IoT Hub实验手册

目录

[IoT Hub实验手册 1](#_Toc468361585)

[实验一：在Azure中创建IoT Hub 3](#_Toc468361586)

[1. 登陆Azure管理门户 3](#_Toc468361587)

[2. 新建IoT Hub 3](#_Toc468361588)

[3. 查看IoT Hub基本信息 5](#_Toc468361589)

[实验二：前端设备向IoT Hub发送消息 6](#_Toc468361590)

[1. 设备注册 6](#_Toc468361591)

[2. 创建模拟设备向IoT Hub发送消息 14](#_Toc468361592)

[3. 通过DeviceExplorer查看发送到IoT Hub的消息 20](#_Toc468361593)

[实验三：通过Azure Stream Analytics进行流数据处理 22](#_Toc468361594)

[1. 新建Stream Analytics Job 22](#_Toc468361595)

[2. 配置Stream Analytics Job输入 24](#_Toc468361596)

[3. 配置Stream Analytics Job输出 26](#_Toc468361597)

[4. 配置Stream Analytics Job查询 33](#_Toc468361598)

[实验四：通过Power BI进行数据展现 35](#_Toc468361599)

[1. 通过Power BI Desktop连接Azure Blob Storage 35](#_Toc468361600)

[2. 通过 Power BI创建报表 40](#_Toc468361601)

# 实验一：在Azure中创建IoT Hub

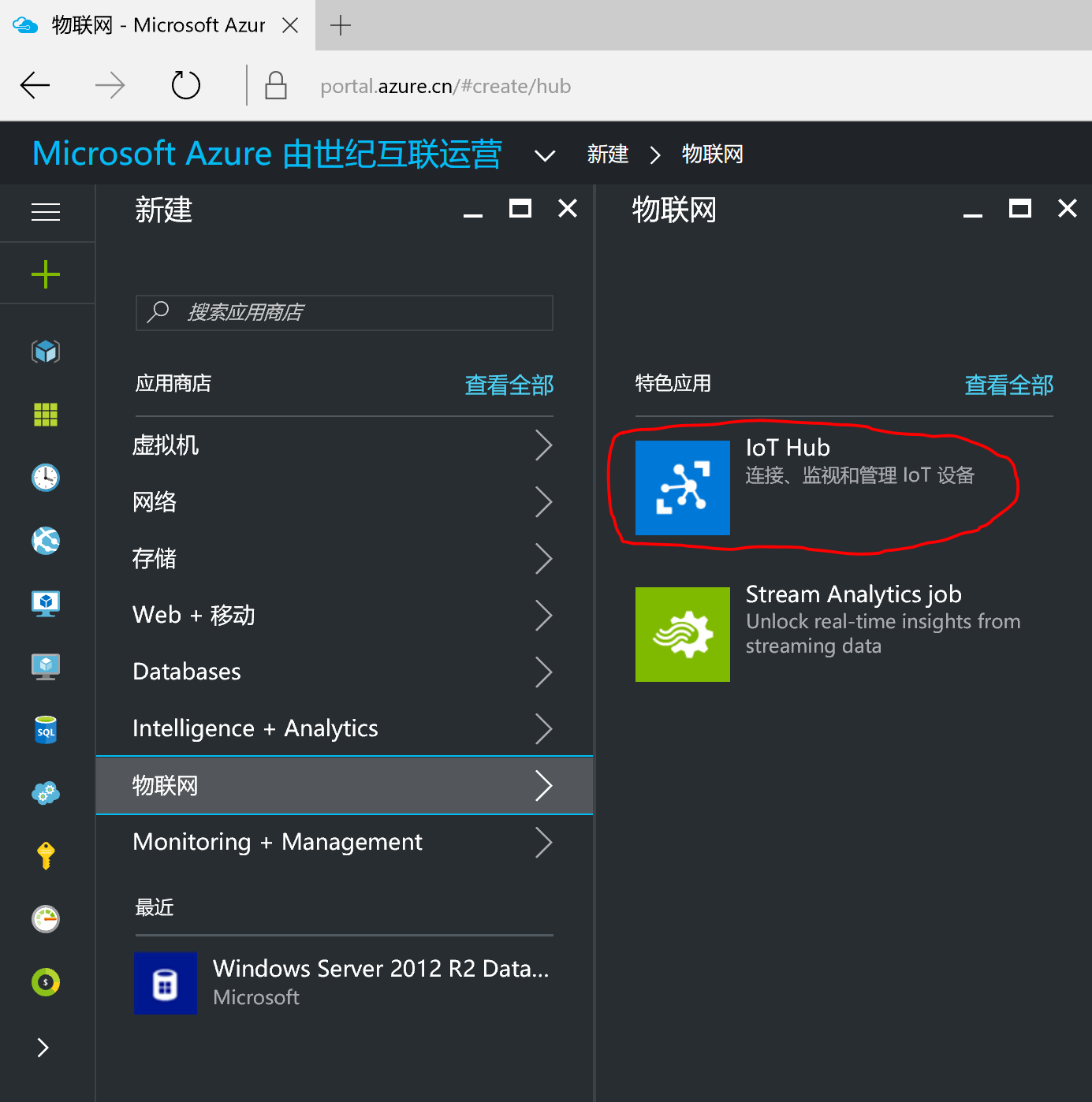
1. **登陆Azure管理门户**

在浏览器中输入<https://www.azure.cn/> 打开Azure中国区官方网站，然后点击页面右上角的“Azure门户预览”：



1. **新建IoT Hub**

输入账户和密码，登陆到管理门户。点击左上角“+新建”按钮，然后依次选择“物联网”，“IoT Hub”，如下图所示：

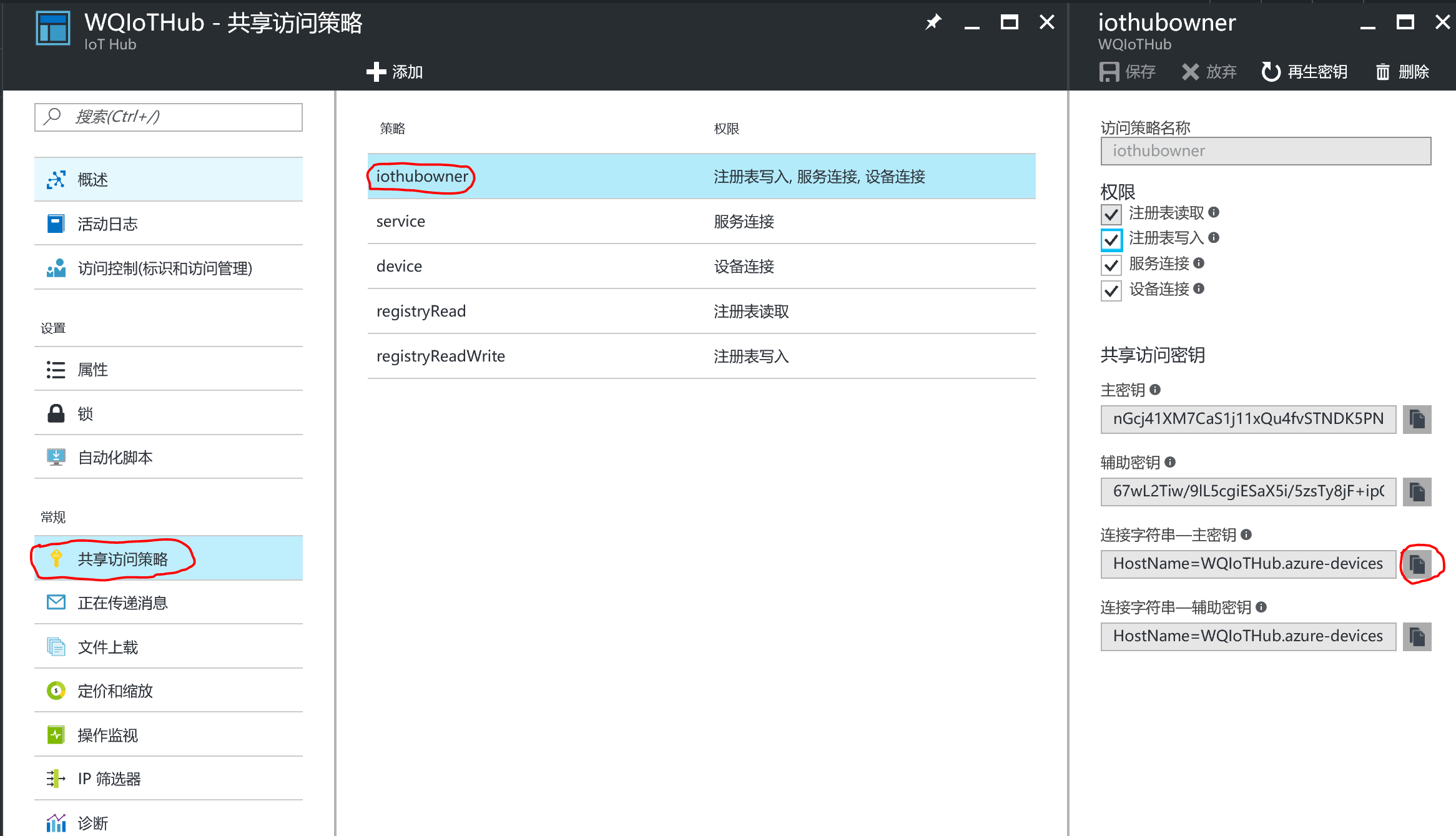


为IoT Hub和新建资源组输入名称，将定价和缩放级别调整为”S2 – 标准”，然后勾选“固定到仪表板”选项，其它配置项保留默认值，然后点击“创建”，如下图所示：



1. **查看IoT Hub基本信息**

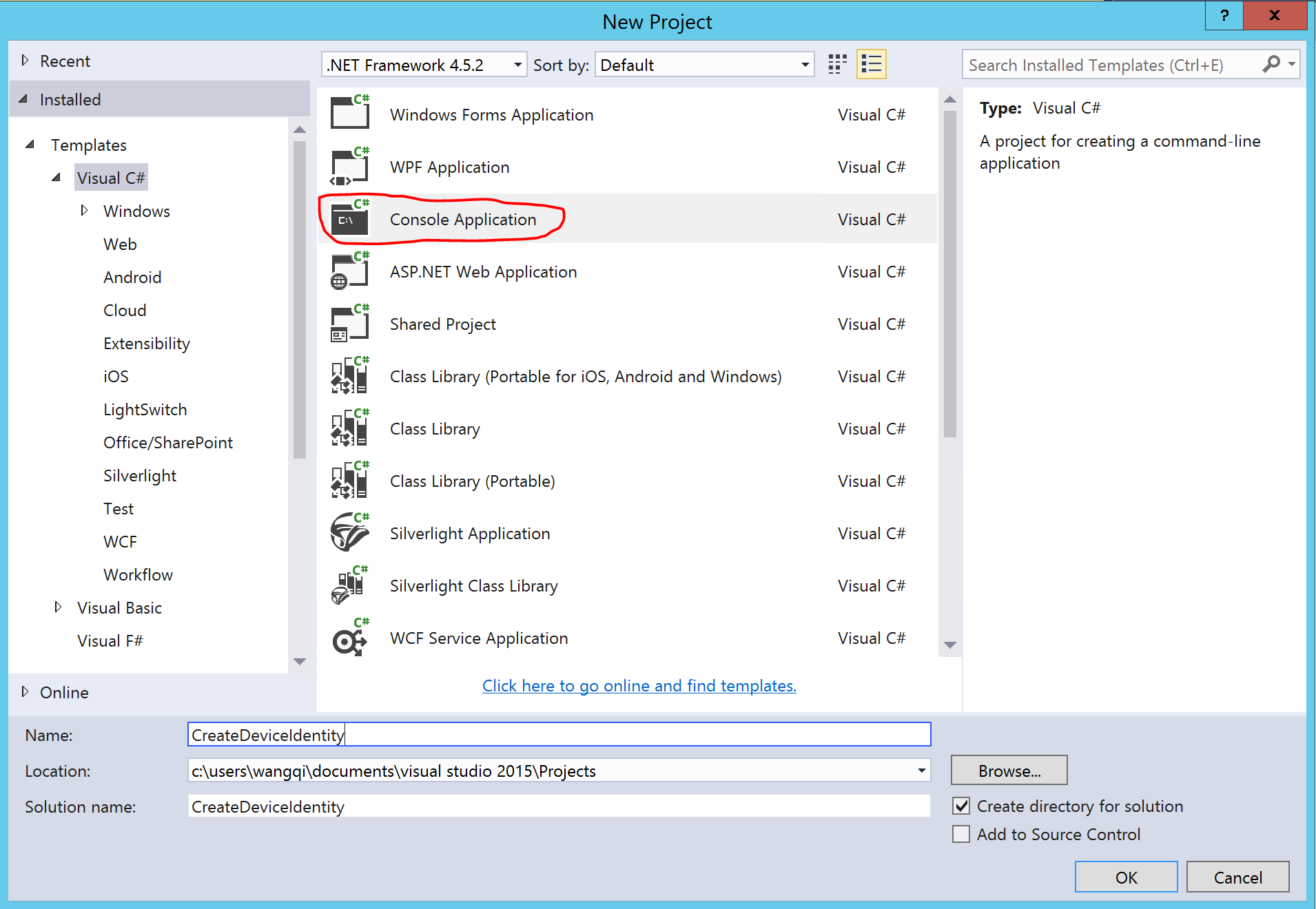
创建完成后在仪表板中点击刚刚创建好的IoT Hub，然后点击“共享访问策略”，选中iothubowner，然后在屏幕右侧的“连接字符串-主密钥”旁边点击复制按钮，如下图所示：



