

70-532 Sample Questions

Q1.

You deploy a cloud service that reads and processes orders from a queue by using a worker role. The service includes a C# class named OrderProcessor.

Your organization is moving all Azure resources to use Azure Resource Manager (ARM) templates. You must migrate the code to Service Fabric. You establish a new Service Fabric cluster to deploy the updated code. You migrate all settings from the ServiceConfiguration.cscfg to a new Settings.xml file that each Service Fabric instance will use.

You need to update the code for the OrderProcessor class.

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

```
namespace WideWorldImportersOrderProcessor
{
    public class OrderProcessor:
    {
        StatelessService
        FabricClient
        ServicePartitionResolver

        private CloudQueue ordersQueue;
        private CloudBlobContainer ordersBlobContainer;
        private DbContext ordersDatabaseContext;
        protected override
        {
            Task RunAsync(CancellationToken cancelServiceInstance)
            IEnumerable<ServiceInstanceListener> CreateServiceInstanceListeners()

            ConfigurationPackage configPackage = this.Context.
                .GetConfigurationPackageObject ("Config");

            NodeContext
            CodePackageActivationContext

            KeyedCollection<string, ConfigurationProperty> parameters =
            configPackage. Settings.Sections ["MyConfigSection"].Parameters;
            string databaseConnectionString = parameters["OrdersDatabaseConnection"]?.Value;
            ordersDatabaseContext = GetOrdersDatabaseContext(databaseConnectionString);
            ordersBlobContainer = GetOrdersBlobStorageContainerReference();
            ordersQueue = GetOrdersQueueReference();
            ProcessOrders();
        }
    }
}
```

A1.

```
namespace WideWorldImportersOrderProcessor
{
    public class OrderProcessor:
    {
        StatelessService
        FabricClient
        ServicePartitionResolver

        private CloudQueue ordersQueue;
        private CloudBlobContainer ordersBlobContainer;
        private DbContext ordersDatabaseContext;
        protected override
        {
            Task RunAsync(CancellationToken cancelServiceInstance)
            IEnumerable<ServiceInstanceListener> CreateServiceInstanceListeners()

            ConfigurationPackage configPackage = this.Context.
                .GetConfigurationPackageObject ("Config");

            NodeContext
            CodePackageActivationContext

            KeyedCollection<string, ConfigurationProperty> parameters =
            configPackage. Settings.Sections ["MyConfigSection"].Parameters;
            string databaseConnectionString = parameters["OrdersDatabaseConnection"]?.Value;
            ordersDatabaseContext = GetOrdersDatabaseContext(databaseConnectionString);
            ordersBlobContainer = GetOrdersBlobStorageContainerReference();
            ordersQueue = GetOrdersQueueReference();
            ProcessOrders();
        }
    }
}
```

Q2.

You use the Windows PowerShell Desired State Configuration (DSC) feature to configure your company's servers. Line numbers are included for reference only.

```
01 $ConfigurationData = @{
02     AllNodes = @(
03         @{NodeName = 'Server1';Role='Web'},
04         @{NodeName = 'Server2';Role='FileShare'}
05         @{NodeName = 'Server3';Role=@('FileShare','Web')}
06     )
07 }
08 configuration RoleConfiguration
09 {
10     param ($Roles)
11     switch ($Roles)
12     {
13         'FileShare'
14         {
15             WindowsFeature FileSharing
16             {
17                 Name = 'FS-FileServer'
18             }
19         }
20         'Web'
21         {
22             WindowsFeature Web
23             {
24                 Name = 'Web-Server'
25                 Ensure = 'Absent'
26             }
27         }
28     }
29 }
30 configuration MyFirstServerConfig
31 {
32     node $allnodes.NodeName
33     {
34         WindowsFeature snmp
35         {
36             Name = 'SNMP-Service'
37         }
38         RoleConfiguration MyServerRoles
39         {
40             Roles = $Node.Role
41             DependsOn = '[WindowsFeature]snmp'
42         }
43     }
44 }
```

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

Answer Area

Yes	No
-----	----

The script configures SNMP service on all servers.

The script configures the Web Server (IIS) role on Server3.

Invoking the script within Windows PowerShell applies the desired state to all servers.

A2.

Yes	No
-----	----

The script configures SNMP service on all servers.

The script configures the Web Server (IIS) role on Server3.

Invoking the script within Windows PowerShell applies the desired state to all servers.

Q3.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are developing a new Azure Logic App. The Logic App requires a custom action to evaluate data from an internal, proprietary system. You create a custom ASP>NET Web API to retrieve data from the system and update the Logic App to use the API.

The Logic App generates a timeout error when it requests data from the API.

You need to eliminate the timeout error and allow the Logic App to retrieve data by using the API.

What should you do?

- A Update the API to immediately return an HTTP '102 PROCESSING' response when a request is received and an HTTP '205 RESET CONTENT' response when the data is returned from the system.
- B Update the Logic App to use a new HTTPWebhook trigger to call out to the API's newly-created subscribe and unsubscribe methods.
- C Update the API to immediately return an HTTP '202 ACCEPTED' response when a request is received and an '200 OK' response when the data is returned from the system.
- D Update the Logic App adding a wait action to include the interval object's unit and count properties set to valid values.

A3.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are developing a new Azure Logic App. The Logic App requires a custom action to evaluate data from an internal, proprietary system. You create a custom ASP>NET Web API to retrieve data from the system and update the Logic App to use the API.

The Logic App generates a timeout error when it requests data from the API.

You need to eliminate the timeout error and allow the Logic App to retrieve data by using the API.

What should you do?

- A Update the API to immediately return an HTTP '102 PROCESSING' response when a request is received and an HTTP '205 RESET CONTENT' response when the data is returned from the system.
- B Update the Logic App to use a new HTTPWebhook trigger to call out to the API's newly-created subscribe and unsubscribe methods.
- C Update the API to immediately return an HTTP '202 ACCEPTED' response when a request is received and an '200 OK' response when the data is returned from the system.
- D Update the Logic App adding a wait action to include the interval object's unit and count properties set to valid values.

Answer: C

Q4.

An entrepreneur has decided to open an e-commerce site to complement their retail store. After researching their options, they decide that a PaaS solution will be sufficient. To reduce upfront cost, the entrepreneur intends to build the site themselves. Which of the following skill-tests will be needed?

- A Firewall Administration
- B Web-Server Administration
- C Security standard development
- D Application development

A4.

An entrepreneur has decided to open an e-commerce site to complement their retail store. After researching their options, they decide that a PaaS solution will be sufficient. To reduce upfront cost, the entrepreneur intends to build the site themselves. Which of the following skill-tests will be needed?

- A Firewall Administration
- B Web-Server Administration
- C Security standard development
- D Application development

Answer: D

Explanation:

Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

PaaS allows you to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware or the development tools and other resources. You manage the applications and services you develop, and the cloud service provider typically manages everything else.

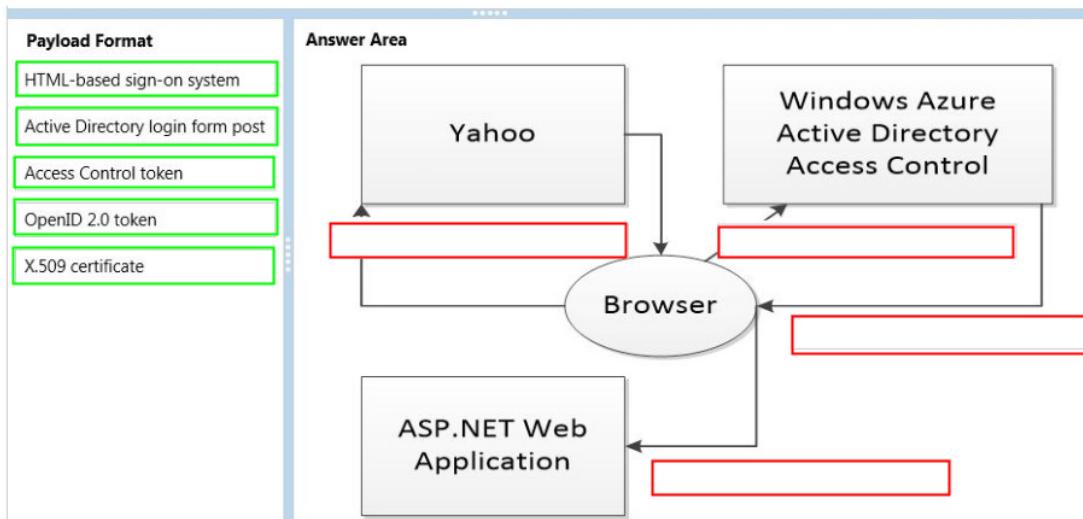
<https://azure.microsoft.com/en-us/overview/what-is-paas/>

Q5.

You are converting an existing ASP.NET web application to use the Azure Active Directory (AD) Access Control service for authentication. The application will authenticate users by using their Yahoo account credentials.

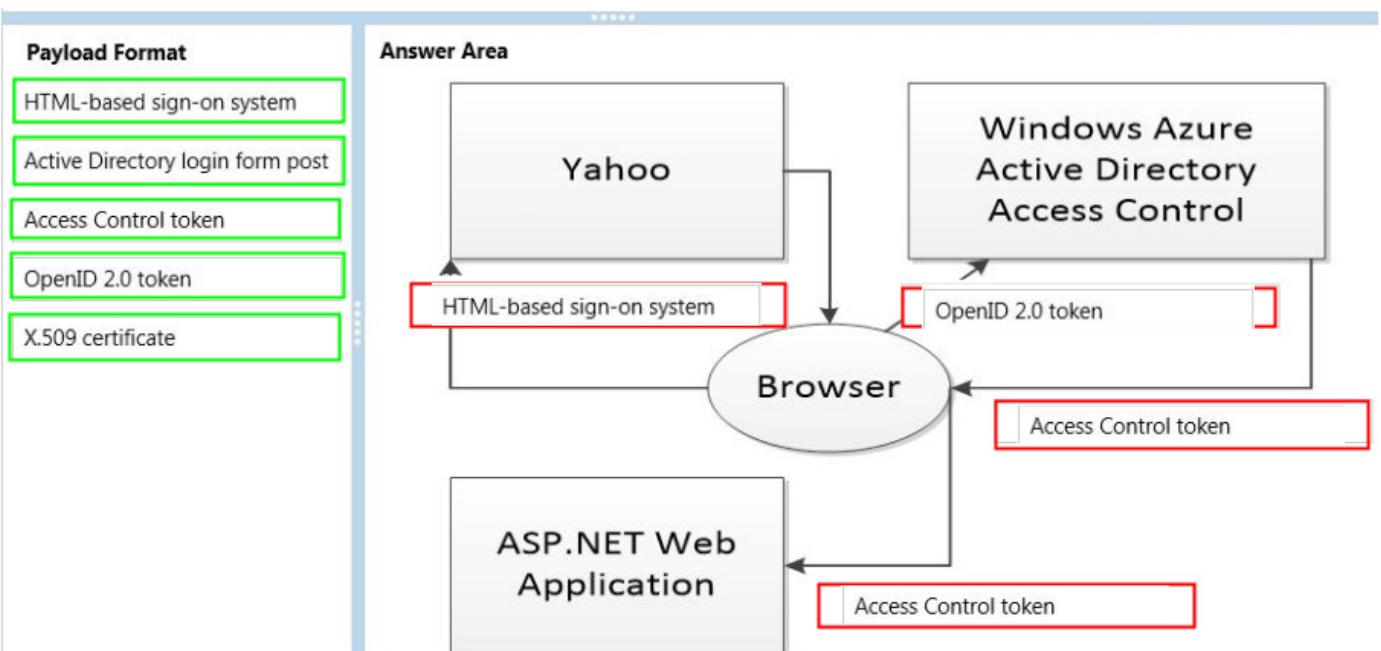
You need to determine the correct payload for each stage of the authentication process.

What should you do? To answer, drag the appropriate payload format to the correct location on the dialog box. Each payload format may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



A5.

Answer:



Q6.

You are developing a multitenant application that uses Azure Search services. You have the following tenants:

Tenant	Requirement
TenantA	The workload and data for this tenant must be isolated from other tenants.
TenantB	The data for this tenant must be isolated from other tenants, but TenantB can share its workload with other tenants.

You must minimize costs associated with implementing any solution. The cost model must be predictable.

You need to design the search experience for the application.

Which Azure Search pattern should you use for each tenant? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer area

Tenant	Pattern
TenantA	<input type="checkbox"/> index-per-tenant <input type="checkbox"/> service per tenant <input type="checkbox"/> mixed model
TenantB	<input type="checkbox"/> index-per-tenant <input type="checkbox"/> service per tenant <input type="checkbox"/> mixed model

A6.

Answer area

Tenant	Pattern
TenantA	<input type="checkbox"/> index-per-tenant <input checked="" type="checkbox"/> service per tenant <input type="checkbox"/> mixed model
TenantB	<input checked="" type="checkbox"/> index-per-tenant <input checked="" type="checkbox"/> service per tenant <input type="checkbox"/> mixed model

Q7.

You have six Ubuntu Linux virtual machines (VMs) that run a Hadoop cluster on Azure. All VMs were deployed by using Azure Resource Manager (ARM) templates and Azure PowerShell cmdlets. One of the VMs runs a custom web user interface that allows users to examine the processing jobs within the Hadoop cluster. You are planning a backup strategy for long-term retention and recovery that includes geo-replication.

The backup and recovery solution must be cost effective.

You need to backup all VMs.

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

Actions

- Select the VMs to include in the backup.
- Select the appropriate backup policy.
- Create a recovery services vault for each VM that has geo-redundant storage replication enabled.
- Set the backup goal to **Azure and VM**.**
- Run and confirm that an initial backup has been completed for all VMs.
- Create a backup vault for the VM backups that has geo-redundant storage replication enabled.
- Create a recovery services vault for the VM backups that has locally-redundant storage replication enabled.
- Create a recovery services vault for the VM backups that has geo-redundant storage replication enabled.

Answer Area

A7.

Answer:

Actions

- Select the VMs to include in the backup.
- Select the appropriate backup policy.
- Create a recovery services vault for each VM that has geo-redundant storage replication enabled.
- Set the backup goal to **Azure and VM**.**
- Run and confirm that an initial backup has been completed for all VMs.
- Create a backup vault for the VM backups that has geo-redundant storage replication enabled.
- Create a recovery services vault for the VM backups that has locally-redundant storage replication enabled.
- Create a recovery services vault for the VM backups that has geo-redundant storage replication enabled.

Answer Area

Create a recovery services vault for the VM backups that has geo-redundant storage replication enabled.
Set the backup goal to Azure and VM .
Select the appropriate backup policy.
Select the VMs to include in the backup.
Run and confirm that an initial backup has been completed for all VMs.

Q8.

You store data in an Azure blob. Data accumulates at a rate of 0.10 GB per day.

You must use storage analytics data to verify that the service level agreement (SLA) has been met and to analyze the performance of VHDs, including the pattern of usage.

Analytics data must be deleted when it is older than 100 days or when the total amount of data exceeds 10 GB.

You need to configure storage analytics and access the storage analytics data.

Which two approaches will achieve the goal? Each correct answer presents part of the solution.

- A Disable the data retention policy.
- B Access analytics data by using the Service Management REST API.
- C Access analytics data by using the APIs used to read blob and table data.
- D Configure a data retention policy of 100 days.

A8.

You store data in an Azure blob. Data accumulates at a rate of 0.10 GB per day.

You must use storage analytics data to verify that the service level agreement (SLA) has been met and to analyze the performance of VHDs, including the pattern of usage.

Analytics data must be deleted when it is older than 100 days or when the total amount of data exceeds 10 GB.

You need to configure storage analytics and access the storage analytics data.

Which two approaches will achieve the goal? Each correct answer presents part of the solution.

- A Disable the data retention policy.
- B Access analytics data by using the Service Management REST API.
- C Access analytics data by using the APIs used to read blob and table data.
- D Configure a data retention policy of 100 days.

Answer: C, D

Q9.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
	ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<pointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29
PC30         public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31         {
PC32             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33             {
PC34                 var collection = await GetCollection();
PC35                 await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36                 return pg;
PC37             }
PC38         }
PC39         private static async Task<DocumentCollection> GetCollection()
PC40         {
PC41             ...
PC42         }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PG01 public class PointGenerator
PG02 {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10    public DateTimeOffset DateLastClaimed { get; set; }
PG12    [JsonProperty("location")]
PG12    public Point Location { get; set; }
PG13 }
```

You need to trigger the EmailPlayer Azure Function when a point generator is claimed.

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

- A Insert code after line PC35 to create a Queue trigger and send a queue message.
- B Create a trigger based on the primary data store.
- C Create a trigger based on the reporting data store.
- D Insert code after line PC14 to create a Service Bus trigger and send a message.

A9.

You need to trigger the EmailPlayer Azure Function when a point generator is claimed.

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

- A Insert code after line PC35 to create a Queue trigger and send a queue message.
- B Create a trigger based on the primary data store.
- C Create a trigger based on the reporting data store.
- D Insert code after line PC14 to create a Service Bus trigger and send a message.

Answer: D

A10.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<PointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29
PC30         public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31         {
PC32             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33             {
PC34                 var collection = await GetCollection();
PC35                 await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36                 return pg;
PC37             }
PC38         }
PC39         private static async Task<DocumentCollection> GetCollection()
PC40         {
PC41             ...
PC42         }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PG01 public class PointGenerator
PG02 {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10    public DateTimeOffset DateLastClaimed { get; set; }
PG12    [JsonProperty("location")]
PG12    public Point Location { get; set; }
PG13 }
```

You need to create the Azure Search index.

How should you configure the Azure Search index? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

FIELD NAME	TYPE	RETRIEVABLE	FILTERABLE	SORTABLE
currentOwner	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
sponsor	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dateLastClaimed	Edm.DateTimeOffset	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

A10.

FIELD NAME	TYPE	RETRIEVABLE	FILTERABLE	SORTABLE
currentOwner	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
sponsor	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dateLastClaimed	Edm.DateTimeOffset	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Q11.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for Fabrikam, a company that specializes in payment processing. Fabrikam is developing a solution to process payments for various events, such as music concerts. You develop an ASP.NET MVC website that is hosted in Azure to support an upcoming music concert. The music concert is expected to generate a large volume of ticket sales in a short amount of time.

The website uploads information to an Azure storage queue. A worker role in Azure retrieves information from the queue and generates the concert tickets in a PDF file format after the financial transaction is approved.

You observe a delay between the time the website adds a message to a queue and the time it becomes available to read from the queue. After examining the queue, you determine that no queue messages have a DequeueCount value greater than zero. The website does not throw any errors.

Business Requirements

Payments

The music concert website must be able to submit event payment information for processing. The website must remain responsive while submitting payment information. Customers must be able to add notes about their orders to a free-form control on the website. These notes must be submitted with the payment when the customer submits an order.

Customers often enter notes that exceed 7 KB in size.

Technical Requirements

Payment Submission and Processing

Event payment information must be sent from the website to a Windows Communication Foundation (WCF) service worker role. The worker role must submit the information to the payment processor in JSON format.

Payment Processing

You have the following payment processing requirements:

- If the number of messages in a queue goes above or below a specified threshold, worker role instances must be created or deleted as needed. This process must be completed by using the least amount of effort. It must be easy to reconfigure role instance thresholds.
- Payments must be retrieved from the queue in the maximum batch sizes that are allowed by the queue and pulled from the queue for 5 minutes.
- The payment queue must not be re-created when processing payments.
- During single Payment processing, the number of tickets available for an event must be updated. The update operation must be retried for 30 seconds or 5 retry attempts, whichever occurs first. Each retry should pause for at least two seconds and for one second longer than the previous attempt. If the update fails, the payment should be placed in the poison queue.

Storage

You have the following storage requirements:

- Payment information must be stored by using Azure Queue storage. Connection to the Azure storage account has been established in a configured setting named StorageConnectionString, which is configured for the web and worker roles.
- A payment processing queue and a poison payment queue must be used when processing payments.
- Azure Queue message content must be XML-safe and UTF-8 encoded.
- An Azure storage account must be established for diagnostic information in a configured setting named DiagnosticsConnectionString, which is configured for both the web and worker roles.

Security and Monitoring

Security

The web role must be secured by using HTTPS.

Monitoring

You must collect diagnostic data for both the web and worker roles by using the Diagnostics module. Diagnostics configuration changes must not require the code of the roles to be rebuilt. The diagnostic data is used for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis and capacity planning, and auditing.

Performance testing must evaluate the roles under normal and stress conditions without incurring changes for running Azure. Memory allocation, function time, and multithreading concurrency issues must be evaluated.

Deployment

You purchase a custom domain name fabrikamfunding.com to host the website, web role, and worker roles. You must deploy an HTTPS certificate with the web role, and you must update associated configuration files accordingly.

Web role and worker role instance sizes must be specified as Medium. You must deploy one web role instance named FabrikamFundingPaymentGenerator, and worker role instances named FabrikamFundingPaymentProcessor.

Application Structure

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

CustomRetryPolicy.cs

```
CR01 public class CustomRetryPolicy : IRetryPolicy
CR02 {
CR03     int _retryCount = 0;
CR04     readonly TimeSpan _baseInterval= TimeSpan.FromSeconds(1);
CR05     readonly string _poisonPaymentQueueName;
CR06     private readonly CloudQueueClient _queueClient;
CR07     private readonly EventPayment _eventPayment;
CR08     public CustomRetryPolicy(string poisonPaymentQueueName, CloudQueueClient
queueClient, EventPayment eventPayment)
CR09     {
CR10         _poisonPaymentQueueName = poisonPaymentQueueName;
CR11         _queueClient = queueClient;
CR12         _eventPayment = eventPayment;
CR13     }
CR14     public IRetryPolicy CreateInstance()
CR15     {
CR16         return new CustomRetryPolicy(_poisonPaymentQueueName, _queueClient,
_eventPayment);
CR17     }
CR18 }
```

Event.cs

```
EV01 public class Event : TableEntity
EV02 {
EV03     public int AvailableTickets { get; set; }
EV04 }
```

EventPayment.cs

```
EP01 [DataContract]
EP02 public class EventPayment
EP03 {
EP04     [DataMember]
EP05     public int EventId { get; set; }
EP06     [DataMember]
EP07     public string Email { get; set; }
EP08     [DataMember]
EP09     public string Notes { get; set; }
EP10     [DataMember]
EP11     public int TicketCount { get; set; }
EP12     [DataMember]
EP13     public DateTime OrderDate { get; set; }
EP14     [DataMember]
EP15     public Guid EventPaymentId { get; set; }
EP16 }
```

QueueManager.cs

```
QM01 public class QueueManager
QM02 {
QM03     private readonly CloudQueueClient _queueClient;
QM04     private readonly CloudTableClient _tableClient;
QM05     private const string PaymentQueueName = "paymentqueue";
QM06     private const string PoisonPaymentQueueName = "poisonpaymentqueue";
QM07     public QueueManager()
QM08     {
QM09         var storageAccount = CloudStorageAccount.Parse(
QM10             CloudConfigurationManager.GetSetting("StorageConnectionString"));
QM11         _queueClient = storageAccount.CreateCloudQueueClient();
QM12         _tableClient = storageAccount.CreateCloudTableClient();
QM13     }
QM14     public async Task SendMessageAsync(EventPayment eventPayment)
QM15     {
QM16         ...
QM17     }
QM18     public async Task ProcessMessagesAsync()
QM19     {
QM20         ...
QM21     }
QM22     public async Task ProcessPayment(EventPayment eventPayment)
QM23     {
QM24         var events = _tableClient.GetTableReference("events");
QM25         var key = eventPayment.EventId.ToString();
QM26         var operation = await events.ExecuteAsync(TableOperation.Re-
trieve<Event> (key, key));
QM27         var @event = operation.Result as Event;
QM28         @event.AvailableTickets = @event.AvailableTickets - eventPay-
ment.TicketCount;
QM29         var requestOptions = new TableRequestOptions
QM30         {
QM31             RetryPolicy = new CustomRetryPolicy(
QM32                 PoisonPaymentQueueName,
QM33                 _queueClient,
QM34                 eventPayment),
QM35         };
QM36         var context = new OperationContext
QM37         {
QM38             StartTime = DateTime.Now,
QM39         };
QM40         await events.ExecuteAsync(TableOperation.Replace(@event),
requestOptions, context);
QM41     }
QM42 }
```

The **SendMessageAsync** method of the **QueueManager** class occasionally throws errors.

You need to correct the errors.

What should you do?

- A Remove all attributes from the **EventPayment** class.
- B Encode the **notes** field content by using UTF-32 encoding.
- C Update the **notes** field to a byte array. Binary encode and decode the **notes** content when sending or receiving an **EventPayment** class.
- D Update the **SendMessageAsync** method of the **QueueManager** class to store the notes field in BLOB storage. Update the **EventPayment** class to store the BLOB uniform resource identifier (URI). Extract the **notes** BLOB information by using the BLOB URI in the **ProcessMessagesAsync** method of the **QueueManager** class.

A11.

The **SendMessageAsync** method of the **QueueManager** class occasionally throws errors.

You need to correct the errors.

What should you do?

- A Remove all attributes from the **EventPayment** class.
- B Encode the **notes** field content by using UTF-32 encoding.
- C Update the **notes** field to a byte array. Binary encode and decode the **notes** content when sending or receiving an **EventPayment** class.
- D Update the **SendMessageAsync** method of the **QueueManager** class to store the notes field in BLOB storage. Update the **EventPayment** class to store the BLOB uniform resource identifier (URI). Extract the **notes** BLOB information by using the BLOB URI in the **ProcessMessagesAsync** method of the **QueueManager** class.

Answer: D

Q12.

You create a web application. You publish the source code of the web application to a GitHub repository by using Microsoft Visual Studio. You create an Azure Web App by using the Azure management portal.

You must continuously deploy the web application from the GitHub repository website to the Azure Web App.

You need to deploy the source code of the web application.

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

Actions	Answer Area
Select the repository and the branch from which to deploy the Azure Web App.	
Select GitHub as the source control method.	
Configure the Azure Web App to use the Always On option.	
In the Azure management portal, configure web endpoint monitoring.	
In the Azure management portal, choose the option to set up deployment from source control.	
Sign in to GitHub by using your deployment credentials.	

A12.

Actions	Answer Area
Select the repository and the branch from which to deploy the Azure Web App.	<hr/> In the Azure management portal, choose the option to set up deployment from source control.
Select GitHub as the source control method.	<hr/> Select GitHub as the source control method.
Configure the Azure Web App to use the Always On option.	<hr/> Sign in to GitHub by using your deployment credentials.
In the Azure management portal, configure web endpoint monitoring.	<hr/> Select the repository and the branch from which to deploy the Azure Web App.
In the Azure management portal, choose the option to set up deployment from source control.	
Sign in to GitHub by using your deployment credentials.	

Q13.

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

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

You deploy a Virtual Machine Scale Set (VMSS) named CorpWebVMSS to Azure by using Azure PowerShell and set the instance count to 1. The VMSS includes a storage account, load balancer, public IP address, and six Standard_A1 Windows virtual machines (VMs) that run Internet Information Services (IIS). All components are deployed to a resource group named CorpWebRG.

You must increase the instance count to support the increased load on IIS.

You need to manually scale out the number of VMs in the scale set to 5.

Solution: You run the following Azure PowerShell commands:

```
$vmss = Get-AzureRmVmss -ResourceGroupName CorpWebRG -VMScalesSetName CorpWebVMSS  
$vmss.Sku.Capacity = 5  
Update-AzureRmVmss -ResourceGroupName CorpWebRG -Name CorpWebVMSS -VirtualMachineScaleSet $vmss
```

Does the solution meet the goal?

- A Yes
- B No

A13.

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

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

You deploy a Virtual Machine Scale Set (VMSS) named CorpWebVMSS to Azure by using Azure PowerShell and set the instance count to 1. The VMSS includes a storage account, load balancer, public IP address, and six Standard_A1 Windows virtual machines (VMs) that run Internet Information Services (IIS). All components are deployed to a resource group named CorpWebRG.

You must increase the instance count to support the increased load on IIS.

You need to manually scale out the number of VMs in the scale set to 5.

Solution: You run the following Azure PowerShell commands:

```
$vmss = Get-AzureRmVmss -ResourceGroupName CorpWebRG -VMScalesSetName CorpWebVMSS  
$vmss.Sku.Capacity = 5  
Update-AzureRmVmss -ResourceGroupName CorpWebRG -Name CorpWebVMSS -VirtualMachineScaleSet $vmss
```

Does the solution meet the goal?

- A Yes
- B No

Answer: B

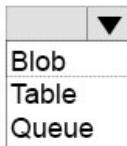
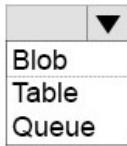
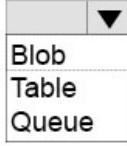
Q14.

You administer an Azure environment that contains multiple virtual machines (VMs).

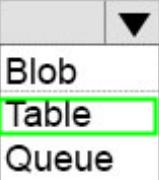
You need to view and retrieve diagnostic logs for all VMs.

Which storage type should you use for each data source? To answer, select the appropriate options in the answer area.

Answer Area

Data source	Storage type
Azure logs	 Blob Table Queue
IIS 7.0 logs	 Blob Table Queue
Windows Event logs	 Blob Table Queue

A14.

Data source	Storage type
Azure logs	 Blob Table Queue
IIS 7.0 logs	 Blob Table Queue
Windows Event logs	 Blob Table Queue

Q15.

Background:

You are a developer for ProseWare Inc., a software-as-a-service (SaaS) company that provides a comment system that websites use to allow for end users to post comments associated with a webpage or topic on a customer's website.

Business requirements

Moderation:

The moderation of comments is a feature of the software, and usually involves the editing of a comment.

Only users who have accounts in a group in Azure Active Directory (Azure AD) have the ability moderate. External users can also become moderators, but only by explicit invitation.

Any moderation action must include the name of the moderator.

Comment navigation:

Each comment is identified by a unique string consisting of a random string of characters.

Within the body of a comment, internal links to other comment threads can be specified using the link format: "/<parent comment id> / <child comment id>"

Comment search:

Comments can be searched using Azure Search. Searches must do the following:

- Searching for email addresses must match email addresses in comments.
- Searching must work for the client's language.
- Internal links to other comments using the link format should be searched.

Content screening:

Comment content is screened for inappropriate language, length, and topic using content analysis. Content must be screened, but can appear prior to be screened.

Mobile App:

The moderation functionality can be accessed using a Universal Windows Platform (UWP) app named ProsewareApp. The app includes functionality that notifies moderators when changes are made to a comment they modified.

Export:

Customers can perform an export of all comments to a customer supplied Microsoft OneDrive folder on demand. The export functionality is implemented as an Azure Logic App, and it must be able to be triggered by the customer from their local network.

Interaction agents:

Interaction agents are parts of the system that interact with comment threads. The main purpose is to modify a comment's body based on the contents of the comment. For example, one of the agents is WikiAgent, which adds links to Wikipedia articles when it sees text in the comment body that exactly matches a Wikipedia article title. Interaction Agents are implemented in Service Fabric.

