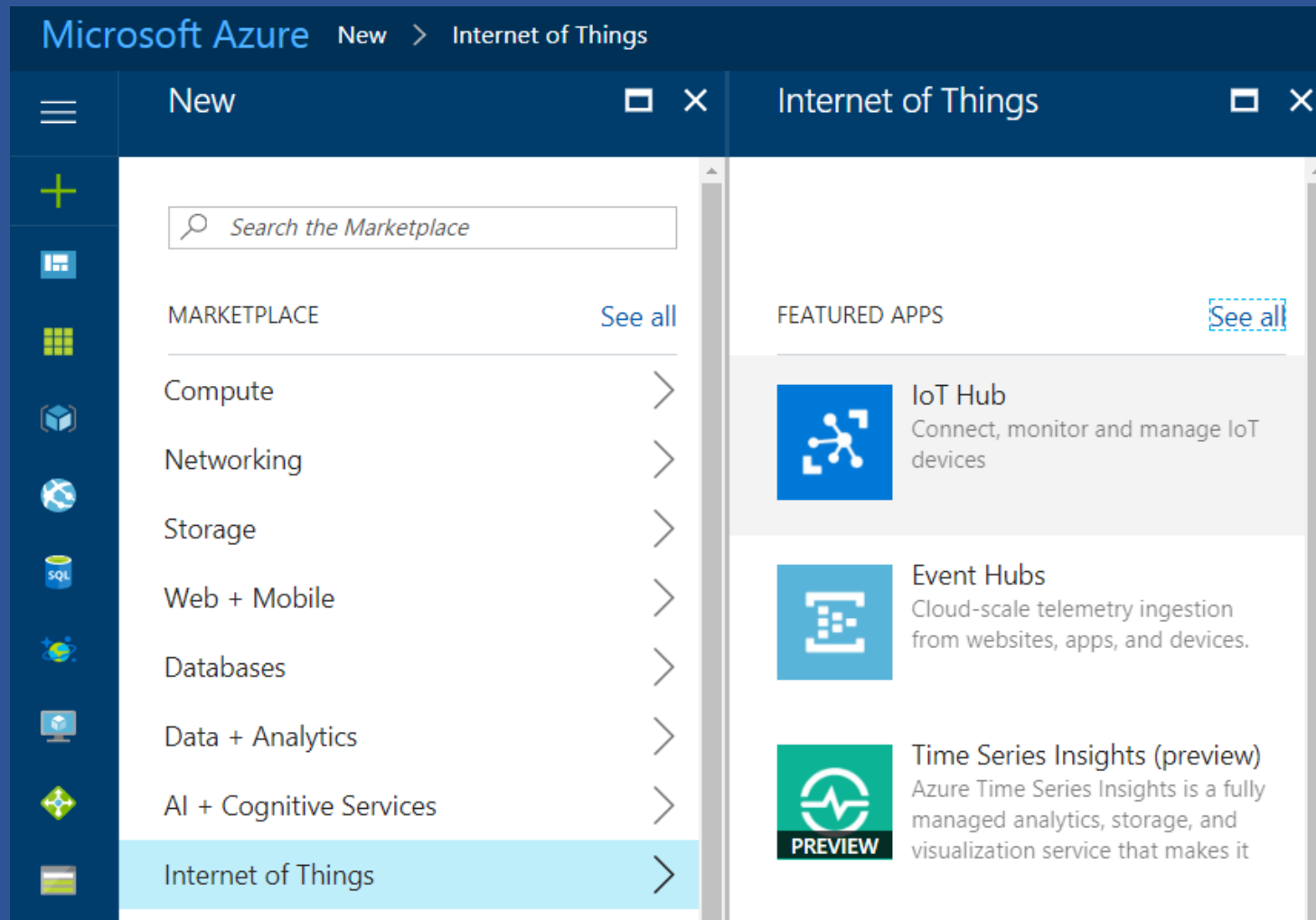


# Create *Azure* function that read message from device

## Do to list

- Create Azure IoT Hub
- Provision a device
- Create device sim in Visual studio
- Make sure the device send message to cloud successfully. Use Win app “device explorer” to monitor message
- Create Azure function app
- Create Azure function
- Make sure function is triggered by IoT Hub event
- See function’s log for the message from device.