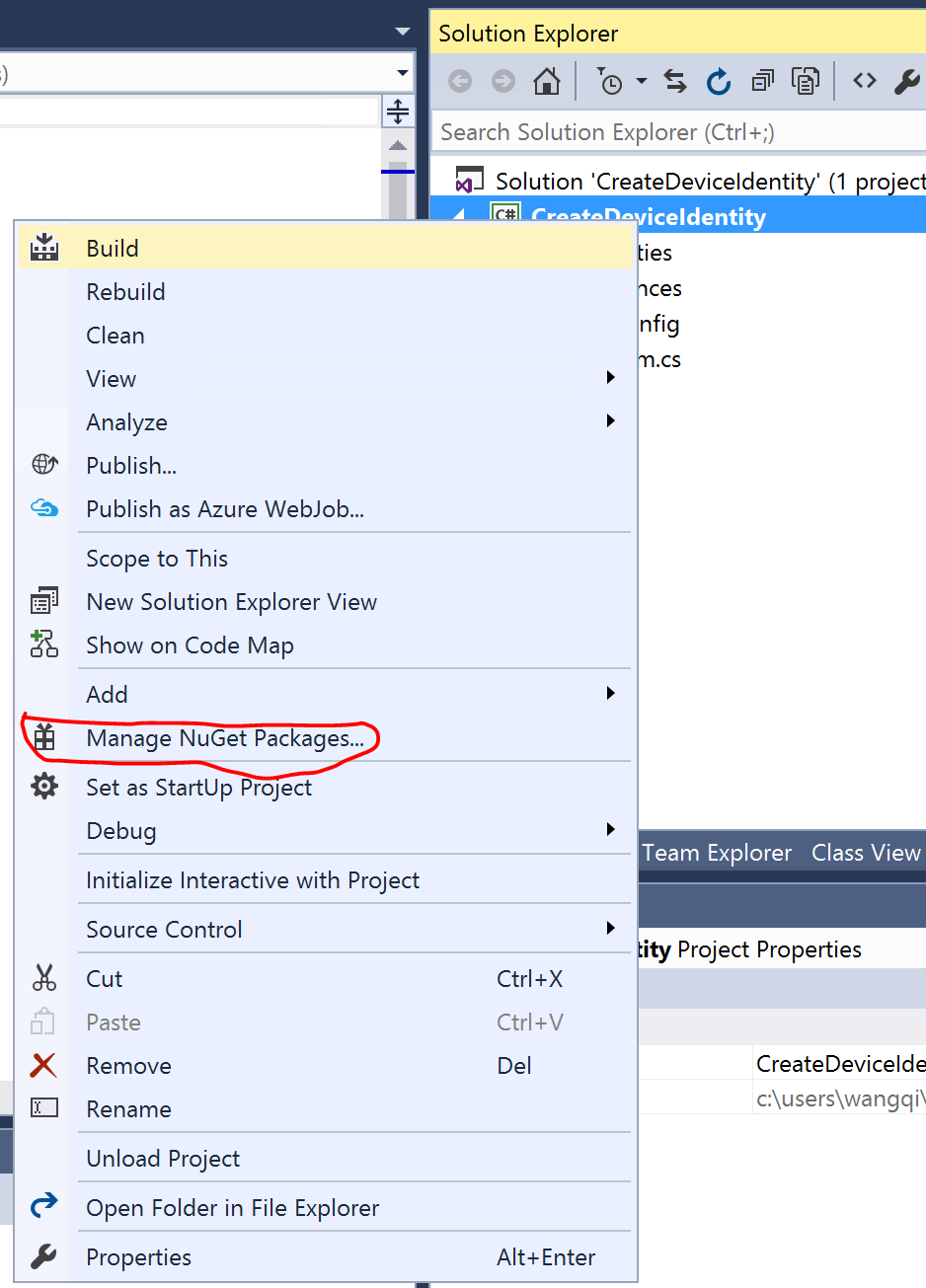
# 实验二：前端设备向IoT Hub发送消息

1. **设备注册**

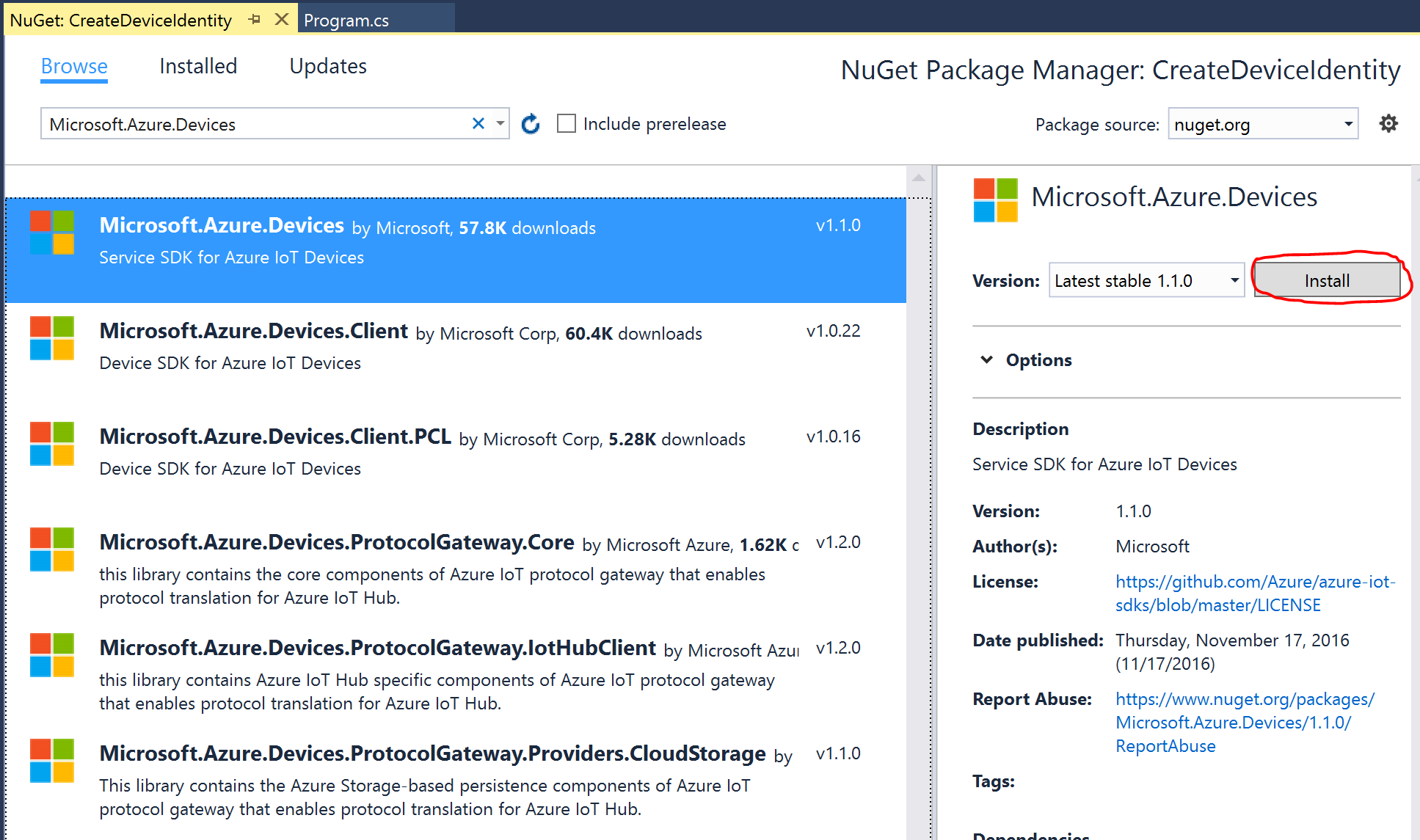
打开Visual Studio 2015，依次点击”File”, “New”, “Project”，然后Visual C#项目模板中选择“Console Application”，并为新创建的项目起一个项目名称，然后点击OK。如下图所示：



右键点击新创建好的项目，然后点击”Manage NuGet Packages”，如下图所示：



在左上角选中“Browse”，然后在搜索框中搜索“Microsoft.Azure.Devices”,并点击”Install”进行安装，如下图所示：



双击打开Program.cs文件，在代码上方添加：

using Microsoft.Azure.Devices;

using Microsoft.Azure.Devices.Common.Exceptions;

在Program类中添加如下代码，并将{iot hub connection string}替换为实验一中复制的IoT Hub的连接字符串：

static RegistryManager registryManager;

static string connectionString = "{iot hub connection string}";

在Program类中添加如下方法，该方法将在IoT Hub中注册ID为myFirstDevice的设备并返回该设备的主密钥，如果该设备ID已经被注册，则直接返回主密钥。该密钥用于将设备连接到IoT Hub：

private static async Task AddDeviceAsync()

{

string deviceId = "myFirstDevice";

Device device;

try

{

device = await registryManager.AddDeviceAsync(new Device(deviceId));

}

catch (DeviceAlreadyExistsException)

{

device = await registryManager.GetDeviceAsync(deviceId);

}

Console.WriteLine("Generated device key: {0}", device.Authentication.SymmetricKey.PrimaryKey);

}

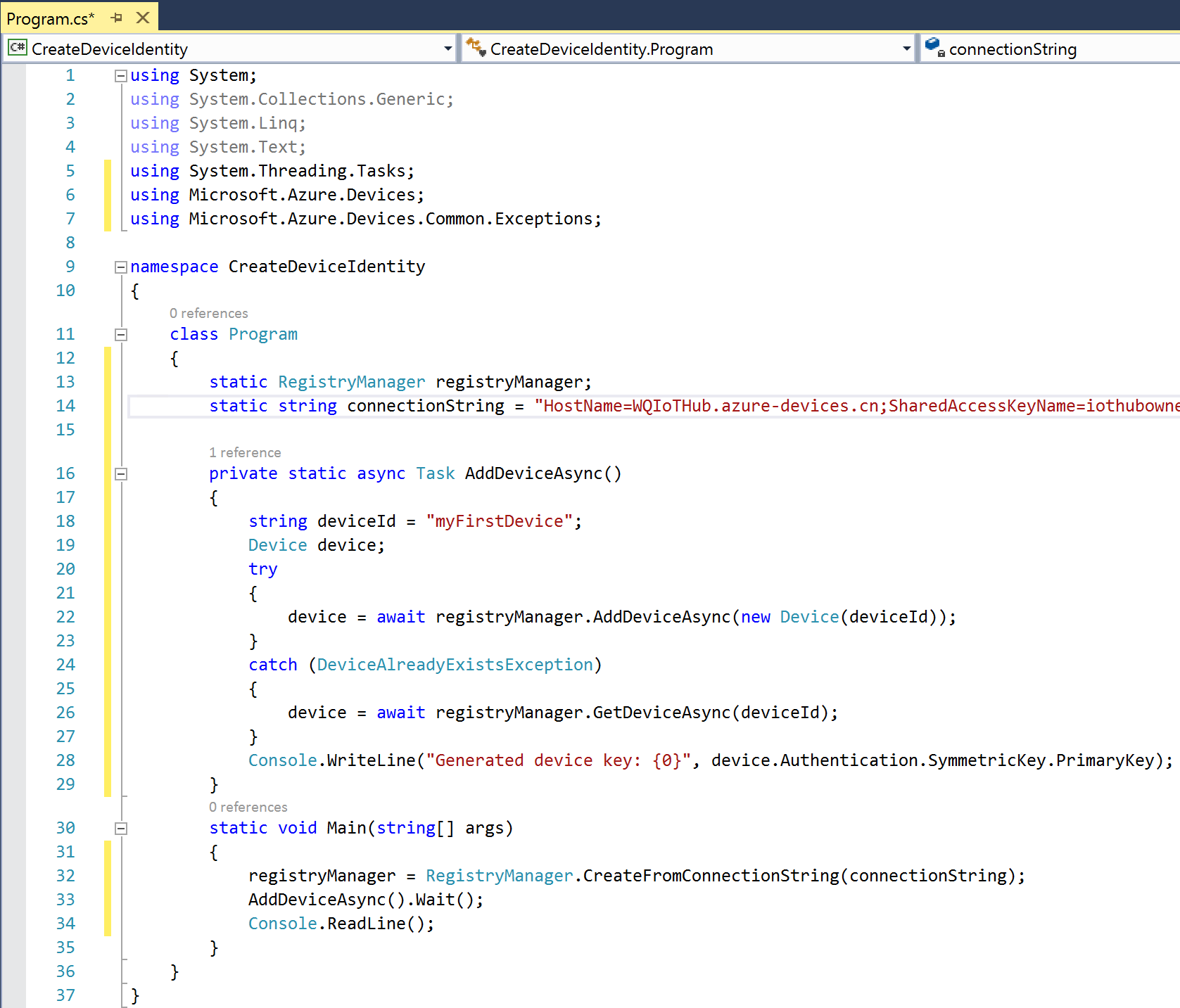
最后，在Main函数中添加如下代码：

registryManager = RegistryManager.CreateFromConnectionString(connectionString);

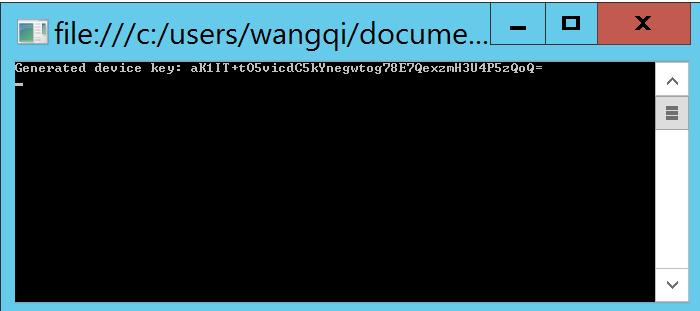
AddDeviceAsync().Wait();

Console.ReadLine();

