KIRK DAHL

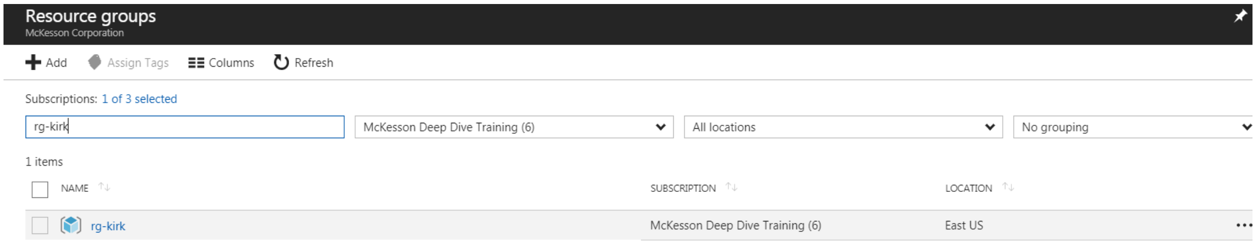
McKesson **Assignment 08** Deep Azure

Handed out: 11/29/2017 Due by 11:59 PM, midnight (CST) on Tuesday, 12/05/2017

You can do this assignment in Java, C# or Python. You do not have to do both problems in the same language. Code samples were collected from various Microsoft repositories.

Problem 01. Start with Java class file `QueueBasics.java`, or with C# file `GettingStarted.cs` or with Python program `Queues.py`. Those files are contained in provided archives `storage-queue-java-getting-started.zip, storage-queue-cshrap-getting-started.zip and storage-queue-python-getting-started.zip`, respectfully. Make any one of those programs work and perform the following operations: create a storage queue; publish a series of 5 short messages to that queue; dequeue two of those messages and print their content; peak into the content of one of remaining messages in the queue, update the content of that messages, and make that updated message invisible for 5 seconds. Try to dequeuer that message right away. Report error or system message received. Then, wait 6 seconds, dequeue the message and print its content. Leave one message in the queue and do nto delete the queue right away. Convince yourself that the queue and your stored message can be seen in Azure Portal.
(100$)