## Create Azure IoT Hub : New / Internet of Things

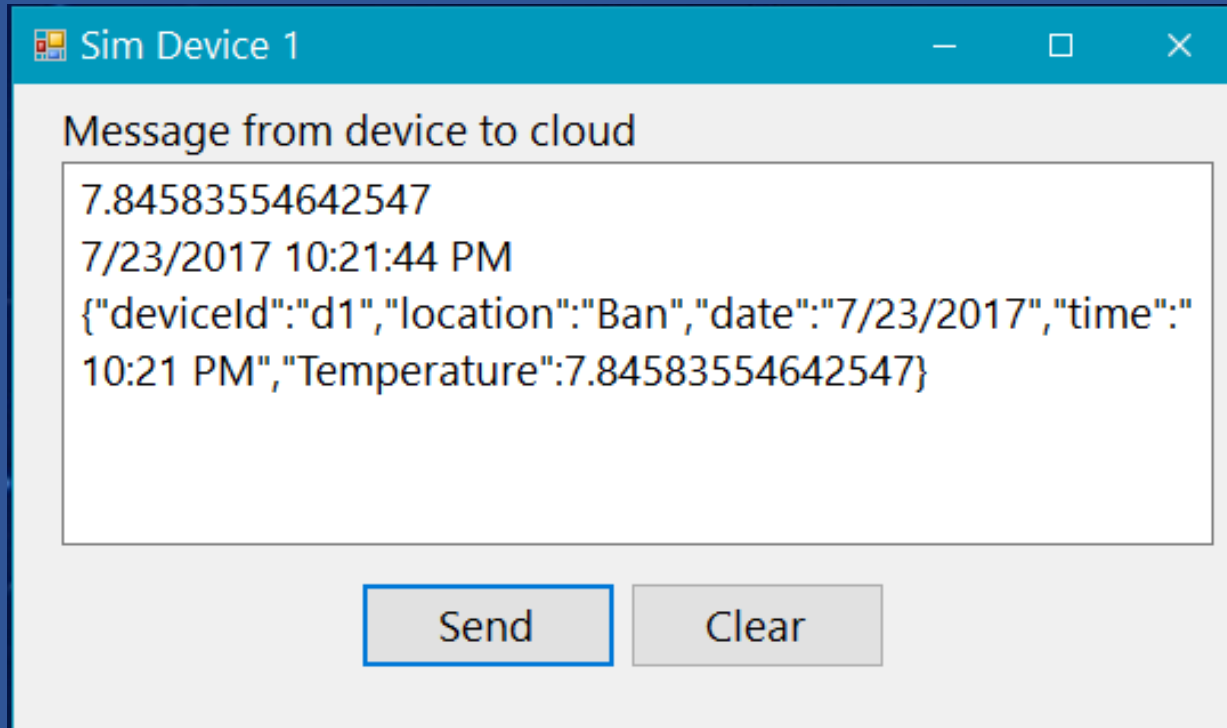


## Provision device

The screenshot shows the 'Device Explorer Twin' web application. It has a teal header with the title and a navigation bar with tabs: 'Configuration', 'Management', 'Data', 'Messages To Device', and 'Call M'. The 'Management' tab is active. Below the tabs, there is an 'Actions' section with buttons for 'Create', 'Refresh', 'Update', and a partially visible 'Delete' button. Below the actions is a 'Devices' section showing 'Total: 2'. A table lists the devices with columns: Id, PrimaryKey, SecondaryKey, and Primary. The first device, 'd1', is highlighted in blue. The second device, 'd2', is listed below it. A third row with a gear icon is partially visible at the bottom.