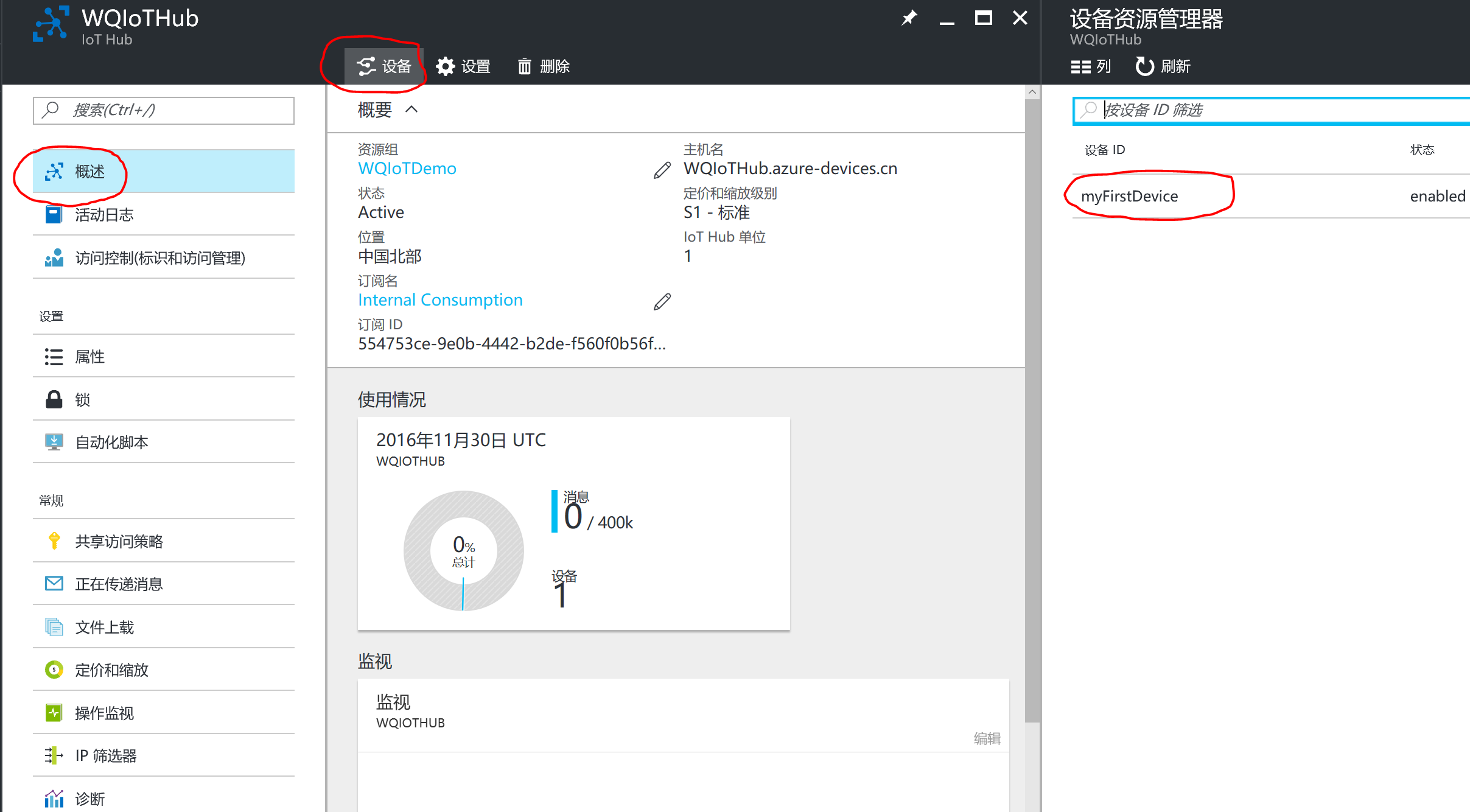
完成后的代码如下图所示：



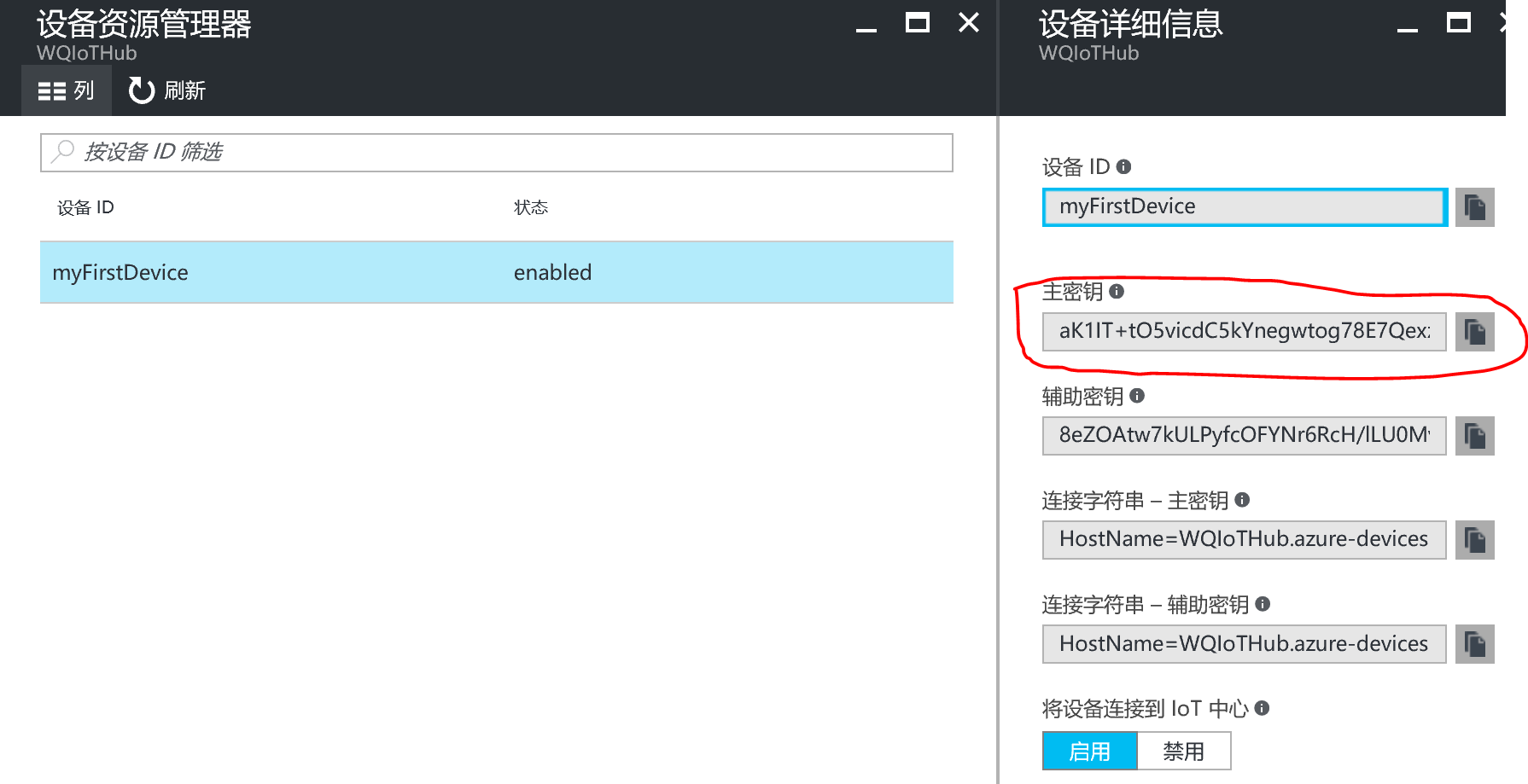
在Visual Studio中点击”Start”按钮运行该程序：



返回浏览器中，在IoT Hub的管理界面中依次点击“概述”，“设备”，可以看到刚刚注册过的名为“myFirstDevice”的设备，其状态为”enabled”：

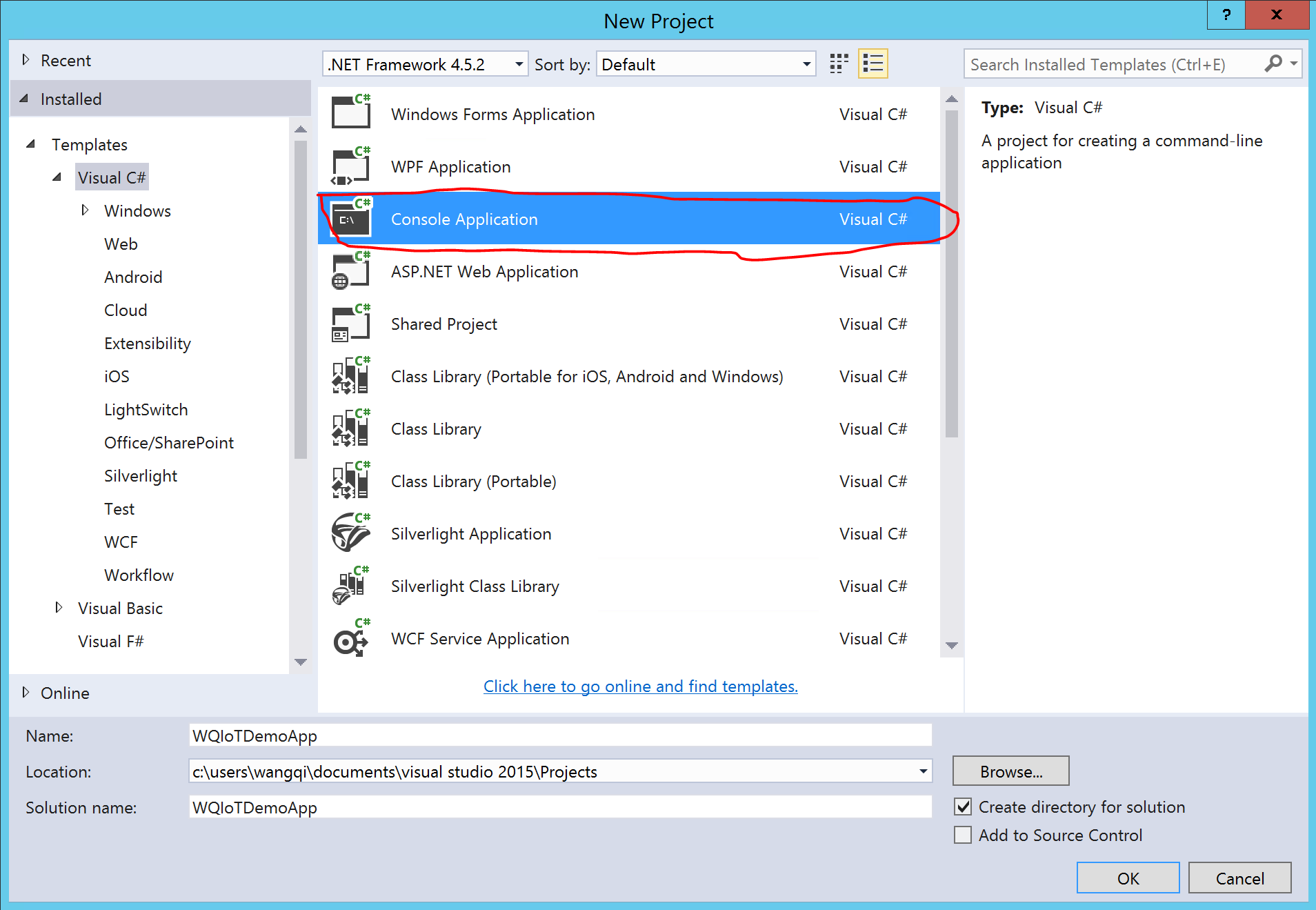


点击”myFirstDevice”，并复制其主密钥，如下图所示：

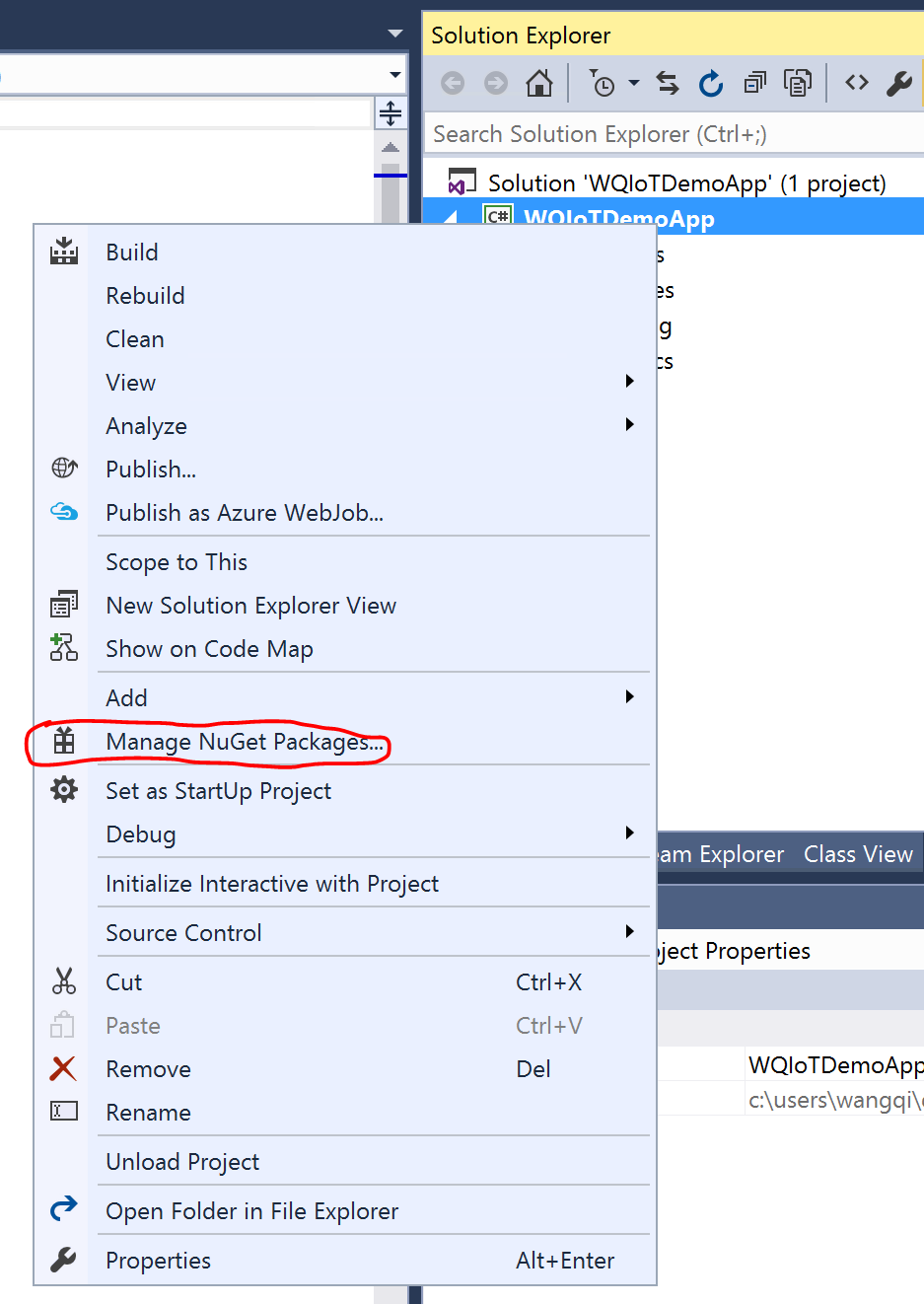


1. **创建模拟设备向IoT Hub发送消息**

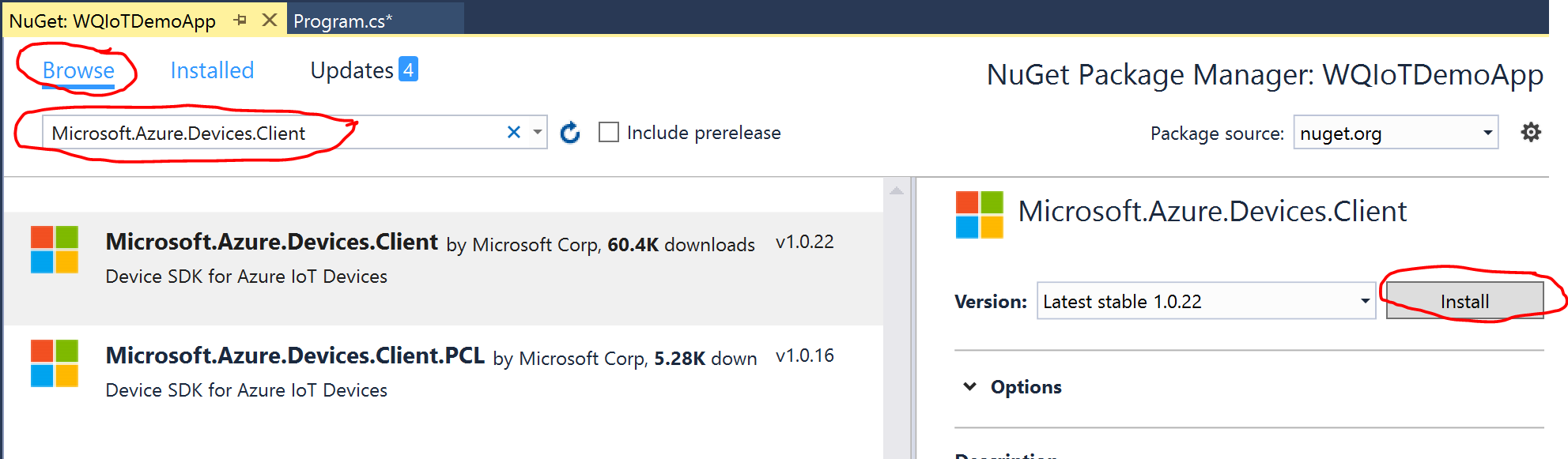
打开Visual Studio 2015，依次点击”File”, “New”, “Project”，然后Visual C#项目模板中选择“Console Application”，并为新创建的项目起一个项目名称，然后点击OK。如下图所示：



右键点击新创建好的项目，然后点击”Manage NuGet Packages”，如下图所示：



在左上角选中“Browse”，然后在搜索框中搜索“Microsoft.Azure.Devices.Client”,并点击”Install”进行安装，如下图所示：



双击打开Program.cs文件，在代码上方添加：

using Microsoft.Azure.Devices.Client;

using Newtonsoft.Json;

在Program类中添加如下代码，并将{iot hub hostname}替换为IoT Hub的主机名，将{device key}替换为”myFirstDevice”的主密钥：

static DeviceClient deviceClient;

static string iotHubUri = "{iot hub hostname}";

static string deviceKey = "{device key}";

在Program类中添加如下方法：

private static async void SendDeviceToCloudMessagesAsync()

{

double avgWindSpeed = 10; // m/s

double avgTemperature = 10;

double avgHumidity = 50;

Random rand = new Random();

while (true)

{

double currentWindSpeed = avgWindSpeed + rand.NextDouble() \* 4 - 2;

double currentTemperature = avgTemperature + rand.NextDouble() \* 4 - 2;

double currentHumidity = avgHumidity + rand.NextDouble() \* 4 - 2;

var telemetryDataPoint = new

{

deviceId = "myFirstDevice",

windSpeed = currentWindSpeed,

Temperature = currentTemperature,

Humidity = currentHumidity

};

var messageString = JsonConvert.SerializeObject(telemetryDataPoint);

var message = new Message(Encoding.ASCII.GetBytes(messageString));

await deviceClient.SendEventAsync(message);

Console.WriteLine("{0} > Sending message: {1}", DateTime.Now, messageString);

Task.Delay(1000).Wait();

