AZ 204 Final exam 2

Done by Ahmed Fouad
Ahmed Fouad | LinkedIn

Nokhba Academy FB page

https://shorturl.at/UUDf2

1. You are working on a smart irrigation system that collects weather and soil moisture data from numerous sensors deployed in fields. The system processes this large dataset using Azure Data Factory and performs predictive analytics to optimize water usage. During peak seasons, the influx of data increases significantly, causing longer processing times. To overcome this challenge, you need to expand the available compute resources. To efficiently handle the increased processing workload during peak season, you decide to leverage Azure Batch to create a pool of compute nodes. Which of the following steps should you undertake to achieve this?

A. In the Azure portal, create a new Azure Batch account to manage the compute resources.

B. In a .NET application, utilize the method: BatchClient.PoolOperations.CreatePoolAsync() to create the necessary compute nodes.

C. In Python, implement the class: JobAddParameter to configure the Azure Batch pool settings.

D. In Python, implement the class: TaskAddParameter to define the tasks to be executed on the Azure Batch compute nodes.

Answer: B

Feedback (if correct):

The correct answer is B, as using the .NET SDK method allows for efficient management of compute resources and supports the simulation of navigation sets for the autonomous transportation system. Creating a Batch pool using BatchClient.PoolOperations.CreatePoolAsync() is the appropriate step for provisioning compute nodes on Azure Batch.

Feedback (if wrong):

A) Wrong: Creating a new Batch account is not necessary for provisioning compute nodes within an existing account.

C) Wrong: Configuring the pool settings alone does not create the necessary compute nodes.

D) Wrong: Defining tasks does not allocate resources for executing them.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Sub-Topic: Provision compute resources on Azure Batch

Difficulty Level: Intermediate

Bloom's Taxonomy: Application

2. You are tasked with configuring container logging for a web application named "CloudApp" hosted on Azure. The application is deployed in an App Service named "CloudAppService" within the resource group "CloudAppRG." The custom Docker container used by the application is named "CloudContainer." Your goal is to complete the Azure CLI script snippet provided below to configure container logging for "CloudAppService" and stream the container logs in real-time. The script includes placeholders marked as `slot1`, `slot2`, `slot3`, and `slot4` for you to fill in. Choose the correct completion for `slot1` to accomplish this task.

```bash
# Set variables
resourceGroup="CloudAppRG"
appName="CloudAppService"
containerName="CloudContainer"

# Configure container logging
az slot1 log config --name $appName --slot2 $resourceGroup \
  slot3 filesystem

# Stream container logs in real-time
az webapp log tail slot4 $appName --resource-group $resourceGroup \
  --container-url $containerName
```

A. `appservice`

B. `webapp`

C. `log`

D. `azure`

Answer: B

Feedback(if correct): The correct completion for `slot1` is the `az webapp container set` command. This command is used to configure container settings for a web app in Azure, including logging configurations. In this scenario, it configures container logging for the specified web app and slot.

The `az webapp container set` command is specifically designed for configuring container settings for Azure web apps, making it the appropriate choice for configuring container logging in this scenario.

Feedback(if wrong):

the `az container log` command is not suitable for configuring container logging settings for a web app. the `az webapp config container` command is not the appropriate command for configuring container logging settings. the `az webapp config log` command is not valid for configuring container logging.

The question falls under the Azure App Service domain. The Bloom's Taxonomy Level is Applying, and the Difficulty Level is Medium.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure compute solutions

Competency: Implement container-based solutions

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

3. You are tasked with configuring container logging for a web application named "CloudApp" hosted on Azure. The application is deployed in an App Service named "CloudAppService" within the resource group "CloudAppRG." The custom Docker container used by the application is named "CloudContainer." Your goal is to complete the Azure CLI script snippet provided below to configure container logging for "CloudAppService" and stream the container logs in real-time. The script includes placeholders marked as `slot1`, `slot2`, `slot3`, and `slot4` for you to fill in. Choose the correct completion for `slot2` to accomplish this task.

```bash

# Set variables

resourceGroup="CloudAppRG"

appName="CloudAppService"

containerName="CloudContainer"
```

```
# Configure container logging
az slot1 log config --name $appName --slot2 $resourceGroup \
  slot3 filesystem

# Stream container logs in real-time
az webapp log tail slot4 $appName --resource-group $resourceGroup \
  --container-url $containerName
```

A `--resource-group`

B `--webspace`

C `--environment`

D `--service-plan`

Answer: A

Feedback(if correct):-

The completed script should look like this:

```bash
# Set variables
resourceGroup="CloudAppRG"
appName="CloudAppService"
containerName="CloudContainer"

# Configure container logging
az webapp log config --name $appName --resource-group $resourceGroup \
  --docker-container-name $containerName --filesystem

# Stream container logs in real-time
```

```
az webapp log tail $appName --resource-group $resourceGroup \
  --container-url $containerName
```

The purpose of this script is to configure container logging for the "CloudAppService" App Service in the "CloudAppRG" resource group. By setting the logging to the filesystem for the specified Docker container, the script enables monitoring and collecting logs efficiently. Streaming the container logs in real-time offers immediate visibility into the application's runtime events and potential issues.

Feedback(if wrong):

`appservice`: This option is incorrect because the correct term to use when configuring container logging for an Azure Web App is `webapp`, not `appservice`. The Azure CLI command for configuring logging is `az webapp log config`.

`log`: While the term "log" is part of the correct Azure CLI command (`az webapp log config`), it alone is insufficient for configuring container logging. The complete and correct term is `webapp`, and the option lacks necessary parameters for configuration.

`azure`: This option is incorrect because "Azure" is not a valid term or parameter when configuring container logging for an Azure Web App. The correct term to use is `webapp`, and the command is `az webapp log config`.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure compute solutions

Competency: Implement container-based solutions

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

4. You are tasked with configuring container logging for a web application named "CloudApp" hosted on Azure. The application is deployed in an App Service named "CloudAppService" within the resource group "CloudAppRG." The custom Docker container used by the application is named "CloudContainer." Your goal is to complete the Azure CLI script snippet provided below to configure container logging for "CloudAppService" and stream the container logs in real-time. The script includes placeholders marked as `slot1`, `slot2`, `slot3`, and `slot4` for you to fill in. Choose the correct completion for `slot3` to accomplish this task.

```bash
# Set variables
resourceGroup="CloudAppRG"
```

```
appName="CloudAppService"

containerName="CloudContainer"


# Configure container logging

az slot1 log config --name $appName --slot2 $resourceGroup \

  slot3 <<your_choice_here>>


# Stream container logs in real-time

az webapp log tail slot4 $appName --resource-group $resourceGroup \

  --container-url $containerName
```

A. `az container`

B. `config`

C. `log`

D. `webapp`

Answer: C

Feedback(if correct): The correct option to complete `slot3` is `log` because the `az log` command is used to configure container logging for "CloudAppService."

Feedback(if wrong):

(`az container`) is incorrect because the `az log` command is used for configuring container logging, not `az container`.

(`config`) is incorrect because it is a placeholder for the `az log config` command, not just `config`.

(`webapp`) is incorrect because it is not part of the `az log` command, which is used for configuring container logging.

5. You are tasked with configuring container logging for a web application named "CloudApp" hosted on Azure. The application is deployed in an App Service named "CloudAppService" within the resource group "CloudAppRG." The custom Docker container used by the application is named "CloudContainer." Your goal is to complete the

Azure CLI script snippet provided below to configure container logging for "CloudAppService" and stream the container logs in real-time. The script includes placeholders marked as `slot1`, `slot2`, `slot3`, and `slot4` for you to fill in. Choose the correct completion for `slot4` to accomplish this task.

```bash
# Set variables
resourceGroup="CloudAppRG"
appName="CloudAppService"
containerName="CloudContainer"

# Configure container logging
az slot1 log config --name $appName --slot2 $resourceGroup \
  slot3 <<your_choice_here>>

# Stream container logs in real-time
az webapp log tail slot4 $appName --resource-group $resourceGroup \
  --container-url $containerName
```

A. `stream`

B. `azure`

C. `filesystem`

D. `container`

Answer: C

Feedback(if correct): `filesystem`, is the correct choice as it represents the correct parameter for configuring container logging using Azure CLI for App Service. The `filesystem` parameter specifies that the logs should be stored in the file system.

Feedback(if wrong):

`stream`, is incorrect. While streaming logs is part of the task, it is not the parameter used for configuring container logging in this context.

`azure`, is incorrect. This parameter is not used for configuring container logging in Azure CLI for App Service.

`container`, is incorrect. This parameter is not used for configuring container logging in Azure CLI for App Service.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure compute solutions

Competency: Implement container-based solutions

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

6. You are deploying a web app to Azure App Service on Linux. You have created an App Service plan and pushed a custom Docker image containing the web app to Azure Container Registry. You now need to access the console logs generated from inside the container in real-time. How should you complete the Azure CLI command?

A. `az webapp log config --name <app_name> --resource-group <resource_group> --docker-container-logging filesystem`

B. `az webapp log config --name <app_name> --resource-group <resource_group> --docker-container-logging off`

C. `az webapp log download --name <app_name> --resource-group <resource_group>`

D. `az webapp log tail --name <app_name> --resource-group <resource_group>`

Answer: A

Feedback(if correct):

The correct answer is A. This command configures logging for the web app running in a Docker container on Azure App Service to enable real-time console log access by setting the `--docker-container-logging` option to `filesystem`.

Feedback(if wrong):

For option B, setting the `--docker-container-logging` option to `off` would disable logging for the Docker container, which is not what is required in this scenario. Option C, `az webapp log download`, is used to download log history as a zip file, but it does not provide real-time access to console logs. Option D, `az webapp log tail`, does not support Docker container logging configuration, making it an incorrect choice in this scenario.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Developing Solutions using Azure Functions

Competencies: Implementing Azure Functions, Integrating Azure Functions with other Azure services

Difficulty Level: Intermediate

Selected Blooms Taxonomy: Application

7. Lous Real Estate Company, an international villa service provider, is expanding its business into flat rentals. To cater to this growth, Louis needs to implement Azure Search for listing the flats. You already created the index in Azure Search. Now, you need to import flat data into the Azure Search service using the Azure Search .NET SDK. Question: Which of the following methods are appropriate for importing flat data into the Azure Search service using the Azure Search .NET SDK? (Select ALL applicable choices)

A. Using the SearchServiceClient class to connect to the search index

B. Creating a DocumentClient instance and setting its Endpoint property to the search index URL

C. Creating a DataContainer holding the documents and instantiating a DataSource pointing to it

D. Building an IndexBatch object containing the documents and using the Documents.Index() method

Answers: C, D

Feedback(if correct):- Creating a DataContainer for documents and instantiating a DataSource pointing to it establishes a means to send data to Azure Search Building an IndexBatch object and using the Documents.Index() method allows batching documents for efficient indexing.

