## References

* Azure Stream Analytics <https://azure.microsoft.com/zh-tw/services/stream-analytics/>
* Azure Storage Account <https://azure.microsoft.com/zh-tw/services/storage/>
* Azure Blob Storage <https://azure.microsoft.com/zh-tw/services/storage/blobs/>
* Azure SQL Database <https://azure.microsoft.com/zh-tw/services/sql-database/>

## Requirements

* Finished the part 3 of HOL
* Simulated Linux & Windows Wind turbines
* NuGet packages
  + WindowsAzure.Storage for Blob storage access
  + Newtonsoft.Json for JSON in C#

## Goals

* Historic Data Processing in Azure Stream Analytics
  + Input

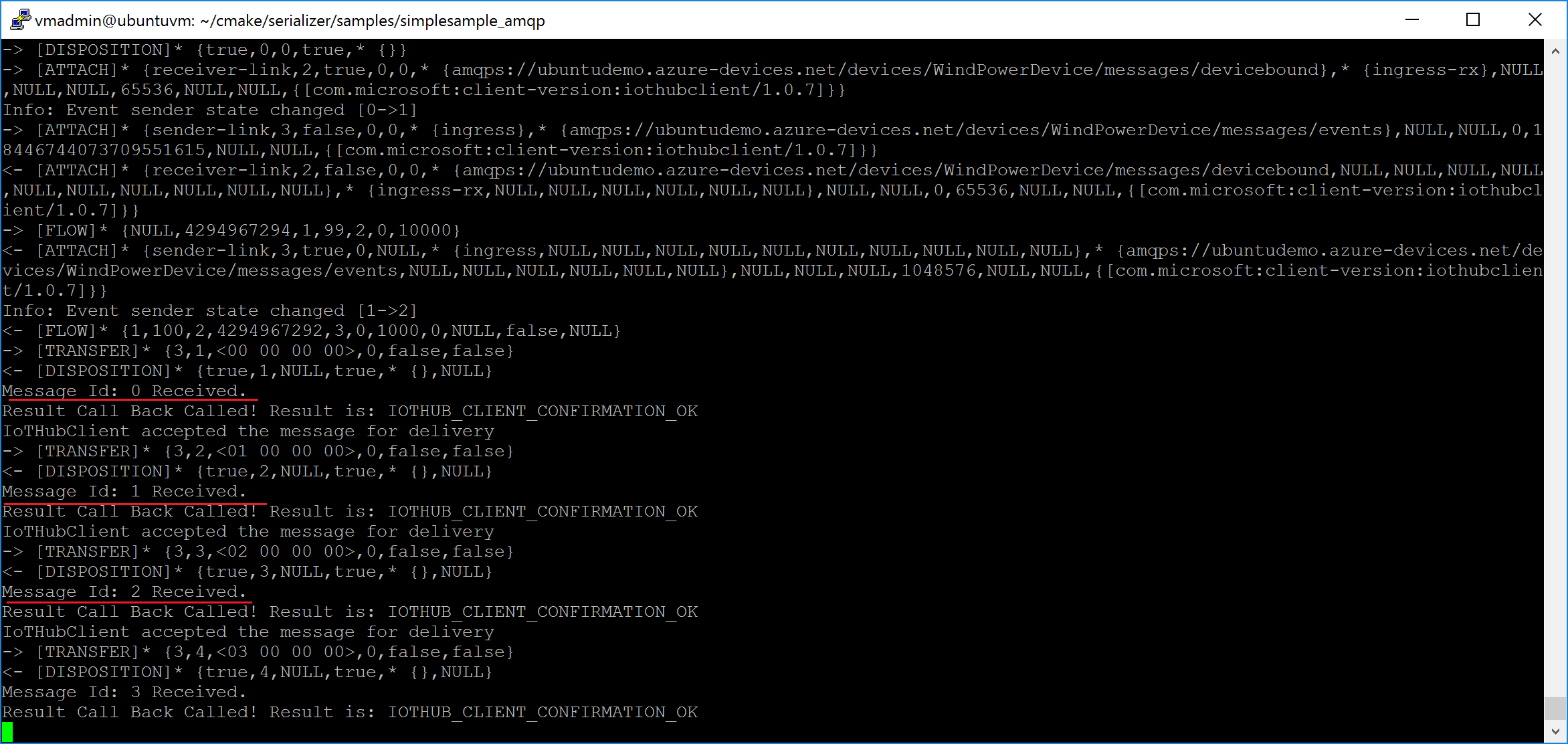
1. Get the telemetry data from IoT Hub
2. Get the device rules from the Blob Storage (Container: devicerules)
   * Output
3. Process the telemetry data with rule alert into Azure SQL database.
4. Process the telemetry data into Blob Storage. (Container: telemetry)

## Step 1: Run the simulated Linux and Windows wind turbines

* LinuxTurbine - Run C Simple Sample of AMQP in C SDK

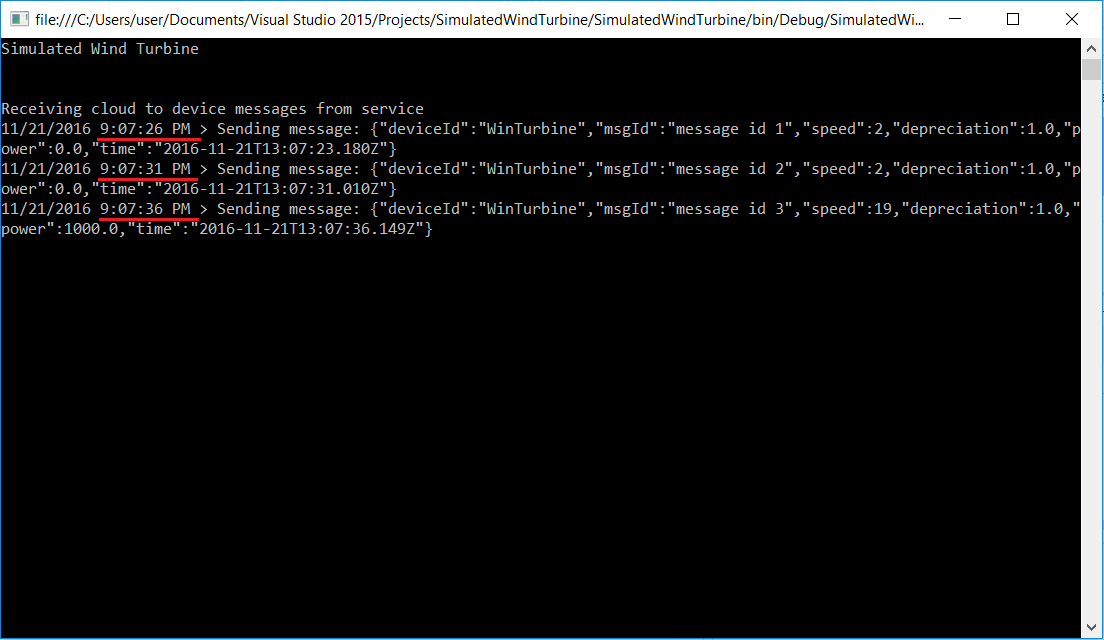
cd ~/azure-iot-sdk-c/cmake/iotsdk\_linux/serializer/samples/simplesample\_amqp/

./simplesample\_amqp



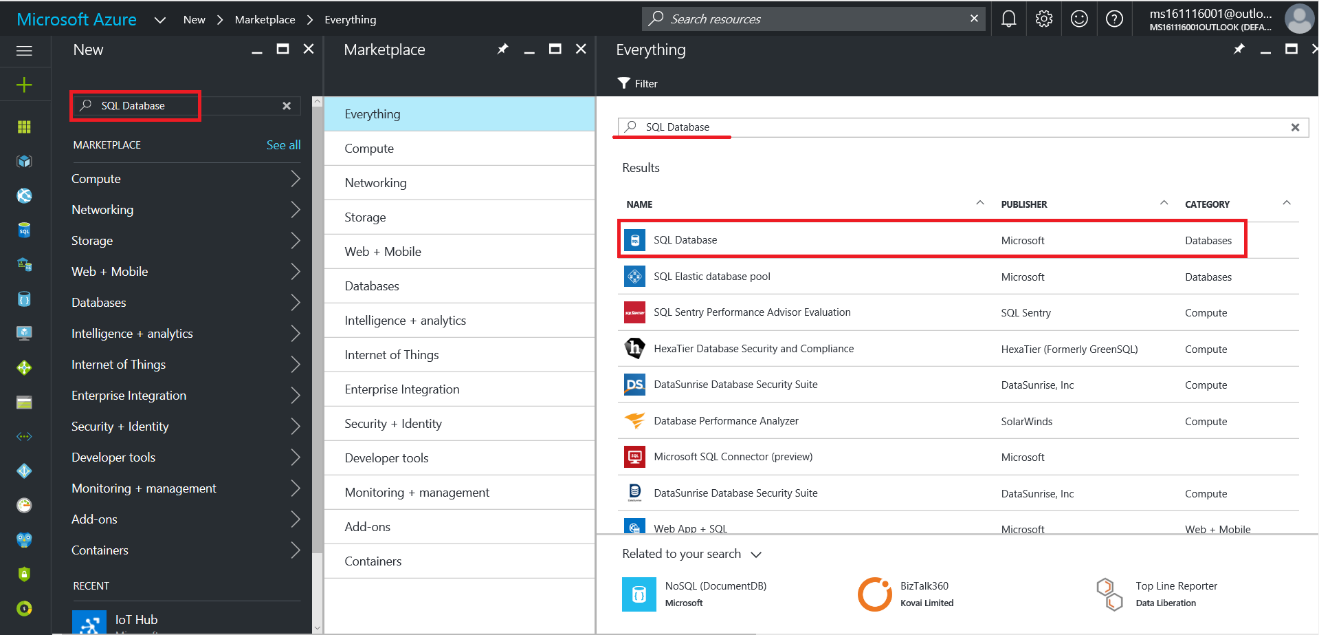
* WinTurbine - Run the executable file (SimulatedWindTurbine.exe) or debugging in the VS
  + The executable file should be located at

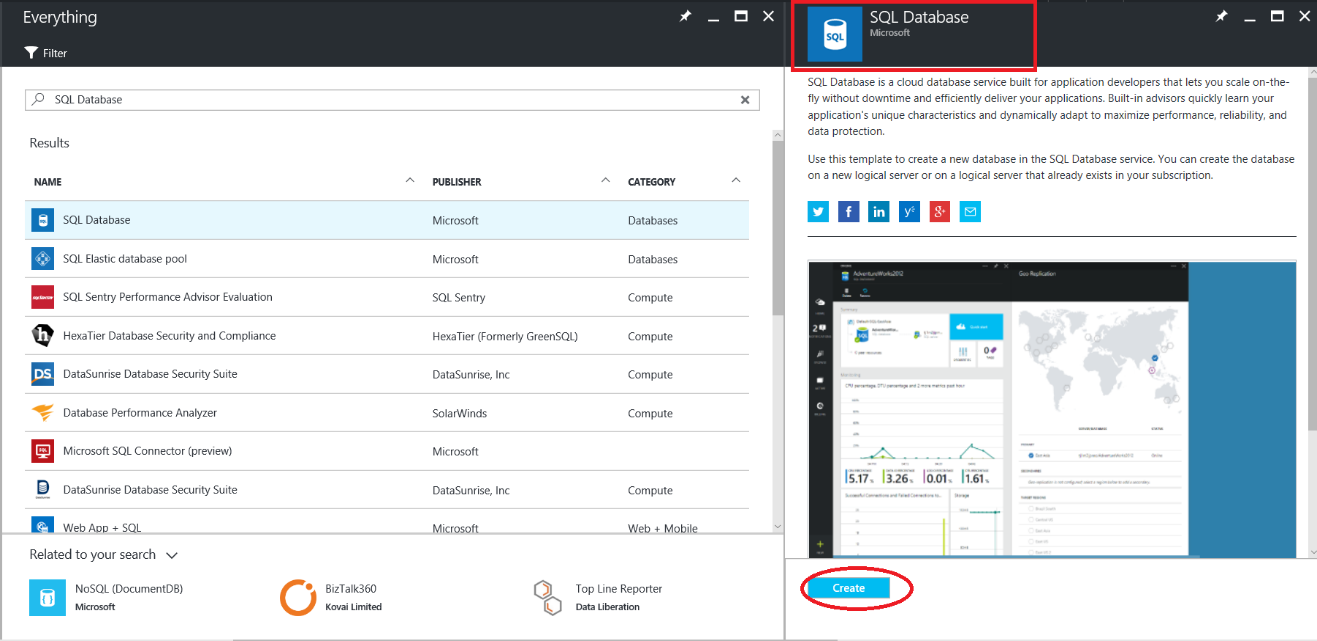
C:\Users\user\Documents\Visual Studio 2015\Projects\SimulatedWindTurbine\SimulatedWindTurbine\bin\Debug



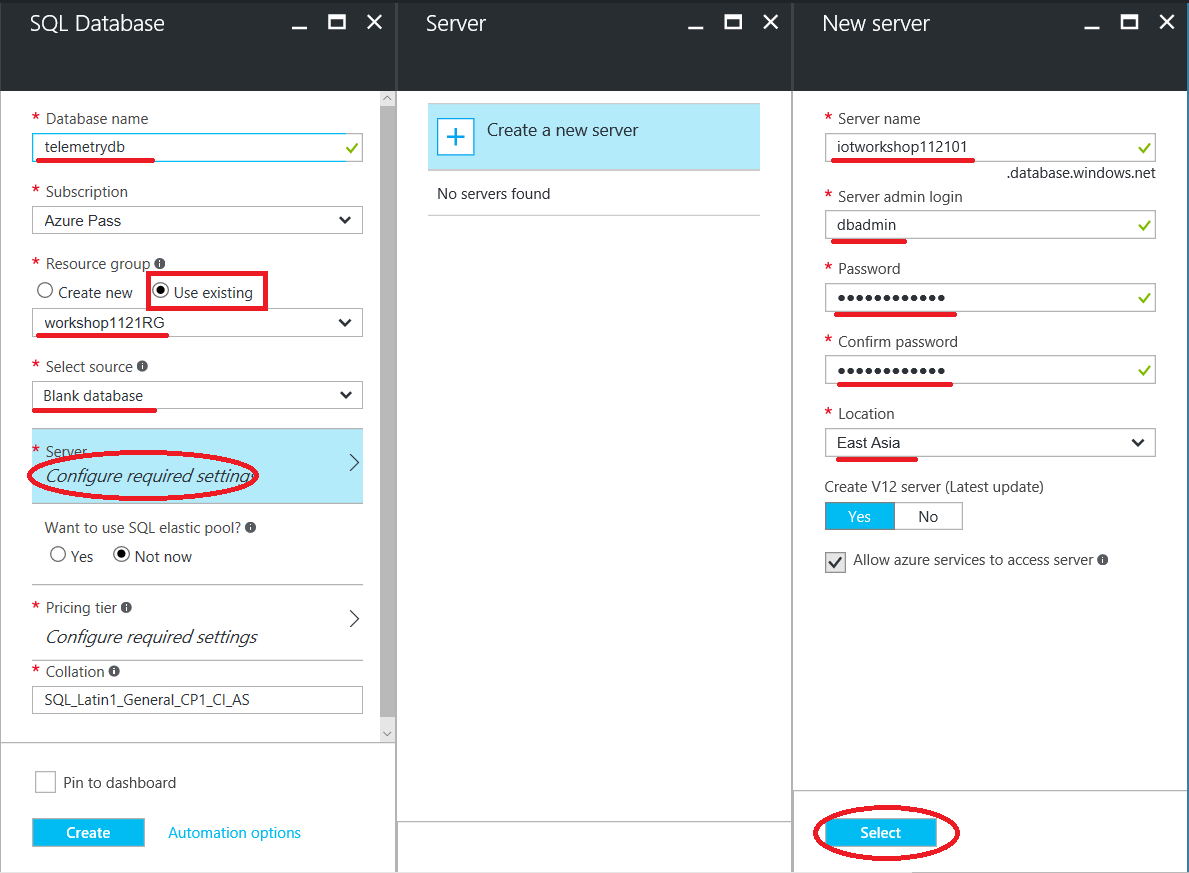
## Step 2: Create Azure SQL Database for the historic data

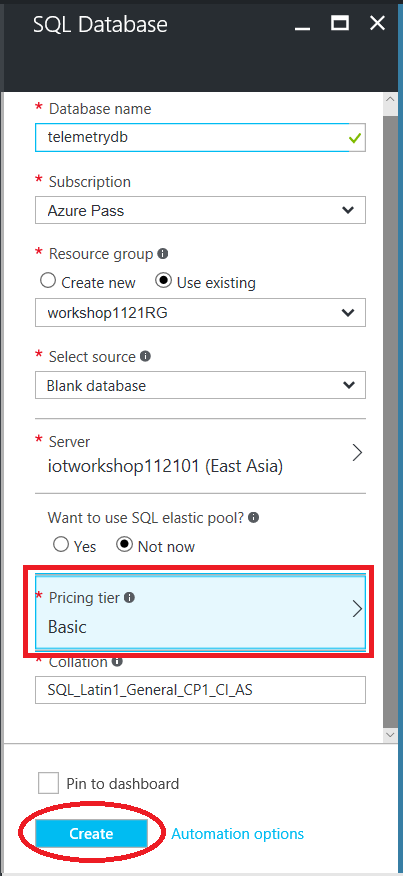
* Login Azure Portal <https://portal.azure.com>
* Search the **SQL Database**



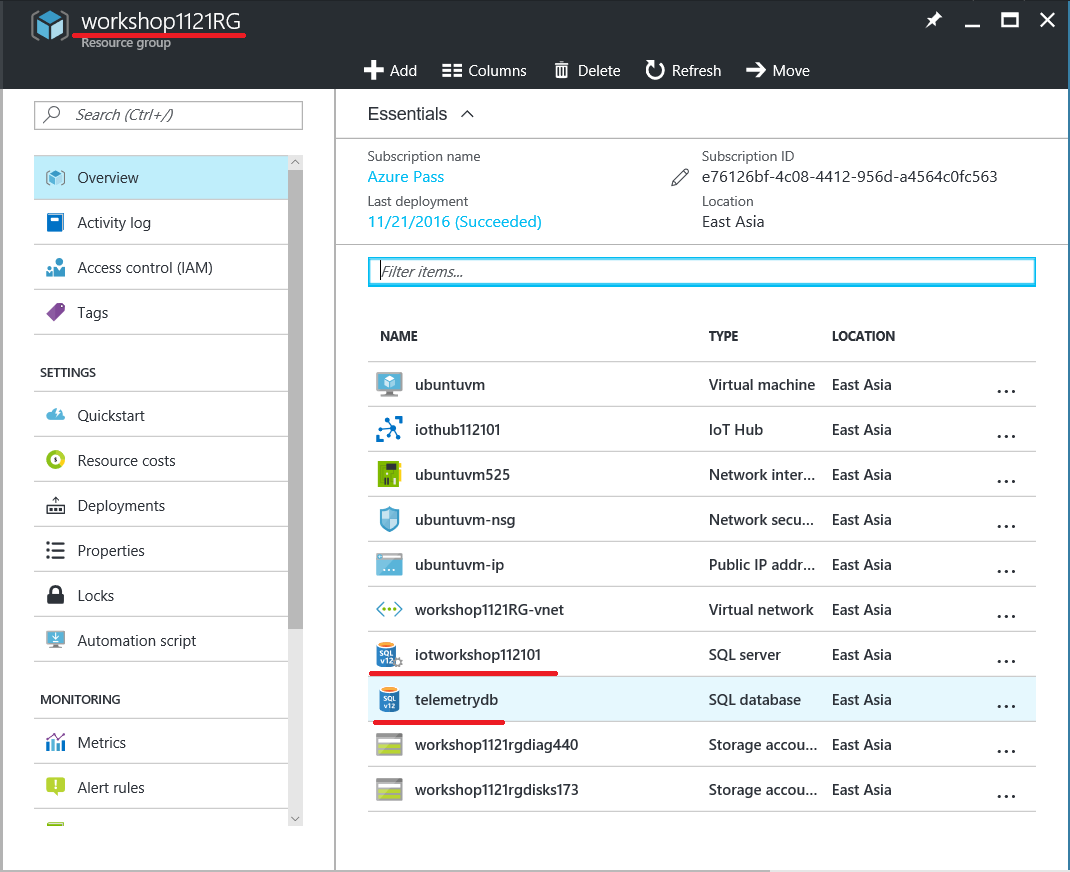


* Create the SQL Database
  + Database name: **telemetrydb (it must be fixed in this workshop)**
  + Select the same resource group
  + Source: **Blank database**
  + Server
    - Server name: **iotworkshop112101** (URL unique, for example)
    - Server admin login: **dbadmin** (for example)
    - Password: **demo1234567!** (for example)
    - Location: **East Asia**
  + Pricing Tire: **Basic**

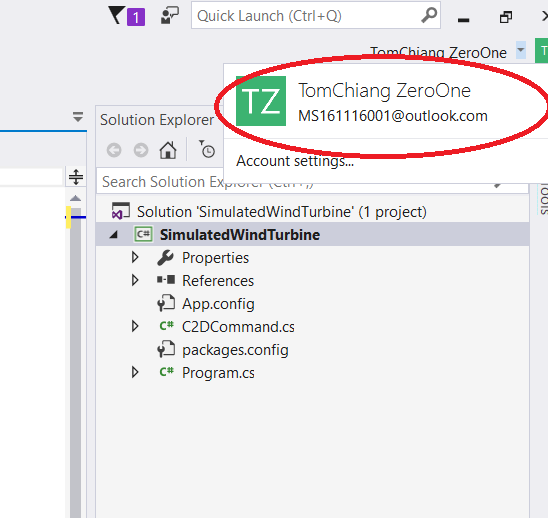




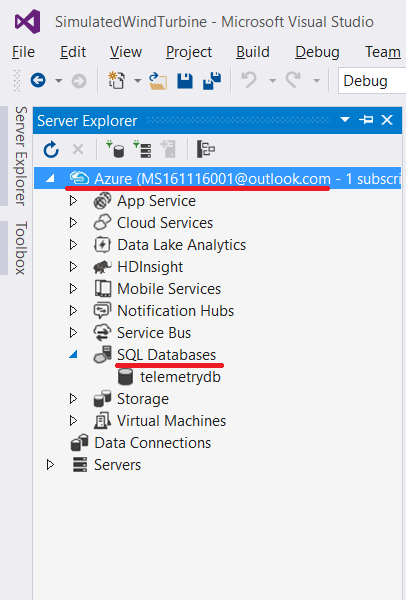
* + Deployment (Wait for a few minutes)



* Login your azure account in **Visual Studio**



* Find the SQL Database in **Server Explorer**

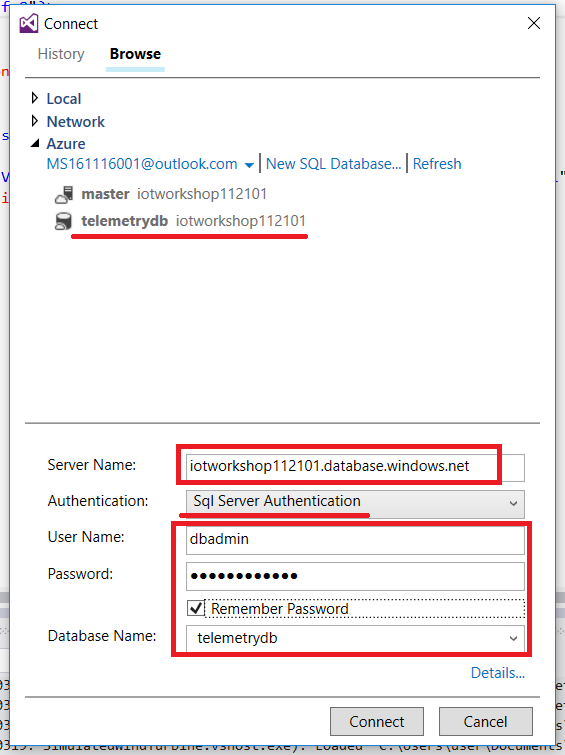


* Right click the **telemetrydb** to open **SQL Server Object Explorer** (or open it from View of the top toolbar)

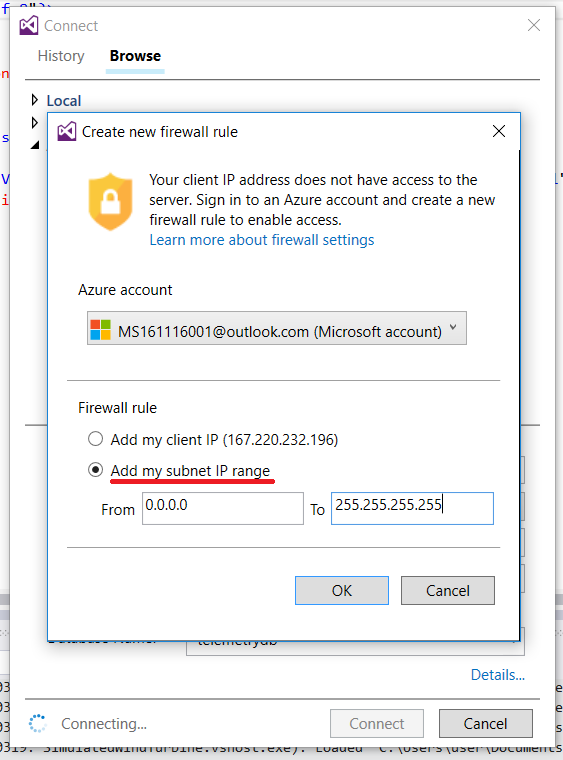
Server Name: **iotworkshop112101** (for example)

Database admin & password: as your previous configurations

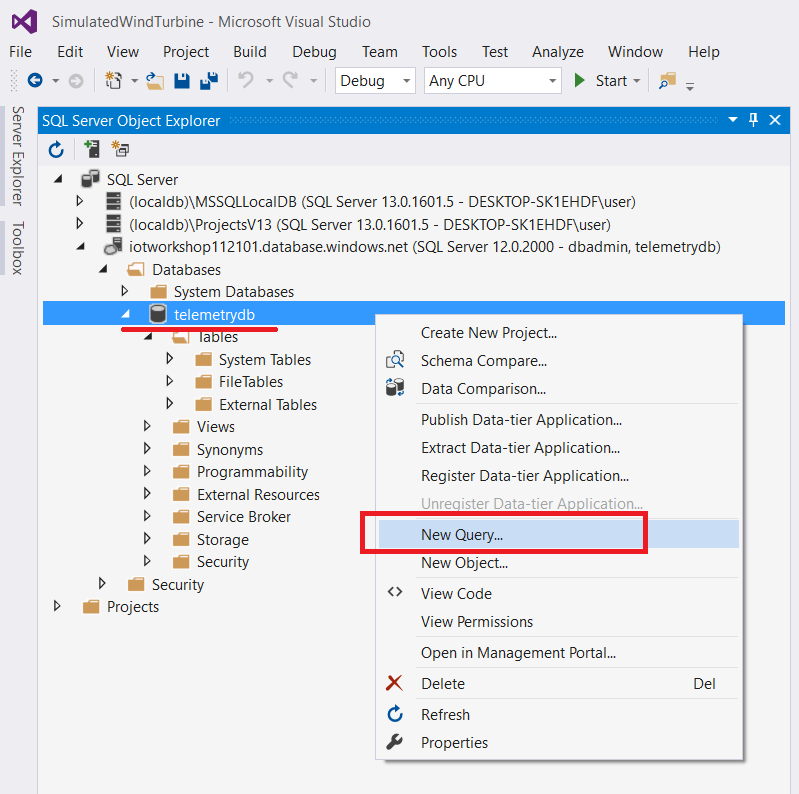
Database Name: **telemetrydb**



* Create new firewall rule
  + From **0.0.0.0** to **255.255.255.255** (demo purpose)



* **New Query** to create new schema



* Paste the following scripts and execute it.

