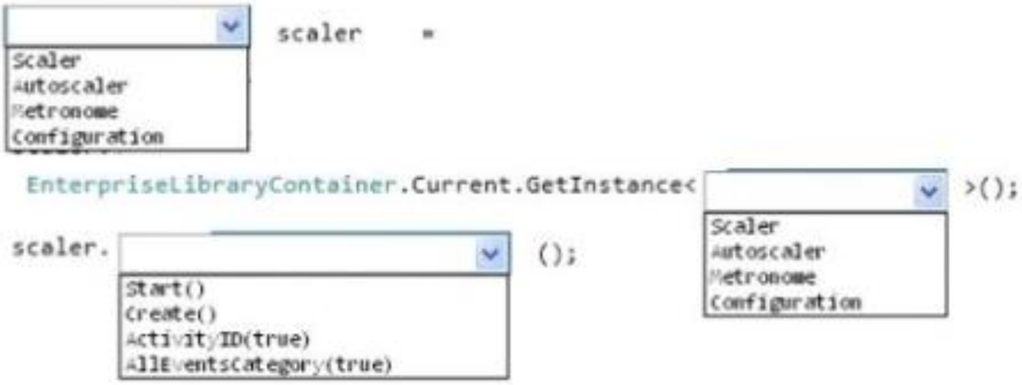
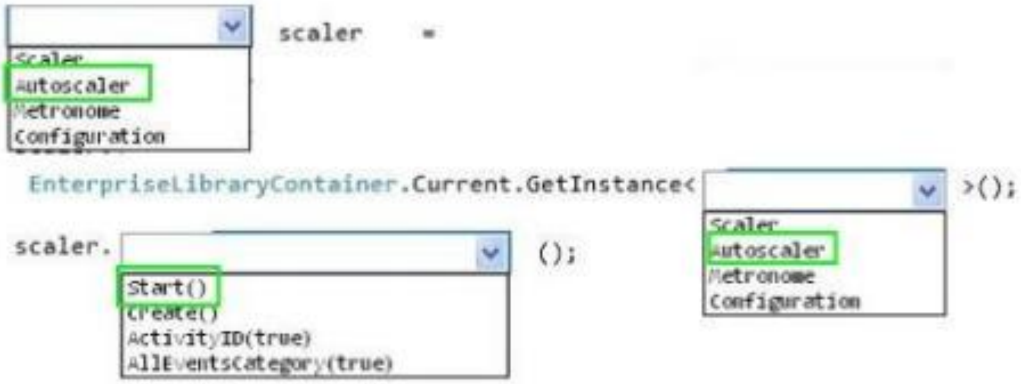
1) A company creates an Azure worker role to manage products. The number of customers who inquire about how many products are in inventory rapidly increases. You need to ensure that the worker role can scale to accommodate the increased workload. How should you complete the relevant code?
To answer, select the appropriate option or options in the answer area.



```
scaler =
EnterpriseLibraryContainer.Current.GetInstance<  >();
scaler.  ();
```

Answer:

