Lab 2 IoT Hub + Huzzah

By: Kevin Saye

IoT Solution Architect

December, 2017

This lab assumes you have completed Lab 1.

If you have any issues or concerns, please email: virtualbootcamphelp@microsoft.com.

Execution Time: 30 minutes.

Required Hardware:

- Windows 10 PC
- IoT Hardware kit: https://www.adafruit.com/product/3605 or similar hardware.
- Access to a WiFi network (without a captive portal aka web page login)

Required Operating System:

- Windows 10

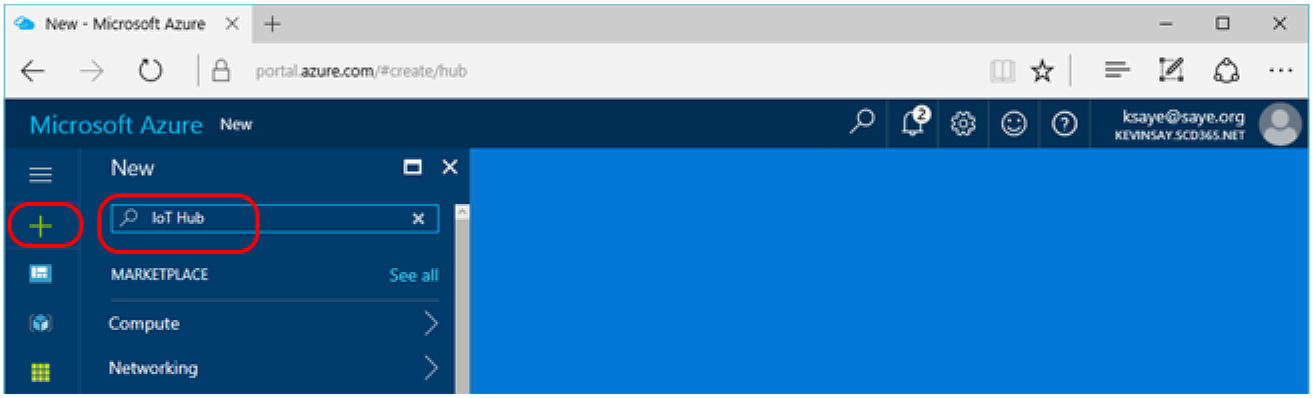
Other Requirements:

- Azure Subscription

Required Software:

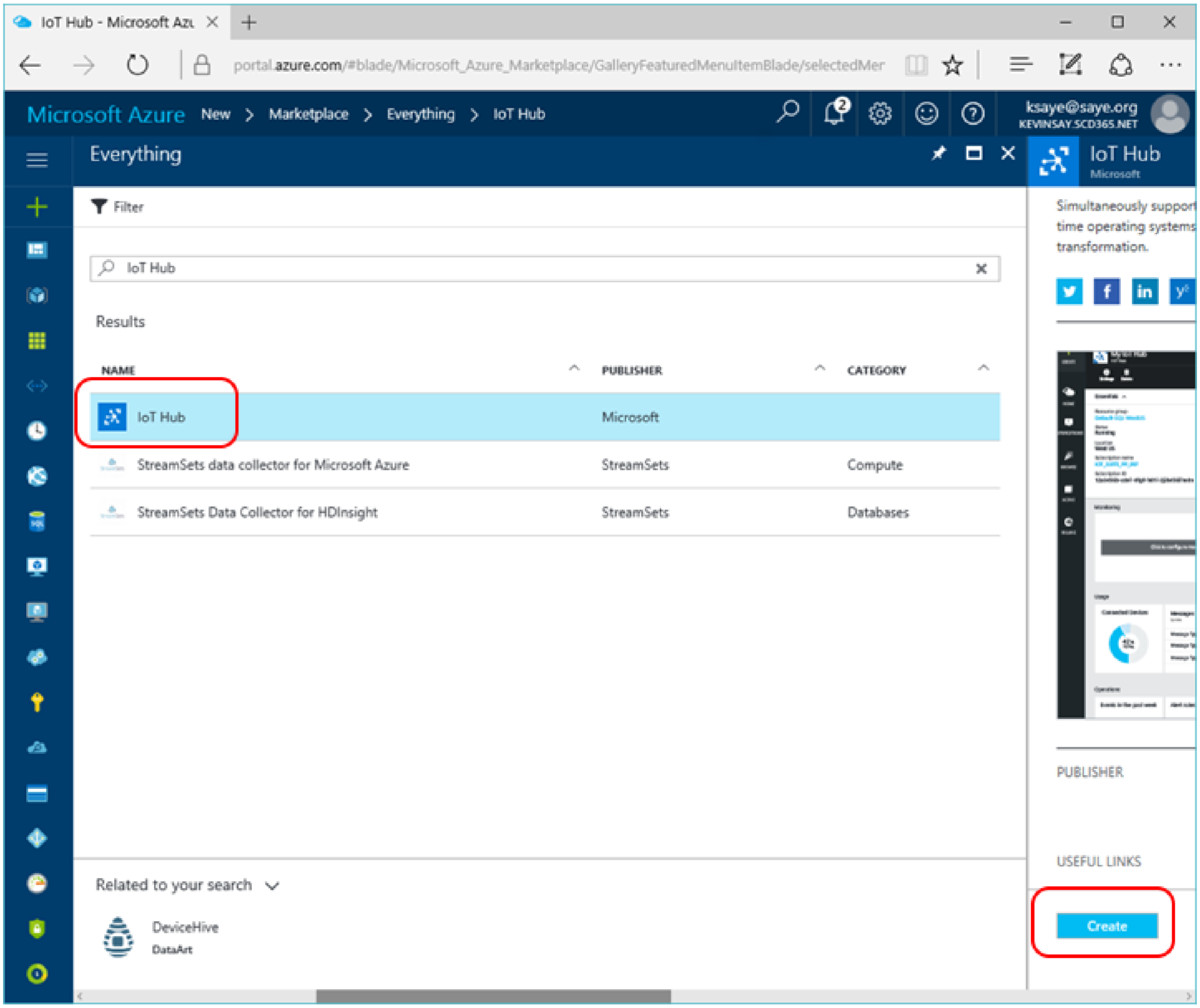
Software	Size	Installation URL
IoT Device Explorer	5 MB	https://github.com/Azure/azure-iot-sdk-csharp/releases/download/2017-10-23/SetupDeviceExplorer.msi

Step 1. Go to http://portal.azure.com, click the Plus sign on the left and search for IoT Hub



IoT Hub

Step 2. Select the IoT Hub by Microsoft and click Create.



