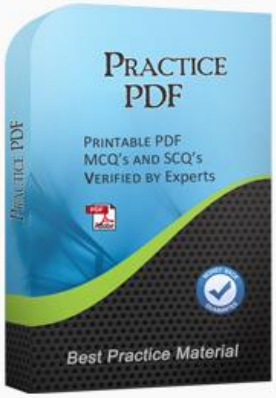
Pass4Training

Pass4Training

- Online Tool, Convenient, easy to study.
- Instant Online Access
- Supports All Web Browsers
- Practice Online Anytime
- Test History and Performance Review
- Supports Windows / Mac / Android / iOS, etc.

- Installable Software Application
- Simulates Real Exam Environment
- Builds Exam Confidence
- Supports MS Operating System
- Two Modes For Practice
- Practice Offline Anytime



- Printable PDF Format
- Prepared by IT Experts
- Instant Access to Download
- Study Anywhere, Anytime
- 365 Days Free Updates
- Free PDF Demo Available

Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.

365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.

Money Back Guarantee

Full refund if you fail the corresponding exam in 90 days after purchasing. And Free get any another product.

Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

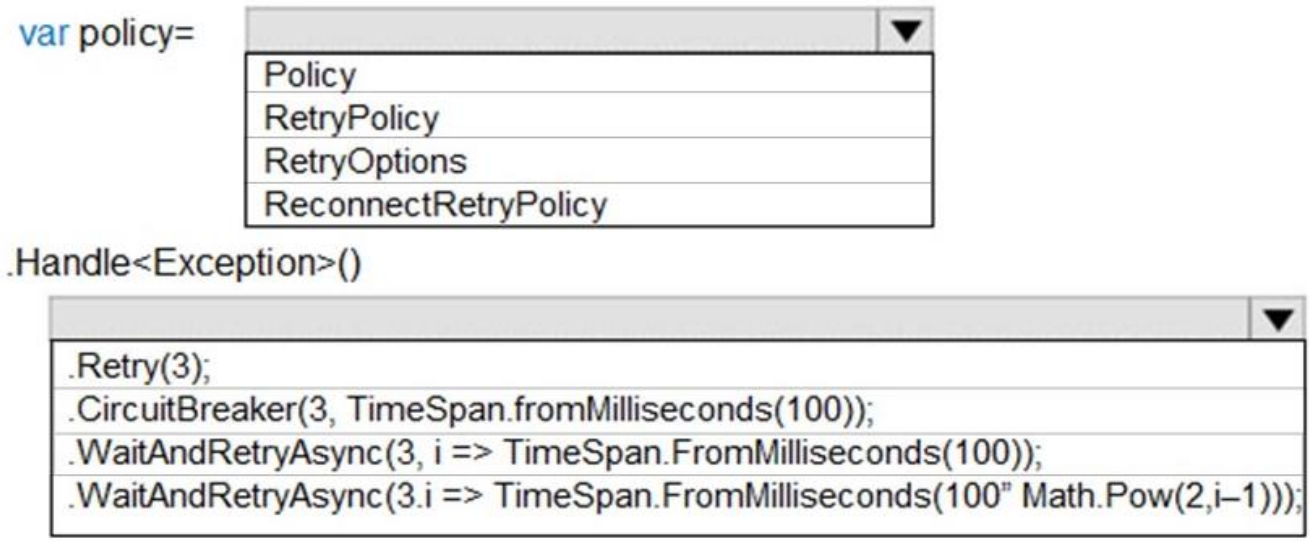
Exam : **AZ-202**

Title : Microsoft Azure Developer Certification Transition (beta)

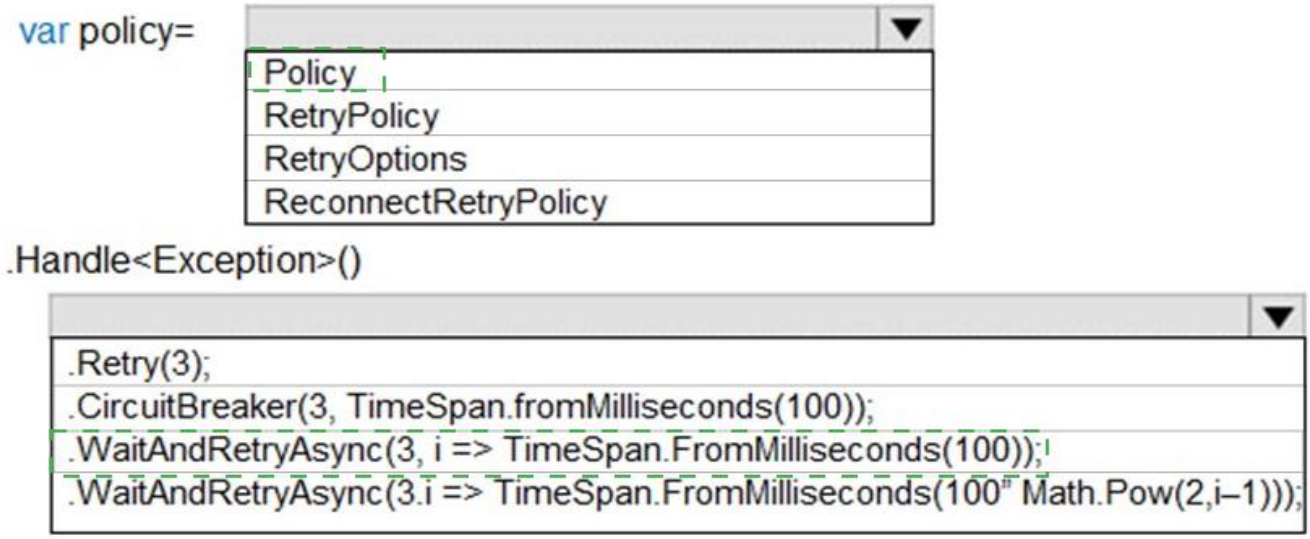
Vendor : Microsoft

Version : DEMO

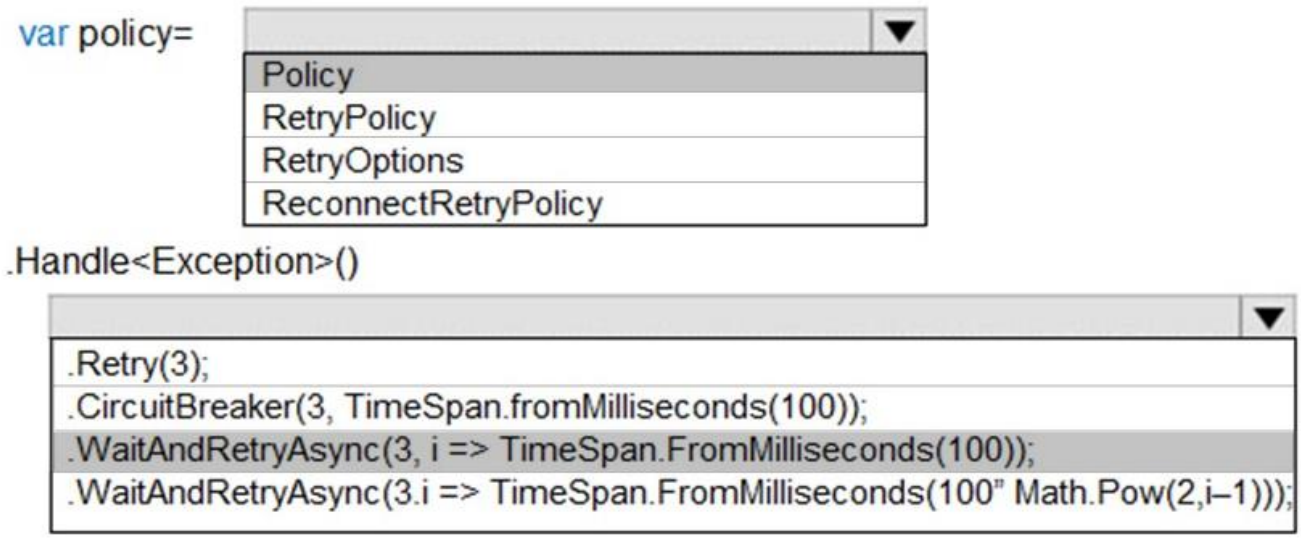
NO.1 HOTSPOT
You need to configure retries in the LoadUserDetails function in the Database class without impacting user experience.
What code should you insert on line DB07?
To answer, select the appropriate options in the answer area.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:



Box 1: Policy
RetryPolicy retry = Policy
Handle<HttpRequestException>()
Retry(3);
The above example will create a retry policy which will retry up to three times if an action fails with an exception handled by the Policy.
Box 2: WaitAndRetryAsync(3,i => TimeSpan.FromMilliseconds(100* Math.Pow(2,i-1))); A common retry strategy is exponential backoff: this allows for retries to be made initially quickly, but then at progressively longer intervals, to avoid hitting a subsystem with repeated frequent calls if the subsystem may be struggling.
Example:
Policy
Handle<SomeExceptionType>()
WaitAndRetry(3, retryAttempt =>
TimeSpan.FromSeconds(Math.Pow(2, retryAttempt))
);
References:
https://github.com/App-vNext/Polly/wiki/Retry

NO.2 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 need to meet the LabelMaker application security requirement.
Solution: Create a Microsoft Azure Active Directory service principal and assign it to the Azure Kubernetes Service (AKS) cluster.
Does the solution meet the goal?
A. Yes
B. No
Answer: A

NO.3 Note: This question is part of a series of questions that present the same solution. Each question in the series contains a unique solution. Determine whether the meets the stated goals.

You connect to Azure by using a workstation that has a slow internet connection. You have two Azure file shares. You plan to transfer a series of large files from one container to another container. The workstation does not have sufficient disk space to store the files.
You define the following variables in Azure PowerShell:

Variable	Description
$sourceServer	This variable represents the container that stores the files.
$destServer	This variable represents the container where files will be copied.
$sourceKey	This variable represents the primary key of the source storage account.
$destKey	This variable represents the primary key of the destination storage account

You need to simultaneously transfer the large files as efficiently as possible. Solution: Run the following Azure PowerShell command:

```
AzCopy /Source:$sourceServer /Dest:$DestServer /SourceKey:$sourceKey /DestKey:$destKey /S
```

Does the solution meet the goal?

