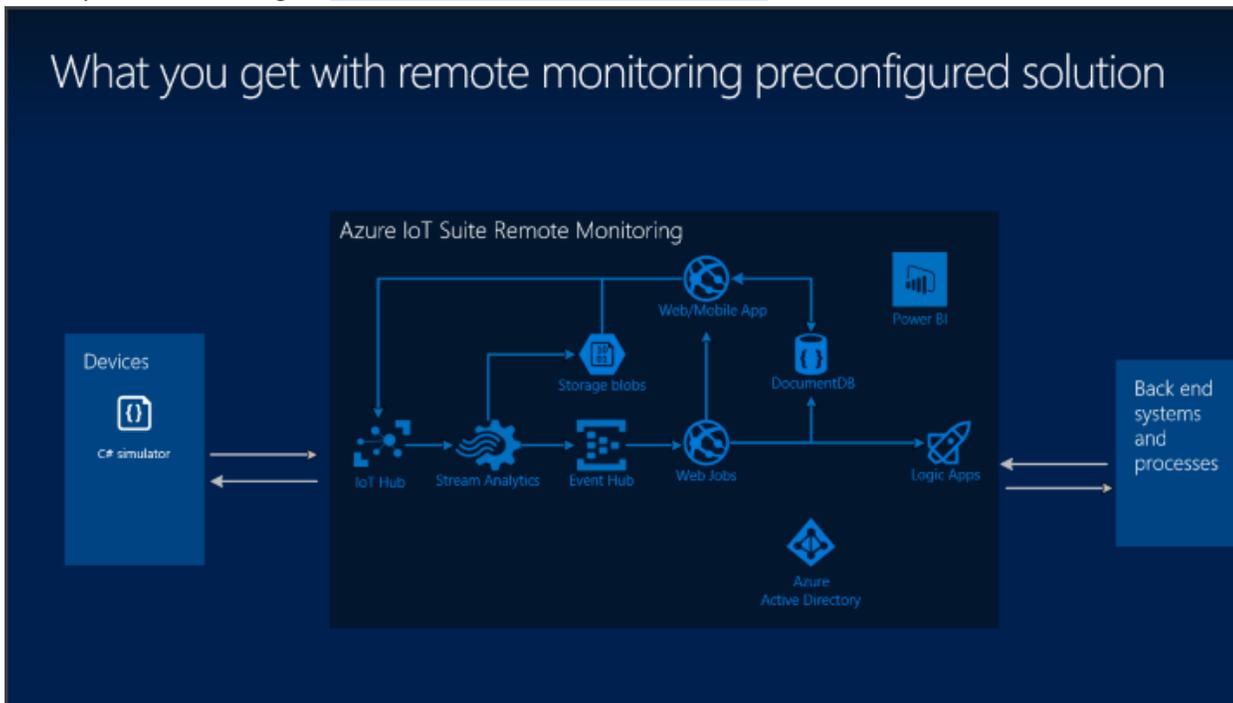


IoT application demonstration

David Chen , Dec./2015 Rev 1.0

1.0 Purpose :

Understand how an IoT structure and how implement IoT client to works with Cloud . The following instructions describe the steps for connecting an [mbed-enabled Freescale FRDM-K64F](#) device to Azure IoT Hub.



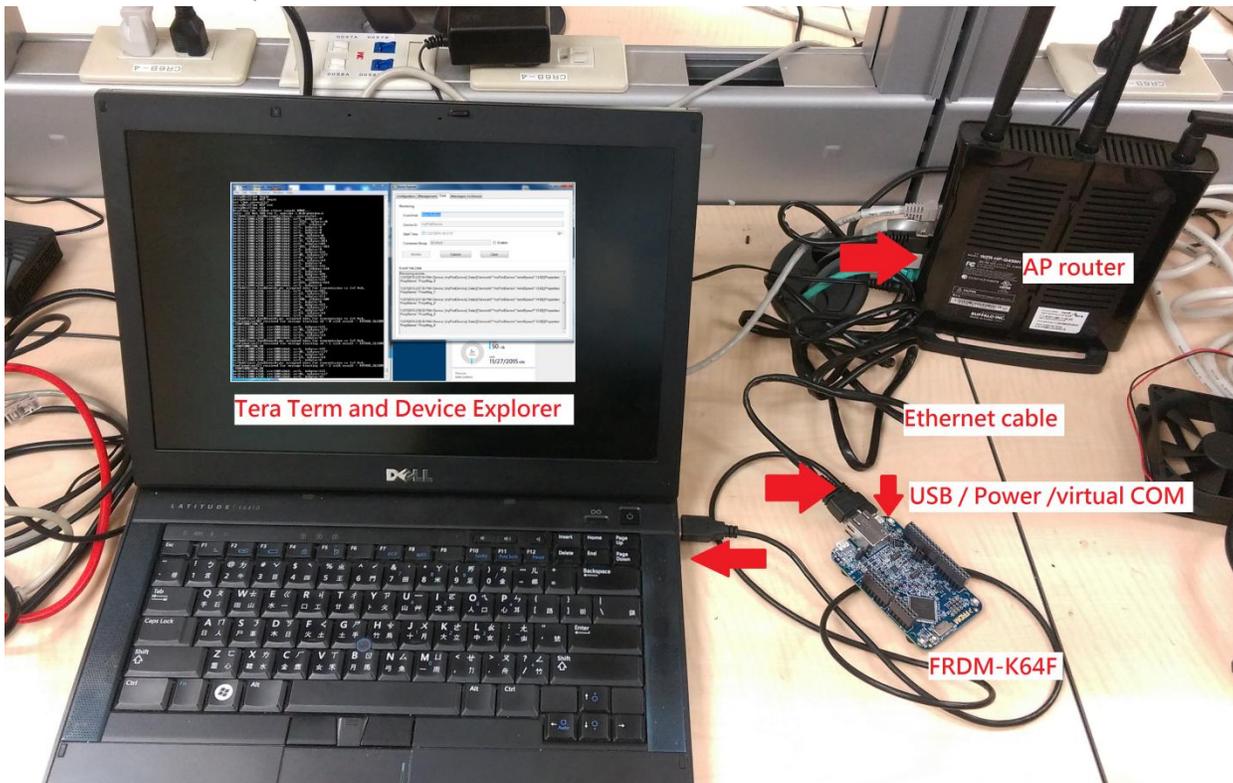
1.1 Demonstration :

1. IoT client (FRDM-K64F) report data to Cloud (Microsoft Azure)
2. IoT client receive data from Cloud
3. Could computing IoT client data and take action

1.2 Tool Requirement :

1. FRDM-K64F (<http://www.freescale.com/FRDM-K64F>)
2. Device Explorer (<http://aka.ms/iot-hub-how-to-use-device-explorer>)
3. Visual Studio 2015
4. SSH client (PuTTY or Tera Term)
5. mbed (<http://www.mbed.com>)
6. Microsoft Azure (<https://azure.microsoft.com>)

1.3 Hardware setup:



2.0 Setup the demo

2.1 : Create an Microsoft Azure IoT Hub

2.1.1 IoT Hub

Azure IoT Hub is a fully managed service that enables reliable and secure bi-directional communications between millions of IoT devices and an application back end.

More IoT Hub information please refer to below Microsoft link .

- IoT Hub documentation

<https://azure.microsoft.com/en-us/documentation/services/iot-hub/>

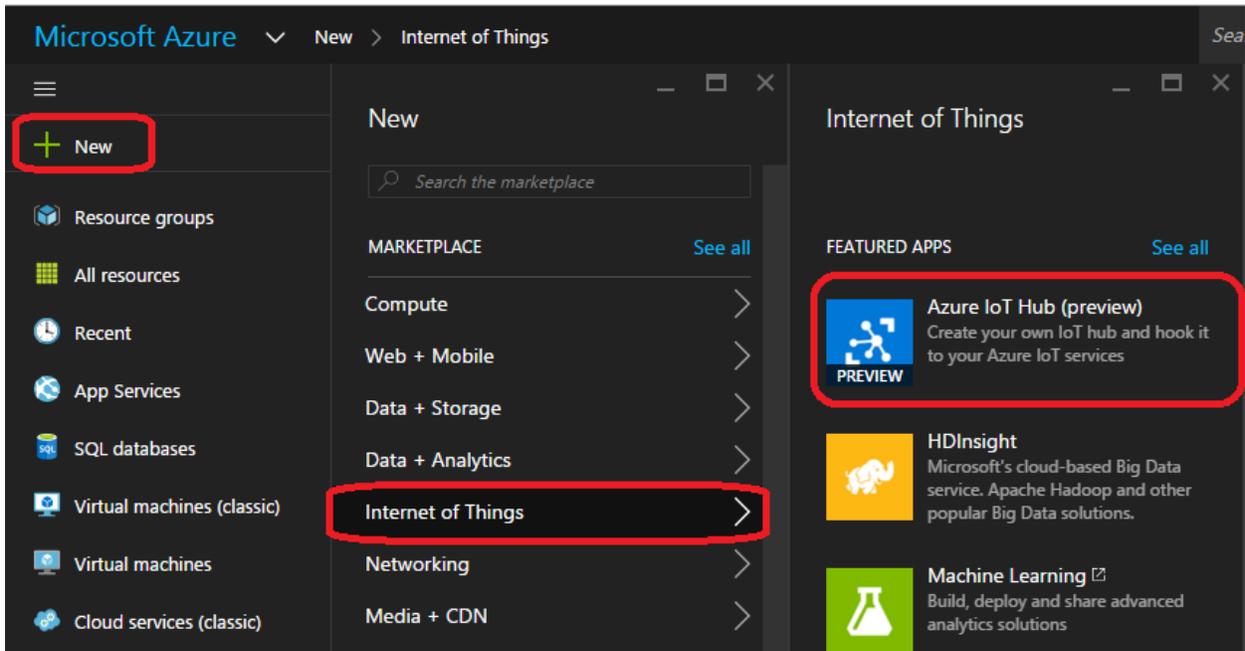
- Azure IoT Hub developer guide

<https://azure.microsoft.com/en-us/documentation/articles/iot-hub-devguide/>

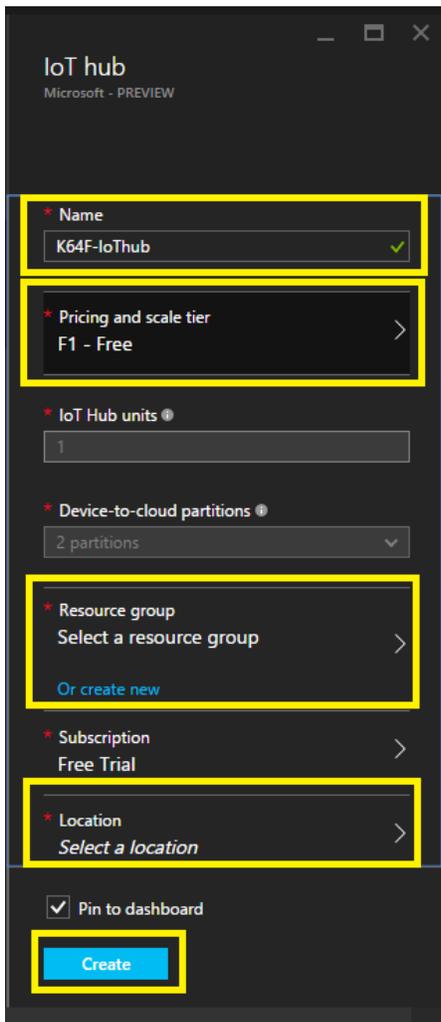
2.1.2 Step of create IoT Hub

Step 1 : Log on to the [Azure portal](#).

Step 2: In the jumpbar, click **New**, then click **Internet of Things**, and then click **IoT Hub**.



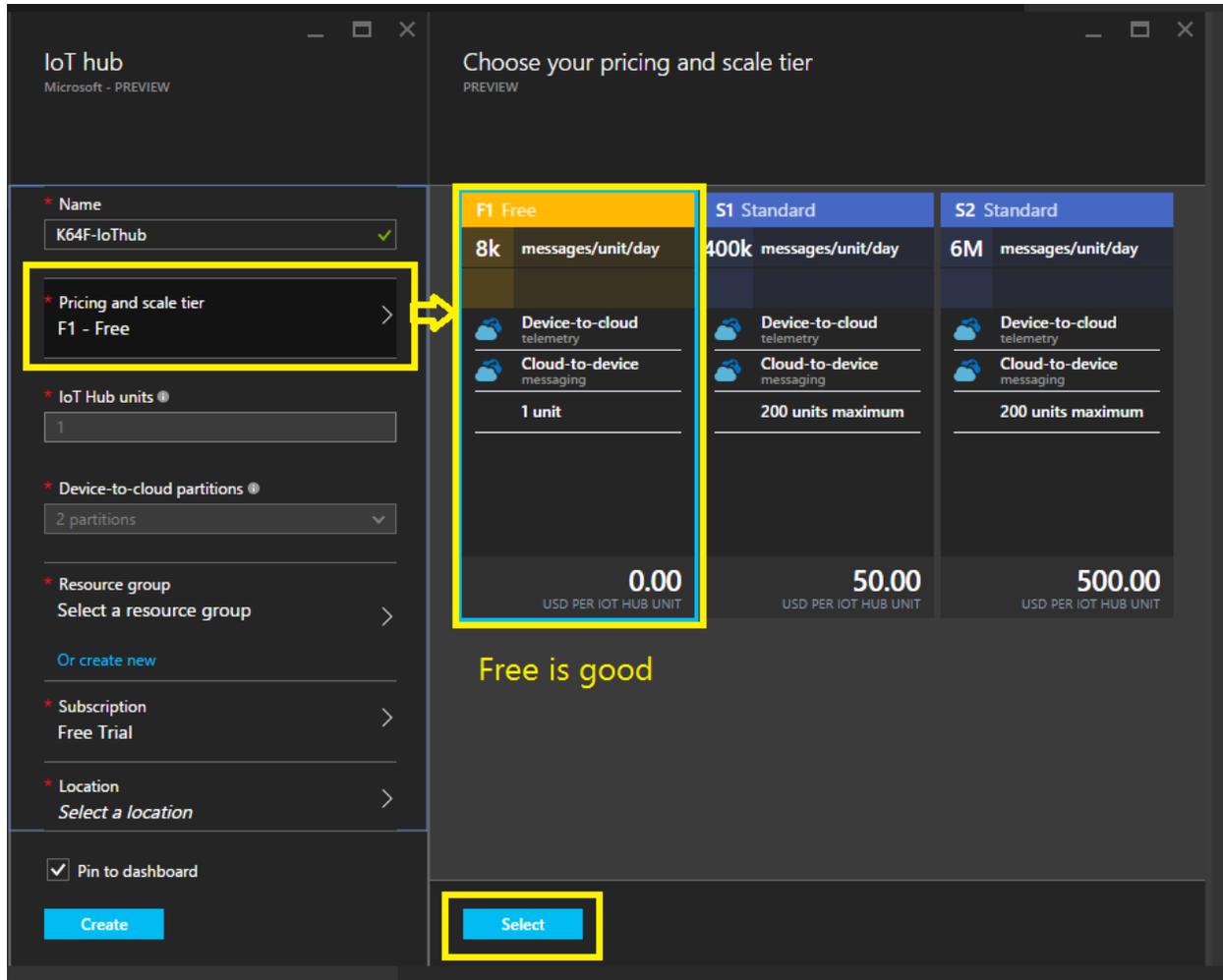
Step 3: In the **New IoT Hub** blade, specify the desired configuration for the IoT Hub.



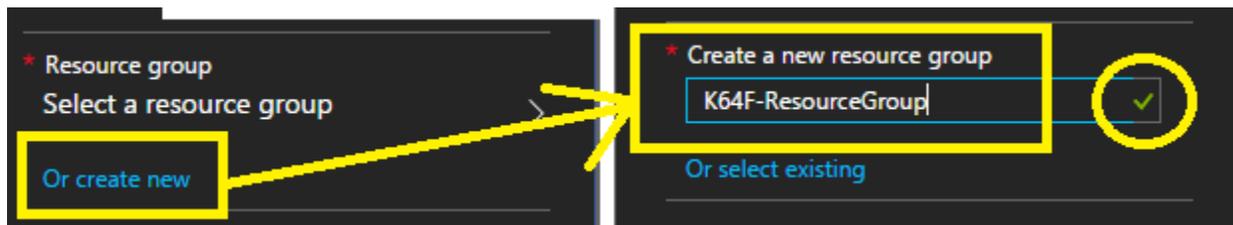
In the Name box, enter a name to identify your IoT hub. When the Name is validated, a green check mark appears in the Name box.