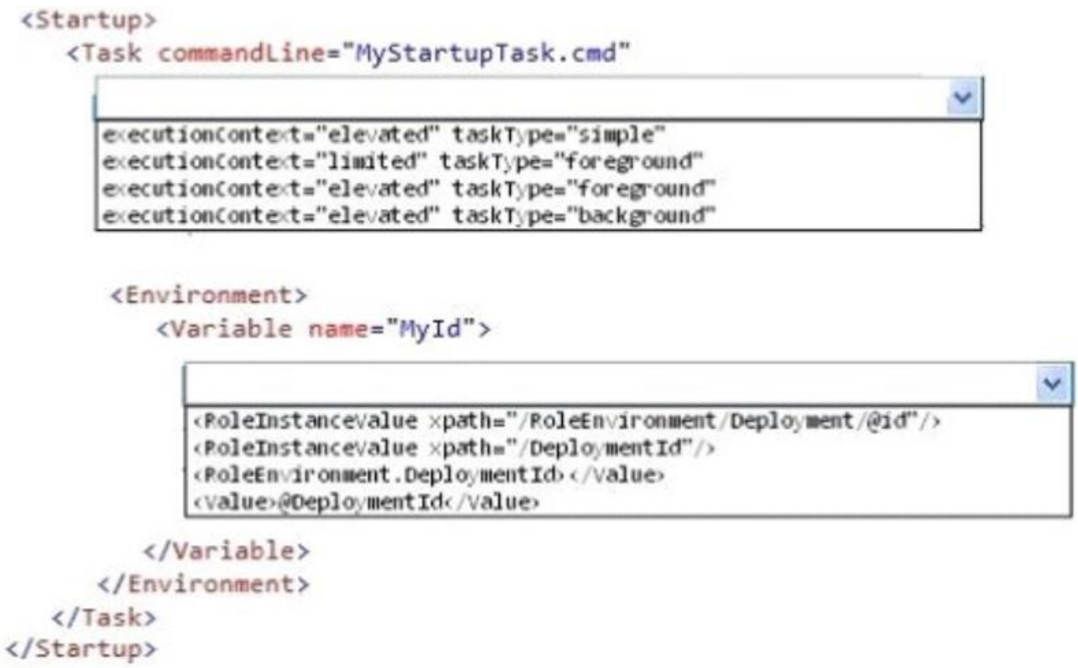


```
scaler =
EnterpriseLibraryContainer.Current.GetInstance<  >();
scaler.  ();
```

2) 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**

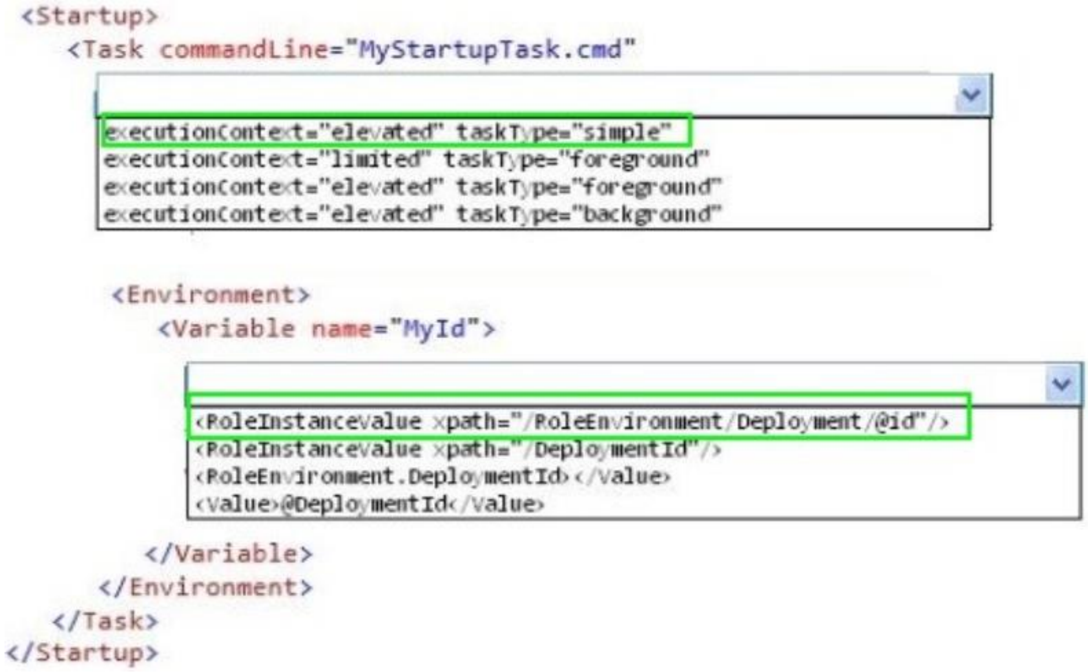
3) You have a cloud service that runs an external process that is named MyStartupTask.cmd. The cloud service runs this external process when the web role starts. The external process writes information to the Windows registry. You set the value of an environment variable named MyID to the deployment 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>
```

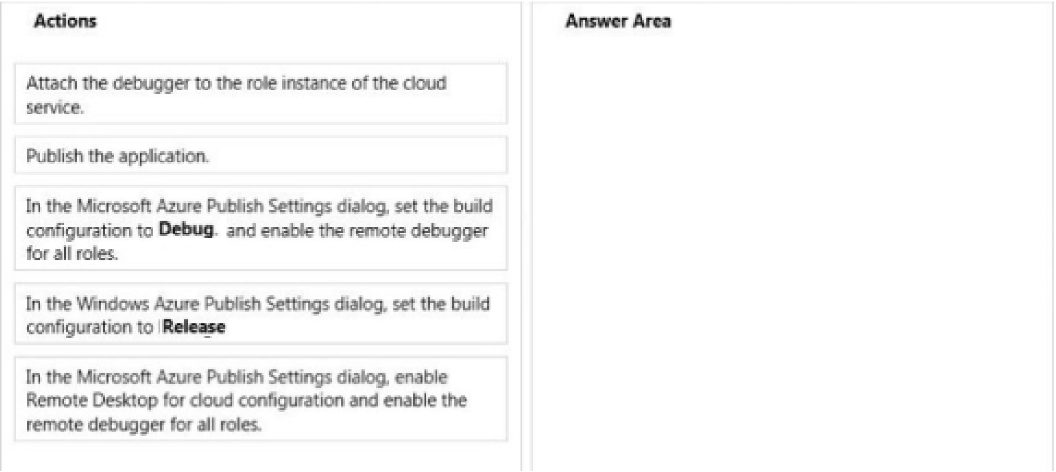
Answer:



```
<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>
```

4) Drag and Drop Question.
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

Attach the debugger to the role instance of the cloud service.

Publish the application.

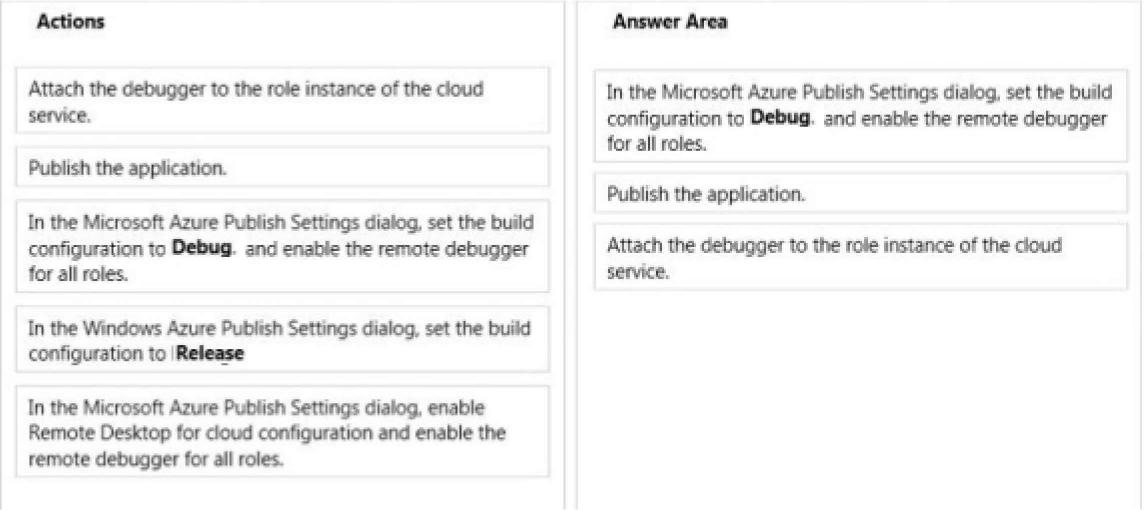
In the Microsoft Azure Publish Settings dialog, set the build configuration to **Debug**. and enable the remote debugger for all roles.

In the Windows Azure Publish Settings dialog, set the build configuration to **Release**

In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.

Answer Area

Answer:



Actions

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 **Debug**. and enable the remote debugger for all roles.

In the Windows Azure Publish Settings dialog, set the build configuration to **Release**

In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.

Answer Area

In the Microsoft Azure Publish Settings dialog, set the build configuration to **Debug**. and enable the remote debugger for all roles.

Publish the application.

Attach the debugger to the role instance of the cloud service.