}

}

在Main()函数中添加如下代码：

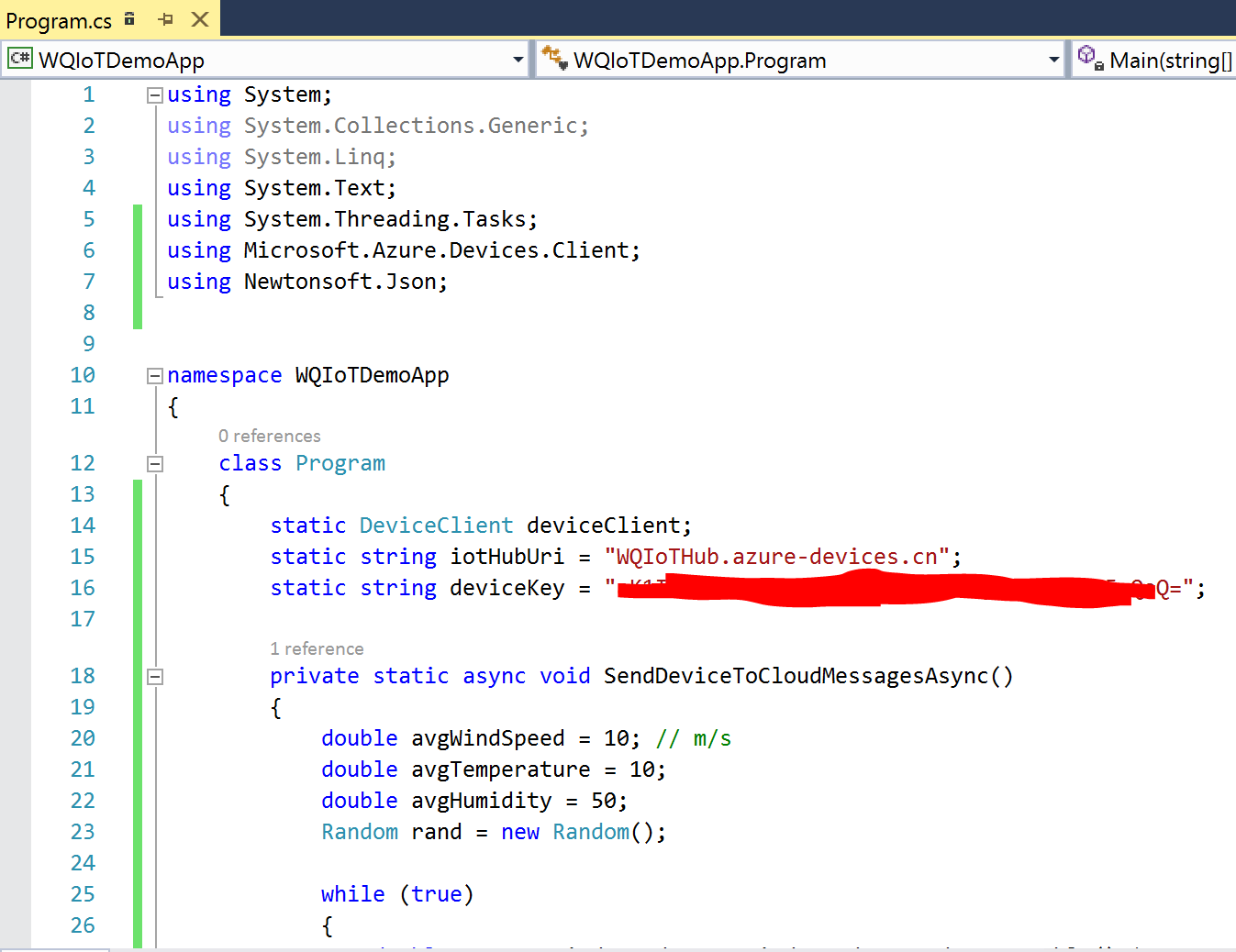
Console.WriteLine("Simulated device\n");

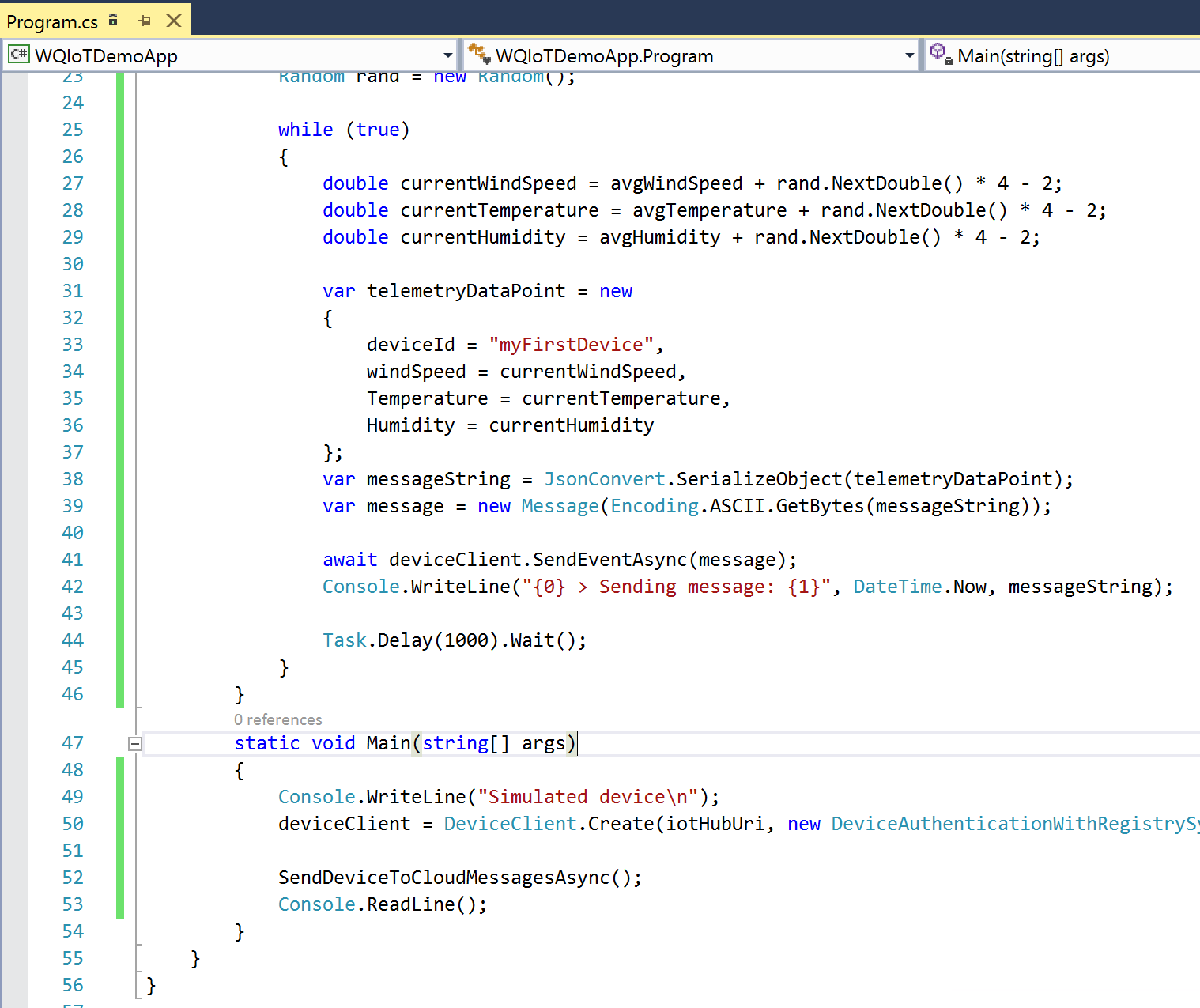
deviceClient = DeviceClient.Create(iotHubUri, new DeviceAuthenticationWithRegistrySymmetricKey("myFirstDevice", deviceKey));

SendDeviceToCloudMessagesAsync();

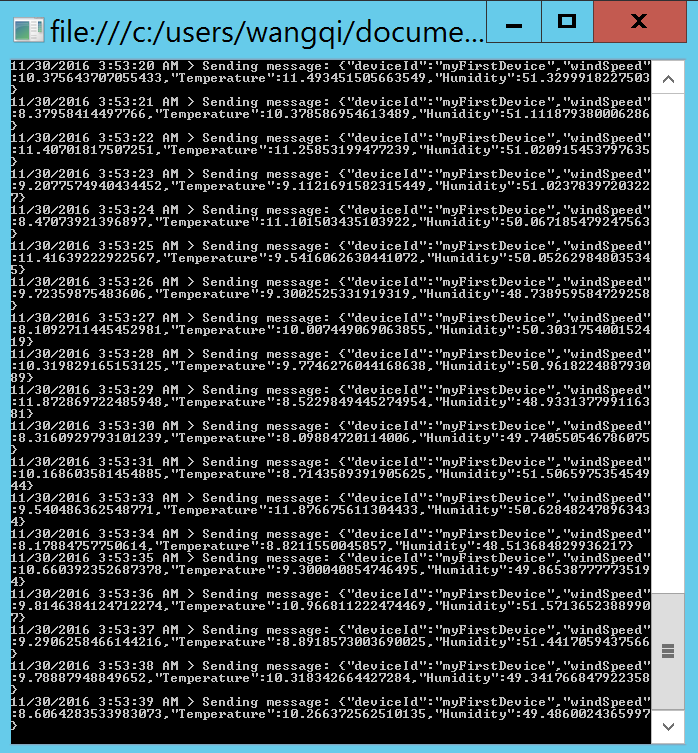
Console.ReadLine();

完成后的代码如下图所示：

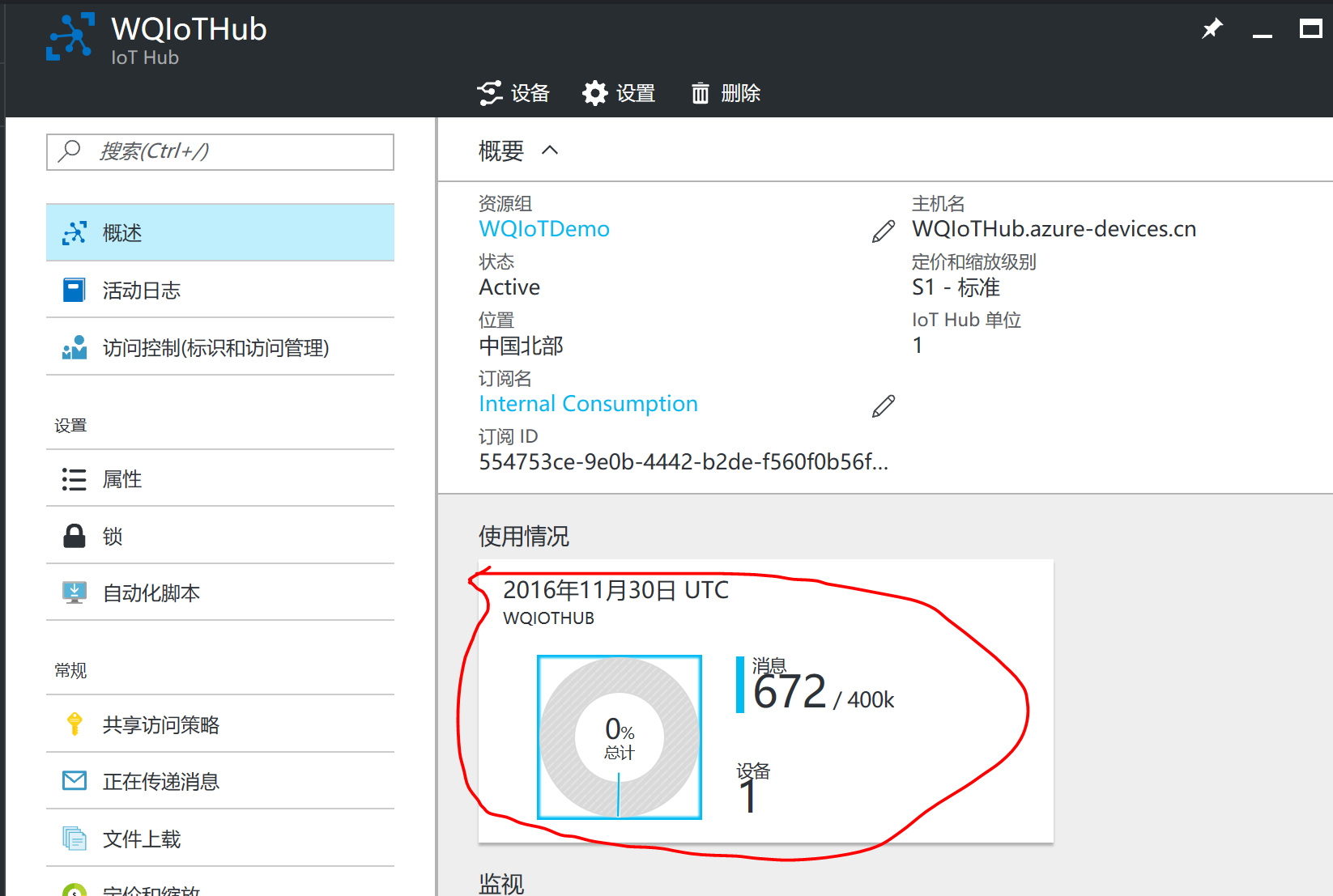




在Visual Studio中点击”Start”按钮，启动程序，可以看到模拟程序每秒向IoT Hub发送一组消息，消息中包含随机生成的风速、温度和湿度的数据，如下图所示：