Interaction agents must meet the following requirements:

- Only successfully process each comment once
- Any errors encountered during the processing of a comment should be retried
- Must run on systems that allow for custom applications to be installed
- Must run in a VNet or private network space
- Must be run on a system that can scale up and down based on demand
- A single user's usage of Interaction Agents must not impact other users' usage of Interaction Agents

Technical requirements

Authentication:

ProseWare Inc. allows for user authentication through Azure AD and Twitter.

Storage:

The application runs as a Web App on Azure. Comments are stored in an Azure DocumentDB database named "Proseware".

Performance:

The product includes a service level agreement (SLA) for individual method performance. All data retrieval methods must return within 100ms 99% of the time.

API:

The ProseWare Inc. API is made available to public callers using an Azure API App. Azure AD and Twitter are the Authentication Providers.

Application structure

CommentController.cs:

```
CC01 [Route("api/[controller]")]
CC02 public class CommentController : Controller
CC03 {
CC04     private IDatabase _redis;
CC05     private DataStore _dataStore;
CC06     private CloudQueue _queue
CC07
CC08     public CommentController ()
CC09     {
CC10         _queue = CloudStorageAccount.Parse(" ").Create-
CloudQueueClient().GetQueueReference ("commentQueue");
CC11         _redis = ConnectionMultiplexer.Connect("...").GetData-
base();
CC12         _dataStore = new DataStore();
CC13     }
CC14
CC15     [HttpGet("{commentId}")]
CC16     public async Task<Comment> Get(string commentId)
CC17     {
CC18         var cached = await _redis.StringGetAsync(commentId);
CC19         if (cached.HasValue)
CC20         {
CC21             return JsonConvert.DeserializeObject<Com-
ment>(cached.ToString());
CC22         }
CC23         return await _dataStore.LoadAsync(commentId);
CC24     }
CC25
CC26     [HttpGet]
CC27     public IEnumerable<Comment> GetChildComments(string com-
mentId)
CC28     {
CC29         IEnumerable<Comment> result = null;
CC30
CC31         if (result == null)
CC32         {
CC33             result = _dataStore.LoadThread(commentId);
CC34         }
CC35         return results;
CC36     }
CC37
CC38     [HttpPost]
CC39     public async Task<IActionResult> New([FromBody]Comment
comment)
CC40     {
CC41         await Save(comment);
CC42         return Ok();
CC43     }
CC44
CC45     [HttpPost]
CC46     public async Task<IActionResult> Reply(string inRe-
sponseTo, [FromBody]Comment comment)
CC47     {
CC48         comment.InResponseTo = inResponseTo;
CC49         await Save(comment);
CC50         return View();
CC51     }
CC52
```

```

CC53     private static Comment Convert(string json)
CC54     {
CC55         return JsonConvert.DeserializeObject<Comment>(json);
CC56     }
CC57
CC58     private async Task Save(Comment comment, string moderator-
Name = null)
CC59     {
CC60         comment.Moderator = moderatorName;
CC61         var json = JsonConvert.SerializeObject(comment);
CC62         _redis.StringSet(comment.Id, json);
CC62
CC63
CC64         await _queue.AddMessageAsync(new CloudQueueMessage(com-
ment.Id));
CC65         _dataStore.Save(comment);
CC66     }
CC67 }
```

cleaner.csx:

```

CL01 #r "Newtonsoft.Json"
CL02
CL03 using System;
CL04 using Newtonsoft.Json;
CL05 using Newtonsoft.Json.Linq;
CL06 public static void Run(string commentId, object result,
TraceWriter log)
CL07 {
CL08     dynamic comment = JObject.Parse(item);
CL09 ...
CL10     result = comment;
CL11 }
```

ICommentAgent.cs:

```

CA01 public interface ICommentAgent: IActor
CA02     {
CA03         Task<string> ModifyCommentText(string id, string body,
string title);
CA04     }
```

WikiAgent.cs:

```

WA01 [StatePersistence(StatePersistence.Persisted)]
WA02 internal class WikiAgent : Agent, ICommentAgent
WA03 {
WA04     public WikiAgent(ActorService, ActorId id) : base(service,
id) {}
WA05     public async Task<string> ModifyCommentText(string id,
string body, string title)
WA06     {
WA07         try
WA08         {
WA09             var newBody = scanForLinks(body);
WA11
WA12             return newBody;
WA13         }
WA14         catch
WA15         {
WA16             throw
WA18         }
WA19     }
WA20 }
```

Comment.cs:

```
CO01 public class Comment
CO02 {
CO03     public string Id {get; set;}
CO04     public string UserId {get; set;}
CO05     public string InResponseTo {get; set;}
CO06     public string Title {get; set;}
CO07     public DateTimeOffset Date {get; set;}
CO08     public string Body {get; set;}
CO09     public string Moderator {get; internal; set;}
CO10 }
```

DataStore.cs:

```
DS01 public class DataStore
DS02 {
DS03     private const string EndpointUrl = "https://proseware.documents.azure.com:443/";
DS04     private const string PrimaryKey = "";
DS05     private const string db = "Proseware";
DS06     private const string col = "Comments";
DS07     private DocumentClient client;
DS08
DS09     public DataStore()
DS10     {
DS11         client = new DocumentClient(new Uri(EndpointUrl), PrimaryKey);
DS12     }
DS13
DS14     public async Task<Comment> LoadAsync(string commentId)
DS15     {
DS16         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS17         return await client.ReadDocumentAsync<Comment>(UriFactory.CreateDocumentUri(db, col, commentId));
DS18     }
DS19

DS20     public async void Save(Comment comment)
DS21     {
DS22         var uri = UriFactory.CreateDocumentCollectionUri(db,
col, comment.Id);
DS23         await client.UpsertDocumentAsync(uri, comment);
DS24     }
DS25     public IEnumerable<Comment> LoadThread(string commentId)
DS26     {
DS27         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS28         return client.CreateDocumentQuery<Comment>(uri).Where(f => f.Id == commentId);
DS29     }
}
```

MainPage.xaml.cs:

```
MP01 public sealed partial class MainPage : Page
MP02 {
MP03     public MainPage()
MP04     {
MP05         InitializeComponent();
MP06     }
MP07
MP08     private async void StartNotify()
MP09     {
MP10     }
MP11
MP12     private void UpdateUI()
MP13     {
MP14     }
MP15 }
```

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

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

You need to implement the infrastructure for the Interaction Agents.

Solution: Create a Service Fabric cluster with Bronze durability and reliability tiers.

Does the solution meet the goal?

- A Yes
- B No

A15.

You need to implement the infrastructure for the Interaction Agents.

Solution: Create a Service Fabric cluster with Bronze durability and reliability tiers.

Does the solution meet the goal?

- A Yes
- B No

Answer: A

Q16.

You deploy an application that uses a secure data storage solution to Azure. You use Redis Cache and select the Premium tier.

You have the following requirements:

- Create point-in-time snapshots of the dataset at specific intervals.
- Limit specific clients from using the cache.
- Use primary/replica cache pairs.

You need to configure the environment.

Which feature should you implement for each requirement? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer area

Requirement	Feature
Create point-in-time snapshots of the dataset at specific intervals.	<input type="checkbox"/> Redis cluster <input type="checkbox"/> Redis persistence <input type="checkbox"/> Azure Virtual Network
Limit specific clients from using the cache.	<input type="checkbox"/> Redis cluster <input type="checkbox"/> Redis persistence <input type="checkbox"/> Azure Virtual Network
Use primary/replica cache pairs.	<input type="checkbox"/> Redis cluster <input type="checkbox"/> Redis persistence <input type="checkbox"/> Azure Virtual Network

A16.

Requirement	Feature
Create point-in-time snapshots of the dataset at specific intervals.	<input type="checkbox"/> Redis cluster <input checked="" type="checkbox"/> Redis persistence <input type="checkbox"/> Azure Virtual Network
Limit specific clients from using the cache.	<input type="checkbox"/> Redis cluster <input checked="" type="checkbox"/> Redis persistence <input checked="" type="checkbox"/> Azure Virtual Network
Use primary/replica cache pairs.	<input checked="" type="checkbox"/> Redis cluster <input checked="" type="checkbox"/> Redis persistence <input type="checkbox"/> Azure Virtual Network

Q17.

You develop an Azure App Service Mobile App.

The Azure App Service must use Twitter as an authentication provider. You start by registering your application with Twitter.

You need to update your app's authentication and authorization in the Azure Portal.

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

NOTE: Each correct selection is worth one point.

- A API Key
- B Azure Active Directory (Azure AD) Bearer Token
- C JSON Web Token (JWT)
- D API Secret
- E Mobile App gateway URL

A17.

You develop an Azure App Service Mobile App.

The Azure App Service must use Twitter as an authentication provider. You start by registering your application with Twitter.

You need to update your app's authentication and authorization in the Azure Portal.

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

NOTE: Each correct selection is worth one point.

- A API Key
- B Azure Active Directory (Azure AD) Bearer Token
- C JSON Web Token (JWT)
- D API Secret
- E Mobile App gateway URL

Answer: A, D

Q18.

You develop a web application that uses table storage in Azure.

You create a storage account named Contoso that stores a table named CityPopulationData.

The web application stores entities in this table.

You need to query the table data by using OData.

Which URL should you use?

- A <http://contoso.table.core.windows.net/citypopulationdata>
- B <http://contoso.table.core.windows.net/odata/citypopulationdata>
- C <http://azurestorage.table.core.windows.net/contoso>
- D <http://microsoft.table.core.windows.net/contoso>
- E <http://azure.table.core.windows.net/contoso/citypopulationdata>

A18.

You develop a web application that uses table storage in Azure.

You create a storage account named Contoso that stores a table named CityPopulationData.

The web application stores entities in this table.

You need to query the table data by using OData.

Which URL should you use?

- A <http://contoso.table.core.windows.net/citypopulationdata>
- B <http://contoso.table.core.windows.net/odata/citypopulationdata>
- C <http://azurestorage.table.core.windows.net/contoso>
- D <http://microsoft.table.core.windows.net/contoso>
- E <http://azure.table.core.windows.net/contoso/citypopulationdata>

Answer: A

Q19.

Background

Contoso, Ltd. is developing a patient monitoring solution for a hospital. The solution consists of an Azure Web App and a set of mobile applications that health care providers use to monitor patients remotely.

Monitoring devices that run the embedded version of Windows will be attached to patients. The devices will collect information from patients and will transmit real-time continuous data to a service that runs on Azure. The service collects and distributes data. The data that the service provides must be accessible by the website and by the mobile applications.

Business Requirements

Patients

All patient data must be stored securely on Azure. Data security must meet or exceed Health Insurance Portability and Accountability Act of 1996 (HIPAA) standards in the United States and must meet or exceed ISO/IEC 27002 data security standards in the rest of the world.

Contractors

Third-party contractors will develop the mobile applications. All contractors must develop the applications by using virtual machines (VMs) that are hosted on Azure. Only authorized contractors and authorized IP addresses are permitted to access the VMs. The contractors can use Near Field Communication (NFC) tags to launch Remote Desktop (RD) connections to the VMs from NFC-enabled devices. For testing purposes, contractors must be able to run multiple instances of mobile applications within the VMs.

Data Collection and Distribution Service

The service must monitor the patient data and send out alerts to health care providers when specific conditions are detected. The service must send the alerts to mobile applications and to the website in real time so that doctors, nurses, and caregivers can attend to the patient. Partner organizations and diagnostic laboratories must be able to securely access the data and the website from remote locations.

Current Issues

A partner that is testing a prototype of the website reports that after signing in to the website, the partner is redirected to the settings page instead of to the home page. The data from the patient devices is slow to appear on the website and does not always appear. All patient devices online have active connections to the data collection service.

Technical Requirements

Contractors

All contractors will use virtual machines that are initially configured as size A3. Contractors must sign in to the assigned VM by using IP addresses from a list of preapproved addresses.

Data Collection and Distribution Service

- The service runs Node.js in a worker role.
- The service must use at least 2048-bit encryption and must use port 8888.
- All patient information must be encrypted and stored by using a NoSQL data store.
- Data must be stored and retrieved securely by using RESTful endpoints.
- Data must NOT be stored within a virtual machine.

All deployed services must send an alert email to watchguard@contoso.com when any of the following conditions is met:

- The CPU Percentage metric is at or above 85 percent for at least 10 minutes.
- The Network In metric is at or above 2 KB for at least 10 minutes.
- The Network Out metric is at or above 2 KB for at least 10 minutes.
- The Disk Write metric is at or above 1 KB/sec for at least 30 minutes.
- The Disk Read metric is at or above 1 KB/sec for at least 30 minutes.

Website and Mobile Devices

The website must be secure and must be accessible only within the hospital's physical grounds. All mobile applications and websites must be responsive. All websites must produce error logs that can be viewed remotely.

Virtual Machines

- All Azure instances must be deployed and tested on staging instances before they are deployed to production instances.
- All deployed instances must scale up to the next available CPU instance at a CPU usage threshold of 90 percent and scale down when the usage is below 10 percent.

Application Structure

Relevant portions of the application files are shown in the following code segments. Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

ControllerFile.cs:

```
CF01  using System;
CF02  using System.Collections.Generic;
CF03  using System.Linq;
CF04  using System.Web;
CF05  using System.Web.Mvc;
CF06  namespace WebApplication1.Controllers
CF07  {
CF08      public class HomeController : Controller
CF09      {
CF10          public ActionResult Index()
CF11          {
CF12              ViewBag.Message = "Welcome to Contoso Patient Monitor.";
CF13
CF14              return View();
CF15          }
CF16          ...
CF17      }
CF18  }
```

Web.config

```
WC01 <?xml version="1.0" encoding="utf-8"?>
WC02  <configuration>
WC03      <appSettings>
WC04          <add key="webpages:Version" value="3.0.0.0" />
WC05          <add key="webpages:Enabled" value="false" />
WC06          <add key="ClientValidationEnabled" value="true" />
WC07          <add key="UnobtrusiveJavaScriptEnabled" value="true" />
WC08
WC09      </appSettings>
WC10      <system.web>
WC11          <authentication mode="None" />
WC12          <compilation debug="true" targetFramework="4.5" />
WC13          <httpRuntime targetFramework="4.5" />
WC14
WC15      </system.web>
WC16  </configuration>
```

The Web App does not receive alerts quickly enough.

There is a lengthy delay between the time an alert is sent and when it is received by the Web App.

You need to resolve the issue.

What should you do?

- A Increase the amount of swap memory for the VM instance,
- B Enable automatic scaling for the Web App.
- C Decrease the instance count for the worker role.
- D Enable automatic scaling for the worker role.
- E Set the monitoring level to Verbose for the worker role.

A19.

The Web App does not receive alerts quickly enough.

There is a lengthy delay between the time an alert is sent and when it is received by the Web App.

You need to resolve the issue.
What should you do?

- A Increase the amount of swap memory for the VM instance,
- B Enable automatic scaling for the Web App.
- C Decrease the instance count for the worker role.
- D Enable automatic scaling for the worker role.
- E Set the monitoring level to Verbose for the worker role.

Answer: C

Explanation:

From scenario: The data collection service runs Node.js in a worker role.

All deployed instances must scale up to the next available CPU instance at a CPU usage threshold of 90 percent and scale down when the usage is below 10 percent.

Q20.

You plan to connect a customer's on-premises infrastructure to Azure. You have several connections available.

You have the following requirements:

- All connections must be secure.
- All on-premises solutions must support hybrid functionality.

You need to recommend connectivity solutions.

Which solutions should you recommend? To answer, drag the appropriate connection strategy to the correct connection. Each connection strategy may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Connection strategies

Existing WAN network
Windows built-in virtual private network (VPN) client
Industry standard IPsec virtual private network (VPN)
External-facing IPv6 address

Answer Area

Secure cross-premises connection	Connection strategy
Site-to-site VPN	Connection strategy
Point-to-site VPN	Connection strategy
Azure ExpressRoute	Connection strategy

A20.

Connection strategies

Existing WAN network
Windows built-in virtual private network (VPN) client
Industry standard IPsec virtual private network (VPN)
External-facing IPv6 address

Answer Area

Secure cross-premises connection	Connection strategy
Site-to-site VPN	Industry standard IPsec virtual private network (VPN)
Point-to-site VPN	Windows built-in virtual private network (VPN) client
Azure ExpressRoute	Existing WAN network

Q21.

A company is designing a new web-based software application that must be highly available and resistant. Which of the following is the BEST environment for the application?

- A The primary instance of the application will be locally hosted with a weekly copy of the instance sent to a cloud service provider.
- B The primary instance of the application will be locally hosted with a nightly file-level backup being performed to an off-site location.
- C The primary instance of the application will be running a cloud service provider's hosted environment with a continuous backup to the company's local infrastructure.
- D The primary instance of the application will be locally hosted with a nightly copy of the instance sent to a client service provider.

A21.

A company is designing a new web-based software application that must be highly available and resistant. Which of the following is the BEST environment for the application?

- A The primary instance of the application will be locally hosted with a weekly copy of the instance sent to a cloud service provider.
- B The primary instance of the application will be locally hosted with a nightly file-level backup being performed to an off-site location.
- C The primary instance of the application will be running a cloud service provider's hosted environment with a continuous backup to the company's local infrastructure.
- D The primary instance of the application will be locally hosted with a nightly copy of the instance sent to a client service provider.

Answer: C

Q22.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to Azure datacenters on two continents.

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Traffic Manager and Application Gateway.

Does the solution meet the goal?

- A Yes
 B No

A22.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to Azure datacenters on two continents.

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Traffic Manager and Application Gateway.

Does the solution meet the goal?

- A Yes
 B No

Answer: A

Explanation:

Application Gateway works at the application layer (Layer 7 in the OSI network reference stack). It acts as a reverse-proxy service, terminating the client connection and forwarding requests to back-end endpoints. It supports the HTTP, HTTPS, and WebSockets protocols.

Application Gateway is useful for applications that require requests from the same user/client session to reach the same back-end virtual machine. Examples of these applications would be shopping cart applications and web mail servers.

Traffic Manager works at the DNS level. It uses DNS responses to direct end-user traffic to globally distributed endpoints. Clients then connect to those endpoints directly.

Microsoft Azure Traffic Manager allows you to control the distribution of user traffic for service endpoints in different datacenters

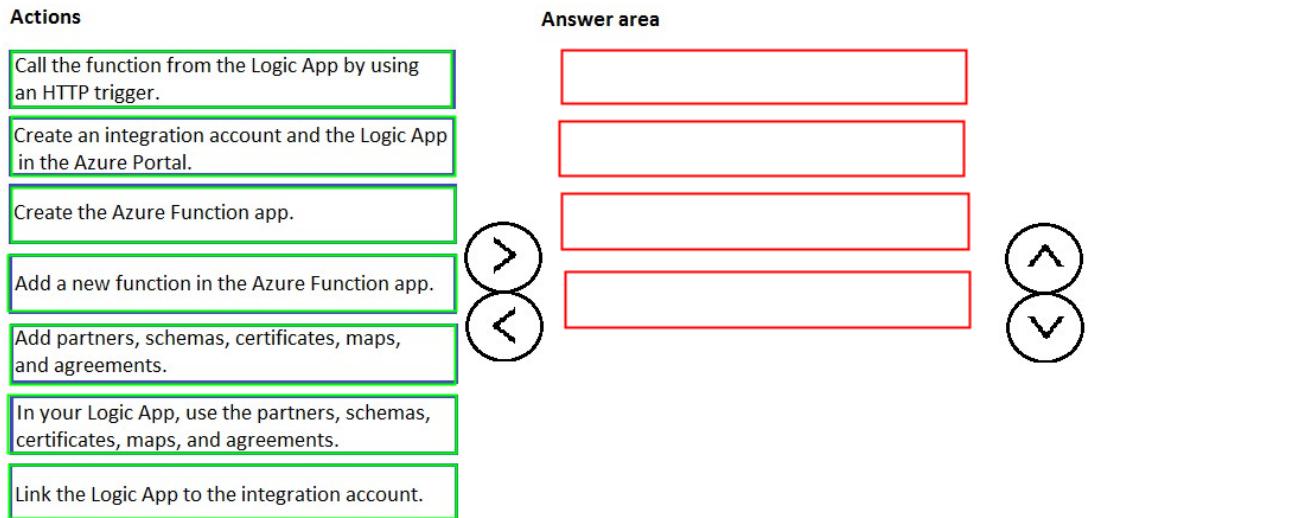
<https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview>">References:
<https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview>

Q23.

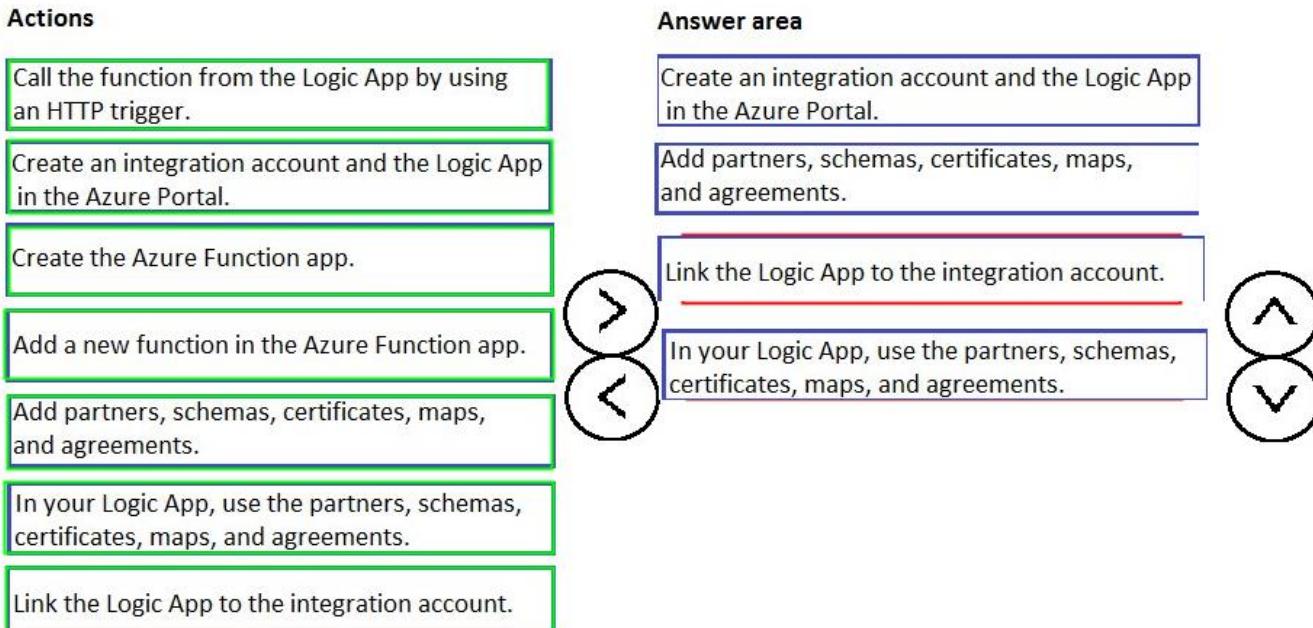
You are developing a business-to-business (B2B) solution by using an Azure Logic App. You plan to use the Enterprise Integration Pack to allow the exchange of the X12 industry standard message format within your Logic App workflow. You start by creating a new Azure Resource Manager (ARM) resource group and Azure App Service plan.

You need to create the B2B solution.

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



A23.



Q24.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to Azure datacenters on two continents.

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Traffic Manager.

Does the solution meet the goal?

- A Yes
 B No

A24.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Traffic Manager.

Does the solution meet the goal?

- A Yes
 B No

Answer: B

Explanation:

A Traffic Manager works at the DNS level. It uses DNS responses to direct end-user traffic to globally distributed endpoints. Clients then connect to those endpoints directly.

An application manager, which works at the Application level (Layer 7), is also required.

Q25.

You create a new web application by using a single Azure website deployment. The deployment uses the shared web hosting plan. User activity varies significantly and unpredictably.

The application must automatically scale to a maximum of eight virtual machines based on CPU utilization.

You need to configure the environment.

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

Actions	Answer Area
Change the value of the web hosting plan to Standard .	
Configure autoscaling to support scaling by metrics based on CPU utilization.	
Enable the Scale by Metric option.	
Configure autoscaling to None .	
Change the value of the web hosting plan to Basic .	

A25.

Answer:

Actions	Answer Area
Change the value of the web hosting plan to Standard .	Change the value of the web hosting plan to Standard .
Configure autoscaling to support scaling by metrics based on CPU utilization.	Enable the Scale by Metric option.
Enable the Scale by Metric option.	
Configure autoscaling to None .	Configure autoscaling to support scaling by metrics based on CPU utilization.
Change the value of the web hosting plan to Basic .	

Q26.

Background:

You are a developer for ProseWare Inc., a software-as-a-service (SaaS) company that provides a comment system that websites use to allow for end users to post comments associated with a webpage or topic on a customer's website.

Business requirements

Moderation:

The moderation of comments is a feature of the software, and usually involves the editing of a comment.

Only users who have accounts in a group in Azure Active Directory (Azure AD) have the ability moderate. External users can also become moderators, but only by explicit invitation.

Any moderation action must include the name of the moderator.

Comment navigation:

Each comment is identified by a unique string consisting of a random string of characters.

Within the body of a comment, internal links to other comment threads can be specified using the link format: "<parent comment id> / <child comment id>"

Comment search:

Comments can be searched using Azure Search. Searches must do the following:

- Searching for email addresses must match email addresses in comments.
- Searching must work for the client's language.
- Internal links to other comments using the link format should be searched.

Content screening:

Comment content is screened for inappropriate language, length, and topic using content analysis. Content must be screened, but can appear prior to be screened.

Mobile App:

The moderation functionality can be accessed using a Universal Windows Platform (UWP) app named ProsewareApp. The app includes functionality that notifies moderators when changes are made to a comment they modified.

Export:

Customers can perform an export of all comments to a customer supplied Microsoft OneDrive folder on demand. The export functionality is implemented as an Azure Logic App, and it must be able to be triggered by the customer from their local network.

Interaction agents:

Interaction agents are parts of the system that interact with comment threads. The main purpose is to modify a comment's body based on the contents of the comment. For example, one of the agents is WikiAgent, which adds links to Wikipedia articles when it sees text in the comment body that exactly matches a Wikipedia article title. Interaction Agents are implemented in Service Fabric.

Interaction agents must meet the following requirements:

- Only successfully process each comment once
- Any errors encountered during the processing of a comment should be retried
- Must run on systems that allow for custom applications to be installed
- Must run in a VNet or private network space
- Must be run on a system that can scale up and down based on demand
- A single user's usage of Interaction Agents must not impact other users' usage of Interaction Agents

Technical requirements

Authentication:

ProseWare Inc. allows for user authentication through Azure AD and Twitter.

Storage:

The application runs as a Web App on Azure. Comments are stored in an Azure DocumentDB database named "Proseware".

Performance:

The product includes a service level agreement (SLA) for individual method performance. All data retrieval methods must return within 100ms 99% of the time.

API:

The ProseWare Inc. API is made available to public callers using an Azure API App. Azure AD and Twitter are the Authentication Providers.

Application structure

CommentController.cs:

```
CC01 [Route("api/[controller]")]
CC02 public class CommentController : Controller
CC03 {
CC04     private IDatabase _redis;
CC05     private DataStore _dataStore;
CC06     private CloudQueue _queue
CC07
CC08     public CommentController ()
CC09     {
CC10         _queue = CloudStorageAccount.Parse(" ").Create-
CloudQueueClient().GetQueueReference ("commentQueue");
CC11         _redis = ConnectionMultiplexer.Connect("...").GetData-
base();
CC12         _dataStore = new DataStore();
CC13     }
CC14
CC15     [HttpGet("{commentId}")]
CC16     public async Task<Comment> Get(string commentId)
CC17     {
CC18         var cached = await _redis.StringGetAsync(commentId);
CC19         if (cached.HasValue)
CC20         {
CC21             return JsonConvert.DeserializeObject<Com-
ment>(cached.ToString());
CC22         }
CC23         return await _dataStore.LoadAsync(commentId);
CC24     }
CC25
CC26     [HttpGet]
CC27     public IEnumerable<Comment> GetChildComments(string com-
mentId)
CC28     {
CC29         IEnumerable<Comment> result = null;
CC30
CC31         if (result == null)
CC32         {
CC33             result = _dataStore.LoadThread(commentId);
CC34         }
CC35         return results;
CC36     }
CC37
CC38     [HttpPost]
CC39     public async Task<IActionResult> New([FromBody]Comment
comment)
CC40     {
CC41         await Save(comment);
CC42         return Ok();
CC43     }
CC44
CC45     [HttpPost]
CC46     public async Task<IActionResult> Reply(string inRe-
sponseTo, [FromBody]Comment comment)
CC47     {
CC48         comment.InResponseTo = inResponseTo;
CC49         await Save(comment);
CC50         return View();
CC51     }
CC52
CC53     private static Comment Convert(string json)
CC54     {
CC55         return JsonConvert.DeserializeObject<Comment>(json);
CC56     }
CC57
```

```

CC58     private async Task Save(Comment comment, string moderator-
Name = null)
CC59    {
CC60        comment.Moderator = moderatorName;
CC61        var json = JsonConvert.SerializeObject(comment);
CC62        _redis.StringSet(comment.Id, json);
CC63
CC64        await _queue.AddMessageAsync(new CloudQueueMessage(com-
ment.Id));
CC65        _dataStore.Save(comment);
CC66    }
CC67 }

```

cleaner.csx:

```

CL01 #r "Newtonsoft.Json"
CL02
CL03 using System;
CL04 using Newtonsoft.Json;
CL05 using Newtonsoft.Json.Linq;
CL06 public static void Run(string commentId, object result,
TraceWriter log)
CL07 {
CL08     dynamic comment = JObject.Parse(item);
CL09 ...
CL10     result = comment;
CL11 }

```

ICommentAgent.cs:

```

CA01 public interface ICommentAgent: IActor
CA02 {
CA03     Task<string> ModifyCommentText(string id, string body,
string title);
CA04 }

```

WikiAgent.cs:

```

WA01 [StatePersistence(StatePersistence.Persisted)]
WA02 internal class WikiAgent : Agent, ICommentAgent
WA03 {
WA04     public WikiAgent(ActorService service, ActorId id) : base(service,
id) {}
WA05     public async Task<string> ModifyCommentText(string id,
string body, string title)
WA06 {
WA07     try
WA08     {
WA09
WA10         var newBody = scanForLinks(body);
WA11
WA12         return newBody;
WA13     }
WA14     catch
WA15     {
WA16
WA17         throw;
WA18     }
WA19 }
WA20 }

```

Comment.cs:

```

CO01 public class Comment
CO02 {
CO03     public string Id {get; set;}
CO04     public string UserId {get; set;}
CO05     public string InResponseTo {get; set;}
CO06     public string Title {get; set;}
CO07     public DateTimeOffset Date {get; set;}
CO08     public string Body {get; set;}
CO09     public string Moderator {get; internal; set;}
CO10 }

```

DataStore.cs:

```
DS01 public class DataStore
DS02 {
DS03     private const string EndpointUrl = "https:
//proseware.documents.azure.com:443/";
DS04     private const string PrimaryKey = "";
DS05     private const string db = "Proseware";
DS06     private const string col = "Comments";
DS07     private DocumentClient client;
DS08
DS09     public DataStore()
DS10     {
DS11         client = new DocumentClient(new Uri(EndpointUrl), Pri-
maryKey);
DS12     }
DS13
DS14     public async Task<Comment> LoadAsync(string commentId)
DS15     {
DS16         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS17         return await client.ReadDocumentAsync<Comment>(UriFac-
tory.CreateDocumentUri(db, col, commentId));
DS18     }
DS19
DS20     public async void Save(Comment comment)
DS21     {
DS22         var uri = UriFactory.CreateDocumentCollectionUri(db,
col, comment.Id);
DS23         await client.UpsertDocumentAsync(uri, comment);
DS24     }
DS25     public IEnumerable<Comment> LoadThread(string commentId)
DS26     {
DS27         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS28         return client.CreateDocumentQuery<Comment>(uri).Where(f
=> f.Id == commentId);
DS29     }
}
}
```

MainPage.xaml.cs:

```
MP01 public sealed partial class MainPage : Page
MP02 {
MP03     public MainPage()
MP04     {
MP05         InitializeComponent();
MP06     }
MP07
MP08     private async void StartNotify()
MP09     {
MP10     }
MP11
MP12     private void UpdateUI()
MP13     {
MP14     }
MP15 }
```

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

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

You need to ensure that moderators can be added to the system.

