Custom View Settings

Question #18 *Topic 2*

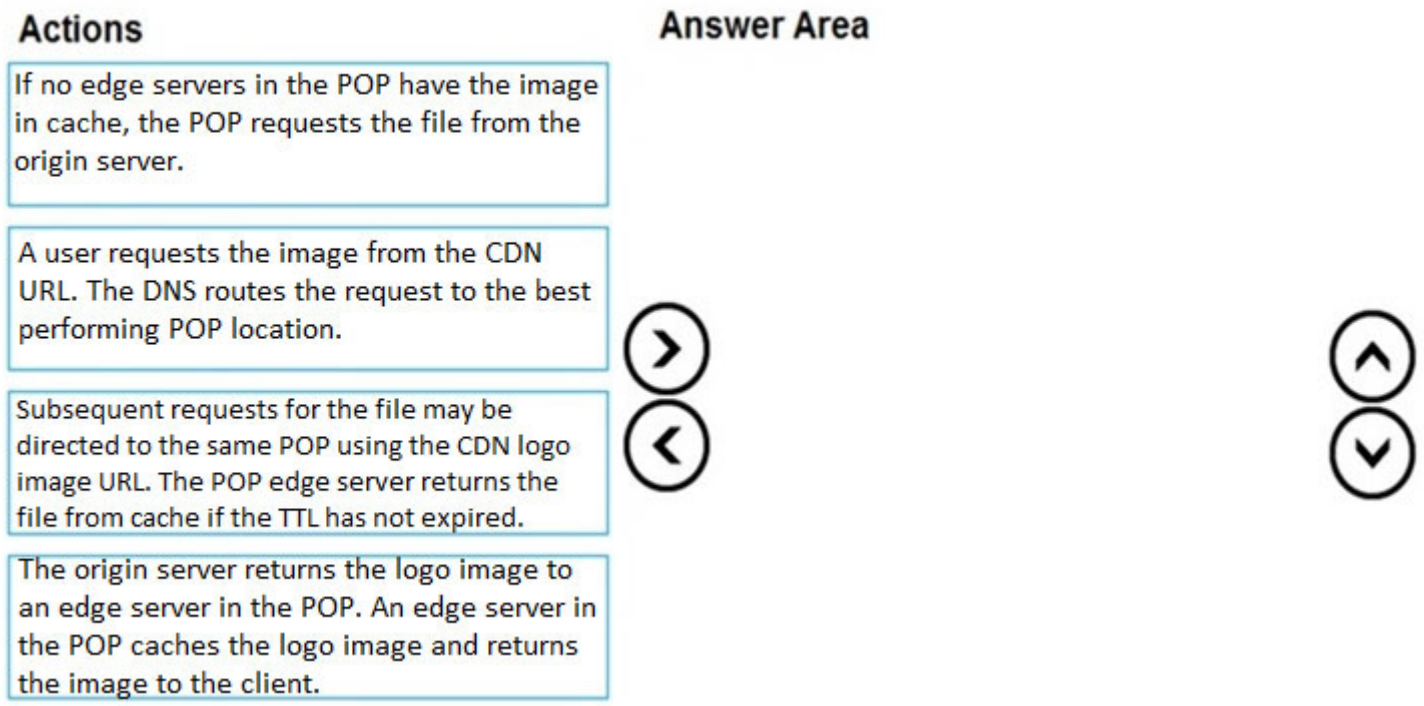
DRAG DROP -

Your company has several websites that use a company logo image. You use Azure Content Delivery Network (CDN) to store the static image.

