through code. See best practices for requesting permissions

Since this is clearly given in the documentation, all other options are incorrect/
<https://docs.microsoft.com/en-us/azure/storage/common/storage-auth-aad-app>

QUESTION 45

You have to build a web application that would be deployed onto Azure. The web application would not allow anonymous access. The authentication would be carried out via Azure AD.

The application needs to above by the following requirements

- Users must be able to log into the web application using their Azure AD credentials
- The personalization of the web application must be based on the membership in Active Directory groups

You have to configure the application manifest file:

```
{  
...  
"appId" :"44d3ece4-2c21-48c1-8857-db3524a086b0"  
    Slot 1 : "All",  
    Slot 2 : true  
}
```

Which of the following would go into Slot 2?

- A. "allowPublicClient"

- B. "oauth2Permissions"
- C. "requiredResourceAccess"
- D. "oauth2AllowImplicitFlow"

Answer: B

Explanation:

The "OAuth2Permissions" is used for web API permissions

The Microsoft documentation mentions the following:

| | | |
|-----|------------|---|
| oau | Collection | Specifies the collection of OAuth 2.0 permission scopes that the web API (resource) app exposes to client apps. These permission scopes may be granted to client apps during consent. |
| th2 | | |
| Per | | |
| mis | | |
| sio | | |
| ns | | |

Option A is incorrect since this is used to specify a fallback application type.

Option C is incorrect since this is used to provide a list of permission scopes and app roles that an application requires from a specified resource.

Option D is incorrect since this is used for single page applications.

<https://docs.microsoft.com/en-us/azure/active-directory/develop/reference-app-manifest>

QUESTION 46

You are developing an application that is going to make use of the Azure Service Bus. You have to create filters based on the different types of subscribers that would subscribe to the topic. The broad classification of these subscribers are:

- Subscribers should be able to receive all messages being sent to the topic
- Subscribers should NOT be able to receive all messages being sent to the topic
- Subscribers should be able to receive messages based on a SQL-like conditional expression

Which of the following would you use as the filter condition for the requirement?

"Subscribers should be able to receive all messages being sent to the topic"

- A. Boolean filters
- B. Primary filters
- C. SQL filters
- D. Correlation filters

Answer: A

Explanation:

Here we have to make use of Boolean filters which could either accept or reject all messages"

Service Bus supports three filter conditions:

- *Boolean filters* - The **TrueFilter** and **FalseFilter** either cause all arriving messages (**true**) or none of the arriving messages (**false**) to be selected for the subscription.
- *SQL Filters* - A **SqlFilter** holds a SQL-like conditional expression that is evaluated in the broker against the arriving messages' user-defined properties and system properties. All system properties must be prefixed with `sys.` in the conditional expression. The [SQL-language subset for filter conditions](#) tests for the existence of properties (`EXISTS`), as well as for null-values (`IS NULL`), logical NOT/AND/OR, relational operators, simple numeric arithmetic, and simple text pattern matching with `LIKE`.
- *Correlation Filters* - A **CorrelationFilter** holds a set of conditions that are matched against one or more of an arriving message's user and system properties. A common use is to match against the **CorrelationId** property, but the application can also choose to match against **ContentType**, **Label**, **MessageId**, **ReplyTo**, **ReplyToSessionId**, **SessionId**, **To**, and any user-defined properties. A match exists when an arriving message's value for a property is equal to the value specified in the correlation filter. For string expressions, the comparison is case-sensitive. When specifying multiple match properties, the filter combines them as a logical AND condition, meaning for the filter to match, all conditions must match.

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

QUESTION 47

A company has a web application that has been deployed using the Azure Web App service. The current service plan being used is D1. It needs to be ensured that the application infrastructure can automatically scale when the CPU load reaches 85 percent.

You also have to ensure costs are minimized.

Which of the following steps would you implement to achieve the requirements? Choose 4 answers from the options given below

- A. Enable autoscaling on the Web application
- B. Configure a scale condition
- C. Configure the web application to use the Standard App Service Plan
- D. Configure the web application to use the Premium App Service Plan
- E. Add a scale rule.

Answer: ABCE

Explanation:

<https://blogs.msdn.microsoft.com/benjaminperkins/2017/07/26/how-to-configure-auto-scaling-for-an-azure-app-service-with-powershell/>

QUESTION 48

A company is implementing an order processing system. The orders are going to be published to an Azure Service Bus topic. The properties of the messages that would be sent are as follows:

| Property | Description |
|---------------|---|
| Location | The region of the shipment |
| CorrelationId | Used as the priority value for the order |
| Quantity | User defined property that defines the order quantity |
| Audited | User defined property that defines the order date |

The following subscriptions will be created. The requirement for each subscription is also given:

| Property | Description |
|--------------------|---|
| LaterOrders | This subscription will be used in the future and should not accept any orders at the moment |
| HighPriorityOrders | Here all the high priority orders should be sent |
| GlobalOrders | Here the order where the region is not USA should be sent |
| HighOrders | Orders where the quantity is greater than 1000 should be sent |
| AllOrders | For auditing purposes, all orders should be sent here |

You need to implement the right filters for each of the subscriptions given above.

Which of the following would you implement for the Subscription - AllOrders?

- A. SqlFilter
- B. CorrelationFilter
- C. TrueFilter
- D. No Filter
- E. FalseFilter

Answer: D

Explanation:

Here since you want all messages to come for the subscription, there is no need to add any filter.
<https://docs.microsoft.com/en-us/azure/service-bus-messaging/topic-filters>

QUESTION 49

Drag and Drop Question

You are developing an application to use Azure Blob storage. You have configured Azure Blob storage to include change feeds.

A copy of your storage account must be created in another region. Data must be copied from the current storage account to the new storage account directly between the storage servers.

You need to create a copy of the storage account in another region and copy the data.

In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Actions**Answer Area**

Use AZCopy to copy the data to the new storage account.

Deploy the template to create a new storage account in the target region.

Export a Resource Manager template.

Create a new template deployment.

Modify the template by changing the storage account name and region.

**Answer:****Actions****Answer Area**

Create a new template deployment.

Export a Resource Manager template.



Modify the template by changing the storage account name and region.

Deploy the template to create a new storage account in the target region.

Use AZCopy to copy the data to the new storage account.

Explanation:

To move a storage account, create a copy of your storage account in another region. Then, move your data to that account by using AzCopy, or another tool of your choice.

The steps are:

- Export a template.
- Modify the template by adding the target region and storage account name.
- Deploy the template to create the new storage account.
- Configure the new storage account.
- Move data to the new storage account.
- Delete the resources in the source region.

Note: You must enable the change feed on your storage account to begin capturing and recording changes. You can enable and disable changes by using Azure Resource Manager templates on Portal or Powershell.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-account-move>

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-change-feed>

QUESTION 50

Hotspot Question

You are developing an ASP.NET Core web application. You plan to deploy the application to Azure Web App for Containers.

The application needs to store runtime diagnostic data that must be persisted across application restarts. You have the following code:

```
public void SaveDiagData(string data)
{
    var path = Environment.GetEnvironmentVariable("DIAGDATA")
    File.WriteAllText(Path.Combine(path, "data"), data);
}
```

You need to configure the application settings so that diagnostic data is stored as required.

How should you configure the web app's settings? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| App setting | Value |
|-------------------------------------|--|
| LOCALAPPDATA | true |
| WEBSITE_LOCALCACHE_ENABLED | |
| DOTNET_HOSTING_OPTIMIZATION_CACHE | |
| WEBSITES_ENABLE_APP_SERVICE_STORAGE | |
| DIAGDATA | /home /local D:\home D:\local |

Answer:

Answer Area

| App setting | Value |
|-------------------------------------|-------|
| LOCALAPPDATA | true |
| WEBSITE_LOCALCACHE_ENABLED | |
| DOTNET_HOSTING_OPTIMIZATION_CACHE | |
| WEBSITES_ENABLE_APP_SERVICE_STORAGE | |
| DIAGDATA | |

Explanation:

Box 1: If WEBSITES_ENABLE_APP_SERVICE_STORAGE

If WEBSITES_ENABLE_APP_SERVICE_STORAGE setting is unspecified or set to true, the /home/ directory will be shared across scale instances, and files written will persist across restarts

Box 2: /home

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/containers/app-service-linux-faq>

QUESTION 51

Hotspot Question

You are implementing a software as a service (SaaS) ASP.NET Core web service that will run as an Azure Web App. The web service will use an on-premises SQL Server database for storage. The web service also includes a WebJob that processes data updates. Four customers will use the web service.

- Each instance of the WebJob processes data for a single customer and must run as a singleton instance.
- Each deployment must be tested by using deployment slots prior to serving production data.
- Azure costs must be minimized.
- Azure resources must be located in an isolated network.

You need to configure the App Service plan for the Web App.

How should you configure the App Service plan? To answer, select the appropriate settings in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| App service plan setting | Value |
|--------------------------|-------|
|--------------------------|-------|

Number of VM instances

| |
|----|
| 2 |
| 4 |
| 8 |
| 16 |

Pricing tier

| |
|-------------|
| Isolated |
| Standard |
| Premium |
| Consumption |

Answer:

Answer Area

| App service plan setting | Value | | | | |
|--------------------------|--|----------|----------|---------|-------------|
| Number of VM instances | <table border="1"><tr><td>2</td></tr><tr><td>4</td></tr><tr><td>8</td></tr><tr><td>16</td></tr></table> | 2 | 4 | 8 | 16 |
| 2 | | | | | |
| 4 | | | | | |
| 8 | | | | | |
| 16 | | | | | |
| Pricing tier | <table border="1"><tr><td>Isolated</td></tr><tr><td>Standard</td></tr><tr><td>Premium</td></tr><tr><td>Consumption</td></tr></table> | Isolated | Standard | Premium | Consumption |
| Isolated | | | | | |
| Standard | | | | | |
| Premium | | | | | |
| Consumption | | | | | |

Explanation:

Number of VM instances: 4

You are not charged extra for deployment slots.

Pricing tier: Isolated

The App Service Environment (ASE) is a powerful feature offering of the Azure App Service that gives network isolation and improved scale capabilities. It is essentially a deployment of the Azure App Service into a subnet of a customer's Azure Virtual Network (VNet).

Reference:

<https://azure.microsoft.com/sv-se/blog/announcing-app-service-isolated-more-power-scale-and-ease-of-use/>

QUESTION 52

Drag and Drop Question

You are a developer for a software as a service (SaaS) company that uses an Azure Function to process orders. The Azure Function currently runs on an Azure Function app that is triggered by an Azure Storage queue.

You are preparing to migrate the Azure Function to Kubernetes using Kubernetes-based Event Driven Autoscaling (KEDA).

You need to configure Kubernetes Custom Resource Definitions (CRD) for the Azure Function.

Which CRDs should you configure? To answer, drag the appropriate CRD types to the correct locations. Each CRD 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.

Answer Area

| CRD types | Setting | CRD type |
|-----------------------|---------------------------------|----------|
| Secret | Azure Function code | |
| Deployment | Polling interval | |
| ScaledObject | Azure Storage connection string | |
| TriggerAuthentication | | |

Answer:

Answer Area

| CRD types | Setting | CRD type |
|-----------------------|---------------------------------|--------------|
| | Azure Function code | Deployment |
| | Polling interval | ScaledObject |
| TriggerAuthentication | Azure Storage connection string | Secret |

Explanation:

Box 1: Deployment

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

Box 2: ScaledObject

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

Example of ScaledObject with polling interval

apiVersion: keda.k8s.io/v1alpha1

kind: ScaledObject

metadata:

name: transformer-fn

namespace: tt

labels:

deploymentName: transformer-fn

spec:

```
scaleTargetRef:  
deploymentName: transformer-fn  
pollingInterval: 5  
minReplicaCount: 0  
maxReplicaCount: 100
```

Box 3: Secret

Store connection strings in Kubernetes Secrets.

Example: to create the Secret in our demo Namespace:

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

Reference:

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

QUESTION 53

Hotspot Question

You are creating a CLI script that creates an Azure web app and related services in Azure App Service. The web app uses the following variables:

| Variable name | Value |
|---------------|---|
| \$gitrepo | https://github.com/Contos/webapp |
| \$webappname | Webapp1103 |

You need to automatically deploy code from Git-Hub to the newly created web app.

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

NOTE: Each correct selection is worth one point.

Answer Area

```
az group create - --location westeurope - --name myResourceGroup
```

```
az webapp create  
az appservice plan create  
az webapp deployment  
az group delete
```

```
--name $webappname - --resource-group myResourceGroup  
az webapp create  
az appservice plan create  
az webapp deployment  
az group delete
```

```
--repo-url $gitrepo - --branch master - --manual-integration  
git clone $gitrepo  
--plan $webappname
```

```
source config --name $webappname  
az webapp create  
az appservice plan create  
az webapp deployment  
az group delete
```

```
--resource-group myResourceGroup
```

```
--repo-url $gitrepo - --branch master - --manual-integration  
git clone $gitrepo  
--plan $webappname
```

Answer:

Answer Area

```
az group create - --location westeurope - --name myResourceGroup  
az webapp create --name $webappname - --resource-group myResourceGroup - --sku FREE  
az appservice plan create  
az webapp deployment  
az group delete
```

```
az webapp create --name $webappname - --resource-group myResourceGroup  
az appservice plan create  
az webapp deployment  
az group delete
```

```
--repo-url $gitrepo - --branch master - --manual-integration  
git clone $gitrepo  
--plan $webappname
```

```
source config --name $webappname  
az webapp create  
az appservice plan create  
az webapp deployment  
az group delete
```

```
--resource-group myResourceGroup  
--repo-url $gitrepo - --branch master - --manual-integration  
git clone $gitrepo  
--plan $webappname
```

Explanation:

Box 1: az appservice plan create

The azure group creates command successfully returns JSON result. Now we can use resource group to create a azure app service plan

Box 2: az webapp create
Create a new web app..

Box 3: --plan \$webappname
..with the serviceplan we created in step 1.

Box 4: az webapp deployment
Continuous Delivery with GitHub. Example:
az webapp deployment source config --name firstsamplewebsite1 --resource-group websites--
repo-url \$gitrepo --branch master --git-token \$token

Box 5: --repo-url \$gitrepo --branch master --manual-integration

Reference:

<https://medium.com/@satish1v/devops-your-way-to-azure-web-apps-with-azure-cli-206ed4b3e9b1>

QUESTION 54

Hotspot Question

You are developing a ticket reservation system for an airline.

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

- Ensure at least 99.99% availability and provide low latency.
- Accept reservations even when localized network outages or other unforeseen failures occur.
- Process reservations in the exact sequence as reservations are submitted to minimize overbooking or selling the same seat to multiple travelers.
- Allow simultaneous and out-of-order reservations with a maximum five-second tolerance window.

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

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

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

NOTE: Each correct selection is worth one point.

Answer Area

```
resourceGroupName- +airlineResourceGroup'
name- +docdb-airline-reservations'
databaseName- 'docdb-tickets-database'
collectionName- 'docdb-tickets-collection'
consistencyLevel-
```

| |
|------------------|
| Strong |
| Eventual |
| ConsistentPrefix |
| BoundedStaleness |

```
az cosmosdb create \
--name $name \
--enable-virtual-network true \
--enable-automatic-failover true \
--kind 'GlobalDocumentDB' \
--kind 'MongoDB' \
--resource group $resourceGroupName \
--max interval 5 \
```

| |
|--|
| --locations 'southcentralus' |
| --locations 'eastus' |
| --locations 'southcentralus=0 eastus=1 westus=2' |
| --locations 'southcentralus=0' |

```
--default-consistency-level - $consistencylevel
```

Answer:

Answer Area

```
resourceGroupName- +airlineResourceGroup'
name- +docdb-airline-reservations'
databaseName- 'docdb-tickets-database'
collectionName- 'docdb-tickets-collection'
consistencyLevel-
```

| |
|------------------|
| Strong |
| Eventual |
| ConsistentPrefix |
| BoundedStaleness |

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

```
--enable-virtual-network true\
--enable-automatic-failover true\
--kind 'GlobalDocumentDB' \
--kind 'MongoDB' \
```

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

```
--locations 'southcentralus'
--locations 'eastus'
--locations'southcentralus=0 eastus=1 westus=2'
--locations 'southcentralus=0'
```

```
--default-consistency-level - $consistencylevel
```

Explanation:

Box 1: BoundedStaleness

Bounded staleness: The reads are guaranteed to honor the consistent-prefix guarantee. The reads might lag behind writes by at most "K" versions (that is, "updates") of an item or by "T" time interval. In other words, when you choose bounded staleness, the "staleness" can be configured in two ways:

The number of versions (K) of the item

The time interval (T) by which the reads might lag behind the writes

Incorrect Answers:

Strong

Strong consistency offers a linearizability guarantee. Linearizability refers to serving requests concurrently. The reads are guaranteed to return the most recent committed version of an item. A client never sees an uncommitted or partial write. Users are always guaranteed to read the latest

committed write.

Box 2: --enable-automatic-failover true\

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

Question: Accept reservations event when localized network outages or other unforeseen failures occur.

Box 3: --locations'southcentralus=0 eastus=1 westus=2

Need multi-region.

Reference:

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

<https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/cosmos-db/manage-with-cli.md>

QUESTION 55

Drag and Drop Question

You are developing a new page for a website that uses Azure Cosmos DB for data storage. The feature uses documents that have the following format:

```
{  
    "name": "John",  
    "city" : "Seattle"  
}
```

You must display data for the new page in a specific order. You create the following query for the page:

```
SELECT*  
FROM People p  
ORDER BY p.name, p.city DESC
```

You need to configure a Cosmos DB policy to support the query.

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

NOTE: Each correct selection is worth one point.

JSON segments

orderBy
sortOrder
ascending
descending
compositeIndexes

Answer Area

```
{
    "automatic": true,
    "ngMode": "Consistent",
    "includedPaths": [
        {
            "path": "/**"
        }
    ],
    "excludedPaths": [],
    "compositeIndexes": [
        [
            {
                "path": "/name", "order": "descending"
            },
            {
                "path": "/city", "order": "descending"
            }
        ]
    ]
}
```

Answer:

JSON segments

orderBy
sortOrder
ascending

Answer Area

```
{
    "automatic": true,
    "ngMode": "Consistent",
    "includedPaths": [
        {
            "path": "/**"
        }
    ],
    "excludedPaths": [],
    "compositeIndexes": [
        [
            {
                "path": "/name", "order": "descending"
            },
            {
                "path": "/city", "order": "descending"
            }
        ]
    ]
}
```

Explanation:

Box 1: compositeIndexes

You can order by multiple properties. A query that orders by multiple properties requires a composite index.

Box 2: descending

Example: Composite index defined for (name ASC, age ASC):

It is optional to specify the order. If not specified, the order is ascending.

```
{
    "automatic": true,
```

```
"indexingMode":"Consistent",
"includedPaths":[
{
"path": "*"
},
],
"excludedPaths":[],
"compositeIndexes":[
[
{
"path": "/name",
},
{
"path": "/age",
}
]
]
}
```

QUESTION 56

Hotspot Question

You are building a traffic monitoring system that monitors traffic along six highways. The system produces time series analysis-based reports for each highway.

Data from traffic sensors are stored in Azure Event Hub.

Traffic data is consumed by four departments. Each department has an Azure Web App that displays the time series-based reports and contains a WebJob that processes the incoming data from Event Hub. All Web Apps run on App Service Plans with three instances.

Data throughput must be maximized. Latency must be minimized.

You need to implement the Azure Event Hub.

Which settings should you use? To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value | | | | |
|----------------------|---|---------|------------|-----------|---------|
| Number of partitions | <table border="1"><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>6</td></tr><tr><td>12</td></tr></table> | 3 | 4 | 6 | 12 |
| 3 | | | | | |
| 4 | | | | | |
| 6 | | | | | |
| 12 | | | | | |
| Partition Key | <table border="1"><tr><td>Highway</td></tr><tr><td>Department</td></tr><tr><td>Timestamp</td></tr><tr><td>VM name</td></tr></table> | Highway | Department | Timestamp | VM name |
| Highway | | | | | |
| Department | | | | | |
| Timestamp | | | | | |
| VM name | | | | | |

Answer:**Answer Area**

| Setting | Value | | | | |
|----------------------|---|---------|------------|-----------|---------|
| Number of partitions | <table border="1"><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>6</td></tr><tr><td>12</td></tr></table> | 3 | 4 | 6 | 12 |
| 3 | | | | | |
| 4 | | | | | |
| 6 | | | | | |
| 12 | | | | | |
| Partition Key | <table border="1"><tr><td>Highway</td></tr><tr><td>Department</td></tr><tr><td>Timestamp</td></tr><tr><td>VM name</td></tr></table> | Highway | Department | Timestamp | VM name |
| Highway | | | | | |
| Department | | | | | |
| Timestamp | | | | | |
| VM name | | | | | |

Explanation:

Box 1: 6

The number of partitions is specified at creation and must be between 2 and 32.
There are 6 highways.

Box 2: Highway

Reference:

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features>**QUESTION 57**

Drag and Drop Question

You are developing a microservices solution. You plan to deploy the solution to a multinode Azure Kubernetes Service (AKS) cluster.

You need to deploy a solution that includes the following features:

- reverse proxy capabilities
- configurable traffic routing
- TLS termination with a custom certificate

Which component should you use? To answer, drag the appropriate components to the correct requirements. Each component 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.

Answer Area

| Components | Action | Component |
|--------------------|--|-----------|
| Helm | | |
| Draft | Deploy solution. | |
| Brigade | View cluster and external IP addressing. | |
| KubeCtl | | |
| Ingress Controller | Implement a single, public IP endpoint that is routed to multiple microservices. | |
| CoreDNS | | |
| Virtual Kubelet | | |

Answer:

Answer Area

| Components | Action | Component |
|-----------------|--|--------------------|
| Draft | Deploy solution. | Helm |
| Brigade | View cluster and external IP addressing. | KubeCtl |
| | Implement a single, public IP endpoint that is routed to multiple microservices. | Ingress Controller |
| CoreDNS | | |
| Virtual Kubelet | | |

Explanation:

Box 1: Helm

To create the ingress controller, use Helm to install nginx-ingress.

Box 2: kubectl

To find the cluster IP address of a Kubernetes pod, use the kubectl get pod command on your local machine, with the option -o wide .

Box 3: Ingress Controller

An ingress controller is a piece of software that provides reverse proxy, configurable traffic routing, and TLS termination for Kubernetes services. Kubernetes ingress resources are used to configure the ingress rules and routes for individual Kubernetes services.

Incorrect Answers:

Virtual Kubelet: Virtual Kubelet is an open-source Kubernetes kubelet implementation that masquerades as a kubelet. This allows Kubernetes nodes to be backed by Virtual Kubelet providers such as serverless cloud container platforms.

CoreDNS: CoreDNS is a flexible, extensible DNS server that can serve as the Kubernetes cluster DNS. Like Kubernetes, the CoreDNS project is hosted by the CNCF.

Reference:

<https://docs.microsoft.com/bs-cyrillic/azure/aks/ingress-basic>

<https://www.digitalocean.com/community/tutorials/how-to-inspect-kubernetes-networking>

QUESTION 58

Drag and Drop Question

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.

Answer Area

| Policy section | Policy | Policy section |
|----------------|--------------------|----------------------|
| | Set-variable | <input type="text"/> |
| Inbound | Cache-lookup-value | <input type="text"/> |
| Outbound | Cache-store-value | <input type="text"/> |
| | Find-and-replace | <input type="text"/> |

Answer:

Answer Area

| Policy section | Policy | Policy section |
|----------------|--------------------|----------------|
| | Set-variable | Inbound |
| Inbound | Cache-lookup-value | Inbound |
| Outbound | Cache-store-value | Outbound |
| | Find-and-replace | Outbound |

Explanation:

Box 1: Inbound.

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

Example:

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

Box 2: Inbound

A cache-lookup-value policy

Example:

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

Box 3: Outbound

A cache-store-value policy.

Example:

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

Box 4: Outbound

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

Example:

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

Reference:

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

QUESTION 59

Drag and Drop Question

You develop a web application.

You need to register the application with an active Azure Active Directory (Azure AD) tenant.

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

| Actions | Answer Area |
|---|---|
| Select Manifest from the middle-tier service registration. | |
| In Enterprise Applications, select New application . | |
| Add a Cryptographic key. | |
| Create a new application and provide the name, account type, and redirect URL |   |
| Select the Azure AD instance. | |
| Use an access token to access the secure resource. | |
| In App Registrations, select New registration . |   |

Answer:

Actions

Select **Manifest** from the middle-tier service registration.

In Enterprise Applications, select **New application**.

Add a Cryptographic key.

Answer Area

In App Registrations, select **New registration**.

Select the Azure AD instance.

Create a new application and provide the name, account type, and redirect URL



Use an access token to access the secure resource.

Explanation:

Register a new application using the Azure portal

1. Sign in to the Azure portal using either a work or school account or a personal Microsoft account.
2. If your account gives you access to more than one tenant, select your account in the upper right corner. Set your portal session to the Azure AD tenant that you want.
3. Search for and select Azure Active Directory. Under Manage, select App registrations.
4. Select New registration. (Step 1)
5. In Register an application, enter a meaningful application name to display to users.
6. Specify who can use the application. Select the Azure AD instance. (Step 2)
7. Under Redirect URI (optional), select the type of app you're building: Web or Public client (mobile & desktop). Then enter the redirect URI, or reply URL, for your application. (Step 3)
8. When finished, select Register.

QUESTION 60

Drag and Drop Question

You are developing an application. You have an Azure user account that has access to two subscriptions.

You need to retrieve a storage account key secret from Azure Key Vault.

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

Powershell commands**Answer Area**

```
$secretvalue = ConvertTo-SecureString  
$storAcctkey -AsPlainText  
-Force  
    Set-AzKeyVaultSecret -VaultName  
$vaultName -Name $secretName  
-SecretValue $secretvalue
```

```
Get-AzStorageAccountKey -  
ResourceGroupName $resGroup -Name  
$storAcct
```

```
Set-AzContext -SubscriptionId  
$subscriptionID
```



```
Get-AzKeyVaultSecret -VaultName  
$vaultName
```

```
Get-AzSubscription
```

Answer:**Powershell commands****Answer Area**

```
Get-AzSubscription
```

```
Set-AzContext -SubscriptionId  
$subscriptionID
```



```
Get-AzStorageAccountKey -  
ResourceGroupName $resGroup -Name  
$storAcct
```

```
$secretvalue = ConvertTo-SecureString  
$storAcctkey -AsPlainText  
-Force  
    Set-AzKeyVaultSecret -VaultName  
$vaultName -Name $secretName  
-SecretValue $secretvalue
```

```
Get-AzKeyVaultSecret -VaultName  
$vaultName
```

Explanation:**Step 1: Get-AzSubscription**

If you have multiple subscriptions, you might have to specify the one that was used to create your key vault. Enter the following to see the subscriptions for your account:

```
Get-AzSubscription
```

Step 2: Set-AzContext -SubscriptionId

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

Set-AzContext -SubscriptionId <subscriptionID>

Step 3: Get-AzStorageAccountKey

You must get that storage account key.

Step 4: \$secretvalue = ConvertTo-SecureString <storageAccountKey> -AsPlainText -Force

Set-AzKeyVaultSecret -VaultName <vaultName> -Name <secretName> -SecretValue

\$secretvalue

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

Step 5: Get-AzKeyVaultSecret

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

Get-AzKeyVaultSecret –VaultName <vaultName>

Reference:

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

QUESTION 61

Hotspot Question

You are using Azure Front Door Service.

You are expecting inbound files to be compressed by using Brotli compression. You discover that inbound XML files are not compressed. The files are 9 megabytes (MB) in size.

You need to determine the root cause for the issue.

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

NOTE: Each correct selection is worth one point.

Answer Area

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| The file MIME type is supported by the service. | <input type="radio"/> | <input type="radio"/> |
| Edge nodes must be purged of all cache assets. | <input type="radio"/> | <input type="radio"/> |
| The compression type is supported. | <input type="radio"/> | <input type="radio"/> |

Answer:**Answer Area**

| Statement | Yes | No |
|---|----------------------------------|----------------------------------|
| The file MIME type is supported by the service. | <input type="radio"/> | <input checked="" type="radio"/> |
| Edge nodes must be purged of all cache assets. | <input checked="" type="radio"/> | <input type="radio"/> |
| The compression type is supported. | <input checked="" type="radio"/> | <input type="radio"/> |

Explanation:

Box 1: No

Front Door can dynamically compress content on the edge, resulting in a smaller and faster response to your clients. All files are eligible for compression.

However, a file must be of a MIME type that is eligible for compression list.

Box 2: No

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

Box 3: Yes

These profiles support the following compression encodings: Gzip (GNU zip), Brotli

Reference:

<https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching>

QUESTION 62

Hotspot Question

You are developing an Azure App Service hosted ASP.NET Core web app to deliver video on-demand streaming media. You enable an Azure Content Delivery Network (CDN) Standard for the web endpoint. Customer videos are downloaded from the web app by using the following example URL.: <http://www.contoso.com/content.mp4?quality=1>

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

You need to configure Azure CDN caching rules.

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

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Action | | | | |
|----------------------------------|---|----------------------|----------------------------------|------------------------|-------|
| Caching behavior | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><p>▼</p><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td>Bypass cache</td></tr><tr><td>Override</td></tr><tr><td>Set if missing</td></tr></table></div> | Bypass cache | Override | Set if missing | |
| Bypass cache | | | | | |
| Override | | | | | |
| Set if missing | | | | | |
| Cache expiration duration | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><p>▼</p><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td>1 second</td></tr><tr><td>1 minute</td></tr><tr><td>1 hour</td></tr><tr><td>1 day</td></tr></table></div> | 1 second | 1 minute | 1 hour | 1 day |
| 1 second | | | | | |
| 1 minute | | | | | |
| 1 hour | | | | | |
| 1 day | | | | | |
| Query string caching behavior | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><p>▼</p><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td>Ignore query strings</td></tr><tr><td>Bypass caching for query strings</td></tr><tr><td>Cache every unique URL</td></tr></table></div> | Ignore query strings | Bypass caching for query strings | Cache every unique URL | |
| Ignore query strings | | | | | |
| Bypass caching for query strings | | | | | |
| Cache every unique URL | | | | | |

Answer:

Answer Area

| Setting | Action |
|-------------------------------|---|
| Caching behavior | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>Bypass cache</p><p>Override</p><p>Set if missing</p></div> |
| Cache expiration duration | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>1 second</p><p>1 minute</p><p>1 hour</p><p>1 day</p></div> |
| Query string caching behavior | <div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>Ignore query strings</p><p>Bypass caching for query strings</p><p>Cache every unique URL</p></div> |

Explanation:

Box 1: Override

Override: Ignore origin-provided cache duration; use the provided cache duration instead. This will not override cache-control: no-cache. Set if missing: Honor origin-provided cache-directive headers, if they exist; otherwise, use the provided cache duration.

Incorrect:

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

Box 2: 1 hour

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

Box 3: Cache every unique URL

Cache every unique URL: In this mode, each request with a unique URL, including the query string, is treated as a unique asset with its own cache. For example, the response from the origin server for a request for example.ashx?q=test1 is cached at the POP node and returned for subsequent caches with the same query string. A request for example.ashx?q=test2 is cached as a separate asset with its own time-to-live setting.

Incorrect Answers:

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

Ignore query strings: Default mode. In this mode, the CDN point-of-presence (POP) node passes

the query strings from the requestor to the origin server on the first request and caches the asset. All subsequent requests for the asset that are served from the POP ignore the query strings until the cached asset expires.

Reference:

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

QUESTION 63

Drag and Drop Question

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

Spikes in traffic have caused increases in page load times.

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

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

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Actions**Answer Area**

Configure the web app to the Premium App Service tier.



Configure the web app to the Standard App Service tier.



Enable autoscaling on the web-app.



Add a Scale rule.



Switch to an Azure App Services consumption plan.

Configure a Scale condition.

Answer:

Actions

Configure the web app to the Premium App Service tier.

Answer Area

Configure the web app to the Standard App Service tier.

Enable autoscaling on the web-app.



Add a Scale rule.



Configure a Scale condition.

Switch to an Azure App Services consumption plan.

Explanation:

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

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

Step 2: Enable autoscaling on the web app

First enable autoscale

Step 3: Add a scale rule

Step 4: Add a Scale condition

Reference:

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

QUESTION 64

Drag and Drop Question

You manage several existing Logic Apps.

You need to change definitions, add new logic, and optimize these apps on a regular basis.

What should you use? To answer, drag the appropriate tools to the correct functionalities. Each tool 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.

Answer Area

| Tools | Functionality | Tool |
|-----------------------------|----------------------------|------|
| Logic Apps Designer | Edit B2B workflows | |
| Code View Editor | Edit definitions in JSON | |
| Enterprise Integration Pack | Visually and functionality | |

Answer:**Answer Area**

| Tools | Functionality | Tool |
|-------|----------------------------|-----------------------------|
| | Edit B2B workflows | Enterprise Integration Pack |
| | Edit definitions in JSON | Code View Editor |
| | Visually and functionality | Logic Apps Designer |

Explanation:

Box 1: Enterprise Integration Pack

For business-to-business (B2B) solutions and seamless communication between organizations, you can build automated scalable enterprise integration workflows by using the Enterprise Integration Pack (EIP) with Azure Logic Apps.

Box 2: Code View Editor

Edit JSON - Azure portal

1. Sign in to the Azure portal.

2. From the left menu, choose All services. In the search box, find "logic apps", and then from the results, select your logic app.

3. On your logic app's menu, under Development Tools, select Logic App Code View.

4. The Code View editor opens and shows your logic app definition in JSON format.

Box 3: Logic Apps Designer

Reference:

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-enterprise-integration-overview><https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-author-definitions>

QUESTION 65

Hotspot Question

You are developing an application that uses Azure Storage Queues.

You have the following code:

```
CloudStorageAccount storageAccount = CloudStorageAccount.Parse  
    (CloudConfigurationManager.GetSetting("StorageConnectionString"))  
CloudQueueClient queueClient = storageAccount.CreateCloudQueueCli  
  
CloudQueue queue = queueClient.GetQueueReference("appqueue") ;  
await queue.CreateIfNotExistsAsync() ;  
  
CloudQueueMessage peekedMessage = await queue.PeekMessageAsync()  
if (peekedMessage != null)  
{  
    Console.WriteLine("The peeked message is: {0}", peekedMessage.  
}  
CloudQueueMessage message = await queue.GetMessageAsync() ;
```

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

NOTE: Each correct selection is worth one point.

Answer Area

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| The code configures the lock duration for the queue. | <input type="radio"/> | <input type="radio"/> |
| The last message read remains in the queue after the code runs. | <input type="radio"/> | <input type="radio"/> |
| The storage queue remains in the storage account after the code runs. | <input type="radio"/> | <input type="radio"/> |

Answer:

Answer Area

| Statement | Yes | No |
|---|----------------------------------|----------------------------------|
| The code configures the lock duration for the queue. | <input type="radio"/> | <input checked="" type="radio"/> |
| The last message read remains in the queue after the code runs. | <input checked="" type="radio"/> | <input type="radio"/> |
| The storage queue remains in the storage account after the code runs. | <input checked="" type="radio"/> | <input type="radio"/> |

Explanation:

Box 1: No

The QueueDescription.LockDuration property gets or sets the duration of a peek lock; that is, the amount of time that the message is locked for other receivers.

The maximum value for LockDuration is 5 minutes; the default value is 1 minute.

Box 2: Yes

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

Box 3: Yes

Reference:

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

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

QUESTION 66

Hotspot Question

You are working for Contoso, Ltd.

You define an API Policy object by using the following XML markup:

```
<set-variable name="bodySize" value="@({context.Request.Headers["Content-Length"] [0]})"/>
<choose>
  <when condition="@(int.Parse(context.Variables.GetValueOrDefault<string> ("bodySize"))<512000)">
  </when>
  <otherwise>
    <rewrite-uri template="/put"/>
    <set-backend-service base-url="http://contoso.com/api/9.1/" />
  </otherwise>
</choose>
```

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

NOTE: Each correct selection is worth one point.

Answer Area

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| The XML segment belongs in the <inbound> section of the policy. | <input type="radio"/> | <input type="radio"/> |
| If the body size is >256k, an error will occur. | <input type="radio"/> | <input type="radio"/> |
| If the request is http://contoso.com/api/9.2/, the policy will retain the higher version. | <input type="radio"/> | <input type="radio"/> |

Answer:**Answer Area**

| Statement | Yes | No |
|---|----------------------------------|----------------------------------|
| The XML segment belongs in the <inbound> section of the policy. | <input checked="" type="radio"/> | <input type="radio"/> |
| If the body size is >256k, an error will occur. | <input type="radio"/> | <input checked="" type="radio"/> |
| If the request is http://contoso.com/api/9.2/, the policy will retain the higher version. | <input type="radio"/> | <input checked="" type="radio"/> |

Explanation:

Box 1: Yes

Use the set-backend-service policy to redirect an incoming request to a different backend than the one specified in the API settings for that operation. Syntax:
<set-backend-service base-url="base URL of the backend service" />

Box 2: No

The condition is on 512k, not on 256k.

Box 3: No

The set-backend-service policy changes the backend service base URL of the incoming request to the one specified in the policy.

Reference:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-transformation-policies>

QUESTION 67**Case Study 3 - City Power & Light****Background**

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

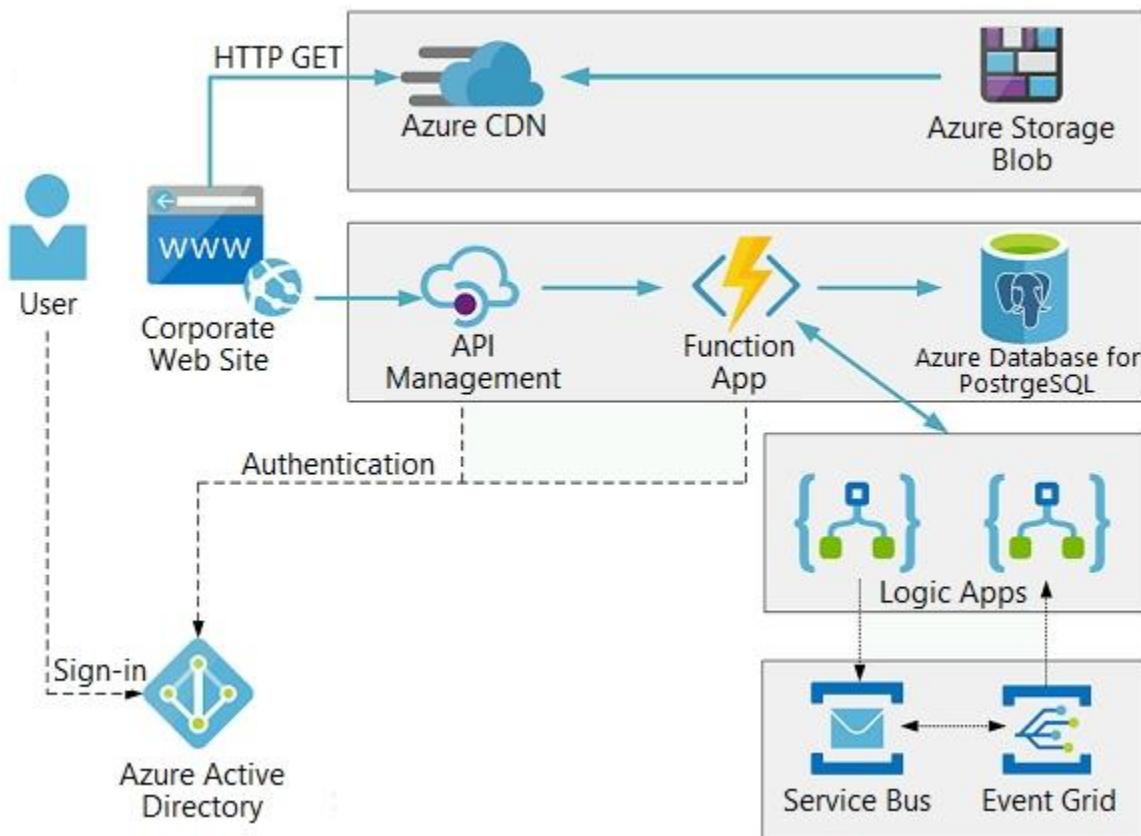
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

1. The user selects **Sign in** in the website.
2. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
3. The user signs in.
4. Azure AD redirects the user's session back to the web application. The URL includes an access token.
5. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.

6. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

FunctionAppLogs

```
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

You need to correct the RequestUserApproval Function app error.

What should you do?

- A. Update line RA13 to use the `async` keyword and return an `HttpRequest` object value.
- B. Configure the Function app to use an App Service hosting plan. Enable the Always On setting of the hosting plan.
- C. Update the function to be stateful by using Durable Functions to process the request payload.
- D. Update the `functionTimeout` property of the `host.json` project file to 15 minutes.

Answer: C

Explanation:

Async operation tracking

The HTTP response mentioned previously is designed to help implement long-running HTTP async APIs with Durable Functions. This pattern is sometimes referred to as the polling consumer pattern.

Both the client and server implementations of this pattern are built into the Durable Functions

HTTP APIs.

Function app

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

'Timeout value of 00:10:00 exceeded by function: RequestUserApproval' The same error message displays when you test the function in an Azure development environment when you run the following Kusto query:

FunctionAppLogs

| where FunctionName == "RequestUserApproval"

References:

<https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-http-features>

QUESTION 68

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

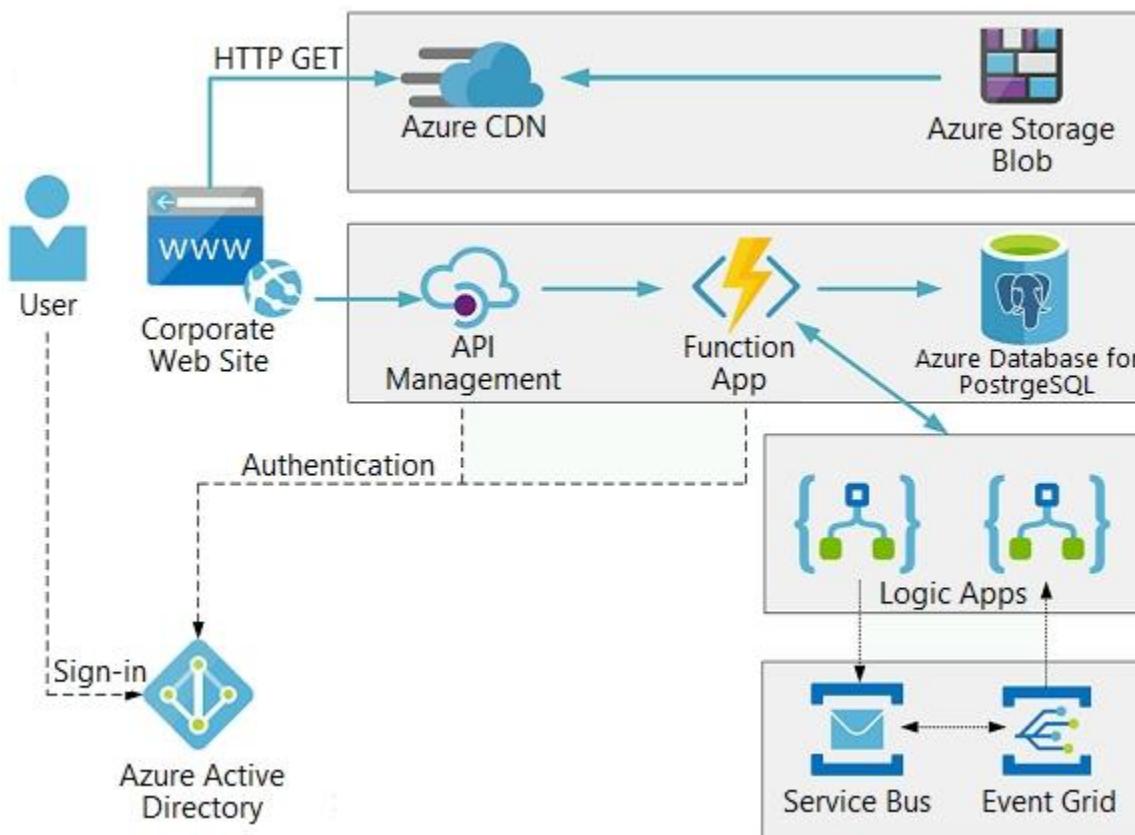
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

7. The user selects **Sign in** in the website.
8. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
9. The user signs in.
10. Azure AD redirects the user's session back to the web application. The URL includes an access token.
11. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
12. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpandlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

You need to authenticate the user to the corporate website as indicated by the architectural diagram.

Which two values should you use? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. ID token signature
- B. ID token claims
- C. HTTP response code
- D. Azure AD endpoint URI
- E. Azure AD tenant ID

Answer: AD**Explanation:**

A: Claims in access tokens

JWTs (JSON Web Tokens) are split into three pieces:

Header - Provides information about how to validate the token including information about the type of token and how it was signed.

Payload - Contains all of the important data about the user or app that is attempting to call your service.

Signature - Is the raw material used to validate the token.

E: Your client can get an access token from either the v1.0 endpoint or the v2.0 endpoint using a variety of protocols.

Scenario: User authentication (see step 5 below)

The following steps detail the user authentication process:

1. The user selects Sign in in the website.
2. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
3. The user signs in.

4. Azure AD redirects the user's session back to the web application. The URL includes an access token.
 5. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
 6. The back-end API validates the access token.
- Reference:
<https://docs.microsoft.com/en-us/azure/api-management/api-management-access-restriction-policies>

QUESTION 69**Case Study 3 - City Power & Light****Background**

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment**Architecture overview**

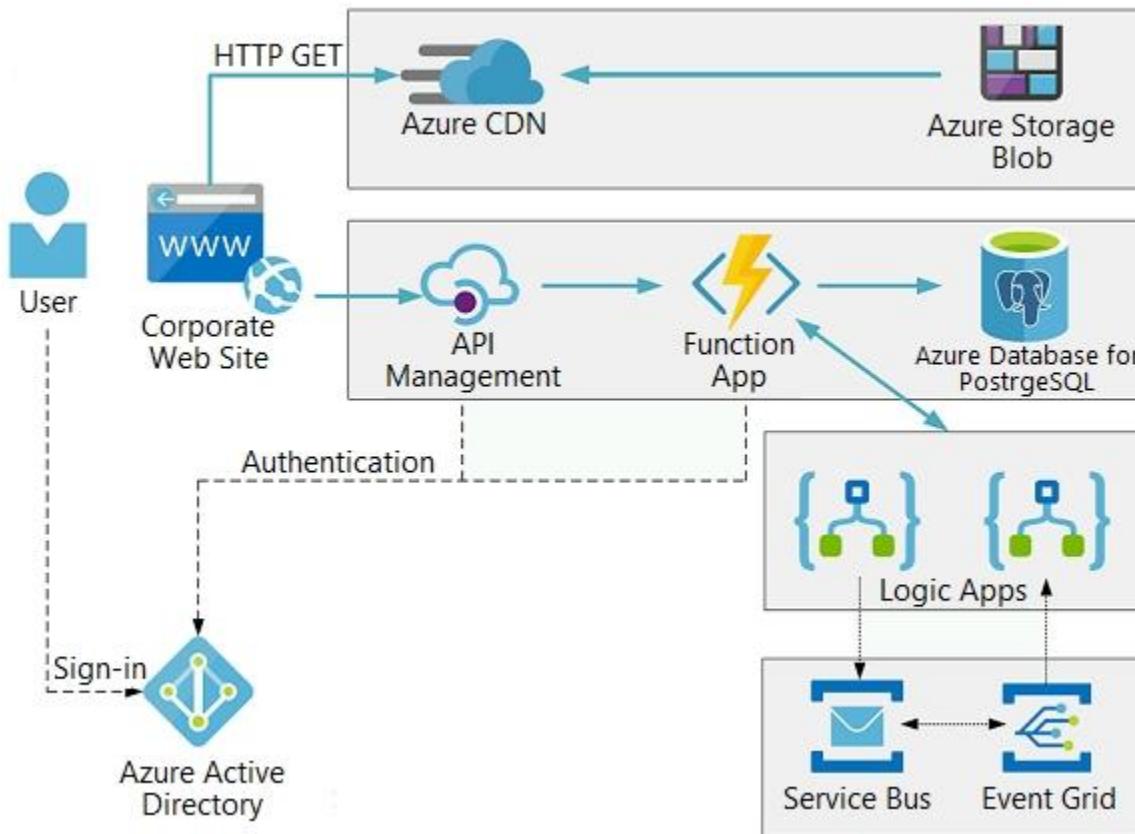
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

13. The user selects **Sign in** in the website.
14. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
15. The user signs in.
16. Azure AD redirects the user's session back to the web application. The URL includes an access token.
17. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
18. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

You need to investigate the Azure Function app error message in the development environment.

What should you do?

- A. Connect Live Metrics Stream from Application Insights to the Azure Function app and filter the metrics.
- B. Create a new Azure Log Analytics workspace and instrument the Azure Function app with Application Insights.
- C. Update the Azure Function app with extension methods from Microsoft.Extensions.Logging to log events by using the log instance.
- D. Add a new diagnostic setting to the Azure Function app to send logs to Log Analytics.

Answer: A

Explanation:

Azure Functions offers built-in integration with Azure Application Insights to monitor functions.

The following areas of Application Insights can be helpful when evaluating the behavior, performance, and errors in your functions:

Live Metrics: View metrics data as it's created in near real-time.

Failures

Performance

Metrics

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-monitoring>

QUESTION 70

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

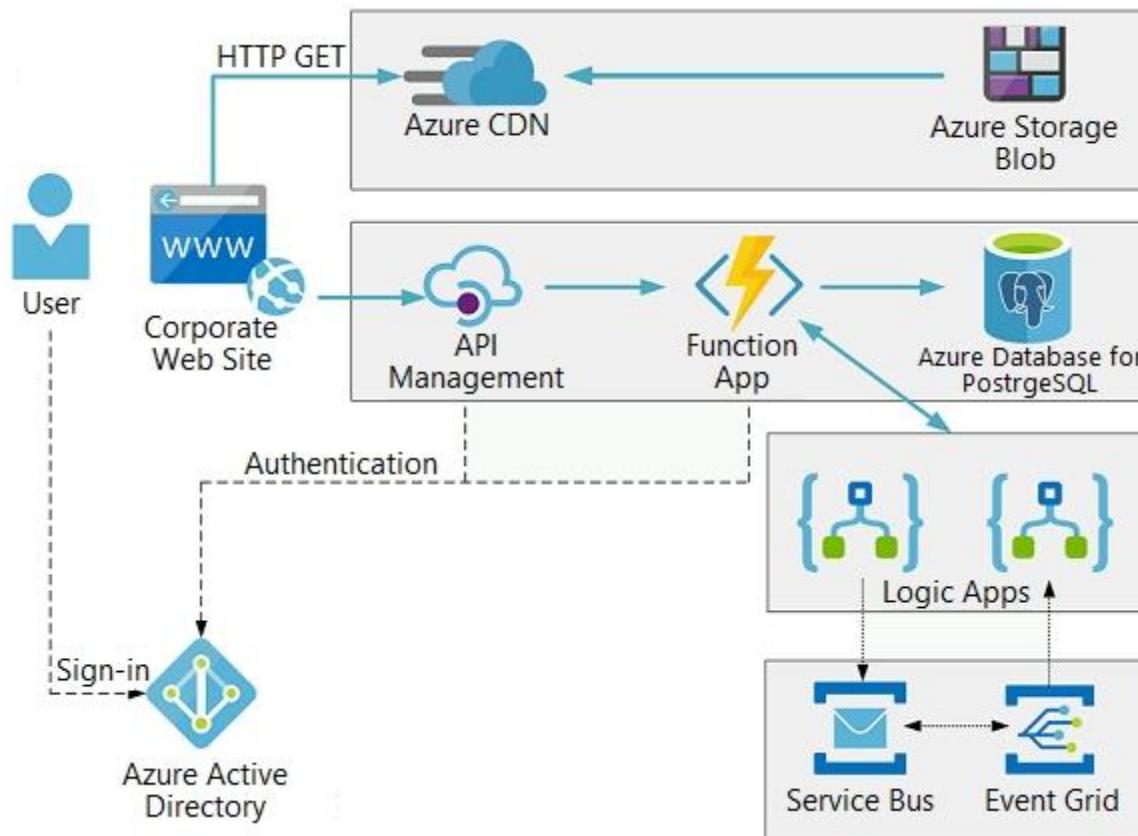
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

19. The user selects **Sign in** in the website.
20. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
21. The user signs in.

22. Azure AD redirects the user's session back to the web application. The URL includes an access token.
23. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
24. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to configure the Account Kind, Replication, and Storage tier options for the corporate website's Azure Storage account.

How should you complete the configuration? To answer, select the appropriate options in the dialog box in the answer area.

NOTE: Each correct selection is worth one point.

Create storage account

X

[Basics](#) [Advanced](#) [Tags](#) [Review + create](#)

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

| | |
|------------------|--------------------------|
| * Subscription | Visual Studio Enterprise |
| | ▼ |
| * Resource group | (New) cplcorporatesite |
| | ▼ |
| | Create new |

INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

| | | | | | | | |
|--|--|---------------------------------|------------------------------|-----------------------------|--|-----------------------------------|--|
| * Storage account name ⓘ | corporatewebsitecontent | | | | | | |
| | ✓ | | | | | | |
| * Location | (US) East US | | | | | | |
| | ▼ | | | | | | |
| Performance ⓘ | <input checked="" type="radio"/> Standard <input type="radio"/> Premium | | | | | | |
| Account kind ⓘ | <table><tr><td>StorageV2 (general purpose v2)</td></tr><tr><td>Storage (general purpose v1)</td></tr><tr><td>BlobStorage</td></tr></table> | StorageV2 (general purpose v2) | Storage (general purpose v1) | BlobStorage | | | |
| StorageV2 (general purpose v2) | | | | | | | |
| Storage (general purpose v1) | | | | | | | |
| BlobStorage | | | | | | | |
| | ▼ | | | | | | |
| Replication ⓘ | <table><tr><td>Locally-redundant storage (LRS)</td></tr><tr><td>Zone-redundant storage (ZRS)</td></tr><tr><td>Geo-redundant storage (GRS)</td></tr><tr><td>Read-access geo-redundant storage (RA-GRS)</td></tr><tr><td>Geo-zone-redundant storage (GZRS)</td></tr><tr><td>Read-access geo-zone-redundant storage (RA-GZRS)</td></tr></table> | Locally-redundant storage (LRS) | Zone-redundant storage (ZRS) | Geo-redundant storage (GRS) | Read-access geo-redundant storage (RA-GRS) | Geo-zone-redundant storage (GZRS) | Read-access geo-zone-redundant storage (RA-GZRS) |
| Locally-redundant storage (LRS) | | | | | | | |
| Zone-redundant storage (ZRS) | | | | | | | |
| Geo-redundant storage (GRS) | | | | | | | |
| Read-access geo-redundant storage (RA-GRS) | | | | | | | |
| Geo-zone-redundant storage (GZRS) | | | | | | | |
| Read-access geo-zone-redundant storage (RA-GZRS) | | | | | | | |
| | ▼ | | | | | | |
| Access tier (default) ⓘ | <input type="radio"/> Cool <input type="radio"/> Hot | | | | | | |

Answer:

Create storage account



Basics Advanced Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

| | |
|------------------|----------------------------------|
| * Subscription | Visual Studio Enterprise |
| | <input type="button" value="▼"/> |
| * Resource group | (New) cplcorporatesite |
| | <input type="button" value="▼"/> |
| | Create new |

INSTANCE DETAILS

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

| | | |
|------------------------|---|-------------------------------------|
| * Storage account name | corporatewebsitecontent | <input checked="" type="checkbox"/> |
| * Location | (US) East US | <input type="button" value="▼"/> |
| Performance | <input checked="" type="radio"/> Standard <input type="radio"/> Premium | |
| Account kind | <input type="button" value="▼"/> | |
| | StorageV2 (general purpose v2) | |
| | Storage (general purpose v1) | |
| | BlobStorage | |
| Replication | <input type="button" value="▼"/> | |
| | Locally-redundant storage (LRS) | |
| | Zone-redundant storage (ZRS) | |
| | <input checked="" type="radio"/> Geo-redundant storage (GRS) | |
| | Read-access geo-redundant storage (RA-GRS) | |
| | Geo-zone-redundant storage (GZRS) | |
| | Read-access geo-zone-redundant storage (RA-GZRS) | |
| Access tier (default) | <input type="radio"/> Cool | <input type="radio"/> Hot |

Explanation:

Account Kind: StorageV2 (general-purpose v2)

Scenario: Azure Storage blob will be used (refer to the exhibit). Data storage costs must be minimized.

General-purpose v2 accounts: Basic storage account type for blobs, files, queues, and tables. Recommended for most scenarios using Azure Storage.

Incorrect Answers:

BlockBlobStorage accounts: Storage accounts with premium performance characteristics for block blobs and append blobs. Recommended for scenarios with high transaction rates, or

scenarios that use smaller objects or require consistently low storage latency.

General-purpose v1 accounts: Legacy account type for blobs, files, queues, and tables. Use general-purpose v2 accounts instead when possible.

Replication: Geo-redundant Storage

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

Geo-redundant storage (GRS) copies your data synchronously three times within a single physical location in the primary region using LRS. It then copies your data asynchronously to a single physical location in the secondary region.

Incorrect Answers:

Geo-zone-redundant storage (GZRS), but it would be more costly.

Storage tier: Cool

Data storage costs must be minimized.

Note: Azure storage offers different access tiers, which allow you to store blob object data in the most cost-effective manner. The available access tiers include:

Hot - Optimized for storing data that is accessed frequently. Cool - Optimized for storing data that is infrequently accessed and stored for at least 30 days.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview>

<https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy>

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers?tabs=azure-portal>

QUESTION 71

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

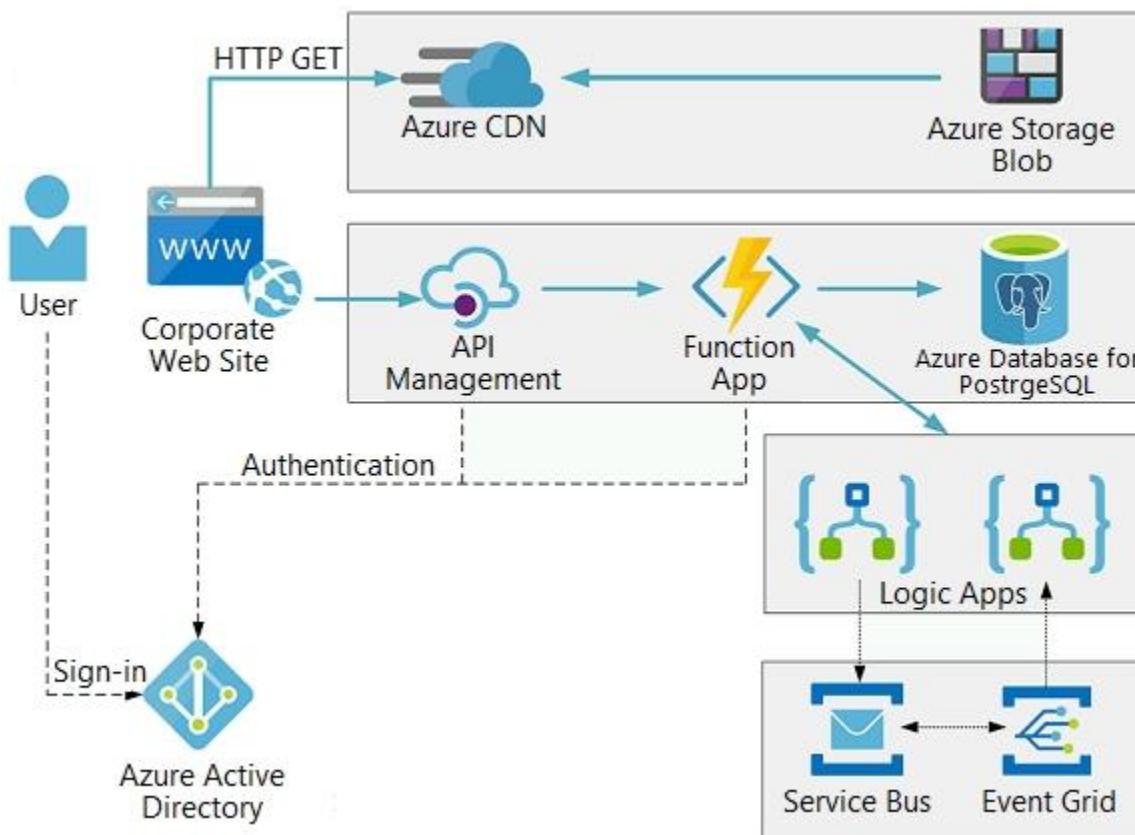
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

25. The user selects **Sign in** in the website.
26. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
27. The user signs in.
28. Azure AD redirects the user's session back to the web application. The URL includes an access token.
29. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
30. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpandlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to retrieve the database connection string.

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

NOTE: Each correct selection is worth one point.

Answer Area

REST API Endpoint:

| | | | |
|----------------------------------|---|----------------------------------|-----|
| https:// | ▼ | .vault.azure.net/secrets/ | ▼ / |
| cpandlkeyvault | | cpandlkeyvault | |
| PostgreSQLConn | | PostgreSQLConn | |
| 80df3e46ffcd4f1cb187f79905e9a1e8 | | 80df3e46ffcd4f1cb187f79905e9a1e8 | |

Variable type to access Azure Key Vault secret values:

| |
|-------------|
| ▼ |
| Environment |
| Session |
| ViewState |
| QueryString |

Answer:

Answer Area

REST API Endpoint:

| | | | | |
|----------|----------------------------------|---------------------------|----------------------------------|---|
| https:// | cpandlkeyvault | .vault.azure.net/secrets/ | cpandlkeyvault | / |
| | PostgreSQLConn | | PostgreSQLConn | |
| | 80df3e46ffcd4f1cb187f79905e9a1e8 | | 80df3e46ffcd4f1cb187f79905e9a1e8 | |

Variable type to access Azure Key Vault secret values:

| |
|-------------|
| Environment |
| Session |
| ViewState |
| QueryString |

Explanation:

Azure database connection string retrieve REST API vault.azure.net/secrets/

Box 1: cpandlkeyvault

We specify the key vault, cpandlkeyvault.

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

Azure Key Vault name: cpandlkeyvault

Secret name: PostgreSQLConn

Id: 80df3e46ffcd4f1cb187f79905e9a1e8

Box 2: PostgreSQLConn

We specify the secret, PostgreSQLConn

Example, sample request:

`https://myvault.vault.azure.net/secrets/mysecretname/4387e9f3d6e14c459867679a90fd0f79?api-version=7.1`

Box 3: Querystring

Reference:

<https://docs.microsoft.com/en-us/rest/api/keyvault/getsecret/getsecret>**QUESTION 72****Case Study 3 - City Power & Light****Background**

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

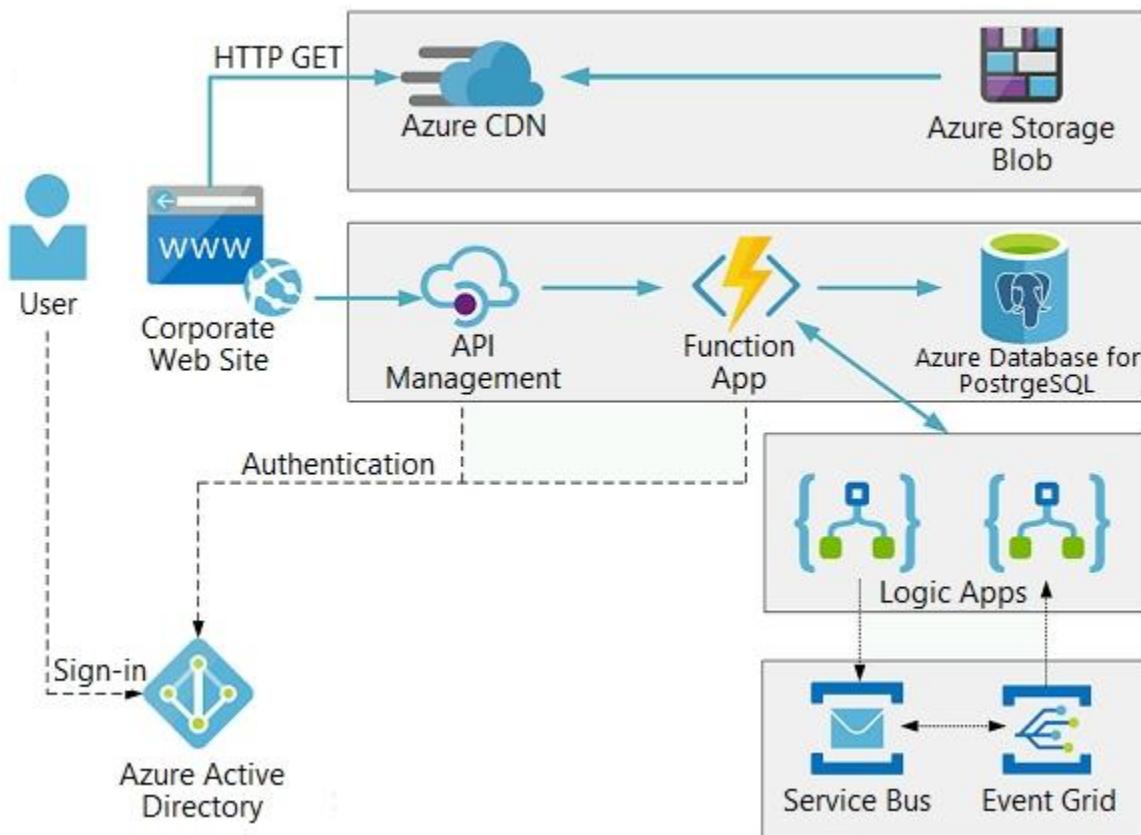
Current environment**Architecture overview**The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

31. The user selects **Sign in** in the website.
32. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
33. The user signs in.
34. Azure AD redirects the user's session back to the web application. The URL includes an access token.
35. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
36. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues**Corporate website**

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code**Corporate website**

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Drag and Drop Question

You need to correct the corporate website error.

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

| Actions | Answer Area |
|---|-------------|
| Upload the certificate to Azure Key Vault. | |
| Update line SC05 of Security.cs to include error handling and then redeploy the code. | ▶ |
| Update line SC03 of Security.cs to include a using statement and then re-deploy the code. | ◀ |
| Add the certificate thumbprint to the WEBSITE_LOAD_CERTIFICATES app setting. | ◀ |
| Upload the certificate to source control. | |
| Import the certificate to Azure App Service. | |
| Generate a certificate. | |

Answer:

| Actions | Answer Area |
|---|-------------|
| Generate a certificate. | |
| Upload the certificate to Azure Key Vault. | |
| Update line SC03 of Security.cs to include a using statement and then re-deploy the code. | ▶ |
| Add the certificate thumbprint to the WEBSITE_LOAD_CERTIFICATES app setting. | ◀ |
| Import the certificate to Azure App Service. | ◀ |
| Update line SC05 of Security.cs to include error handling and then redeploy the code. | ▼ |
| Upload the certificate to source control. | |

Explanation:

Scenario: Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Step 1: Generate a certificate

Step 2: Upload the certificate to Azure Key Vault

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

Step 3: Import the certificate to Azure App Service

Step 4: Update line SCO5 of Security.cs to include error handling and then redeploy the code

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/configure-ssl-certificate>

QUESTION 73

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

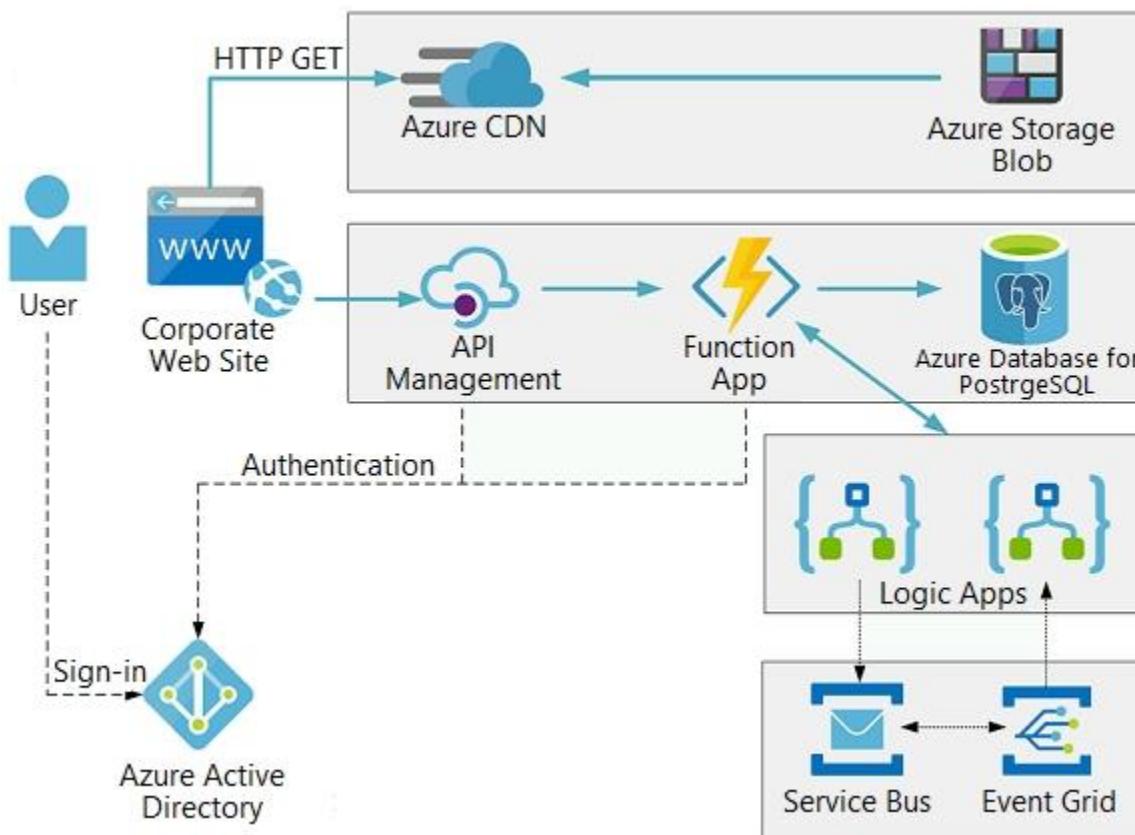
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

37. The user selects **Sign in** in the website.
38. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
39. The user signs in.
40. Azure AD redirects the user's session back to the web application. The URL includes an access token.
41. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
42. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to configure API Management for authentication.

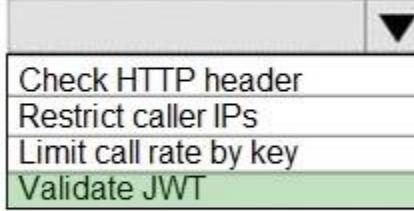
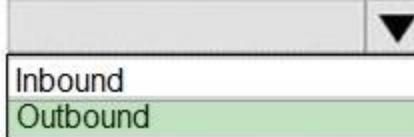
Which policy values should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|----------------|---|
| Policy | <div style="border: 1px solid black; padding: 5px;"><input type="checkbox"/> Check HTTP header <input type="checkbox"/> Restrict caller IPs <input type="checkbox"/> Limit call rate by key <input type="checkbox"/> Validate JWT</div> |
| Policy section | <div style="border: 1px solid black; padding: 5px;"><input type="checkbox"/> Inbound <input type="checkbox"/> Outbound</div> |

Answer:**Answer Area**

| Setting | Value |
|----------------|---|
| Policy |  <ul style="list-style-type: none">Check HTTP headerRestrict caller IPsLimit call rate by keyValidate JWT |
| Policy section |  <ul style="list-style-type: none">InboundOutbound |

Explanation:

Box 1: Validate JWT

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

Scenario: User authentication (see step 5 below)

The following steps detail the user authentication process:

1. The user selects Sign in in the website.
2. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
3. The user signs in.
4. Azure AD redirects the user's session back to the web application. The URL includes an access token.
5. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
6. The back-end API validates the access token.

Incorrect Answers:

Limit call rate by key - Prevents API usage spikes by limiting call rate, on a per key basis.

Restrict caller IPs - Filters (allows/denies) calls from specific IP addresses and/or address ranges.

Check HTTP header - Enforces existence and/or value of a HTTP Header.

Box 2: Outbound

Reference:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-access-restriction-policies>

QUESTION 74**Case Study 3 - City Power & Light**

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment**Architecture overview**

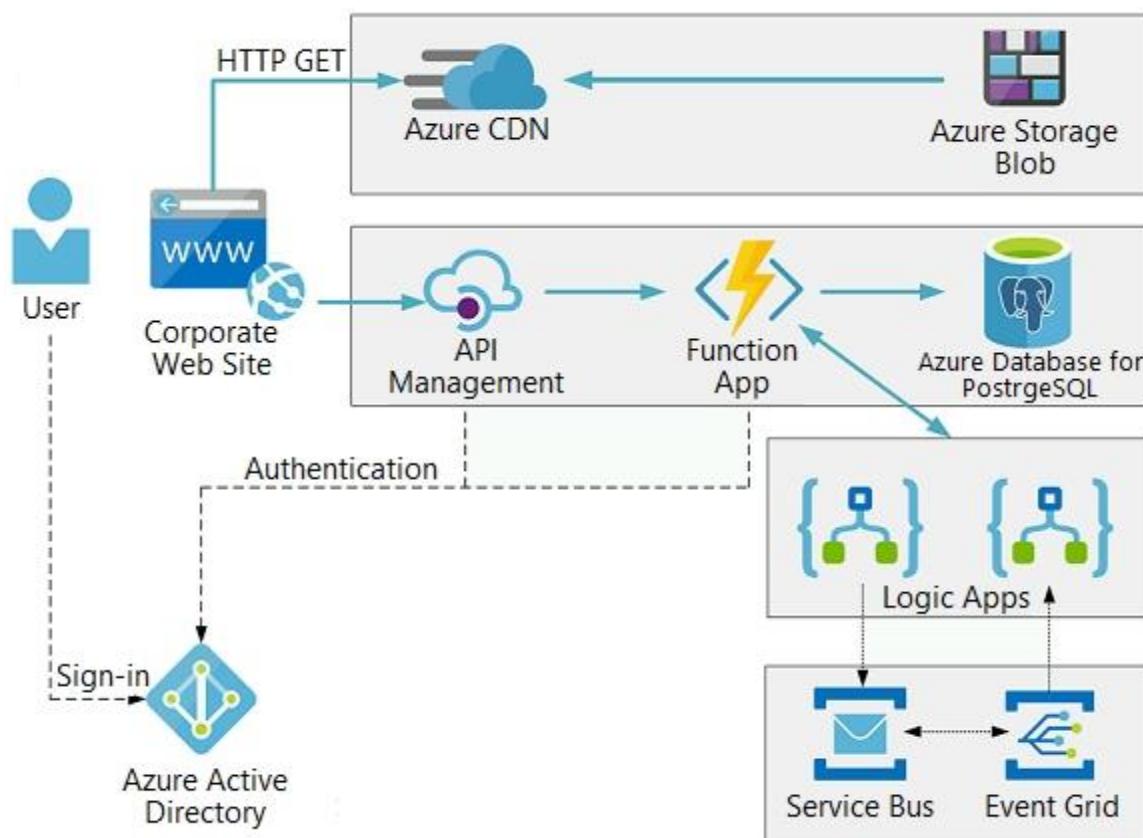
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.

**User authentication**

The following steps detail the user authentication process:

43. The user selects **Sign in** in the website.
44. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.

45. The user signs in.
46. Azure AD redirects the user's session back to the web application. The URL includes an access token.
47. The web application calls an API and includes the access token in the authentication header. The application ID is sent as the audience ('aud') claim in the access token.
48. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

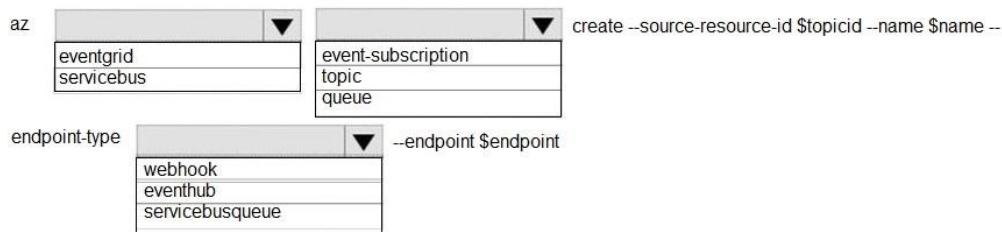
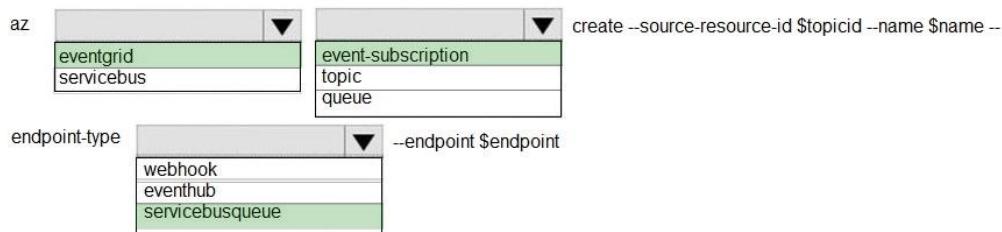
```
RA01 public static class RequestUserApproval  
RA02 {  
RA03 [FunctionName("RequestUserApproval")]  
RA04 public static async Task<IActionResult> Run(  
RA05 [HttpTrigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,  
ILogger log)  
RA06 {  
RA07     log.LogInformation("RequestUserApproval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