CREATE SCHEMA [Prod]

CREATE TABLE [Prod].[HistoricData]

(

[Id] INT IDENTITY (1, 1) NOT NULL,

[IoTHubDeviceID] NVARCHAR(128) NULL,

[MessageID] NVARCHAR(128) NULL,

[WindSpeed] FLOAT(10) NULL,

[Depreciation] FLOAT(10) NULL,

[Power] FLOAT(10) NULL,

[Altitude] Decimal(9,6) NULL,

[Longitude] Decimal(9,6) NULL,

[Latitude] Decimal(9,6) NULL,

[CutOutSpeed] FLOAT NULL,

[CutOutSpeedAlarm] INT DEFAULT ((0)) NULL,

[Repair] FLOAT NULL,

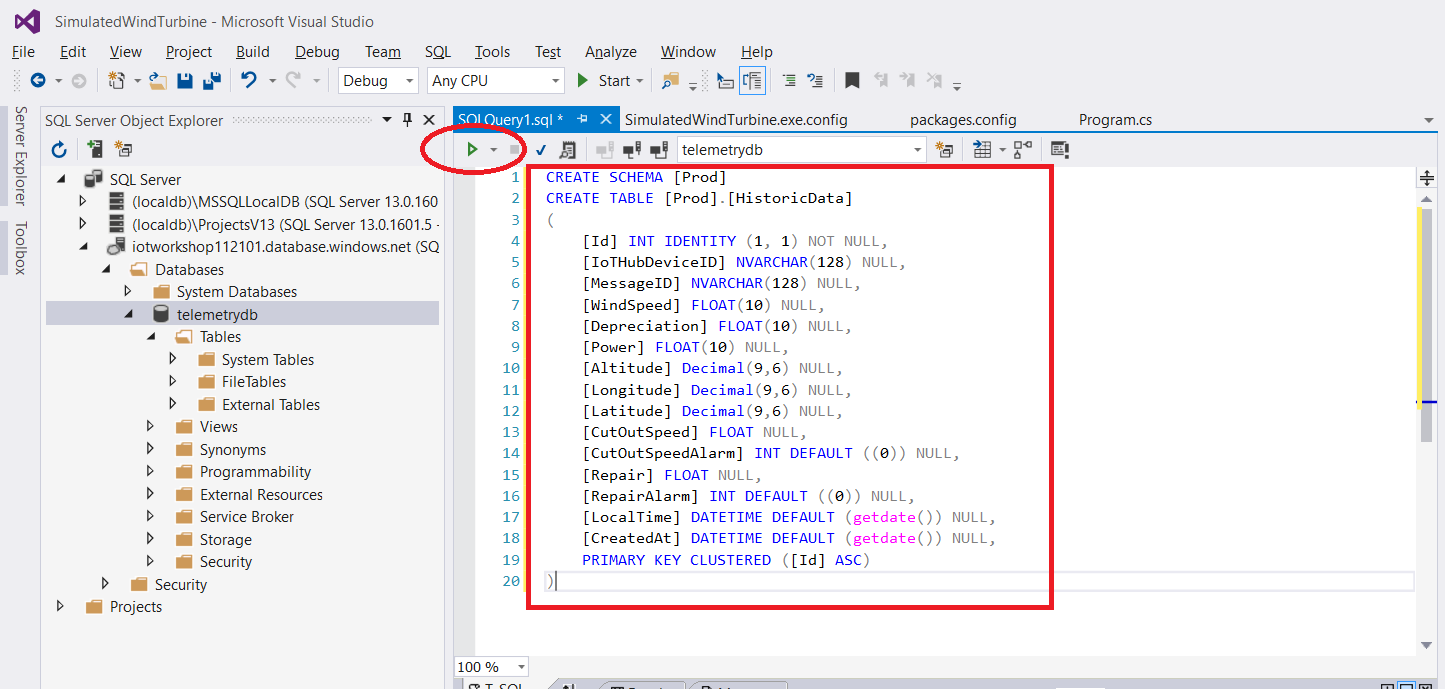
[RepairAlarm] INT DEFAULT ((0)) NULL,

[LocalTime] DATETIME DEFAULT (getdate()) NULL,

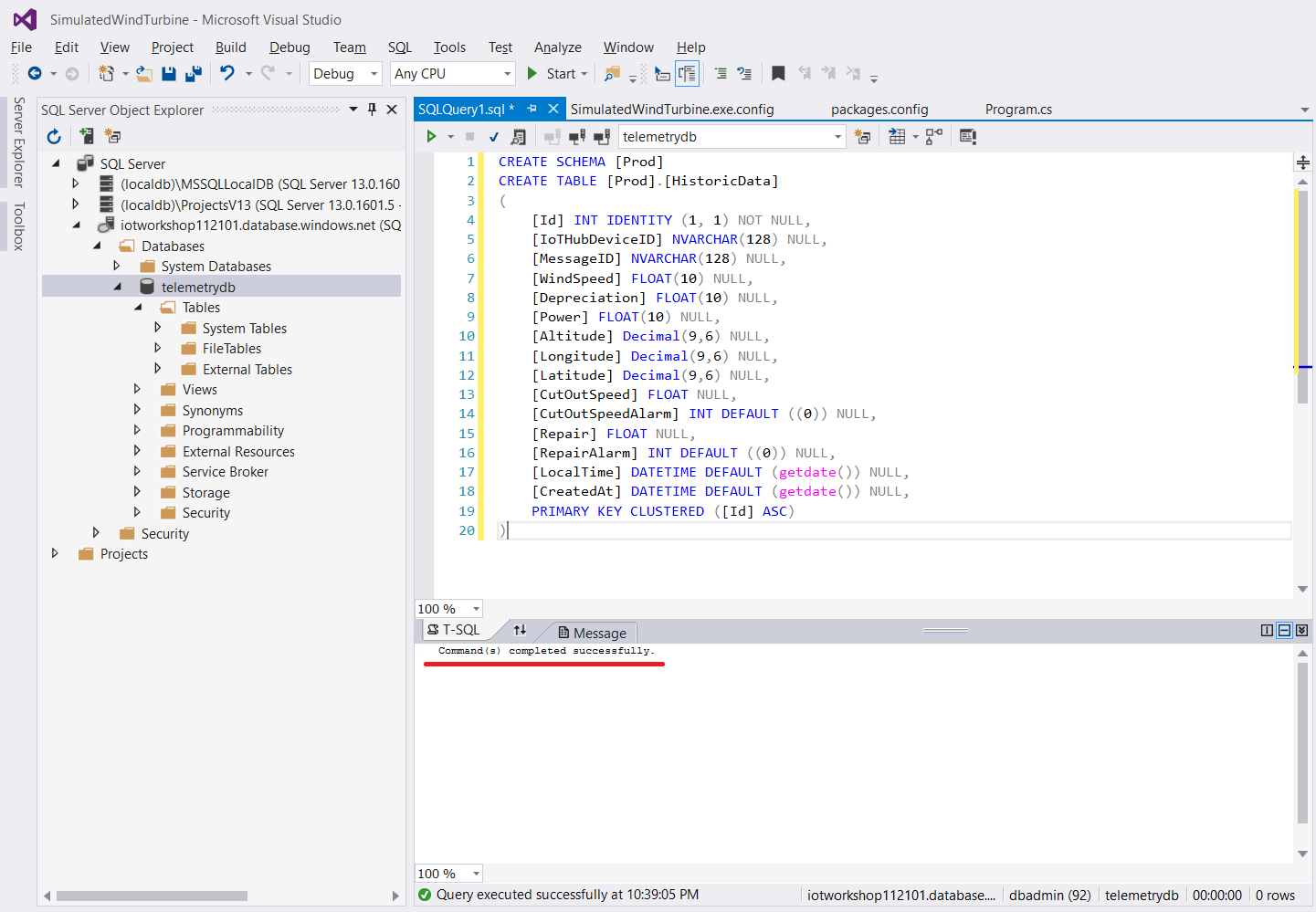
[CreatedAt] DATETIME DEFAULT (getdate()) NULL,

PRIMARY KEY CLUSTERED ([Id] ASC)

)