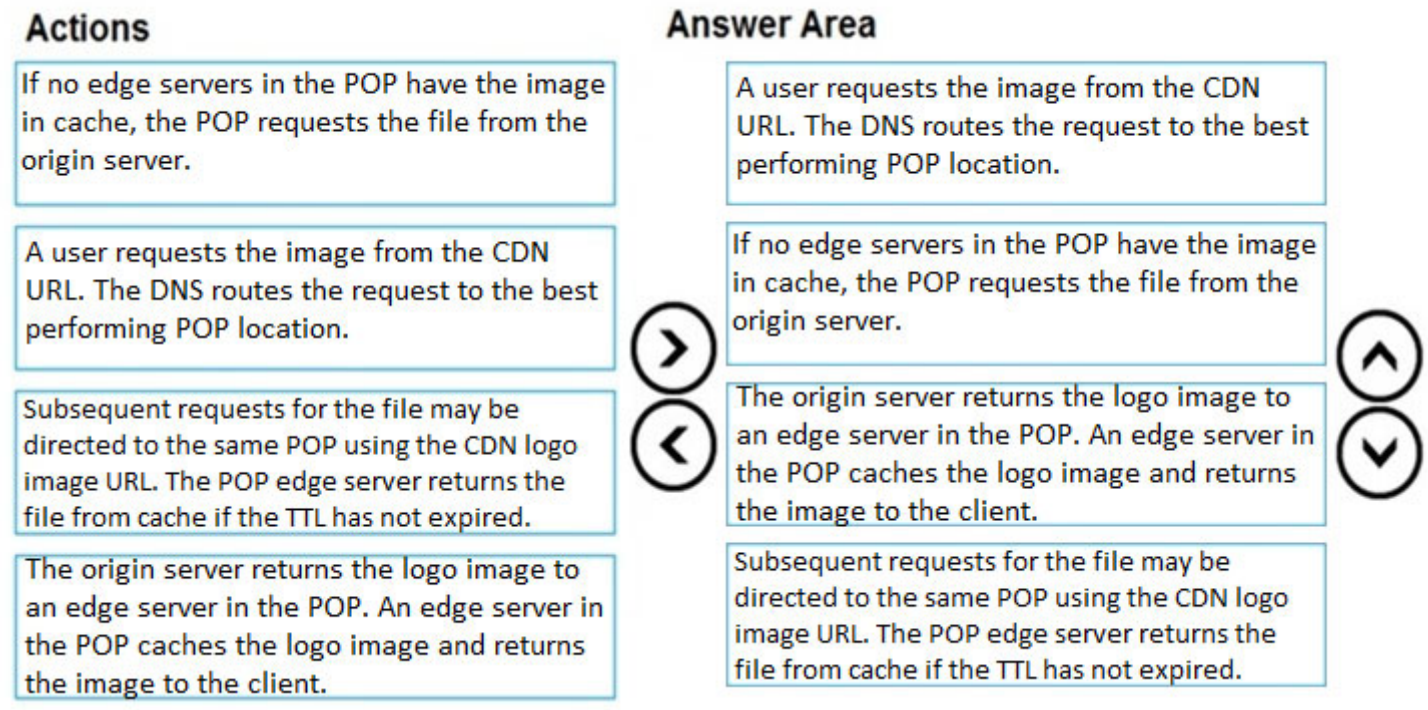
You need to determine the correct process of how the CDN and the Point of Presence (POP) server will distribute the image and list the items in the correct order.

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

Select and Place:




**Correct Answer:**




Step 1: A user requests the image..

A user requests a file (also called an asset) by using a URL with a special domain name, such as <endpoint name>.azureedge.net. This name can be an endpoint hostname or a custom domain. The DNS routes the request to the best performing POP location, which is usually the POP that is geographically closest to the user.

Step 2: If no edge servers in the POP have the..

If no edge servers in the POP have the file in their cache, the POP requests the file from the origin server. The origin server can be an Azure Web App, Azure

Cloud Service, Azure Storage account, or any publicly accessible web server.

Step 3: The origin server returns the..

The origin server returns the file to an edge server in the POP

The origin server returns the file to an edge server in the POP.
An edge server in the POP caches the file and returns the file to the original requestor (Alice). The file remains cached on the edge server in the

POP until the time-to-live (TTL) specified by its HTTP headers expires. If the origin server didn't specify a TTL, the default TTL is seven days.
Step 4: Subsequent requests for..
Additional users can then request the same file by using the same URL that the original user used, and can also be directed to the same POP.
If the TTL for the file hasn't expired, the POP edge server returns the file directly from the cache. This process results in a faster, more responsive user experience.
Reference:
https://docs.microsoft.com/en-us/azure/cdn/cdn-overview

Question #19 *Topic 2*

You are developing an Azure Cosmos DB solution by using the Azure Cosmos DB SQL API. The data includes millions of documents. Each document may contain hundreds of properties.
The properties of the documents do not contain distinct values for partitioning. Azure Cosmos DB must scale individual containers in the database to meet the performance needs of the application by spreading the workload evenly across all partitions over time.
You need to select a partition key.
Which two partition keys can you use? Each correct answer presents a complete solution.
NOTE: Each correct selection is worth one point.

A. a single property value that does not appear frequently in the documents

B. a value containing the collection name

C. a single property value that appears frequently in the documents

D. a concatenation of multiple property values with a random suffix appended

E. a hash suffix appended to a property value

Question #20 *Topic 2*

HOTSPOT -

You are developing an Azure-hosted e-commerce web application. The application will use Azure Cosmos DB to store sales orders. You are using the latest SDK to manage the sales orders in the database.