Which authentication approach should you use?

- A Microsoft Office 365 directory
- B Azure AD self-service signup
- C Azure AD Organizational Units (OU)
- D Active Directory Federation

A26.

You need to ensure that moderators can be added to the system.

Which authentication approach should you use?

- A Microsoft Office 365 directory
- B Azure AD self-service signup
- C Azure AD Organizational Units (OU)
- D Active Directory Federation

Answer: C

Q27.

You are developing an Azure-hosted application that processes request messages for multiple office locations. You create an Azure Service Bus topic named Requests. The topic has a maximum size of 5 gigabytes (GB) and a default message time to live (TTL) of 5 minutes. You also create subscriptions named PriorityRequest and StandardRequest and include appropriate logic to route the messages.

Users report that the application has not processed messages from PriorityRequest in several days.

You need to retrieve the number of messages in the PriorityRequest subscription.

Which metric Should you use?

- A Subscription Length
- B Subscription Incoming Requests
- C Topic Incoming Messages
- D Topic Size

A27.

You are developing an Azure-hosted application that processes request messages for multiple office locations. You create an Azure Service Bus topic named Requests. The topic has a maximum size of 5 gigabytes (GB) and a default message time to live (TTL) of 5 minutes. You also create subscriptions named PriorityRequest and StandardRequest and include appropriate logic to route the messages.

Users report that the application has not processed messages from PriorityRequest in several days.

You need to retrieve the number of messages in the PriorityRequest subscription.

Which metric Should you use?

- A Subscription Length
- B Subscription Incoming Requests
- C Topic Incoming Messages
- D Topic Size

Answer: D

Q28.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a web app that is deployed to Azure.

You need to download a compressed collection of the diagnostic logs.

What should you use?

- A Azure PowerShell
- B File Transfer Protocol (FTP)
- C Application Insights
- D Microsoft Visual Studio

A28.

You have a web app that is deployed to Azure.

You need to download a compressed collection of the diagnostic logs.

What should you use?

- A Azure PowerShell
- B File Transfer Protocol (FTP)
- C Application Insights
- D Microsoft Visual Studio

Answer: A

Explanation:

Diagnostic information stored to the web app file system can be accessed directly using FTP.

It can also be downloaded as a Zip archive using Azure PowerShell or the Azure Command-Line Interface.

Q29.

You are building an ASP.NET Azure Web App that is built from source code on GitHub. Automatic deployment is used for integration testing. The web.config file has settings that are updated during development deployments by using a TransformXml MSBuild task.

The settings in the web.config must be set to specific values during integration testing.

You need to ensure that the web.config is updated when the Web App is deployed to Azure.

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

- A In Azure, add an app setting namedSCM_BUILD_ARGSwith the value/p:Environment=Integration.
- B Add the integration settings and values to the ServiceDefinition.csdef and ServiceConfiguration.csfg files.
- C In Azure, create a new deployment slot namedIntegration.
- D Create an XML Document Transform (XDT) file namedweb.Integration.configthat converts the values to the integration test values.
- E In Azure, add a tag with the keyEnvironmentand the valueIntegration.

A29.

You are building an ASP.NET Azure Web App that is built from source code on GitHub. Automatic deployment is used for integration testing. The web.config file has settings that are updated during development deployments by using a TransformXml MSBuild task.

The settings in the web.config must be set to specific values during integration testing.

You need to ensure that the web.config is updated when the Web App is deployed to Azure.

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

- A In Azure, add an app setting namedSCM_BUILD_ARGSwith the value/p:Environment=Integration.
- B Add the integration settings and values to the ServiceDefinition.csdef and ServiceConfiguration.csfg files.
- C In Azure, create a new deployment slot namedIntegration.
- D Create an XML Document Transform (XDT) file namedweb.Integration.configthat converts the values to the integration test values.
- E In Azure, add a tag with the keyEnvironmentand the valueIntegration.

Answer: C, D

Q30.

Virtual Desktop Interface (VDI) will present challenges for the network administrator as they move their users to the cloud. Which of the following would be considered a major challenge?

- A Developing a backup environment for the end user
- B Troubleshooting the users' applications
- C Supporting multiple devices (e.g. tablets, thin clients)
- D Centralizing the applications

A30.

Virtual Desktop Interface (VDI) will present challenges for the network administrator as they move their users to the cloud. Which of the following would be considered a major challenge?

- A Developing a backup environment for the end user
- B Troubleshooting the users' applications
- C Supporting multiple devices (e.g. tablets, thin clients)
- D Centralizing the applications

Answer: C

A31.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<PointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29
PC30         public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31         {
PC32             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33             {
PC34                 var collection = await GetCollection();
PC35                 await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36                 return pg;
PC37             }
PC38         }
PC39         private static async Task<DocumentCollection> GetCollection()
PC40         {
PC41             ...
PC42         }
PC43     }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PG01 public class PointGenerator
PG02     {
PG02
PG04         public string Name { get; set; }
PG05         [JsonProperty("currentOwner")]
PG06         public string CurrentOwner { get; set; }
PG07         [JsonProperty("sponsor")]
PG08         public string Sponsor { get; set; }
PG09         [JsonProperty("dateLastClaimed")]
PG10         public DateTimeOffset DateLastClaimed { get; set; }
PG11         [JsonProperty("location")]
PG12         public Point Location { get; set; }
PG13     }
```

You need to add a routing constraint.

Which code segment should you add at line PC29?

- A [HttpDelete]
- B [HttpPost]
- C [HttpOptions]
- D [HttpsHead]

A31.

You need to add a routing constraint.

Which code segment should you add at line PC29?

- A [HttpDelete]
- B [HttpPost]
- C [HttpOptions]
- D [HttpsHead]

Answer: B

Q32.

You create a cache for a project by using Azure Redis Cache. You are writing test code that verifies that the cache is available.

You need to ensure that data can be saved to the cache and retrieved from the cache.

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

Answer Area

```
using System;
using StackExchange.Redis;
using Microsoft.WindowsAzure.Caching;
using Microsoft.ApplicationServer.Caching;

public class RedisCacheTester
{
    public bool TestRedisCache(string name, string key)
    {
        var redisConfiguration = String.Format("{0}.redis.cache.windows.net, password={1}", name, key);
        var redisConnection = ConnectionMultiplexer.Connect(redisConfiguration);

        IDatabase cache = redisConnection.GetDatabase();
        IDatabase cache = redisConnection.GetDatabase(name);
        System.Web.Caching.Cache cache = redisConnection.GetDatabase();
        System.Web.Caching.Cache cache = redisConnection.GetDatabase(name);

        var cacheKey = "test key";
        var cacheValue = "test data";

        cache.StringSet(cacheKey, cacheValue);
        cache.StringSetOrUpdate(cacheKey, cacheValue);
        cache.StringSet(name, cacheKey, cacheValue);

        return (cacheValue == cache.StringGet(cacheKey));
    }
}
```

A32.

```
using System;
using StackExchange.Redis;
using Microsoft.WindowsAzure.Caching;
using Microsoft.ApplicationServer.Caching;

public class RedisCacheTester
{
    public bool TestRedisCache(string name, string key)
    {
        var redisConfiguration = String.Format("{0}.redis.cache.windows.net, password={1}", name, key);
        var redisConnection = ConnectionMultiplexer.Connect(redisConfiguration);

        IDatabase cache = redisConnection.GetDatabase();
        IDatabase cache = redisConnection.GetDatabase(name);
        System.Web.Caching.Cache cache = redisConnection.GetDatabase();
        System.Web.Caching.Cache cache = redisConnection.GetDatabase(name);

        var cacheKey = "test key";
        var cacheValue = "test data";

        cache.StringSet(cacheKey, cacheValue);
        cache.StringSetOrUpdate(cacheKey, cacheValue);
        cache.StringSet(name, cacheKey, cacheValue);

        return (cacheValue == cache.StringGet(cacheKey));
    }
}
```

Q33.

You need to add code to CommentController.cs to enable moderation of comments.

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

NOTE: Each correct selection is worth point.

Code segments

Authorize
AllowAnonymous
AutoValidateAntiforgeryToken
Headers ["X-MS-CLIENT-PRINCIPAL-NAME"] .First()
Headers ["Proxy-Authorization"] .First()
Headers ["X-Forwarded-For"] .First()

Answer area

[**HttpPost**]
[**Code segment**]
public async Task<ActionResult> Moderate([FromBody] Comment comment)
{
 var moderatorName = Request.
 Code segment ;
 await Save(comment, moderatorName);
 return Ok();
}

A33.

Code segments

Authorize
AllowAnonymous
AutoValidateAntiforgeryToken
Headers ["X-MS-CLIENT-PRINCIPAL-NAME"] .First()
Headers ["Proxy-Authorization"] .First()
Headers ["X-Forwarded-For"] .First()

Answer area

[**HttpPost**]
[**Authorize**]
public async Task<ActionResult> Moderate([FromBody] Comment comment)
{
 var moderatorName = Request.
 Headers ["X-MS-CLIENT-PRINCIPAL-NAME"] .First() ;
 await Save(comment, moderatorName);
 return Ok();
}

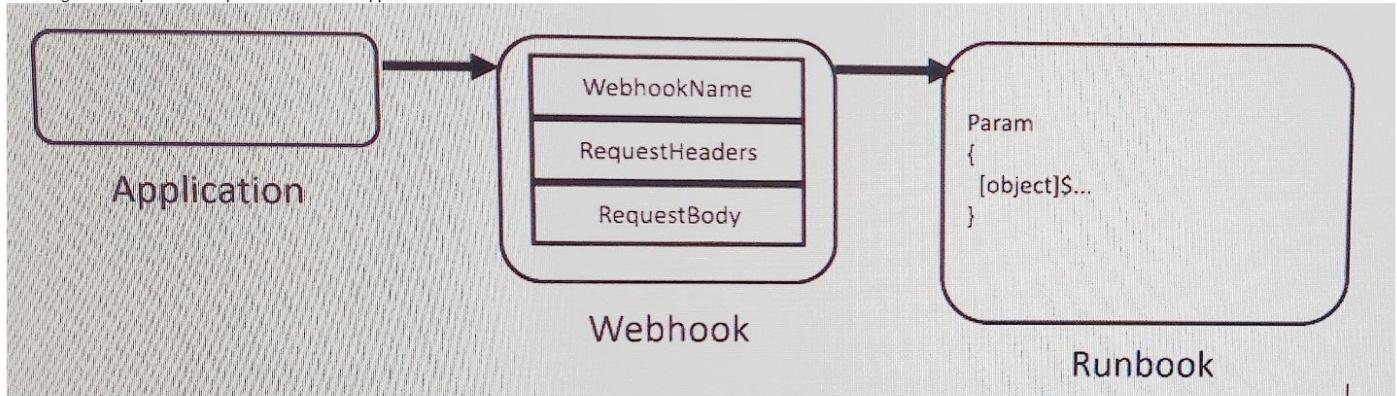
Q34.

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

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

You are developing an Azure application.

The image below represents the process flow for the application.



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

Action	Solution
Trigger the webhook.	<input checked="" type="checkbox"/> Azure Alert HTTP GET HTTP POST
Select the webhook object for passing data to the runbook.	<input checked="" type="checkbox"/> WebhookName RequestHeaders RequestBody
Select the object type to use as parameters in the runbook.	<input checked="" type="checkbox"/> Webhook Request Webhook Data Response

Azure Alert HTTP GET HTTP POST



WebhookName RequestHeaders RequestBody
--



Webhook Request Webhook Data Response
--



A34.

Trigger the webhook.

Azure Alert	<input checked="" type="checkbox"/>
HTTP GET	
HTTP POST	

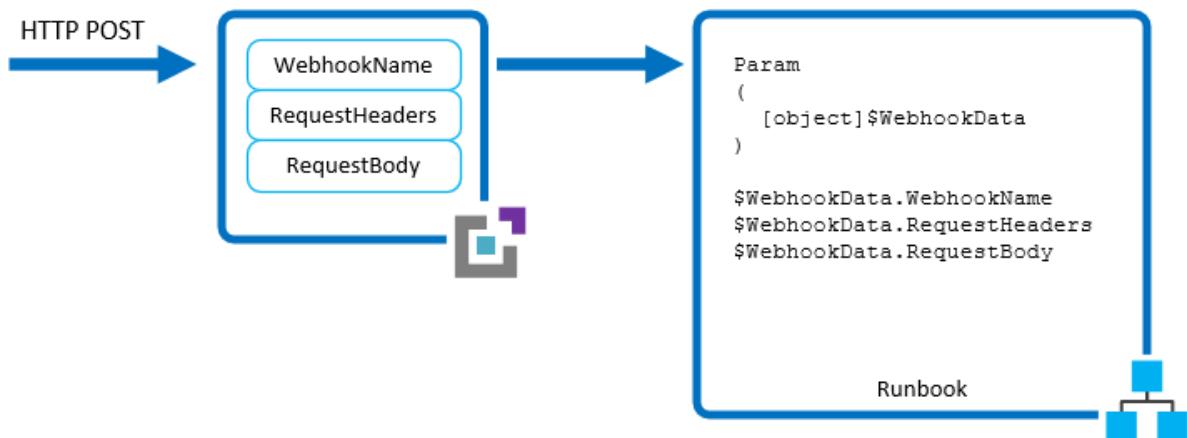
Select the webhook object for passing data to the runbook.

WebhookName	<input checked="" type="checkbox"/>
RequestHeaders	
RequestBody	

Select the object type to use as parameters in the runbook.

Webhook	<input checked="" type="checkbox"/>
Request	
Webhook Data	
Response	

A webhook allows you to start a particular runbook in Azure Automation through a single HTTP request.



Q35.

CASE STUDY

Background:

You are developer for Fabrikam, a company that specializes in payment processing. Fabrikam is developing a solution to process payments for various events, such as music concerts. You develop an ASP.NET MVC website that is hosted in Azure to support an upcoming music concert. The music concert is expected to generate a large volume of ticket sales in a short amount of time.

The website uploads information to an Azure storage queue. A worker role in Azure retrieves information from the queue and generates the concert tickets in a PDF file format after the financial transaction is approved.

You observe a delay between the time the website adds a message to a queue and the time it becomes available to read from the queue. After examining the queue, you determine that no queue messages have a DequeueCount value greater than zero. The website does not throw any errors.

Business Requirements

Payments:

The music concert website must be able to submit event payment information for processing. The website must remain responsive while submitting payment information. Customers must be able to add notes about their orders to a free-form control on the website. These notes must be submitted with the payment when the customer submits an order.

Customers often enter notes that exceed 7 KB in size.

Technical Requirements

Payment Submission and Processing:

Event payment information must be sent from the website to a Windows Communication Foundation (WCF) service worker role. The worker role must submit the information to the payment processor in JSON format.

Payment Processing

You have the following payment processing requirements:

- If the number of messages in a queue goes above or below a specified threshold, worker role instances must be created or deleted as needed. This process must be completed by using the least amount of effort. It must be easy to reconfigure role instance thresholds.
- Payments must be retrieved from the queue in the maximum batch sizes that are allowed by the queue and pulled from the queue for 5 minutes.
- The payment queue must not be re-created when processing payments.
- During single Payment processing, the number of tickets available for an event must be updated. The update operation must be retried for 30 seconds or 5 retry attempts, whichever occurs first. Each retry should pause for at least two seconds and for one second longer than the previous attempt. If the update fails, the payment should be placed in the poison queue.

Storage:

You have the following storage requirements:

- Payment information must be stored by using Azure Queue storage. Connection to the Azure storage account has been established in a configured setting namedStorageConnectionString, which is configured for the web and worker roles.
- A payment processing queue and a poison payment queue must be used when processing payments.
- Azure Queue message content must be XML-safe and UTF-8 encoded.
- An Azure storage account must be established for diagnostic information in a configured setting namedDiagnosticsConnectionString, which is configured for both the web and worker roles.

Security and Monitoring

Security

The web role must be secured by using HTTPS.

Monitoring

You must collect diagnostic data for both the web and worker roles by using the Diagnostics module. Diagnostics configuration changes must not require the code of the roles to be rebuilt. The diagnostic data is used for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis and capacity planning, and auditing.

Performance testing must evaluate the roles under normal and stress conditions without incurring changes for running Azure. Memory allocation, function time, and multithreading concurrency issues must be evaluated.

Deployment:

You purchase a custom domain name fabrikamfunding.com to host the website, web role, and worker roles. You must deploy an HTTPS certificate with the web role, and you must update associated configuration files accordingly.

Web role and worker role instance sizes must be specified as Medium. You must deploy one web role instance named FabrikamFundingPaymentGenerator, and worker role instances named FabrikamFundingPaymentProcessor.

Application Structure:

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

CustomRetryPolicy.cs

```
CR01 public class CustomRetryPolicy : IRetryPolicy
CR02 {
CR03     int _retryCount = 0;
CR04     readonly TimeSpan _baseInterval= TimeSpan.FromSeconds(1);
CR05     readonly string _poisonPaymentQueueName;
CR06     private readonly CloudQueueClient _queueClient;
CR07     private readonly EventPayment _eventPayment;
CR08     public CustomRetryPolicy(string poisonPaymentQueueName, CloudQueueClient
queueClient, EventPayment eventPayment)
CR09     {
CR10         _poisonPaymentQueueName = poisonPaymentQueueName;
CR11         _queueClient = queueClient;
CR12         _eventPayment = eventPayment;
CR13     }
CR14     public IRetryPolicy CreateInstance()
CR15     {
CR16         return new CustomRetryPolicy(_poisonPaymentQueueName, _queueClient,
_eventPayment);
CR17     }
CR18 }
```

Event.cs

```
EV01 public class Event : TableEntity
EV02 {
EV03     public int AvailableTickets { get; set; }
EV04 }
```

EventPayment.cs

```
EP01 [DataContract]
EP02 public class EventPayment
EP03 {
EP04     [DataMember]
EP05     public int EventId { get; set; }
EP06     [DataMember]
EP07     public string Email { get; set; }
EP08     [DataMember]
EP09     public string Notes { get; set; }
EP10     [DataMember]
EP11     public int TicketCount { get; set; }
EP12     [DataMember]
EP13     public DateTime OrderDate { get; set; }
EP14     [DataMember]
EP15     public Guid EventPaymentId { get; set; }
EP16 }
```

QueueManager.cs

```
QM01 public class QueueManager
QM02 {
QM03     private readonly CloudQueueClient _queueClient;
QM04     private readonly CloudTableClient _tableClient;
QM05     private const string PaymentQueueName = "paymentqueue";
QM06     private const string PoisonPaymentQueueName = "poisonpaymentqueue";
QM07     public QueueManager()
QM08     {
QM09         var storageAccount = CloudStorageAccount.Parse(
QM10             CloudConfigurationManager.GetSetting("StorageConnectionString"));
QM11         _queueClient = storageAccount.CreateCloudQueueClient();
QM12         _tableClient = storageAccount.CreateCloudTableClient();
QM13     }
QM14     public async Task SendMessageAsync(EventPayment eventPayment)
QM15     {
QM16         ...
QM17     }
QM18     public async Task ProcessMessagesAsync()
QM19     {
QM20         ...
QM21     }
QM22     private async Task ProcessPayment(EventPayment eventPayment)
QM23     {
QM23         var events = _tableClient.GetTableReference("events");
QM23         var key = eventPayment.EventId.ToString();
QM23         var operation = await
QM23             events.ExecuteAsync(TableOperation.Retrieve<Event>(key, key));
QM23         var @event = operation.Result as Event;
QM23         @event.AvailableTickets = @event.AvailableTickets - eventPay-
ment.TicketCount;
QM23         var requestOptions = new TableRequestOptions
QM23         {
QM23             RetryPolicy = new CustomRetryPolicy(
QM23                 PoisonPaymentQueueName,
QM23                 _queueClient,
QM23                 eventPayment),
QM23         };
QM23         var context = new OperationContext
QM23         {
QM23             StartTime = DateTime.Now,
QM23         };
QM23         await events.ExecuteAsync(TableOperation.Replace(@event),
requestOptions, context);
QM40     }
QM41 }
```

You need to diagnose the source of the performance issues when preparing concert tickets.

Which two actions should you perform? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A Examine the Storage Logging logs for any queue operations that have higher than expected latency.
- B Examine the Storage Client Library logs to determine whether there is a decrease in the total number of requests for storage operations.
- C Examine the diagnostic message logs for the worker role to determine whether the worker role is failing to process messages.
- D Examine the Storage Client Library logs to determine whether there are repeated retries for storage operations.

A35.

You need to diagnose the source of the performance issues when preparing concert tickets.

Which two actions should you perform? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A Examine the Storage Logging logs for any queue operations that have higher than expected latency.
- B Examine the Storage Client Library logs to determine whether there is a decrease in the total number of requests for storage operations.
- C Examine the diagnostic message logs for the worker role to determine whether the worker role is failing to process messages.
- D Examine the Storage Client Library logs to determine whether there are repeated retries for storage operations.

Answer: A, D

Q36.

Background:

You are a developer for ProseWare Inc., a software-as-a-service (SaaS) company that provides a comment system that websites use to allow for end users to post comments associated with a webpage or topic on a customer's website.

Business requirements

Moderation:

The moderation of comments is a feature of the software, and usually involves the editing of a comment.

Only users who have accounts in a group in Azure Active Directory (Azure AD) have the ability moderate. External users can also become moderators, but only by explicit invitation.

Any moderation action must include the name of the moderator.

Comment navigation:

Each comment is identified by a unique string consisting of a random string of characters.

Within the body of a comment, internal links to other comment threads can be specified using the link format: "/<parent comment id> / <child comment id>"

Comment search:

Comments can be searched using Azure Search. Searches must do the following:

- Searching for email addresses must match email addresses in comments.
- Searching must work for the client's language.
- Internal links to other comments using the link format should be searched.

Content screening:

Comment content is screened for inappropriate language, length, and topic using content analysis. Content must be screened, but can appear prior to be screened.

Mobile App:

The moderation functionality can be accessed using a Universal Windows Platform (UWP) app named ProsewareApp. The app includes functionality that notifies moderators when changes are made to a comment they modified.

Export:

Customers can perform an export of all comments to a customer supplied Microsoft OneDrive folder on demand. The export functionality is implemented as an Azure Logic App, and it must be able to be triggered by the customer from their local network.

Interaction agents:

Interaction agents are parts of the system that interact with comment threads. The main purpose is to modify a comment's body based on the contents of the comment. For example, one of the agents is WikiAgent, which adds links to Wikipedia articles when it sees text in the comment body that exactly matches a Wikipedia article title. Interaction Agents are implemented in Service Fabric.

Interaction agents must meet the following requirements:

- Only successfully process each comment once
- Any errors encountered during the processing of a comment should be retried
- Must run on systems that allow for custom applications to be installed
- Must run in a VNet or private network space
- Must be run on a system that can scale up and down based on demand
- A single user's usage of Interaction Agents must not impact other users' usage of Interaction Agents

Technical requirements

Authentication:

ProseWare Inc. allows for user authentication through Azure AD and Twitter.

Storage:

The application runs as a Web App on Azure. Comments are stored in an Azure DocumentDB database named "Proseware".

Performance:

The product includes a service level agreement (SLA) for individual method performance. All data retrieval methods must return within 100ms 99% of the time.

API:

The ProseWare Inc. API is made available to public callers using an Azure API App. Azure AD and Twitter are the Authentication Providers.

Application structure

CommentController.cs:

```
CC01 [Route("api/[controller]")]
CC02 public class CommentController : Controller
CC03 {
CC04     private IDatabase _redis;
CC05     private DataStore _dataStore;
CC06     private CloudQueue _queue
CC07
CC08     public CommentController ()
CC09     {
CC10         _queue = CloudStorageAccount.Parse(" ").Create-
CloudQueueClient().GetQueueReference ("commentQueue");
CC11         _redis = ConnectionMultiplexer.Connect("...").GetData-
base();
CC12         _dataStore = new DataStore();
CC13     }
CC14
CC15     [HttpGet("{commentId}")]
CC16     public async Task<Comment> Get(string commentId)
CC17     {
CC18         var cached = await _redis.StringGetAsync(commentId);
CC19         if (cached.HasValue)
CC20         {
CC21             return JsonConvert.DeserializeObject<Com-
ment>(cached.ToString());
CC22         }
CC23         return await _dataStore.LoadAsync(commentId);
CC24     }
CC25
CC26     [HttpGet]
CC27     public IEnumerable<Comment> GetChildComments(string com-
mentId)
CC28     {
CC29         IEnumerable<Comment> result = null;
CC30
CC31         if (result == null)
CC32         {
CC33             result = _dataStore.LoadThread(commentId);
CC34         }
CC35         return results;
CC36     }
CC37
CC38     [HttpPost]
CC39     public async Task<IActionResult> New([FromBody]Comment
comment)
CC40     {
CC41         await Save(comment);
CC42         return Ok();
CC43     }
CC44
CC45     [HttpPost]
CC46     public async Task<IActionResult> Reply(string inRe-
sponseTo, [FromBody]Comment comment)
CC47     {
CC48         comment.InResponseTo = inResponseTo;
CC49         await Save(comment);
CC50         return View();
CC51     }
CC52
```

```

CC53  private static Comment Convert(string json)
CC54  {
CC55      return JsonConvert.DeserializeObject<Comment>(json);
CC56  }
CC57
CC58  private async Task Save(Comment comment, string moderator-
Name = null)
CC59  {
CC60      comment.Moderator = moderatorName;
CC61      var json = JsonConvert.SerializeObject(comment);
CC62      _redis.StringSet(comment.Id, json);
CC62
CC63
CC64      await _queue.AddMessageAsync(new CloudQueueMessage(com-
ment.Id));
CC65      _dataStore.Save(comment);
CC66  }
CC67 }

```

cleaner.csx:

```

CL01 #r "Newtonsoft.Json"
CL02
CL03 using System;
CL04 using Newtonsoft.Json;
CL05 using Newtonsoft.Json.Linq;
CL06 public static void Run(string commentId, object result,
TraceWriter log)
CL07 {
CL08     dynamic comment = JObject.Parse(item);
CL09 ...
CL10     result = comment;
CL11 }

```

ICommentAgent.cs:

```

CA01 public interface ICommentAgent: IActor
CA02  {
CA03     Task<string> ModifyCommentText(string id, string body,
string title);
CA04  }

```

WikiAgent.cs:

```

WA01 [StatePersistence(StatePersistence.Persisted)]
WA02 internal class WikiAgent : Agent, ICommentAgent
WA03 {
WA04     public WikiAgent(ActorService, ActorId id) : base(service,
id) {}
WA05     public async Task<string> ModifyCommentText(string id,
string body, string title)
WA06 {
WA07     try
WA08     {
WA09
WA10         var newBody = scanForLinks(body);
WA11
WA12         return newBody;
WA13     }
WA14     catch
WA15     {
WA16
WA17         throw
WA18     }
WA19 }
WA20 }

```

Comment.cs:

```
CO01 public class Comment
CO02 {
CO03     public string Id {get; set;}
CO04     public string UserId {get; set;}
CO05     public string InResponseTo {get; set;}
CO06     public string Title {get; set;}
CO07     public DateTimeOffset Date {get; set;}
CO08     public string Body {get; set;}
CO09     public string Moderator {get; internal; set;}
CO10 }
```

DataStore.cs:

```
DS01 public class DataStore
DS02 {
DS03     private const string EndpointUrl = "https://
//proseware.documents.azure.com:443/";
DS04     private const string PrimaryKey = "";
DS05     private const string db = "Proseware";
DS06     private const string col = "Comments";
DS07     private DocumentClient client;
DS08
DS09     public DataStore()
DS10     {
DS11         client = new DocumentClient(new Uri(EndpointUrl), Pri-
maryKey);
DS12     }
DS13
DS14     public async Task<Comment> LoadAsync(string commentId)
DS15     {
DS16         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS17         return await client.ReadDocumentAsync<Comment>(UriFac-
tory.CreateDocumentUri(db, col, commentId));
DS18     }
DS19
DS20     public async void Save(Comment comment)
DS21     {
DS22         var uri = UriFactory.CreateDocumentCollectionUri(db,
col, comment.Id);
DS23         await client.UpsertDocumentAsync(uri, comment);
DS24     }
DS25     public IEnumerable<Comment> LoadThread(string commentId)
DS26     {
DS27         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS28         return client.CreateDocumentQuery<Comment>(uri).Where(f
=> f.Id == commentId);
DS29     }
}
}
```

MainPage.xaml.cs:

```
MP01 public sealed partial class MainPage : Page
MP02 {
MP03     public MainPage()
MP04     {
MP05         InitializeComponent();
MP06     }
MP07
MP08     private async void StartNotify()
MP09     {
MP10     }
MP11
MP12     private void UpdateUI()
MP13     {
MP14     }
MP15 }
```

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

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

You need to implement the infrastructure for the Interaction Agents.

Solution: Create an Azure virtual machine (VM) scale set and use Azure Desired State Configuration (DSC) extension handler to install Service Fabric runtime.

Does the solution meet the goal?

- A Yes
- B No

A36.

You need to implement the infrastructure for the Interaction Agents.

Solution: Create an Azure virtual machine (VM) scale set and use Azure Desired State Configuration (DSC) extension handler to install Service Fabric runtime.

Does the solution meet the goal?

- A Yes
- B No

Answer: B

Q37.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create a Virtual Machine Scale Set (VMSS) that has an instance count of 4.

Does the solution meet the goal?

- A Yes
 B No

A37.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create a Virtual Machine Scale Set (VMSS) that has an instance count of 4.

Does the solution meet the goal?

- A Yes
 B No

Answer: B

Q38.

You deploy an application as a cloud service to Azure. The application contains a web role to convert temperatures between Celsius and Fahrenheit.

The application does not correctly convert temperatures. You must use Microsoft Visual Studio to determine why the application does not correctly convert temperatures.

You need to debug the source code in Azure.

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

Actions	Answer Area
Attach the debugger to the role instance of the cloud service.	
Publish the application.	
In the Microsoft Azure Publish Settings dialog, set the build configuration to Release and enable the remote debugger for all roles.	
In the Windows Azure Publish Settings dialog, set the build configuration to Debug .	
In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.	

A38.

Actions	Answer Area
Attach the debugger to the role instance of the cloud service.	
Publish the application.	Publish the application.
In the Microsoft Azure Publish Settings dialog, set the build configuration to Release and enable the remote debugger for all roles.	In the Microsoft Azure Publish Settings dialog, set the build configuration to Release and enable the remote debugger for all roles.
In the Windows Azure Publish Settings dialog, set the build configuration to Debug .	Attach the debugger to the role instance of the cloud service.
In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.	

Q39.

Background:

You are a developer for ProseWare Inc., a software-as-a-service (SaaS) company that provides a comment system that websites use to allow for end users to post comments associated with a webpage or topic on a customer's website.

Business requirements

Moderation:

The moderation of comments is a feature of the software, and usually involves the editing of a comment.

Only users who have accounts in a group in Azure Active Directory (Azure AD) have the ability moderate. External users can also become moderators, but only by explicit invitation.

Any moderation action must include the name of the moderator.

Comment navigation:

Each comment is identified by a unique string consisting of a random string of characters.

Within the body of a comment, internal links to other comment threads can be specified using the link format: "/<parent comment id> / <child comment id>"

Comment search:

Comments can be searched using Azure Search. Searches must do the following:

- Searching for email addresses must match email addresses in comments.
- Searching must work for the client's language.
- Internal links to other comments using the link format should be searched.

Content screening:

Comment content is screened for inappropriate language, length, and topic using content analysis. Content must be screened, but can appear prior to be screened.

Mobile App:

The moderation functionality can be accessed using a Universal Windows Platform (UWP) app named ProsewareApp. The app includes functionality that notifies moderators when changes are made to a comment they modified.

Export:

Customers can perform an export of all comments to a customer supplied Microsoft OneDrive folder on demand. The export functionality is implemented as an Azure Logic App, and it must be able to be triggered by the customer from their local network.

Interaction agents:

Interaction agents are parts of the system that interact with comment threads. The main purpose is to modify a comment's body based on the contents of the comment. For example, one of the agents is WikiAgent, which adds links to Wikipedia articles when it sees text in the comment body that exactly matches a Wikipedia article title. Interaction Agents are implemented in Service Fabric.

Interaction agents must meet the following requirements:

- Only successfully process each comment once
- Any errors encountered during the processing of a comment should be retried
- Must run on systems that allow for custom applications to be installed
- Must run in a VNet or private network space
- Must be run on a system that can scale up and down based on demand
- A single user's usage of Interaction Agents must not impact other users' usage of Interaction Agents

Technical requirements

Authentication:

ProseWare Inc. allows for user authentication through Azure AD and Twitter.

Storage:

The application runs as a Web App on Azure. Comments are stored in an Azure DocumentDB database named "Proseware".

Performance:

The product includes a service level agreement (SLA) for individual method performance. All data retrieval methods must return within 100ms 99% of the time.

API:

The ProseWare Inc. API is made available to public callers using an Azure API App. Azure AD and Twitter are the Authentication Providers.

Application structure

CommentController.cs:

```
CC01 [Route("api/[controller]")]
CC02 public class CommentController : Controller
CC03 {
CC04     private IDatabase _redis;
CC05     private DataStore _dataStore;
CC06     private CloudQueue _queue
CC07
CC08     public CommentController ()
CC09     {
CC10         _queue = CloudStorageAccount.Parse(" ").Create-
CloudQueueClient().GetQueueReference ("commentQueue");
CC11         _redis = ConnectionMultiplexer.Connect("...").GetData-
base();
CC12         _dataStore = new DataStore();
CC13     }
CC14
CC15     [HttpGet("{commentId}")]
CC16     public async Task<Comment> Get(string commentId)
CC17     {
CC18         var cached = await _redis.StringGetAsync(commentId);
CC19         if (cached.HasValue)
CC20         {
CC21             return JsonConvert.DeserializeObject<Com-
ment>(cached.ToString());
CC22         }
CC23         return await _dataStore.LoadAsync(commentId);
CC24     }
CC25
CC26     [HttpGet]
CC27     public IEnumerable<Comment> GetChildComments(string com-
mentId)
CC28     {
CC29         IEnumerable<Comment> result = null;
CC30
CC31         if (result == null)
CC32         {
CC33             result = _dataStore.LoadThread(commentId);
CC34         }
CC35         return results;
CC36     }
CC37
CC38     [HttpPost]
CC39     public async Task<IActionResult> New([FromBody]Comment
comment)
CC40     {
CC41         await Save(comment);
CC42         return Ok();
CC43     }
CC44
CC45     [HttpPost]
CC46     public async Task<IActionResult> Reply(string inRe-
sponseTo, [FromBody]Comment comment)
CC47     {
CC48         comment.InResponseTo = inResponseTo;
CC49         await Save(comment);
CC50         return View();
CC51     }
CC52
CC53     private static Comment Convert(string json)
CC54     {
CC55         return JsonConvert.DeserializeObject<Comment>(json);
CC56     }
CC57
```

```

CC58     private async Task Save(Comment comment, string moderator-
Name = null)
CC59     {
CC60         comment.Moderator = moderatorName;
CC61         var json = JsonConvert.SerializeObject(comment);
CC62         _redis.StringSet(comment.Id, json);
CC63
CC64         await _queue.AddMessageAsync(new CloudQueueMessage(com-
ment.Id));
CC65         _dataStore.Save(comment);
CC66     }
CC67 }
```

cleaner.csx:

```

CL01 #r "Newtonsoft.Json"
CL02
CL03 using System;
CL04 using Newtonsoft.Json;
CL05 using Newtonsoft.Json.Linq;
CL06 public static void Run(string commentId, object result,
TraceWriter log)
CL07 {
CL08     dynamic comment = JObject.Parse(item);
CL09 ...
CL10     result = comment;
CL11 }
```

ICommentAgent.cs:

```

CA01 public interface ICommentAgent: IActor
CA02     {
CA03     Task<string> ModifyCommentText(string id, string body,
string title);
CA04 }
```

WikiAgent.cs:

```

WA01 [StatePersistence(StatePersistence.Persisted)]
WA02 internal class WikiAgent : Agent, ICommentAgent
WA03 {
WA04     public WikiAgent(ActorService, ActorId id) : base(service,
id) {}
WA05     public async Task<string> ModifyCommentText(string id,
string body, string title)
WA06     {
WA07         try
WA08         {
WA09             var newBody = scanForLinks(body);
WA11
WA12             return newBody;
WA13         }
WA14         catch
WA15         {
WA16             throw
WA18     }
WA19 }
WA20 }
```

Comment.cs:

```

CO01 public class Comment
CO02 {
CO03     public string Id {get; set;}
CO04     public string UserId {get; set;}
CO05     public string InResponseTo {get; set;}
CO06     public string Title {get; set;}
CO07     public DateTimeOffset Date {get; set;}
CO08     public string Body {get; set;}
CO09     public string Moderator {get; internal; set;}
CO10 }
```

DataStore.cs:

```
DS01 public class DataStore
DS02 {
DS03     private const string EndpointUrl = "https:
//proseware.documents.azure.com:443/";
DS04     private const string PrimaryKey = "";
DS05     private const string db = "Proseware";
DS06     private const string col = "Comments";
DS07     private DocumentClient client;
DS08
DS09     public DataStore()
DS10     {
DS11         client = new DocumentClient(new Uri(EndpointUrl), Pri-
maryKey);
DS12     }
DS13
DS14     public async Task<Comment> LoadAsync(string commentId)
DS15     {
DS16         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS17         return await client.ReadDocumentAsync<Comment>(UriFac-
tory.CreateDocumentUri(db, col, commentId));
DS18     }
DS19
DS20     public async void Save(Comment comment)
DS21     {
DS22         var uri = UriFactory.CreateDocumentCollectionUri(db,
col, comment.Id);
DS23         await client.UpsertDocumentAsync(uri, comment);
DS24     }
DS25     public IEnumerable<Comment> LoadThread(string commentId)
DS26     {
DS27         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS28         return client.CreateDocumentQuery<Comment>(uri).Where(f
=> f.Id == commentId);
DS29     }
}
}
```

MainPage.xaml.cs:

```
MP01 public sealed partial class MainPage : Page
MP02 {
MP03     public MainPage()
MP04     {
MP05         InitializeComponent();
MP06     }
MP07
MP08     private async void StartNotify()
MP09     {
MP10     }
MP11
MP12     private void UpdateUI()
MP13     {
MP14     }
MP15 }
```

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

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

You need to implement the infrastructure for the Interaction Agents.

Solution: Create a set of Azure virtual machines (VMs) using Azure Resource Manager (ARM) templates, and use Chef to install the Service Fabric runtime.

Does the solution meet the goal?

- A Yes
- B No

A39.

You need to implement the infrastructure for the Interaction Agents.

Solution: Create a set of Azure virtual machines (VMs) using Azure Resource Manager (ARM) templates, and use Chef to install the Service Fabric runtime.

Does the solution meet the goal?

- A Yes
- B No

Answer: B

Q40.

You are developing a web application that integrates with Azure Active Directory (AD). The application uses the OAuth 2.0 protocol to authorize secure connections to a web service that is at <https://service.adatum.com>.

The application must request an access token to invoke the web service methods.

You need to submit an HTTP request to the Azure AD endpoint.

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

HTTP Request Segments	Answer Area
adatum.com/oauth2/token	POST <input type="text"/> HTTP/1.1
common/oauth2/token	...
grant_type	resource=https%3A%2F%2Fservice.adatum.com%2F&
response_type	<input type="text"/> = <input type="text"/> &
client_credentials	<input type="text"/> =F92FC9B0-F515-433E-BB72-482CC2303E62
client_id	

A40.

HTTP Request Segments	Answer Area
adatum.com/oauth2/token	POST <input type="text"/> adatum.com/oauth2/token HTTP/1.1
common/oauth2/token	...
grant_type	resource=https%3A%2F%2Fservice.adatum.com%2F&
response_type	<input type="text"/> grant_type = <input type="text"/> client_credentials &
client_credentials	<input type="text"/> client_id =F92FC9B0-F515-433E-BB72-482CC2303E62
client_id	

Q41.

You need to add code at line CC63 to ensure that the Interaction Agent is invoked.

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

NOTE: Each correct selection is worth point.

Code fragments

ActorId.CreateRandom()

new ActorId(comment.UserId)

new ActorId(comment.Id)

ActorProxy

ServiceProxy

ActorServiceProxy

Answer area

```
var actorId = Code fragment;  
var actor = Code fragment.Create<ICommentAgent>(actorId, "...");  
await actor.ModifyCommentText(comment.Id, commetn.Body,  
commey.Title);
```

A41.

Code fragments

ActorId.CreateRandom()

new ActorId(comment.UserId)

new ActorId(comment.Id)

ActorProxy

ServiceProxy

ActorServiceProxy

Answer area

```
var actorId = ActorId.CreateRandom();  
var actor = ActorProxy.Create<ICommentAgent>(actorId, "...");  
await actor.ModifyCommentText(comment.Id, commetn.Body,  
commey.Title);
```

Q42.

You store JSON data in a blob by using the Azure Blob service. Web applications access the JSON data by using client-side JavaScript calls.

JSON data is stored in a container that is configured to allow anonymous access. Web applications that are allowed to make updates to the data have access to any necessary shared access signatures (SASs) and storage keys.

[You configure one Cross-Origin Resource Sharing \(CORS\) rule for the https://fabrikam.com domain and then run the following method. Line numbers are provided for reference only.](#)

```
01 void ConfigureBlobCorsRules(CloudBlobClient blobClient)
02 {
03     var blobServiceProperties = blobClient.GetServiceProperties();
04     var partnerCorsRule = new CorsRule();
05     partnerCorsRule.AllowedOrigins.Add("https://contoso.com");
06     partnerCorsRule.AllowedMethods = CorsHttpMethods.Post | CorsHttpMethods.Put;
07     partnerCorsRule.ExposedHeaders.Add("*");
08     partnerCorsRule.AllowedHeaders.Add("*");
09     blobServiceProperties.Cors.CorsRules.Add(partnerCorsRule);
10    var publicCorsRule = new CorsRule();
11    publicCorsRule.AllowedOrigins.Add("*");
12    publicCorsRule.AllowedMethods = CorsHttpMethods.Get;
13    publicCorsRule.ExposedHeaders.Add("*");
14    publicCorsRule.AllowedHeaders.Add("*");
15    blobServiceProperties.Cors.CorsRules.Add(publicCorsRule);
16    blobClient.SetServiceProperties(blobServiceProperties);
17 }
```

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

Answer Area

Yes	No
-----	----

The CORS rule that was previously configured for https://fabrikam.com is no longer in effect after this method runs.

Partners from the https://contoso.com domain can access the configured storage by using the **HTTP HEAD** operation.

Partners from the https://contoso.com domain can access the configured storage service by using the **HTTP GET** operation.

A42.

Answer Area

Yes	No
-----	----

The CORS rule that was previously configured for https://fabrikam.com is no longer in effect after this method runs.

Partners from the https://contoso.com domain can access the configured storage by using the **HTTP HEAD** operation.

Partners from the https://contoso.com domain can access the configured storage service by using the **HTTP GET** operation.

Q43.

You deploy a stateless ASP.NET application to an Azure website. You scale out the application by adding website instances.

Only newly signed in users are routed to the recently added website instances. Users must be evenly distributed among all of the instances.

You need to configure the environment to ensure that the load balancer evenly distributes requests.

What should you do?

- A. Add the following markup to the web.config file for the application:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Arr-Disable-Session-Affinity" value="False" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

- B. Configure autoscaling rules based on metrics.

- C. Add the following markup to the web.config file for the application:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Arr-Disable-Session-Affinity" value="True" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

- D. Enable Always On support.

A43.

You deploy a stateless ASP.NET application to an Azure website. You scale out the application by adding website instances.

Only newly signed in users are routed to the recently added website instances. Users must be evenly distributed among all of the instances.

You need to configure the environment to ensure that the load balancer evenly distributes requests.

What should you do?

- A. Add the following markup to the web.config file for the application:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Arr-Disable-Session-Affinity" value="False" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

- B. Configure autoscaling rules based on metrics.

- C. Add the following markup to the web.config file for the application:

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Arr-Disable-Session-Affinity" value="True" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

- D. Enable Always On support.

A Option A

B Option B

C Option C

D Option D

Answer: C

Q44.

You are developing a web application that uses Azure push notifications to interact with users.

You need to send a text notification to users to alert them that the application is ready to test.

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

Code Segments	Answer Area
"toast"	<code>var payload = new XElement(</code> [red box] <code>,</code>
"visual"	<code>new XElement(</code> [red box]
"binding"	<code>new XElement(</code> [red box]
"template"	<code>new XAttribute(</code> [red box] <code>, "ToastText02"),</code>
"text"	<code>new XElement(</code> [red box] <code>, "System Ready")));</code>
"notification"	<code>var message = new WindowsNotification(payload.ToString());</code>

A44.

Code Segments	Answer Area
"toast"	<code>var payload = new XElement(</code> [red box] <code>"toast"</code> <code>,</code>
"visual"	<code>new XElement(</code> [red box] <code>"visual"</code> <code>,</code>
"binding"	<code>new XElement(</code> [red box] <code>"binding"</code> <code>,</code>
"template"	<code>new XAttribute(</code> [red box] <code>"template"</code> <code>,</code> <code>"ToastText02"),</code>
"text"	<code>new XElement(</code> [red box] <code>"text"</code> <code>,</code> <code>"System Ready")));</code>
"notification"	<code>var message = new WindowsNotification(payload.ToString());</code>

Q45.

You are creating virtual machines (VMs) that are hosted on Azure.

You must be able to change the Remote Desktop access settings for the VMs. You must also be able to change the password for the built-in administrator account on all VMs. You identify the VMAccess VM extensions that have the required capabilities.

You need to enable the VMAccess VM extensions.

Which approach should you use?

- A Download and install the Microsoft Installer file to enable the VM Agent on each VM.
- B Use the Azure management portal to restart each VM.
- C When you configure the new VMs, use the Azure management portal to install the VM Agent.
- D For each VM, use Windows PowerShell cmdlets to enable the VM Agent and the VMAccess VM extensions.

A45.

You are creating virtual machines (VMs) that are hosted on Azure.

You must be able to change the Remote Desktop access settings for the VMs. You must also be able to change the password for the built-in administrator account on all VMs. You identify the VMAccess VM extensions that have the required capabilities.

You need to enable the VMAccess VM extensions.

Which approach should you use?

- A Download and install the Microsoft Installer file to enable the VM Agent on each VM.
- B Use the Azure management portal to restart each VM.
- C When you configure the new VMs, use the Azure management portal to install the VM Agent.
- D For each VM, use Windows PowerShell cmdlets to enable the VM Agent and the VMAccess VM extensions.

Answer: D

Q46.

You deploy an application as a cloud service in Azure.

The application consists of five instances of a web role.

You need to move the web role instances to a different subnet.

Which file should you update?

- A Service definition
- B Diagnostics configuration
- C Service configuration
- D Network configuration

A46.

You deploy an application as a cloud service in Azure.

The application consists of five instances of a web role.

You need to move the web role instances to a different subnet.

Which file should you update?

- A Service definition
- B Diagnostics configuration
- C Service configuration
- D Network configuration

Answer: C

Q47.

Your team uses a proprietary source control product. You use FTP to manually deploy an Azure Web App.

You must move your source code from the proprietary source control product to a secure on-premises Git versioning system. Instead of deploying the website by using FTP, the website must automatically deploy to Azure each time developers check-in source files.

You need to implement the new deployment strategy.

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

- In the Azure management portal, configure Web Apps to support deployment from the local Git repository.
- Commit the website to the Git repository.
- In the Azure portal, configure Web App to support deployment from Microsoft Visual Studio Team Services.
- In the Azure portal, configure Web App to support deployment from external repository sources.
- Create a local Git repository.

Answer Area

A47.

Actions

- In the Azure management portal, configure Web Apps to support deployment from the local Git repository.
- Commit the website to the Git repository.
- In the Azure portal, configure Web App to support deployment from Microsoft Visual Studio Team Services.
- In the Azure portal, configure Web App to support deployment from external repository sources.
- Create a local Git repository.

Answer Area

In the Azure management portal, configure Web Apps to support deployment from the local Git repository.	
	Create a local Git repository.
	Commit the website to the Git repository.

Q48.

You connect to an existing service over the network by using HTTP. The service listens on HTTP port 80. You plan to create a test environment for this existing service by using an Azure virtual machine (VM) that runs Windows Server.

The service must be accessible from the public Internet over HTTP port 8080.

You need to configure the test environment.

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

- A Configure a Network Security Group to route traffic from port 8080 to port 80
- B Configure a Network Security Group to route traffic from port 80 to port 8080.
- C Ensure that the public IP address is configured as a static IP address.
- D Configure the Windows Server firewall to allow incoming and outgoing traffic on port 8080.
- E Configure the Windows Server firewall to allow incoming and outgoing traffic on port 80.

A48.

You connect to an existing service over the network by using HTTP. The service listens on HTTP port 80. You plan to create a test environment for this existing service by using an Azure virtual machine (VM) that runs Windows Server.

The service must be accessible from the public Internet over HTTP port 8080.

You need to configure the test environment.

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

- A Configure a Network Security Group to route traffic from port 8080 to port 80
- B Configure a Network Security Group to route traffic from port 80 to port 8080.
- C Ensure that the public IP address is configured as a static IP address.
- D Configure the Windows Server firewall to allow incoming and outgoing traffic on port 8080.
- E Configure the Windows Server firewall to allow incoming and outgoing traffic on port 80.

Answer: A, E

Q49.

You create a software-as-a-service (SaaS) application. Websites, cloud services, and virtual machines (VMs) read common data values from the database for the application.

The application does not scale efficiently. All VMs, websites, and cloud services must read from the same data source.

You need to design a cache solution for the SaaS application.

What should you do?

- A Deploy a cache by using Azure Redis Cache. Access the cache from the websites, cloud services, and VMs.
- B Configure a cache by using ASP.NET. Access the cache from the websites, cloud services, and VMs.
- C Use Azure Redis Cache to deploy one cache for each website, one cache for each cloud service, and one cache for each VM. Configure each cache to ensure that data is consistent in all the cache instances.
- D Deploy a cache by using Azure Redis Cache. Configure the cache to use database connection strings.

A49.

You create a software-as-a-service (SaaS) application. Websites, cloud services, and virtual machines (VMs) read common data values from the database for the application.

The application does not scale efficiently. All VMs, websites, and cloud services must read from the same data source.

You need to design a cache solution for the SaaS application.

What should you do?

- A Deploy a cache by using Azure Redis Cache. Access the cache from the websites, cloud services, and VMs.
- B Configure a cache by using ASP.NET. Access the cache from the websites, cloud services, and VMs.
- C Use Azure Redis Cache to deploy one cache for each website, one cache for each cloud service, and one cache for each VM. Configure each cache to ensure that data is consistent in all the cache instances.
- D Deploy a cache by using Azure Redis Cache. Configure the cache to use database connection strings.

Answer: A

Q50.

You use the storage client library to develop an application that manages Azure table storage data.

The application reports error codes when it saves data. You must use a custom retry policy to handle the error codes.

The custom retry policy must meet the following requirements:

- Retry when a conflict error code is encountered.
- Retry when a storage exception is encountered.
- Retry until the maximum number of retry attempts is reached.

You create the following code segment. Line numbers are included for reference only.

```
01 public class CustomRetryPolicy : IRetryPolicy
02 {
03     private readonly int _maxRetryAttempts = 10;
04     private readonly TimeSpan _defaultRetryInterval = TimeSpan.FromSeconds(5);
05     public CustomRetryPolicy(TimeSpan deltaBackoff, int retryAttempts)
06     {
07         _maxRetryAttempts = retryAttempts;
08         _defaultRetryInterval = deltaBackoff;
09     }
10     public IRetryPolicy CreateInstance()
11     {
12         return new CustomRetryPolicy(_defaultRetryInterval, _maxRetryAttempts);
13     }
14 }
15 }
```

You need to insert code at line 14 to implement the retry policy.

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

Answer Area

```
public bool ShouldRetry(int currentRetryCount, int statusCode,
    Exception lastException, out TimeSpan retryInterval,
    OperationContext operationContext)
{
    retryInterval = _defaultRetryInterval;

    if ( )
        _maxRetryAttempts != currentRetryCount
        currentRetryCount &gt;= _maxRetryAttempts
        retryInterval &gt;= _defaultRetryInterval
        retryInterval == _defaultRetryInterval

    {
        return false;
    }

    if ( )
        (HttpStatusCode) statusCode != HttpStatusCode.Conflict
        (HttpStatusCode) statusCode == HttpStatusCode.Moved
        (HttpStatusCode) statusCode != HttpStatusCode.ExpectationFailed
        (HttpStatusCode) statusCode != HttpStatusCode.Unauthorized

    {
        return false;
    }

    if ( )
        lastException.GetType() == typeof(AccessViolationException)
        lastException.GetType() == typeof(ContextMarshalException)
        !lastException.GetType() != typeof(UnauthorizedAccessException)
        lastException.GetType() != typeof(StorageException)

    {
        return false;
    }
    return true;
}
```

A50.

```
public bool ShouldRetry(int currentRetryCount, int statusCode,
    Exception lastException, out TimeSpan retryInterval,
    OperationContext operationContext)
{
    retryInterval = _defaultRetryInterval;

    if (
        maxRetryAttempts != currentRetryCount
        currentRetryCount >= maxRetryAttempts
        retryInterval >= _defaultRetryInterval
        retryInterval == _defaultRetryInterval
    )

    {
        return false;
    }

    if (
        (HttpStatusCode) statusCode != HttpStatusCode.Conflict
        (HttpStatusCode) statusCode == HttpStatusCode.Moved
        (HttpStatusCode) statusCode != HttpStatusCode.ExpectationFailed
        (HttpStatusCode) statusCode != HttpStatusCode.Unauthorized
    )

    {
        return false;
    }

    if (
        lastException.GetType() == typeof(AccessViolationException)
        lastException.GetType() == typeof(ContextMarshalException)
        lastException.GetType() != typeof(UnauthorizedAccessException)
        lastException.GetType() != typeof(StorageException)
    )

    {
        return false;
    }
    return true;
}
```

Q51.

You are modifying a web application so that it uses Azure Active Directory to manage users. You create a security group named Users and a security group named Administrators. The Administrators security group is a member of the Users security group.

You create the following code segment. Line numbers are included for reference only.

```
01 function canAccessUserResources(userId) {  
02  
03 }  
04 function getGroupId(groupName) {  
05 ...  
06 }  
07 function domain() {  
08 ...  
09 }
```

You need to implement the canAccessUserResources function.

Which code segment should you insert at line 02?

- A.

```
var groupId = getGroupId("Users");  
var link = domain().concat("/users/", userId, "/memberOf?api-version=2013-04-05");  
var json = $.getJSON(link);  
for (entry in json.Value)  
    if (entry.objectId == groupId)  
        return true;  
return false;
```
- B.

```
var groupId = getGroupId("Users");  
var link = domain().concat("/isMemberOf?api-version=2013-04-05");  
var json = $.post(link, { groupId: groupId, memberId: userId });  
return json.value;
```
- C.

```
var groupId = getGroupId("User");  
var link = domain().concat("/roles/", groupId, "?api-version=2013-04-05");  
var json = $.getJSON(link);  
return json.value;
```
- D.

```
var groupId = getGroupId("Users");  
var link = domain().concat("/groups/", groupId, "/members?api-version=2013-04-05");  
var json = $.getJSON(link);  
for (entry in json.Value)  
    if (entry.objectId == userId)  
        return true;  
return false;
```

A51.

- A Option A
- B Option B
- C Option C
- D Option D

Answer: C

Q52.

You manage an on-premises server that runs Windows Server 2016. The server has a disk that contains 4 terabytes (TB) of data and thousands of files. None of the individual files are larger than 1 TB. You plan to create a virtual machine (VM) in Azure to process the workload currently handled by the on-premises server.

You need to create a storage location for the data.

What should you do?

- A Create premium storage account. Use a D-series VM.
- B Configure a StorSimple virtual array. Configure the VM to use the array with the SMB protocol.
- C Add a new table storage account. Update the VM workload to use the table storage.
- D Add a single file share to the VM. In the VM operating system, assign a drive letter.

A52.

You manage an on-premises server that runs Windows Server 2016. The server has a disk that contains 4 terabytes (TB) of data and thousands of files. None of the individual files are larger than 1 TB. You plan to create a virtual machine (VM) in Azure to process the workload currently handled by the on-premises server.

You need to create a storage location for the data.

What should you do?

- A Create premium storage account. Use a D-series VM.
- B Configure a StorSimple virtual array. Configure the VM to use the array with the SMB protocol.
- C Add a new table storage account. Update the VM workload to use the table storage.
- D Add a single file share to the VM. In the VM operating system, assign a drive letter.

Answer: D

Q53.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create two resource groups by using the Azure portal. Create four VMs. Assign two VMs to the first resource group and two to the second group.

Does the solution meet the goal?

- A Yes
 B No

A53.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create two resource groups by using the Azure portal. Create four VMs. Assign two VMs to the first resource group and two to the second group.

Does the solution meet the goal?

- A Yes
 B No

Answer: B

Q54.

You manage an on-premises server that runs Windows Server 2016. The server has a disk that contains 4 terabytes (TB) of data and thousands of files. None of the individual files are larger than 1 TB. You plan to create a virtual machine (VM) in Azure to process the workload currently handled by the on-premises server.

You need to create a storage location for the data.

What should you do?

- A Create premium storage account. Use a D-series VM.
- B Configure a StorSimple virtual array. Configure the VM to use the array with the SMB protocol.
- C Add a new table storage account. Update the VM workload to use the table storage.
- D Add a single file share to the VM. In the VM operating system, assign a drive letter.

A54.

You manage an on-premises server that runs Windows Server 2016. The server has a disk that contains 4 terabytes (TB) of data and thousands of files. None of the individual files are larger than 1 TB. You plan to create a virtual machine (VM) in Azure to process the workload currently handled by the on-premises server.

You need to create a storage location for the data.

What should you do?

- A Create premium storage account. Use a D-series VM.
- B Configure a StorSimple virtual array. Configure the VM to use the array with the SMB protocol.
- C Add a new table storage account. Update the VM workload to use the table storage.
- D Add a single file share to the VM. In the VM operating system, assign a drive letter.

Answer: D

Q55.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create two resource groups by using the Azure portal. Create four VMs. Assign two VMs to the first resource group and two to the second group.

Does the solution meet the goal?

- A Yes
 B No

A55.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: Create two resource groups by using the Azure portal. Create four VMs. Assign two VMs to the first resource group and two to the second group.

Does the solution meet the goal?

- A Yes
 B No

Answer: B

Q56.

Tailspin Toys uses a website to manage its inventory. The website is hosted on Azure. You are writing a Windows Store app that uses data from the blob storage.

You need to retrieve an image from the following URI:

<https://tailspintoys.blob.core.windows.net/Trains/Caboose2.jpg>.

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

Answer Area

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

  BlobEndpoint  

  FileEndpoint  

  CreateCloudBlobClient  

  CreateCloudFileClient  

CloudBlobContainer blobContainer =
blobClient.   ("trains");
  GetContainerReference  

  GetBlobReferenceFromServerAsync  

CloudBlockBlob myBlob =
blobContainer.   ("Caboose2.jpg");
  GetBlockBlobReference  

  GetDirectoryReference
using (var fileStream = System.IO.File.OpenWrite
(@"path\myfile"))
{
    myBlob.DownloadToStream(fileStream);
}
```

A56.