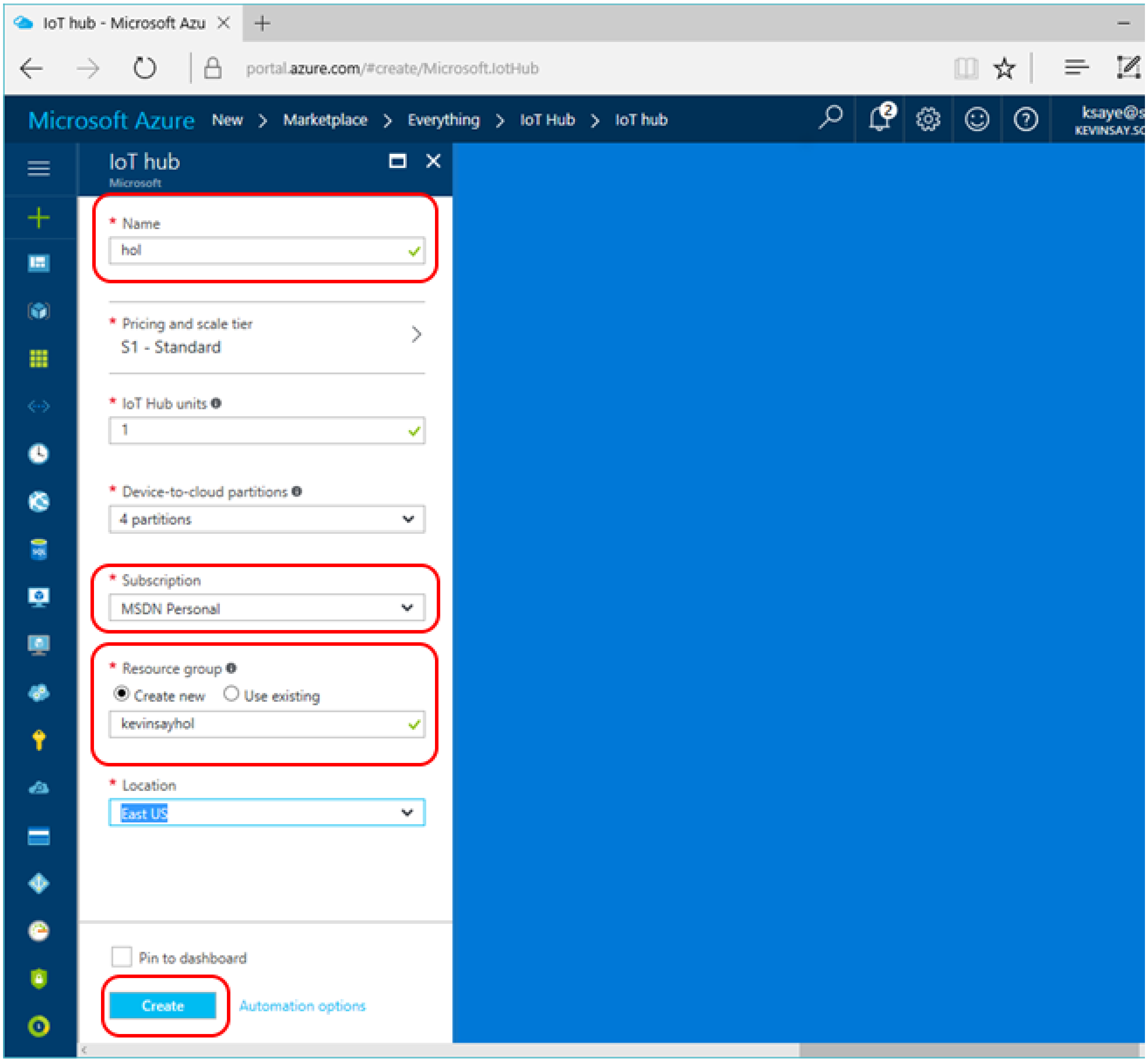
IoT Hub

Results

NAME	PUBLISHER	CATEGORY
IoT Hub	Microsoft	
StreamSets data collector for Microsoft Azure	StreamSets	Compute
StreamSets Data Collector for HDInsight	StreamSets	Databases

Simultaneously support time operating systems transformation.

Step 3. Give your hub a unique name, select the correct Azure Subscription, create a new (or use an existing Resource Group) and click create. Change the Pricing and Scale Tier to F1, if available to save cost.



