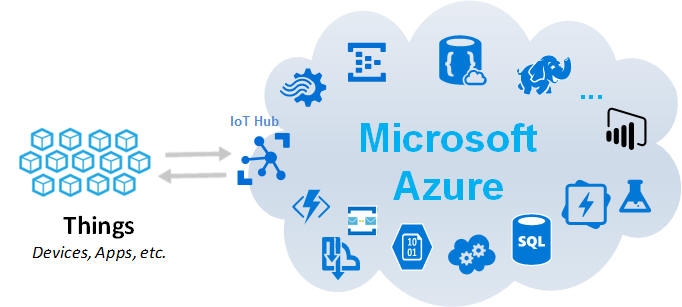
**PHẦN 1: TỔNG QUAN HƯỚNG DẪN ĐĂNG KÝ TÀI KHOẢN**

**VÀ CÀI ĐẶT CÔNG CỤ HỖ TRỢ**

***I Tổng quan:***

Sự bùng nổ về cuộc cách mạng 4.0 đã khiến cụm từ[*Internet of Things*](https://giaiphapiot.vn/tag/internet-of-things/)  trở lên không còn quá xa lạ với nhiều người. Để bắt kịp xu hướng này, các hãng công nghệ lớn như Microsoft, Amazon, Google đều tung ra những bộ công cụ và nền tảng để phục vụ cho một giải pháp tổng thể về IoT.

Cụ thể, Microsoft đã tạo ra các phiên bản dịch vụ điện toán đám mây Azure IoT, một mảng dịch vụ được vận hành trên nền tảng Microsoft Azure. Với Azure IoT, nó  cung cấp giao tiếp hai chiều an toàn giữa các thực thể IoTs và Azure Cloud( các thực thể có thể là cảm biến, vi điều khiển nhúng hoặc thiết bị thông minh), giúp các nhà phát triển có thể kết nối, quản lý thiết bị hay thu thập, phân tích và trích xuất dữ liệu. Azure IoT bao gồm nhiều dịch vụ con với chức năng khác nhau như Azure IoT Hub, Azure Stream Analytics on IoT Edge, Microsoft IoT Central.