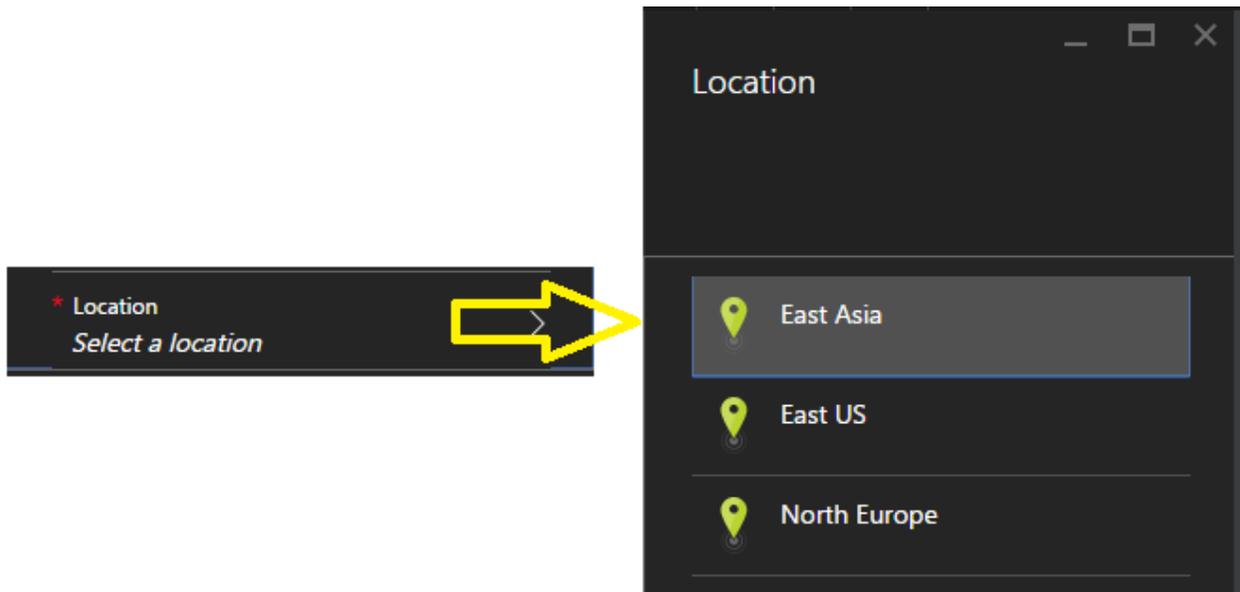
Change the Pricing and scale tier *as desired*. This demo does not require a specific tier.



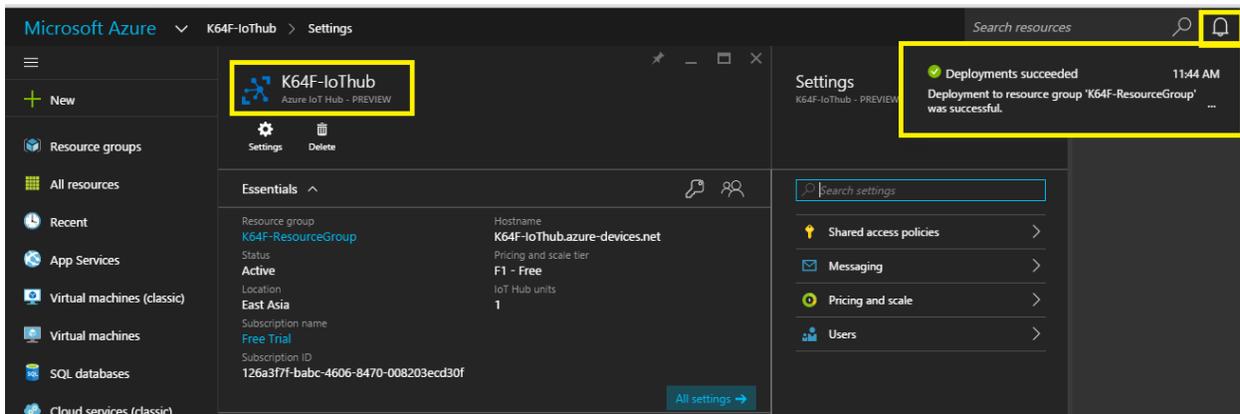
In the Resource group box, create a new resource group, or select an existing one. For more information, see [Using resource groups to manage your Azure resources](#).



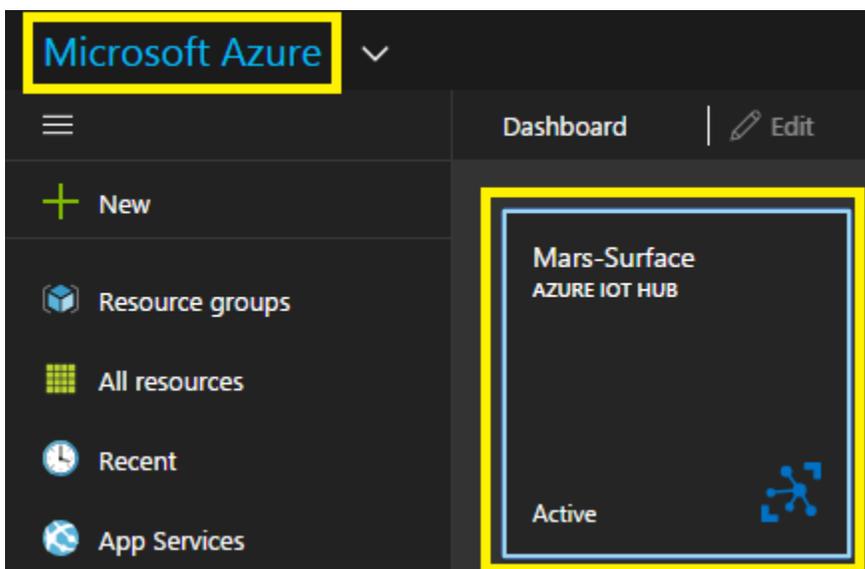
Use Location to specify the geographic location in which to host your IoT hub.

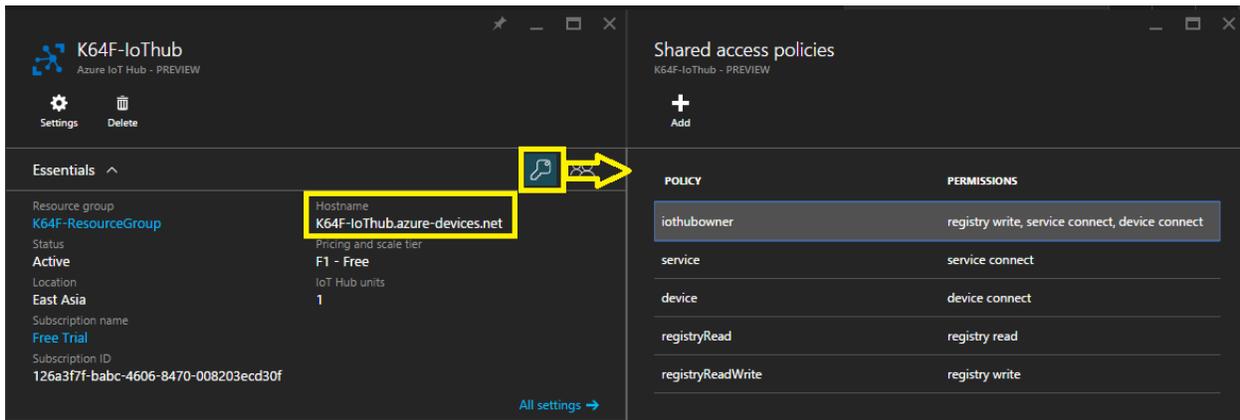


Step 4 : Once the new IoT hub options are configured, click **Create**. It can take a few minutes for the IoT hub to be created. To check the status, you can monitor the progress on the Startboard. Or, you can monitor your progress from the Notifications section.

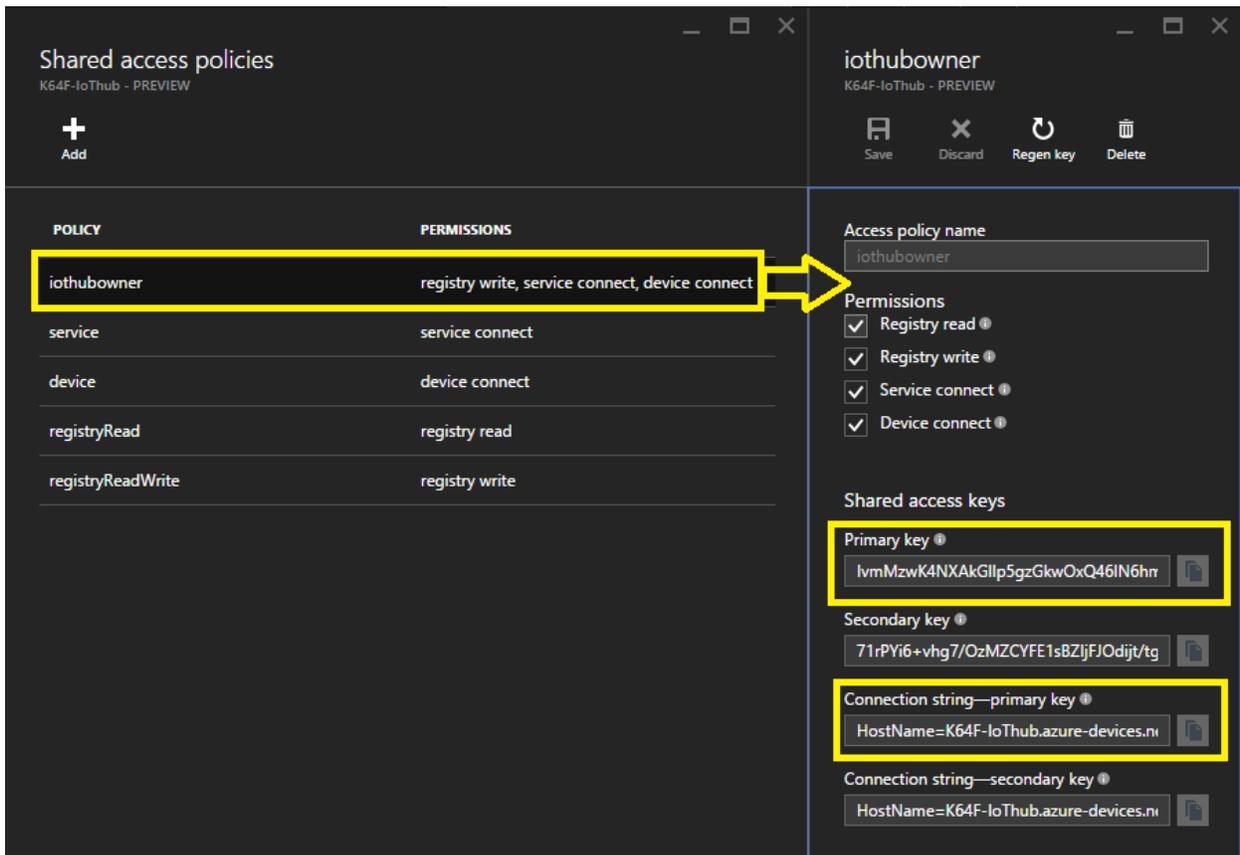


Step 5 : After the IoT hub has been created successfully, open the blade of the new IoT hub, take note of the **Hostname** , and select the **Key** icon on the top.





Step 6 : Select the Shared access policy called **iothubowner**, then copy and take note of the **Primary key** and **Connection string** on the right blade.



IoT hub is now created, and we have the **Hostname** ,**Primary Key** and **Connection string** that we need to configure IoT client .

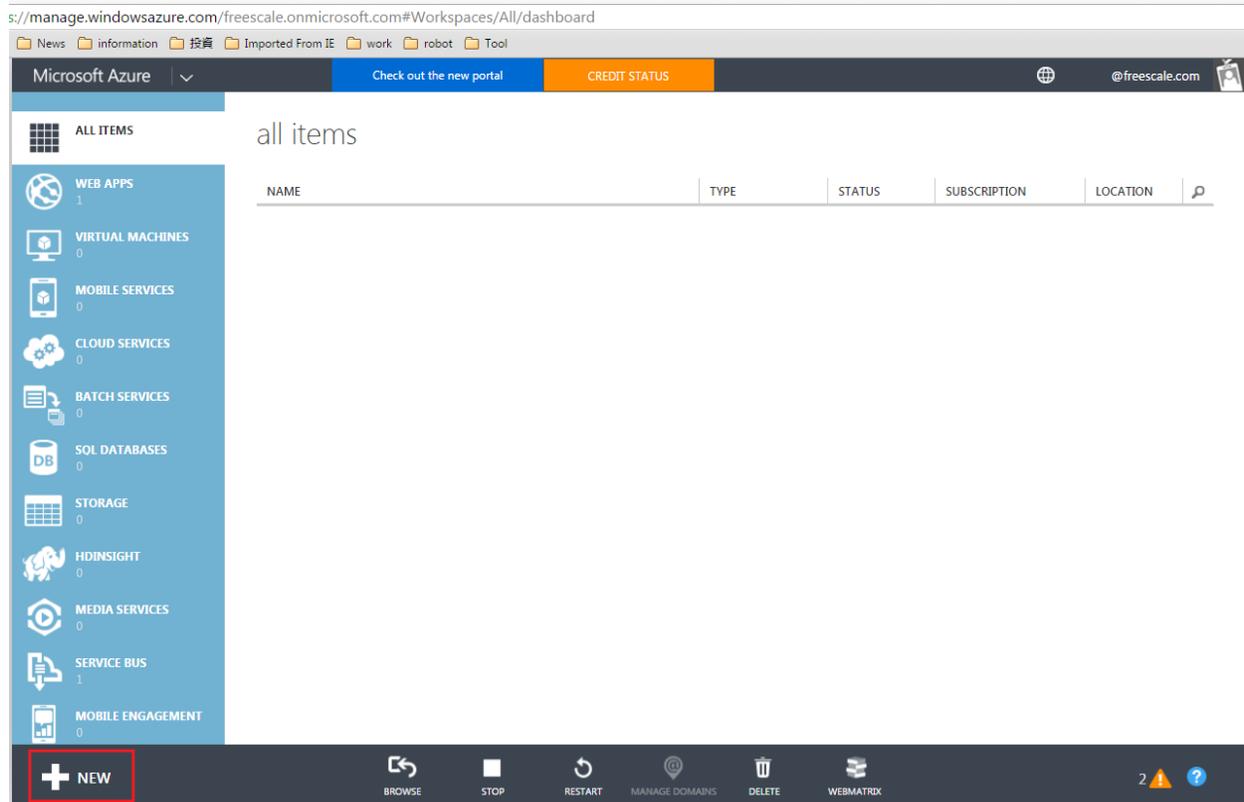
2.2 Create an Microsoft Azure Event Hub

2.2.1 Event Hub

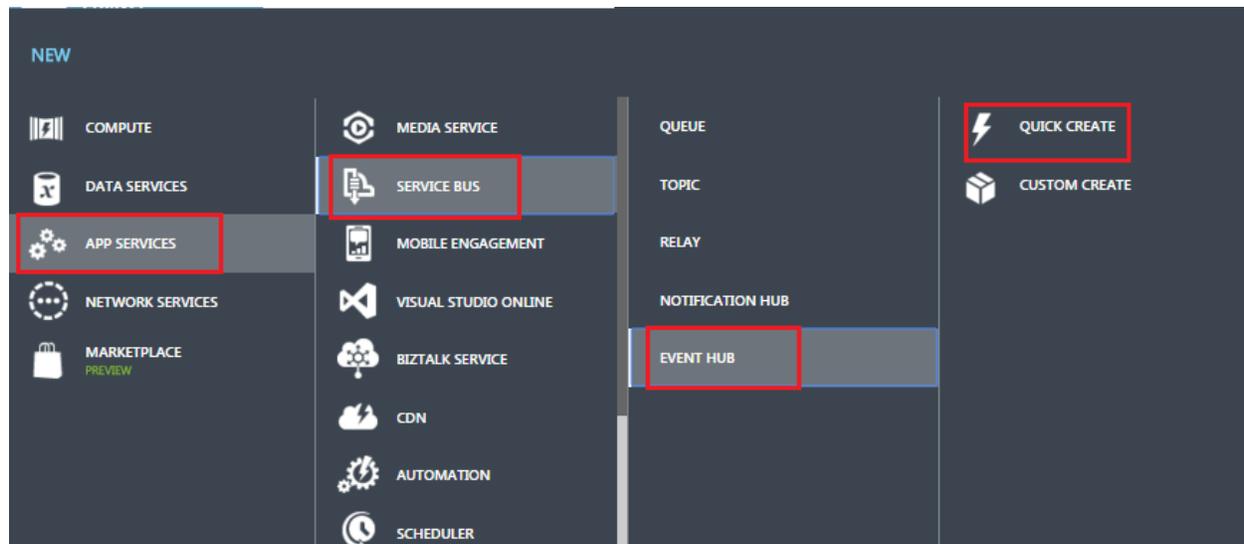
2.2.2 : Step of Event Hub

Step 1 : Log on to the [Azure Management Portal](#).

Step 2: In the lower left corner of the page, click on the + **NEW** button.