	Id	PrimaryKey	SecondaryKey	Primary
▶	d1	K2JW/PI2qPx...	6jM6bx5qMQL...	
	d2	Un4NPE/g8L...	RQm+G2tP0p...	
⚙				

## Create device sim



The screenshot shows a web application window titled "Sim Device 1". Inside the window, there is a section labeled "Message from device to cloud". Below this label is a text area containing the following text:

```
7.84583554642547  
7/23/2017 10:21:44 PM  
{ "deviceId": "d1", "location": "Ban", "date": "7/23/2017", "time": "10:21 PM", "Temperature": 7.84583554642547 }
```

At the bottom of the window, there are two buttons: "Send" and "Clear". The "Send" button is highlighted with a blue border.

Make sure you can send message to cloud. Use Win app “device explorer” to monitor message.

The screenshot shows the 'Device Explorer Twin' application window. It has a teal title bar and five tabs: 'Configuration', 'Management', 'Data' (which is selected), 'Messages To Device', and 'Call Method on Device'. The 'Data' tab contains a 'Monitoring' section with the following fields: 'Event Hub' (containing 'loy2017bh1'), 'Device ID' (containing 'd1'), 'Start Time' (with a checked checkbox and the value '07/23/2017 22:27:42'), and 'Consumer Group' (containing '\$Default' and an unchecked 'Enable' checkbox). Below these fields are three buttons: 'Monitor', 'Cancel', and 'Clear'. At the bottom of the window is an 'Event Hub Data' section with a text area showing received events. The first event is: '7/23/2017 10:28:28 PM> Device: [d1], Data:[{"deviceId":"d1","location":"Ban","date":"7/23/2017 10:28:28 PM","Temperature":8.70771254864881}]'. The second event is: '7/23/2017 10:28:37 PM> Device: [d1], Data:[{"deviceId":"d1","location":"Ban","date":"7/23/2017 10:28:37 PM","Temperature":8.70771254864881}]'.

Device Explorer Twin

Configuration Management **Data** Messages To Device Call Method on Device

Monitoring

Event Hub:

Device ID:

Start Time: ☒ 07/23/2017 22:27:42

Consumer Group:  ☐ Enable

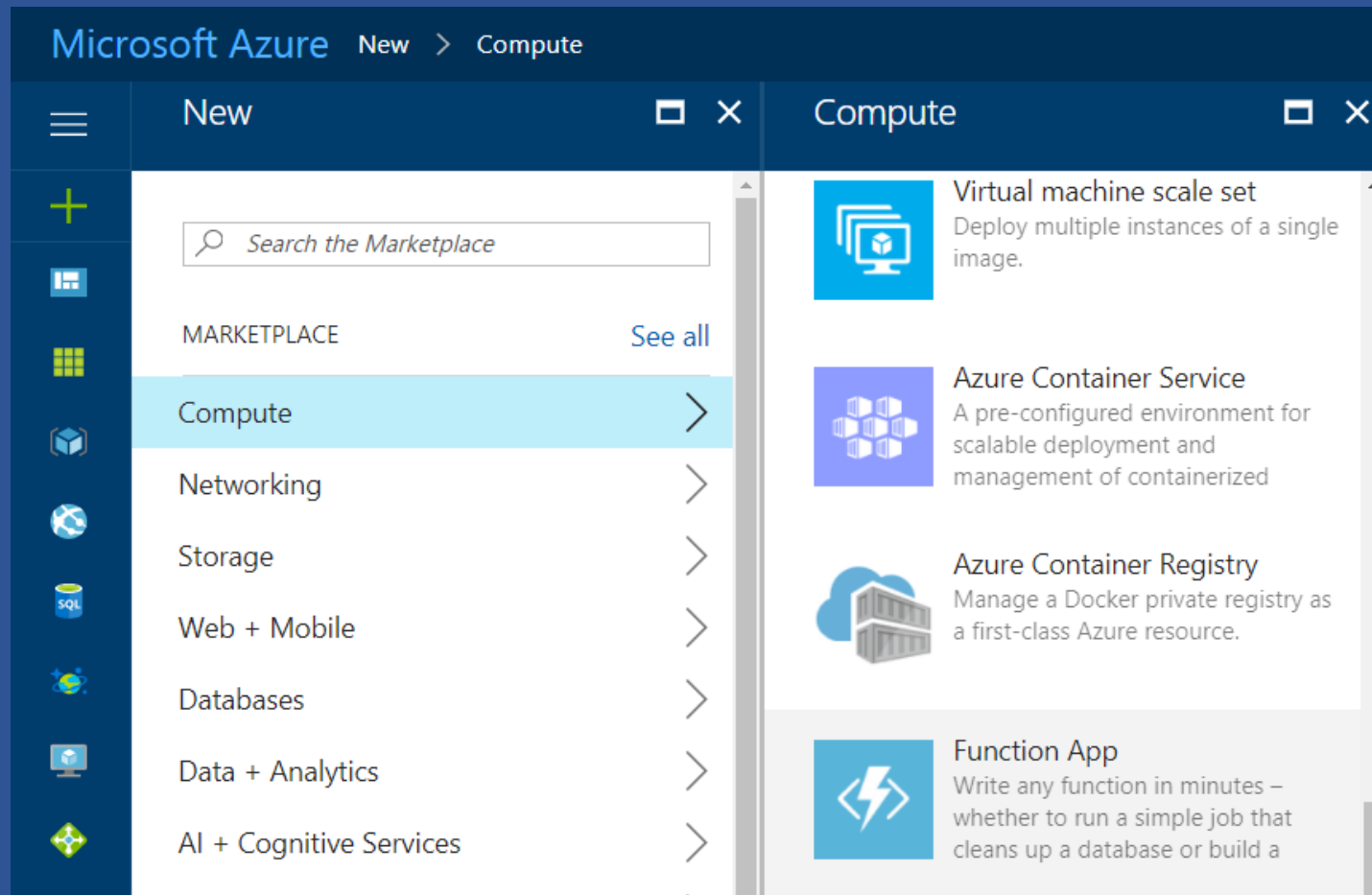
Event Hub Data

Receiving events...

7/23/2017 10:28:28 PM> Device: [d1], Data:[{"deviceId":"d1","location":"Ban","date":"7/23/2017 10:28:28 PM","Temperature":8.70771254864881}]

7/23/2017 10:28:37 PM> Device: [d1], Data:[{"deviceId":"d1","location":"Ban","date":"7/23/2017 10:28:37 PM","Temperature":8.70771254864881}]

## Create new Function App



## Function app properties

Function App

Create

\* App name

loy2017bfa2

.azurewebsites.net

\* Subscription

Loy2017b

\* Resource Group ⓘ

☐ Create new ☒ Use existing

loy2017brs1

\* Hosting Plan ⓘ

Consumption Plan

\* Location

South Central US

\* Storage ⓘ

☐ Create New ☒ Select Existing

loy2017bs1

Application Insights ⓘ



## Create new function “EventHubTrigger – C#”

The screenshot shows the Microsoft Azure portal interface for creating a new function. The top navigation bar indicates the current location: **Microsoft Azure** > **All resources** > **Function Apps - Functions**.

The left sidebar contains a navigation menu with the following items:

- Function Apps
- loy2017bfa2 (selected)
- Functions (highlighted with a blue border and a plus icon)
- Proxies (preview)
- Slots (preview)

The main content area displays the search results for "loy2017bfa2". It shows a search bar with the text "loy2017bfa2" and a dropdown menu with "Loy2017b". Below the search bar, there are two filters: **Language:** All and **Scenario:** Core.