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

  BlobEndpoint  

  FileEndpoint  

  CreateCloudBlobClient  

  CreateCloudFileClient  

CloudBlobContainer blobContainer =
blobClient.   ("trains");
  GetContainerReference  

  GetBlobReferenceFromServerAsync  

CloudBlockBlob myBlob =
blobContainer.   ("Caboose2.jpg");
  GetBlockBlobReference  

  GetDirectoryReference
using (var fileStream = System.IO.File.OpenWrite
(@"path\myfile"))
{
    myBlob.DownloadToStream(fileStream);
}
```

Q57.

You host an application on an Azure virtual machine (VM) that uses a data disk. The application performs several input and output operations per second.

You need to disable disk caching for the data disk.

Which two actions will achieve the goal? Each answer presents a complete solution.

- A Use the Azure Resource Manager REST API
- B Use the Service Management REST API.
- C Run the following Windows PowerShell cmdlet: Remove-AzureDataDisk
- D Run the following Windows PowerShell cmdlet: Set-AzureDataDisk

A57.

You host an application on an Azure virtual machine (VM) that uses a data disk. The application performs several input and output operations per second.

You need to disable disk caching for the data disk.

Which two actions will achieve the goal? Each answer presents a complete solution.

- A Use the Azure Resource Manager REST API
- B Use the Service Management REST API.
- C Run the following Windows PowerShell cmdlet: Remove-AzureDataDisk
- D Run the following Windows PowerShell cmdlet: Set-AzureDataDisk

Answer: A, D

Q58.

You plan to run SQL Server Enterprise Edition by using an Azure virtual machine (VM).

You must configure the VM to run all SQL Server high volume workloads.

You need to optimize SQL Server performance for workloads that run on the new VM.

What should you do? To answer, drag the appropriate optimization technique to the correct configuration option. Each optimization technique may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Optimization technique

- Keep the VM in the same region and disable geo-replication.
- Use Premium Storage and enable read caching for data files and TempDB.
- Enable locked pages and instant file initialization for data files.
- Use a DS3-series or higher-level VM.
- Move the VM to a different region and enable geo-replication.
- Enable autogrow and autoshrink.

Answer Area

Configuration option	Optimization technique
VM size	
Storage	
Disks	
I/O	

A58.

Optimization technique

- Keep the VM in the same region and disable geo-replication.
- Use Premium Storage and enable read caching for data files and TempDB.
- Enable locked pages and instant file initialization for data files.
- Use a DS3-series or higher-level VM.
- Move the VM to a different region and enable geo-replication.
- Enable autogrow and autoshrink.

Answer Area

Configuration option	Optimization technique
VM size	Use a DS3-series or higher-level VM.
Storage	Keep the VM in the same region and disable geo-replication.
Disks	Use Premium Storage and enable read caching for data files and TempDB.
I/O	Enable locked pages and instant file initialization for data files.

Q59.

You manage API management policies in Azure.

You attempt to add a policy that is marked as unavailable.

You need to ensure that you can add the desired policy.

What should you do?

- A Modify the API Management policy definition.
- B Enable custom caching for the API Management service.
- C Modify the scope of the API policy.
- D Integrate the API Management service with the Azure Event Hub service.

A59.

You manage API management policies in Azure.

You attempt to add a policy that is marked as unavailable.

You need to ensure that you can add the desired policy.

What should you do?

- A Modify the API Management policy definition.
- B Enable custom caching for the API Management service.
- C Modify the scope of the API policy.
- D Integrate the API Management service with the Azure Event Hub service.

Answer: C

Q60.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<pointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
```

```

PC29
PC30     public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31     {
PC32         using (var client = new DocumentClient(new Uri(EndpointUrl),
PC33             AuthorizationKey))
PC33         {
PC34             var collection = await GetCollection();
PC35             await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36             return pg;
PC37         }
PC38     }
PC39     private static async Task<DocumentCollection> GetCollection()
PC40     {
PC41         ...
PC42     }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

PG01 public class PointGenerator
PG02 {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10     public DateTimeOffset DateLastClaimed { get; set; }
PG12     [JsonProperty("location")]
PG12     public Point Location { get; set; }
PG13 }
```

You need to write an Azure Search Query to return data for the first subsection of the Daily Sponsor Report.

Which query string should you use?

- A facets=currentOwner&sort=dateLastClaimed
- B \$filter=sponsor&sort=dateLastClaimed
- C search=currentOwner&sort=dateLastClaimed
- D group=sponsor&sort=dateLastClaimed
- E facets=sponsor&sort=dateLastClaimed

A60.

You need to write an Azure Search Query to return data for the first subsection of the Daily Sponsor Report.

Which query string should you use?

- A facets=currentOwner&sort=dateLastClaimed
- B \$filter=sponsor&sort=dateLastClaimed
- C search=currentOwner&sort=dateLastClaimed
- D group=sponsor&sort=dateLastClaimed
- E facets=sponsor&sort=dateLastClaimed

Answer: C

Q61.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to Azure datacenters on two continents.

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Application Gateway.

Does the solution meet the goal?

- A Yes
 B No

A61.

You develop an enterprise application that will be used only by the employees of a company. The application is not Internet-facing. You deploy instances of the application to Azure datacenters on two continents.

You must implement a load balancing solution that meets the following requirements:

- Provide network-level distribution of traffic across all instances of the application.
- Support HTTP and HTTPS protocols.
- Manage all inbound and outbound connections.

Any back-end virtual machine (VM) must be able to service requests from the same user or client session.

Solution: You implement Application Gateway.

Does the solution meet the goal?

- A Yes
 B No

Answer: B

Explanation:

Traffic Manager is also needed. It allows you to control the distribution of user traffic for service endpoints in different datacenters

Q62.

You need to implement the StartNotify method in MainPage.xaml.cs to enable the receiving of notifications.

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

NOTE: Each correct selection is worth point.



Code segments

PushNotificationChannelManager
Queue
NotificationHub
Registration
moderationnotify
commentQueue

Answer area

```
var a = await  
    [REDACTED].CreatePushNotificationChannelForApplicationAsync();  
  
var b = new [REDACTED] ("[REDACTED]", "...");  
  
var result = await b.RegisterNativeAsync(a.Uri);  
if (result.RegistrationId != null)  
{  
    UpdateUI();  
}
```

A62.

Code segments

PushNotificationChannelManager
Queue
NotificationHub
Registration
moderationnotify
commentQueue

Answer area

```
var a = await  
    PushNotificationChannelManager.CreatePushNotificationChannelForApplicationAsync();  
  
var b = new NotificationHub ("[REDACTED]", "moderationnotify", "...");  
  
var result = await b.RegisterNativeAsync(a.Uri);  
if (result.RegistrationId != null)  
{  
    UpdateUI();  
}
```

Q63.

You create a web application. You publish the source code of the web application to a GitHub repository by using Microsoft Visual Studio. You create a website by using the Azure management portal.

You must continuously deploy the web application from the GitHub repository website to the Azure website.

You need to deploy the source code of the web application.

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

Actions	Answer Area
Select the repository and the branch from which to deploy the Azure website.	
Select GitHub as the source control method.	
Configure the Azure website to use the Always On option.	
In the Azure management portal, configure web endpoint monitoring.	
In the Azure management portal, choose the option to set up deployment from source control.	
Sign in to GitHub by using your deployment credentials.	

A63.

Actions	Answer Area
Select the repository and the branch from which to deploy the Azure website.	In the Azure management portal, choose the option to set up deployment from source control.
Select GitHub as the source control method.	Select GitHub as the source control method.
Configure the Azure website to use the Always On option.	Sign in to GitHub by using your deployment credentials.
In the Azure management portal, configure web endpoint monitoring.	Select the repository and the branch from which to deploy the Azure website.
In the Azure management portal, choose the option to set up deployment from source control.	
Sign in to GitHub by using your deployment credentials.	

Q64.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<PointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29 }
```

```

PC29
PC30     public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31     {
PC32         using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33         {
PC34             var collection = await GetCollection();
PC35             await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36             return pg;
PC37         }
PC38     }
PC39     private static async Task<DocumentCollection> GetCollection()
PC40     {
PC41         ...
PC42     }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

PG01 public class PointGenerator
PG02 {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10     public DateTimeOffset DateLastClaimed { get; set; }
PG11     [JsonProperty("location")]
PG12     public Point Location { get; set; }
PG13 }
```

You need to write a method to return the email address for a given sponsor.

What should you do? To answer, drag the appropriate code segment to the correct location. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Code segments

<https://graph.windows.net/litwaregamerdata>

<https://litwaregamerdata.onmicrosoft.com/graph>

<https://graph.windows.net/litwareinc>

<https://litwareinc.onmicrosoft.com/graph>

/users?api-version=1.6&\$filter=displayName eq \'{pg.Sponsor}\'

/users?api-version=1.6&\$filter=displayName eq \'{pg.CurrentOwner}\'

result.value[0].mail

result.value[0].sipProxyAddress

Answer Area

```
private async Task<string> getEmail(PointGenerator pg, string accessToken)
{
    var url = "https://graph.microsoft.com/v1.0/me";
    url += $"?$select=mail";
    var http = new HttpClient();
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization =
        new AuthenticationHeaderValue("Bearer", accessToken);
    var response = await http.SendAsync(request);
    dynamic result =
        JsonConvert.DeserializeObject(await response.Content.ReadAsStringAsync());
    return result.mail;
}
```

A64.

```
private async Task<string> getEmail(PointGenerator pg, string accessToken)
{
    var url = "https://graph.windows.net/litwaregamerdata";
    url += $"?api-version=1.6&$filter=displayName eq '{pg.Sponsor}'";
    var http = new HttpClient();
    var request = new HttpRequestMessage(HttpMethod.Get, url);
    request.Headers.Authorization =
        new AuthenticationHeaderValue("Bearer", accessToken);
    var response = await http.SendAsync(request);
    dynamic result =
        JsonConvert.DeserializeObject(await response.Content.ReadAsStringAsync());
    return result.value[0].mail;
}
```

Q65.

You are developing a messaging solution to integrate two applications named WeatherSummary and WeatherDetails. The WeatherSummary application displays a summary of weather information for major cities. The WeatherDetails application displays weather details for a specific city.

You need to ensure that the WeatherDetails application displays the weather details for the city that the user selects in the WeatherSummary application.

What should you do?

- A Create an Azure Service Bus Queue communication. In the WeatherDetails application, implement the PeekLock method.
- B Create an Azure Service Bus Topics object. In the WeatherDetails application, create a filter.
- C Create an Azure Service Bus Relay object. In the WeatherDetails application, create a filter.
- D Create an Azure Service Bus Queue communication. In the WeatherDetails application, implement the ReceiveAndDelete method.

A65.

You are developing a messaging solution to integrate two applications named WeatherSummary and WeatherDetails. The WeatherSummary application displays a summary of weather information for major cities. The WeatherDetails application displays weather details for a specific city.

You need to ensure that the WeatherDetails application displays the weather details for the city that the user selects in the WeatherSummary application.

What should you do?

- A Create an Azure Service Bus Queue communication. In the WeatherDetails application, implement the PeekLock method.
- B Create an Azure Service Bus Topics object. In the WeatherDetails application, create a filter.
- C Create an Azure Service Bus Relay object. In the WeatherDetails application, create a filter.
- D Create an Azure Service Bus Queue communication. In the WeatherDetails application, implement the ReceiveAndDelete method.

Answer: B

Q66.

You develop a web application that will use the Azure Table service. The web application will store entities in the form of XML data within a single table.

The web application must support high traffic throughput.

You need to avoid exceeding the throttle limit for the table.

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

- A Add additional partition keys to the table.
- B Batch transactions for entities that are in the same partition group in the table.
- C Compress the entities before storing them in the table.
- D Store the entities in JSON format.

A66.

You develop a web application that will use the Azure Table service. The web application will store entities in the form of XML data within a single table.

The web application must support high traffic throughput.

You need to avoid exceeding the throttle limit for the table.

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

- A Add additional partition keys to the table.
- B Batch transactions for entities that are in the same partition group in the table.
- C Compress the entities before storing them in the table.
- D Store the entities in JSON format.

Answer: B, D

Q67.

You develop a service that runs on a worker role in Azure. The service caches a large amount of data from a database at startup. The service has a configuration file that includes two settings named ConnectionString and SleepInterval.

The service must restart when the value of the ConnectionString setting changes. The service must NOT restart when the value of the SleepInterval setting changes.

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

```
01 public class WorkerRole : RoleEntryPoint
02 {
03     int _sleepInterval = 10000;
04     string _connString = "Server=tcp:contoso.database.windows.net;Database=db1;
05     User ID=sa@contoso;Password=password123!;
06     Trusted_Connection=True;Encrypt=True;";
07     public override void Run()
08     {
09         CacheTableData(_connString);
10         while (true)
11         {
12             Thread.Sleep(10000);
13             ProcessQueueMessages();
14         }
15     }
16     public override bool OnStart()
17     {
18         RoleEnvironment.Changing += RoleEnvironment_Changing;
19         return base.OnStart();
20     }
21     void RoleEnvironment_Changing(object sender, RoleEnvironmentChangingEventArgs e)
22     {
23 }
```

You need to configure the service.

Which code segment should you insert at line 21?

- A.

```
var settingChanges = e.Changes.OfType<RoleEnvironmentConfigurationSettingChange>();
if (settingChanges.Any(chg => chg.ConfigurationSettingName == "ConnectionString"))
{
    e.Cancel = true;
}
```
- B.

```
var newValue = RoleEnvironment.GetConfigurationSettingValue("ConnectionString");
if (newValue == _connString)
{
    e.Cancel = false;
}
```
- C.

```
var settingChanges = e.Changes.OfType<RoleEnvironmentConfigurationSettingChange>();
if (settingChanges.Any(chg => chg.ConfigurationSettingName == "ConnectionString"))
{
    e.Cancel = false;
}
```
- D.

```
var newValue = RoleEnvironment.GetConfigurationSettingValue("ConnectionString");
if (newValue == _connString)
{
    e.Cancel = true;
}
```

A67.

A Option A

B Option B

C Option C

D Option D

Answer: A

Q68.

You have an on-premises Windows Identity Foundation (WIF) application. A section of the application uses resources that are hosted in Azure. The application uses Azure Active Directory (Azure AD) to control access to the section of the application that accesses Azure resources. You synchronize all user principals to Azure Active Directory.

The application has the following requirements:

- Use Windows integrated credentials for single sign-on (SSO).
- Use Azure Active Directory as an identity provider.

You need to create an endpoint to use for web sign-in to the secured section of the application.

Which endpoint should you use?

- A SAML-P
- B OAuth
- C Azure AD Graph API
- D WS-Federation

A68.

You have an on-premises Windows Identity Foundation (WIF) application. A section of the application uses resources that are hosted in Azure. The application uses Azure Active Directory (Azure AD) to control access to the section of the application that accesses Azure resources. You synchronize all user principals to Azure Active Directory.

The application has the following requirements:

- Use Windows integrated credentials for single sign-on (SSO).
- Use Azure Active Directory as an identity provider.

You need to create an endpoint to use for web sign-in to the secured section of the application.

Which endpoint should you use?

- A SAML-P
- B OAuth
- C Azure AD Graph API
- D WS-Federation

Answer: D

Q69.

Background:

You are a developer for ProseWare Inc., a software-as-a-service (SaaS) company that provides a comment system that websites use to allow for end users to post comments associated with a webpage or topic on a customer's website.

Business requirements

Moderation:

The moderation of comments is a feature of the software, and usually involves the editing of a comment.

Only users who have accounts in a group in Azure Active Directory (Azure AD) have the ability moderate. External users can also become moderators, but only by explicit invitation.

Any moderation action must include the name of the moderator.

Comment navigation:

Each comment is identified by a unique string consisting of a random string of characters.

Within the body of a comment, internal links to other comment threads can be specified using the link format: "<parent comment id> / <child comment id>"

Comment search:

Comments can be searched using Azure Search. Searches must do the following:

- Searching for email addresses must match email addresses in comments.
- Searching must work for the client's language.
- Internal links to other comments using the link format should be searched.

Content screening:

Comment content is screened for inappropriate language, length, and topic using content analysis. Content must be screened, but can appear prior to be screened.

Mobile App:

The moderation functionality can be accessed using a Universal Windows Platform (UWP) app named ProsewareApp. The app includes functionality that notifies moderators when changes are made to a comment they modified.

Export:

Customers can perform an export of all comments to a customer supplied Microsoft OneDrive folder on demand. The export functionality is implemented as an Azure Logic App, and it must be able to be triggered by the customer from their local network.

Interaction agents:

Interaction agents are parts of the system that interact with comment threads. The main purpose is to modify a comment's body based on the contents of the comment. For example, one of the agents is WikiAgent, which adds links to Wikipedia articles when it sees text in the comment body that exactly matches a Wikipedia article title. Interaction Agents are implemented in Service Fabric.

Interaction agents must meet the following requirements:

- Only successfully process each comment once
- Any errors encountered during the processing of a comment should be retried
- Must run on systems that allow for custom applications to be installed
- Must run in a VNet or private network space
- Must be run on a system that can scale up and down based on demand
- A single user's usage of Interaction Agents must not impact other users' usage of Interaction Agents

Technical requirements

Authentication:

ProseWare Inc. allows for user authentication through Azure AD and Twitter.

Storage:

The application runs as a Web App on Azure. Comments are stored in an Azure DocumentDB database named "Proseware".

Performance:

The product includes a service level agreement (SLA) for individual method performance. All data retrieval methods must return within 100ms 99% of the time.

API:

The ProseWare Inc. API is made available to public callers using an Azure API App. Azure AD and Twitter are the Authentication Providers.

Application structure

CommentController.cs:

```
CC01 [Route("api/[controller]")]
CC02 public class CommentController : Controller
CC03 {
CC04     private IDatabase _redis;
CC05     private DataStore _dataStore;
CC06     private CloudQueue _queue
CC07
CC08     public CommentController ()
CC09     {
CC10         _queue = CloudStorageAccount.Parse(" ").Create-
CloudQueueClient().GetQueueReference ("commentQueue");
CC11         _redis = ConnectionMultiplexer.Connect("...").GetData-
base();
CC12         _dataStore = new DataStore();
CC13     }
CC14
CC15     [HttpGet("{commentId}")]
CC16     public async Task<Comment> Get(string commentId)
CC17     {
CC18         var cached = await _redis.StringGetAsync(commentId);
CC19         if (cached.HasValue)
CC20         {
CC21             return JsonConvert.DeserializeObject<Com-
ment>(cached.ToString());
CC22         }
CC23         return await _dataStore.LoadAsync(commentId);
CC24     }
CC25
CC26     [HttpGet]
CC27     public IEnumerable<Comment> GetChildComments(string com-
mentId)
CC28     {
CC29         IEnumerable<Comment> result = null;
CC30
CC31         if (result == null)
CC32         {
CC33             result = _dataStore.LoadThread(commentId);
CC34         }
CC35         return results;
CC36     }
CC37
CC38     [HttpPost]
CC39     public async Task<IActionResult> New([FromBody]Comment
comment)
CC40     {
CC41         await Save(comment);
CC42         return Ok();
CC43     }
CC44
CC45     [HttpPost]
CC46     public async Task<IActionResult> Reply(string inRe-
sponseTo, [FromBody]Comment comment)
CC47     {
CC48         comment.InResponseTo = inResponseTo;
CC49         await Save(comment);
CC50         return View();
CC51     }
CC52
```

```

CC53    private static Comment Convert(string json)
CC54    {
CC55        return JsonConvert.DeserializeObject<Comment>(json);
CC56    }
CC57
CC58    private async Task Save(Comment comment, string moderator-
Name = null)
CC59    {
CC60        comment.Moderator = moderatorName;
CC61        var json = JsonConvert.SerializeObject(comment);
CC62        _redis.StringSet(comment.Id, json);
CC62
CC63
CC64        await _queue.AddMessageAsync(new CloudQueueMessage(com-
ment.Id));
CC65        _dataStore.Save(comment);
CC66    }
CC67 }

```

cleaner.csx:

```

CL01 #r "Newtonsoft.Json"
CL02
CL03 using System;
CL04 using Newtonsoft.Json;
CL05 using Newtonsoft.Json.Linq;
CL06 public static void Run(string commentId, object result,
TraceWriter log)
CL07 {
CL08     dynamic comment = JObject.Parse(item);
CL09 ...
CL10     result = comment;
CL11 }

```

ICommentAgent.cs:

```

CA01 public interface ICommentAgent: IActor
CA02    {
CA03        Task<string> ModifyCommentText(string id, string body,
string title);
CA04    }

```

WikiAgent.cs:

```

WA01 [StatePersistence(StatePersistence.Persisted)]
WA02 internal class WikiAgent : Agent, ICommentAgent
WA03 {
WA04     public WikiAgent(ActorService, ActorId id) : base(service,
id) {}
WA05     public async Task<string> ModifyCommentText(string id,
string body, string title)
WA06    {
WA07        try
WA08        {
WA09            var newBody = scanForLinks(body);
WA11
WA12            return newBody;
WA13        }
WA14        catch
WA15        {
WA16            throw
WA18        }
WA19    }
WA20 }

```

Comment.cs:

```
CO01 public class Comment
CO02 {
CO03     public string Id {get; set;}
CO04     public string UserId {get; set;}
CO05     public string InResponseTo {get; set;}
CO06     public string Title {get; set;}
CO07     public DateTimeOffset Date {get; set;}
CO08     public string Body {get; set;}
CO09     public string Moderator {get; internal; set;}
CO10 }
```

DataStore.cs:

```
DS01 public class DataStore
DS02 {
DS03     private const string EndpointUrl = "https://
//proseware.documents.azure.com:443/";
DS04     private const string PrimaryKey = "";
DS05     private const string db = "Proseware";
DS06     private const string col = "Comments";
DS07     private DocumentClient client;
DS08
DS09     public DataStore()
DS10     {
DS11         client = new DocumentClient(new Uri(EndpointUrl), Pri-
maryKey);
DS12     }
DS13
DS14     public async Task<Comment> LoadAsync(string commentId)
DS15     {
DS16         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS17         return await client.ReadDocumentAsync<Comment>(UriFac-
tory.CreateDocumentUri(db, col, commentId));
DS18     }
DS19
DS20     public async void Save(Comment comment)
DS21     {
DS22         var uri = UriFactory.CreateDocumentCollectionUri(db,
col, comment.Id);
DS23         await client.UpsertDocumentAsync(uri, comment);
DS24     }
DS25     public IEnumerable<Comment> LoadThread(string commentId)
DS26     {
DS27         var uri = UriFactory.CreateDocumentCollectionUri(db,
col);
DS28         return client.CreateDocumentQuery<Comment>(uri).Where(f
=> f.Id == commentId);
DS29     }
}
```

MainPage.xaml.cs:

```
MP01 public sealed partial class MainPage : Page
MP02 {
MP03     public MainPage()
MP04     {
MP05         InitializeComponent();
MP06     }
MP07
MP08     private async void StartNotify()
MP09     {
MP10     }
MP11
MP12     private void UpdateUI()
MP13     {
MP14     }
MP15 }
```

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

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

You need to implement the infrastructure for the Interaction Agents.

Solution: Create an Azure Container Service cluster and create a container for running Service Fabric.

Does the solution meet the goal?

- A Yes
- B No

A69.

You need to implement the infrastructure for the Interaction Agents.

Solution: Create an Azure Container Service cluster and create a container for running Service Fabric.

Does the solution meet the goal?

- A Yes
- B No

Answer: A

Q70.

Your company works with trusted partners. These partners upload files into a storage account that you control.

Partners must be able to create, read, and write files. Partners must NOT be allowed to see files from other partners. You generate a shared access signature (SAS) for each partner.

You create the following Windows PowerShell script to create a new container for each partner. Line numbers are included for reference only.

```
01 $containerName = "partner123files"
02 $key = (Get-AzureStorageKey -StorageAccountName $storageAccountName).Primary
03 $context = New-AzureStorageContext -StorageAccountName $storageAccountName
   -StorageAccountKey $key
04 New-AzureStorageContainer -Name $containerName -Context $context
05 $filepath = "welcome.txt"
06 $blobname = "welcome.txt"
07 Set-AzureStorageBlobContent -Container $containerName -File "$filepath"
   -Blob $blobname -Context $context -Properties @{"ContentType"="text/plain"}
08 $oneYearFromNow = (Get-Date).AddYears(1)
09 $sasToken = New-AzureStorageContainerSASToken -Name $containerName
   -Permission 'rwdl' -ExpiryTime $oneYearFromNow -Context $context
10 $sasBlobUri = New-AzureStorageBlobSASToken -Container $containerName
   -Permission 'r' -ExpiryTime $oneYearFromNow -Context $context
   -FullUri -Blob $blobname
```

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

Answer Area

Yes	No
------------	-----------

Running the command at line 10 a second time invalidates the previously generated SAS token.

Web browsers can open the welcome.txt file directly by using the full URI and the SAS token for the file.

If the primary storage key is regenerated, the SAS token is still valid until its expiration date is reached.

A70.

Answer Area

Yes	No
------------	-----------

Running the command at line 10 a second time invalidates the previously generated SAS token.

Web browsers can open the welcome.txt file directly by using the full URI and the SAS token for the file.

If the primary storage key is regenerated, the SAS token is still valid until its expiration date is reached.

Q71.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18
PC19         [Route("nearby")]
PC20         [HttpGet]
PC21         public IEnumerable<pointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22         {
PC23             var location = new Point(longitude, latitude);
PC24             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29 }
```

```

PC30     public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31     {
PC32         using (var client = new DocumentClient(new Uri(EndpointUrl),
PC33             AuthorizationKey))
PC33         {
PC34             var collection = await GetCollection();
PC35             await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36             return pg;
PC37         }
PC38     }
PC39     private static async Task<DocumentCollection> GetCollection()
PC40     {
PC41         ...
PC42     }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

PG01 public class PointGenerator
PG02 {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10     public DateTimeOffset DateLastClaimed { get; set; }
PG11     [JsonProperty("location")]
PG12     public Point Location { get; set; }
PG13 }
```

You need to decrease the amount of time it takes to query point generators by configuring API management caching.

In the Azure portal, which value should you use for the Vary by Query string parameters setting?

- A name
- B longitude;latitude;minDistance
- C longitude;latitude;dateLastClaimed
- D Id

A71.

You need to decrease the amount of time it takes to query point generators by configuring API management caching.

In the Azure portal, which value should you use for the Vary by Query string parameters setting?

- A name
- B longitude;latitude;minDistance
- C longitude;latitude;dateLastClaimed
- D Id

Answer: B

Q72.

An application sends Azure push notifications to a client application that runs on Windows Phone, iOS, and Android devices.

Users cannot use the application on some devices. The authentication mechanisms that the application uses are the source of the problem.

You need to monitor the number of notifications that failed because of authentication errors.

Which three metrics should you monitor? Each correct answer presents part of the solution.

- A Microsoft Push Notification Service (MPNS) authentication errors
- B External notification system errors
- C Apple Push Notification Service (APNS) authentication errors
- D Channel errors
- E Windows Push Notification Services (WNS) authentication errors
- F Google Cloud Messaging (GCM) authentication errors

A72.

An application sends Azure push notifications to a client application that runs on Windows Phone, iOS, and Android devices.

Users cannot use the application on some devices. The authentication mechanisms that the application uses are the source of the problem.

You need to monitor the number of notifications that failed because of authentication errors.

Which three metrics should you monitor? Each correct answer presents part of the solution.

- A Microsoft Push Notification Service (MPNS) authentication errors
- B External notification system errors
- C Apple Push Notification Service (APNS) authentication errors
- D Channel errors
- E Windows Push Notification Services (WNS) authentication errors
- F Google Cloud Messaging (GCM) authentication errors

Answer: A, C, F

Q73.

You are managing an application. The application uses data that is stored in an Azure SQL database.

You must be able to reset the application to the state that existed on any day in the previous 35 days.

You need to choose a backup solution.

What should you do?

- A Run SQL replication on the SQL database once a day.
- B Use Microsoft Azure SQL Database Point in Time Restore
- C Use the SQL Server Data-Tier Application Framework to build a data-tier application (DAC) file once a day.
- D Use the bcp utility to export data to an Azure page blob once a day.

A73.

You are managing an application. The application uses data that is stored in an Azure SQL database.

You must be able to reset the application to the state that existed on any day in the previous 35 days.

You need to choose a backup solution.

What should you do?

- A Run SQL replication on the SQL database once a day.
- B Use Microsoft Azure SQL Database Point in Time Restore
- C Use the SQL Server Data-Tier Application Framework to build a data-tier application (DAC) file once a day.
- D Use the bcp utility to export data to an Azure page blob once a day.

Answer: B

Q74.

You administer an Azure-based solution that performs image processing. You have four Standard D3 Azure Resource Manager (ARM) virtual machines (VMs). All VMs are deployed in a Virtual Machine Scale Set (VMSS).

The servers must scale up or down as the workload increases or decreases.

You need to configure auto-scaling to scale the VMSS when the server workload is above 95 percent or below 5 percent.

What should you do?

- A Navigate to the VM's Size panel and increase the instance count.
- B Navigate to the VMSS Metric panel and add a new alert for the CPU Percentage Metric. Configure the alert to notify Via email.
- C Navigate to the VM's Metric panel and enable diagnostics for basic metrics,
- D Navigate to the VMSS Metric panel and add a new alert for the CPU Percentage Metric. Configure the alert to notify via webhook.

A74.

You administer an Azure-based solution that performs image processing. You have four Standard D3 Azure Resource Manager (ARM) virtual machines (VMs). All VMs are deployed in a Virtual Machine Scale Set (VMSS).

The servers must scale up or down as the workload increases or decreases.

You need to configure auto-scaling to scale the VMSS when the server workload is above 95 percent or below 5 percent.

What should you do?

- A Navigate to the VM's Size panel and increase the instance count.
- B Navigate to the VMSS Metric panel and add a new alert for the CPU Percentage Metric. Configure the alert to notify Via email.
- C Navigate to the VM's Metric panel and enable diagnostics for basic metrics,
- D Navigate to the VMSS Metric panel and add a new alert for the CPU Percentage Metric. Configure the alert to notify via webhook.

Answer: D

Q75.

You are developing a REST API service that provides data about products. The service will be hosted in an Azure virtual machine (VM).

The product data must be stored in Azure tables and replicated to multiple geographic locations.

API calls that use the HTTP GET operation must continue to function when the data tables at the primary Azure datacenter are not accessible.

You need to configure storage for the service.

Which type of replication should you choose?

- A Locally Redundant Storage replication
- B Geo-Redundant Storage replication
- C Zone-Redundant Storage replication
- D Read-Access Geo-Redundant Storage replication

A75.

You are developing a REST API service that provides data about products. The service will be hosted in an Azure virtual machine (VM).

The product data must be stored in Azure tables and replicated to multiple geographic locations.

API calls that use the HTTP GET operation must continue to function when the data tables at the primary Azure datacenter are not accessible.

You need to configure storage for the service.

Which type of replication should you choose?

- A Locally Redundant Storage replication
- B Geo-Redundant Storage replication
- C Zone-Redundant Storage replication
- D Read-Access Geo-Redundant Storage replication

Answer: D

Q76.

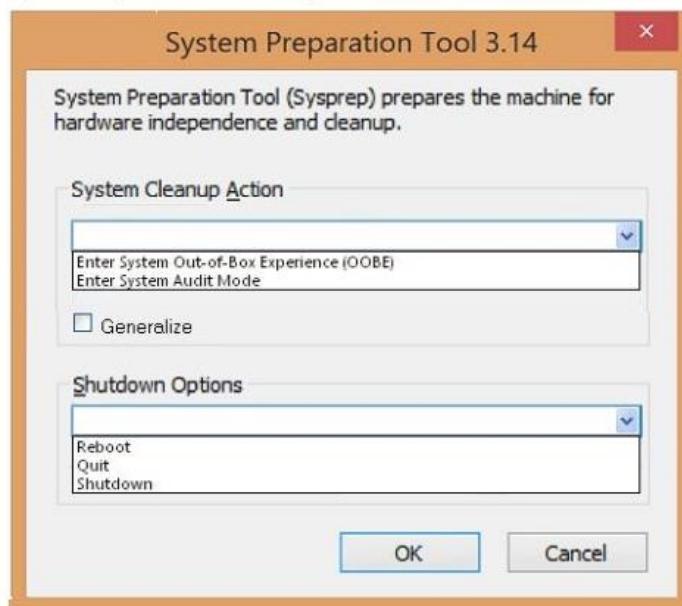
You have an existing server that runs Windows Server. You plan to create a base image of this server. You will use this base image to prepare several virtual servers for future use. After the base image is prepared, you will capture it by using the Azure management portal.