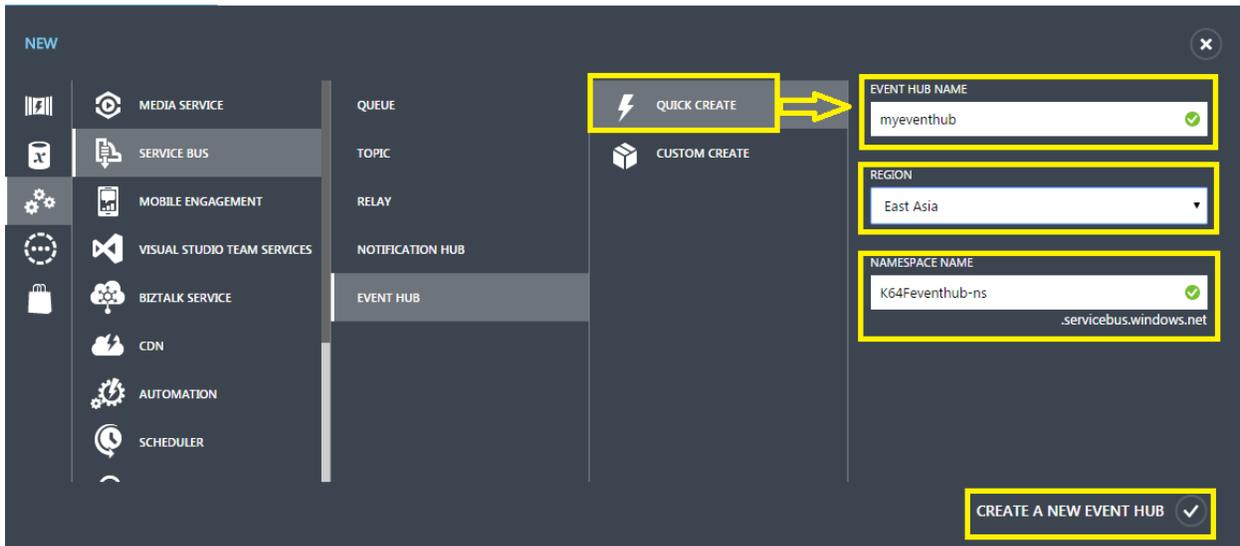


Step 3: Select **App Services, Service Bus, Event Hub, Quick Create**

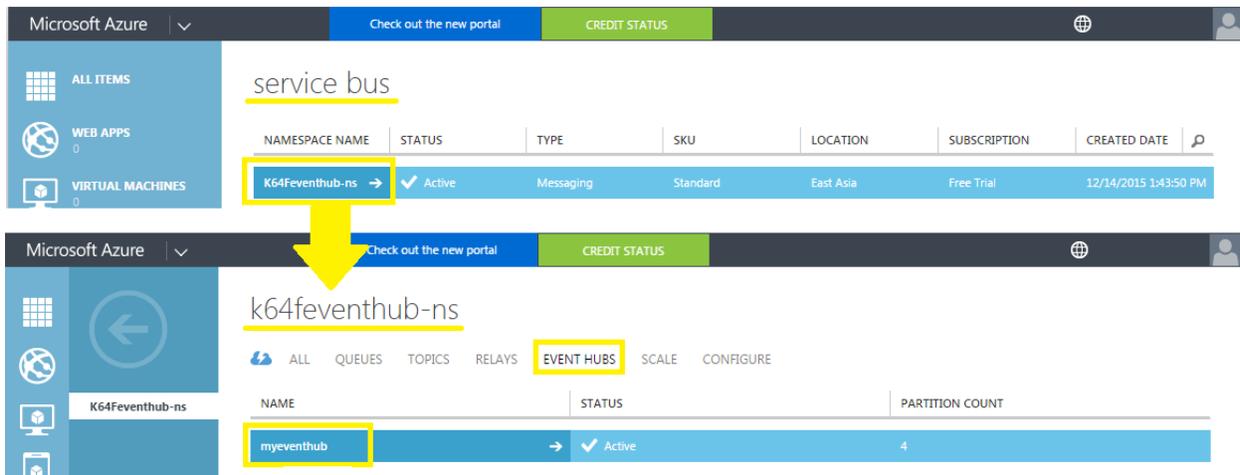


Step 4: Enter the following settings for the Event Hub (use a name of your choice for the event hub and the namespace):

- Event Hub Name: "*myeventhubname*"
- Region: your choice
- Namespace Name: "*k64Feventhub-ns*"



Step 5 : Select the *K64Feventhub-ns* (namespace) and go in the **Event Hub** tab



Step 6 : Select the *myeventhubname* event hub and go in the **Configure** tab , in the **Shared Access Policies** section, add a new policy:

- Name = "readwrite"
- Permissions = Send, Listen

Microsoft Azure | Check out the new portal | CREDIT STATUS

k64feventhub-ns

ALL | QUEUES | TOPICS | RELAYS | **EVENT HUBS** | SCALE | CONFIGURE

NAME	STATUS	PARTITION COUNT
myeventhub	Active	4

Microsoft Azure | Check out the new portal | CREDIT STATUS

myeventhub

DASHBOARD | **CONFIGURE** | CONSUMER GROUPS

general

MESSAGE RETENTION: 1 days

EVENT HUB STATE: Enabled

PARTITION COUNT: 4 Partitions

shared access policies

NAME	PERMISSIONS
readwrite	Send, Listen
<input type="text" value="NEW POLICY NAME"/>	<input type="checkbox"/> Manage

Send

Listen

+ NEW | **SAVE** | DISCARD | 1 | 1

Step 7: Click **Save**, then you will find there are two shared policy access key generated . Write down the "Policy Name" and "Primary Key" .

Microsoft Azure | Check out the new portal | CREDIT STATUS

MESSAGE RETENTION: 1 days

EVENT HUB STATE: Enabled

PARTITION COUNT: 4 Partitions

shared access policies

NAME	PERMISSIONS
readwrite	Send, Listen
<input type="text" value="NEW_POLICY_NAME"/>	<input type="text"/>

shared access key generator

POLICY NAME: readwrite

PRIMARY KEY: M8NpQ39Z45fk7ldjPRT4iwdYImSxh6LmUzgMJxuZzk=

SECONDARY KEY: cM9RzSZ3NqaT065iWfanH8MkLI2uyqb0gEpOdjujla0=

Step 8 : Then go to the event hub **Dashboard** tab and click on **Connection Information** at the bottom . Write down the connection string for the readwrite policy name.

Microsoft Azure | Check out the new portal | CREDIT STATUS

myeventhub

DASHBOARD | CONFIGURE | CONSUMER GROUPS

INCOMING MESSAGES | INCOMING THROUGHPUT | 7 MORE

RELATIVE | 1 HOUR

+ NEW | + CREATE CONSUMER GROUP | **CONNECTION INFORMATION** | DELETE

Access connection information

Use this connection information to connect to event hub 'myeventhub'.

SAS ?

NAME	CONNECTION STRING
readwrite	Endpoint=sb://k64feventhub-ns.servicebus.windows.net;/SharedAccessKeyName=rea

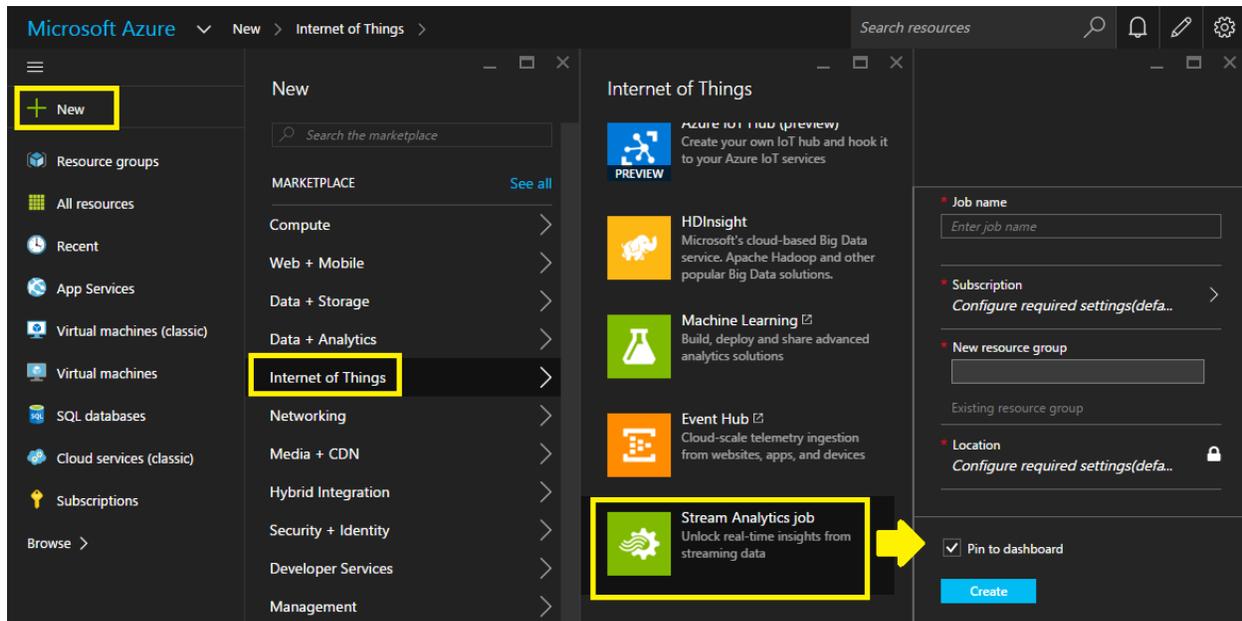
2.3 Create a Stream Analytics job

2.3.1 Stream Analytics

2.3.2 Step of create a Stream Analytics job

Step 1 : Log on to the [Azure Preview Portal](#).

Step 2 : In the jumpbar, click **New**, then click **Internet of Things**, and then click **Stream Analytics job**.



Step 3 : Enter a name for the job, a preferred region, choose your subscription. At this stage you are also offered to create a new or to use an existing resource group.

Subscription

* Job name
K64FStreamAnalyticsJob 

* Subscription
Configure required settings(defa... 

* New resource group

Existing resource group

* Location
Configure required settings(defa... 

Pin to dashboard

Create

 Free Trial
126a3f7f-babc-4606-8...

Resource group

Resource groups are containers that help you manage a collection of Azure resources. [Learn more](#)

* Job name
K64FStreamAnalyticsJob ✓

* Subscription
Free Trial >

Resource group
Configure required settings >

Enter resource group name

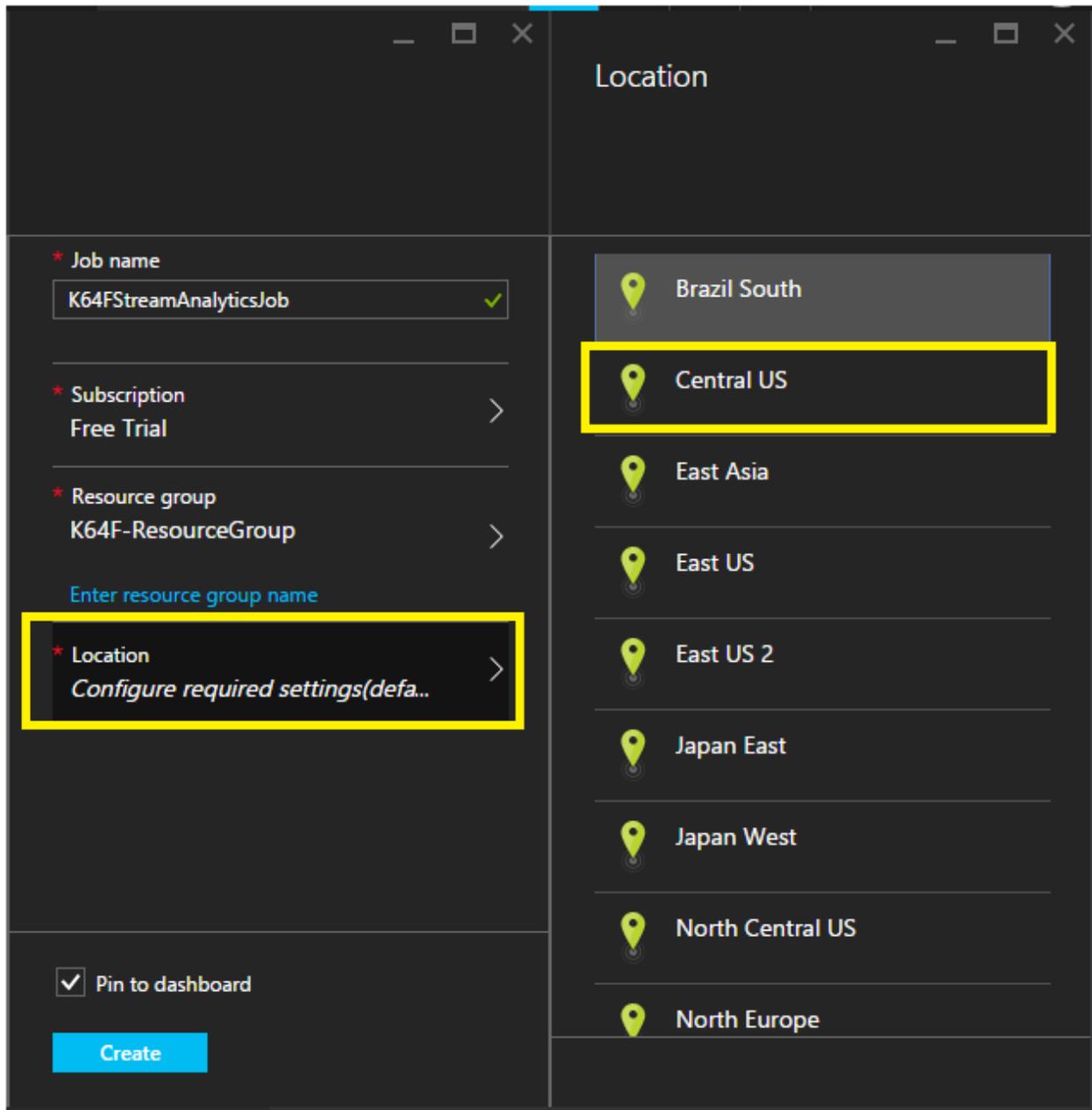
* Location
Configure required settings(defa... >

Pin to dashboard

Create

Default-ServiceBus-EastAsia
eastasia

K64F-ResourceGroup
eastasia



Step 4 : After the job is created , click on the **Inputs** tile in the **job topology** section. In the **Inputs blade**, click on **Add**

The screenshot displays the Azure Stream Analytics job configuration page. At the top, the job name is 'K64FStreamAnalyticsJob'. Below the job name are control buttons for Settings, Start, Stop, and Delete. The 'Essentials' section provides job metadata: Resource group (K64F-ResourceGroup), Status (Created), Created time (Monday, December 14, 2015, 2:36:57 PM), Last output, Subscription id (126a3f7f-babc-4606-8470-008203ecd30f), Region (East Asia), and Subscription name (Free Trial). The 'Job Topology' section shows a diagram with three tiles: 'Inputs', 'Query', and 'Outputs'. The 'Inputs' tile is highlighted with a yellow box and contains '0' and 'No results.'. A yellow arrow points from this tile to the 'Add' button in the 'Inputs' table on the right. The 'Inputs' table has columns for NAME, SOURCE TYPE, and SOURCE, and is currently empty.

Step 5 : Enter the following settings:

- Input Alias = "tempsensors"
- Type = "Data Stream"
- Source = "IoT Hub"
- IoT Hub = "K64F-IoThub" (The name for the IoT Hub that we create at chapter 2.1.2 step 2)
- Shared Access Policy Name = "iothubowner"
- Shared Access Policy Key = "**iothubowner Primary Key**" (That's the key you wrote down when creating the IoT Hub , pls refer to chapter 2.1.2 step 6)
- IoT Hub Consumer Group = "" (leave it to the default empty value)
- Event serialization format = "JSON"
- Encoding = "UTF-8"

New input

* Input alias
tempensors ✓

* Source Type ⓘ
Data stream ▼

* Source ⓘ
IoT hub ▼

* IoT hub ⓘ
K64F-IoThub ✓

* Shared access policy name ⓘ
iothubowner ✓

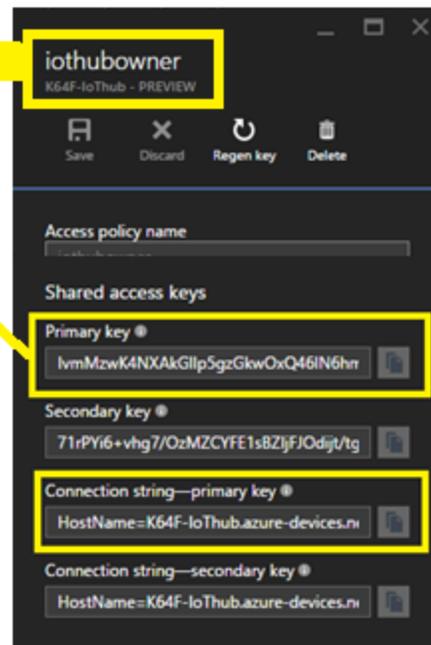
* Shared access policy key ⓘ
..... ✓

Consumer group ⓘ
.....

