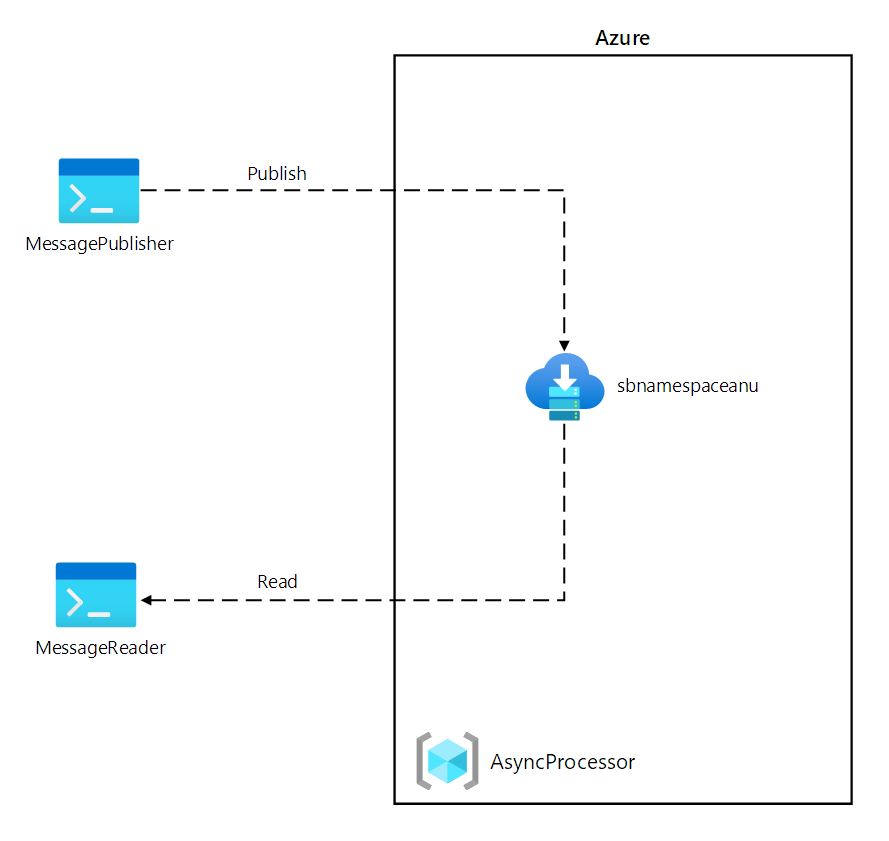
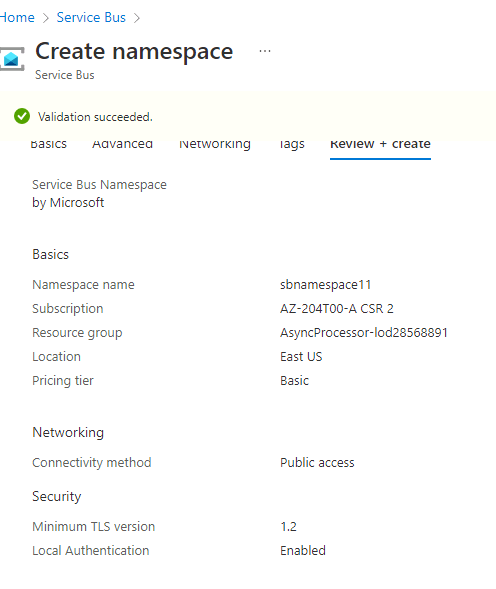
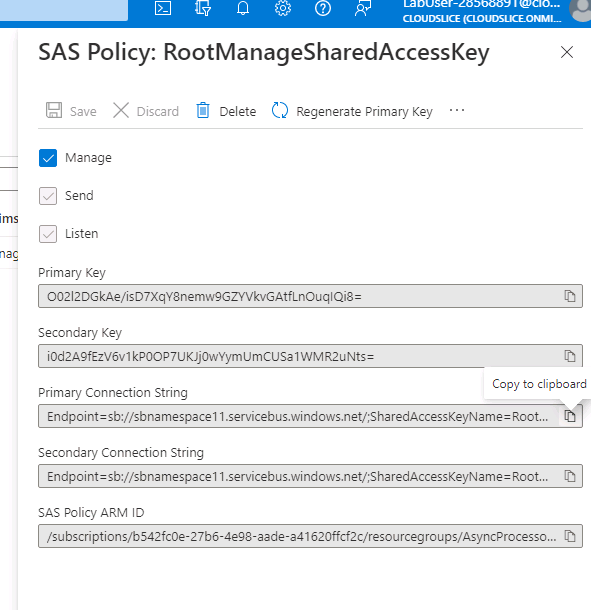
Develop message-based solutions