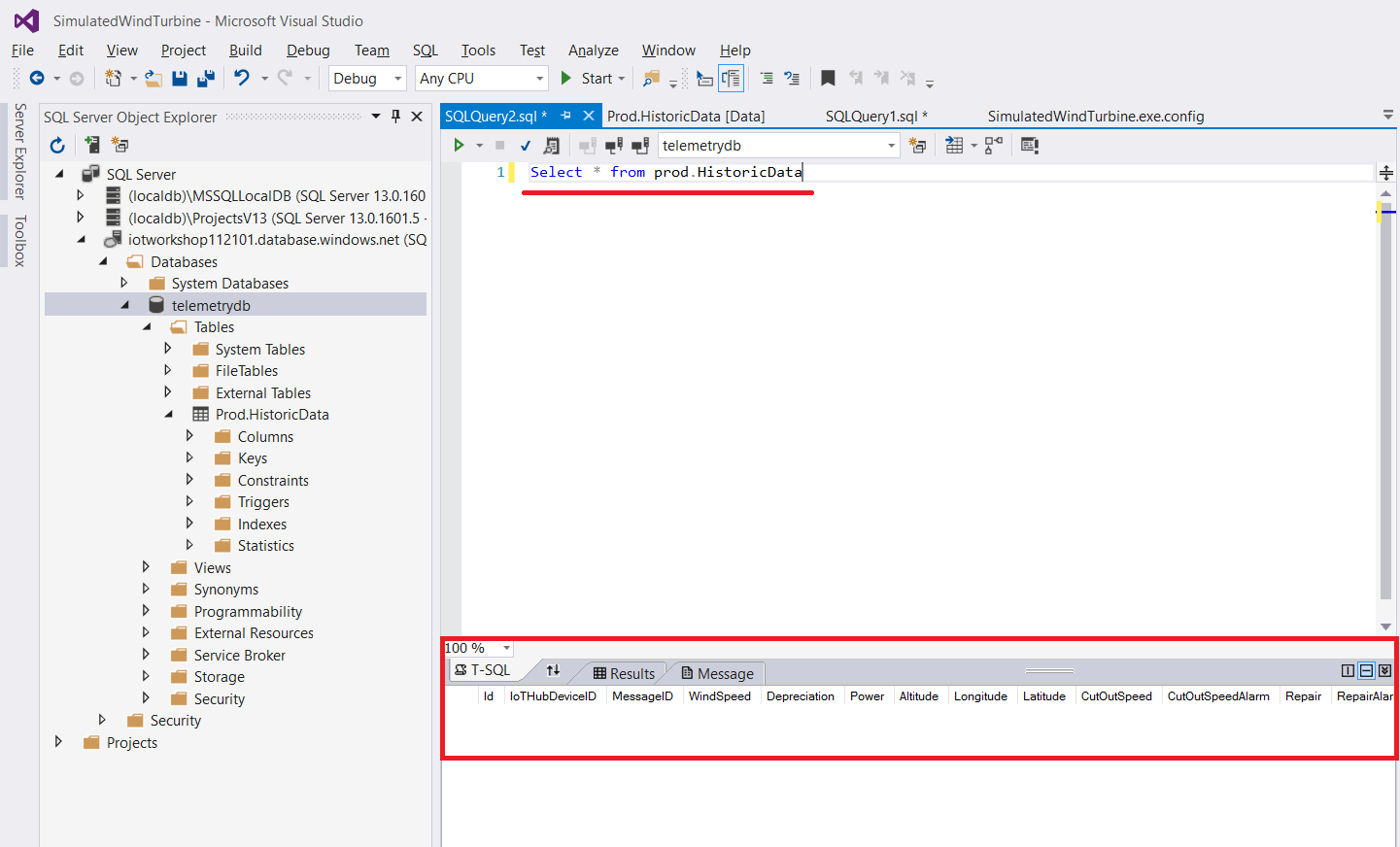


* Confirmation



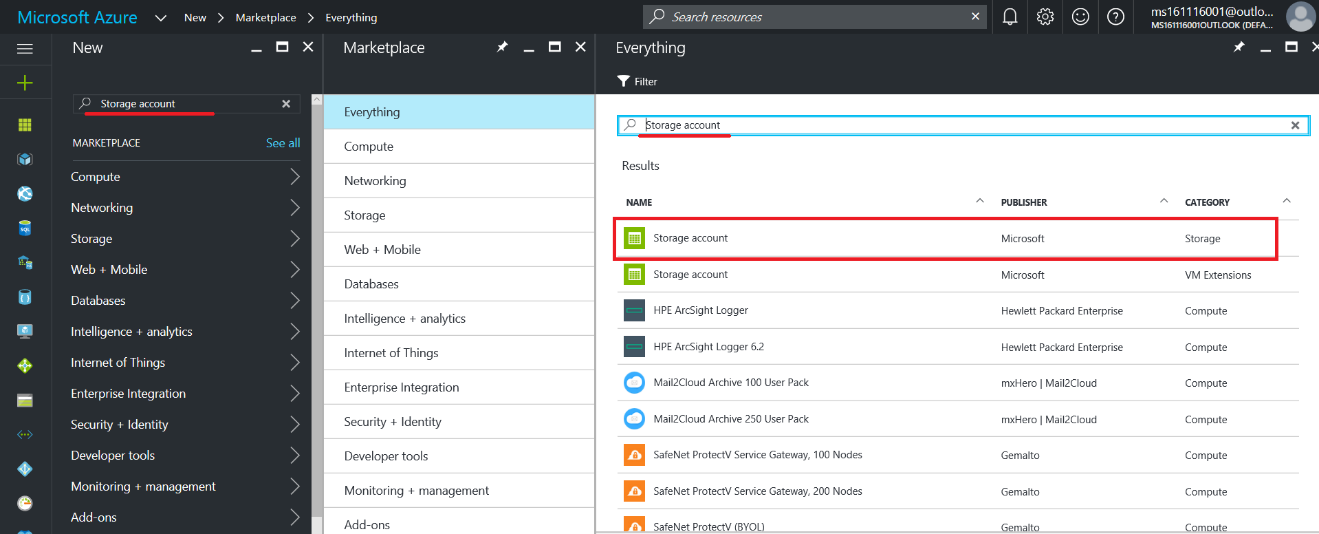
* Confirm the schema by **New Query**

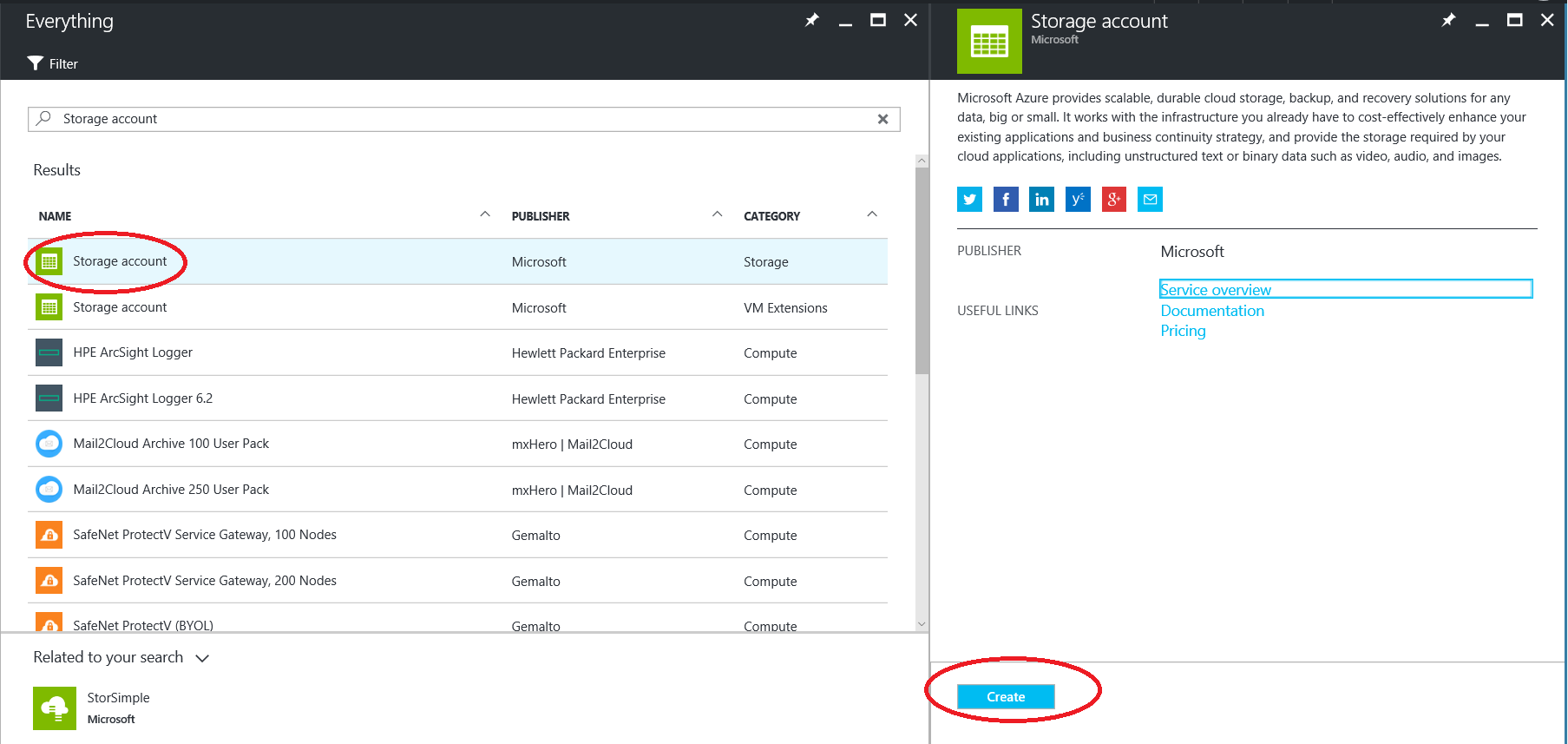
Select \* from prod.HistoricData



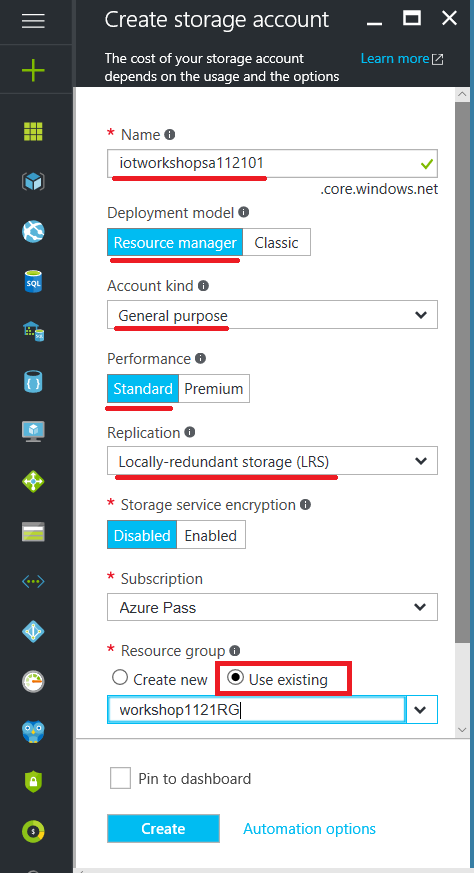
## Step 3: Create a Reference Blob for device rules

* Provision a Storage Account for Blob
  + Login Azure Portal <https://portal.azure.com>
  + Search the **Storage account** and create it.

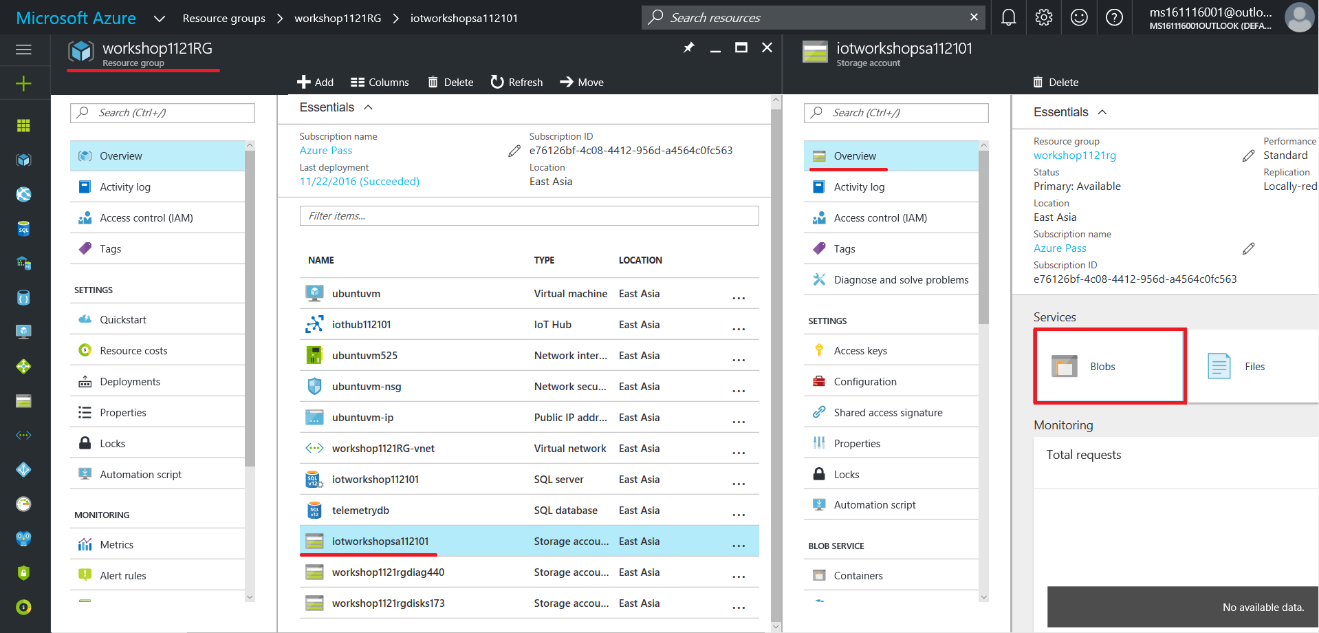




* + Create storage account
    - Name: **iotworkshop112101**
    - Account kind: **General purpose**
    - Use existing resource group

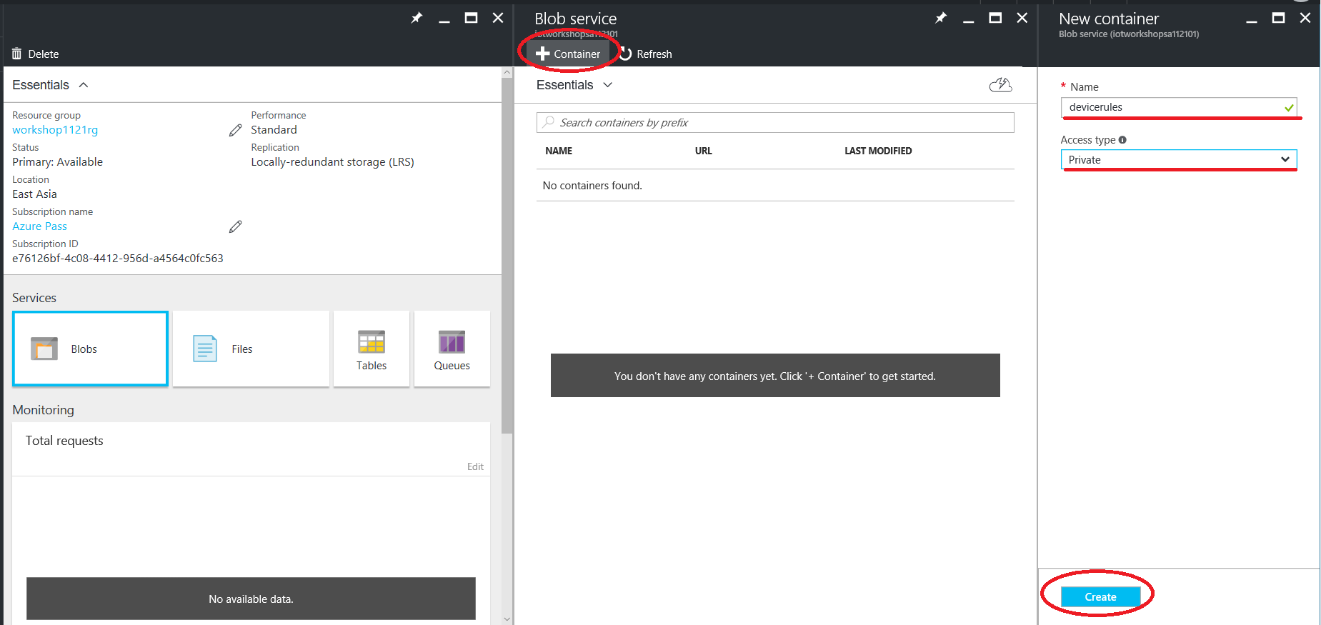


* + Create and wait for the deployment
  + Find the Blob service and create a **container** for reference blob

