whether the user holds an editor claim of partner.
- Limit access to the Manage action of the controller to users with an editor claim of partner who are also members of the SysAdmin role.

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

NOTE: Each correct selection is worth one point.

Code segments

```
[Authorize(Policy ="ProviderPartner")]
[Authorize(Role ="SysAdmin")]
```

```
[Authorize(Role ="ProviderAdmin")]
[Authorize(Role ="SysAdmin")]
```

```
[Authorize(Role ="SysAdmin", "ProviderAdmin")]
```

```
[Authorize(Policy ="ProviderPartner",
Role="SysAdmin")]
```

Answer Area

```
[ ]
public class PartnerController : Controller
{
  . . .
[ ]
Public ActionResult Manage()
{
  . . .
}
}
```

Correct Answer:

Code segments

```
[Authorize(Policy ="ProviderPartner")]
[Authorize(Role ="SysAdmin")]
```

```
[Authorize(Role ="ProviderAdmin")]
[Authorize(Role ="SysAdmin")]
```

```
[Authorize(Role ="SysAdmin", "ProviderAdmin")]
```

```
[Authorize(Policy ="ProviderPartner",
Role="SysAdmin")]
```

Answer Area

```
[Authorize(Role ="SysAdmin", "ProviderAdmin")]
public class PartnerController : Controller
{
  . . .
[Authorize(Policy ="ProviderPartner")]
[Authorize(Role ="SysAdmin")]
Public ActionResult Manage()
{
  . . .
}
}
```

QUESTION 31
DRAG DROP
You are developing Azure WebJobs.

You need to recommend a WebJob type for each scenario.

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

NOTE: Each correct selection is worth one point.

Answer Area

WebJob types	Scenario	WebJob type
Triggered	Run on all instances that the web app runs on. Optionally restrict the WebJob to a single instance.	
Continuous	Run on a single instance that Azure select for load balancing.	
	Supports remote debugging	

Correct Answer:

Answer Area

WebJob types	Scenario	WebJob type
Triggered	Run on all instances that the web app runs on. Optionally restrict the WebJob to a single instance.	Continuous
Continuous	Run on a single instance that Azure select for load balancing.	Triggered
	Supports remote debugging	Continuous

QUESTION 32

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You are developing and deploying several ASP.NET web applications to Azure App Service. You plan to save session state information and HTML output.

You must use a storage mechanism with the following requirements:

- Share session state across all ASP.NET web applications.
- Support controlled, concurrent access to the same session state data for multiple readers and a single writer.
- Save full HTTP responses for concurrent requests.

You need to store the information.

Solution: Enable Application Request Routing (ARR).

Does the solution meet the goal?

A. Yes
B. No

Correct Answer: B
Explanation:
Instead deploy and configure Azure Cache for Redis. Update the web applications.

Reference:
https://docs.microsoft.com/en-us/azure/architecture/best-practices/caching#managing-concurrency-in-a-cache

QUESTION 33
HOTSPOT
You are developing an app that manages users for a video game. You plan to store the region, email address, and phone number for the player. Some players may not have a phone number. The player's region will be used to load-balance data.

Data for the app must be stored in Azure Table Storage.

You need to develop code to retrieve data for an individual player.

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

NOTE: Each correct selection is worth one point.

```
public class PlayerEntity : TableEntity
{
   public PlayerEntity()
   {
   }
   public PlayerEntity(string region, string email)
   {
     ParitionKey = [email|phone|region] ;
     RowKey= [email|phone|region] ;
   }
     public string Phone { get; set; }
   }
   public class Player
   }

protected PlayerEntity player;
async void GetPlayer(string cs, [CloudTable|CloudTableClient|TableEntity|TableEntityAdapter] table, string pk, string rk)
{
   [TableEntity query =TableEntity.Retrieve<PlayerEntity>(pk, rk);
   TableOperation query =TableOperation.Retrieve<PlayerEntity>(pk,rk);
   TableResult query =TableQuery.Retrieve<PlayerEntity>(pk,rk);
   TableResultSegment query =TableResult.Retrieve<PlayerEntity>(pk, rk);]

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

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

Correct Answer:

```
public class PlayerEntity : TableEntity
{
   public PlayerEntity()
   {
   }
   public PlayerEntity(string region, string email)
   {
     ParitionKey = [ ▼ ] ;
                   email
                   phone
                   region
     RowKey=       [ ▼ ] ;
                   email
                   phone
                   region
   }
     public string Phone { get; set; }
   }
   public class Player
   }

protected PlayerEntity player;
async void GetPlayer(string cs, [ ▼ ] table, string pk, string rk)
                                 CloudTable
                                 CloudTableClient
                                 TableEntity
                                 TableEntityAdapter
{
   [ ▼ ]
     TableEntity query =TableEntity.Retrieve<PlayerEntity>(pk, rk);
     TableOperation query =TableOperation.Retrieve<PlayerEntity>(pk,rk);
     TableResult query =TableQuery.Retrieve<PlayerEntity>(pk,rk);
     TableResultSegment query =TableResult.Retrieve<PlayerEntity>(pk, rk);
   [ ▼ ]
     TableEntity data =await table.ExecuteAsync(query);
     TableOperation data =await.table.ExeucteAsync(query);
     TableQuery data =await table.ExecuteAsync(query);
     TableResult data =await table.ExecuteAsync(query);
 player=data.Result as PlayerEntity;
 }
}
```

QUESTION 34
You are developing an application to store information about the organizational structure for a company.

Users must be able to determine which people report to a particular manager, the office where employees work, and the projects that are assigned to an employee.

Which Azure Cosmos DB API should you use for the application?