Lab 10: Asynchronously process messages by using Azure Service Bus Queues



### **Create Azure resources**



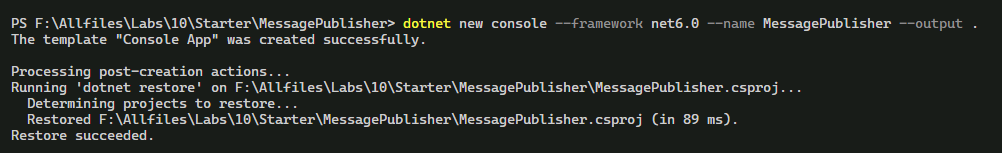


Endpoint=sb://sbnamespace11.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=O02l2DGkAe/isD7XqY8nemw9GZYVkvGAtfLnOuqIQi8=

Create a queque

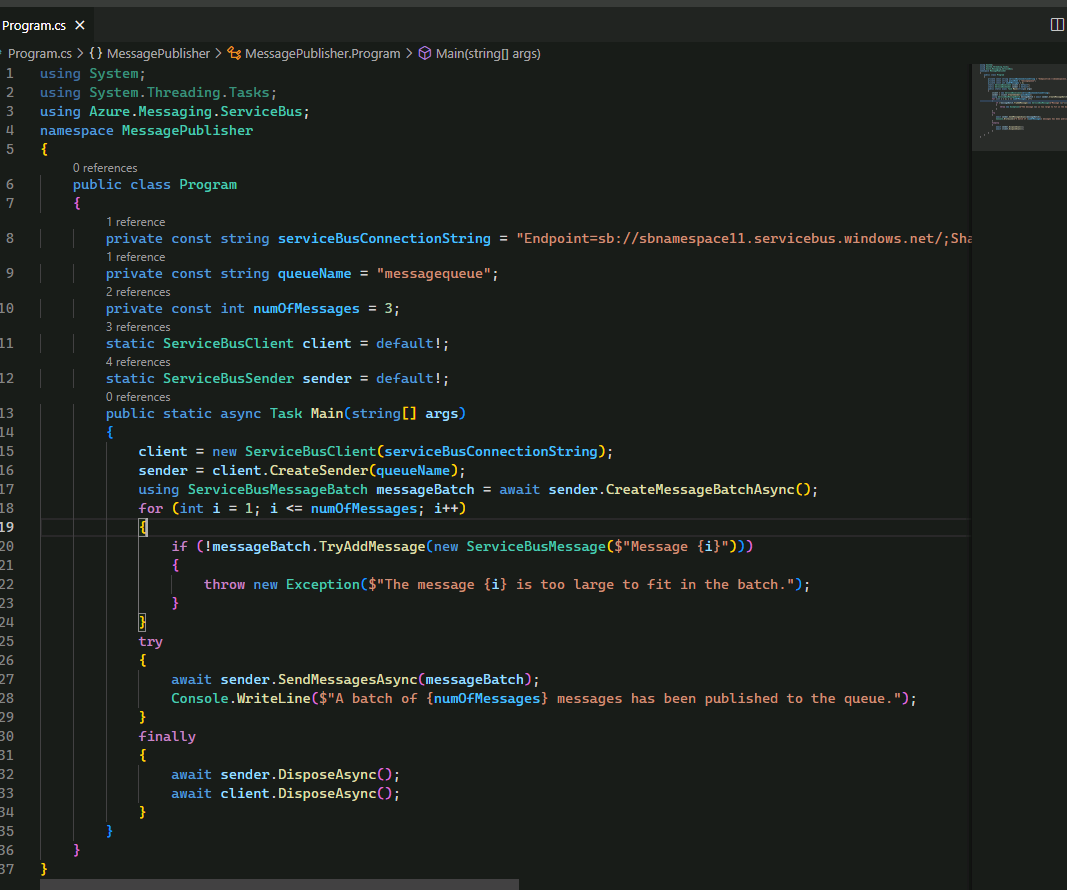
in this exercise, you created an Azure **Service Bus** namespace and a **Service Bus** queue that you'll use through the remainder of the lab.