Feedback(if wrong):- SearchServiceClient is deprecated; hence it shouldn't be used anymore. There is no DocumentClient class in the Azure Search .NET SDK.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Importing data into Azure Search

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

8. You are developing a global e-commerce platform using Azure. The platform involves processing large volumes of transaction data and requires a highly scalable solution. The architecture includes Azure Functions for

microservices, Azure SQL Database for transaction storage, and Azure Storage for file storage. It must be resilient to failures and perform efficiently. How should you design the Azure infrastructure to meet these requirements?

A. Utilize Azure Virtual Machines for microservices, Azure SQL Database for transactions, and Azure Blob Storage for file storage.

B. Leverage Azure Kubernetes Service (AKS) for microservices, Azure Cosmos DB for transactions, and Azure Queue Storage for file storage.

C. Implement Azure Functions for microservices, Azure SQL Database for transactions, and Azure Blob Storage for file storage.

D. Use Azure Container Instances for microservices, Azure Table Storage for transactions, and Azure File Storage for file storage.

Answer: C

Feedback(if correct): Implement Azure Functions for microservices, Azure SQL Database for transactions, and Azure Blob Storage for file storage.

The combination of Azure Functions, Azure SQL Database, and Azure Blob Storage ensures a scalable, cost-effective, and resilient solution.

By choosing option C, you benefit from the inherently scalable nature of Azure Functions for hosting microservices, coupled with Azure SQL Database to handle structured transaction data and Azure Blob Storage to deal with unstructured file storage. Both Azure Functions and Azure SQL Database offer built-in redundancy, automatic failover capabilities, and geo-replication, ensuring high availability and resilience to failures. Moreover, Azure Blob Storage excels in durability, scalability, and cost-effectiveness for storing files. Overall, this combination meets the requirements of high scalability, failure resilience, and efficiency.

Feedback(if wrong):

A. Utilizing Azure Virtual Machines introduces operational overhead and compromises scalability. Azure SQL Database and Azure Blob Storage are still viable options, but the lack of native scaling abilities makes this less desirable compared to other managed services.

B. Azure Kubernetes Service (AKS) is typically used for container orchestration purposes and works well with stateless microservices. However, the use case for transaction storage implies statefulness, which conflicts with the ephemeral nature of containers in AKS. Azure Cosmos DB is a compelling NoSQL offering but introducing such drastic changes without justification could lead to higher costs and complexity without added benefits. Azure Queue Storage is meant for messaging scenarios, not file storage.

D. Azure Container Instances may appear attractive initially but suffer from limited scalability, slower cold starts, and potentially higher costs compared to Azure Functions. Similarly, Azure Table Storage falls short in supporting ACID

transactions, leading to challenges in preserving data integrity. Lastly, Azure File Storage, though reliable, comes at a premium price tag. These factors combined result in lower suitability for this specific scenario.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Azure Functions, Azure SQL Database, Azure Storage

Difficulty Level: Intermediate

Bloom's Taxonomy: Analysis

9. Your team is building a serverless application in Azure that analyzes audio files uploaded by customers. Upon receiving an audio file, the application must initiate transcription and perform sentiment analysis. Transcriptions and analyses should commence immediately after file upload completion. Given the large volume of anticipated daily submissions, efficiency and scalability are paramount concerns. To address these requirements, you contemplate designing a process centered around Azure Functions, activated via Azure Blob Storage events. You intend to utilize Cognitive Services Text Analytics API for sentiment analysis and Speech Services SDK for transcription tasks. Considering the described context, select the appropriate statement regarding this proposed solution:

Answer: A

Feedback(if correct):- Option A correctly identifies that using Azure Functions triggered by Blob Storage events is a suitable approach for initiating transcription and sentiment analysis tasks immediately after file upload completion. Azure Functions can efficiently handle this event-driven workload, ensuring scalability and responsiveness. Additionally, integrating with Cognitive Services Text Analytics API and Speech Services SDK within the Azure Functions allows for seamless execution of transcription and sentiment analysis tasks. Therefore, Option A aligns with the project requirements and addresses concerns regarding efficiency and scalability.

Feedback(if wrong):-

B. Regrettably, this proposal falls short as neither Cognitive Services nor Speech Services APIs integrate natively with Azure Functions, necessitating additional architectural layers for interaction mediation.

C. Despite harnessing Azure Functions' responsiveness to Blob Storage events, this design encounters difficulties in catering to the expected surge in submission rates, thus impacting overall system effectiveness negatively.

D. While utilizing Azure Functions for activating downstream processes sounds promising, picking General Purpose v1 Storage Account over V2 hinders event integration functionality, rendering this setup insufficient for our purposes.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Azure Compute Solutions: Azure Functions

Azure Services Integration: Cognitive Services Text Analytics API, Speech Services SDK

Difficulty Level: Expert

Bloom's Taxonomy Level: Application

10. You are developing a microservices-based solution on Azure. The solution comprises several Azure Functions that are responsible for processing different types of events. Each function must run independently, scale automatically, and handle varying workloads. Additionally, you need to optimize costs and ensure efficient resource utilization. How should you configure the Azure Functions to meet the specified requirements? Choose the best solution from the options below:

A. Implement a single Azure Functions Premium Plan with multiple function apps, each handling a specific event type.

B. Use a Consumption Plan for each Azure Function, allowing automatic scaling and billing based on actual consumption.

C. Provision an isolated App Service Environment (ASE) for the Azure Functions, ensuring network isolation and improved scale capabilities.

D. Deploy the Azure Functions in a Standard Plan with multiple instances for redundancy, manually adjusting the scale based on workload changes.

Answer: A

Feedback(if correct):

Choosing a single Premium Plan for multiple-function apps optimizes costs, ensures scalability, and aligns with the microservices architecture.

Feedback(if wrong):

Option B (Consumption Plan): While the Consumption Plan provides automatic scaling, it may lead to less predictable performance in certain scenarios.

Option C (ASE): ASE is typically used for different purposes, such as running Azure Web Apps in an isolated network. It might be an overkill for Azure Functions.

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Azure Functions, Scaling, Cost Optimization

Difficulty Level: Intermediate

Bloom's Taxonomy: Application

11. You are tasked with developing a cloud-native microservices architecture on Azure. The solution involves multiple microservices that communicate with each other through HTTP. Each microservice is responsible for a specific business capability. To ensure resilience and scalability, you decide to implement Azure Service Fabric. What are the key considerations when implementing Azure Service Fabric for the microservices architecture? Choose the best solution from the options below:

A. Implement a single Service Fabric cluster for all microservices to simplify management and reduce costs.

B. Utilize stateless services for microservices that don't require a durable state and stateful services for microservices that need to maintain state.

C. Use Azure Service Bus for communication between microservices, ensuring reliable and asynchronous communication.

D. Deploy microservices in Docker containers and orchestrate them using Kubernetes to leverage containerization benefits.

Answer: B, D

Feedback(if correct):

Option B: Stateless services are suitable for microservices without durable state, while stateful services are ideal for those needing to maintain state within the Service Fabric.

Option D: Deploying microservices in Docker containers and orchestrating them with Kubernetes allows for easy scaling, management, and deployment of individual microservices.

Feedback(if wrong):

Option A: A single Service Fabric cluster may lead to increased complexity and reduced flexibility in managing microservices independently.

Option C: Azure Service Bus may introduce additional latency and might not be the most suitable choice for communication within a Service Fabric environment.

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Azure Service Fabric, Microservices Communication, Container Orchestration

Difficulty Level: Intermediate

Bloom's Taxonomy: Synthesis

12. You are assigned the task of deploying an Azure Web App named `<WebAppName>` using an ARM template. The template includes configurations for HTTPS traffic. Your objective is to identify the correct configuration to enforce that the Azure Web App only accepts HTTPS traffic. Select the appropriate option for Slot1

```json
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "resources": [
    {
      "type": "Microsoft.Web/sites",
      "apiVersion": "2019-08-01",
      "name": "<WebAppName>",
      "location": "<Location>",
      "properties": {
        "siteConfig": {
          // Slot1 - Hidden: Complete the missing elements here for HTTPS-only configuration
          // Hidden Configuration
        }
      }
    }
  ]
}
```

A. `"httpTrafficOnly": false`

B. `"httpsOnly": true`

C. `"trafficEncryption": "https"`

D. `"enableHttpTraffic": false`

Answer: B

Feedback(if correct):

": true`, is the correct configuration to enforce that the Azure Web App only accepts HTTPS traffic.

Feedback(if wrong):

`"httpTrafficOnly": false` is not the correct setting to enforce HTTPS-only traffic. `"trafficEncryption": "https"` is not a valid configuration for enabling HTTPS-only traffic.`"enableHttpTraffic": false` is not the correct configuration for ensuring that only HTTPS traffic is accepted.

Question Topic: Configuring HTTPS-only traffic in Azure Web Apps using ARM templates

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure compute solutions

Competency: Implement ARM templates for Azure resource deployment

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

13. You are working on an ARM template to deploy an Azure Storage Account named `<StorageAccountName>`. The template includes configurations for defining firewall rules. Your goal is to identify the correct configuration to define a firewall rule allowing traffic from all IP addresses. Select the appropriate option for Slot2:

```json
{
```

```
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "resources": [
    {
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2019-06-01",
      "name": "<StorageAccountName>",
      "location": "<Location>",
      "properties": {
        // Slot2 - Hidden: Complete the missing elements here for firewall rule allowing all IP addresses
        // Hidden Configuration
      }
    }
  ]
}
```

A. `"firewallEnabled": true`

B. `"firewallRules": [ { "ruleName": "AllowAll", "startIpAddress": "0.0.0.0", "endIpAddress": "255.255.255.255" } ]`

C. `"allowAllIPs": true`

D. `"networkRuleSet": { "defaultAction": "Allow", "bypass": "None", "virtualNetworkRules": [] }`

Answer: D

Feedback(if correct):

`"networkRuleSet": { "defaultAction": "Allow", "bypass": "None", "virtualNetworkRules": [] }`, is the correct configuration to define a firewall rule allowing traffic from all IP addresses.

Feedback(if wrong):

`"firewallEnabled": true` is not the correct configuration for defining specific firewall rules.

`"firewallRules": [ { "ruleName": "AllowAll", "startIpAddress": "0.0.0.0", "endIpAddress": "255.255.255.255" } ]` is the correct configuration, but it's not the option with the correct answer in this context.`"allowAllIPs": true` is not a valid configuration for defining firewall rules in an Azure Storage Account ARM template.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure compute solutions

Competency: Implement ARM templates for Azure resource deployment

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

14. You're managing a web application hosted on Azure Web App service using a D1 plan. Your company requires the application to automatically scale when CPU load reaches 75% while minimizing costs. Given the limitations of the D1 plan, which two actions should you prioritize to achieve these goals? (Choose 2)

A. Upgrade the App Service Plan to Basic or higher to enable autoscaling.
B. Configure a scale condition based on CPU load.
C. Set up manual scaling rules to adjust instances as needed.
D. Rearchitect the application to reduce resource consumption.

Answer: A, B

Feedback(if correct): -

Upgrading the App Service Plan is essential to unlock autoscaling capabilities, which are not available in the D1 plan. Basic, Standard, and Premium plans support autoscaling. Configuring a scale condition ensures that scaling is triggered when the CPU load reaches 85%, optimizing resource usage and preventing bottlenecks.

Key takeaways:

- Understand the limitations of different App Service Plans regarding autoscaling.
- Prioritize upgrading to a plan that supports autoscaling when it's a critical requirement.
- Configure scale conditions to fine-tune scaling behavior and align it with specific CPU load thresholds.
- Consider manual scaling or application optimization as supplementary strategies if needed

Feedback(if wrong): Manual scaling requires human intervention and might not be as responsive as autoscaling. It could lead to performance issues or overspending if not managed proactively. Rearchitecting the application might reduce resource needs but could be a more time-intensive and costly solution compared to upgrading the App Service Plan.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure Compute Solutions

Competency: Implement autoscaling for Azure Web Apps

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

15. You're a developer-facing two common issues that outbreak your VMs: backups and performance. You need to demonstrate your understanding of appropriate tools and troubleshooting skills.

Challenge 1: Backup and Restore Blues

Your production VM recently crashed, and you need to restore it from a backup. Your colleague suggests using Azure Site Recovery. While appreciating their support, you know a different tool is the optimal choice.

Which Azure tool should you use for reliable and efficient VM backups and restores, ensuring data protection and swift recovery?

A. Azure Site Recovery

B. Azure Backup

C. Azure Data Box

D. Azure Migrate

Answer: B

Feedback(if correct):

Azure Backup is the recommended tool for reliable and efficient VM backups and restores, ensuring data protection and swift recovery.

Feedback(if wrong):

A. Azure Site Recovery - Although useful for disaster recovery, it's not the optimal choice for regular VM backups and restores.

C. Azure Data Box - Used for offline data transfer, not for VM backups and restores.

D. Azure Migrate - Primarily for workload migration, not for VM backups and restores.

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure Compute Solutions

Competency: Implement IaaS solutions, Implement solutions for backup and restore of virtual machines

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

16. You're a developer and are facing two common issues outbreak your VMs: backups and performance. You need to demonstrate your understanding of appropriate tools and troubleshooting skills.

Challenge 2: Performance Perplexity

Your development VM exhibits frequent lag and sluggishness. While your network seems stable, pinpointing the root cause feels like navigating a maze. Which Azure tools could you leverage to diagnose and troubleshoot the performance bottleneck impacting your development VM? Choose two that offer the most relevant insights.

A. Azure Traffic Manager

B. Azure Network Watcher

C. Azure Monitor

D. ExpressRoute

Answer: C, B

Feedback(if correct): Azure Network Watcher and Azure Monitor are the correct choices. Azure Network Watcher provides network performance monitoring and diagnostics, while Azure Monitor offers comprehensive monitoring and insights into VM performance metrics, helping identify and troubleshoot performance bottlenecks effectively.

Feedback(if wrong): Azure Traffic Manager is primarily used for traffic distribution across global Azure regions and does not provide insights into VM performance. ExpressRoute is a dedicated private connection to Azure data centers and does not offer performance troubleshooting capabilities for VMs.

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop Azure Compute Solutions

Competency: Implement IaaS solutions, implement solutions for backup and restore of virtual machines

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

17. You are Working with an Azure Storage Account featuring soft delete, an application submits a blog labeled 'sample_file.txt'. Two snapshots result (Snapshot X and Snapshot Y). System fault causes the application to

remove the blob and accompanying snapshots. Reflect on the chances of restoring the blob and its snapshots owing to the soft delete attribute. Can you restore the original blob 'sample_file.txt'?

A. Yes, soft delete holds onto recently purged materials

B. No, since the elapsed time exceeds retention limits

C. No, because the blob vanished irrecoverably

D. Yes, if the object still falls inside the grace period

Answer: A

Feedback(if correct): Soft delete indeed retains the recently purged blob and its snapshots for a certain amount of time, usually known as the retention period or grace period, allowing you to recover the deleted data.

Feedback(if wrong):

No, since the elapsed time surpasses retention limits. In this scenario, the problem statement doesn't indicate the length of time elapsed between deletion and the attempted recovery. Also, retention limits apply to the permanence of deleted data, not the restoration window.

No, because the blob vanished irretrievably. The presence of a soft delete attribute signifies that, except under exceptional situations, blobs and their snapshots stay recoverable following deletion. Irretrievability is often observed after the conclusion of the retention interval.

Yes, assuming the object exists within the grace period! The term "grace period" denotes leeway or forgiveness intervals. In the context of Azure Storage Accounts, the retention period is commonly utilized to denote the survival span of deleted data. Despite slight variations in wording, the notion of restoring the blob within a confined timeframe echoes the core idea of soft delete.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop for Azure storage

Competencies: Azure Storage: Understanding soft delete feature and its functionality, Difficulty Level: Intermediate

Bloom's Taxonomy Level: Evaluation

18. You are Working with an Azure Storage Account featuring soft delete, an application submits a blog labeled 'sample_file.txt'. Two snapshots result (Snapshot X and Snapshot Y). System fault causes the application to remove the blob and accompanying snapshots. Reflect on the chances of restoring the blob and its snapshots owing to the soft delete attribute. Is it feasible to restore Snapshot X attached to the blob?

A. Yes, benefitting from the protective shield offered by soft delete

B. No, unless recovery instructions are explicitly issued

C. Yes, as snapshots persist irrespective of removal

D. No, when Snapshot X transcends the allowed deletion window

Answer: C

Feedback(if correct): Soft delete in Azure Storage Accounts allows for the recovery of deleted blobs and their associated snapshots. Specifically, when a snapshot (Snapshot X, in this case) is removed, it stays accessible in a "deleted" state for a determined period. This state is governed by the retention policy you set up, enabling you to recover Snapshot X and the original blob using built-in Azure management tools or APIs. So, yes, it is indeed feasible to restore Snapshot X due to the protective shield offered by the soft delete attribute. Keep up the great work!

Feedback(if wrong):

Yes, benefitting from the protective shield offered by soft delete.

Soft delete mainly protects blobs, not necessarily their snapshots.

No, unless recovery instructions are explicitly issued.

Recovery instructions aren't strictly required to restore snapshots.

No, when Snapshot X transcends the allowed deletion window.

There is no concrete evidence suggesting a restriction on the deletion window.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop for Azure storage

Competencies: Azure Storage: Understanding soft delete feature and its functionality

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Evaluation

19. You are Working with an Azure Storage Account featuring soft delete, an application submits a blog labeled 'sample_file.txt'. Two snapshots result (Snapshot X and Snapshot Y). System fault causes the application to remove the blob and accompanying snapshots. Reflect on the chances of restoring the blob and its snapshots owing to the soft delete attribute. Can you salvage Snapshot Y bound to the blob?

A. Sure, granted Snapshot Y antedates the moment of final deletion

B. No, since the simultaneous extinction compromises recoverability

C. No, excluding cases wherein snapshots evade soft delete's grasp

D. Yes, abiding by the standard protocol of soft delete operation

Answer: A

Feedback(if correct): as long as Snapshot Y came into existence before the ultimate deletion point, you can successfully recover it. This highlights the advantage of soft delete, which grants users opportunities to resurrect data even when accidents happen. Well done!

Feedback(if wrong):

No, since the simultaneous extinction compromises recoverability.

The timing of deletion doesn't negatively impact the recoverability of Snapshot Y, as soft delete saves it for a set duration.

No, excluding cases wherein snapshots evade soft delete's grasp.

Soft delete covers all snapshots equally, leaving little room for escape.

Yes, abiding by the standard protocol of soft delete operation.

The soft delete protocol doesn't inherently determine whether Snapshot Y can be saved or not. Timing plays a bigger role in this situation.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop for Azure storage

Competencies: Azure Storage: Understanding the soft delete feature and its functionality

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Evaluation

20. You are developing a cloud-native application that requires consistent and reliable access to a distributed database. The application's workload fluctuates throughout the day, and you want to ensure optimal performance and cost-effectiveness. What should you consider when choosing the database deployment model in Azure?

A. Deploy the database to an Azure SQL Database managed instance for maximum control and customization.

B. Choose Azure Cosmos DB with multi-region writes for global distribution and automatic scaling based on demand.

C. Single-node virtual machine (VM) running a traditional relational database system for simplicity and cost-effectiveness.

D. Implement Azure Cache for Redis to store frequently accessed data, reducing the load on the database.

Answer: B

Feedback(if correct): Azure Cosmos DB with multi-region writes meets all the key requirements: consistent and reliable access via global distribution, optimal performance with elastic scaling, and cost-effectiveness for fluctuating workloads. It eliminates the need for manual scaling and adjusts resources automatically based on demand.

Feedback(if wrong):- (Azure SQL Database managed instance): Offers control and customization, but lacks automatic scaling and global distribution for fluctuating workloads, potentially leading to higher costs.

(Single-node VM): Lacks scalability and high availability, not suitable for consistent and reliable access in a cloud-native application.

(Azure Cache for Redis): Improves performance by caching frequently accessed data, but doesn't address the core requirement of a scalable and globally distributed database.

21. An e-commerce website uses an Azure storage account for product images. Soft delete is enabled on the account with no configured retention policy. A data migration error accidentally deletes a product image blob named "img.jpg" along with its two existing snapshots. The website administrators need to recover the blob as soon as possible. Within what timeframe can the administrators recover the deleted blob "img.jpg" and its two snapshots from soft delete?