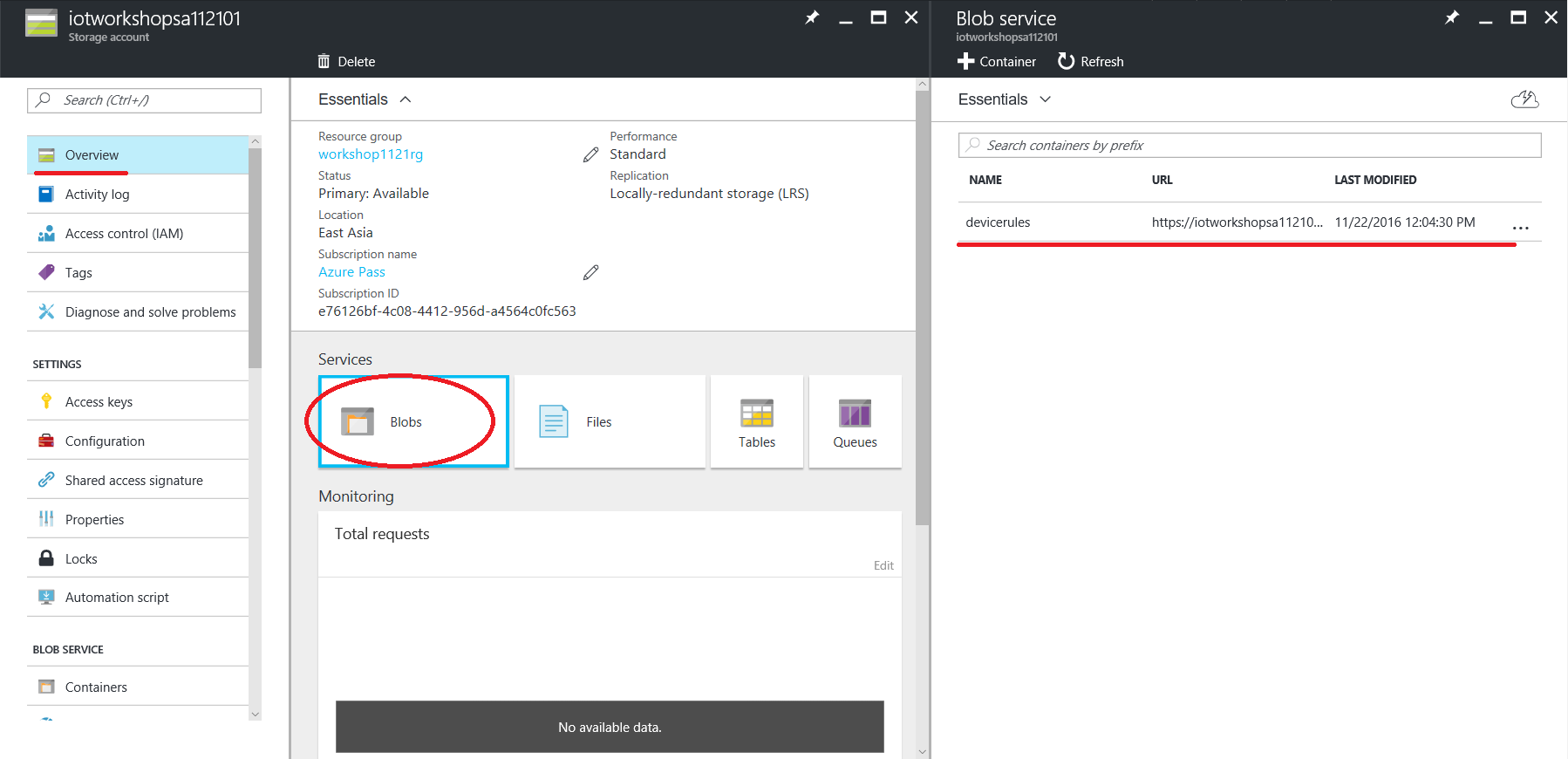


* + New container name: **devicerules**

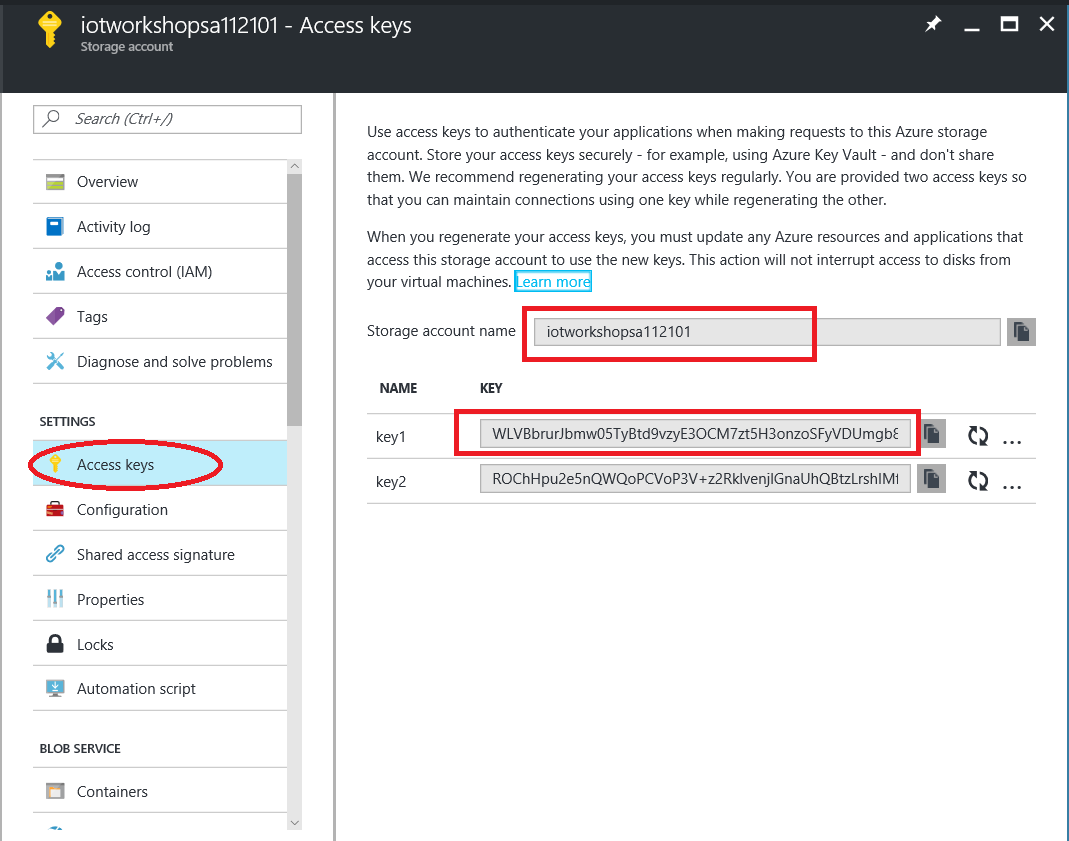
**(The name, “devicerules” must be fixed)**



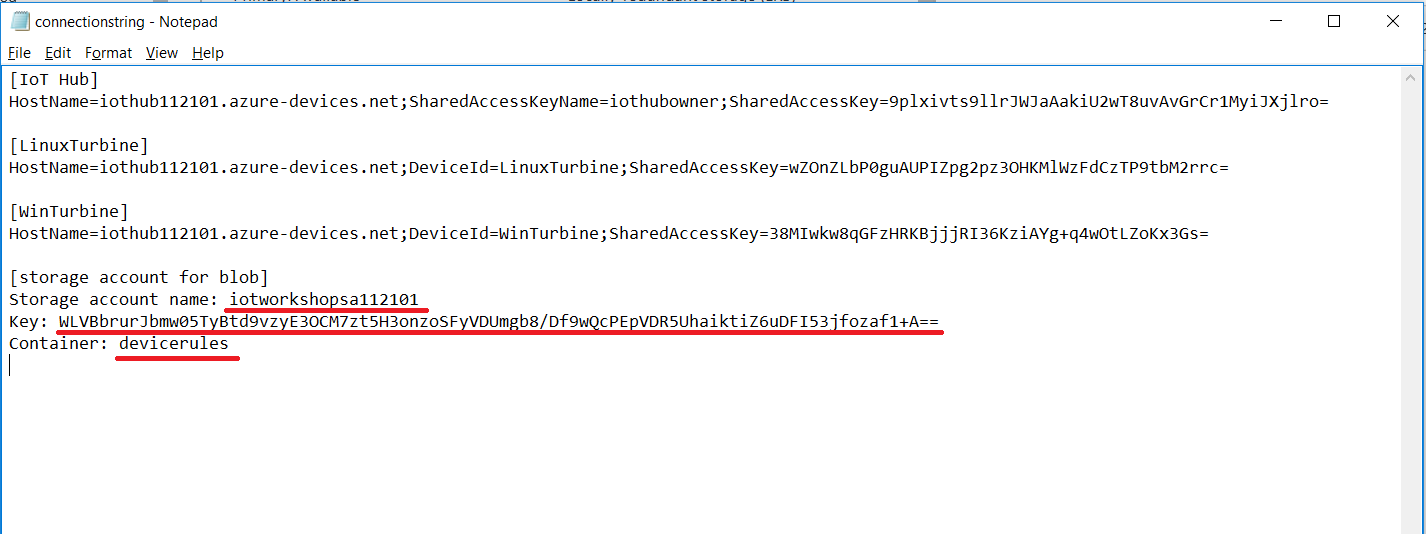
* + Created successful



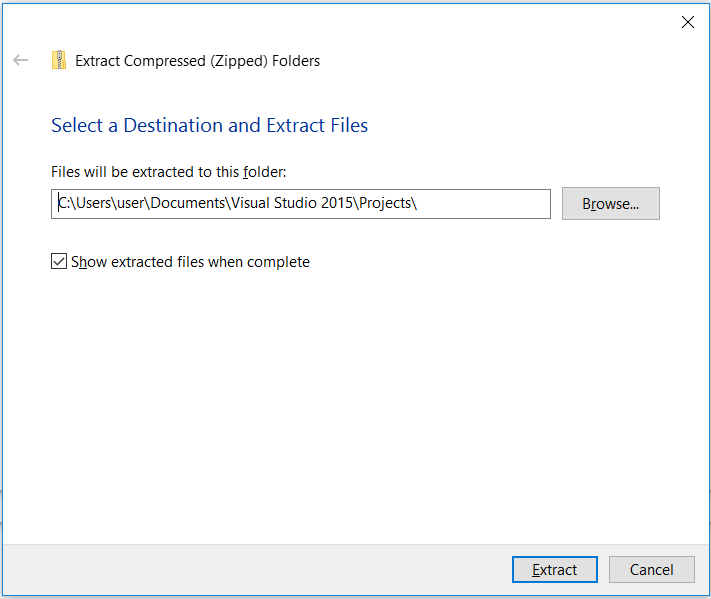
* + Get the name and key of Storage Account



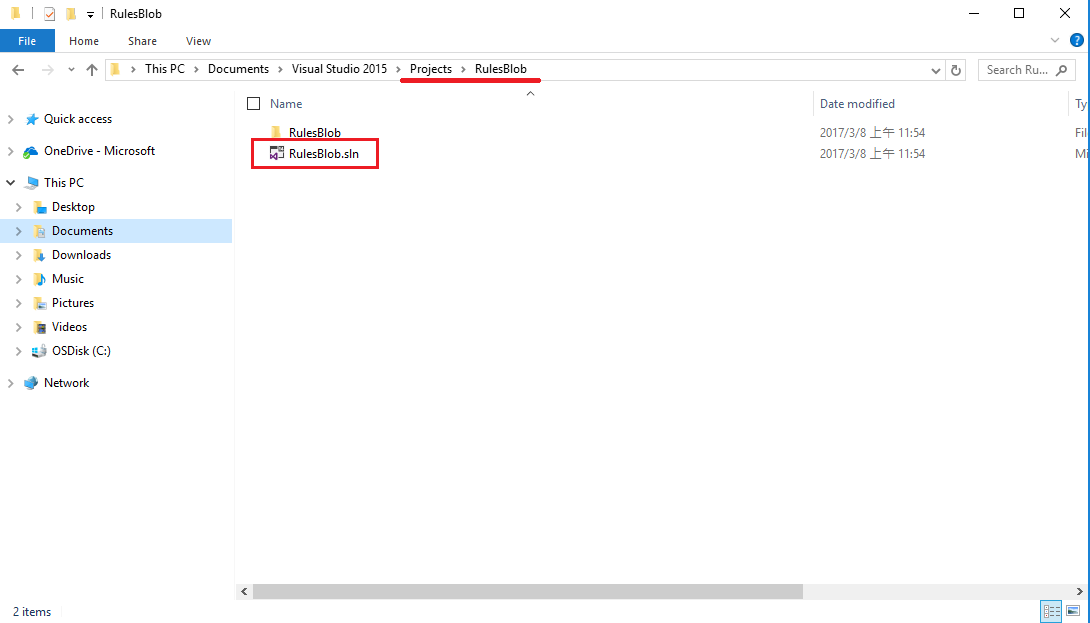
* + Save it for the later used



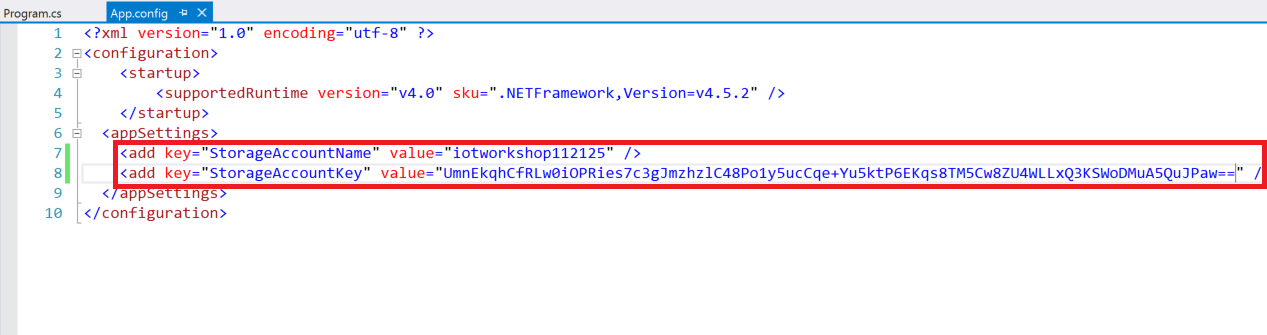
* Build the Console App for Rules Blob
  + Unzip the **RulesBlob.zip** to the projects of VS