Created my resource group and storage account



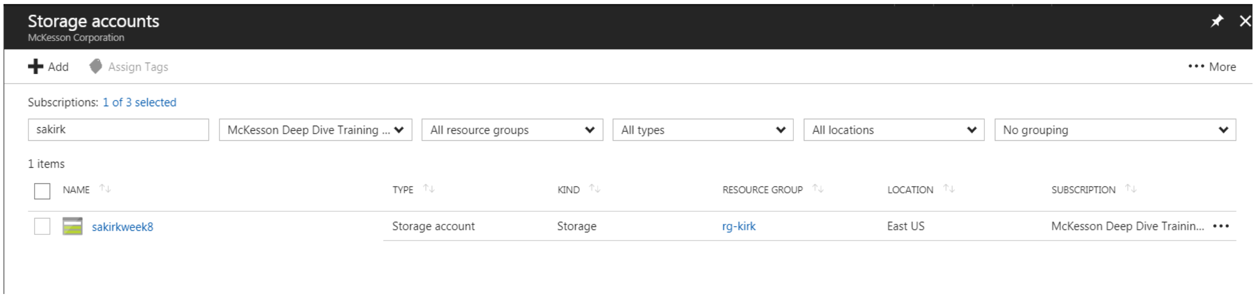
Resource groups
McKesson Corporation

Subscriptions: 1 of 3 selected

rg-kirk | McKesson Deep Dive Training (6) | All locations | No grouping

1 items

NAME	SUBSCRIPTION	LOCATION
rg-kirk	McKesson Deep Dive Training (6)	East US



Storage accounts
McKesson Corporation

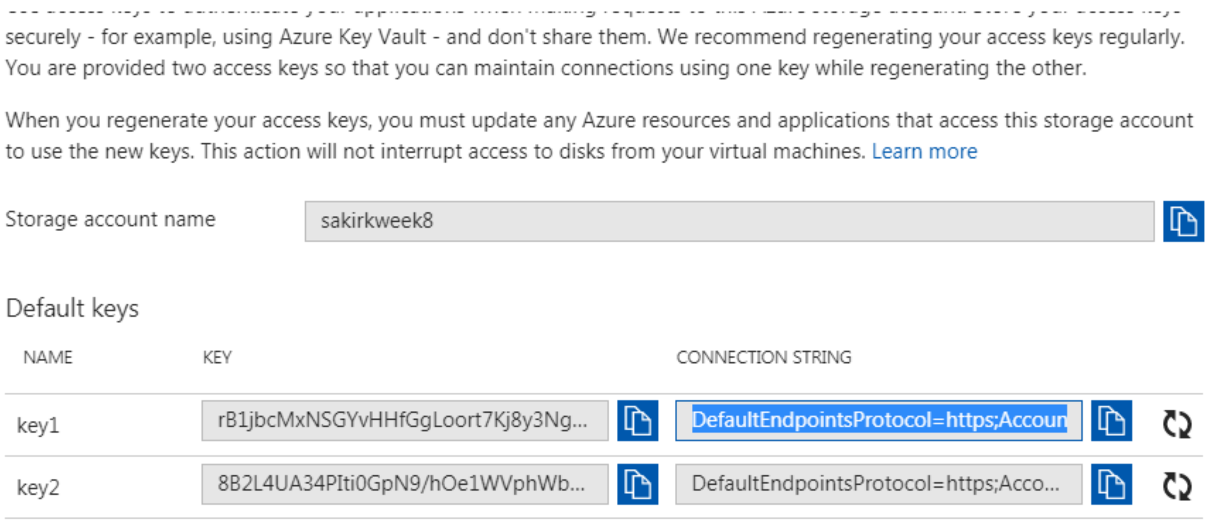
Subscriptions: 1 of 3 selected

sakirk | McKesson Deep Dive Training ... | All resource groups | All types | All locations | No grouping

1 items

NAME	TYPE	KIND	RESOURCE GROUP	LOCATION	SUBSCRIPTION
sakirkweek8	Storage account	Storage	rg-kirk	East US	McKesson Deep Dive Trainin...

Gather the Connection string that I will use in the app.config



securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

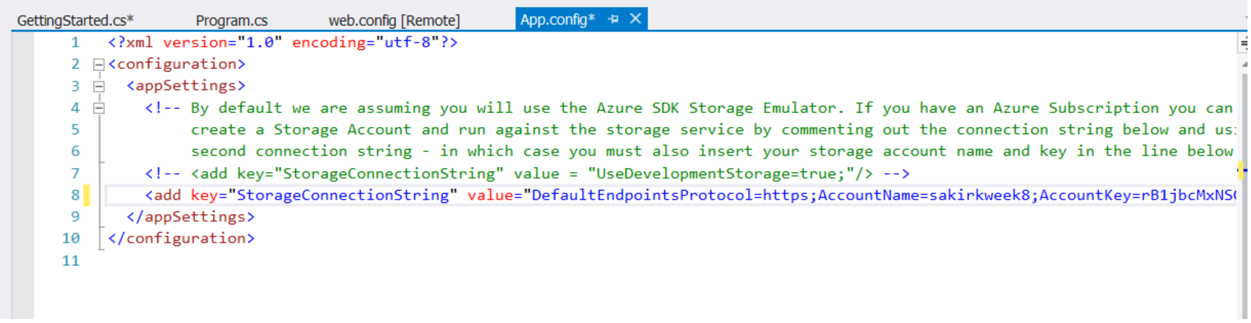
When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. Learn more