A. Immediately

B. Within 1 hour

C. Within 7 days

D. Within 14 days

Answer: C.

Feedback(if correct):- When soft delete is enabled without a retention policy, deleted blobs and snapshots are retained for the default period of 7 days. After this period, they are permanently deleted and unrecoverable. Therefore, the website administrators have up to 7 days to restore the deleted blob and its snapshots from the Azure portal, PowerShell, CLI, or other supported tools.

Feedback(if wrong):- Immediately: While recovery can be initiated immediately, the actual restoration process might take some time depending on the blob size and storage account performance. Within 1 hour: The default retention period is 7 days, not 1 hour. Within 14 days: Some Azure services like Azure Backup have a default retention period of 14 days for soft-deleted data, but not Azure Blob Storage in this scenario.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop for Azure storage

Competencies: Blob storage

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Comprehension

22. As you develop an Azure Cosmos DB solution using the Azure Cosmos DB SQL API, your dataset comprises millions of documents, each potentially featuring hundreds of properties. However, these properties lack distinct values for effective partitioning. To ensure a balanced workload distribution across all partitions over time and meet your application's performance needs, you must choose a suitable partition key. Which options are appropriate choices for defining a partition key in this scenario? Select all that apply.

A. Utilizing a single property value that appears frequently in the documents.

B. Concatenating multiple property values with a random suffix appended.

C. Assigning a hash suffix to a property value.

D. Employing a value containing the collection name.

Answer: A. C

Feedback (if correct):

Utilizing a single property value that appears frequently in the documents (Option A) and assigning a hash suffix to a property value Assigning a hash suffix to a property value are appropriate choices for defining a partition key in this scenario. These options promote even distribution across partitions, facilitating parallel write operations and ensuring workload balance.

Feedback (if wrong):

Concatenating multiple property values with a random suffix appended and Employing a value containing the collection name are incorrect. Concatenating multiple property values with a random suffix appended may introduce uneven

distribution due to the random suffix, and Option D is not a recommended practice for an effective partition key in this scenario.

Skill: Azure Developer Certification (AZ-204)

Subskill: Develop for Azure storage.

Competency: Azure Cosmos DB Partitioning Strategy, Choosing Effective Partition Keys, Understanding the factors influencing partition key selection in Azure Cosmos DB.

Difficulty Level: Expert

Blooms Taxonomy Level: Analysis

23. You design an ASP.NET Core website that stores images in Azure Blob Storage. Authenticate users via Azure Active Directory (Azure AD). Implement role-based access control (RBAC) on Blob Storage containers and sync user permissions with the Azure AD Application. What should you do?

A. Configure user_impersonation and delegated permissions with Microsoft Graph and Application in Azure AD.

B. Assign rbacrole and ciétid for Azure Storage API and Authentication.

C. Allow Authentication, Authorization, user_impersonation, delegated permissions, and Microsoft Graph with User Read permission.

D. Connect user_impersonation, delegated permissions, and Microsoft Graph with Permission, Cié tenant id, and Authentication.

Answer: A

Feedback(if correct): Configuring "user_impersonation" and "delegated" permissions with Microsoft Graph for the Azure Storage API and Application in Azure AD allows the ASP.NET Core website to act on behalf of the user, implement RBAC on Blob Storage containers, and sync user permissions.

Feedback(if wrong):

terms like "rbacrole" and "ciétid," which are not recognized or relevant in the context of Azure AD authentication and RBAC for Azure Blob Storage. allowing a mix of terms, including "rbacrole," "user_impersonation," and "delegated permissions," which may lead to confusion and misconfiguration. The question specifically mentions syncing user permissions with Azure AD, which is best achieved through "user_impersonation" and "delegated" permissions. combines terms like "user_impersonation," "delegated permissions," and "Microsoft Graph" with unrelated terms such

as "Permission" and "Cié tenant id." These terms do not align with the standard Azure AD authentication and RBAC configuration.

Skills: Azure Developer Certification (AZ-204)

Subskills: Implement Azure security

Competencies: Azure Security Authentication, Role-Based Access Control (RBAC)

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

24. You are developing a web application that will be hosted on Azure Web App and will require users to sign in with their Azure Active Directory (Azure AD) credentials. The website has three permission levels: admin, normal, and reader. User permissions will depend on their Azure AD group membership.

Goal: Configure authorization for the Azure Web App to grant permission levels based on Azure AD group membership.

Statement:

You have created an Azure AD application for the web application and now need to update its manifest to support the desired authorization scheme.

Proposed Solution: Within the Azure AD application's manifest, set the `groupMembershipClaims` option to `"All"`.

Does this solution meet the stated goals?

A) Yes

B) No

Feedback (if correct): setting the `groupMembershipClaims` option to `"All"` is not sufficient to meet the stated goal. Further steps are required to successfully configure authorization based on Azure AD group membership.

Feedback(if wrong):- While setting the `groupMembershipClaims` option to `"All"` is a necessary step, it is not sufficient to meet the stated goal. Further steps are required to successfully configure authorization based on Azure AD group membership.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Configure authentication and authorization for Azure resources

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

25. Your company manages a customer relationship management (CRM) system built upon Microsoft Dynamics 365, which runs inside an Azure App Service environment. Access to the CRM system requires secure authentication through Azure Active Directory (Azure AD). Specifically, you need to grant three different user roles—Admin, Sales Agent, and Support Specialist—with varying degrees of functionality and data visibility. To maintain compliance and enhance auditing capabilities, logging mechanisms must be configured to capture user activities accurately. How should you proceed to enable Azure AD authentication and apply customized access levels based on predefined user roles within your Microsoft Dynamics 365 instance operating within an Azure App Service environment?

A. Activate anonymous access and embed user role validation checks in the application code.

B. Limit the authentication mechanism to Azure AD, transmit user role information as query string parameters, and enforce role mapping at the middleware layer.

C. Empower Azure AD authentication solely and modify the Azure AD application's manifest file to propagate pertinent group claims.

D. Configure Azure AD authentication for the App Service environment and build a custom policy to map user roles to predefined Microsoft Dynamics 365 roles.

Feedback (if correct):

Configure Azure AD authentication for the App Service environment and build a custom policy to map user roles to predefined Microsoft Dynamics 365 role. This method provides better security, easier maintenance, and improved compatibility with Microsoft Dynamics 365 features.

Feedback(if wrong):-

activating anonymous access and embedding user role validation checks in the application code. However, this approach is not recommended because it exposes sensitive data and processes to unauthorized users who haven't logged in via Azure AD. Enabling anonymous access increases the risk of malicious attacks and violates compliance regulations. limiting the authentication mechanism to Azure AD, transferring user role information as query string parameters, and performing role mapping at the middleware layer. Although this technique uses Azure AD authentication, sending user role info through URL parameters raises serious security risks due to possible interception or manipulation by attackers. Moreover, middleware-layer enforcement doesn't offer direct integration with Microsoft Dynamics 365 roles, leading to complex and error-prone manual maintenance efforts. empowering Azure AD authentication exclusively and tweaking the Azure AD application's manifest file to disseminate appropriate group claims. Even though this approach implements Azure AD authentication, manually editing the manifest introduces management overhead and poses synchronization difficulties between evolving Azure AD groups and changing Microsoft Dynamics 365 roles. Furthermore, this technique fails to demonstrate explicit mapping between Azure AD groups and Microsoft Dynamics 365 roles.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Configure authentication and authorization

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

26. Your company manages a customer relationship management (CRM) system built upon Microsoft Dynamics 365, which runs inside an Azure App Service environment. Access to the CRM system requires secure authentication through Azure Active Directory (Azure AD). Specifically, you need to grant three different user roles—Admin, Sales Agent, and Support Specialist—with varying degrees of functionality and data visibility. To maintain compliance and enhance auditing capabilities, logging mechanisms must be configured to capture user activities accurately. Once Azure AD authentication is enabled, how can you isolate sensitive internal resources tied to the CRM system while maintaining seamless connectivity amongst trusted entities?

A. By establishing Point-to-Site VPN tunnels connecting approved devices to the underlying virtual network infrastructure supporting the CRM system.

B. Through integrating Private Link technology, thereby exposing managed platform endpoints privately to selected consumers over standard HTTP(S) channels.

C. With employing Hybrid Connections aimed at bridging gaps between on-premises systems and cloud components utilized by the CRM solution.

D. By configuring a service endpoint within the Azure virtual network, restricting access to the CRM system solely to resources within that network or those connected via approved VPN or ExpressRoute gateways.

Answer : D

Feedback (if correct): By configuring a service endpoint within the Azure virtual network, access gets limited strictly to resources within network or those attached via certified VPN or ExpressRoute gateways.

Feedback(if wrong):-

putting together Point-to-Site VPN burrows hooking accepted gadgets to the backing virtual network foundation for the CRM structure. Whilst helpful for remote associations, this setup cannot independently separate delicate inner reserves from general virtual network traffic. blending Private Link innovation, thus divulging oversaw stage endpoints secretly to

handpicked clients through ordinary HTTP(S) channels. Despite being advantageous for shielding public exposure, this alternative still leaves interior resources open to every asset inside the virtual network.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Configure network security

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

27. You are developing a .NET Core model-view controller (MVC) application hosted on Azure for a financial system. The application involves multiple roles and access requirements. You develop the following code:

```csharp
services.AddAuthorization(options =>
{
   options.AddPolicy("FinanceRoles", policy =>
   {
      // Code segment needed to meet authorization requirements
   });
});
```

You define roles named "AccountAdmin" and "FinanceManager."

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

1. Allow users with the "AccountAdmin" and "FinanceManager" roles access to the FinancialController, regardless of their claims.

2. Limit access to the Transfer action of the controller to users with a "transactionEditor" claim who are also members of the "FinanceManager" role.

How should you complete the code? To answer, choose the appropriate code segments from the options below. Each code segment may be used once, more than once, or not at all. You may need to scroll to view content.

Select the correct combination of code segments to meet the authorization requirements.

A.

```csharp
policy.RequireRole("AccountAdmin", "FinanceManager");
policy.RequireAuthenticatedUser();
```

B.

```csharp
policy.RequireRole("FinanceManager");
policy.RequireClaim("transactionEditor");
```

C.

```csharp
policy.RequireRole("AccountAdmin");
```

D.

```csharp
policy.RequireRole("FinanceManager");
policy.RequireClaim("transactionEditor");
policy.RequireRole("AccountAdmin");
```

Answer: A, B.