Hotspot Question

You need to configure the integration for Azure Service Bus and Azure Event Grid.

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

NOTE: Each correct selection is worth one point.

Answer Area

Answer:
Answer Area

Explanation:

Box 1: eventgrid

To create event subscription use: az eventgrid event-subscription create

Box 2: event-subscription

Box 3: servicebusqueue

Scenario: Azure Service Bus and Azure Event Grid

Azure Event Grid must use Azure Service Bus for queue-based load leveling. Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering. Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Reference:

https://docs.microsoft.com/en-us/cli/azure/eventgrid/event-subscription?view=azure-cli-latest#az_eventgrid_event_subscription_create

QUESTION 75

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?

- 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 a virtual machine.
Configure a Scale Set to increase the virtual machine instance count when the CPU load is high.
- D. 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.

Answer: D

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 76

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 develop an HTTP triggered Azure Function app to process Azure Storage blob data. The app is triggered using an output binding on the blob.

The app continues to time out after four minutes. The app must process the blob data.

You need to ensure the app does not time out and processes the blob data.

Solution: Use the Durable Function async pattern to process the blob data.

Does the solution meet the goal?

- A. Yes
B. No

Answer: B

Explanation:

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

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

Whenever possible, refactor large functions into smaller function sets that work together and return responses fast. For example, a webhook or HTTP trigger function might require an

acknowledgment response within a certain time limit; it's common for webhooks to require an immediate response. You can pass the HTTP trigger payload into a queue to be processed by a queue trigger function. This approach lets you defer the actual work and return an immediate response.

Reference:

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

QUESTION 77

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 develop an HTTP triggered Azure Function app to process Azure Storage blob data. The app is triggered using an output binding on the blob.

The app continues to time out after four minutes. The app must process the blob data.

You need to ensure the app does not time out and processes the blob data.

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

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

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

Reference:

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

QUESTION 78

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 develop an HTTP triggered Azure Function app to process Azure Storage blob data. The app is triggered using an output binding on the blob.

The app continues to time out after four minutes. The app must process the blob data.

You need to ensure the app does not time out and processes the blob data.

Solution: Configure the app to use an App Service hosting plan and enable the Always On setting.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

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

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

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

Reference:

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

QUESTION 79

You are developing an Azure Cosmos DB solution by using the Azure Cosmos DB SQL API. The data includes millions of documents. Each document may contain hundreds of properties.

The properties of the documents do not contain distinct values for partitioning. Azure Cosmos DB must scale individual containers in the database to meet the performance needs of the application by spreading the workload evenly across all partitions over time.

You need to select a partition key.

Which two partition keys can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. a single property value that does not appear frequently in the documents
- B. a value containing the collection name
- C. a single property value that appears frequently in the documents
- D. a concatenation of multiple property values with a random suffix appended
- E. a hash suffix appended to a property value

Answer: DE

Explanation:

You can form a partition key by concatenating multiple property values into a single artificial partitionKey property. These keys are referred to as synthetic keys.

Another possible strategy to distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.

Note: It's the best practice to have a partition key with many distinct values, such as hundreds or thousands. The goal is to distribute your data and workload evenly across the items associated with these partition key values. If such a property doesn't exist in your data, you can construct a synthetic partition key.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/synthetic-partition-keys>

QUESTION 80

You are building a website that uses Azure Blob storage for data storage. You configure Azure Blob storage lifecycle to move all blobs to the archive tier after 30 days.

Customers have requested a service-level agreement (SLA) for viewing data older than 30 days.

You need to document the minimum SLA for data recovery.

Which SLA should you use?

- A. at least two days
- B. between one and 15 hours
- C. at least one day
- D. between zero and 60 minutes

Answer: B

Explanation:

The archive access tier has the lowest storage cost. But it has higher data retrieval costs compared to the hot and cool tiers. Data in the archive tier can take several hours to retrieve depending on the priority of the rehydration. For small objects, a high priority rehydrate may retrieve the object from archive in under 1 hour.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers?tabs=azure-portal>

QUESTION 81

You develop an app that allows users to upload photos and videos to Azure storage. The app uses a storage REST API call to upload the media to a blob storage account named Account1. You have blob storage containers named Container1 and Container2.

Uploading of videos occurs on an irregular basis.

You need to copy specific blobs from Container1 to Container2 when a new video is uploaded.

What should you do?

- A. Copy blobs to Container2 by using the Put Blob operation of the Blob Service REST API
- B. Create an Event Grid topic that uses the Start-AzureStorageBlobCopy cmdlet
- C. Use AzCopy with the Snapshot switch to copy blobs to Container2
- D. Download the blob to a virtual machine and then upload the blob to Container2

Answer: B

Explanation:

The Start-AzureStorageBlobCopy cmdlet starts to copy a blob.

Example 1: Copy a named blob

```
C:\PS>Start-AzureStorageBlobCopy -SrcBlob "ContosoPlanning2015" -DestContainer  
"ContosoArchives" - SrcContainer "ContosoUploads"
```

This command starts the copy operation of the blob named ContosoPlanning2015 from the container named ContosoUploads to the container named ContosoArchives.

Reference:

<https://docs.microsoft.com/en-us/powershell/module/azure.storage/start-azurerestorageblobcopy?view=azurermps-6.13.0>

QUESTION 82

You are developing an ASP.NET Core website that uses Azure FrontDoor. The website is used to build custom weather data sets for researchers. Data sets are downloaded by users as Comma Separated Value (CSV) files. The data is refreshed every 10 hours.

Specific files must be purged from the FrontDoor cache based upon Response Header values.

You need to purge individual assets from the Front Door cache.

Which type of cache purge should you use?

- A. single path
- B. wildcard
- C. root domain

Answer: A

Explanation:

These formats are supported in the lists of paths to purge:

Single path purge: Purge individual assets by specifying the full path of the asset (without the protocol and domain), with the file extension, for example, /pictures/strasbourg.png; Wildcard purge: Asterisk (*) may be used as a wildcard. Purge all folders, subfolders, and files under an endpoint with /* in the path or purge all subfolders and files under a specific folder by specifying the folder followed by /*, for example, /pictures/*.

Root domain purge: Purge the root of the endpoint with "/" in the path.

Reference:

<https://docs.microsoft.com/en-us/azure/frontdoor/front-door-caching>

QUESTION 83

You are developing a Java application that uses Cassandra to store key and value data. You plan to use a new Azure Cosmos DB resource and the Cassandra API in the application. You create an Azure Active Directory (Azure AD) group named Cosmos DB Creators to enable provisioning of Azure Cosmos accounts, databases, and containers.

The Azure AD group must not be able to access the keys that are required to access the data.

You need to restrict access to the Azure AD group.

Which role-based access control should you use?

- A. DocumentDB Accounts Contributor
- B. Cosmos Backup Operator
- C. Cosmos DB Operator
- D. Cosmos DB Account Reader

Answer: C**Explanation:**

Azure Cosmos DB now provides a new RBAC role, Cosmos DB Operator. This new role lets you provision Azure Cosmos accounts, databases, and containers, but can't access the keys that are required to access the data. This role is intended for use in scenarios where the ability to grant access to Azure Active Directory service principals to manage deployment operations for Cosmos DB is needed, including the account, database, and containers.

Reference:

<https://azure.microsoft.com/en-us/updates/azure-cosmos-db-operator-role-for-role-based-access-control-rbac-is-now-available/>

QUESTION 84

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 are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.

You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.

You need to configure authorization.

Solution: Configure the Azure Web App for the website to allow only authenticated requests and require Azure AD log on.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B**Explanation:**

Instead in the Azure AD application's manifest, set value of the groupMembershipClaims option to All.

Reference:

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

QUESTION 85

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 are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.

You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.

You need to configure authorization.

Solution: Create a new Azure AD application. In the application's manifest, set value of the groupMembershipClaims option to All.

In the website, use the value of the groups claim from the JWT for the user to determine permissions.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

To configure Manifest to include Group Claims in Auth Token

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

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

3. Locate the "groupMembershipClaims" setting. Set its value to either "SecurityGroup" or "All".

To help

you decide which:

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

"All" - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member

Now your application will include group claims in your manifest and you can use this fact in your code.

Reference:

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

QUESTION 86

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 are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.

You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level.

You need to configure authorization.

Solution:

Create a new Azure AD application. In the application's manifest, define application roles that match the required permission levels for the application.

Assign the appropriate Azure AD group to each role. In the website, use the value of the roles claim from the JWT for the user to determine permissions.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B**Explanation:**

To configure Manifest to include Group Claims in Auth Token

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

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

3. Locate the "groupMembershipClaims" setting. Set its value to either "SecurityGroup" or "All".

To help you decide which:

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

"All" - groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member

Now your application will include group claims in your manifest and you can use this fact in your code.

Reference:

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

QUESTION 87

You develop and deploy an ASP.NET web app to Azure App Service. You use Application Insights telemetry to monitor the app.

You must test the app to ensure that the app is available and responsive from various points around the world and at regular intervals. If the app is not responding, you must send an alert to support staff.

You need to configure a test for the web app.

Which two test types can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. integration
- B. multi-step web
- C. URL ping
- D. unit
- E. load

Answer: BC**Explanation:**

There are three types of availability tests:

URL ping test: a simple test that you can create in the Azure portal.

Multi-step web test: A recording of a sequence of web requests, which can be played back to test more complex scenarios. Multi-step web tests are created in Visual Studio Enterprise and uploaded to the portal for execution.

Custom Track Availability Tests: If you decide to create a custom application to run availability tests, the `TrackAvailability()` method can be used to send the results to Application Insights.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/monitor-web-app-availability>

QUESTION 88

You are developing an e-commerce solution that uses a microservice architecture.

You need to design a communication backplane for communicating transactional messages between various parts of the solution. Messages must be communicated in first-in-first-out (FIFO) order.

What should you use?

- A. Azure Storage Queue
- B. Azure Event Hub
- C. Azure Service Bus
- D. Azure Event Grid

Answer: C

Explanation:

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

Your solution requires the queue to provide a guaranteed first-in-first-out (FIFO) ordered delivery.

Reference:

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

QUESTION 89

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 are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

- Queue size must not grow larger than 80 gigabytes (GB).
- Use first-in-first-out (FIFO) ordering of messages.
- Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Storage Queue from the mobile

application. Create an Azure Function App that uses an Azure Storage Queue trigger.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Create an Azure Function App that uses an Azure Service Bus Queue trigger.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

QUESTION 90

A company is developing a solution that allows smart refrigerators to send temperature information to a central location. You have an existing Service Bus.

The solution must receive and store messages until they can be processed. You create an Azure Service Bus instance by providing a name, pricing tier, subscription, resource group, and location.

You need to complete the configuration.

Which Azure CLI or PowerShell command should you run?

- A. `az group create
--name fridge-rg
--location fridge-loc`
- B. `New-AzureRmServiceBusNamespace
-ResourceGroupName fridge-rg
-NamespaceName fridge-ns
-Location fridge-loc`
- C. `New-AzureRmServiceBusQueue
-ResourceGroupName fridge-rg
-NamespaceName fridge-ns
-Name fridge-q
-EnablePartitioning $False`
- D. `Get-AzureRmServiceBusKey
-ResourceGroupName fridge-rg
-Namespace fridge-ns
-Name RootManageSharedAccessKey`

Answer: C

Explanation:

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

Note:

Steps:

Step 1: # Create a resource group

```
resourceGroupName="myResourceGroup"
```

```
az group create --name $resourceGroupName --location eastus
```

Step 2: # Create a Service Bus messaging namespace with a unique name

```
namespaceName=myNameSpace$RANDOM
```

```
az servicebus namespace create --resource-group $resourceGroupName --name  
$namespaceName --location eastus
```

Step 3: # Create a Service Bus queue

```
az servicebus queue create --resource-group $resourceGroupName --namespace-name  
$namespaceName --name BasicQueue
```

Step 4: # Get the connection string for the namespace

```
connectionString=$(az servicebus namespace authorization-rule keys list --resource-group  
$resourceGroupName --namespace-name $namespaceName --name  
RootManageSharedAccessKey --query primaryConnectionString --output tsv)
```

Reference:

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

QUESTION 91

Hotspot Question

You are developing an Azure Web App. You configure TLS mutual authentication for the web app.

You need to validate the client certificate in the web app. To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Property | Value |
|-----------------------------|---|
| Client certificate location | <p>HTTP request header Client cookie HTTP message body URL query string</p> |
| Encoding type | <p>HTML URL Unicode Base64</p> |

Answer:

Answer Area

| Property | Value |
|-----------------------------|---|
| Client certificate location | <p>HTTP request header Client cookie HTTP message body URL query string</p> |
| Encoding type | <p>HTML URL Unicode Base64</p> |

Explanation:

Accessing the client certificate from App Service.

If you are using ASP.NET and configure your app to use client certificate authentication, the certificate will be available through the `HttpRequest.ClientCertificate` property. For other application stacks, the client cert will be available in your app through a base64 encoded value in the "X-ARR-ClientCert" request header. Your application can create a certificate from this value

and then use it for authentication and authorization purposes in your application.

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/app-service-web-configure-tls-mutual-auth>

QUESTION 92

Drag and Drop Question

You are developing a Docker/Go using Azure App Service Web App for Containers. You plan to run the container in an App Service on Linux. You identify a Docker container image to use.

None of your current resource groups reside in a location that supports Linux. You must minimize the number of resource groups required.

You need to create the application and perform an initial deployment.

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

Azure CLI Commands**Answer Area**

az group create

az group update

az webapp update

az webapp create

az appservice plan create



Answer:

Azure CLI Commands**Answer Area**

az group update

az webapp update

az group create

az appservice plan create

az webapp create

**Explanation:**

You can host native Linux applications in the cloud by using Azure Web Apps. To create a Web App for Containers, you must run Azure CLI commands that create a group, then a service plan, and finally the web app itself.

Step 1: az group create

In the Cloud Shell, create a resource group with the az group create command.

Step 2: az appservice plan create

In the Cloud Shell, create an App Service plan in the resource group with the az appservice plan create command.

Step 3: az webapp create

In the Cloud Shell, create a web app in the myAppServicePlan App Service plan with the az webapp create command. Don't forget to replace with a unique app name, and <docker-ID> with your Docker ID.

Reference:

<https://docs.microsoft.com/mt-mt/azure/app-service/containers/quickstart-docker-go?view=sql-server-ver15>

QUESTION 93

Drag and Drop Question

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 commands from the list of commands to the answer area and arrange them in the correct order.

Azure CLI Commands**Answer Area**

```
az webapp config container set  
--docker-custom-image-name  
$dockerHubContainerPath  
--name $appName  
--resource-group  
fourthCoffeePublicWebResourceGroup
```



```
az webapp config hostname add  
--webapp-name $appName  
--resource-group  
fourthCoffeePublicWebResourceGroup \  
--hostname $fqdn
```

```
az webapp create  
--name $appName  
--plan AppServiceLinuxDockerPlan  
--resource-group  
fourthCoffeePublicWebResourceGroup
```

```
#/bin/bash  
appName="FourthCoffeePublicWeb$random"  
location="WestUS"  
dockerHubContainerPath="FourthCoffee/publicweb:v1"  
fqdn="http://www.fourthcoffee.com">www.fourthcoffee.com
```

Answer:

Azure CLI Commands**Answer Area**

```
#!/bin/bash  
appName="FourthCoffeePublicWeb$random"  
location="WestUS"  
dockerHubContainerPath="FourthCoffee/publicweb:v1"  
fqdn="http://www.fourthcoffee.com">www.fourthcoffee.com
```



```
az webapp create  
--name $appName  
--plan AppServiceLinuxDockerPlan  
--resource-group  
fourthCoffeePublicWebResourceGroup
```



```
az webapp config container set  
--docker-custom-image-name  
$dockerHubContainerPath  
--name $appName  
--resource-group  
fourthCoffeePublicWebResourceGroup
```

```
az webapp config hostname add  
--webapp-name $appName  
--resource-group  
fourthCoffeePublicWebResourceGroup \  
--hostname $fqdn
```

Explanation:

Step 1: #bin/bash

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

Step 2: 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 3: 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.

Step 4: az webapp config hostname add

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

The custom domain mapping is done after the app is in place running. So it will be the last step in the process.

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

<https://docs.microsoft.com/en-us/azure/app-service/app-service-web-tutorial-custom-domain?toc=/azure/app-service/containers/toc.json>

QUESTION 94

Drag and Drop Question

You are developing a serverless Java application on Azure. You create a new Azure Key Vault to work with secrets from a new Azure Functions application.

The application must meet the following requirements:

- Reference the Azure Key Vault without requiring any changes to the Java code.
- Dynamically add and remove instances of the Azure Functions host based on the number of incoming application events.
- Ensure that instances are perpetually warm to avoid any cold starts.
- Connect to a VNet.
- Authentication to the Azure Key Vault instance must be removed if the Azure Function application is deleted.

You need to grant the Azure Functions application access to the Azure Key Vault.

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.

| Actions | Answer Area |
|--|---|
| Create a user-assigned managed identity for the application. | |
| Create the Azure Functions app with a Premium plan type. | |
| Create an access policy in Azure Key Vault for the application identity. |  |
| Create an SSL certification in Azure Key Vault for the application identity. |  |
| Create the Azure Functions app with an App Service plan type. | |
| Create the Azure Functions app with a Consumption plan type. | |
| Create a system-assigned managed identity for the application. | |

Answer:

Actions**Answer Area**

Create the Azure Functions app with a Premium plan type.

Create the Azure Functions app with a Consumption plan type.

Create an SSL certification in Azure Key Vault for the application identity.

Create a user-assigned managed identity for the application.

Create the Azure Functions app with an App Service plan type.

Create an access policy in Azure Key Vault for the application identity.

Create a system-assigned managed identity for the application.

**Explanation:**

Step 1: Create the Azure Functions app with a Consumption plan type.
Use the Consumption plan for serverless.

Step 2: Create a system-assigned managed identity for the application. Create a system-assigned managed identity for your application.

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

Step 3: Create an access policy in Key Vault for the application identity. Create an access policy in Key Vault for the application identity you created earlier. Enable the "Get" secret permission on this policy. Do not configure the "authorized application" or applicationId settings, as this is not compatible with a managed identity.

Reference:

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

QUESTION 95

Hotspot Question

A company is developing a Java web app. The web app code is hosted in a GitHub repository located at <https://github.com/Contoso/webapp>.

The web app must be evaluated before it is moved to production. You must deploy the initial code release to a deployment slot named staging.

You need to create the web app and deploy the code.

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

NOTE: Each correct selection is worth one point.

Answer Area

```
gitrepo=https://github.com/Contoso/webapp
webappname=BusinessWebApp
resourcegroupname=BusinessAppResourceGroup

az group create --location centralus --name $resourcegroupname

az webapp create --name $webappname --resource-group $resourcegroupname --sku S3

az webapp create --name $webappname --resource-group $resourcegroupname --plan $webappname

az webapp deployment slot staging create --name $webappname --resource-group $resourcegroupname --slot staging

az webapp config --name $webappname --resource-group $resourcegroupname \
--slot staging --repo-url $gitrepo --branch master --manual-integration
```

Answer:

Answer Area

```
gitrepo=https://github.com/Contoso/webapp
webappname=BusinessWebApp
resourcegroupname=BusinessAppResourceGroup

az group create --location centralus --name $resourcegroupname
az webapp appservice plan create --name $webappname --resource-group $resourcegroupname --sku S3
az webapp create --name $webappname --resource-group $resourcegroupname --plan $webappname
az webapp deployment slot create --name $webappname --resource-group $resourcegroupname --slot staging
az webapp deployment source config --name $webappname --resource-group $resourcegroupname \
--slot staging --repo-url $gitrepo --branch master --manual-integration
```

The code block shows five Azure CLI commands. Each command is preceded by a dropdown menu showing options: 'group', 'webapp', 'appservice plan', 'webapp deployment slot', and 'webapp deployment source'. In each menu, the selected option is highlighted in green.

- 1st command: 'group' is selected.
- 2nd command: 'appservice plan' is selected.
- 3rd command: 'webapp' is selected.
- 4th command: 'webapp deployment slot' is selected.
- 5th command: 'webapp deployment source' is selected.

Explanation:

Box 1: group

Create a resource group.

```
az group create --location westeurope --name myResourceGroup
```

Box 2: appservice plan

Create an App Service plan in STANDARD tier (minimum required by deployment slots). az appservice plan create --name \$webappname --resource-group myResourceGroup --sku S1

Box 3: webapp

Create a web app.

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

Box 4: webapp deployment slot

Create a deployment slot with the name "staging".

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

Box 5: webapp deployment source

Deploy sample code to "staging" slot from GitHub.

```
az webapp deployment source config --name $webappname --resource-group myResourceGroup \
\ --slot staging --repo-url $gitrepo --branch master --manual-integration
```

Reference:

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

QUESTION 96

Hotspot Question

You have a web service that is used to pay for food deliveries. The web service uses Azure Cosmos DB as the data store.

You plan to add a new feature that allows users to set a tip amount. The new feature requires that a property named tip on the document in Cosmos DB must be present and contain a numeric value.

There are many existing websites and mobile apps that use the web service that will not be updated to set the tip property for some time.

How should you complete the trigger?

NOTE: Each correct selection is worth one point.

Answer Area

```
function ensureTip() {
    var r = 
        .value();
        .readDocument('item');
        getContext().getRequest();
        getContext().getResponse();

    var i = r.getBody();

    if (!("tip" in i)) {
        if (request.getValue("tip") == null) {
            if (isNaN(i["tip"]) || i["tip"] == null) {
                if (typeof __.pluck("tip") == 'number') {
                    i["tip"] = 0;
                }
            }
        }
    }

    r.setBody(i);
    r.setValue(i);
    __.upsertDocument(i);
    __.replaceDocument(i);
}
```

Answer:

Answer Area

```
function ensureTip() {
    var r = ___.value();
      ___.readDocument('item');
      getContext().getRequest();
      getContext().getResponse();
    var i = r.getBody();

    if (!("tip" in i)) {
        if (request.getValue("tip") == null) {
            if (isNaN(i["tip"]) || i["tip"] == null) {
                if (typeof __.pluck("tip") == 'number') {
                    i["tip"] = 0;
                }
            }
        }
        r.setBody(i);
        r.setValue(i);
        ___.upsertDocument(i);
        ___.replaceDocument(i);
    }
}
```

QUESTION 97

Drag and Drop Question

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

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

The system has the following requirements for subscriptions:

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

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

Which filter types should you implement? To answer, drag the appropriate filter types to the correct subscriptions. Each filter type may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

| Filter types | Answer Area | | | | | | | | | | | | |
|---------------------|---|--------------|-------------|--------------|-------------|--------------------|-------------|---------------------|-------------|--------------------|-------------|-----------|-------------|
| SQLFilter | <table><thead><tr><th>Subscription</th><th>Filter type</th></tr></thead><tbody><tr><td>FutureOrders</td><td>filter type</td></tr><tr><td>HighPriorityOrders</td><td>filter type</td></tr><tr><td>InternationalOrders</td><td>filter type</td></tr><tr><td>HighQuantityOrders</td><td>filter type</td></tr><tr><td>AllOrders</td><td>filter type</td></tr></tbody></table> | Subscription | Filter type | FutureOrders | filter type | HighPriorityOrders | filter type | InternationalOrders | filter type | HighQuantityOrders | filter type | AllOrders | filter type |
| Subscription | Filter type | | | | | | | | | | | | |
| FutureOrders | filter type | | | | | | | | | | | | |
| HighPriorityOrders | filter type | | | | | | | | | | | | |
| InternationalOrders | filter type | | | | | | | | | | | | |
| HighQuantityOrders | filter type | | | | | | | | | | | | |
| AllOrders | filter type | | | | | | | | | | | | |
| CorrelationFilter | | | | | | | | | | | | | |
| No Filter | | | | | | | | | | | | | |

Answer:

Filter types

| |
|-------------------|
| SQLFilter |
| CorrelationFilter |
| No Filter |
| |

Answer Area

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

Explanation:

FutureOrders: SQLFilter

HighPriorityOrders: CorrelationFilter

CorrelationID only

InternationalOrders: SQLFilter

Country NOT USA requires an SQL Filter

HighQuantityOrders: SQLFilter

Need to use relational operators so an SQL Filter is needed.

AllOrders: No Filter

SQL Filter: SQL Filters - A SqlFilter holds a SQL-like conditional expression that is evaluated in the broker against the arriving messages' user-defined properties and system properties. All system properties must be prefixed with sys. in the conditional expression. The SQL-language subset for filter conditions tests for the existence of properties (EXISTS), as well as for null-values (IS NULL), logical NOT/AND/OR, relational operators, simple numeric arithmetic, and simple text pattern matching with LIKE.

Correlation Filters: A CorrelationFilter holds a set of conditions that are matched against one or more of an arriving message's user and system properties. A common use is to match against the CorrelationId property, but the application can also choose to match against ContentType, Label, MessageId, ReplyTo, ReplyToSessionId, SessionId, To, and any user-defined properties. A match exists when an arriving message's value for a property is equal to the value specified in the correlation filter. For string expressions, the comparison is case-sensitive. When specifying multiple match properties, the filter combines them as a logical AND condition, meaning for the filter to match, all conditions must match.

Boolean filters: The TrueFilter and FalseFilter either cause all arriving messages (true) or none of the arriving messages (false) to be selected for the subscription.

Reference:<https://docs.microsoft.com/en-us/azure/service-bus-messaging/topic-filters>**QUESTION 98**

Drag and Drop Question

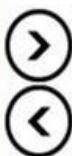
Your company has several websites that use a company logo image. You use Azure Content Delivery Network (CDN) to store the static image.

You need to determine the correct process of how the CDN and the Point of Presence (POP) server will distribute the image and list the items in the correct order.

In which order do the actions occur? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Actions**Answer Area**

If no edge servers in the POP have the image in cache, the POP requests the file from the origin server.



A user requests the image from the CDN URL. The DNS routes the request to the best performing POP location.



Subsequent requests for the file may be directed to the same POP using the CDN logo image URL. The POP edge server returns the file from cache if the TTL has not expired.



The origin server returns the logo image to an edge server in the POP. An edge server in the POP caches the logo image and returns the image to the client.

Answer:**Actions****Answer Area**

A user requests the image from the CDN URL. The DNS routes the request to the best performing POP location.



If no edge servers in the POP have the image in cache, the POP requests the file from the origin server.



The origin server returns the logo image to an edge server in the POP. An edge server in the POP caches the logo image and returns the image to the client.

Subsequent requests for the file may be directed to the same POP using the CDN logo image URL. The POP edge server returns the file from cache if the TTL has not expired.

Explanation:

Step 1: A user requests the image..

A user requests a file (also called an asset) by using a URL with a special domain name, such as <endpoint name>.azureedge.net. This name can be an endpoint hostname or a custom domain.

The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

Step 2: If no edge servers in the POP have the..

If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure Cloud Service, Azure Storage account, or any publicly accessible web server.

Step 3: The origin server returns the..

The origin server returns the file to an edge server in the POP. An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the POP until the time-to-live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.

Step 4: Subsequent requests for..

Additional users can then request the same file by using the same URL that the original user used, and can also be directed to the same POP.

If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.

Reference:

<https://docs.microsoft.com/en-us/azure/cdn/cdn-overview>

QUESTION 99

Hotspot Question

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

You have the following requirements:

- Code must be scalable and allow for growth.
- Each record must consist of a playerId, 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 plan to store customer information in Azure Cosmos DB. 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 to save scores in the data store. (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 }
```

You develop the following code to query the database. (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.GenerateFilterCondition (PartitionKey, QueryComparisons.Equal , "Smith"),
06 TableOperators.And , TableQuery.GenerateFilterCondition (Email, QueryComparisons.Equal, "ssmith@contoso.com")
07 ));
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.

Answer Area

| Statements | Yes | No |
|---|-----------------------|-----------------------|
| SaveScore will work with Cosmos DB. | <input type="radio"/> | <input type="radio"/> |
| SaveScore will update and replace a record if one already exists with the same playerId and gameId. | <input type="radio"/> | <input type="radio"/> |
| Leader board data for the game will be automatically partitioned using gameId. | <input type="radio"/> | <input type="radio"/> |
| SaveScore will store the values for the gameId and playerId parameters in the database. | <input type="radio"/> | <input type="radio"/> |

Answer:

Answer Area

| Statements | Yes | No |
|---|----------------------------------|----------------------------------|
| SaveScore will work with Cosmos DB. | <input checked="" type="radio"/> | <input type="radio"/> |
| SaveScore will update and replace a record if one already exists with the same playerId and gameId. | <input type="radio"/> | <input checked="" type="radio"/> |
| Leader board data for the game will be automatically partitioned using gameId. | <input checked="" type="radio"/> | <input type="radio"/> |
| SaveScore will store the values for the gameId and playerId parameters in the database. | <input checked="" type="radio"/> | <input type="radio"/> |

Explanation:

Box 1: Yes

Create a table.

A CloudTableClient object lets you get reference objects for tables and entities. The following code creates a CloudTableClient object and uses it to create a new CloudTable object, which represents a table // Retrieve storage account from connection-string.

```
CloudStorageAccount storageAccount =  
CloudStorageAccount.parse(storageConnectionString);
```

```
// Create the table client.
```

```
CloudTableClient tableClient = storageAccount.createCloudTableClient();
```

```
// Create the table if it doesn't exist.
```

```
String tableName = "people";
```

```
CloudTable cloudTable = tableClient.getTableReference(tableName);  
cloudTable.createIfNotExists();
```

Box 2: No

New records are inserted with TableOperation.insert. Old records are not updated.

To update old records TableOperation.insertOrReplace should be used instead.

Box 3: No

Box 4: Yes

Reference:

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

QUESTION 100

Hotspot Question

You are developing a solution that uses the Azure Storage Client library for .NET. You have the following code: (Line numbers are included for reference only.)

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

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

NOTE: Each correct selection is worth one point.

Answer Area

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| The code creates an infinite lease | <input type="radio"/> | <input type="radio"/> |
| The code at line 06 always creates a new blob | <input type="radio"/> | <input type="radio"/> |
| The finally block releases the lease | <input type="radio"/> | <input type="radio"/> |

Answer:

Answer Area

| Statement | Yes | No |
|---|----------------------------------|----------------------------------|
| The code creates an infinite lease | <input checked="" type="radio"/> | <input type="radio"/> |
| The code at line 06 always creates a new blob | <input type="radio"/> | <input checked="" type="radio"/> |
| The finally block releases the lease | <input type="radio"/> | <input checked="" type="radio"/> |

Explanation:

Box 1: Yes

AcquireLeaseAsync does not specify leaseTime.

leaseTime is a TimeSpan representing the span of time for which to acquire the lease, which will be rounded down to seconds. If null, an infinite lease will be acquired. If not null, this must be 15 to 60 seconds.

Box 2: No

The GetBlockBlobReference method just gets a reference to a block blob in this container.

Box 3: Yes

The BreakLeaseAsync method initiates an asynchronous operation that breaks the current lease on this container.

Reference:

[https://docs.microsoft.com/en-](https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.acquireleaseasync)

[us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.getblockblobreference](https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.getblockblobreference)

https://docs.microsoft.com/en-

[us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.breakleaseasync](https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.breakleaseasync)

QUESTION 101

Hotspot Question

You are preparing to deploy a Python website to an Azure Web App using a container. The solution will use multiple containers in the same container group. The Dockerfile that builds the container is as follows:

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

You build a container by using the following command. The Azure Container Registry instance named images is a private registry.

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

The user name and password for the registry is admin.

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

You need to create an Azure Web App to run the website.

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

NOTE: Each correct selection is worth one point.

Answer Area

```
az configure --defaults web=website
az configure --defaults group=website
az appservice plan create --name websitePlan
az webapp create --plan websitePlan
az webapp config
```

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

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

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

Answer:**Answer Area**

```
az configure --defaults web=website
az configure --defaults group=website
az appservice plan create --name websitePlan
az webapp create --plan websitePlan
az webapp config
```

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

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

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

Explanation:

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

--hyper-v

Host web app on Windows container.

Box 2: --deployment-source-url images.azurecr.io/website:v1.0.0 --deployment-source-url -u
Git repository URL to link with manual integration.

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

Incorrect:

--deployment-container-image-name -i

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

Box 3: az webapp config container set -url https://images.azurecr.io -u admin -p admin

az webapp config container set
Set a web app container's settings.

Parameter: --docker-registry-server-url -r
The container registry server url.

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

Example:

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

Reference:

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

QUESTION 102

Hotspot Question

You are building a website to access project data related to teams within your organization. The website does not allow anonymous access. Authentication is performed using an Azure Active Directory (Azure AD) app named internal.

The website has the following authentication requirements:

- Azure AD users must be able to login to the website.
- Personalization of the website must be based on membership in Active Directory groups.

You need to configure the application's manifest to meet the authentication requirements.

How should you configure the manifest? To answer, select the appropriate configuration in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
{  
    ...  
    "appId": "d61126e3-089b-4adb-b721-d5023213df7d",  
    "displayName": "internal",  
    "optionalClaims": "All",  
    "groupMembershipClaims":  
    "allowPublicClient": true  
    "oauth2Permissions":  
    "requiredResourceAccess":  
    "oauth2AllowImplicitFlow":  
    ...  
}
```

Answer:**Answer Area**

```
{  
    ...  
    "appId": "d61126e3-089b-4adb-b721-d5023213df7d",  
    "displayName": "internal",  
    "optionalClaims": "All",  
    "groupMembershipClaims":  
    "allowPublicClient": true  
    "oauth2Permissions":  
    "requiredResourceAccess":  
    "oauth2AllowImplicitFlow":  
    ...  
}
```

Explanation:

Box 1: groupMembershipClaims

Scenario: Personalization of the website must be based on membership in Active Directory groups.

Group claims can also be configured in the Optional Claims section of the Application Manifest. Enable group membership claims by changing the groupMembershipClaim

The valid values are:

- "All"
- "SecurityGroup"
- "DistributionList"
- "DirectoryRole"

Box 2: oauth2Permissions

Scenario: Azure AD users must be able to login to the website.

oauth2Permissions specifies the collection of OAuth 2.0 permission scopes that the web API (resource) app exposes to client apps. These permission scopes may be granted to client apps during consent.

Incorrect Answers:

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

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-fed-group-claims>

QUESTION 103

Hotspot Question

You are debugging an application that is running on Azure Kubernetes cluster named cluster1. The cluster uses Azure Monitor for containers to monitor the cluster.

The application has sticky sessions enabled on the ingress controller.

Some customers report a large number of errors in the application over the last 24 hours.

You need to determine on which virtual machines (VMs) the errors are occurring.

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

NOTE: Each correct selection is worth one point.

Answer Area

let startTimestamp =

| |
|------------------|
| ago(1d) |
| since(1d) |
| totimespan(1d) |
| date(now() - 1d) |

let ContainerIDs = KubePodInventory
| where ClusterName == "Cluster1"

| |
|----------------------|
| top ContainerID |
| union ContainerID |
| sample ContainerID |
| distinct ContainerID |

ContainerLog

| |
|--|
| fork containerIDs |
| where ContainerID in (ContainerIDs) |
| restrict ContainerID in (ContainerIDs) |
| join ContainerID == ContainerIDs.ContainerID |

| where TimeGenerated > startTimestamp
| where LogEntrySource == "stderr"

| |
|-------------------------------|
| project by Computer |
| summarize by Computer |
| partition count() by Computer |
| summarize count() by Computer |

Answer:

Answer Area

```
let startTimestamp =
```

| |
|------------------|
| ago(1d) |
| since(1d) |
| totimespan(1d) |
| date(now() - 1d) |

```
let ContainerIDs = KubePodInventory  
| where ClusterName == "Cluster1"
```

| |
|----------------------|
| top ContainerID |
| union ContainerID |
| sample ContainerID |
| distinct ContainerID |

```
ContainerLog
```

| |
|--|
| fork containerIDs |
| where ContainerID in (ContainerIDs) |
| restrict ContainerID in (ContainerIDs) |
| join ContainerID == ContainerIDs.ContainerID |

```
| where TimeGenerated > startTimestamp  
| where LogEntrySource == "stderr"
```

| |
|-------------------------------|
| project by Computer |
| summarize by Computer |
| partition count() by Computer |
| summarize count() by Computer |

Explanation:

Box 1: ago(1d)

Box 2: distinct containerID

Box 3: where ContainerID in (ContainerIDs)

Box 4: summarize Count by Computer

Summarize: aggregate groups of rows

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

Reference:

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

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

QUESTION 104

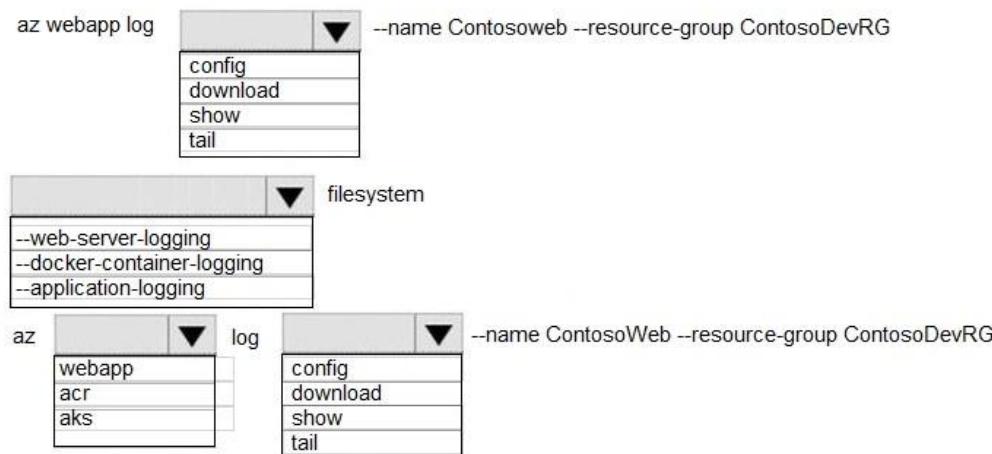
Hotspot Question

You plan to deploy a web app to App Service on Linux. You create an App Service plan. You create and push a custom Docker image that contains the web app to Azure Container Registry.

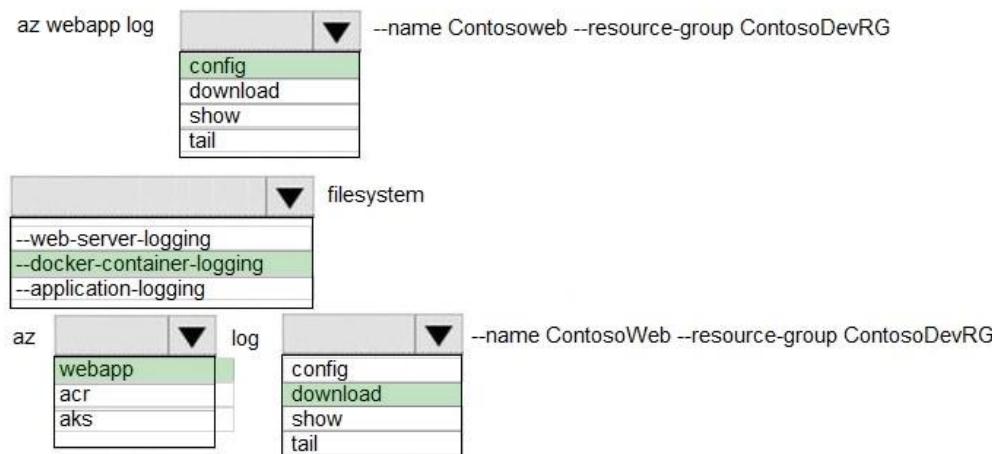
You need to access the console logs generated from inside the container in real-time.

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

NOTE: Each correct selection is worth one point.

Answer Area

Answer:

Answer Area

Explanation:

Box 1: config

To Configure logging for a web app use the command:

az webapp log config

Box 2: --docker-container-logging

Syntax include:

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

Box 3: webapp

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

az webapp log download

Box 4: download

Reference:

<https://docs.microsoft.com/en-us/cli/azure/webapp/log>**QUESTION 105**

Drag and Drop Question

A web service provides customer summary information for e-commerce partners. The web service is implemented as an Azure Function app with an HTTP trigger. Access to the API is provided by an Azure API Management instance. The API Management instance is configured in consumption plan mode. All API calls are authenticated by using OAuth.

API calls must be cached. Customers must not be able to view cached data for other customers.

You need to configure API Management policies for caching.

How should you complete the policy statement?

| Targets | Answer Area |
|---------------|--|
| Expect | <policies> |
| Public | <inbound> |
| Private | <base /> |
| Internal | <cache-lookup caching-type=" |
| External | Target |
| Authorization | " downstream-caching-type = " Target " |
| | </cache-lookup> |
| | <vary-by-header> |
| | Target |
| | </vary-by-header> |
| | </Cache-lookup> |
| | </inbound> |
| | </policies> |

Answer:

| Targets | Answer Area |
|----------|-------------|
| Expect | |
| Public | |
| Private | |
| External | |

<policies>
<inbound>
<base />
<cache-lookup caching-type=" Internal " downstream-caching-type = " Target ">
<vary-by-header>
 Authorization
</vary-by-header>
</cache-lookup>
</inbound>
</policies>

Explanation:

Box 1: internal
caching-type

Choose between the following values of the attribute:

internal to use the built-in API Management cache, external to use the external cache as Azure Cache for Redis prefer-external to use external cache if configured or internal cache otherwise.

Box 2: private

downstream-caching-type

This attribute must be set to one of the following values.

none - downstream caching is not allowed.

private - downstream private caching is allowed.

public - private and shared downstream caching is allowed.

Box 3: Authorization

<vary-by-header>Authorization</vary-by-header>
<!-- should be present when allow-private-response-caching is "true"-->

Note: Start caching responses per value of specified header, such as Accept, Accept-Charset, Accept- Encoding, Accept-Language, Authorization, Expect, From, Host, If-Match

Reference:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-caching-policies>

QUESTION 106

Hotspot Question

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.

Answer Area

```
void ClearCachedTeams()
```

```
{
```

```
    IDatabase cache = Connection.GetDatabase();  
    ICache cache = Connection.GetDatabase();
```

```
    cache.KeyDelete("Team");  
    cache.StringSet("Team", "");  
    cache.ValueDelete("Team");  
    cache.StringGet("Team", "");
```

```
    ViewBag.msg += "Team data removed from cache.";
```

```
}
```

Answer:

Answer Area

```
void ClearCachedTeams()

{
    IDatabase cache = Connection.GetDatabase();
    ICache cache = Connection.GetDatabase();

    cache.KeyDelete("Team");
    cache.StringSet("Team", "");
    cache.ValueDelete("Team");
    cache.StringGet("Team");

    ViewBag.msg += "Team data removed from cache.";
}
```

Explanation:

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

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

Reference:

<https://azure.microsoft.com/sv-se/blog/lap-around-azure-redis-cache-preview/>
<https://docs.microsoft.com/en-us/cli/azure/webapp/config/container>

QUESTION 107

Drag and Drop Question

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.

| Policy types | Answer Area | Policy type |
|--------------|---|-------------|
| Inbound | Requirement Rewrite the request URL to match to the format expected by the web service. | policy type |
| Outbound | Requirement Remove formatting text from responses. | policy type |
| Backend | Requirement Forward the user ID that is associated with the subscription key for the original request to the back-end service. | policy type |
| ... | ... | ... |

Answer:

| Policy types | Answer Area | Policy type |
|--------------|---|-------------|
| ... | Requirement Rewrite the request URL to match to the format expected by the web service. | Outbound |
| ... | Requirement Remove formatting text from responses. | Inbound |
| ... | Requirement Forward the user ID that is associated with the subscription key for the original request to the back-end service. | Backend |
| ... | ... | ... |

QUESTION 108

Drag and Drop Question

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

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

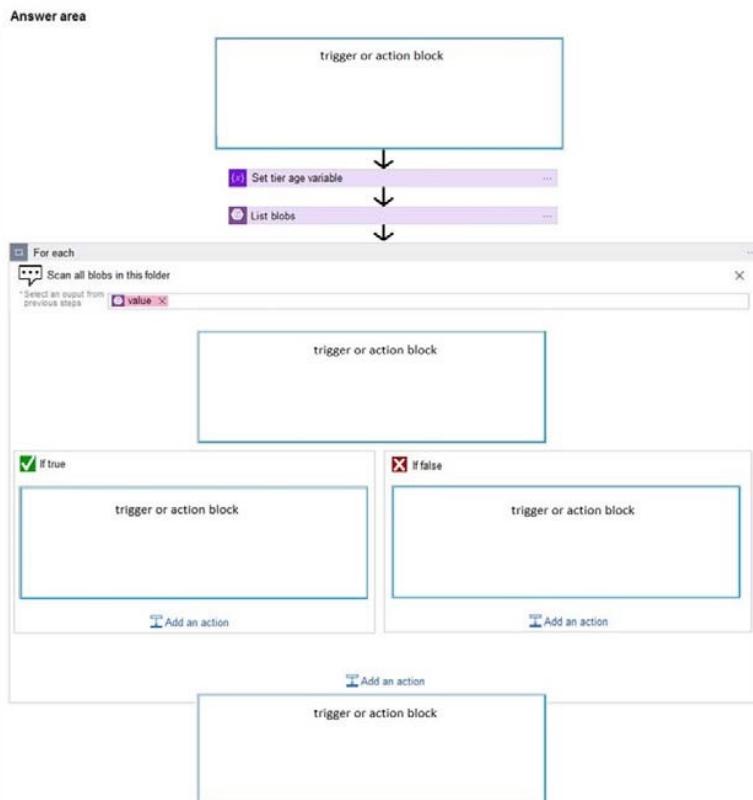
How should you configure the Logic App? To answer, drag the appropriate triggers or action blocks to the correct trigger or action slots. Each trigger or action block may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

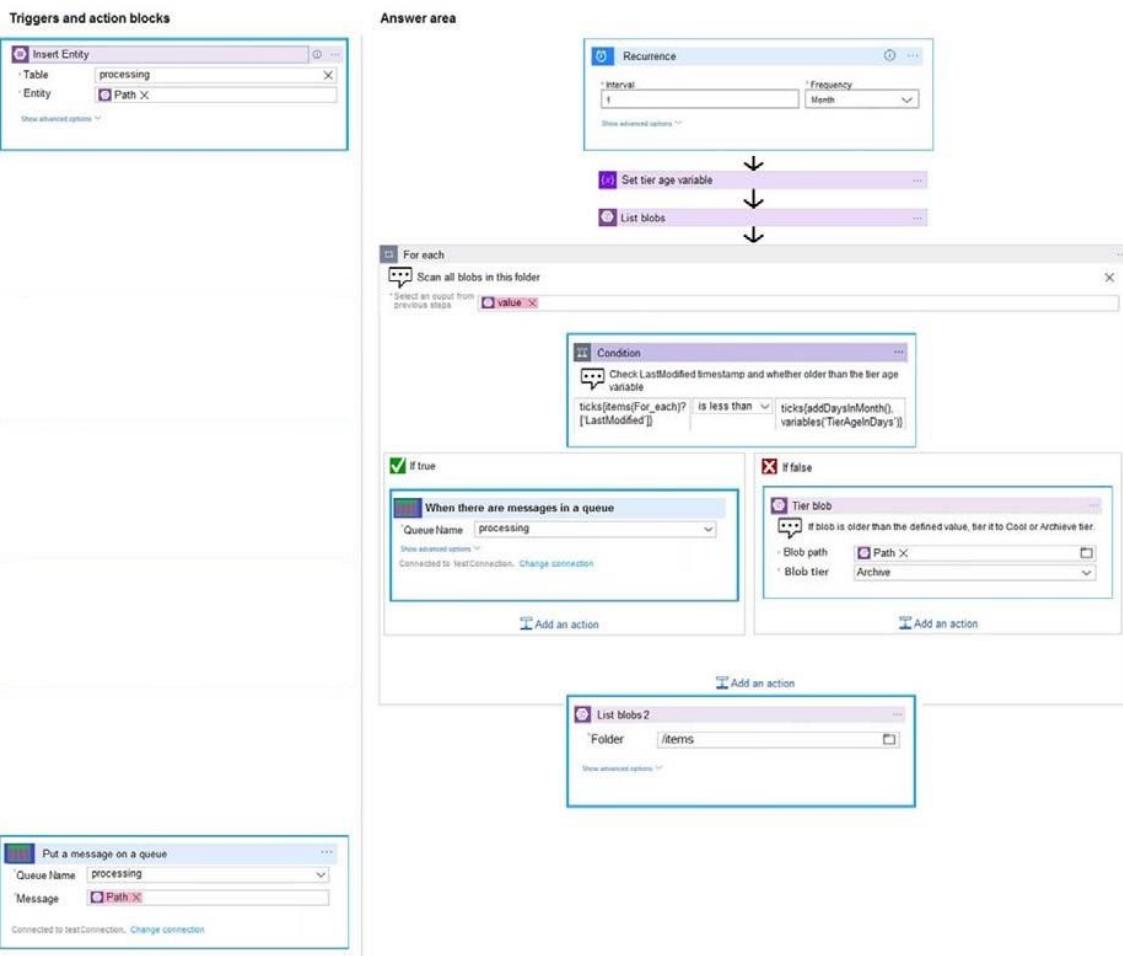
Triggers and action blocks

- Insert Entity**: Table -> processing, Entity -> Path
- Tier blob**: Blob path -> Path, Blob tier -> Archive
- When there are messages in a queue**: Queue Name -> processing
- Recurrence**: Interval -> 1, Frequency -> Month
- List blobs2**: Folder -> /items
- Condition**: Check LastModified timestamp and whether older than the tier age variable


```
ticks[items[For_each]?] is less than ticks[addDaysInMonth(), [LastModified]]]
variables[TierAgeInDays]]
```
- Put a message on a queue**: Queue Name -> processing, Message -> Path



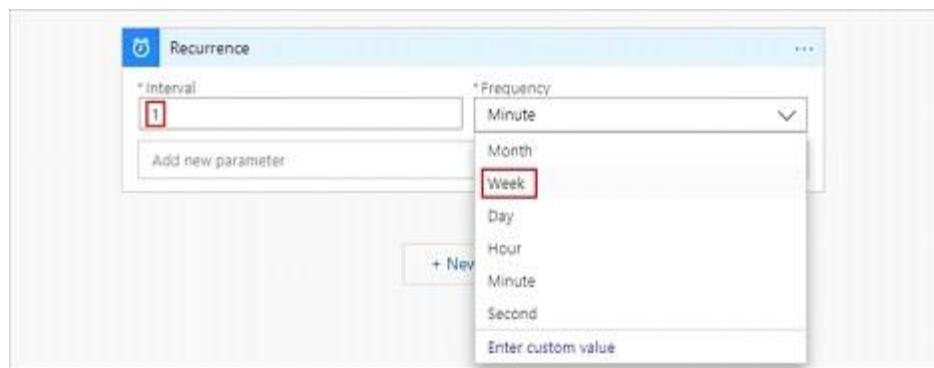
Answer:



Explanation:

Box 1: Recurrence..