A. No

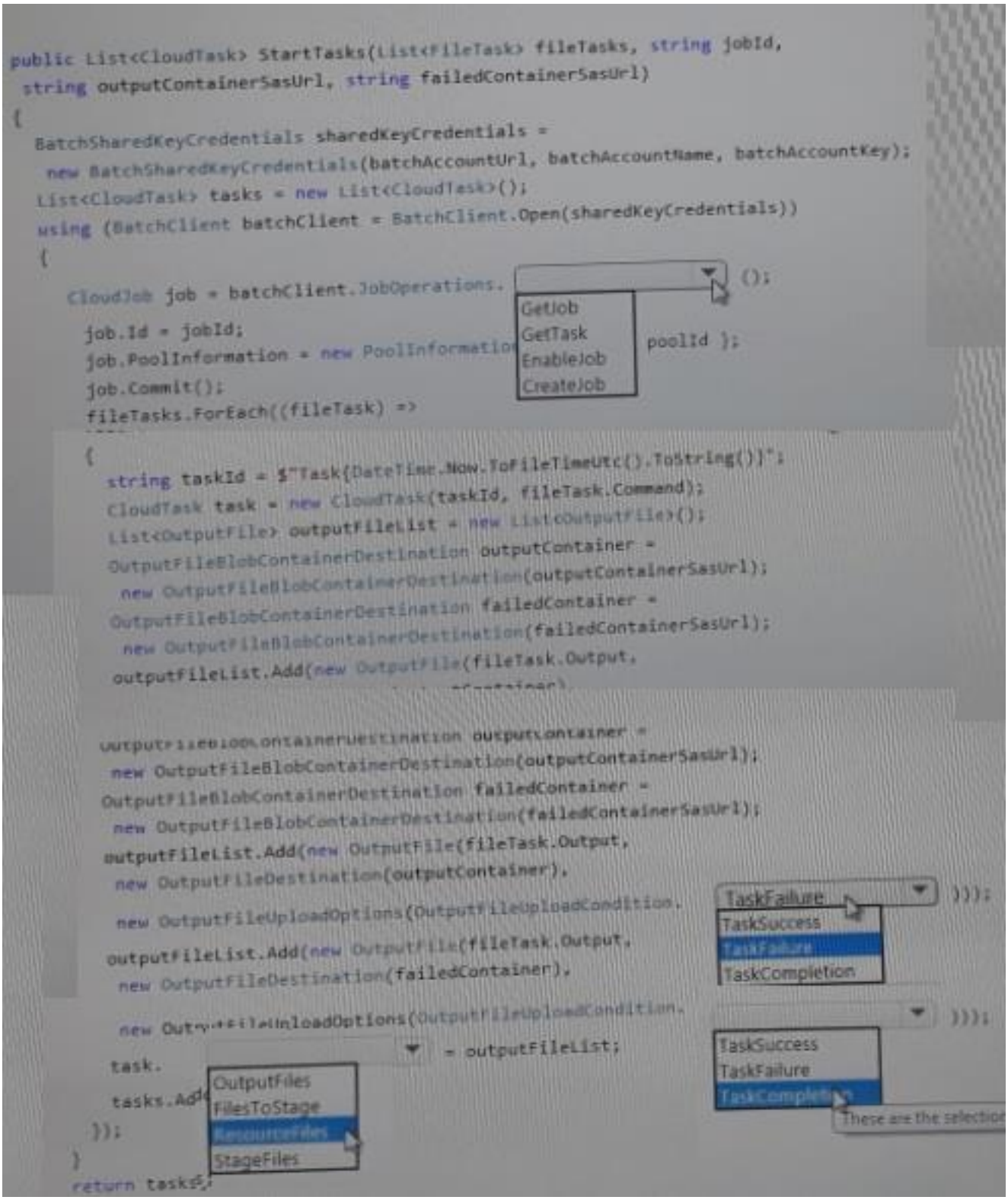
B. Yes

Answer: A

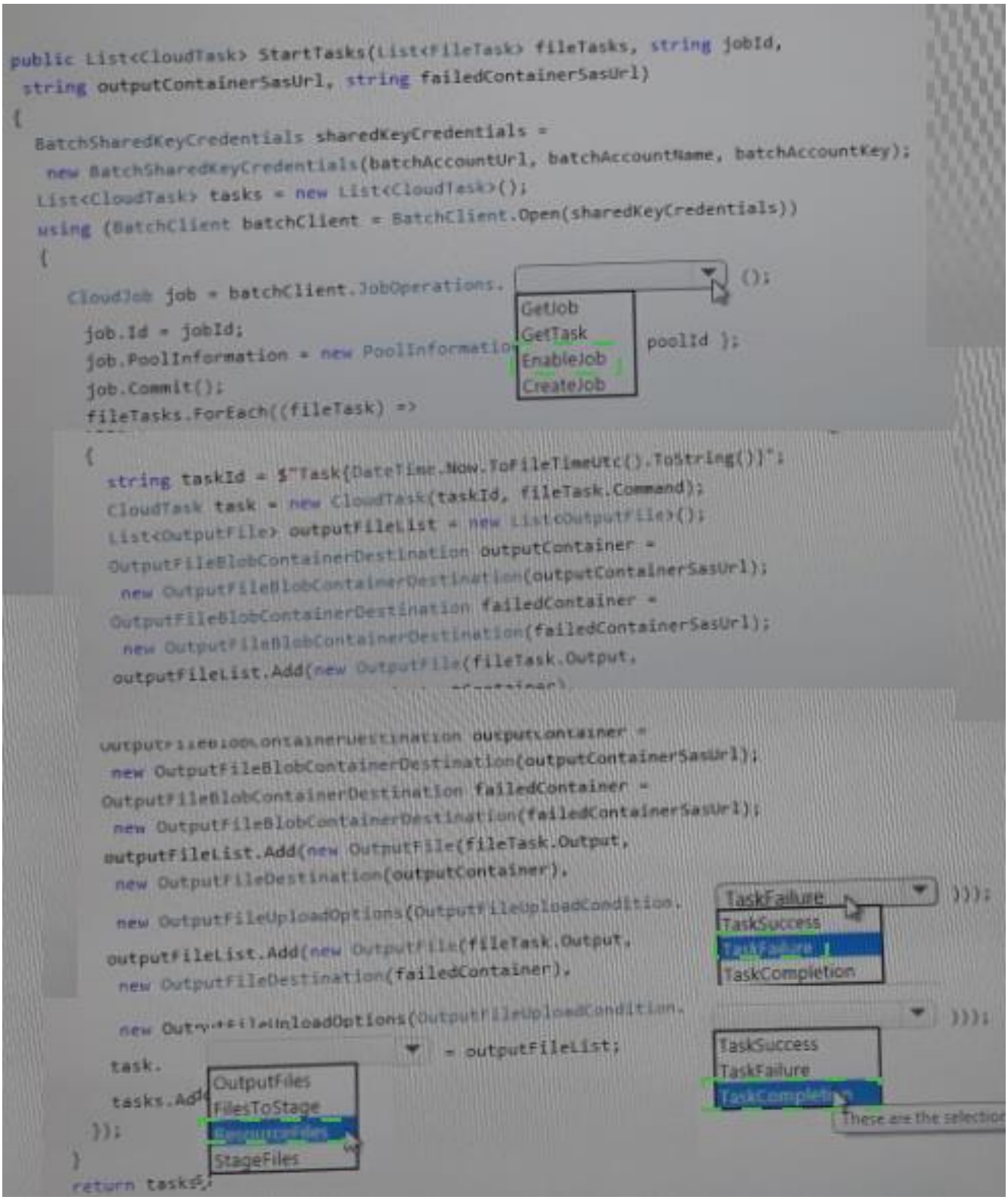
NO.4 HOTSPOT
You have an Azure Batch project that processes and converts files and stores the files in Azure storage. You are developing a function to start the batch job.
You add the following parameters to the function:

Parameter name	Description
fileTasks	a list of tasks to be run
jobId	the identifier that must be assigned to the job
outputContainerSasUrl	a storage SAS URL to store successfully converted files
failedContainerSasUrl	a storage SAS URL to store copies of files that failed to convert.

