

Options

An Azure Key Vault

An Azure Storage account

Azure Active Directory (AD)
Identity Protection

An access policy

An Azure policy

A backup policy

Answer

An Azure Key Vault

An access policy

Options

Azure Cosmos DB
connector

SendGrid action

Consumption plan

Azure Event Hubs
binding

SendGrid binding

Answer

Consumption plan

SendGrid binding

Answer Area

Require MFA for Azure port...

Info

* Name
MFA required for Azure portal access

Assignments

Users and groups >
0 users selected

Cloud apps >
0 apps selected

Conditions >
0 conditions selected

Access controls

Grant >
0 controls selected

Session >
0 controls selected

Answer Area

```
az monitor metrics alert create -n myAlert -g myResourceGroup  
--scopes targetResourceId --condition "

|                        |                    |
|------------------------|--------------------|
| 5m                     | > 85"              |
| --window-size          | CPU Usage          |
| --evaluation-frequency | Percentage CPU     |
| --auto-mitigate        | avg Percentage CPU |


```

Create a DataSource instance and set its Container property to the DataContainer.

Create an IndexBatch that contains the documents which must be added.

Set the DataSources property of the SearchServiceClient.

Create a SearchIndexClient object to connect to the search index.

Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.

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

Create a SearchIndexClient object to connect to the search index.

Create an IndexBatch that contains the documents which must be added.

Call the Documents.Index method of the SearchIndexClient and pass the IndexBatch.

```
function.json
```

```
{
```

▼
“type”: “http”
“platform”: “gcm”
“datatype”: “stream”
“path”: “process.exe”

```
“direction”: “out”,
```

```
“name” : “result”
```

```
}
```

```
host.json
```

▼
“customHandler”: { “description”: {
“languageWorker”: { “path”: {
“extensions”: {“worker”: {
“extensionBundle”: {

```
“defaultExecutablePath”: “process.exe”  
},
```

▼
“enableForwardingHttpRequest”: true
“enableForwardingHttpRequest”: false

```
}
```

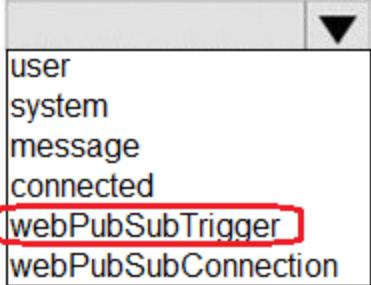
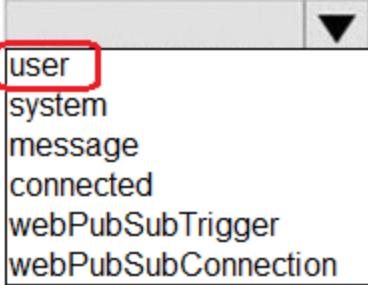
Answer Area

```
{  
  "routes": {  
    "headers": {},  
    "responseOverrides": {  
      "statusCodes": {  
        "400": {},  
        "401": {},  
        "403": {},  
        "404": {}  
      },  
      "redirect": "/auth/login/?post_login=true",  
      "statusCode": 302  
    }  
  }  
}
```

Answer Area

```
az ts create\nstorage account create\nstorage account update\nblueprint artifact template create\n\n--name templateStore \
--version "1.0" \
--resource-group templatesRG \
--location "eastus" \
\n\n--template-file \"\nmainTemplate.json\nlinkedTemplate1.json\nlinkedTemplate2.json\nmainTemplate.parameters.json\n\n\"C:\\templates\\\n--tags Dept=HumanResources Environment=Production
```

Answer Area

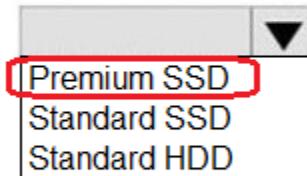
```
{  
  "bindings": [  
    {  
      "type": "",  
        "direction": "in",  
        "name": "data",  
        "eventName": "message",  
        "eventType": ""  
      }  
    ]  
  }  
}
```

Answer Area

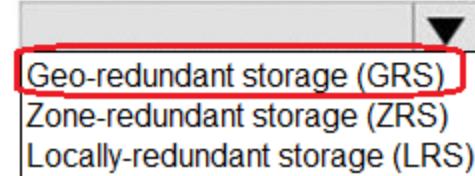
Requirement

Solution

Disk type



Redundancy



Configuration setting	Configuration value	
Shared lifecycle	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">Container group</div> Container image Service endpoint Resource group </div>	
Storage volume	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">Cloned Git repo</div> Azure file share Secret Empty directory Cloned Git repo </div>	
App service plan setting	Value	
Number of VM instances	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">2</div> 4 8 16 </div>	
Pricing tier	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid #ccc; padding-bottom: 5px; margin-bottom: 5px;">Isolated</div> Standard Premium Consumption </div>	
CRD types	Setting	CRD type
Secret	Azure Function code	Deployment
Deployment		
ScaledObject	Polling interval	ScaledObject
TriggerAuthentication	Azure Storage connection string	Secret

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

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

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

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

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

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

Property

Client certificate location

Value

```
HTTP request header  
Client cookie  
HTTP message body  
URL query string
```

Encoding type

```
HTML  
URL  
Unicode  
Base64
```

Azure CLI Commands

az group create

az group update

az webapp update

az webapp create

az appservice plan create

Azure CLI Commands

```
az webapp config container set
--docker-custom-image-name
$dockerHubContainerPath
--name $appName
--resource-group
fourthCoffeePublicWebResourceGroup
```

```
az webapp config hostname add
--webapp-name $appName
--resource-group
fourthCoffeePublicWebResourceGroup \
--hostname $fqdn
```

```
az webapp create
--name $appName
--plan AppServiceLinuxDockerPlan
--resource-group
fourthCoffeePublicWebResourceGroup
```

```
#/bin/bash
appName="FourthCoffeePublicWeb$random"
location="WestUS"
dockerHubContainerPath="FourthCoffee/publicweb:v1"
fqdn="http://www.fourthcoffee.com">www.fourthcoffee.com
```

Answer Area

az group create

az appservice plan create

az webapp create



Answer Area

```
#!/bin/bash
appName="FourthCoffeePublicWeb$random"
location="WestUS"
dockerHubContainerPath="FourthCoffee/publicweb:v1"
fqdn="http://www.fourthcoffee.com">www.fourthcoffee.com
```

```
az webapp create
--name $appName
--plan AppServiceLinuxDockerPlan
--resource-group
fourthCoffeePublicWebResourceGroup
```

```
az webapp config container set
--docker-custom-image-name
$dockerHubContainerPath
--name $appName
--resource-group
fourthCoffeePublicWebResourceGroup
```

```
az webapp config hostname add
--webapp-name $appName
--resource-group
fourthCoffeePublicWebResourceGroup \
--hostname $fqdn
```

Actions

Create a user-assigned managed identity for the application.

Create the Azure Functions app with a Premium plan type.

Create an access policy in Azure Key Vault for the application identity.

Create an SSL certification in Azure Key Vault for the application identity.

Create the Azure Functions app with an App Service plan type.

Create the Azure Functions app with a Consumption plan type.

Create a system-assigned managed identity for the application.

Answer Area

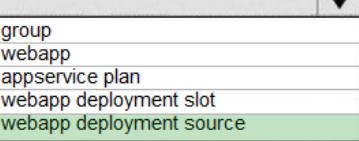
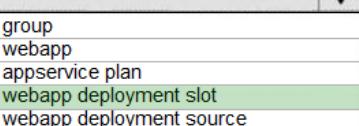
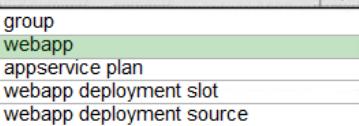
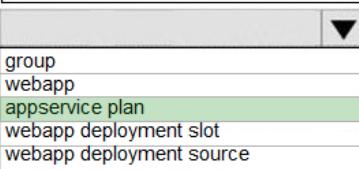
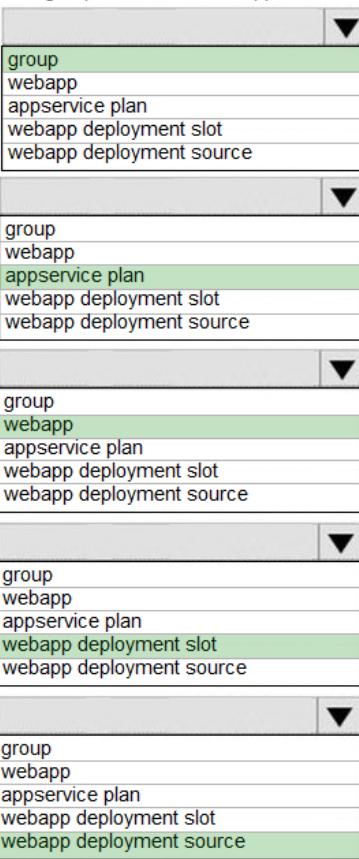
Create the Azure Functions app with a Consumption plan type.

Create a user-assigned managed identity for the application.

Create an access policy in Azure Key Vault for the application identity.



```
gitrepo=https://github.com/Contoso/webapp
webappname=businesswebapp
resourcegroupname=BusinessAppResourceGroup
az group create --location centralus --name $resourcegroupname
az appservice plan create --name $webappname --resource-group $resourcegroupname --sku S3
az webapp create --name $webappname --resource-group $resourcegroupname --plan $webappname
az webapp deployment slot create --name $webappname --resource-group $resourcegroupname --slot staging
az webapp deployment source config --name $webappname --resource-group $resourcegroupname \
--slot staging --repo-url $gitrepo --branch master --manual-integration
```



```

function ensureTip() {
  var r = ___.value();
          ___.readDocument('item');
          getContext().getRequest();
          getContext().getResponse();

  var i = r.getBody();

  if (!("tip" in i)) {
    if (request.getValue("tip") === null) {
      if (isNaN(i["tip"]) || i["tip"] === null) {
        if (typeof __.pluck("tip") === 'number') {

          i["tip"] = 0;
        }
      }
    }
  }

  r.setBody(i);
  r.setValue(i);
  ___.upsertDocument(i);
  ___.replaceDocument(i);
}

```

Commands

```

FROM microsoft/aspnetcore:latest
WORKDIR /apps/ContosoApp
CMD ["dotnet", "ContosoApp.dll"]
COPY ./ .
RUN powershell ./setupScript.ps1

```

Answer Area

```

CMD ["dotnet", "ContosoApp.dll"]
FROM microsoft/aspnetcore:latest
WORKDIR /apps/ContosoApp
COPY ./ .
RUN powershell ./setupScript.ps1

```

```
{
  ...
  "resources": [
    {
      "apiVersion": "2016-01-01",
      "type": "Microsoft.Storage/storageAccounts",
      "name": "[concat(          (), 'storage', uniqueString(resourceGroup().id))]",
      "copy": {
        "copyIndex": 0,
        "priority": 10,
        "dependsOn": [
          "copy"
        ]
      }
    },
    {
      "location": "[resourceGroup().location]",
      ...
      "sku": {
        "name": "Standard_LRS"
      },
      "kind": "Storage",
      "properties": {},
      "copy": {
        "copyIndex": 0,
        "priority": 10,
        "dependsOn": [
          "copy"
        ]
      }
    },
    {
      "name": "storagesetup",
      "count": 3
    }
  ],
  {
    "apiVersion": "2015-06-15",
    "type": "Microsoft.Compute/virtualMachines",
    "name": "[concat('VM', uniqueString(resourceGroup().id))]",
    "dependsOn": [
      "[variables('loadBalancerName')]",
      "[variables('virtualNetworkName')]",
      "storagesetup",
      ...
    ],
    ...
  }
],
}
```

Consistency levels	
Strong	Bounded Staleness
Consistent Prefix	Eventual

Answer Area

Return the most recent patient status.

Strong

Return health monitoring data that is no less than one version behind.

Bounded Staleness

After patient is discharged and all charges are assessed, retrieve the correct billing data with the final charges.

Eventual

Action	Tool or service
Generalize the VM.	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="background-color: #90EE90; color: white; padding: 2px 0; margin-bottom: 2px;">Azure PowerShell</div> Visual Studio command prompt Azure Migrate Azure Backup </div>
Store images.	<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <div style="background-color: #90EE90; color: white; padding: 2px 0; margin-bottom: 2px;">Azure Blob Storage</div> Azure Data Lake Storage Azure File Storage Azure Table Storage </div>
Actions	Answer Area
Use AZCopy to copy the data to the new storage account.	Create a new template deployment.
Deploy the template to create a new storage account in the target region.	Export a Resource Manager template.
Export a Resource Manager template.	<div style="display: flex; align-items: center;"> ◀ Modify the template by changing the storage account name and region. ▶ </div>
Create a new template deployment.	Deploy the template to create a new storage account in the target region.
Modify the template by changing the storage account name and region.	Use AZCopy to copy the data to the new storage account.
Features	Answer Area
Run Command	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Requirement</p> <p>Firewall configuration</p> <p>Supporting services script</p> </div> <div style="width: 45%;"> <p>Feature</p> <p>Run Command</p> <p>Hybrid Runbook Worker</p> </div> </div>
Serial console	
Hybrid Runbook Worker	
Custom Script Extension	

Answer Area

```
$gitrepo="https://github.com/TailSpinToys/webapp"
$webappname="TailSpinToysWeb"
$location="WestUS2"
-Name myResourceGroup -Location $location
New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -ResourceGroupName myResourceGroup -Tier Standard
New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -Location $location -AppServicePlan $webappname -ResourceGroupName myResourceGroup
New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
-Name $webappname -ResourceGroupName myResourceGroup -Slot review
New-AzWebAppSlot
New-AzWebApp
New-AzAppServicePlan
New-AzResourceGroup
$PropertiesObject = @{$repoUrl = "$gitrepo";branch = "master";}
Set-AzResource -PropertyObject $PropertiesObject -ResourceGroupName myResourceGroup -ResourceType
Microsoft.Web/sites/slots/sourcecontrols -ResourceName $webappname/review/web -ApiVersion 2015-08-01 -Force
Switch-AzWebAppSlot -Name $webappname -ResourceGroupName myResourceGroup
-SourceSlotName review -DestinationSlotName production
```