IoT hub - Microsoft Azu
portal.azure.com/#create/Microsoft.IotHub
Microsoft Azure
New > Marketplace > Everything > IoT Hub > IoT hub
ksaye@s
KEVINSAY.S
IoT hub
Microsoft

* Name
hol

* Pricing and scale tier
S1 - Standard

* IoT Hub units
1

* Device-to-cloud partitions
4 partitions

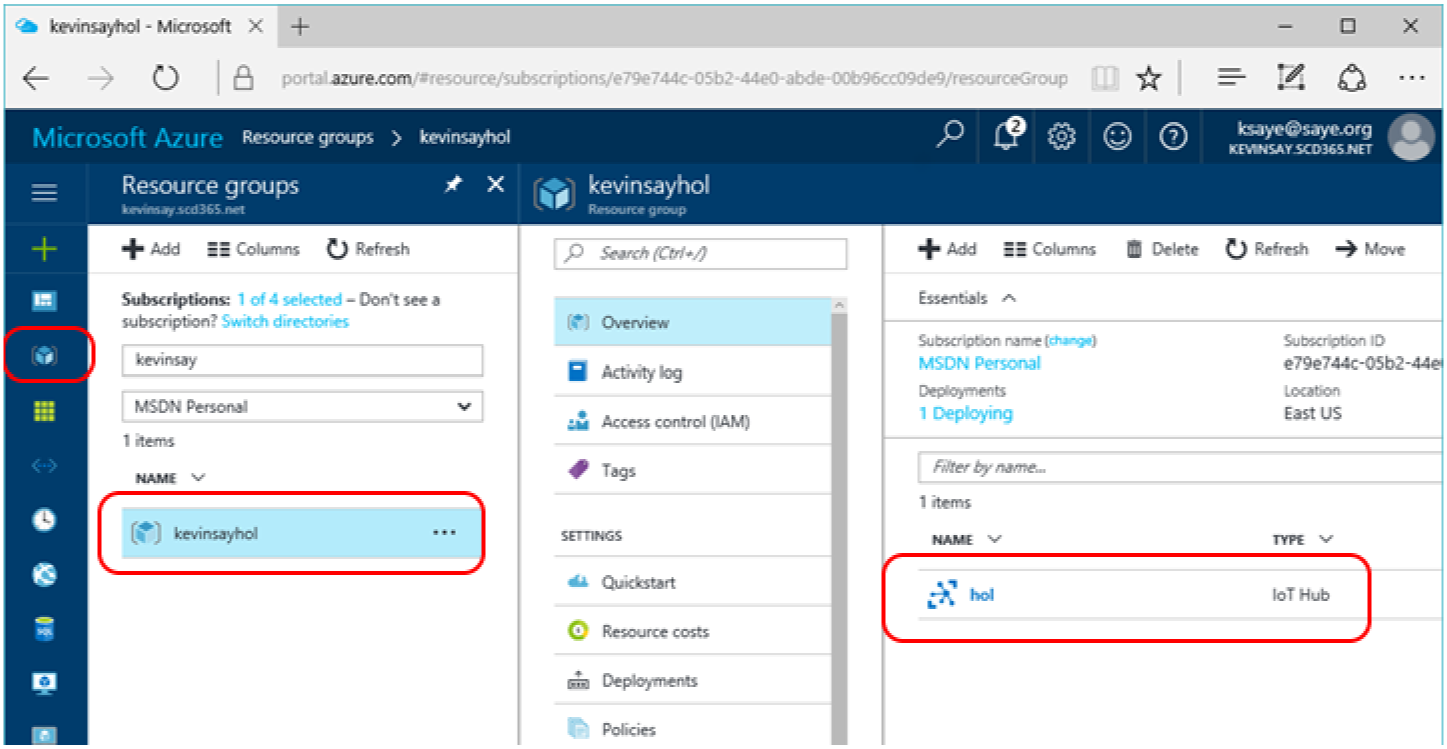
* Subscription
MSDN Personal

* Resource group
● Create new ○ Use existing
kevinsayhol

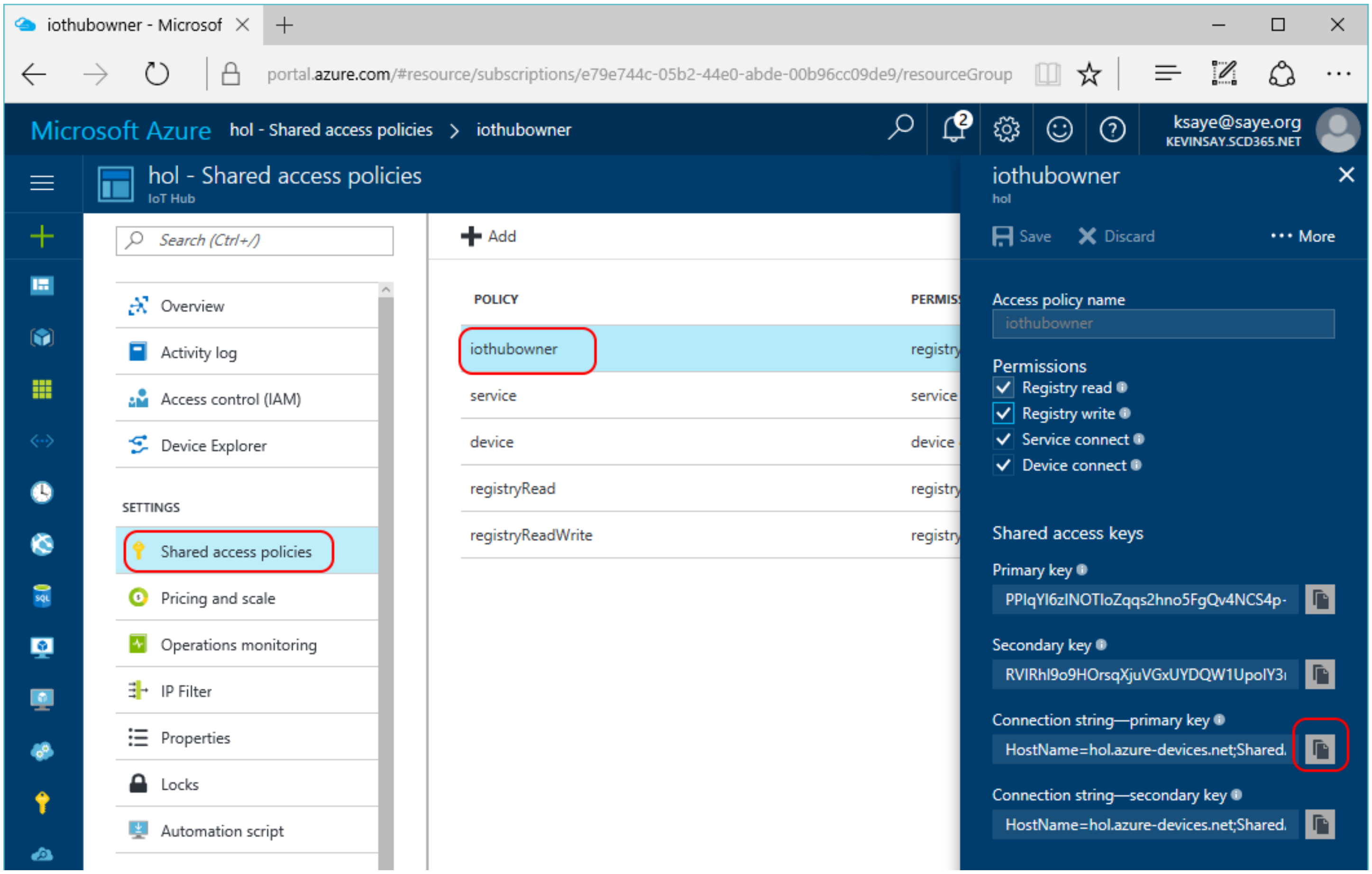
* Location
East US

□ Pin to dashboard

Step 4. Click the Resource Group Icon on the left, select the resource group you just created and click on your IoT Hub just created.



Step 5. Once the Hub is crated (about 2 minutes), click on the Shared Access Policies and then the iothubowner and then the copy icon to copy the connection string to the clipboard and save in a text file. This will be used in the next Lab.



POLICY	PERMIS
iothubowner	registry
service	service
device	device
registryRead	registry
registryReadWrite	registry