You must ensure that converted files are placed in the container referenced by the outputContainerSasUrl parameter. Files which fail to convert are placed in the container referenced by the failedContainerSasUrt parameter.
You need to ensure the files are correctly processed.
How should you complete the code segment? To answer, select the appropriate options in the answer area.


public List<CloudTask> StartTasks(List<FileTask> fileTasks, string jobId,
string outputContainerSasUrl, string failedContainerSasUrl)
BatchSharedKeyCredentials sharedKeyCredentials =
new BatchSharedKeyCredentials(batchAccountUrl, batchAccountName, batchAccountKey);
List<CloudTask> tasks = new List<CloudTask>();
using (BatchClient batchClient = BatchClient.Open(sharedKeyCredentials))
CloudJob job = batchClient.JobOperations.
();
GetJob
GetTask
EnableJob
CreateJob
poolId };
job.Id = jobId;
job.PoolInformation = new PoolInformation
job.Commit();
fileTasks.ForEach((fileTask) =>
string taskId = $"Task{DateTime.Now.ToFileTimeUtc().ToString()}";
CloudTask task = new CloudTask(taskId, fileTask.Command);
List<OutputFile> outputFileList = new List<OutputFile>();
OutputFileBlobContainerDestination outputContainer =
new OutputFileBlobContainerDestination(outputContainerSasUrl);
OutputFileBlobContainerDestination failedContainer =
new OutputFileBlobContainerDestination(failedContainerSasUrl);
outputFileList.Add(new OutputFile(fileTask.Output,
new OutputFileDestination(outputContainer),
new OutputFileUploadOptions(OutputFileUploadCondition.
TaskFailure
TaskSuccess
TaskFailure
TaskCompletion
outputFileList.Add(new OutputFile(fileTask.Output,
new OutputFileDestination(failedContainer),
TaskSuccess
TaskFailure
TaskCompletion
= outputFileList;
task.
tasks.Add
OutputFiles
FilesToStage
ResourceFiles
StageFiles
These are the selection
return tasks;

Answer:

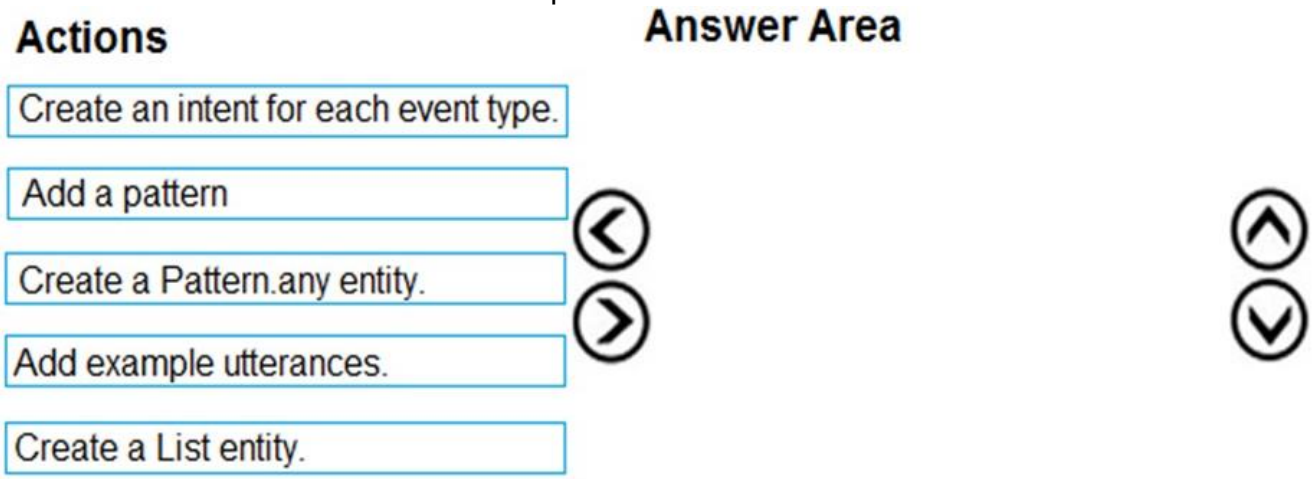


Explanation: EnableJob
TaskFailure
Taskcompletion
ResourceFiles

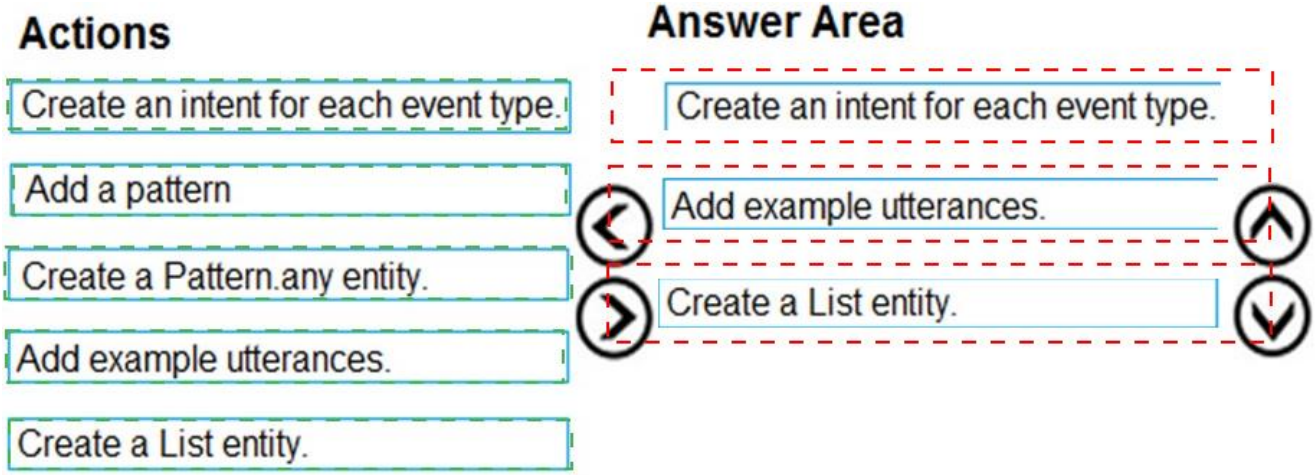
NO.5 DRAG DROP

You develop a bot by using Language Understanding Intelligence Service (LUIS) and the .NET Bot framework. You use LUIS in the Azure portal to optimize the bot.

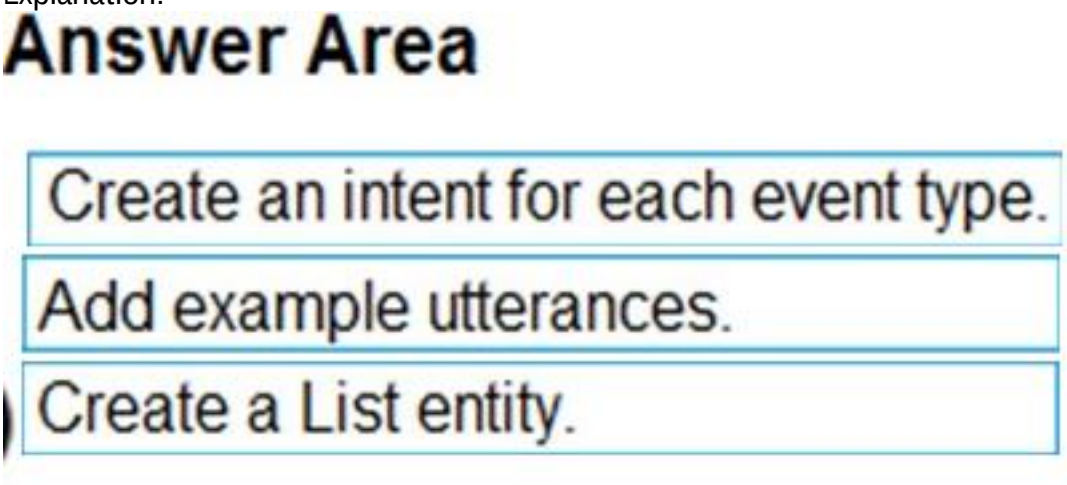
You review the utterances and determine that users are requesting time and venue information for events.
You need to improve the prediction efficiency of the bot.
Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.
NOTE: Each correct selection is worth one point.



Answer:



Explanation:



Step 1: Create an intent for each event type

Identify your intents
Step 2: Add example utterances
Create example utterances for each intent
Step 3: Create a List Entity
Identify your entities
A list entity is an explicitly specified list of values. Each value consists of one or more synonyms. In a travel app, you might choose to create a list entity to represent airport names.
References:
https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-plan-your-app

NO.6 You need to access user claims in the e-commerce web app* What should you do first?
A. Using the Azure CU enable Cross-origin resource sharing (CORS) from the e-commerce checkout API to the e-commerce web app
B. Assign the Contributor RBAC role to the e-commerce web app by using the Resource Manager create role assignment API.
C. Update the e-commerce web app to read the HTTP request header values.
D. Write custom code to make a Microsoft Graph API call from the e-commerce web app.
Answer: D

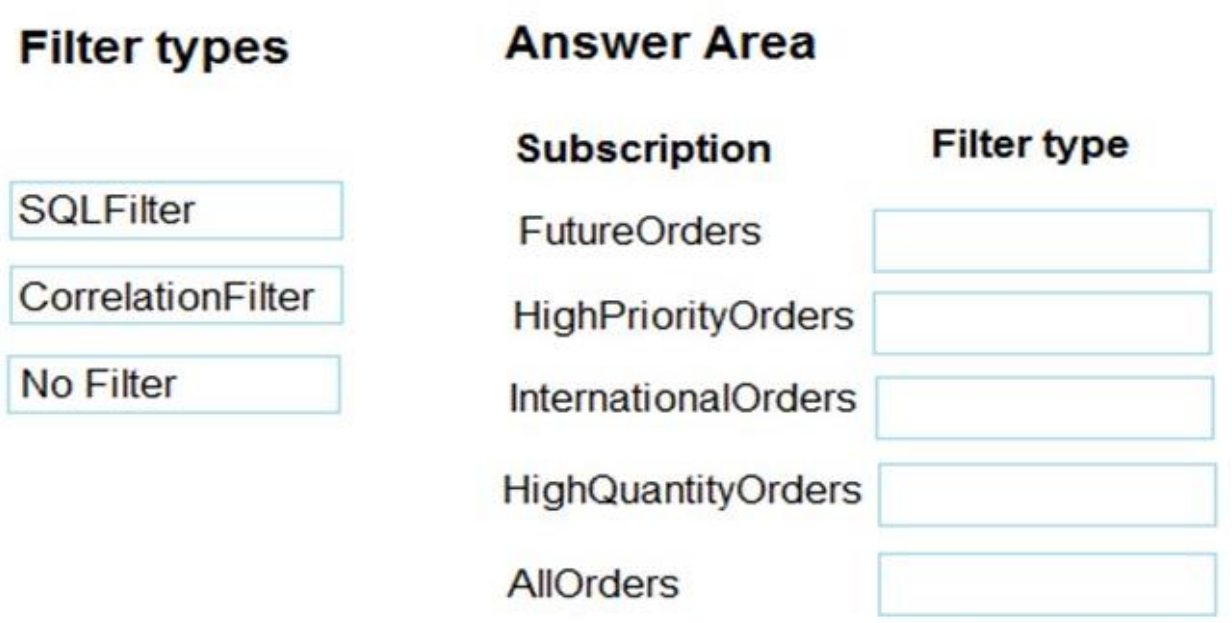
NO.7 DRAG DROP
You are implementing an order processing system. A point of sale application publishes orders to topics in an Azure Service Bus queue. The label property for the topic includes the following data:

Property	Description
ShipLocation	the country/region where the order will be shipped
CorrelationId	a priority value for the order
Quantity	a user-defined field that stores the quantity of items in an order
AuditedAt	a user-defined field that records the date an order is audited

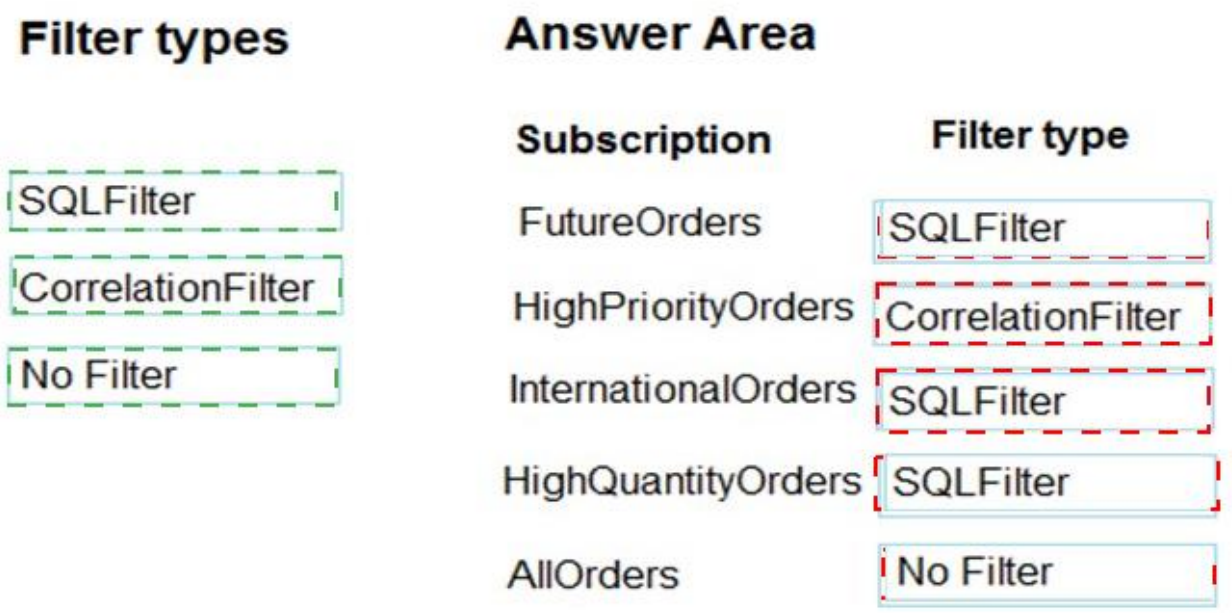
The system has the following requirements for subscriptions:

Subscription type	Comments
FutureOrders	This subscription is reserved for future use and must not receive any orders.
HighPriorityOrders	Handle all high priority orders and International orders.
InternationalOrders	Handle orders where the country/region is not United States.
HighQuantityOrders	Handle only orders with quantities greater than 100 units.
AllOrders	This subscription is used for auditing purposes. This subscription must receive every single order. AllOrders has an Action defined that updates the AuditedAt property to include the date and time it was received by the subscription.

You need to implement filtering and maximize throughput while evaluating filters.
Which filter types should you implement? To answer, drag the appropriate filter types to the correct subscriptions. Each filter 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.

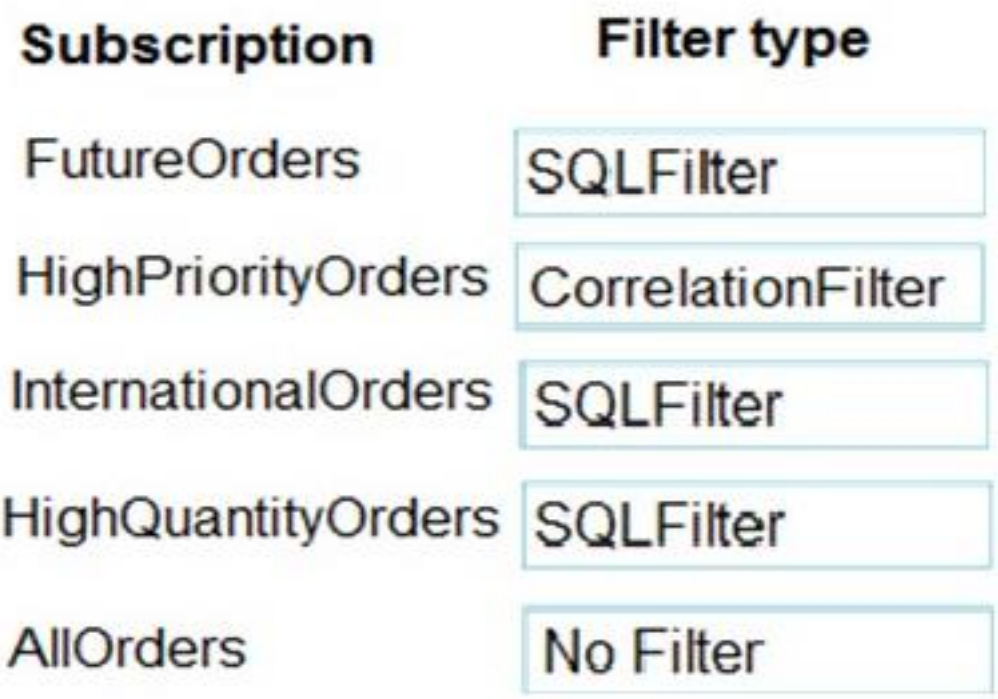


Answer:



Explanation:

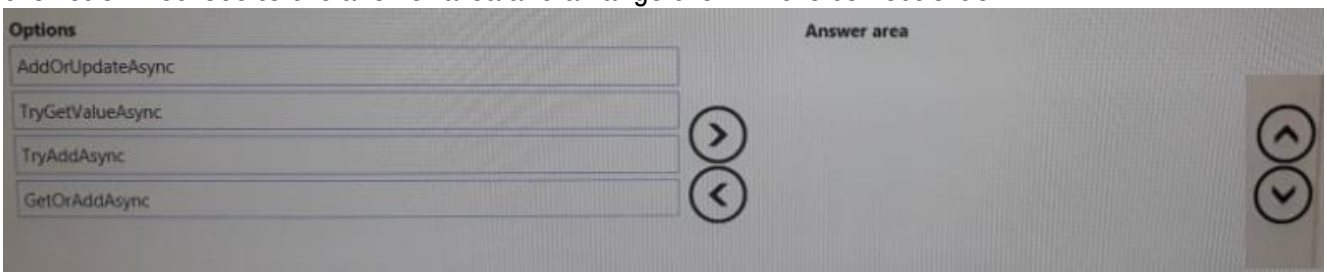
Answer Area



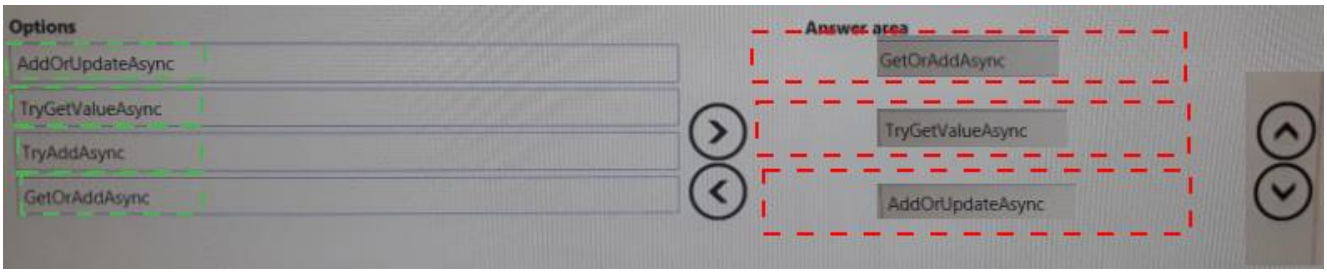
FutureOrders: SQLFilter
HighPriortyOrders: CorrelationFilter
CorrelationID only
InternationalOrders: SQLFilter
Country NOT USA requires an SQL Filter
HighQuantityOrders: SQLFilter
Need to use relational operators so an SQL Filter is needed.
AllOrders: No Filter
SQL Filter: SQL Filters - A SqlFilter holds a SQL-like conditional expression that is evaluated in the broker against the arriving messages' user-defined properties and system properties. All system properties must be prefixed with sys. in the conditional expression.
The SQL-language subset for filter conditions tests for the existence of properties (EXISTS), as well as for null-values (IS NULL), logical NOT/AND/OR, relational operators, simple numeric arithmetic, and simple text pattern matching with LIKE.
Correlation Filters - A CorrelationFilter holds a set of conditions that are matched against one or more of an arriving message's user and system properties. A common use is to match against the CorrelationId property, but the application can also choose to match against ContentType, Label, MessageId, ReplyTo, ReplyToSessionId, SessionId, To, and any user-defined properties. A match exists when an arriving message's value for a property is equal to the value specified in the correlation filter. For string expressions, the comparison is case-sensitive. When specifying multiple match properties, the filter combines them as a logical AND condition, meaning for the filter to match, all conditions must match.
Boolean filters - The TrueFilter and FalseFilter either cause all arriving messages (true) or none of the arriving messages (false) to be selected for the subscription.

References:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/topic-filters

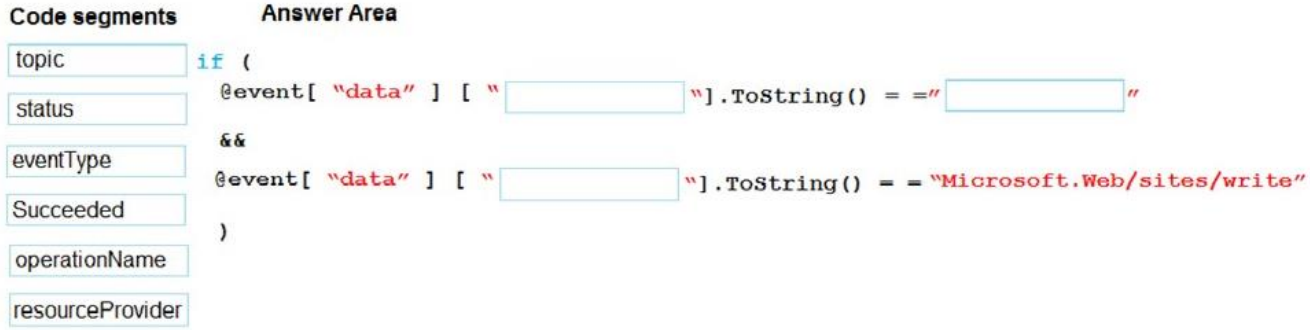
NO.8 DRAG DROP
You are developing a stateful service to deploy to Azure Service Fabric. You plan to implement the RunAsync method.
You need to implement the methods to interface with an instance of the IReliable dictionary interface to increment a count each time the service is called. The first time the service is called, you must initialize the count to 1 if it does not yet exist and then update it by one each time it is called.
Which three methods should you run in sequence. To answer, move the appropriate methods from the list of methods to the answer area and arrange them in the correct order.