Feedback (if correct):

```csharp
policy.RequireRole("AccountAdmin", "FinanceManager");
policy.RequireAuthenticatedUser();
```

meets the first authorization requirement by allowing users with the "AccountAdmin" and "FinanceManager" roles access to the FinancialController, regardless of their claims.

```csharp
policy.RequireRole("FinanceManager");
policy.RequireClaim("transactionEditor");
```

meets the second authorization requirement by limiting access to the Transfer action to users with a "transactionEditor" claim who are also members of the "FinanceManager" role.

Feedback (if wrong):

```csharp
policy.RequireRole("AccountAdmin");
```

only meets the first authorization requirement partially, as it only allows users with the "AccountAdmin" role access to the FinancialController, but not users with the "FinanceManager" role. D.

```csharp
policy.RequireRole("FinanceManager");
policy.RequireClaim("transactionEditor");
policy.RequireRole("AccountAdmin");
```

is not the most efficient solution as it includes redundant code.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Implement authentication and authorization

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

28. You are developing a multi-tiered Azure solution that includes an Azure App Service hosting a web application, an Azure SQL Database for data storage, and Azure Blob Storage for file storage. The web application needs to securely access both the SQL Database and Blob Storage without exposing sensitive credentials. How should you configure the Azure App Service and related Azure services to ensure secure access to both Azure SQL Database and Azure Blob Storage? Choose the best solution from the options below:

A. Use Managed Identity for the Azure App Service. Configure the SQL Database and Blob Storage to accept requests only from the Managed Identity.

B. Store the sensitive credentials in environment variables within the Azure App Service. Implement encryption for the environment variables to ensure secure storage.

C. Create Shared Access Signatures (SAS) for both Azure SQL Database and Blob Storage. Embed the SAS tokens in the web application's code for authentication.

D. Implement Azure Key Vault to store and retrieve sensitive credentials. Configure the Azure App Service to authenticate with Azure Key Vault for access to both SQL Database and Blob Storage.

Answer: D

Feedback(if correct):

Using Azure Key Vault allows secure storage and retrieval of sensitive credentials. Configuring the Azure App Service to authenticate with Azure Key Vault ensures secure access to both Azure SQL Database and Azure Blob Storage without exposing credentials.

Utilizing Azure Key Vault aligns with best practices for secure credential management and ensures a robust, centralized solution for sensitive data.

Feedback(if wrong):

(Managed Identity): While Managed Identity is a good practice, it doesn't directly address secure storage for credentials.

(Environment Variables): Storing sensitive credentials directly in environment variables may expose them, and encryption alone may not be sufficient.

(Shared Access Signatures): Embedding SAS tokens in code can be a security risk, especially for long-term access.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Secure access to Azure storage services and SQL databases

Bloom's Taxonomy Level: Application

Difficulty Level: Medium

29. our organization hosts a static marketing site on Azure Static Web Apps, featuring user registration forms that collect confidential contact information, which is encrypted directly on clients' browsers prior to transmission. Upon obtaining user consent, the data gets decrypted and moved to Cosmos DB for durable storage. Determine the suitable components needed to tackle this challenge, accounting for both frontend and backend aspects.

A) Deploy the static web assets to Azure Static Web Apps, apply JavaScript-based encryption/decryption libraries in HTML/CSS files, and utilize Azure Functions for backend APIs linking to Cosmos DB.

B) Preserve the static web assets on GitHub Pages, capitalize on browser-native crypto APIs for encryption/decryption intentions, and establish the backend architecture comprising Azure Kubernetes Service joined to Cosmos DB.

C) Distribute the static material with Azure Storage, capitalize on Azure SDK for encryption/decryption measures, and merge the frontend with Azure Functions communicating with Cosmos DB by means of HTTPS petitions.

D) Offer the static entities with Netlify, insert OpenSSL collection snippets intended for client-side encryption, and depend on Cloudflare employees for backend functions liaising with Cosmos DB.

Answer: A

Feedback(if correct):- Using Azure Static Web Apps provides an easy way to host static websites while embedding encryption/decryption libraries in client-side scripts ensures data protection. Azure Functions facilitates communication between the client and Cosmos DB, keeping everything consistent.

Feedback(if wrong):-

Store the static web assets on GitHub Pages, leverage browser-native crypto APIs for encryption/decryption purposes, and deploy the backend infrastructure consisting of Azure Kubernetes Service linked to Cosmos DB. This option is incorrect because Github Pages aren't primarily designed for hosting applications that require encryption and decryption, unlike Azure Static Web Apps. Moreover, using Github Pages implies missing out on Azure services integration, which makes managing the environment challenging. Furthermore, opting for Azure Kubernetes Service seems excessive for a static website and creates additional complexity.

Publish the static content using Azure Ad, and harness Azure SDK for encryption/decryption procedures is invalid since Azure AD serves identity and access management purposes rather than functioning as a content delivery network or hosting provider. Thus, publishing static content via Azure AD doesn't suit the purpose. Besides, while Azure SDK includes useful cryptographic packages, applying them for direct implementation in client-side scripts may not be feasible or secure.

Serve the static artifacts with Netlify, embed OpenSSL library snippets for client-side encryption, and rely on Cloudflare workers for backend operations interfacing with Cosmos DB. This option is erroneous mainly because mixed technological choices undermine the coherence of the solution. Both Netlify and Cloudflare serve overlapping roles, raising concerns about excess complexity. More importantly, opening SSL library snippets within client-side scripts poses potential risks, as exposing encryption keys publicly compromises security. Instead, safer approaches ought to protect credentials and restrict key exposure to trusted environments.

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Secure Data Handling and Encryption

Difficulty Level: Intermediate

Blooms Taxonomy Level: Application

30. You are tasked with developing a solution that utilizes an Azure SQL Database to store sensitive user information for a mobile application. To ensure the protection of sensitive data, you need to configure dynamic data masking to hide this information from developers who query the data for the mobile app. You are configuring dynamic data masking for an Azure SQL Database storing sensitive user information for a mobile app. Which three items must you identify when configuring dynamic data masking?

A) Column

B) Table

C) Trigger

D) Schema

Correct Answer: A, B, D

Feedback(if correct):-

When configuring dynamic data masking in Azure SQL Database, you must identify the following items:

- A) Column: Specify the columns containing sensitive data that need to be masked.

- B) Table: Identify the tables where the sensitive columns are located for applying dynamic data masking.

- D) Schema: Determine the schema where the tables containing sensitive data reside, as dynamic data masking can be applied at the schema level to mask multiple tables simultaneously.

These selections collectively form the solution for configuring dynamic data masking to protect sensitive information in the Azure SQL Database.

Feedback(if wrong):-

Option C) Trigger: This option is incorrect. Triggers are not directly related to dynamic data masking in Azure SQL Database. Triggers are database objects that are activated or executed automatically in response to specified events, such as insertions, updates, or deletions of data in a table. However, dynamic data masking is a feature used to hide sensitive data in query results, not to trigger actions based on data changes.

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Secure Data Handling and Encryption

Difficulty Level: Intermediate

Blooms Taxonomy Level: Application

31. You are tasked with enhancing the security of an HR intranet site that handles confidential staff information. The site uses Azure Active Directory (AAD) for authentication. To meet compliance standards, you need to enforce multi-factor authentication. Which fist Essential step should you take?

A. Set up Multi-Factor Authentication in Azure AD.

B. Establish Conditional Access Policies in Azure AD.

C. Designate Multi-Factor Authentication coverage for specific users or groups.

D Update Azure AD subscription tier to Premium edition.

Answer: A

Feedback(if correct):-

This step involves configuring and enabling Multi-Factor Authentication in Azure AD to add an extra layer of security during the authentication process. It helps ensure that users provide additional verification, such as a phone call, text message, or mobile app notification, in addition to their password.

Feedback(if wrong):-

B. Establish Conditional Access Policies in Azure AD: While Conditional Access Policies can be used to further enhance security by defining specific conditions and requirements for accessing resources, they are not an essential step for enforcing multi-factor authentication.

C. Designate Multi-Factor Authentication coverage for specific users or groups: While designating Multi-Factor Authentication coverage for specific users or groups can be useful for targeting specific individuals or groups, it is not an essential step for enforcing multi-factor authentication.

D. Update Azure AD subscription tier to Premium edition: Upgrading the Azure AD subscription tier to Premium edition is not a necessary step for enforcing multi-factor authentication. Multi-Factor Authentication can be implemented without upgrading to the Premium edition.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Configure authentication and authorization for Azure resources

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

32. You are tasked with securing an Azure API Management managed web service that connects to a back-end service implementing HTTP Strict Transport Security (HSTS). Each request to the back-end service must include a valid HTTP authorization header. Which policies can you configure in Azure API Management to meet this requirement? Choose the correct options.

A. Certificate Authentication

B. Token-based Authentication

C. OAuth Client Credential Grant

D. Digest Authentication

Answer: A

Feedback(if correct):

Certificate Authentication and OAuth Client Credential Grant, align with the implementation of security policies in Azure API Management, specifically for authenticating and authorizing client requests. Certificate Authentication ensures secure communication between the client and the API Management instance, while the OAuth Client Credential Grant allows clients to obtain access tokens for authenticating with the back-end service without requiring user credentials.

Feedback(if wrong):

Basic Authentication and Digest Authentication are not suitable for securing communication between the client and the API Management instance, especially when dealing with sensitive data and strict security requirements.

Token-based Authentication is a generic term and not a specific policy in Azure API Management. Additionally, it does not provide the same level of security and flexibility as Certificate Authentication or OAuth Client Credential Grant.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Implement Azure security

Competency: Azure Security: Authentication and authorization, Azure Active Directory

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

33. You are assigned with controlling Application Insights expenses for your Azure web app within a fixed budget, you're currently monitoring the app's performance metrics via Application Insights. What measures can you take to stay within the spending limit?

A. Enable ingestion sampling in the Azure portal.

B. Set a daily cap for the Application Insights instance.

C. Utilize adaptive sampling in the Azure portal.

D. Implement adaptive sampling with the Application Insights SDK.

Answer: D

Feedback(if correct): Implementing adaptive sampling using the Application Insights SDK allows the web app to automatically adjust telemetry collection based on traffic volume, effectively managing costs while still providing valuable insights into app performance.

Feedback(if wrong):

A. Incorrect. Ingestion sampling via the Azure portal reduces the amount of data stored in Application Insights but does not automatically adjust based on traffic volume like adaptive sampling.