Access policy name: iothubowner

Permissions
- ☑ Registry read
- ☑ Registry write
- ✓ Service connect
- ✓ Device connect

Shared access keys

Primary key: PPlqYl6zINOTloZqqs2hno5FgQv4NCS4p-

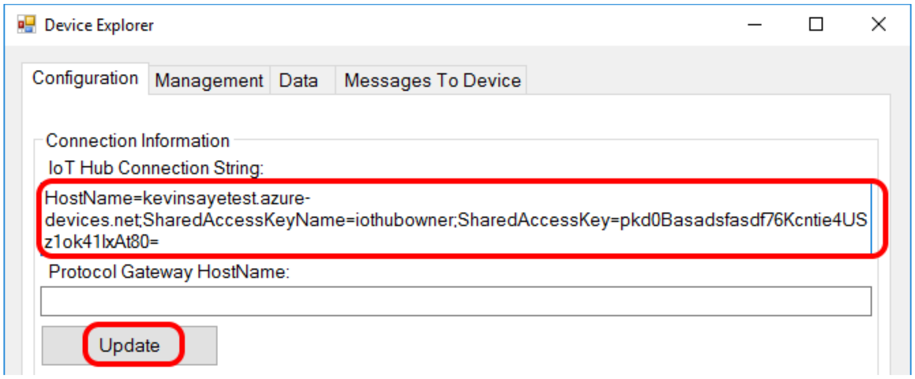
Secondary key: RVIRhI9o9HOrsqXjuVGxUYDQW1UpolY3ı

Connection string—primary key: HostName=hol.azure-devices.net;Shared.

Connection string—secondary key: HostName=hol.azure-devices.net;Shared.

Step 6. If not already installed, install (https://github.com/Azure/azure-iot-sdk-csharp/releases/download/2017-10-23/SetupDeviceExplorer.msi) and run the Device Explorer.

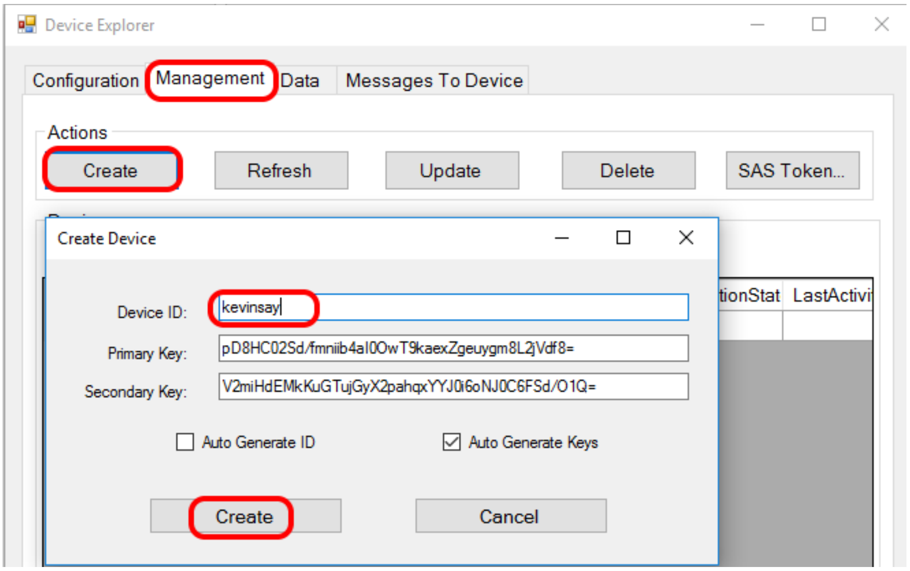
Step 7. Using the connection string copied from step 5, past it in Device Explorer and click Update.



IoT Hub Connection String:
HostName=kevinsayetest.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=pkd0Basadsfasdf76Kcntie4USz1ok41lxAt80=

Protocol Gateway HostName:

Click the Management Tab and Create. Then type in your device name and click Create.