To regularly run tasks, processes, or jobs on specific schedule, you can start your logic app workflow with the built-in Recurrence - Schedule trigger. You can set a date and time as well as a time zone for starting the workflow and a recurrence for repeating that workflow. Set the interval and frequency for the recurrence. In this example, set these properties to run your workflow every week.



Box 2: Condition..

To run specific actions in your logic app only after passing a specified condition, add a conditional

statement. This control structure compares the data in your workflow against specific values or fields. You can then specify different actions that run based on whether or not the data meets the condition.

Box 3: Put a message on a queue

The path for any item that is not archived must be placed in an existing queue.

Note: Under If true and If false, add the steps to perform based on whether the condition is met.

Box 4: ...tier it to Cool or Archive tier.

Archive item.

Box 5: List blobs 2

Reference:

<https://docs.microsoft.com/en-us/azure/connectors/connectors-native-recurrence>

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-control-flow-loops>

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-control-flow-conditional-statement>

QUESTION 109

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

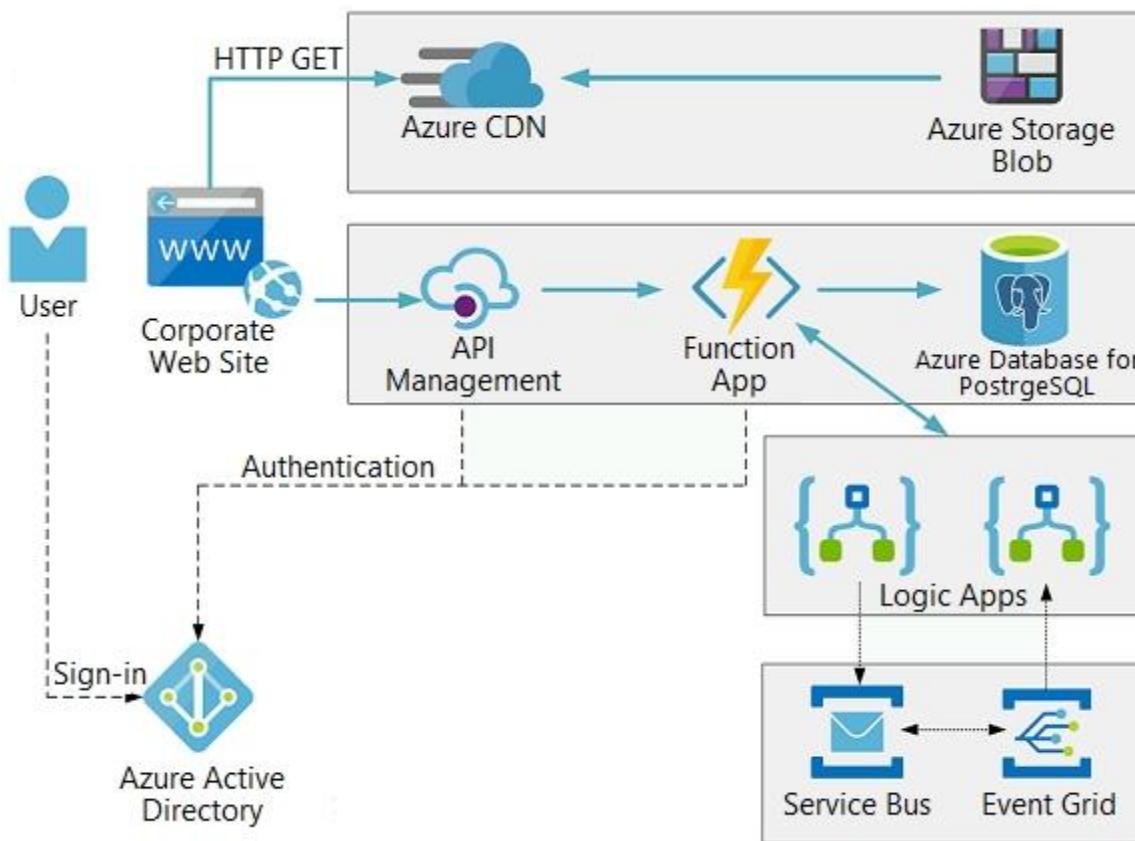
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

49. The user selects **Sign in** in the website.
50. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
51. The user signs in.
52. Azure AD redirects the user's session back to the web application. The URL includes an access token.
53. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
54. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

You need to ensure that all messages from Azure Event Grid are processed.

What should you use?

- A. Azure Event Grid topic
- B. Azure Service Bus topic
- C. Azure Service Bus queue
- D. Azure Storage queue
- E. Azure Logic App custom connector

Answer: C

Explanation:

As a solution architect/developer, you should consider using Service Bus queues when: Your solution needs to receive messages without having to poll the queue. With Service Bus, you can achieve it by using a long-polling receive operation using the TCP-based protocols that Service Bus supports.

Reference:

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

QUESTION 110

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

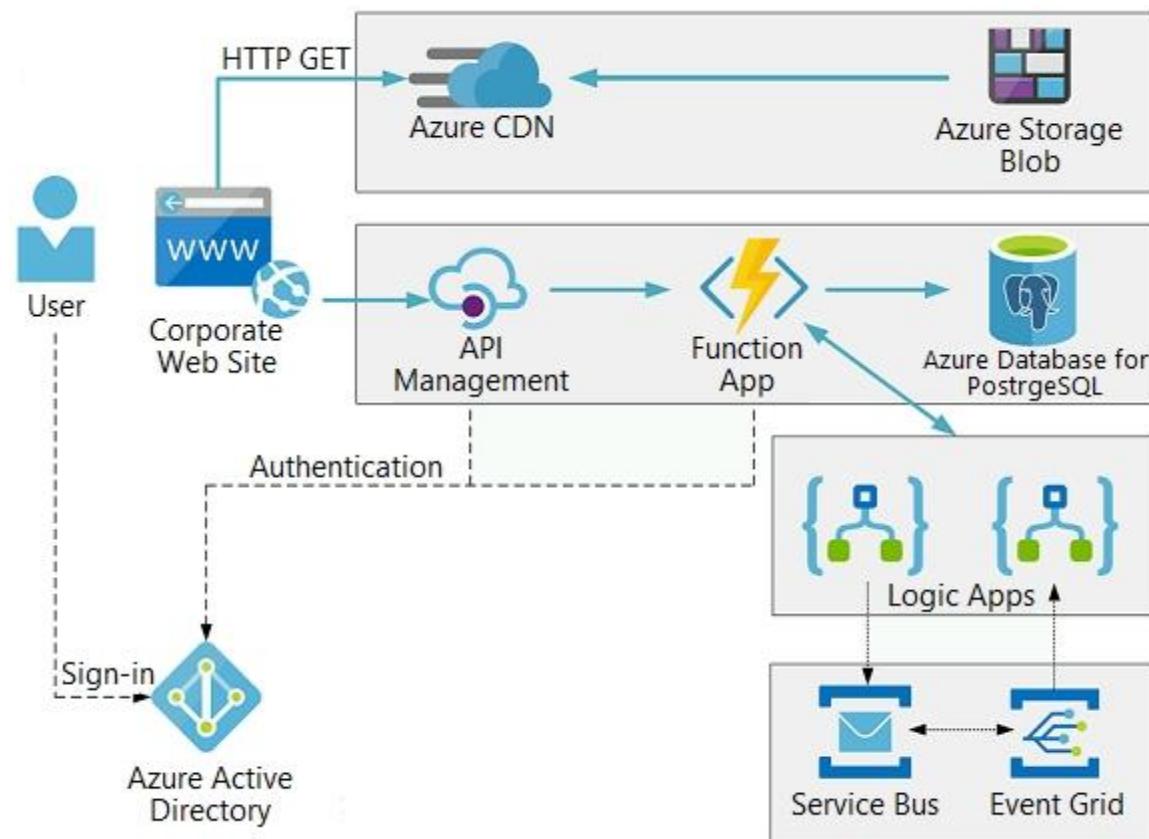
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

55. The user selects **Sign in** in the website.
56. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
57. The user signs in.
58. Azure AD redirects the user's session back to the web application. The URL includes an access token.
59. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
60. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

FunctionAppLogs

| where FunctionName == "RequestUserApproval"

Logic app

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

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

Code**Corporate website**

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to correct the Azure Logic app error message.

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

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|----------------------|--|
| authentication level | <div style="border: 1px solid black; padding: 5px; width: fit-content;">▼</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; height: 100px;"><p>anonymous</p><p>function</p><p>admin</p></div> |
| managed identity | <div style="border: 1px solid black; padding: 5px; width: fit-content;">▼</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; height: 100px;"><p>system-assigned</p><p>user-assigned</p></div> |

Answer:

Answer Area

| Setting | Value |
|----------------------|--|
| authentication level | <div style="border: 1px solid black; padding: 5px; width: fit-content;">▼</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; height: 100px;"><p>anonymous</p><p>function</p><p>admin</p></div> |
| managed identity | <div style="border: 1px solid black; padding: 5px; width: fit-content;">▼</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; height: 100px;"><p>system-assigned</p><p>user-assigned</p></div> |

Explanation:

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

'400 Bad Request'

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

Note: If the inbound call's request body doesn't match your schema, the trigger returns an HTTP 400 Bad Request error.

Box 1: function

If you have an Azure function where you want to use the system-assigned identity, first enable authentication for Azure functions.

Box 2: system-assigned

Your logic app or individual connections can use either the system-assigned identity or a single user-assigned identity, which you can share across a group of logic apps, but not both.

Reference:

<https://docs.microsoft.com/en-us/azure/logic-apps/create-managed-service-identity>

QUESTION 111

Case Study 3 - City Power & Light

Background

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment

Architecture overview

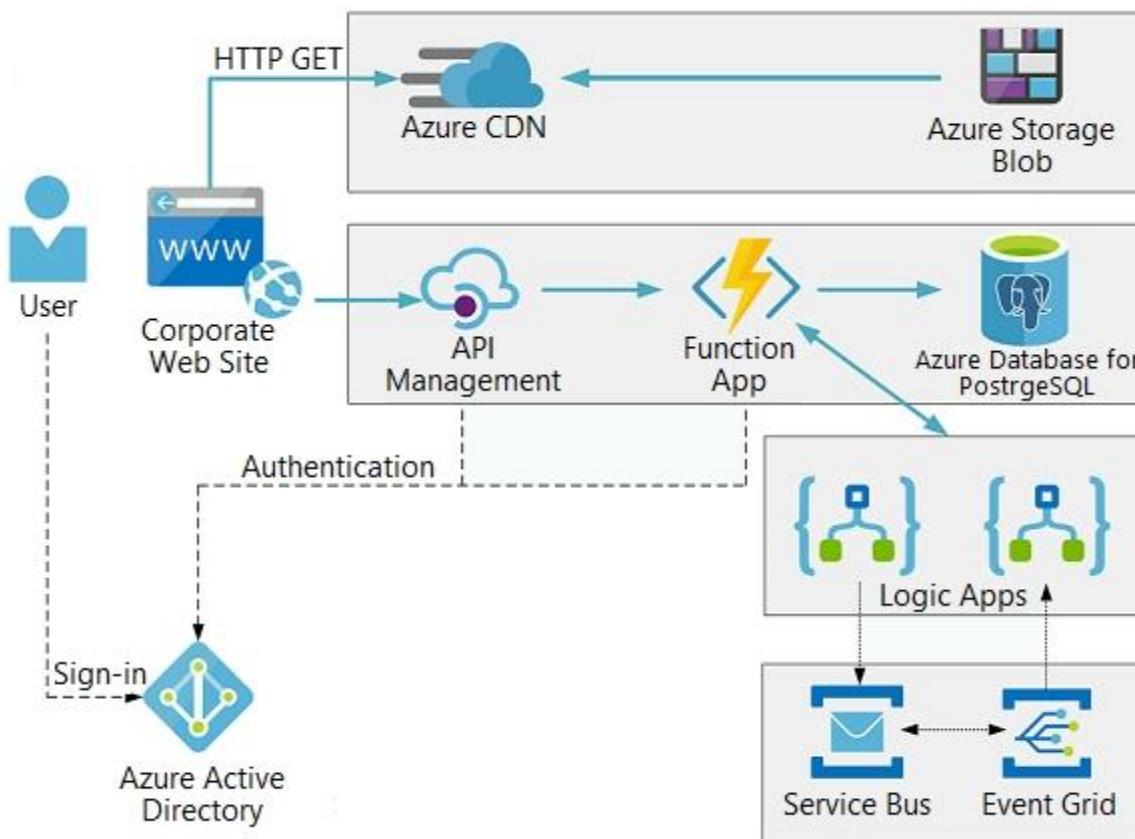
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.



User authentication

The following steps detail the user authentication process:

61. The user selects **Sign in** in the website.
62. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
63. The user signs in.
64. Azure AD redirects the user's session back to the web application. The URL includes an access token.
65. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
66. The back-end API validates the access token.

Requirements

Corporate website

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to configure Azure Service Bus to Event Grid integration.

Which Azure Service Bus settings should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| | Setting | Value |
|-----------|----------------|---|
| Tier | | <input type="checkbox"/> Basic <input type="checkbox"/> Standard <input type="checkbox"/> Premium |
| RBAC role | | <input type="checkbox"/> Owner <input type="checkbox"/> Contributor <input type="checkbox"/> Azure Service Bus Data Owner <input type="checkbox"/> Azure Service Bus Data Receiver |

Answer:

Answer Area

| Setting | Value |
|----------------|--|
| Tier | <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">Basic Standard Premium</div> |
| RBAC role | <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">Owner Contributor Azure Service Bus Data Owner Azure Service Bus Data Receiver</div> |

Explanation:

Box 1: Premium

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

To enable the feature, you need the following items:

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

Contributor access to the Service Bus namespace.

Box 2: Contributor

Reference:

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

QUESTION 112**Case Study 3 - City Power & Light****Background**

City Power & Light company provides electrical infrastructure monitoring solutions for homes and businesses. The company is migrating solutions to Azure.

Current environment**Architecture overview**

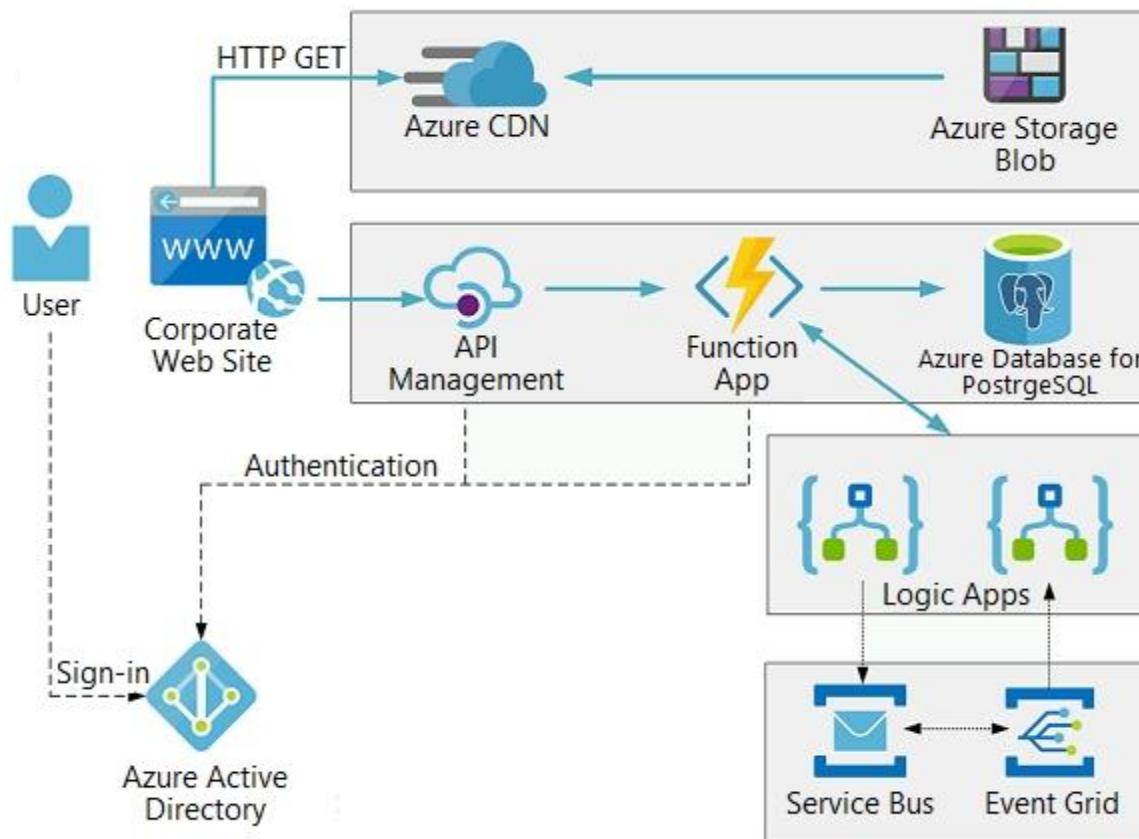
The company has a public website located at <http://www.cpndl.com/>. The site is a single-page web application that runs in Azure App Service on Linux. The website uses files stored in Azure Storage and cached in Azure Content Delivery Network (CDN) to serve static content.

API Management and Azure Function App functions are used to process and store data in Azure Database for PostgreSQL. API Management is used to broker communications to the Azure Function app functions for Logic app integration. Logic apps are used to orchestrate the data processing while Service Bus and Event Grid handle messaging and events.

The solution uses Application Insights, Azure Monitor, and Azure Key Vault.

Architecture diagram

The company has several applications and services that support their business. The company plans to implement serverless computing where possible. The overall architecture is shown below.

**User authentication**

The following steps detail the user authentication process:

67. The user selects **Sign in** in the website.
68. The browser redirects the user to the Azure Active Directory (Azure AD) sign in page.
69. The user signs in.
70. Azure AD redirects the user's session back to the web application. The URL includes an access token.
71. The web application calls an API and includes the access token in the authentication header.
The application ID is sent as the audience ('aud') claim in the access token.
72. The back-end API validates the access token.

Requirements**Corporate website**

- Communications and content must be secured by using SSL.
- Communications must use HTTPS.
- Data must be replicated to a secondary region and three availability zones.
- Data storage costs must be minimized.

Azure Database for PostgreSQL

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

- Azure Key Vault name: cpndlkeyvault
- Secret name: PostgreSQLConn
- Id: 80df3e46ffcd4f1cb187f79905e9a1e8

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

Azure Service Bus and Azure Event Grid

- Azure Event Grid must use Azure Service Bus for queue-based load leveling.
- Events in Azure Event Grid must be routed directly to Service Bus queues for use in buffering.
- Events from Azure Service Bus and other Azure services must continue to be routed to Azure Event Grid for processing.

Security

- All SSL certificates and credentials must be stored in Azure Key Vault.
- File access must restrict access by IP, protocol, and Azure AD rights.
- All user accounts and processes must receive only those privileges which are essential to perform their intended function.

Compliance

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

Issues

Corporate website

While testing the site, the following error message displays:

CryptographicException: The system cannot find the file specified.

Function app

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

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

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

```
FunctionAppLogs  
| where FunctionName == "RequestUserApproval"
```

Logic app

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

'400 Bad Request'

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

Code

Corporate website

Security.cs:

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

Function app

RequestUserApproval.cs:

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

Hotspot Question

You need to configure security and compliance for the corporate website files.

Which Azure Blob storage settings should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Action | Setting |
|----------------------|---|
| Restrict file access | <ul style="list-style-type: none">role-based access control (RBAC)managed identityshared access signature (SAS) tokenconnection string |
| Enable file auditing | <ul style="list-style-type: none">access tierchange feedblob indexerstorage account type |

Answer:

Answer Area

| Action | Setting |
|----------------------|--|
| Restrict file access | role-based access control (RBAC) managed identity shared access signature (SAS) token connection string |
| Enable file auditing | access tier change feed blob indexer storage account type |

Explanation:

Box 1: role-based access control (RBAC)

Azure Storage supports authentication and authorization with Azure AD for the Blob and Queue services via Azure role-based access control (Azure RBAC).

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

Box 2: storage account type

Scenario: The website uses files stored in Azure Storage Auditing of the file updates and transfers must be enabled to comply with General Data Protection Regulation (GDPR).

Creating a diagnostic setting:

1. Sign in to the Azure portal.
2. Navigate to your storage account.
3. In the Monitoring section, click Diagnostic settings (preview).

| NAME | RESOURCE TYPE | RESOURCE GROUP | DIAGNOSTICS STATUS |
|------------------|-----------------|-----------------|--------------------|
| mystorageaccount | Storage account | myresourcegroup | Disabled |
| blob | Storage account | myresourcegroup | Disabled |
| queue | Storage account | myresourcegroup | Disabled |
| table | Storage account | myresourcegroup | Disabled |
| file | Storage account | myresourcegroup | Disabled |

4. Choose file as the type of storage that you want to enable logs for.
5. Click Add diagnostic setting.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

<https://docs.microsoft.com/en-us/azure/storage/files/storage-files-monitoring>

Exam B

QUESTION 1

Case Study 1 - Litware Inc

Background

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

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

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details

Users table

| Column | Description |
|----------------|---|
| UserId | unique identifier for an employee |
| ExpenseAccount | employee's expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL

database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements

Receipt processing

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues

Upload format issue

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

Capacity issue

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

Log capacity issue

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

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlock (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().CreateIfNotExistAsync();
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
```

Hotspot Question

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

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

NOTE: Each correct selection is worth one point.

Answer Area

| Operating system | Supported |
|------------------|-----------|
|------------------|-----------|

| | |
|-----------|----------------------------------|
| Windows 7 | <input type="button" value="▼"/> |
|-----------|----------------------------------|

| |
|-----|
| No |
| Yes |

| | |
|-------------|----------------------------------|
| Windows 8.1 | <input type="button" value="▼"/> |
|-------------|----------------------------------|

| |
|-----|
| No |
| Yes |

| | |
|------------|----------------------------------|
| Windows 10 | <input type="button" value="▼"/> |
|------------|----------------------------------|

| |
|-----|
| No |
| Yes |

Answer:**Answer Area**

| Operating system | Supported |
|------------------|-----------|
|------------------|-----------|

| | |
|-----------|----------------------------------|
| Windows 7 | <input type="button" value="▼"/> |
|-----------|----------------------------------|

| |
|-----|
| No |
| Yes |

| | |
|-------------|----------------------------------|
| Windows 8.1 | <input type="button" value="▼"/> |
|-------------|----------------------------------|

| |
|-----|
| No |
| Yes |

| | |
|------------|----------------------------------|
| Windows 10 | <input type="button" value="▼"/> |
|------------|----------------------------------|

| |
|-----|
| No |
| Yes |

Explanation:

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

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

²Windows Server, version 1709 and 1803.

Scenario: Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

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

References:

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

QUESTION 2

Case study 1 - Litware Inc

Background

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

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

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details**Users table**

| Column | Description |
|----------------|--|
| UserId | unique identifier for an employee |
| ExpenseAccount | employees expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements**Receipt processing**

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.

- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues

Upload format issue

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

Capacity issue

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

Log capacity issue

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

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBBlob (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().CreateIfNotExistAsync();
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     }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

Drag and Drop Question

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

What code should you add at line RU14?

To answer, drag the appropriate code fragments to the correct locations. Each code fragment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

| Values | Answer Area |
|------------------------------------|--------------------------------------|
| SMBDeletePending | |
| ShareBeingDeleted | |
| HttpStatusCode.Conflict | return response.StatusCode = = |
| CannotDeleteFileOrDirectory | && response.ReasonPhrase = = "" ; |
| HttpStatusCode.InternalServerError | |

Answer:

| Values | Answer Area |
|-------------------------|--|
| SMBDeletePending | |
| ShareBeingDeleted | |
| HttpStatusCode.Conflict | return response.StatusCode = = HttpStatusCode.InternalServerError && response.ReasonPhrase = = "" CannotDeleteFileOrDirectory ; |

Explanation:

Box 1: HttpStatusCode.InternalServerError

HttpStatusCode.InternalServerError is equivalent to HTTP status 500. InternalServerError indicates that a generic error has occurred on the server.

Box 2: CannotDeleteFileOrDirectory

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

Scenario: Upload format issue

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

References:

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

QUESTION 3**Case study 1 - Litware Inc****Background**

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

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

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details**Users table**

| Column | Description |
|----------------|---|
| UserId | unique identifier for an employee |
| ExpenseAccount | employee's expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements**Receipt processing**

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues**Upload format issue**

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

Capacity issue

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

Log capacity issue

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

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlock (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().CreateIfNotExistAsync();
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(HttpStatusCode 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
```

Hotspot Question

You need to ensure that security requirements are met.

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

NOTE: Each correct selection is worth one point.

Answer Area

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

| | |
|---------------------------------|----|
| Integrated Security = SSPI | ; |
| Trusted_Connection = False | |
| Network Library = DBNSOCN | |
| MultipleActiveResultSets = True | |
| Encrypt = True | ;" |
| Integrated Security = True | |
| Failover Partner = False | |
| Named Pipes = True | |

Answer:**Answer Area**

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

| | |
|---------------------------------|----|
| Integrated Security = SSPI | ; |
| Trusted_Connection = False | |
| Network Library = DBNSOCN | |
| MultipleActiveResultSets = True | |
| Encrypt = True | ;" |
| Integrated Security = True | |
| Failover Partner = False | |
| Named Pipes = True | |

Explanation:

Box 1: Integrated Security=SSPI

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

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

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

Box 2: Encrypt = True

Scenario: All data must be protected in transit.

References:

<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-syntax>**QUESTION 4****Case study 1 - Litware Inc****Background**

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

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

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details**Users table**

| Column | Description |
|----------------|---|
| UserId | unique identifier for an employee |
| ExpenseAccount | employee's expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements**Receipt processing**

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues**Upload format issue**

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

Capacity issue

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

Log capacity issue

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

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlock (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().CreateIfNotExistAsync();
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
```

Hotspot Question

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.

Answer Area

```
var policy=
```

| |
|----------------------|
| Policy |
| RetryPolicy |
| RetryOptions |
| ReconnectRetryPolicy |

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

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

Answer:**Answer Area**

```
var policy=
```

| |
|----------------------|
| Policy |
| RetryPolicy |
| RetryOptions |
| ReconnectRetryPolicy |

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

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

Explanation:

Box 1: Policy

RetryPolicy retry = Policy

```
.Handle<HttpRequestException>()
```

```
.Retry(3);
```

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

Box 2: WaitAndRetryAsync(3,i => TimeSpan.FromMilliseconds(100* Math.Pow(2,i-1)));

A common retry strategy is exponential backoff: this allows for retries to be made initially quickly, but then at progressively longer intervals, to avoid hitting a subsystem with repeated frequent calls if the subsystem may be struggling.

Example:

Policy
.Handle<SomeExceptionType>()
.WaitAndRetry(3, retryAttempt =>
TimeSpan.FromSeconds(Math.Pow(2, retryAttempt))
);
References:
<https://github.com/App-vNext/Polly/wiki/Retry>

QUESTION 5**Case Study 2 - Coho Winery****LabelMaker app**

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

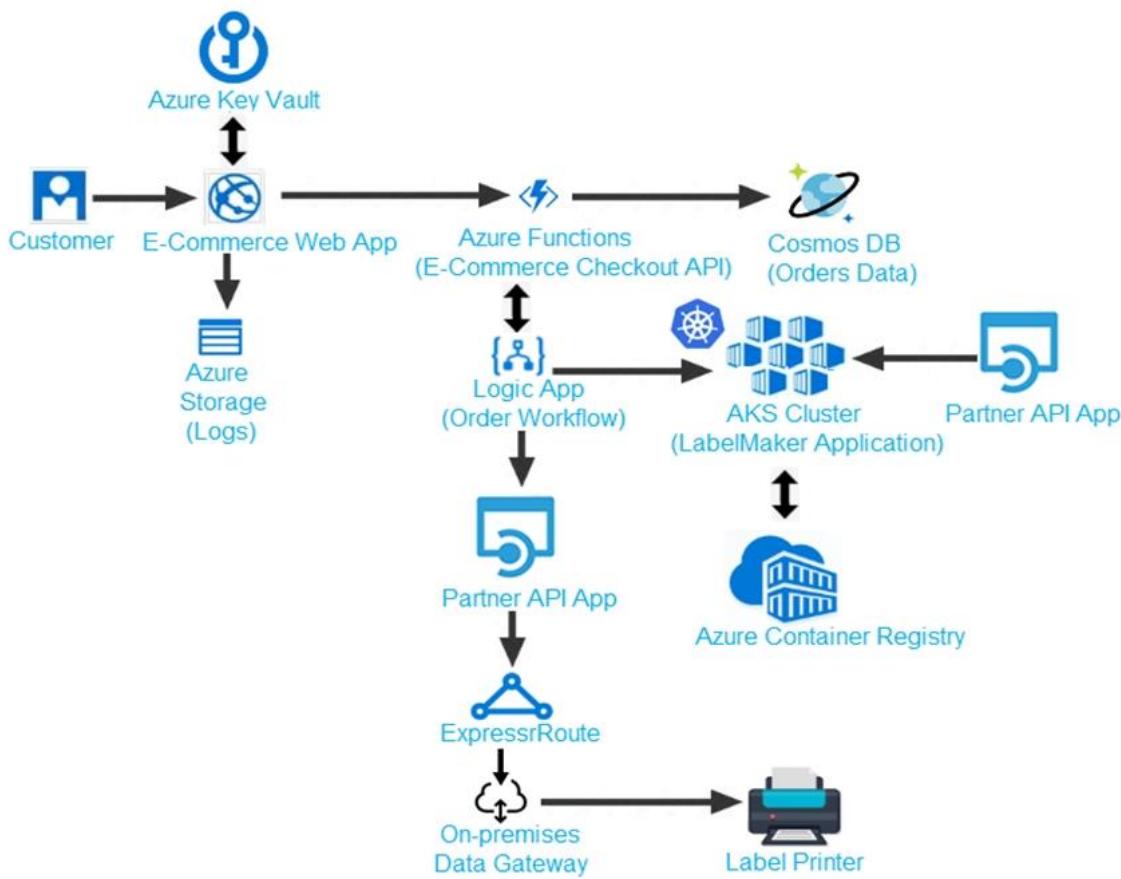
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order .json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

You need to troubleshoot the order workflow.

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

NOTE: Each correct selection is worth one point.

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

Answer: BD

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 6

Case Study 2 - Coho Winery

LabelMaker app

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

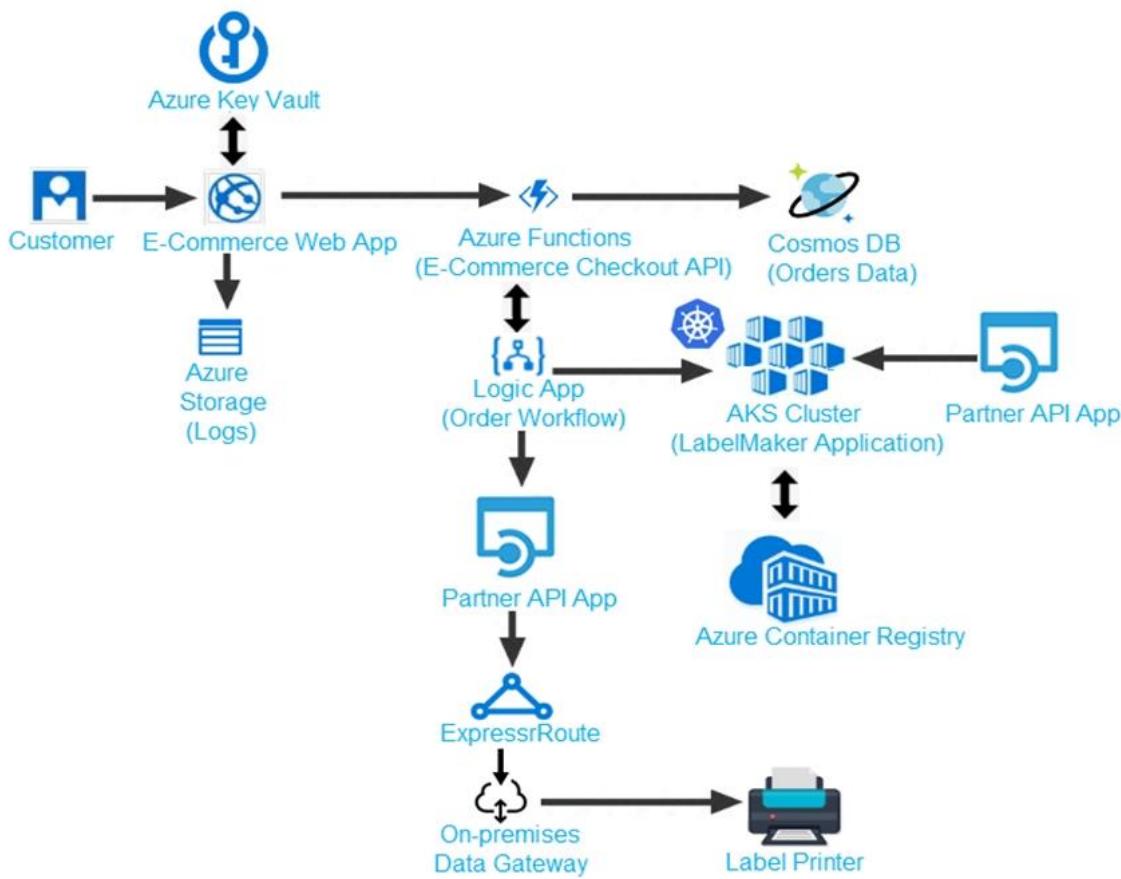
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order .json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

Drag and Drop Question

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

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

NOTE: Each correct selection is worth one point.

Actions

Restart the cluster.

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

Build a new application image by using msbuild.

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



Build a new application image by using dockerfile.

Download the image to your local computer.

Log in to the registry and push image.

Answer Area**Answer Area**

Build a new application image by using dockerfile.

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

Log in to the registry and push image.



Download the image to your local computer.

Explanation:

Step 1: Build a new application image by using dockerfile

Step 2: Create an alias if the image with the fully qualified path to the registry

Before you can push the image to a private registry, you've to ensure a proper image name. This can be achieved using the docker tag command. For demonstration purpose, we'll use Docker's hello world image, rename it and push it to ACR.

```
# pulls hello-world from the public docker hub
```

```
$ docker pull hello-world
```

```
# tag the image in order to be able to push it to a private registry
```

```
$ docker tag hello-world <REGISTRY_NAME>/hello-world
```

```
# push the image
```

```
$ docker push <REGISTRY_NAME>/hello-world
```

Step 3: Log in to the registry and push image

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

Scenario:

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

LabelMaker app

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

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

References:

<https://thorsten-hans.com/how-to-use-a-private-azure-container-registry-with-kubernetes-9b86e67b93b6>

<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-tutorial-quick-task>

QUESTION 7**Case Study 2 - Coho Winery****LabelMaker app**

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

You have the following security requirements:

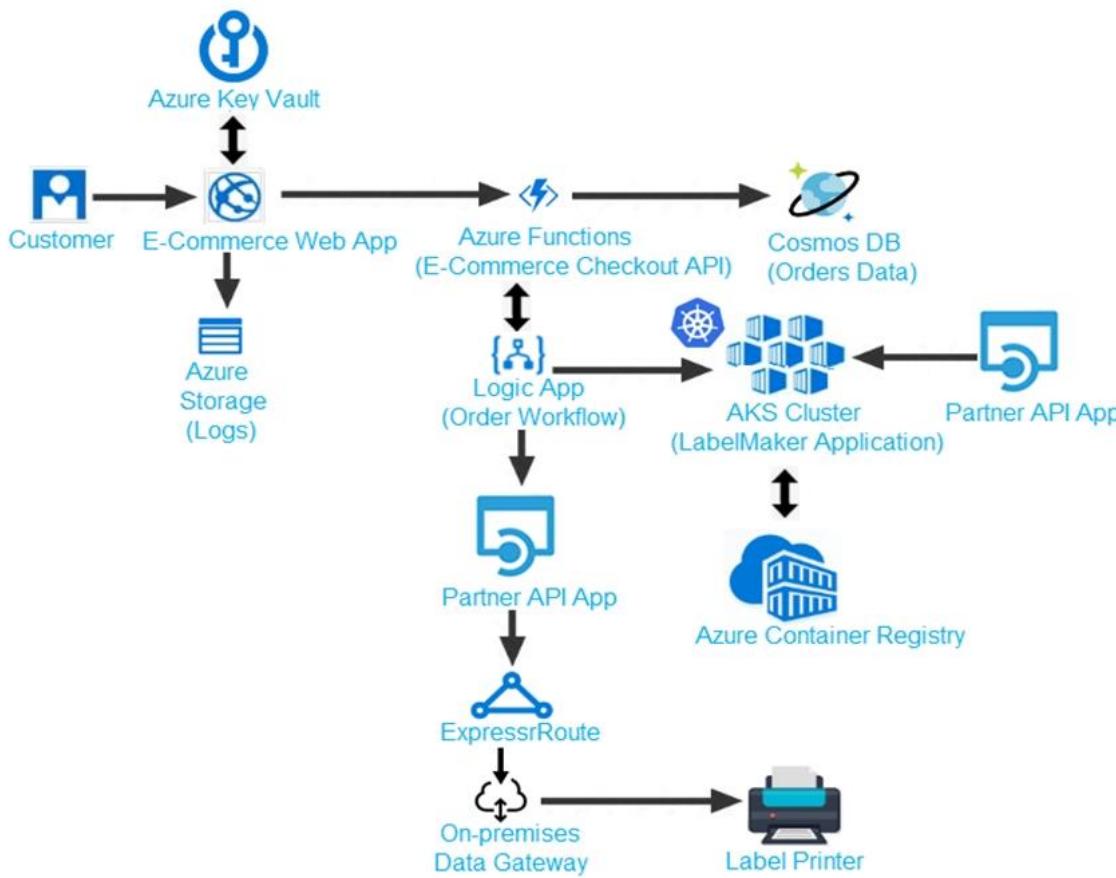
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.

- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order.json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

Hotspot Question

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

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

NOTE: Each correct selection is worth one point.

Answer Area

```
az group create --name CohoWIneryLabelMaker --location eastus
az group create --resource-group CohoWIneryLabelMaker --name
LabelMakerCluster --node-count 5 --enable-addons monitoring
http_application_routing
```

Answer:
Answer Area

```
az group create --name CohoWIneryLabelMaker --location eastus
az group create --resource-group CohoWIneryLabelMaker --name
LabelMakerCluster --node-count 5 --enable-addons monitoring
http_application_routing
```

Explanation:

Box 1: group

Create a resource group with the az group create command. An Azure resource group is a logical group in which Azure resources are deployed and managed.

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

```
az group create --name myResourceGroup --location westeurope
```

Box 2: CohoWinterLabelMaker

Use the resource group named, which is used in the second command.

Box 3: aks

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

Box 4: monitoring

Scenario: LabelMaker app

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

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

QUESTION 8
Case Study 2 - Coho Winery

LabelMaker app

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

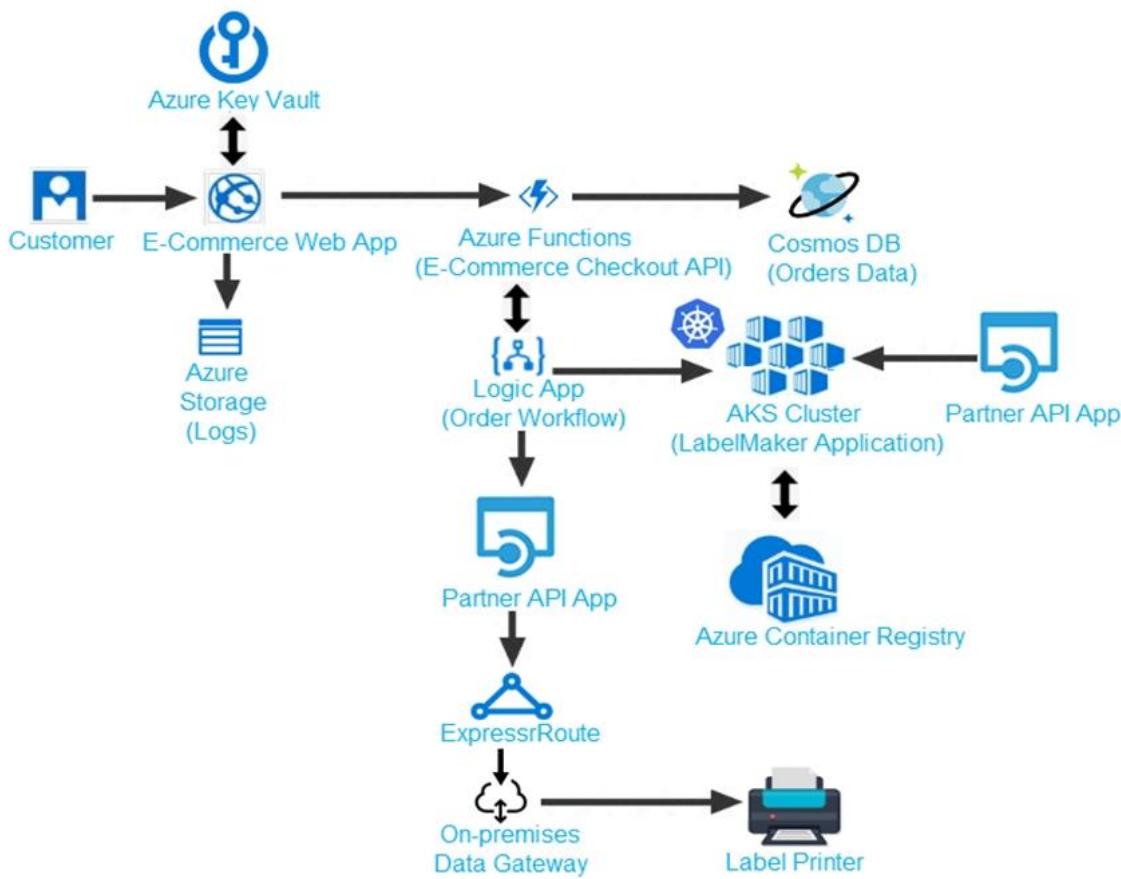
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order .json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

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

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

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

You need to meet the LabelMaker application security requirement.

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

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 9

Case Study 2 - Coho Winery

LabelMaker app

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

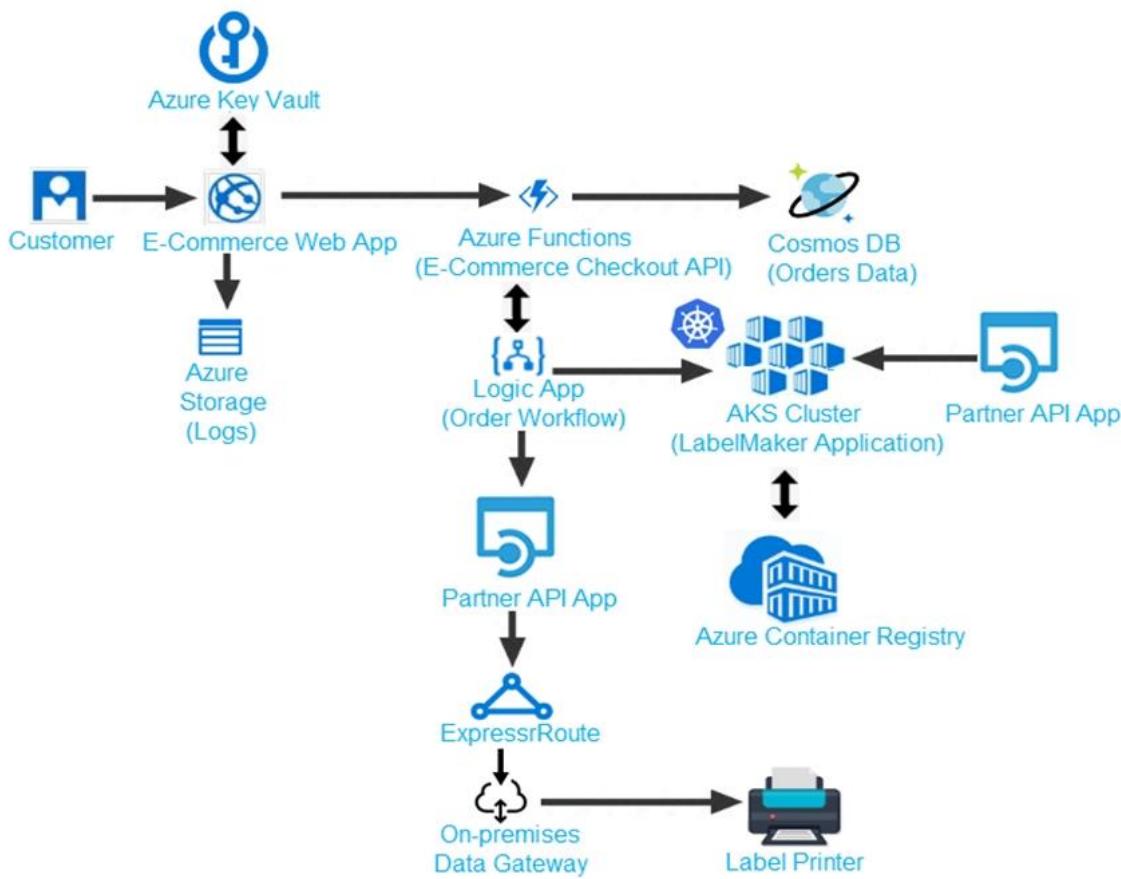
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order .json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

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

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

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

You need to meet the LabelMaker application security requirement.

Solution: Create a conditional access policy and assign it to the Azure Kubernetes Service cluster.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

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

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

References:

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

QUESTION 10

Case Study 2 - Coho Winery

LabelMaker app

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

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

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

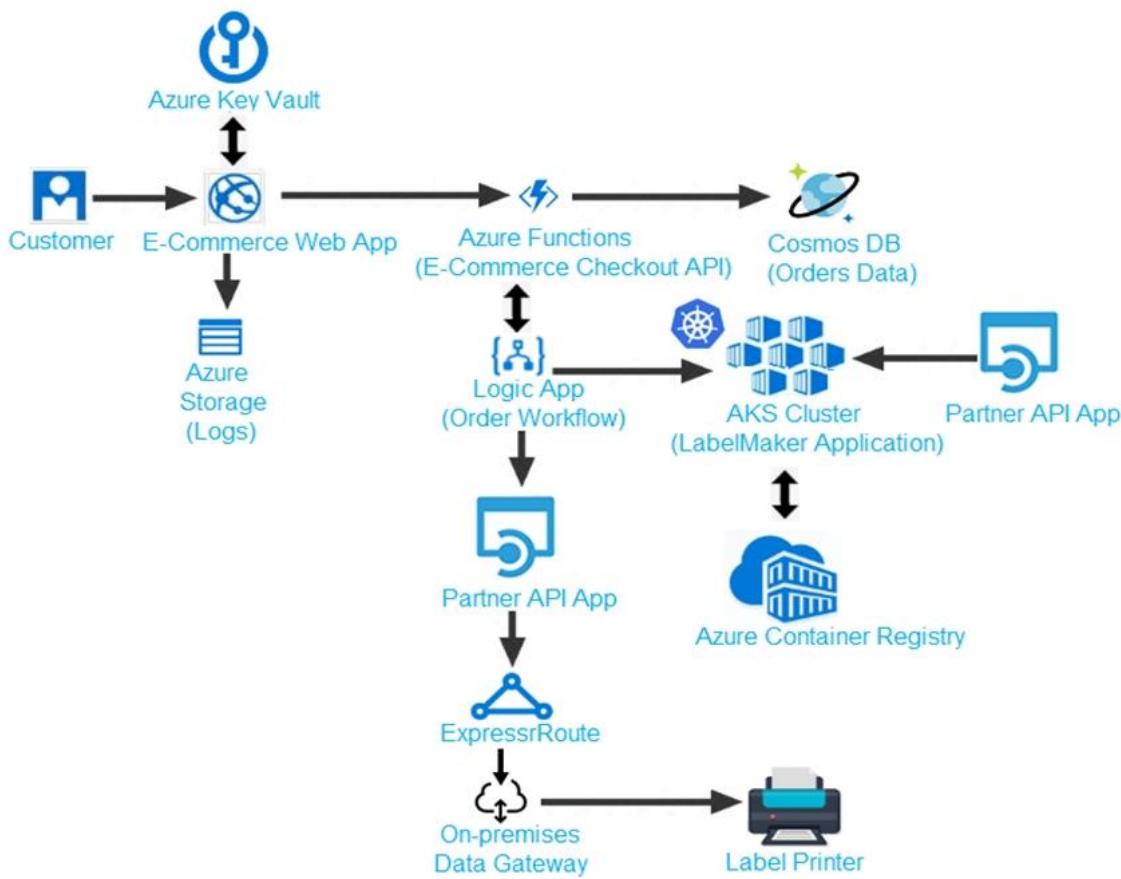
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

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

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



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

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

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

Order json.

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

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

Order .json

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

```
23 "tax_lines" : [
24 {
25   "title" : "State Tax",
26   "price" : "3.98",
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",
42   "city" : "NY"
43 }
44 }
```

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

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

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

You need to meet the LabelMaker application security requirement.

Solution: Create a RoleBinding and assign it to the Azure AD account.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

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 11

Case Study 3 - Proseware, Inc

Background

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

Requirements

Policy service

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

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

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

Policies

Log Policy

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

Authentication events

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

PolicyLib

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

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

Other

Anomaly detection service

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

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

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

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

LoginEvent.cs

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

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

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

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

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

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

QUESTION 12

Case Study 3 - Proseware, Inc

Background

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

Requirements

Policy service

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

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

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

Policies

Log Policy

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

Authentication events

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

PolicyLib

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

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

Other

Anomaly detection service

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

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

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

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

LoginEvent.cs

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

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

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

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

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

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

QUESTION 13

Case Study 3 - Proseware, Inc

Background

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

Requirements

Policy service

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

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

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

Policies

Log Policy

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

Authentication events

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

PolicyLib

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

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

Other

Anomaly detection service

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

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

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

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

LoginEvent.cs

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

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

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

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

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

QUESTION 14

Case Study 3 - Proseware, Inc

Background

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

Requirements

Policy service

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

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

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

Policies

Log Policy

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

Authentication events

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

PolicyLib

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

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

Other

Anomaly detection service

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

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

```

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

```

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

LoginEvent.cs

```

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

```

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.

| Code segments | Answer Area |
|------------------|---|
| topic | if (@event["data"] ["status"].ToString() == "Succeeded") |
| status | |
| eventType | && @event["data"] ["operationName"].ToString() == "Microsoft.Web/sites/write" |
| Succeeded | |
| operationName |) |
| resourceProvider | |

Answer:

| Code segments | Answer Area |
|------------------|---|
| topic | if (@event["data"] ["status"].ToString() == "Succeeded") |
| status | |
| eventType | && @event["data"] ["operationName"].ToString() == "Microsoft.Web/sites/write" |
| operationName |) |
| resourceProvider | |

Explanation:

Box 1: Status

Box 2: Succeeded

Box 3: operationName

Scenario: Policy service

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

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

QUESTION 15

Case Study 3 - Proseware, Inc

Background

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

Requirements

Policy service

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

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

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

Policies

Log Policy

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

Authentication events

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

PolicyLib

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

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

Other

Anomaly detection service

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

```
EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody]) string eventsJson
EG05  {
EG06      var events = JArray.Parse(eventsJson);
EG07
EG08      foreach (var @event in events)
EG09      {
EG10          EventId.Value = @event ["id"].ToString();
EG11          if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12          {
EG13              SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14          }
EG15
EG16          {
EG17              EnsureLogging(@event["subject"].ToString());
EG18          }
EG19      }
EG20      return null;
EG21  }
EG22  private void EnsureLogging(string resource)
EG23  {
EG24      . .
EG25  }
EG26  private async Task SendToAnomalyDetectionService(string uri)
EG27  {
EG28      var content = GetLogData(uri);
EG29      var scoreRequest = new
EG30      {
EG31          Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32          {
EG33              {
EG34                  "input1",
EG35                  new List<Dictionary<string, string>>()
EG36                  {
EG37                      new Dictionary<string, string>()
EG38                      {
EG39                          {
EG40                              "logcontent", content
EG41                          }
EG42                      }
EG43                  }
EG44              },
EG45          },
EG46          GlobalParameters = new Dictionary<string, string>() { }
EG47      };
EG48      var result = await (new HttpClient()).PostAsJsonAsync(" . . . ", scoreRequest);
EG49      var rawModelResult = await result.Content.ReadAsStringAsync();
EG50      var modelResult = JObject.Parse(rawModelResult);
EG51      if (modelResult["notify"].HasValues)
EG52      {
EG53          . .
EG54      }
EG55  }
EG56  private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57  {
EG58      . .
EG59  }
EG60  private string GetLogData(string uri)
EG61  {
EG62      . .
EG63  }
EG64  static string BlobStoreAccountSAS(string containerName)
EG65  {
EG66      . .
EG67  }
EG68 }
```

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

LoginEvent.cs

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

Drag and Drop Question

You need to implement the Log policy.

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

NOTE: Each correct selection is worth one point.

| Code segment | Answer Area |
|-------------------------------|---|
| All | { |
| WebHook | "name": "newlogs", "properties" : { |
| EventHub | "topic" : "/subscriptions/ . . . /providers/Microsoft.EventGrid/topics/. . . ", "destination": { |
| subjectEndsWith | "endpointType" : " [] ", "filter": { |
| Microsoft.Storage | " [] " : "/blobServices/default/containers/logdrop/", "includeEventTypes": [" [] "], |
| subjectBeginsWith | }, "labels": [], "eventDeliverySchema": "EventGridSchema" } } |
| Microsoft.Storage.BlobCreated | |

Answer:

| Code segment | Answer Area |
|-------------------|--|
| All | { |
| EventHub | "name": "newlogs", "properties": { "topic" : "/subscriptions/ . . ./providers/Microsoft.EventGrid/topics/ . . .", "destination": { "endpointType" : " WebHook ", "filter": { "subjectBeginsWith": "/blobServices/default/containers/logdrop/", "includeEventTypes": [" Microsoft.Storage.BlobCreated "]}, "labels": [], "eventDeliverySchema": "EventGridSchema" } } |
| subjectEndsWith | |
| Microsoft.Storage | |

Explanation:

Box 1:WebHook

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

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

Box 2: SubjectBeginsWith

Box 3: Microsoft.Storage.BlobCreated

Scenario: Log Policy

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

Example subscription schema

```
{  
    "properties": {  
        "destination": {  
            "endpointType": "webhook",  
            "properties": {  
                "endpointUrl":  
                    "https://example.azurewebsites.net/api/HttpTriggerCSharp1?code=VXbGWce53I48Mt8wuotr0GP  
myJ/nDT4hgdFj9DpBiRt38qqnnm5OFg==" }  
            },  
            "filter": {  
                "includedEventTypes": [ "Microsoft.Storage.BlobCreated", "Microsoft.Storage.BlobDeleted" ],  
                "subjectBeginsWith": "blobServices/default/containers/mycontainer/log",  
                "subjectEndsWith": ".jpg",  
                "isSubjectCaseSensitive ": "true"  
            }  
        }  
    }  
}
```

References:

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

QUESTION 16**Case Study 3 - Proseware, Inc****Background**

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

Requirements**Policy service**

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

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

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

Policies**Log Policy**

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

Authentication events

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

PolicyLib

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

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

Other**Anomaly detection service**

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

Health monitoring

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

Policy loss

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

Performance issue

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

Notification latency

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

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