B. Incorrect. Setting a daily cap for the Application Insights instance limits the amount of data stored but does not dynamically adjust sampling rates based on traffic volume.

C. Incorrect. Adaptive sampling in the Azure portal does not exist; it's only available through the Application Insights SDK and automatically adjusts telemetry collection based on traffic volume.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competencies: Monitoring, Optimization, and Troubleshooting

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

34. Within a mobile app environment operating on Azure, you plan to harness Azure Monitor's Application Insights through the Azure Mobile Apps SDK to diligently follow user interactions. Designate the MOST crucial data point to track, yielding substantial insights into user activities.

A. Dependency Call

B. Page View

C. Authentication

D. Custom Event

Answer: B

Feedback(if correct):

Option B, "Page View," is the most suitable choice as it provides essential insights into user interactions by tracking the navigation between different screens or pages within the mobile app. This data point allows developers to understand user behavior, preferences, and engagement patterns, forming the basis for improving the overall user experience.

Feedback(if wrong):

A. Dependency Call: Tracking dependency calls can be useful for monitoring external dependencies and service integrations within the app, but it may not provide direct insights into user interactions.

C. Authentication: While monitoring authentication events is important for security and user management purposes, it does not directly capture user interactions within the app.

D. Custom Event: Tracking custom events allows developers to capture specific user actions or behaviors tailored to the app's functionality. While valuable for certain use cases, it may not be as universally applicable or fundamental as tracking page views for overall user interaction analysis.

Skill: Aure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competency: Azure Monitor, Application Insights

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

35. You are developing an application that utilizes Azure Blob Storage to store critical data. The application regularly creates snapshots of the blobs to enable rollback to previous states. Soft delete is enabled on the storage account. After a series of operations, including updates and snapshot creation, an unexpected system error occurs, resulting in the deletion of the data blob and all snapshots. Which of the following accurately describes the restorability of the application data?

A) The data blob and all snapshots can be restored entirely from the soft delete feature.

B) Only the snapshots can be restored; the data blob cannot be recovered.

C) Both the data blob and all snapshots cannot be restored due to the system error.

D) The data blob can be restored, but the snapshots are irretrievable.

Answer: A

Feedback (if correct):

Option A is correct. Soft delete enables the recovery of deleted blobs and their snapshots, ensuring that both the data blob and all snapshots can be restored from the storage account.

Feedback (if wrong):

B) Only the snapshots can be restored; the data blob cannot be recovered.

This option is incorrect because soft delete enables the recovery of both blobs and snapshots. The soft-deleted data blob and its associated snapshots can be restored entirely from the soft delete feature.

C) Both the data blob and all snapshots cannot be restored due to the system error.

This option is incorrect because soft delete provides the capability to restore both the data blob and its snapshots. Even in the event of a system error or accidental deletion, the soft-deleted data blob and snapshots can be recovered.

D) The data blob can be restored, but the snapshots are irretrievable.

This option is incorrect because soft delete allows for the recovery of both the data blob and its snapshots. The soft-deleted data blob and all associated snapshots can be restored entirely from the soft delete feature.

Skill mapping:

Skill: Aure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competency: Collect telemetry data, Troubleshoot common issue

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

36. You are developing a cloud-native application that consists of multiple microservices deployed as Docker containers to Azure Kubernetes Service (AKS). Each microservice generates log data that you need to monitor for troubleshooting and performance analysis purposes. You want to set up logging for your AKS cluster to capture both application logs and system logs from the underlying infrastructure. You are tasked with configuring logging for your AKS cluster to capture both application logs and system logs. Which Azure CLI command should you use to accomplish this task?

A) az aks update --name MyAKSCluster --resource-group MyResourceGroup --enable-addons monitoring --workspace-resource-id /subscriptions/<subscription_id>/resourcegroups/<resource_group>/providers/microsoft.operationalinsights/workspaces/<workspace_name>

B) az aks enable-addons --name MyAKSCluster --resource-group MyResourceGroup --addons monitoring --workspace-resource-id /subscriptions/<subscription_id>/resourcegroups/<resource_group>/providers/microsoft.operationalinsights/workspaces/<workspace_name>

C) az aks logging enable --name MyAKSCluster --resource-group MyResourceGroup --workspace-resource-id /subscriptions/<subscription_id>/resourcegroups/<resource_group>/providers/microsoft.operationalinsights/workspaces/<workspace_name>

D) az aks monitor enable --name MyAKSCluster --resource-group MyResourceGroup --workspace-resource-id /subscriptions/<subscription_id>/resourcegroups/<resource_group>/providers/microsoft.operationalinsights/workspaces/<workspace_name>

Answer: C

Feedback (if-correct):

The chosen answer, C) az aks logging enable --name MyAKSCluster --resource-group MyResourceGroup --workspace-resource-id /subscriptions/<subscription_id>/resourcegroups/<resource_group>/providers/microsoft.operationalinsights/workspaces/<workspace_name>, is correct as it explicitly enables logging for the AKS cluster and establishes a connection to the OMS workspace, capturing both application and system logs.

Feedback (if-wrong):

For incorrect answers A, B, and D, the reason is that they do not enable logging explicitly for the AKS cluster. Instead, they focus on updating, enabling add-ons, or monitoring the cluster, which does not necessarily activate logging for both application and system logs.

Skill mapping:

Skill: Aure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competency: Azure Monitor, Log Analytics, and Application Insights

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

37. You are developing a cloud-based application that utilizes Azure Service Bus for message processing. The application needs to scale dynamically based on the number of messages in the Service Bus queue. Currently, there is a rule in place to scale up the application when the average queue length of unprocessed and valid messages exceeds 1000. Now, you need to configure a new rule that will continuously scale down the application as long as the scale-up condition is not met. How should you configure the scale rule? Choose the best option from the answer choices provided.

A. Set the criteria to "Active Message Count" and the operator to "Less than or equal to." Set the count to "1000" and the action to "Decrease count by 1."

B. Set the criteria to "Message Count" and the operator to "Less than or equal to." Set the count to "1000" and the action to "Decrease count by 1."

C. Set the criteria to "Total Maximum Average Count" and the operator to "Less than or equal to." Set the count to "1000" and the action to "Decrease count by 1."

D. Set the criteria to "Message Count" and the operator to "Greater than or equal to." Set the count to "1000" and the action to "Decrease count by 1."

Answer: B

Feedback(if correct): Option B is correct. Setting the criteria to "Message Count" and the operator to "Less than or equal to 1000" ensures that the application scales down when the number of messages in the queue falls below the specified threshold, aligning with the requirement to continuously scale down the application until the scale-up condition is met.

Feedback(if wrong): Option A uses "Active Message Count" instead of "Message Count," which may not accurately reflect the total number of messages in the queue. Option C uses "Total Maximum Average Count," which is not suitable for determining when to scale down the application. Option D uses "Greater than or equal to" as the operator, which would

trigger the scale-down action when the message count exceeds 1000, contrary to the requirement to scale down when it falls below that threshold.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competencies: Implementing and configuring monitoring solutions

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

38. You are developing a cloud-native application on Azure that requires background processing tasks to be executed regularly. Additionally, certain tasks need to be executed immediately upon the occurrence of specific events. You also need the ability to debug these background tasks remotely for troubleshooting purposes. You are tasked with recommending the appropriate WebJob type for different scenarios in your Azure application development. Which WebJob type should you recommend for each scenario?

Scenario 1:

You need to execute background processing tasks on all instances that the web app runs on. Additionally, you want the option to restrict the WebJob to a single instance if necessary. What is the recommended WebJob type:

Select the appropriate WebJob type for Scenario 1 from the options provided.

A. Continuous
B. Triggered
C. Incessant
D. Activated

Answer: A

Feedback(if correct):

The correct answer is A) Continuous. Continuous WebJobs are suitable for executing background processing tasks on all instances that the web app runs on, and they support remote debugging, fulfilling the requirements of the scenario.

Feedback(if wrong):

Option B) Triggered is not recommended because Triggered WebJobs are designed to execute tasks immediately upon the occurrence of specific events, rather than regularly or on all instances of the web app.

Option C) Incessant is not a valid WebJob type in Azure App Service.

Option D) Activated is not a valid WebJob type in Azure App Service.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Develop Azure compute solutions

Competencies: Azure App Service

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

39. You are developing a cloud-native application on Azure that requires background processing tasks to be executed regularly. Additionally, certain tasks need to be executed immediately upon the occurrence of specific events. You also need the ability to debug these background tasks remotely for troubleshooting purposes. You are tasked with recommending the appropriate WebJob type for different scenarios in your Azure application development. Which WebJob type should you recommend for each scenario?

Scenario 2:

You need to execute tasks immediately upon the occurrence of specific events, and these tasks should run on a single instance selected by Azure for load balancing.

Select the appropriate WebJob type for Scenario 2 from the options provided.

A. Continuous
B. Triggered
C. Incessant
D. Activated

Answer: B

Feedback(if correct):-

For Scenario 2, where tasks need to be executed immediately upon the occurrence of specific events and run on a single instance selected by Azure for load balancing, the recommended WebJob type is Triggered. Triggered WebJobs are designed to execute tasks in response to specific events and run on a single instance selected by Azure for load balancing. This aligns with the requirements of the scenario.

Feedback(if wrong):

Option A) Continuous is not recommended because Continuous WebJobs are designed to execute tasks regularly, not immediately upon the occurrence of specific events.

Option C) Incessant is not a valid WebJob type in Azure App Service.

Option D) Activated is not a valid WebJob type in Azure App Service.

These options are incorrect because they do not align with the requirements of the scenario, which specifies the need for executing tasks immediately upon the occurrence of specific events and running on a single instance selected by Azure for load balancing.

Skill Mapping:

Skills: Azure Developer Certification (AZ-204)

Subskills: Monitor, optimize, and troubleshoot Azure solutions

Competencies: Optimizing Azure Solutions, Troubleshooting Azure Solutions

Difficulty Level: Intermediate

Bloom's Taxonomy Level: Application

40. You are tasked with managing security policies for a solution involving various Azure services, including Azure Kubernetes Service (AKS), Azure Functions, Cosmos DB, and Event Hub. Your client requires strict control over access to the AKS cluster and its associated resources using Azure Policy. Specifically, they want to prevent unauthorized updates to specific Azure Function app settings and enforce naming conventions on Cosmos DB databases. Which approach should you take to meet these requirements?

A) Apply a 'Deny' effect to an Azure Policy definition governing undesired updates to Function app settings and enforce consistent naming conventions on Cosmos DB databases through Azure Policy assignment.