Storage account name: sakirkweek8

Default keys

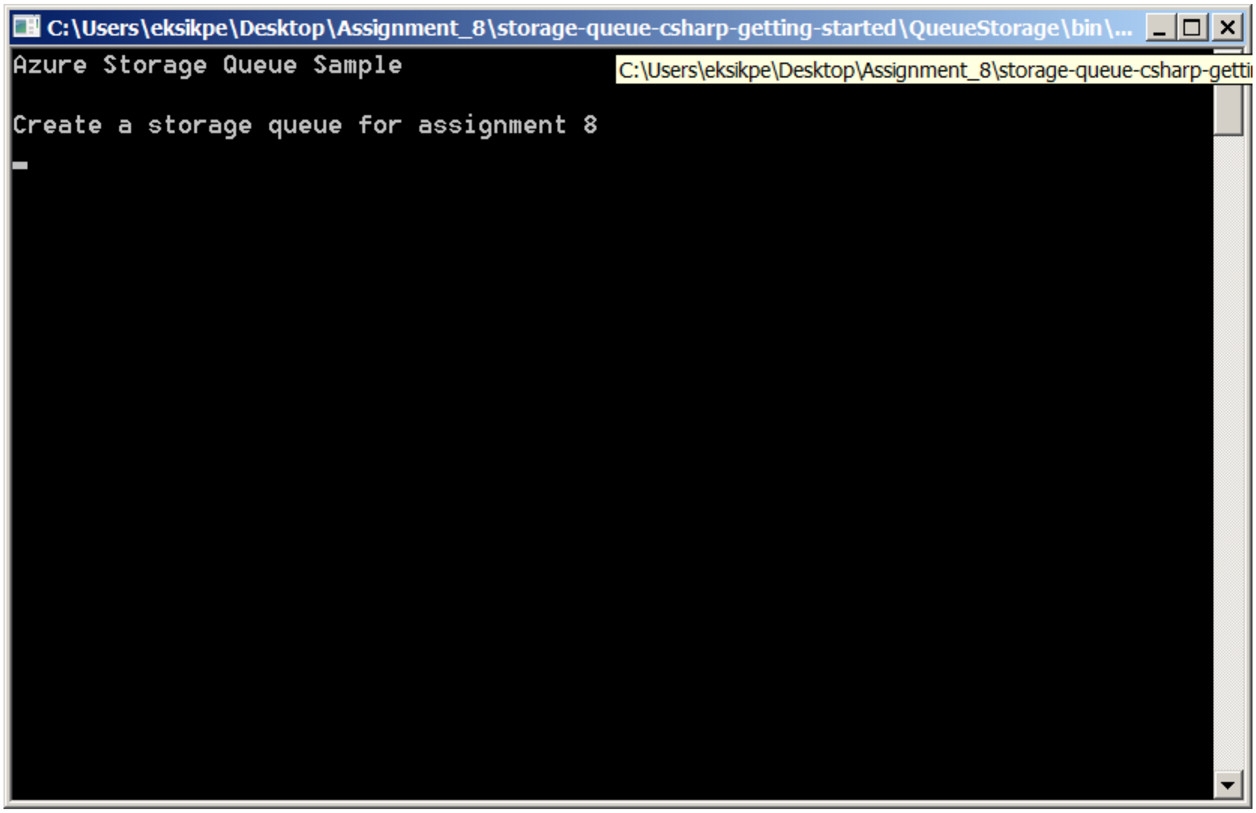
NAME	KEY	CONNECTION STRING
key1	rB1jbcMxNSGYvHHfGgLoort7Kj8y3Ng...	DefaultEndpointsProtocol=https;Accoun
key2	8B2L4UA34PIti0GpN9/hOe1WVphWb...	DefaultEndpointsProtocol=https;Acco...