在Azure管理界面中，可以看到IoT Hub已经接收到消息：

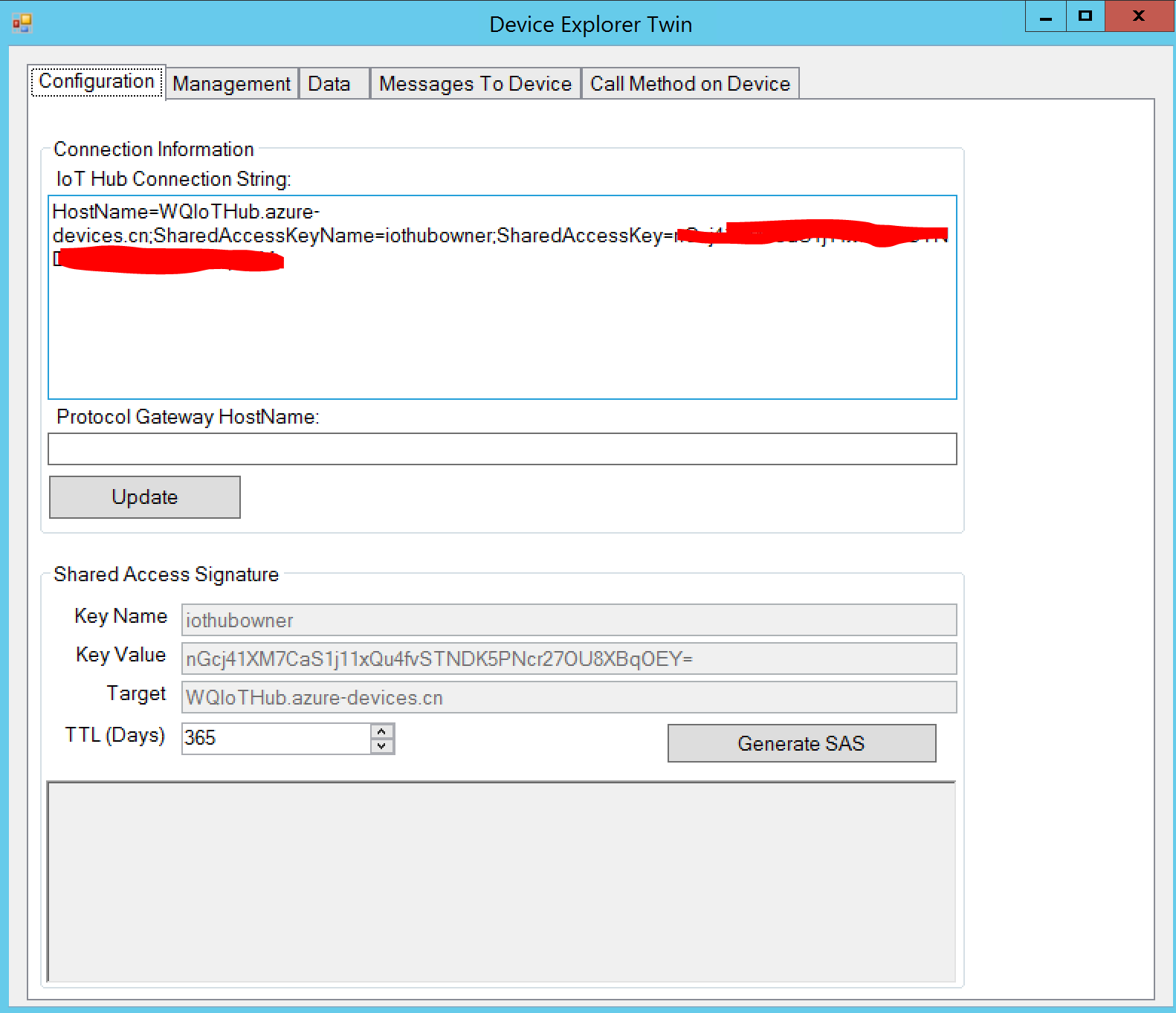


1. **通过DeviceExplorer查看发送到IoT Hub的消息**

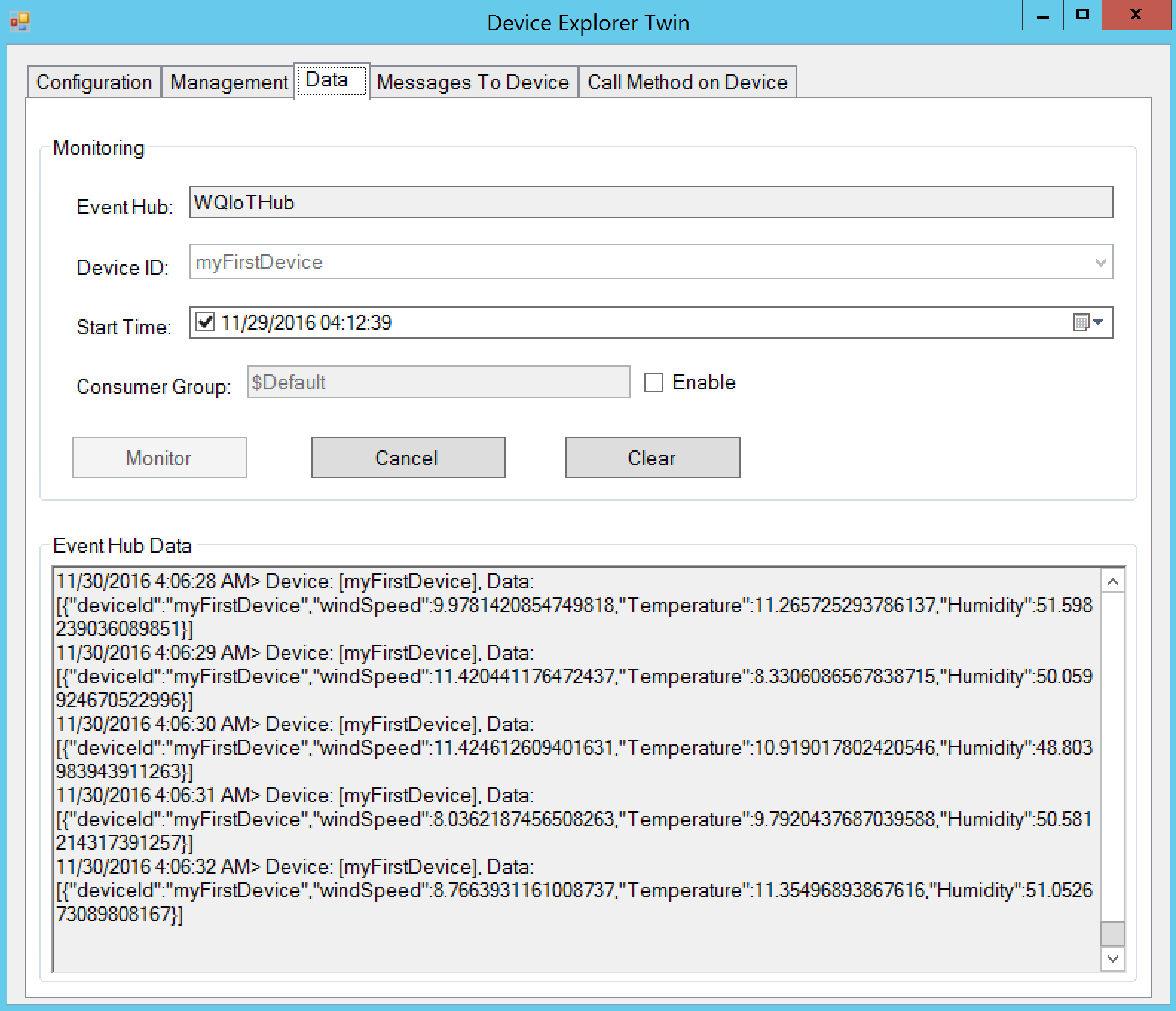
通过如下链接下载并安装DeviceExplorer：

<https://github.com/Azure/azure-iot-sdks/releases/download/2016-11-17/SetupDeviceExplorer.msi>

打开DeviceExplorer，在Configuration选项卡中数据IoT Hub的连接字符串，然后点击Update, 如下图所示：



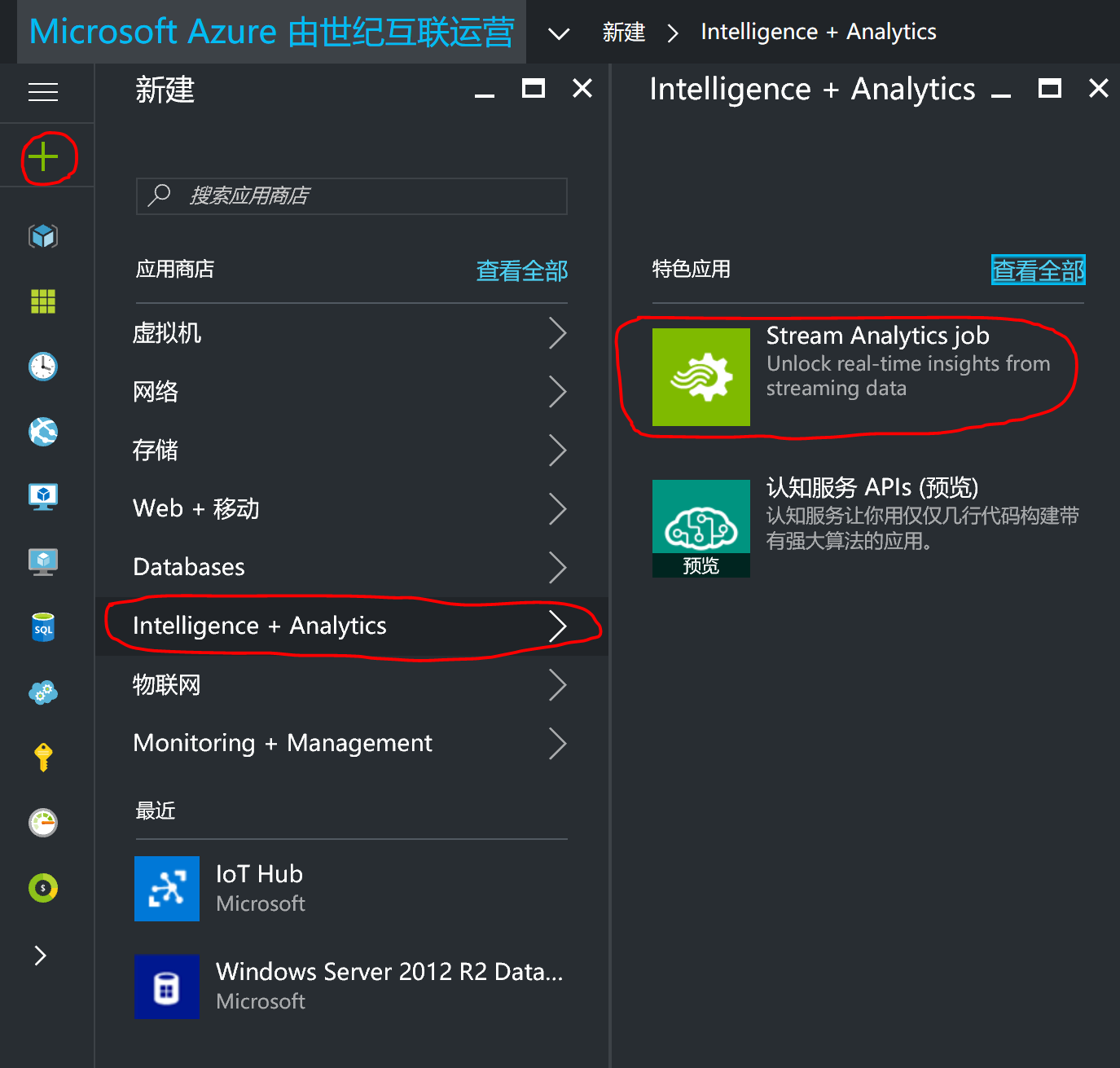
点击”Data”选项卡，可以看到之前发送到IoT Hub的消息，如下图所示：



# 实验三：通过Azure Stream Analytics进行流数据处理

1. **新建Stream Analytics Job**

打开Azure管理门户，点击左上角的+号按钮，然后依次点击”Intelligence + Analytics”, “Stream Analytics Job”，如下图所示：



为流分析作业输入名称，并选择使用现有资源组，然后点击“创建”按钮，如下图所示：



1. **配置Stream Analytics Job输入**

等待Stream Analytics Job创建好以后，点击“输入”，如下图所示：



点击“添加”按钮，然后输入如下信息：

输入别名 input

源类型 数据流

源 IoT中心

IoT中心 *{IoT Hub的名称}*

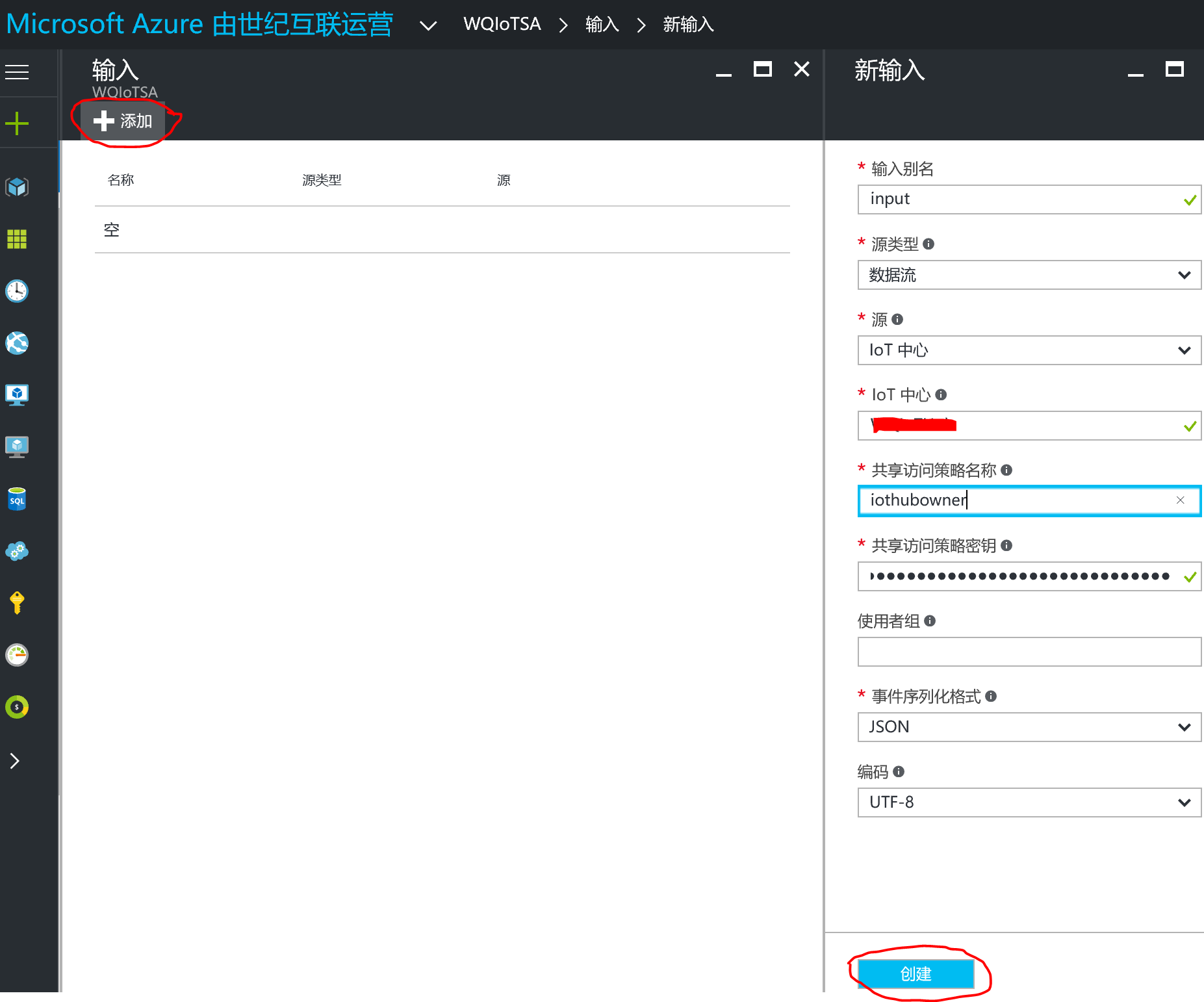
共享访问策略名称 iothubowner

共享访问策略密钥 *{iothubowner对应的主密钥}*

事件序列化格式 JSON

编码 UTF-8

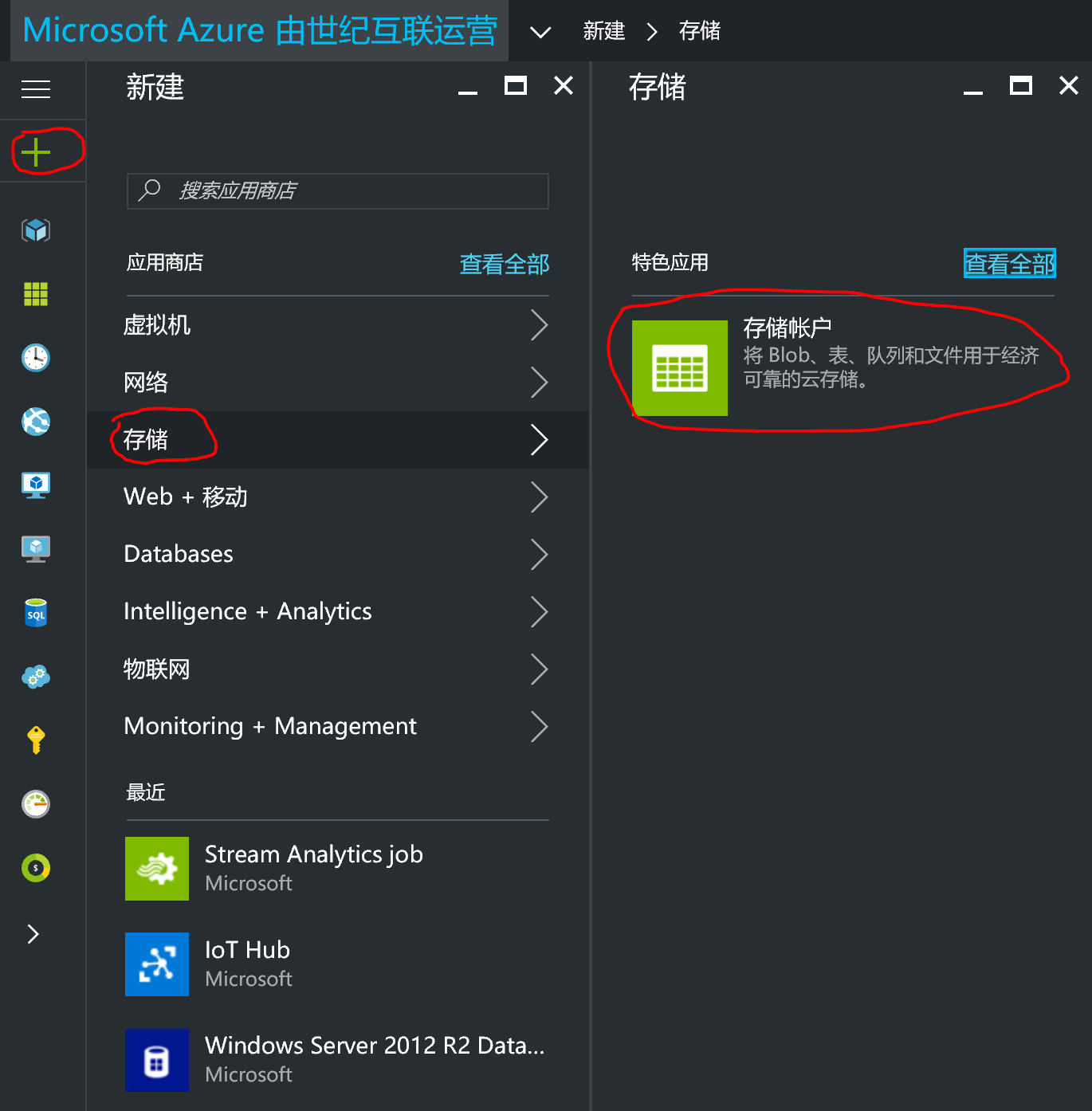
然后点击“创建”，如下图所示：



1. **配置Stream Analytics Job输出**

接下来我们要为Stream Analytics Job创建输出。此次动手实验中我们将数据输出到Azure Blob Storage中，因此先要配置一个Azure Blob Storage的存储账号。

