Answer Area

```
public list<CloudTasks> 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 = batchClient.JobOperations. [GetJob / GetTask / EnableJob / CreateJob] ();
        job.Id = jobId,
        job.PoolInfromation = new PoolInformation { PoolId = poolId };
        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. [TaskSuccess / TaskFailure / TaskCompletion] )));
   outputFileList.Add(new OutputFile(fileTask.Output,
    new OutputFileDestination(failedContainer),
    new OutputFileUploadOptions(OutputFileUploadCondition, [TaskSuccess / TaskFailure / TaskCompletion] )));
    task [OutputFiles / FilesToStage / ResourceFiles / StageFiles] =outputFileList;
    task.Add(task);
    });
    ]
    return tasks,
}
```

Correct Answer:

Answer Area

```
public list<CloudTasks> 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 = batchClient.JobOperations. [GetJob / GetTask / EnableJob / CreateJob] ();
        job.Id = jobId,
        job.PoolInfromation = new PoolInformation { PoolId = poolId };
        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. [TaskSuccess / TaskFailure / TaskCompletion] )));
   outputFileList.Add(new OutputFile(fileTask.Output,
    new OutputFileDestination(failedContainer),
    new OutputFileUploadOptions(OutputFileUploadCondition, [TaskSuccess / TaskFailure / TaskCompletion] )));
    task [OutputFiles / FilesToStage / ResourceFiles / StageFiles] =outputFileList;
    task.Add(task);
    });
    ]
    return tasks,
}
```

QUESTION 20
HOTSPOT
You are developing a back-end Azure App Service that scales based on the number of messages contained in a Service Bus queue.

A rule already exists to scale up the App Service when the average queue length of unprocessed and valid queue messages is greater than 1000.

You need to add a new rule that will continuously scale down the App Service as long as the scale up condition is not met.

How should you configure the Scale rule? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area


Scale rule
Metric source
Storage queue
Service Bus queue
Current resource
Storage queue (classic)
Resource type
Service Bus Namespaces
Resource
MessageQueue1103
* Queues
itemqueue
Criteria
* Metric name
Message Count
Active Message Count
1 minute time grain
* Time grain statistic
Total
Maximum
Average
Count
Greater than
Greater than or equal to
Less than
Less than or equal to
* Threshold
1000
Action
* Operation
Increase count by
Increase count to
Decrease count by
Decrease count to
* Instance count
1
* Cool down (minutes)
5

Correct Answer: