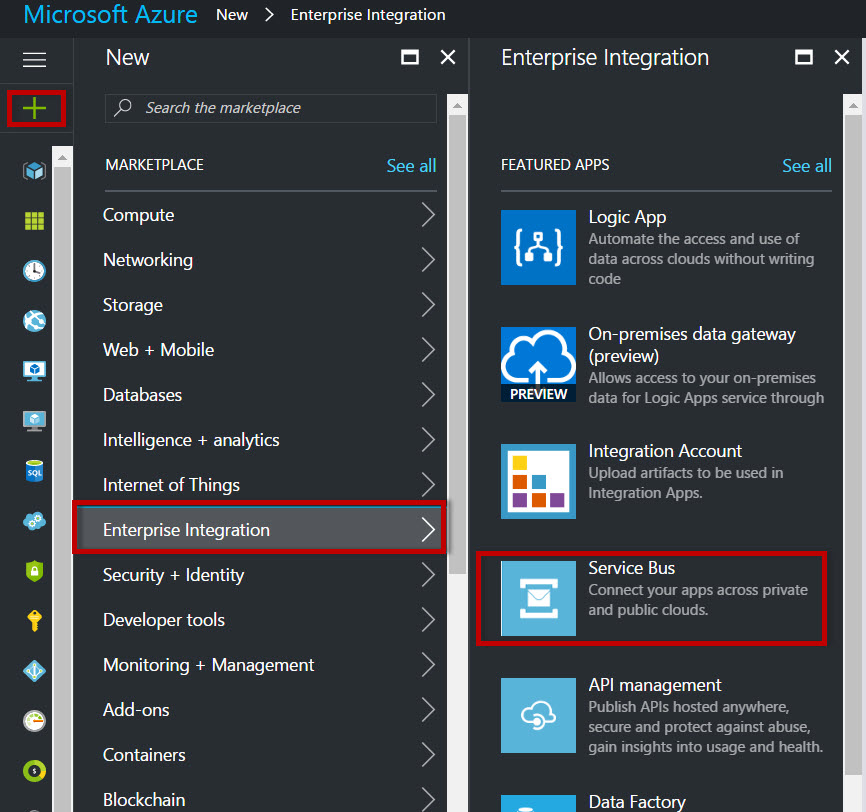
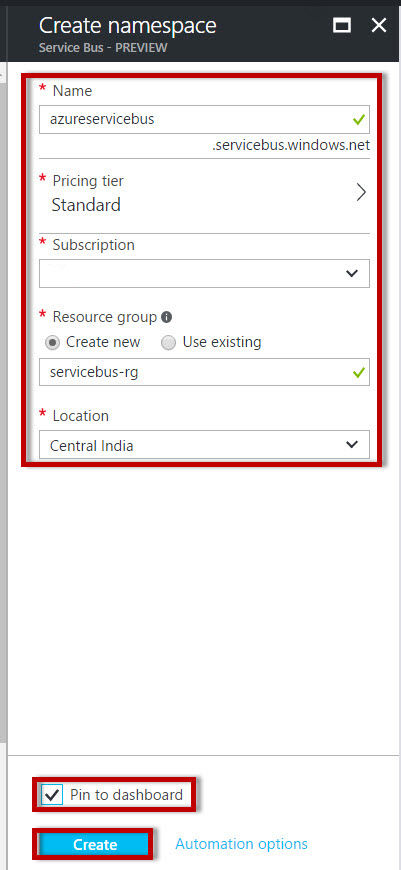
**Azure Service Bus – Queue**

Step 1: Navigate to Azure Portal.

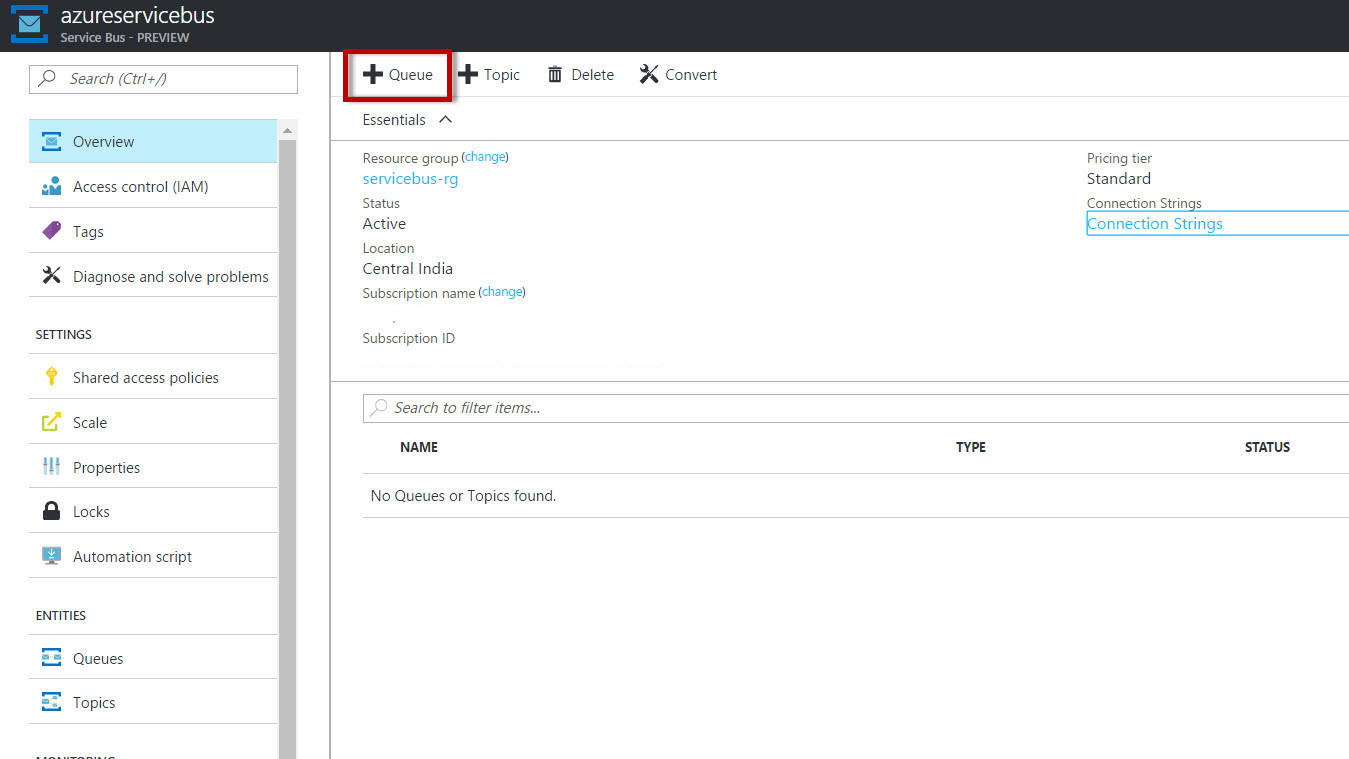
Step 2: Click on +New -> Enterprise Integration -> Service Bus