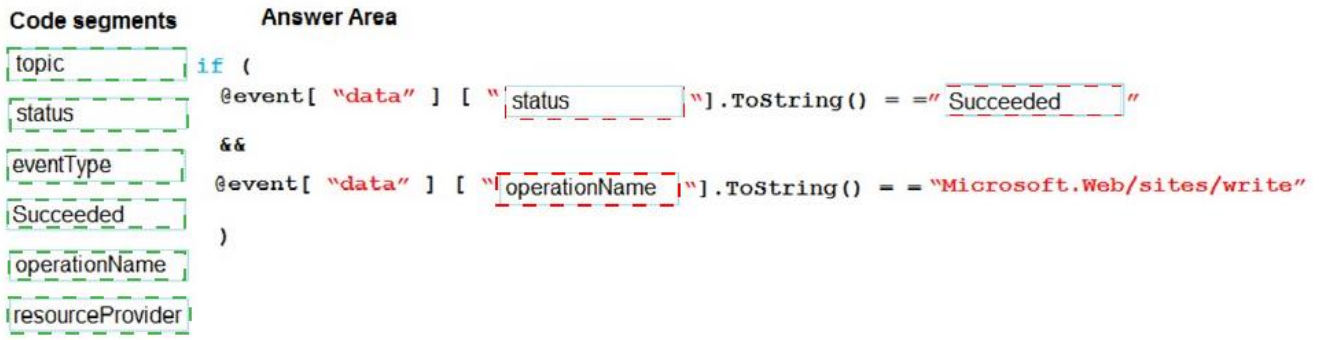
Answer:



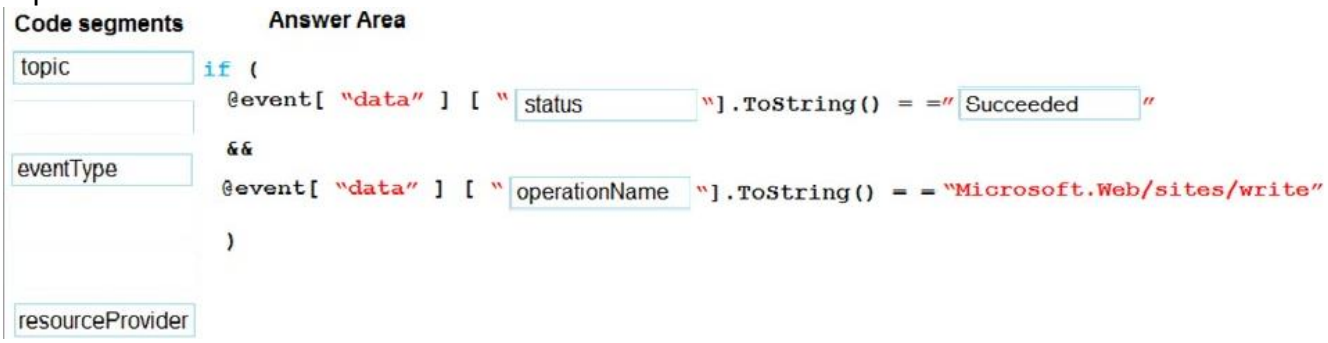
NO.9 DRAG DROP
You need to add code at line EG15 in EventGridController.cs to ensure that the Log policy applies to all services.
How should you complete the code? To answer, drag the appropriate code segments to the correct locations. Each code segment may be used once, more than once, or not at all.
You may need to drag the split bar between panes or scroll to view content.
NOTE: Each correct selection is worth one point.



Answer:



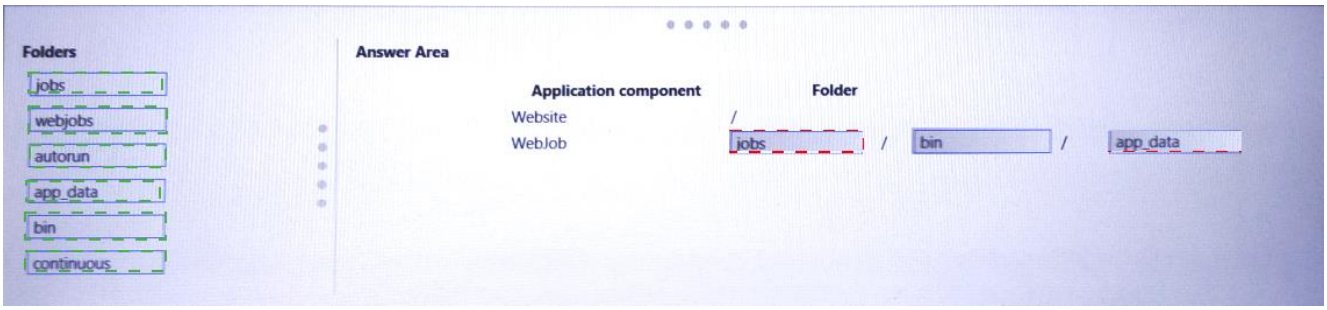
Explanation:



Box 1: Status
Box 2: Succeded
Box 3: operationName
Scenario: Policy service
You develop and deploy a stateful ASP.NET Core 2.1 web application named Policy service to an Azure App Service Web App. The application reacts to events from Azure Event Grid and performs policy actions based on those events.
The application must include the Event Grid Event ID field in all Application Insights telemetry.

NO.10 DRAG DROP
You are developing an application that consists of an ASP.NET Core Web API website and a WebJob that starts automatically and runs continuously. You are building the deployment process for the application.
You need to ensure that both the website and the WebJob are deployed.
How should you structure the deployment folders? To answer, drag the appropriate path segments to the correct locations. Each path 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.

Answer:


Folders
jobs
webjobs
autorun
app_data
bin
continuous
Answer Area
Application component
Folder
Website
/
WebJob
jobs
/
bin
/
app_data