You create a new Azure Cosmos DB instance. You include a valid endpoint and valid authorization key to an appSettings.json file in the code project.

You are evaluating the following application code: (Line number are included for reference only.)

```
01 using System;
02 using System.Threading.Tasks;
03 using Microsoft.Azure.Cosmos;
04 using Microsoft.Extensions.Configuration;
05 using Newtonsoft.Json;
06 namespace SalesOrders
07 {
08   public class SalesOrder
09   {
10     . . .
11   }
12   internal class ManageSalesOrders
13     {
14       private static async Task GenerateSalesOrders()
15         {
16           IConfigurationRoot configuration = new ConfigurationBuilder().AddJsonFile("appSettings.json").Build();
17           string endpoint = configuration["EndPointUrl"];
18           string authKey = configuration["AuthorizationKey"];
19           using CosmosClient client = new CosmosClient(endpoint, authKey);
20           Database database = null;
21           using (await client.GetDatabase("SalesOrders").DeleteStreamAsync()) { }
22           database = await client.CreateDatabaseIfNotExistsAsync("SalesOrders");
23           Container container1 = await database.CreateContainerAsync(id: "Container1", partitionKeyPath: "/AccountNumber");
24           Container container2 = await database.CreateContainerAsync(id: "Container2", partitionKeyPath: "/AccountNumber");
25           SalesOrder salesOrder1 = new SalesOrder() { AccountNumber = "123456" };
26           await container1.CreateItemAsync(salesOrder1, new PartitionKey(salesOrder1.AccountNumber));
27           SalesOrder salesOrder2 = new SalesOrder() { AccountNumber = "654321" };
28           await container1.CreateItemAsync(salesOrder2, new PartitionKey(salesOrder2.AccountNumber));
29           SalesOrder salesOrder3 = new SalesOrder() { AccountNumber = "109876" };
30           await container2.CreateItemAsync(salesOrder3, new PartitionKey(salesOrder3.AccountNumber));
31           _ = await database.CreateUserAsync("User1");
32           User user1 = database.GetUser("User1");
33           _ = await user1.ReadAsync();
34     }
35   }
36 }
```

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

NOTE: Each correct selection is worth one point.

Hot Area:

**Answer Area**

| Statements | Yes | No |
|---|---|---|
| A database named SalesOrders is created. The database will include two containers. | ○ | ○ |
| Container1 will contain two items. | ○ | ○ |
| Container2 will contain one item. | ○ | ○ |

**Answer Area**

| Statements | Yes | No |
|---|---|---|
| A database named SalesOrders is created. The database will include two containers. | ○ | ○ |
| Container1 will contain two items. | ○ | ○ |
| Container2 will contain one item. | ○ | ○ |

**Correct Answer:**

Box 1: Yes -

The createDatabaseIfNotExistsAsync method checks if a database exists, and if it doesn't, create it.

The Database.CreateContainerAsync method creates a container as an asynchronous operation in the Azure Cosmos service.

Box 2: Yes -

The CosmosContainer.CreateItemAsync method creates an item as an asynchronous operation in the Azure Cosmos service.

Box 3: Yes -

Reference:

https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.cosmosclient.createdatabaseifnotexistsasync

https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cosmos.database.createcontainerasync https://docs.microsoft.com/en-us/dotnet/api/azure.cosmos.cosmoscontainer.createitemasync

Question #21 *Topic 2*

DRAG DROP -

You develop an Azure solution that uses Cosmos DB.

The current Cosmos DB container must be replicated and must use a partition key that is optimized for queries.

You need to implement a change feed processor solution.

Which change feed processor components should you use? To answer, drag the appropriate components to the correct requirements. Each component may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view the content.

NOTE: Each correct selection is worth one point.

Select and Place:

**Components**

- Host
- Delegate
- Lease container
- Monitored container

**Answer Area**

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

**Correct Answer:**

**Components**

**Answer Area**

| Requirement | Component |
|---|---|
| Store the data from which the change feed is generated. | Monitored container |
| Coordinate processing of the change feed across multiple workers. | Lease container |
| Use the change feed processor to listen for changes. | Host |
| Handle each batch of changes. | Delegate |

Box 1: The monitored container -

The monitored container has the data from which the change feed is generated. Any inserts and updates to the monitored container are reflected in the change feed of the container.

Box 2: The lease container -

The lease container acts as a state storage and coordinates processing the change feed across multiple workers. The lease container can be stored in the same account as the monitored container or in a separate account.