Answer Area

```
$vm = Get-AzVM -ResourceGroupName "ContosoRG" -Name "ContosoVM"
```

```
Update-AzVM -ResourceGroupName "ContosoRG" -VM $vm
```

-AssignIdentity:	<input type="radio"/> \$SystemAssigned <input checked="" type="radio"/> \$UserAssigned
------------------	---

```
$vm = Get-AzVM -ResourceGroupName myResourceGroup -Name myVM
```

```
Update-AzVM -ResourceGroupName myResourceGroup -VM $vm -IdentityType SystemAssigned
```

Reference:

Features

- Custom handler
- Extension bundle
- Trigger
- Runtime
- Policy
- Hosting plan

Answer Area

Requirement

Enable developers to write the functions by using the Rust language.

Feature

Custom handler

Declaratively connect to an Azure Blob Storage account.

Trigger

Box 1: Custom handler -

Custom handlers can be used to create functions in any language or runtime by running an HTTP server process, for example Go or Rust.

Box 2: Trigger -

Functions are invoked by a trigger and can have exactly one. In addition to invoking the function, certain triggers also serve as bindings. You may also define multiple bindings in addition to the trigger. Bindings provide a declarative way to connect data to your code.

Answer Area

App setting	Value
LOCALAPPDATA	true
WEBSITE_LOCALCACHE_ENABLED	
DOTNET_HOSTING_OPTIMIZATION_CACHE	
WEBSITES_ENABLE_APP_SERVICE_STORAGE	
DIAGDATA	

You are developing an Azure Function App. You develop code by using a language that is not supported by the Azure Function App host. The code language supports HTTP primitives.

Configuration parameter	Configuration value
Publish	<div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>Code</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>Docker Container</div>
Runtime stack	<div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>Node.js</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>Python</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>PowerShell Core</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>Custom Handler</div>
Version	<div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>14 LTS</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>7.0</div> <div style="border: 1px solid black; padding: 2px;"><div style="background-color: #e0f2e0; height: 15px;"></div>custom</div>

You provision virtual machines (VMs) as development environments.

One VM does not start. The VM is stuck in a Windows update process. You attach the OS disk for the affected VM to a recovery VM.

You need to correct the issue.

Actions

Run the following command at an elevated command prompt:

```
dism /image:\ /get=packages > c:\temp\Patch.txt
```

Run the following command at an elevated command prompt:

```
dism /Image:<Attached OS disks>:\ /Remove  
Package /PackageName:<package name to delete>
```

Detach the OS disk and recreate the VM

Open C:\temp\Patch.txt file and locate the update that is in a pending state

Answer Area

Run the following command at an elevated command prompt:

```
dism /image:\ /get=packages > c:\temp\Patch.txt
```

Open C:\temp\Patch.txt file and locate the update that is in a pending state

Run the following command at an elevated command prompt:

```
dism /Image:<Attached OS disks>:\ /Remove  
Package /PackageName:<package name to delete>
```

Detach the OS disk and recreate the VM

Answer Area

```
[FunctionName("App")]
public static async Task<List<string>> RunOrchestrator(
    [OrchestrationTrigger] IDurableOrchestrationContext context)
{
    EntityId[] input = . . .
    int errIndex = . . .

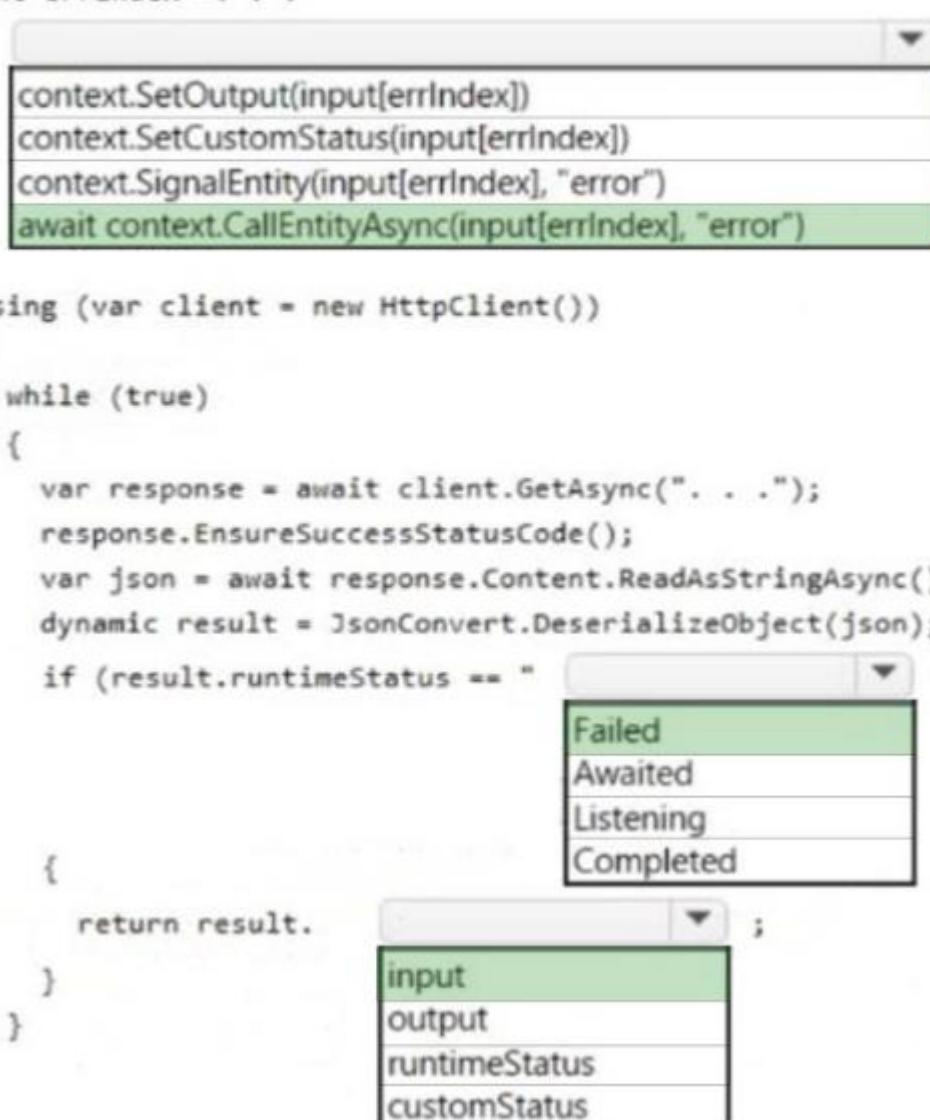
    context.SetOutput(input[errIndex])
    context.SetCustomStatus(input[errIndex])
    context.SignalEntity(input[errIndex], "error")
    await context.CallEntityAsync(input[errIndex], "error")
}

using (var client = new HttpClient())
{
    while (true)
    {
        var response = await client.GetAsync("...");  

        response.EnsureSuccessStatusCode();
        var json = await response.Content.ReadAsStringAsync();
        dynamic result = JsonConvert.DeserializeObject(json);
        if (result.runtimeStatus == "Failed")  

        {
            return result.  

        }
    }
}
```



The screenshot shows a portion of an Azure Functions orchestrator's run function. The code handles errors by setting the output to the entity at the error index and calling the entity again. It then enters a loop where it makes an HTTP request to a local endpoint, checks the status code, reads the JSON response, and deserializes it into a dynamic object. If the runtime status is 'Failed', it returns the result. A tooltip for the 'runtimeStatus' property lists 'Failed', 'Awaited', 'Listening', and 'Completed'. Another tooltip for the 'input' parameter lists 'input', 'output', 'runtimeStatus', and 'customStatus'.

You are authoring a set of nested Azure Resource Manager templates to deploy multiple Azure resources.

The templates must be tested before deployment and must follow recommended practices.

You need to validate and test the templates before deployment.

Tools	Answer Area	Requirement	Tool
Parameter file		Determine whether the templates follow recommended practices.	Azure Resource Manager test toolkit
Template function			
Azure Resource Manager test toolkit			
User-defined function			
What-if operation		Test and validate changes that templates will make to the environment.	What-if operation
Azure Deployment Manager			

You are developing an Azure Function app. All functions in the app meet the following requirements:

- Run until either a successful run or until 10 run attempts occur.
- Ensure that there are at least 20 seconds between attempts for up to 15 minutes.

```
{
```

```
"
```

```
": {
```

```
retry  
healthMonitor  
singleton
```

```
"strategy": "
```

```
",
```

```
exponentialBackoff  
counterThreshold  
fixedDelay
```

```
"
```

```
": 10,
```

```
maxRetryCount  
healthCheckInterval  
healthCheckThreshold
```

```
"minimumInterval": "00:00:20",  
"maximumInterval": "00:15:00"
```

```
}
```

```
}
```

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

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


az cosmosdb create \
--name $name \


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


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


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


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

```

az configure --defaults web=website
az configure --defaults group=website
az appservice plan create --name websitePlan

```

```

az webapp create --plan websitePlan

```

```

az webapp config

```

You are developing a back-end Azure App Service that scales based on the number of messages contained in a Service Bus queue.
A rule already exists to scale up the App Service when the average queue length of unprocessed and valid queue messages is greater than 1000.

Scale rule

Metric source

Storage queue

Service Bus queue

Current resource

Storage queue (classic)

Criteria

*** Metric name**

Message Count

Active Message Count

1 minute time grain

*** Time grain statistic**

Count

Total

Maximum

Average

*** Queues**

itemqueue

*** Operator**

Less than or equal to

*** Operator**

Decrease count by

You have an application that uses Azure Blob storage.

You need to update the metadata of the blobs.

Methods

Metadata.Add

SetMetadataAsync

FetchAttributesAsync

UploadFileStream

SetPropertiesAsync

Answer Area

Metadata.Add

SetMetadataAsync

SetPropertiesAsync

You are maintaining an existing application that uses an Azure Blob GPv1 Premium storage account. Data older than three months is rarely used.

Data newer than three months must be available immediately. Data older than a year must be saved but does not need to be available immediately.

You need to configure the account to support a lifecycle management rule that moves blob data to archive storage for data not modified in the last year.

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

Actions

Upgrade the storage account to GPv2

Create a new GPv2 Standard account and set its default access tier level to cool

Change the storage account access tier from hot to cool

Copy the data to be archived to a Standard GPv2 storage account and then delete the data from the original storage account

Answer Area

Upgrade the storage account to GPv2

Copy the data to be archived to a Standard GPv2 storage account and then delete the data from the original storage account

Change the storage account access tier from hot to cool

Code segment 1

`http://localhost:50342/oauth2/token`

`http://169.254.169.254:50432/oauth2/token`

`http://localhost/metadata/identity/oauth2/token`

`http://169.254.169.254/metadata/identity/oauth2/token`

Answer Area

```
var url = " http://169.254.169.254/metadata/identity/oauth2/token " ;
```

```
var queryString = "...";  
var client = new HttpClient();  
var response = await client.GetAsync(url + queryString);  
var payload = await response.Content.ReadAsStringAsync();
```

```
return JsonConvert.DeserializeObject<Dictionary<string, string>>(payload);
```

Code segment 2

`XDocument.Parse(payload);`

`new MultipartContent(payload);`

`new NetworkCredential("Azure", payload);`

`JsonConvert.DeserializeObject<Dictionary<string, string>>(payload);`

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

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

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

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

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

JSON segments	Answer Area
orderBy	{
sortOrder	"automatic": true, "ngMode": "Consistent", "includedPaths": [{ "path": "/*" }], "excludedPaths": [], "compositeIndexes": [
ascending	{ "path": "/name", "order": "descending" }, { "path": "/city", "order": "descending" }] }
descending	} }, { "path": "/name", "order": "descending" }, { "path": "/city", "order": "descending" }]
compositeIndexes	You can order by multiple properties. A query that orders by multiple properties requires a composite index. Example: Composite index defined for (name ASC, age ASC): It is optional to specify the order. If not specified, the order is ascending. ↓

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

Data from traffic sensors are stored in Azure Event Hub.

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

Data throughput must be maximized. Latency must be minimized.

You need to implement the Azure Event Hub.

Setting	Value
Number of partitions	<input type="text" value="12"/> 3 The number of partitions is specified 4 at creation and must be between 2 and 32. 6 There are 6 highways. 12
Partition Key	<input type="text" value="Highway"/> Highway Department Timestamp VM name

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

Filter types	Answer Area	Filter type
SQLFilter	Subscription	SQLFilter
CorrelationFilter	FutureOrders	CorrelationFilter
No Filter	HighPriorityOrders	SQLFilter
	InternationalOrders	SQLFilter
	HighQuantityOrders	SQLFilter
	AllOrders	No Filter

Your company has several websites that use a company logo image. You use Azure Content Delivery Network (CDN) to store the static image. You need to determine the correct process of how the CDN and the Point of Presence (POP) server will distribute the image and list the items in the correct order. In which order do the actions occur? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:

Actions

- If no edge servers in the POP have the image in cache, the POP requests the file from the origin server.
- A user requests the image from the CDN URL. The DNS routes the request to the best performing POP location.
- Subsequent requests for the file may be directed to the same POP using the CDN logo image URL. The POP edge server returns the file from cache if the TTL has not expired.
- The origin server returns the logo image to an edge server in the POP. An edge server in the POP caches the logo image and returns the image to the client.