```

EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody]) string eventsJson
EG05  {
EG06      var events = JArray.Parse(eventsJson);
EG07
EG08      foreach (var @event in events)
EG09      {
EG10          EventId.Value = @event ["id"].ToString();
EG11          if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12          {
EG13              SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14          }
EG15
EG16          {
EG17              EnsureLogging(@event["subject"].ToString());
EG18          }
EG19      }
EG20      return null;
EG21  }
EG22  private void EnsureLogging(string resource)
EG23  {
EG24      . .
EG25  }
EG26  private async Task SendToAnomalyDetectionService(string uri)
EG27  {
EG28      var content = GetLogData(uri);
EG29      var scoreRequest = new
EG30      {
EG31          Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32          {
EG33              {
EG34                  "input1",
EG35                  new List<Dictionary<string, string>>()
EG36                  {
EG37                      new Dictionary<string, string>()
EG38                      {
EG39                          {
EG40                              "logcontent", content
EG41                          }
EG42                      }
EG43                  }
EG44              },
EG45          },
EG46          GlobalParameters = new Dictionary<string, string>() { }
EG47      };
EG48      var result = await (new HttpClient()).PostAsJsonAsync(" . . . ", scoreRequest);
EG49      var rawModelResult = await result.Content.ReadAsStringAsync();
EG50      var modelResult = JObject.Parse(rawModelResult);
EG51      if (modelResult["notify"].HasValues)
EG52      {
EG53          . .
EG54      }
EG55  }
EG56  private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57  {
EG58      . .
EG59  }
EG60  private string GetLogData(string uri)
EG61  {
EG62      . .
EG63  }
EG64  static string BlobStoreAccountSAS(string containerName)
EG65  {
EG66      . .
EG67  }
EG68  }

```

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

LoginEvent.cs

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

You need to resolve a notification latency issue.

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

NOTE: Each correct selection is worth one point.

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

Answer: BD

Explanation:

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

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

Anomaly detection service: You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service.

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

References:

<https://github.com/Azure/Azure-Functions/wiki/Enable-Always-On-when-running-on-dedicated-App-Service-Plan>

QUESTION 17**Case Study 4 - Best for You Organics****Background**

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!



John Doe

Sun, Aug 26, 2018

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

Specials: Chicken Marsala

Order Pickup

Order Delivery

Vendor API

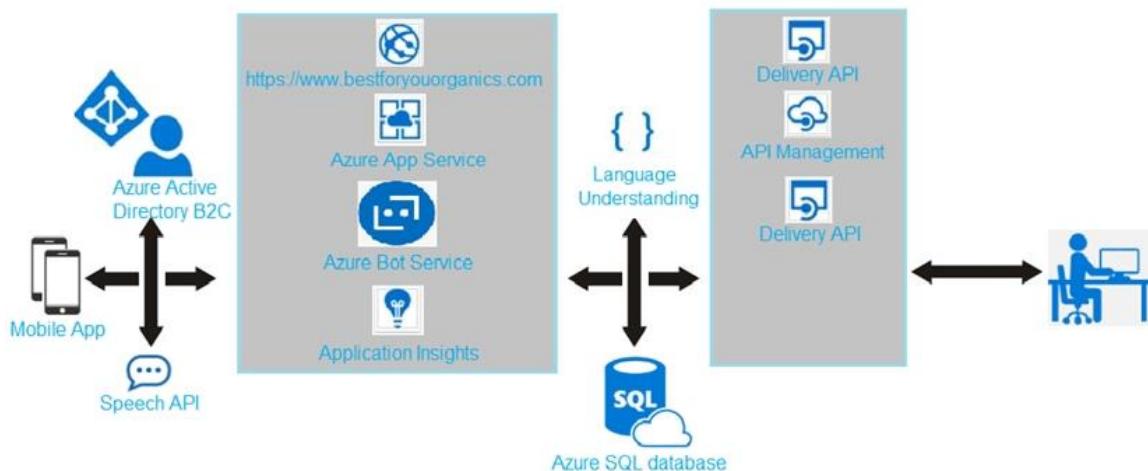
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

- API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.
- If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.
- APIs must prevent API usage spikes on a per-subscription basis by limiting the call rate to 100 calls per minute.
- The Inventory API must be written by using ASP.NET Core and Node.js.

- The API must be updated to provide an interface to Azure SQL Database. Database objects must be managed by using code.
- The Delivery API must be protected by using the OAuth 2.0 protocol with Azure Active Directory (Azure AD) when called from the Azure web app. You register the Delivery API and web app in Azure AD. You enable OAuth 2.0 in the web app.
- The delivery API must update the Products table, the Vendor transactions table, and the Billing table in a single transaction.

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

Hotspot Question

You need to update the Inventory API.

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

NOTE: Each correct selection is worth one point.

Answer Area

Development Tool

Technology

| |
|-----------------------|
| ADO.NET |
| Entity Framework |
| Entity Framework Core |
| WCF Data Services |

Workflow

| |
|----------------|
| Model first |
| Database first |
| Code first |

Answer:

Answer Area

Development Tool

Technology

| |
|-----------------------|
| ADO.NET |
| Entity Framework |
| Entity Framework Core |
| WCF Data Services |

Workflow

| |
|----------------|
| Model first |
| Database first |
| Code first |

Explanation:

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

Box 1: Entity Framework Core

Box 2: Code first

References:

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

QUESTION 18**Case Study 4 - Best for You Organics****Background**

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala

Order Pickup

Order Delivery

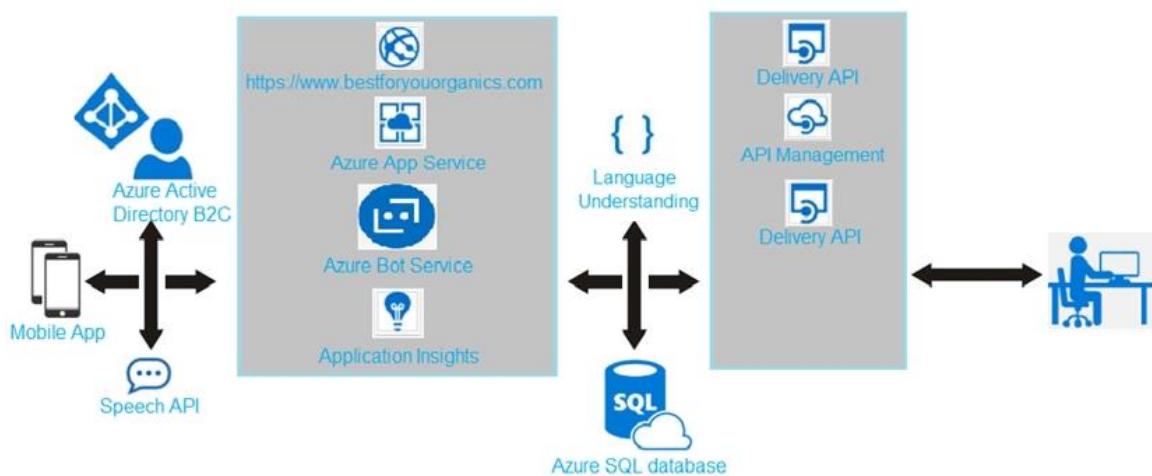
Vendor API

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

APIs must meet the following conditions:

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

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

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

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

You need to meet the vendor notification requirement.

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

Scenario:

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

(API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.) In Azure API Management (APIM), policies are a powerful capability of the system that allow the publisher to change the behavior of the API through configuration.

Policies are a collection of Statements that are executed sequentially on the request or response of an API. Popular Statements include format conversion from XML to JSON and call rate limiting to restrict the amount of incoming calls from a developer. Many more policies are available out of the box.

References:

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

QUESTION 19**Case Study 4 - Best for You Organics****Background**

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

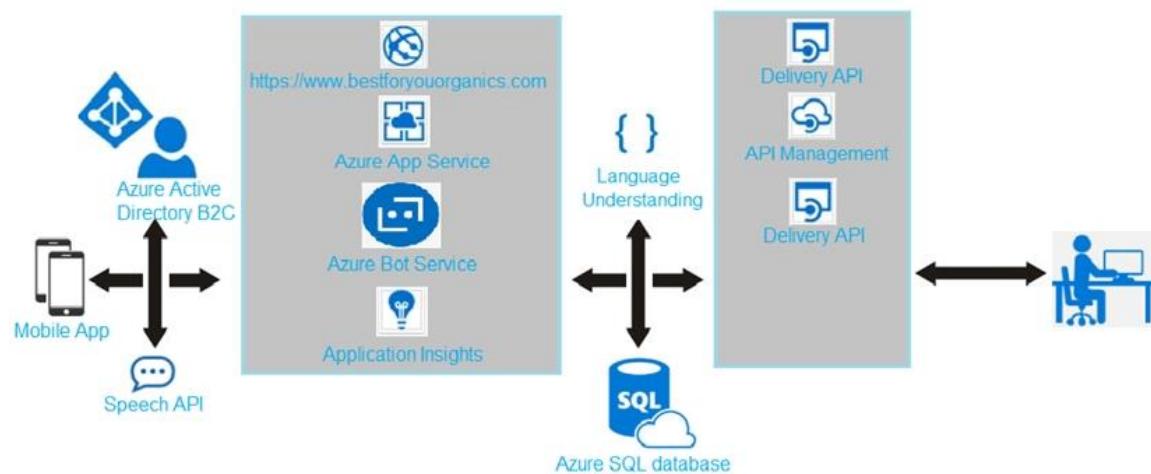
Specials: Chicken Marsala**Order Pickup****Order Delivery****Vendor API**

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

APIs must meet the following conditions:

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

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

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

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

You need to meet the vendor notification requirement.

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

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Use a custom outbound Azure API Management policy.

Scenario:

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

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

References:

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

QUESTION 20**Case Study 4 - Best for You Organics****Background**

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala**Order Pickup****Order Delivery****Vendor API**

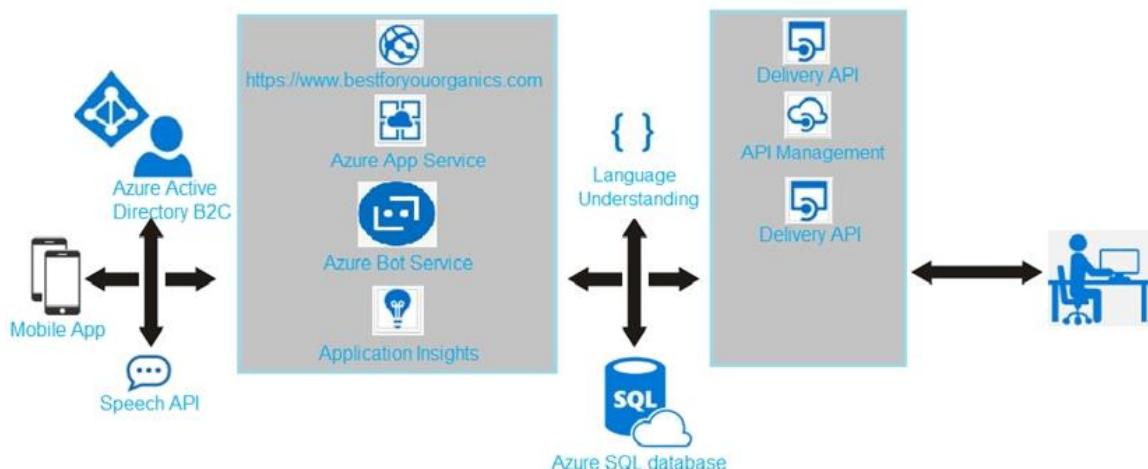
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

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

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

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

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

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

You need to meet the vendor notification requirement.

Solution: Configure notifications in the Azure API Management instance.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation: Use a custom outbound Azure API Management policy.

Scenario:

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

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

References:

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

QUESTION 21

Case Study 4 - Best for You Organics

Background

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala**Order Pickup****Order Delivery****Vendor API**

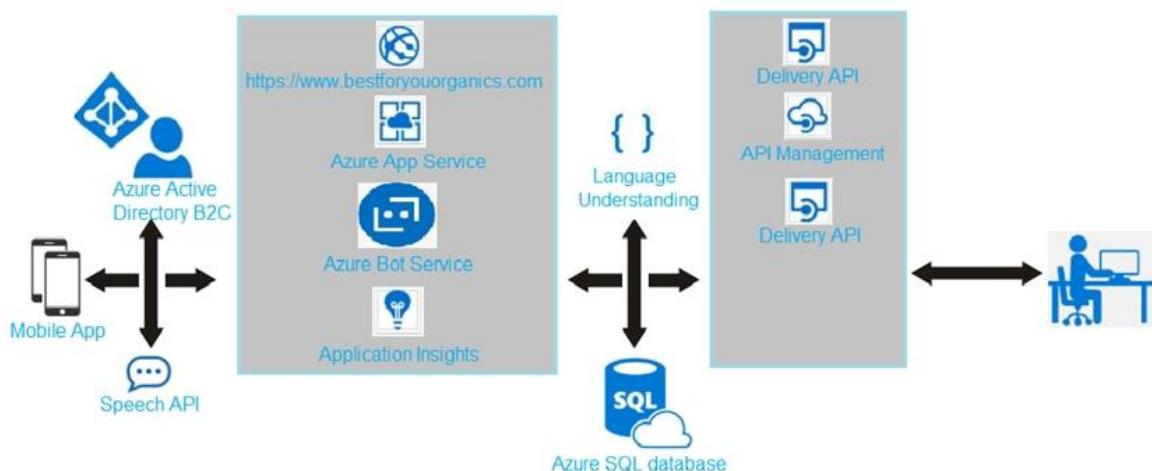
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

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

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

You need to resolve the delivery API error.

What should you do?

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

Answer: A

Explanation:

Scenario: The Delivery API intermittently throws the following exception:

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

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

```
services.AddDbContext< DbContext > ( options =>
options.UseSqlServer ( 'yourconnectionstring' , ...sqlServerOptionsAction: sqlOptions =>
...
.....sqlOptions.EnableRetryOnFailure(
.....maxRetryCount: 5,
.....maxRetryDelay: TimeSpan.FromSeconds ( 30 ),
.....errorNumbersToAdd: new List< int > ( { 19 } ) );
...));
In Retry logic, error 19 is not included. So you have to pass the error code 19 to set retry logic for
error code 19.
```

References:

<https://stackoverflow.com/questions/47558062/error-19-physical-connection-error/47559967>

QUESTION 22**Case Study 4 - Best for You Organics****Background**

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala

Order Pickup

Order Delivery

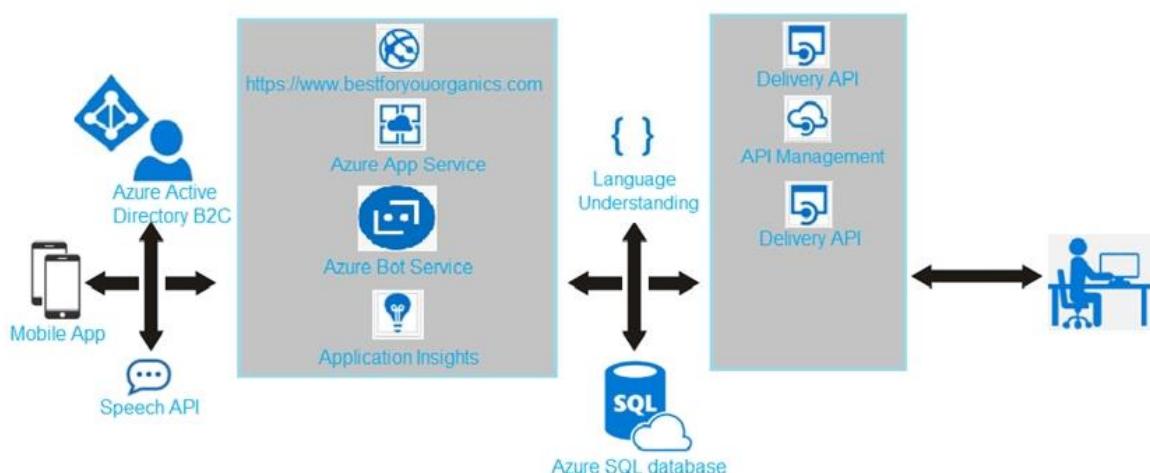
Vendor API

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

APIs must meet the following conditions:

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

You need to implement the purchase requirement.

What should you do?

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

Answer: A

Explanation:

Scenario: Enable users to place an order for delivery or pickup by using their voice. You must develop a chatbot by using the Bot Builder SDK and Language Understanding Intelligence Service (LUIS). The chatbot must allow users to order food for pickup or delivery. The Bot Framework REST APIs enable you to build bots that exchange messages with channels configured in the Bot Framework Portal, store and retrieve state data, and connect your own client applications to your bots. All Bot Framework services use industry-standard REST and JSON over HTTPS.

The Speech Service API is used to recognize intents.

References:

<https://docs.microsoft.com/en-us/azure/bot-service/rest-api/bot-framework-rest-connector-concepts?view=azure-bot-service-4.0>

<https://docs.microsoft.com/en-us/azure/cognitive-services/speech-service/how-to-recognize-intents-from-speech-cpp>

QUESTION 23

Case Study 4 - Best for You Organics

Background

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala**Order Pickup****Order Delivery****Vendor API**

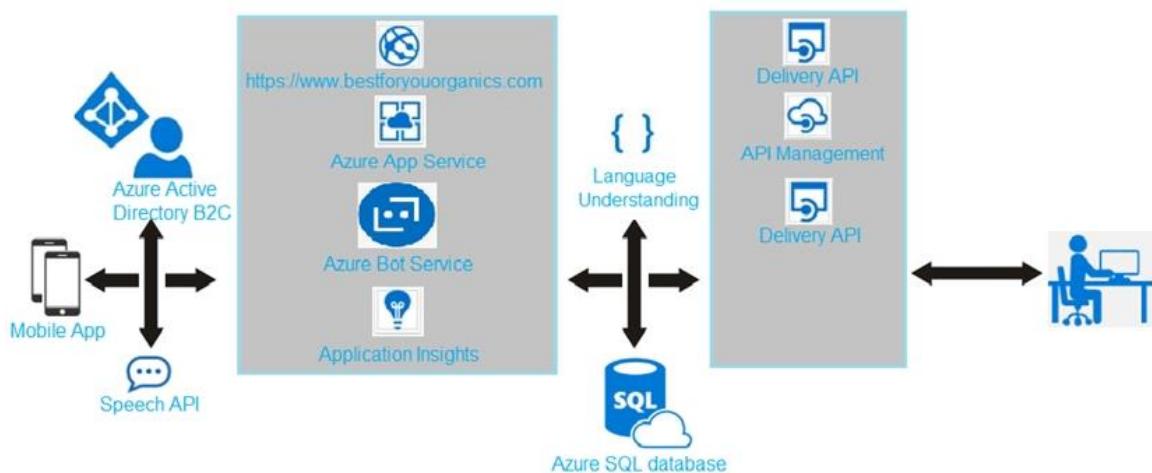
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

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

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

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

You need to debug the user greeting issue.

What should you use?

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

Answer: B

Explanation:

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

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

References:

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

QUESTION 24

Case Study 4 - Best for You Organics

Background

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

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

Requirements

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

The chatbot must meet the following requirements:

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

Welcome to the Restaurant!

**John Doe**

Sun, Aug 26, 2018

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

Specials: Chicken Marsala

Order Pickup

Order Delivery

Vendor API

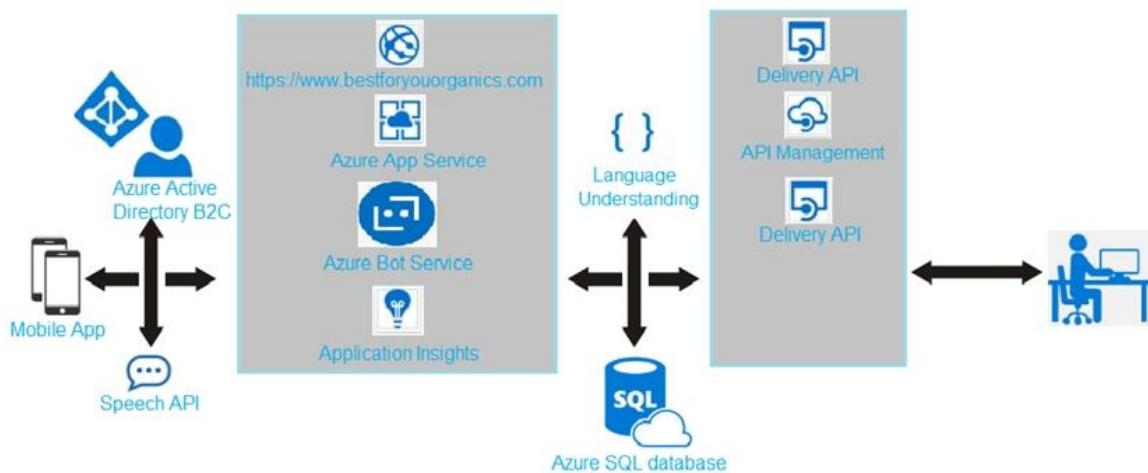
Vendors receive and provide updates for the restaurant inventory and delivery services by using Azure API Management hosted APIs. Each vendor uses their own subscription to access each of the APIs.

APIs must meet the following conditions:

- API usage must not exceed 5,000 calls and 50,000 kilobytes of bandwidth per hour per vendor.
- If a vendor is nearing the number of calls or bandwidth limit, the API must trigger email notifications to the vendor.
- APIs must prevent API usage spikes on a per-subscription basis by limiting the call rate to 100 calls per minute.
- The Inventory API must be written by using ASP.NET Core and Node.js.
- The API must be updated to provide an interface to Azure SQL Database. Database objects must be managed by using code.

- The Delivery API must be protected by using the OAuth 2.0 protocol with Azure Active Directory (Azure AD) when called from the Azure web app. You register the Delivery API and web app in Azure AD. You enable OAuth 2.0 in the web app.
- The delivery API must update the Products table, the Vendor transactions table, and the Billing table in a single transaction.

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



Delivery API

The Delivery API intermittently throws the following exception:

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

Chatbot greeting

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

Language processing

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

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

Startup.cs

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

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

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

NOTE: Each correct selection is worth one point.

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

Answer: AD

Explanation:

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

Welcome to the Restaurant!



John Doe

Sun, Aug 26, 2018

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

Specials: Chicken Marsala

Order Pickup

Order Delivery

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

The Bot Framework currently supports eight types of rich cards:

Thumbnail Card. A card that typically contains a single thumbnail image, one or more buttons, and text.

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

Incorrect Answers:

B: Animation Card. A card that can play animated GIFs or short videos.

C Hero Card. A card that typically contains a single large image, one or more buttons, and text.

E: Adaptive Card. A customizable card that can contain any combination of text, speech, images, buttons, and input fields.

Note:

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

Video Card. A card that can play videos.

References:

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

QUESTION 25

Hotspot Question

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

Customer data exists in all databases.

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

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

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

Which actions and Azure PowerShell command should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

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

Answer:**Answer Area**

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

Explanation:

Box 1: Use a transaction scope

The TransactionScope class establishes an ambient transaction in .NET. (An "ambient transaction" is one that lives in the current thread.) All connections opened within the TransactionScope participate in the transaction. If different databases participate, the transaction is automatically elevated to a distributed transaction.

Box 2: New-AzureRmSqlServerCommunicationLink

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

References:

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

QUESTION 26

Hotspot Question

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

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

You need to complete the configuration.

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

NOTE: Each correct solution is worth one point.

Answer Area

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

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

    Console.WriteLine ("Finished.");
}

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

Answer:

Answer Area

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

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

    Console.WriteLine ("Finished.");
}

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

Explanation:

Box 1: ProcessAsync();

Box 2: await cloudBlobContainer.CreatAsync();

If you specify that a method is an `async` method by using the `async` modifier, you enable the following two capabilities.

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

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

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

References:

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

QUESTION 27

Drag and Drop Question

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

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

You need to implement optimistic concurrency for uploading images.

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

NOTE: Each correct solution is worth one point.

Actions**Answer Area**

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

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

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



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

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

Answer:**Actions****Answer Area**

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

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

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



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

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

Explanation:

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

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

References:

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

QUESTION 28

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

You need to develop an Azure Function that ingests data from the modified workbook and places

it into a Microsoft Word document.

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

NOTE: Each correct selection is worth one point.

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

Answer: BC

Explanation:

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

References:

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

QUESTION 29

Drag and Drop Question

You have an Azure subscription.

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

| Variable name | Usage |
|----------------------|-----------------|
| \$resourceGroupName | files |
| \$storageAccountName | filestorage1103 |
| \$fileShareName | userfiles |

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

Actions

```
az storage share create  
--name $fileShareName  
--account -name $acct  
--account -key $key  
--quota 2048
```

```
$acct=$(az storage account create --resource-  
group $resourceGroupName  
--name $storageAccountName  
--location eastus  
--sku Standard_LRS  
--query "name" | tr -d '\n')
```

```
az group create  
--name $resourceGroupName  
--location eastus
```

```
$key=$(az storage account keys list  
--resource-group $resourceGroupName  
--account-name $acct  
--query "[0].value" | tr -d '\n')
```

Answer Area**Answer:****Actions****Answer Area**

```
az group create  
--name $resourceGroupName  
--location eastus
```

```
$acct=$(az storage account create --resource-  
group $resourceGroupName  
--name $storageAccountName  
--location eastus  
--sku Standard_LRS  
--query "name" | tr -d '\n')
```



```
$key=$(az storage account keys list  
--resource-group $resourceGroupName  
--account-name $acct  
--query "[0].value" | tr -d '\n')
```

```
az storage share create  
--name $fileShareName  
--account -name $acct  
--account -key $key  
--quota 2048
```

Explanation:**Step 1:**

Create a resource group

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

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

```
az group create --name myResourceGroup --location eastus
```

Step 2:

Create a storage account

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

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

Step 3:

Get the storage account key

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

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

Step 4:

Now, you can create your Azure file share. Create file shares by using the az storage share create command. This example creates an Azure file share named myshare:

```
az storage share create \
--account-name $STORAGEACCT \
--account-key $STORAGEKEY \
--name "myshare"
```

References:

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

QUESTION 30

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

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

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

NOTE: Each correct selection is worth one point.

- A. Navigate to the following URL:
`PUT https://management.azure.com/subscriptions/00000000-1111-2222-3333-444444444444
/resourceGroups/rg01/providers/Microsoft.Sql/servers/server01/databases/customers
/transparentDataEncryption/current?api-version=2014-04-01`
- B. Exclude the managers group.
- C. Include the managers group.
- D. Run the following Azure PowerShell command:
`New-AzureRmSqlDatabaseDataMaskingRule -SchemaName "dbo" -TableName "customers" '
-ColumnName "ssn" -MaskingFunction "Default"`
- E. Exclude the administrators group.

Answer: BD

Explanation:

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

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

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

References:

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

QUESTION 31

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

The app stores sensitive information about users.

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

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

NOTE: Each correct selection is worth one point.

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

Answer: ABD

Explanation:

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

The screenshot shows the Microsoft Azure Dynamic Data Masking portal. At the top, it displays "Dynamic Data Masking" and "demo_database". Below the header are three buttons: "Save" (with a blue square icon), "Discard" (with a red cross icon), and "Add Mask" (with a plus sign icon). A message bar indicates that downlevel clients require the use of Security Enabled Connection Strings, with a checkbox and a link icon.

Masking Rules

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

SQL users excluded from masking (administrators are always excluded) ⓘ

| | |
|--|---|
| SQL users excluded from masking (administrators are always excluded) | ✓ |
|--|---|

Recommended fields to mask

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

References:

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

QUESTION 32

Hotspot Question

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

```
01 public static async Task QueueItemReceived(string queueItem, ILogger log)
02 {
03     CloudStorageAccount storageAccount = CloudStorageAccount.Parse(ConnectionString);
04     CloudFileClient fileClient = storageAccount.CreateCloudFileClient();
05     CloudFileShare share = fileClient.GetShareReference(fileContainerName);
06     if (await share.ExistsAsync())
07     {
08         var workItem = JsonConvert.DeserializeObject<WorkItem>(queueItem);
09         CloudFileDirectory rootDir = share.GetRootDirectoryReference();
10         CloudFileDirectory importDir = rootDir.GetDirectoryReference("import");
11         CloudFile file = importDir.GetFileReference(workItem.FileName);
12         if (!file.Metadata.ContainsKey(workItem.AttributeName))
13         {
14             file.Metadata.Add(workItem.AttributeName, ${workItem.AttributeValue});
15             file.Metadata.Add("Timestamp", ${DateTime.UtcNow.ToString("O")});
16             ProcessFile(workItem);
17         }
18     }
19     else
20     {
21         log.LogInformation($"Metadata key {workItem.AttributeName} already exists on item {workItem.FileName}");
22     }
23     await file.SetMetadataAsync();
24     log.LogInformation($"processed: {queueItem}");
25 }
26 public class WorkItem
27 {
28     public string FileName {get; set;}
29     public string AttributeName {get; set;}
30     public stringAttributeValue {get; set;}
31 }
```

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

NOTE: Each correct selection is worth one point.

Answer Area

- | | Yes | No |
|---|-----------------------|-----------------------|
| The code will properly set and save metadata using <code>AttributeName</code> as the metadata key, and <code>AttributeValue</code> as the metadata value | <input type="radio"/> | <input type="radio"/> |
| If metadata with the same key exists on the file in Azure, the function will properly load the metadata and only log an error, keeping the existing value and timestamp on the file in Azure. | <input type="radio"/> | <input type="radio"/> |
| Changing line 12 to the following causes the code on line 19 to run when Azure already contains a file with a matching name, and metadata with a key matching <code>workItem.AttributeName</code> | <input type="radio"/> | <input type="radio"/> |

```
if (await file.ExistsAsync() &&
!file.Metadata.ContainsKey(workItem.AttributeName))
```

Answer:**Answer Area**

| | Yes | No |
|---|----------------------------------|----------------------------------|
| The code will properly set and save metadata using <code>AttributeName</code> as the metadata key, and <code>AttributeValue</code> as the metadata value | <input checked="" type="radio"/> | <input type="radio"/> |
| If metadata with the same key exists on the file in Azure, the function will properly load the metadata and only log an error, keeping the existing value and timestamp on the file in Azure. | <input type="radio"/> | <input checked="" type="radio"/> |
| Changing line 12 to the following causes the code on line 19 to run when Azure already contains a file with a matching name, and metadata with a key matching <code>workItem.AttributeName</code> | <input checked="" type="radio"/> | <input type="radio"/> |

```
if (await file.ExistsAsync() &&
!file.Metadata.ContainsKey(workItem.AttributeName))
```

Explanation:

Box 1: Yes

The following code example sets metadata on a container. One value is set using the collection's Add method.

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

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

Box 2: No

It would not load the metadata

Box 3: Yes

QUESTION 33

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

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

What should you do?

- A. In a .NET method, call the method: `BatchClient.PoolOperations.CreateJob`
- B. In Python, implement the class: `JobAddParameter`
- C. In the Azure portal, create a Batch account.
- D. In Azure CLI, run the command: `az batch pool create`

Answer: A**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.

Note:

Step 1: Create a pool of compute nodes. When you create a pool, you specify the number of compute nodes for the pool, their size, and the operating system. When each task in your job runs, it's assigned to execute on one of the nodes in your pool.

Step 2 : Create a job. A job manages a collection of tasks. You associate each job to a specific pool where that job's tasks will run.

Step 3: Add tasks to the job. Each task runs the application or script that you uploaded to process the data files it downloads from your Storage account. As each task completes, it can upload its output to Azure Storage.

References:

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

QUESTION 34

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

You need to optimize the Dockerfile.

What should you do?

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

Answer: C

Explanation:

Minimize the number of layers.

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

References:

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

QUESTION 35

Hotspot Question

You are developing a workflow solution using Azure technologies.

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

NOTE: Each correct selection is worth one point.

Answer Area

| Requirement | Tool |
|--|---|
| Debug the solution by using Visual Studio. | Durable functions only Logic Apps only Durable functions and Logic Apps |
| Use a collection of ready-made actions. | Durable functions only Logic Apps only Durable functions and Logic Apps |
| Deploy the component by using Visual Studio Team Services. | Durable functions only Logic Apps only Durable functions and Logic Apps |

Answer:**Answer Area**

| Requirement | Tool |
|--|--|
| Debug the solution by using Visual Studio. | Durable functions only Logic Apps only Durable functions and Logic Apps |
| Use a collection of ready-made actions. | Durable functions only Logic Apps only Durable functions and Logic Apps |
| Deploy the component by using Visual Studio Team Services. | Durable functions only Logic Apps only Durable functions and Logic Apps |

Explanation:

Box 1: Logic Apps only

You can manually trigger a logic app deployed in Azure from Visual Studio. On the Logic App Designer toolbar, choose Run Trigger.

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

Box 2: Durable functions only

Box 3: Durable functions and Logic Apps

References:

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

QUESTION 36

Hotspot Question

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

```
01 public class MessageController : ApiController
02 {
03     public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
04 {
05     if (activity.GetActivityType() == ActivityTypes.Message)
06     {
07         await Conversation.SendAsync(activity, () => new Dialogs.RootDialog());
08     }
09     else
10     {
11         HandleSystemMessage(activity);
12     }
13     var response = Request.CreateResponse(HttpStatusCode.OK);
14     return response;
15 }
16 }
17 [Serializable]
18 public class RootDialog : IDialog<object>
19 {
20     public class RootDialog : IDialog<object>
21     {
22         context.Wait(MessageReceivedAsync);
23         return Task.CompletedTask;
24     }
25     public virtual async Task MessageReceivedAsync (IDialogContext context, IAwaitable<IMessageActivity> result)
26     {
27         var message = await result;
28         if (message.Text.ToLower().Contains("help") || message.Text.ToLower().Contains ("support"))
29         {
30             await context.Forward(new SupportDialog(), this.ResumeAfterSupportDialog, message);
31         }
32         else
33         {
34             await context.PostAsync($"Hello. I can help you with the following keywords : help | support");
35             context.Wait (MessageReceivedAsync);
36         }
37     }
38     private async Task ResumeAfterSupportDialog(IDialogContext context, IAwaitable<object> result)
39     {
40         try
41         {
42             var message = await result;
43         }
44         finally
45         {
46             context.Wait(this.MessageReceivedAsync);
47         }
48     }
49 }
```

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

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

NOTE: Each correct selection is worth one point.

Answer Area**Yes** **No**

A child dialog will handle any messages that contain the word **support**.

RootDialog will return to the top of the stack after SupportDialog handles a request

Answer:**Answer Area****Yes** **No**

A child dialog will handle any messages that contain the word **support**.

RootDialog will return to the top of the stack after SupportDialog handles a request

QUESTION 37

Drag and Drop Question

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

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

How should you configure the Logic App? To answer, drag the appropriate triggers or action blocks to the correct trigger or action slots. Each trigger or action block may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

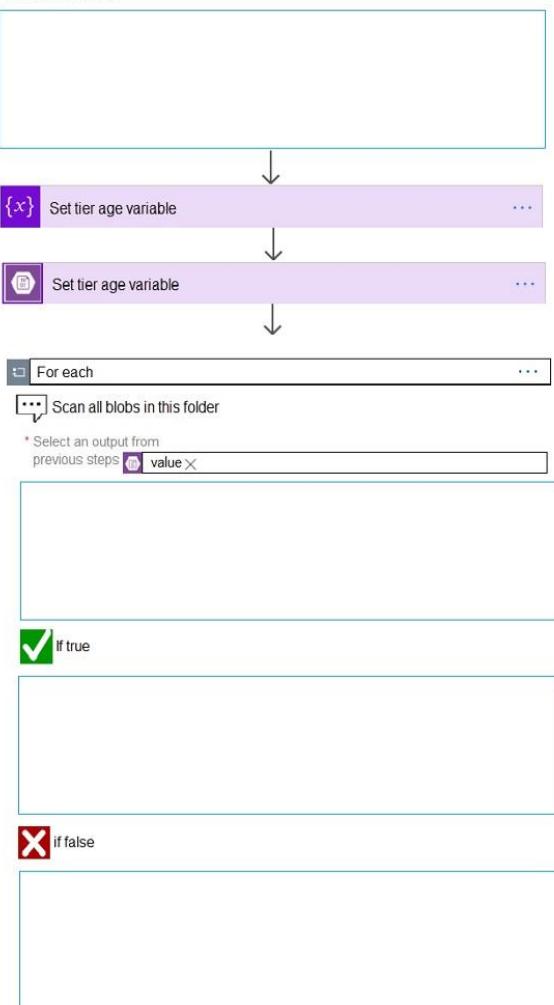
NOTE: Each correct selection is worth one point.

Triggers and Action Blocks

The screenshot shows the Logic App designer interface with several components:

- Insert Entity**: Triggered by a Table entity named "processing". It has a Path input.
- Tier blob**: Action block that checks if a blob is older than a defined value and tiers it to Cool or Archive tier. It takes Blob path and Blob Tier as inputs.
- When there are messages in a queue**: Triggered by a Queue named "processing". It connects to "tableStorageAccountConnection".
- Recurrence**: Action block that runs every month.

Answer Area



Answer:

Triggers and Action Blocks

The screenshot shows the Azure Logic App designer interface. A trigger named "When there are messages in a queue" is connected to a "Recurrence" action. The "Recurrence" action has an "Interval" of 1 and a "Frequency" of Month. Below these, there is a "Set tier age variable" action, followed by a "List blobs" action. This is followed by a "For each" loop action with the condition "Scan all blobs in this folder". Inside the loop, there is an "Insert Entity" action and an "If true" condition block. The "If true" block contains a "Tier blob" action. There are also "Add an action" buttons at the bottom.

Answer Area



Explanation:

Box 1: Recurrence

Box 2: Insert Entity

Box 3 (if true): Tier Blob

Box 4: (if false):

Leave blank.

References:

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

QUESTION 38

Hotspot Question

You develop a microservice solution for a company.

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

You need to implement code for the GPS system.

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

NOTE: Each correct selection is worth one point.

Answer Area

```
public interface IGPS_System : IActor
{
    Task<int> GetAccountAsync(CancellationToken cancellationToken);
    Task SetCountAsync (int count, CancellationToken cancellationToken);
}
[StatePersistence (StatePersistence.
```

| |
|-----------|
| Persisted |
| Volatile |
| None |
| Actor |

```
internal class GPS_System : 
```

| |
|---------------|
| Actor, |
| IActor, |
| ActorId, |
| ActorService, |

| |
|---------------|
| Actor, |
| IActor, |
| IGPS_System |
| ActorService, |

```

{
    public GPS_System(ActorService p_pService, ActorId p_pID) " base(p_pService,
p_pID)
    {
    }
...
}
```

Answer:

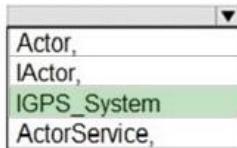
Answer Area

```
public interface IGPS_System : IActor
{
    Task<int> GetAccountAsync(CancellationToken cancellationToken);
    Task SetCountAsync (int count, CancellationToken cancellationToken);
}
[StatePersistence (StatePersistence.
```



```

internal class GPS_System : Actor,
```



```

{
    public GPS_System(ActorService p_pService, ActorId p_pID) : base(p_pService,
p_pID)
    {
    }
...
}
```

Explanation:**Example:**

The state manager retrieval methods return a reference to an object in local memory. Modifying this object in local memory alone does not cause it to be saved durably. When an object is retrieved from the state manager and modified, it must be reinserted into the state manager to be saved durably.

```
[StatePersistence(StatePersistence.Persisted)]
class MyActor : Actor, IMyActor
{
    public MyActor(ActorService actorService, ActorId actorId)
        : base(actorService, actorId)
    {
    }
```

```
    public Task SetCountAsync(int value)
    {
        return this.StateManager.SetStateAsync<int>("MyState", value);
    }
}
```

References:

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-actors-access-save-remove-state>

QUESTION 39

You host an on-premises ASP.NET Web API at the company headquarters.

The Web API is consumed by applications running at the company's branch offices using the Azure Relay service.

All the users of the applications are on the same Azure Active Directory (Azure AD).

You need to ensure that the applications can consume the Web API.

What should you do?

- A. Create a Shared Access policy for Send permissions and another for Receive permissions.
- B. Create separate Azure AD groups named Senders and Receivers. In Access Control (IAM) for the Relay namespace, assign Senders the Reader role and assign Receivers the Reader role.
- C. Create dedicated Azure AD identities named Sender and Receiver. Assign Sender the Azure AD Identity Reader role. Assign Receiver the Azure AD Identity Reader role. Configure applications to use the respective identities.
- D. Create a Shared Access policy for the namespace. Use a connection string in Web API and use a different connection string in consumer applications.

Answer: D

Explanation:

To begin using Service Bus messaging entities in Azure, you must first create a namespace with a name that is unique across Azure. A namespace provides a scoping container for addressing Service Bus resources within your application.

When you publish an application through Azure Active Directory Application Proxy, you create an external URL for your users to go to when they're working remotely. This URL gets the default domain `yourtenant.msappproxy.net`.

References:

<https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy-configure-custom-domain>

QUESTION 40

Drag and Drop Question

You are developing a .NET Core model-view controller (MVC) application hosted on Azure for a health care system that allows providers access to their information.

You develop the following code:

```
services.AddAuthorization (options =>
{
    options.AddPolicy("ProviderPartner", policy =>
    {
        policy.AddAuthenticationSchemes("Cookie, Bearer");
        policy.RequireAuthenticatedUser();
        policy.RequireRole("ProviderAdmin", "SysAdmin");
        policy.RequireClaim("editor", "partner");
    });
})
```

You define a role named SysAdmin.

You need to ensure that the application meets the following authorization requirements:

- Allow the ProviderAdmin and SysAdmin roles access to the Partner controller regardless of whether the user holds an editor claim of partner.

- Limit access to the Manage action of the controller to users with an editor claim of partner who are also members of the SysAdmin role.

How should you complete the code? To answer, drag the appropriate code segments to the correct locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

| Code Segments | Answer Area |
|---|-------------|
| [Authorize (Policy = "ProviderEditor")] [Authorize(Role = "SysAdmin")] | |
| [Authorize(Role = "ProviderAdmin")] [Authorize(Role = "SysAdmin")] | |
| [Authorize(Role = "SysAdmin", "ProviderAdmin")] | |
| [Authorize(Policy = "ProviderEditor", Role= "SysAdmin")] | |

public class PartnerController : Controller
{
 . . .
}

Public ActionResult Manage()
{
 . . .
}

Answer:

| Code Segments | Answer Area |
|---|---|
| [Authorize (Policy = "ProviderEditor")] [Authorize(Role = "SysAdmin")] | [Authorize(Role = "ProviderAdmin")] [Authorize(Role = "SysAdmin")] |
| [Authorize(Role = "SysAdmin", "ProviderAdmin")] | |
| [Authorize(Policy = "ProviderEditor", Role= "SysAdmin")] | public class PartnerController : Controller { . . . } |

Public ActionResult Manage()
{
 . . .
}

Explanation:

Box 1:

Allow the ProviderAdmin and SysAdmin roles access to the Partner controller regardless of whether the user holds an editor claim of partner.

Box 2:

Limit access to the Manage action of the controller to users with an editor claim of partner who are also members of the SysAdmin role.

QUESTION 41

You have implemented code that uses elastic transactions spanning across three different Azure SQL Database logical servers.

Database administration report that some transactions take longer to complete than expected.

You need to use the correct tool to monitor all the transactions originating from the elastic transaction implementation.

Which tool should you use?

- A. Use the dependencies section of Azure Applications Insights.
- B. Run the sys.dm_tran_active_transactions dynamic management view.
- C. Run the sys.dm_tran_current_snapshot dynamic management view.
- D. Run the sys.dm_tran_active_snapshot_database_transactions dynamic management view.

Answer: B

Explanation:

Use Dynamic Management Views (DMVs) in SQL DB to monitor status and progress of your ongoing elastic database transactions.

These DMVs are particularly useful:

sys.dm_tran_active_transactions: Lists currently active transactions and their status. The UOW (Unit Of Work) column can identify the different child transactions that belong to the same distributed transaction. All transactions within the same distributed transaction carry the same UOW value.

sys.dm_tran_database_transactions: Provides additional information about transactions, such as placement of the transaction in the log.

sys.dm_tran_locks: Provides information about the locks that are currently held by ongoing transactions

References:
<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-transactions-overview>

QUESTION 42

A company is migrating an existing on-premises third-party website to Azure. The website is stateless.

The company does not have access to the source code for the website. They do have the original installer.

The number of visitors at the website varies throughout the year. The on-premises infrastructure was resized to accommodate peaks but the extra capacity was not used.

You need to implement a virtual machine scale set instance.

What should you do?

- A. Use an autoscale setting to scale instances vertically.
- B. Use an autoscale setting with unlimited maximum number of instances.
- C. Scale out by one instance when the average CPU usage of one of the instance is over 80 percent.
- D. Use Azure Monitor to create autoscale settings using custom metrics.

Answer: D

Explanation:

Azure Monitor autoscaling allows you to scale the number of running instances up or down, based on telemetry data (metrics).

References:

<https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-autoscale-common-metrics>

QUESTION 43

You are developing an Azure Batch solution to perform CPU intensive calculations. The calculations occur at a specific time each week and last for approximately one hour.

Before any changes are made, a timer must be created to measure the initial duration. The timer

must start before the first calculation is queued to run on the computer node.

You need to implement the timer.

Before which line should the timer be created?

- A. CloudJob = batchClient.JobOperations.CreateJob();
- B. batchClient.JobOperations.AddTask(JobId, tasks);
- C. batchClient.Utilities.CreateTaskStateMonitor().WaitAll(addedTasks, TaskState.Completed, timeout);
- D. using (BatchClient batchClient = BatchClient.Open(cred))

Answer: B

Explanation:

References:

<https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet>

QUESTION 44

Drag and Drop Question

You develop a web app that uses the tier D1 app service plan by using the Web Apps feature of Microsoft Azure App Service.

Spikes in traffic have caused increases in page load times.

You need to ensure that the web app automatically scales when CPU load is about 85 percent and minimize costs.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

NOTE: Each correct selection is worth one point.

| Actions | Answer Area |
|---|---|
| Enable autoscaling on the web app. | |
| Configure a Scale condition. | |
| Configure the web app to the Standard App Service tier. |  |
| Configure the web app to the Premium App Service tier. |  |
| Switch to an Azure App Services consumption plan. | |
| Add a Scale rule. | |

Answer:

| Actions | Answer Area |
|--|---|
| | Configure the web app to the Standard App Service tier. |
| | Enable autoscaling on the web app. |
| |  Add a Scale rule. |
| Configure the web app to the Premium App Service tier. |  Configure a Scale condition. |
| |   |
| Switch to an Azure App Services consumption plan. | |

Explanation:

Step 1: Configure the web app to the Standard App Service Tier
The Standard tier supports auto-scaling, and we should minimize the cost.

Step 2: Enable autoscaling on the web app
First enable autoscale

Step 3: Add a scale rule
Step 4: Add a Scale condidation

References:

<https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-autoscale-get-started>
<https://azure.microsoft.com/en-us/pricing/details/app-service/plans/>

QUESTION 45

Drag and Drop Question

You are developing a multi-tenant ASP.NET Core application that will be hosted on Azure. The application will support multiple database platforms, including Azure SQL and on-premises SQL Server instances.

You need to ensure the application supports distributed transactions.

Which technologies should you use? To answer, drag the appropriate technologies to the correct scenarios. Each technology may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

| Technologies | Answer Area | |
|---------------------------------|--------------------------------------|------------|
| Elastic Database Client Library | Platform | Technology |
| SQLite | Sharded Azure SQL Database instances | |
| MSDTC | On-premises SQL Server databases | |
| SQLCMD | | |

Answer:

Technologies**Answer Area**

| Platform | Technology |
|----------|---|
| SQLite | Sharded Azure SQL Database instances Elastic Database Client Library |
| SQLCMD | On-premises SQL Server databases MSDTC |

Explanation:

Box 1: Elastic Database Client Library

Sharded database applications in Azure: With this scenario, the data tier uses the Elastic Database client library or self-sharding to horizontally partition the data across many databases in SQL DB.

Box 2: MSDTC

Elastic database transactions cannot stretch across on premises SQL Server and Azure SQL Database. For distributed transactions on premises, continue to use MSDTC.

References:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-transactions-overview>

QUESTION 46

Drag and Drop Question

You are implementing an order processing system. A point of sale application publishes orders to topics in an Azure Service Bus queue. The label property for the topic includes the following data:

| Property | Description |
|---------------|--|
| ShipLocation | the country/region where the order will be shipped |
| CorrelationId | a priority value for the order |
| Quantity | a user-defined field that stores the quantity of items in an order |
| AuditedAt | a user-defined field that records the date an order is audited |

The system has the following requirements for subscriptions:

| Subscription type | Comments |
|---------------------|---|
| FutureOrders | This subscription is reserved for future use and must not receive any orders. |
| HighPriorityOrders | Handle all high priority orders and International orders. |
| InternationalOrders | Handle orders where the country/region is not United States. |
| HighQuantityOrders | Handle only orders with quantities greater than 100 units. |
| AllOrders | This subscription is used for auditing purposes. This subscription must receive every single order. AllOrders has an Action defined that updates the AuditedAt property to include the date and time it was received by the subscription. |

You need to implement filtering and maximize throughput while evaluating filters.

Which filter types should you implement? To answer, drag the appropriate filter types to the correct subscriptions. Each filter type may be used once, more than once, or not at all. You may

need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Filter types Answer Area

| Filter types | Subscription | Filter type |
|-------------------|---------------------|-------------|
| SQLFilter | FutureOrders | |
| CorrelationFilter | HighPriorityOrders | |
| No Filter | InternationalOrders | |
| | HighQuantityOrders | |
| | AllOrders | |

Answer:

Filter types Answer Area

| Filter types | Subscription | Filter type |
|-------------------|---------------------|-------------------|
| SQLFilter | FutureOrders | SQLFilter |
| CorrelationFilter | HighPriorityOrders | CorrelationFilter |
| No Filter | InternationalOrders | SQLFilter |
| | HighQuantityOrders | SQLFilter |
| | AllOrders | No Filter |

Explanation:

FutureOrders: SQLFilter

HighPriorityOrders: CorrelationFilter

CorrelationID only

InternationalOrders: SQLFilter

Country NOT USA requires an SQL Filter

HighQuantityOrders: SQLFilter

Need to use relational operators so an SQL Filter is needed.

AllOrders: No Filter

SQL Filter: SQL Filters - A SqlFilter holds a SQL-like conditional expression that is evaluated in

the broker against the arriving messages' user-defined properties and system properties. All system properties must be prefixed with sys. in the conditional expression. The SQL-language subset for filter conditions tests for the existence of properties (EXISTS), as well as for null-values (IS NULL), logical NOT/AND/OR, relational operators, simple numeric arithmetic, and simple text pattern matching with LIKE.

Correlation Filters - A CorrelationFilter holds a set of conditions that are matched against one or more of an arriving message's user and system properties. A common use is to match against the CorrelationId property, but the application can also choose to match against ContentType, Label, MessageId, ReplyTo, ReplyToSessionId, SessionId, To, and any user-defined properties. A match exists when an arriving message's value for a property is equal to the value specified in the correlation filter. For string expressions, the comparison is case-sensitive. When specifying multiple match properties, the filter combines them as a logical AND condition, meaning for the filter to match, all conditions must match.

Boolean filters - The TrueFilter and FalseFilter either cause all arriving messages (true) or none of the arriving messages (false) to be selected for the subscription.

References:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/topic-filters>

QUESTION 47

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

Routing Service - Routes a request to the appropriate service and must not persist data across sessions

Account Service - Stores and manages all account information and authentication and requires data to persist across sessions.

User Service - Stores and manages all user information and requires data to persist across sessions.

Housing Network Service - Stores and manages the current real-estate economy and requires data to persist across sessions.

Trade Service - Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Create a Service Fabric Cluster with a stateless Reliable Service for Routing Service. Create stateful Reliable Services for all other components.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

We should use stateful services when we want data to persist, and stateless service when data must not persist.

Note:

For stateful services, the Reliable Services programming model allows you to consistently and reliably store your state right inside your service by using Reliable Collections.

A stateless service is one where there is no state maintained within the service across calls. Any state that is present is entirely disposable and doesn't require synchronization, replication, persistence, or high availability.

References:

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction>

QUESTION 48

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

Routing Service - Routes a request to the appropriate service and must not persist data across sessions

Account Service - Stores and manages all account information and authentication and requires data to persist across sessions.

User Service - Stores and manages all user information and requires data to persist across sessions.

Housing Network Service - Stores and manages the current real-estate economy and requires data to persist across sessions.

Trade Service - Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Create a Service Fabric Cluster with a stateful Reliable Service for Routing Service. Deploy a Guest Executable to Service Fabric for each component.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B**Explanation:**

We should use stateful services when we want data to persist, and stateless service when data must not persist.

References:

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction>

QUESTION 49

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You develop an entertainment application where users can buy and trade virtual real estate. The application must scale to support thousands of users.

The current architecture includes five Azure Virtual Machines (VM) that connect to an Azure SQL Database for account information and Azure Table Storage for backend services. A user interacts with these components in the cloud at any given time

Routing Service - Routes a request to the appropriate service and must not persist data across sessions

Account Service - Stores and manages all account information and authentication and requires data to persist across sessions.

User Service - Stores and manages all user information and requires data to persist across sessions.

Housing Network Service - Stores and manages the current real-estate economy and requires data to persist across sessions.

Trade Service - Stores and manages virtual trade between accounts and requires data to persist across sessions.

Due to volatile user traffic, a microservices solution is selected for scale agility.

You need to migrate to a distributed microservices solution on Azure Service Fabric.

Solution: Deploy a Windows container to Azure Service Fabric for each component.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

We should use stateful services when we want data to persist, and stateless service when data must not persist.

References:

<https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-reliable-services-introduction>

QUESTION 50

You are developing an Azure Durable Function instance. You need to add a delay by using a durable timer.

What type of function should you use?

- A. Activity
- B. Orchestrator
- C. Client
- D. Webhook

Answer: B

Explanation:

Durable Functions provides durable timers for use in orchestrator functions to implement delays or to set up timeouts on async actions.

References:

<https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-overview>

QUESTION 51

A company sells products worldwide and provides customer service in many languages. The company has a customer service email address for customer requests.

The language of the email is written in needs to be recognized and routed to the appropriate local language department.

You need to use the appropriate cognitive service to detect the language of the email.

How should you initiate language detection?

- A. Use the Spark natural language processing functionality on Azure HDInsight.
- B. Implement the DetectLanguageAsync method of the TextAnalyticsClient object.
- C. Use the RecognizeAsync method of the SpeechRecognizer object.
- D. Pass the content as a query parameter to <https://api.cognitive.microsoft.com/bing/v7.0/search>.

Answer: B**Explanation:**

The DetectLanguageAsync(String) method detects the language of a text.

Incorrect Answers:

A: the Spark natural language processing functionality on Azure HDInsight Doesn't support language detection.

C: The RecognizeAsync method of the SpeechRecognizer object is used for speech recognition.

D: The request URL to <https://api.cognitive.microsoft.com/bing/v7.0/search> is used for customized searches.

References:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.toolkit.services.microsofttranslator.translatorservice.detectlanguageasync?view=win-comm-toolkit-dotnet-stable>

QUESTION 52

Hotspot Question

You are developing an SMS-based testing solution. The solution sends users a question by using SMS. Early responders may qualify for prizes.

Users must respond with an answer choice within 90 seconds.

You must be able to track how long it takes each user to respond.

You create a durable Azure Function named SendSmsQuizQuestion that uses Twilio to send messages.

You need to write the code for MessageQuiz.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
[FunctionName("MessageQuiz")]
public static async Task<bool> Run([OrchestrationTrigger] DurableorchestrationContext context)
{
    string phoneNumber = context.GetInput<string>();
    int correctAnswerCode = await context.CallActivityAsync<int>("SendSmsQuizQuestion", phoneNumber);
    using (var cts = new CancellationTokenSource())
    {
        DateTime expiration = DateTime.UtcNow;
        DateTime expiration = DateTime.UtcNow.AddSeconds(90);
        DateTime expiration = DateTime.Now();
        DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);

        var timeoutTask = context.CallActivityAsync<DateTime> ("timeout", expiration);
        var timeoutTask = context.CreateTimer(expiration, cts.Token);
        var timeoutTask = context.WaitForExternalEvent("timeout", 90000);
        var timeoutTask = context.CallSubOrchestratorAsync("timeout", expiration);

        bool isWinner = false
        for (int retryCount = 0; retryCount <= 3; retryCount++)
        {
            Task<int> challengeResponseTask = context.WaitForExternalEvent<int>("SmsQuizResponse");
            Task winner = await Task.WhenAny(challengeResponseTask, timeoutTask);
            if (winner == challengeResponseTask)
            {
                if (challengeResponseTask.Result == correctAnswerCode)
                {
                    isWinner = true;
                    break;
                }
            }
            else
            {
                break;
            }
        }

        if (!timeoutTask.IsCompleted)
        if (!timeoutTask.IsCanceled)
        if (!context.IsReplaying)
        if (!cts.IsCancellationRequested)

    }
    cts.Cancel();
}
return isWinner;
```

Answer:

Answer Area

```
[FunctionName ("MessageQuiz")]
public static async Task<bool> Run([OrchestrationTrigger] DurableorchestrationContext context)
{
    string phoneNumber = context.GetInput<string>();
    int correctAnswerCode = await context.CallActivityAsync<int>("SendSmsQuizQuestion", phoneNumber);
    using (var cts = new CancellationTokenSource())
    {
        DateTime expiration = DateTime.UtcNow;
        DateTime expiration = DateTime.UtcNow.AddSeconds(90);
        DateTime expiration = DateTime.Now();
        DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);

        var timeoutTask = context.CallActivityAsync<DateTime> ("timeout", expiration);
        var timeoutTask = context.CreateTimer(expiration, cts.Token);
        var timeoutTask = context.WaitForExternalEvent("timeout", 90000);
        var timeoutTask = context.CallSubOrchestratorAsync("timeout", expiration);

        bool isWinner = false
        for (int retryCount = 0; retryCount <= 3; retryCount++)
        {
            Task<int> challengeResponseTask = context.WaitForExternalEvent<int>("SmsQuizResponse");
            Task winner = await Task.WhenAny(challengeResponseTask, timeoutTask);
            if (winner == challengeResponseTask)
            {
                if (challengeResponseTask.Result == correctAnswerCode)
                {
                    isWinner = true;
                    break;
                }
            }
            else
            {
                break;
            }
        }

        if (!timeoutTask.IsCompleted)
        if (!timeoutTask.IsCanceled)
        if (!context.IsReplaying)
        if (!cts.IsCancellationRequested)

        {
            cts.Cancel();
        }
        return isWinner;
    }
}
```

Explanation:

Box 1: DateTime expiration = context.CurrentUtcDateTime.AddSeconds(90);

The user has 90 seconds to respond with the code they received in the SMS message.

Box 2: var timeoutTask = context.CreateTimer(expiration, cts.Token);

Create a timer.

Box 3: if (!timeoutTask.IsCompleted)

All pending timers must be complete or canceled before the function exits.

References:

<https://github.com/Azure/azure-functions-durable-extension/blob/master/samples/precompiled/PhoneVerification.cs>

QUESTION 53

Hotspot Question

You are creating an app that uses Event Grid to connect with other services. Your app's event data will be sent to a serverless function that checks compliance.

This function is maintained by your company.

You write a new event subscription at the scope of your resource. The event must be invalidated after a specific period of time.

You need to configure Event Grid to ensure security.

What should you implement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Authentication | Type |
|------------------------|--|
| WebHook event delivery | SAS tokens Key authentication JWT token |
| Topic publishing | ValidationCode handshake ValidationURL handshake Management Access Control |

Answer:

Answer Area

| Authentication | Type |
|------------------------|--|
| WebHook event delivery | SAS tokens Key authentication JWT token |
| Topic publishing | ValidationCode handshake ValidationURL handshake Management Access Control |

Explanation:

Box 1: SAS tokens

Custom topics use either Shared Access Signature (SAS) or key authentication. Microsoft recommends SAS, but key authentication provides simple programming, and is compatible with

many existing webhook publishers.

In this case we need the expiration time provided by SAS tokens.

Box 2: ValidationCode handshake

Event Grid supports two ways of validating the subscription: ValidationCode handshake (programmatic) and ValidationURL handshake (manual).

If you control the source code for your endpoint, this method is recommended.

Incorrect Answers:

ValidationURL handshake (manual): In certain cases, you can't access the source code of the endpoint to implement the ValidationCode handshake. For example, if you use a third-party service (like Zapier or IFTTT), you can't programmatically respond with the validation code.

References:

<https://docs.microsoft.com/en-us/azure/event-grid/security-authentication>

QUESTION 54

You are developing a speech-enabled home automation control bot.

The bot interprets some spoken words incorrectly.

You need to improve the spoken word recognition for the bot.

What should you implement?

- A. The Skype for Business Channel and use scorable dialogs for improving conversation flow
- B. The Skype Channel and Speech priming using a LUIS app
- C. The Cortana Channel and Speech priming using a LUIS app
- D. The Web Chat Channel and use scorable dialogs for improving conversation flow

Answer: C

Explanation:

Speech priming improves the recognition of spoken words and phrases that are commonly used in your bot. For speech-enabled bots that use the Web Chat and Cortana channels, speech priming uses examples specified in Language Understanding (LUIS) apps to improve speech recognition accuracy for important words.

References:

<https://docs.microsoft.com/en-us/azure/bot-service/bot-service-manage-speech-priming?view=azure-bot-service-4.0>

QUESTION 55

A company is creating an IoT solution for connecting to smart refrigerators. You plan to use the Azure IoT Hub Device Provisioning Service for this process.

You need to provision the devices automatically.

Which feature of Device Provisioning Service should you use?

- A. Device registration and configuration
- B. Template registration
- C. Error policy
- D. Device simulation

Answer: A

Explanation:

Azure IoT auto-provisioning can be broken into three phases:

- Service configuration - a one-time configuration of the Azure IoT Hub and IoT Hub Device Provisioning Service instances, establishing them and creating linkage between them.
- Device enrollment - the process of making the Device Provisioning Service instance aware of the devices that will attempt to register in the future. Note: The Device Provisioning Service is a helper service that enables just-in-time provisioning of devices to an IoT hub, without requiring human intervention. After successful provisioning, devices connect directly with their designated IoT Hub. This process is referred to as auto-provisioning, and provides an out-of-the-box registration and initial configuration experience for devices.
- Device registration and configuration - initiated upon boot up by registration software, which is built using a Device Provisioning Service client SDK appropriate for the device and attestation mechanism. The software establishes a connection to the provisioning service for authentication of the device, and subsequent registration in the IoT Hub. Upon successful registration, the device is provided with its IoT Hub unique device ID and connection information, allowing it to pull its initial configuration and begin the telemetry process. In production environments, this phase can occur weeks or months after the previous two phases.

References:

<https://docs.microsoft.com/en-us/azure/iot-dps/concepts-auto-provisioning>

QUESTION 56

You are creating an IoT solution using Azure Time Series Insights.

You configure the environment to ensure that all data for the current year is available.

What should you do?

- A. Create a reference data set.
- B. Add a disaster recovery (DR) strategy.
- C. Change the pricing tier.
- D. Set a value for the Data Retention time setting.

Answer: D

Explanation:

The data is retained in Time Series Insights based on the selected retention days or maximum limits.

Retention is configurable in the Azure portal. The longest allowable retention period is a rolling year of 12 months + 1 month, which is defined as 400 days.

References:

<https://azure.microsoft.com/en-us/pricing/details/time-series-insights/>

QUESTION 57

Member of the finance department for a company review and make changes to a Microsoft Excel workbook that is hosted on OneDrive. The workbook contains projected costs and revenue for a project.

You need to develop an Azure Function that ingests data from the modified workbook and place it into a Microsoft Word document.

Which two objects should you implement? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. An auth token input binding
- B. A group subscription
- C. An Excel table output binding
- D. A group conversation subscription

- E. An Excel table input binding

Answer: BE

QUESTION 58

You are a developer for a SaaS company that offers many web services. All web services for the company must meet the following requirements:

- Use API Management to access the services
- Use OpenID Connect for authentication.
- Prevent anonymous usage

A recent security audit found that several web services can be called without any authentication. Which API Management policy should you implement?

- A. validate-jwt
- B. authentication-certificate
- C. check-header
- D. jsonp

Answer: A

Explanation:

Add the validate-jwt policy to validate the OAuth token for every incoming request.

Incorrect Answers:

B: The jsonp policy adds JSON with padding (JSONP) support to an operation or an API to allow cross-domain calls from JavaScript browser-based clients. JSONP is a method used in JavaScript programs to request data from a server in a different domain. JSONP bypasses the limitation enforced by most web browsers where access to web pages must be in the same domain.

JSONP - Adds JSON with padding (JSONP) support to an operation or an API to allow cross-domain calls from JavaScript browser-based clients.

References:<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad>

QUESTION 59

You are developing a speech-enabled home automation control bot.

The bot interprets some spoken words incorrectly.

You need to improve the spoken word recognition for the bot.

Should you implement?