* Event serialization format ⓘ
JSON ▼

Encoding ⓘ
UTF-8 ▼

Create



Step 6 : Back to the Stream Analytics Job blade, click on the **Query** tile. In the Query settings blade, type in the below query and click **Save**

```
SELECT
  System.timestamp AS timestart,
```

```
ObjectName AS dsplalert,  
ObjectType AS alerttype,  
Version AS message,  
TargetAlarmDevice AS targetalarmdevice  
INTO  
eventhub  
FROM  
tempsensors  
WHERE temp>82
```

K64FStreamAnalyticsJob
Stream Analytics job

Settings Start Stop Delete

Created

Essentials

Resource group
K64F-ResourceGroup

Status
Created

Created
Monday, December 14, 2015, 2:36:57 PM

Last output
-

Subscription id
126a3f7f-babc-4606-8470-008203ecd30f

Send feedback
UserVoice

Region
East Asia

Started
-

Subscription name
Free Trial

All settings →

Job Topology

Inputs

1
tempsensors

Query

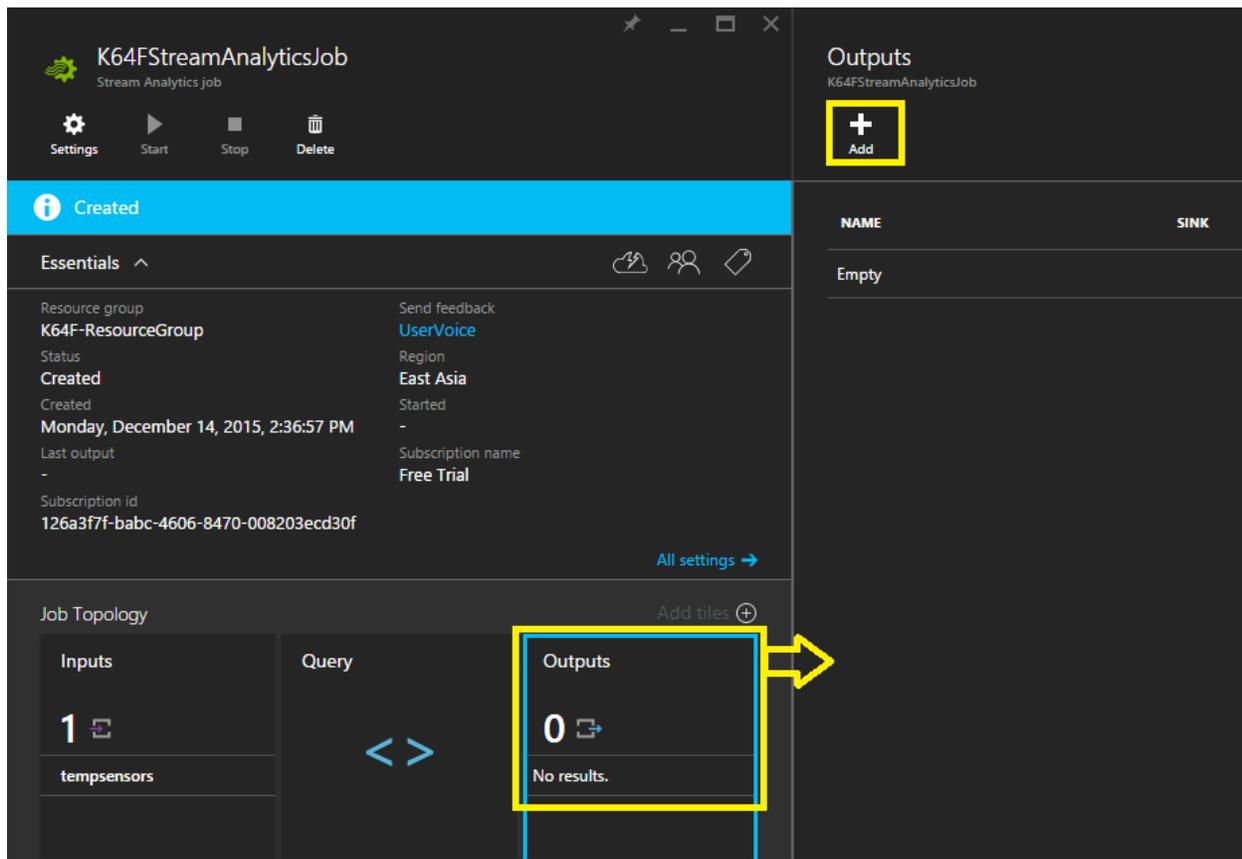
0
No results.

K64FStreamAnalyticsJob
Query

Save Discard Test

```
1 SELECT  
2 System.timestamp AS timestart,  
3 ObjectName AS dsplalert,  
4 ObjectType AS alerttype,  
5 Version AS message,  
6 TargetAlarmDevice AS targetalarmdevice  
7 INTO  
8 eventhub  
9 FROM  
10 tempsensors  
11 WHERE temp>82  
12
```

Step 7: Back to the Stream Analytics Job blade, click on the **Outputs** tile and in the Outputs blade, click on **Add**



Step 8 : Enter the following settings then click on **create**:

- Output Alias = "eventhub" (the name is used at chapter 2.3.2 step 6)
- Source = "Event Hub"
- Service Bus Namespace = "K64Feventhub-ns" (please refer to chapter 2.2.2 step 4)
- Event Hub Name = "myeventhub" (please refer to chapter 2.2.2 step 4)
- Event Hub Policy Name = "readwrite" (please refer to chapter 2.2.2 step 6)
- Event Hub Policy Key = "Primary Key for readwrite Policy name" (please refer to chapter 2.2.2 step 7)
- Partition Key Column = "4"
- Event Serialization format = "JSON"
- Encoding = "UTF-8"
- Format = "Line separated"

The image shows the configuration of a new output for a K64FStreamAnalyticsJob. The 'New output' window on the left lists the following settings:

- Output alias: eventhub
- Sink: Event hub
- Service bus namespace: k64Feventhub-ns
- Event hub name: myeventhub
- Event hub policy name: readwrite
- Event hub policy key: [Redacted]
- Partition key column: 4
- Event serialization format: JSON
- Encoding: UTF-8
- Format: Line separated

Yellow arrows point from these settings to corresponding elements in other application windows:

- The 'Output alias' 'eventhub' points to the 'INTO eventhub' line in the SQL query.
- The 'Service bus namespace' 'k64Feventhub-ns' points to the 'NAMESPACE NAME' field in the 'service bus' window.
- The 'Event hub name' 'myeventhub' points to the 'NAME' field in the 'service bus' window.
- The 'Event hub policy name' 'readwrite' points to the 'POLICY NAME' field in the 'shared access key generator' window.
- The 'Event hub policy key' points to the 'PRIMARY KEY' field in the 'shared access key generator' window.

The 'K64FStreamAnalyticsJob' window shows the following SQL query:

```

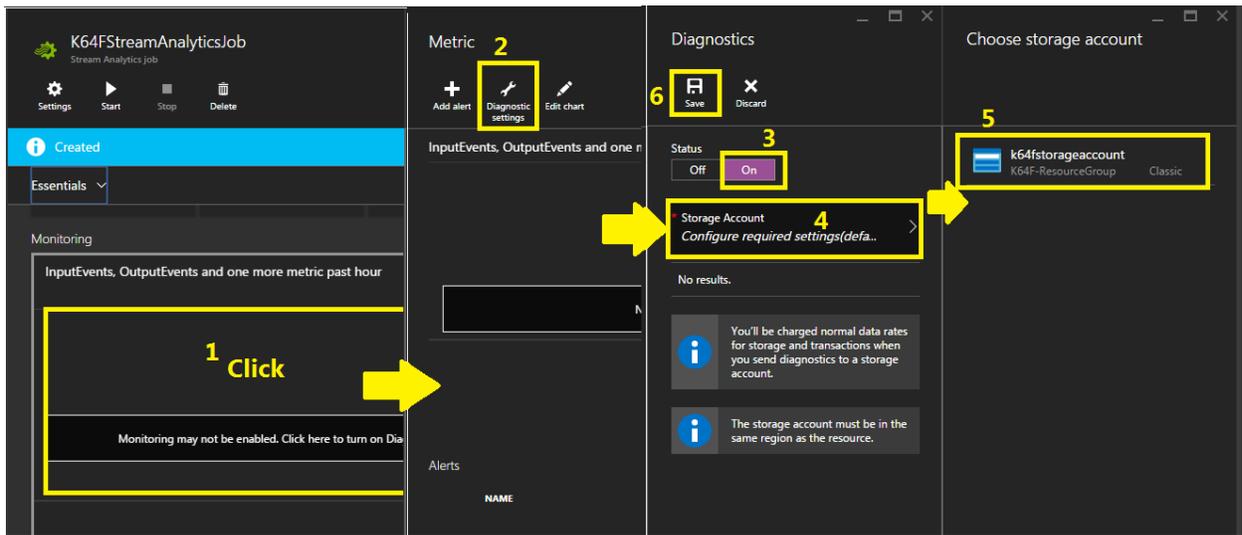
1 SELECT
2   System.timestamp AS timestart,
3   ObjectName AS dsplalert,
4   ObjectType AS alerttype,
5   Version AS message,
6   TargetAlarmDevice AS targetalarmdevice
7 INTO
8   eventhub
9 FROM
10  tempsensors
11 WHERE temp>82
  
```

The 'service bus' window shows the namespace 'k64feventhub-ns' and the event hub 'myeventhub'.

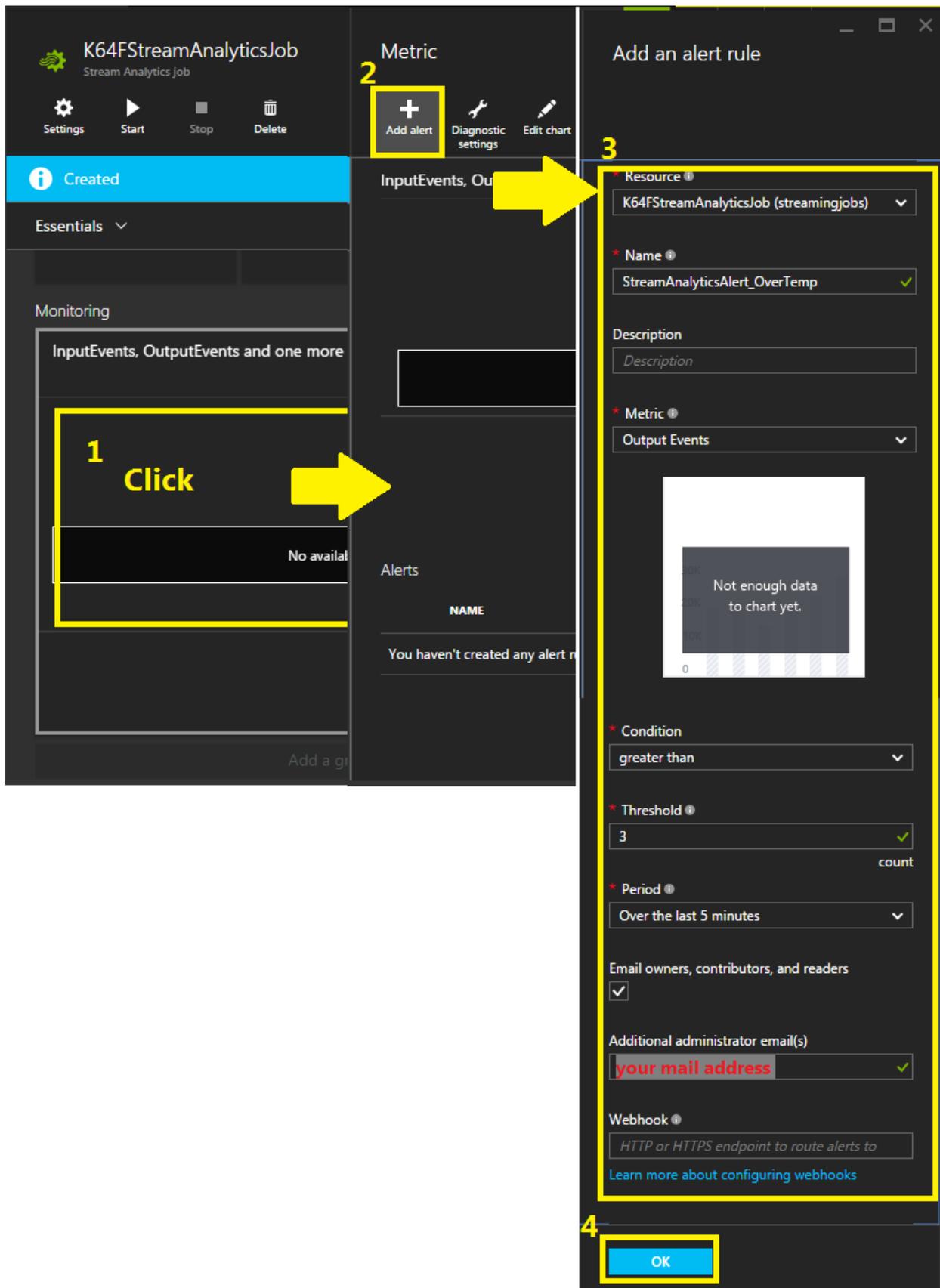
The 'shared access key generator' window shows the policy name 'readwrite' and the primary key 'M8NpQ39Z45fk7ldjPRT4iwdYImSxh6LmUzgMUxuZzk='.

A 'Create' button is highlighted at the bottom of the 'New output' window.

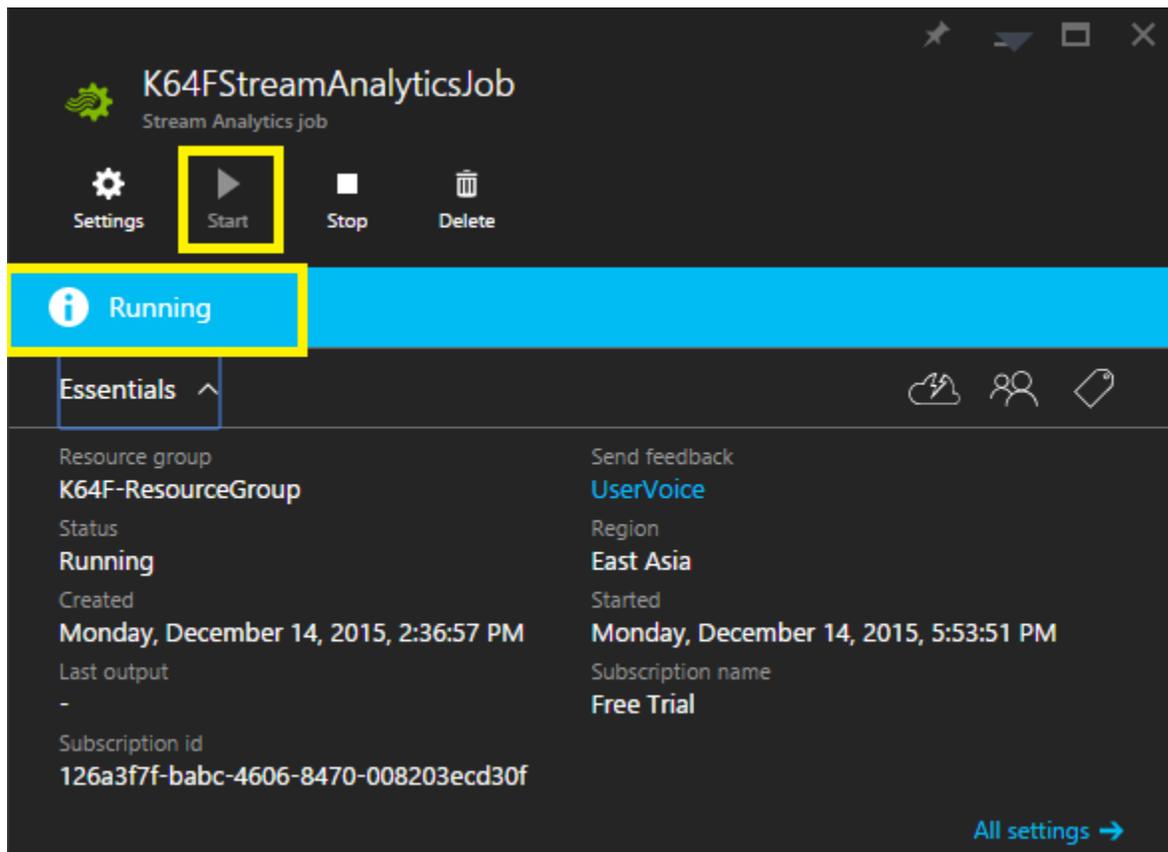
Step 9 : Configuration and Enable Monitoring .