5) You are designing an upgrade strategy for a Windows Azure application that includes one web role with one instance. You have the following requirements:
- Test the application on the Windows Azure platform.
- Ensure that application upgrades can be rolled back.
- Ensure that upgrade and rollback processes do not cause downtime.
You need to recommend an approach for upgrading the application. What should you recommend?

a) Deploy to the Production slot.
Test the application, and then perform a VIP swap.

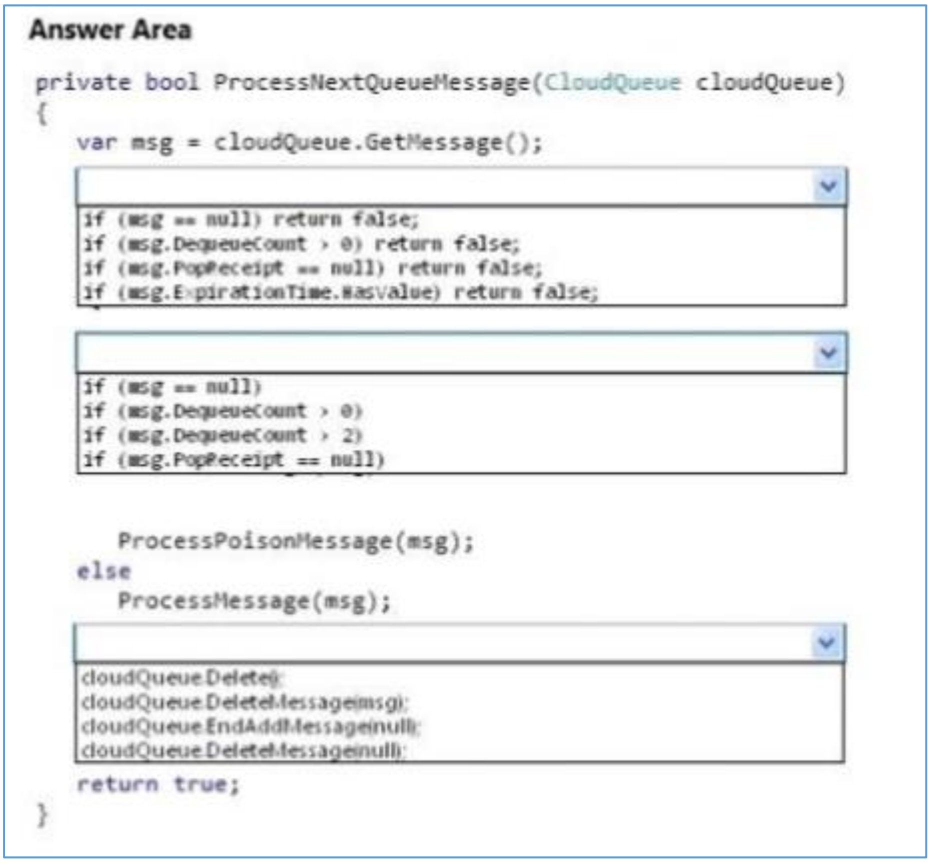
b) Deploy to the Staging slot.
Test the application, and then perform a VIP swap.

c) Deploy to the Staging slot.
Test the application, and then perform a manual in-place upgrade to the Production slot.

d) Deploy to the Staging slot.
Test the application, and then perform an automatic in-place upgrade to the Production slot.

Answer: B

6) 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;

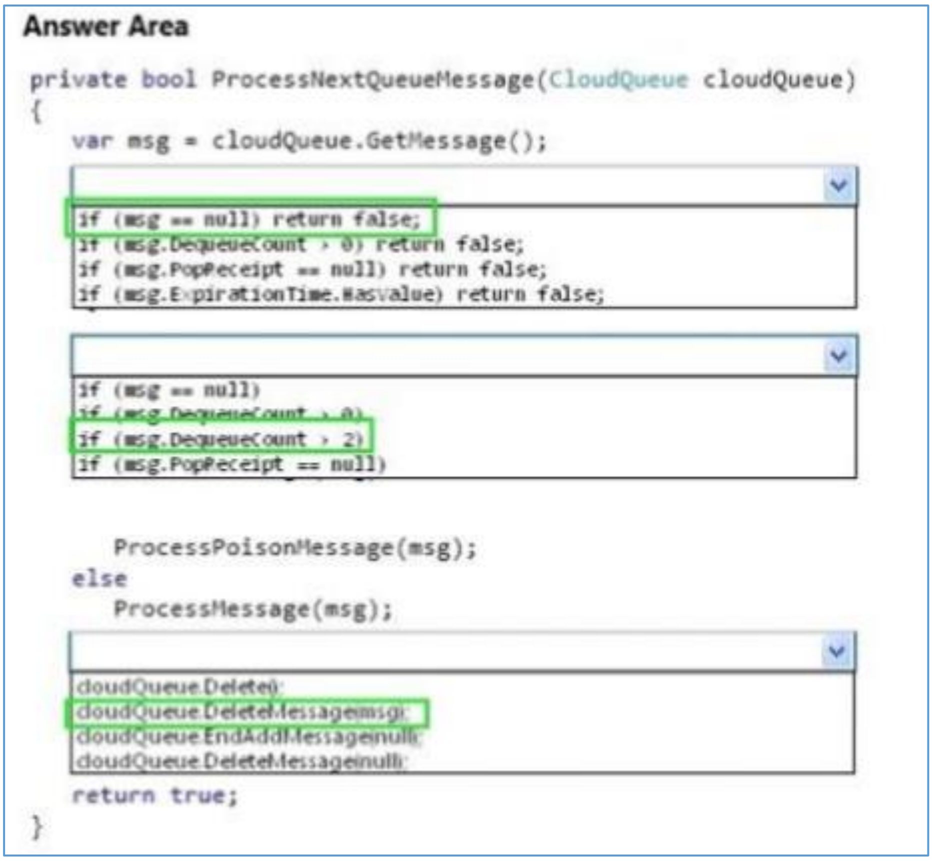
- if (msg == null)
- if (msg.DequeueCount > 0)
- if (msg.DequeueCount > 2)
- if (msg.PopReceipt == null)