Answer Area

- A user requests the image from the CDN URL. The DNS routes the request to the best performing POP location.
- If no edge servers in the POP have the image in cache, the POP requests the file from the origin server.
- The origin server returns the logo image to an edge server in the POP. An edge server in the POP caches the logo image and returns the image to the client.
- Subsequent requests for the file may be directed to the same POP using the CDN logo image URL. The POP edge server returns the file from cache if the TTL has not expired.

Components

- Host
- Delegate
- Lease container
- Monitored container

Answer Area

Requirement

- Store the data from which the change feed is generated.
- Coordinate processing of the change feed across multiple workers.
- Use the change feed processor to listen for changes.
- Handle each batch of changes.

Component

- Monitored container
- Lease container
- Host
- Delegate

You are developing a web application that will use Azure Storage. Older data will be less frequently used than more recent data.

You need to configure data storage for the application. You have the following requirements:

- Retain copies of data for five years.
- Minimize costs associated with storing data that is over one year old.
- Implement Zone Redundant Storage for application data.

Requirement

Configure an Azure Storage account

Solution

- | |
|--|
| ▼ |
| Implement Blob Storage |
| Implement Azure Cosmos DB |
| Implement Storage (general purpose v1) |
| Implement StorageV2 (general purpose v2) |

Configure data retention

- | |
|---|
| ▼ |
| Snapshot blobs and move them to the archive tier |
| Set a lifecycle management policy to move blobs to the cool tier |
| Use AzCopy to copy the data to an on-premises device for backup |
| Set a lifecycle management policy to move blobs to the archive tier |

You are developing an application to collect the following telemetry data for delivery drivers:
first name, last name, package count, item id, and current location
coordinates. The app will store the data in Azure Cosmos DB.
You need to configure Azure Cosmos DB to query the data.

Configuration Parameter

Azure Cosmos DB API

Core(SQL) API stores data in document format.

	▼
Gremlin	
Table API	
Core (SQL)	

Azure Cosmos DB partition key

item id is a unique identifier and is suitable for the partition key.

	▼
first name	
last name	
package count	
item id	

Features

- Change feed estimator
- Dead-letter queue
- Deployment unit
- Lease container

Answer Area

Requirement

- Monitor the progress of the change feed processor
- Prevent the change feed processor from retrying the entire batch when one document cannot be read

Feature

- Change feed estimator
- Dead-letter queue

Requirement

Read a new batch of documents while keeping track of the failing batch of documents.

Feature

- Lease container
- Dead-letter queue
- Life-cycle notifications
- Change feed estimator

Handle errors in the change feed processor.

- Lease container
- Dead-letter queue
- Life-cycle notifications
- Change feed estimator

You are developing a solution to store documents in Azure Blob storage. Customers upload documents to multiple containers. Documents consist of PDF, CSV, Microsoft Office format and plain text files.

- ⇒ Allow filtering by the customer identifier.
- ⇒ Allow searching of information contained within a document
- ⇒ Minimize costs.

Requirement

Search and filter by customer identifier

Solution

Azure Cognitive Search
Azure Blob index tags
Azure Blob inventory policy
Azure Blob metadata

Search information inside documents

Azure Cognitive Search
Azure Blob index tags
Azure Blob inventory policy
Azure Blob metadata

You are developing a web application by using the Azure SDK. The web application accesses data in a zone-redundant BlockBlobStorage storage account.

The application must determine whether the data has changed since the application last read the data. Update operations must use the latest data changes when writing data to the storage account.

Code evaluation

Value

HTTP Header value

ETag
Last Modified
VersionId

Conditional header

If-Match
If-Modified-Since
If-None-Match

An organization deploys a blob storage account. Users take multiple snapshots of the blob storage account over time.

You need to delete all snapshots of the blob storage account. You must not delete the blob storage account itself.

```
Delete (Azure.Storage.Blobs.Models.DeleteSnapshotsOption  
snapshotsOption = Azure.Storage.Blobs.Models.
```

DeleteIfExists
DeleteSnapshotsOption
WithSnapshot
WithSnapshotCore

IncludeSnapshots
None
OnlySnapshots

```
delete_blob (
```

delete_container
delete_snapshots
snapshot_blob
snapshots_present

=
False
Include
Only

You are developing an application that monitors data added to an Azure Blob storage account.
You need to process each change made to the storage account.

```
var changeFeedClient = new BlobServiceClient("...").GetChangeFeedClient();
var x = default(string);
while (true)
{
    var changeFeed = changeFeedClient. ; 
    foreach (var c in changeFeed)
    {
        x = c. ; 
        ProcessChanges(c.Values);
    }
}
```

You develop an application that sells AI generated images based on user input. You recently started a marketing campaign that displays unique ads **every second day**.
Sales data is stored in Azure Cosmos DB with the date of each sale being stored in a property named **'whenFinished'**.
The marketing department requires a view that shows the number of sales for each unique ad.
You need to implement the query for the view.

```
SELECT
    max(c.whenFinished),
    sum(c.whenFinished),
    count(c.whenFinished) ; 
    DateTimeBin(c.whenFinished, 'day', 2); 
    DateTimePart(c.whenFinished, 'day', 2)
    DateTimeBin(c.whenFinished, 'hour', 12)
    DateTimePart(c.whenFinished, 'hour', 12)

FROM c
group by ; 
    DateTimeBin(c.whenFinished, 'day', 2); 
    DateTimePart(c.whenFinished, 'day', 2)
    DateTimeBin(c.whenFinished, 'hour', 12)
    DateTimePart(c.whenFinished, 'hour', 12)
```

Features	Answer Area	Action	Feature
Access policy		Enable retention period and accidental deletion.	Soft delete
Purge protection			
Soft delete		Enforce retention period and accidental deletion.	Purge protection
Shared access signature			

Settings

```
client_id  
profile  
delegated  
application  
user_impersonation
```

Answer Area

API	Permission	Type
Azure Storage	user_impersonation	delegated
Microsoft Graph	User.Read	delegated

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
    }

    app.();  

    UseAuthentication  

    UseStaticFiles  

    UseSession  

    UseCookiePolicy

    app.();  

    UseAuthorization  

    UseHttpsRedirection  

    UseSession  

    UseCookiePolicy

    app.();  

    UseAzureAppConfiguration  

    UseRequestLocalization  

    UseCors  

UseStaticFiles

    app.UseEndpoint(endpoints =>
    {
        endpoints.MapRazorPages();
    });
}
```

```
az provider register -n Microsoft.KeyVault
resourcegroup="myResourceGroup"
az group create --name $resourcegroup --location westus
keyvault_name=myvaultname$RANDOM

az [▼] create \
    

|               |
|---------------|
| vm            |
| keyvault      |
| keyvault key  |
| vm encryption |



--name $keyvault_name \
--resource-group $resourcegroup \
--location eastus \
--enabled-for-disk-encryption True

az [▼] create \
    

|               |
|---------------|
| vm            |
| keyvault      |
| keyvault key  |
| vm encryption |



--vault-name $keyvault_name \
--name Name1 \
--protection software

az [▼] create \
    

|               |
|---------------|
| vm            |
| keyvault      |
| keyvault key  |
| vm encryption |