Step 10 : Setup alert condition and action



Step 11 : Back in the Stream Analytics blade, start the job by clicking on the **Start** button at the top



2.4 Create a storage account

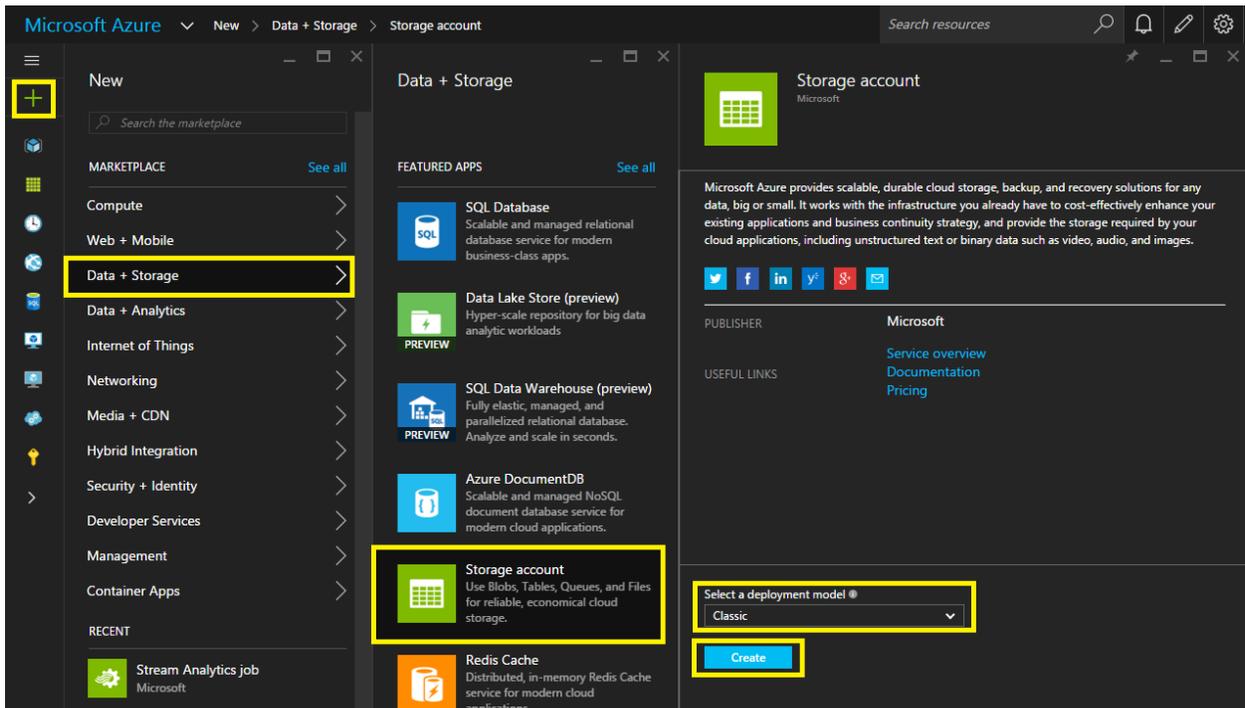
2.4.1 Storage account

2.4.2 Step of create a storage account

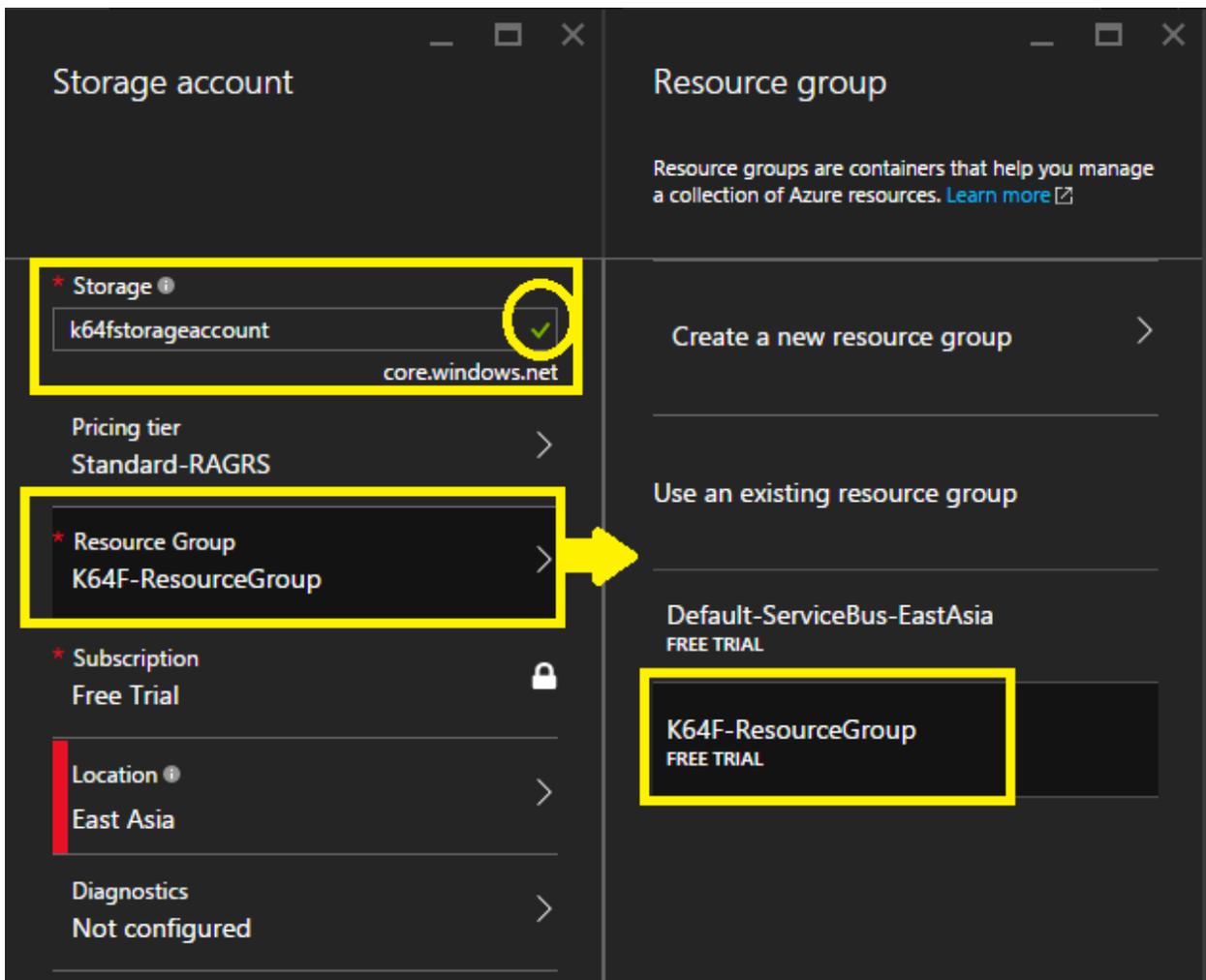
Step 1 : Log on to the [Azure Preview Portal](#).

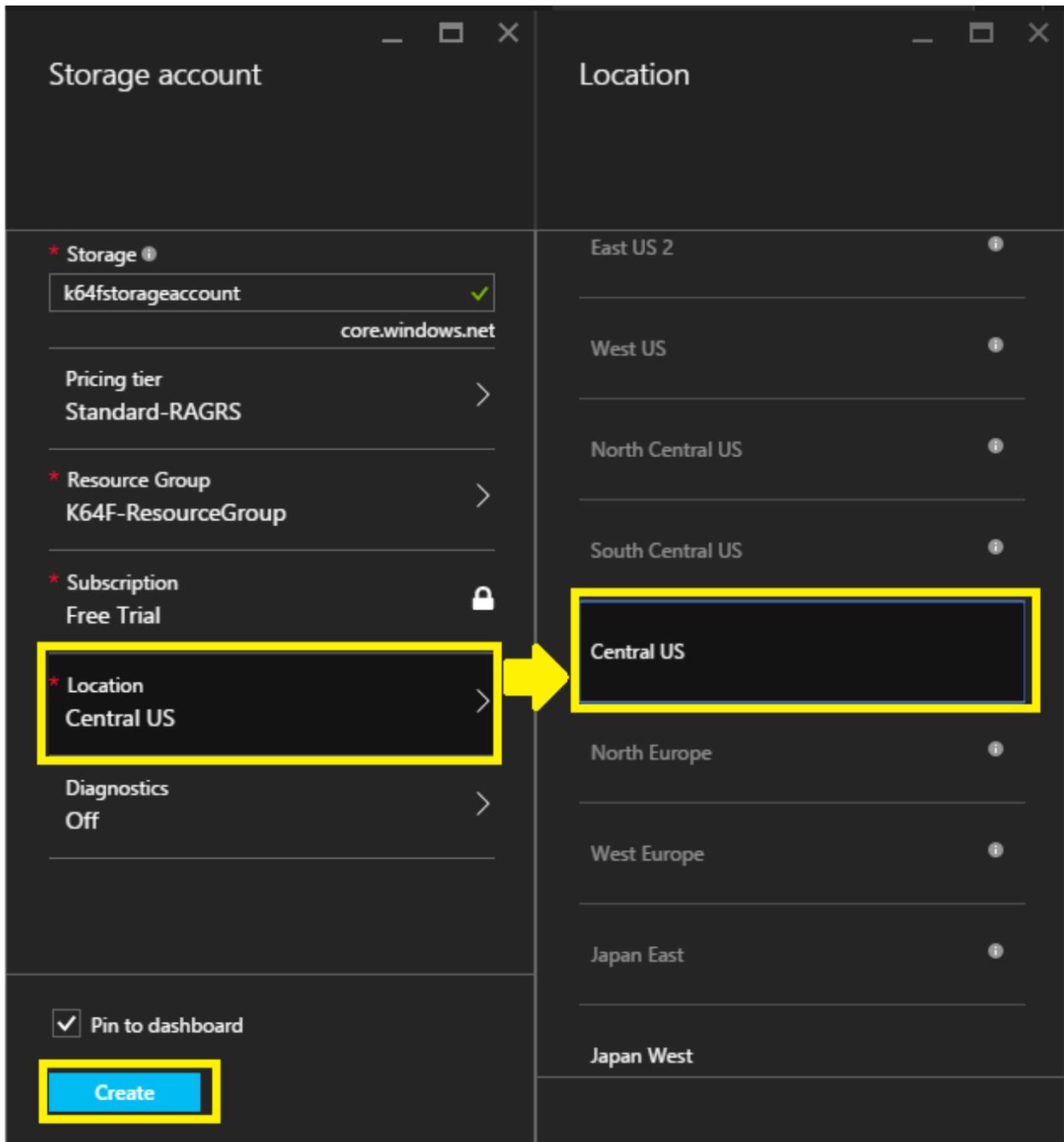
Step 2 : In the jumpbar, click **New** and select **Data + Storage** then **Storage Account**

Step 3 : Choose **Classic** for the deployment model and click on **create**

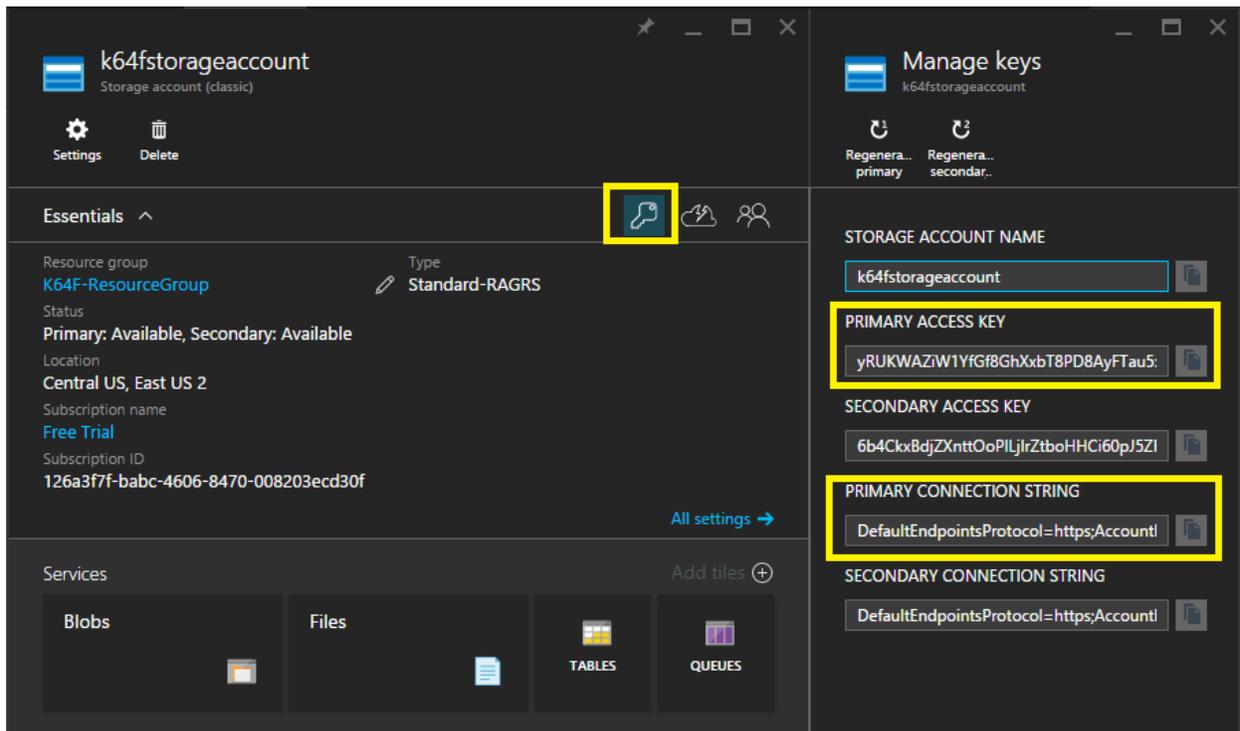


Step 4 : Enter the name of your choice (i.e. "mystorageaccountname" for the account name and select your resource group, subscription,... then click on "Create"





Step 5 : Once the account is created, find it in the resources blade and write down the primary connection string for it to configure the worker role



2.5 Create a new device identity in the IoT Hub

To connect your device to the IoT Hub instance, you need to generate a unique identity and connection string. IoT Hub does that for you. Here we use the [Device Explorer tool](#) (runs only on Windows for now) to create a new device identity .

2.5.1 Step of create device identity

Step 1 : Getting Device Explorer

You can either download a pre-built version of Device Explorer or build it yourself.

link: [Downloads](#)

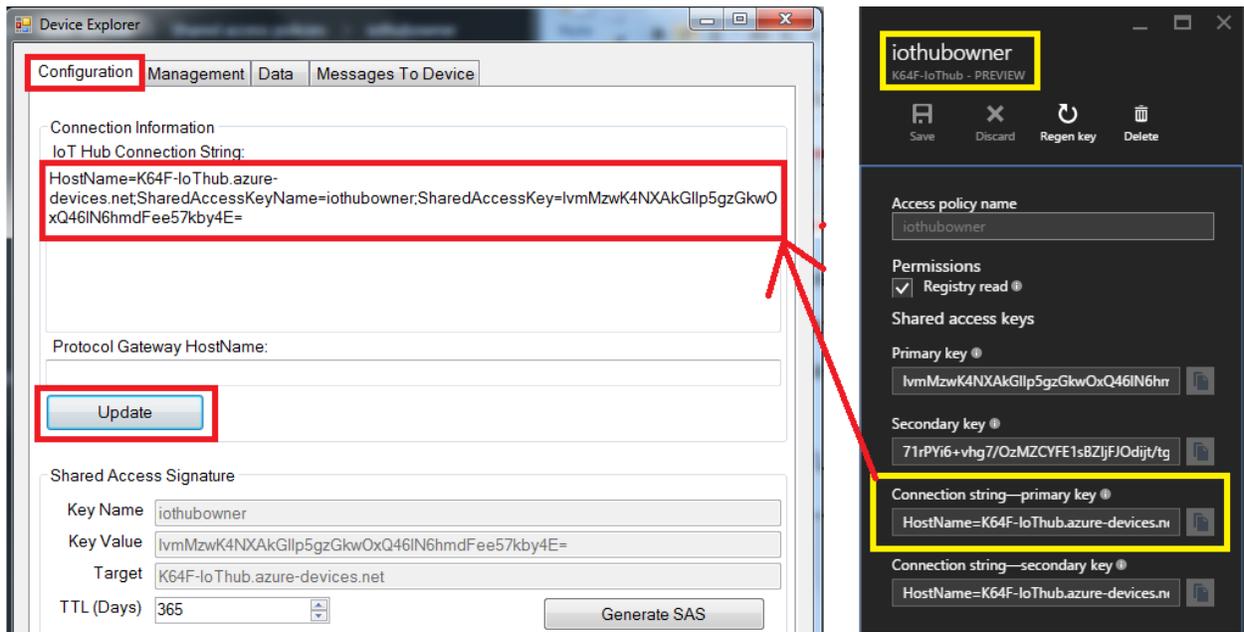
- Download a pre-built version of the Device Explorer application
→ Scroll down for SetupDeviceExplorer.msi
- Build the Device Explorer application
→ Scroll down for Source code(zip) .
→ To open the **tools\DeviceExplorer\DeviceExplorer.sln** file in your local copy of this repository ([azure-iot-sdks](#)) in Visual Studio 2015. Then build and run the solution.