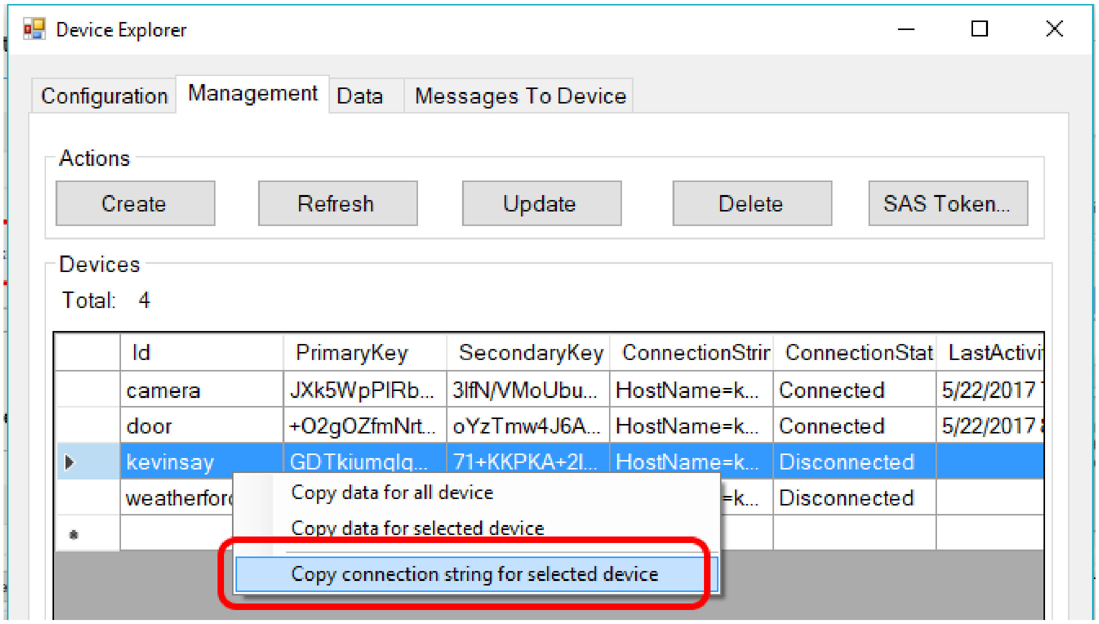
Device ID: kevinsay

Primary Key: pD8HC02Sd/fmniib4aI0OwT9kaexZgeuygm8L2jVdf8=

Secondary Key: V2miHdEMkKuGTujGyX2pahqxYYJ0i6oNJ0C6FSd/O1Q=

☐ Auto Generate ID ☑ Auto Generate Keys

Next, select your device, right click and copy the device connection string and save it to notepad to be used for our device.