You must use the System Preparation Tool (Sysprep) to prepare the server so that the base image can be captured.

You need to prepare the server so that the base image can be captured.

What should you do? To answer, configure the appropriate options in the dialog box in the answer area.

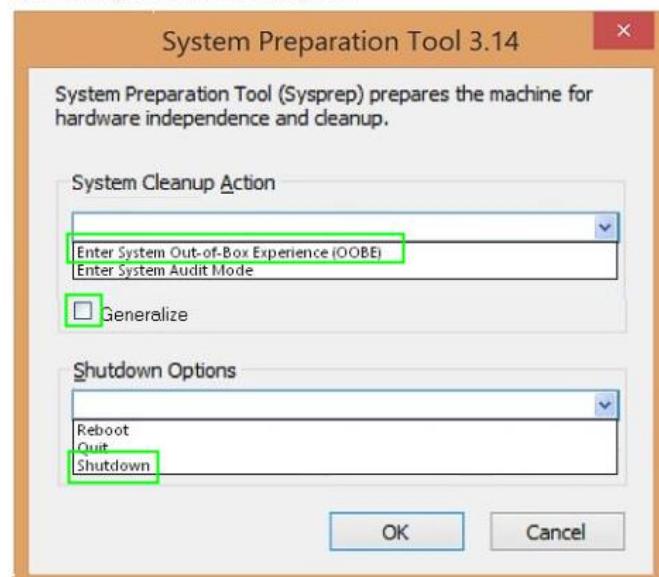
System Preparation Tool dialog box



A76.

Answer:

System Preparation Tool dialog box



Q77.

Contoso has an Azure DocumentDB database that contains contact information for customers.

You have a collection named Companies. The collection includes the following data:

```
{  
  "id" : "ContosoCompany",  
  "name" : "Contoso",  
  "contacts" : [  
    {  
      "giveName" : "Lola",  
      "surName" : "Jacobsen",  
      "regions" : [  
        { "regionName" : "West" },  
        { "regionName" : "South" }  
      ]  
    },  
    {  
      "giveName" : "David",  
      "surName" : "Jones",  
      "regions" : [  
        { "regionName" : "North" },  
        { "regionName" : "South" }  
      ]  
    }  
,  
  ],  
  "address" : {"state": "co", "city": "Denver"}  
}
```

You plan to collect the following information for contacts that are located in the South region only:

- Company name
- Given name
- Surname

You need to create the query.

Which three Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

Transact-SQL segments

```
FROM Contoso c  
SELECT Name, givenName, surName  
SELECT contoso.Name, c.contacts.givenName,  
c.contacts.surName  
WHERE regionName = 'South'  
WHERE c.contacts.regions.regionName = 'South'  
FROM Companies c  
SELECT c.Name, c.contacts.givenName,  
c.contacts.surName
```

Answer area



A77.

Transact-SQL segments

```
FROM Contoso c
SELECT Name, givenName, surName
SELECT contoso.Name, c.contacts.givenName,
c.contacts.surName
WHERE regionName = 'South'
WHERE c.contacts.regions.regionName = 'South'
FROM Companies c
SELECT c.Name, c.contacts.givenName,
c.contacts.surName
```

Answer area

```
SELECT Name, givenName, surName
FROM Contoso c
WHERE c.contacts.regions.regionName = 'South'
```



Q78.

The Azure Queue service hosts a queue named userRegistrationQueue. You are developing a web job to process messages from the queue. You create a new console application by using Microsoft Visual Studio. You also create an Azure storage connection string and store the connection string in the application configuration file.

All trigger listeners and jobs must run on the current thread.

You need to ensure that the web job processes the messages from the queue.

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

Code Segments

```
var host = new Microsoft.Azure.Jobs.JobHost();
var host = new Microsoft.Azure.Jobs.JobHostConfiguration();
host.RunOnBackgroundThread();
host.RunAndBlock();
host.GetService(typeof(CloudQueue));
host.NameResolver.Resolve("userRegistrationQueue");
```

Answer Area

```
static void Main()
{
    var cloudQueue = CreateCloudQueue();
    AddMessageToQueue(cloudQueue);
}
```

A78.

Code Segments

```
var host = new Microsoft.Azure.Jobs.JobHost();
var host = new Microsoft.Azure.Jobs.JobHostConfiguration();
host.RunOnBackgroundThread();
host.RunAndBlock();
host.GetService(typeof(CloudQueue));
host.NameResolver.Resolve("userRegistrationQueue");
```

Answer Area

```
static void Main()
{
    var cloudQueue = CreateCloudQueue();
    AddMessageToQueue(cloudQueue);
    var host = new Microsoft.Azure.Jobs.JobHost();
    host.RunAndBlock();
}
```

Q79.

You are creating a set of load-balanced virtual machines (VMs) that are hosted on Azure.

You run the following Windows PowerShell script. Line numbers are included for reference only.

```
01 Add-AzureInternalLoadBalancer -ServiceName "Contoso-Chicago" -InternalLoadBalancerName "Data-LB"
   -SubnetName "DataFarm1" -StaticVNetIPAddress 192.168.100.10
02 Get-AzureVM -ServiceName "Contoso-Chicago" -Name "DATA1" | Add-AzureEndpoint -Name "DataFarm"
   -Protocol "TCP" -LocalPort 1433 -PublicPort 1337 -DefaultProbe -InternalLoadBalancerName "Data-LB" | Update-AzureVM
03 Get-AzureService -ServiceName "Contoso-Chicago" | Get-AzureInternalLoadBalancer
```

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

Answer Area

	Yes	No
The internal IP address of the VM named DATA1 is 192.168.100.10.	<input type="radio"/>	<input type="radio"/>
The endpoint named DataFarm can be accessed by using external port 1337.	<input type="radio"/>	<input type="radio"/>
The internal load balancer for the Contoso-Chicago service is named Data-LB .	<input type="radio"/>	<input type="radio"/>

A79.

	Yes	No
The internal IP address of the VM named DATA1 is 192.168.100.10.	<input type="radio"/>	<input checked="" type="radio"/>
The endpoint named DataFarm can be accessed by using external port 1337.	<input checked="" type="radio"/>	<input type="radio"/>
The internal load balancer for the Contoso-Chicago service is named Data-LB .	<input type="radio"/>	<input checked="" type="radio"/>

Q80.

You are migrating an existing solution to Azure. The solution includes a user interface tier and a database tier. The user interface tier runs on multiple virtual machines (VMs). The user interface tier has a website that uses Node.js. The user interface tier has a background process that uses Python. This background process runs as a scheduled job. The user interface tier is updated frequently. The database tier uses a self-hosted MySQL database.

The user interface tier requires up to 25 CPU cores. You must be able to revert the user interface tier to a previous version if updates to the website cause technical problems. The database requires up to 50 GB of memory. The database must run in a single VM.

You need to deploy the solution to Azure.

What should you do first?

- A Deploy the entire solution to an Azure Web App. Use a web job that runs continuously to host the database.
- B Configure Microsoft Visual Team Services to continuously deploy the user interface tier to the Azure Web App service. Deploy the production builds and the staging builds of the user interface tier to separate slots.
- C Deploy the entire solution to an Azure Web App. Use a web job that runs continuously to host the user interface tier.
- D Deploy the user interface tier to a VM. Use multiple availability sets to continuously deploy updates from Microsoft Visual Studio Online.

A80.

You are migrating an existing solution to Azure. The solution includes a user interface tier and a database tier. The user interface tier runs on multiple virtual machines (VMs). The user interface tier has a website that uses Node.js. The user interface tier has a background process that uses Python. This background process runs as a scheduled job. The user interface tier is updated frequently. The database tier uses a self-hosted MySQL database.

The user interface tier requires up to 25 CPU cores. You must be able to revert the user interface tier to a previous version if updates to the website cause technical problems. The database requires up to 50 GB of memory. The database must run in a single VM.

You need to deploy the solution to Azure.

What should you do first?

- A Deploy the entire solution to an Azure Web App. Use a web job that runs continuously to host the database.
- B Configure Microsoft Visual Team Services to continuously deploy the user interface tier to the Azure Web App service. Deploy the production builds and the staging builds of the user interface tier to separate slots.
- C Deploy the entire solution to an Azure Web App. Use a web job that runs continuously to host the user interface tier.
- D Deploy the user interface tier to a VM. Use multiple availability sets to continuously deploy updates from Microsoft Visual Studio Online.

Answer: B

Q81.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: You create an availability set that has two fault domains and two update domains by using the Azure portal. You create four virtual machines and assign the new availability set to each VM.

Does the solution meet the goal?

- A Yes
 B No

A81.

You are administrating an Azure environment for your company. You plan to deploy virtual machines (VMs) for a mobile application. You have the following requirements:

- Ensure that all VMs use the **Standard D3** size.
- Ensure that at least two of the four servers must be available at all times.
- Ensure that users of the application do not experience downtime or loss of connection.

You need to configure four VMs for application development.

Solution: You create an availability set that has two fault domains and two update domains by using the Azure portal. You create four virtual machines and assign the new availability set to each VM.

Does the solution meet the goal?

- A Yes
 B No

Answer: A

Q82.

Using https instead of http for accessing a cloud service is considered more secure.

- A True
- B False

A82.

Using https instead of http for accessing a cloud service is considered more secure.

- A True
- B False

Answer: A

Explanation:

HTTPS (also called HTTP over TLS, HTTP over SSL, and HTTP Secure) is a protocol for secure communication over a computer network which is widely used on the Internet.

en.wikipedia.org/wiki/HTTPS">References: <https://en.wikipedia.org/wiki/HTTPS>

Q83.

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

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

You deploy a Virtual Machine Scale Set (VMSS) named CorpWebVMSS to Azure by using Azure PowerShell and set the instance count to 1. The VMSS includes a storage account, load balancer, public IP address, and six Standard_A1 Windows virtual machines (VMs) that run Internet Information Services (IIS). All components are deployed to a resource group named CorpWebRG.

You must increase the instance count to support the increased load on IIS.

You need to manually scale out the number of VMs in the scale set to 5.

Solution: You deploy the following JSON template by using Azure PowerShell:

```
{  
  "$schema": "http://schema.management.azure.com/schemas/2015-01-01-preview/deploymentTemplate.json",  
  "contentVersion": "1.0.0.0",  
  "resources": [  
    {  
      "type": "Microsoft.Compute/virtualMachineScaleSets",  
      "apiVersion": "2016-03-30",  
      "name": "CorpWebVMSS",  
      "location": "[resourceGroup().location]",  
      "sku": {  
        "name": "Standard_A1",  
        "tier": "Standard",  
        "capacity": "5"  
      }  
    }  
  ]  
}
```

Does the solution meet the goal?

A Yes

B No

A83.

Does the solution meet the goal?

A Yes

B No

Answer: A

Q84.

You need to implement exception handling for the ModifyCommentText method in the WikiAgent class.

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

NOTE: Each correct selection is worth point.

Answer Area

```
try
{
    await StateManager.RemoveStateAsync(id);
    if (await StateManager.ContainsStateAsync(id)) return body;
    await StateManager.GetOrAddStateAsync(id, id),
    if (await StateManager.GetStateAsync<string>(id) == null) return body;

    var newBody = ...
    await StateManager.SaveStateAsync();
    await StateManager.ClearCacheAsync();
    await StateManager.AddStateAsync(id, id);
    await StateManager.RemoveStateAsync(id);

    return newBody;
}
catch
{
    await StateManager.SaveStateAsync();
    await StateManager.ClearCacheAsync();
    await StateManager.RemoveStateAsync(id);
    await StateManager.TryRemoveStateAsync(id);

    throw;
}
```

A84.

```
try
{
    await StateManager.RemoveStateAsync(id);
    if (await StateManager.ContainsStateAsync(id)) return body;
    await StateManager.GetOrAddStateAsync(id, id),
    if (await StateManager.GetStateAsync<string>(id) == null) return body;

    var newBody = ...

    await StateManager.SaveStateAsync();
    await StateManager.ClearCacheAsync();
    await StateManager.AddStateAsync(id, id);
    await StateManager.RemoveStateAsync(id);

    return newBody;
}
catch
{
    await StateManager.SaveStateAsync();
    await StateManager.ClearCacheAsync();
    await StateManager.RemoveStateAsync(id);
    await StateManager.TryRemoveStateAsync(id);

    throw;
}
```

Q85.

A company plans to increase its virtual network capacity by adding virtual network subscriptions.

You must increase the number of subscriptions from 3 to 15.

You need to configure the virtual networks.

What should you do?

- A Export and modify the network configuration file. Then import the modified file.
- B Export and modify the service definition file. Then import the modified file.
- C Create and import a new network configuration file.
- D Create a multi-site virtual network.

A85.

A company plans to increase its virtual network capacity by adding virtual network subscriptions.

You must increase the number of subscriptions from 3 to 15.

You need to configure the virtual networks.

What should you do?

- A Export and modify the network configuration file. Then import the modified file.
- B Export and modify the service definition file. Then import the modified file.
- C Create and import a new network configuration file.
- D Create a multi-site virtual network.

Answer: A

Q86.

You deploy a new version of a cloud-service application to a staging slot. The application consists of one web role. You prepare to swap the new version of the application into the production slot. Your Azure account has access to multiple Azure subscriptions. You load the Azure PowerShell cmdlets into the Windows PowerShell command shell. The command shell is NOT configured for certificate-based authentication.

You must use the Windows PowerShell command window to configure the application.

You need to create five instances of the web role.

How should you configure the relevant Windows PowerShell script? To answer, select the appropriate option or options in the answer area.

Answer Area

```
$subscription = 'mysubscription'  
$service = 'myservice'  
$rolename = 'myrole'
```

```
Add-AzureAccount  
Get-AzureAccount -Name $subscription  
Get-AzureAccount
```

```
Select-AzureSubscription -SubscriptionName $subscription  
Set-AzureSubscription -SubscriptionName $subscription  
Set-AzureSubscription -SubscriptionId $subscription
```

```
Set-AzureRole -ServiceName $service -Slot Staging -RoleName $rolename -Count 5  
Set-AzureRole -ServiceName $service -RoleName $rolename -Count 5  
Set-AzureRole -ServiceName $service -Slot Production -RoleName $rolename -Count 5  
Add-AzureWebRole -Name $service -Instances 5
```

A86.

```
$subscription = 'mysubscription'  
$service = 'myservice'  
$rolename = 'myrole'
```

```
Add-AzureAccount  
Get-AzureAccount -Name $subscription  
Get-AzureAccount
```

```
Select-AzureSubscription -SubscriptionName $subscription  
Set-AzureSubscription -SubscriptionName $subscription  
Set-AzureSubscription -SubscriptionId $subscription
```

```
Set-AzureRole -ServiceName $service -Slot Staging -RoleName $rolename -Count 5  
Set-AzureRole -ServiceName $service -RoleName $rolename -Count 5  
Set-AzureRole -ServiceName $service -Slot Production -RoleName $rolename -Count 5  
Add-AzureWebRole -Name $service -Instances 5
```

Q87.

Your company runs existing applications on virtual machines (VMs) that are hosted on Azure.

You are preparing additional Azure services to support the existing applications.

You run the following script. Line numbers are provided for reference only.

```
01 Add-AzureAccount
02 Select-AzureSubscription -SubscriptionName (Get-AzureSubscription)[0].SubscriptionName
03 New-AzureStorageAccount -Location "East US" -StorageAccountName "store314159265"
04 Set-AzureSubscription -CurrentStorageAccountName "store314159265" -SubscriptionName $subscriptionName
05 $vmImageNameDb = 'c290a6b031d841e09f2da759bbabe71f_Oracle-Database-121010.v3-SE-Lnx'
06 $vmImageNameApp = 'a699494373c04fc0bc8f2bb1389d6106_Windows-Server-2012-R2-201405.01-en.us-127GB.vhd'
07 $cs = New-AzureService -ServiceName "myService27182" -Location "East US"
08 $vmConfigDb = New-AzureVMConfig -Name "MyDb" -InstanceSize Large -ImageName $vmImageNameDb | ^
    Add-AzureProvisioningConfig -Linux -LinuxUser 'dbadmin314' -Password 'ou812?_159265' | ^
    Add-AzureDataDisk -CreateNew -DiskSizeInGB 250 -DiskLabel 'dbdata' -LUN 0
09 $vmConfigDb | New-AzureVM -ServiceName "myService27182"
10 $vmConfigApp = New-AzureVMConfig -Name "MyApp" -InstanceSize Medium -ImageName $vmImageNameApp | ^
    Add-AzureProvisioningConfig -Windows -AdminUsername 'winadm314' -Password 'W!3d03_K05t07'
11 $vmConfigApp | New-AzureVM -ServiceName "myService27182"
```

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

Answer Area

	Yes	No
The command in line 11 creates a new VM that has one local data disk that uses Azure blob storage.	<input type="radio"/>	<input type="radio"/>
The VM that runs Linux and the VM that runs Windows can communicate with each other by using internal IP addresses.	<input type="radio"/>	<input type="radio"/>
The VM that runs Windows can accept HTTP requests from the public Internet.	<input type="radio"/>	<input type="radio"/>

A87.

	Yes	No
The command in line 11 creates a new VM that has one local data disk that uses Azure blob storage.	<input type="radio"/>	<input checked="" type="radio"/>
The VM that runs Linux and the VM that runs Windows can communicate with each other by using internal IP addresses.	<input checked="" type="radio"/>	<input type="radio"/>
The VM that runs Windows can accept HTTP requests from the public Internet.	<input checked="" type="radio"/>	<input type="radio"/>

Q88.

A company creates an API and makes it accessible on an Azure website. External partners use the API occasionally. The website uses the Standard web hosting plan. Partners report that the first API call in a sequence of API calls occasionally takes longer than expected to run. Subsequent API calls consistently perform as expected. You need to ensure that all API calls perform consistently.

What should you do?

- A Configure the website to use the Basic web hosting plan.
- B Enable Always On support.
- C Configure the website to automatically scale.
- D Add a trigger to the web.config file for the website that causes the website to recycle periodically.

A88.

A company creates an API and makes it accessible on an Azure website. External partners use the API occasionally. The website uses the Standard web hosting plan. Partners report that the first API call in a sequence of API calls occasionally takes longer than expected to run. Subsequent API calls consistently perform as expected. You need to ensure that all API calls perform consistently.

What should you do?

- A Configure the website to use the Basic web hosting plan.
- B Enable Always On support.
- C Configure the website to automatically scale.
- D Add a trigger to the web.config file for the website that causes the website to recycle periodically.

Answer: B

Q89.

You are using Microsoft Visual Studio to develop an App Service Web App named WebApp.

The app must collect the statistics and details on the application dependencies.

You need to set up, configure, and validate monitoring using Application Insights.

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

Upload the project and run it to generate log data.

Add Application Insights to the solution. Then, add the **Microsoft.ApplicationInsights.TraceListener** NuGet package to the project.

Start a new instance of Azure PowerShell and run the following Azure PowerShell command:

Get-AzureWebSiteLog -Name WebApp -Tail

In the Azure Portal, browse to the Application Insights resource and open **Search**.

Use the Azure Command-Line interface to run the following command:
azure site log tail WebApp

Answer Area

A89.

Answer:

Actions

Upload the project and run it to generate log data.

Add Application Insights to the solution. Then, add the **Microsoft.ApplicationInsights.TraceListener** NuGet package to the project.

Start a new instance of Azure PowerShell and run the following Azure PowerShell command:

Get-AzureWebSiteLog -Name WebApp -Tail

In the Azure Portal, browse to the Application Insights resource and open **Search**.

Use the Azure Command-Line interface to run the following command:
azure site log tail WebApp

Answer Area

Add Application Insights to the solution. Then, add the **Microsoft.ApplicationInsights.TraceListener** NuGet package to the project.

Upload the project and run it to generate log data.

In the Azure Portal, browse to the Application Insights resource and open **Search**.

Q90.

You plan to migrate a website named Contoso from one hosting plan to another hosting plan. The website is currently in a hosting plan named webhostingplan1. You create a resource group named ContosoGroup.

You create the following PowerShell script by using the Azure PowerShell tools. Line numbers are included for reference only.

```
01 $webhostingplan = @{"serverfarm" = "webhostingplan2"}  
02 Set-AzureResource -name Contoso -ResourceGroupName ContosoGroup -ResourceType Microsoft.Web/sites  
    -apiversion 2014-04-01 -PropertyObject $webhostingplan  
03 Get-AzureResource -name Contoso -ResourceGroupName ContosoGroup -ResourceType Microsoft.Web/sites  
    -apiversion 2014-04-01
```

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

Answer Area

	Yes	No
The command in line 01 defines a variable that stores a hash table.	<input type="radio"/>	<input type="radio"/>
The command in line 02 assigns the website to the ContosoGroup resource group.	<input type="radio"/>	<input type="radio"/>
The command in line 02 assigns the website to a hosting plan named webhostingplan2 .	<input type="radio"/>	<input type="radio"/>

A90.

	Yes	No
The command in line 01 defines a variable that stores a hash table.	<input checked="" type="checkbox"/>	<input type="radio"/>
The command in line 02 assigns the website to the ContosoGroup resource group.	<input type="radio"/>	<input checked="" type="checkbox"/>
The command in line 02 assigns the website to a hosting plan named webhostingplan2 .	<input type="radio"/>	<input checked="" type="checkbox"/>

Q91.

A company maintains an Azure storage account. The storage account uses blobs and tables.

Customers access the storage account by using shared access signatures (SASs).

You need to monitor the usage of the storage services. You need to do the following:

- Understand which storage areas perform operations that incur an Azure fee.
- Understand which requests are denied because of insufficient permissions.
- Validate that the performance of the storage account meets the service level agreement (SLA) for the Azure Storage service.

Which three data analysis tasks should you perform? Each correct answer presents part of the solution.

- A Use data from the logs of the storage services to find individual storage access attempts that do not comply with the SLA.
- B Use data from the logs of the storage services to calculate aggregate server latency across individual requests. Determine whether the results of this calculation indicate that the Azure Storage service is in compliance with the SLA.
- C Analyze the logs of the storage services to determine which storage services were inaccessible because of permissions issues.
- D Review the Azure documentation to determine which storage operations are billable. Then find records of those operations in the logs of the storage services.
- E Analyze the logs of the storage services to find records of operations that are marked as billable.
- F Correlate the data logged from the storage service with the permissions to store data in the individual blobs and containers. Determine which storage services were inaccessible because of permissions issues.

A91.

A company maintains an Azure storage account. The storage account uses blobs and tables.

Customers access the storage account by using shared access signatures (SASs).

You need to monitor the usage of the storage services. You need to do the following:

- Understand which storage areas perform operations that incur an Azure fee.
- Understand which requests are denied because of insufficient permissions.
- Validate that the performance of the storage account meets the service level agreement (SLA) for the Azure Storage service.

Which three data analysis tasks should you perform? Each correct answer presents part of the solution.

- A Use data from the logs of the storage services to find individual storage access attempts that do not comply with the SLA.
- B Use data from the logs of the storage services to calculate aggregate server latency across individual requests. Determine whether the results of this calculation indicate that the Azure Storage service is in compliance with the SLA.
- C Analyze the logs of the storage services to determine which storage services were inaccessible because of permissions issues.
- D Review the Azure documentation to determine which storage operations are billable. Then find records of those operations in the logs of the storage services.
- E Analyze the logs of the storage services to find records of operations that are marked as billable.
- F Correlate the data logged from the storage service with the permissions to store data in the individual blobs and containers. Determine which storage services were inaccessible because of permissions issues.

Answer: B, C, D

Q92.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You administer an Azure environment that includes six Azure Resource Manager (ARM) virtual machines (VMs) that support development. The development team uses Azure SQL databases and Azure Queues for application storage. All Azure resources are grouped within a single subscription and resource group. You need to reduce the recurring monthly Azure costs without degrading server performance. You must minimize the administrative effort involved.

What should you do?

- A Configure an auto-shutdown schedule for each VM by using the Azure Portal.
- B Update the development environment to use Azure Table storage.
- C Create an Azure Automation runbook that compresses unused virtual hard disk (VHD) files daily.
- D Create an Azure PowerShell script that backs up and deprovisions all Azure SQL databases daily.

A92.

You administer an Azure environment that includes six Azure Resource Manager (ARM) virtual machines (VMs) that support development. The development team uses Azure SQL databases and Azure Queues for application storage. All Azure resources are grouped within a single subscription and resource group. You need to reduce the recurring monthly Azure costs without degrading server performance. You must minimize the administrative effort involved.

What should you do?

- A Configure an auto-shutdown schedule for each VM by using the Azure Portal.
- B Update the development environment to use Azure Table storage.
- C Create an Azure Automation runbook that compresses unused virtual hard disk (VHD) files daily.
- D Create an Azure PowerShell script that backs up and deprovisions all Azure SQL databases daily.

Answer: A

Explanation:

You can set any ARM-based Virtual Machines to auto-shutdown with a few simple clicks. This was a feature originally available only to VMs in Azure Dev/Test Labs: your self-service sandbox environment in Azure to quickly create Dev/Test environments while minimizing waste and controlling costs. In case you haven't heard it before, the goal for this service is to solve the problems that IT and development teams have been facing: delays in getting a working environment, time-consuming environment configuration, production fidelity issues, and high maintenance cost. It has been helping our customers to quickly get "ready to test" with a worry-free self-service environment.

Q93.

You are creating virtual machines (VMs) that are hosted on Azure.

You must be able to change the Remote Desktop access settings for the VMs. You must also be able to change the password for the built-in administrator account on all VMs. You identify the VMAccess VM extensions that have the required capabilities.

You need to enable the VMAccess VM extensions.

Which approach should you use?

- A Use Azure PowerShell cmdlets to change the name of the availability set to the same name for all of the VMs.
- B Use the Azure portal to restart each VM.
- C In the Azure Portal, create an image from a virtual hard disk (VHD) for each VM by running Sysprep.
- D For each VM, use Azure PowerShell cmdlets to enable the VM Agent and the VMAccess VM extensions.

A93.

You are creating virtual machines (VMs) that are hosted on Azure.

You must be able to change the Remote Desktop access settings for the VMs. You must also be able to change the password for the built-in administrator account on all VMs. You identify the VMAccess VM extensions that have the required capabilities.

You need to enable the VMAccess VM extensions.

Which approach should you use?

- A Use Azure PowerShell cmdlets to change the name of the availability set to the same name for all of the VMs.
- B Use the Azure portal to restart each VM.
- C In the Azure Portal, create an image from a virtual hard disk (VHD) for each VM by running Sysprep.
- D For each VM, use Azure PowerShell cmdlets to enable the VM Agent and the VMAccess VM extensions.

Answer: D

Q94.

You are administering an Azure environment for your company that requires multiple virtual machines (VMs) for a production application. You have the following requirements:

- Two VMs are required for application data.
- Seven VMs are required for image processing.
- VM sizes should be set to Standard D2.
- Only two image processing servers can be rebooted at a time.

You need to configure an availability set for the image processing VMs.

How many fault domain and update domains should you implement? To answer, configure the appropriate options in the dialog box in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Availability set	Value			
fault domains	<table><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr></table>	1	2	3
1				
2				
3				
update domains	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr></table>	2	3	4
2				
3				
4				

A94.

Answer Area

Availability set	Value			
fault domains	<table><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr></table>	1	2	3
1				
2				
3				
update domains	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr></table>	2	3	4
2				
3				
4				

Q95.

You maintain an application that is used by local food delivery companies. When a customer requests a delivery, the application sends a message to all of the delivery companies. One company accepts the request and fulfills the order.

The application currently supports orders of 100 products or fewer. Some of the delivery companies can now deliver large orders that contain up to 500 products.

You must modify the application so that it supports both small orders and large orders. Messages about large orders should be sent to only delivery companies that can fulfill them. Messages about small orders should be sent to all delivery companies.

Which service should you use?

- A Azure Service Bus Queue
- B Azure Service Bus Relay
- C Azure Service Bus Topics
- D Azure Service Bus Namespace

A95.

You maintain an application that is used by local food delivery companies. When a customer requests a delivery, the application sends a message to all of the delivery companies. One company accepts the request and fulfills the order.

The application currently supports orders of 100 products or fewer. Some of the delivery companies can now deliver large orders that contain up to 500 products.

You must modify the application so that it supports both small orders and large orders. Messages about large orders should be sent to only delivery companies that can fulfill them. Messages about small orders should be sent to all delivery companies.

Which service should you use?

- A Azure Service Bus Queue
- B Azure Service Bus Relay
- C Azure Service Bus Topics
- D Azure Service Bus Namespace

Answer: C

Q96.

A company would like to move an application to the cloud which resides on a single physical server in their datacenter. The server has two drives, one of which hosts operating system, and the other hosts the application data. The operating system has been showing errors recently and the application data was corrupted last Friday at 4:00PM. Data is backed up every day at 1:00PM. Which of the following would be the BEST option for migrating this application to the cloud?

- A Setup a server in the cloud, install an operating system, install the application and copy the data to the cloud server from last Friday's backup.
- B Setup a server in the cloud, install an operating system, install and configure the application and copy the data to the cloud server from last Thursday's backup.
- C Clone or P2V the server with both drivers to the cloud platform.
- D Clone or P2V the server with the application to the cloud platform and copy the operating system to the cloud server.

A96.

A company would like to move an application to the cloud which resides on a single physical server in their datacenter. The server has two drives, one of which hosts operating system, and the other hosts the application data. The operating system has been showing errors recently and the application data was corrupted last Friday at 4:00PM. Data is backed up every day at 1:00PM. Which of the following would be the BEST option for migrating this application to the cloud?

- A Setup a server in the cloud, install an operating system, install the application and copy the data to the cloud server from last Friday's backup.
- B Setup a server in the cloud, install an operating system, install and configure the application and copy the data to the cloud server from last Thursday's backup.
- C Clone or P2V the server with both drivers to the cloud platform.
- D Clone or P2V the server with the application to the cloud platform and copy the operating system to the cloud server.

Answer: A

Explanation:

Use the latest backup of the application data.

Q97.

You are migrating an existing solution to Azure. The solution includes a user interface tier and a database tier. The user interface tier runs on multiple virtual machines (VMs). The user interface tier has a website that uses Node.js. The user interface tier has a background process that uses Python. This background process runs as a scheduled job. The user interface tier is updated frequently. The database tier uses a self-hosted MySQL database.

The user interface tier requires up to 25 CPU cores. You must be able to revert the user interface tier to a previous version if updates to the website cause technical problems. The database requires up to 50 GB of memory. The database must run in a single VM.

You need to deploy the solution to Azure.

What should you do first?