### **Exercise 2: Create a .NET Core project to publish messages to a Service Bus queue**

create a new .NET project named **MessagePublisher** in the current folder

dotnet build

#### **Publish messages to an Azure Service Bus queue**

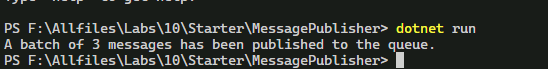


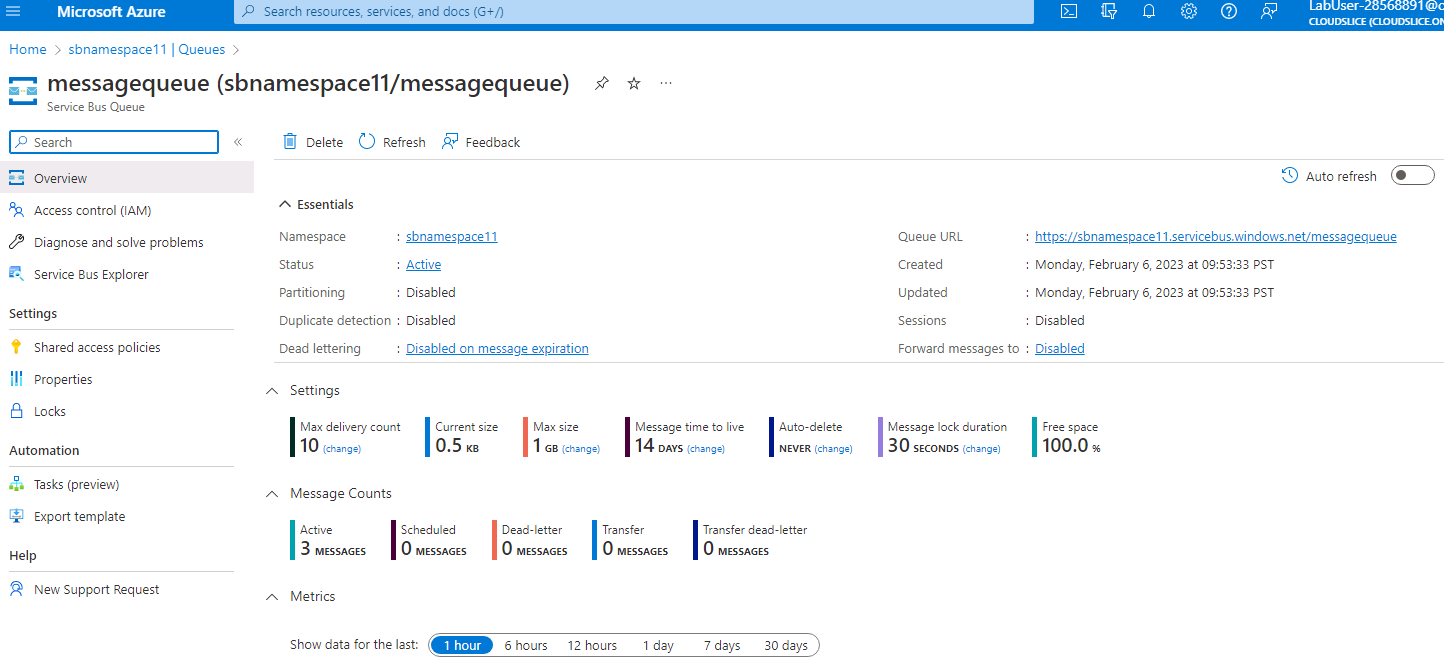
add the following code to initialize client of type **ServiceBusClient** that will provide connectivity to the Service Bus namespace and **sender** that will be responsible for sending messages

The Service Bus client is safe to cache and use as a singleton for the lifetime of the application. This is considered one of the best practices when publishing and reading messages on a regular basis.