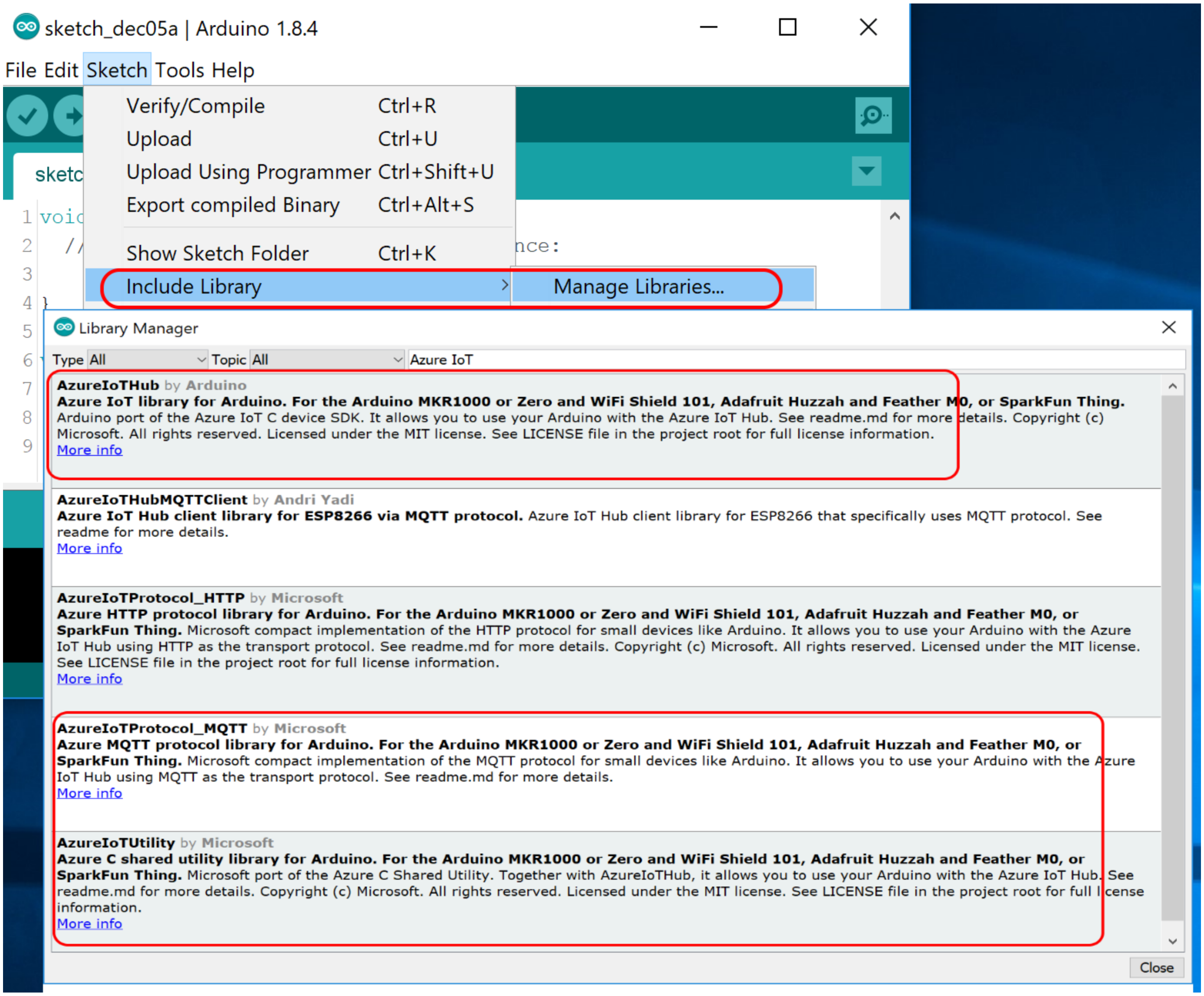


Devices

Total: 4

	Id	PrimaryKey	SecondaryKey	ConnectionStrir	ConnectionStat	LastActivi
	camera	JXk5WpPIRb...	3lfN/VMoUbu...	HostName=k...	Connected	5/22/2017
	door	+O2gOZfmNrt...	oYzTmw4J6A...	HostName=k...	Connected	5/22/2017
▶	kevinsay	GDTkiumqlq...	71+KKPKA+2I...	HostName=k...	Disconnected	
	weatherforc			=k...	Disconnected	
*						

Step 8. Launch Arduino. Add the Microsoft IoT and Arduino libraries by clicking Sketch → Include Library → Manage Libraries and search for “AzureIoT”. Install the library AzureIoTHub by Arduino, the AzureIoTProtocol_MQTT by Microsoft and AzureIoTUtility by Microsoft, as shown below.


sketch_dec05a | Arduino 1.8.4
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Manage Libraries...
Library Manager
Type All
Topic All
Azure IoT
Close

AzureIoTHub by Arduino
Azure IoT library for Arduino. For the Arduino MKR1000 or Zero and WiFi Shield 101, Adafruit Huzzah and Feather M0, or SparkFun Thing. Arduino port of the Azure IoT C device SDK. It allows you to use your Arduino with the Azure IoT Hub. See readme.md for more details. Copyright (c) Microsoft. All rights reserved. Licensed under the MIT license. See LICENSE file in the project root for full license information.
More info

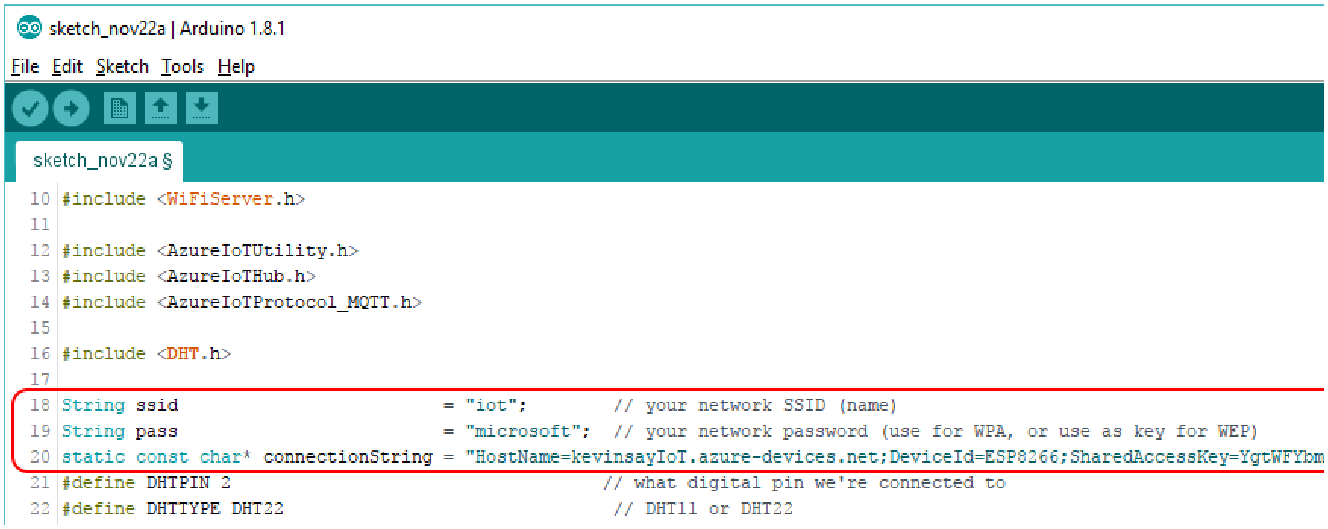
AzureIoTHubMQTTClient by Andri Yadi
Azure IoT Hub client library for ESP8266 via MQTT protocol. Azure IoT Hub client library for ESP8266 that specifically uses MQTT protocol. See readme for more details.
More info

AzureIoTProtocol_HTTP by Microsoft
Azure HTTP protocol library for Arduino. For the Arduino MKR1000 or Zero and WiFi Shield 101, Adafruit Huzzah and Feather M0, or SparkFun Thing. Microsoft compact implementation of the HTTP protocol for small devices like Arduino. It allows you to use your Arduino with the Azure IoT Hub using HTTP as the transport protocol. See readme.md for more details. Copyright (c) Microsoft. All rights reserved. Licensed under the MIT license. See LICENSE file in the project root for full license information.
More info

AzureIoTProtocol_MQTT by Microsoft
Azure MQTT protocol library for Arduino. For the Arduino MKR1000 or Zero and WiFi Shield 101, Adafruit Huzzah and Feather M0, or SparkFun Thing. Microsoft compact implementation of the MQTT protocol for small devices like Arduino. It allows you to use your Arduino with the Azure IoT Hub using MQTT as the transport protocol. See readme.md for more details.
More info