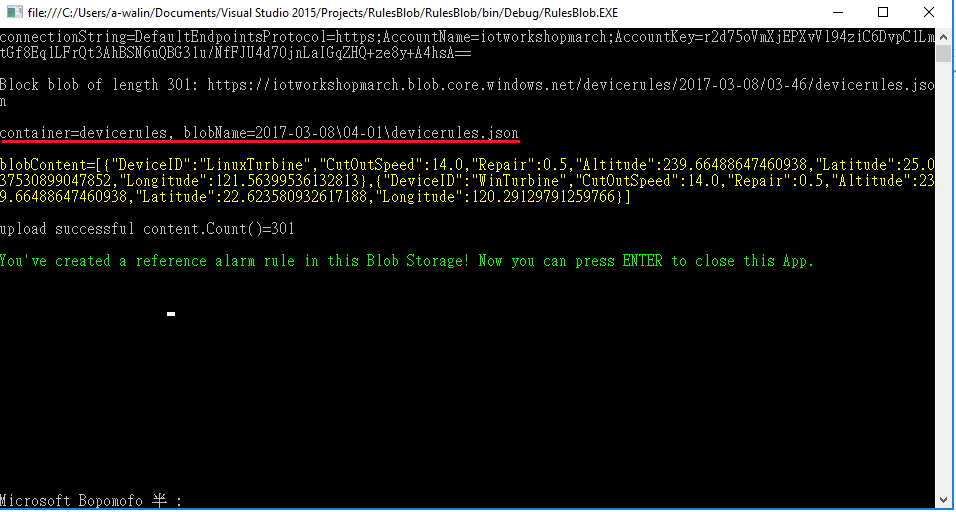
* + Open the solution of VS



* + Update the **name** and **key** of **Storage Account** in **App.config** file



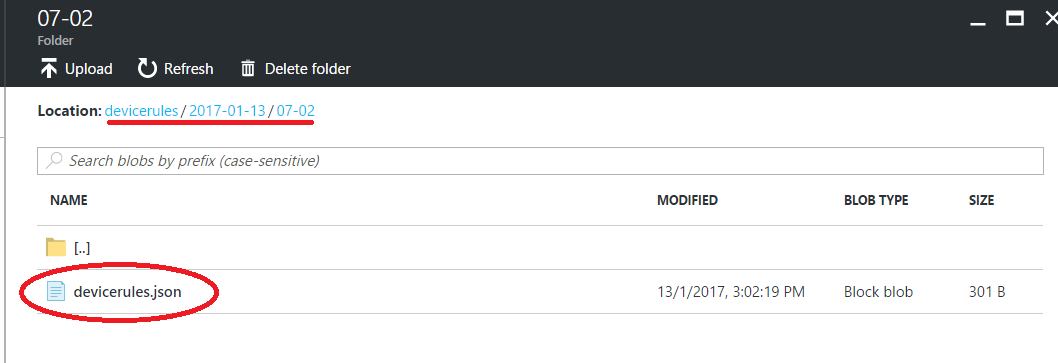
* + Press **F5** to build and debug the project
  + Check the output of console



* + JSON Content

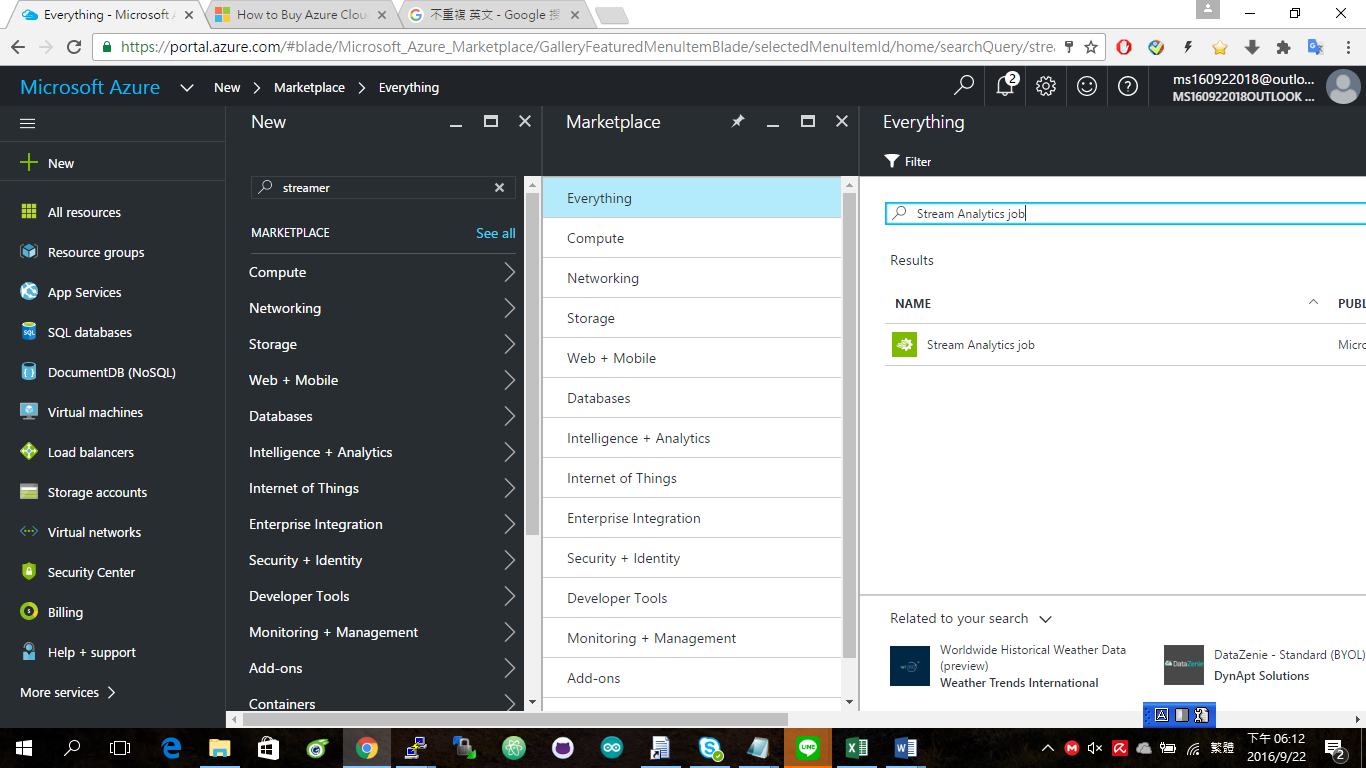


* + The **devicerules.json** file has been uploaded in the **devicerules** container of Blob Storage by date and time.

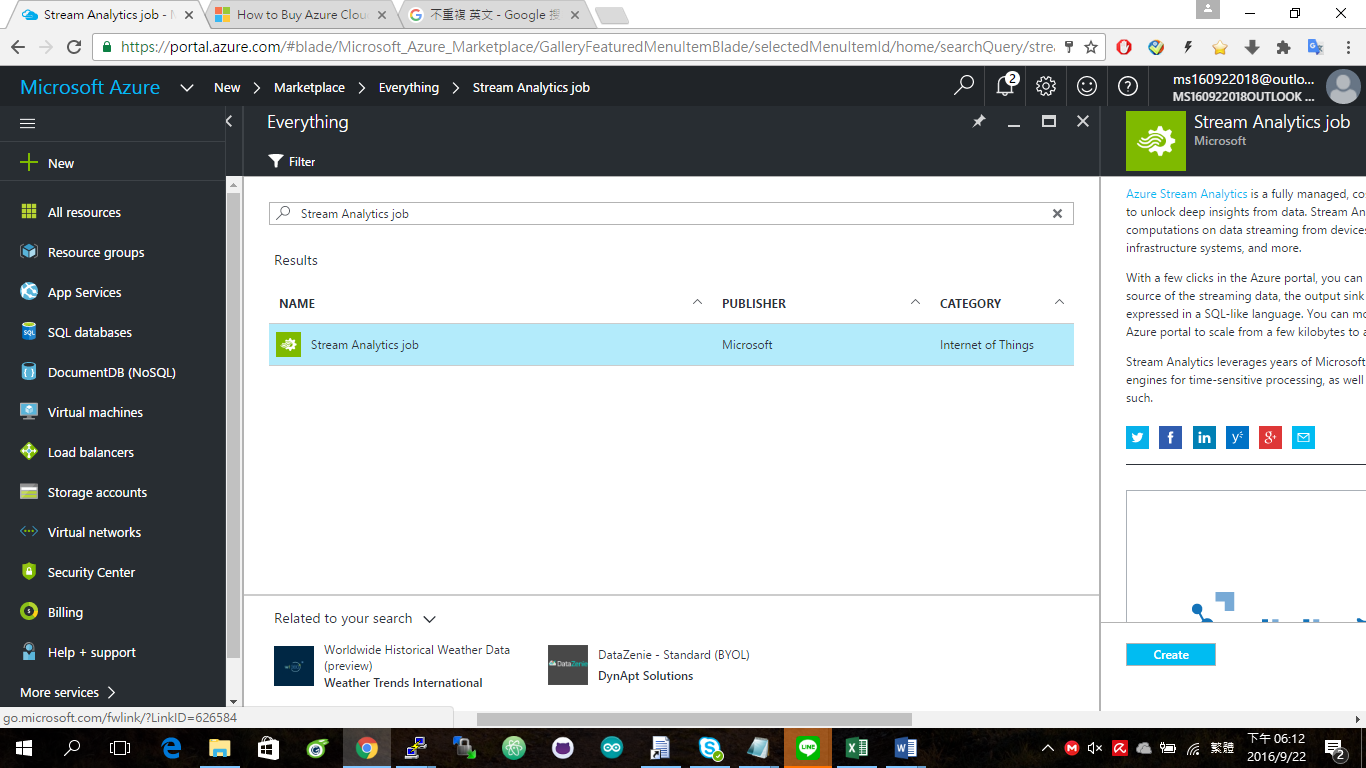


## Step 4: Create a Stream Analytics Job for Historic Data (SQL DB and Blob)

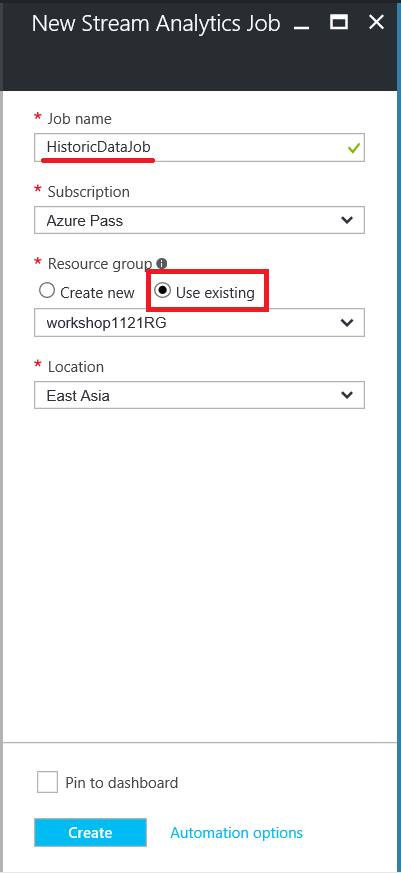
* Login Azure Portal <https://portal.azure.com>
* Create a Stream Analytics Job
  + Add and search **Stream Analytics job**