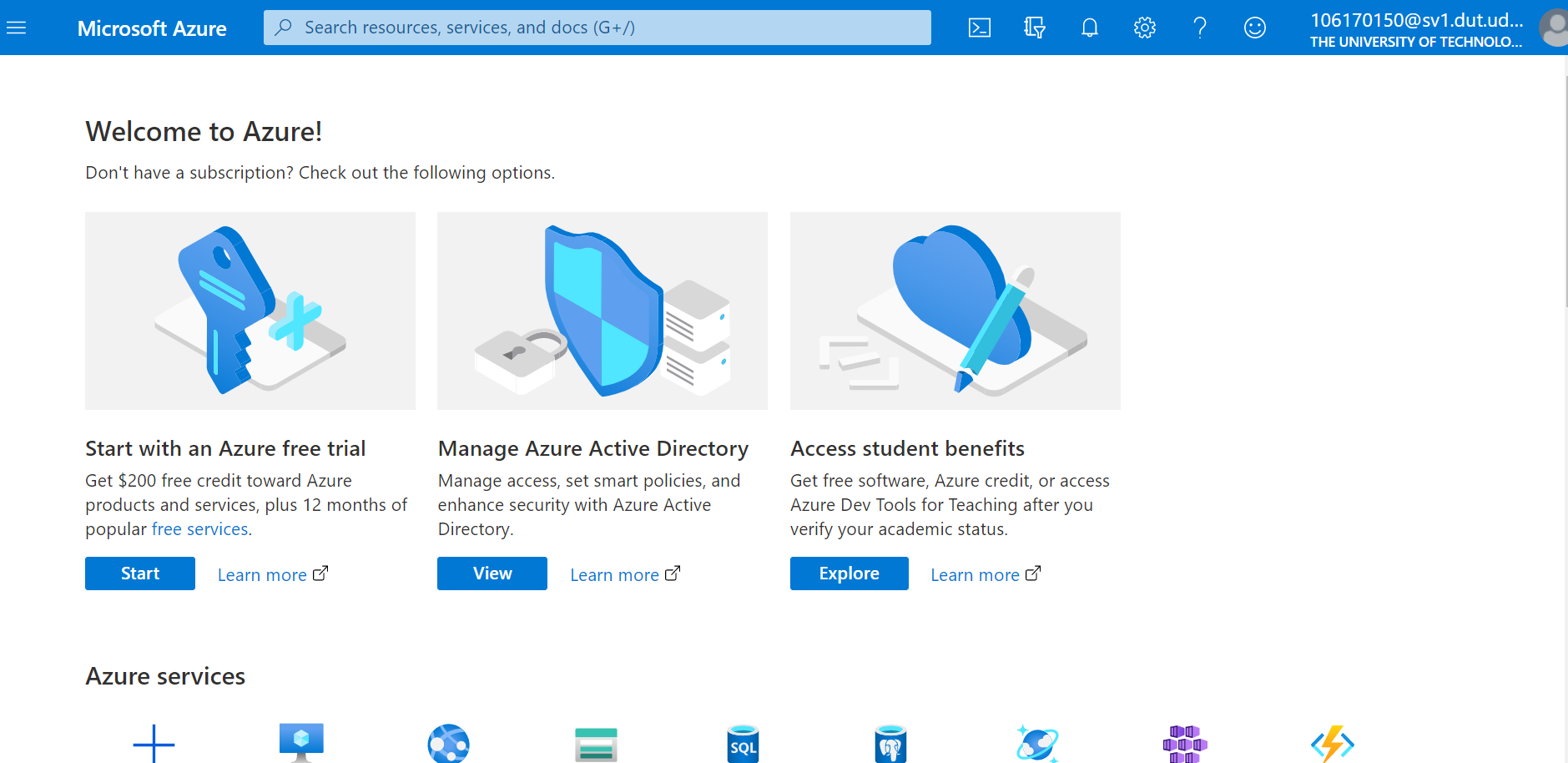
Trong chuỗi bài viết này, nhóm sẽ hướng dẫn các bạn sử dụng dịch vụ Azure IoT Hub để kết nối và quản lý hệ thống thiết bị IoT. Với dịch vụ này nó có thể cung cấp:

* Nhiều tùy chọn giao tiếp từ thiết bị đến cloud và từ cloud về thiết bị.
* Bảo mật kết nối bằng các sử dụng thông tin xác thực bảo mật cho mỗi thiết bị và kiểm soát truy cập.
* Có thể truy vấn thông tin trạng thái trên mỗi thiết bị.
* Kết nối thiết bị dễ dàng với các ngôn ngữ và nền tảng phổ biến.

***II Hướng dẫn đăng ký tài khoản****:*

Để bắt đầu với Azure IoT hub, trước tiên bạn cần chuẩn bị một tài khoản Microsoft ( các bạn nên đăng ký thông tin trùng với thẻ VISA để đăng ký là tốt nhất), một thẻ VISA của bất kỳ ngân hàng nào cũng được và một số điện thoại để verify (xác thực) tin nhắn lúc bạn thực hiện đăng ký( bạn lưu ý là số điện thoại này phải chưa từng đăng ký Azure nhé ).

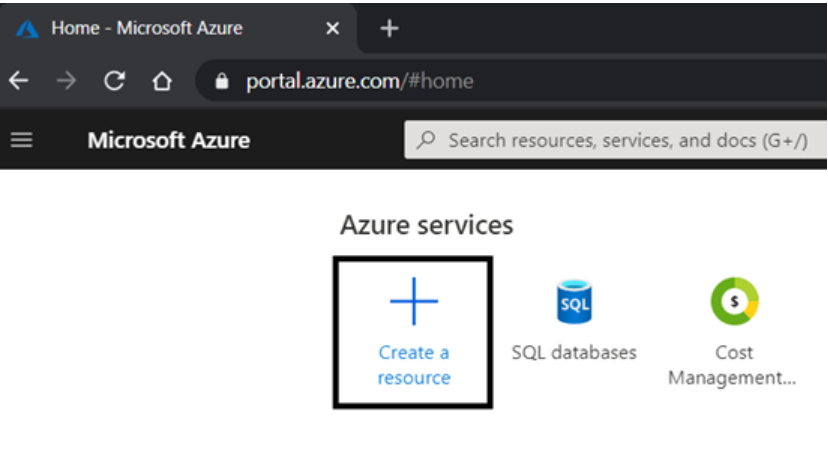
Sau khi chúng ta đã chuẩn bị xong, để sử dụng dịch vụ Azure IoT hub, chúng ta truy cập vào đường link <https://portal.azure.com/#home> , sử dụng tài khoản Microsoft đã chuẩn bị để đăng nhập vào. Lúc này, giao diện như hình bên sẽ xuất hiện, ta click vào Start rồi lần lượt xát minh thông tin cá nhân, số điện thoại,thẻ VISA .



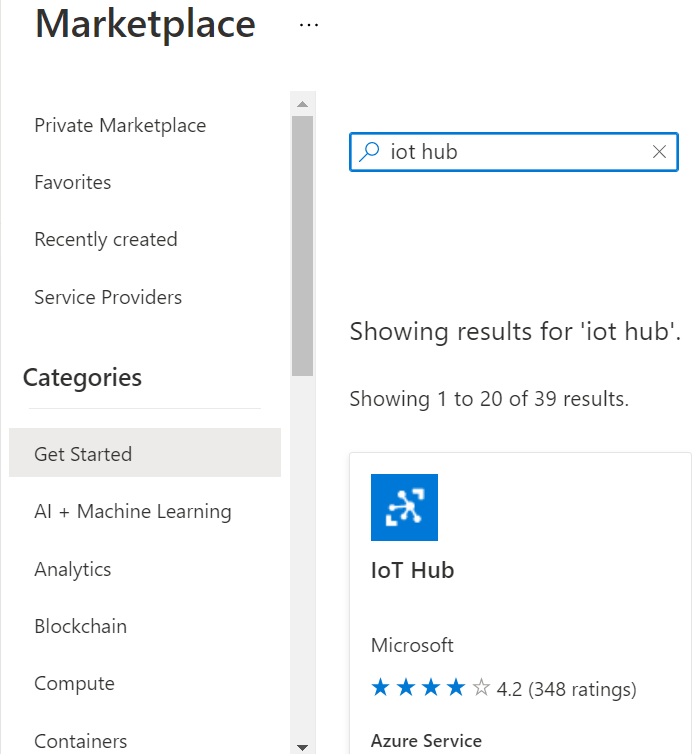
Để bắt đầu khởi tạo một project chúng ta cần tạo một IoT Hub.Quá trình này trải qua các bước:

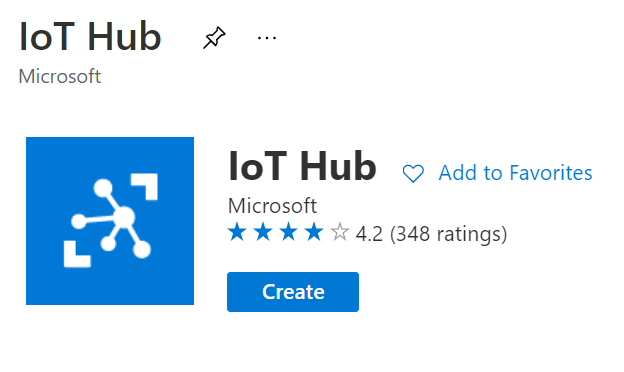
Bước 1: Đăng nhập vào <https://portal.azure.com/>

Bước 2: Từ trang chủ Azure, hãy chọn nút ***+ Create a resource***, sau đó nhập *IoT Hub* vào trường *Search the Marketplace*.