Update the app.config for my storage account connection string



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <!-- By default we are assuming you will use the Azure SDK Storage Emulator. If you have an Azure Subscription you can
         create a Storage Account and run against the storage service by commenting out the connection string below and us
         second connection string - in which case you must also insert your storage account name and key in the line below
    <!-- <add key="StorageConnectionString" value = "UseDevelopmentStorage=true;"/> -->
    <add key="StorageConnectionString" value="DefaultEndpointsProtocol=https;AccountName=sakirkweek8;AccountKey=rB1jbcMxNS
  </appSettings>
</configuration>
```

Modified the `Gettingstarted.cs` to add some break points to illustrate the message queue.

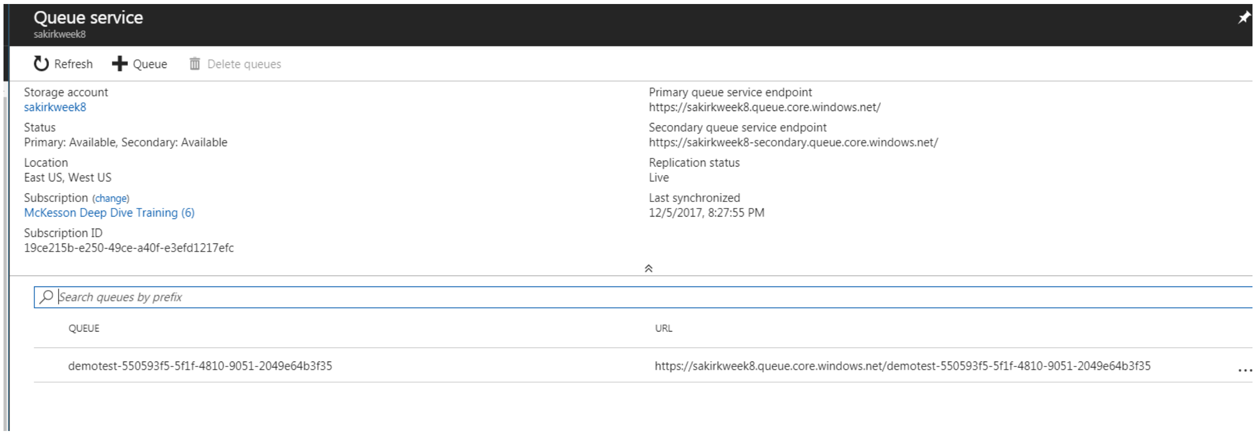


C:\Users\eksikpe\Desktop\Assignment_8\storage-queue-csharp-getting-started\QueueStorage\bin\...

C:\Users\eksikpe\Desktop\Assignment_8\storage-queue-csharp-getti

