LoginEvent.cs

```csharp
public class LoginEvent
{

  public string subject { get; set; }
  public DateTime eventTime { get; set; }
  public Dictionary<string, string> data { get; set; }
  public string Serialize()
  {
    return JsonConvert.SerializeObject(this);
  }
}
```

QUESTION 1

HOTSPOT

You need to implement the Log policy.

How should you complete the EnsureLogging method in EventGridController.cs? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

```csharp
var client = new WebSiteManagementClient(. . .);
var id = ParseResourceID(resource);
var appSettings = new StringDictionary(name: "properties",
  properties: new Dictionary<string, string> {
  {"DIAGNOSTICS_AZUREBLOBCONTAINERSASURL", BlobStoreAccountSAS("          ")},
                                                                   logs
                                                                   logdrop
  {"DIAGNOSTICS_AZUREBLOBRETENTIONINDAYS", "      "}
                                            15
                                            30
});
client.WebApps.                           (
               UploadLoggingSettings
               UpdateApplicationSetting
  id.resourceGroup,
  id.name, appSettings);
```

Correct Answer:

QUESTION 2
You need to resolve a notification latency issue.

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

NOTE: Each correct selection is worth one point.

A. Set Always On to true .
B. Ensure that the Azure Function is using an App Service plan.
C. Set Always On to false .
D. Ensure that the Azure Function is set to use a consumption plan.

Correct Answer: AB
Explanation:
Azure Functions can run on either a Consumption Plan or a dedicated App Service Plan. If you run in a dedicated mode, you need to turn on the Always On setting for your Function App to run properly. The Function runtime will go idle after a few minutes of inactivity, so only HTTP triggers will actually "wake up" your functions. This is similar to how WebJobs must have Always On enabled.

Scenario: Notification latency: Users report that anomaly detection emails can sometimes arrive several minutes after an anomaly is detected.

Anomaly detection service: You have an anomaly detection service that analyzes log information for anomalies. It is implemented as an Azure Machine Learning model. The model is deployed as a web service. If an anomaly is detected, an Azure Function that emails administrators is called by using an HTTP WebHook.

Reference:
https://github.com/Azure/Azure-Functions/wiki/Enable-Always-On-when-running-on-dedicated-App-Service-Plan

QUESTION 3
DRAG DROP
You need to implement telemetry for non-user actions.

How should you complete the Filter class? 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

- /health
- /status
- RequestTelemetry
- PageViewTelemetry
- ITelementryProcessor
- ITelementryInitializer

Answer Area

```
public class Filter : [ code segment ]
{
   private readonly [ code segment ] _next;
   public (Filter [ code segment ] next)
   {
   _next = next;
   }
   public void Process(ITelemetry item)
   {
     var x = item as [ code segment ] ;
      if (x?.Url.AbsolutePath == " [ code segment ] " )
     {
       return;
     }
     _next.Process(item);
   }
}
```

Correct Answer:

Code segments

- /health
- /status
- RequestTelemetry
- PageViewTelemetry
- ITelementryProcessor
- ITelementryInitializer

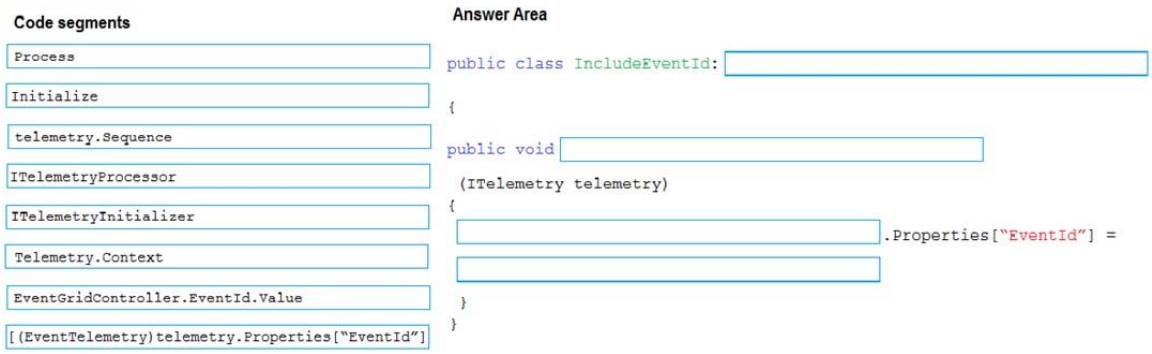
Answer Area

```
public class Filter : ITelementryProcessor
{
   private readonly ITelementryProcessor _next;
   public (Filter ITelementryProcessor next)
   {
   _next = next;
   }
   public void Process(ITelemetry item)
   {
     var x = item as RequestTelemetry ;
      if (x?.Url.AbsolutePath == " /health " )
     {
       return;
     }
     _next.Process(item);
   }
}
```