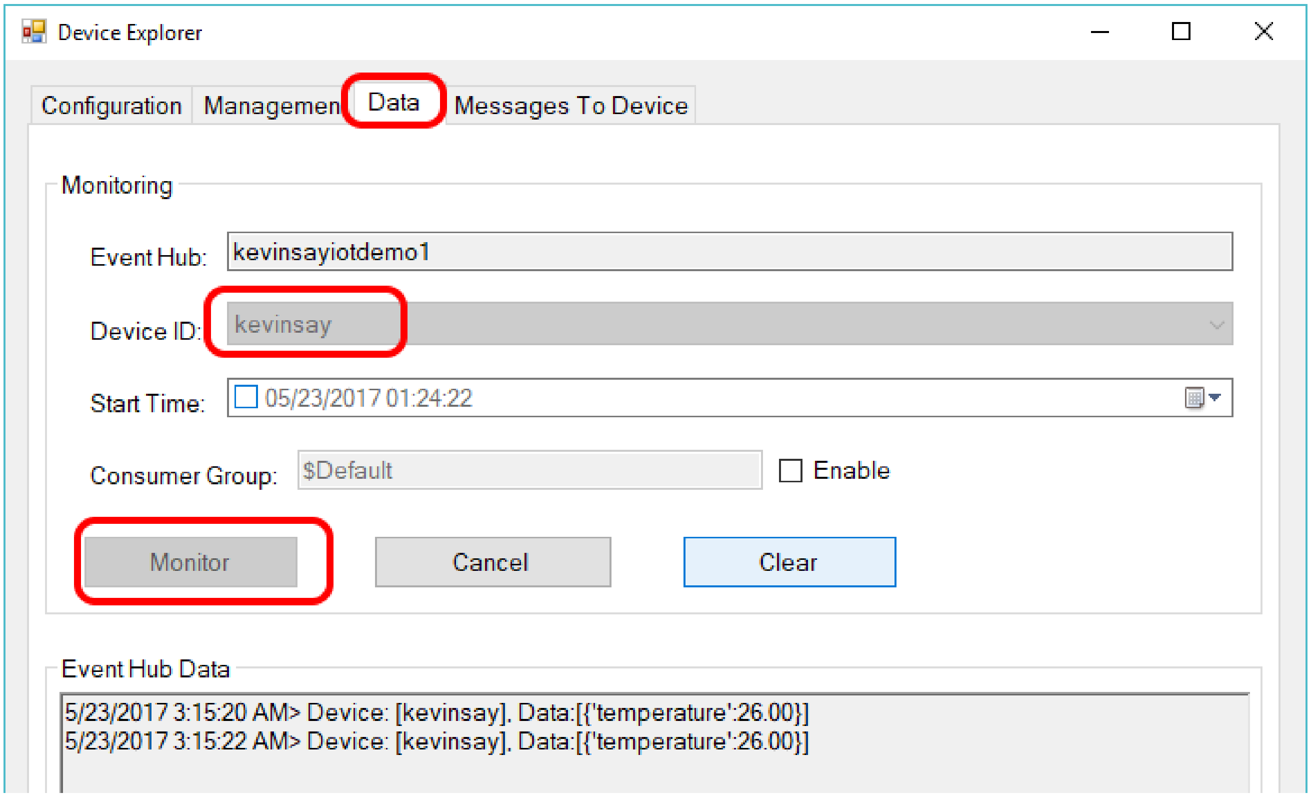
AzureIoTUtility by Microsoft
Azure C shared utility library for Arduino. For the Arduino MKR1000 or Zero and WiFi Shield 101, Adafruit Huzzah and Feather M0, or SparkFun Thing. Microsoft port of the Azure C Shared Utility. Together with AzureIoTHub, it allows you to use your Arduino with the Azure IoT Hub. See readme.md for more details. Copyright (c) Microsoft. All rights reserved. Licensed under the MIT license. See LICENSE file in the project root for full license information.
More info

Step 9. Starting with a new sketch and copy the content from: http://tinyurl.com/MayNodeMCU and modify the ssid, pass, and connectionString on lines 18 - 20 as shown below and click the "Compile and Upload" button.



```
10 #include <WiFiServer.h>
11
12 #include <AzureIoTUtility.h>
13 #include <AzureIoTHub.h>
14 #include <AzureIoTProtocol_MQTT.h>
15
16 #include <DHT.h>
17
18 String ssid                          = "iot";         // your network SSID (name)
19 String pass                          = "microsoft";  // your network password (use for WPA, or use as key for WEP)
20 static const char* connectionString = "HostName=kevinsayIoT.azure-devices.net;DeviceId=ESP8266;SharedAccessKey=YgtWFYbm
21 #define DHTPIN 2                                     // what digital pin we're connected to
22 #define DHTTYPE DHT22                                 // DHT11 or DHT22
```

Step 10. Using Device Explorer, click Data, select the correct device and click Monitor. The screen should look something like (calculating the temperature in Celsius):