**ServiceBusMessageBatch** object that will allow you to combine multiple messages into a batch by using the **TryAddMessage** method

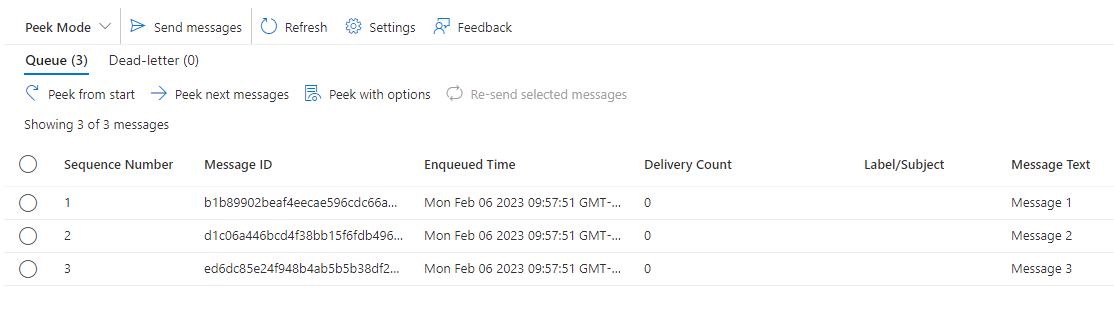
Dotnet run



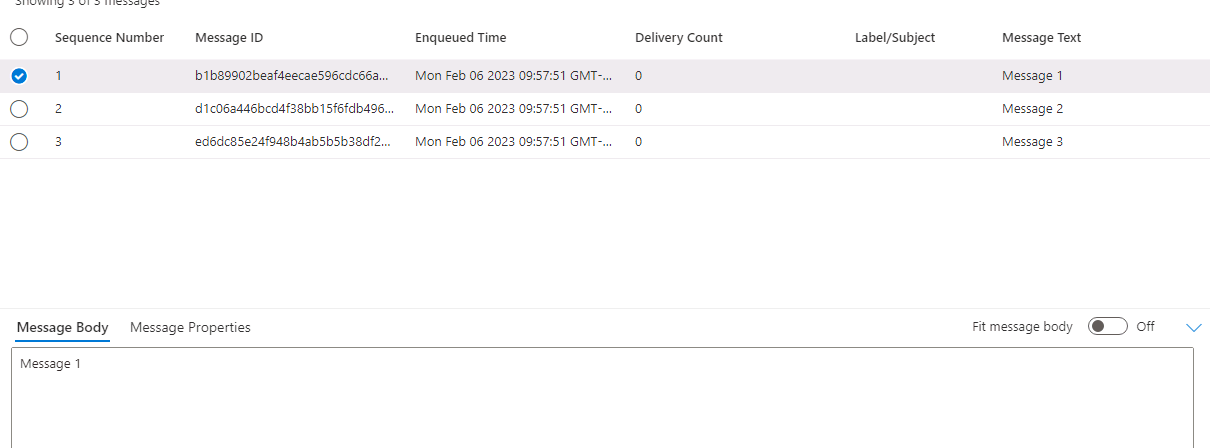


3 active messages

Peek from start:



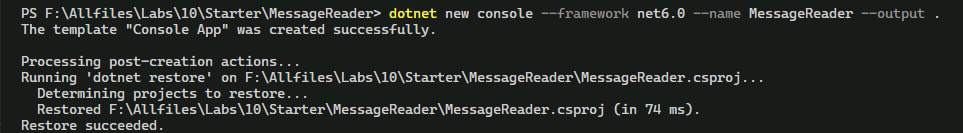
Содержимое сообщения:



n this exercise, you configured your .NET project that published messages into an Azure Service Bus queue.