B) Utilize Azure Blueprints to create a predefined template with the desired policy restrictions, while granting the client permission to modify the AKS cluster scope.

C) Utilize the Azure Policy Add-on for Kubernetes to integrate Azure Policy with the AKS cluster's GateKeeper admission controller, and implement conditional tags for Cosmos DB databases and Function apps to align with the client's preferences.

D) Enable Azure Arc for Servers on the AKS cluster and assign custom roles to restrict modifications to Function app settings and Cosmos DB database management.

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Implement solutions that use Azure Queue Storage queues

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

41. As part of your responsibilities in developing a microservices-based architecture for a SaaS company, you are tasked with ensuring the security of the web services. Each service must securely connect to external APIs and Azure services while adhering to specific authentication requirements. After a security audit, it is discovered that certain web services lack proper authentication mechanisms, potentially allowing unauthorized access. Which Azure API Management policy should you apply to address this security vulnerability and enforce proper authentication?

A. enforce-ssl

B. validate-jwt

C. restrict-cors

D. rate-limit

Answer: D

Feedback(if correct):

The correct answer is Option D, "validate-jwt." This policy is used in Azure API Management to authenticate and validate OAuth tokens, ensuring that only authorized users can access the web services, aligning with the security requirements mentioned in the scenario.

Feedback(if wrong):

Option A, "rate-limit," is focused on controlling the number of requests and does not directly address authentication, making it irrelevant to the scenario.

Option B, "enforce-ssl," ensures secure connections but does not handle authentication concerns.

Option C, "jwt-bearer," involves JWT bearer authentication but may not specifically prevent unauthorized or anonymous access to the web services.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Configure authentication for APIs, Develop an App Service Logic App, Create a Logic App, Create a custom connector for Logic Apps

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

42. You are developing a hazard notification system that utilizes Azure Service Bus for publishing alarms and communicating with alarm controllers. Each transaction involving alarm signals needs to be audited, including details about the activated alarm type. Which two actions should you take to implement a reply trail auditing solution? Select the correct options:

A. Assign the value of the hazard message SessionID property to the SequenceNumber property.

B. Assign the value of the hazard message SequenceNumber property to the DeliveryCount property.

C. Assign the value of the hazard message MessageId property to the DeliveryCount property.

D. Assign the value of the hazard message SessionID property to the ReplyToSessionId property.

Feedback (if correct):

Correct! To implement a reply trail auditing solution, you should assign the value of the hazard message SessionID property to the ReplyToSessionId property, allowing you to track the session of the original message. Additionally, assigning the value of the hazard message SequenceNumber property to the DeliveryCount property helps in tracking the delivery count for each message, ensuring accurate auditing.

Feedback (if wrong):

Option A: Incorrect. Assigning the SessionID property to the SequenceNumber property does not align with the requirement for implementing a reply trail auditing solution.

Option B: Incorrect. The SequenceNumber property should not be assigned to the DeliveryCount property for auditing purposes.

Option C: Incorrect. Assigning the MessageId property to the DeliveryCount property does not contribute to implementing a reply trail auditing solution.

Option D: Correct! Assigning the SessionID property to the ReplyToSessionId property helps in tracking the session of the original message, facilitating reply trail auditing.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Implement solutions that use Azure Service Bus

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

43. You are developing an Azure solution to manage inventory data from multiple warehouses located across different regions. Each warehouse sends inventory updates every hour, with each update ranging from 1 to 5 megabytes (MB) in size. The inventory updates must be stored in Azure Blob storage, and each update must be associated with a unique warehouse identifier. New warehouses may be added in the future. You need to implement a solution to receive and store the inventory updates securely and efficiently. Which approach should you take?

A. Provision of an Azure Service Bus. Configure a topic to receive the inventory updates by using a correlation filter.
B. Create an Azure Event Grid subscription for each warehouse. Configure the subscription to route inventory updates to Azure Blob storage.
C. Implement Azure Logic Apps with a scheduled trigger to retrieve inventory updates from each warehouse. Use Azure Blob storage as the destination for storing the updates.
D. Deploy Azure Functions with a Blob storage trigger to listen for inventory updates. Configure the trigger to capture updates based on the warehouse identifier.

Answer: D

Feedback (if correct):

Azure Functions with a Blob storage trigger provide a serverless approach to handling inventory updates efficiently. By configuring the trigger to listen for updates based on the warehouse identifier, you can ensure that each update is associated with the correct warehouse. This solution offers scalability, cost-effectiveness, and the ability to handle future additions of warehouses seamlessly.

Feedback (if wrong):

Option A is incorrect because Azure Service Bus is not the ideal choice for receiving and storing large amounts of inventory data in Blob storage.

Option B is incorrect because Azure Event Grid is better suited for event-based routing and not necessarily for receiving and storing large data payloads like inventory updates.

Option C is incorrect because Azure Logic Apps are more suitable for orchestrating workflows and integrations, but they are not optimized for handling large-scale data ingestion tasks like storing inventory updates in Blob storage.

Skill Mapping:

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Implement solutions that use Azure Blob storage

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

44. You are developing a web application that needs to be registered with an active Azure Active Directory (Azure AD) tenant. To complete the registration, you need to perform multiple actions in a specific sequence. Choose the correct sequence of actions from the options below:

Please choose the correct sequence of actions by selecting the option that reflects the correct order.

A. (Step 1) Select "New registration" in App Registrations.

(Step 2) Enter the application name and specify the account type and redirect URL.

(Step 3) Select the Azure AD instance.

B. (Step 1) Select "New application" in Enterprise Applications.

(Step 2) Use an access token to access the secure resource.

(Step 3) Add a cryptographic key.

C. (Step 1) Select "Manifest" from the middle-tier service registration.

(Step 2) Create a new application and provide the name, account type, and redirect URL.

(Step 3) In Azure AD, select the Azure AD instance.

D. (Step 1) Select "New application" in Enterprise Applications.

(Step 2) Select "New registration" in App Registrations.

(Step 3) Use an access token to access the secure resource.

Answer: A

The correct sequence of actions to register a web application with an active Azure Active Directory (Azure AD) tenant is:

A. (Step 1) Select "New registration" in App Registrations.

(Step 2) Enter the application name and specify the account type and redirect URL.

(Step 3) Select the Azure AD instance.

Answer: A

Feedback(if correct):-

Sequence A describes the correct order of operations needed to register a new web application with Azure AD.

1. By selecting "New registration" in App Registrations, you initiate the registration process.

2. Providing essential details such as the application name, account type, and redirect URL falls under entering the application information.

3. Choosing the Azure AD instance helps to establish connections between the web application and other Azure AD features.

Feedback(if wrong):-

B. (Step 1) Select "New application" in Enterprise Applications.

(Step 2) Use an access token to access the secure resource.

(Step 3) Add a cryptographic key.

This sequence jumps straight into creating an enterprise application without actually registering the web application. Moreover, requesting an access token and generating keys come later in the development cycle.

C. (Step 1) Select "Manifest" from the middle-tier service registration.

(Step 2) Create a new application and provide the name, account type, and redirect URL.

(Step 3) In Azure AD, select the Azure AD instance.

There is no concept of middle-tier service registration in Azure AD. Besides, modifying the Manifest is typically done after initially setting up the web application.

D. (Step 1) Select "New application" in Enterprise Applications.

(Step 2) Select "New registration" in App Registrations.

(Step 3) Use an access token to access the secure resource.

Similar to option B, this skips crucial steps and goes directly to creating an enterprise application. Furthermore, selecting "New registration" twice makes little sense.

Remember that understanding the underlying concepts is important when choosing the correct sequence of actions. While memorizing individual steps plays a role, grasping how things fit together holistically leads to better retention and performance.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Active Directory (Azure AD) integration, Web application registration and configuration in Azure AD, Azure CLI commands for Azure App Service configuration, Understanding of Azure Blob Storage and its features

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

45. You are developing a cloud-native web application that needs to connect to Azure services and third-party services for various functionalities. To ensure seamless integration with Azure Active Directory (Azure AD), proper registration and configuration of the web application in Azure AD are required. Additionally, you need to utilize Azure CLI commands for configuring Azure App Service and understand how to interact with Azure Blob Storage. Which set of commands should you use to achieve the required integration and configuration from the following in the correct order?

1) az ad app create
2) az webapp config appsettings set
3) az storage blob upload-batch

4) az ad app update

Select correct orders from the following:

A. 1,2,3,4
B. 1,4,2,3
C. 1,3,2,4
D. 2,1,3,4

Answer: B. 1,4,2,3

Feedback(if correct):-

1. Register and create a new application in Azure AD for integration with Azure services and third-party services using the 'az ad app create' command.

2. Update the Azure AD application settings using the 'az ad app update' command to configure permissions and access rights.

3. Configure the Azure App Service application settings using the 'az webapp config appsettings set' command to provide necessary configurations for connecting to Azure services and third-party services.

4. Upload files to Azure Blob Storage using the 'az storage blob upload-batch' command to store and manage data required by the web application.

This order ensures proper registration and configuration of the web application in Azure AD, configuration of Azure App Service, and interaction with Azure Blob Storage, aligning with the requirements specified in the scenario.

Feedback(if wrong):-

Option A) 1,2,3,4: This order is incorrect.

- az ad app create should be the first command to register and create a new application in Azure Active Directory (Azure AD), so step 1 is correct.

- However, az webapp config appsettings set (step 2) is not directly related to Azure Blob Storage interaction or Azure AD integration. This command is used to configure application settings for an Azure App Service, not for interacting with Azure Blob Storage or registering/configuring an application in Azure AD.

- az storage blob upload-batch (step 3) is used to upload files to Azure Blob Storage, but it should come after configuring the Azure App Service application settings and registering the application in Azure AD.

- az ad app update (step 4) is used to update the Azure AD application settings, which should occur after registering the application (az ad app create) but before configuring the Azure App Service (az webapp config appsettings set) and interacting with Azure Blob Storage (az storage blob upload-batch).

Option B) 1,4,2,3: This order is incorrect.

- az ad app create (step 1) is correct for registering and creating a new application in Azure AD.

- az ad app update (step 2) should come after registering the application (az ad app create) but before configuring the Azure App Service (az webapp config appsettings set) and interacting with Azure Blob Storage (az storage blob upload-batch).

- az webapp config appsettings set (step 3) is not directly related to Azure Blob Storage interaction or Azure AD integration. This command is used to configure application settings for an Azure App Service, not for interacting with Azure Blob Storage or registering/configuring an application in Azure AD.

- az storage blob upload-batch (step 4) is used to upload files to Azure Blob Storage, but it should come after configuring the Azure App Service application settings and registering the application in Azure AD.

Option D) 2,1,3,4: This order is incorrect.

- az webapp config appsettings set (step 1) is not directly related to Azure Blob Storage interaction or Azure AD integration. This command is used to configure application settings for an Azure App Service, not for interacting with Azure Blob Storage or registering/configuring an application in Azure AD.

- az ad app create (step 2) should be the first command to register and create a new application in Azure Active Directory (Azure AD), so step 2 is incorrect.

- az storage blob upload-batch (step 3) is used to upload files to Azure Blob Storage, but it should come after configuring the Azure App Service application settings and registering the application in Azure AD.

- az ad app update (step 4) should come after registering the application (az ad app create) but before configuring the Azure App Service (az webapp config appsettings set) and interacting with Azure Blob Storage (az storage blob upload-batch).

Correct Order:

1. A) az ad app create

2. D) az ad app update

3. B) az webapp config appsettings set

4. C) az storage blob upload-batch

Explanation:

1. Use the 'az ad app create' command to register and create a new application in Azure AD for integrating with Azure services and third-party services.

2. Update the Azure AD application settings using the 'az ad app update' command to configure permissions and access rights.

3. Configure the Azure App Service application settings using the 'az webapp config appsettings set' command to provide necessary configurations for connecting to Azure services and third-party services.

4. Upload files to Azure Blob Storage using the 'az storage blob upload-batch' command to store and manage data required by the web application.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Active Directory (Azure AD) integration, Web application registration and configuration in Azure AD, Azure CLI commands for Azure App Service configuration, Understanding of Azure Blob Storage and its features

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

46. You are developing an application that requires connecting to a No-SQL globally-distributed database using the .NET API. You need 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);`

D) `new DocumentClient(EndpointUri, PrimaryKey);`

Answer: C

Feedback (if correct):

Option C is correct. The `CosmosClient` class is specifically designed for connecting to Azure Cosmos DB, allowing configuration and execution of database requests.

To connect to a No-SQL globally-distributed database using the .NET API, you should use the `CosmosClient` class. This class provides the necessary functionalities to configure and execute requests in the database. Example usage:

```csharp

// Create a new instance of the Cosmos Client
```

this.cosmosClient = new CosmosClient(EndpointUri, PrimaryKey);

await this.CreateDatabaseAsync();

```

Feedback (if wrong):

Options A, B, and D are incorrect. Option A refers to a `Container` object, which is used to work with specific containers within the Cosmos DB, not to establish the initial connection. Option B refers to a `Database` object, which does not exist in the context of Cosmos DB. Option D refers to a `DocumentClient` object, which was used in previous versions of the Cosmos DB SDK but has been superseded by the `CosmosClient` class. Therefore, Option C is the correct choice for connecting to the Cosmos DB in this scenario.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Storage, Blob storage, Table storage, Queue storage, File storage, Azure Services Integration, Azure Cosmos DB

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

47. You're developing a cloud-based application that requires interacting with Azure Storage and Azure Cosmos DB. Your task involves efficiently storing user data in Azure Table Storage and performing batch operations for optimal performance. Identify the missing line(s) of code to achieve this goal. Considerations: Ensure the solution maximizes efficiency and leverages Azure Table Storage's inherent capabilities for bulk operations, while also integrating seamlessly with Azure Cosmos DB. Complete the missing code in Slot1

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
    CloudConfigurationManager.GetSetting("StorageConnectionString"));

CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
```
```

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code

Slot1 Slot2 = new Slot3();

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));

CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code

Slot1 Slot2 = new Slot3();
```

A. TableBatchOperation
B. batchOperation
C. tableClient
D. CreateCloudTableClient

Answer: A

Feedback(if correct):-

To efficiently perform batch operations for storing user data in Azure Table Storage, you need to use the `TableBatchOperation` class, which allows you to combine multiple insert, update, delete, or merge operations into a single batch. This approach maximizes efficiency and leverages Azure Table Storage's capabilities for bulk operations.

A. `TableBatchOperation`

This line of code instantiates a new `TableBatchOperation` object, enabling you to perform batch operations efficiently. Therefore, the correct answer is:

Feedback(if wrong):-

B. batchOperation: This option is incorrect because "batchOperation" is not a valid class or method in the Azure Table Storage SDK. The correct class for performing batch operations is TableBatchOperation.

C. tableClient: This option is incorrect because the "tableClient" variable is already initialized earlier in the code and is used to create a reference to the Azure Table Storage client. It is not related to batch operations.

D. CreateCloudTableClient: This option is incorrect because "CreateCloudTableClient" is a method used to initialize the CloudTableClient object, which is responsible for creating, retrieving, and managing table references in Azure Table Storage. However, it is not directly related to performing batch operations.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Storage, Blob storage, Table storage, Queue storage, File storage, Azure Services Integration, Azure Cosmos DB

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

48. You're developing a cloud-based application that requires interacting with Azure Storage and Azure Cosmos DB. Your task involves efficiently storing user data in Azure Table Storage and performing batch operations for optimal performance. Identify the missing line(s) of code to achieve this goal. Considerations: Ensure the solution maximizes efficiency and leverages Azure Table Storage's inherent capabilities for bulk operations, while also integrating seamlessly with Azure Cosmos DB. Complete the missing code in Slot1

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));
```

```
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code
Slot1 Slot2 = new Slot3();
```

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));

CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code
Slot1 Slot2 = new Slot3();
```

A. TableBatchOperation
B. batchOperation
C. tableClient
D. CreateCloudTableClient

Answer: B

Feedback(if correct):-

`batchOperation` is a variable name representing an instance of the `TableBatchOperation` class. This class is used to batch multiple insert operations together, allowing for efficient storage of user data in Azure Table Storage and performing batch operations for optimal performance. By leveraging `batchOperation`, the code can efficiently insert large quantities of user data into Azure Table Storage concurrently, maximizing throughput and minimizing latency. Therefore, option B, `batchOperation`, is indeed the correct choice for Slot2.

Feedback(if wrong):-

A. TableBatchOperation: This option represents the class used to define a batch operation in Azure Table Storage. While it is related to performing batch operations, it is not used as a variable in the provided code snippet. Therefore, it cannot be directly assigned to Slot2 in the code.

C. tableClient: This option represents the client object used to interact with Azure Table Storage. However, it is not directly related to defining or performing batch operations for storing data in Azure Table Storage. It is used for creating, accessing, and managing tables within Azure Table Storage.

D. CreateCloudTableClient: This option represents a method used to create an instance of the `CloudTableClient` class, which is responsible for interacting with Azure Table Storage. Similar to option C, it is not directly related to defining or performing batch operations for storing data in Azure Table Storage. It focuses on the creation of the client object rather than the execution of batch operations.

Therefore, options A, C, and D are incorrect choices for Slot2.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Storage, Blob storage, Table storage, Queue storage, File storage, Azure Services Integration, Azure Cosmos DB

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

49. You're developing a cloud-based application that requires interacting with Azure Storage and Azure Cosmos DB. Your task involves efficiently storing user data in Azure Table Storage and performing batch operations for optimal performance. Identify the missing line(s) of code to achieve this goal. Considerations: Ensure the solution maximizes efficiency and leverages Azure Table Storage's inherent capabilities for bulk operations, while also integrating seamlessly with Azure Cosmos DB. Complete the missing code in Slot1

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));

CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code
Slot1 Slot2 = new Slot3();
```

Code Sample:

```csharp
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
   CloudConfigurationManager.GetSetting("StorageConnectionString"));

CloudTableClient tableClient = storageAccount.CreateCloudTableClient();

CloudTable table = tableClient.GetTableReference("users");

table.CreateIfNotExists();

// Missing code
```

Slot1 Slot2 = new Slot3();

A. TableBatchOperation
B. batchOperation
C. tableClient
D. CreateCloudTableClient

Answer: A

Feedback(if correct):-

The missing line(s) of code in Slot3 should aim to efficiently perform batch operations in Azure Table Storage for storing user data, ensuring optimal performance while also integrating seamlessly with Azure Cosmos DB.

Given these considerations, the correct option for Slot3 should align with these requirements.

Utilizing `TableBatchOperation` allows for batching multiple insert operations together, optimizing performance by minimizing the number of network calls and leveraging parallelism. This aligns with the requirement to efficiently store user data in Azure Table Storage using batch operations for optimal performance while integrating seamlessly with Azure Cosmos DB.

Feedback(if wrong):-

B. batchOperation: This option is not directly related to performing batch operations in Azure Table Storage. It is a generic variable name and does not represent a specific class or operation related to Azure Table Storage's batch functionality. Therefore, it is not the appropriate choice for Slot3.

C. tableClient: While the `tableClient` object is used to interact with Azure Table Storage, it is not directly related to performing batch operations for storing user data. It is primarily used for creating, accessing, and managing tables within Azure Table Storage. Therefore, it is not the correct choice for Slot3.

D. CreateCloudTableClient: This option represents a method used to create an instance of the `CloudTableClient` class, which is responsible for interacting with Azure Table Storage. However, it is not directly related to defining or performing batch operations for storing data in Azure Table Storage. It focuses on creating the client object rather than the execution of batch operations. Therefore, it is not the appropriate choice for Slot3.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Azure Storage, Blob storage, Table storage, Queue storage, File storage, Azure Services Integration, Azure Cosmos DB

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate

50. You are developing a mobile application and need to implement Application Insights instrumentation capabilities using the Azure Mobile Apps SDK to analyze user interactions effectively. Which three data values should you capture to enable the Usage Analytics feature of Application Insights? Choose the correct options from the following:

A. Trace

B. Session Id

C. User Id

D. Events

Feedback(if correct):

Options A, C, and D are correct. To implement the Usage Analytics feature of Application Insights, you need to capture traces (A), user IDs (C), and events (D) to analyze user interactions and application usage effectively.

Feedback(if wrong):

Option B is incorrect. Application Insights automatically manages the Session Id, so capturing it manually is not necessary for implementing the Usage Analytics feature. Instead, capturing traces, user IDs, and events provide the necessary data for usage analysis.

Skill Mapping :

Skill: Azure Developer Certification (AZ-204)

Subskill: Connect to and consume Azure services and third-party services

Competency: Monitoring, Optimization, and Troubleshooting: Application Insights.

Bloom's Taxonomy Level: Application

Difficulty Level: Intermediate