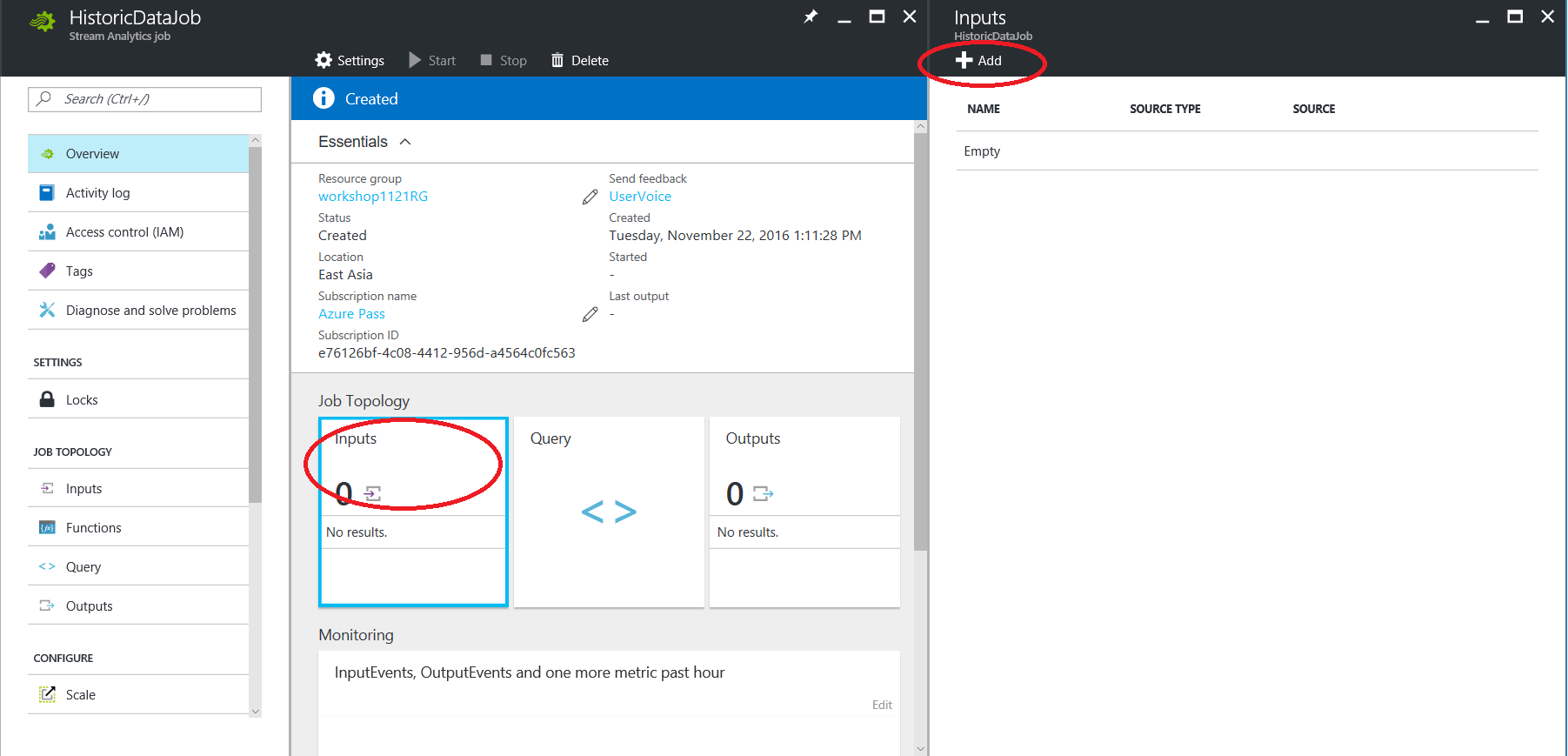
* + **Create** a Stream Analytics Job



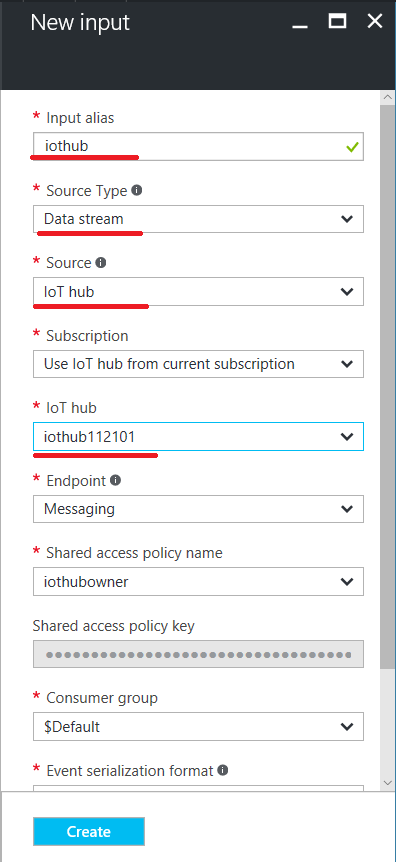
* + **Create** a new Job
    - job name: **HistoricDataJob** (for example)
    - Choose a subscription
    - Choose a resource group (using the same resources group is preferred)

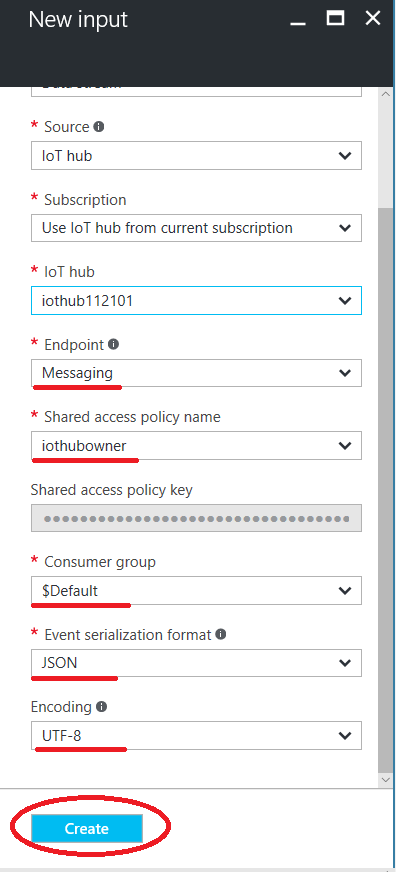


* + Add an **IoT Hub** as the **input**

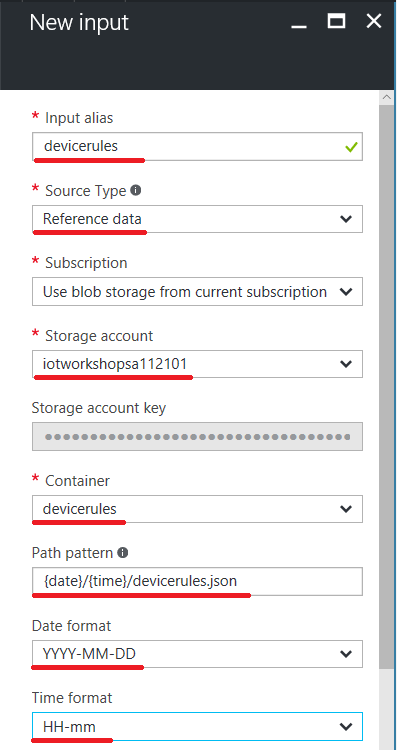


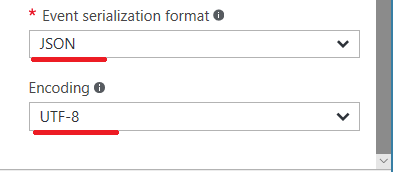
* + - Input alias: **iothub (must be fixed)**
    - Source Type: **Data stream**
    - Source: **IoT Hub**
    - IoT Hub: select the used IoT Hub
    - The others should be set as below



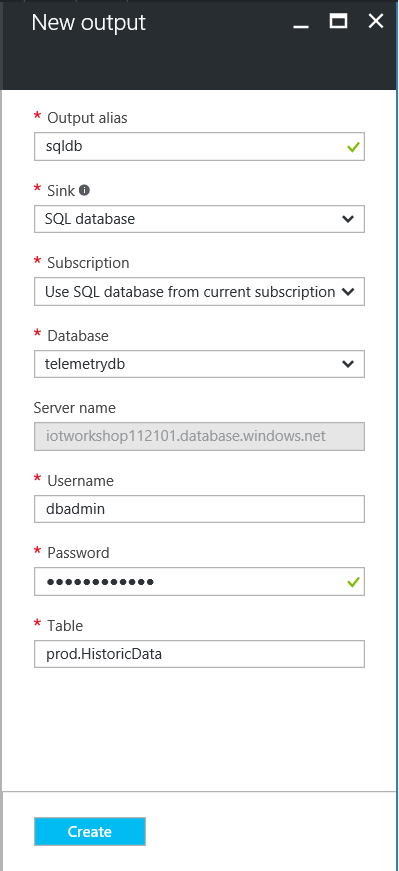


* + Add a Reference **Blob** as the **input**
    - Input alias: **devicerules (must be fixed)**
    - Source Type: **Reference data**
    - Storage account: select the storage account as you provisioned
    - Container: **devicerules (select this one)**
    - Path pattern: **{date}/{time}/devicerules.json (must be fixed)**
    - Date format: **YYYY-MM-DD (IMPORTANT!! Please choose this format for this workshop)**
    - Time format: **HH-mm ((IMPORTANT!! Please choose this format for this workshop)**
    - The **JSON** and **UTF-8** also must be fixed.

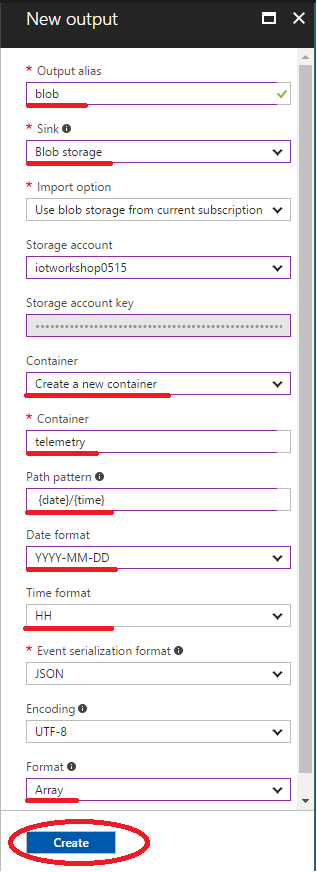




* Press the **Create** button
  + Add a **SQL database** as the **output**
    - Output alias: **sqldb (must be fixed)**
    - Sink: **SQL database**
    - Database: **telemetrydb**
    - Username & password: please input your name and password of SQL database (for example, dbadmin/demo1234567!)
    - Table: **prod.HistoricData** **(must be fixed and please check the filled string without any space)**



* + Add a **Blob Storage** as the **output**
    - Output alias: **blob (must be fixed)**
    - Sink: **Blob storage**
    - Storage account: the same as the previous storage account
    - Container: **telemetry (We need to create a new one)**
    - Path pattern: **{date}/{time} (must be fixed)**
    - Date format: **YYYY-MM-DD (must be fixed)**
    - Time format: **HH (must be fixed)**
    - Format: **Array** (it’s easy to read the JSON array)
    - The others should be set as below



* + Add a query as the historic data
    - Paste the following scripts to the query of Stream Analytics.

WITH HistoricData AS (

SELECT

Stream.IoTHub.[ConnectionDeviceId] AS [IoTHubDeviceID],

Stream.[msgId] AS [MessageID],

Stream.[speed] AS [WindSpeed],

Stream.[depreciation] AS [Depreciation],

Stream.[power] AS [Power],

Ref.[Altitude] AS [Altitude],

Ref.[Longitude] AS [Longitude],

Ref.[Latitude] AS [Latitude],

Ref.[CutOutSpeed] AS [CutOutSpeed],

CASE

WHEN Stream.[speed] > Ref.[CutOutSpeed] THEN 1

ELSE 0

END AS [CutOutSpeedAlarm],

Ref.[Repair] AS [Repair],

CASE

WHEN Stream.[depreciation] < Ref.[Repair] THEN 1

ELSE 0

END AS [RepairAlarm],

Stream.[time] AS [LocalTime],

Stream.[EventEnqueuedUtcTime] AS [CreatedAt]

FROM

[iothub] Stream

JOIN [devicerules] Ref

ON

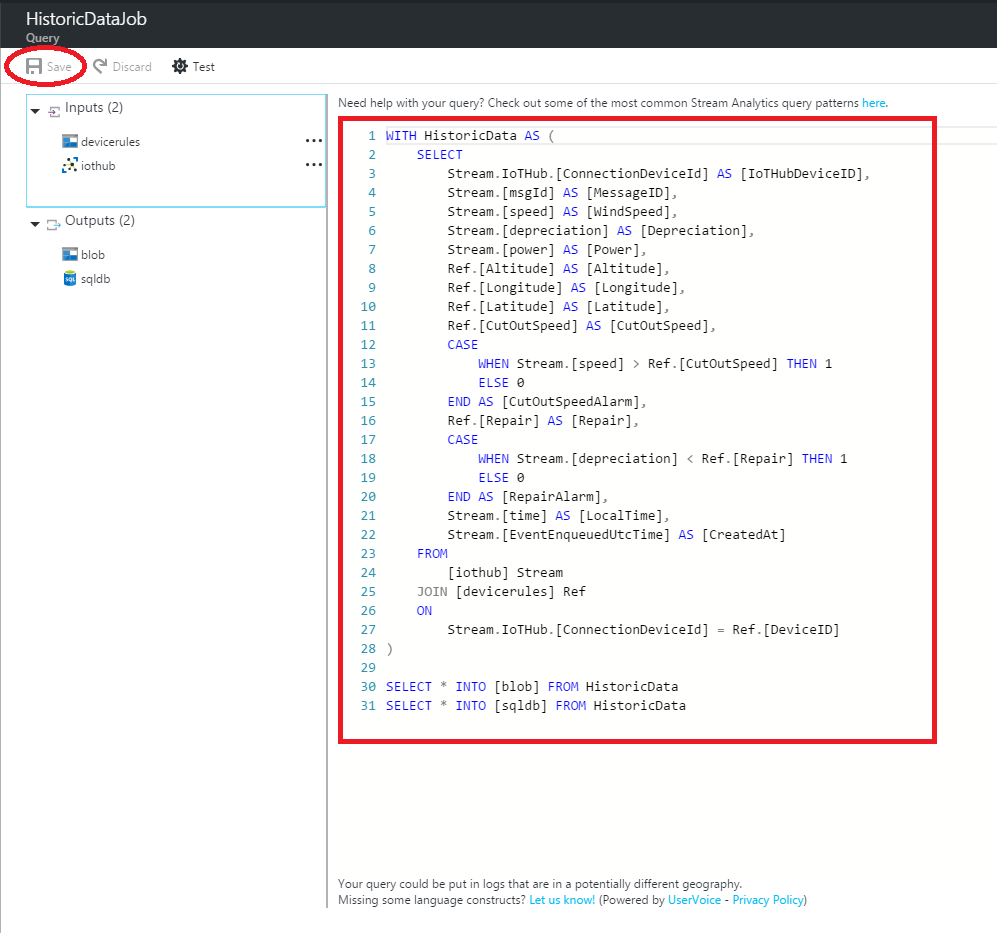
Stream.IoTHub.[ConnectionDeviceId] = Ref.[DeviceID]