```
Azure Storage Queue Sample

Create a storage queue for assignment 8
```



Storage account
sakirkweek8

Status
Primary: Available, Secondary: Available

Location
East US, West US

Subscription (change)
McKesson Deep Dive Training (6)

Subscription ID
19ce215b-e250-49ce-a40f-e3efd1217efc

Primary queue service endpoint
https://sakirkweek8.queue.core.windows.net/

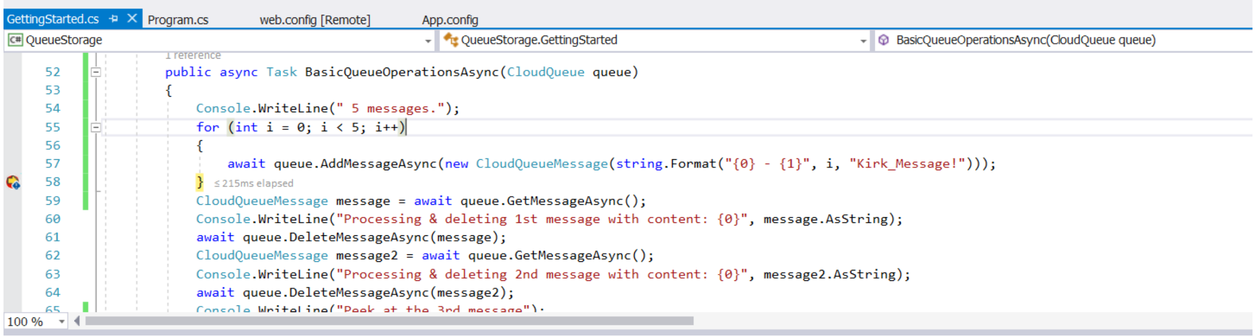
Secondary queue service endpoint
https://sakirkweek8-secondary.queue.core.windows.net/

Replication status
Live

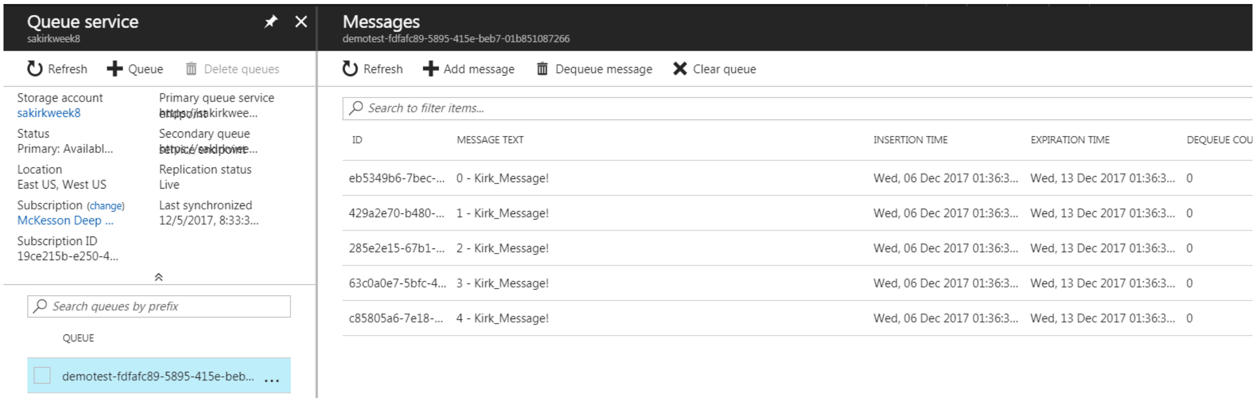
Last synchronized
12/5/2017, 8:27:55 PM

QUEUE	URL
demotest-550593f5-5f1f-4810-9051-2049e64b3f35	https://sakirkweek8.queue.core.windows.net/demotest-550593f5-5f1f-4810-9051-2049e64b3f35

Continue through the break points until it had created my 5 messages