- A Deploy the entire solution to an Azure website. Use a web job that runs continuously to host the database.
- B Deploy the database to a VM that runs Windows Server on the Standard tier.
- C Deploy the entire solution to an Azure website. Run the database by using the Azure data management services.
- D Deploy the user interface tier to a VM. Use multiple availability sets to continuously deploy updates from Microsoft Visual Studio Online.

A97.

You are migrating an existing solution to Azure. The solution includes a user interface tier and a database tier. The user interface tier runs on multiple virtual machines (VMs). The user interface tier has a website that uses Node.js. The user interface tier has a background process that uses Python. This background process runs as a scheduled job. The user interface tier is updated frequently. The database tier uses a self-hosted MySQL database.

The user interface tier requires up to 25 CPU cores. You must be able to revert the user interface tier to a previous version if updates to the website cause technical problems. The database requires up to 50 GB of memory. The database must run in a single VM.

You need to deploy the solution to Azure.

What should you do first?

- A Deploy the entire solution to an Azure website. Use a web job that runs continuously to host the database.
- B Deploy the database to a VM that runs Windows Server on the Standard tier.
- C Deploy the entire solution to an Azure website. Run the database by using the Azure data management services.
- D Deploy the user interface tier to a VM. Use multiple availability sets to continuously deploy updates from Microsoft Visual Studio Online.

Answer: C

Q98.

You are developer working on a project that will be deployed to Azure. The project includes a local SQL Server database.

You need to migrate the database to Azure SQL.

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

Code segment

sqlpackage.exe

dtexec.exe

Start-AzureSqlDatabaseImport

Start-AzureSqlDatabaseRestore

db.bacpac

db.mdf

Answer Area

/a:Export
/ssn:devsql
/sdn:proddb
/tf:C:\Temp\

azure storage blob upload C:\Temp\db.bacpac db db.bacpac
\$ctx = New-AzureSqlDatabaseServerContext -ServerName \$ServerName -Credential \$credential

-SqlConnectionContext \$ctx
-StorageContainer db
-DatabaseName proddb
-BlobName

A98.

Code segment

sqlpackage.exe

dtexec.exe

Start-AzureSqlDatabaseImport

Start-AzureSqlDatabaseRestore

db.bacpac

db.mdf

Answer Area

sqlpackage.exe

/a:Export
/ssn:devsql
/sdn:proddb
/tf:C:\Temp\

db.bacpac

azure storage blob upload C:\Temp\db.bacpac db db.bacpac
\$ctx = New-AzureSqlDatabaseServerContext -ServerName \$ServerName -Credential \$credential

Start-AzureSqlDatabaseImport

-SqlConnectionContext \$ctx
-StorageContainer db
-DatabaseName proddb
-BlobName

db.bacpac

Q99.

You have a WebJob object that runs as part of an Azure website. The WebJob object uses features from the Azure SDK for .NET. You use a well-formed but invalid storage key to create the storage account that you pass into the UploadDataToAzureStorage method.

The WebJob object contains the following code segment. Line numbers are included for reference only.

```
01 void UploadDataToAzureStorage(CloudStorageAccount storageAccount,
    string storageContainerName, string blobpath, string localpath)
02 {
03     var blobClient = storageAccount.CreateCloudBlobClient();
04     var container = blobClient.GetContainerReference(storageContainerName);
05     CloudBlockBlob blockBlob = container.GetBlockBlobReference(blobpath);
06     blockBlob.UploadFromFile(localpath, FileMode.Open);
07 }
```

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

Answer Area

Yes	No
-----	----

If the storage container does not already exist when the code runs, a file can still be uploaded successfully.

If a transient fault occurs when the code segment on line 06 runs, the Azure SDK will attempt to upload the file again.

The code segment at line 06 will fail when the code runs.

A99.

Yes	No
-----	----

If the storage container does not already exist when the code runs, a file can still be uploaded successfully.

If a transient fault occurs when the code segment on line 06 runs, the Azure SDK will attempt to upload the file again.

The code segment at line 06 will fail when the code runs.

Q100.

CASE STUDY

Background:

You are developer for Fabrikam, a company that specializes in payment processing. Fabrikam is developing a solution to process payments for various events, such as music concerts. You develop an ASP.NET MVC website that is hosted in Azure to support an upcoming music concert. The music concert is expected to generate a large volume of ticket sales in a short amount of time.

The website uploads information to an Azure storage queue. A worker role in Azure retrieves information from the queue and generates the concert tickets in a PDF file format after the financial transaction is approved.

You observe a delay between the time the website adds a message to a queue and the time it becomes available to read from the queue. After examining the queue, you determine that no queue messages have a DequeueCount value greater than zero. The website does not throw any errors.

Business Requirements

Payments:

The music concert website must be able to submit event payment information for processing. The website must remain responsive while submitting payment information. Customers must be able to add notes about their orders to a free-form control on the website. These notes must be submitted with the payment when the customer submits an order.

Customers often enter notes that exceed 7 KB in size.

Technical Requirements

Payment Submission and Processing:

Event payment information must be sent from the website to a Windows Communication Foundation (WCF) service worker role. The worker role must submit the information to the payment processor in JSON format.

Payment Processing

You have the following payment processing requirements:

- If the number of messages in a queue goes above or below a specified threshold, worker role instances must be created or deleted as needed. This process must be completed by using the least amount of effort. It must be easy to reconfigure role instance thresholds.
- Payments must be retrieved from the queue in the maximum batch sizes that are allowed by the queue and pulled from the queue for 5 minutes.
- The payment queue must not be re-created when processing payments.
- During single Payment processing, the number of tickets available for an event must be updated. The update operation must be retried for 30 seconds or 5 retry attempts, whichever occurs first. Each retry should pause for at least two seconds and for one second longer than the previous attempt. If the update fails, the payment should be placed in the poison queue.

Storage:

You have the following storage requirements:

- Payment information must be stored by using Azure Queue storage. Connection to the Azure storage account has been established in a configured setting named StorageConnectionString, which is configured for the web and worker roles.
- A payment processing queue and a poison payment queue must be used when processing payments.
- Azure Queue message content must be XML-safe and UTF-8 encoded.
- An Azure storage account must be established for diagnostic information in a configured setting named DiagnosticsConnectionString, which is configured for both the web and worker roles.

Security and Monitoring

Security

The web role must be secured by using HTTPS.

Monitoring

You must collect diagnostic data for both the web and worker roles by using the Diagnostics module. Diagnostics configuration changes must not require the code of the roles to be rebuilt. The diagnostic data is used for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis and capacity planning, and auditing.

Performance testing must evaluate the roles under normal and stress conditions without incurring charges for running Azure. Memory allocation, function time, and multithreading concurrency issues must be evaluated.

Deployment:

You purchase a custom domain name fabrikamfunding.com to host the website, web role, and worker roles. You must deploy an HTTPS certificate with the web role, and you must update associated configuration files accordingly.

Web role and worker role instance sizes must be specified as Medium. You must deploy one web role instance named FabrikamFundingPaymentGenerator, and worker role instances named FabrikamFundingPaymentProcessor.

Application Structure:

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

CustomRetryPolicy.cs

```
CR01 public class CustomRetryPolicy : IRetryPolicy
CR02 {
CR03     int_retryCount = 0;
CR04     readonly TimeSpan _baseInterval= TimeSpan.FromSeconds(1);
CR05     readonly string _poisonPaymentQueueName;
CR06     private readonly CloudQueueClient _queueClient;
CR07     private readonly EventPayment _eventPayment;
CR08     public CustomRetryPolicy(string poisonPaymentQueueName, CloudQueueClient
queueClient, EventPayment eventPayment)
CR09     {
CR10         _poisonPaymentQueueName = poisonPaymentQueueName;
CR11         _queueClient = queueClient;
CR12         _eventPayment = eventPayment;
CR13     }
CR14     public IRetryPolicy CreateInstance()
CR15     {
CR16         return new CustomRetryPolicy(_poisonPaymentQueueName, _queueClient,
_eventPayment);
CR17     }
CR18 }
```

Event.cs

```
EV01 public class Event : TableEntity
EV02 {
EV03     public int AvailableTickets { get; set; }
EV04 }
```

EventPayment.cs

```
EP01 [DataContract]
EP02 public class EventPayment
EP03 {
EP04     [DataMember]
EP05     public int EventId { get; set; }
EP06     [DataMember]
EP07     public string Email { get; set; }
EP08     [DataMember]
EP09     public string Notes { get; set; }
EP10     [DataMember]
EP11     public int TicketCount { get; set; }
EP12     [DataMember]
EP13     public DateTime OrderDate { get; set; }
EP14     [DataMember]
EP15     public Guid EventPaymentId { get; set; }
EP16 }
```

QueueManager.cs

```
QM01 public class QueueManager
QM02 {
QM03     private readonly CloudQueueClient _queueClient;
QM04     private readonly CloudTableClient _tableClient;
QM05     private const string PaymentQueueName = "paymentqueue";
QM06     private const string PoisonPaymentQueueName = "poisonpaymentqueue";
QM07     public QueueManager()
QM08     {
QM09         var storageAccount = CloudStorageAccount.Parse(
QM10             CloudConfigurationManager.GetSetting("StorageConnectionString"));
QM11         _queueClient = storageAccount.CreateCloudQueueClient();
QM12         _tableClient = storageAccount.CreateCloudTableClient();
QM13     }
QM14     public async Task SendMessageAsync(EventPayment eventPayment)
QM15     {
QM16         ...
QM17     }
QM18     public async Task ProcessMessagesAsync()
QM19     {
QM20         ...
QM21     }
QM22     private async Task ProcessPayment(EventPayment eventPayment)
QM23     {
QM23         var events = _tableClient.GetTableReference("events");
QM23         var key = eventPayment.EventId.ToString();
QM23         var operation = await
QM23             events.ExecuteAsync(TableOperation.Retrieve<Event>(key, key));
QM23         var @event = operation.Result as Event;
QM23         @event.AvailableTickets = @event.AvailableTickets - eventPay-
ment.TicketCount;
QM23         var requestOptions = new TableRequestOptions
QM23         {
QM23             RetryPolicy = new CustomRetryPolicy(
QM23                 PoisonPaymentQueueName,
QM23                 _queueClient,
QM23                 eventPayment),
QM23         };
QM23         var context = new OperationContext
QM23         {
QM23             StartTime = DateTime.Now,
QM23         };
QM23         await events.ExecuteAsync(TableOperation.Replace(@event),
requestOptions, context;
QM40     }
QM41 }
```

The SendMessageAsync method of the QueueManager class occasionally throws errors.

You need to correct the errors.

What should you do?

- A Update the QueueManager to use the Put Message operation of the Queue Service REST API. Use HTTP compression for all calls made to the REST API.
 - B Encode the notesfield content by using UTF-32 encoding.
 - C UpdateSendMessageAsyncmethod of the QueueManagerclass to store the notesfield in BLOB storage. Update the EventPaymentclass to store the BLOB uniform resource identifier (URI). Extract the notes BLOB information by using the BLOB URI in the ProcessMessagesAsyncmethod of the QueueManagerclass.
 - D Update the notesfield to a byte array. Binary encode and decode the notescontent when sending or receiving an EventPaymentclass.
-

A100.

The SendMessageAsync method of the QueueManager class occasionally throws errors.

You need to correct the errors.

What should you do?

- A Update the QueueManager to use the Put Message operation of the Queue Service REST API. Use HTTP compression for all calls made to the REST API.
- B Encode the notesfield content by using UTF-32 encoding.
- C UpdateSendMessageAsyncmethod of the QueueManagerclass to store the notesfield in BLOB storage. Update the EventPaymentclass to store the BLOB uniform resource identifier (URI). Extract the notes BLOB information by using the BLOB URI in the ProcessMessagesAsyncmethod of the QueueManagerclass.
- D Update the notesfield to a byte array. Binary encode and decode the notescontent when sending or receiving an EventPaymentclass.

Answer: C

Q101.

You have an existing virtual network with a gateway that is deployed to Azure. You delete all objects that are deployed to the virtual network.

You use the Azure portal to delete the virtual network, but the deletion fails.

You need to determine the cause of the error.

What should you do first?

- A Delete any local network settings and DNS servers.
- B Save all settings.
- C Delete all point-to-site virtual network connections.
- D Delete the virtual network gateway.

A101.

You have an existing virtual network with a gateway that is deployed to Azure. You delete all objects that are deployed to the virtual network

You use the Azure portal to delete the virtual network, but the deletion fails.

You need to determine the cause of the error.

What should you do first?

- A Delete any local network settings and DNS servers.
- B Save all settings.
- C Delete all point-to-site virtual network connections.
- D Delete the virtual network gateway.

Answer: D

Q102.

You are developing a messaging solution for a financial services company named Adatum. The solution must integrate an application named Enrollment and an application named Activation.

The Enrollment application is used to enroll new customers. The Activation application is used to activate accounts for new customers.

You need to ensure that each message that the Enrollment application sends is stored in a queue for ten minutes before the Activation application uses the message.

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

Answer Area

```
var address =  
    ServiceBusEnvironment.CreateServiceUri("sb  
adatum.activation  
adatum.servicebus.windows.net/activate",  
    string.Empty);  
  
var ns = new NamespaceManager(address, new NamespaceManagerSettings()  
{  
    OperationTimeout =  
        TimeSpan.FromSeconds(10)  
});  
ns.CreateQueue("ActivationQueue");
```

A102.

```
var address =  
    ServiceBusEnvironment.CreateServiceUri("sb  
adatum.activation  
adatum.servicebus.windows.net/activate",  
    string.Empty);  
  
var ns = new NamespaceManager(address, new NamespaceManagerSettings()  
{  
    OperationTimeout =  
        TimeSpan.FromSeconds(10)  
});  
ns.CreateQueue("ActivationQueue");
```

Q103.

You plan to migrate an Azure Web App named Contoso from an App Service plan named AppServicePlan1 to another App Service plan. You create a resource group named ContosoGroup.

You create the following Azure PowerShell script. Line numbers are included for reference only.

```
01 $AppServicePlan = @{"serverfarm" = "AppServicePlan2"}  
02 Set-AzureResource –name Contoso –ResourceGroupName ContosoGroup –ResourceType  
Microsoft.Web/sites ~  
    -apiversion 2014-04-01 –PropertyObject $AppServicePlan  
03 Get-AzureResource –name Contoso –ResourceGroupName ContosoGroup –ResourceType  
Microsoft.Web/sites ~  
    -apiversion 2014-04-01
```

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

NOTE: Each correct selection is worth one point.

Answer Area

	Yes	No
The command in line 01 defines a variable that stores a hash table.	<input type="radio"/>	<input type="radio"/>
The command in line 02 assigns the Web App to the ContosoGroup resource group.	<input type="radio"/>	<input type="radio"/>
The command in line 02 assigns the Web App to a hosting plan named webhostingplan2 .	<input type="radio"/>	<input type="radio"/>

A103.

	Yes	No
The command in line 01 defines a variable that stores a hash table.	<input checked="" type="checkbox"/>	<input type="radio"/>
The command in line 02 assigns the Web App to the ContosoGroup resource group.	<input type="radio"/>	<input checked="" type="checkbox"/>
The command in line 02 assigns the Web App to a hosting plan named webhostingplan2 .	<input type="radio"/>	<input checked="" type="checkbox"/>

Q104.

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution. After you answer a question in this sections, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You plan to deploy an application as a cloud service. The application uses a virtual network to extend your on-premises network into Azure.

You need to configure a site-to-site VPN for cross-premises network connections.

Which two objects should you configure? Each correct answer presents part of the solution.

- A VNet peering
- B External-facing IPv6 address
- C VPN gateway
- D External-facing IPv4 address

A104.

You plan to deploy an application as a cloud service. The application uses a virtual network to extend your on-premises network into Azure.

You need to configure a site-to-site VPN for cross-premises network connections.

Which two objects should you configure? Each correct answer presents part of the solution.

- A VNet peering
- B External-facing IPv6 address
- C VPN gateway
- D External-facing IPv4 address

Answer: C, D

Explanation:

A Site-to-Site VPN gateway connection is used to connect your on-premises network to an Azure virtual network over an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel.

Verify that you have an externally facing public IPv4 address for your VPN device. This IP address cannot be located behind a NAT.

Q105.

Which of the following are the MOST important benefits of a cloud computing solution for an application development provider? (Select two.)

- A Reduced training time for new developers
- B Reduced storage requirements.
- C Reduced complexity for users.
- D Reduced bandwidth usage.
- E Reduced cost.
- F Reduced development timeframe.

A105.

Which of the following are the MOST important benefits of a cloud computing solution for an application development provider? (Select two.)

- A Reduced training time for new developers
- B Reduced storage requirements.
- C Reduced complexity for users.
- D Reduced bandwidth usage.
- E Reduced cost.
- F Reduced development timeframe.

Answer: E, F

Explanation:

The biggest promise of Azure-based applications is the ability to write them to scale as needed in real-time. Customers will therefore only use the amount of resources they need, rather than budgeting a set amount of resources that can overtax or underutilize their current setup.

Q106.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18 }
```

```

PC19     [Route("nearby")]
PC20     [HttpGet]
PC21     public IEnumerable<pointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22     {
PC23         var location = new Point(longitude, latitude);
PC24         using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25         {
PC26
PC27         }
PC28     }
PC29
PC30     public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31     {
PC32         using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33         {
PC34             var collection = await GetCollection();
PC35             await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36             return pg;
PC37         }
PC38     }
PC39     private static async Task<DocumentCollection> GetCollection()
PC40     {
PC41         ...
PC42     }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

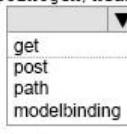
```

PG01 public class PointGenerator
PG02     {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10     public DateTimeOffset DateLastClaimed { get; set; }
PG11     [JsonProperty("location")]
PG12     public Point Location { get; set; }
PG13 }
```

You need to build a Swagger specification for creating the GamerData managed API.

What should you do? To answer, select the appropriate options in the answer area.

```

"paths": {
    "/api/pointgen/{name}": {
        "get": {
            "parameters": [
                {
                    "name": "name", "in": "path", "required": true,
                    
                    "type": "string"
                }
            ],
            "responses": {
                "200": {
                    "description": "Success response"
                }
            }
        }
    },
    "/api/pointgen/nearby": {
        "get": {
            "parameters": [
                {
                    "name": "longitude", "in": "query", "required": true,
                    
                    "type": "number"
                },
                {
                    "name": "latitude", "in": "query", "required": true,
                    "type": "number"
                },
                {
                    "name": "minDistance", "in": "query", "required": true,
                    "type": "number"
                }
            ],
            "responses": {
                "200": {
                    "description": "Success response"
                }
            }
        }
    }
}
```

```
"parameters": [
    {
        "name": "longitude", "in": "query", "required": true, "type": "number", "format": "double"
    },
    {
        "name": "latitude", "in": "query", "required": true, "type": "number", "format": "double"
    },
    {
        "name": "minDistance", "in": "query", "required": true, "type": "number", "format": "int64"
    }
]}]
```

A106.

```
"paths": {
    "/api/pointgen/{name)": {
        "get": {
            "parameters": [
                {
                    "name": "name", "in": "path", "required": true,
                    "type": "string"
                }
            ],
        },
        "/api/pointgen/nearby": {
            "get": {
                "parameters": [
                    {
                        "name": "longitude", "in": "path", "required": true,
                        "type": "number", "format": "double"
                    },
                    {
                        "name": "latitude", "in": "path", "required": true,
                        "type": "number", "format": "double"
                    },
                    {
                        "name": "minDistance", "in": "path", "required": true,
                        "type": "number", "format": "int64"
                    }
                ]
            }
        }
    }
}
```

Q107.

You have an ASP.NET application that runs in a cloud service. A new version of the application is ready for release. The new version contains code changes and new SSL certificates. The application consists of six instances of a web role and four instances of a worker role.

The application performs at or near full capacity. The cloud service uses the default number of fault domains and upgrade domains.

You plan to deploy the new version of the application. The performance and capacity of the web roles must not degrade during the deployment. Temporary degradation of the worker roles is acceptable. The deployment must take a maximum of six hours.

You need to deploy the new version of the ASP.NET application to the cloud service.

Which two approaches will achieve the goal? Each correct answer presents a complete solution.

- A Increase the number of web role instances to eight, and then deploy the new version of the application by using an in-place update. Reduce the number of web role instances to six after the upgrade is completed.
- B Deploy the new version of the application by using an in-place update. Use upgrade domains to ensure that there is sufficient capacity during the upgrade.
- C Deploy the new version of the application into the staging slot for the cloud service. Then activate the new version of the application by swapping virtual IP (VIP) addresses.
- D Delete the old version of the application, and deploy the new version of the application.

A107.

You have an ASP.NET application that runs in a cloud service. A new version of the application is ready for release. The new version contains code changes and new SSL certificates. The application consists of six instances of a web role and four instances of a worker role.

The application performs at or near full capacity. The cloud service uses the default number of fault domains and upgrade domains.

You plan to deploy the new version of the application. The performance and capacity of the web roles must not degrade during the deployment. Temporary degradation of the worker roles is acceptable. The deployment must take a maximum of six hours.

You need to deploy the new version of the ASP.NET application to the cloud service.

Which two approaches will achieve the goal? Each correct answer presents a complete solution.

- A Increase the number of web role instances to eight, and then deploy the new version of the application by using an in-place update. Reduce the number of web role instances to six after the upgrade is completed.
- B Deploy the new version of the application by using an in-place update. Use upgrade domains to ensure that there is sufficient capacity during the upgrade.
- C Deploy the new version of the application into the staging slot for the cloud service. Then activate the new version of the application by swapping virtual IP (VIP) addresses.
- D Delete the old version of the application, and deploy the new version of the application.

Answer: B, C

Q108.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for Fabrikam, a company that specializes in payment processing. Fabrikam is developing a solution to process payments for various events, such as music concerts. You develop an ASP.NET MVC website that is hosted in Azure to support an upcoming music concert. The music concert is expected to generate a large volume of ticket sales in a short amount of time.

The website uploads information to an Azure storage queue. A worker role in Azure retrieves information from the queue and generates the concert tickets in a PDF file format after the financial transaction is approved.

You observe a delay between the time the website adds a message to a queue and the time it becomes available to read from the queue. After examining the queue, you determine that no queue messages have a DequeueCount value greater than zero. The website does not throw any errors.

Business Requirements

Payments

The music concert website must be able to submit event payment information for processing. The website must remain responsive while submitting payment information. Customers must be able to add notes about their orders to a free-form control on the website. These notes must be submitted with the payment when the customer submits an order.

Customers often enter notes that exceed 7 KB in size.

Technical Requirements

Payment Submission and Processing

Event payment information must be sent from the website to a Windows Communication Foundation (WCF) service worker role. The worker role must submit the information to the payment processor in JSON format.

Payment Processing

You have the following payment processing requirements:

- If the number of messages in a queue goes above or below a specified threshold, worker role instances must be created or deleted as needed. This process must be completed by using the least amount of effort. It must be easy to reconfigure role instance thresholds.
- Payments must be retrieved from the queue in the maximum batch sizes that are allowed by the queue and pulled from the queue for 5 minutes.
- The payment queue must not be re-created when processing payments.
- During single Payment processing, the number of tickets available for an event must be updated. The update operation must be retried for 30 seconds or 5 retry attempts, whichever occurs first. Each retry should pause for at least two seconds and for one second longer than the previous attempt. If the update fails, the payment should be placed in the poison queue.

Storage

You have the following storage requirements:

- Payment information must be stored by using Azure Queue storage. Connection to the Azure storage account has been established in a configured setting named StorageConnectionString, which is configured for the web and worker roles.
- A payment processing queue and a poison payment queue must be used when processing payments.
- Azure Queue message content must be XML-safe and UTF-8 encoded.
- An Azure storage account must be established for diagnostic information in a configured setting named DiagnosticsStorageConnectionString, which is configured for both the web and worker roles.

Security and Monitoring

Security

The web role must be secured by using HTTPS.

Monitoring

You must collect diagnostic data for both the web and worker roles by using the Diagnostics module. Diagnostics configuration changes must not require the code of the roles to be rebuilt. The diagnostic data is used for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis and capacity planning, and auditing.

Performance testing must evaluate the roles under normal and stress conditions without incurring changes for running Azure. Memory allocation, function time, and multithreading concurrency issues must be evaluated.

Deployment

You purchase a custom domain name fabrikamfunding.com to host the website, web role, and worker roles. You must deploy an HTTPS certificate with the web role, and you must update associated configuration files accordingly.

Web role and worker role instance sizes must be specified as Medium. You must deploy one web role instance named FabrikamFundingPaymentGenerator, and worker role instances named FabrikamFundingPaymentProcessor.

Application Structure

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

CustomRetryPolicy.cs

```
CR01 public class CustomRetryPolicy : IRetryPolicy
CR02 {
CR03     int _retryCount = 0;
CR04     readonly TimeSpan _baseInterval= TimeSpan.FromSeconds(1);
CR05     readonly string _poisonPaymentQueueName;
CR06     private readonly CloudQueueClient _queueClient;
CR07     private readonly EventPayment _eventPayment;
CR08     public CustomRetryPolicy(string poisonPaymentQueueName, CloudQueueClient
queueClient, EventPayment eventPayment)
CR09     {
CR10         _poisonPaymentQueueName = poisonPaymentQueueName;
CR11         _queueClient = queueClient;
CR12         _eventPayment = eventPayment;
CR13     }
CR14     public IRetryPolicy CreateInstance()
CR15     {
CR16         return new CustomRetryPolicy(_poisonPaymentQueueName, _queueClient,
_eventPayment);
CR17     }
CR18 }
```

Event.cs

```
EV01 public class Event : TableEntity
EV02 {
EV03     public int AvailableTickets { get; set; }
EV04 }
```

EventPayment.cs

```
EP01 [DataContract]
EP02 public class EventPayment
EP03 {
EP04     [DataMember]
EP05     public int EventId { get; set; }
EP06     [DataMember]
EP07     public string Email { get; set; }
EP08     [DataMember]
EP09     public string Notes { get; set; }
EP10     [DataMember]
EP11     public int TicketCount { get; set; }
EP12     [DataMember]
EP13     public DateTime OrderDate { get; set; }
EP14     [DataMember]
EP15     public Guid EventPaymentId { get; set; }
EP16 }
```

QueueManager.cs

```
QM01 public class QueueManager
QM02 {
QM03     private readonly CloudQueueClient _queueClient;
QM04     private readonly CloudTableClient _tableClient;
QM05     private const string PaymentQueueName = "paymentqueue";
QM06     private const string PoisonPaymentQueueName = "poisonpaymentqueue";
QM07     public QueueManager()
QM08     {
QM09         var storageAccount = CloudStorageAccount.Parse(
QM10             CloudConfigurationManager.GetSetting("StorageConnectionString"));
QM11         _queueClient = storageAccount.CreateCloudQueueClient();
QM12         _tableClient = storageAccount.CreateCloudTableClient();
QM13     }
QM14     public async Task SendMessageAsync(EventPayment eventPayment)
QM15     {
QM16         ...
QM17     }
QM18     public async Task ProcessMessagesAsync()
QM19     {
QM20         ...
QM21     }
QM22     public async Task ProcessPayment(EventPayment eventPayment)
QM23     {
QM24         var events = _tableClient.GetTableReference("events");
QM25         var key = eventPayment.EventId.ToString();
QM26         var operation = await events.ExecuteAsync(TableOperation.Retrieve<Event>(key, key));
QM27         var @event = operation.Result as Event;
QM28         @event.AvailableTickets = @event.AvailableTickets - eventPayment.TicketCount;
QM29         var requestOptions = new TableRequestOptions
QM30         {
QM31             RetryPolicy = new CustomRetryPolicy(
QM32                 PoisonPaymentQueueName,
QM33                 _queueClient,
QM34                 eventPayment),
QM35         };
QM36         var context = new OperationContext
QM37         {
QM38             StartTime = DateTime.Now,
QM39         };
QM40         await events.ExecuteAsync(TableOperation.Replace(@event),
requestOptions, context,
QM41     }
QM42 }
```

You need to implement the `SendMessagesAsync` method in the `QueueManager` class.

How should you complete the relevant code? To answer, select the appropriate code segment from each list in the answer area.

```
public async Task SendMessageAsync(EventPayment eventPayment)
{
    var queue = _queueClient.GetQueueReference(PaymentQueueName);
    await queue.CreateIfNotExistsAsync();

    await queue.ClearAsync();
    queue.EncodeMessage = true;
    var eventPaymentMessage = new XmlSerializer(typeof(EventPayment)).Serialize(eventPayment);
    var eventPaymentMessage = JsonConvert.SerializeObject(eventPayment);

    await queue.PeekMessageAsync();
    queue.Metadata.Add("message", eventPaymentMessage);
    var message = new CloudQueueMessage(eventPaymentMessage);
    var message = CloudConfigurationManager.GetSetting("eventPaymentMessage");

    queue.AddMessage(message);
    queue.DeleteMessage(message);
    await queue.AddMessageAsync(message);
    await queue.DeleteMessageAsync(message);
}
```

A108.

```
public async Task SendMessageAsync(EventPayment eventPayment)
{
    var queue = _queueClient.GetQueueReference(PaymentQueueName);
    await queue.CreateIfNotExistsAsync();

    await queue.ClearAsync();
    queue.EncodeMessage = true;
    var eventPaymentMessage = new XmlSerializer(typeof(EventPayment)).Serialize(eventPayment);
    var eventPaymentMessage = JsonConvert.SerializeObject(eventPayment);

    await queue.PeekMessageAsync();
    queue.Metadata.Add("message", eventPaymentMessage);
    var message = new CloudQueueMessage(eventPaymentMessage);
    var message = CloudConfigurationManager.GetSetting("eventPaymentMessage");

    queue.AddMessage(message);
    queue.DeleteMessage(message);
    await queue.AddMessageAsync(message);
    await queue.DeleteMessageAsync(message);
}
```

Q109.

You store data by using table storage in Azure.

The storage analytics logs do not contain any data.

You must configure the Azure storage account to retain logs for the maximum length of time that Azure permits.

In the Azure management portal, what should you do?

- A Set the monitoring level to Minimal, and set the number of days the data in the logs is retained to 0.
- B Set the monitoring level to Verbose, and set the number of days the data in the logs is retained to 365.
- C Set the monitoring level to Minimal, and set the number of days the data in the logs is retained to 99.
- D Set the monitoring level to Verbose, and set the number of days the data in the logs is retained to 30.

A109.

You store data by using table storage in Azure.

The storage analytics logs do not contain any data.

You must configure the Azure storage account to retain logs for the maximum length of time that Azure permits.

In the Azure management portal, what should you do?

- A Set the monitoring level to Minimal, and set the number of days the data in the logs is retained to 0.
- B Set the monitoring level to Verbose, and set the number of days the data in the logs is retained to 365.
- C Set the monitoring level to Minimal, and set the number of days the data in the logs is retained to 99.
- D Set the monitoring level to Verbose, and set the number of days the data in the logs is retained to 30.

Answer: A

Q110.

You are maintaining an application that uses the Azure Content Delivery Network (CDN) to serve terabytes of content that is stored in page blobs.

Your bill for CDN services is higher than you expect.

You need to monitor the application to find issues that increase costs.