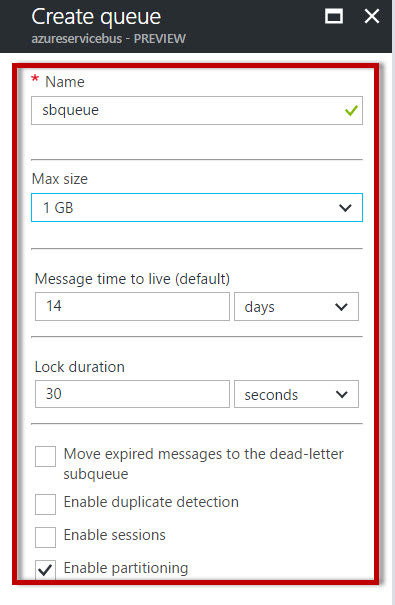
Step 3: Enter Service Bus Namespace Name, Choose Pricing Tier, Resource Group, Location



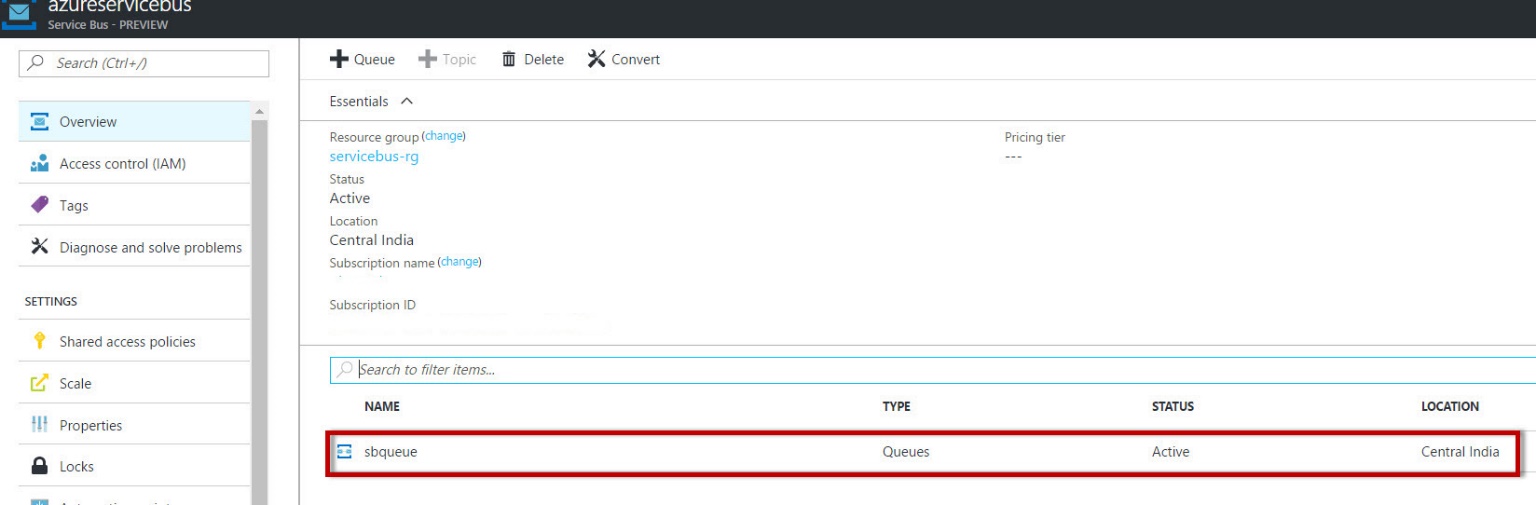
Step 4: Click on **+** **Queue** to create new Queue



Step 5: Enter Queue Name, Max Size, Message time to live, Lock duration,etc.

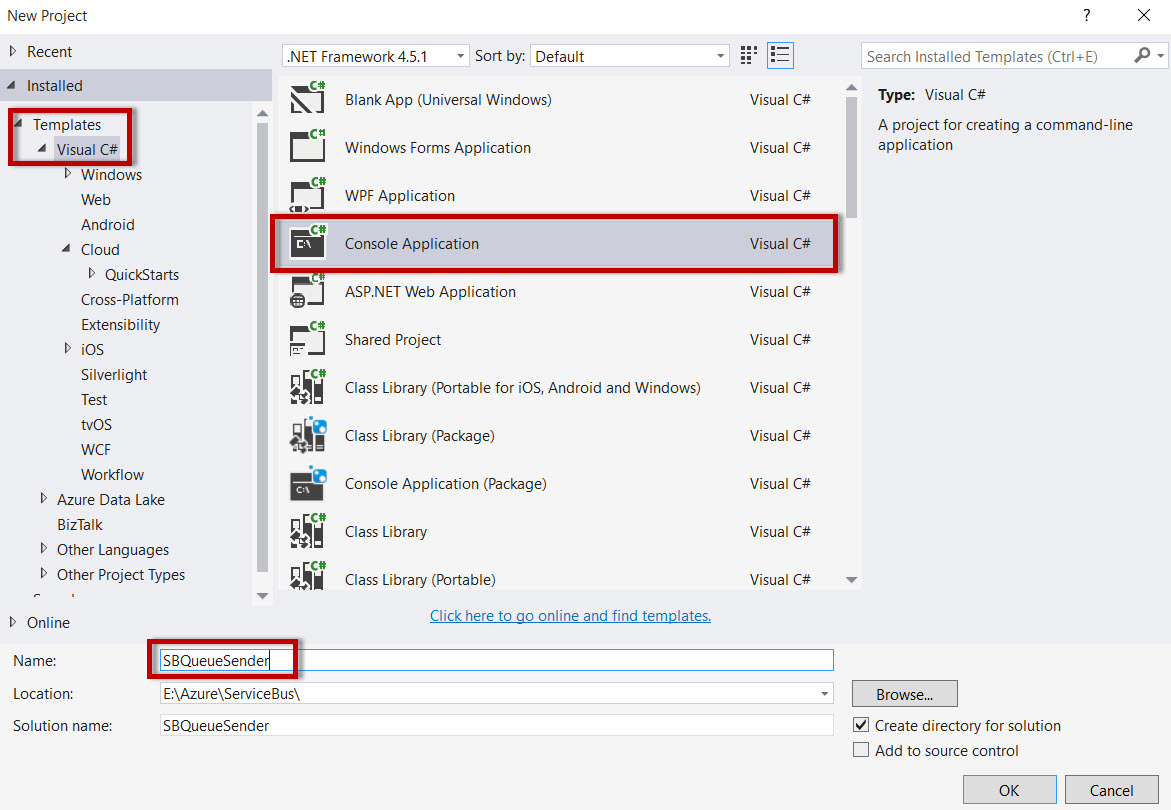


Queue created successfully.

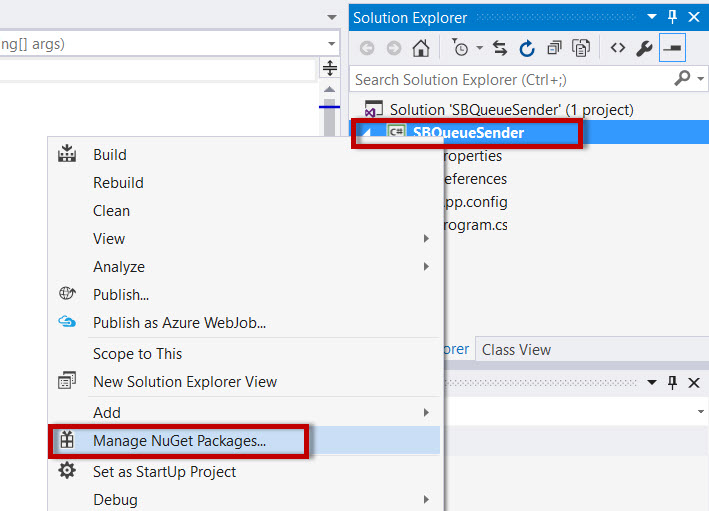


Step 6: Start Visual Studio & Choose Visual C# Template.

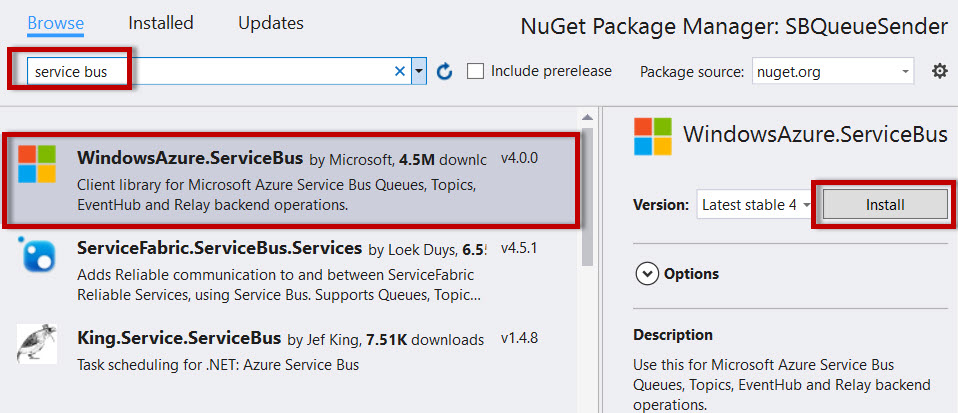
Create New Console Application



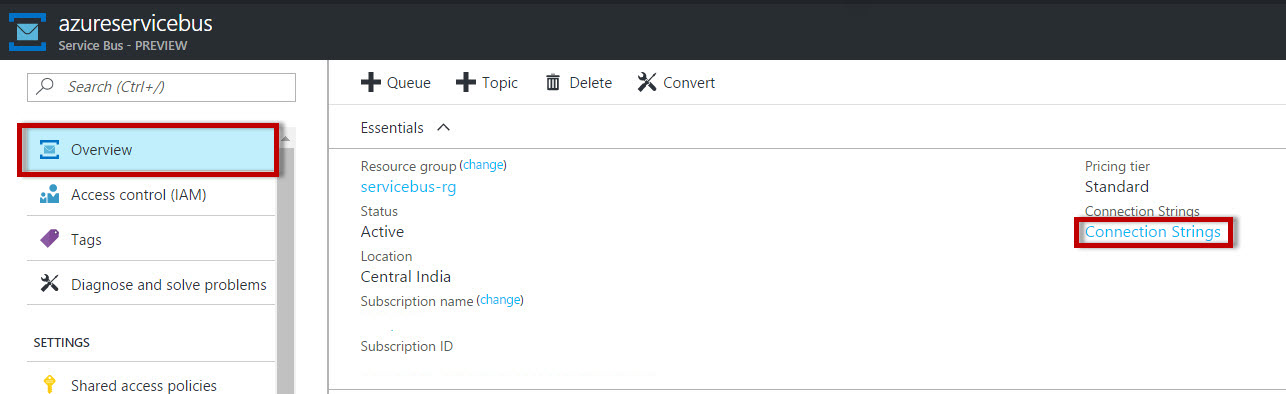
Step 7: Right click on Project Name & select “**Manage NuGet Packages…**”



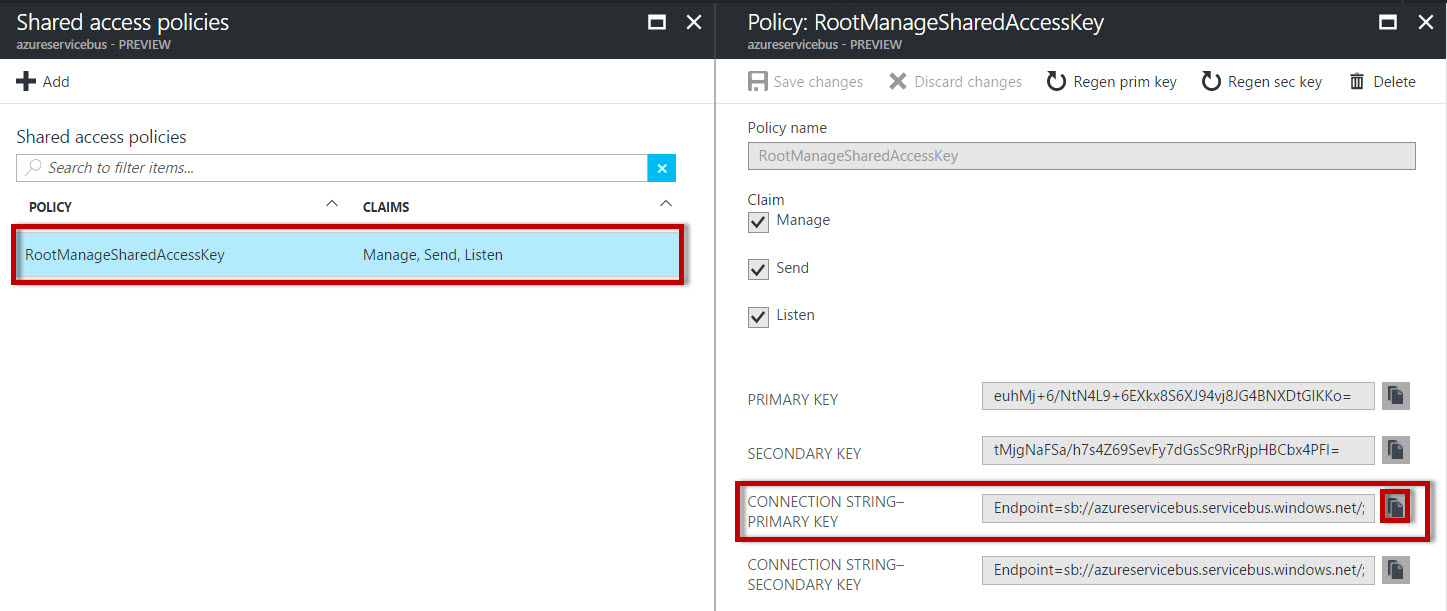
Step 8: Search for “service bus” & install WindowsAzure.ServiceBus packages.



Step 9: Navigate to Azure Portal & Select **Connection String** option.



Click on Copy option & paste into **program.cs** file



Step 10: Add Service Bus Namespace

*using Microsoft.ServiceBus.Messaging;*

namespace SBQueueSender

{

class Program

{

static string ConnectionString = "Endpoint=sb://azureservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=<copy-servicebus-sharedkey-from-azure-portal>";

static string QueuePath = "sbqueue";

static void Main(string[] args)

{

//Service Bus Queue Sender

var queueClient = QueueClient.CreateFromConnectionString(ConnectionString, QueuePath);

for (int i = 0; i < 10; i++)

{

var message = new BrokeredMessage("Sender's Message ==> " + i);

// message.SessionId = “test”;

queueClient.Send(message);

Console.Write("\nSent Message : = " + i );

}

Console.WriteLine("Press Enter to Exit...");

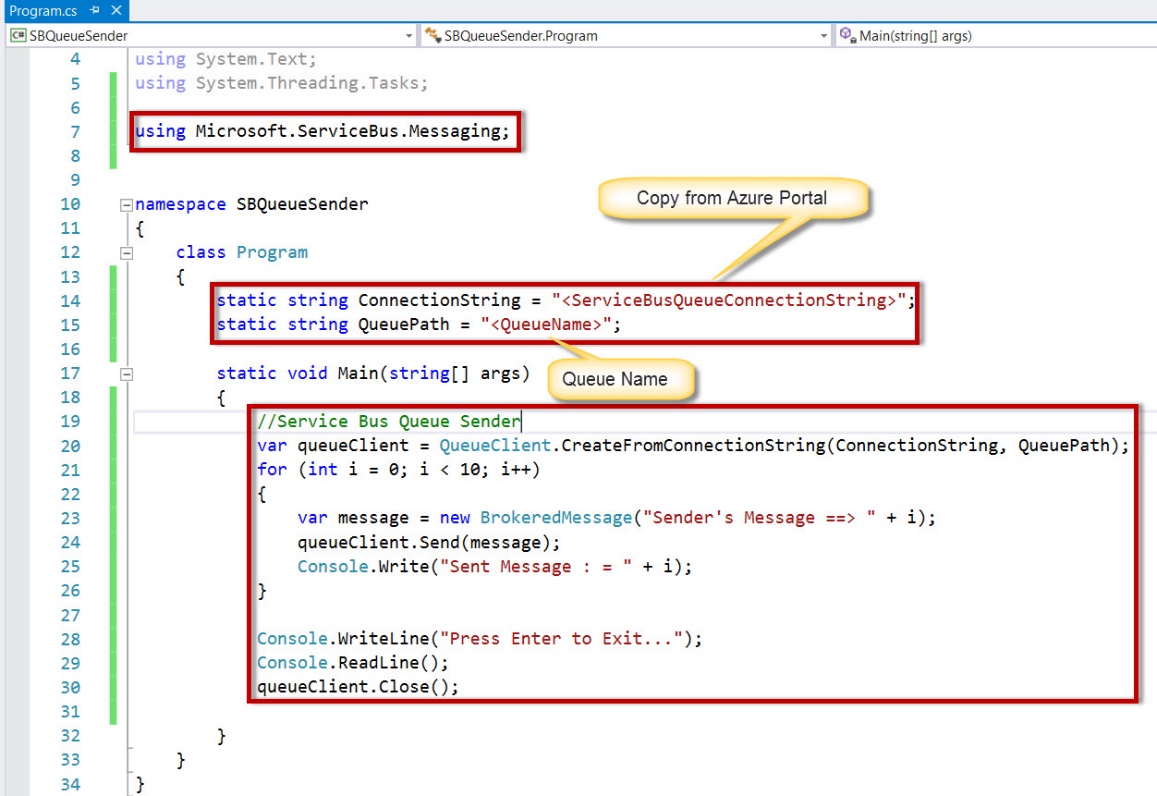
Console.ReadLine();

queueClient.Close();

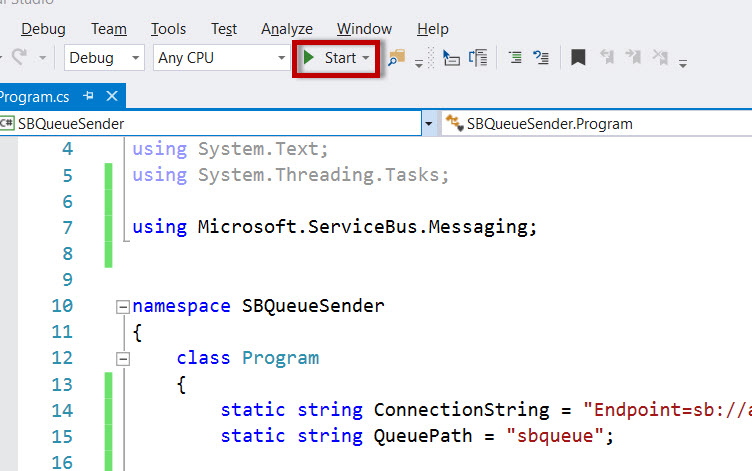
}

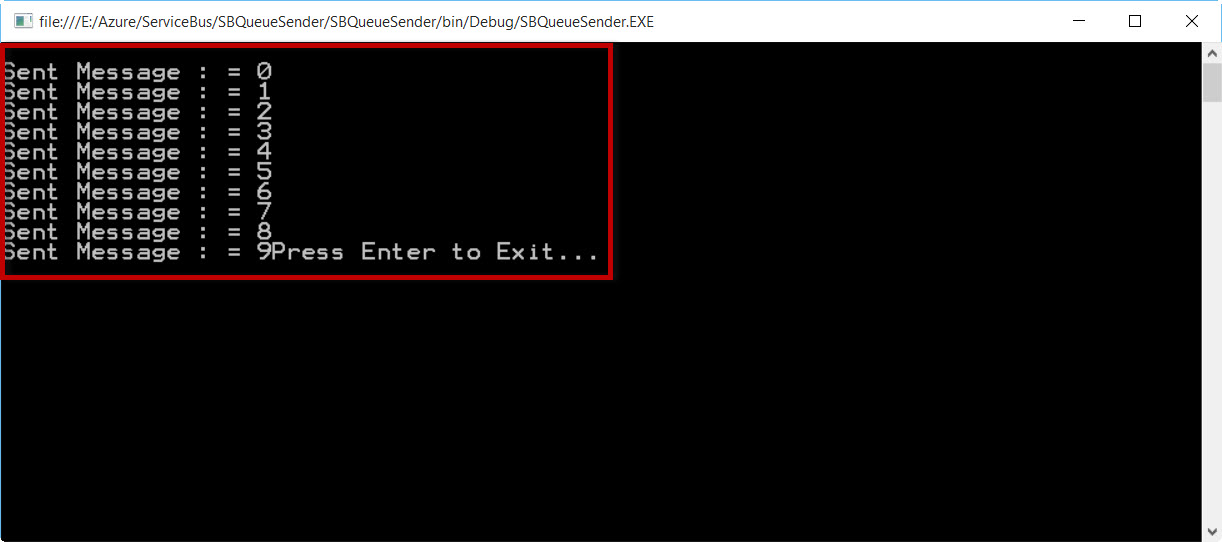
}

}

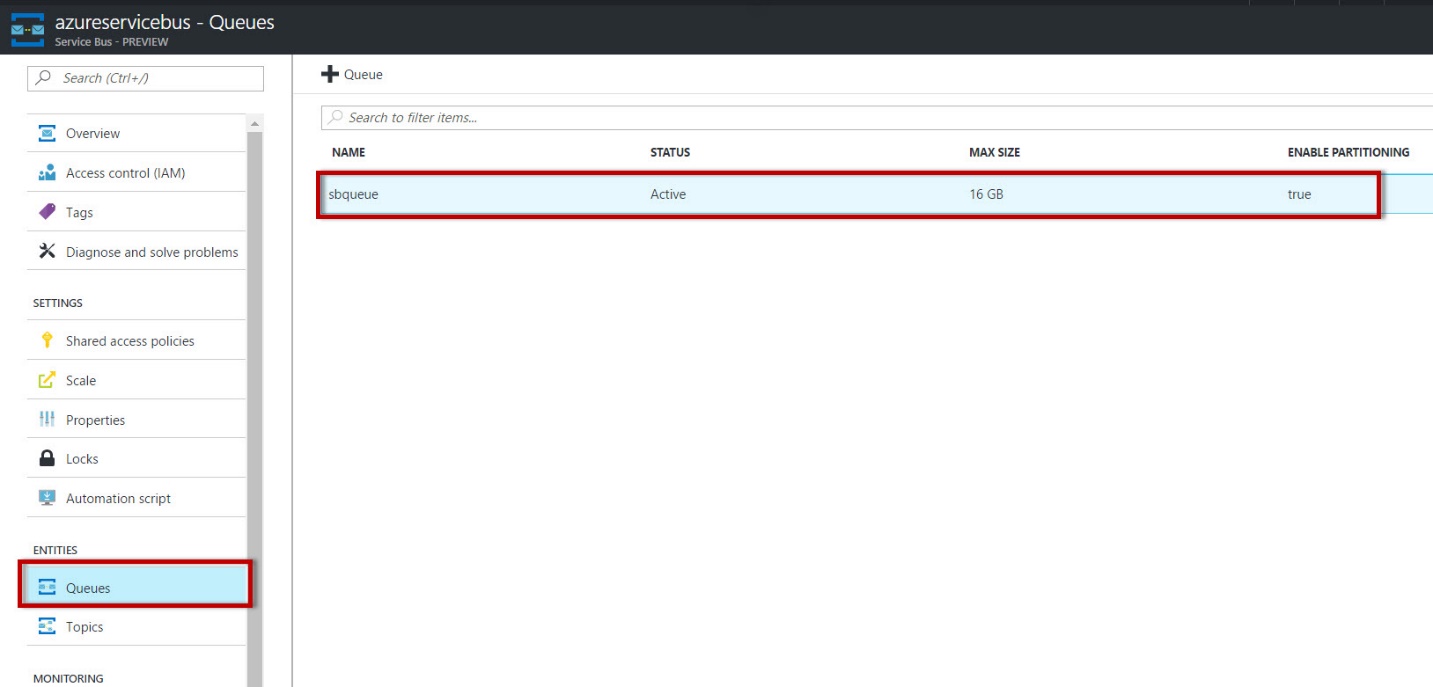


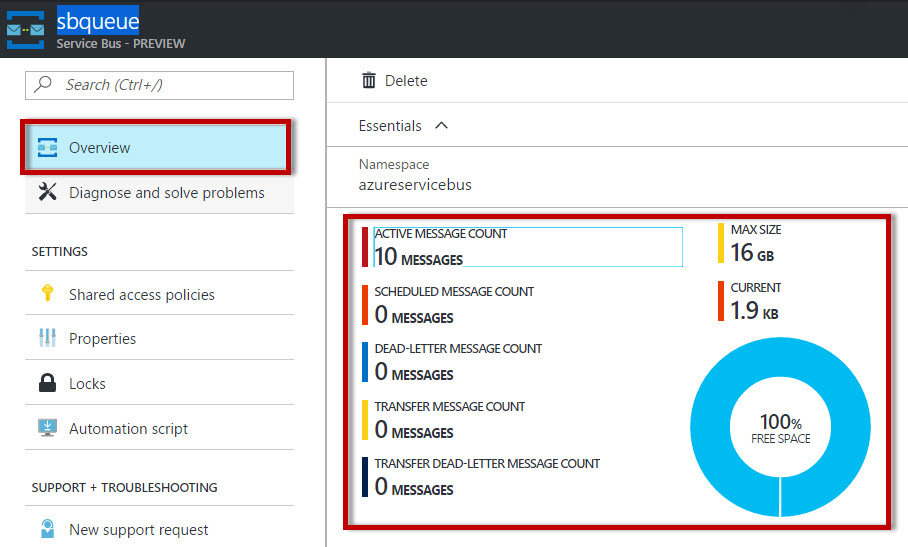
Step 11: Now run the Sender program



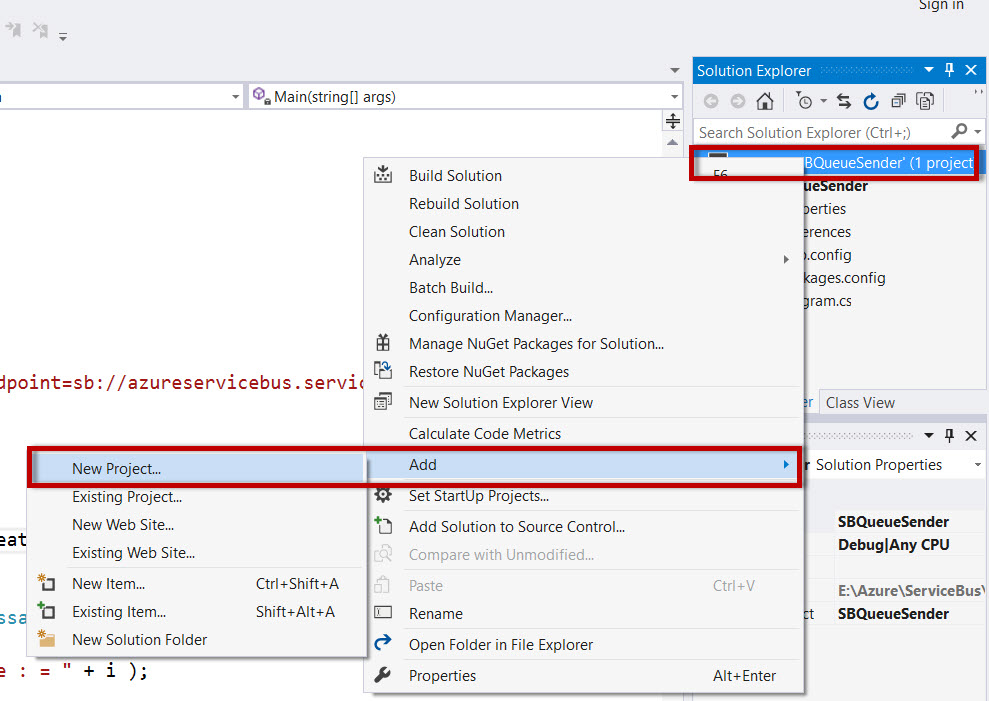


Again Navigate to Service Bus Queue option

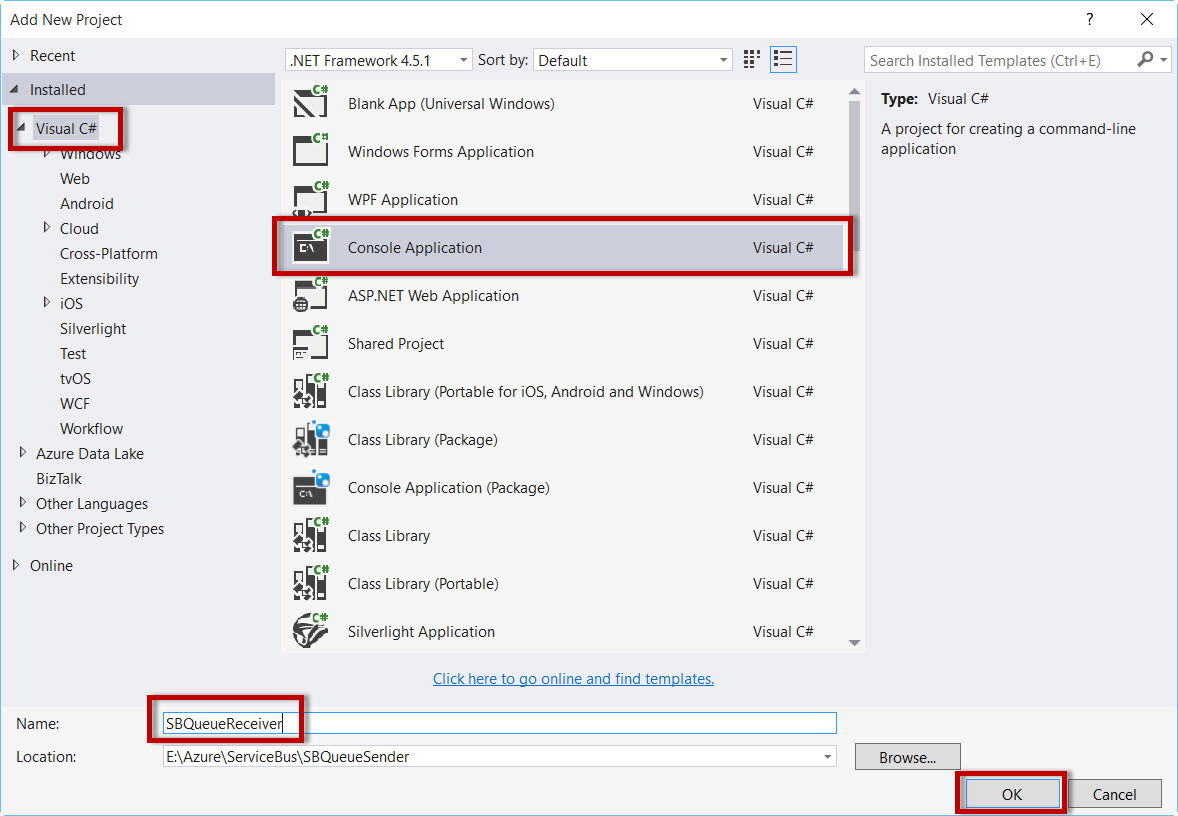




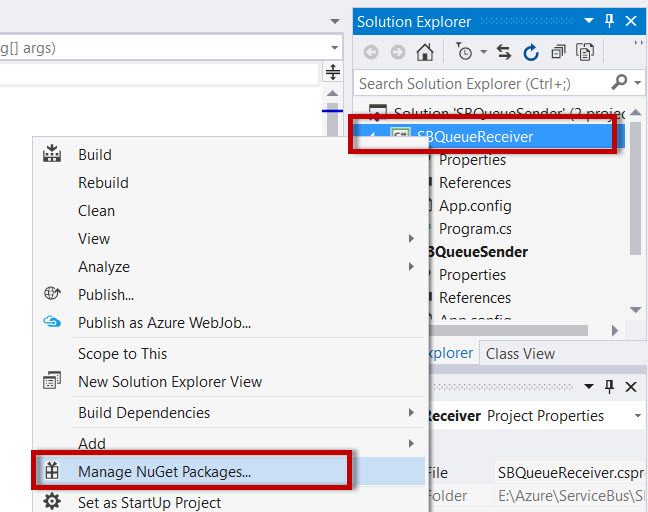
Step 12: Create one more project for Receiver using Visual Studio.



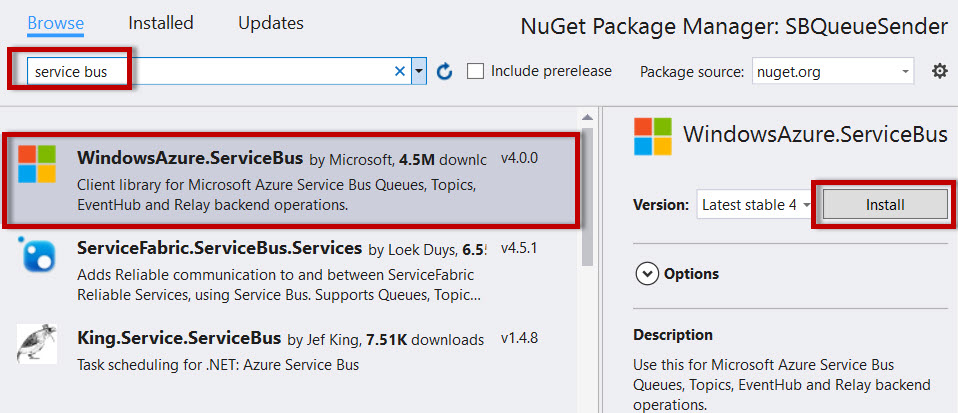
Create Console Application.



repeat the same steps to install NuGet Packages of Azure Service Bus.



Search for “service bus” & install NuGet Packages.

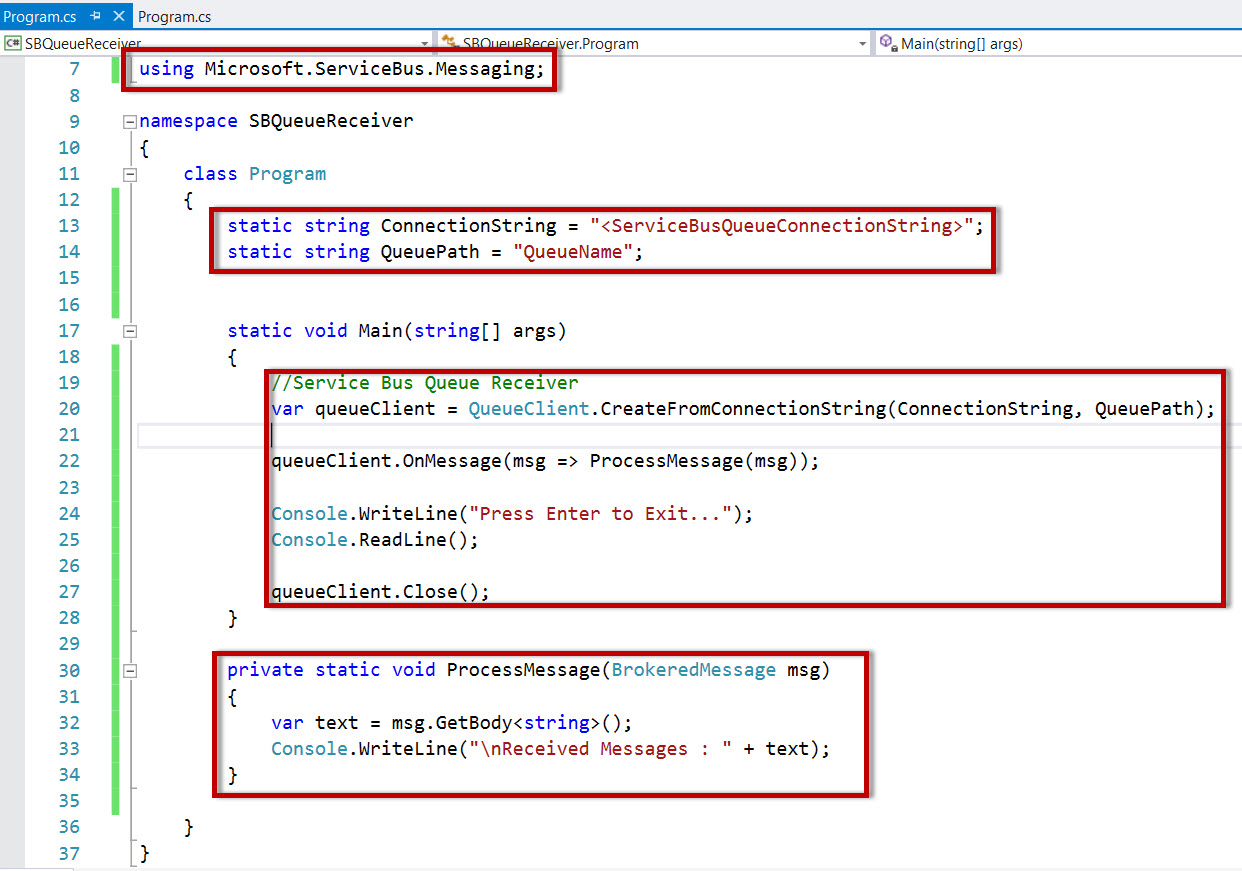


Step 13:

Add Service Bus Namespace

*using Microsoft.ServiceBus.Messaging;*

Get ConnectionString & QueueName from Azure Portal or copy from Sender’s Project.



namespace SBQueueReceiver

{

class Program

{

static string ConnectionString = "Endpoint=sb://azureservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=<copy-servicebus-sharedkey-from-azure-portal>";

static string QueuePath = "sbqueue";

static void Main(string[] args)

{

//Service Bus Queue Receiver

var queueClient = QueueClient.CreateFromConnectionString(ConnectionString, QueuePath);

queueClient.OnMessage(msg => ProcessMessage(msg));

Console.WriteLine("Press Enter to Exit...");

Console.ReadLine();

queueClient.Close();

}

private static void ProcessMessage(BrokeredMessage msg)

{

var text = msg.GetBody<string>();

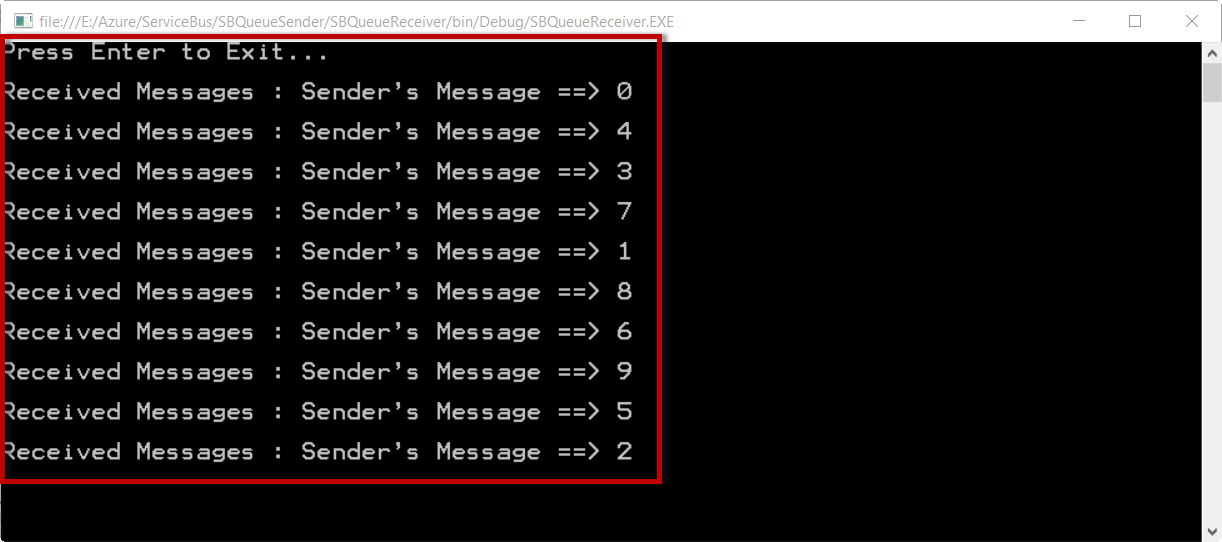
Console.WriteLine("\nReceived Messages : " + text);

}

}

}

Step 14: Now run the Receiver project.



Navigate to Azure Portal.