- A. The Skype Channel and use scorable dialogs for improving conversation flow
- B. The Skype Channel and Speech priming using a LUIS app
- C. The Web Chat Channel and use scorable dialogs for improving conversation flow
- D. The Cortana Channel and Speech priming using a LUIS app

Answer: D

QUESTION 60

Case Study 3 - Proseware, Inc

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody]) string eventsJson
EG05      {
EG06          var events = JArray.Parse(eventsJson);
EG07          foreach (var @event in events)
EG08          {
EG09              EventId.Value = @event["id"].ToString();
EG10              if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG11              {
EG12                  SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG13              }
EG14          }
EG15          EnsureLogging(@event["subject"].ToString());
EG16      }
EG17      return null;
EG18  }
EG19  private void EnsureLogging(string resource)
EG20  {
EG21      . .
EG22  }
EG23  private async Task SendToAnomalyDetectionService(string uri)
EG24  {
EG25      var content = GetLogData(uri);
EG26      var scoreRequest = new
EG27      {
EG28          Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG29          {
EG30              {
EG31                  "input1",
EG32                      new List<Dictionary<string, string>>()
EG33                      {
EG34                          new Dictionary<string, string>()
EG35                          {
EG36                              "logcontent", content
EG37                          }
EG38                      }
EG39                  {
EG40                      "logcontent", content
EG41                  }
EG42              }
EG43          }
EG44      },
EG45      GlobalParameters = new Dictionary<string, string>() { }
EG46  };
EG47  var result = await (new HttpClient()).PostAsJsonAsync("../", scoreRequest);
EG48  var rawModelResult = await result.Content.ReadAsStringAsync();
EG49  var modelResult = JObject.Parse(rawModelResult);
EG50  if (modelResult["notify"].HasValues)
EG51  {
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 }
```

You need to ensure that the solution can meet the scaling requirements for Policy Service. Which Azure Application Insights data model should you use?

- A. an Application Insights metric
- B. an Application Insights dependency
- C. an Application Insights trace
- D. an Application Insights event

Answer: A

Explanation:

Application Insights provides three additional data types for custom telemetry:

Trace - used either directly, or through an adapter to implement diagnostics logging using an instrumentation framework that is familiar to you, such as Log4Net or System.Diagnostics.

Event - typically used to capture user interaction with your service, to analyze usage patterns.
Metric - used to report periodic scalar measurements.

Scenario:

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

References:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/data-model>

QUESTION 61

Drag and Drop Question

You develop an IoT solution by using Node.js. The solution is ready to deploy to the production environment.

You must implement the device twin capabilities of Azure IoT Hub. You must register a sensor named Sensor00. The IoT Hub name is Hub01.

You need to register the endpoint with ContosoHub01 so that you can configure them from your solution.

Which four commands should you use to develop the solution? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

Answer Area

```
npm install azure-iothub  
--save
```

```
npm install azure-iot-device azure-iot-device-  
mqtt  
--save
```

```
az extension add  
--name azure-cli-iot-ext
```

```
az iot hub show-connection-string  
--name {Sensor00}  
--output table
```



```
az iot hub device-identity create  
--hub name {Hub01}  
--device-id Sensor00
```

```
az iot hub show-connection-string  
--name {Hub01}  
--output table
```

```
az iot hub device-identity show-connection-  
string  
--hub-name {Hub01}  
--device-id Sensor00  
--output table
```

Answer:**Answer Area**

```
az extension add  
--name azure-cli-iot-ext
```

```
npm install azure-iot-device azure-iot-device-  
mqtt  
--save
```

```
az iot hub device-identity create  
--hub name {Hub01}  
--device-id Sensor00
```

```
az iot hub show-connection-string  
--name {Sensor00}  
--output table
```



```
npm install azure-iothub  
--save
```

```
az iot hub show-connection-string  
--name {Hub01}  
--output table
```

```
az iot hub device-identity show-connection-  
string  
--hub-name {Hub01}  
--device-id Sensor00  
--output table
```

Explanation:

Step 1: az extension add --name azure-cli-iot-ext

Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension:

```
az extension add --name azure-cli-iot-ext
```

Step 2: az iot hub device-identity create

```
--hub-name {Hub01}
```

```
--device-id Sensor00
```

Create a new device identity called myDeviceId and retrieve the device connection string with these commands:

```
az iot hub device-identity create --device-id myDeviceId --hub-name {Your IoT Hub name} az iot hub device-identity show-connection-string --device-id myDeviceId --hub-name {Your IoT Hub name} -o table
```

Step 3: az iot hub device-identity show-connection-string

```
--hub-name {Hub01}
```

```
--device-id Sensor00
```

```
--output table
```

Step 4:

Create the service app

In this section, you create a Node.js console app that adds location metadata to the device twin.
npm install azure-iothub --save

References:

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-node-node-twin-getstarted>

QUESTION 62

Hotspot Question

You are creating a bot for a company by using QnA Maker.

You need to ensure that the company can update the bot without third-party assistance.

What should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Scenario

Add customer question/answer content

Component

| |
|------------------------------|
| QnA Maker runtime |
| QnA Maker management service |

Use an Azure Application Insights resource for analytics.

| |
|------------------------------|
| QnA Maker runtime |
| QnA Maker management service |

Update and train a knowledge base

| |
|------------------------------|
| QnA Maker runtime |
| QnA Maker management service |

Answer:**Answer Area**

| Scenario | Component |
|---|---|
| Add customer question/answer content | <div style="background-color: #e0f2e0; padding: 5px; border: 1px solid black; width: fit-content; margin: auto;">QnA Maker runtime QnA Maker management service</div> |
| Use an Azure Application Insights resource for analytics. | <div style="background-color: #e0f2e0; padding: 5px; border: 1px solid black; width: fit-content; margin: auto;">QnA Maker runtime QnA Maker management service</div> |
| Update and train a knowledge base | <div style="background-color: #e0f2e0; padding: 5px; border: 1px solid black; width: fit-content; margin: auto;">QnA Maker runtime QnA Maker management service</div> |

QUESTION 63

Drag and Drop Question

You are developing an IoT solution. The solution requires bidirectional communication between a .NET application and Azure IoT Hub.

You need to obtain connection information for a single test device.

Which three commands should you use to develop the solution? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

NOTE: Each correct selection is worth one point.

| Commands | Answer Area |
|--|---|
| <pre>az iot hub device-identity show-connection-string --hub-name <iot-hub-name> --device-id <device-id></pre> | |
| <pre>az iot hub device-identity export --hub-name <iot-hub-name> --device-id <device-id></pre> | |
| <pre>az extension add --name <iot-extension-name></pre> |   |
| <pre>az iot hub device-identity create --hub-name <iot-hub-name> --device-id <device-id></pre> | |
| <pre>az iot hub device-identity import --hub-name <iot-hub-name> --device-id <device-id></pre> | |

Answer:**Commands**

```
az iot hub device-identity export  
--hub-name <iot-hub-name>  
--device-id <device-id>
```

Answer Area

```
az extension add  
--name <iot-extension-name>
```

```
az iot hub device-identity create  
--hub-name <iot-hub-name>  
--device-id <device-id>
```

```
az iot hub device-identity show-connection-string  
--hub-name <iot-hub-name>  
--device-id <device-id>
```



```
az iot hub device-identity import  
--hub-name <iot-hub-name>  
--device-od <device-id>
```

Explanation:

Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension:

Step 1: az extension add

```
--name <iot-extension-name>
```

Run the following command in the command-line environment where you are using the Azure CLI to install the IoT extension:

```
az extension add --name azure-cli-iot-ext
```

Step 2: az iot hub device-identity create

```
--hub-name <iot-hub-name>
```

```
--device-id <device-id>
```

Create a new device in the IoT Hub "DemoHub"

Command: az iot hub device-identity create --hub-name DemoHub --device-id testDevice

Step 3: az iot hub device-identity show-connection-string

```
--hub-name { iot-hub-name }
```

```
--device-id <device-id>
```

References:

<https://github.com/MicrosoftDocs/azure-docs/blob/master/includes/iot-hub-get-started-create-device-identity.md>

QUESTION 64

Hotspot Question

You are developing an IoT solution.

The solution requires bidirectional communication between a client .NET application and Azure IoT hub. A .NET back-end application will connect to the IoT Hub to process information.

You need to collect the values required for the back-end application to connect with the newly created IoT Hub.

How should you complete the commands? To answer, select the appropriate options in the answer area.

Answer Area

```
az iot hub
    create
    show
    list
    certificate
--query properties.eventHubEndpoints.events.endpoint
--name <iot-hub-name>

az iot hub
    create
    show
    list
    certificate
--query properties.eventHubEndpoints.events.path
--name<iot-hub-name>
az iot hub policy
    create
    show
    list
    certificate
    connection-string
    primaryKey
--query
--name <iot-owner-name>
--hub-name <iot-hub-name>
```

Answer:

Answer Area

```
az iot hub
    create
    show
    list
    certificate
--query properties.eventHubEndpoints.events.endpoint
--name <iot-hub-name>

az iot hub
    create
    show
    list
    certificate
--query properties.eventHubEndpoints.events.path
--name<iot-hub-name>
az iot hub policy
    create
    show
    list
    certificate
    connection-string
    primaryKey
--query
--name <iot-owner-name>
--hub-name <iot-hub-name>
```

Explanation:

Box 1: show

Event Hub-compatible endpoint

az iot hub show --query properties.eventHubEndpoints.events.endpoint --name {your IoT Hub name}

Box 2: show

Event Hub-compatible name

az iot hub show --query properties.eventHubEndpoints.events.path --name {your IoT Hub name}

Box 3: show

az iot hub policy show --name iothubowner --query primaryKey --hub-name {your IoT Hub name}

Box 4: primaryKey

References:

<https://github.com/Azure-Samples/azure-iot-samples-csharp/blob/master/iot-hub/Quickstarts/read-d2c-messages/ReadDeviceToCloudMessages.cs>**QUESTION 65**

Drag and Drop Question

You develop a bot by using Language Understanding Intelligence Service (LUIS) and the .NET Bot framework. You use LUIS in the Azure portal to optimize the bot.

You review the utterances and determine that users are requesting time and venue information for events.

You need to improve the prediction efficiency of the bot.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

NOTE: Each correct selection is worth one point.

Actions

Create an intent for each event type.

Add a pattern

Create a Pattern.any entity.

Add example utterances.

Create a List entity.

Answer Area

Answer:

Actions

Add a pattern

Create a Pattern.any entity.

Answer Area

Create an intent for each event type.

Add example utterances.

Create a List entity.

Explanation:

Step 1: Create an intent for each event type

Identify your intents

Step 2: Add example utterances

Create example utterances for each intent

Step 3: Create a List Entity

Identify your entities

A list entity is an explicitly specified list of values. Each value consists of one or more synonyms.

In a travel app, you might choose to create a list entity to represent airport names.

References:

<https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-plan-your-app>

QUESTION 66

Case Study 5**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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

You need to construct the link to the summary report for the email that is sent to users.

What should you do?

- A. Create a SharedAccessBlobPolicy and add it to the containers SharedAccessPolicies. Call GetSharedAccessSignature on the blob and use the resulting link.
- B. Create a SharedAccessAccountPolicy and call GetSharedAccessSignature on storage account and use the resulting link.
- C. Create a SharedAccessBlobPolicy and set the expiry time to two weeks from today. Call GetSharedAccessSignature on the blob and use the resulting link.
- D. Create a SharedAccessBlobPolicy and set the expiry time to two weeks from today. Call GetSharedAccessSignature on the container and use the resulting link.

Answer: D

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>

QUESTION 67**Case Study 5****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 GetDRBlock(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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

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

Answer: B

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 68

Case Study 5

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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

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

Answer: B

Explanation:

Instead of DataMasking, 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 69

Case Study 5

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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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 -EnableEncryptionServiceFile `
CS13 -KeyVaultEncryption `
CS14 -KeyName $key.Name
CS15 -KeyVersion $key.Version
CS16 -KeyVaultUri $keyVault.VaultUri
```

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

Answer: A

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 70

Case Study 5

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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

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

Answer: A

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/azurerm.keyvault/set-azurermkeyvaultaccesspolicy>

QUESTION 71

Case Study 5

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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

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.

Answer: D

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>

QUESTION 72

Case Study 5

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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

Hotspot Question

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.

Answer Area

The screenshot shows a code editor with the following code snippet:

```
var policy =
```

A dropdown menu is open above the code, listing the following options:

- Policy
- RetryPolicy
- RetryOptions
- ReconnectRetryPolicy

Below the code, another dropdown menu is open, showing the following suggestions:

- .Retry(3);
- .CircuitBreaker(3, TimeSpan.FromMilliseconds(100));
- .WaitAndRetryAsync(3, i=> TimeSpan.FromMilliseconds(100));

Answer:

Answer Area

```
var policy =  
    Policy  
    RetryPolicy  
    RetryOptions  
    ReconnectRetryPolicy  
.Handle<Exception>()  
  
.Retry(3);  
.CircuitBreaker(3, TimeSpan.FromMilliseconds(100));  
.WaitAndRetryAsync(3, i=> TimeSpan.FromMilliseconds(100));
```

Explanation:

Box 1: Policy

RetryPolicy retry = Policy

.Handle<HttpRequestException>()

.Retry(3);

The above example will create a retry policy which will retry up to three times if an action fails with an exception handled by the Policy.

Box 2: WaitAndRetryAsync(3,i => TimeSpan.FromMilliseconds(100* Math.Pow(2,i-1))); A common retry strategy is exponential backoff: this allows for retries to be made initially quickly, but then at progressively longer intervals, to avoid hitting a subsystem with repeated frequent calls if the subsystem may be struggling.

Example:

Policy

.Handle<SomeExceptionType>()

.WaitAndRetry(3, retryAttempt =>

TimeSpan.FromSeconds(Math.Pow(2, retryAttempt))

);

References:

<https://github.com/App-vNext/Polly/wiki/Retry>**QUESTION 73****Case Study 5****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 }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

Drag and Drop Question

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.

| Values | Answer Area |
|------------------------------|--|
| true | var copyOptions = new CopyOptions {}; |
| false | var context= new [](source,destination)=>Task.FromResult(true); |
| SingleTransferContext | context. [](source, destination) => Task.FromResult(true); |
| DirectoryTransferContext | await TransferManager.CopyAsync(blob, GetDRBlob(blob), isServiceCopy:[] |
| ShouldTransferCallbackAsync | , context:context, options: copyOptions); copyOptions, context); |
| ShouldOverwriteCallbackAsync | |

Answer:

| Values | Answer Area |
|------------------------------|--|
| true | var copyOptions = new CopyOptions {}; |
| SingleTransferContext | var context= new [](source,destination)=>Task.FromResult(true); |
| DirectoryTransferContext | context. [ShouldTransferCallbackAsync](source, destination) => Task.FromResult(true); |
| ShouldOverwriteCallbackAsync | await TransferManager.CopyAsync(blob, GetDRBlob(blob), isServiceCopy: false |
| | , context:context, options: copyOptions); copyOptions, context); |

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 locally.

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 74

Case Study 5

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 GetDRBlock(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     }
PC38     private KeyVaultClient _keyVaultClient = new KeyVaultClient(" . . .");
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(HttpStatusCode 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
```

Drag and Drop Question

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.

| Code segments | Answer Area |
|---|-----------------------------------|
| MSITokenProvider("...", null) | var tp=new |
| tp.GetAccessTokenAsync("...") | var t=new TokenCredential(await |
| AzureServiceTokenProvider() | return new StorageCredentials(t); |
| StringTokenProvider("storage","msi") | |
| tp.GetAuthenticationHeaderAsync (CancellationToken.None) | |

Answer:

| Code segments | Answer Area |
|---|--|
| MSITokenProvider("...", null) | var tp=new AzureServiceTokenProvider() |
| | var t=new TokenCredential(await tp.GetAccessTokenAsync("...")) |
| | return new StorageCredentials(t); |
| StringTokenProvider("storage","msi") | |
| tp.GetAuthenticationHeaderAsync (CancellationToken.None) | |

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 75**Case Study 6****LabelMaker app**

Coho Winery produces bottles, and distributes a variety of wines globally. You are developer implementing highly scalable and resilient applications to support online order processing by using Azure solutions.

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

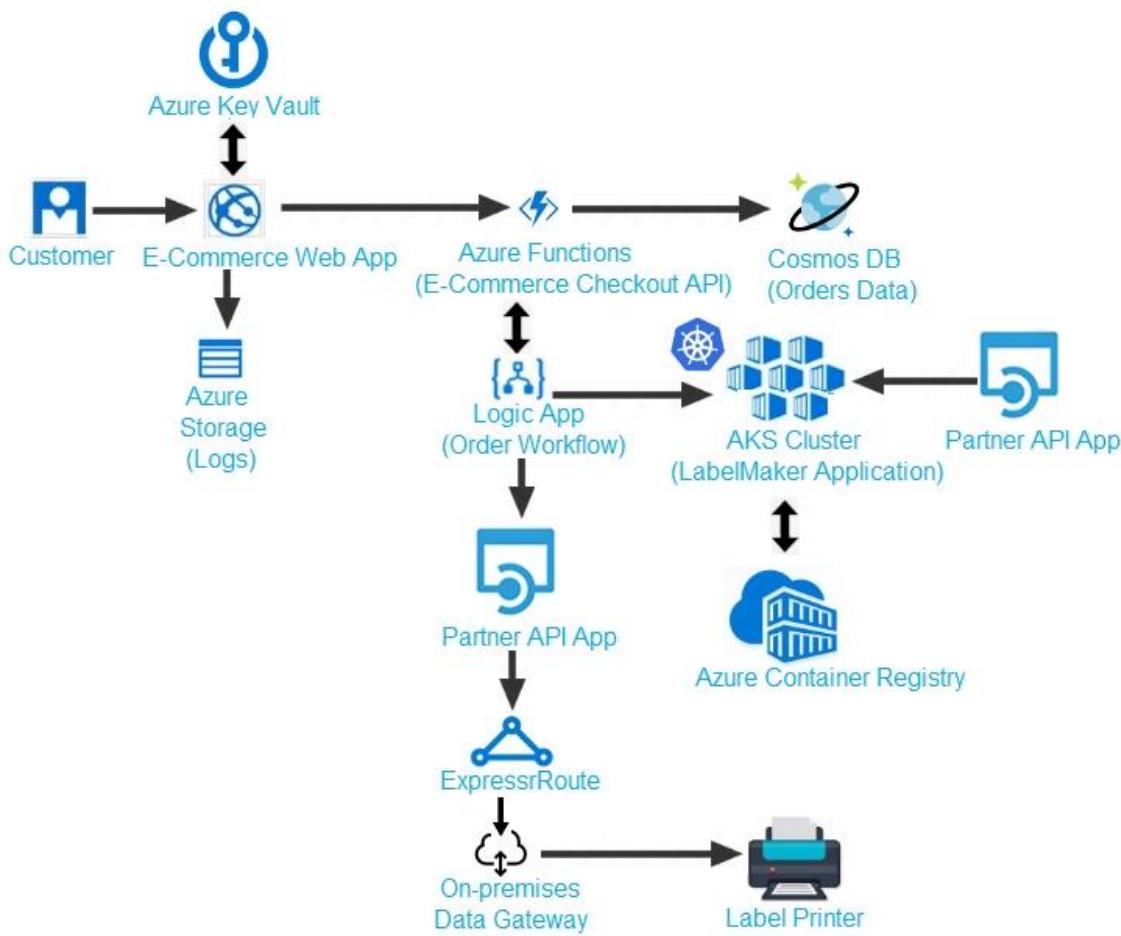
You have the following security requirements:

- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

Azure Monitor Container Health must be used to monitor the performance of workloads that are deployed to Kubernetes environments and hosted on Azure Kubernetes Service (AKS).

You must use Azure Container Registry to publish images that support the AKS deployment.



Calls to the Printer API App fail periodically due to printer communication timeouts.

Printer communications timeouts occur after 10 seconds. The label printer must only receive up to 5 attempts within one minute.

The order workflow fails to run upon initial deployment to Azure.

Order .json

```
01 {
02   "id" : 1,
03   "customers" : [
04     {
05       "familyName" : "Doe",
06       "givenName" : "John",
07       "customerid" : 5
08     }
09   ],
10   "line_items" : [
11     {
12       "fulfillable_quantity" : 1,
13       "id" : 6,
14       "price" : "199.99",
15       "product_id" : 7513594,
16       "quantity": 1,
17       "requires_shipping" : true,
18       "sku" : "SFC-342-N",
19       "title" : "Surface Go",
20       "vendor" : "Microsoft",
21       "name" : "Surface Go - 8GB",
22       "taxable" : true,
23       "tax_lines" : [
24         {
25           "title" : "State Tax",
26           "price" : "3.98",
27           "rate" : 0.06
28         }
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",
42     "city" : "NY"
43   }
44 }
```

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.

Answer: ABD

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 76

Case Study 6

LabelMaker app

Coho Winery produces bottles, and distributes a variety of wines globally. You are developer implementing highly scalable and resilient applications to support online order processing by using Azure solutions.

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

You have the following security requirements:

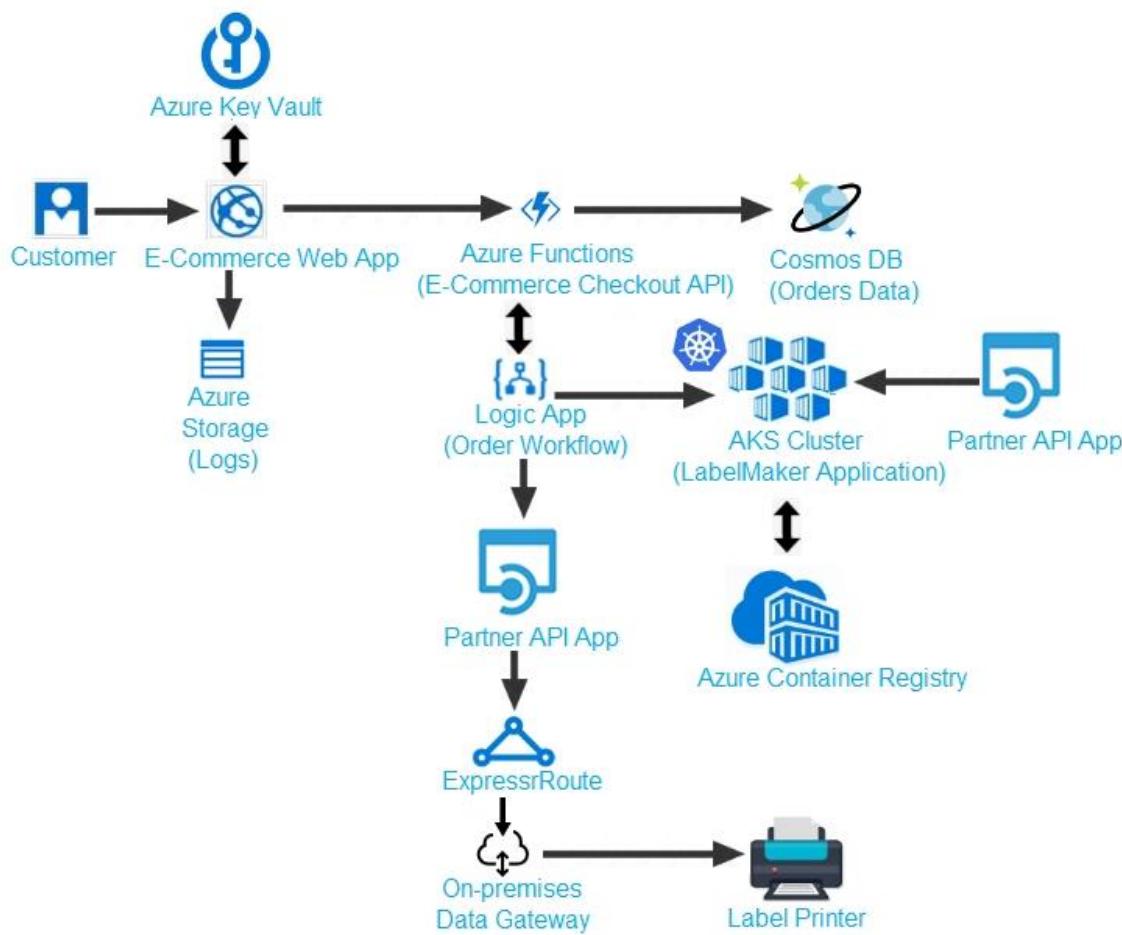
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.

- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

Azure Monitor Container Health must be used to monitor the performance of workloads that are deployed to Kubernetes environments and hosted on Azure Kubernetes Service (AKS).

You must use Azure Container Registry to publish images that support the AKS deployment.



Calls to the Printer API App fail periodically due to printer communication timeouts.

Printer communications timeouts occur after 10 seconds. The label printer must only receive up to 5 attempts within one minute.

The order workflow fails to run upon initial deployment to Azure.

Order .json

```
01 {
02   "id" : 1,
03   "customers" : [
04     {
05       "familyName" : "Doe",
06       "givenName" : "John",
07       "customerid" : 5
08     }
09   ],
10   "line_items" : [
11     {
12       "fulfillable_quantity" : 1,
13       "id" : 6,
14       "price" : "199.99",
15       "product_id" : 7513594,
16       "quantity": 1,
17       "requires_shipping" : true,
18       "sku" : "SFC-342-N",
19       "title" : "Surface Go",
20       "vendor" : "Microsoft",
21       "name" : "Surface Go - 8GB",
22       "taxable" : true,
23       "tax_lines" : [
24         {
25           "title" : "State Tax",
26           "price" : "3.98",
27           "rate" : 0.06
28         }
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",
42     "city" : "NY"
43   }
44 }
```

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.

Answer: B

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 77

Case Study 6

LabelMaker app

Coho Winery produces bottles, and distributes a variety of wines globally. You are developer implementing highly scalable and resilient applications to support online order processing by using Azure solutions.

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

You have the following security requirements:

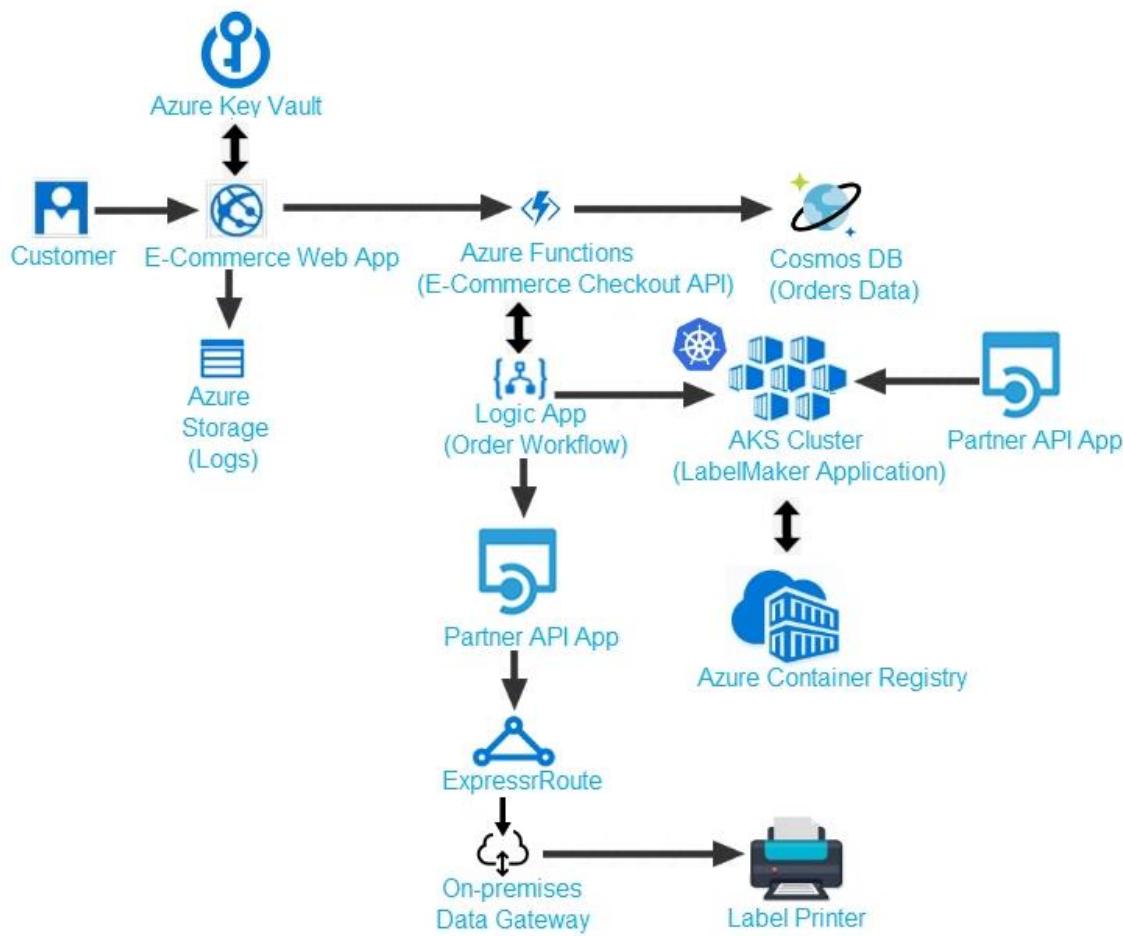
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).

- Conditional access policies must be applied at the application level to protect company content
- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

Azure Monitor Container Health must be used to monitor the performance of workloads that are deployed to Kubernetes environments and hosted on Azure Kubernetes Service (AKS).

You must use Azure Container Registry to publish images that support the AKS deployment.



Calls to the Printer API App fail periodically due to printer communication timeouts.

Printer communications timeouts occur after 10 seconds. The label printer must only receive up to 5 attempts within one minute.

The order workflow fails to run upon initial deployment to Azure.

Order .json

```
01 {
02   "id" : 1,
03   "customers" : [
04     {
05       "familyName" : "Doe",
06       "givenName" : "John",
07       "customerid" : 5
08     }
09   ],
10   "line_items" : [
11     {
12       "fulfillable_quantity" : 1,
13       "id" : 6,
14       "price" : "199.99",
15       "product_id" : 7513594,
16       "quantity": 1,
17       "requires_shipping" : true,
18       "sku" : "SFC-342-N",
19       "title" : "Surface Go",
20       "vendor" : "Microsoft",
21       "name" : "Surface Go - 8GB",
22       "taxable" : true,
23       "tax_lines" : [
24         {
25           "title" : "State Tax",
26           "price" : "3.98",
27           "rate" : 0.06
28         }
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",
42     "city" : "NY"
43   }
44 }
```

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.

Answer: AD

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 78

Case Study 6

LabelMaker app

Coho Winery produces bottles, and distributes a variety of wines globally. You are developer implementing highly scalable and resilient applications to support online order processing by using Azure solutions.

Coho Winery has a LabelMaker application that prints labels for wine bottles. The application sends data to several printers. The application consists of five modules that run independently on virtual machines (VMs). Coho Winery plans to move the application to Azure and continue to support label creation.

External partners send data to the LabelMaker application to include artwork and text for custom label designs.

Data

You identify the following requirements for data management and manipulation:

- Order data is stored as nonrelational JSON and must be queried using Structured Query Language (SQL).
- Changes to the Order data must reflect immediately across all partitions. All reads to the Order data must fetch the most recent writes.

You have the following security requirements:

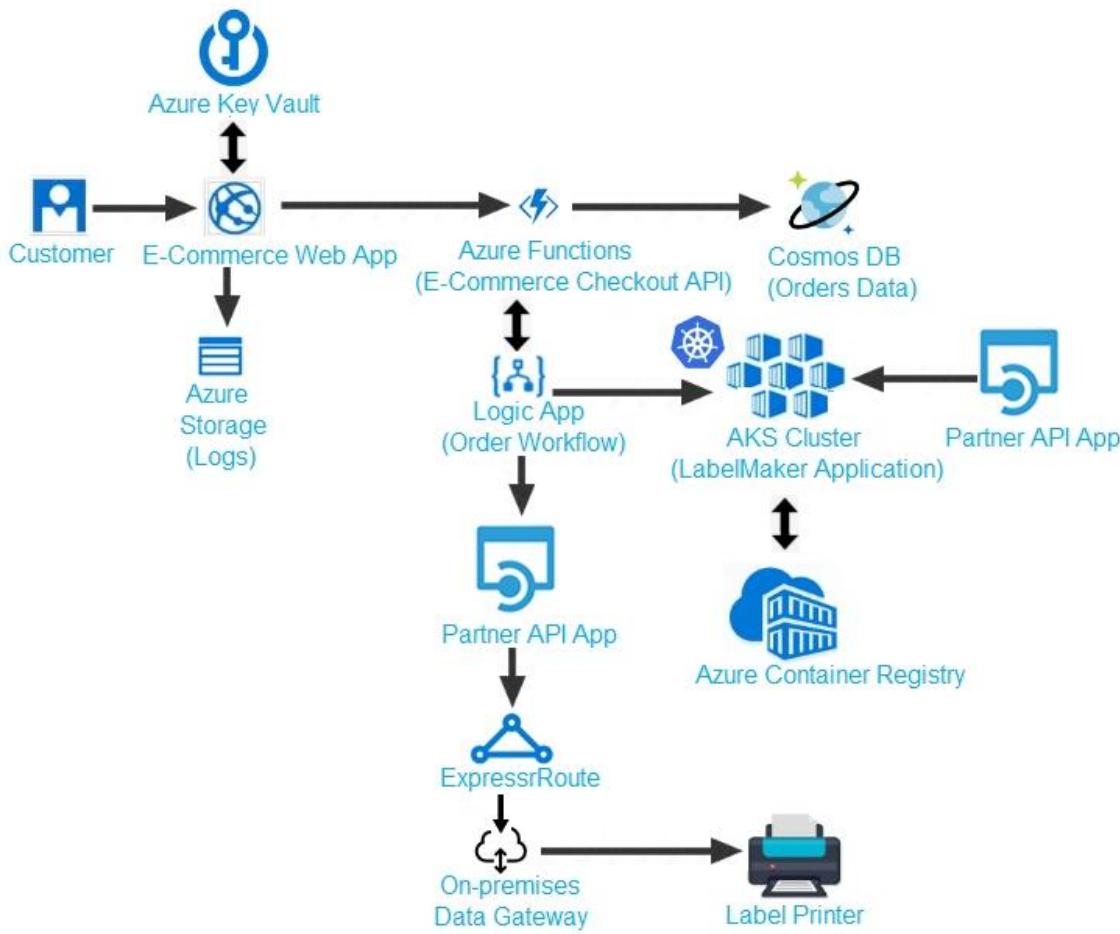
- Users of Coho Winery applications must be able to provide access to documents, resources, and applications to external partners.
- External partners must use their own credentials and authenticate with their organization's identity management solution.
- External partner logins must be audited monthly for application use by a user account administrator to maintain company compliance.
- Storage of e-commerce application settings must be maintained in Azure Key Vault.
- E-commerce application sign-ins must be secured by using Azure App Service authentication and Azure Active Directory (AAD).
- Conditional access policies must be applied at the application level to protect company content

- The LabelMaker applications must be secured by using an AAD account that has full access to all namespaces of the Azure Kubernetes Service (AKS) cluster.

LabelMaker app

Azure Monitor Container Health must be used to monitor the performance of workloads that are deployed to Kubernetes environments and hosted on Azure Kubernetes Service (AKS).

You must use Azure Container Registry to publish images that support the AKS deployment.



Calls to the Printer API App fail periodically due to printer communication timeouts.

Printer communications timeouts occur after 10 seconds. The label printer must only receive up to 5 attempts within one minute.

The order workflow fails to run upon initial deployment to Azure.

Order .json

```
01 {
02   "id" : 1,
03   "customers" : [
04     {
05       "familyName" : "Doe",
06       "givenName" : "John",
07       "customerid" : 5
08     }
09   ],
10   "line_items" : [
11     {
12       "fulfillable_quantity" : 1,
13       "id" : 6,
14       "price" : "199.99",
15       "product_id" : 7513594,
16       "quantity": 1,
17       "requires_shipping" : true,
18       "sku" : "SFC-342-N",
19       "title" : "Surface Go",
20       "vendor" : "Microsoft",
21       "name" : "Surface Go - 8GB",
22       "taxable" : true,
23       "tax_lines" : [
24         {
25           "title" : "State Tax",
26           "price" : "3.98",
27           "rate" : 0.06
28         }
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",
42     "city" : "NY"
43   }
44 }
```

Hotspot Question

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.

Answer Area

| Requirement | Option |
|----------------|--|
| Authentication | <div style="text-align: right;">▼</div> <ul style="list-style-type: none">B2CB2BSelf-service signupOrganizational Units (OU) |
| Login Auditing | <div style="text-align: right;">▼</div> <ul style="list-style-type: none">Access reviewRisky sign-ins reportIdentity ProtectionPrivileged Identity Management |

Answer:

Answer Area

| Requirement | Option |
|----------------|--|
| Authentication | <div style="text-align: right;">▼</div> <ul style="list-style-type: none">B2CB2BSelf-service signupOrganizational Units (OU) |
| Login Auditing | <div style="text-align: right;">▼</div> <ul style="list-style-type: none">Access reviewRisky sign-ins reportIdentity ProtectionPrivileged Identity Management |

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>

QUESTION 79**Case Study 7****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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
>LoginEvent.cs
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

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

Answer: D

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 80

Case Study 7

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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
>LoginEvent.cs
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

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

Answer: B

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 81

Case Study 7

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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
>LoginEvent.cs
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08 {
LE09         return JsonConvert.SerializeObject(this);
LE10 }
LE11 }
```

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

Answer: B

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 82

Case Study 7

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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
>LoginEvent.cs
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08 {
LE09         return JsonConvert.SerializeObject(this);
LE10 }
LE11 }
```

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

Answer: A

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 83

Case Study 7

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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

Drag and Drop Question

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.

| Code segments | Answer Area |
|-----------------------|--|
| /health | public class filter : <input type="text"/> |
| /status | { private readonly <input type="text"/> _next; |
| RequestTelemetry | public filter(<input type="text"/> next) { _next = next; } |
| PageViewTelemetry | public void Process(ITelemetry item) { var x = item as <input type="text"/> ; if (x?.Url.AbsolutePath == " <input type="text"/>) |
| ITelemetryProcessor | { return; } _next.Process(item); } |
| ITelemetryInitializer | } |

Answer:

| Code segments | Answer Area |
|-----------------------|---|
| /health | public class filter : ITelemetryProcessor |
| /status | { |
| RequestTelemetry | private readonly ITelemetryProcessor _next; |
| PageViewTelemetry | public filter(ItelemetryProcessor next) |
| ITelemetryProcessor | { |
| ITelemetryInitializer | _next = next; |
| | } |
| | public void Process(ITelemetry item) |
| | { |
| | var x = item as RequestTelemetry ; |
| | if (x?.Url.AbsolutePath == "/health") |
| | { |
| | return; |
| | } |
| | _next.Process(item); |
| | } |
| | } |

Explanation:

Scenario: Exclude non-user actions from Application Insights telemetry.

Box 1: ITelemetryProcessor

To create a filter, implement ITelemetryProcessor. This technique gives you more direct control over what is included or excluded from the telemetry stream.

Box 2: ITelemetryProcessor

Box 3: ITelemetryProcessor

Box 4: RequestTelemetry

Box 5: /health

To filter out an item, just terminate the chain.

References:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/api-filtering-sampling>

QUESTION 84**Case Study 7****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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

Drag and Drop Question

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.

| Code segments | Answer Area |
|---|---|
| Process | public class IncludeEventId: [REDACTED] |
| Initialize | { |
| telemetry.Sequence | public void [REDACTED] |
| ITelemetryProcessor | (ITelemetry telemetry) |
| ITelemetryInitializer | { |
| Telemetry.Context | [REDACTED].Properties["EventId"] = |
| EventGridController.EventId.Value | [REDACTED] |
| [EventTelemetry]telemetry.Properties["EventId"] | } |

Answer:

| Code segments | Answer Area |
|-----------------------------------|--|
| Process | public class IncludeEventId: ITelemetryInitializer |
| | { |
| telemetry.Sequence | public void Initialize |
| ITelemetryProcessor | (ITelemetry telemetry) |
| | { |
| Telemetry.Context | [REDACTED].Properties["EventId"] = |
| | [EventTelemetry]telemetry.Properties["EventId"] |
| EventGridController.EventId.Value | [REDACTED] |
| | } |

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")]

QUESTION 85**Case Study 7****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 = JArray.Parse(eventsJson);
EG07
EG08         foreach (var @event in events)
EG09         {
EG10             EventId.Value = @event["id"].ToString();
EG11             if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12             {
EG13                 SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14             }
EG15
EG16             {
EG17                 EnsureLogging(@event["subject"].ToString());
EG18             }
EG19         }
EG20         return null;
EG21     }
EG22     private void EnsureLogging(string resource)
EG23     {
EG24         . .
EG25     }
EG26     private async Task SendToAnomalyDetectionService(string uri)
EG27     {
EG28         var content = GetLogData(uri)
EG29         var scoreRequest = new
EG30         {
EG31             Inputs = new Dictionary<string, List<Dictionary<string, string>>()
EG32             {
EG33                 {
EG34                     "input1",
EG35                     new List<Dictionary<string, string>>()
EG36                     {
EG37                         new Dictionary<string, string>()
EG38                         {
EG39                             {
EG40                                 "logcontent", content
EG41                             }
EG42                         }
EG43                     }
EG44                 },
EG45             },
EG46             GlobalParameters = new Dictionary<string, string>() ( )
EG47         };
EG48         var result = await (new HttpClient()).PostAsJsonAsync(" ... ", scoreRequest);
EG49         var rawModelResult = await result.Content.ReadAsStringAsync();
EG50         var modelResult = JObject.Parse(rawModelResult);
EG51         if (modelResult ["notify"].HasValues)
EG52         {
EG53             . .
EG54         }
EG55     }
EG56     private (string name, string resourceGroup) ParseResourceId(string resourceId)
EG57     {
EG58         . .
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62         . .
EG63     }
EG64     static string BlobStoreAccountSAS (string containerName)
EG65     {
EG66         . .
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get ; set ; }
LE05     public DateTime eventTime { get ; set ; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

Drag and Drop Question

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.

| Code segments | Answer Area |
|------------------|---|
| topic | if (|
| status | @event["data"]["status"].ToString() == " [] " |
| eventType | && |
| Succeeded | @event["data"]["operationName"].ToString() == "Microsoft.Web/sites/write" |
| operationName |) |
| resourceProvider | |

Answer:

| Code segments | Answer Area |
|------------------|---|
| topic | if (|
| eventType | @event["data"]["status"].ToString() == "Succeeded" |
| resourceProvider | && |
| operationName | @event["data"]["operationName"].ToString() == "Microsoft.Web/sites/write" |
| |) |

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>

QUESTION 86

You are writing code to create and run an Azure Batch job.

You have created a pool of compute nodes.

You need to choose the right class and its method to submit a batch job to the Batch service.

Which method should you use?

- A. JobOperations.EnableJobAsync(String, IEnumerable<BatchClientBehavior>, CancellationToken)
- B. JobOperations.CreateJob()
- C. CloudJob.Enable(IEnumerable<BatchClientBehavior>)
- D. JobOperations.EnableJob(String, IEnumerable<BatchClientBehavior>)
- E. CloudJob.CommitAsync(IEnumerable<BatchClientBehavior>, CancellationToken)

Answer: E

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 87

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

Answer: B

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 88

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

Answer: B

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 89

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

Answer: A

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 documents

Something like:

```
var hotels = new Hotel[];  
{  
    new Hotel()  
    {  
        HotelId = "3",  
        BaseRate = 129.99,  
        Description = "Close to town hall and the river"
```

```
}

};

...

var batch = IndexBatch.Upload(hotels);
3. The next step is to populate the newly-created index
Example:
var batch = IndexBatch.Upload(hotels);
try
{
indexClient.Documents.Index(batch);
}
References:
https://docs.microsoft.com/en-us/azure/search/search-howto-dotnet-sdk
```

QUESTION 90

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);

Answer: C

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 91

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.

Answer: E

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 92

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

Answer: AE

Explanation:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/custom-operations-tracking>

QUESTION 93

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")

Answer: C

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>

QUESTION 94

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.

Answer: C

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 95

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)

Answer: B

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 96

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.

Answer: BE

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>

QUESTION 97

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.

Answer: AC

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 98

Hotspot Question

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 places 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.

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.NowToFileTimeUtc().ToString()}";
            CloudTask task = new CloudTask (taskId, fileTask.Command);
            List<OutputFile> outputFileList = new List<OutputFile>();
            OutputFileBlobContainerDestination outputContainer =
                new OutputFileBlobContainerDestination(outputContainerSasUrl);
            OutputFileBlobContainerDestination failedContainer =
                new OutputFileBlobContainerDestination (failedContainerSasUrl);
            outputFileList.Add(new outputFile(fileTask.Output,
                new outputFileDestination(outputContainer),
                new outputFileUploadOptions(OutputFileUploadCondition. ▼ ))));
            TaskSuccess
            TaskFailure
            TaskCompletion

            outputFileList.Add(new outputFile(fileTask.Output,
                new outputFileDestination(failedContainer),
                new outputFileUploadOptions(OutputFileUploadCondition, ▼ ))));
            TaskSuccess
            TaskFailure
            TaskCompletion

            task. ▼ =outputFileList;
            OutputFiles
            FilesToStage
            ResourceFiles
            StageFiles

            task.Add(task);
        });
    }
    return tasks;
}
```

Answer:

Answer Area

```
public list<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
    string outputContainerSasUrl, string failedContainerSasUrl)
{
    BatchSharedKeyCredentials sharedKeyCredentials =
        new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
    List<CloudTask> tasks = new List<CloudTask>();
    using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
    {
        CloudJob = batchClient.JobOperations. ▾ () ;
        GetJob
        GetTask
        EnableJob
        CreateJob
        job.Id = jobId,
        job.PoolInformation = new PoolInformation { PoolId = poolId };
        job.Commit();
        fileTasks.ForEach((fileTask) =>
        {
            string taskId = $"Task{DateTime.NowToFileTimeUtc().ToString()}";
            CloudTask task = new CloudTask (taskId, fileTask.Command);
            List<OutputFile> outputFileList = new List<OutputFile>();
            outputFileBlobContainerDestination outputContainer =
                new outputFileBlobContainerDestination(outputContainerSasUrl);
            outputFileBlobContainerDestination failedContainer =
                new outputFileBlobContainerDestination (failedContainerSasUrl);
            outputFileList.Add(new outputFile(fileTask.Output,
                new outputFileDestination(outputContainer),
                new outputFileUploadOptions(OutputFileUploadCondition. ▾ )) );
            TaskSuccess
            TaskFailure
            TaskCompletion
            outputFileList.Add(new outputFile(fileTask.Output,
                new outputFileDestination(failedContainer),
                new outputFileUploadOptions(OutputFileUploadCondition, ▾ )) );
            TaskSuccess
            TaskFailure
            TaskCompletion
            task. ▾ =outputFileList;
            OutputFiles
            FilesToStage
            ResourceFiles
            StageFiles
            task.Add(task);
        });
    }
    return tasks;
}
```

Explanation:

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 99

Drag and Drop Question

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.

| Actions | Answer area |
|--|-------------|
| Create a Service Bus subscription for each restaurant for which a driver can receive orders. | |
| Create a single Service Bus topic. | |
| Create a single Service Bus subscription. | ◀ |
| Create a Service Bus topic for each restaurant for which a driver can receive messages | ▶ |
| Create a single Service Bus Namespace. | |
| Create a Service Bus Namespace for each restaurant for which a driver can receive messages. | ↑ |

Answer:

| Actions | Answer area |
|--|---|
| Create a Service Bus subscription for each restaurant for which a driver can receive orders. | Create a single Service Bus Namespace. |
| Create a single Service Bus topic. | Create a Service Bus topic for each restaurant for which a driver can receive messages |
| Create a single Service Bus subscription. | Create a Service Bus Namespace for each restaurant for which a driver can receive messages. |

**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 100

Hotspot Question

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.

Answer Area

```
[SerializePropertyNameAsCamelCase]
public class Restaurant
{
    [Key, IsFilterable]
    public int RestaurantId { get; set; }
    [IsSearchable, IsFilterable, IsSortable]
    public string Name { get; set; }

    [IsSearchable, IsFilterable, IsSortable, IsFacetable]
    [IsFilterable, IsFacetable, Required]
    [IsSearchable]
    [IsSearchable, Required]

    public string location { get; set; }
    public string Phone { get; set; }

    [Required]
    [IsSearchable]
    [IsFilterable, IsFacetable, Required]
    [IsFilterable, IsFacetable, IsSortable]

    public string Description { get; set; }

    [IsFiltrable, IsSortable, IsSearchable]
    [IsFilterable, IsSortable, IsFacetable]
    [IsFiltrable, IsSortable, Key]
    [IsFilterable, IsSortable, IsSearchable, Required]

    public double Rating { get; set; }

    [IsSearchable, IsFilterable, IsFacetable]
    [IsFilterable, IsSortable, Key]
    [IsFilterable, IsSortable, IsSearchable]
    [IsFilterable, IsSortable, Key, Required]

    public List<string> Cuisines { get; set; }

    [IsFilterable, IsSortable, Key, Required]
    [IsSearchable, IsSortable, IsFacetable]
    [IsFilterable, IsSortable, Key, IsSearchable]
    [IsFilterable, IsFacetable]
```

Answer:**Answer Area**

```
[SerializePropertyNameAsCamelCase]
public class Restaurant
{
    [Key, IsFilterable]
    public int RestaurantId { get; set; }
    [IsSearchable, IsFilterable, IsSortable]
    public string Name { get; set; }

    [IsSearchable, IsFilterable, IsSortable, IsFacetable]
    [IsFilterable, IsFacetable, Required]
    [IsSearchable]
    [IsSearchable, Required]

    public string location { get; set; }
    public string Phone { get; set; }

    [Required]
    [IsSearchable]
    [IsFilterable, IsFacetable, Required]
    [IsFilterable, IsFacetable, IsSortable]

    public string Description { get; set; }

    [IsFiltrable, IsSortable, IsSearchable]
    [IsFilterable, IsSortable, IsFacetable]
    [IsFiltrable, IsSortable, Key]
    [IsFilterable, IsSortable, IsSearchable, Required]

    public double Rating { get; set; }

    [IsSearchable, IsFilterable, IsFacetable]
    [IsFilterable, IsSortable, Key]
    [IsFilterable, IsSortable, IsSearchable]
    [IsFilterable, IsSortable, Key, Required]

    public List<string> Cuisines { get; set; }

    [IsFilterable, IsSortable, Key, Required]
    [IsSearchable, IsSortable, IsFacetable]
    [IsFilterable, IsSortable, Key, IsSearchable]
    [IsFilterable, IsFacetable]
```

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 101

Drag and Drop Question

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.

Answer Area**Values**

| Azure Resource | Configuration parameter | Value |
|------------------|-------------------------|-------|
| HTTP(s) endpoint | Target | |
| Basic | Gateway credentials | |
| Client cert | | |

Answer:**Answer Area****Values**

| | Configuration parameter | Value |
|------------------|-------------------------|----------------|
| HTTP(s) endpoint | Target | Azure Resource |
| Basic | Gateway credentials | Client cert |

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 102**

Drag and Drop Question

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.

Answer Area

| Policy types | Requirement | Policy type |
|--------------|---|-------------|
| Inbound | Rewrite the request URL to match to the format expected by the web service. | |
| Outbound | Remove formatting text from responses. | |
| Backend | Forward the user ID that is associated with the subscription key for the original request to the back-end service | |

Answer:

Answer Area

| Policy types | Requirement | Policy type |
|--------------|---|-------------|
| Inbound | Rewrite the request URL to match to the format expected by the web service. | Inbound |
| Outbound | Remove formatting text from responses. | Outbound |
| Backend | Forward the user ID that is associated with the subscription key for the original request to the back-end service | Inbound |

Explanation:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-howto-policies>

QUESTION 103

Hotspot Question

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.

Answer Area

Scenario

Search the index by using regular expressions.

Property

| |
|------------|
| QueryType |
| OrderBy |
| SearchMode |

Organize results by counts for name-value pairs.

| |
|------------|
| Facets |
| Filter |
| SearchMode |

List hotels within a specified distance to an airport and that fall within a specific price range.

| |
|----------|
| Order by |
| Top |
| Filter |

Answer:

Answer Area

| Scenario | Property | | | |
|--|--|-----------|---------|------------|
| Search the index by using regular expressions. | <table border="1"><tr><td>QueryType</td></tr><tr><td>OrderBy</td></tr><tr><td>SearchMode</td></tr></table> | QueryType | OrderBy | SearchMode |
| QueryType | | | | |
| OrderBy | | | | |
| SearchMode | | | | |
| Organize results by counts for name-value pairs. | <table border="1"><tr><td>Facets</td></tr><tr><td>Filter</td></tr><tr><td>SearchMode</td></tr></table> | Facets | Filter | SearchMode |
| Facets | | | | |
| Filter | | | | |
| SearchMode | | | | |
| List hotels within a specified distance to an airport and that fall within a specific price range. | <table border="1"><tr><td>Order by</td></tr><tr><td>Top</td></tr><tr><td>Filter</td></tr></table> | Order by | Top | Filter |
| Order by | | | | |
| Top | | | | |
| Filter | | | | |

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 104**Hotspot Question**

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.

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
hub= NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails
NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails
GetInstallation
CreateClientFromConnectionString
CreateOrUpdateInstallation
PatchInstallation
(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"><toast><visual><binding template=""ToastText01""><text id=""1"">" +
@"New item to view" + @"/text></binding></visual></toast>";
try
{
var result=
await hub.
SendWindowsNativeNotificationAsync
SubmitNotificationHubJobAsync
ScheduleNotificationAsync
SendAppleNativeNotificationAsync
.
.
}
catch (System.Exception ex)
{
.
.
}
.
.
```

Answer:

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
hub= NotificationHubClient.CreateClientFromConnectionString(listenConnString, hubName);
NotificationHubClient hub;
try
{
    var result=
        await hub.SendWindowsNativeNotificationAsync(windowsToastPayload);
}
catch (System.Exception ex)
{
    . . .
}
. . .
```

The screenshot shows three separate code snippets with corresponding IntelliSense dropdowns:

- Top snippet:** Shows the constructor for `NotificationHubClient`. The dropdown lists: `NotificationHubClient`, `NotificationHubClientSettings`, `NotificationHubJob`, and `NotificationDetails`.
- Middle snippet:** Shows the `SendWindowsNativeNotificationAsync` method. The dropdown lists: `GetInstallation`, `CreateClientFromConnectionString` (highlighted in green), `CreateOrUpdateInstallation`, and `PatchInstallation`.
- Bottom snippet:** Shows the `SendWindowsNativeNotificationAsync` method again. The dropdown lists: `SendWindowsNativeNotificationAsync` (highlighted in green), `SubmitNotificationHubJobAsync`, `ScheduleNotificationAsync`, and `SendAppleNativeNotificationAsync`.

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>

QUESTION 105

Hotspot Question

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.

Answer Area

```
void ClearCachedTeams()
{
    IDatabase cache = Connection.GetDatabase();
    ICache cache = Connection.GetDatabase();

    cache.KeyDelete("teams");
    cache.StringSet("teams", "");
    cache.ValueDelete("teams");
    cache.StringGet("teams", "");

    ViewBag.ngs += Team data removed from cache.
}
```

Answer:

Answer Area

```
void ClearCachedTeams()
{
    IDatabase cache = Connection.GetDatabase();
    ICache cache = Connection.GetDatabase();

    cache.KeyDelete("teams");
    cache.StringSet("teams", "");
    cache.ValueDelete("teams");
    cache.StringGet("teams", "");

    ViewBag.nsg += Team data removed from cache. ";
}
```

Explanation:

Box 1: IDatabase cache = connection.GetDatabase();
Connection refers to a previously configured ConnectionMultiplexer.

Box 2: cache.StringSet("teams", "")
To specify the expiration of an item in the cache, use the TimeSpan parameter of StringSet.
cache.StringSet("key1", "value1", TimeSpan.FromMinutes(90));

References:

<https://azure.microsoft.com/sv-se/blog/lap-around-azure-redis-cache-preview/>

QUESTION 106

Hotspot Question

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.

Answer Area

| Item | Value | | | | |
|-------------------------------|---|-------------------------------|---------|------------------------|-----|
| Powershell command | <pre>Get-AzureRmRoleDefinition-Name"Reader" ConvertTo-Json Out-File C:\SupportRole.json Get-AzureRmRoleDefinition-Name"Operator" ConvertTo-Json Out-File C:\SupportRole.json Set-AzureRmRoleDefinition-Name"Reader" Input-File C:\SupportRole.json Set-AzureRmRoleDefinition Input-File C:\SupportRole.json</pre> | | | | |
| Actions section | <table border="1"><tr><td>*/read*, *Microsoft.Support/*</td></tr><tr><td>*/read*</td></tr><tr><td>, *Microsoft.Support/*</td></tr><tr><td>*/*</td></tr></table> | */read*, *Microsoft.Support/* | */read* | , *Microsoft.Support/* | */* |
| */read*, *Microsoft.Support/* | | | | | |
| */read* | | | | | |
| , *Microsoft.Support/* | | | | | |
| */* | | | | | |

Answer:**Answer Area**

| Item | Value | | | | |
|-------------------------------|---|-------------------------------|---------|------------------------|-----|
| Powershell command | <pre>Get-AzureRmRoleDefinition-Name"Reader" ConvertTo-Json Out-File C:\SupportRole.json Get-AzureRmRoleDefinition-Name"Operator" ConvertTo-Json Out-File C:\SupportRole.json Set-AzureRmRoleDefinition-Name"Reader" Input-File C:\SupportRole.json Set-AzureRmRoleDefinition Input-File C:\SupportRole.json</pre> | | | | |
| Actions section | <table border="1"><tr><td>*/read*, *Microsoft.Support/*</td></tr><tr><td>*/read*</td></tr><tr><td>, *Microsoft.Support/*</td></tr><tr><td>*/*</td></tr></table> | */read*, *Microsoft.Support/* | */read* | , *Microsoft.Support/* | */* |
| */read*, *Microsoft.Support/* | | | | | |
| */read* | | | | | |
| , *Microsoft.Support/* | | | | | |
| */* | | | | | |

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 107

Drag and Drop Question

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.

Answer Area

| Policy section | Policy | Policy section |
|----------------|--------------------|----------------------|
| Inbound | Set-variable | <input type="text"/> |
| Outbound | Cache-lookup-value | <input type="text"/> |
| | Cache-store-value | <input type="text"/> |
| | Find-and-replace | <input type="text"/> |

Answer:

Answer Area

| Policy section | Policy | Policy section |
|----------------|--------------------|----------------|
| Inbound | Set-variable | Inbound |
| Outbound | Cache-lookup-value | Inbound |
| | Cache-store-value | Outbound |
| | Find-and-replace | Outbound |

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 108**Hotspot Questions**

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 gameId)
2 {
3     TableQuery<DynamicTableEntity> query = new TableQuery<DynamicTableEntity>().Select(new string[] { "Score" })
4         .Where(TableQuery.GenerateFilterConditionForInt("Score", QueryComparisons.GreaterThanOrEqual, 15000)).Take
(20);
5     EntityResolver<KeyValuePair<string, int?>> resolver =
6         (partitionKey, rowKey, ts, props, etag) => new KeyValuePair<string, int?>(rowKey, props["Score"].Int32Value);
7     foreach (var scoreItem in scoreTable.ExecuteQuery(query, resolver, null, null))
8     {
9         Console.WriteLine($"{scoreItem.Key} {scoreItem.Value}");
10    }
11 }
12
13 public class PlayerScore : TableEntity
14 {
15     PartitionKey = gameId;
16     RowKey = playerId;
17     Score = score;
18     TimePlayed = timePlayed;
19 }
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.GenerateAnd, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal, "Smith")
06 TableOperstors.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal,
07 "ssmith@contoso.com")
08 ));  
09 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.

Answer Area

| | Yes | No |
|---|-----------------------|-----------------------|
| The code queries the Azure table and retrieves the TimePlayed property from the table | <input type="radio"/> | <input type="radio"/> |
| The code will display a maximum of twenty records. | <input type="radio"/> | <input type="radio"/> |
| All records will be sent to the client. The client will display records for scores greater than or equal to 15,000. | <input type="radio"/> | <input type="radio"/> |
| The scoreItem.Key property of the KeyValuePairs that ExecuteQuery returns will contain a value for PlayerID. | <input type="radio"/> | <input type="radio"/> |

Answer:

Answer Area

| | Yes | No |
|---|----------------------------------|----------------------------------|
| The code queries the Azure table and retrieves the TimePlayed property from the table | <input type="radio"/> | <input checked="" type="radio"/> |
| The code will display a maximum of twenty records. | <input checked="" type="radio"/> | <input type="radio"/> |
| All records will be sent to the client. The client will display records for scores greater than or equal to 15,000. | <input checked="" type="radio"/> | <input type="radio"/> |
| The scoreItem.Key property of the KeyValuePairs that ExecuteQuery returns will contain a value for PlayerID. | <input checked="" type="radio"/> | <input type="radio"/> |

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 109

Hotspot Question

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(ModelBuilder modelBuilder)
09.         {
10.             modelBuilder.Entity<Player>().HasMany(x => x.Games). WithMany (x => x.Players);
11.         }
12.     }
13.     internal sealed 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.

Answer Area

| | Yes | No |
|---|-----------------------|-----------------------|
| The code will successfully insert a player record. | <input type="radio"/> | <input type="radio"/> |
| The code has a bug and will insert an additional copy of the Game record with a new Id. | <input type="radio"/> | <input type="radio"/> |
| The code has a bug and will insert the wrong gameld value. | <input type="radio"/> | <input type="radio"/> |
| There is a valid many-to-many relationship between Players and Games. | <input type="radio"/> | <input type="radio"/> |

Answer:**Answer Area**

| | Yes | No |
|---|----------------------------------|----------------------------------|
| The code will successfully insert a player record. | <input checked="" type="radio"/> | <input type="radio"/> |
| The code has a bug and will insert an additional copy of the Game record with a new Id. | <input type="radio"/> | <input checked="" type="radio"/> |
| The code has a bug and will insert the wrong gameld value. | <input checked="" type="radio"/> | <input type="radio"/> |
| There is a valid many-to-many relationship between Players and Games. | <input type="radio"/> | <input checked="" type="radio"/> |

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 110

Drag and Drop Question

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.

Azure CLI commands

```
az webapp config hostname add  
--webapp-name $appName  
--resource-group fourthCoffeePublicWebResourceGroup  
--hostname $fqdn
```

Answer area

```
#!/bin/bash  
appName="FourthCoffeePublicWeb$random".  
location "WestUS"  
dockerHubContainerPath="FourthCoffee/publicweb:v1"  
fqdn=http://www.fourthcoffee.com>www.fourthcoffee.com
```



```
az webapp create  
--name $appName  
--plan AppServiceLinuxDockerPlan  
--resource-group fourthCoffeePublicWebResourceGroup
```



```
az webapp config container set  
--docker-custom-image-name $dockerHubContainerPath  
--name $appName  
--resource-group fourthCoffeePublicWebResourceGroup
```

Answer:

Azure CLI commands**Answer area**

```
#!/bin/bash  
appName="FourthCoffeePublicWeb$random".  
location "WestUS"  
dockerHubContainerPath="FourthCoffee/publicweb:v1"  
fqdn=http://www.fourthcoffee.com>www.fourthcoffee.com
```

```
az webapp config hostname add  
--webapp-name $appName  
--resource-group fourthCoffeePublicWebResourceGroup  
--hostname $fqdn
```



```
az webapp create  
--name $appName  
--plan AppServiceLinuxDockerPlan  
--resource-group fourthCoffeePublicWebResourceGroup
```

```
az webapp config container set  
--docker-custom-image-name $dockerHubContainerPath  
--name $appName  
--resource-group fourthCoffeePublicWebResourceGroup
```

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 111

Hotspot Question

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 playerId, 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 and 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.GenerateAnd, TableQuery.GenerateFilterCondition("Email", QueryComparisons.Equal, "Smith")
06         TableOperators.And, TableQuery.GenerateFilterCondition("Email", QueryComparisons.Equal,
07         "ssmith@contoso.com")
08     ));
09 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.