Box 3: The host: A host is an application instance that uses the change feed processor to listen for changes. Multiple instances with the same lease configuration can run in parallel, but each instance should have a different instance name.

Box 4: The delegate -

The delegate is the code that defines what you, the developer, want to do with each batch of changes that the change feed processor reads.

Reference:

https://docs.microsoft.com/en-us/azure/cosmos-db/change-feed-processor

## Topic 3 - Question Set 3

Question #1 *Topic 3*

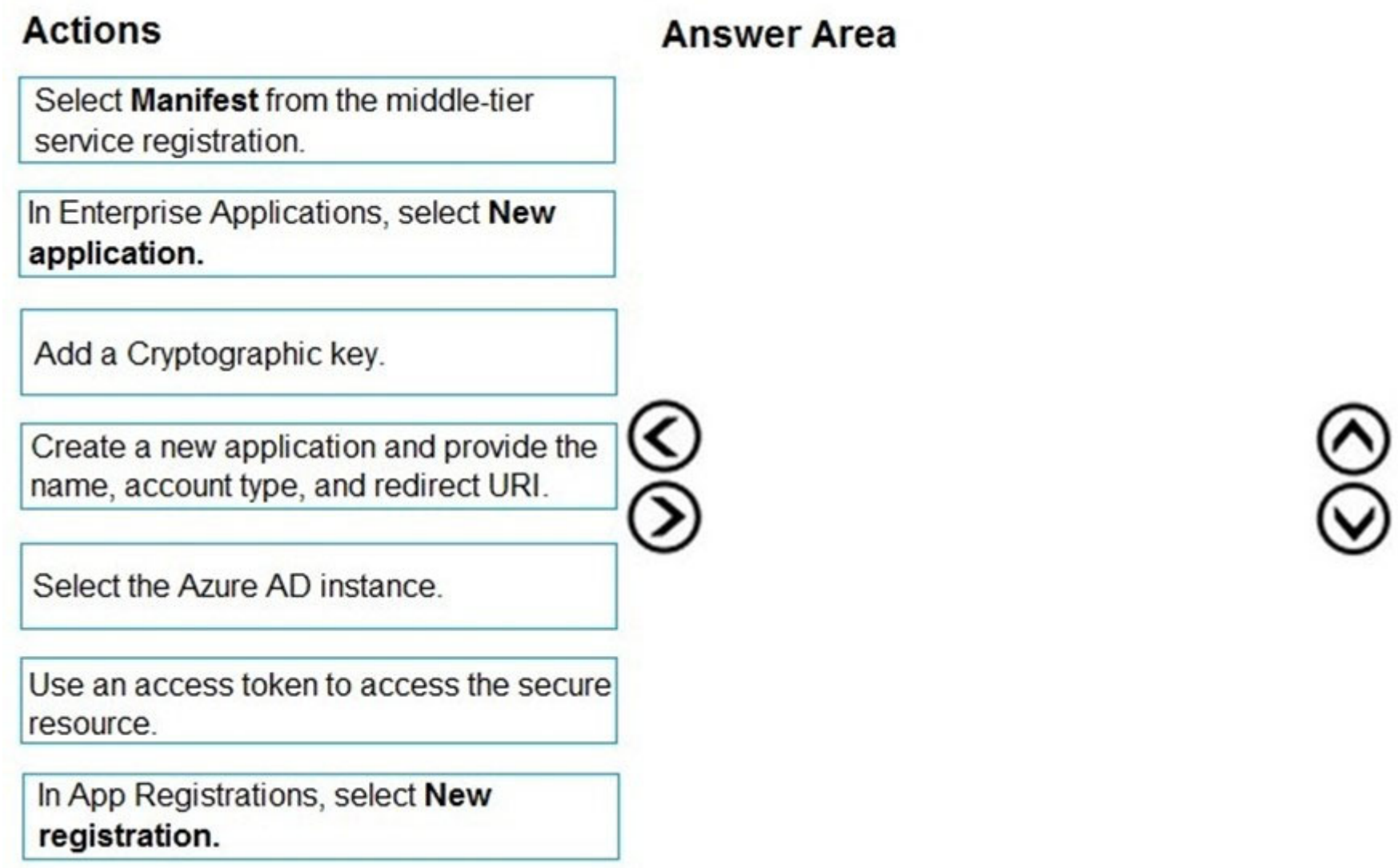
DRAG DROP -

You develop a web application.