在Azure门户中，点击左上角的“新建”按钮，然后依次点击“存储”，“存储账户”，如下图所示：



存储账户的信息按如下内容输入：

名称 *{输入一个账户名}*

部署模型 Resource Manager

账户类型 Blob存储

性能 标准

复制 本地冗余存储(LRS)

访问层 热

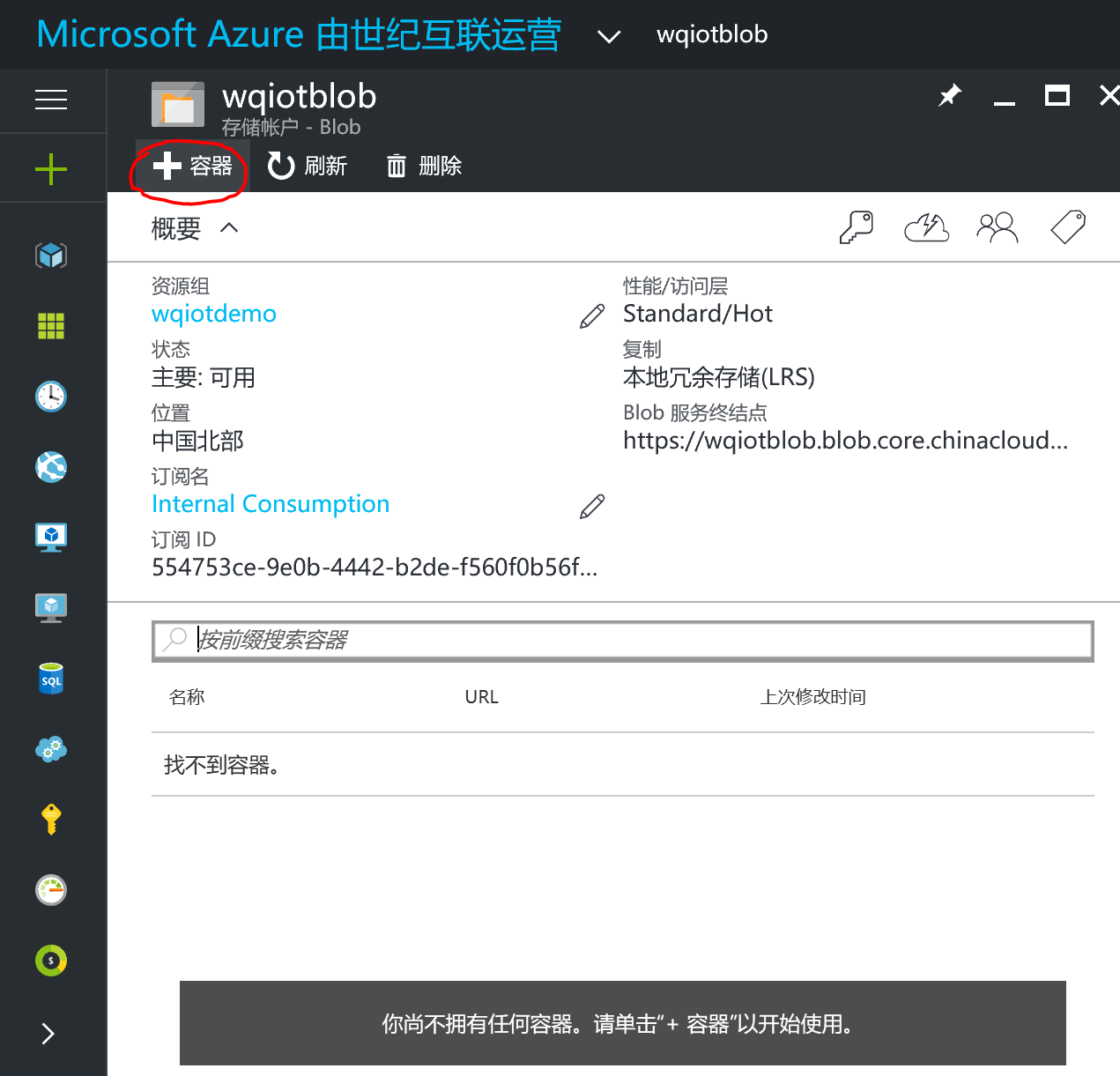
存储服务加密 已禁用

资源组 使用现有

然后点击“创建”按钮，如下图所示：



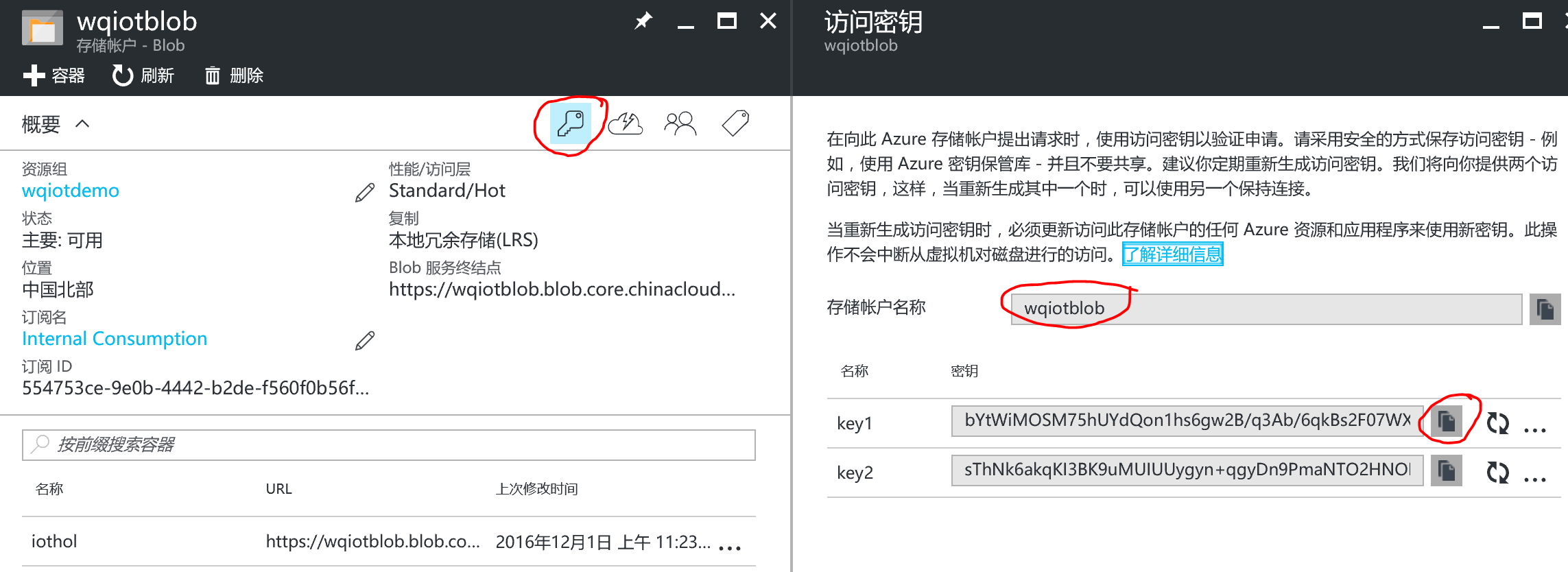
在新创建好的存储账户中，点击“+容器”按钮，新建一个容器



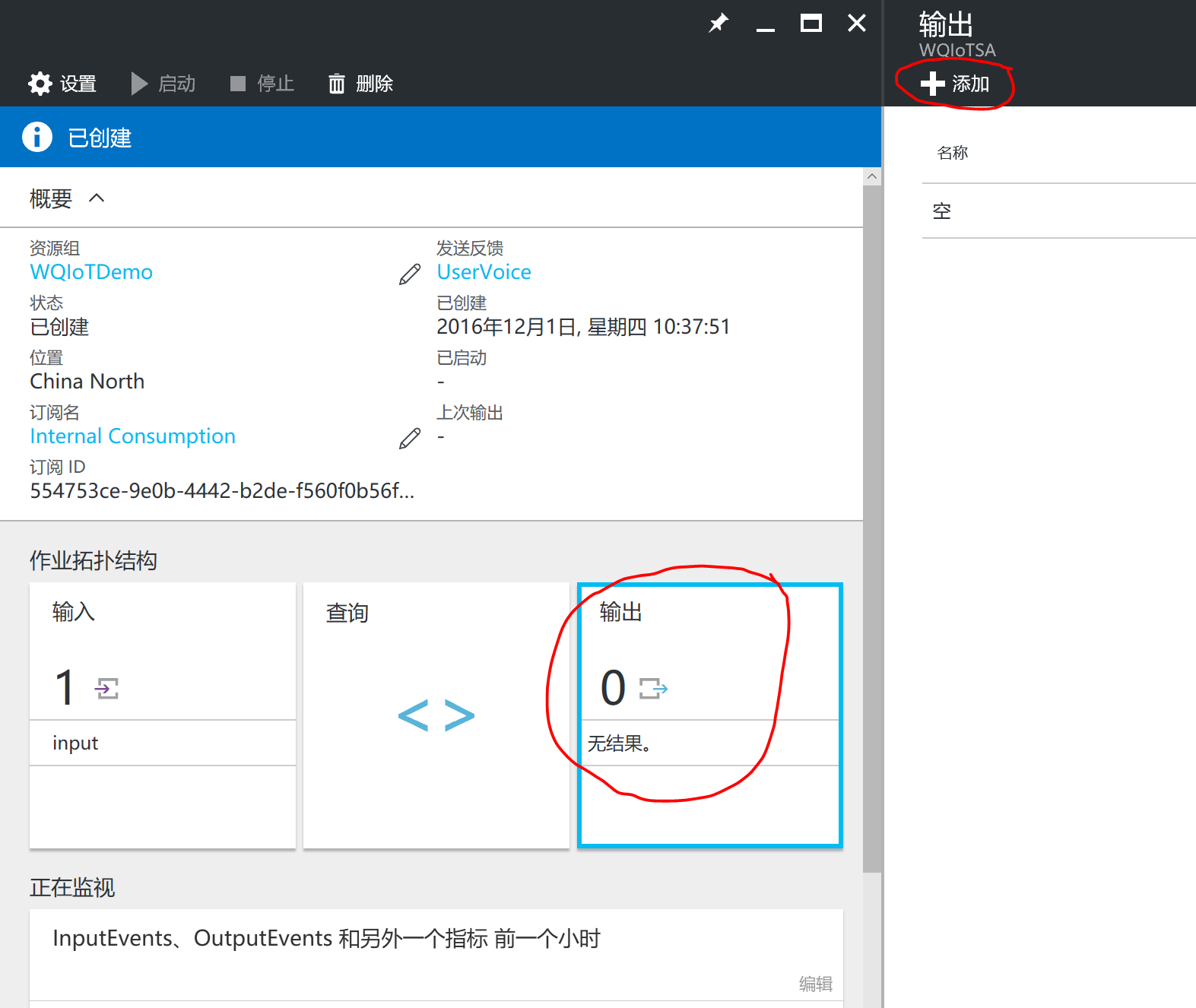
为容器起一个名称，访问类型选择“Blob”，然后点击“创建”，如下图所示：



在存储账号的配置界面中，点击“密钥”图标，然后记录下存储账户名称和密钥，如下图所示：



关闭存储账户配置界面，在Azure管理门户中打开之前所创建的Stream Analytics Job，点击“输出”，然后点击“+添加”按钮，如下图所示：



输入如下信息，然后点击“创建“：

输出别名 output

接收器 Blob存储

存储账户 *{刚创建的存储账户}*

存储账户密钥 *{存储账户的密钥}*

容器 *{存储账户中创建的容器的名称}*

事件序列化格式 CSV

分隔符 逗号(,)