Step 2 : Configure an IoT Hub connection

Execute Device Explorer , In the **Configuration** tab, add the connection string for your IoT Hub.

The IoT Hub connection string is what we created at chapter 2.1.2 step 6 .

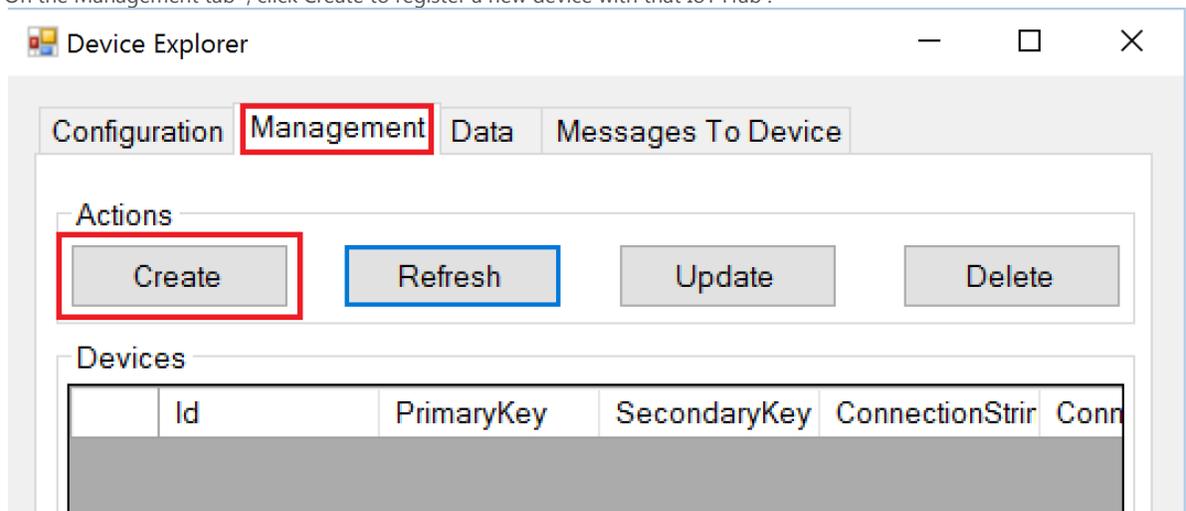
Then click **Update**.



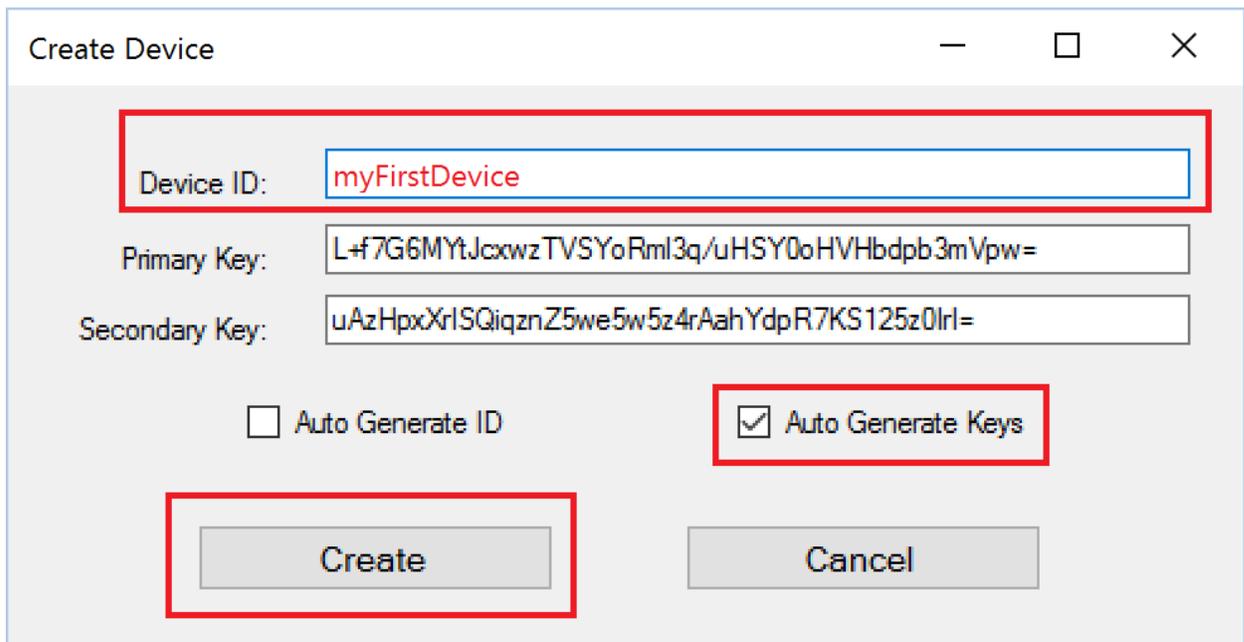
Step 3 : Create device

Creating a device adds device details to the device identity registry. **IoT Hub uses this information to generate a device-specific connection string that enables the device to connect to the IoT hub.**

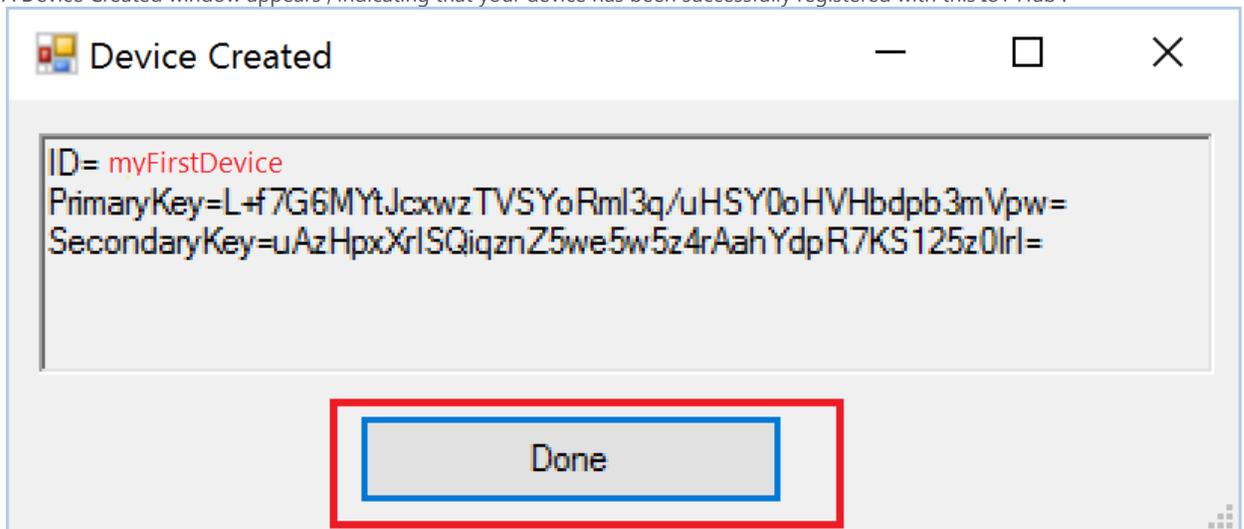
- Click the Management tab
- On the Management tab , click Create to register a new device with that IoT Hub .



- The **Create Device** dialog appears . In the **Device ID** field , type a unique name for your device or select **Auto Generate ID** to generate a unique ID instead . In this demo please type **"myFirstDevice"** that will be used in FRDM-K64F platform . Then click **Create** .



d) A Device Created window appears , indicating that your device has been successfully registered with this IoT Hub .

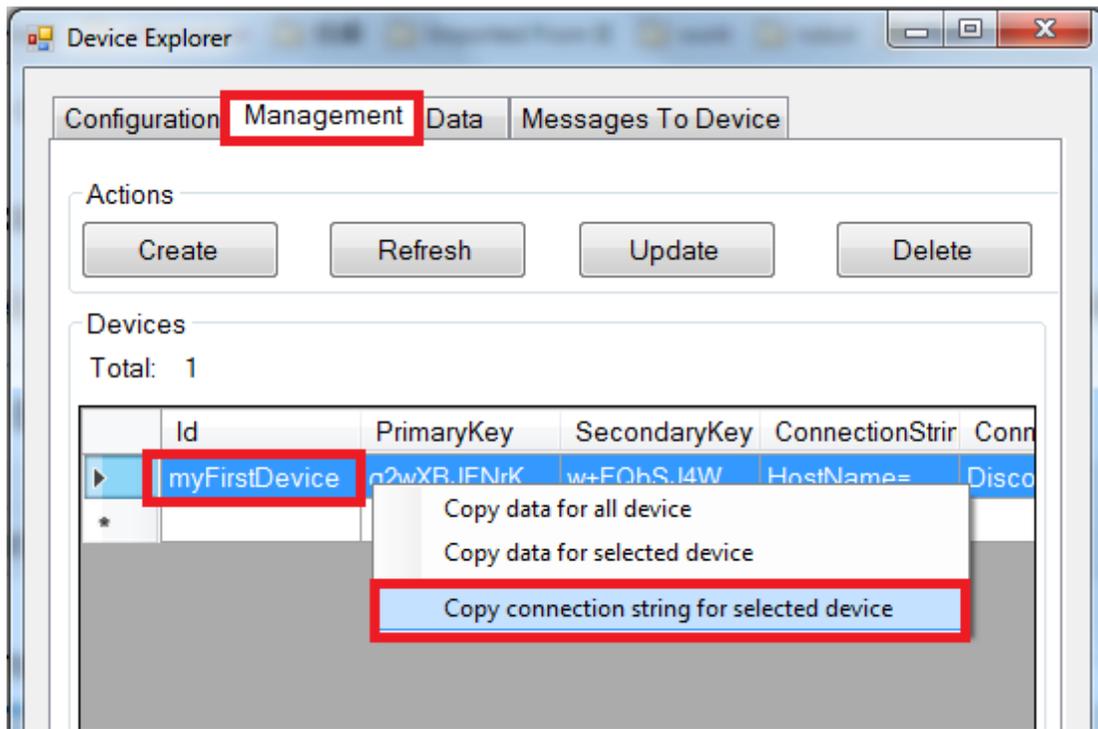


Step 4 : Get device connection string

Use the mouse right click for context menu for the selected device and select "Copy connection string for selected device" of options . Then please save the connection string that you copied .

Ex. Device connection string -->

HostName=Mars-Surface.azure-devices.net;DeviceId=myFirstDevice;SharedAccessKey=q2wXBJFNrKQF6t4pPyrPQF3H3UjInXv1e/Hasy38yU=



2.6 Connect the device

1. Connect the board FRDM-K64F to your network using an Ethernet cable. This step is required, as the sample depends on Internet access.
2. Plug the device into your computer using a micro-USB cable. Be sure to attach the cable to the correct USB port on the device (the CMSIS-DAP USB one, see [here](#) to find which one it is).
3. Follow the [instructions on the mbed handbook](#) to setup the serial connection with your device from your development machine. If you are on Windows, install the Windows serial port drivers located [here](#).

2.7 Program IoT Client -- FRDM-K64F

2.7.1 FRDM-K64F

The Freescale Freedom-K64F is an ultra-low-cost development platform for Kinetis K64, K63, and K24 MCUs.

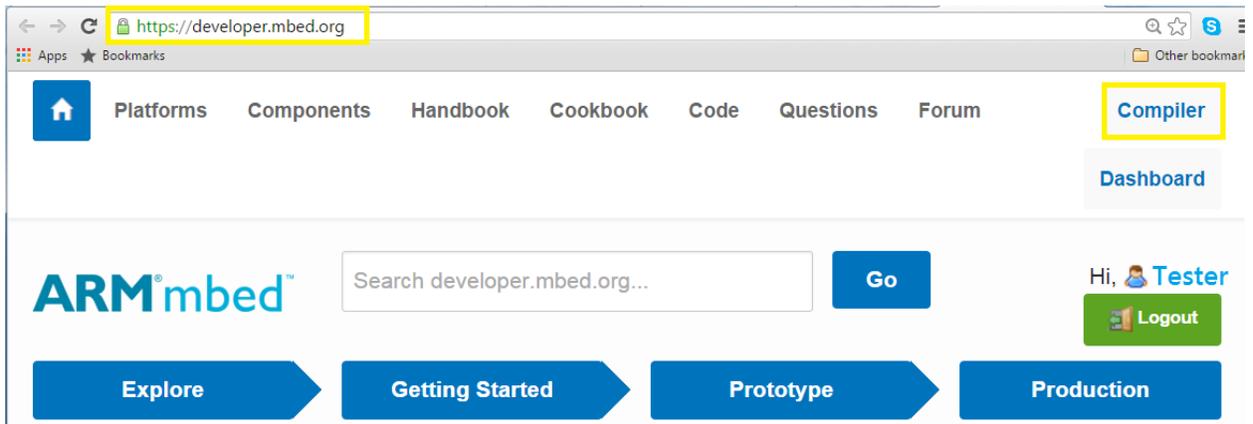
- Form-factor compatible with the Arduino R3 pin layout
- Peripherals enable rapid prototyping, including a 6-axis digital accelerometer and magnetometer to create full eCompass capabilities, a tri-colored LED and 2 user push-buttons for direct interaction, a microSD card slot, and connectivity using onboard Ethernet port and headers for use with Bluetooth® and 2.4 GHz radio add-on modules
- [Microsoft has certified the ARM® mbed Enabled™ Freescale® FRDM-K64F development board for Internet of Things \(IoT\) products](#)
 - *More detail please refer to below Microsoft link .*

<http://www.freescale.com/FRDM-K64F>

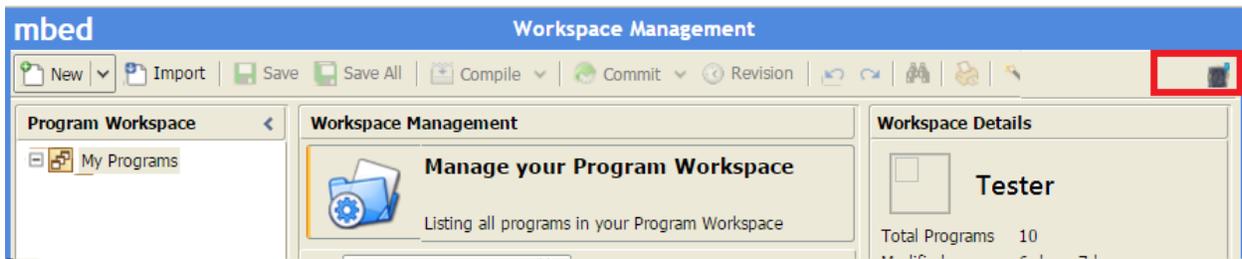
2.7.2 Create mbed project and import the sample code

Step 1 : In your web browser, go to the mbed.org [developer site](#).

If you haven't signed up, you will see an option to create a new account (it's free). Otherwise, log in with your account credentials. Then click on **Compiler** in the upper right-hand corner of the page. This should bring you to the Workspace Management interface.



Step2 : Make sure the hardware platform you're using appears in the upper right-hand corner of the window, or click the icon in the right-hand corner to select your hardware platform.



[Home](#)
[Platforms](#)
[Components](#)
[Handbook](#)
[Cookbook](#)
[Code](#)
[Questions](#)
[Forum](#)
[Dashboard](#)
[Compiler](#)

ARMmbed™

 Hi, [Logout](#)

Platforms

Filter

Target vendor

- ARM
- Atmel
- Freescale**
- Semiconductor
 - Maxim Integrated
 - NXP
- Semiconductors
 - Nordic
- Semiconductor ASA
 - Renesas
- STMicroelectronics
- Silicon Labs
- WIZnet

Platforms



FRDM-KL25Z

- Cortex-M0+
- 128KB Flash, 16KB RAM
- USB OTG



FRDM-KL46Z

- Cortex-M0+, 48MHz
- 256KB Flash, 32KB RAM
- USB OTG



FRDM-K64F

- Cortex-M4, 120MHz
- 1MB Flash, 256KB RAM
- Ethernet, SD Filesystem

[Home](#)
[Platforms](#)
[Components](#)
[Handbook](#)
[Cookbook](#)
[Code](#)
[Questions](#)
[Forum](#)
[Dashboard](#)
[Compiler](#)

ARMmbed™

 Hi, [Logout](#)

Platforms » FRDM-K64F

FRDM-K64F

Freescale Freedom Development Platform



Platform Partner



Freescale

Freescale is a leader in embedded processing solutions for the automotive, consumer, industrial and networking markets.