--resource-group $resourcegroup \
--name Name2 \
--image Canonical:UbuntuServer:16.04-LTS:latest \
--admin-username azureuser \
--generate-ssh-keys \
--data-disk-sizes-gb 5

az [▼] enable \
    

|               |
|---------------|
| vm            |
| keyvault      |
| keyvault key  |
| vm encryption |



--resource-group $resourcegroup \
--name Name2 \
--disk-encryption-keyvault $keyvault_name \
--key-encryption-key Name1 \
--volume-type \
    

|      |
|------|
| all  |
| data |
| os   |


```

Powershell commands

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

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

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

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

```
Get-AzSubscription
```

Answer Area

```
Get-AzSubscription
```

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

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

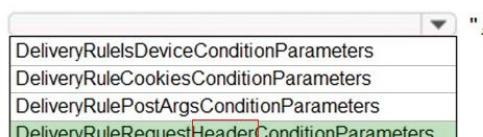
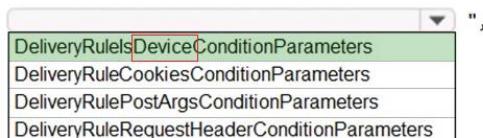
```
$secretvalue = ConvertTo-SecureString
$storAcctkey -AsPlainText
-Force
    Set-AzKeyVaultSecret -VaultName
$vaultName -Name $secretName
-SecretValue $secretvalue
```

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

You are building a website that is used to review restaurants. The website will use an Azure CDN to improve performance and add functionality to requests. You build and deploy a mobile app for Apple iPhones. Whenever a user accesses the website from an iPhone, the user must be redirected to the app store.

Answer Area

```
"conditions": [ {
    "name": "IsDevice",
    "parameters": {
        "@odata.type": "#Microsoft.Azure.Cdn.Models.
        "operator": "Equal",
        "matchValues": [ " " ]
    }
},
{
    "name": "RequestHeader",
    "parameters": {
        "@odata.type": "#Microsoft.Azure.Cdn.Models.
        "operator": "Contains",
        "selector": " "
    }
}
]
}
] }
```



You are building a website to access project data related to teams within your organization. The website does not allow anonymous access. Authentication is performed using an Azure Active Directory (Azure AD) app named internal.

The website has the following authentication requirements:

- ⇒ Azure AD users must be able to login to the website.
- ⇒ Personalization of the website must be based on membership in Active Directory groups.

You need to configure the application's manifest to meet the authentication requirements.

```
{  
    ...  
    "appId": "d61126e3-089b-4adb-b721-d5023213df7d",  
    "displayName": "internal",  
    "optionalClaims": { "target": "All" },  
    "groupMembershipClaims": "  
        "optionalClaims"  
        "groupMembershipClaims"  
    ",  
    "allowPublicClient": true,  
    "oauth2Permissions": { "target": "true" },  
    "requiredResourceAccess": { "target": "true" },  
    "oauth2AllowImplicitFlow": true  
}
```

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

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

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

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

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

You are developing an Azure solution.

You need to develop code to access a secret stored in Azure Key Vault.

Code segments

```
DefaultAzureCredential
```

```
ClientSecretCredential
```

```
CloudClients
```

```
SecretClient
```

Answer Area

```
string var1 = Environment.GetEnvironmentVariable("KEY_VAULT_URI");  
var var2 = new SecretClient ( new Uri(var1), new DefaultAzureCredential () );
```

You are developing an Azure-hosted application that must use an on-premises hardware security module (HSM) key.

The key must be transferred to your existing Azure Key Vault by using the Bring Your Own Key (BYOK) process.

You need to securely transfer the key to Azure Key Vault.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions:

Actions

Generate a key transfer blob file by using the HSM vendor-provided tool.

Generate a Key Exchange Key (KEK).

Create a custom policy definition in Azure Policy.

Run the `az keyvault key import` command.

Run the `az keyvault key restore` command.

Retrieve the Key Exchange Key (KEK) public key.

Answer Area

Generate a Key Exchange Key (KEK).

Retrieve the Key Exchange Key (KEK) public key.

Generate a key transfer blob file by using the HSM vendor-provided tool.

Run the `az keyvault key import` command.

Exchange KEK, Exchange KEK Publik KEY, HSM Tool, key Import

You are developing a web application that makes calls to the Microsoft Graph API. You register the application in the Azure portal and upload a valid X509 certificate. You create an `appsettings.json` file containing the certificate name, client identifier for the application, and the tenant identifier of the Azure Active Directory (Azure AD). You create a method named `ReadCertificate` to return the X509 certificate by name.

You need to implement code that acquires a token by using the certificate.

```
AuthenticationConfig config = AuthenticationConfig.ReadFromJsonFile("appsettings.json");
X509Certificate2 certificate = ReadCertificate(config.CertificateName);
var app =
    ConfidentialClientApplicationBuilder
        .GetAccountAsync()
        .GetAccountsAsync()
        .ConfidentialClientApplication
    .WithCertificate(certificate)
    .WithAuthority(new Uri(config.Authority))
    .Build();
string[] scopes = new string[] { $"{config.ApiUrl}.default" };
AuthenticationResult result = await app.AcquireTokenForClient(
    scopes
    ).ExecuteAsync();
```

You develop a containerized application. You plan to deploy the application to a new Azure Container instance by using a third-party continuous integration and continuous delivery (CI/CD) utility.

The deployment must be unattended and include all application assets. The third-party utility must only be able to push and pull images from the registry. The authentication must be managed by Azure Active Directory (Azure AD). The solution must use the principle of least privilege.

Authentication

Registry authentication method

Option

Service principal
Individual identity
Repository-scoped access token
Managed identity for Azure resources

RBAC role

AcrPull
Owner
AcrPush
Contributor

You are a developer building a web site using a web app. The web site stores configuration data in Azure App Configuration.

Access to **Azure App Configuration** has been configured to use the identity of the web app for authentication. Security requirements specify that no other authentication systems must be used.

You need to load configuration data from Azure App Configuration.

```
Host .CreateDefaultBuilder(args)
.ConfigureWebHostDefaults(wb =>
{
    wb.ConfigureAppConfiguration((hc, config) =>
    {
        var settings = config.Build();
        config.                                     (options =>
            AddAzureKeyVault
            DefaultAzureCredential
            ChainedTokenCredential
            ManagedIdentityCredential
            AddAzureAppConfiguration
        options.Connect(new Uri(settings["AppConfig:Endpoint"]),
            new                                         ());
            AddAzureKeyVault
            DefaultAzureCredential
            ChainedTokenCredential
            ManagedIdentityCredential
            AddAzureAppConfiguration
        });
    });
});
```

You are building an application that stores sensitive customer data in Azure Blob storage. The data must be encrypted with a key that is unique for each customer.

If the encryption key has been corrupted it must not be used for encryption.

You need to ensure that the blob is encrypted.