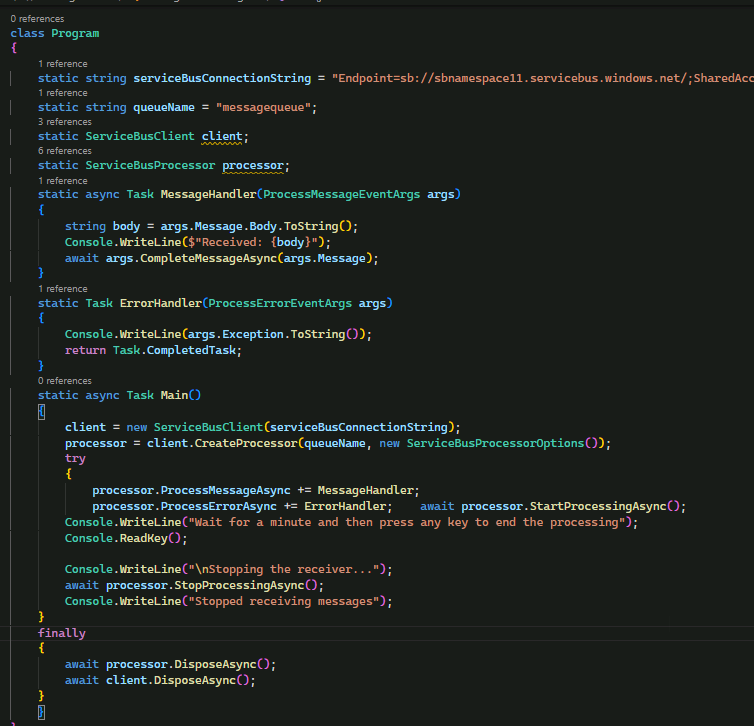
### **Create a .NET Core project to read messages from a Service Bus queue**

un the following command to create a new .NET project named **MessageReader** in the current folder



Build

#### **Read messages from an Azure Service Bus queue**

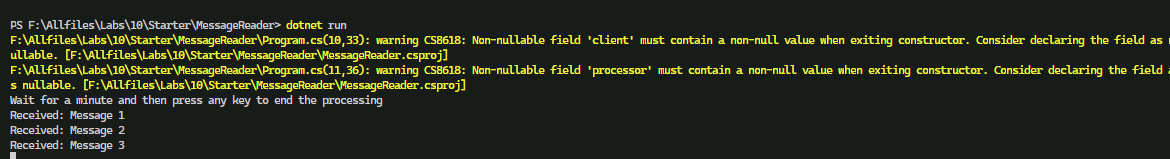


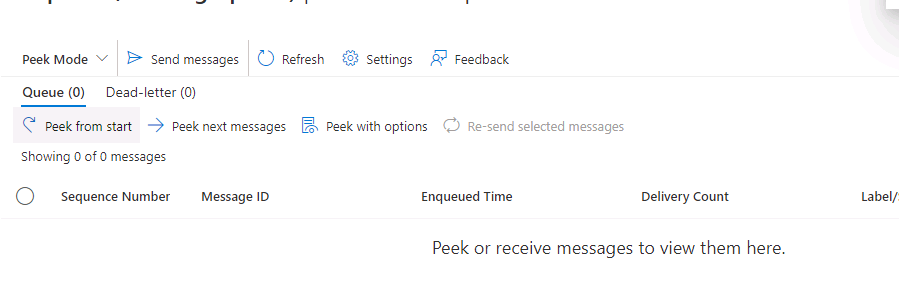
**MessageHandler** task that displays the body of messages in the queue as they are being processed and deletes them after the processing completes

**ServiceBusClient** that will provide connectivity to the Service Bus namespace and **processor** that will be responsible for processing of messages

**Main** method, add the following lines of code to create a try block, which first implements a message and error processing handler, initiates message processing, and stops processing following a user input

Dotnet run





Теперь peekfromstart возвращает 0

In this exercise, you read and deleted messages from the Azure Service Bus queue by using the .NET library.