)

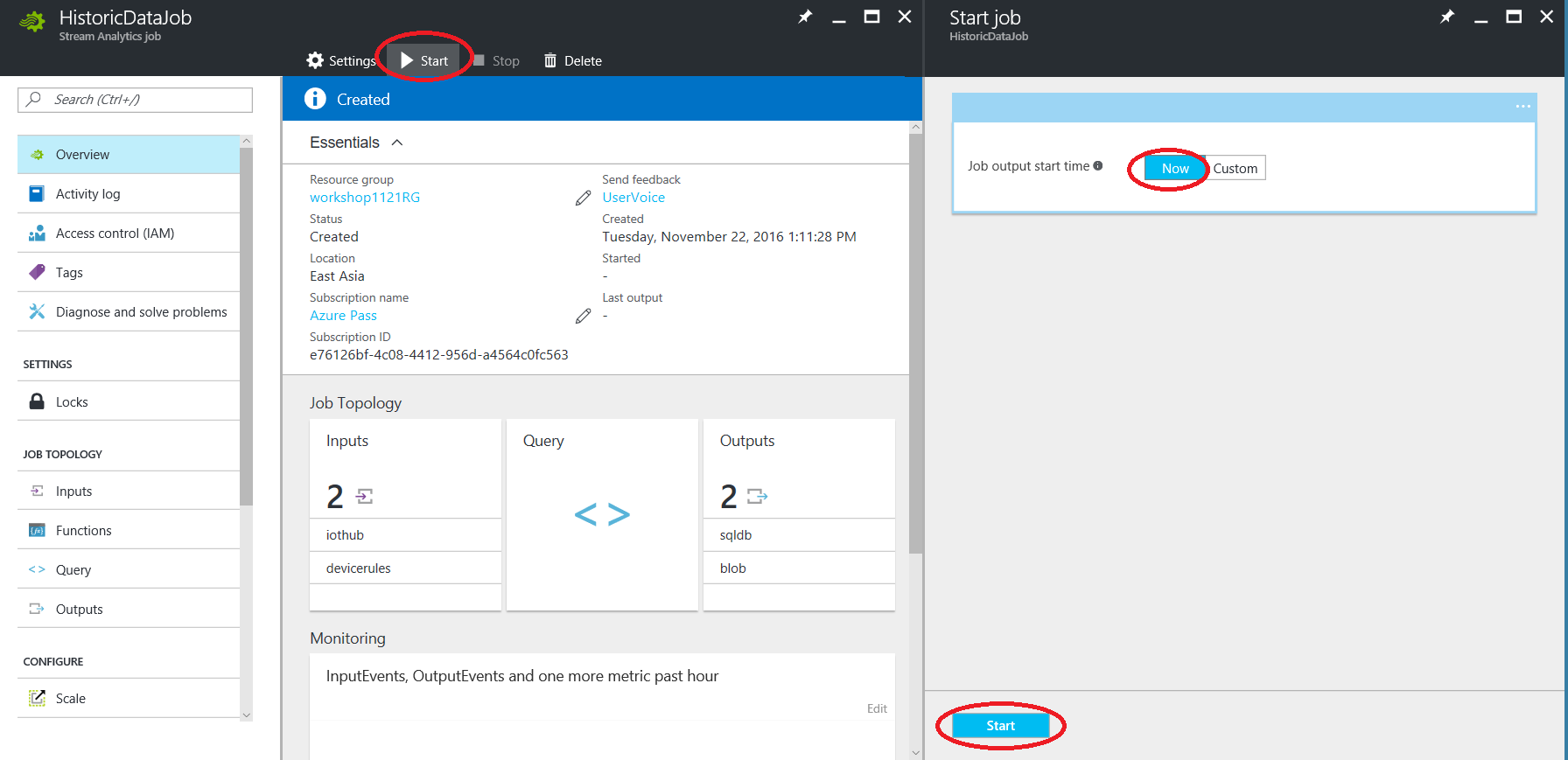
SELECT \* INTO [blob] FROM HistoricData

SELECT \* INTO [sqldb] FROM HistoricData

* + - Save



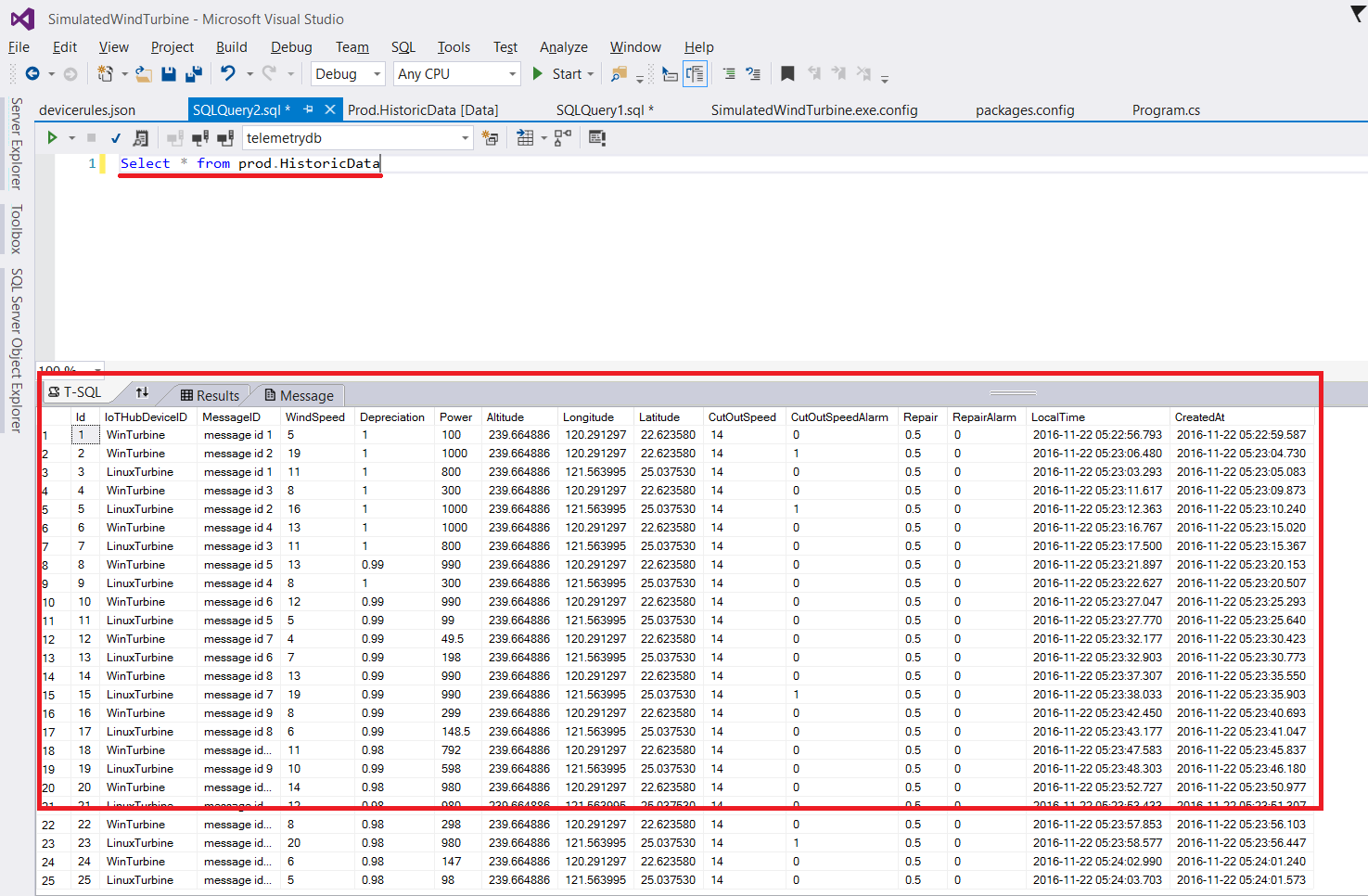
* + - Start now



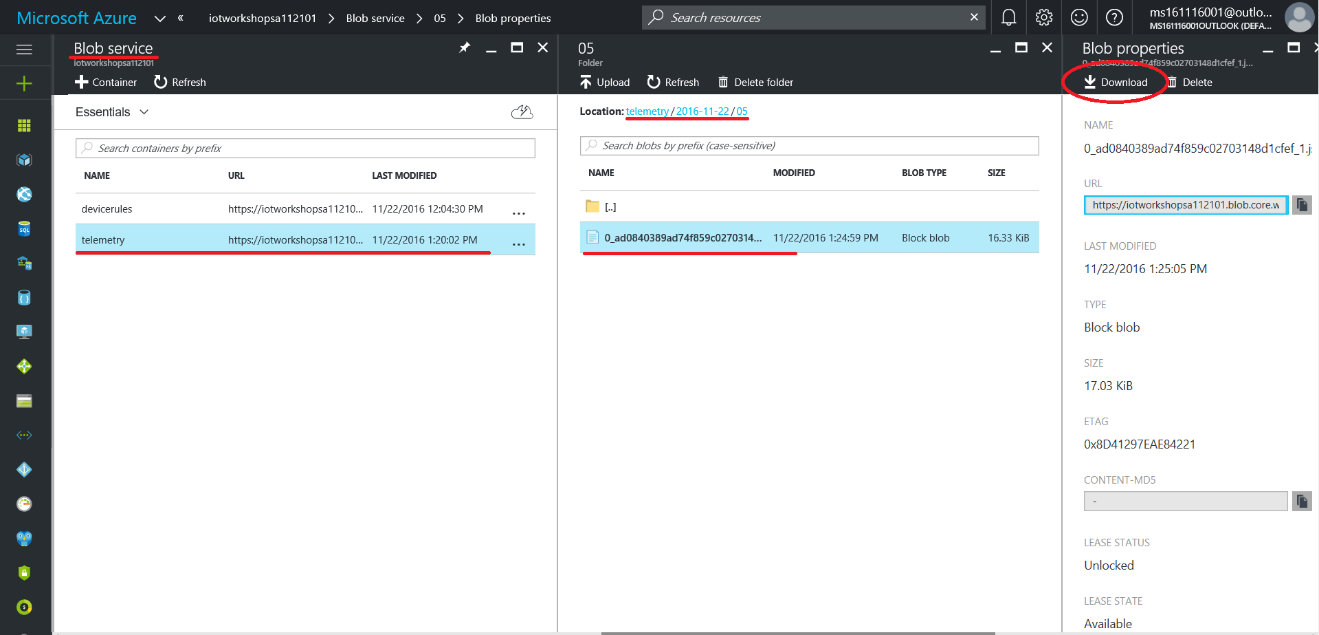
Step 5: Watch the historic data in SQL DB and Blob Storage

* **New Query** from the **telemetrydb** database in VS

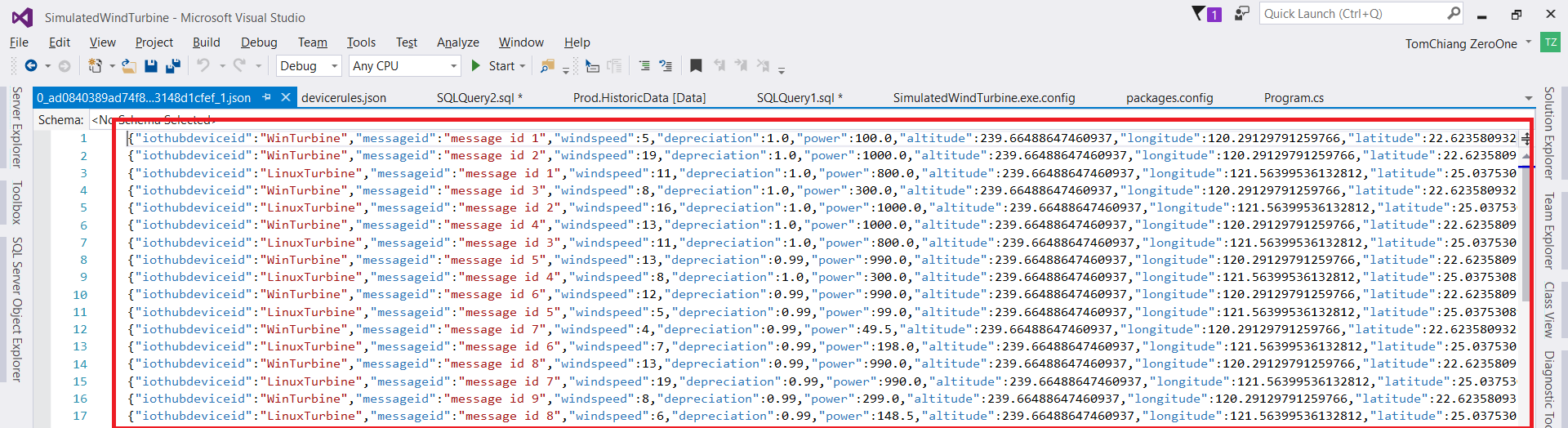
Select \* from prod.HistoricData



* Find and download the blob in the Storage Account



* + View the content of blob



* *The HOL 4 has been completed. You have already learned how to process the telemetry data from the devices, and join a device rules by another input source from blob. By the real-time data processing with a query pattern, we output the historical data into the Azure SQL Database and Azure Blob Storage through Azure Stream Analytics job.*
* *The next hands-on lab, we will implement a rule alarm and use the Service Bus to send the cloud-to-device commands to the device.*