```
Uri blobUri = ... ; TokenCredential c = ...
byte[] key = ...; string verify = ...
var x = new                                         ;
    AesManaged(key)
    AsnEncodedData(key)
    CustomerProvidedKey(key)
    BlobContainerEncryptionScopeOptions { DefaultEncryptionScope = key }
if (                                         ) {
    x.IV == verify
    x.RawData == verify
    x.Encryption == verify
    x.PreventEncryptionScopeOverride == verify
}
var o = new BlobClientOptions()
{
    = x
    Version
    Transport
    EncryptionScope
    CustomerProvidedKey
};
var blobClient = new BlobClient(blobUri, c, o);
}
```

You are developing a new web service by using a web app named ExcelParser. The web app contains a third-party library for processing Microsoft Excel files. The license for the third-party library stipulates that you can only run a single instance of the library.

You need to configure the service.

```
Set-AzAppServicePlan `
```

```
-ResourceGroupName $rg `
```

```
-Name "PrimaryASP" `
```

```
NumberofSites 1
PerSiteScaling $true
TargetWorkerCount = 1
MaxNumberOfWorkers = 1
SiteConfig.NumberOfWorkers = 1
```

```
$app = Get-AzWebApp `
```

```
-ResourceGroupName $rg `
```

```
-Name "ExcelParser"
```

```
$app. NumberofSites 1
PerSiteScaling $true
TargetWorkerCount = 1
MaxNumberOfWorkers = 1
SiteConfig.NumberOfWorkers = 1
```

```
Set-AzWebApp $app
```

Answer Area

Features

horizontal partitioning

channel

list

set

Requirement

Create a data structure for storing collections of related items

Create a data structure for the most recently accessed cache items

Send messages through a high-performance publisher/subscriber mechanism

Feature

set

list

channel

You are developing a REST web service. Customers will access the service by using an Azure API Management instance.

The web service does not correctly handle conflicts. Instead of returning an HTTP status code of 409, the service returns a status code of 500. The body of the status message contains only the word conflict.

You need to ensure that conflicts produce the correct response.

Policy segments

server

context

on-error

set-status

when-error

override-status

Answer Area

```
< on-error >
<base />
<choose>
  <when condition = " @ context .Response.StatusCode == 500
    && context .LastError.Message.Contains
      " conflict = " )>
    <return-response>
      < set-status >
        </return-response>
    </when>
    <otherwise />
  </choose>
< on-error >
```

The application must be granted access to the Azure Blob storage account with a start time, expiry time, and read permissions.
The Azure Blob storage account access
must use the Azure AD credentials of the application to secure data access.
Data access must be able to be revoked if the client application security is breached.

Component	Security Feature
Application (Client)	<ul style="list-style-type: none">Storage Account Access KeySystem-assigned Managed IdentityShared access signature (SAS) token
Azure Storage (Server)	<ul style="list-style-type: none">Stored Access PolicyUser-assigned Managed IdentityCross-Origin Resource Sharing (CORS)

You develop and deploy a web app to Azure App Service in a production environment. You scale out the web app to four instances and configure a staging slot to support changes.

You must monitor the web app in the environment to include the following requirements:

- Increase web app availability by re-routing requests away from instances with error status codes and automatically replace instances if they remain in an error state after one hour.
- Send web server logs, application logs, standard output, and standard error messaging to an Azure Storage blob account.

You need to configure Azure App Service.

Configuration values	Answer Area	Configuration value
Health check	Requirement	Health check
Diagnostic setting	Increase availability	
Deployment slot	Send logs	Diagnostic setting
Autoscale rule		
Zone redundancy		

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

You plan to implement an Azure Redis Cache instance to improve the efficiency of data operations for entities that rarely change. You need to invalidate the cache when team data is changed.

```
void ClearCachedTeams()  
{  
    Database cache = Connection.GetDatabase();  
    ICache cache = Connection.GetDatabase();  
  
    cache.KeyDelete("Team");  
    cache.StringSet("Team", "");  
    cache.ValueDelete("Team");  
    cache.StringGet("Team", "");  
  
    ViewBag.msg += "Team data removed from cache.";  
}
```

You develop and deploy an Azure Logic App that calls an Azure Function app. The Azure Function App includes an OpenAPI ([Swagger](#)) definition and uses an Azure Blob storage account. All resources are secured by using Azure Active Directory (Azure AD). The Logic App must use Azure Monitor logs to record and store information about runtime data and events. The logs must be stored in the Azure Blob storage account. You need to set up Azure Monitor logs and collect diagnostics data for the Azure Logic App.

Actions

- Create action groups and alert rules.
- Create a Log Analytics workspace.
- Install the Logic Apps Management solution.
- Add a diagnostic setting to the Azure Function App.
- Create an Azure storage account.
- Add a diagnostic setting to the Azure Logic App.

Answer Area

Create a Log Analytics [workspace](#).



Install the Logic Apps Management solution.

Add a diagnostic setting to the Azure Logic App.

Technologies

- Azure Event Hub
- Azure Event Grid
- Azure Service Bus
- Azure Blob Storage
- Azure App Service
- Azure Logic App

Answer Area

Object

Event Source

Event Receiver

Event Handler

Technology

Azure Blob Storage

Azure Event Grid

Azure Logic App

You develop a web application.

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

Actions

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

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

Add a Cryptographic key.

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

Select the Azure AD instance.

Use an access token to access the secure resource.

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

Answer Area

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

Select the Azure AD instance.

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



New Registration -
Select Azure AD
Create new Application

You are developing an app that manages users for a video game. You plan to store the region, email address, and phone number for the player. Some players may not have a phone number. The player's region will be used to load-balance data.

```
public class PlayerEntity : TableEntity
{
    public PlayerEntity()
    {
    }
    public PlayerEntity(string region, string email)
    {
        PartitionKey = region;
        RowKey = email;
    }
    public string Phone { get; set; }
}
public class Player
{
    protected PlayerEntity player;
    async void GetPlayer(string cs,
        CloudTable table, string pk, string rk)
    {
        TableEntity query = TableEntity.Retrieve<PlayerEntity>(pk, rk);
        TableOperation query = TableOperation.Retrieve<PlayerEntity>(pk, rk);
        TableResult query = TableQuery.Retrieve<PlayerEntity>(pk, rk);
        TableResultSegment query = TableResult.Retrieve<PlayerEntity>(pk, rk);

        TableEntity data = await table.ExecuteAsync(query);
        TableOperation data = await table.ExecuteAsync(query);
        TableQuery data = await table.ExecuteAsync(query);
        TableResult data = await table.ExecuteAsync(query);

        player = data.Result as PlayerEntity;
    }
}
```

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

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

```
string notificationHubName = "contoso hub";
string notificationHubConnection = "connection_string";
hub =
NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails
.
.
.
NotificationHubClient
NotificationHubClientSettings
NotificationHubJob
NotificationDetails
(notificationHubConnection, notificationHubName);
string windowsToastPayload =
@"<toast><visual><binding template=""ToastText01""><text id=""1"">" +
@"New item to view" + @"</text></binding></visual></toast>";
try
{
var result =
await hub.
SendWindowsNativeNotificationAsync(
windowsToastPayload);
.
.
.
}
catch (System.Exception ex)
{
...
}
```

Settings
client_id
profile
delegated
application
user_impersonation

Answer Area

API	Permission	Type
Azure Storage	user_impersonation	delegated
Microsoft Graph	User.Read	delegated

You develop a web app that uses the tier D1 app service plan by using the Web Apps feature of Microsoft Azure App Service. Spikes in traffic have caused increases in page load times.

Actions

Answer Area

Configure the web app to the Premium App Service tier.

Configure the web app to the Standard App Service tier.

Configure the web app to the Standard App Service tier.

Enable autoscaling on the web app.

Enable autoscaling on the web app.

Add a Scale rule.

Add a Scale rule.

Configure a Scale condition.

Switch to an Azure App Services consumption plan.

Select Standard Tier

Enable Scaling

Add scale rule

Configure

Configure a Scale condition.