Answer Area

| | Yes | No |
|---|-----------------------|-----------------------|
| The code will work with Cosmos DB. | <input type="radio"/> | <input type="radio"/> |
| The save score function will update and replace a record if one already exists with the same playerId and gameId. | <input type="radio"/> | <input type="radio"/> |
| The data for the game will be automatically partitioned. | <input type="radio"/> | <input type="radio"/> |
| This code will store the values for the gameId and playerId parameters in the database. | <input type="radio"/> | <input type="radio"/> |

Answer:**Answer Area**

| | Yes | No |
|---|----------------------------------|----------------------------------|
| The code will work with Cosmos DB. | <input checked="" type="radio"/> | <input type="radio"/> |
| The save score function will update and replace a record if one already exists with the same playerId and gameId. | <input type="radio"/> | <input checked="" type="radio"/> |
| The data for the game will be automatically partitioned. | <input type="radio"/> | <input checked="" type="radio"/> |
| This code will store the values for the gameId and playerId parameters in the database. | <input checked="" type="radio"/> | <input type="radio"/> |

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 112**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

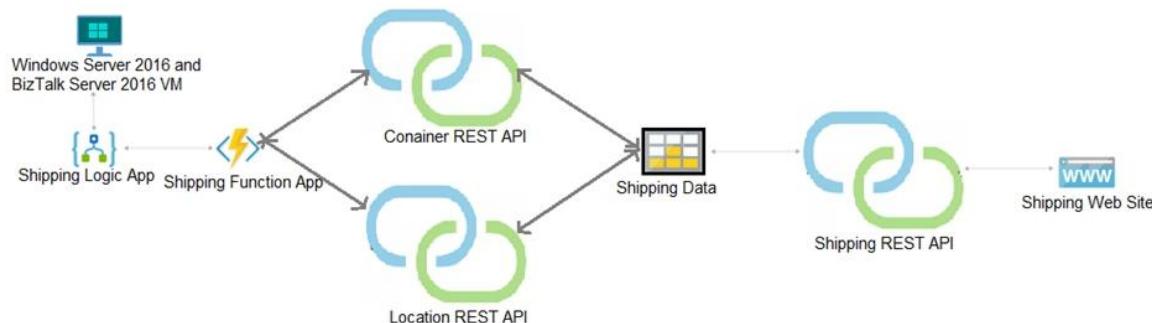
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Shipping Logic App**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

You need to support the requirements for the Shipping Logic App.

What should you use?

- Azure Active Directory Application Proxy
- Point-to-Site (P2S) VPN connection
- Site-to-Site (S2S) VPN connection
- On-premises Data Gateway

Answer: D

Explanation:

Before you can connect to on-premises data sources from Azure Logic Apps, download and install the onpremises data gateway on a local computer. The gateway works as a bridge that provides quick data transfer and encryption between data sources on premises (not in the cloud) and your logic apps.

The gateway supports BizTalk Server 2016.

Note: Microsoft have now fully incorporated the Azure BizTalk Services capabilities into Logic Apps and Azure App Service Hybrid Connections.

Logic Apps Enterprise Integration pack bring some of the enterprise B2B capabilities like AS2 and X12, EDI standards support

Scenario: The Shipping Logic app must meet the following requirements:

Support the ocean transport and inland transport workflows by using a Logic App.

Support industry standard protocol X12 message format for various messages including vessel content details and arrival notices.

Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.

Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

References:

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-gateway-install>

QUESTION 113

Case Study 5 - Wide World Importers

Background

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment

Windows Server 2016 virtual machine

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

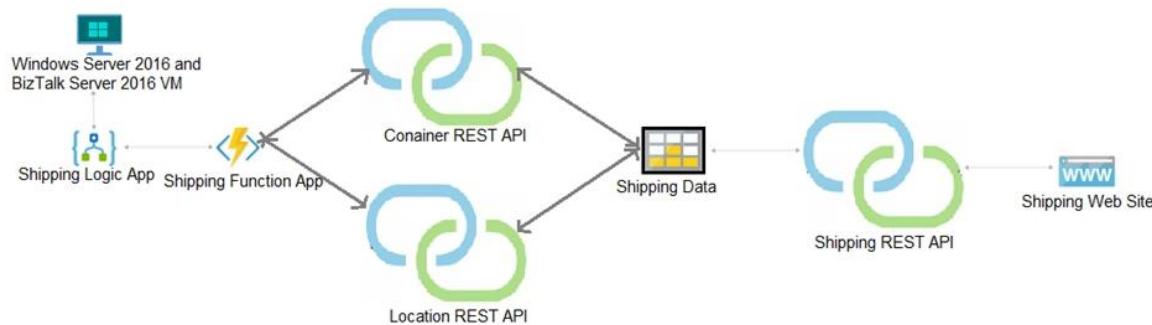
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

Failed to load <http://test-shippingapi.wideworldimporters.com/>: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.

You need to migrate on-premises shipping data to Azure.

What should you use?

- A. Azure Cosmos DB Data Migration tool (dt.exe)
- B. Azure Database Migration service
- C. AzCopy
- D. Azure Migrate

Answer: B

Explanation:

Migrate from on-premises or cloud implementations of MongoDB to Azure Cosmos DB with minimal downtime by using Azure Database Migration Service. Perform resilient migrations of MongoDB data at scale and with high reliability.

Scenario: Data migration from on-premises to Azure must minimize costs and downtime.

The application uses MongoDB JSON document storage database for all container and transport information.

References:

<https://azure.microsoft.com/en-us/updates/mongodb-to-azure-cosmos-db-online-and-offline-migrationsare-now-available/>

QUESTION 114**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

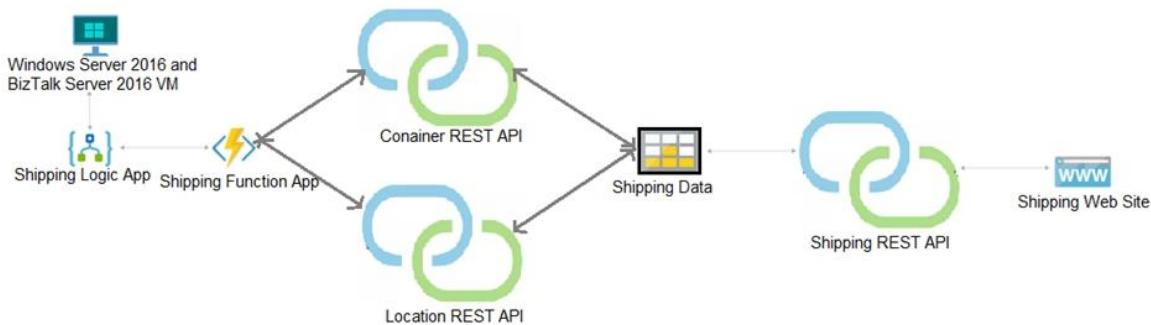
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Shipping Logic App**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

You need to secure the Shipping Logic App.

What should you use?

- A. Azure App Service Environment (ASE)
- B. Azure AD B2B integration
- C. Integration Service Environment (ISE)
- D. VNet service endpoint

Answer: C

Explanation:

Scenario: The Shipping Logic App requires secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.

You can access to Azure Virtual Network resources from Azure Logic Apps by using integration service environments (ISEs).

Sometimes, your logic apps and integration accounts need access to secured resources, such as virtual machines (VMs) and other systems or services, that are inside an Azure virtual network.

To set up this access, you can create an integration service environment (ISE) where you can run your logic apps and create your integration accounts.

References:

<https://docs.microsoft.com/en-us/azure/logic-apps/connect-virtual-network-vnet-isolated-environmentoverview>

QUESTION 115**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

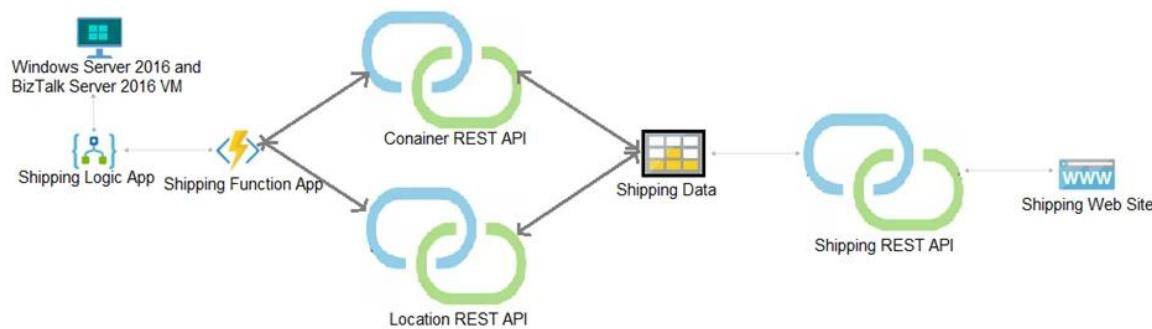
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

Hotspot Question

You need to resolve the Shipping web site error.

How should you configure the Azure Table Storage service? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
<?xml version="1.0" encoding="utf-8"?>
<StorageServiceProperties>
    ...
    <Cors>
        <CorsRule>
            <AllowedHeaders>
                AllowedHeaders
                ExposedHeaders
                AllowedMethods
                AllowedOrigins
            </AllowedHeaders>
            <AllowedMethods>
                GET,PUT
                GET
                POST
                GET,HEAD
            </AllowedMethods>
            ...
        </CorsRule>
    </Cors>
</StorageServiceProperties>
```

Answer:**Answer Area**

```
<?xml version="1.0" encoding="utf-8"?>
<StorageServiceProperties>
    ...
    <Cors>
        <CorsRule>
            <AllowedHeaders>
                AllowedHeaders
                ExposedHeaders
                AllowedMethods
                AllowedOrigins
            </AllowedHeaders>
            <AllowedMethods>
                GET,PUT
                GET
                POST
                GET,HEAD
            </AllowedMethods>
            ...
        </CorsRule>
    </Cors>
</StorageServiceProperties>
```

Explanation:

Box 1: AllowedOrigins

A CORS request will fail if Access-Control-Allow-Origin is missing.

Scenario:

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimp...' therefore not allowed access.
```

Box 2: http://test-shippingapi.wideworldimporters.com

Syntax: Access-Control-Allow-Origin: *

Access-Control-Allow-Origin: <origin>

Access-Control-Allow-Origin: null

<origin> Specifies an origin. Only a single origin can be specified.

Box 3: AllowedOrigins

Box 4: POST

The only allowed methods are GET, HEAD, and POST. In this case POST is used.

<Corsrule> "allowedmethods" Failed to load no "Access-control-Origin" header is present

References:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin>

QUESTION 116

Case Study 5 - Wide World Importers

Background

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment

Windows Server 2016 virtual machine

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

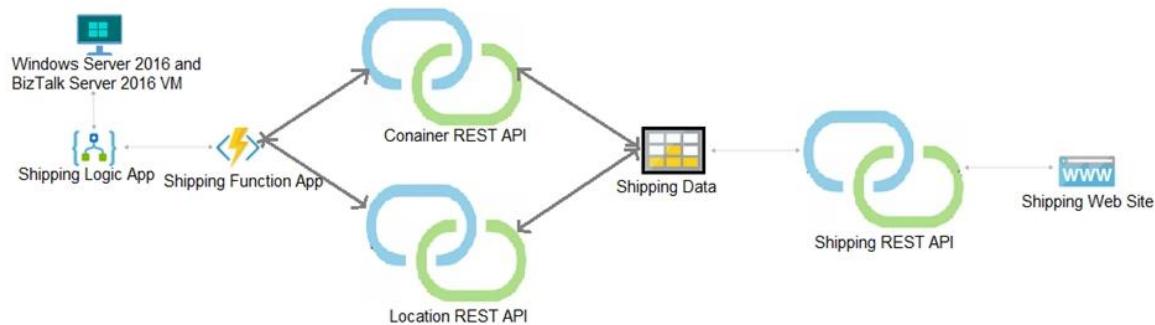
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

Failed to load <http://test-shippingapi.wideworldimporters.com/>: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.

Hotspot Question

You need to secure the Shipping Function app.

How should you configure the app? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|---------------------|--|
| Authorization level | <ul style="list-style-type: none">FunctionAnonymousAdmin |
| User claims | <ul style="list-style-type: none">JSON Web Token (JWT)Shared Access Signature (SAS) tokenAPI Key |
| Trigger type | <ul style="list-style-type: none">blobHTTPqueuetimer |

Answer:

Answer Area

| Setting | Value | | | | |
|-------------------------------------|--|----------------------|-------------------------------------|---------|-------|
| Authorization level | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="background-color: #a9f5e0; color: black; padding: 2px;">Function</td></tr><tr><td style="padding: 2px;">Anonymous</td></tr><tr><td style="padding: 2px;">Admin</td></tr></table></div> | Function | Anonymous | Admin | |
| Function | | | | | |
| Anonymous | | | | | |
| Admin | | | | | |
| User claims | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="background-color: #a9f5e0; color: black; padding: 2px;">JSON Web Token (JWT)</td></tr><tr><td style="padding: 2px;">Shared Access Signature (SAS) token</td></tr><tr><td style="padding: 2px;">API Key</td></tr></table></div> | JSON Web Token (JWT) | Shared Access Signature (SAS) token | API Key | |
| JSON Web Token (JWT) | | | | | |
| Shared Access Signature (SAS) token | | | | | |
| API Key | | | | | |
| Trigger type | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"><table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="padding: 2px;">blob</td></tr><tr><td style="background-color: #a9f5e0; color: black; padding: 2px;">HTTP</td></tr><tr><td style="padding: 2px;">queue</td></tr><tr><td style="padding: 2px;">timer</td></tr></table></div> | blob | HTTP | queue | timer |
| blob | | | | | |
| HTTP | | | | | |
| queue | | | | | |
| timer | | | | | |

Explanation:

Scenario: Shipping Function app: Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

Box 1: Function

Box 2: JSON based Token (JWT)

Azure AD uses JSON based tokens (JWTs) that contain claims

Box 3: HTTP

How a web app delegates sign-in to Azure AD and obtains a token User authentication happens via the browser. The OpenID protocol uses standard HTTP protocol messages.

References:

<https://docs.microsoft.com/en-us/azure/active-directory/develop/authentication-scenarios>

QUESTION 117**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

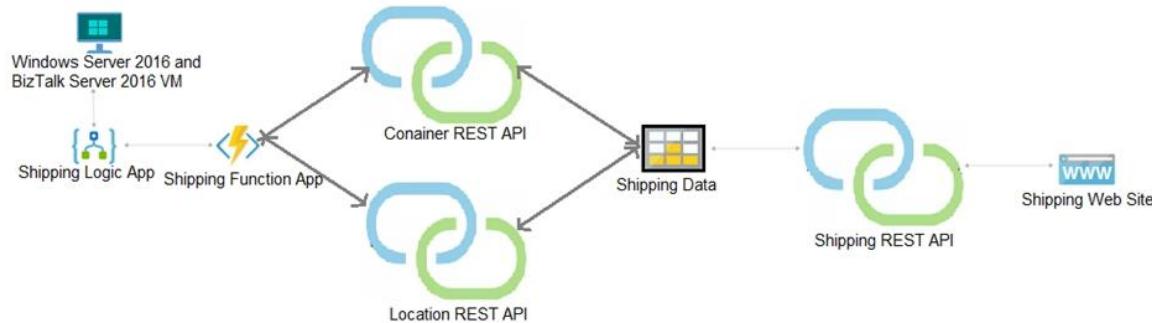
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

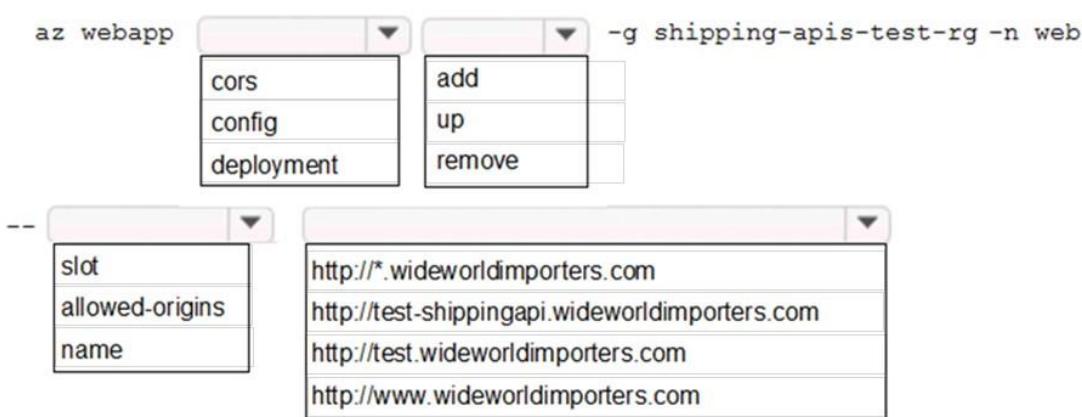
```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

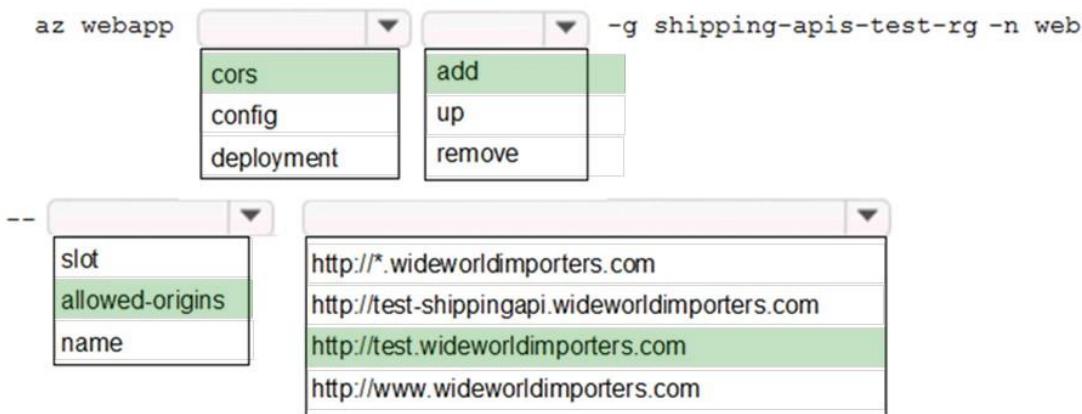
Hotspot Question

You need to update the APIs to resolve the testing error.

How should you complete the Azure CLI command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.



Answer:**Explanation:**

Enable Cross-Origin Resource Sharing (CORS) on your Azure App Service Web App. Enter the full URL of the site you want to allow to access your WEB API or * to allow all domains.

Box 1: cors

Box 2: add

Box 3: allowed-origins

Box 4: `http://testwideworldimporters.com/`

References:

<http://donovanbrown.com/post/How-to-clear-No-Access-Control-Allow-Origin-header-error-with-Azure-AppService>

QUESTION 118**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

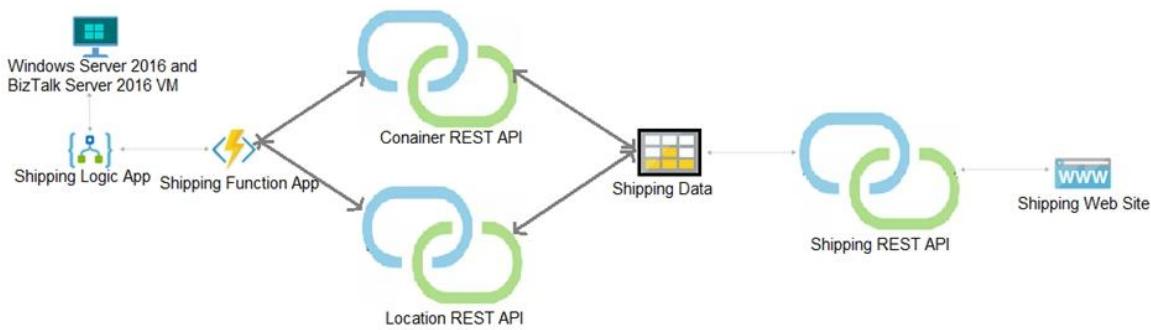
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Shipping Logic App**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

Hotspot Question

You need to correct the VM issues.

Which tools should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Issue**Tool**

Backup and Restore

- Azure Site Recovery
- Azure Backup
- Azure Data Box
- Azure Migrate

Performance

- Azure Network Watcher
- Azure Traffic Manager
- ExpressRoute
- Accelerated Networking

Answer:

Answer Area

| Issue | Tool |
|--------------------|--|
| Backup and Restore | Azure Site Recovery Azure Backup Azure Data Box Azure Migrate |
| Performance | Azure Network Watcher Azure Traffic Manager ExpressRoute Accelerated Networking |

Explanation:

Backup and Restore: Azure Backup

Scenario: The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

In-Place restore of disks in IaaS VMs is a feature of Azure Backup.

Performance: Accelerated Networking

Scenario: The VM shows high network latency, jitter, and high CPU utilization. Accelerated networking enables single root I/O virtualization (SR-IOV) to a VM, greatly improving its networking performance. This high-performance path bypasses the host from the datapath, reducing latency, jitter, and CPU utilization, for use with the most demanding network workloads on supported VM types.

References:

<https://azure.microsoft.com/en-us/blog/an-easy-way-to-bring-back-your-azure-vm-with-in-place-restore/>

QUESTION 119**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

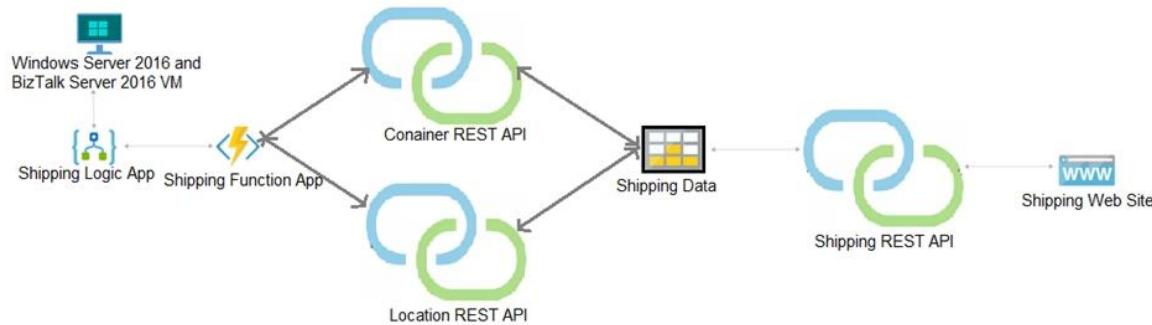
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

Hotspot Question

You need to configure Azure CDN for the Shipping web site.

Which configuration options should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Option | Value |
|--------------|--|
| Tier | <div style="border: 1px solid black; padding: 5px; width: fit-content;">Standard Premium</div> |
| Profile | <div style="border: 1px solid black; padding: 5px; width: fit-content;">Akamai Microsoft</div> |
| Optimization | <div style="border: 1px solid black; padding: 5px; width: fit-content;">general web delivery large file download dynamic site acceleration video-on-demand media streaming</div> |

Answer:

Answer Area

| Option | Value |
|--------------|---|
| Tier | <div style="border: 1px solid black; padding: 5px;"><p>Standard</p><p>Premium</p></div> |
| Profile | <div style="border: 1px solid black; padding: 5px;"><p>Akamai</p><p>Microsoft</p></div> |
| Optimization | <div style="border: 1px solid black; padding: 5px;"><p>general web delivery</p><p>large file download</p><p>dynamic site acceleration</p><p>video-on-demand media streaming</p></div> |

Explanation:

Scenario: Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Tier: Standard

Profile: Akamai

Optimization: Dynamic site acceleration

Dynamic site acceleration (DSA) is available for Azure CDN Standard from Akamai, Azure CDN Standard from Verizon, and Azure CDN Premium from Verizon profiles.

DSA includes various techniques that benefit the latency and performance of dynamic content.

Techniques include route and network optimization, TCP optimization, and more.

You can use this optimization to accelerate a web app that includes numerous responses that aren't cacheable. Examples are search results, checkout transactions, or real-time data. You can continue to use core Azure CDN caching capabilities for static data.

References:

<https://docs.microsoft.com/en-us/azure/cdn/cdn-optimization-overview>

QUESTION 120**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment

Windows Server 2016 virtual machine

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

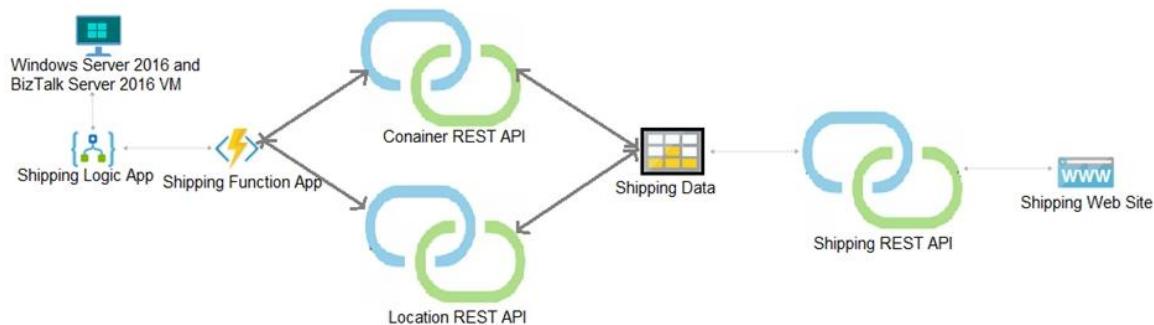
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:



Shipping Logic App

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.

Drag and Drop Question

You need to support the message processing for the ocean transport workflow.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order?

| Actions | Answer Area |
|--|---|
| Create an integration account in the Azure portal. | |
| Link the custom connector to the Logic App. | |
| Update the Logic App to use the partners, schemas, certificates, maps, and agreements. |   |
| Create a custom connector for the Logic App. | |
| Add partners, schemas, certificates, maps, and agreements. | |
| Link the Logic App to the integration account. | |

Answer:

| Actions | Answer Area |
|--|---|
| Create an integration account in the Azure portal. | |
| Link the custom connector to the Logic App. | Link the Logic App to the integration account. |
| Update the Logic App to use the partners, schemas, certificates, maps, and agreements. |   |
| Add partners, schemas, certificates, maps, and agreements. | |
| Create a custom connector for the Logic App. | |

Explanation:

Step 1: Create an integration account in the Azure portal You can define custom metadata for artifacts in integration accounts and get that metadata during runtime for your logic app to use. For example, you can provide metadata for artifacts, such as partners, agreements, schemas, and maps all store metadata using key-value pairs.

Step 2: Link the Logic App to the integration account

A logic app that's linked to the integration account and artifact metadata you want to use.

Step 3: Add partners, schemas, certificates, maps, and agreements

Step 4: Create a custom connector for the Logic App.

References:

<https://docs.microsoft.com/bs-latn-ba/azure/logic-apps/logic-apps-enterprise-integration-metadata>

QUESTION 121

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 are developing a solution that will be deployed to an Azure Kubernetes Service (AKS) cluster. The solution will include a custom VNet, Azure Container Registry images, and an Azure Storage account.

The solution must allow dynamic creation and management of all Azure resources within the AKS cluster.

You need to configure an AKS cluster for use with the Azure APIs.

Solution: Enable the Azure Policy Add-on for Kubernetes to connect the Azure Policy service to the GateKeeper admission controller for the AKS cluster. Apply a built-in policy to the cluster.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead create an AKS cluster that supports network policy. Create and apply a network to allow traffic only from within a defined namespace

References:

<https://docs.microsoft.com/en-us/azure/aks/use-network-policies>

QUESTION 122

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 are developing a solution that will be deployed to an Azure Kubernetes Service (AKS) cluster. The solution will include a custom VNet, Azure Container Registry images, and an Azure Storage account.

The solution must allow dynamic creation and management of all Azure resources within the AKS cluster.

You need to configure an AKS cluster for use with the Azure APIs.

Solution: Create an AKS cluster that supports network policy. Create and apply a network to allow traffic only from within a defined namespace.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

When you run modern, microservices-based applications in Kubernetes, you often want to control which components can communicate with each other. The principle of least privilege should be applied to how traffic can flow between pods in an Azure Kubernetes Service (AKS) cluster. Let's say you likely want to block traffic directly to back-end applications. The Network Policy feature in Kubernetes lets you define rules for ingress and egress traffic between pods in a cluster.

References:

<https://docs.microsoft.com/en-us/azure/aks/use-network-policies>

QUESTION 123

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You are developing and deploying several ASP.Net web applications to Azure App Service. You plan to save session state information and HTML output. You must use a storage mechanism with the following requirements:

- Share session state across all ASP.NET web applications
- Support controlled, concurrent access to the same session state data for multiple readers and a single writer
- Save full HTTP responses for concurrent requests

You need to store the information.

Proposed Solution: Deploy and configure an Azure Database for PostgreSQL. Update the web applications.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

The worst solution from a performance and scalability standpoint is to use a database backed session state provider. Instead use Azure Cache for Redis.

QUESTION 124

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You are developing and deploying several ASP.Net web applications to Azure App Service. You plan to save session state information and HTML output. You must use a storage mechanism with the following requirements:

- Share session state across all ASP.NET web applications
- Support controlled, concurrent access to the same session state data

for multiple readers and a single writer
- Save full HTTP responses for concurrent requests

You need to store the information.

Proposed Solution: Deploy and configure Azure Cache for Redis. Update the web applications.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

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. To use the caching session state provider, first configure your cache, and then configure your ASP.NET application for cache using the Azure Cache for Redis Session State NuGet package.

Reference: <https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-aspnet-session-stateprovider>

QUESTION 125

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Event Hub. Configure the machine identifier as the partition key and enable capture.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-programming-guide>

QUESTION 126

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future. You need to implement a solution to receive the device data.

Solution: Provision an Azure Event Grid. Configure event filtering to evaluate the device identifier.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/event-grid/event-filtering>

QUESTION 127

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Notification Hub. Register all devices with the hub.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead provision an Azure Event Hub. Configure the machine identifier as the partition key and

enable capture.

References:

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-programming-guide>

QUESTION 128

You develop and deploy a Java RESTful API to Azure App Service.

You open a browser and navigate to the URL for the API.

You receive the following error message:

```
Failed to load http://api.azurewebsites.net:6000/#/api/Products: No 'Access-Control-Allow-Origin' header is present on the requested resource.  
Origin 'http://localhost:6000' is therefore not allowed access
```

You need to resolve the error.

What should you do?

- A. Bind an SSL certificate
- B. Enable authentication
- C. Enable CORS
- D. Map a custom domain
- E. Add a CDN

Answer: C

Explanation:

We need to enable Cross-Origin Resource Sharing (CORS).

References:

<https://medium.com/@xinganwang/a-practical-guide-to-cors-51e8fd329a1f>

QUESTION 129

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You are developing and deploying several ASP.Net web applications to Azure App Service. You plan to save session state information and HTML output. You must use a storage mechanism with the following requirements:

- Share session state across all ASP.NET web applications
- Support controlled, concurrent access to the same session state data for multiple readers and a single writer
- Save full HTTP responses for concurrent requests

You need to store the information.

Proposed Solution: Add the web applications to Docker containers. Deploy the containers. Deploy the containers to Azure Kubernetes Service (AKS).

Does the solution meet the goal?

- A. Yes

B. No

Answer: B

Explanation:

Instead use Azure Cache for Redis.

Note: 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. To use the caching session state provider, first configure your cache, and then configure your ASP.NET application for cache using the Azure Cache for Redis Session State NuGet package.

References:

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-aspnet-session-state-provider>

QUESTION 130

You are developing an Azure Cosmos DB solution by using the Azure Cosmos DB SQL API. The data includes millions of documents. Each document may contain hundreds of properties.

The properties of the documents do not contain distinct values for partitioning. Azure Cosmos DB must scale individual containers in the database to meet the performance needs of the application by spreading the workload evenly across all partitions over time.

You need to select a partition key.

Which two partition keys can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. a concatenation of multiple property values with a random suffix appended
- B. a single property value that does not appear frequently in the documents
- C. a hash suffix appended to a property value
- D. a value containing the collection name
- E. a single property value that appears frequently in the documents

Answer: AC

Explanation:

You can form a partition key by concatenating multiple property values into a single artificial partitionKey property. These keys are referred to as synthetic keys.

Another possible strategy to distribute the workload more evenly is to append a random number at the end of the partition key value. When you distribute items in this way, you can perform parallel write operations across partitions.

Note: It's the best practice to have a partition key with many distinct values, such as hundreds or thousands. The goal is to distribute your data and workload evenly across the items associated with these partition key values. If such a property doesn't exist in your data, you can construct a synthetic partition key.

References:

<https://docs.microsoft.com/en-us/azure/cosmos-db/synthetic-partition-keys>

QUESTION 131

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 are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.

You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level. You need to configure authorization.

Solution:

Create a new Azure AD application's manifest, set value of the groupMembershipClaims option to All.

In the website, use the value of the groups claim from the JWI for the user to determine permissions.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

To configure Manifest to include Group Claims in Auth Token

1. Go to Azure Active Directory to configure the Manifest. Click on Azure Active Directory, and go to App registrations to find your application:

2. Click on your application (or search for it if you have a lot of apps) and edit the Manifest by clicking on it.

3. Locate the "groupMembershipClaims" setting. Set its value to either "SecurityGroup" or "All".

To help you decide which:

"SecurityGroup" groups claim will contain the identifiers of all security groups of which the user is a member.

"All" groups claim will contain the identifiers of all security groups and all distribution lists of which the user is a member

Now your application will include group claims in your manifest and you can use this fact in your code.

References:

<https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/>

QUESTION 132

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 are developing a website that will run as an Azure Web App. Users will authenticate by using their Azure Active Directory (Azure AD) credentials.

You plan to assign users one of the following permission levels for the website: admin, normal, and reader. A user's Azure AD group membership must be used to determine the permission level. You need to configure authorization.

Solution: Configure the Azure Web App for the website to allow only authenticated requests and require Azure AD log on.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead in the Azure AD application's manifest, set value of the groupMembershipClaims option to All.

References:

<https://blogs.msdn.microsoft.com/waws/2017/03/13/azure-app-service-authentication-aad-groups/>

QUESTION 133

Drag and Drop Question

You are preparing to deploy an Azure virtual machine (VM)-based application. The VMs that run the application have the following requirements:

- When a VM is provisioned the firewall must be automatically configured before it can access Azure resources
- Supporting services must be installed by using an Azure PowerShell script that is stored in Azure Storage

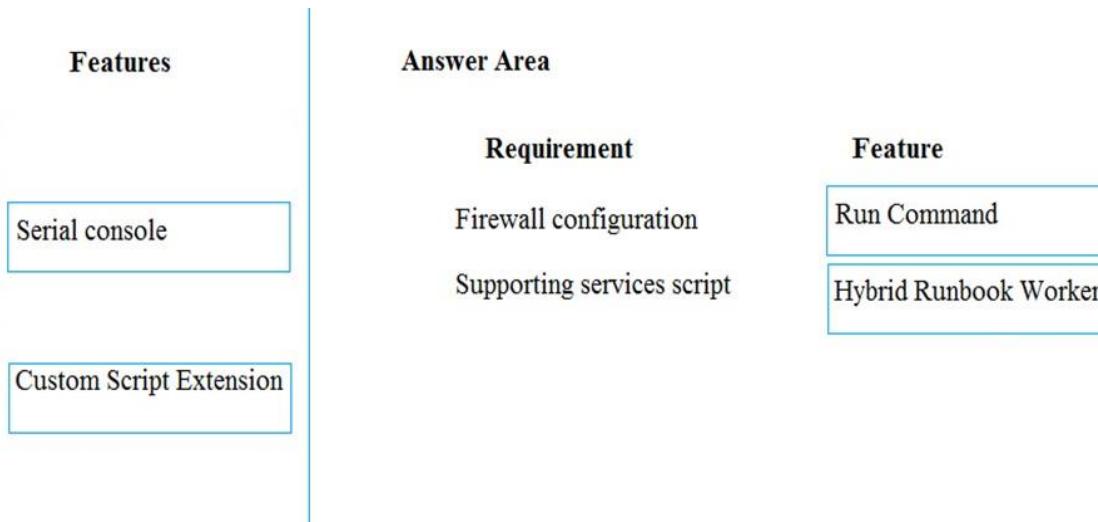
You need to ensure that the requirements are met.

Which features should you use? To answer, drag the appropriate features to the correct requirements. Each feature 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.

| Features | Answer Area | |
|-------------------------|----------------------------|---------|
| | Requirement | Feature |
| Run Command | Firewall configuration | |
| Serial console | Supporting services script | |
| Hybrid Runbook Worker | | |
| Custom Script Extension | | |

Answer:

**Explanation:**

<https://docs.microsoft.com/en-us/azure/automation/automation-hybrid-runbook-worker>
<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/run-command>

QUESTION 134

Drag and Drop Question

You are developing a microservices solution. You plan to deploy the solution to a multinode Azure Kubernetes Service (AKS) cluster.

You need to deploy a solution that includes the following features:

- reverse proxy capabilities
- configurable traffic routing
- TLS termination with a custom certificate

Which components should you use? To answer, drag the appropriate components to the correct requirements. Each component 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.

| Components | Answer Area | |
|--------------------|--|-----------|
| | Action | Component |
| Helm | Deploy solution. | Component |
| Draft | View cluster and external IP addressing. | Component |
| Brigade | | |
| KubeCtl | | |
| Ingress Controller | Implement a single, public IP endpoint that is routed to multiple microservices. | Component |
| CoreDNS | | |
| Virtual Kubelet | | |

Answer:

| Components | Answer Area | |
|-----------------|--|--------------------|
| | Action | Component |
| Draft | Deploy solution. | Helm |
| Brigade | View cluster and external IP addressing. | KubeCtl |
| | | |
| CoreDNS | Implement a single, public IP endpoint that is routed to multiple microservices. | Ingress Controller |
| Virtual Kubelet | | |

Explanation:

Box 1: Helm

To create the ingress controller, use Helm to install nginx-ingress.

Box 2: kubectl

To find the cluster IP address of a Kubernetes pod, use the kubectl get pod command on your local machine, with the option -o wide .

Box 3: Ingress Controller

An ingress controller is a piece of software that provides reverse proxy, configurable traffic routing, and TLS termination for Kubernetes services. Kubernetes ingress resources are used to configure the ingress rules and routes for individual Kubernetes services.

Incorrect Answers:

Virtual Kubelet: Virtual Kubelet is an open-source Kubernetes kubelet implementation that masquerades as a kubelet. This allows Kubernetes nodes to be backed by Virtual Kubelet providers such as serverless cloud container platforms.

CoreDNS: CoreDNS is a flexible, extensible DNS server that can serve as the Kubernetes cluster DNS. Like Kubernetes, the CoreDNS project is hosted by the CNCF.

Reference:

<https://docs.microsoft.com/bs-cyrl-ba/azure/aks/ingress-basic>

<https://www.digitalocean.com/community/tutorials/how-to-inspect-kubernetes-networking>

QUESTION 135

Hotspot Question

You are configuring a development environment for your team. You deploy the latest Visual Studio image from the Azure Marketplace to your Azure subscription.

The development environment requires several software development kits (SDKs) and third-party components to support application development across the organization. You install and customize the deployed virtual machine (VM) for your development team. The customized VM must be saved to allow provisioning of a new team member development environment.

You need to save the customized VM for future provisioning. Which tools or services should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Action | Tool or service |
|--------------------|---|
| Generalize the VM. | Azure Power Shell Visual Studio command prompt Azure Migrate Azure Backup |
| Store images. | Azure Blob Storage Visual Data Lake Storage Azure File Storage Azure Table Storage |

Answer:

Answer Area

| Action | Tool or service |
|--------------------|---|
| Generalize the VM. | Azure Power Shell Visual Studio command prompt Azure Migrate Azure Backup |
| Store images. | Azure Blob Storage Visual Data Lake Storage Azure File Storage Azure Table Storage |

Explanation:

Box 1: Azure Powershell

Creating an image directly from the VM ensures that the image includes all of the disks associated with the VM, including the OS disk and any data disks.

Before you begin, make sure that you have the latest version of the Azure PowerShell module. You use Sysprep to generalize the virtual machine, then use Azure PowerShell to create the image.

Box 2: Azure Blob Storage

References:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/capture-image-resource#create-an-image-of-a-vm-using-powershell>

QUESTION 136

Drag and Drop Question

You are preparing to deploy an application to an Azure Kubernetes Service (AKS) cluster. The application must only be available from within the VNet that includes the cluster.

You need to deploy the application.

How should you complete the deployment YAML? To answer, drag the appropriate YAML segments to the correct locations. Each YAML 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.

Code segments

Ingress
Service
LoadBalancer
Deployment
ingress.class
azure-load-balancer-internal

Answer Area

```
apiVersion: v1
kind: Code segment
metadata:
  name: web-app
  annotations:
    service.beta.kubernetes. Code segment : "true"
spec:
  type: Code segment
  ports:
  - port: 80
  selector:
    app: web-app
```

Answer:

Code segments

Ingress

Deployment
ingress.class

Answer Area

```
apiVersion: v1
kind: Service
metadata:
  name: web-app
  annotations:
    service.beta.kubernetes.azure-load-balancer-internal: "true"
spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: web-app
```

Explanation:

To create an internal load balancer, create a service manifest named internal-lb.yaml with the service type LoadBalancer and the azure-load-balancer-internal annotation as shown in the following example:

YAML:

```
apiVersion: v1
kind: Service
metadata:
  name: internal-app
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: "true"
spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: internal-app
  References:
    https://docs.microsoft.com/en-us/azure/aks/internal-lb
```

QUESTION 137

Hotspot Question

A company is developing a Node.js web app. The web app code is hosted in a GitHub repository located at <https://github.com/TailSpinToys/webapp>.

The web app must be reviewed before it is moved to production. You must deploy the initial code release to a deployment slot named review.

You need to create the web app and deploy the code.

How should you complete the commands? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
$gitrepo="https://github.com/TailSpinToys/webapp"
$webappname="TailSpinToysWeb"
$location="WestUS2"

New-AzResourceGroup -Name myResourceGroup -Location $location
New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup

New-AzWebAppSlot -Name $webappname -Location $location -ResourceGroupName myResourceGroup -Tier Standard
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup

New-AzWebAppSlot -Name $webappname -Location $location -AppServicePlan $webappname -ResourceGroupName myResourceGroup
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup

New-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup -Slot review
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup

@PropertiesObject = @{repoUrl = "$gitrepo";branch = "master";}
Set-AzResource -PropertyObject $PropertiesObject -ResourceGroupName myResourceGroup -ResourceType
Microsoft.Web/sites/slots/sourcecontrols -ResourceName $webappname/review/web -ApiVersion 2015-08-01 -Force
Switch-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup `
-SourceSlotName review -DestinationSlotName production
```

Answer:

Answer Area

```
$gitrepo="https://github.com/TailSpinToys/webapp"
$webappname="TailSpinToysWeb"
$location="WestUS2"

 -Name myResourceGroup -Location $location
 New-AzWebAppSlot
 New-AzWebApp
 New-AzAppServicePlan
 New-AzResourceGroup

 -Name $webappname -Location $location -ResourceGroupName myResourceGroup -Tier Standard
 New-AzWebAppSlot
 New-AzWebApp
 New-AzAppServicePlan
 New-AzResourceGroup

 -Name $webappname -Location $location -AppServicePlan $webappname -ResourceGroupName myResourceGroup
 New-AzWebAppSlot
 New-AzWebApp
 New-AzAppServicePlan
 New-AzResourceGroup

 -Name $webappname -ResourceGroupName myResourceGroup -Slot review
 New-AzWebAppSlot
 New-AzWebApp
 New-AzAppServicePlan
 New-AzResourceGroup

@PropertiesObject = @{repoUrl = "$gitrepo";branch = "master";}
Set-AzResource -PropertyObject $PropertiesObject -ResourceGroupName myResourceGroup -ResourceType
Microsoft.Web/sites/slots/sourcecontrols -ResourceName $webappname/review/web -ApiVersion 2015-08-01 -Force
Switch-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup -
-SourceSlotName review -DestinationSlotName production
```

Explanation:

The New-AzResourceGroup cmdlet creates an Azure resource group.

The New-AzAppServicePlan cmdlet creates an Azure App Service plan in a given location

The New-AzWebApp cmdlet creates an Azure Web App in a given a resource group

The New-AzWebAppSlot cmdlet creates an Azure Web App slot.

QUESTION 138

Hotspot Question

You are implementing a software as a service (SaaS) ASP.NET Core web service that will run as an Azure Web App. The web service will use an on-premises SQL Server database for storage. The web service also includes a WebJob that processes data updates. Four customers will use the web service.

- Each instance of the WebJob processes data for a single customer and must run as a singleton instance.
- Each deployment must be tested by using deployment slots prior to serving production data.
- Azure costs must be minimized.
- Azure resources must be located in an isolated network.

You need to configure the App Service plan for the Web App.

How should you configure the App Service plan? To answer, select the appropriate settings in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| App service plan setting | Value | | | | |
|---------------------------------|--|----------|----------|---------|-------------|
| Number of VM instances | <input type="button" value="▼"/> <table border="1"><tr><td>2</td></tr><tr><td>4</td></tr><tr><td>8</td></tr><tr><td>16</td></tr></table> | 2 | 4 | 8 | 16 |
| 2 | | | | | |
| 4 | | | | | |
| 8 | | | | | |
| 16 | | | | | |
| Pricing tier | <input type="button" value="▼"/> <table border="1"><tr><td>Isolated</td></tr><tr><td>Standard</td></tr><tr><td>Premium</td></tr><tr><td>Consumption</td></tr></table> | Isolated | Standard | Premium | Consumption |
| Isolated | | | | | |
| Standard | | | | | |
| Premium | | | | | |
| Consumption | | | | | |

Answer:**Answer Area**

| App service plan setting | Value | | | | |
|---------------------------------|--|----------|----------|---------|-------------|
| Number of VM instances | <input type="button" value="▼"/> <table border="1"><tr><td>2</td></tr><tr><td>4</td></tr><tr><td>8</td></tr><tr><td>16</td></tr></table> | 2 | 4 | 8 | 16 |
| 2 | | | | | |
| 4 | | | | | |
| 8 | | | | | |
| 16 | | | | | |
| Pricing tier | <input type="button" value="▼"/> <table border="1"><tr><td>Isolated</td></tr><tr><td>Standard</td></tr><tr><td>Premium</td></tr><tr><td>Consumption</td></tr></table> | Isolated | Standard | Premium | Consumption |
| Isolated | | | | | |
| Standard | | | | | |
| Premium | | | | | |
| Consumption | | | | | |

Explanation:

Number of VM instances: 4

You are not charged extra for deployment slots.

Pricing tier: Isolated

The App Service Environment (ASE) is a powerful feature offering of the Azure App Service that gives network isolation and improved scale capabilities. It is essentially a deployment of the Azure App Service into a subnet of a customer's Azure Virtual Network (VNet).

References:

[https://azure.microsoft.com/sv-se/blog/announcing-app-service-isolated-more-power-scale-and-ease-ofuse/](https://azure.microsoft.com/sv-se/blog/announcing-app-service-isolated-more-power-scale-and-ease-of-use/)

QUESTION 139

Hotspot Question

You are building a website to access project data related to terms within your organization. The website does not allow anonymous access. Authentication performed using an Azure Active Directory (Azure AD) app named internal.

The website has the following authentication requirements:

- Azure AD users must be able to login to the website.
- Personalization of the website must be based on membership in Active Directory groups.

You need to configure the application's manifest to meet the authentication requirements.

How should you configure the manifest? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
{  
    ...  
    "appId": "d61126e3-089b-4adb-b721-  
d5023213df7d",  
     : "All",  


"optionalClaims"  
        "groupMembershipClaims"

  
     : true  


"allowPublicClient"  
        "oauth2Permissions"  
        "requiredResourceAccess"  
        "oauth2AllowImplicitFlow"

  
    ...  
}
```

Answer:

Answer Area

```
{  
    ...  
    "appId": "d61126e3-089b-4adb-b721-  
d5023213df7d",  
    "optionalClaims": "All",  
    "groupMembershipClaims":  
    "allowPublicClient": true  
    "oauth2Permissions":  
    "requiredResourceAccess":  
    "oauth2AllowImplicitFlow":  
    ...  
}
```

Explanation:

Box 1: groupMembershipClaims

Scenario: Personalization of the website must be based on membership in Active Directory groups.

Group claims can also be configured in the Optional Claims section of the Application Manifest.

Enable group membership claims by changing the groupMembershipClaim

The valid values are:

"All"

"SecurityGroup"

"DistributionList"

"DirectoryRole"

Box 2: oauth2Permissions

Scenario: Azure AD users must be able to login to the website.

oauth2Permissions specifies the collection of OAuth 2.0 permission scopes that the web API (resource) app exposes to client apps. These permission scopes may be granted to client apps during consent.

Incorrect Answers:

oauth2AllowImplicitFlow. oauth2AllowImplicitFlow specifies whether this web app can request OAuth2.0 implicit flow access tokens. The default is false. This flag is used for browser-based

apps, like Javascript single-page apps.

References:

<https://docs.microsoft.com/en-us/azure/active-directory/hybrid/how-to-connect-fed-group-claims>

QUESTION 140

Drag and Drop Question

You are developing an ASP.NET Core website that can be used to manage photographs which are stored in Azure Blob Storage containers.

Users of the website authenticate by using their Azure Active Directory (Azure AD) credentials.

You implement role-based access control (RBAC) role permission on the containers that store photographs. You assign users to RBAC role.

You need to configure the website's Azure AD Application so that user's permissions can be used with the Azure Blob containers.

How should you configure the application? To answer, drag the appropriate setting to the correct location. Each setting 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.

| Settings | | Answer Area | | |
|--------------------|--|-----------------|------------|---------|
| | | API | Permission | Type |
| client_id | | Azure Storage | Setting | Setting |
| delegated | | Microsoft Graph | User.Read | Setting |
| profile | | | | |
| application | | | | |
| user_impersonation | | | | |

Answer:

| Settings | Answer Area | | |
|--------------------|---------------|--------------------|-----------|
| | API | Permission | Type |
| client_id | Azure Storage | user_impersonation | delegated |
| delegated | | User.Read | delegated |
| profile | | | |
| application | | | |
| user_impersonation | | | |

Explanation:

Box 1: user_impersonation

Box 2: delegated

Example:

1. Select the API permissions section
2. Click the Add a permission button and then:
Ensure that the My APIs tab is selected
3. In the list of APIs, select the API TodoListService-aspnetcore.
4. In the Delegated permissions section, ensure that the right permissions are checked:
user_impersonation.
5. Select the Add permissions button.

Box 3: delegated

Example

1. Select the API permissions section
2. Click the Add a permission button and then,
Ensure that the Microsoft APIs tab is selected
3. In the Commonly used Microsoft APIs section, click on Microsoft Graph
4. In the Delegated permissions section, ensure that the right permissions are checked:
User.Read. Use the search box if necessary.
5. Select the Add permissions button

References:

<https://docs.microsoft.com/en-us/samples/azure-samples/active-directory-dotnet-webapp-webapiopenidconnect-aspnetcore/calling-a-web-api-in-an-aspnet-core-web-application-using-azure-ad/>

QUESTION 141

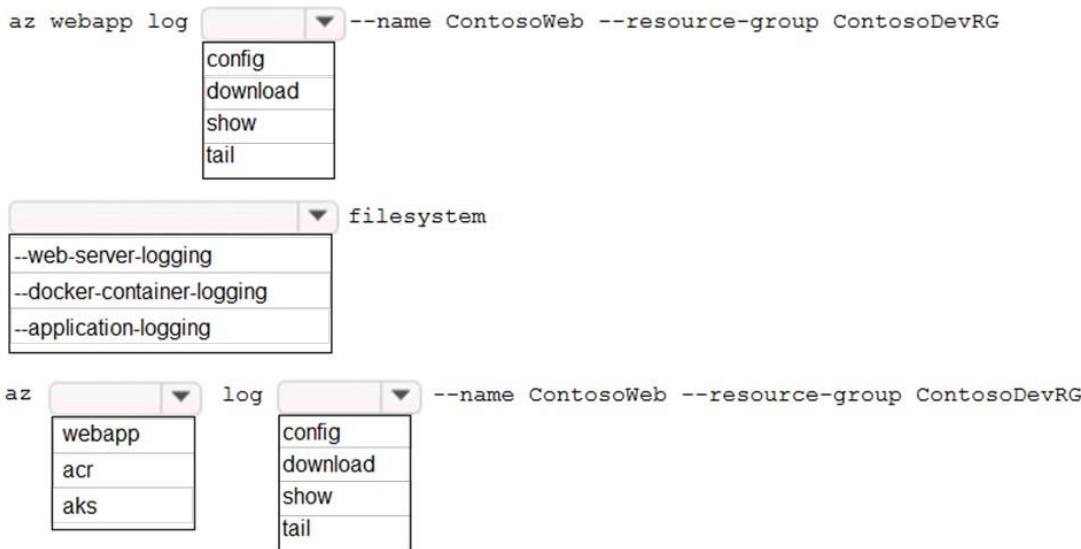
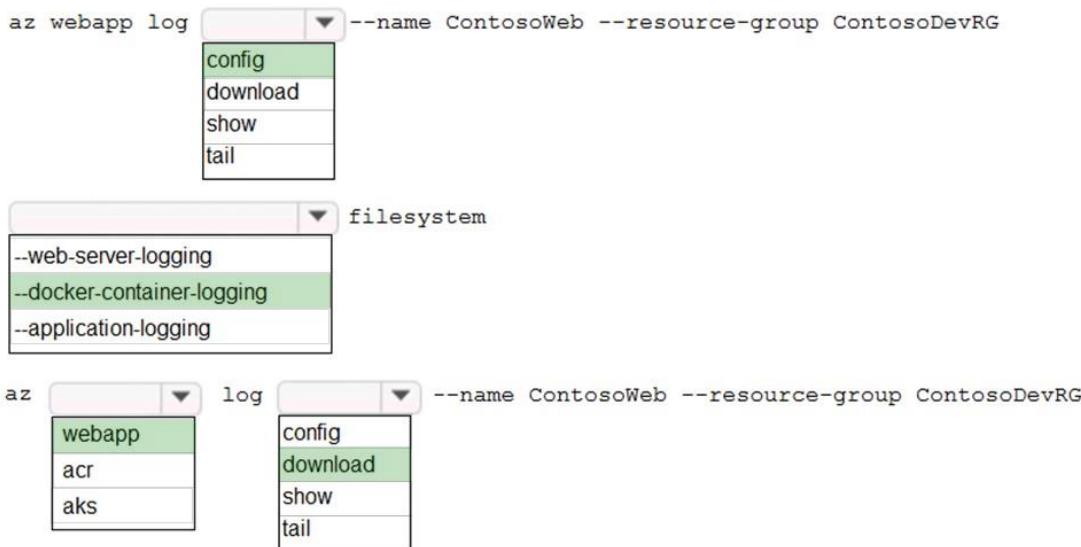
Hotspot Question

You plan to deploy a web app to App Service on Linux. You create an App Service plan. You create and push a custom Docker image that contains the web app to Azure Container Registry.

You need to access the console logs generated from inside the container in real-time.

How should you complete the Azure CLI command? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area**Answer:****Answer Area****Explanation:**

Box 1: config

To Configure logging for a web app use the command:

`az webapp log config`

Box 2: `--docker-container-logging`

Syntax include:

`az webapp log config [--docker-container-logging {filesystem, off}]`

Box 3: `webapp`

To download a web app's log history as a zip file use the command:

az webapp log download

Box 4: download

References:

<https://docs.microsoft.com/en-us/cli/azure/webapp/log>

QUESTION 142

A team has created an Index in the Azure Search service. You have to upload data into the Index. You propose the following steps to carry out from your .Net program

- Create a SearchServiceClient object to connect to the search index.
- Create a DataContainer that contains the documents which must be added.
- Create a DataSource instance and set its Container property to the DataContainer.
- Set the DataSource property of the SearchServiceClient

Does the list of steps fulfil the requirement?

- A. Yes
- B. No

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/search/search-what-is-azure-search#how-to-use-azure-search>

QUESTION 143

A team has created an Index in the Azure Search service. You have to upload data into the Index. You propose the following steps to carry out from your .Net program

- Create a SearchIndexClient object to connect to the search index
- Create an IndexBatch that contains the documents which must be added.
- Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.

Does the list of steps fulfil the requirement?

- A. Yes
- B. No

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/azure/search/search-import-data-dotnet>

In order to push documents into your index using the .NET SDK, you will need to:

1. Create a `SearchIndexClient` object to connect to your search index.
2. Create an `IndexBatch` containing the documents to be added, modified, or deleted.
3. Call the `Documents.Index` method of your `SearchIndexClient` to send the `IndexBatch` to your search index.

QUESTION 144

A team has created an Index in the Azure Search service. You have to upload data into the Index. You propose the following steps to carry out from your .Net program

- Create a SearchIndexClient object to connect to the search index.
- Create a DataContainer that contains the documents which must be added.
- Create a DataSource instance and set its Container property to the DataContainer
- Call the Documents.Search method of the SearchIndexClient and pass the DataSource.

Does the list of steps fulfil the requirement?

- A. Yes
- B. No

Answer: B

Explanation:

In order to push documents into your index using the .NET SDK, you will need to:

1. Create a `SearchIndexClient` object to connect to your search index.
2. Create an `IndexBatch` containing the documents to be added, modified, or deleted.
3. Call the `Documents.Index` method of your `SearchIndexClient` to send the `IndexBatch` to your search index.

<https://docs.microsoft.com/en-us/azure/search/search-import-data-dotnet>

QUESTION 145

A team is developing container-based applications that need to be deployed to a Kubernetes cluster in Azure. You have to create the cluster and ensure the services are running as desired. Which of the following commands would you execute? Choose 4 answers from the options given below

- A. az aks create
- B. az group create
- C. kubectl apply
- D. az appservice plan create
- E. az aks get-credentials

Answer: ABCE

Explanation:

<https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough>

QUESTION 146

A development team is developing an application. The application will be storing its data in Azure Table storage. Below are the fields that are going to be stored in the table

- Region
- Email address
- Phone number

The following snippet of code needs to be completed that would be used to insert a batch of records.

```
private static void InsertBatch()
{
    CloudStorageAccount whizlabs_storage = CloudStorageAccount.Parse(conn_string);
    CloudTableClient whizlabs_table_client = whizlabs_storage.CreateCloudTableClient();
    CloudTable whizlabs_table = whizlabs_table_client.GetTableReference("Customer");

    Slot1 whizlabs_batch = new Slot2

    Customer customer_obj1 = new Customer(4, "May");
    customer_obj1.Email = "May@whizlabs.com";

    Customer customer_obj2 = new Customer(4, "Carrie");
    customer_obj2.Email = "Carrie@whizlabs.com";

    whizlabs_batch.Insert(customer_obj1);
    whizlabs_batch.Insert(customer_obj2);

    whizlabs_table. Slot3 (whizlabs_batch);

    Console.WriteLine("Records Inserted");

    Console.ReadKey();
}
```

Which of the following will go into Slot1?

- A. TableOperation
- B. TableBatchOperation
- C. TableEntity
- D. TableQuery

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet>

QUESTION 147

A development team is developing an application. The application will be storing its data in Azure Table storage. Below are the fields that are going to be stored in the table

- Region
- Email address
- Phone number

The following snippet of code needs to be completed that would be used to insert a batch of records.

```
private static void InsertBatch()
{
    CloudStorageAccount whizlabs_storage = CloudStorageAccount.Parse(conn_string);
    CloudTableClient whizlabs_table_client = whizlabs_storage.CreateCloudTableClient();
    CloudTable whizlabs_table = whizlabs_table_client.GetTableReference("Customer");

    Slot1 whizlabs_batch = new Slot2

    Customer customer_obj1 = new Customer(4, "May");
    customer_obj1.Email = "May@whizlabs.com";

    Customer customer_obj2 = new Customer(4, "Carrie");
    customer_obj2.Email = "Carrie@whizlabs.com";

    whizlabs_batch.Insert(customer_obj1);
    whizlabs_batch.Insert(customer_obj2);

    whizlabs_table. Slot3 (whizlabs_batch);

    Console.WriteLine("Records Inserted");

    Console.ReadKey();
}
```

Which of the following will go into Slot2?

- A. TableOperation
- B. TableBatchOperation
- C. TableEntity
- D. TableQuery

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet>

QUESTION 148

A development team is developing an application. The application will be storing its data in Azure Table storage. Below are the fields that are going to be stored in the table

- Region
- Email address
- Phone number

The following snippet of code needs to be completed that would be used to insert a batch of records.

```
private static void InsertBatch()
{
    CloudStorageAccount whizlabs_storage = CloudStorageAccount.Parse(conn_string);
    CloudTableClient whizlabs_table_client = whizlabs_storage.CreateCloudTableClient();
    CloudTable whizlabs_table = whizlabs_table_client.GetTableReference("Customer");

    Slot1 whizlabs_batch = new Slot2

    Customer customer_obj1 = new Customer(4, "May");
    customer_obj1.Email = "May@whizlabs.com";

    Customer customer_obj2 = new Customer(4, "Carrie");
    customer_obj2.Email = "Carrie@whizlabs.com";

    whizlabs_batch.Insert(customer_obj1);
    whizlabs_batch.Insert(customer_obj2);

    whizlabs_table. Slot3 (whizlabs_batch);

    Console.WriteLine("Records Inserted");

    Console.ReadKey();
}
```

Which of the following will go into Slot3?

- A. ExecuteBatch
- B. Execute
- C. Insert
- D. InsertOrMerge

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet>

QUESTION 149

A development team is developing an application. The application will be working with customer data.

The application will also be making use of Azure Redis Cache.

You need to invalidate the cache when the customer data is changed.

You have to complete the below code to comply with the requirement

```
void clearCustomerCache(string p_Customer)
{
    //Establish the cache connection
    Slot1
    //Invalidate the cache
    Slot2
}
```

Which of the following will go into Slot1?

- A. IDatabase cache=Connection.GetDatabase();
- B. IDatabase cache=Connection.GetCache();
- C. ICache cache=Connection.GetDatabase();
- D. ICache cache=Connection.GetCache();

Answer: A

Explanation:

The right way is to use the IDatabase interface. Also you need to use the GetDatabase() method. This is also mentioned in the Microsoft documentation.

