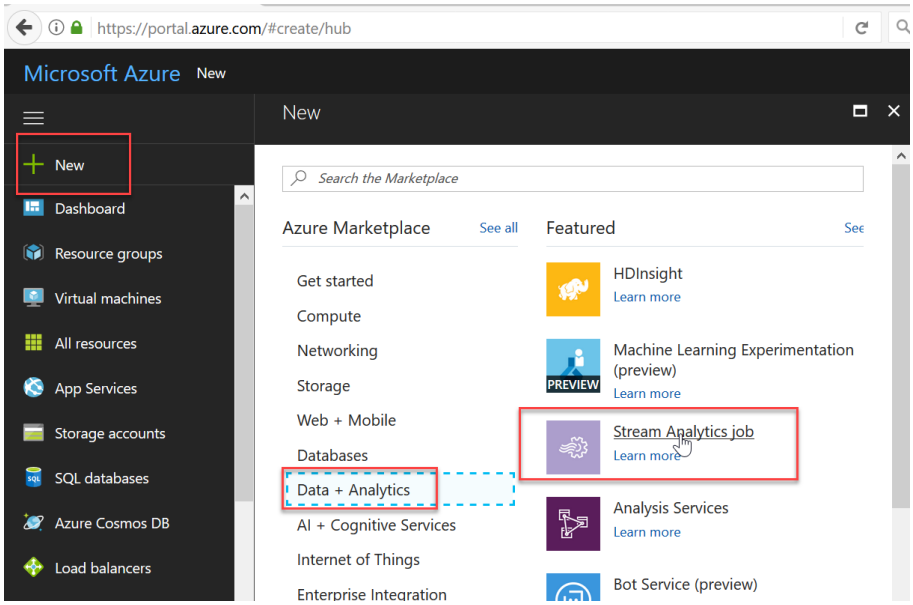
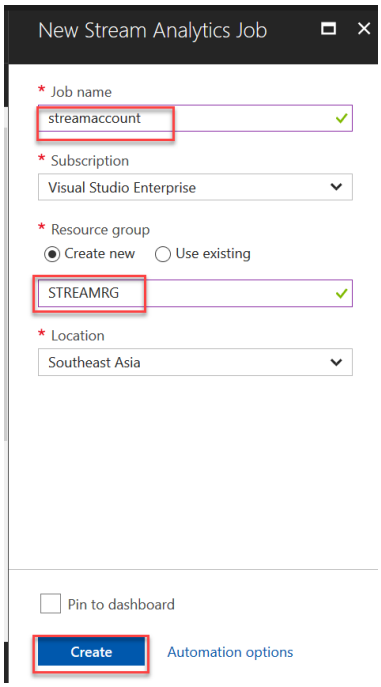
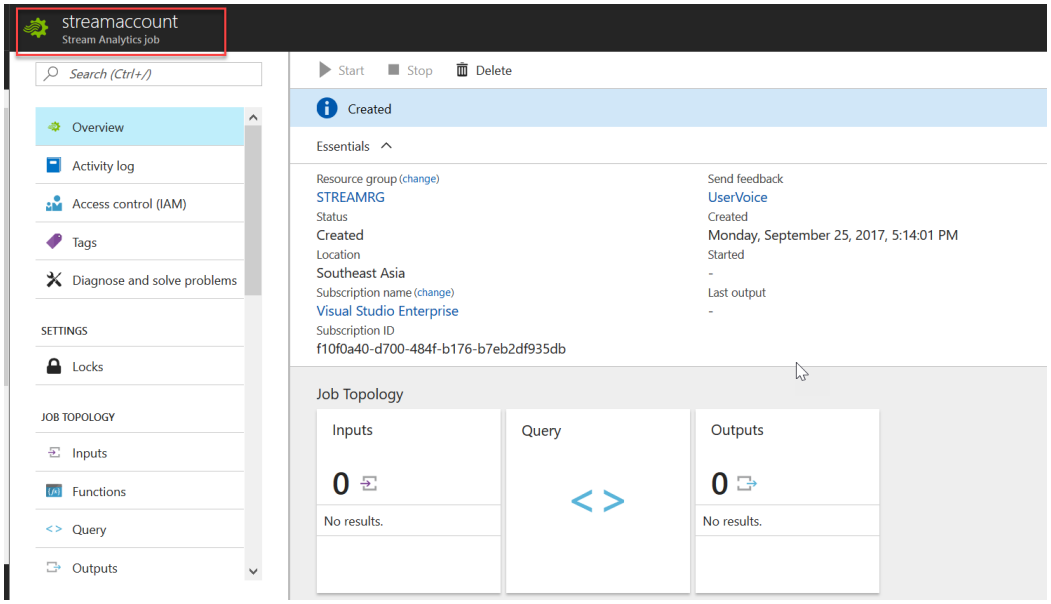
**Azure Stream Analytics – Event Hub – Service Bus – Storage**

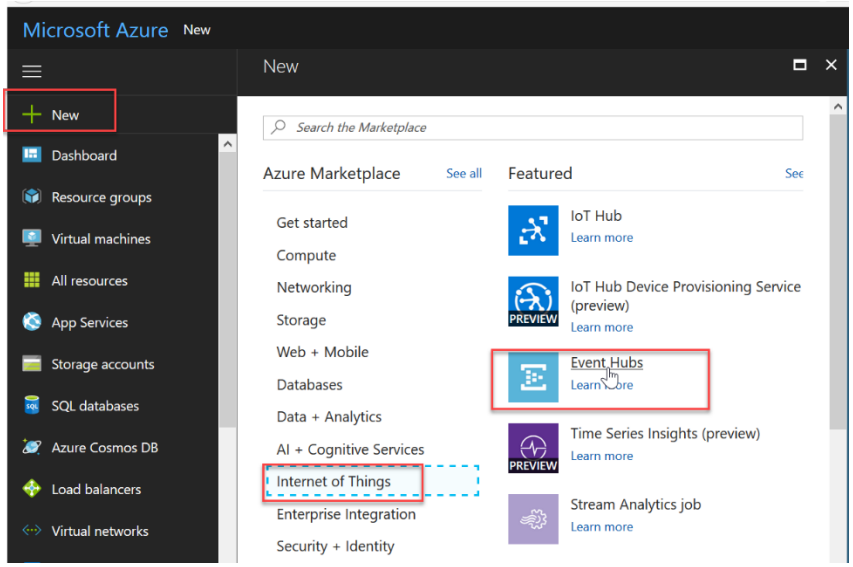
Step – Create Stream Analytics Service in Azure Portal.



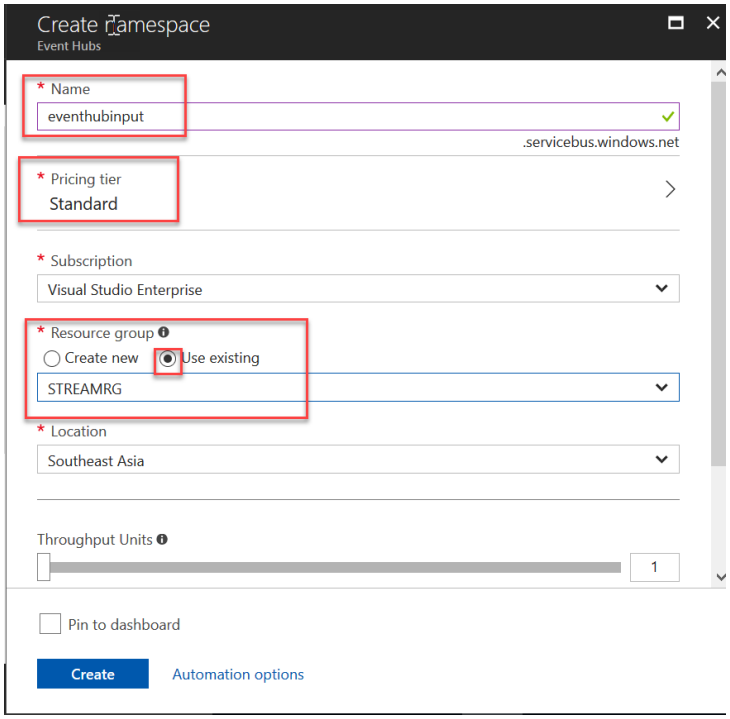





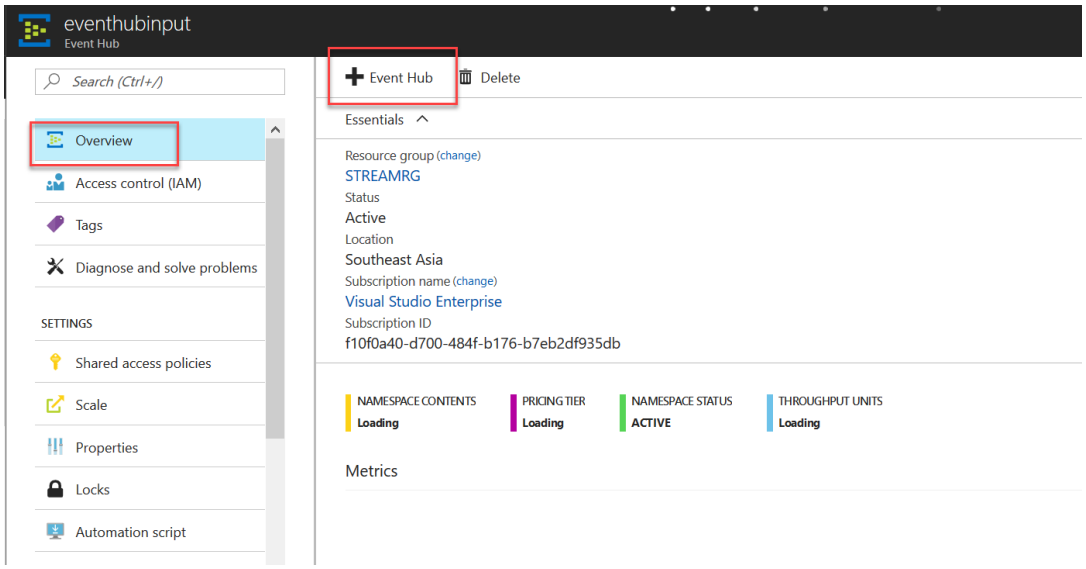

Now Create Event Hub for Data Input

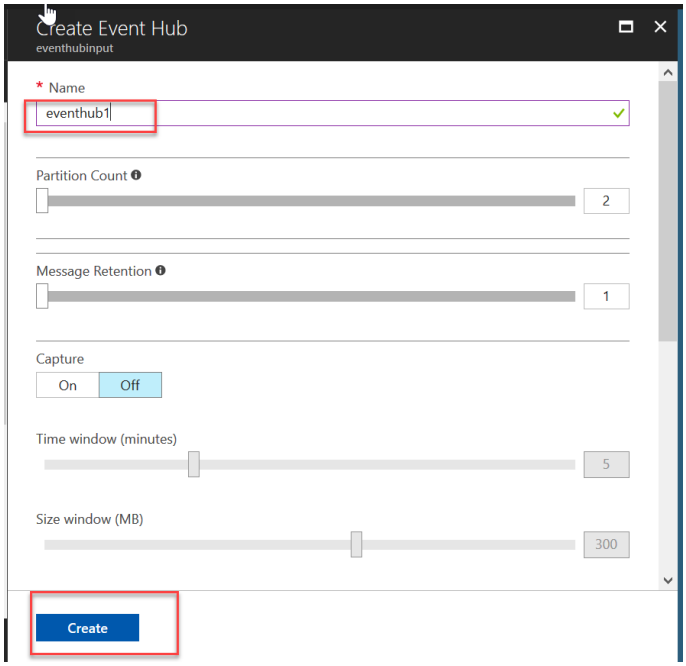


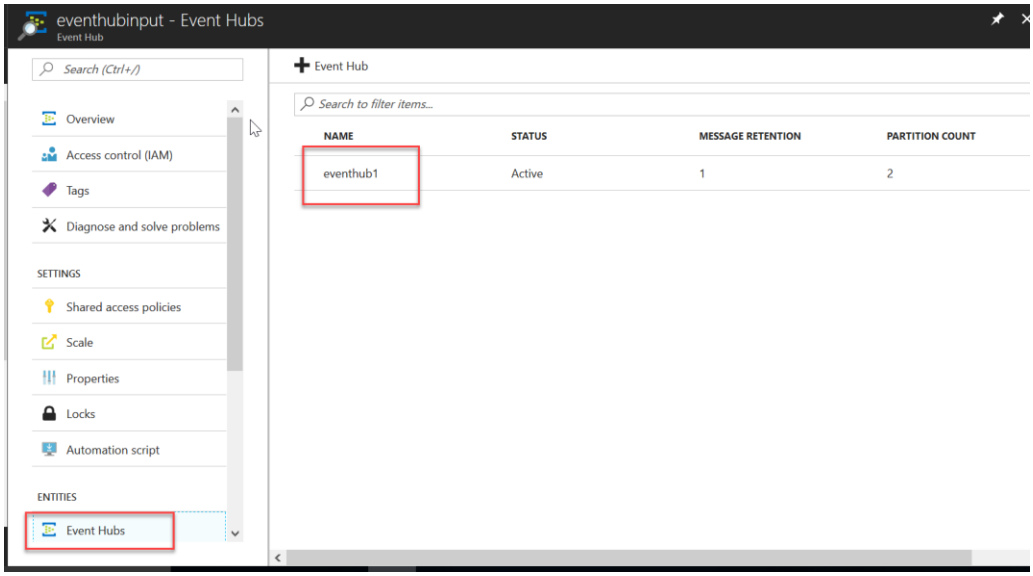

Create Event Hub and Select Pricing Tier - Standard




## Add Event Hub





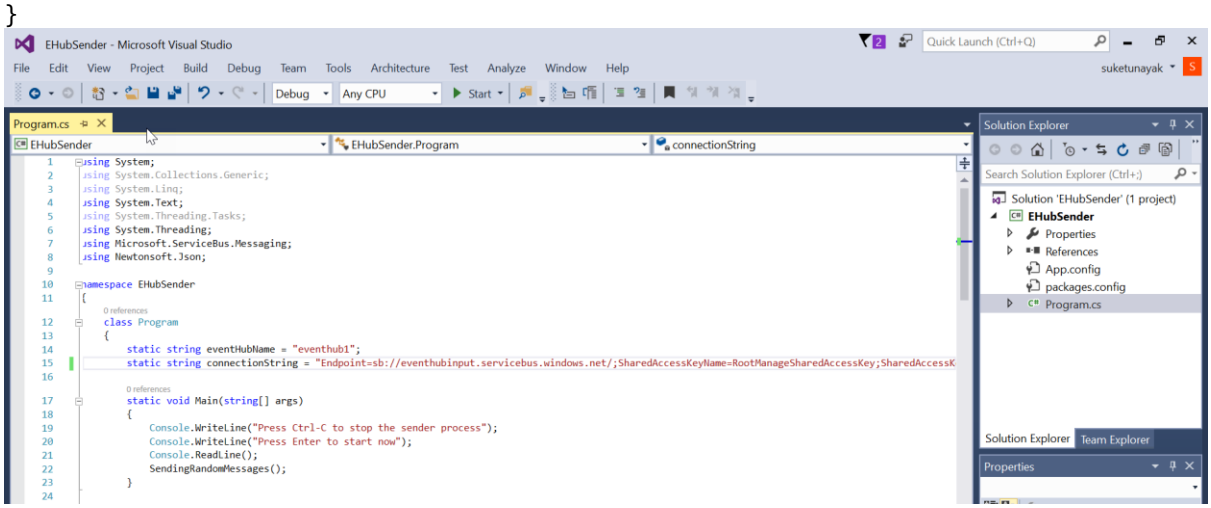




Open Shared Access Policies -> RootManageSharedAccessKey -> Copy and Paste Connection String in Notepad, Now We need to create Console Application of Event Hub to send Messages to Stream Analytics. Open Visual Studio and Click File -> New Project - > Console Application and Copy and Paste below code and in that Code change Connection String of Event hub with your connection string.

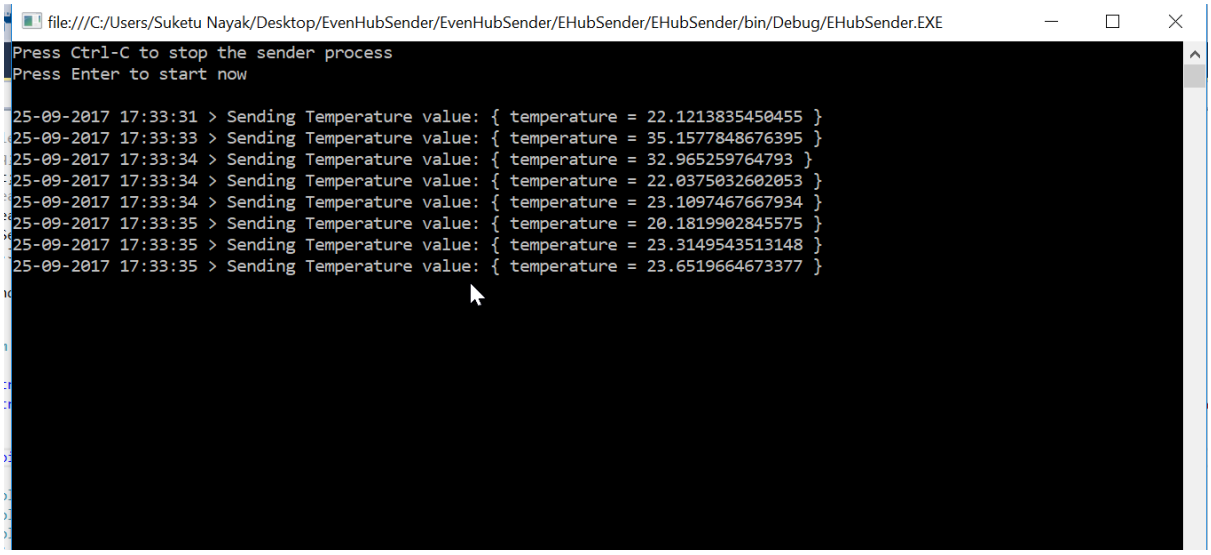
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
using Microsoft.ServiceBus.Messaging;
using Newtonsoft.Json;

namespace EHubSender
{
    class Program
```

```
    {
        static string eventHubName = "eventhub1";
        static string connectionString =
"Endpoint=sb://eventhubinput.servicebus.windows.net/;SharedAccessKeyName=RootManageSha
redAccessKey;SharedAccessKey=blfYZqefLsCwOkN1J/xIn3m+LTNwarrtn8atgCGLqCA=";

        static void Main(string[] args)
        {
            Console.WriteLine("Press Ctrl-C to stop the sender process");
            Console.WriteLine("Press Enter to start now");
            Console.ReadLine();
            SendingRandomMessages();
        }



        static void SendingRandomMessages()
        {
            double temp;

            var eventHubClient =
EventHubClient.CreateFromConnectionString(connectionString, eventHubName);
            while (true)
            {
                try
                {
                    // Simulate reading Temperature
                    var rand = new Random();
                    double humi = 45 + rand.NextDouble() * 4 - 2;

                    if (rand.NextDouble() > 0.8)
                    {
                        temp = 22 + rand.NextDouble() * 4 + 10;
                    }
                    else
                    {
                        temp = 22 + rand.NextDouble() * 4 - 2;
                    }

                    var messageInfo = new
                    {

                        temperature = temp
                    };

                    var serialJson = JsonConvert.SerializeObject(messageInfo);

                    EventData data = new
EventData(Encoding.UTF8.GetBytes(serialJson));

                    Console.WriteLine("{0} > Sending Temperature value: {1}",
DateTime.Now, messageInfo);
                    eventHubClient.Send(data);
                }
                catch (Exception exception)
                {
                    Console.ForegroundColor = ConsoleColor.Red;
                    Console.WriteLine("{0} > Exception: {1}", DateTime.Now,
exception.Message);
                    Console.ResetColor();
                }
```

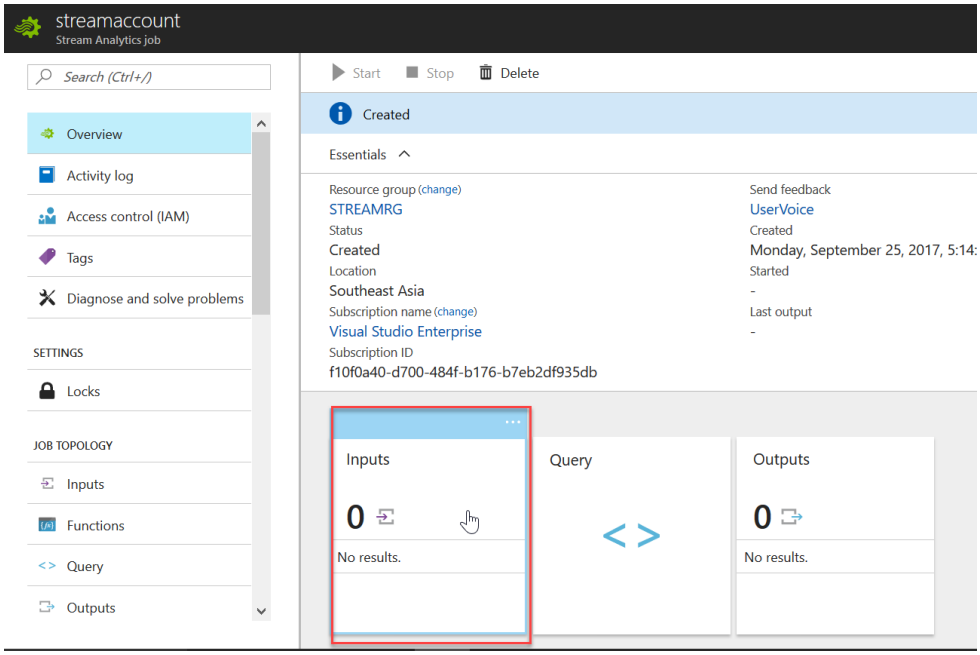
```
                Thread.Sleep(200);
            }
        }
    }
}
```



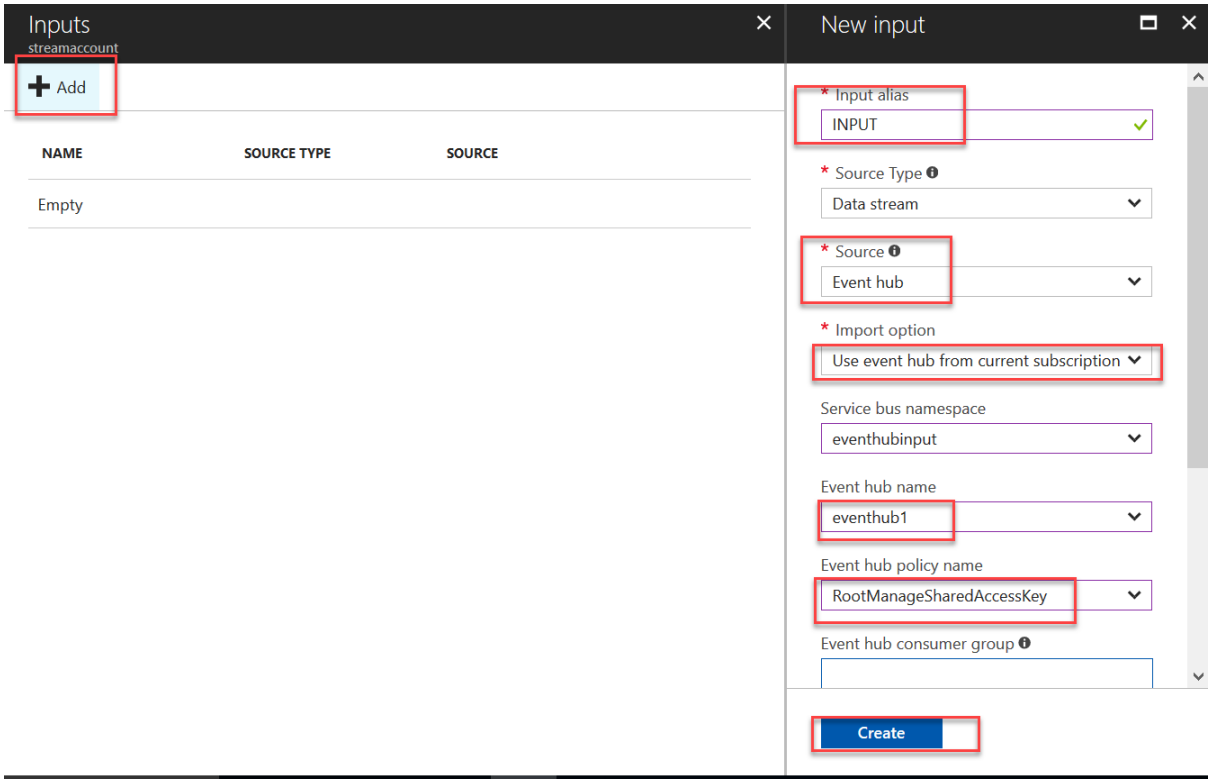

This Console Application will keep sending random temperature to Stream Analytics.



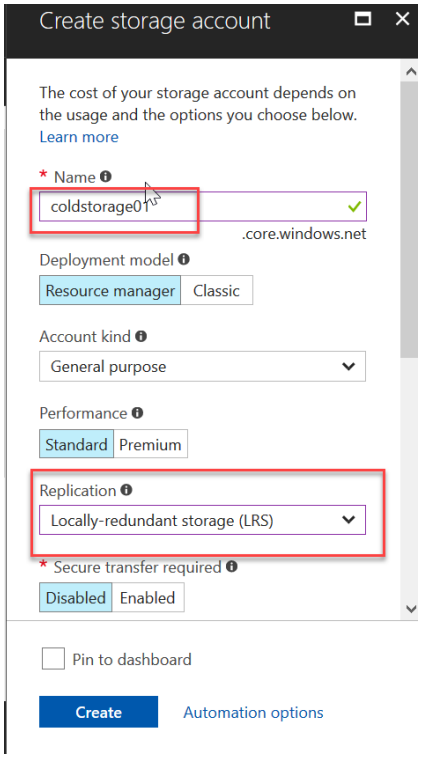

Now Add Event Hub as a Input of Stream Analytics Service in Azure Portal



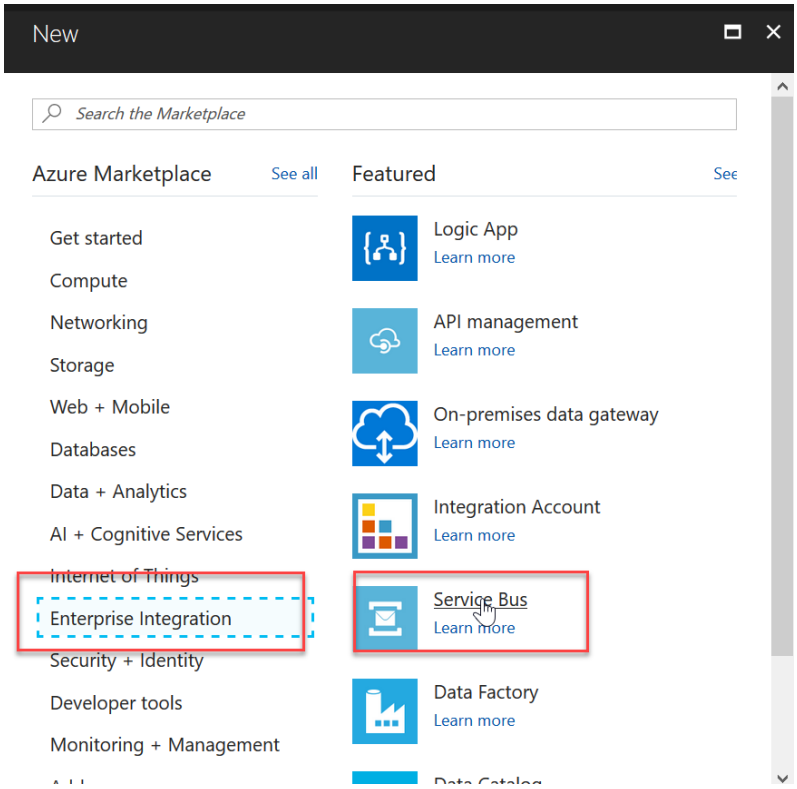

Add Event Hub

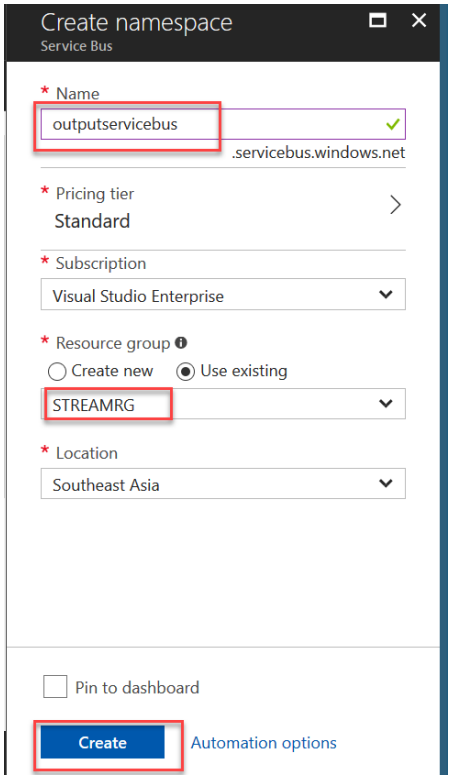



Now Create a Storage Account

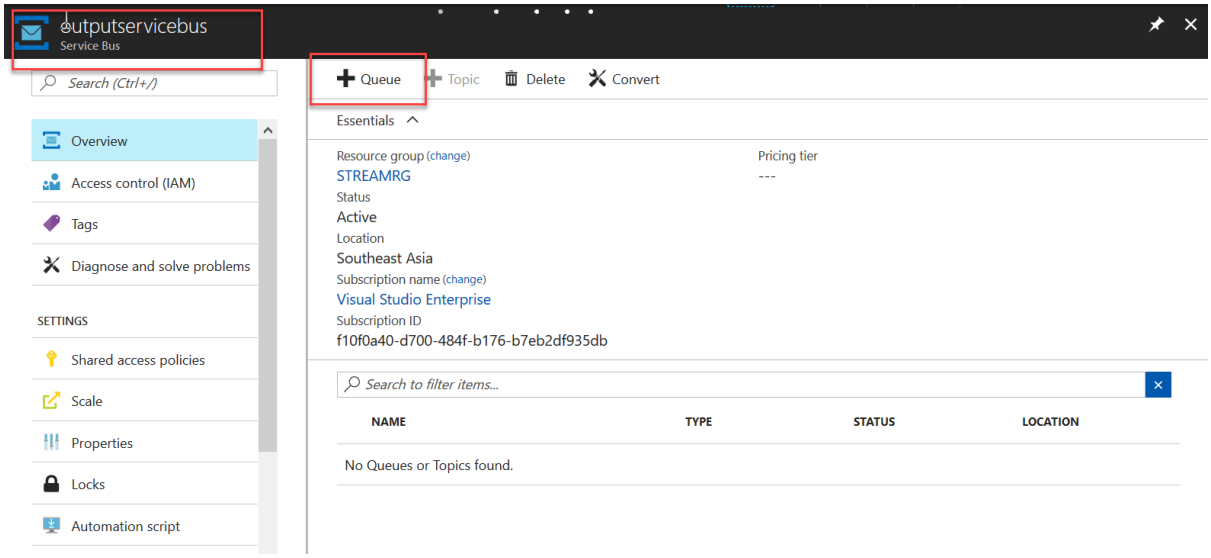


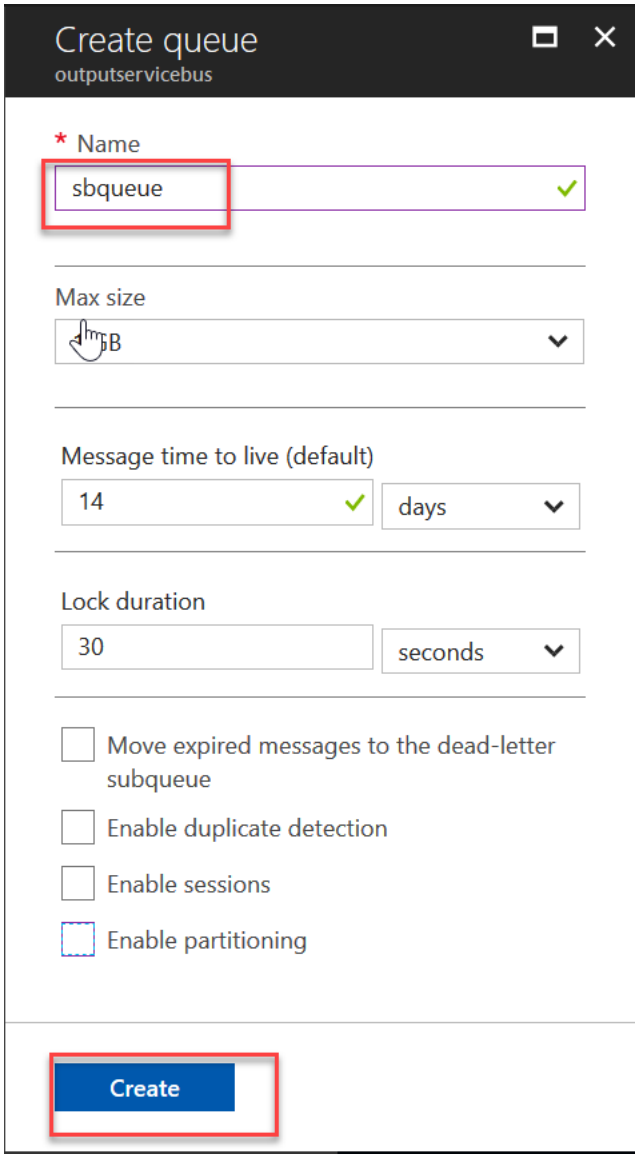

Create Service Bus for another output of Stream Analytics



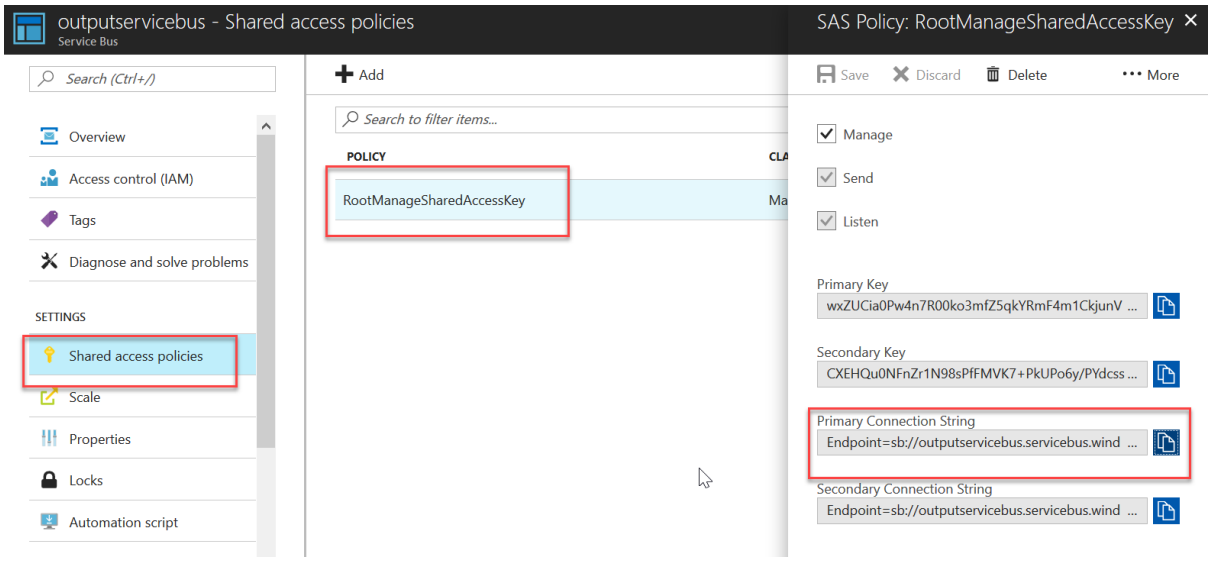



## Add Queue in Service Bus

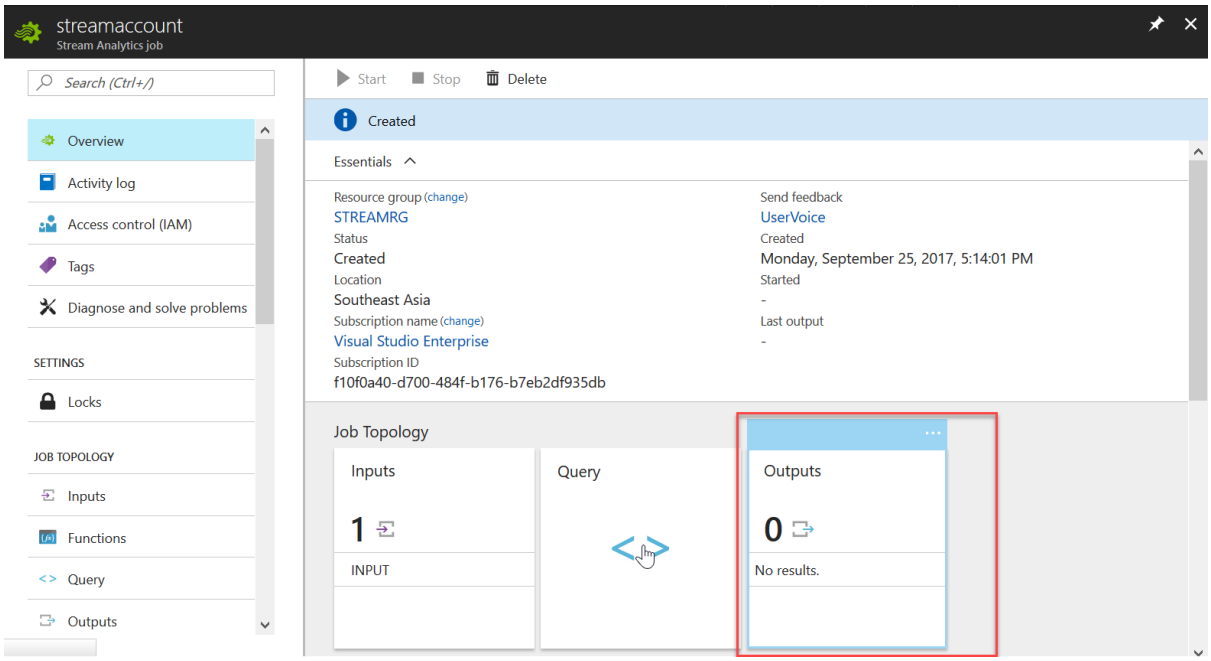




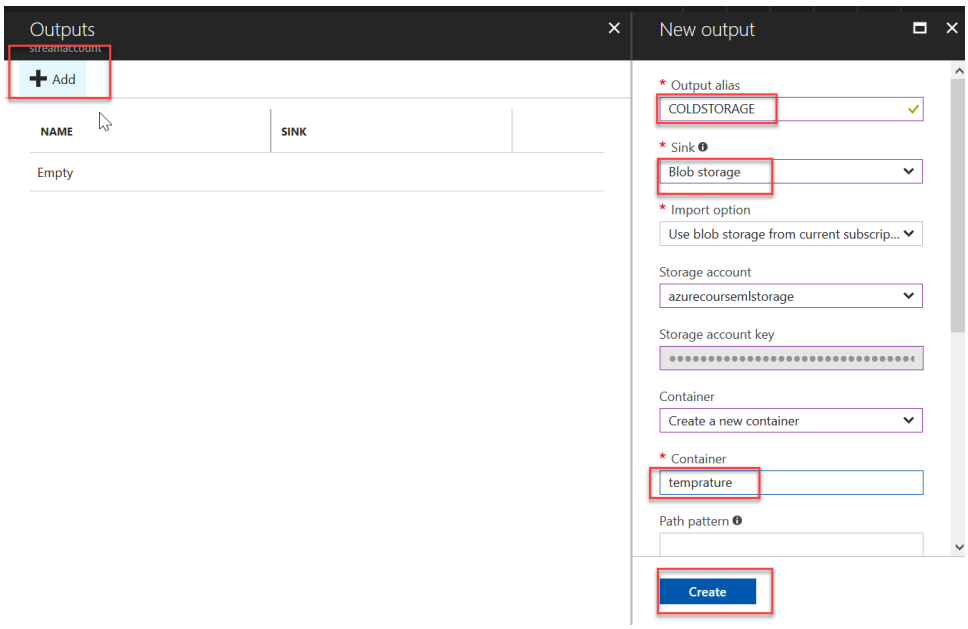

Copy and Paste Connection String of Service Bus in Notepad



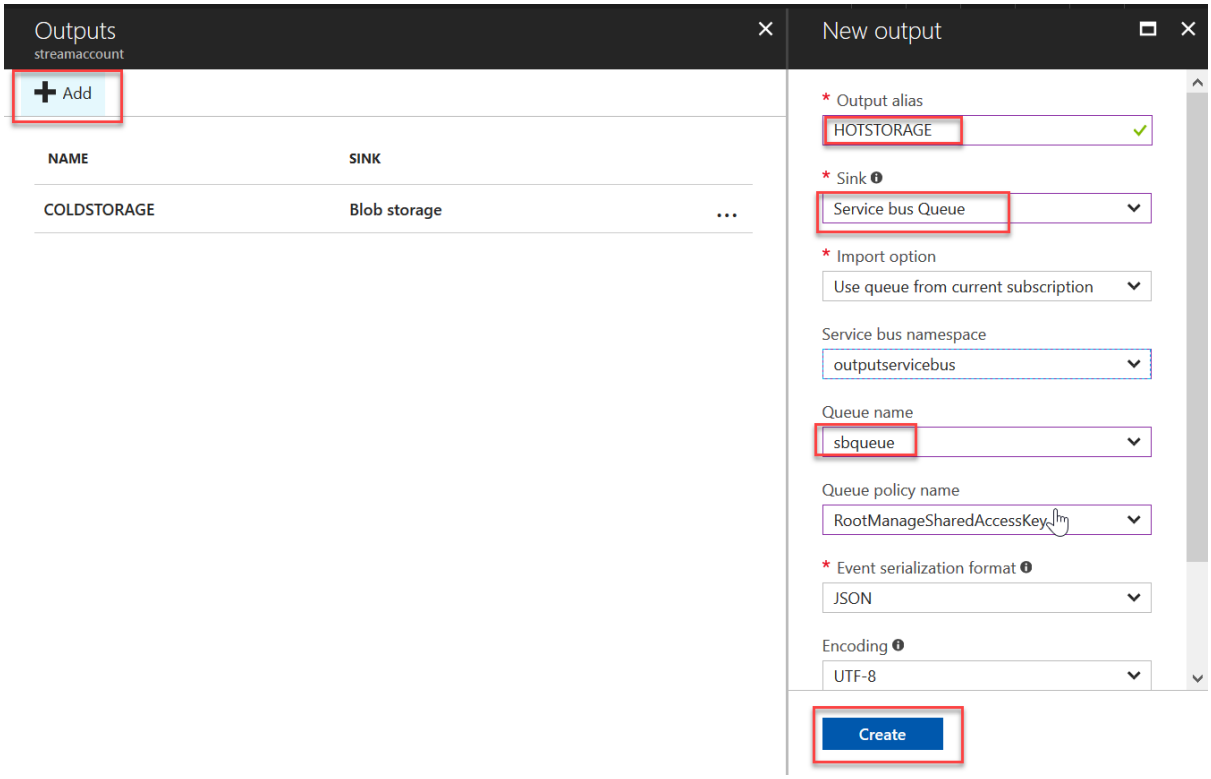
Now Add Output of Stream Analytics 1. Storage and 2. Service Bus Queue two Output one for cold storage another for hot storage.

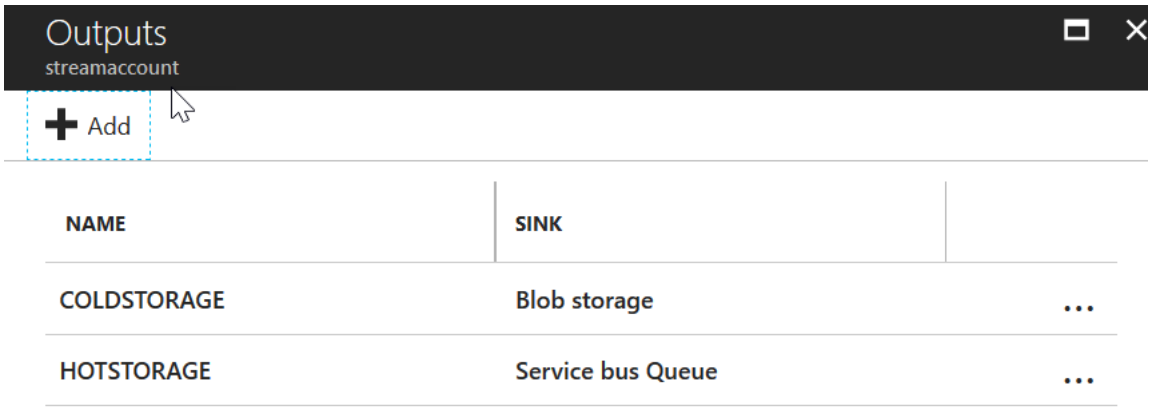



Add Cold Storage as a Azure Storage

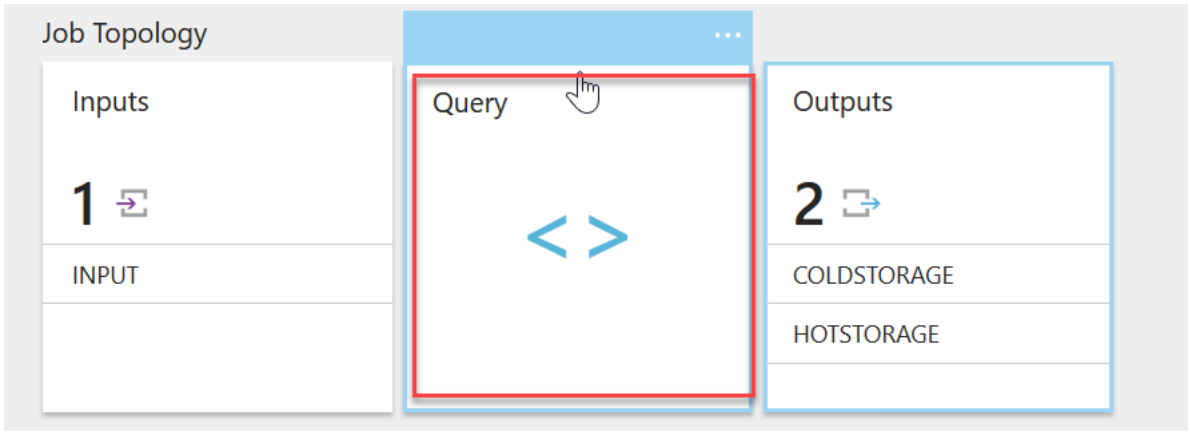



Now Add Hot Storage as a Service Bus



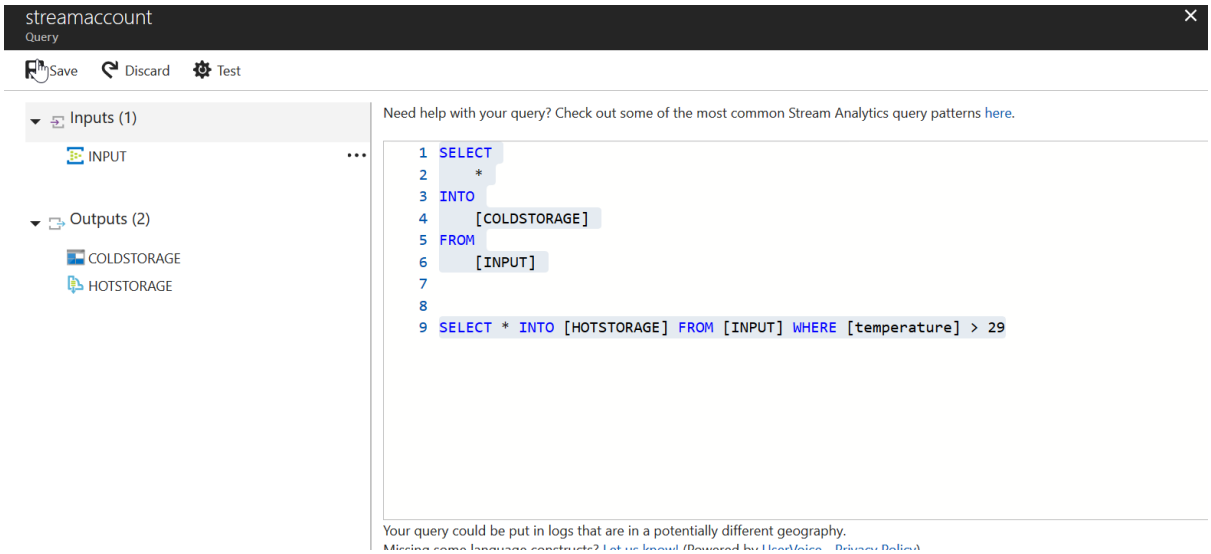

Now add Query in Stream Analytics




Type Query

```
SELECT
   *
INTO
   [COLDSTORAGE]
FROM
   [INPUT]
```

```
SELECT * INTO [HOTSTORAGE] FROM [INPUT] WHERE [temperature] > 29
```




All Temperature Data will go to Azure Storage but Temperature > 29 data will goes to Service Bus Queue.

Click on SAVE

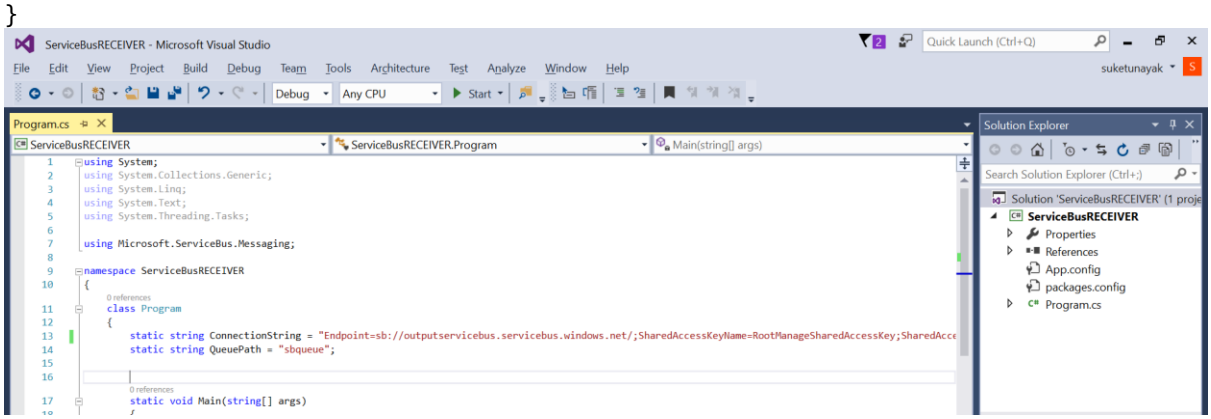
Now Create Visual Studio Console Application for Service Bus Queue to read Service Bus Hot Temperature Data.

In VS 2015 - > File -> New Project -> Console Application

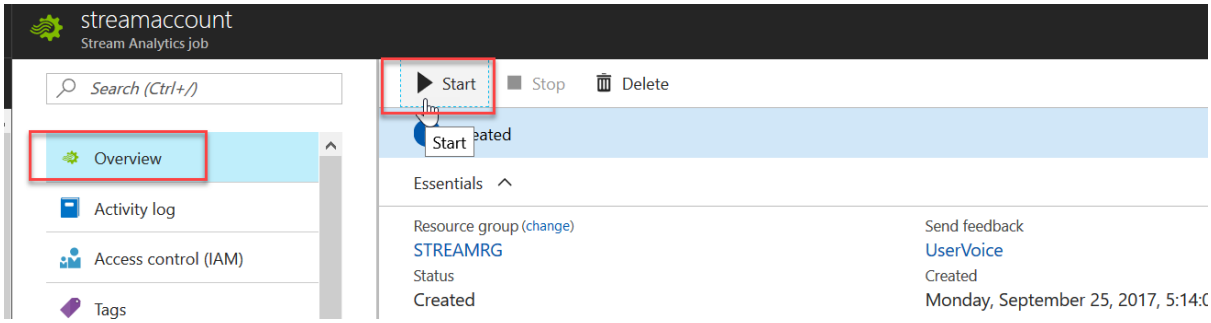
Add Below Code and replace your service bus connection string

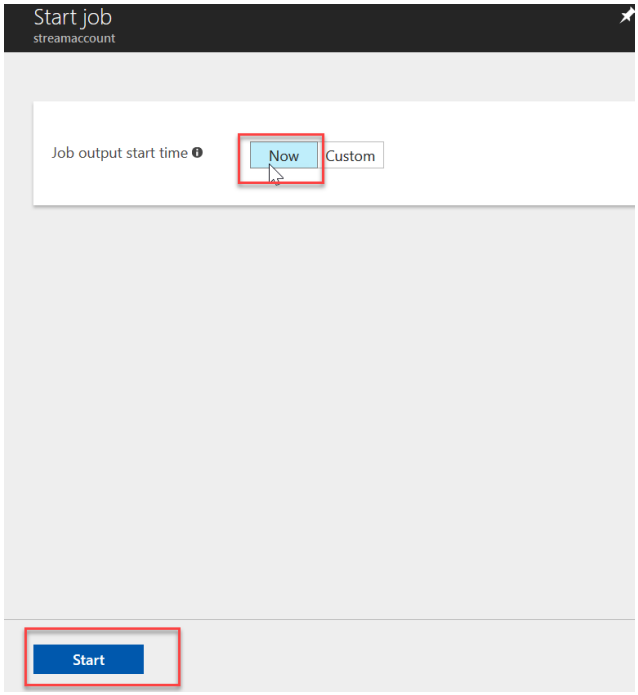
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using Microsoft.ServiceBus.Messaging;

namespace ServiceBusRECEIVER
{
    class Program
    {
```

```
        static string ConnectionString =
"Endpoint=sb://outputservicebus1.servicebus.windows.net/;SharedAccessKeyName=RootManag
eSharedAccessKey;SharedAccessKey=HFxNhHapH6Or819wFW4iP0bJ7AmEeiYJ4jneqJCaTxY=";
        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);
        }


    }
}
```




Now Click Start Button of Stream Analytics Service in Portal






Now Start Event Hub Console App to produce Temperature Data and it will keep sending that data to stream analytics and stream analytics will keep store data in storage blob and Temperature value > 29 data will be stored in Service Bus Queue. And also start service bus console application to read that Hot Temperature values.