Which two operations should you monitor? Each correct answer presents part of the solution.

- A The Time-To-Live (TTL) of the blobs.
- B The country of origin for the client computer and the CDN region.
- C The number of requests that result in an HTTP status code over 400.
- D The allocated size of page blobs.
- E The expiration date of the blobs.

A110.

You are maintaining an application that uses the Azure Content Delivery Network (CDN) to serve terabytes of content that is stored in page blobs.

Your bill for CDN services is higher than you expect.

You need to monitor the application to find issues that increase costs.

Which two operations should you monitor? Each correct answer presents part of the solution.

- A The Time-To-Live (TTL) of the blobs.
- B The country of origin for the client computer and the CDN region.
- C The number of requests that result in an HTTP status code over 400.
- D The allocated size of page blobs.
- E The expiration date of the blobs.

Answer: B, D

Q111.

You are developing an Azure cloud service for a company. The cloud service monitors a queue for incoming messages and then processes invoices based on the contents of these messages.

Some messages are formed incorrectly and cause exceptions. There is no time limit for how long the service takes to process an individual message.

All messages must be processed at least once by using the ProcessMessage method. Messages must not be processed more than twice by using the ProcessMessage method. Messages that fail normal processing must be processed by using the ProcessPoisonMessage method.

You need to configure message processing.

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

Answer Area

```
private bool ProcessNextQueueMessage(CloudQueue cloudQueue)
{
    var msg = cloudQueue.GetMessage();

    if (msg == null) return false;
    if (msg.DequeueCount > 0) return false;
    if (msg.PopReceipt == null) return false;
    if (msg.ExpirationTime.HasValue) return false;

    ProcessPoisonMessage(msg);
    else
        ProcessMessage(msg);

    cloudQueue.Delete();
    cloudQueue.DeleteMessage(msg);
    cloudQueue.EndAddMessage(null);
    cloudQueue.DeleteMessage(null);

    return true;
}
```

A111.

```
private bool ProcessNextQueueMessage(CloudQueue cloudQueue)
{
    var msg = cloudQueue.GetMessage();

    if (msg == null) return false;
    if (msg.DequeueCount > 0) return false;
    if (msg.PopReceipt == null) return false;
    if (msg.ExpirationTime.HasValue) return false;

    if (msg == null)
        if (msg.DequeueCount > 0)
            if (msg.DequeueCount > 2)
                if (msg.PopReceipt == null)

    ProcessPoisonMessage(msg);
    else
        ProcessMessage(msg);

    cloudQueue.Delete();
    cloudQueue.DeleteMessage(msg);
    cloudQueue.EndAddMessage(null);
    cloudQueue.DeleteMessage(null);

    return true;
}
```

Q112.

You have six Ubuntu Linux virtual machines (VMs) that run a Hadoop cluster on Azure. One of the VMs hosts a custom web user interface that allows users to examine the processing jobs within the Hadoop Cluster.

You need to select the appropriate Azure Storage type for each Azure VM scenario.

Which Azure Storage types should you use? To answer, drag the appropriate Azure Storage type to the correct target. Each Azure Storage type may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Azure Storage types	Answer area	Storage Type
Azure Files	Provide a Server Message Block (SMB) interface in addition to a REST interface to access files from the VM.	Storage Type
Azure Blobs	Use REST APIs to store unstructured data for random access and streaming for the VM.	Storage Type
Azure Disks	Provide persistent storage attached to the VM.	Storage Type
	Mount the share from Ubuntu Linux and access the share by using file system APIs.	Storage Type
	Snapshot the VM storage to create point in time read-only backups.	Storage Type

A112.

Azure Storage types	Answer area	Storage Type
Azure Files	Provide a Server Message Block (SMB) interface in addition to a REST interface to access files from the VM.	Azure Files
Azure Blobs	Use REST APIs to store unstructured data for random access and streaming for the VM.	Azure Blobs
Azure Disks	Provide persistent storage attached to the VM.	Azure Disks
	Mount the share from Ubuntu Linux and access the share by using file system APIs.	Azure Files
	Snapshot the VM storage to create point in time read-only backups.	Azure Disks

Q113.

You deploy a website to Azure. When the website starts, it loads and caches common data.

Updates to the website must occur without downtime or performance degradation that is noticeable to users.

You need to upgrade to a new version of website code.

What should you do?

- A. Create a staging slot for the new version of the website. Run the following Windows PowerShell command, and then deploy the new code.

```
Switch-AzureWebsiteSlot -Name "MyWebsiteName"
```

- B. Create a staging slot for the new version of the website. Deploy the new code to that slot. Then run the following Windows PowerShell command:

```
Switch-AzureWebsiteSlot -Name "MyWebsiteName"
```

- C. Run the following Windows PowerShell command:

```
New-AzureWebsite -Name "Staging" -Location "East US"
```

Deploy the new code to the staging site. Then run the following Windows PowerShell command:

```
Switch-AzureWebsiteSlot -Name "MyWebsiteName"
```

- D. Create a staging slot for the new version of the website. Run the following Windows PowerShell command:

```
Switch-AzureWebsiteSlot -Name "MyWebsiteName"
```

Then deploy the new code to the staging slot.

A113.

A Option A

B Option B

C Option C

D Option D

Answer: B

Q114.

[You have a website that is hosted on Azure. You connect to the site by using the URI http://www.contoso.com. You plan to publish a new version of the website.](#)

You need to acquire the publishing profile for the website.

Which two actions will achieve the goal? Each correct answer presents a complete solution.

- A Run the following Windows PowerShell cmdlet: Get-AzurePublishSettingsFile
- B Run the following Windows PowerShell cmdlet: Get-AzureSubscription
- C [Navigate to the following URI: https://www.contoso.com/download/publishprofile.aspx](https://www.contoso.com/download/publishprofile.aspx)
- D [Navigate to the following URI: https://windows.azure.com/download/publishprofile.aspx](https://windows.azure.com/download/publishprofile.aspx)

A114.

[You have a website that is hosted on Azure. You connect to the site by using the URI http://www.contoso.com. You plan to publish a new version of the website.](#)

You need to acquire the publishing profile for the website.

Which two actions will achieve the goal? Each correct answer presents a complete solution.

- A [Run the following Windows PowerShell cmdlet: Get-AzurePublishSettingsFile](#)
- B Run the following Windows PowerShell cmdlet: Get-AzureSubscription
- C [Navigate to the following URI: https://www.contoso.com/download/publishprofile.aspx](https://www.contoso.com/download/publishprofile.aspx)
- D [Navigate to the following URI: https://windows.azure.com/download/publishprofile.aspx](https://windows.azure.com/download/publishprofile.aspx)

Answer: A, D

Q115.

You connect to an existing service over the network by using HTTP. The service listens on HTTP port 80. You plan to create a test environment for this existing service by using an Azure virtual machine (VM) that runs Windows Server.

The service must be accessible from the public Internet over HTTP port 8080.

You need to configure the test environment.

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

- A Configure an endpoint to route traffic from port 8080 to port 80.
- B Configure an endpoint to route traffic from port 80 to port 8080.
- C Ensure that the public IP address is configured as a static IP address.
- D Configure the Windows Server firewall to allow incoming and outgoing traffic on port 8080.
- E Configure the Windows Server firewall to allow incoming and outgoing traffic on port 80.

A115.

You connect to an existing service over the network by using HTTP. The service listens on HTTP port 80. You plan to create a test environment for this existing service by using an Azure virtual machine (VM) that runs Windows Server.

The service must be accessible from the public Internet over HTTP port 8080.

You need to configure the test environment.

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

- A Configure an endpoint to route traffic from port 8080 to port 80.
- B Configure an endpoint to route traffic from port 80 to port 8080.
- C Ensure that the public IP address is configured as a static IP address.
- D Configure the Windows Server firewall to allow incoming and outgoing traffic on port 8080.
- E Configure the Windows Server firewall to allow incoming and outgoing traffic on port 80.

Answer: A, E

Q116.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for LitWare, Inc., a game development company. You are developing a backend service for an online social gaming platform named GamerData. The game is built around point generators, which are associated with physical landmarks. Players claim point generators which give them a set amount of points per day.

Business Requirements

Mobile App

The game itself runs on various mobile devices and is developed by TailSpin Toys, a company that specializes in mobile game development. The mobile app will periodically make calls to the GamerData service to find the five closest point generators that are located less than the specified distance from the player's current location. If no point generators are found, the search distance increases until one is found.

The mobile app shows all the point generators owned by each player. The mobile app allows for each player to search for claimed point generators by player name. This search does not require exact spelling of names. The details for each claimed generator is shown in the app.

When a player claims a point generator, they should receive an email notification. An Azure Function named EmailPlayer has been developed to email players with details about recently claimed point generators.

Sponsors

The platform allows business to sponsor point generators within a business location.

Reports

A report named Daily Sponsor Report must be generated each day at midnight. The report must contain a section for each sponsor. Each sponsor section must contain two subsections.

The first subsection of the report contains the names of the point generators for that sponsor, ordered by the last time the point generator was claimed. The second subsection contains the current owners for each of the point generators for the sponsor. Generation of reports must not impact the GamerData service.

Technical Requirements

GamerData Service

All data for the GamerData service is stored in an Azure DocumentDB instance named GamerData. Business and players interact with the service by using a REST API.

The REST API must:

- Produce valid Swagger API specifications for non-obsolete actions.
- Be optimized for loading specific point generators.
- Follow REST best practices.
- Include appropriate terms of service.

Costs for all Azure services must be minimized.

Build and Deployment

The GamerData service will be deployed to Azure in a private VNet.

Security

Sponsors have accounts in an Azure Active Directory (Azure AD) with business-to-consumer (B2C) enabled named litwaregamerdata.onmicrosoft.com managed by Litware, Inc. for both GamerData and LitWare, Inc. services.

Only Litware, Inc. developers and automated testing tools should be able to directly access the GamerData service. All other use of the service must be through Azure API Management. A description of the security practices used during development, available on Microsoft SharePoint, must be available to users of the API under the terms of service.

Reporting

Azure Search will be used as the source for running reports. The properties of indexes in Azure Search must match the names of the properties in DocumentDB.

Performance

The Azure DocumentDB must not be used for reporting purposes. All services must perform queries in the data store when possible.

Application Structure

Startup.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
SP01 public class Startup
SP02 {
SP03     public IConfigurationRoot Configuration { get; }
SP04     public Startup(IHostingEnvironment env)
SP05     {
SP06         var builder = new ConfigurationBuilder().SetBasePath(env.ContentRoot-
Path).AddJsonFile ("appsettings.json");
SP07         Configuration = builder.Build();
SP08     }
SP09     public void ConfigureServices(IServiceCollection services)
SP10     {
SP11         services.AddMvc();
SP12         Services.AddSwaggerGen();
SP13     }
SP14     public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
SP15     {
SP16         app.UseMvc();
SP17         app.UseSwagger();
SP18     }
SP19 }
```

PointController.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PC01 [Route("api/pointgen")]
PC02     public class PointGeneratorController : Controller
PC03     {
PC04         private static readonly string DatabaseName = "GamerData";
PC05         private static readonly string CollectionName = "PointGenerators";
PC06         private static readonly string EndpointUrl = "...";
PC07         private static readonly string AuthorizationKey = "...";
PC08
PC09         [HttpGet("{name}")]
PC10         public async Task<PointGenerator> Get(string name)
PC11         {
PC12             using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC13             {
PC14                 var response = await client.ReadDocumentAsync(UriFactory.Create-
DocumentUri(DatabaseName, CollectionName, name));
PC15                 return (PointGenerator)(dynamic)response.Resource;
PC16             }
PC17         }
PC18 }
```

```
PC19     [Route("nearby")]
PC20     [HttpGet]
PC21     public IEnumerable<pointGenerator> Nearby(double longitude, double
latitude, long minDistance)
PC22     {
PC23         var location = new Point(longitude, latitude);
PC24         using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC25             {
PC26
PC27             }
PC28         }
PC29
PC30     public async Task<PointGenerator> Update[FromBody] PointGenerator pg)
PC31     {
PC32         using (var client = new DocumentClient(new Uri(EndpointUrl),
AuthorizationKey))
PC33             {
PC34                 var collection = await GetCollection();
PC35                 await client.UpsertDocumentAsync(collection.SelfLink, pg);
PC36                 return pg;
PC37             }
PC38         }
PC39     private static async Task<DocumentCollection> GetCollection()
PC40     {
PC41         ...
PC42     }
PC43 }
```

PointGenerator.cs

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```
PG01 public class PointGenerator
PG02     {
PG02
PG04     public string Name { get; set; }
PG05     [JsonProperty("currentOwner")]
PG06     public string CurrentOwner { get; set; }
PG07     [JsonProperty("sponsor")]
PG08     public string Sponsor { get; set; }
PG09     [JsonProperty("dateLastClaimed")]
PG10    public DateTimeOffset DateLastClaimed { get; set; }
PG11    [JsonProperty("location")]
PG12    public Point Location { get; set; }
PG13 }
```

Actions

- Create a new Azure AD named litware.onmicrosoft.com.
- Enable OAuth 2.0 user authorization in Azure AD.
- Configure an API Management OAuth 2.0 authorization server.
- Configure an Azure AD OAuth 2.0 authorization server.
- Enable OAuth 2.0 user authorization in API Management.
- Register the API Management developer portal as an Azure AD application in litware.onmicrosoft.com.
- Register the API Management developer portal as an Azure AD application in litware2.onmicrosoft.com.
- Register the API Management developer portal as an Azure AD application in litwaregamedata.onmicrosoft.com.

Answer Area



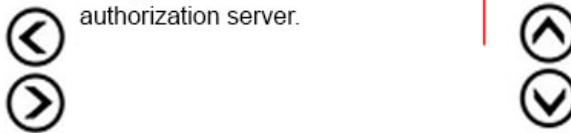
A116.

Actions

- Create a new Azure AD named litware.onmicrosoft.com.
- Enable OAuth 2.0 user authorization in Azure AD.
- Configure an API Management OAuth 2.0 authorization server.
- Configure an Azure AD OAuth 2.0 authorization server.
- Enable OAuth 2.0 user authorization in API Management.
- Register the API Management developer portal as an Azure AD application in litware.onmicrosoft.com.
- Register the API Management developer portal as an Azure AD application in litware2.onmicrosoft.com.
- Register the API Management developer portal as an Azure AD application in litwaregamedata.onmicrosoft.com.

Answer Area

- Enable OAuth 2.0 user authorization in Azure AD.
- Register the API Management developer portal as an Azure AD application in litwaregamedata.onmicrosoft.com.
- Configure an Azure AD OAuth 2.0 authorization server.



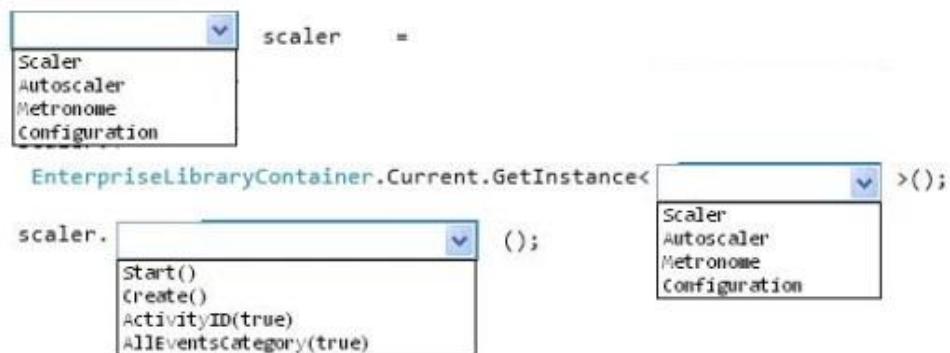
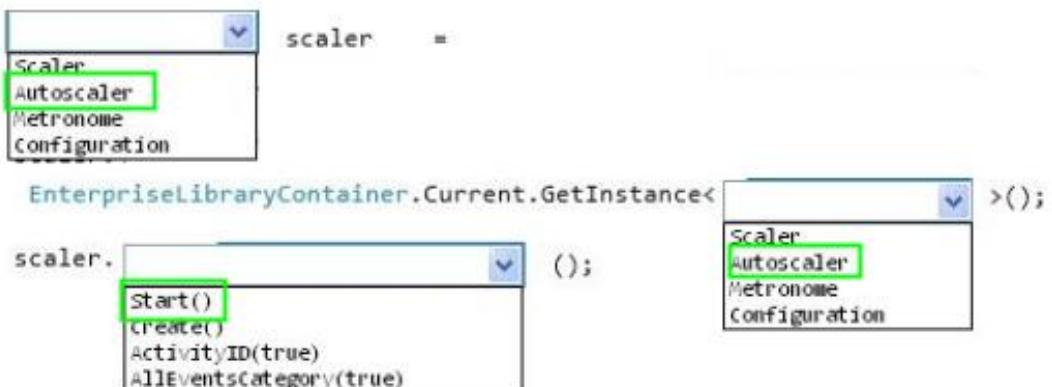
Q117.

A company creates an Azure worker role to manage products.

The number of customers who inquire about how many products are in inventory rapidly increases.

You need to ensure that the worker role can scale to accommodate the increased workload.

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

Answer Area**A117.**

Q118.

You are developing an ASP.NET Web App that makes a large number of calls to Azure Blob storage.
You observe that the app suffers from Azure Blob storage throttling.

You need to resolve throttling failures when loading data from Azure Blob storage.

What should you do? To answer, drag the appropriate code segment to the correct location. Each code segment may be used once, more than once, or not at all.
You may need to drag the split bar between panes or scroll to view content

NOTE: Each correct selection is worth one point.

400

403

500

503

waitMillisecnd * 2;

waitMillisecnd + 2;

Answer Area

```
var sasBlobUri = "....";
var waitMillisecond = 1000;
while (true) {
    using (var client = new System.Net.Http.HttpClient())
    {
        var response = await client.GetAsync(sasBlobUri);
        if (response.IsSuccessStatusCode)
        {
            return await response.Content.ReadAsByteArrayAsync();
        }
        else
        {
            var statusCode = (int)response.StatusCode;
            if (statusCode == [redacted])
                || statusCode == [redacted])
            {
                waitMillisecond = [redacted];
                await Task.Delay(waitMillisecond);
            }
            else
            {
                response.EnsureSuccessStatusCode();
            }
        }
    }
}
```

A118.

400
403
500
503
waitMillisecnd * 2;
waitMillisecnd + 2;

Answer Area

```
var sasBlobUri = "...";
var waitMillisecond = 1000;
while (true) {
    using (var client = new System.Net.Http.HttpClient())
    {
        var response = await client.GetAsync(sasBlobUri);
        if (response.IsSuccessStatusCode)
        {
            return await response.Content.ReadAsByteArrayAsync();
        }
        else
        {
            var statusCode = (int)response.StatusCode;
            if (statusCode == 500
                || statusCode == 503)
            {
                waitMillisecond = waitMillisecnd * 2;

                await Task.Delay(waitMillisecond);
            }
            else
            {
                response.EnsureSuccessStatusCode();
            }
        }
    }
}
```

Q119.

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

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

You deploy a Virtual Machine Scale Set (VMSS) named CorpWebVMSS to Azure by using Azure PowerShell and set the instance count to 1. The VMSS includes a storage account, load balancer, public IP address, and six Standard_A1 Windows virtual machines (VMs) that run Internet Information Services (IIS). All components are deployed to a resource group named CorpWebRG.

You must increase the instance count to support the increased load on IIS.

You need to manually scale out the number of VMs in the scale set to 5.

Solution: You run the following command by using the Azure Command-Line Interface (CLI):

```
azure vmss scale -g CorpWebRG -n CorpWebVMSS -c 5
```

Does the solution meet the goal?

- A Yes
- B No

A119.

You deploy a Virtual Machine Scale Set (VMSS) named CorpWebVMSS to Azure by using Azure PowerShell and set the instance count to 1. The VMSS includes a storage account, load balancer, public IP address, and six Standard_A1 Windows virtual machines (VMs) that run Internet Information Services (IIS). All components are deployed to a resource group named CorpWebRG.

You must increase the instance count to support the increased load on IIS.

You need to manually scale out the number of VMs in the scale set to 5.

Solution: You run the following command by using the Azure Command-Line Interface (CLI):

```
azure vmss scale -g CorpWebRG -n CorpWebVMSS -c 5
```

Does the solution meet the goal?

- A Yes
- B No

Answer: A

Q120.

Your team uses a proprietary source control product. You use FTP to manually deploy an Azure website.

You must move your source code from the proprietary source control product to a secure on-premises Git versioning system. Instead of deploying the website by using FTP, the website must automatically deploy to Azure each time developers check-in source files.

You need to implement the new deployment strategy.

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

Actions	Answer Area
In the Azure management portal, configure websites to support deployment from the local Git repository.	
In the Azure management portal, configure websites to support deployment from external repository sources.	
In the Azure management portal, configure websites to support deployment from Microsoft Visual Studio Online.	
Commit the website to Azure.	
Create the website and add it to the local Git repository.	

A120.

Actions	Answer Area
In the Azure management portal, configure websites to support deployment from the local Git repository.	Create the website and add it to the local Git repository.
In the Azure management portal, configure websites to support deployment from external repository sources.	In the Azure management portal, configure websites to support deployment from the local Git repository.
In the Azure management portal, configure websites to support deployment from Microsoft Visual Studio Online.	Commit the website to Azure.
Commit the website to Azure.	
Create the website and add it to the local Git repository.	

Q121.

Case Study

This is a case study. Case studies are not timed separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are a developer for Fabrikam, a company that specializes in payment processing. Fabrikam is developing a solution to process payments for various events, such as music concerts. You develop an ASP.NET MVC website that is hosted in Azure to support an upcoming music concert. The music concert is expected to generate a large volume of ticket sales in a short amount of time.

The website uploads information to an Azure storage queue. A worker role in Azure retrieves information from the queue and generates the concert tickets in a PDF file format after the financial transaction is approved.

You observe a delay between the time the website adds a message to a queue and the time it becomes available to read from the queue. After examining the queue, you determine that no queue messages have a DequeueCount value greater than zero. The website does not throw any errors.

Business Requirements

Payments

The music concert website must be able to submit event payment information for processing. The website must remain responsive while submitting payment information. Customers must be able to add notes about their orders to a free-form control on the website. These notes must be submitted with the payment when the customer submits an order.

Customers often enter notes that exceed 7 KB in size.

Technical Requirements

Payment Submission and Processing

Event payment information must be sent from the website to a Windows Communication Foundation (WCF) service worker role. The worker role must submit the information to the payment processor in JSON format.

Payment Processing

You have the following payment processing requirements:

- If the number of messages in a queue goes above or below a specified threshold, worker role instances must be created or deleted as needed. This process must be completed by using the least amount of effort. It must be easy to reconfigure role instance thresholds.
- Payments must be retrieved from the queue in the maximum batch sizes that are allowed by the queue and pulled from the queue for 5 minutes.
- The payment queue must not be re-created when processing payments.
- During single Payment processing, the number of tickets available for an event must be updated. The update operation must be retried for 30 seconds or 5 retry attempts, whichever occurs first. Each retry should pause for at least two seconds and for one second longer than the previous attempt. If the update fails, the payment should be placed in the poison queue.

Storage

You have the following storage requirements:

- Payment information must be stored by using Azure Queue storage. Connection to the Azure storage account has been established in a configured setting named StorageConnectionString, which is configured for the web and worker roles.
- A payment processing queue and a poison payment queue must be used when processing payments.
- Azure Queue message content must be XML-safe and UTF-8 encoded.
- An Azure storage account must be established for diagnostic information in a configured setting named DiagnosticsConnectionString, which is configured for both the web and worker roles.

Security and Monitoring

Security

The web role must be secured by using HTTPS.

Monitoring

You must collect diagnostic data for both the web and worker roles by using the Diagnostics module. Diagnostics configuration changes must not require the code of the roles to be rebuilt. The diagnostic data is used for debugging and troubleshooting, measuring performance, monitoring resource usage, traffic analysis and capacity planning, and auditing.

Performance testing must evaluate the roles under normal and stress conditions without incurring changes for running Azure. Memory allocation, function time, and multithreading concurrency issues must be evaluated.

Deployment

You purchase a custom domain name fabrikamfunding.com to host the website, web role, and worker roles. You must deploy an HTTPS certificate with the web role, and you must update associated configuration files accordingly.

Web role and worker role instance sizes must be specified as Medium. You must deploy one web role instance named FabrikamFundingPaymentGenerator, and worker role instances named FabrikamFundingPaymentProcessor.

Application Structure

Relevant portions of the app files are shown below. Line numbers are included for reference only and include a two-character prefix that denotes the specific file to which they belong.

CustomRetryPolicy.cs

```
CR01 public class CustomRetryPolicy : IRetryPolicy
CR02 {
CR03     int _retryCount = 0;
CR04     readonly TimeSpan _baseInterval= TimeSpan.FromSeconds(1);
CR05     readonly string _poisonPaymentQueueName;
CR06     private readonly CloudQueueClient _queueClient;
CR07     private readonly EventPayment _eventPayment;
CR08     public CustomRetryPolicy(string poisonPaymentQueueName, CloudQueueClient
queueClient, EventPayment eventPayment)
CR09     {
CR10         _poisonPaymentQueueName = poisonPaymentQueueName;
CR11         _queueClient = queueClient;
CR12         _eventPayment = eventPayment;
CR13     }
CR14     public IRetryPolicy CreateInstance()
CR15     {
CR16         return new CustomRetryPolicy(_poisonPaymentQueueName, _queueClient,
 _eventPayment);
CR17     }
CR18 }
```

Event.cs

```
EV01 public class Event : TableEntity
EV02 {
EV03     public int AvailableTickets { get; set; }
EV04 }
```

EventPayment.cs

```
EP01 [DataContract]
EP02 public class EventPayment
EP03 {
EP04     [DataMember]
EP05     public int EventId { get; set; }
EP06     [DataMember]
EP07     public string Email { get; set; }
EP08     [DataMember]
EP09     public string Notes { get; set; }
EP10     [DataMember]
EP11     public int TicketCount { get; set; }
EP12     [DataMember]
EP13     public DateTime OrderDate { get; set; }
EP14     [DataMember]
EP15     public Guid EventPaymentId { get; set; }
EP16 }
```

QueueManager.cs

```
QM01 public class QueueManager
QM02 {
QM03     private readonly CloudQueueClient _queueClient;
QM04     private readonly CloudTableClient _tableClient;
QM05     private const string PaymentQueueName = "paymentqueue";
QM06     private const string PoisonPaymentQueueName = "poisonpaymentqueue";
QM07     public QueueManager()
QM08     {
QM09         var storageAccount = CloudStorageAccount.Parse(
QM10             CloudConfigurationManager.GetSetting("StorageConnectionString"));
QM11         _queueClient = storageAccount.CreateCloudQueueClient();
QM12         _tableClient = storageAccount.CreateCloudTableClient();
QM13     }
QM14     public async Task SendMessageAsync(EventPayment eventPayment)
QM15     {
QM16         ...
QM17     }
QM18     public async Task ProcessMessagesAsync()
QM19     {
QM20         ...
QM21     }
QM22     public async Task ProcessPayment(EventPayment eventPayment)
QM23     {
QM24         var events = _tableClient.GetTableReference("events");
QM25         var key = eventPayment.EventId.ToString();
QM26         var operation = await events.ExecuteAsync(TableOperation.Re-
trieve<Event>(key, key));
QM27         var @event = operation.Result as Event;
QM28         @event.AvailableTickets = @event.AvailableTickets - eventPay-
ment.TicketCount;
QM29         var requestOptions = new TableRequestOptions
QM30         {
QM31             RetryPolicy = new CustomRetryPolicy(
QM32                 PoisonPaymentQueueName,
QM33                 _queueClient,
QM34                 eventPayment),
QM35         };
QM36         var context = new OperationContext
QM37         {
QM38             StartTime = DateTime.Now,
QM39         };
QM40         await events.ExecuteAsync(TableOperation.Replace(@event),
requestOptions, context);
QM41     }
QM42 }
```

You need to implement the `ProcessPaymentAsync` method in the `QueueManager` class.

Develop the solution by selecting and arranging the required code blocks in the correct order.

NOTE: You will not need all of the code segments.

Code segments

```
while (true)
{
    var messages =
        await queue.GetMessagesAsync(32, TimeSpan.FromMinutes(5),
        null, null);
    foreach (var message in messages.Where(message => message != null))
    {

```

```
        var eventPayment = JsonConvert.
        DeserializeObject<EventPayment>(messageAsString);
    }
}
```

```
    await queue.DeleteMessageAsync(message);
}
}
}
```

```
public async Task ProcessPaymentsAsync()
{
    var queue = _queueClient.GetQueueReference(PaymentQueueName);
    await queue.CreateAsync();
}
```

```
public async Task ProcessPaymentsAsync()
{
    var queue = _queueClient.GetQueueReference(PaymentQueueName);
    await queue.CreateIfNotExistsAsync();
}
```

```
    await ProcessPayment(eventPayment);
}
```

Answer Area



A121.

Worker Area

```
public async Task ProcessPaymentsAsync()
{
    var queue = _queueClient.GetQueueReference(PaymentQueueName);
    await queue.CreateIfNotExistsAsync();
```

```
while (true)
{
    var messages =
        await queue.GetMessagesAsync(32, TimeSpan.FromMinutes(5),
        null, null);
    foreach (var message in messages.Where(message => message != null))
    {
```



```
        await ProcessPayment
        (eventPayment);
```

```
    await queue.DeleteMessageAsync
    (message);
}
```

Q122.

You have a cloud service that runs an external process that is named MyStartupTask.cmd. The cloud service runs this external process when the web role starts. The external process writes information to the Windows registry. You set the value of an environment variable named MyId to the deployment ID for the current web role instance.

The external process must complete writing the information to the Windows registry before the web role starts to accept web traffic.

You need to configure the cloud service.

How should you complete the relevant markup? To answer, select the appropriate option or options in the answer area.

```
<Startup>
  <Task commandLine="MyStartupTask.cmd"
    executionContext="elevated" taskType="simple"
    executionContext="limited" taskType="foreground"
    executionContext="elevated" taskType="foreground"
    executionContext="elevated" taskType="background">

    <Environment>
      <Variable name="MyId">
        <RoleInstanceValue xpath="/RoleEnvironment/Deployment/@id"/>
        <RoleInstanceValue xpath="/DeploymentId"/>
        <RoleEnvironment.DeploymentId></value>
        <Value>@DeploymentId</Value>
      </Variable>
    </Environment>
  </Task>
</Startup>
```

A122.

```
<Startup>
  <Task commandLine="MyStartupTask.cmd"
    executionContext="elevated" taskType="simple"
    executionContext="limited" taskType="foreground"
    executionContext="elevated" taskType="foreground"
    executionContext="elevated" taskType="background">

    <Environment>
      <Variable name="MyId">
        <RoleInstanceValue xpath="/RoleEnvironment/Deployment/@id"/>
        <RoleInstanceValue xpath="/DeploymentId"/>
        <RoleEnvironment.DeploymentId></value>
        <Value>@DeploymentId</Value>
      </Variable>
    </Environment>
  </Task>
</Startup>
```