Event Hub: kevinsayiotdemo1

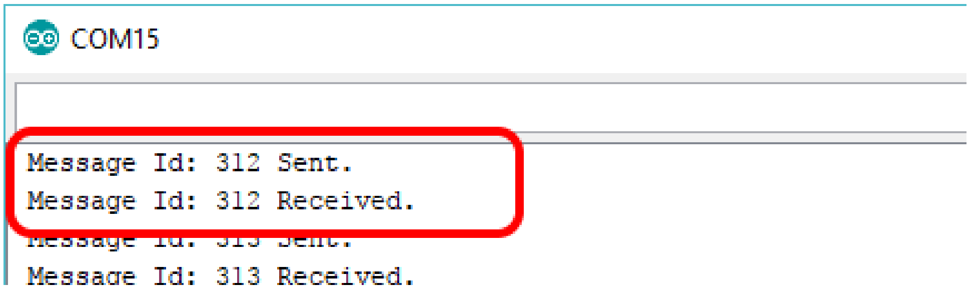
Device ID: kevinsay

Start Time: □ 05/23/2017 01:24:22

Consumer Group: $Default □ Enable

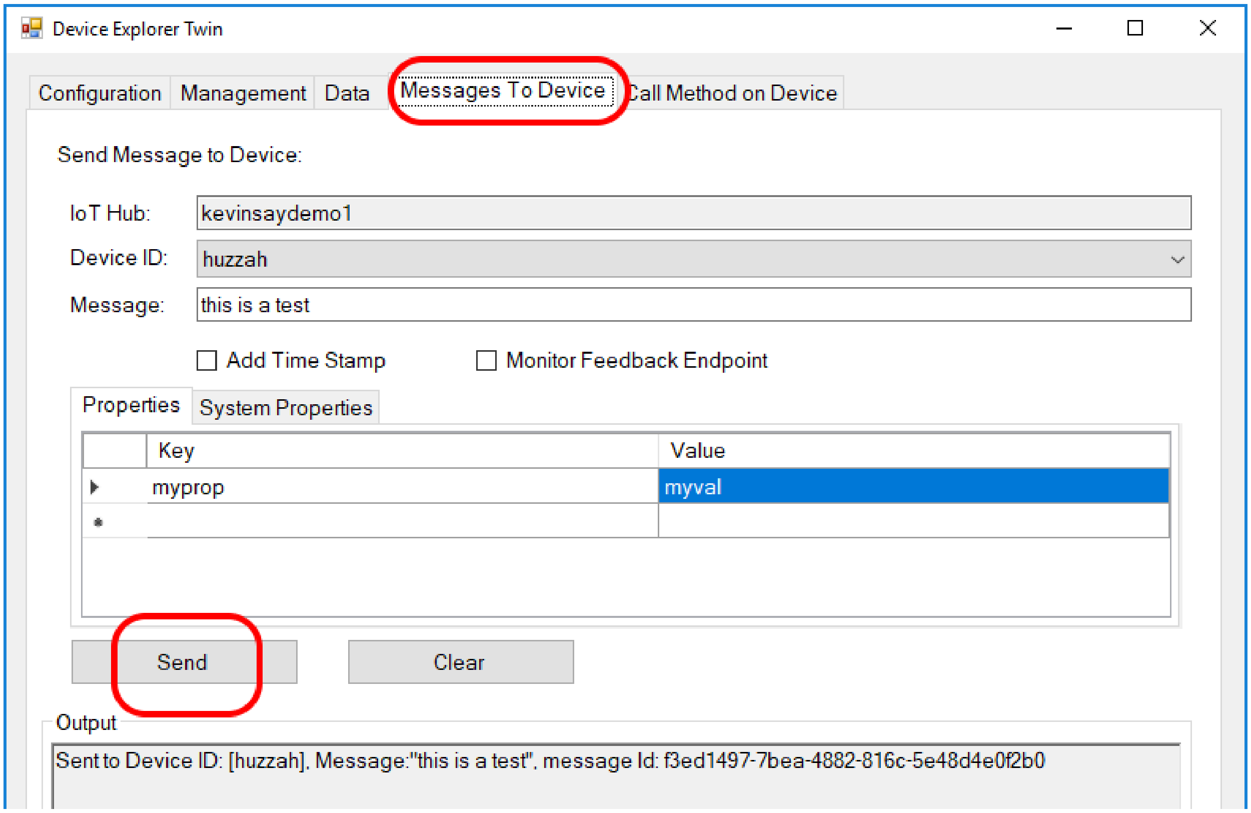
```
5/23/2017 3:15:20 AM> Device: [kevinsay], Data:[{'temperature':26.00}]
5/23/2017 3:15:22 AM> Device: [kevinsay], Data:[{'temperature':26.00}]
```

Step 11. Using Serial Monitor, you can see that messages are successfully sent and we get confirmation from IoT Hub that the message was received:



```
Message Id: 312 Sent.
Message Id: 312 Received.
Message Id: 313 Sent.
Message Id: 313 Received.
```

Step 12. Use Device Explorer, Click Message To Device. Type a message add any desired properties and click send, as shown below:



Send Message to Device:

IoT Hub: kevinsaydemo1

Device ID: huzzah

Message: this is a test

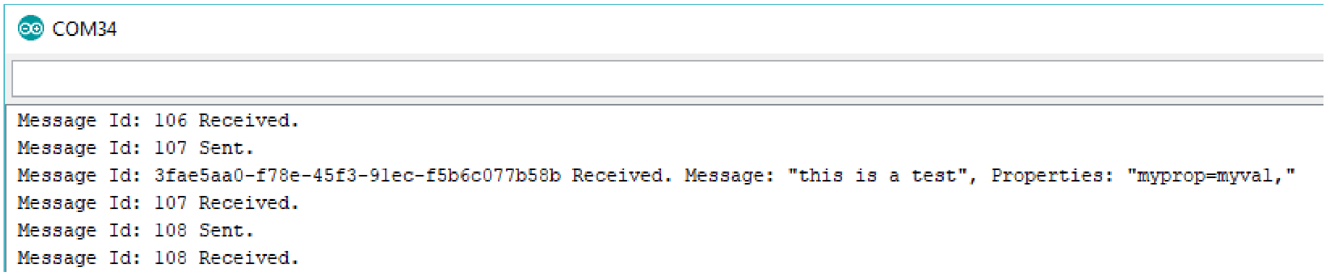
□ Add Time Stamp □ Monitor Feedback Endpoint

Key	Value
myprop	myval

Output

```
Sent to Device ID: [huzzah], Message:"this is a test", message Id: f3ed1497-7bea-4882-816c-5e48d4e0f2b0
```

Step 13. Using Serial Monitor, you can see that the message was received, and any properties:



```
Message Id: 106 Received.
Message Id: 107 Sent.
Message Id: 3fae5aa0-f78e-45f3-91ec-f5b6c077b58b Received. Message: "this is a test", Properties: "myprop=myval,"
Message Id: 107 Received.
Message Id: 108 Sent.
Message Id: 108 Received.
```

This completes this lab. If time permits, read the setup(), loop() and supporting functions. Note that the temperature is in Celsius, as we will adjust this later.