```
static void Main(string[] args)
{
    // Connection refers to a property that returns a ConnectionMultiplexer
    // as shown in the previous example.
    IDatabase cache = lazyConnection.Value.GetDatabase();
```

Since this is clearly given in the Microsoft documentation, all other options are incorrect.

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-dotnet-how-to-use-azure-redis-cache>

QUESTION 150

A development team is developing an application. The application will be working with customer data. The application will also be making use of Azure Redis Cache. You need to invalidate the cache when the customer data is changed.

You have to complete the below code to comply with the requirement

```
void clearCustomerCache(string p_Customer)
{
    //Establish the cache connection
    Slot1
    //Invalidate the cache
    Slot2
}
```

Which of the following will go into Slot2?

- A. cache.KeyDelete(p_Customer);
- B. cache.ValueDelete(p_Customer);
- C. cache.StringGet(p_Customer);
- D. cache.StringSet(p_Customer);

Answer: A

Explanation:

Since you have to invalidate the cache, you have to delete the Key itself

Option B is incorrect since you need to work with keys and not the values

Option C is incorrect this is used to get the string value

Option D is incorrect this is used to set the string value

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-dotnet-how-to-use-azure-redis-cache>

QUESTION 151

A development team is developing an application that works with Azure Table storage.

| Column | |
|-----------|---------------|
| FirstName | Partition Key |
| LastName | RowKey |
| Email | Property |

Below are some of the rows in the table

| Azure Table Storage - Customer | | | |
|--------------------------------|--------|------------------------------|--------------------|
| PartitionKey | RowKey | Timestamp | Email |
| James | Smith | 2019-03-06T10:39:17.263Z | Smith@whizlabs.com |
| John | Smith | 2019-03-06T10:29:18.0667815Z | Smith@whizlabs.com |
| Mark | Carter | 2019-03-06T10:29:34.843Z | Mark@whizlabs.com |

You have the following code statement from a C# program

```
TableQuery<CustomerEntity> rangeQuery = new TableQuery<CustomerEntity>().Where(
    TableQuery.CombineFilters(
        TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "James"),
        TableOperators.And,
        TableQuery.GenerateFilterCondition("RowKey", QueryComparisons.Equal, "Smith@whizlabs.com")));
```

Would this return all the entities where the RowKey is Smith@whizlabs.com?

- A. Yes
- B. No

Answer: B

Explanation:

<https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet>

QUESTION 152

A development team is developing an application that works with Azure Table storage.

| Column | |
|-----------|---------------|
| FirstName | Partition Key |
| LastName | RowKey |
| Email | Property |

Below are some of the rows in the table

| Customer | | | |
|--------------|--------|------------------------------|--------------------|
| PARTITIONKEY | ROWKEY | TIMESTAMP | EMAIL |
| James | Smith | 2019-03-06T10:39:17.263Z | Smith@whizlabs.com |
| John | Smith | 2019-03-06T10:29:18.0667815Z | Smith@whizlabs.com |
| Mark | Carter | 2019-03-06T10:29:34.843Z | Mark@whizlabs.com |

Is the below .Net Code query

```
TableQuery<CustomerEntity> rangeQuery = new TableQuery<CustomerEntity>().Where(
    TableQuery.CombineFilters(
        TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "James"),
        TableOperators.And,
        TableQuery.GenerateFilterCondition("RowKey", QueryComparisons.Equal, "Smith")));
```

Same as executing the below REST API call along with a valid Shared Access Signature
[https://whizlabsstore.table.core.windows.net/Customer\(PartitionKey='James',RowKey='Smith'\)](https://whizlabsstore.table.core.windows.net/Customer(PartitionKey='James',RowKey='Smith'))

- A. Yes
- B. No

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/rest/api/storageservices/querying-tables-and-entities>

QUESTION 153

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You are developing and deploying several ASP.NET web applications to Azure App Service. You plan to save session state information and HTML output. You must use a storage mechanism with the following requirements:

- Share session state across all ASP.NET web applications
- Support controlled, concurrent access to the same session state data for multiple readers and a single writer
- Save full HTTP responses for concurrent requests

You need to store the information.

Proposed Solution: Enable Application Request Routing (ARR)

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead deploy and configure Azure Cache for Redis. Update the web applications.

Reference:

<https://docs.microsoft.com/en-us/azure/architecture/best-practices/caching#managing-concurrency-in-a-cache>

QUESTION 154

Hotspot Question

You have a web service that is used to pay for food deliveries. The web service uses Azure Cosmos DB as the data store.

You plan to add a new feature that allows users to set a tip amount. The new feature requires that a property named tip on the document in Cosmos DB must be present and contain a numeric value.

There are many existing websites and mobile apps that use the web service that will not be updated to set the tip property for some time.

How should you complete the trigger?

NOTE: Each correct selection is worth one point.

Answer Area

```
function ensureTip() {  
    var r = _.value();  
_.readDocument('item');  
getContext().getRequest();  
getContext().getResponse();  
    var i = r.getBody();  
  
    if (!("tip" in i)) {  
        if (request.getValue("tip") === null){  
            if (isNaN(i["tip"]) || i["tip"] === null) {  
                if (typeof_.pluck("tip") == 'number') {  
                    i["tip"] = 0;  
                }  
            }  
        }  
  
        r.setBody(i);  
        r.setValue(i);  
        _.upsertDocument(i);  
_.replaceDocument(i)  
    }  
}
```

Answer:

Answer Area

```
function ensureTip() {
    var r =
        __.value();
        __.readDocument('item');
        getContext().getRequest();
        getContext().getResponse();

    var i = r.getBody();

    if (!("tip" in i)) {
        if (request.getValue("tip") === null) {
            if (isNaN(i["tip"]) || i["tip"] === null) {
                if (typeof __.pluck("tip") === 'number') {

                    i["tip"] = 0;
                }
            }
        }
    }

    r.setBody(i);
    r.setValue(i);
    __.upsertDocument(i);
    __.replaceDocument(i)
}
```

Explanation:

Box 1: getContext().getRequest();

Box 2: if(isNaN(i)["tip"] ..

In JavaScript, there are two ways to check if a variable is a number :

isNaN() ?Stands for "is Not a Number", if variable is not a number, it return true, else return false.
typeof ?If variable is a number, it will returns a string named "number".

Box 3:r.setBody(i);

// update the item that will be created

References:

<https://docs.microsoft.com/en-us/bs-latn-ba/azure/cosmos-db/how-to-write-stored-procedures-triggers-udfs>

<https://mkyong.com/javascript/check-if-variable-is-a-number-in-javascript/>

QUESTION 155

Hotspot Question

You are developing an application that uses an Azure blob named `data` to store application data. The application creates blob snapshots to allow application state to be reverted to an earlier state. The Azure storage account has soft delete enabled.

The system performs the following operations in order:

- The blob is updated
- Snapshot 1 is created.
- Snapshot 2 is created.
- Snapshot 1 is deleted.

A system error then deletes the data blob and all snapshots.

You need to determine which application states can be restored.

What is the restorability of the application data? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Application State | Restorability |
|-------------------|--|
| Data blob | <input type="checkbox"/> Can be restored <input checked="" type="checkbox"/> Cannot be restored |
| Snapshot 1 | <input type="checkbox"/> Can be restored <input checked="" type="checkbox"/> Cannot be restored |
| Snapshot 2 | <input type="checkbox"/> Can be restored <input checked="" type="checkbox"/> Cannot be restored |

Answer:

Answer Area

Application State Restorability

| Data blob | Restorability |
|------------|---------------------------------------|
| | Can be restored Cannot be restored |
| Snapshot 1 | Can be restored Cannot be restored |
| Snapshot 2 | Can be restored Cannot be restored |

Explanation:

Box 1: Can be restored

When enabled, soft delete enables you to save and recover your data when blobs or blob snapshots are deleted. This protection extends to blob data that is erased as the result of an overwrite.

Box 2: Cannot be restored

It has been deleted.

Box 3: Can be restored

It has not been deleted.

References:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-soft-delete>

QUESTION 156

Hotspot Question

You are building a traffic monitoring system that monitors traffic along six highways. The system produces time series analysis-based reports for each highway. Data from traffic sensors are stored in Azure Event Hub.

Traffic data is consumed by four departments. Each department has an Azure Web App that displays the time-series-based reports and contains a WebJob that processes the incoming data from Event Hub. All Web Apps run on App Service Plans with three instances.

Data throughput must be maximized. Latency must be minimized.

You need to implement the Azure Event Hub.

Which settings should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|---------|-------|
|---------|-------|

Number of partitions

| |
|----|
| 3 |
| 4 |
| 6 |
| 12 |

Partition Key

| |
|------------|
| Highway |
| Department |
| Timestamp |
| VM name |

Answer:

Answer Area

| Setting | Value |
|---------|-------|
|---------|-------|

Number of partitions

| |
|----|
| 3 |
| 4 |
| 6 |
| 12 |

Partition Key

| |
|------------|
| Highway |
| Department |
| Timestamp |
| VM name |

Explanation:

Box 1: 6

The number of partitions is specified at creation and must be between 2 and 32.

There are 6 highways.

Box 2: Highway

References:

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features>

QUESTION 157

You are preparing to deploy an ASP.NET Core website to an Azure Web App from a GitHub repository. The website includes static content generated by a script.

You plan to use the Azure Web App continuous deployment feature.

You need to run the static generation script before the website starts serving traffic.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Create a file named .deployment in the root of the repository that calls a script which generates the static content and deploys the website.
- B. Add a PreBuild target in the websites csproj project file that runs the static content generation script.
- C. Create a file named run.cmd in the folder /run that calls a script which generates the static content and deploys the website.

- D. Add the path to the static content generation tool to WEBSITE_RUN_FROM_PACKAGE setting in the host.json file.

Answer: AD

Explanation:

A: To customize your deployment, include a .deployment file in the repository root.

You just need to add a file to the root of your repository with the name .deployment and the content:

[config]

command = YOUR COMMAND TO RUN FOR DEPLOYMENT

this command can be just running a script (batch file) that has all that is required for your deployment, like copying files from the repository to the web root directory for example.

D: In Azure, you can run your functions directly from a deployment package file in your function app. The other option is to deploy your files in the d:\home\site\wwwroot directory of your function app (see A above).

To enable your function app to run from a package, you just add a WEBSITE_RUN_FROM_PACKAGE setting to your function app settings.

Note: The host.json metadata file contains global configuration options that affect all functions for a function app.

References:

<https://github.com/projectkudu/kudu/wiki/Custom-Deployment-Script>

<https://docs.microsoft.com/bs-latn-ba/azure/azure-functions/run-functions-from-deployment-package>

QUESTION 158

A company is developing a solution that allows smart refrigerators to send temperature information to a central location. You have an existing Service Bus.

The solution must receive and store messages until they can be processed. You create an Azure Service Bus instance by providing a name, pricing tier, subscription, resource group, and location.

You need to complete the configuration.

Which Azure CLI or PowerShell command should you run?

- A. **New-AzureRmServiceBusNamespace**
 -ResourceGroupName **fridge-rg**
 -NamespaceName **fridge-ns**
 -Location **fridge-loc**
- B. **az servicebus namespace create**
 - -resource-group **fridge-rg**
 - -name **fridge-ns**
 - -location **fridge-loc**
- C. **New-AzureRmResourceGroup**
 -Name **fridge-rg**
 -Location **fridge-loc**

- D. **az servicebus queue create
--resource-group fridge-rg
--namespace-name fridge-ns
--name fridge-q**

Answer: D

Explanation:

A service bus instance has already been created (Step 2 below). Next is step 3, Create a Service Bus queue.

Note:

Steps:

Step 1: # Create a resource group

```
resourceGroupName="myResourceGroup"
```

```
az group create --name $resourceGroupName --location eastus Step 2: # Create a Service Bus messaging namespace with a unique name namespaceName=myNameSpace$RANDOM
```

```
az servicebus namespace create --resource-group $resourceGroupName --name  
$namespaceName -- location eastus
```

Step 3: # Create a Service Bus queue

```
az servicebus queue create --resource-group $resourceGroupName --namespace-name  
$namespaceName --name BasicQueue
```

Step 4: # Get the connection string for the namespace

```
connectionString=$(az servicebus namespace authorization-rule keys list --resource-group  
$resourceGroupName --namespace-name $namespaceName --name  
RootManageSharedAccessKey --query primaryConnectionString --output tsv)
```

References:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-quickstart-cli>

QUESTION 159

A company is developing a solution that allows smart refrigerators to send temperature information to a central location. You have an existing Service Bus.

The solution must receive and store messages until they can be processed. You create an Azure Service Bus instance by providing a name, pricing tier, subscription, resource group, and location.

You need to complete the configuration.

Which Azure CLI or PowerShell command should you run?

- A. **az servicebus namespace create
- -resource-group fridge-rg
- -name fridge-ns
- -location fridge-loc**
- B. **az servicebus queue create
--resource-group fridge-rg
--namespace-name fridge-ns
--name fridge-q**

- C. `connectionString=$(az servicebus namespace authorization-rule key list --resource-group fridge-rg --namespace-name fridge-ns --name RootManageSharedAccessKey --query primaryConnectionString --output tsv)`
- D. `az group create --name fridge-rg --location fridge-log`

Answer: B

Explanation:

A service bus instance has already been created (Step 2 below). Next is step 3, Create a Service Bus queue.

Note:

Steps:

Step 1: # Create a resource group

`resourceGroupName="myResourceGroup"`

`az group create --name $ResourceGroupName --location eastus`

Step 2: # Create a Service Bus messaging namespace with a unique name

`namespaceName=myNameSpace$RANDOM`

`az servicebus namespace create --resource-group $resourceGroupName --name`

`$namespaceName -- location eastus`

Step 3: # Create a Service Bus queue

`az servicebus queue create --resource-group $resourceGroupName --namespace-name`

`$namespaceName --name BasicQueue`

Step 4: # Get the connection string for the namespace

`connectionString=$(az servicebus namespace authorization-rule keys list --resource-group`

`$resourceGroupName --namespace-name $namespaceName --name`

`RootManageSharedAccessKey -- query primaryConnectionString --output tsv)`

References:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-quickstart-cli>

QUESTION 160

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Service Bus. Configure a topic to receive the device data by using a

correlation filter.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead provision an Azure Event Hub. Configure the machine identifier as the partition key and enable capture.

References:

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-programming-guide>

QUESTION 161

Case Study 3 - Proseware, Inc

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service.

If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody]) string eventsJson
EG05      {
EG06          var events = JArray.Parse(eventsJson);
EG07
EG08          foreach (var @event in events)
EG09          {
EG10              EventId.Value = @event["id"].ToString();
EG11              if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12              {
EG13                  SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14              }
EG15
EG16              {
EG17                  EnsureLogging(@event["subject"].ToString());
EG18              }
EG19          }
EG20          return null;
EG21      }
EG22      private void EnsureLogging(string resource)
EG23      {
EG24          . .
EG25      }
EG26      private async Task SendToAnomalyDetectionService(string uri)
EG27      {
EG28          var content = GetLogData(uri);
EG29          var scoreRequest = new
EG30          {
EG31              Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32          {
EG33              {
EG34                  "input1",
```

```
EG35         new List<Dictionary<string, string>>()
EG36     {
EG37         new Dictionary<string, string>()
EG38     {
EG39         {
EG40             "logcontent", content
EG41         }
EG42     }
EG43 }
EG44 },
EG45 },
EG46     GlobalParameters = new Dictionary<string, string>() { }
EG47 };
EG48 var result = await (new HttpClient()).PostAsJsonAsync("...", scoreRequest);
EG49 var rawModelResult = await result.Content.ReadAsStringAsync();
EG50 var modelResult = JObject.Parse(rawModelResult);
EG51 if (modelResult["notify"].HasValues)
EG52 {
EG53     ...
EG54 }
EG55 }
EG56 private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57 {
EG58     ...
EG59 }
EG60     private string GetLogData(string uri)
EG61 {
EG62     ...
EG63 }
EG64     static string BlobStoreAccountSAS(string containerName)
EG65 {
EG66     ...
EG67 }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get; set; }
LE05     public DateTime eventTime { get; set; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

You need to ensure that authentication events are triggered and processed according to the authentication events policy.

What should you do?

- A. Ensure that signout events have a subject prefix. Create an Azure Event Grid subscription that uses the subjectBeginsWith filter.
- B. Create a new Azure Event Grid topic and add a subscription for the events.
- C. Create a new Azure Event Grid subscription for all authentication that delivers messages to an Azure Event Hub. Use the subscription to process signout events.
- D. Create separate Azure Event Grid topics and subscriptions for sign-in and sign-out events.

Answer: A

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 162

Drag and Drop Question

You are developing a software solution for an autonomous transportation system. The solution uses large data sets and Azure Batch processing to simulate navigation sets for entire fleets of vehicles.

You need to create compute nodes for the solution on Azure Batch.

What should you do?

| Select these | Place here |
|---|------------|
| In the Azure portal, add a Job | |
| In the Azure portal, create a Batch account. | |
| In the Azure portal, create tasks | |
| In the Azure portal, create a pool of compute nodes | |

Answer:

| Select these | Place here |
|--------------|---|
| | In the Azure portal, create a Batch account. |
| | In the Azure portal, create a pool of compute nodes |
| | In the Azure portal, add a Job |
| | In the Azure portal, create tasks |

Explanation:

With the Azure Portal:

Step 1: In the Azure portal, create a Batch account.

First we create a batch account.

Step 2: In the Azure portal, create a pool of compute nodes Now that you have a Batch account, create a sample pool of Windows compute nodes for test purposes.

Step 3: In the Azure portal, add a Job.

Now that you have a pool, create a job to run on it. A Batch job is a logical group for one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. Initially the job has no tasks.

Step 4: In the Azure portal, create tasks

Now create sample tasks to run in the job. Typically you create multiple tasks that Batch queues and distributes to run on the compute nodes.

References:

<https://docs.microsoft.com/en-us/azure/batch/quick-create-portal>

QUESTION 163

Drag and Drop Question

You are developing a software solution for an autonomous transportation system. The solution uses large data sets and Azure Batch processing to simulate navigation sets for entire fleets of vehicles.

You need to create compute nodes for the solution on Azure Batch.

What should you do?

| Select these | Place here |
|--|------------|
| In a .NET method, call the method: BatchClient.PoolOperations.Create Pool | |
| In a .NET method, call the method: BatchClient.PoolOperations.CreateJob | |
| In the Azure portal, create a Batch account | |
| In a .NET method, call the method: BatchClient.JobOperations.AddTask | |

Answer:

| Select these | Place here |
|--------------|--|
| | In the Azure portal, create a Batch account |
| | In a .NET method, call the method: BatchClient.PoolOperations.Create Pool |
| | In a .NET method, call the method: BatchClient.PoolOperations.CreateJob |
| | In a .NET method, call the method: BatchClient.JobOperations.AddTask |

Explanation:

With .NET:

Step 1: In the Azure portal, create a Batch account.

First we create a batch account.

Step 2: In a .NET method, call the method: BatchClient.PoolOperations.CreatePool Now that you have a Batch account, create a sample pool of Windows compute nodes for test purposes. To create a Batch pool, the app uses the BatchClient.PoolOperations.CreatePool method to set the number of nodes, VM size, and a pool configuration.

Step 3: In a .NET method, call the method: BatchClient.PoolOperations.CreateJob Now that you have a pool, create a job to run on it. A Batch job is a logical group for one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. Initially the job has no tasks. The app uses the BatchClient.JobOperations.CreateJob method to create a job on your pool.

Step 4: In a .NET method, call the method: batchClient.JobOperations.AddTask Now create sample tasks to run in the job. Typically you create multiple tasks that Batch queues and distributes to run on the compute nodes. The app adds tasks to the job with the AddTask method, which queues them to run on the compute nodes.

For example: batchClient.JobOperations.AddTask(JobId, tasks);

References:

<https://docs.microsoft.com/en-us/azure/batch/quick-create-portal>

<https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet>

QUESTION 164

Drag and Drop Question

You are developing a software solution for an autonomous transportation system. The solution uses large data sets and Azure Batch processing to simulate navigation sets for entire fleets of vehicles.

You need to create compute nodes for the solution on Azure Batch.

What should you do?

| Select these | Place here |
|--|------------|
| In Python, implement the class: TaskAddParameter | |
| In Python, implement the class: JobAddParameter | |
| In the Azure portal, create a Batch account. | |
| In Python, implement the class: PoolAddParameter | |

Answer:

| Select these | Place here |
|--------------|--|
| | In the Azure portal, create a Batch account. |
| | In Python, implement the class: PoolAddParameter |
| | In Python, implement the class: JobAddParameter |
| | In Python, implement the class: TaskAddParameter |

Explanation:

With Python:

Step 1: In the Azure portal, create a Batch account.

First we create a batch account.

Step 2: In Python, implement the class: PoolAddParameter Now that you have a Batch account, create a sample pool of Windows compute nodes for test purposes. To create a Batch pool, the app uses the PoolAddParameter class to set the number of nodes, VM size, and a pool configuration.

Step 3: In Python, implement the class: JobAddParameter

Now that you have a pool, create a job to run on it. A Batch job is a logical group for one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. Initially the job has no tasks. The app uses the JobAddParameter class to create a job on your pool.

Step 4: In Python, implement the class: TaskAddParameter Now create sample tasks to run in the job. Typically you create multiple tasks that Batch queues and distributes to run on the compute nodes. The app creates a list of task objects using the TaskAddParameter class.

References:

<https://docs.microsoft.com/en-us/azure/batch/quick-create-portal>

<https://docs.microsoft.com/en-us/azure/batch/quick-run-python>

QUESTION 165**Case Study 5 - Wide World Importers****Background**

Wide World Importers is moving all their datacenters to Azure. The company has developed several applications and services to support supply chain operations and would like to leverage serverless computing where possible.

Current environment**Windows Server 2016 virtual machine**

This virtual machine (VM) runs Biz Talk Server 2016. The VM runs the following workflows:

- Ocean Transport – This workflow gathers and validates container information including container contents and arrival notices at various shipping ports.
- Inland Transport – This workflow gathers and validates trucking information including fuel usage, number of stops, and routes.

The VM supports the following REST API calls:

- Container API – This API provides container information including weight, contents, and other attributes.
- Location API – This API provides location information regarding shipping ports of call and truck stops.
- Shipping REST API – This API provides shipping information for use and display on the shipping website.

Shipping Data

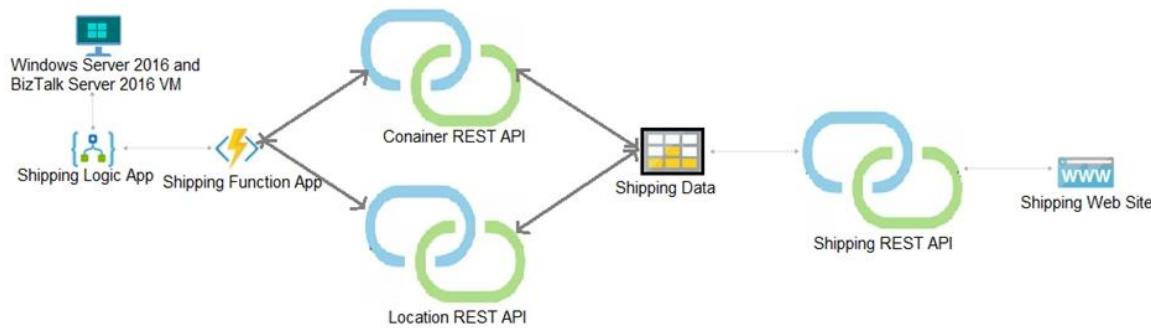
The application uses MongoDB JSON document storage database for all container and transport information.

Shipping Web Site

The site displays shipping container tracking information and container contents. The site is located at <http://shipping.wideworldimporters.com>

Proposed solution

The on-premises shipping application must be moved to Azure. The VM has been migrated to a new Standard_D16s_v3 Azure VM by using Azure Site Recovery and must remain running in Azure to complete the BizTalk component migrations. You create a Standard_D16s_v3 Azure VM to host BizTalk Server. The Azure architecture diagram for the proposed solution is shown below:

**Shipping Logic App**

The Shipping Logic app must meet the following requirements:

- Support the ocean transport and inland transport workflows by using a Logic App.
- Support industry-standard protocol X12 message format for various messages including vessel content details and arrival notices.
- Secure resources to the corporate VNet and use dedicated storage resources with a fixed costing model.
- Maintain on-premises connectivity to support legacy applications and final BizTalk migrations.

Shipping Function app

Implement secure function endpoints by using app-level security and include Azure Active Directory (Azure AD).

REST APIs

The REST API's that support the solution must meet the following requirements:

- Secure resources to the corporate VNet.
- Allow deployment to a testing location within Azure while not incurring additional costs.
- Automatically scale to double capacity during peak shipping times while not causing application downtime.
- Minimize costs when selecting an Azure payment model.

Shipping data

Data migration from on-premises to Azure must minimize costs and downtime.

Shipping website

Use Azure Content Delivery Network (CDN) and ensure maximum performance for dynamic content while minimizing latency and costs.

Issues

Windows Server 2016 VM

The VM shows high network latency, jitter, and high CPU utilization. The VM is critical and has not been backed up in the past. The VM must enable a quick restore from a 7-day snapshot to include in-place restore of disks in case of failure.

Shipping website and REST APIs

The following error message displays while you are testing the website:

```
Failed to load http://test-shippingapi.wideworldimporters.com/: No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'http://testwideworldimporters.com/' is therefore not allowed access.
```

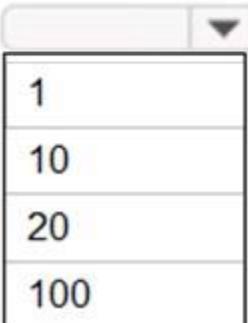
Hotspot Question

You need to configure Azure App Service to support the REST API requirements.

Which values should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Setting | Value |
|----------------|--|
| Plan |  <ul style="list-style-type: none">BasicStandardPremiumIsolated |
| Instance Count |  <ul style="list-style-type: none">11020100 |

Answer:

Answer Area

| Setting | Value |
|----------------|---|
| Plan | <div style="border: 1px solid black; padding: 5px; width: fit-content;">Basic Standard Premium Isolated</div> |
| Instance Count | <div style="border: 1px solid black; padding: 5px; width: fit-content;">1 10 20 100</div> |

Explanation:

Plan: Standard

Standard support auto-scaling

Instance Count: 10

Max instances for standard is 10.

Scenario:

The REST API's that support the solution must meet the following requirements:

Allow deployment to a testing location within Azure while not incurring additional costs.

Automatically scale to double capacity during peak shipping times while not causing application downtime.

Minimize costs when selecting an Azure payment model.

References:

<https://azure.microsoft.com/en-us/pricing/details/app-service/plans/>

QUESTION 166

In the Azure Search service a team 'XPN' has created an Index. Ryan has upload data into the Index. He proposed the following steps to carryout from his .Net program:

- Create a `SearchServiceClient` object to connect to the search index.
- Create a `DataContainer` that contains the documents which must be added.
- Create a `DataSource` instance and set its `Container` property to the `DataContainer`.
- Set the `DataSource` property of the `SearchServiceClientState`

Whether the requirement is being fulfilled by the listed steps?

- A. TRUE
- B. FALSE

Answer: B

QUESTION 167

A development team is preparing code that will work with the Azure Batch service. The code will be used to process videos and store them in a container in an Azure storage account.
You need to ensure that the code will be able to store the processed videos from the Batch jobs to the storage container.
You need to ensure code is in place to prepare the storage for the output videos.
You propose code to generate a CORS signature for the storage container.
Would this solution fulfil the requirements?

- A. Correct
- B. Incorrect

Answer: B

QUESTION 168

A consultant needs to deploy Web Applications to the Azure Web App service for 4 customers. Each customer needs to have the application running on a separate individual instance.
The following key requirements are also in place:

- Ability to automatically scale on demand
- Ability to use deployment slots to test staging environments
- All Azure resources should be located in a separate isolated network
- Costs need to be minimized

How many instances would you keep running for the requirement?

- A. 16
- B. 8
- C. 12
- D. 4

Answer: D

QUESTION 169

A development team 'NPR' is developing an application.
The application will be storing its data in Azure Table storage.
Below are the fields that are going to be stored in the table:

- Region
- Email address
- Phone number

The following snippet of code needs to be completed that would be used to insert a batch of

records. _____ will go into Slot1.

- A. TableOperation
- B. TableBatchOperation
- C. TableEntity
- D. TableQuery

Answer: B

QUESTION 170

A development team 'XRP' is developing an application.

The application will be storing its data in Azure Table storage.

Below are the fields that are going to be stored in the table:

- Region
- Email address
- Phone number

To insert a batch of records _____ snippet of code needs to be completed that would be used .

- A. ExecuteBatch
- B. Execute
- C. Insert
- D. InsertOrMerge

Answer: A

QUESTION 171

A team 'PNX' has created an Index in the Azure Search service.

Ted has to upload data into the Index.

To carry out from his .Net program he Proposed the following steps:

- Create a SearchIndexClient object to connect to the search index.
- Create a DataContainer that contains the documents which must be added.
- Create a DataSource instance and set its Container property to the DataContainer
- Call the Documents.Search method of the SearchIndexClient and pass the DataSource.State

Whether the list of steps have fulfilled the requirement?

- A. TRUE
- B. FALSE

Answer: B

QUESTION 172

A development team 'PNX' is developing an application.

The application will be storing its data in Azure Table storage.

Below are the fields that are going to be stored in the table:

- Region
- Email address
- Phone number

The following snippet of code needs to be completed that would be used to insert a batch of records.

Out of the given Options select which will go into Slot2?

- A. TableOperation
- B. TableBatchOperation
- C. TableEntity
- D. TableQuery

Answer: B

QUESTION 173

A team 'XNP' is developing container-based applications that need to be deployed to a Kubernetes cluster in Azure.

Parker has to create thecluster and ensure the services are running as desired.

Considering the following, select the Command would he execute?

- 1.) az aks create
- 2.) az group create
- 3.) kubectl apply
- 4.) az appservice plan create

- A. Only 1, 2 and 4
- B. Only 2 and 3
- C. Only 3 and 4
- D. Only 1, 2 and 3

Answer: D

QUESTION 174

For an application Nea as developer needs to Create a Dockerfile.

The application will be based on ASP.Net core. The application has the following requirements:

- Ensure that the application whizlabsApp.dll runs at the startup of the docker container
- Run a powershell script called whizlabscsript.ps1 in the Docker container

The tptApp.dll and the tptcsript.ps1 are in the same location as the DockerFile.
Which of the following commands wouldNea place in the DockerFile?

- 1.) FROM Microsoft/dotnet:2.2-aspnetcore-runtime
- 2.) EXPOSE tptApp.dll ,tptcsript.ps1
- 3.) ENTRYPOINT ["dotnet", " MySingleContainerWebApp.dll
- 4.) ENTRYPOINT ["tptApp.dll" , "tptcsript.ps1"]

- A. Only 2 and 3

- B. Only 1, 3 and 4
- C. Only 1 and 2
- D. Only 3 and 4

Answer: D

QUESTION 175

A team 'X' has created an Index in the Azure Search service.

You have to upload data into the Index.

Riley Proposed the following steps to carryout from his .Net program:

- Create a SearchIndexClient object to connect to the search index
- Create an IndexBatch that contains the documents which must be added.
- Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.

Is the Requirement being fulfilled by the listed steps?

- A. TRUE
- B. FALSE

Answer: A

QUESTION 176

You have the following code (line numbers are included for reference only):

```
BatchTokenCredentials = GetCredentials();
BatchClient client^ BatchClient.Open(credentials);
CloudJob job = client.JobOperations.CreateJob();
CloudTask task = new CloudTask("sampleTask", "cmd /c SampleTask.exe");
job.AddTask(task);
job.Commit();
```

You need to ensure that the SampleTask.exe process runs.

Which two actions should you perform? Each answer presents part of the complete solution.

- A. Add the following code between lines 02 and 03: var pool = client.PoolOperations.CreatePool();
- B. Add the following code between lines 03 and 04: job.PoolInformation.PoolId = pool.Id;
- C. Modify line 05 as follows:
client.JobOperations.AddTask("SampleTask", task);
- D. Remove line 06.

Answer: AB

Explanation:

You should add the following code between lines 02 and 03: var pool = client.PoolOperations.CreatePool();

This creates a batch pool. A pool is a set of virtual machines (VMs) for running batch jobs.

You should also add the following code between lines 03 and 04: job.PoolInformation.PoolId = pool.Id;

This allows you to assign the job to the pool you created. A job cannot run outside of a pool.

You should not remove line 06. The Commit method creates the job in Azure.

You should not modify line 05 as follows: client.JobOperations.AddTask("MSampleTask", task);

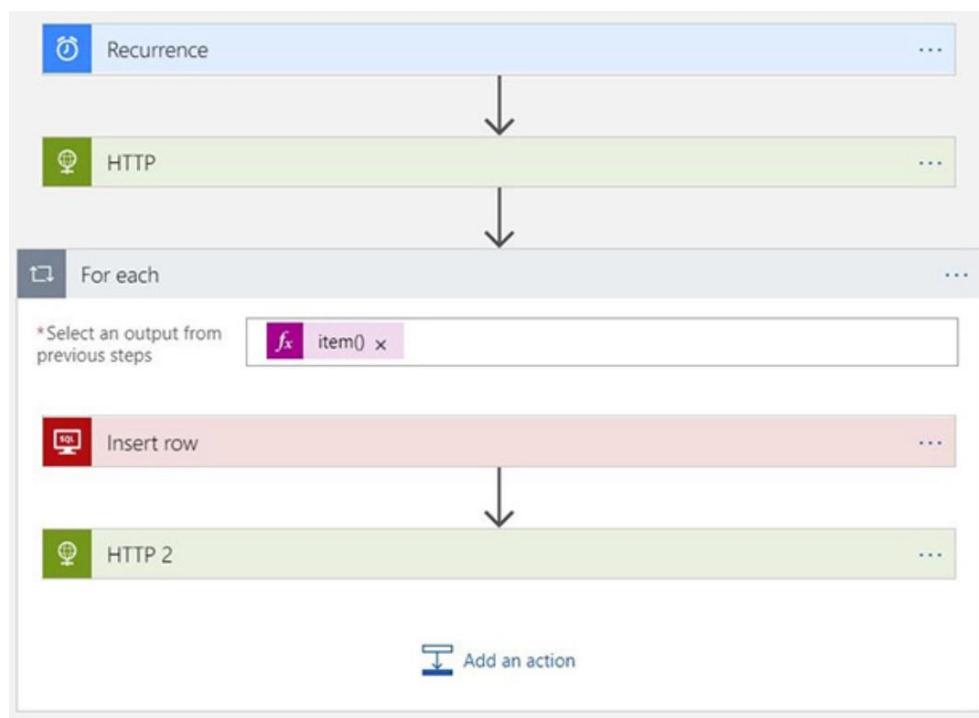
You can either use client.JobOperations.AddTask or job.AddTask. Using the former, you must

specify the ID of the job to which the task applies. In this scenario, you did not create a job with an ID named SampleTask.

QUESTION 177

Hotspot Question

You create the Logic App shown in the exhibit to integrate with a DevOps system. Each step in the Logic App uses default settings.



Every night at 9:00 P.M., the Logic App issues an HTTP request to retrieve a list of all new work items. It adds each open work item to an Azure SQL Database and submits them to another HTTP endpoint.

You need to answer questions about the setup.

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| All work items are processed in parallel | <input type="radio"/> | <input type="radio"/> |
| A work item is added to the Azure SQL Database and sent to the HTTP endpoint in parallel. | <input type="radio"/> | <input type="radio"/> |
| If an exception is thrown processing one work item, all processing stops. | <input type="radio"/> | <input type="radio"/> |

Answer:

| Statement | Yes | No |
|---|-------------------------------------|-------------------------------------|
| All work items are processed in parallel | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| A work item is added to the Azure SQL Database and sent to the HTTP endpoint in parallel. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| If an exception is thrown processing one work item, all processing stops. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Explanation:

All work items are processed in parallel. A For each action iterates its items in parallel by default. Work items are added to the Azure SQL Database and sent to the HTTP endpoint in sequence, not in parallel. Each child action of a For each action executes in sequence. If an exception is thrown processing one work item, the For each loop continues processing other work items. By default, a For each action iterates items in parallel, which means that the child actions run on separate threads. This means that failure during one iteration does not affect the remaining iterations.

QUESTION 178

You are developing a .Net Core application that manages images for an advertising company. The images are stored in blob storage with read access geo-zone-redundant storage (RA-GZRS) redundancy to achieve high availability and maximum durability. In case of regional outage, the application should be able to read data from secondary storage replication. Secondary storage replication should only be used if primary storage is unavailable.

You develop the following code to create the object responsible for uploading and reading images from blob storage:

```
string storageConnectionString =  
Environment.GetEnvironmentVariable("storageconnectionstring");  
CloudStorageAccount storageAccount =  
CloudStorageAccount.Parse(storageConnectionString);  
blobClient = storageAccount.CreateCloudBlobClient();
```

You need to configure the blob client to meet the high availability requirement. Which location mode option should you use?

- A. blobClient.DefaultRequestOptions.LocationMode = LocationMode.SecondaryOnly;
- B. blobClient.DefaultRequestOptions.LocationMode = LocationMode.SecondaryThenPrimary;
- C. blobClient.DefaultRequestOptions.LocationMode = LocationMode.PrimaryThenSecondary;
- D. blobClient.DefaultRequestOptions.LocationMode = LocationMode.PrimaryOnly;

Answer: C**Explanation:**

You should configure the blob client default request option for the location mode as PrimaryThenSecondary. In this mode, in case of a regional outage affecting primary storage, the application reads the images replicated to another region as a fallback until primary storage is available again.

You should not configure the blob client default request option for the location mode as PrimaryOnly. In this mode, only primary storage is used. In case of a regional outage affecting

primary storage, the application will fail to read images even if they are replicated to secondary storage. This is the default blob client mode.

You should not configure the blob client default request option for the location mode as SecondaryOnly or SecondaryThenPrimary. With these modes, the blob client will first read the images from secondary storage. Secondary storage should only be used if primary storage is unavailable.

QUESTION 179

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 develop Azure solutions.

You must grant a virtual machine (VM) access to specific resource groups in Azure Resource Manager.

You need to obtain an Azure Resource Manager access token.

Solution: Run the Invoke-RestMethod cmdlet to make a request to the local managed identity for Azure resources endpoint.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

Get an access token using the VM's system-assigned managed identity and use it to call Azure Resource Manager

You will need to use PowerShell in this portion.

1. In the portal, navigate to Virtual Machines and go to your Windows virtual machine and in the Overview, click Connect.
2. Enter in your Username and Password for which you added when you created the Windows VM.
3. Now that you have created a Remote Desktop Connection with the virtual machine, open PowerShell in the remote session.
4. Using the Invoke-WebRequest cmdlet, make a request to the local managed identity for Azure resources endpoint to get an access token for Azure Resource Manager.

Example:

```
$response = Invoke-WebRequest -Uri  
'http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-  
01&resource=https://management.azure.com/' -Method GET -Headers @{Metadata="true"}
```

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/tutorial-windows-vm-access-arm>

QUESTION 180

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 develop and deploy an Azure App Service API app to a Windows-hosted deployment slot named Development. You create additional deployment slots named Testing and Production. You enable auto swap on the Production deployment slot.

You need to ensure that scripts run and resources are available before a swap operation occurs.

Solution: Update the web.config file to include the applicationInitialization configuration element. Specify custom initialization actions to run the scripts.

Does the solution meet the goal?

- A. No
- B. Yes

Answer: B

Explanation:

Specify custom warm-up.

Some apps might require custom warm-up actions before the swap. The applicationInitialization configuration element in web.config lets you specify custom initialization actions. The swap operation waits for this custom warm-up to finish before swapping with the target slot. Here's a sample web.config fragment.

```
<system.webServer>
  <applicationInitialization>
    <add initializationPage="/" hostName="[app hostname]" />
    <add initializationPage="/Home/About" hostName="[app hostname]" />
  </applicationInitialization>
</system.webServer>
```

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/deploy-staging-slots#troubleshoot-swaps>

QUESTION 181

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 develop and deploy an Azure App Service API app to a Windows-hosted deployment slot named Development. You create additional deployment slots named Testing and Production. You

enable auto swap on the Production deployment slot.

You need to ensure that scripts run and resources are available before a swap operation occurs.

Solution: Enable auto swap for the Testing slot. Deploy the app to the Testing slot.

Does the solution meet the goal?

- A. No
- B. Yes

Answer: B

Explanation:

Instead update the web.config file to include the applicationInitialization configuration element. Specify custom initialization actions to run the scripts.

Note: Some apps might require custom warm-up actions before the swap. The applicationInitialization configuration element in web.config lets you specify custom initialization actions. The swap operation waits for this custom warm-up to finish before swapping with the target slot. Here's a sample web.config fragment.

```
<system.webServer>
    <applicationInitialization>
        <add initializationPage="/" hostName="[app hostname]" />
        <add initializationPage="/Home/About" hostName="[app hostname]" />
    </applicationInitialization>
</system.webServer>
```

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/deploy-staging-slots#troubleshoot-swaps>

QUESTION 182

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 develop and deploy an Azure App Service API app to a Windows-hosted deployment slot named Development. You create additional deployment slots named Testing and Production. You enable auto swap on the Production deployment slot.

You need to ensure that scripts run and resources are available before a swap operation occurs.

Solution: Disable auto swap. Update the app with a method named statuscheck to run the scripts. Re-enable auto swap and deploy the app to the Production slot.

Does the solution meet the goal?

- A. No
- B. Yes

Answer: A**Explanation:**

Instead update the web.config file to include the applicationInitialization configuration element. Specify custom initialization actions to run the scripts.

Note: Some apps might require custom warm-up actions before the swap. The applicationInitialization configuration element in web.config lets you specify custom initialization actions. The swap operation waits for this custom warm-up to finish before swapping with the target slot. Here's a sample web.config fragment.

```
<system.webServer>
    <applicationInitialization>
        <add initializationPage="/" hostName="[app hostname]" />
        <add initializationPage="/Home/About" hostName="[app hostname]" />
    </applicationInitialization>
</system.webServer>
```

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/deploy-staging-slots#troubleshoot-swaps>**QUESTION 183**

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 develop Azure solutions.

You must grant a virtual machine (VM) access to specific resource groups in Azure Resource Manager.

You need to obtain an Azure Resource Manager access token.

Solution: Use an X.509 certificate to authenticate the VM with Azure Resource Manager.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B**Explanation:**

Instead run the Invoke-RestMethod cmdlet to make a request to the local managed identity for Azure resources endpoint.

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/tutorial-windows-vm-access-arm>**QUESTION 184**

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 develop Azure solutions.

You must grant a virtual machine (VM) access to specific resource groups in Azure Resource Manager.

You need to obtain an Azure Resource Manager access token.

Solution: Use the Reader role-based access control (RBAC) role to authenticate the VM with Azure Resource Manager.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead run the Invoke-RestMethod cmdlet to make a request to the local managed identity for Azure resources endpoint.

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/tutorial-windows-vm-access-arm>

QUESTION 185

Drag and Drop Question

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 from 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.

| Actions | Answer Area |
|--|-------------|
| Create a single Service Bus topic. | |
| Create a Service Bus Namespace for each restaurant for which a driver can receive messages. | |
| Create a single Service Bus subscription. | |
| Create a Service Bus subscription for each restaurant for which a driver can receive orders. | |
| Create a single Service Bus Namespace. | |
| Create a Service Bus topic for each restaurant for which a driver can receive messages. | |

Answer:

| Actions | Answer Area |
|---|--|
| Create a single Service Bus topic. | Create a single Service Bus Namespace. |
| Create a Service Bus Namespace for each restaurant for which a driver can receive messages. | Create a Service Bus topic for each restaurant for which a driver can receive messages. |
| Create a single Service Bus subscription. | Create a Service Bus subscription for each restaurant for which a driver can receive orders. |

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.

Reference:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview>

QUESTION 186

Hotspot Question

You develop a news and blog content 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.

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
hub =
NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

GetInstallation
CreateClientFromConnectionString
CreateOrUpdateInstallation
PatchInstallation

(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"<toast><visual><binding template=""ToastText01""><text id=""1"">" +
@"New item to view" + @"</text></binding></visual></toast>";
try
{
var result =
await hub.
SendWindowsNativeNotificationAsync(windowsToastPayload);

SendWindowsNativeNotificationAsync
SubmitNotificationHubJobAsync
ScheduleNotificationAsync
SendAppleNativeNotificationAsync

...
}
catch (System.Exception ex)
{
...
}
...
```

Answer:

Answer Area

```
string notificationHubName = "contoso_hub";
string notificationHubConnection = "connection_string";
hub =
NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

.
.
.

NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails

GetInstallation
CreateClientFromConnectionString
CreateOrUpdateInstallation
PatchInstallation

(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"<toast><visual><binding template=""ToastText01""><text id=""1"">" +
@"New item to view" + @"/<text></binding></visual></toast>";
try
{
var result =
await hub.
SendWindowsNativeNotificationAsync
SubmitNotificationHubJobAsync
ScheduleNotificationAsync
SendAppleNativeNotificationAsync
...
}
catch (System.Exception ex)
{
...
}
...
```

Explanation:

Box 1: NotificationHubClient

Box 2: NotificationHubClient

Box 3: CreateClientFromConnectionString

```
// Initialize the Notification Hub
NotificationHubClient hub =
NotificationHubClient.CreateClientFromConnectionString(listenConnString, hubName);
```

Box 4: SendWindowsNativeNotificationAsync

Send the push notification.

```
var result = await hub.SendWindowsNativeNotificationAsync(windowsToastPayload);
```

Reference:

<https://docs.microsoft.com/en-us/azure/notification-hubs/notification-hubs-push-notification-registration-management>
<https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/app-service-mobile/app-service-mobile-windows-store-dotnet-get-started-push.md>

QUESTION 187

Hotspot Question

You are developing an application that uses Azure Storage Queues.

You have the following code:

```
CloudStorageAccount storageAccount = CloudStorageAccount.Parse  
(CloudConfigurationManager.GetSetting("StorageConnectionString"));  
CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();  
  
CloudQueue queue = queueClient.GetQueueReference("appqueue");  
await queue.CreateIfNotExistsAsync();  
  
CloudQueueMessage peekedMessage = await queue.PeekMessageAsync();  
if (peekedMessage != null)  
{  
    Console.WriteLine("The peeked message is: {0}", peekedMessageAsString);  
}  
CloudQueueMessage message = await queue.GetMessageAsync();
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Answer Area

| Statement | Yes | No |
|---|-----------------------|-----------------------|
| The code configures the lock duration for the queue. | <input type="radio"/> | <input type="radio"/> |
| The last message is no longer visible in the queue after the code runs. | <input type="radio"/> | <input type="radio"/> |
| The storage queue remains in the storage account after the code runs. | <input type="radio"/> | <input type="radio"/> |

Answer:

Answer Area

| Statement | Yes | No |
|---|----------------------------------|----------------------------------|
| The code configures the lock duration for the queue. | <input type="radio"/> | <input checked="" type="radio"/> |
| The last message is no longer visible in the queue after the code runs. | <input checked="" type="radio"/> | <input type="radio"/> |
| The storage queue remains in the storage account after the code runs. | <input checked="" type="radio"/> | <input type="radio"/> |

Explanation:

Box 1: No

The code does not configure the lock duration.

Note: Default lock duration is 30 seconds. Lock duration will be set even if it is not specified in code.

Box 2: Yes

Peeking messages will not remove them from the queue.

Box 3: Yes

Reference:

<https://docs.microsoft.com/en-us/azure/storage/queues/storage-dotnet-how-to-use-queues>

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.servicebus.messaging.queuedescription.lockduration>

QUESTION 188

Drag and Drop Question

You develop an ASP.NET Core MVC application. You configure the application to track webpages and custom events.

You need to identify trends in application usage.

Which Azure Application Insights Usage Analysis features should you use? To answer, drag the appropriate features to the correct requirements. Each feature 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.

| Features | Answer Area | Requirement | Feature |
|------------|-------------|--|---------|
| Users | | Which pages visited by users most often correlate to a product purchase? | Feature |
| Funnels | | How does load time of the product display page affect a user's decision to purchase a product? | Feature |
| Impact | | Which events most influence a user's decision to continue to use the application? | Feature |
| Retention | | Are there places in the application that users often perform repetitive actions? | Feature |
| User Flows | | | |

Answer:

| Features | Answer Area | Requirement | Feature |
|----------|-------------|--|------------|
| Funnels | | Which pages visited by users most often correlate to a product purchase? | Users |
| | | How does load time of the product display page affect a user's decision to purchase a product? | Impact |
| | | Which events most influence a user's decision to continue to use the application? | Retention |
| | | Are there places in the application that users often perform repetitive actions? | User Flows |

Explanation:

Box 1: Users

Box 2: Impact

One way to think of Impact is as the ultimate tool for settling arguments with someone on your team about how slowness in some aspect of your site is affecting whether users stick around. While users may tolerate a certain amount of slowness, Impact gives you insight into how best to balance optimization and performance to maximize user conversion.

Box 3: Retention

The retention feature in Azure Application Insights helps you analyze how many users return to your app, and how often they perform particular tasks or achieve goals. For example, if you run a game site, you could compare the numbers of users who return to the site after losing a game with the number who return after winning. This knowledge can help you improve both your user experience and your business strategy.

Box 4: User flows

The User Flows tool visualizes how users navigate between the pages and features of your site. It's great for answering questions like:

How do users navigate away from a page on your site?

What do users click on a page on your site?

Where are the places that users churn most from your site?

Are there places where users repeat the same action over and over?

Incorrect Answers:

Funnel: If your application involves multiple stages, you need to know if most customers are progressing through the entire process, or if they are ending the process at some point. The progression through a series of steps in a web application is known as a funnel. You can use Azure Application Insights Funnels to gain insights into your users, and monitor step-by-step conversion rates.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-monitor/app/usage-impact>

QUESTION 189**Case Study 3 - Proseware, Inc****Background**

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements**Policy service**

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies**Log Policy**

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```

EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody] string eventsJson
EG05      {
EG06          var events = JArray.Parse(eventsJson);
EG07
EG08          foreach (var @event in events)
EG09          {
EG10              EventId.Value = @event["id"].ToString();
EG11              if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12              {
EG13                  SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14              }
EG15
EG16              {
EG17                  EnsureLogging(@event["subject"].ToString());
EG18              }
EG19          }
EG20          return null;
EG21      }
EG22      private void EnsureLogging(string resource)
EG23      {
EG24          . .
EG25      }
EG26      private async Task SendToAnomalyDetectionService(string uri)
EG27      {
EG28          var content = GetLogData(uri);
EG29          var scoreRequest = new
EG30          {
EG31              Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32              {
EG33                  {
EG34                      "input1",
EG35                      new List<Dictionary<string, string>>()
EG36                      {
EG37                          new Dictionary<string, string>()
EG38                          {
EG39                              {
EG40                                  "logcontent", content
EG41                              }
EG42                          }
EG43                      }
EG44                  },
EG45              },
EG46              GlobalParameters = new Dictionary<string, string>() { }
EG47          };
EG48          var result = await (new HttpClient()).PostAsJsonAsync("...", scoreRequest);
EG49          var rawModelResult = await result.Content.ReadAsStringAsync();
EG50          var modelResult = JObject.Parse(rawModelResult);
EG51          if (modelResult["notify"].HasValues)
EG52          {
EG53              . .
EG54          }
EG55      }
EG56      private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57      {
EG58          . .
EG59      }
EG60      private string GetLogData(string uri)
EG61      {
EG62          . .
EG63      }
EG64      static string BlobStoreAccountSAS(string containerName)
EG65      {
EG66          . .
EG67      }
EG68  }

```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```
LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get; set; }
LE05     public DateTime eventTime { get; set; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

Hotspot Question

You need to insert code at line LE03 of LoginEvent.cs to ensure that all authentication events are processed correctly.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

public string (get; set;)

id
eventType
dataVersion
metadataVersion

public string (get; set;)

id
eventType
dataVersion
metadataVersion

public string (get; set;)

id
eventType
dataVersion
metadataVersion

Answer:

Answer Area

public string (get; set;)

id
eventType
dataVersion
metadataVersion

public string (get; set;)

id
eventType
dataVersion
metadataVersion

public string (get; set;)

id
eventType
dataVersion
metadataVersion

Explanation:

Box 1: id

id is a unique identifier for the event.

Box 2: eventType

eventType is one of the registered event types for this event source.

Box 3: dataVersion

dataVersion is the schema version of the data object. The publisher defines the schema version.

Scenario: Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

The following example shows the properties that are used by all event publishers:

```
[  
{  
  "topic": string,  
  "subject": string,  
  "id": string,  
  "eventType": string,  
  "eventTime": string,  
  "data":{  
    object-unique-to-each-publisher  
  }  
}
```

```
},
  "dataVersion": string,
  "metadataVersion": string
}
```

]

Reference:

<https://docs.microsoft.com/en-us/azure/event-grid/event-schema>

QUESTION 190

Case Study 3 - Proseware, Inc

Background

You are a developer for Proseware, Inc. You are developing an application that applies a set of governance policies for Proseware's internal services, external services, and applications. The application will also provide a shared library for common functionality.

Requirements

Policy service

You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.

The application must include the Event Grid Event ID field in all Application Insights telemetry.

Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Policies

Log Policy

All Azure App Service Web Apps must write logs to Azure Blob storage. All log files should be saved to a container named **logdrop**. Logs must remain in the container for 15 days.

Authentication events

Authentication events are used to monitor users signing in and signing out. All authentication events must be processed by Policy service. Sign outs must be processed as quickly as possible.

PolicyLib

You have a shared library named **PolicyLib** that contains functionality common to all ASP.NET Core web services and applications. The **PolicyLib** library must:

- Exclude non-user actions from Application Insights telemetry.
- Provide methods that allow a web service to scale itself
- Ensure that scaling actions do not disrupt application usage

Other

Anomaly detection service

You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Health monitoring

All web applications and services have health monitoring at the /health service endpoint.

Policy loss

When you deploy Policy service, policies may not be applied if they were in the process of being applied during the deployment.

Performance issue

When under heavy load, the anomaly detection service undergoes slowdowns and rejects connections.

Notification latency

Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

```
EventGridController.cs
EG01  public class EventGridController : Controller
EG02  {
EG03      public static AsyncLocal<string> EventId = new AsyncLocal<string>();
EG04      public IActionResult Process([FromBody] string eventsJson)
EG05      {
EG06          var events = JArray.Parse(eventsJson);
EG07
EG08          foreach (var @event in events)
EG09          {
EG10              EventId.Value = @event["id"].ToString();
EG11              if (@event["topic"].ToString().Contains("providers/Microsoft.Storage"))
EG12              {
EG13                  SendToAnomalyDetectionService(@event["data"]["url"].ToString());
EG14              }
EG15
EG16              {
EG17                  EnsureLogging(@event["subject"].ToString());
EG18              }
EG19          }
EG20          return null;
EG21      }
EG22      private void EnsureLogging(string resource)
EG23      {
EG24          .
EG25      }
EG26      private async Task SendToAnomalyDetectionService(string uri)
EG27      {
EG28          var content = GetLogData(uri);
EG29          var scoreRequest = new
EG30          {
EG31              Inputs = new Dictionary<string, List<Dictionary<string, string>>>()
EG32          {
EG33              {
EG34                  "input1",
```

```

EG35         new List<Dictionary<string, string>>()
EG36     {
EG37         new Dictionary<string, string>()
EG38     {
EG39         {
EG40             "logcontent", content
EG41         }
EG42     }
EG43     }
EG44 },
EG45 },
EG46     GlobalParameters = new Dictionary<string, string>() { }
EG47 };
EG48     var result = await (new HttpClient()).PostAsJsonAsync("...", scoreRequest);
EG49     var rawModelResult = await result.Content.ReadAsStringAsync();
EG50     var modelResult = JObject.Parse(rawModelResult);
EG51     if (modelResult["notify"].HasValues)
EG52     {
EG53         ...
EG54     }
EG55 }
EG56     private (string name, string resourceGroup) ParseResourceId(string
resourceId)
EG57     {
EG58     ...
EG59     }
EG60     private string GetLogData(string uri)
EG61     {
EG62     ...
EG63     }
EG64     static string BlobStoreAccountSAS(string containerName)
EG65     {
EG66     ...
EG67     }
EG68 }
```

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

LoginEvent.cs

```

LE01 public class LoginEvent
LE02 {
LE03
LE04     public string subject { get; set; }
LE05     public DateTime eventTime { get; set; }
LE06     public Dictionary<string, string> data { get; set; }
LE07     public string Serialize()
LE08     {
LE09         return JsonConvert.SerializeObject(this);
LE10     }
LE11 }
```

Hotspot Question

You need to implement the Log policy.