```
        ProcessPoisonMessage(msg);
    else
        ProcessMessage(msg);
```

- cloudQueue.Delete();
- cloudQueue.DeleteMessage(msg);
- cloudQueue.EndAddMessage(null);
- cloudQueue.DeleteMessage(null);

```
    return true;
}
```

Answer:



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;

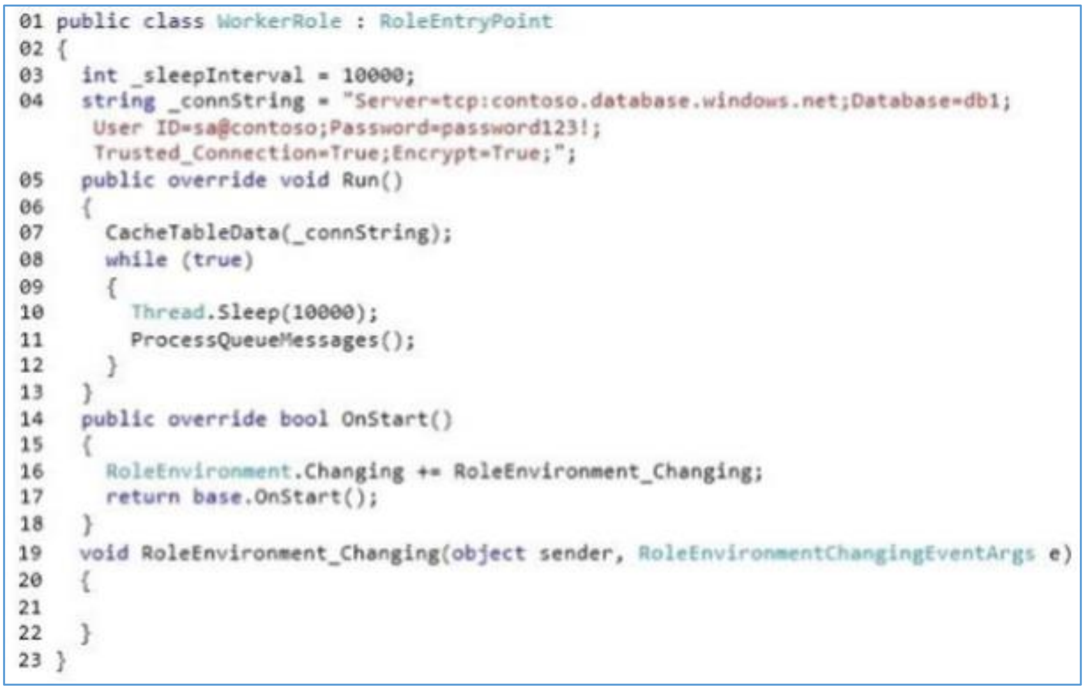
if (msg == null)
if (msg.DequeueCount > 0)
if (msg.DequeueCount > 2)
if (msg.PopReceipt == null)