Overview

The Flagship FRDM-K64F has been designed by Freescale in collaboration with mbed for prototyping

[Table of Contents](#)

mbed Workspace Management

Program Workspace

- My Programs

Workspace Management

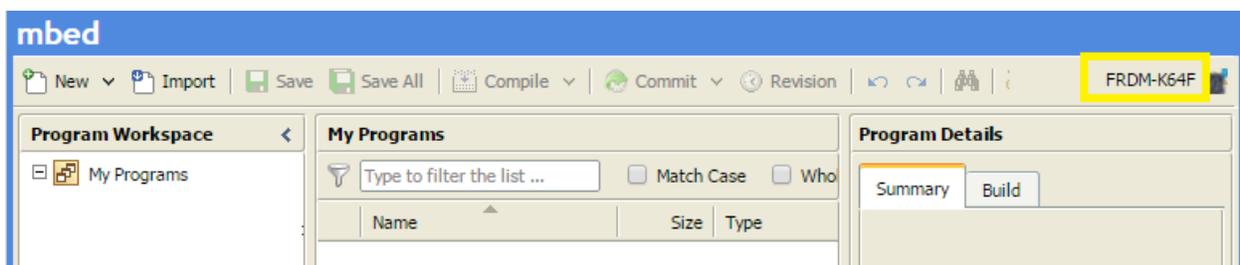
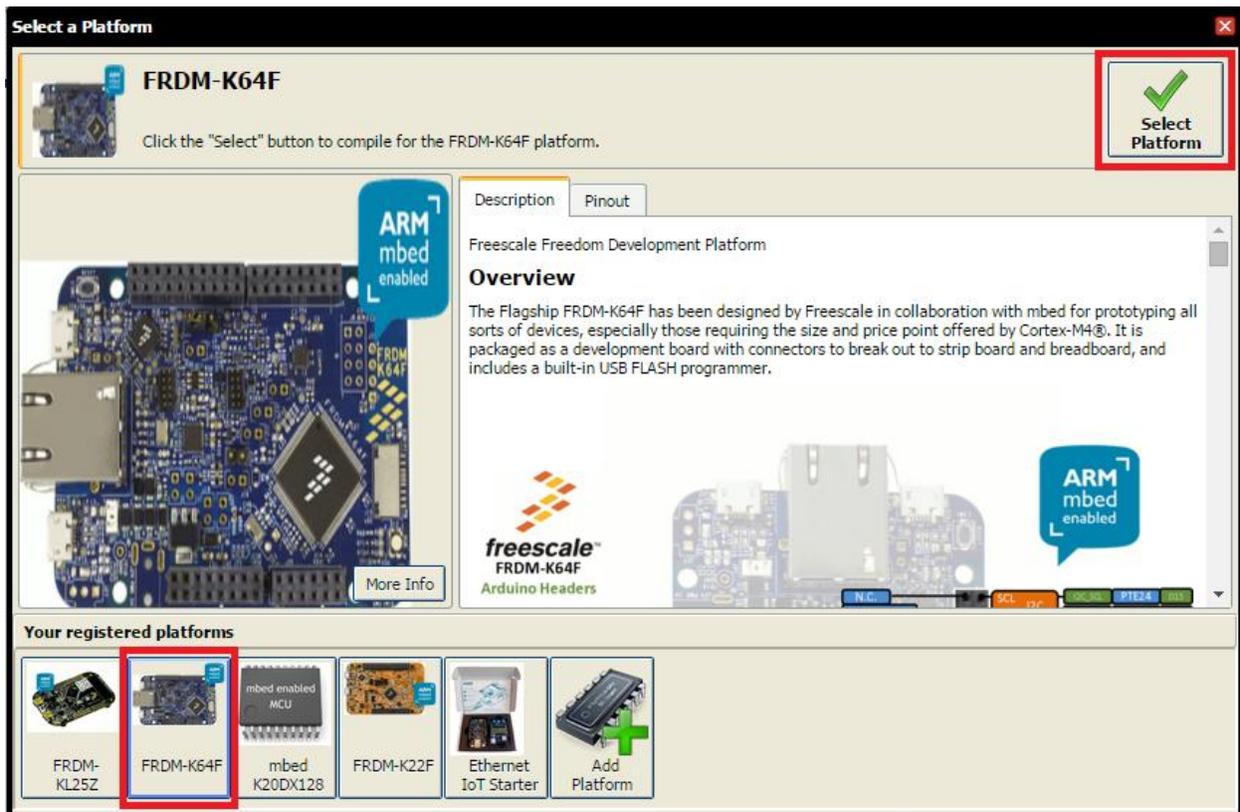
Manage your Program Workspace

Listing all programs in your Program Workspace

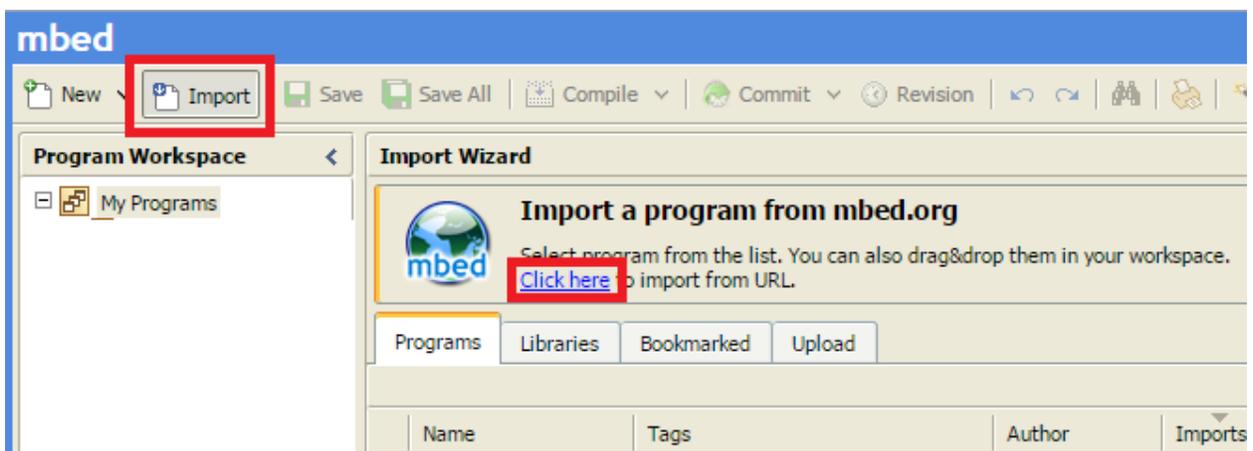
Workspace Details

Tester

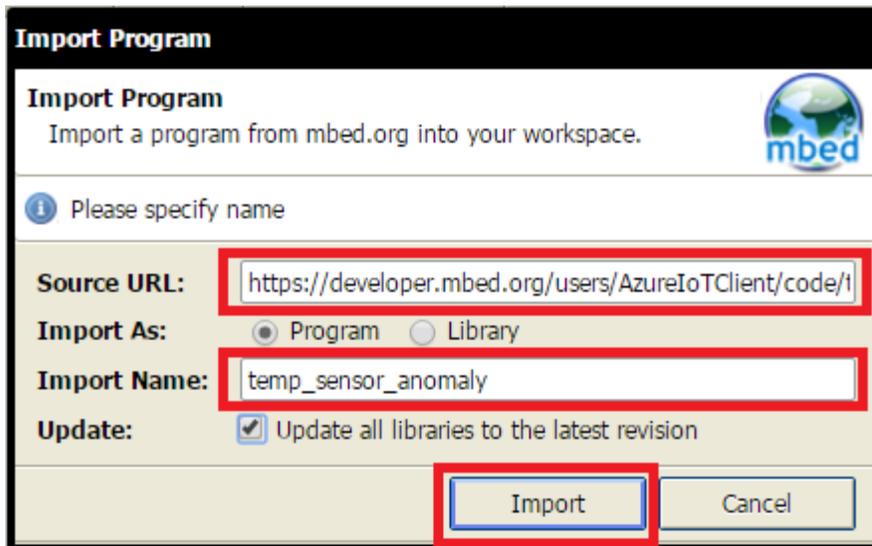
Total Programs 10



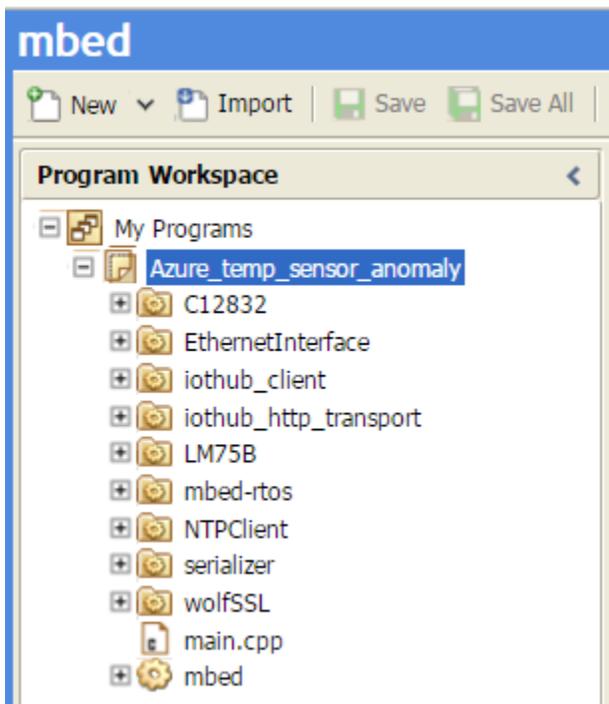
Step3 : Click **Import** on the main menu. Then click the **Click here** to import from URL link next to the mbed globe logo.



Step4 : In the popup window, enter the link for the sample code
https://developer.mbed.org/users/AzureIoTClient/code/temp_sensor_anomaly/



Step 5 : You can see in the mbed compiler that importing this project imported various libraries. Some are provided and maintained by the Azure IoT team ([azureiot_common](#), [iothub_client](#), [iothub_http_transport](#), [proton-c-mbed](#)), while others are third party libraries available in the mbed libraries catalog .



Step 6 : In the temp_sensor_anomaly\main.cpp file, find and replace values in the following lines of code with your

- a) Device connection string that is what we created at Chapter 2.5.1 Step 4 .
- b) device Id that is what we created at Chapter 2.5.1 Step 3 . (" myFirstDevice " is what we used on this demo)

The screenshot shows the mbed IDE interface. The 'Program Workspace' on the left lists 'My Programs' with 'Azure_temp_sensor_and' selected. The main editor displays the following code in main.cpp:

```

27 static const char* connectionString = "HostName=Mars-Surface.azure-dev;
28 static const char* deviceId = "myFirstDevice";//"[deviceName]"; /*must
30 static Timer led_timer;
31 static unsigned char alarm_type;
32 static unsigned char led_on;
33 static unsigned int last_alarm_time;
34 static unsigned int last_edge_time;
35 static unsigned int blink_interval;
36 static float temp;
37 static char* sensorId = NULL;
38
39 #define ALARM_NONE 0
40 #define ALARM_ANOMALY 1
41 #define ATARM THRESHOLD 2

```

Step 7: In case we don't have "mbed Application shield" then we need to modify the software to generate a random number to instead temperature sensor .

The screenshot shows the mbed IDE with the main.cpp file open. The code includes logic for calculating a new temperature value and setting an alarm:

```

127 //b36932 new_temp_value = (sensor.temp() * 9 / 5) + 32;
128 //b36932 temp = temp + (new_temp_value - temp) / 2;
130 temp = led_timer.read_ms() & 0x7F; //b36932
131 if(temp > 100)
132     temp = 50 ;

```

The screenshot shows the mbed IDE with the main.cpp file open. The code includes device initialization and a comment about sending the device into debug mode:

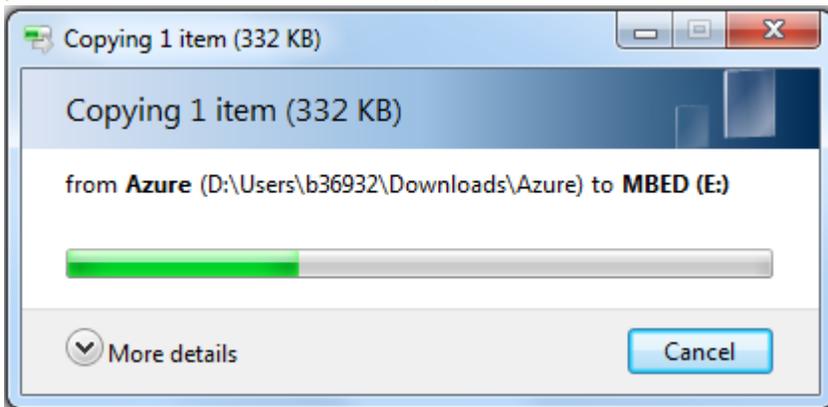
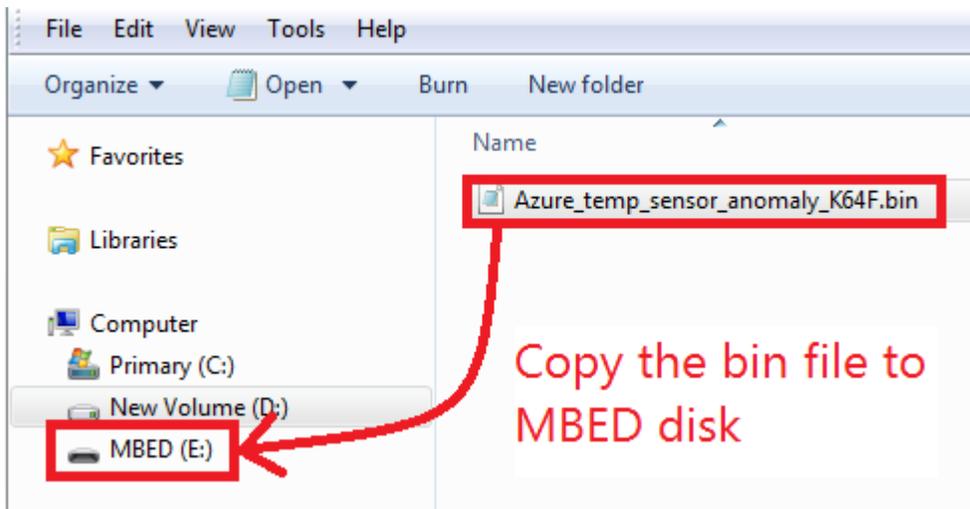
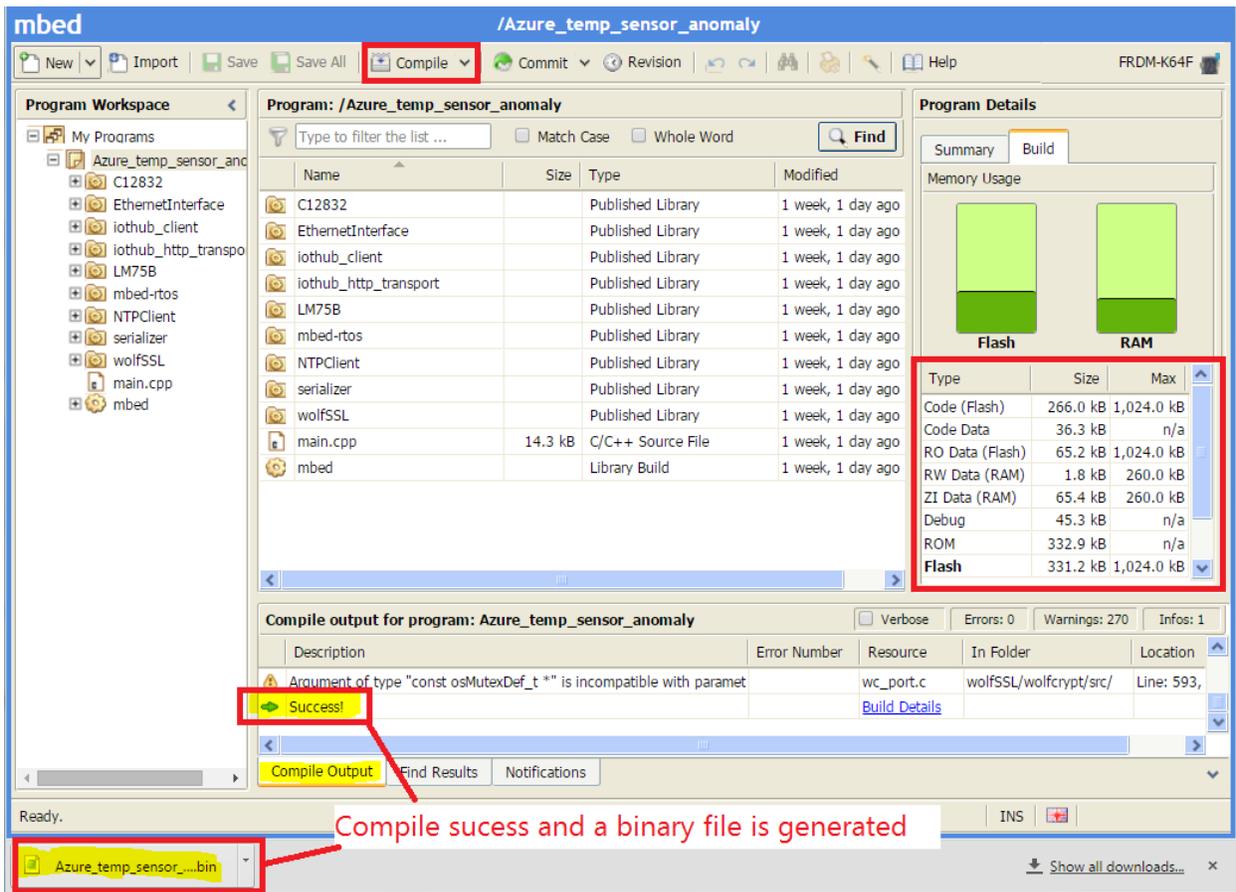
```

367 {
368     STRING_HANDLE commandsMetadata;
369
370     //b36932 temp = (sensor.temp() * 9 / 5) + 32;
371     //temp = 89 ; //b36932
372     temp = led_timer.read_ms() & 0x7F; //b36932
373     if(temp > 100)
374         temp = 50 ;
375
376     /* Send the device into debug mode so that the cloud app k
377     what commands are available and the fact that the device is
378     frdmDevice->ObjectType = "DeviceInfo-HW";
379     frdmDevice->ObjectName = "An ALARM device";
380     frdmDevice->Version = "1.0";
381     frdmDevice->SystemProperties.DeviceID = (char*) deviceId;
382     frdmDevice->SystemProperties.Enabled = true;