```

public list<CloudTasks> StartTasks(List<FileTask> fileTasks, string jobId,
    string outputContainerSasUrl, string failedContainerSasUrl)
{
    BatchSharedKeyCredentials sharedKeyCredentials =
        new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName,
batchAccountKey);
    List<CloudTask> tasks = new List<CloudTask>();
    using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
    {
        CloudJob = batchClient.JobOperations. ▼ ();
        GetJob
        GetTask
        EnableJob
        CreateJob
        job.Id = jobId,
        job.PoolInfromation = new PoolInformation { PoolId = poolId };
        job.Commit();
        fileTasks.ForEach((fileTask) =>
        {
            string taskId = $"Task{DateTime.NowToFileTimeUtc().ToString()}";
            CloudTask task = new CloudTask(taskId, fileTask.Command);
            List<OutputFile> output fileList = new List<OutputFile>();
            OutputFileBlobContainerDestination outputContainer =
                new OutputFileBlobContainerDestination(outputContainerSasUrl);
            OutputFileBlobContainerDestination failedContainer =
                new OutputFileBlobContainerDestination(failedContainerSasUrl);
            output fileList.Add(new OutputFile(fileTask.Output,
                new OutputFileDestination(outputContainer),
                new OutputFileUploadOptions(OutputFileUploadCondition. ▼ ))));
            TaskSuccess
            TaskFailure
            TaskCompletion
            output fileList.Add(new OutputFile(fileTask.Output,
                new OutputFileDestination(failedContainer),
                new OutputFileUploadOptions(OutputFileUploadCondition. ▼ ))));
        });
    }
}

```

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

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

You need to implement an Azure Service Bus solution.

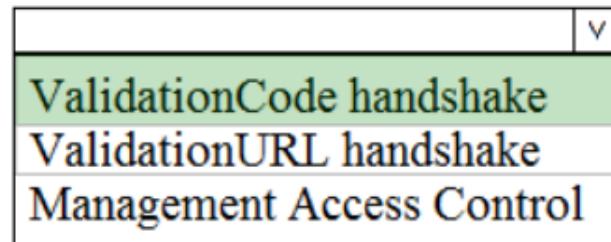
Actions
Create a single Service Bus topic.
Create a Service Bus Namespace for each restaurat for which a driver can receive messages.
Create a single Service Bus subscription.
Create a Service Bus subscription for each restaurant for which a driver can receive orders.
Create s single Service Bus Namespace.
Create a Service Bus topic for each restaurant for which a driver can receive messages.

Answer Area
Create s single Service Bus Namespace.
Create a Service Bus topic for each restaurant for which a driver can receive messages.
Create a Service Bus subscription for each restaurant for which a driver can receive orders.
Single Namespace Service Bus topic Service Bus Subscription

Authentication

WebHook event delivery

Type



Topic publishing

Triggers and action blocks

- Insert Entity (Table: processing, Entity: Path X)
- Tier blob (If blob is older than the defined value, tier it to Cool or Archive tier. Blob path: Path X, Blob tier: Archive)
- When there are messages in a queue (Queue Name: processing)
- Recurrence (Interval: 1, Frequency: Month)
- List blobs 2 (Folder: /items)
- Condition (Check LastModified timestamp and whether older than the tier age variable. ticks[items[For_each]?] is less than ticks[addDaysInMonth([LastModified])].variables('TierAgeInDays'))
- Put a message on a queue (Queue Name: processing, Message: Path X)

Answer area

```
graph TD; Recurrence --> SetTierAge[Set tier age variable]; SetTierAge --> ListBlobs1[List blobs]; ListBlobs1 --> Condition[Condition: Check LastModified timestamp and whether older than the tier age variable]; Condition --> IfTrue[If true: When there are messages in a queue]; Condition --> IfFalse[Tier blob: If blob is older than the defined value, tier it to Cool or Archive tier. Blob path: Path X, Blob tier: Archive]; IfTrue --> ListBlobs2[List blobs 2: Folder: /items];
```

Recurrence - Condition - When There are message inQueue - Tierblob - List Blobs 2

Triggers and Action Blocks

The 'Triggers and Action Blocks' section contains the following triggers:

- Insert Entity:** Table: processing, Entity: Path X.
- Tier blob:** If blob is older than the defined value, tier it to Cool or Archive tier. Blob path: Path X, Blob Tier: Archive.
- When there are messages in a queue:** Queue Name: processing. Connected to tableStorageAccountConnection. Change connection.
- Recurrence:** Interval: 1, Frequency: Month.

Box 1: Recurrence -
Box 2: Insert Entity -
Box 3 (if true): Tier Blob -
Box 4: (if false): Leave blank.

Answer Area

The 'Answer Area' shows a workflow structure:

- Recurrence: Interval: 1, Frequency: Month.
- Set tier age variable (purple box).
- List blobs (purple box).
- For each (grey box):
 - Scan all blobs in this folder
 - Select an output from previous steps: value X.
 - Insert Entity (purple box):
 - Table: processing, Entity: Path X.
- If true (green checkmark):
 - Tier blob (purple box):
 - If blob is older than the defined value, tier it to Cool or Archive tier.
 - Blob path: Path X.
 - Blob Tier: Archive.
- If false (red X):
 - Empty box.

Add an action buttons are located at the bottom of the 'For each' and 'If true' sections.

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

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

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

An organization plans to deploy Azure storage services.

You need to configure shared access signature (SAS) for granting access to Azure Storage.

SAS types	Answer Area	SAS type
Account-level	Requirement Delegate access to resources in one or more of the storage services	Account-level
Service-level	Requirement Delegate access to a resource in a single storage service	Service-level
User delegation	Requirement Secure a resource by using Azure AD credentials	User delegation

You are developing an ASP.NET Core time sheet application that runs as an Azure Web App. Users of the application enter their time sheet information on the first day of every month.

```

"type": "Microsoft.Insights/metricAlerts",
"properties": {
  "criteria": {
    "odata.type": "...",
    "allOf": [
      {
        "criterionType": "DynamicThresholdCriterion",
        "SingleResourceMultipleMetricCriteria"
      },
      {
        "metricName": "Http4xx",
        "Http5xx"
      }
    ],
    "alertSensitivity": "Low"
  }
}
  
```