```
        ProcessPoisonMessage(msg);
    else
        ProcessMessage(msg);
```

cloudQueue.Delete();
cloudQueue.DeleteMessage(msg);
cloudQueue.EndAddMessage(null);
cloudQueue.DeleteMessage(null);

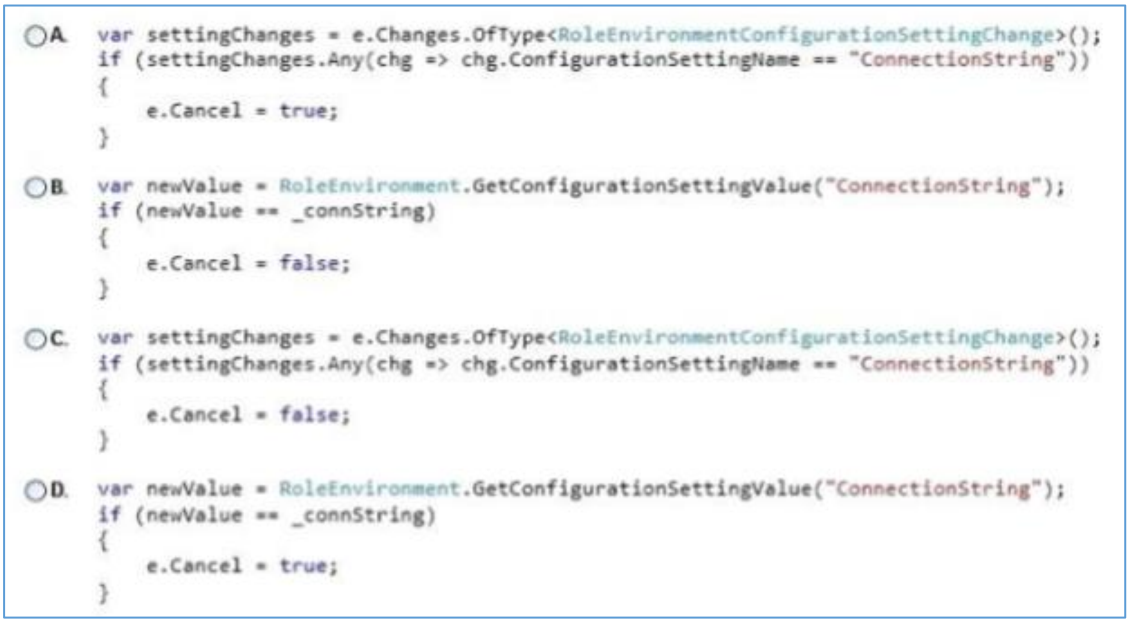
```
    return true;
}
```

7) 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;
      User ID=sa@contoso;Password=password123!;
      Trusted_Connection=True;Encrypt=True;";
05   public override void Run()
06   {
07     CacheTableData(_connString);
08     while (true)
09     {
10       Thread.Sleep(10000);
11       ProcessQueueMessages();
12     }
13   }
14   public override bool OnStart()
15   {
16     RoleEnvironment.Changing += RoleEnvironment_Changing;
17     return base.OnStart();
18   }
19   void RoleEnvironment_Changing(object sender, RoleEnvironmentChangingEventArgs e)
20   {
21
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;
}
```

Answer: A

8)

9)