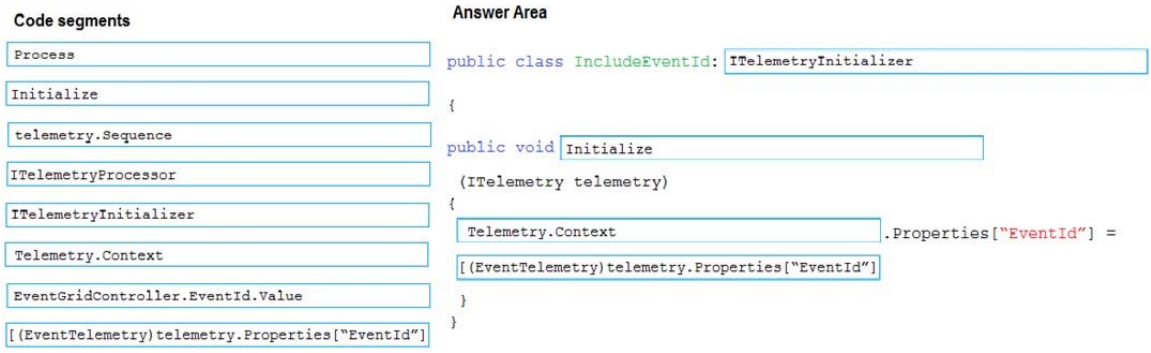
QUESTION 4
DRAG DROP
You need to ensure that PolicyLib requirements are met.

How should you complete the code segment? 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.



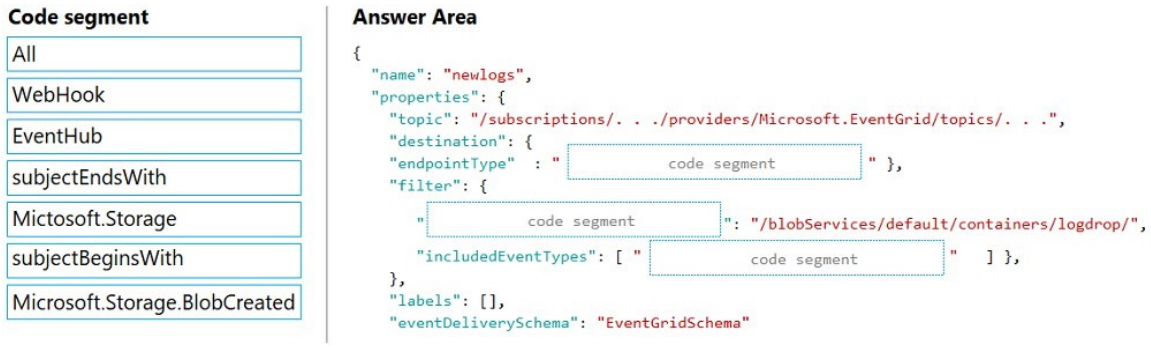
Correct Answer:



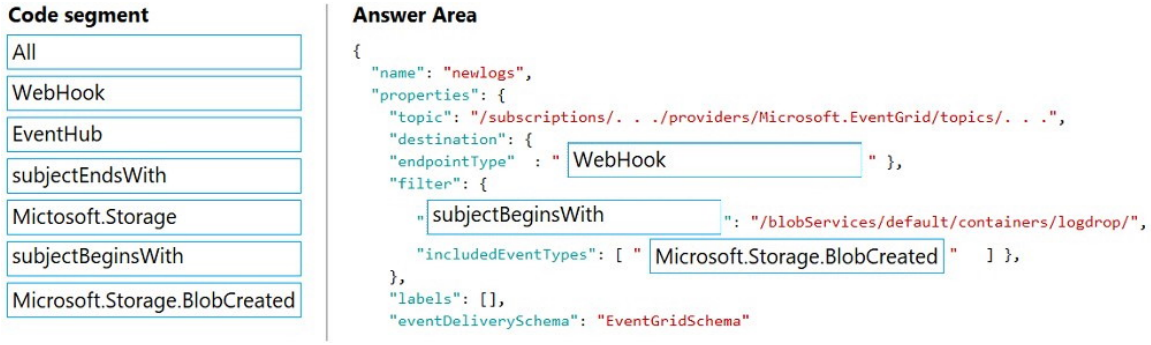
QUESTION 5
DRAG DROP
You need to implement the Log policy.

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

NOTE: Each correct selection is worth one point.



Correct Answer:



QUESTION 6
You need to ensure that the solution can meet the scaling requirements for Policy Service. Which Azure Application Insights data model should you use?

A. an Application Insights dependency
B. an Application Insights event
C. an Application Insights trace
D. an Application Insights metric

Correct Answer: D
Explanation:
Application Insights provides three additional data types for custom telemetry:

- Trace - used either directly, or through an adapter to implement diagnostics logging using an instrumentation framework that is familiar to you, such as Log4Net or System.Diagnostics.
- Event - typically used to capture user interaction with your service, to analyze usage patterns.
- Metric - used to report periodic scalar measurements.

Scenario:
Policy service must use Application Insights to automatically scale with the number of policy actions that it is performing.

Reference:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/data-model