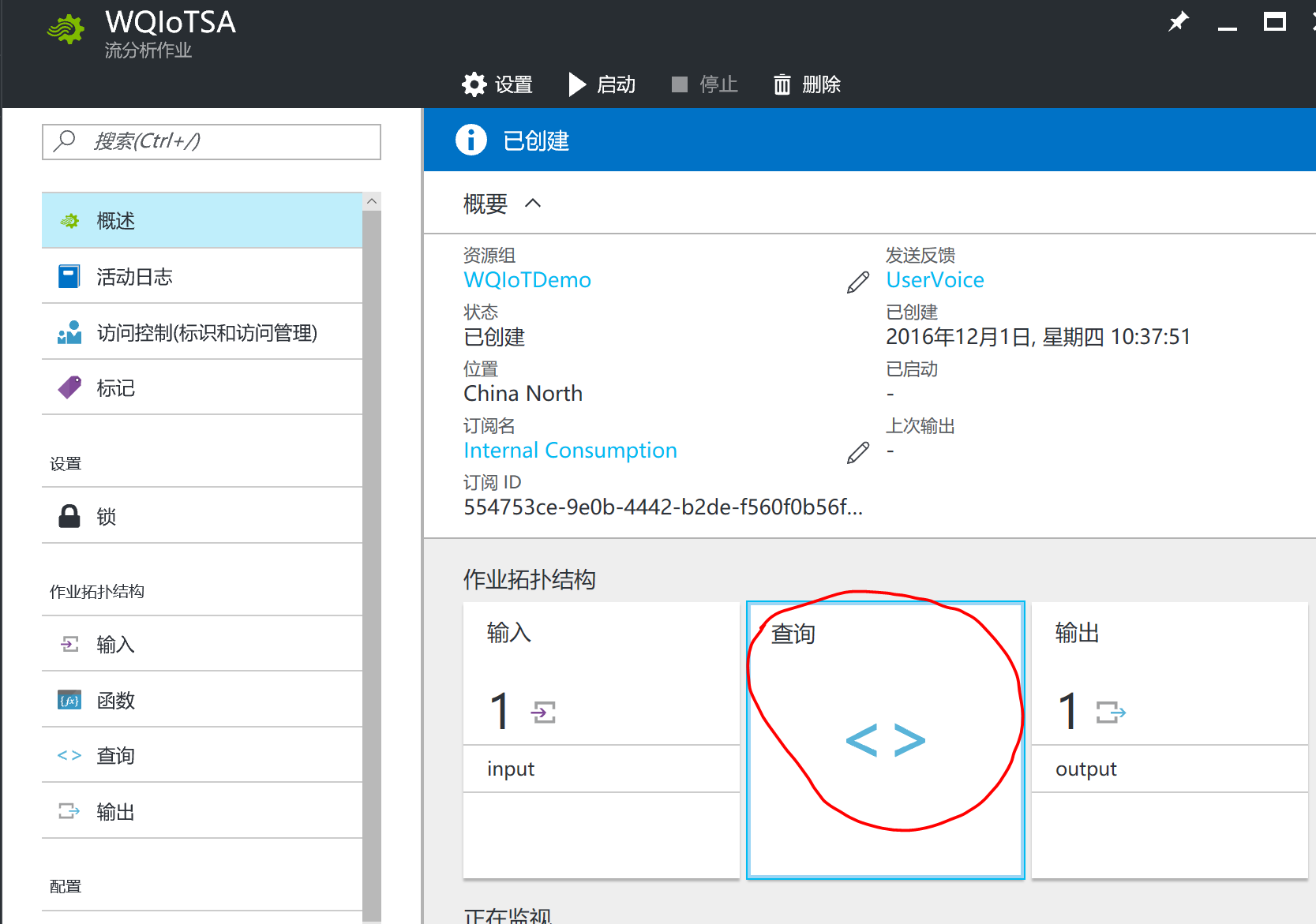
编码 UTF-8

如下图所示：



1. **配置Stream Analytics Job查询**

在创建好的Stream Analytics Job中点击“查询“，如下图所示：



然后输入如下查询语句，并点击“保存“：

Select

system.TimeStamp as Time,

Avg(windSpeed) as WindSpeed,

Avg(Temperature) as Temperature,

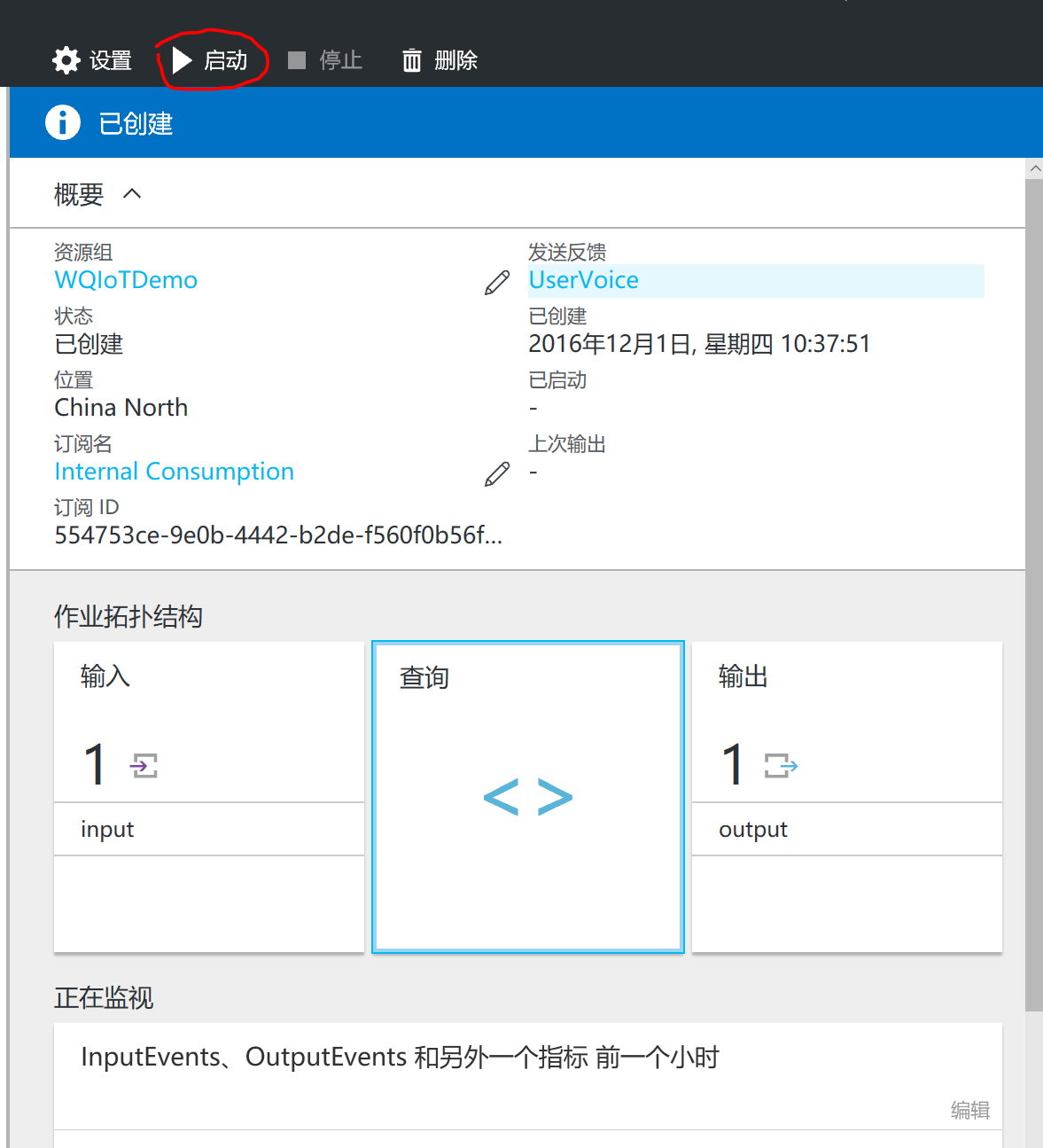
Avg(Humidity) as Humidity

From input

Group by deviceid,

TumblingWindow(Second, 10)

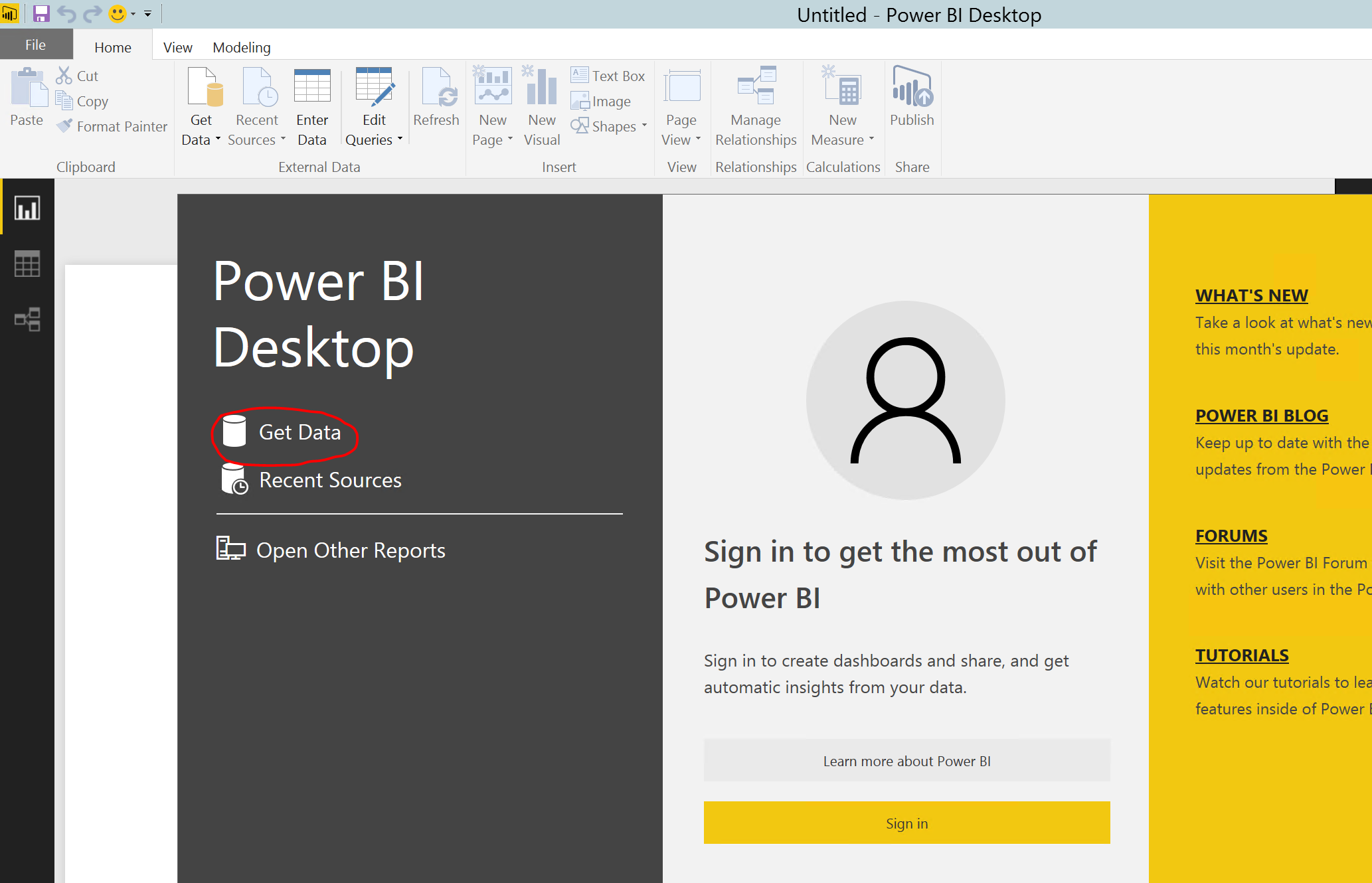
在Stream Analytics Job界面中，点击“启动‘，如下图所示：



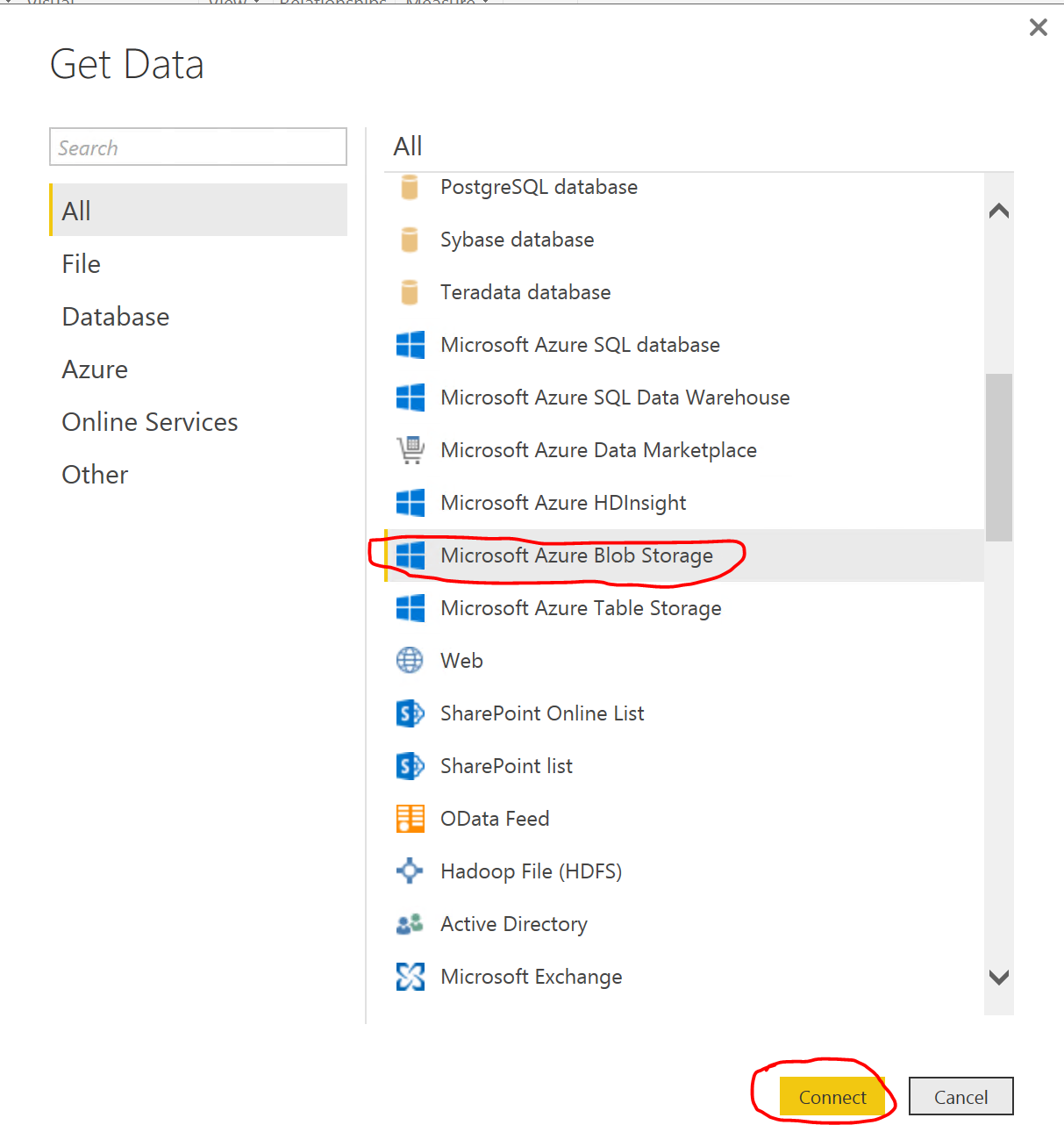
# 实验四：通过Power BI进行数据展现

1. **通过Power BI Desktop连接Azure Blob Storage**

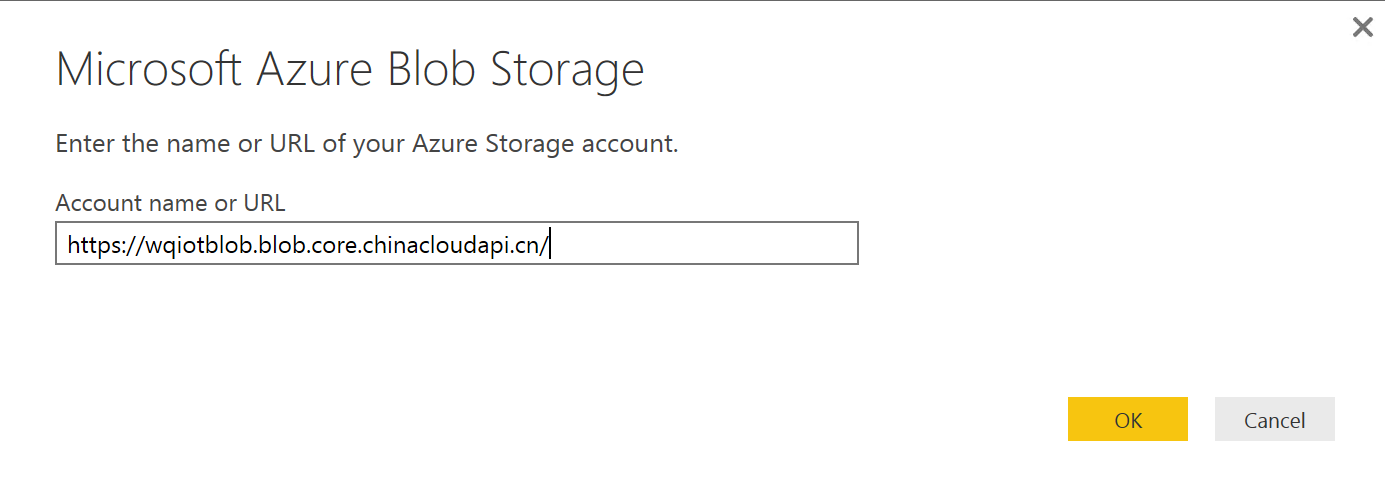
双击打开Power BI Desktop，然后点击“Get Data”，如下图所示：



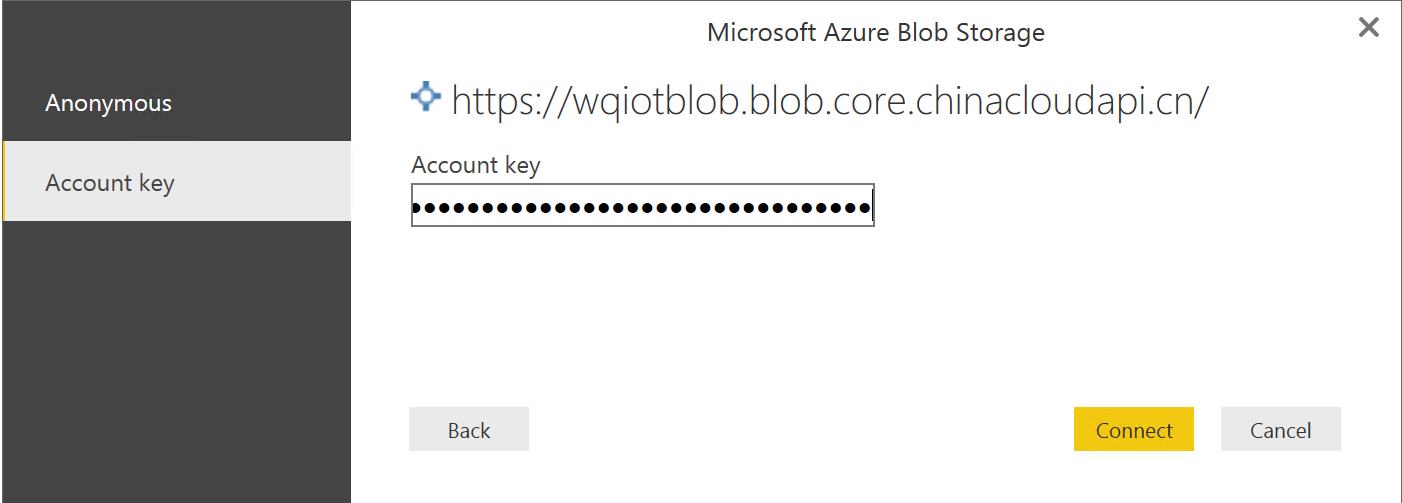
在数据源中选择“Microsoft Azure Blob Storage”，然后点击”Connect”，如下图所示：