```

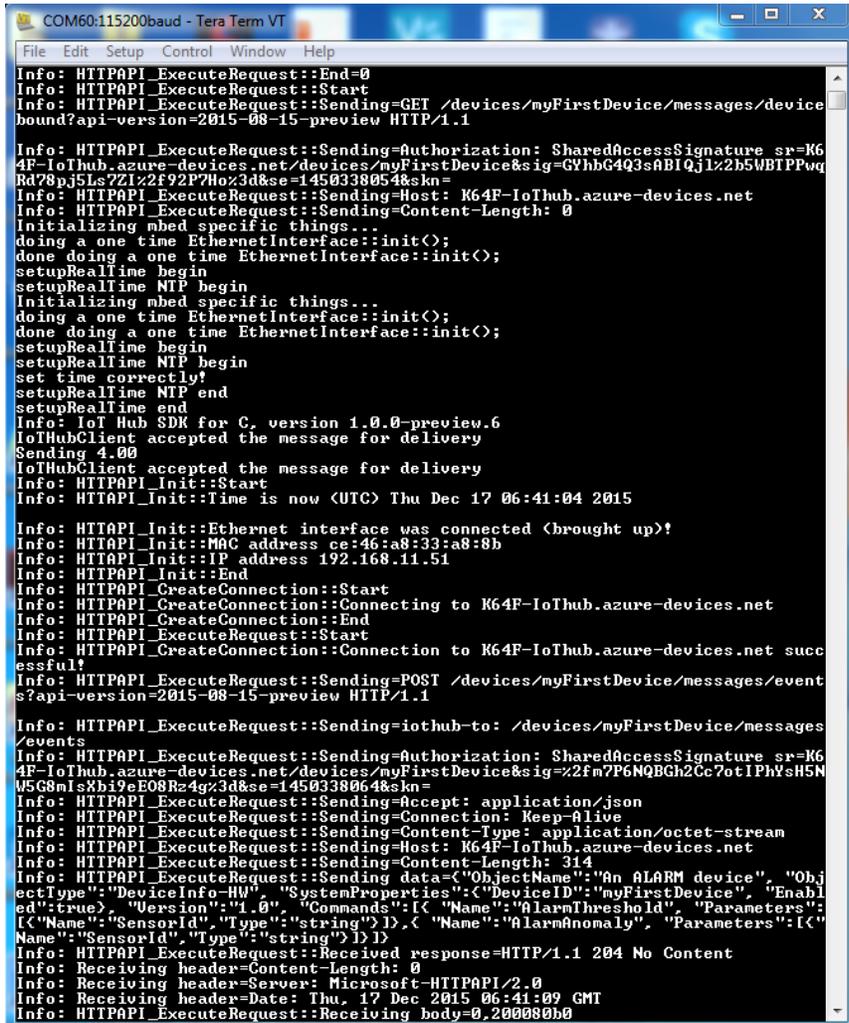
Step 8: Click **Compile** to build the program. If the build generates errors, fix them before proceeding.

Step 9: If the build is successful, a .bin file with the name of your project is generated. Copy the .bin file to the device.



Step 10: After firmware update , unplug and plug the FRDM-K64F board USB again . Then connect to the device using an serial terminal client application, such as PuTTY / Tera Term. Baud rate is 115200 .

The program starts executing. You may have to reset the board (press on the board's reset button) .

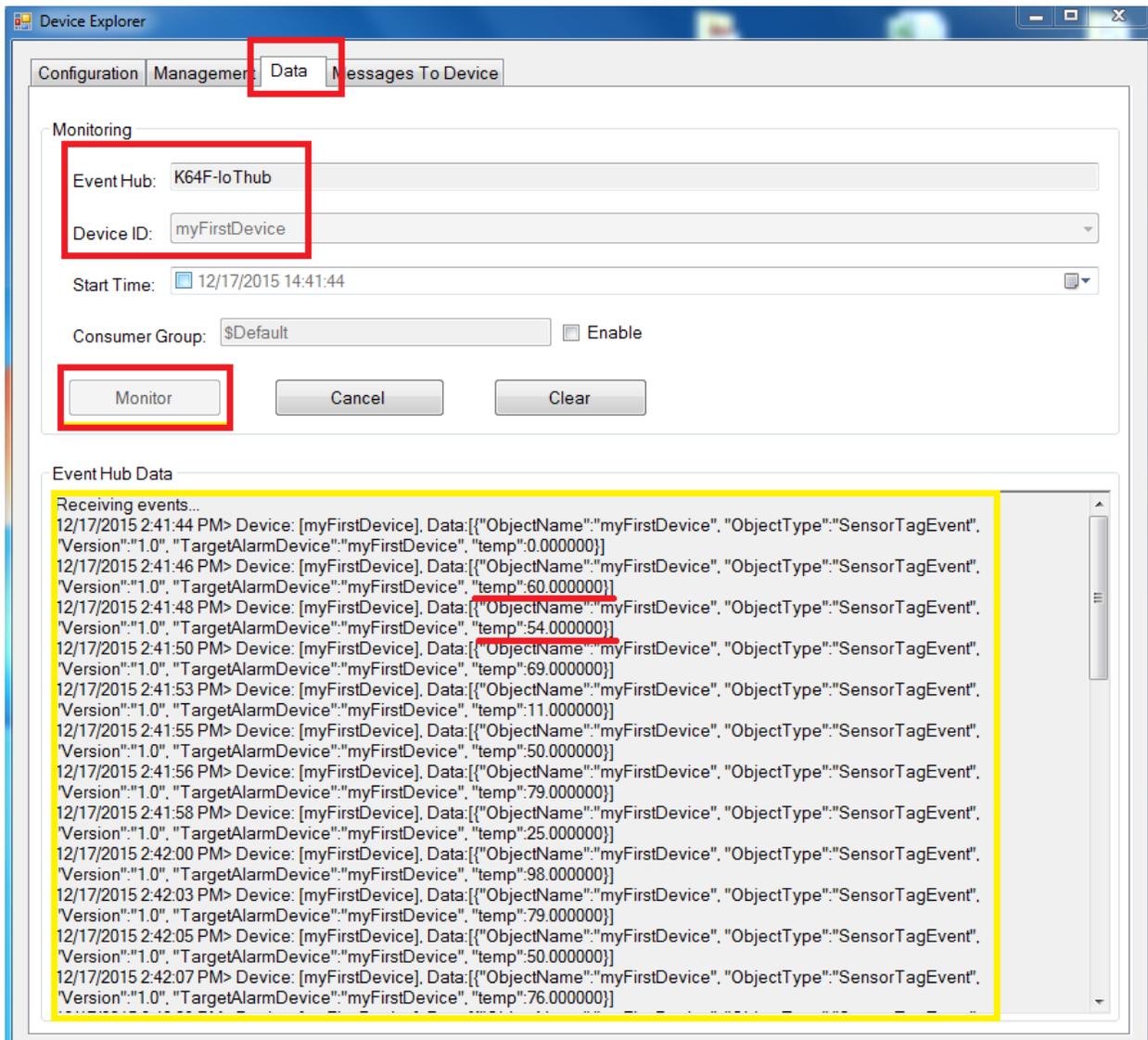


```
COM60:115200baud - Tera Term VT
File Edit Setup Control Window Help
Info: HTTPAPI_ExecuteRequest::End=0
Info: HTTPAPI_ExecuteRequest::Start
Info: HTTPAPI_ExecuteRequest::Sending=GET /devices/myFirstDevice/messages/devicebound?api-version=2015-08-15-preview HTTP/1.1
Info: HTTPAPI_ExecuteRequest::Sending=Authorization: SharedAccessSignature sr=K64F-IoThub.azure-devices.net/devices/myFirstDevice&sig=GYhbG4Q3sABlQj1z2b5WBTppwqRd78pJ5Ls721z2f92P7HoZ3d&se=1450338054&skn=
Info: HTTPAPI_ExecuteRequest::Sending=Host: K64F-IoThub.azure-devices.net
Info: HTTPAPI_ExecuteRequest::Sending=Content-Length: 0
Initializing mbed specific things...
doing a one time EthernetInterface::init();
done doing a one time EthernetInterface::init();
setupRealtime begin
setupRealtime NTP begin
Initializing mbed specific things...
doing a one time EthernetInterface::init();
done doing a one time EthernetInterface::init();
setupRealtime begin
setupRealtime NTP begin
set time correctly!
setupRealtime NTP end
setupRealtime end
Info: IoT Hub SDK for C, version 1.0.0-preview.6
IoTHubClient accepted the message for delivery
Sending 4.00
IoTHubClient accepted the message for delivery
Info: HTTPAPI_Init::Start
Info: HTTPAPI_Init::Time is now (UTC) Thu Dec 17 06:41:04 2015
Info: HTTPAPI_Init::Ethernet interface was connected (brought up)!
Info: HTTPAPI_Init::MAC address ce:46:a8:33:a8:8b
Info: HTTPAPI_Init::IP address 192.168.11.51
Info: HTTPAPI_Init::End
Info: HTTPAPI_CreateConnection::Start
Info: HTTPAPI_CreateConnection::Connecting to K64F-IoThub.azure-devices.net
Info: HTTPAPI_CreateConnection::End
Info: HTTPAPI_ExecuteRequest::Start
Info: HTTPAPI_CreateConnection::Connection to K64F-IoThub.azure-devices.net successful!
Info: HTTPAPI_ExecuteRequest::Sending=POST /devices/myFirstDevice/messages/events?api-version=2015-08-15-preview HTTP/1.1
Info: HTTPAPI_ExecuteRequest::Sending=iOTHUB-TO: /devices/myFirstDevice/messages/events
Info: HTTPAPI_ExecuteRequest::Sending=Authorization: SharedAccessSignature sr=K64F-IoThub.azure-devices.net/devices/myFirstDevice&sig=%2Fm7P6NQBgh2Cc7otIPhYsH5NWSG8m1sXbi9eE08Rz4g%3d&se=1450338064&skn=
Info: HTTPAPI_ExecuteRequest::Sending=Accept: application/json
Info: HTTPAPI_ExecuteRequest::Sending=Connection: Keep-Alive
Info: HTTPAPI_ExecuteRequest::Sending=Content-Type: application/octet-stream
Info: HTTPAPI_ExecuteRequest::Sending=Host: K64F-IoThub.azure-devices.net
Info: HTTPAPI_ExecuteRequest::Sending=Content-Length: 314
Info: HTTPAPI_ExecuteRequest::Sending data=<"ObjectName": "An ALARM device", "ObjectType": "DeviceInfo-HW", "SystemProperties": {"DeviceID": "myFirstDevice", "Enabled": true}, "Version": "1.0", "Commands": [{"Name": "AlarmThreshold", "Parameters": [{"Name": "SensorId", "Type": "string"}]}, {"Name": "AlarmAnomaly", "Parameters": [{"Name": "SensorId", "Type": "string"}]}]>
Info: HTTPAPI_ExecuteRequest::Received response=HTTP/1.1 204 No Content
Info: Receiving header=Content-Length: 0
Info: Receiving header=Server: Microsoft-HTTPAPI/2.0
Info: Receiving header=Date: Thu, 17 Dec 2015 06:41:09 GMT
Info: HTTPAPI_ExecuteRequest::Receiving body=0,200080b0
```

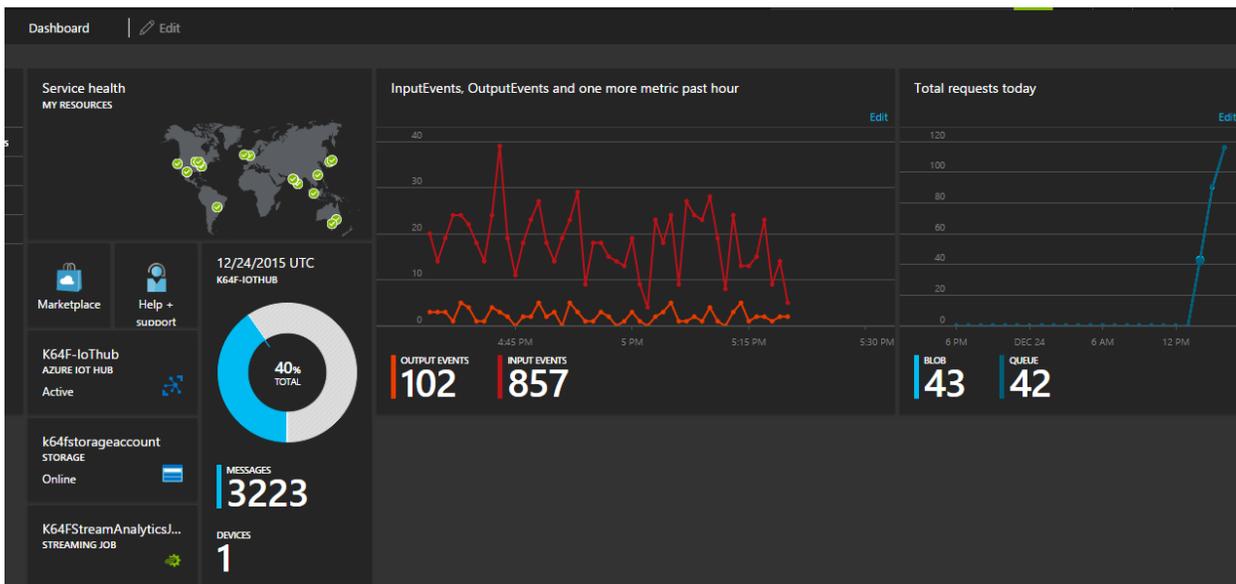
2.8 Monitor IoT Client information

2.8.1 Receive IoT Client data by “Device Explorer”

- In the Device Explorer **Data** tab, select the device name you created from the drop-down list of device IDs and leave the other fields with their default values for now.
- Click **Monitor**.



2.8.2 The data on Azure



[ALERT ACTIVATED] - Output Events GreaterThan 3 (Count) in the last 30 minutes...

FILE MESSAGE

Thu 12/24/2015 3:51 PM

Microsoft Azure Alerts <alerts-noreply@mail.windowsazure.com>
[ALERT ACTIVATED] - Output Events GreaterThan 3 (Count) in the last 30 minutes

To David Chen

Bing Maps Unsubscribe + Get more apps

Microsoft
Azure

Dear Customer,

⚠ 'Output Events GreaterThan 3 (Count) in the last 30 minutes' was activated for Streaming Job: K64FStreamAnalyticsJob (k64f-resourcegroup)

Microsoft Azure Alerts [ALERT ACTIVATED] - Output Events GreaterThan 1 (Count) in the last 60 minutes

Dear Customer,

 'Output Events GreaterThan 3 (Count) in the last 30 minutes' was activated for Streaming Job: K64FStreamAnalyticsJob (k64f-resourcegroup)

You can view more details for this alert in the [Microsoft Azure Management Portal](#).

RULE NAME: StreamAnalyticsAlert_OverTemp

RULE DESCRIPTION:

SERVICE: Streaming Job: K64FStreamAnalyticsJob (k64f-resourcegroup)

METRIC: Total Output Events

ALERT ACTIVATED TIME (UTC): 12/24/2015 7:51:22 AM

SUBSCRIPTION NAME: Free Trial