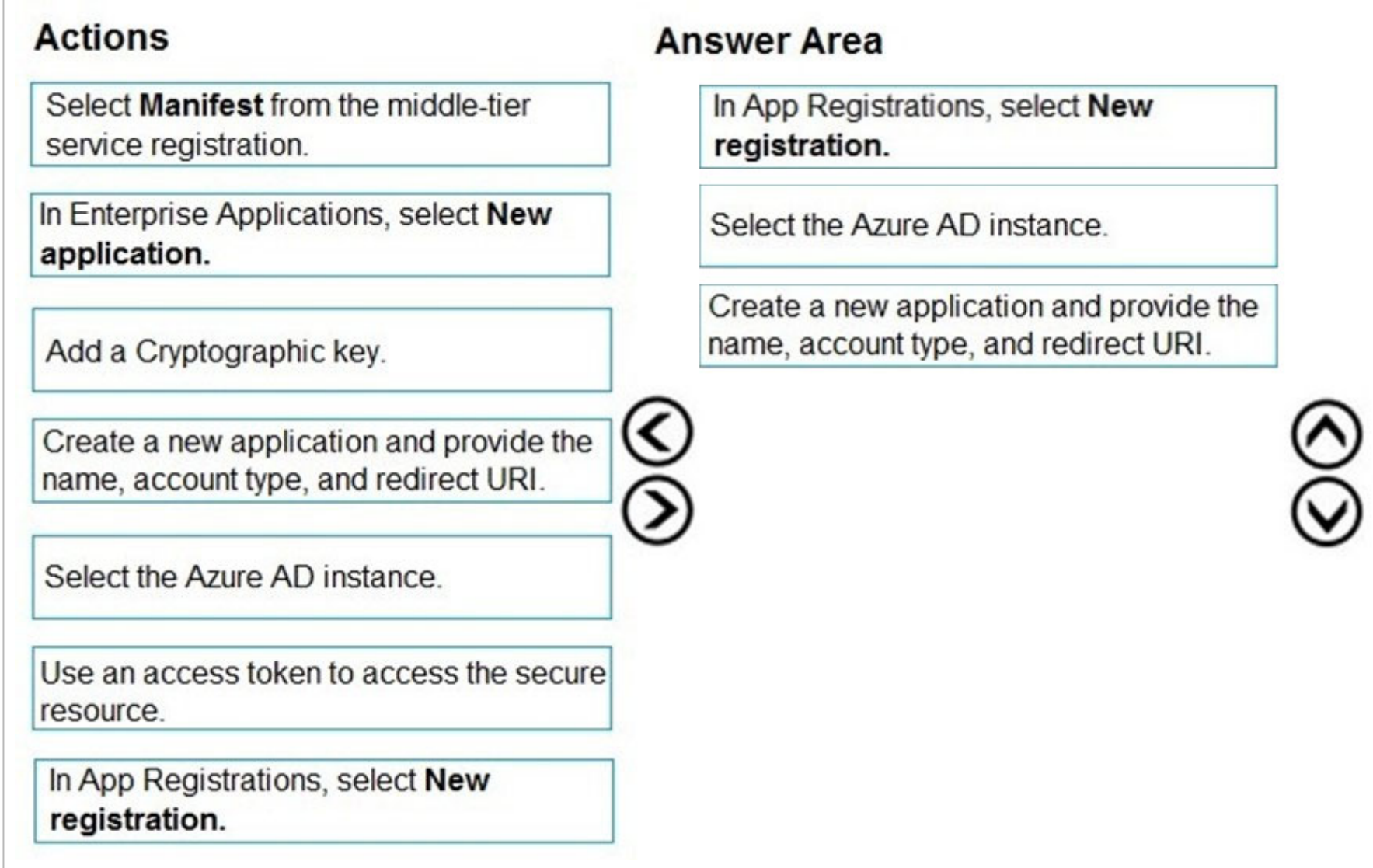
You need to register the application with an active Azure Active Directory (Azure AD) tenant.

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

Select and Place:




**Correct Answer:**




Register a new application using the Azure portal

1. Sign in to the Azure portal using either a work or school account or a personal Microsoft account.

2. If your account gives you access to more than one tenant, select your account in the upper right corner. Set your portal session to the Azure AD tenant that you want.

3. Search for and select Azure Active Directory. Under Manage, select App registrations.
4. Select New registration. (Step 1)
5. In Register an application, enter a meaningful application name to display to users.
6. Specify who can use the application. Select the Azure AD instance. (Step 2)
7. Under Redirect URI (optional), select the type of app you're building: Web or Public client (mobile & desktop). Then enter the redirect URI, or reply URL, for your application. (Step 3)
8. When finished, select Register.

Previous Questions

Next Questions