```
public async Task BasicQueueOperationsAsync(CloudQueue queue)
{
    Console.WriteLine(" 5 messages.");
    for (int i = 0; i < 5; i++)
    {
        await queue.AddMessageAsync(new CloudQueueMessage(string.Format("{0} - {1}", i, "Kirk_Message!")));
    } ≤215ms elapsed
    CloudQueueMessage message = await queue.GetMessageAsync();
    Console.WriteLine("Processing & deleting 1st message with content: {0}", message.AsString);
    await queue.DeleteMessageAsync(message);
    CloudQueueMessage message2 = await queue.GetMessageAsync();
    Console.WriteLine("Processing & deleting 2nd message with content: {0}", message2.AsString);
    await queue.DeleteMessageAsync(message2);
    Console.WriteLine("Peek at the 3rd message");
```



Messages
demotest-fdfafc89-5895-415e-beb7-01b851087266

Storage account
sakirkweek8

Primary queue service endpoint
https://sakirkwee...

Status
Primary: Availabl...

Secondary queue service endpoint
https://sakirkwee...

Location
East US, West US

Replication status
Live

Subscription (change)
McKesson Deep ...

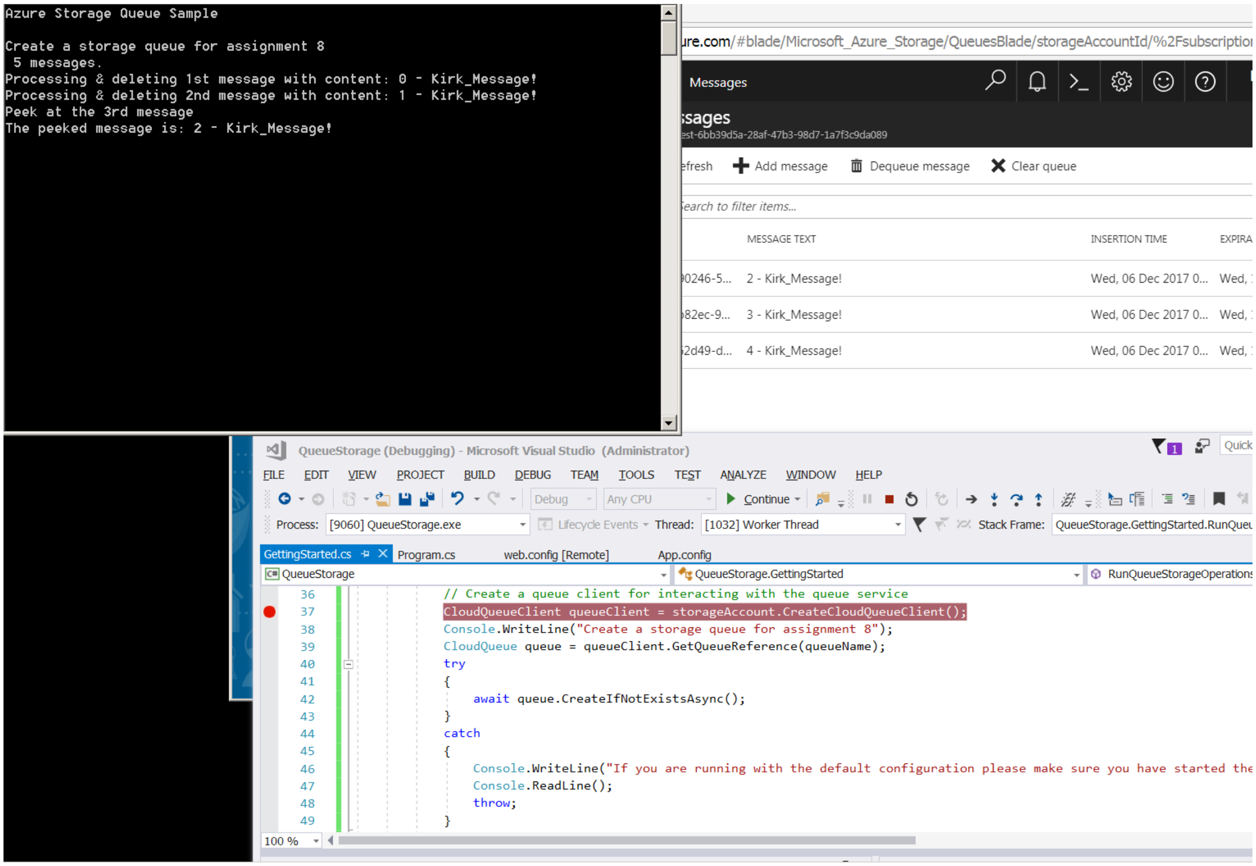
Last synchronized
12/5/2017, 8:33:3...

Subscription ID
19ce215b-e250-4...

QUEUE
demotest-fdfafc89-5895-415e-beb...

ID	MESSAGE TEXT	INSERTION TIME	EXPIRATION TIME	DEQUEUE COU
eb5349b6-7bec-...	0 - Kirk_Message!	Wed, 06 Dec 2017 01:36:3...	Wed, 13 Dec 2017 01:36:3...	0
429a2e70-b480-...	1 - Kirk_Message!	Wed, 06 Dec 2017 01:36:3...	Wed, 13 Dec 2017 01:36:3...	0
285e2e15-67b1-...	2 - Kirk_Message!	Wed, 06 Dec 2017 01:36:3...	Wed, 13 Dec 2017 01:36:3...	0
63c0a0e7-5bfc-4...	3 - Kirk_Message!	Wed, 06 Dec 2017 01:36:3...	Wed, 13 Dec 2017 01:36:3...	0
c85805a6-7e18-...	4 - Kirk_Message!	Wed, 06 Dec 2017 01:36:3...	Wed, 13 Dec 2017 01:36:3...	0

Continued through code and it deleted 2 messages and peeked at the 3rd message