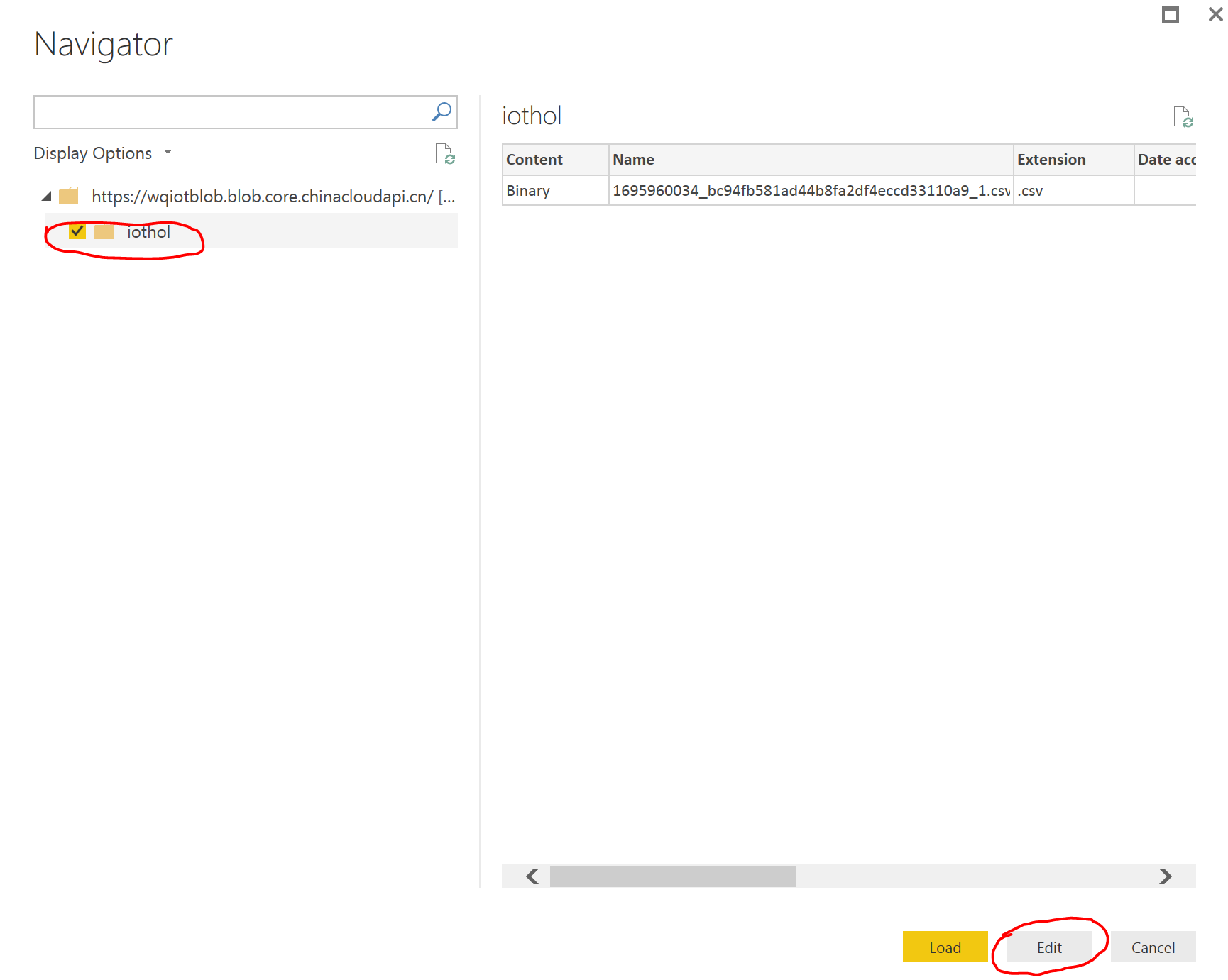
输入存储账户的URL，然后点击OK，如下图所示：



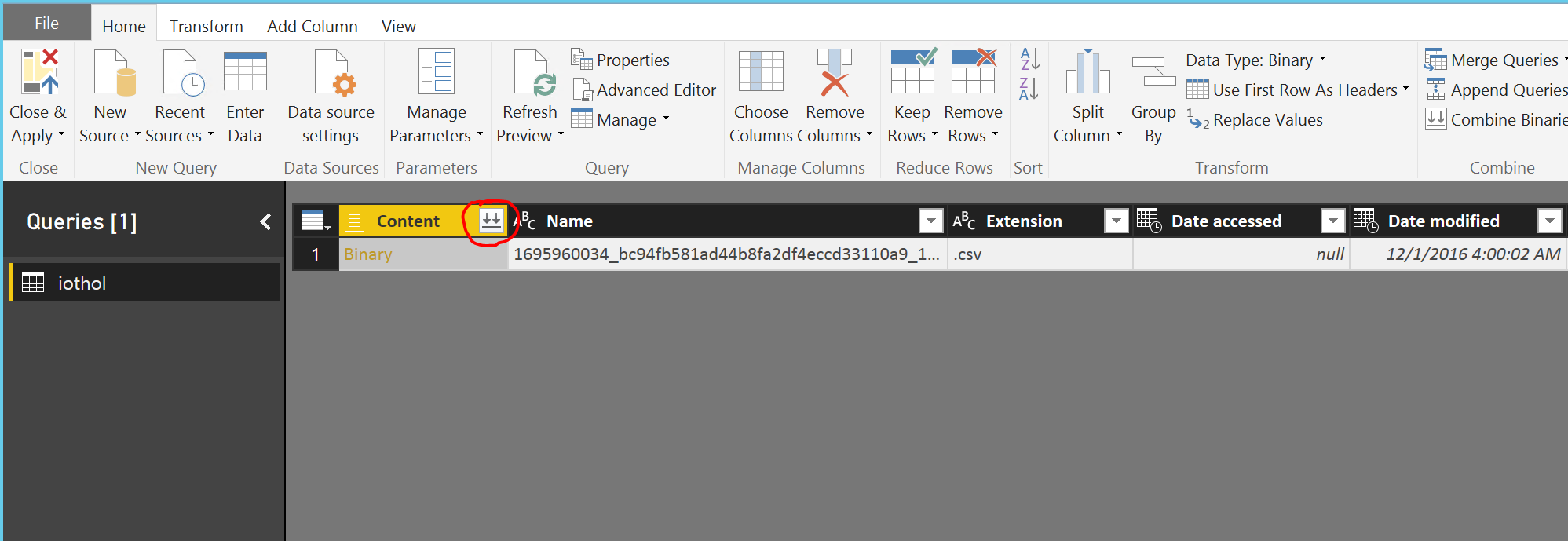
输入存储账户的密钥，然后点击“Connect”，如下图所示：



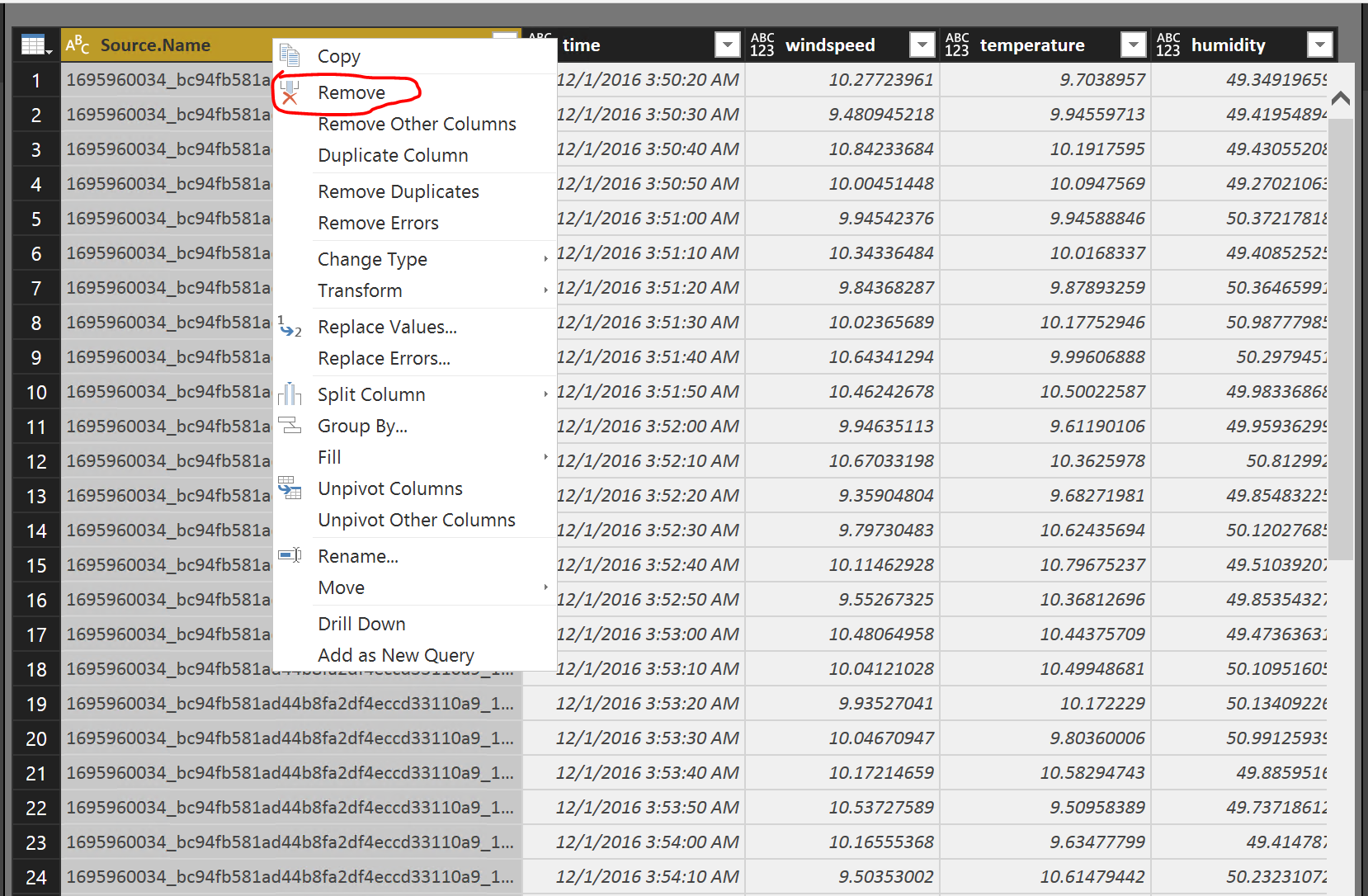
勾选对应的容器，然后点击“Load”按钮，如下图所示：



在编辑界面中点击“Content”旁边的展开按钮，然后点击OK，如下图所示：

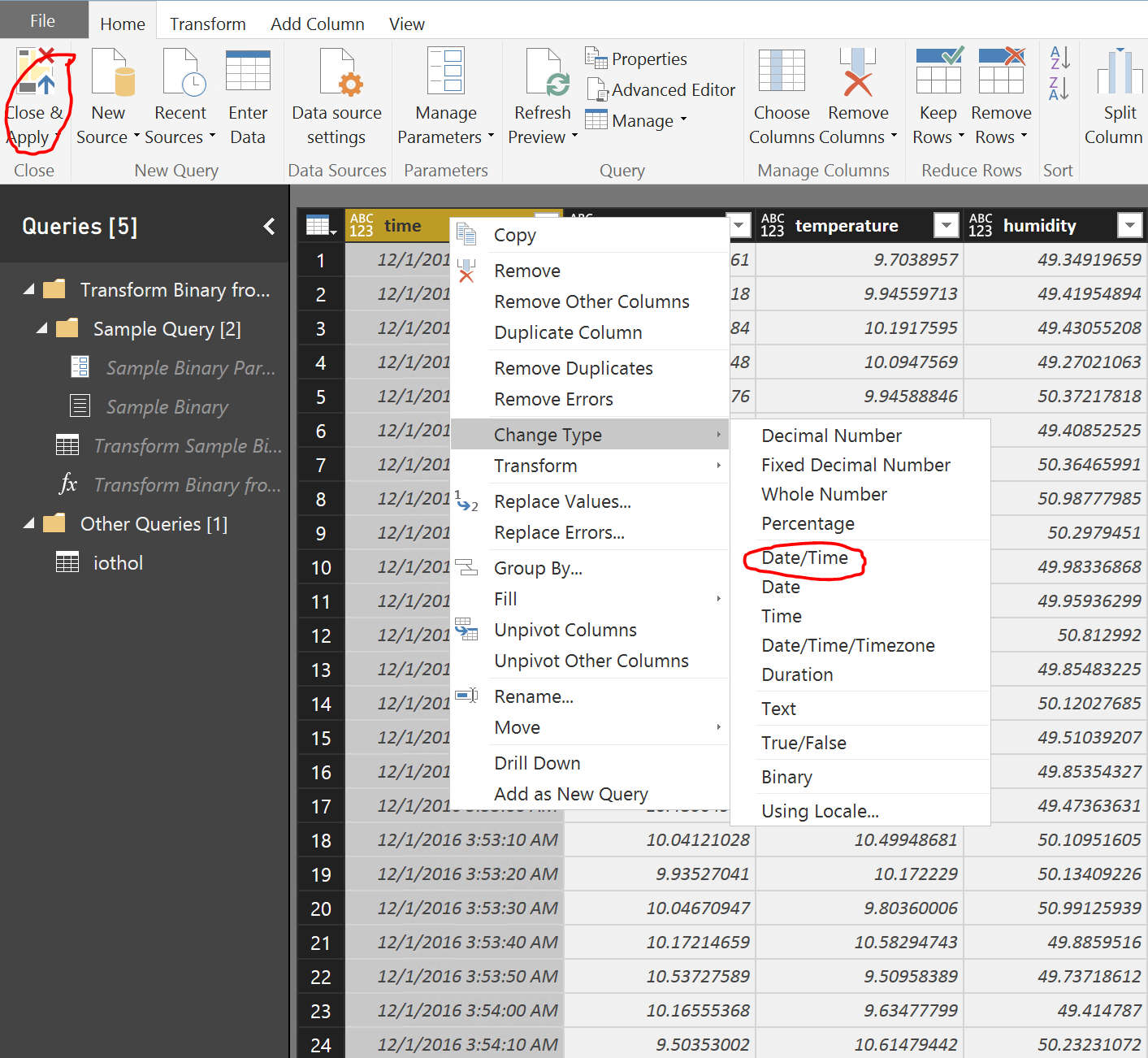


右键点击Source.Name字段，然后点击Remove，如下图所示：



右键点击time字段，然后选择Change Type，勾选Date/Time；然后将wind speed，temperature和humidity字段类型设置为”Decimal Number”

然后点击Close&Apply按钮，如下图所示：



1. **通过 Power BI创建报表**

选择折线图展示数据，其中横轴为时间字段，对应的数值为平均风速、平均温度以及平均湿度，最终的展示结果如下图所示：