The search results show a list of function templates. The first two templates are:

- whenever a message is added to a specified Azure Queue Storage
- whenever a message is added to a specified Azure Queue Storage

The third template is highlighted with a blue border:

- EventHubTrigger - C#**  
A C# function that will be run whenever an event hub receives a new event

The fourth template is:

- EventHubTrigger - F#**  
An F# function that will be run whenever an event hub receives a new event

Name = f1 / Click New

### Name your function

f1

### Azure Event Hub trigger

Event Hub name ⓘ

ehn1

Event Hub connection ⓘ

show value

new

Click IoT hub / Select

**Connection** ✕

Event Hub

**IoT hub**

Custom

**IoT hub**

loy2017bh1 ▼

**Endpoint**

Events (built-in endpoint) ▼

Select

## Click Create

Name your function

Azure Event Hub trigger

Event Hub name ⓘ

Event Hub connection ⓘ  show value *new*

Create

Enter code : Now we have the function / Save / Run

Microsoft Azure All resources > Function Apps - Functions

Function Apps - Functions

Function Apps

Loy2017b

Function Apps

loy2017bfa2

Functions

f1

Integrate

Manage

Monitor

Proxies (preview)

Slots (preview)

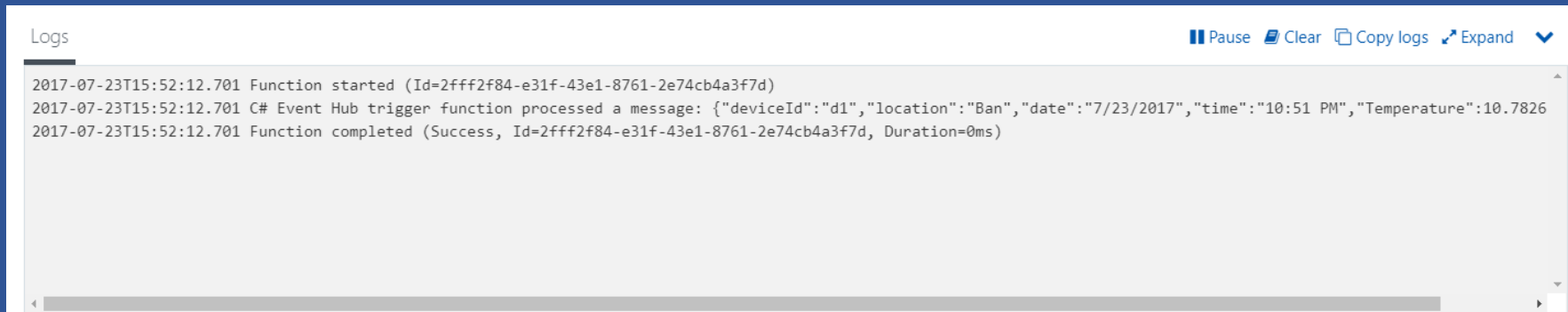
run.csx

Save

Run

```
1 using System;
2
3 public static void Run(string myEventHubMessage, TraceWriter log)
4 {
5     log.Info($"C# Event Hub trigger function processed a message: {myEventHubMessage}");
6 }
7
```

Send message from Device and look at the Logs of Function, you should see the message



The screenshot displays the 'Logs' tab of an Azure Function. At the top right, there are controls: a pause icon, a 'Clear' button, a 'Copy logs' button, an 'Expand' button, and a dropdown arrow. The log content shows three entries for the function 'Function started', 'C# Event Hub trigger function processed a message', and 'Function completed'. The message body is a JSON object with fields: deviceId, location, date, time, and Temperature. A scrollbar is visible at the bottom of the log area.

```
Logs
2017-07-23T15:52:12.701 Function started (Id=2fff2f84-e31f-43e1-8761-2e74cb4a3f7d)
2017-07-23T15:52:12.701 C# Event Hub trigger function processed a message: {"deviceId":"d1","location":"Ban","date":"7/23/2017","time":"10:51 PM","Temperature":10.7826}
2017-07-23T15:52:12.701 Function completed (Success, Id=2fff2f84-e31f-43e1-8761-2e74cb4a3f7d, Duration=0ms)
```