```
Azure Storage Queue Sample

Create a storage queue for assignment 8
 5 messages.
Processing & deleting 1st message with content: 0 - Kirk_Message!
Processing & deleting 2nd message with content: 1 - Kirk_Message!
Peek at the 3rd message
The peeked message is: 2 - Kirk_Message!
```

ure.com/#blade/Microsoft_Azure_Storage/QueuesBlade/storageAccountId/%2Fsubscriptio

ssages
est-6bb39d5a-28af-47b3-98d7-1a7f3c9da089

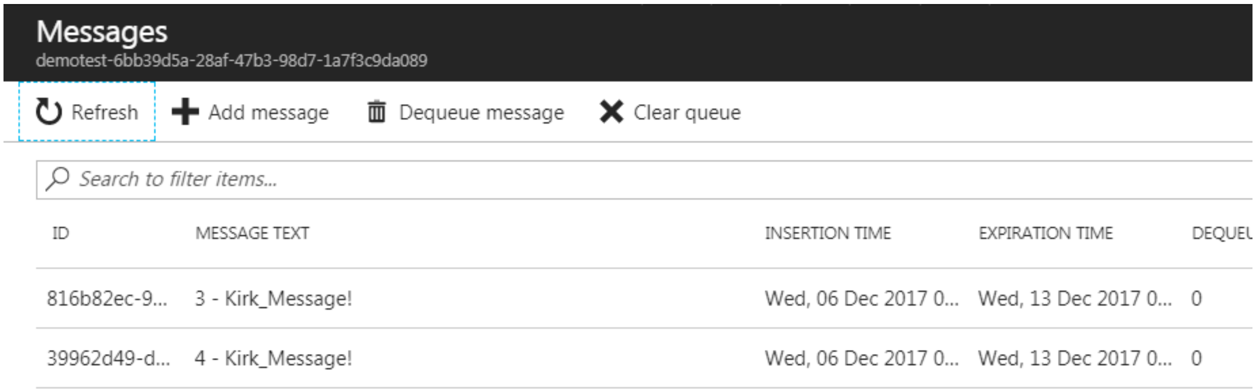
	MESSAGE TEXT	INSERTION TIME	EXPIRA
90246-5...	2 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed,
82ec-9...	3 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed,
2d49-d...	4 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed,

QueueStorage (Debugging) - Microsoft Visual Studio (Administrator)

Process: [9060] QueueStorage.exe Thread: [1032] Worker Thread Stack Frame: QueueStorage.GettingStarted.RunQueu