How should you complete the EnsureLogging method in EventGridController.cs? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
var client = new WebSiteManagementClient(. . .);
var id = ParseResourceID(resource);
var appSettings = new StringDictionary(name: "properties",
    properties: new Dictionary<string, string> {
        {"DIAGNOSTICS_AZUREBLOBCONTAINERSASURL", BlobStoreAccountSAS("  ")},
        {"DIAGNOSTICS_AZUREBLOBRETENTIONINDAYS", "  "}
    });
     ( 

);
client.WebApps. 
    id.resourceGroup,
    id.name, appSettings);
```

Answer:

Answer Area

```
var client = new WebSiteManagementClient(. . .);
var id = ParseResourceID(resource);
var appSettings = new StringDictionary(name: "properties",
    properties: new Dictionary<string, string> {
        {"DIAGNOSTICS_AZUREBLOBCONTAINERSASURL", BlobStoreAccountSAS("  ")},
        {"DIAGNOSTICS_AZUREBLOBRETENTIONINDAYS", "  "}
    });
     ( 

);
client.WebApps. 
    id.resourceGroup,
    id.name, appSettings);
```

Explanation:

Box 1: logdrop

All log files should be saved to a container named logdrop.

Box 2: 15

Logs must remain in the container for 15 days.

Box 3: UpdateApplicationSettings

All Azure App Service Web Apps must write logs to Azure Blob storage.

Reference:

<https://blog.hompus.nl/2017/05/29/adding-application-logging-blob-to-a-azure-web-app-service-using-powershell/>

QUESTION 191

You are developing an Azure Function App that processes images that are uploaded to an Azure Blob container.

Images must be processed as quickly as possible after they are uploaded, and the solution must minimize latency. You create code to process images when the Function App is triggered.

You need to configure the Function App.

What should you do?

- A. Use an App Service plan. Configure the Function App to use an Azure Blob Storage input trigger.
- B. Use a Consumption plan. Configure the Function App to use an Azure Blob Storage trigger.
- C. Use a Consumption plan. Configure the Function App to use a Timer trigger.
- D. Use an App Service plan. Configure the Function App to use an Azure Blob Storage trigger.
- E. Use a Consumption plan. Configure the Function App to use an Azure Blob Storage input trigger.

Answer: B

Explanation:

The Blob storage trigger starts a function when a new or updated blob is detected. The blob contents are provided as input to the function.

The Consumption plan limits a function app on one virtual machine (VM) to 1.5 GB of memory.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-blob-trigger>

QUESTION 192

You are preparing to deploy a website to an Azure Web App from a GitHub repository. The website includes static content generated by a script.

You plan to use the Azure Web App continuous deployment feature.

You need to run the static generation script before the website starts serving traffic.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Add the path to the static content generation tool to WEBSITE_RUN_FROM_PACKAGE setting in the host.json file.
- B. Add a PreBuild target in the websites csproj project file that runs the static content generation script.
- C. Create a file named run.cmd in the folder /run that calls a script which generates the static content and deploys the website.

- D. Create a file named .deployment in the root of the repository that calls a script which generates the static content and deploys the website.

Answer: AD

Explanation:

A: In Azure, you can run your functions directly from a deployment package file in your function app. The other option is to deploy your files in the d:\home\site\wwwroot directory of your function app (see A above).

To enable your function app to run from a package, you just add a WEBSITE_RUN_FROM_PACKAGE setting to your function app settings.

Note: The host.json metadata file contains global configuration options that affect all functions for a function app.

D: To customize your deployment, include a .deployment file in the repository root.

You just need to add a file to the root of your repository with the name .deployment and the content:

[config]

command = YOUR COMMAND TO RUN FOR DEPLOYMENT

this command can be just running a script (batch file) that has all that is required for your deployment, like copying files from the repository to the web root directory for example.

Reference:

<https://github.com/projectkudu/kudu/wiki/Custom-Deployment-Script>

<https://docs.microsoft.com/bs-latn-ba/azure/azure-functions/run-functions-from-deployment-package>

QUESTION 193

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 are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device data in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Event Grid. Configure the machine identifier as the partition key and enable capture.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

<https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

QUESTION 194

You provide an Azure API Management managed web service to clients. The back-end web service implements HTTP Strict Transport Security (HSTS).

Every request to the backend service must include a valid HTTP authorization header.

You need to configure the Azure API Management instance with an authentication policy.

Which two policies can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Basic Authentication
- B. Digest Authentication
- C. Certificate Authentication
- D. OAuth Client Credential Grant

Answer: CD

QUESTION 195

You have an application that includes an Azure Web app and several Azure Function apps. Application secrets including connection strings and certificates are stored in Azure Key Vault.

Secrets must not be stored in the application or application runtime environment. Changes to Azure Active Directory (Azure AD) must be minimized.

You need to design the approach to loading application secrets.

What should you do?

- A. Create a single user-assigned Managed Identity with permission to access Key Vault and configure each App Service to use that Managed Identity.
- B. Create a single Azure AD Service Principal with permission to access Key Vault and use a client secret from within the App Services to access Key Vault.
- C. Create a system assigned Managed Identity in each App Service with permission to access Key Vault.
- D. Create an Azure AD Service Principal with Permissions to access Key Vault for each App Service and use a certificate from within the App Services to access Key Vault.

Answer: C

Explanation:

Use Key Vault references for App Service and Azure Functions.

Key Vault references currently only support system-assigned managed identities. User-assigned identities cannot be used.

Reference:

<https://docs.microsoft.com/en-us/azure/app-service/app-service-key-vault-references>

QUESTION 196

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 are developing a medical records document management website. The website is used to store scanned copies of patient intake forms.

If the stored intake forms are downloaded from storage by a third party, the contents of the forms must not be compromised.

You need to store the intake forms according to the requirements.

Solution:

1. Create an Azure Key Vault key named skey.
2. Encrypt the intake forms using the public key portion of skey.
3. Store the encrypted data in Azure Blob storage.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

QUESTION 197

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 are developing a medical records document management website. The website is used to store scanned copies of patient intake forms.

If the stored intake forms are downloaded from storage by a third party, the contents of the forms must not be compromised.

You need to store the intake forms according to the requirements.

Solution:

1. Create an Azure Cosmos DB database with Storage Service Encryption enabled.
2. Store the intake forms in the Azure Cosmos DB database.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead use an Azure Key vault and public key encryption. Store the encrypted from in Azure Storage Blob storage.

QUESTION 198

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 are developing a medical records document management website. The website is used to store scanned copies of patient intake forms.

If the stored intake forms are downloaded from storage by a third party, the contents of the forms must not be compromised.

You need to store the intake forms according to the requirements.

Solution: Store the intake forms as Azure Key Vault secrets.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Instead use an Azure Key vault and public key encryption. Store the encrypted form in Azure Storage Blob storage.

QUESTION 199

You develop a gateway solution for a public facing news API. The news API back end is implemented as a RESTful service and uses an OpenAPI specification.

You need to ensure that you can access the news API by using an Azure API Management service instance.

Which Azure PowerShell command should you run?

- A. Import-AzureRmApiManagementApi -Context \$ApiMgmtContext -SpecificationFormat "Swagger" -SpecificationPath \$SwaggerPath -Path \$Path
- B. New-AzureRmApiManagementBackend -Context \$ApiMgmtContext -Url \$Url -Protocol http
- C. New-AzureRmApiManagement -ResourceGroupName \$ResourceGroup -Name \$Name -Location \$Location -Organization \$Org -AdminEmail \$AdminEmail
- D. New-AzureRmApiManagementBackendProxy -Url \$ApiUrl

Answer: D

Explanation:

New-AzureRmApiManagementBackendProxy creates a new Backend Proxy Object which can be piped when creating a new Backend entity.

Example: Create a Backend Proxy In-Memory Object

```
PS C:\>$secpassword = ConvertTo-SecureString "PlainTextPassword" -AsPlainText -Force  
PS C:\>$proxycreds = New-Object System.Management.Automation.PSCredential ("foo",  
$secpassword)  
PS C:\>$credential = New-AzureRmApiManagementBackendProxy -Url "http://12.168.1.1:8080" -  
ProxyCredential $proxycreds  
PS C:\>$apimcontext = New-AzureRmApiManagementContext -ResourceGroupName "Api-  
Default-WestUS" -ServiceName "contoso"  
PS C:\>$backend = New-AzureRmApiManagementBackend -Context $apimcontext -BackendId  
123 -Url 'https://contoso.com/awesomeapi' -Protocol http -Title "first backend" -  
SkipCertificateChainValidation $true -Proxy $credential -Description "backend with proxy server"  
Creates a Backend Proxy Object and sets up Backend
```

Incorrect Answers:

- A: The Import-AzureRmApiManagementApi cmdlet imports an Azure API Management API from a file or a URL in Web Application Description Language (WADL), Web Services Description Language (WSDL), or Swagger format.
- B: New-AzureRmApiManagementBackend creates a new backend entity in Api Management.
- C: The New-AzureRmApiManagement cmdlet creates an API Management deployment in Azure API Management.

Reference:

<https://docs.microsoft.com/en-us/powershell/module/azurerm.apimanagement/new-azurermapimanagementbackendproxy?view=azurermps-6.13.0>

QUESTION 200

You are creating a hazard notification system that has a single signaling server which triggers audio and visual alarms to start and stop.

You implement Azure Service Bus to publish alarms. Each alarm controller uses Azure Service Bus to receive alarm signals as part of a transaction. Alarm events must be recorded for audit purposes. Each transaction record must include information about the alarm type that was activated.

You need to implement a reply trail auditing solution.

Which two actions should you perform? Each correct answer resents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Assign the value of the hazard message SessionID property to the ReplyToSessionId property.
- B. Assign the value of the hazard message MessageId property to the DeliveryCount property.
- C. Assign the value of the hazard message SessionID property to the SequenceNumber property.
- D. Assign the value of the hazard message MessageId property to the CorrelationId property.
- E. Assign the value of the hazard message SequenceNumber property to the DeliveryCount property.
- F. Assign the value of the hazard message MessageId property to the SequenceNumber property.

Answer: AD

Explanation:

D: CorrelationId: Enables an application to specify a context for the message for the purposes of correlation; for example, reflecting the MessageId of a message that is being replied to.

A: ReplyToSessionId: This value augments the ReplyTo information and specifies which

SessionId should be set for the reply when sent to the reply entity.

Incorrect Answers:

B, E: DeliveryCount

Number of deliveries that have been attempted for this message. The count is incremented when a message lock expires, or the message is explicitly abandoned by the receiver. This property is read-only.

C, E: SequenceNumber

The sequence number is a unique 64-bit integer assigned to a message as it is accepted and stored by the broker and functions as its true identifier. For partitioned entities, the topmost 16 bits reflect the partition identifier. Sequence numbers monotonically increase and are gapless. They roll over to 0 when the 48-64 bit range is exhausted. This property is read-only.

Reference:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messages-payloads>

QUESTION 201

You are developing an Azure messaging solution.

You need to ensure that the solution meets the following requirements:

- Provide transactional support.
- Provide duplicate detection.
- Store the messages for an unlimited period of time.

Which two technologies will meet the requirements? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Azure Service Bus Topic
- B. Azure Service Bus Queue
- C. Azure Storage Queue
- D. Azure Event Hub

Answer: AB

Explanation:

The Azure Service Bus Queue and Topic has duplicate detection.

Enabling duplicate detection helps keep track of the application-controlled MessageId of all messages sent into a queue or topic during a specified time window.

Incorrect Answers:

C: There is just no mechanism that can query a Storage queue and find out if a message with the same contents is already there or was there before.

D: Azure Event Hub does not have duplicate detection

Reference:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/duplicate-detection>

QUESTION 202

Drag and Drop Question

You plan to create a Docker image that runs an ASP.NET Core application named ContosoApp. You have a setup script named setupScript.ps1 and a series of application files including ContosoApp.dll.

You need to create a Dockerfile document that meets the following requirements:

- Call setupScripts.ps1 when the container is built.
- Run ContosoApp.dll when the container starts.

The Dockerfile document must be created in the same folder where ContosoApp.dll and setupScript.ps1 are stored.

Which five commands should you use to develop the solution? To answer, move the appropriate commands from the list of commands to the answer area and arrange them in the correct order.

| Commands | Answer Area |
|----------------------------------|-------------|
| FROM microsoft/aspnetcore:latest | |
| WORKDIR /apps/ContosoApp | |
| CMD ["dotnet", "ContosoApp.dll"] | |
| COPY ./ . | |
| RUN powershell ./setupScript.ps1 | |

Answer:

| Commands | Answer Area |
|----------|----------------------------------|
| | CMD ["dotnet", "ContosoApp.dll"] |
| | FROM microsoft/aspnetcore:latest |
| | WORKDIR /apps/ContosoApp |
| | COPY ./ . |
| | RUN powershell ./setupScript.ps1 |

Explanation:

Box 1: CMD [..]

Cmd starts a new instance of the command interpreter, Cmd.exe.

Syntax: CMD <string>

Specifies the command you want to carry out.

Box 2: FROM microsoft/aspnetcore-build:latest

Box 3: WORKDIR /apps/ContosoApp

Bx 4: COPY ./ .

Box 5: RUN powershell ./setupScript.ps1

QUESTION 203

Hotspot Question

You are configuring a new development environment for a Java application.

The environment requires a Virtual Machine Scale Set (VMSS), several storage accounts, and networking components.

The VMSS must not be created until the storage accounts have been successfully created and an associated load balancer and virtual network is configured.

How should you complete the Azure Resource Manager template? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
{  
  . . .  
  "resources": [  
    {  
      "apiVersion": "2016-01-01",  
      "type": "Microsoft.Storage/storageAccounts",  
      "name": "[concat(  (), 'storage', uniqueString(resourceGroup().id))]",  
      "dependsOn": [  
        "copy",  
        "copyIndex",  
        "priority",  
        "dependsOn"  
      ]  
      "location": "[resourceGroup().location]",  
      . . .  
      "sku": {  
        "name": "Standard_LRS"  
      },  
      "kind": "Storage",  
      "properties": {},  
      "dependsOn": [  
        "copy",  
        "copyIndex",  
        "priority",  
        "dependsOn"  
      ]  
      "name": "storagesetup",  
      "count": 3  
    }  
  },  
  {  
    "apiVersion": "2015-06-15",  
    "type": "Microsoft.Compute/virtualMachines",  
    "name": "[concat('VM', uniqueString(resourceGroup().id))]",  
    "dependsOn": [  
      "copy",  
      "copyIndex",  
      "priority",  
      "dependsOn"  
    ]  
    "[variables('loadBalancerName')]",  
    "[variables('virtualNetworkName')]",  
    "storagesetup",  
    ],  
    . . .  
  }  
],  
"outputs": {}  
}
```

Answer:

Answer Area

```
{  
  . . .  
  "resources": [  
    {  
      "apiVersion": "2016-01-01",  
      "type": "Microsoft.Storage/storageAccounts",  
      "name": "[concat(          (), 'storage', uniqueString(resourceGroup().id))]",  
      "dependsOn": [  
        "copy",  
        "copyIndex",  
        "priority",  
        "dependsOn"  
      ]  
    },  
    {  
      "location": "[resourceGroup().location]",  
      "sku": {  
        "name": "Standard_LRS"  
      },  
      "kind": "Storage",  
      "properties": {},  
      "copy": {  
        "dependsOn": [  
          "storageSetup",  
          "count": 3  
        ]  
      },  
      "dependsOn": [  
        "copy",  
        "copyIndex",  
        "priority",  
        "dependsOn"  
      ]  
    },  
    {  
      "apiVersion": "2015-06-15",  
      "type": "Microsoft.Compute/virtualMachines",  
      "name": "[concat('VM', uniqueString(resourceGroup().id))]",  
      "dependsOn": [  
        "copy",  
        "copyIndex",  
        "priority",  
        "dependsOn"  
      ],  
      "variables": [  
        "[variables('loadBalancerName')]",  
        "[variables('virtualNetworkName')]",  
        "storageSetup",  
        []  
      ],  
      "copy": {  
        "dependsOn": [  
          "loadBalancerSetup",  
          "virtualNetworkSetup",  
          "storageSetup",  
          "vmSetup",  
          "count": 3  
        ]  
      },  
      "outputs": {}  
    }  
  ]  
}
```

Explanation:**Box 1: copyIndex**

Notice that the name of each resource includes the `copyIndex()` function, which returns the current iteration in the loop. `copyIndex()` is zero-based.

Box 2: copy

By adding the `copy` element to the resources section of your template, you can dynamically set the number of resources to deploy.

Box 3: dependsOn**Example:**

```
"type": "Microsoft.Compute/virtualMachineScaleSets",  
"apiVersion": "2020-06-01",  
"name": "[variables('namingInfix')]",  
"location": "[parameters('location')]",  
"sku": {  
  "name": "[parameters('vmSku')]"  
}
```

```
"tier": "Standard",
"capacity": "[parameters('instanceCount')]"
},
"dependsOn": [
"[resourceId('Microsoft.Network/loadBalancers', variables ('loadBalancerName'))]",
"[resourceId('Microsoft.Network/virtualNetworks', variables ('virtualNetworkName'))]"
],
```

Reference:

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/copy-resources>
<https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/quick-create-template-windows>

QUESTION 204

Hotspot Question

You are developing an Azure Function App by using Visual Studio. The app will process orders input by an Azure Web App. The web app places the order information into Azure Queue Storage.

You need to review the Azure Function App code shown below.

```
public static class OrderProcessor
{
    [FunctionName("ProcessOrders")]
    public static void ProcessOrders([QueueTrigger("incoming-orders")]CloudQueueMessage myQueueItem, [Table("Orders")]ICollector<Order> tableBindings, TraceWriter log)
    {
        log.Info($"Processing Order: {myQueueItem.Id}");
        log.Info($"Queue Insertion Time: {myQueueItem.InsertionTime}");
        log.Info($"Queue Expiration Time: {myQueueItem.ExpirationTime}");
        tableBindings.Add(JsonConvert.DeserializeObject<Order>(myQueueItem.AsString));
    }
    [FunctionName("ProcessOrders-Poison")]
    public static void ProcessFailedOrders([QueueTrigger("incoming-orders-poison")]CloudQueueMessage myQueueItem, TraceWriter log)
    {
        log.Error($"Failed to process order: {myQueueItem.AsString}");
    }
}
```

NOTE: Each correct selection is worth one point.

Answer Area

| Yes | No |
|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> |

The code will log the time that the order was processed from the queue.

When the ProcessOrders function fails, the function will retry up to five times for a given order, including the first try.

When there are multiple orders in the queue, a batch of orders will be retrieved from the queue and the ProcessOrders function will run multiple instances concurrently to process the orders.

The ProcessOrders function will output the order to an Orders table in Azure Table Storage.

Answer:

Answer Area

| | Yes | No |
|--|----------------------------------|----------------------------------|
| The code will log the time that the order was processed from the queue. | <input type="radio"/> | <input checked="" type="radio"/> |
| When the ProcessOrders function fails, the function will retry up to five times for a given order, including the first try. | <input checked="" type="radio"/> | <input type="radio"/> |
| When there are multiple orders in the queue, a batch of orders will be retrieved from the queue and the ProcessOrders function will run multiple instances concurrently to process the orders. | <input checked="" type="radio"/> | <input type="radio"/> |
| The ProcessOrders function will output the order to an Orders table in Azure Table Storage. | <input checked="" type="radio"/> | <input type="radio"/> |

Explanation:

Box 1: No

ExpirationTime -The time that the message expires.

InsertionTime -The time that the message was added to the queue.

Box 2: Yes

maxDequeueCount -The number of times to try processing a message before moving it to the poison queue. Default value is 5.

Box 3: Yes

When there are multiple queue messages waiting, the queue trigger retrieves a batch of messages and invokes function instances concurrently to process them. By default, the batch size is 16. When the number being processed gets down to 8, the runtime gets another batch and starts processing those messages. So the maximum number of concurrent messages being processed per function on one virtual machine (VM) is 24.

Box 4: Yes

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-queue>**QUESTION 205**

Drag and Drop Question

You are developing a solution for a hospital to support the following use cases:

- The most recent patient status details must be retrieved even if multiple users in different locations have updated the patient record.
- Patient health monitoring data retrieved must be the current version or the prior version.
- After a patient is discharged and all charges have been assessed, the patient billing record contains the final charges.

You provision a Cosmos DB NoSQL database and set the default consistency level for the database account to Strong. You set the value for Indexing Mode to Consistent.

You need to minimize latency and any impact to the availability of the solution. You must override the default consistency level at the query level to meet the required consistency guarantees for the scenarios.

Which consistency levels should you implement? To answer, drag the appropriate consistency

levels to the correct requirements. Each consistency level 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.

| Consistency levels | Answer Area |
|--------------------|---|
| Strong | Return the most recent patient status. |
| Consistent Prefix | Return health monitoring data that is no less than one version behind. |
| | After patient is discharged and all charges are assessed, retrieve the correct billing data with the final charges. |

Answer:

| Consistency levels | Answer Area |
|--------------------|---|
| Consistent Prefix | Return the most recent patient status. |
| | Return health monitoring data that is no less than one version behind. |
| | After patient is discharged and all charges are assessed, retrieve the correct billing data with the final charges. |

Explanation:

Box 1: Strong

Strong: Strong consistency offers a linearizability guarantee. The reads are guaranteed to return the most recent committed version of an item. A client never sees an uncommitted or partial write. Users are always guaranteed to read the latest committed write.

Box 2: Bounded staleness

Bounded staleness: The reads are guaranteed to honor the consistent-prefix guarantee. The reads might lag behind writes by at most "K" versions (that is "updates") of an item or by "t" time interval. When you choose bounded staleness, the "staleness" can be configured in two ways:

The number of versions (K) of the item

The time interval (t) by which the reads might lag behind the writes

Box 3: Eventual

Eventual: There's no ordering guarantee for reads. In the absence of any further writes, the replicas eventually converge.

Incorrect Answers:

Consistent prefix: Updates that are returned contain some prefix of all the updates, with no gaps.

Consistent prefix guarantees that reads never see out-of-order writes.

Reference:

<https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels>

QUESTION 206

Hotspot Question

You are configuring a development environment for your team. You deploy the latest Visual Studio image from the Azure Marketplace to your Azure subscription.

The development environment requires several software development kits (SDKs) and third-party

components to support application development across the organization. You install and customize the deployed virtual machine (VM) for your development team. The customized VM must be saved to allow provisioning of a new team member development environment.

You need to save the customized VM for future provisioning.

Which tools or services should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

| Action | Tool or service |
|--------------------|--|
| Generalize the VM. | Azure PowerShell Visual Studio command prompt Azure Migrate Azure Backup |
| Store images. | Azure Blob Storage Azure Data Lake Storage Azure File Storage Azure Table Storage |

Answer:

Answer Area

| Action | Tool or service |
|--------------------|--|
| Generalize the VM. | Azure PowerShell Visual Studio command prompt Azure Migrate Azure Backup |
| Store images. | Azure Blob Storage Azure Data Lake Storage Azure File Storage Azure Table Storage |

Explanation:

Box 1: Azure Powershell

Creating an image directly from the VM ensures that the image includes all of the disks associated with the VM, including the OS disk and any data disks.

Before you begin, make sure that you have the latest version of the Azure PowerShell module.

You use Sysprep to generalize the virtual machine, then use Azure PowerShell to create the image.

Box 2: Azure Blob Storage

You can store images in Azure Blob Storage.

Reference:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/capture-image-resource#create-an-image-of-a-vm-using-powershell>

QUESTION 207

Hotspot Question

You are developing a back-end Azure App Service that scales based on the number of messages contained in a Service Bus queue.

A rule already exists to scale up the App Service when the average queue length of unprocessed and valid queue messages is greater than 1000.

You need to add a new rule that will continuously scale down the App Service as long as the scale up condition is not met.

How should you configure the Scale rule? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer AreaScale rule x

Metric source

| |
|-------------------------|
| Storage queue |
| Service Bus queue |
| Current resource |
| Storage queue (classic) |

Resource type

| |
|------------------------|
| Service Bus Namespaces |
|------------------------|

Resource

| |
|------------------|
| MessageQueue1103 |
|------------------|

* Queues

| |
|-----------|
| itemqueue |
|-----------|

Criteria

* Metric name

| |
|----------------------|
| Message Count |
| Active Message Count |

1 minute time grain

* Time grain statistic •

| |
|---------|
| Total |
| Maximum |
| Average |
| Count |

* Operator

| |
|--------------------------|
| Greater than |
| Greater than or equal to |
| Less than |
| Less than or equal to |

* Threshold

| |
|------|
| 1000 |
|------|

Action

* Operation

| |
|-------------------|
| Increase count by |
| Increase count to |
| Decrease count by |
| Decrease count to |

Answer:**Answer Area**Scale rule x

Metric source

| |
|-------------------------|
| Storage queue |
| Service Bus queue |
| Current resource |
| Storage queue (classic) |

Resource type

| |
|------------------------|
| Service Bus Namespaces |
|------------------------|

Resource

| |
|------------------|
| MessageQueue1103 |
|------------------|

* Queues

| |
|-----------|
| itemqueue |
|-----------|

Criteria

* Metric name

| |
|----------------------|
| Message Count |
| Active Message Count |

1 minute time grain

* Time grain statistic •

| |
|---------|
| Total |
| Maximum |
| Average |
| Count |

* Operator

| |
|--------------------------|
| Greater than |
| Greater than or equal to |
| Less than |
| Less than or equal to |

* Threshold

| |
|------|
| 1000 |
|------|

Action

* Operation

| |
|-------------------|
| Increase count by |
| Increase count to |
| Decrease count by |
| Decrease count to |

Explanation:

Box 1: Service bus queue

You are developing a back-end Azure App Service that scales based on the number of messages contained in a Service Bus queue.

Box 2: ActiveMessage Count

ActiveMessageCount: Messages in the queue or subscription that are in the active state and ready for delivery.

Box 3: Count

Box 4: Less than or equal to

You need to add a new rule that will continuously scale down the App Service as long as the scale up condition is not met.

Box 5: Decrease count by

QUESTION 208

Drag and Drop Question

You have an application that uses Azure Blob storage.

You need to update the metadata of the blobs.

Which three methods should you use to develop the solution? To answer, move the appropriate methods from the list of methods to the answer area and arrange them in the correct order.

Methods

Metadata.Add

SetMetadataAsync

FetchAttributesAsync

UploadFileStream

SetPropertiesAsync

Answer Area

Answer:

Methods

FetchAttributesAsync

UploadFileStream

Answer Area

Metadata.Add

SetMetadataAsync

SetPropertiesAsync

Explanation:

Metadata.Add example:

```
// Add metadata to the dictionary by calling the Add method metadata.Add("docType",  
"textDocuments");
```

SetMetadataAsync example:

```
// Set the blob's metadata.  
await blob.SetMetadataAsync(metadata);
```

// Set the blob's properties.

```
await blob.SetPropertiesAsync();
```

Reference:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-properties-metadata>

QUESTION 209

Drag and Drop Question

You are developing an application to securely transfer data between on-premises file systems and Azure Blob storage. The application stores keys, secrets, and certificates in Azure Key Vault. The application uses the Azure Key Vault APIs.

The application must allow recovery of an accidental deletion of the key vault or key vault objects. Key vault objects must be retained for 90 days after deletion.

You need to protect the key vault and key vault objects.

Which Azure Key Vault feature should you use? To answer, drag the appropriate features to the correct actions. Each feature 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.

| Features | Answer Area | Action | Feature |
|-------------------------|-------------|---|---------|
| Access policy | | Enable retention period and accidental deletion. | Feature |
| Purge protection | | | Feature |
| Soft delete | | | |
| Shared access signature | | Enforce retention period and accidental deletion. | Feature |

Answer:

| Features | Answer Area | Action | Feature |
|-------------------------|-------------|---|------------------|
| Access policy | | Enable retention period and accidental deletion. | Soft delete |
| Shared access signature | | Enforce retention period and accidental deletion. | Purge protection |

Explanation:

Box 1: Soft delete

When soft-delete is enabled, resources marked as deleted resources are retained for a specified period (90 days by default). The service further provides a mechanism for recovering the deleted object, essentially undoing the deletion.

Box 2: Purge protection

Purge protection is an optional Key Vault behavior and is not enabled by default. Purge protection can only be enabled once soft-delete is enabled.

When purge protection is on, a vault or an object in the deleted state cannot be purged until the retention period has passed. Soft-deleted vaults and objects can still be recovered, ensuring that the retention policy will be followed.

Reference:

<https://docs.microsoft.com/en-us/azure/key-vault/general/soft-delete-overview>

QUESTION 210

Drag and Drop Question

You are developing an ASP.NET Core website that can be used to manage photographs which are stored in Azure Blob Storage containers.

Users of the website authenticate by using their Azure Active Directory (Azure AD) credentials.

You implement role-based access control (RBAC) role permissions on the containers that store photographs. You assign users to RBAC roles.

You need to configure the website's Azure AD Application so that user's permissions can be used with the Azure Blob containers.

How should you configure the application? To answer, drag the appropriate setting to the correct location. Each setting can 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.

| Settings | Answer Area | |
|--------------------|-------------|---------|
| client_id | | |
| profile | | |
| delegated | | |
| application | | |
| user_impersonation | | |
| API | Permission | Type |
| Azure Storage | Setting | Setting |
| Microsoft Graph | User.Read | Setting |

Answer:

| Settings | Answer Area | |
|--------------------|--------------------|-----------|
| client_id | | |
| profile | | |
| delegated | | |
| application | | |
| user_impersonation | | |
| API | Permission | Type |
| Azure Storage | user_impersonation | delegated |
| Microsoft Graph | User.Read | delegated |

Explanation:

Box 1: user_impersonation

Box 2: delegated

Example:

1. Select the API permissions section
2. Click the Add a permission button and then:
Ensure that the My APIs tab is selected
3. In the list of APIs, select the API TodoListService-aspnetcore.
4. In the Delegated permissions section, ensure that the right permissions are checked:
user_impersonation.
5. Select the Add permissions button.

Box 3: delegated

Example

1. Select the API permissions section
2. Click the Add a permission button and then,
Ensure that the Microsoft APIs tab is selected
3. In the Commonly used Microsoft APIs section, click on Microsoft Graph
4. In the Delegated permissions section, ensure that the right permissions are checked:
User.Read. Use the search box if necessary.
5. Select the Add permissions button

Reference:

<https://docs.microsoft.com/en-us/samples/azure-samples/active-directory-dotnet-webapp-webapi-openidconnect-aspnetcore/calling-a-web-api-in-an-aspnet-core-web-application-using-azure-ad/>

QUESTION 211

Hotspot Question

You are developing an ASP.NET Core app that includes feature flags which are managed by Azure App Configuration. You create an Azure App Configuration store named AppFeatureFlagStore that contains a feature flag named Export.

You need to update the app to meet the following requirements:

- Use the Export feature in the app without requiring a restart of the app.
- Validate users before users are allowed access to secure resources.
- Permit users to access secure resources.

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.

Answer Area

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
    }

    app.      ();
```

| |
|-------------------|
| UseAuthentication |
| UseStaticFiles |
| UseSession |
| UseCookiePolicy |

```
    app.      ();
```

| |
|---------------------|
| UseAuthorization |
| UseHttpsRedirection |
| UseSession |
| UseCookiePolicy |

```
    app.      ();
```

| |
|--------------------------|
| UseAzureAppConfiguration |
| UseRequestLocalization |
| UseCors |
| UseStaticFiles |

```
    app.UseEndpoint(endpoints =>
    {
        endpoints.MapRazorPages();
    });
}
```

Answer:

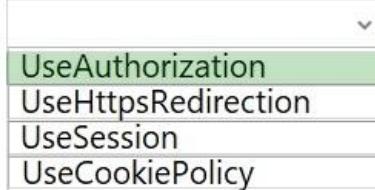
Answer Area

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
    }

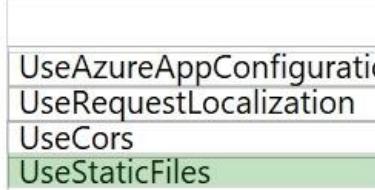
    app. ();
```



```
app. ();
```



```
app. ();
```



```
app.UseEndpoint(endpoints =>
{
    endpoints.MapRazorPages();
});
```

Explanation:

Box 1: UseAuthentication

Need to validate users before users are allowed access to secure resources.

UseAuthentication adds the AuthenticationMiddleware to the specified IApplicationBuilder, which enables authentication capabilities.

Box 2: UseAuthorization

Need to permit users to access secure resources.

UseAuthorization adds the AuthorizationMiddleware to the specified IApplicationBuilder, which enables authorization capabilities.

Box 3: UseStaticFiles

Need to use the Export feature in the app without requiring a restart of the app.

UseStaticFiles enables static file serving for the current request path

Reference:

<https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.builder.iapplicationbuilder?view=aspnetcore-5.0>

QUESTION 212

Drag and Drop Question

A company has multiple warehouse. Each warehouse contains IoT temperature devices which deliver temperature data to an Azure Service Bus queue.

You need to send email alerts to facility supervisors immediately if the temperature at a warehouse goes above or below specified threshold temperatures.

Which five 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.

| Actions | Answer Area |
|---|-------------|
| Add a logic app trigger that fires when one or more messages arrive in the queue. | |
| Add a Recurrence trigger that schedules the app to run every 15 minutes. | |
| Add an action that sends an email to specified personnel if the temperature is outside of those thresholds. | |
| Add a trigger that reads IoT temperature data from a Service Bus queue. | |
| Add a logic app action that fires when one or more messages arrive in the queue. | |
| Add a condition that compares the temperature against the upper and lower thresholds. | |
| Create a blank Logic app. | |
| Add an action that reads IoT temperature data from the Service Bus queue. | |

Answer:

Actions

Add a logic app trigger that fires when one or more messages arrive in the queue.

Add a Recurrence trigger that schedules the app to run every 15 minutes.

Add a trigger that reads IoT temperature data from a Service Bus queue.

Answer Area

Create a blank Logic app.

Add a logic app action that fires when one or more messages arrive in the queue.

Add an action that reads IoT temperature data from the Service Bus queue.

Add an action that sends an email to specified personnel if the temperature is outside of those thresholds.

Add a condition that compares the temperature against the upper and lower thresholds.

Explanation:

Step 1: Create a blank Logic app.
Create and configure a Logic App.

Step 2: Add a logical app trigger that fires when one or more messages arrive in the queue.
Configure the logic app trigger.
Under Triggers, select When one or more messages arrive in a queue (auto-complete).

Step 3: Add an action that reads IoT temperature data from the Service Bus queue

Step 4: Add a condition that compares the temperature against the upper and lower thresholds.

Step 5: Add an action that sends an email to specified personnel if the temperature is outside of those thresholds

Reference: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-monitoring-notifications-with-azure-logic-apps>

QUESTION 213**Case Study 1 - Litware Inc****Background**

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

- Uploading using an Azure Files mounted folder

- Uploading using the web application

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details**Users table**

| Column | Description |
|----------------|--|
| UserId | unique identifier for an employee |
| ExpenseAccount | employees expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements**Receipt processing**

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.

- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues

Upload format issue

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

Capacity issue

During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

Log capacity issue

Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlob (CloudBlockBlob sourceBlob)
PC20     {
PC21         ...
PC22     }
PC23     private static async Task<CloudBlobContainer> GetCloudBlobContainer()
PC24     {
PC25         var cloudBlobClient = new CloudBlobClient (new Uri(" . . ."), await GetCredentials());
PC26
PC27         await cloudBlobClient.GetRootContainerReference().CreateIfNotExistAsync();
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(HttpStatusCode 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
```

You need to ensure receipt processing occurs correctly.

What should you do?

- A. Use blob properties to prevent concurrency problems
- B. Use blob SnapshotTime to prevent concurrency problems
- C. Use blob metadata to prevent concurrency problems
- D. Use blob leases to prevent concurrency problems

Answer: B

Explanation:

You can create a snapshot of a blob. A snapshot is a read-only version of a blob that's taken at a point in time. Once a snapshot has been created, it can be read, copied, or deleted, but not modified. Snapshots provide a way to back up a blob as it appears at a moment in time.

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.

Reference:

<https://docs.microsoft.com/en-us/rest/api/storageservices/creating-a-snapshot-of-a-blob>

QUESTION 214

Case Study 1 - Litware Inc

Background

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

- Uploading using an Azure Files mounted folder
- Uploading using the web application

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details

Users table

| Column | Description |
|----------------|--|
| UserId | unique identifier for an employee |
| ExpenseAccount | employees expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements

Receipt processing

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.

- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues

Upload format issue

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

Capacity issue

During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

Log capacity issue

Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlob (CloudBlockBlob sourceBlob)
PC20     {
PC21         ...
PC22     }
PC23     private static async Task<CloudBlobContainer> GetCloudBlobContainer()
PC24     {
PC25         var cloudBlobClient = new CloudBlobClient (new Uri(" . . ."), await GetCredentials());
PC26
PC27         await cloudBlobClient.GetRootContainerReference().CreateIfNotExistAsync();
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(HttpStatusCode 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
```

You need to resolve the capacity issue.

What should you do?

- A. Convert the trigger on the Azure Function to an Azure Blob storage trigger
- B. Ensure that the consumption plan is configured correctly to allow scaling
- C. Move the Azure Function to a dedicated App Service Plan
- D. Update the loop starting on line PC09 to process items in parallel

Answer: D

Explanation:

If you want to read the files in parallel, you cannot use foreach. Each of the async callback function calls does return a promise. You can await the array of promises that you'll get with Promise.all.

Scenario: Capacity issue: During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

```
PC08     var container = await GetCloudBlobContainer();
PC09     foreach (var fileItem in await ListFiles())
PC10     {
PC11         var file = new CloudFile(fileItem.StorageUri.PrimaryUri);
PC12         var ms = new MemoryStream();
PC13         await file.DownloadToStreamAsync(ms);
PC14         var blob = container.GetBlockBlobReference(fileItem.Uri.ToString());
PC15         await blob.UploadFromStreamAsync(ms);
PC16
PC17 }
```

Reference:

<https://stackoverflow.com/questions/37576685/using-async-await-with-a-foreach-loop>

QUESTION 215

Case Study 1 - Litware Inc

Background

You are a developer for Litware Inc., a SaaS company that provides a solution for managing employee expenses. The solution consists of an ASP.NET Core Web API project that is deployed as an Azure Web App.

Overall architecture

Employees upload receipts for the system to process. When processing is complete, the employee receives a summary report email that details the processing results. Employees then use a web application to manage their receipts and perform any additional tasks needed for reimbursement.

Receipt processing

Employees may upload receipts in two ways:

- Uploading using an Azure Files mounted folder
- Uploading using the web application

Data Storage

Receipt and employee information is stored in an Azure SQL database.

Documentation

Employees are provided with a getting started document when they first use the solution. The documentation includes details on supported operating systems for Azure File upload, and instructions on how to configure the mounted folder.

Solution details**Users table**

| Column | Description |
|----------------|---|
| UserId | unique identifier for an employee |
| ExpenseAccount | employee's expense account number in the format 1234-123-1234 |
| AllowedAmount | limit of allowed expenses before approval is needed |
| SupervisorId | unique identifier for employee's supervisor |
| SecurityPin | value used to validate user identity |

Web Application

You enable MSI for the Web App and configure the Web App to use the security principal name.

Processing

Processing is performed by an Azure Function that uses version 2 of the Azure Function runtime. Once processing is completed, results are stored in Azure Blob Storage and an Azure SQL database. Then, an email summary is sent to the user with a link to the processing report. The link to the report must remain valid if the email is forwarded to another user.

Requirements**Receipt processing**

Concurrent processing of a receipt must be prevented.

Logging

Azure Application Insights is used for telemetry and logging in both the processor and the web application. The processor also has TraceWriter logging enabled. Application Insights must always contain all log messages.

Disaster recovery

Regional outage must not impact application availability. All DR operations must not be dependent on application running and must ensure that data in the DR region is up to date.

Security

- Users' SecurityPin must be stored in such a way that access to the database does not allow the viewing of SecurityPins. The web application is the only system that should have access to SecurityPins.
- All certificates and secrets used to secure data must be stored in Azure Key Vault.
- You must adhere to the Least Privilege Principal.
- All access to Azure Storage and Azure SQL database must use the application's Managed Service Identity (MSI)
- Receipt data must always be encrypted at rest.
- All data must be protected in transit.
- User's expense account number must be visible only to logged in users. All other views of the expense account number should include only the last segment with the remaining parts obscured.
- In the case of a security breach, access to all summary reports must be revoked without impacting other parts of the system.

Issues**Upload format issue**

Employees occasionally report an issue with uploading a receipt using the web application. They report that when they upload a receipt using the Azure File Share, the receipt does not appear in their profile. When this occurs, they delete the file in the file share and use the web application, which returns a 500 Internal Server error page.

Capacity issue

During busy periods, employees report long delays between the time they upload the receipt and when it appears in the web application.

Log capacity issue

Developers report that the number of log messages in the trace output for the processor is too high, resulting in lost log messages.

Processing.cs

```
PC01 public static class Processing
PC02 {
PC03     public static class Function
PC04     {
PC05         [FunctionName ("IssueWork")]
PC06         public static async Task Run ([TimerTrigger("0 */5 * * *")] TimerInfo timer, ILogger log)
PC07         {
PC08             var container = await GetCloudBlobContainer();
PC09             foreach (var fileItem in await ListFiles())
PC10             {
PC11                 var file = new CloudFile (fileItem.StorageUri.PrimaryUri);
PC12                 var ms = new MemoryStream();
PC13                 await file.DownloadToStreamAsync(ms);
PC14                 var blob = container.GetBlockBlobReference (fileItem.Uri.ToString());
PC15                 await blob.UploadFromStreamAsync(ms);
PC16             }
PC17         }
PC18     }
PC19     private static CloudBlockBlob GetDRBlock (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().CreateIfNotExistAsync();
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(HttpStatusCode 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
```

Hotspot Question

You need to add code at line PC26 of Processing.cs to ensure that security policies are met.

How should you complete the code that you will add at line PC26? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
var resolver = new KeyVaultKeyResolver(_keyVaultClient);
var keyBundle = await _keyVaultClient.GetKeyAsync("...", "...");
```

```
var key = keyBundle.Key;
var key = keyBundle.KeyIdentifier.Identifier;
var key = await resolver.ResolveKeyAsync("encrypt", null);
var key = await resolver.ResolveKeyAsync(keyBundle.KeyIdentifier.Identifier, CancellationToken.None);
```

```
var x = keyBundle.Managed;
var x = AuthenticationScheme.SharedKey;
var x = new BlobEncryptionPolicy(key, resolver);
var x = new DeleteRetentionPolicy {Enabled = key.Kid != null};
```

```
cloudBlobClient.AuthenticationScheme = x;
cloudBlobClient.DefaultRequestOptions.RequireEncryption = x;
cloudBlobClient.DefaultRequestOptions.EncryptionPolicy = x;
cloudBlobClient.SetServiceProperties(new ServiceProperties(deleteRetentionPolicy:x));
```

Answer:**Answer Area**

```
var resolver = new KeyVaultKeyResolver(_keyVaultClient);
var keyBundle = await _keyVaultClient.GetKeyAsync("...", "...");
```

```
var key = keyBundle.Key;
var key = keyBundle.KeyIdentifier.Identifier;
var key = await resolver.ResolveKeyAsync("encrypt", null);
var key = await resolver.ResolveKeyAsync(keyBundle.KeyIdentifier.Identifier, CancellationToken.None);
```

```
var x = keyBundle.Managed;
var x = AuthenticationScheme.SharedKey;
var x = new BlobEncryptionPolicy(key, resolver);
var x = new DeleteRetentionPolicy {Enabled = key.Kid != null};
```

```
cloudBlobClient.AuthenticationScheme = x;
cloudBlobClient.DefaultRequestOptions.RequireEncryption = x;
cloudBlobClient.DefaultRequestOptions.EncryptionPolicy = x;
cloudBlobClient.SetServiceProperties(new ServiceProperties(deleteRetentionPolicy:x));
```

Explanation:

Box 1: var key = await
Resolver.ResolveKeyAsyn(keyBundle,KeyIdentifier.CancellationToken.None);

Box 2: var x = new BlobEncryptionPolicy(key,resolver); Example:
// We begin with cloudKey1, and a resolver capable of resolving and caching Key Vault secrets.
BlobEncryptionPolicy encryptionPolicy = new BlobEncryptionPolicy(cloudKey1, cachingResolver);
client.DefaultRequestOptions.EncryptionPolicy = encryptionPolicy;

Box 3: cloudblobClient. DefaultRequestOptions.EncryptionPolicy = x;

Reference:

<https://github.com/Azure/azure-storage-net/blob/master/Samples/GettingStarted/EncryptionSamples/KeyRotation/Program.cs>

QUESTION 216

Case Study 8 - Contoso, Ltd

Background

Overview

You are a developer for Contoso, Ltd. The company has a social networking website that is developed as a Single Page Application (SPA). The main web application for the social networking website loads user uploaded content from blob storage.

You are developing a solution to monitor uploaded data for inappropriate content. The following process occurs when users upload content by using the SPA:

- Messages are sent to ContentUploadService.
- Content is processed by ContentAnalysisService.
- After processing is complete, the content is posted to the social network or a rejection message is posted in its place.

The ContentAnalysisService is deployed with Azure Container Instances from a private Azure Container Registry named contosoiimages.

The solution will use eight CPU cores.

Azure Active Directory

Contoso, Ltd. uses Azure Active Directory (Azure AD) for both internal and guest accounts.

Requirements

ContentAnalysisService

The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

You must create an Azure Function named CheckUserContent to perform the content checks.

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from internal Virtual Networks (VNets).
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso, Ltd. to review content, store cookies on user devices, and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location: westus
CS04 name: contentUploadService
CS05 properties:
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpu: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22     protocol: TCP
CS23
CS24
CS25 networkProfile:
CS26
id: /subscriptions/98...19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

ApplicationManifest

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14     "legalAgeGroupRule" : "Allow"
AM15   },
AM16   "passwordCredentials" : []
AM17 }
```

You need to investigate the http server log output to resolve the issue with the ContentUploadService.

Which command should you use first?

- A. az webapp log
- B. az ams live-output
- C. az monitor activity-log

D. az container attach

Answer: C

Explanation:

Scenario: Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

"502 bad gateway" and "503 service unavailable" are common errors in your app hosted in Azure App Service.

Microsoft Azure publicizes each time there is a service interruption or performance degradation. The az monitor activity-log command manages activity logs.

Note: Troubleshooting can be divided into three distinct tasks, in sequential order:

1. Observe and monitor application behavior
2. Collect data
3. Mitigate the issue

Reference:

<https://docs.microsoft.com/en-us/cli/azure/monitor/activity-log>

QUESTION 217

Case Study 8 - Contoso, Ltd

Background

Overview

You are a developer for Contoso, Ltd. The company has a social networking website that is developed as a Single Page Application (SPA). The main web application for the social networking website loads user uploaded content from blob storage.

You are developing a solution to monitor uploaded data for inappropriate content. The following process occurs when users upload content by using the SPA:

- Messages are sent to ContentUploadService.
- Content is processed by ContentAnalysisService.
- After processing is complete, the content is posted to the social network or a rejection message is posted in its place.

The ContentAnalysisService is deployed with Azure Container Instances from a private Azure Container Registry named contosoimages.

The solution will use eight CPU cores.

Azure Active Directory

Contoso, Ltd. uses Azure Active Directory (Azure AD) for both internal and guest accounts.

Requirements

ContentAnalysisService

The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

You must create an Azure Function named CheckUserContent to perform the content checks.

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from internal Virtual Networks (VNets).
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso, Ltd. to review content, store cookies on user devices, and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code

ContentUploadService

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location: westus
CS04 name: contentUploadService
CS05 properties:
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpu: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol: TCP
CS23
CS24
CS25 networkProfile:
CS26
id: /subscriptions/98...19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

ApplicationManifest

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14       "legalAgeGroupRule" : "Allow"
AM15   },
AM16   "passwordCredentials" : []
AM17 }
```

Hotspot Question

You need to ensure that validation testing is triggered per the requirements.

How should you complete the code segment? To answer, select the appropriate values in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
var event = getEvent();
if (event.eventType === 'ImagePushed'
    && event.data.target === 'contentanalysisservice'
    && event.data.contains('contosoimages'))
{
    startValidationTesting();
}
```

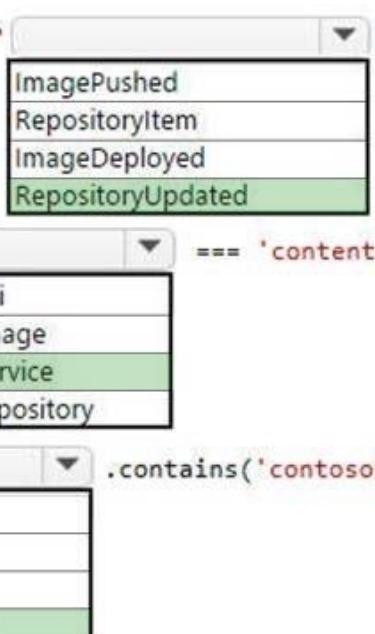
The screenshot shows a code editor with three dropdown menus:

- The first dropdown menu contains: ImagePushed, RepositoryItem, ImageDeployed, and RepositoryUpdated.
- The second dropdown menu contains: aci, image, service, and repository.
- The third dropdown menu contains: topic, service, repository, and imageCollection.

Answer:

Answer Area

```
var event = getEvent();
if (event.eventType === 'RepositoryUpdated' && event.data.target.service === 'contentanalysisservice' && event.imageCollection.contains('contosoimages')) {
    startValidationTesting();
}
```

**Explanation:**

Box 1: RepositoryUpdated

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Box 2: service

Box 3: imageCollection

Reference:

<https://docs.microsoft.com/en-us/azure/devops/notifications/oob-supported-event-types>

QUESTION 218**Case Study 8 - Contoso, Ltd****Background****Overview**

You are a developer for Contoso, Ltd. The company has a social networking website that is developed as a Single Page Application (SPA). The main web application for the social networking website loads user uploaded content from blob storage.

You are developing a solution to monitor uploaded data for inappropriate content. The following process occurs when users upload content by using the SPA:

- Messages are sent to ContentUploadService.
- Content is processed by ContentAnalysisService.
- After processing is complete, the content is posted to the social network or a rejection message is posted in its place.

The ContentAnalysisService is deployed with Azure Container Instances from a private Azure Container Registry named contosoimages.

The solution will use eight CPU cores.

Azure Active Directory

Contoso, Ltd. uses Azure Active Directory (Azure AD) for both internal and guest accounts.

Requirements

ContentAnalysisService

The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

You must create an Azure Function named CheckUserContent to perform the content checks.

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from internal Virtual Networks (VNets).
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso, Ltd. to review content, store cookies on user devices, and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code

ContentUploadService

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location: westus
CS04 name: contentUploadService
CS05 properties:
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpu: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22       protocol: TCP
CS23
CS24
CS25 networkProfile:
CS26
id: /subscriptions/98...19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

ApplicationManifest

```
AM01 {  
AM02     "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",  
AM03     "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",  
AM04  
AM05     "createdDateTime" : "2019-12-24T06:01:44Z",  
AM06     "logoUrl" : null,  
AM07     "logoutUrl" : null,  
AM08     "name" : "ContentAnalysisService",  
AM09  
AM10  
AM11     "orgRestrictions" : [],  
AM12     "parentalControlSettings" : {  
AM13         "countriesBlockedForMinors" : [],  
AM14         "legalAgeGroupRule" : "Allow"  
AM15     },  
AM16     "passwordCredentials" : []  
AM17 }
```

Drag and Drop Question

You need to add YAML markup at line CS17 to ensure that the ContentUploadService can access Azure Storage access keys.

How should you complete the YAML markup? To answer, drag the appropriate YAML segments to the correct locations. Each YAML 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.

| YAML segments | Answer Area |
|----------------------|--|
| secret | YAML segment : - mountPath: /mnt/secrets name: accesskey |
| envVar | YAML segment : - name: accesskey |
| secretValues | YAML segment : key: TXkgZmlyc3Qgc2VjcmV0IEZPTwo= |
| volumes | |
| volumeMounts | |
| environmentVariables | |

Answer:

YAML segments

envVar

secretValues

volumeMounts

Answer Area

```
environmentVariables :  
  - mountPath: /mnt/secrets  
    name: accesskey  
volumes :  
  - name: accesskey  
secret :  
  key: TXkgZmlyc3Qgc2VjcmV0IEZPTwo=
```

Explanation:

Box 1: volumeMounts

Example:

volumeMounts:

- mountPath: /mnt/secrets

name: secretvolume1

volumes:

- name: secretvolume1

secret:

mysecret1: TXkgZmlyc3Qgc2VjcmV0IEZPTwo=

Box 2: volumes

Box 3: secret

Reference:

<https://docs.microsoft.com/en-us/azure/container-instances/container-instances-volume-secret>**QUESTION 219****Case Study 8 - Contoso, Ltd****Background****Overview**

You are a developer for Contoso, Ltd. The company has a social networking website that is developed as a Single Page Application (SPA). The main web application for the social networking website loads user uploaded content from blob storage.

You are developing a solution to monitor uploaded data for inappropriate content. The following process occurs when users upload content by using the SPA:

- Messages are sent to ContentUploadService.
- Content is processed by ContentAnalysisService.
- After processing is complete, the content is posted to the social network or a rejection message is posted in its place.

The ContentAnalysisService is deployed with Azure Container Instances from a private Azure Container Registry named contosoimages.

The solution will use eight CPU cores.

Azure Active Directory

Contoso, Ltd. uses Azure Active Directory (Azure AD) for both internal and guest accounts.

Requirements

ContentAnalysisService

The company's data science group built ContentAnalysisService which accepts user generated content as a string and returns a probable value for inappropriate content. Any values over a specific threshold must be reviewed by an employee of Contoso, Ltd.

You must create an Azure Function named CheckUserContent to perform the content checks.

Costs

You must minimize costs for all Azure services.

Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role. All completed reviews must include the reviewer's email address for auditing purposes.

High availability

All services must run in multiple regions. The failure of any service in a region must not impact overall application availability.

Monitoring

An alert must be raised if the ContentUploadService uses more than 80 percent of available CPU cores.

Security

You have the following security requirements:

- Any web service accessible over the Internet must be protected from cross site scripting attacks.
- All websites and services must use SSL from a valid root certificate authority.
- Azure Storage access keys must only be stored in memory and must be available only to the service.
- All Internal services must only be accessible from internal Virtual Networks (VNets).
- All parts of the system must support inbound and outbound traffic restrictions.
- All service calls must be authenticated by using Azure AD.

User agreements

When a user submits content, they must agree to a user agreement. The agreement allows employees of Contoso, Ltd. to review content, store cookies on user devices, and track user's IP addresses.

Information regarding agreements is used by multiple divisions within Contoso, Ltd.

User responses must not be lost and must be available to all parties regardless of individual service uptime. The volume of agreements is expected to be in the millions per hour.

Validation testing

When a new version of the ContentAnalysisService is available the previous seven days of content must be processed with the new version to verify that the new version does not significantly deviate from the old version.

Issues

Users of the ContentUploadService report that they occasionally see HTTP 502 responses on specific pages.

Code**ContentUploadService**

```
CS01 apiVersion: '2018-10-01'
CS02 type: Microsoft.ContainerInstance/containerGroups
CS03 location: westus
CS04 name: contentUploadService
CS05 properties:
CS06   containers:
CS07     - name: service
CS08       properties:
CS09         image: contoso/contentUploadService:latest
CS10       ports:
CS11         - port: 80
CS12           protocol: TCP
CS13       resources:
CS14         requests:
CS15           cpu: 1.0
CS16           memoryInGB: 1.5
CS17
CS18 ipAddress:
CS19   ip: 10.23.121.112
CS20   ports:
CS21     - port: 80
CS22     protocol: TCP
CS23
CS24
CS25 networkProfile:
CS26
id: /subscriptions/98...19/resourceGroups/container/providers/Microsoft.Network/networkProfiles/subnet
```

ApplicationManifest

```
AM01 {
AM02   "id" : "2b079f03-9b06-2d44-98bb-e9182901fcb6",
AM03   "appId" : "7118a7f0-b5c2-4c9d-833c-3d711396fe65",
AM04
AM05   "createdDateTime" : "2019-12-24T06:01:44Z",
AM06   "logoUrl" : null,
AM07   "logoutUrl" : null,
AM08   "name" : "ContentAnalysisService",
AM09
AM10
AM11   "orgRestrictions" : [],
AM12   "parentalControlSettings" : {
AM13     "countriesBlockedForMinors" : [],
AM14     "legalAgeGroupRule" : "Allow"
AM15   },
AM16   "passwordCredentials" : []
AM17 }
```

Hotspot Question

You need to add code at line AM10 of the application manifest to ensure that the requirement for manually reviewing content can be met.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

"optionalClaims": [
 "
 acct
 platt
 sid
 tenant_ctry
 ",
 "
 sid
 upn
 email
 enfpolids
 "],

Answer:

Answer Area

```
"optionalClaims": [
    "acct",
    "platt",
    "sid", sid
    "tenant_ctry"
],
    "sid",
    "upn",
    "email", email
    "enfpolids"
],
```

Explanation:

Box 1: sid

Sid: Session ID, used for per-session user sign-out. Personal and Azure AD accounts.

Scenario: Manual review

To review content, the user must authenticate to the website portion of the ContentAnalysisService using their Azure AD credentials. The website is built using React and all pages and API endpoints require authentication. In order to review content a user must be part of a ContentReviewer role.

Box 2: email

Scenario: All completed reviews must include the reviewer's email address for auditing purposes.

QUESTION 220

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 develop a software as a service (SaaS) offering to manage photographs. Users upload photos to a web service which then stores the photos in Azure Storage Blob storage. The storage account type is General-purpose V2.

When photos are uploaded, they must be processed to produce and save a mobile-friendly

version of the image. The process to produce a mobile-friendly version of the image must start in less than one minute.

You need to design the process that starts the photo processing.

Solution: Create an Azure Function app that uses the Consumption hosting model and that is triggered from the blob upload.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

In the Consumption hosting plan, resources are added dynamically as required by your functions.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-blob-triggered-function>

QUESTION 221

You develop Azure solutions. A .NET application needs to receive a message each time an Azure virtual machine finishes processing data. The messages must NOT persist after being processed by the receiving application.

You need to implement the .NET object that will receive the messages.

Which object should you use?

- A. QueueClient
- B. SubscriptionClient
- C. TopicClient
- D. CloudQueueClient

Answer: D

Explanation:

A queue allows processing of a message by a single consumer. Need a CloudQueueClient to access the Azure VM.

Incorrect Answers:

B, C: In contrast to queues, topics and subscriptions provide a one-to-many form of communication in a publish and subscribe pattern. It's useful for scaling to large numbers of recipients.

Reference:

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-queues-topics-subscriptions>

QUESTION 222

You are developing an Azure function that connects to an Azure SQL Database instance. The function is triggered by an Azure Storage queue.

You receive reports of numerous System.InvalidOperationExceptions with the following message:

"Timeout expired. The timeout period elapsed prior to obtaining a connection from the pool. This

may have occurred because all pooled connections were in use and max pool size was reached."

You need to prevent the exception.

What should you do?

- A. In the host.json file, decrease the value of the batchSize option
- B. Convert the trigger to Azure Event Hub
- C. Convert the Azure Function to the Premium plan
- D. In the function.json file, change the value of the type option to queueScaling

Answer: C

Explanation:

With the Premium plan the max outbound connections per instance is unbounded compared to the 600 active (1200 total) in a Consumption plan.

Note: The number of available connections is limited partly because a function app runs in a sandbox environment. One of the restrictions that the sandbox imposes on your code is a limit on the number of outbound connections, which is currently 600 active (1,200 total) connections per instance. When you reach this limit, the functions runtime writes the following message to the logs: Host thresholds exceeded: Connections.

Reference:

<https://docs.microsoft.com/en-us/azure/azure-functions/manage-connections>

<https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale#service-limits>

QUESTION 223

Hotspot Question

You are developing an application that needs access to an Azure virtual machine (VM).

The access lifecycle for the application must be associated with the VM service instance.

You need to enable managed identity for the VM.

How should you complete the PowerShell segment? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

\$vm = Get-AzVM -ResourceGroupName "ContosoRG" -Name "ContosoVM"

Update-AzVM -ResourceGroupName "ContosoRG" -VM \$vm

| | |
|------------------|---|
| -AssignIdentity: | ▼ |
| -IdentityId: | ▼ |
| \$SystemAssigned | |
| \$UserAssigned | |

Answer:

Answer Area

```
$vm = Get-AzVM -ResourceGroupName "ContosoRG" -Name "ContosoVM"
```

```
Update-AzVM -ResourceGroupName "ContosoRG" -VM $vm
```

| | |
|------------------------------------|---|
| -AssignIdentity: | ▼ |
| \$SystemAssigned \$UserAssigned | ▼ |

Explanation:

Box 1: -IdentityType

Enable system-assigned managed identity on an existing Azure VM:

To enable a system-assigned managed identity, use the -IdentityType switch on the Update-AzVM cmdlet (see below).

Box 2: \$SystemAssigned

```
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM Update-AzVM -  
ResourceGroupName myResourceGroup -VM $vm -IdentityType SystemAssigned
```

Reference:

<https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/configure-powershell-windows-vm>

QUESTION 224

Drag and Drop Question

You are maintaining an existing application that uses an Azure Blob GPv1 Premium storage account. Data older than three months is rarely used.

Data newer than three months must be available immediately. Data older than a year must be saved but does not need to be available immediately.

You need to configure the account to support a lifecycle management rule that moves blob data to archive storage for data not modified in the last year.

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.

Actions**Answer Area**

Upgrade the storage account to GPv2

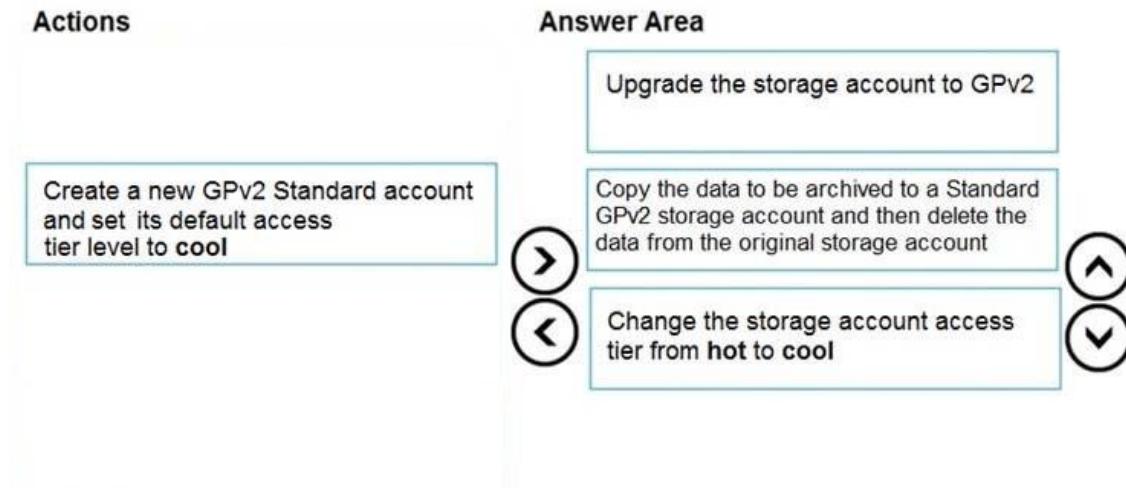


Create a new GPv2 Standard account and set its default access tier level to cool



Change the storage account access tier from hot to cool

Copy the data to be archived to a Standard GPv2 storage account and then delete the data from the original storage account

Answer:**Explanation:**

Step 1: Upgrade the storage account to GPv2

Object storage data tiering between hot, cool, and archive is supported in Blob Storage and General Purpose v2 (GPv2) accounts. General Purpose v1 (GPv1) accounts don't support tiering. You can easily convert your existing GPv1 or Blob Storage accounts to GPv2 accounts through the Azure portal.

Step 2: Copy the data to be archived to a Standard GPv2 storage account and then delete the data from the original storage account

Step 3: Change the storage account access tier from hot to cool

Note: Hot - Optimized for storing data that is accessed frequently. Cool - Optimized for storing data that is infrequently accessed and stored for at least 30 days. Archive - Optimized for storing data that is rarely accessed and stored for at least 180 days with flexible latency requirements, on the order of hours.

Only the hot and cool access tiers can be set at the account level. The archive access tier can only be set at the blob level.

Reference:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers>

QUESTION 225

Hotspot Question

You plan to deploy a new application to a Linux virtual machine (VM) that is hosted in Azure.

The entire VM must be secured at rest by using industry-standard encryption technology to address organizational security and compliance requirements.

You need to configure Azure Disk Encryption for the VM.

How should you complete the Azure CLI commands? To answer, select the appropriate options

in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
az provider register -n Microsoft.KeyVault  
resourcegroup="myResourceGroup"  
az group create --name $resourcegroup --location westus  
keyvault_name=myvaultname$RANDOM  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --name $keyvault_name \  
  --resource-group $resourcegroup \  
  --location eastus \  
  --enabled-for-disk-encryption True  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --vault-name $keyvault_name \  
  --name Name1 \  
  --protection software  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --resource-group $resourcegroup \  
  --name Name2 \  
  --image Canonical:UbuntuServer:16.04-LTS:latest \  
  --admin-username azureuser \  
  --generate-ssh-keys \  
  --data-disk-sizes-gb 5  
  
az [▼] enable\  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --resource-group $resourcegroup \  
  --name Name2 \  
  --disk-encryption-keyvault $keyvault_name \  
  --key-encryption-key Name1 \  
  --volume-type [▼]  
    all  
    data  
    os
```

Answer:**Answer Area**

```
az provider register -n Microsoft.KeyVault  
resourcegroup="myResourceGroup"  
az group create --name $resourcegroup --location westus  
keyvault_name=myvaultname$RANDOM  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --name $keyvault_name \  
  --resource-group $resourcegroup \  
  --location eastus \  
  --enabled-for-disk-encryption True  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --vault-name $keyvault_name \  
  --name Name1 \  
  --protection software  
  
az [▼] create \  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --resource-group $resourcegroup \  
  --name Name2 \  
  --image Canonical:UbuntuServer:16.04-LTS:latest \  
  --admin-username azureuser \  
  --generate-ssh-keys \  
  --data-disk-sizes-gb 5  
  
az [▼] enable\  
  vm  
  keyvault  
  keyvault key  
  vm encryption  
  
  --resource-group $resourcegroup \  
  --name Name2 \  
  --disk-encryption-keyvault $keyvault_name \  
  --key-encryption-key Name1 \  
  --volume-type [▼]  
    all  
    data  
    os
```

Explanation:

Box 1: keyvault

Create an Azure Key Vault with az keyvault create and enable the Key Vault for use with disk encryption. Specify a unique Key Vault name for keyvault_name as follows:

```
keyvault_name=myvaultname$RANDOM  
az keyvault create \  
--name $keyvault_name \  
--resource-group $resourcegroup \  
--location eastus \  
--enabled-for-disk-encryption True
```

Box 2: keyvault key

The Azure platform needs to be granted access to request the cryptographic keys when the VM boots to decrypt the virtual disks. Create a cryptographic key in your Key Vault with az keyvault key create. The following example creates a key named myKey:

```
az keyvault key create \  
--vault-name $keyvault_name \  
--name myKey \  
--protection software
```

Box 3: vm

Create a VM with az vm create. Only certain marketplace images support disk encryption. The following example creates a VM named myVM using an Ubuntu 16.04 LTS image:

```
az vm create \  
--resource-group $resourcegroup \  
--name myVM \  
--image Canonical:UbuntuServer:16.04-LTS:latest \  
--admin-username azureuser \  
--generate-ssh-keys \  
--volume-type all
```

Box 4: vm encryption

Encrypt your VM with az vm encryption enable:

```
az vm encryption enable \  
--resource-group $resourcegroup \  
--name myVM \  
--disk-encryption-keyvault $keyvault_name \  
--key-encryption-key myKey \  
--volume-type all
```

Note: seems to an error in the question. Should have enable instead of create.

Box 5: all

Encrypt both data and operating system.

Reference:

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/disk-encryption-cli-quickstart>

About Lead2pass.com

Lead2pass.com was founded in 2006. We provide latest & high quality IT Certification Training Exam Questions, Study Guides, Practice Tests. Lead the way to help you pass any IT Certification exams, 100% Pass Guaranteed or Full Refund. Especially [Cisco](#), [Microsoft](#), [CompTIA](#), [Citrix](#), [EMC](#), [HP](#), [Oracle](#), [VMware](#), [Juniper](#), [Check Point](#), [LPI](#), [Nortel](#), [EXIN](#) and so on.

Our Slogan: First Test, First Pass.

Help you to pass any IT Certification exams at the first try.

You can reach us at any of the email addresses listed below.

Sales: sales@lead2pass.com

Support: support@lead2pass.com

Technical Assistance Center: technology@lead2pass.com

Any problems about IT certification or our products, you could rely upon us, we will give you satisfactory answers in 24 hours.

View list of all certification exams: <http://www.lead2pass.com/all-products.html>



Microsoft



ORACLE



CITRIX



JUNIPER
NETWORKS



EXIN

EMC²
where information lives[®]