```
            // Create a queue client for interacting with the queue service
            CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();
            Console.WriteLine("Create a storage queue for assignment 8");
            CloudQueue queue = queueClient.GetQueueReference(queueName);
            try
            {
                await queue.CreateIfNotExistsAsync();
            }
            catch
            {
                Console.WriteLine("If you are running with the default configuration please make sure you have started the
                Console.ReadLine();
                throw;
            }
```

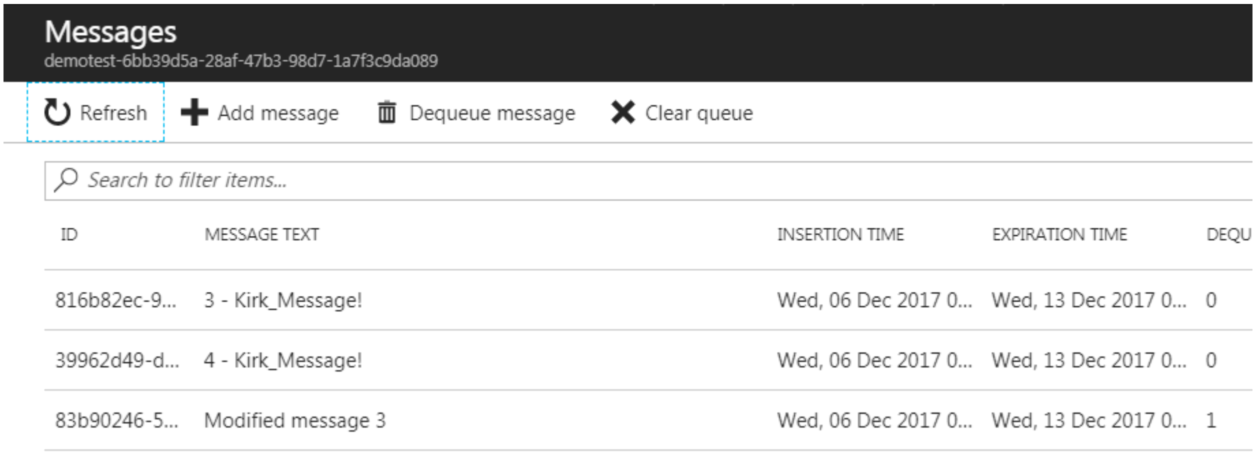
We don't see the modified message yet as we wait.



Messages
demotest-6bb39d5a-28af-47b3-98d7-1a7f3c9da089

ID	MESSAGE TEXT	INSERTION TIME	EXPIRATION TIME	DEQUEU
816b82ec-9...	3 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed, 13 Dec 2017 0...	0
39962d49-d...	4 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed, 13 Dec 2017 0...	0

And now we see the modified message


Messages
demotest-6bb39d5a-28af-47b3-98d7-1a7f3c9da089
Refresh
Add message
Dequeue message
Clear queue
Search to filter items...

ID	MESSAGE TEXT	INSERTION TIME	EXPIRATION TIME	DEQU
816b82ec-9...	3 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed, 13 Dec 2017 0...	0
39962d49-d...	4 - Kirk_Message!	Wed, 06 Dec 2017 0...	Wed, 13 Dec 2017 0...	0
83b90246-5...	Modified message 3	Wed, 06 Dec 2017 0...	Wed, 13 Dec 2017 0...	1

Problem 02 Optional. In this second problem write/modify code so that it performs al of operations you performed in Problem 01. If you are a Java developer, start withclass file `ServiceBusQueueBasic.java` which you could find in the attached archive: `service-bus-java-manage-queue-with-basic-features-master.zip`. This archive was downloaded from https://azure.microsoft.com/en-us/resources/samples/service-bus-java-manage-queue-with-basic-features/.
If you find it easier to work with C#, please examine C Sharp class file `Program.cs` which you could find in the attached archive: `service-bus-dotnet-manage-queue-with-basic-features-master.zip`. This archive was downloaded from https://azure.microsoft.com/en-us/resources/samples/service-bus-dotnet-manage-queue-with-basic-features/
Both referenced sites have very useful practical information.
If you want to do this problem in Python, please visit: https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-python-how-to-use-queues
(100%)

Did not perform

SUBMISSION INSTRUCTIONS:

Your main submission should be a MS Word or PDF document containing descriptions of your action while configuring Azure services. If your MS Word document is larger than 1 MB, save it as a MINIMIZED PDF. Please be merciful and capture small JPGs. Describe the purpose of every action and the significance of the results. Start with the text of this homework assignment as the template. Please add the entire text of your JAVA, C# or Python programs to the end of your MS Word/PDF document. Please write your solution as if you are writing a tutorial for your colleagues. Please make your text readable. Make sure that your fonts, especially in captured images are not unreadable. Please do not provide ZIP or RAR or any other archives. Canvas cannot open those archives and they turn into a nuisance for us.