****

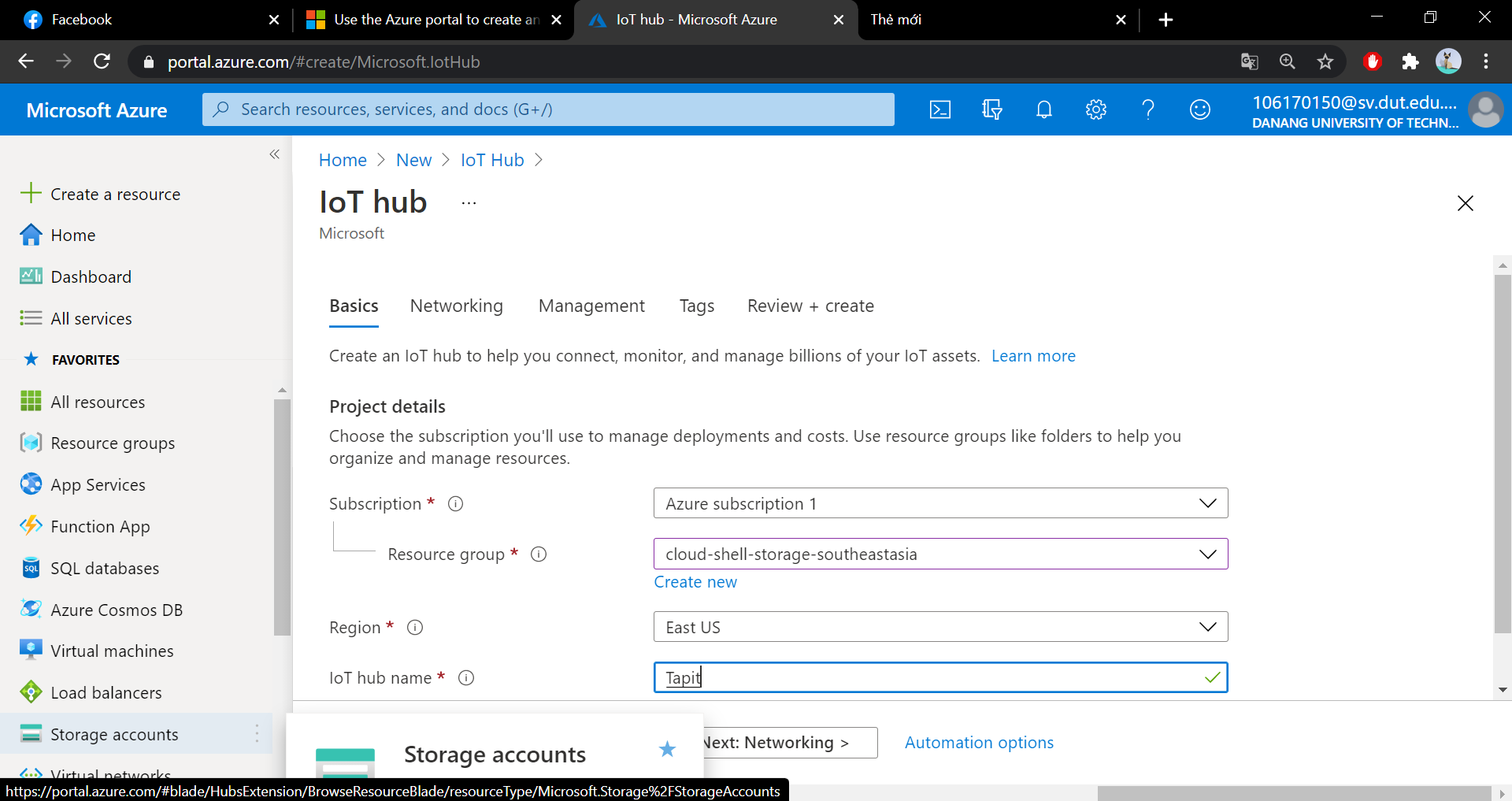
**Bước 3:** Chọn ***IoT Hub*** từ kết quả tìm kiếm, sau đó chọn ***Create***

****

****

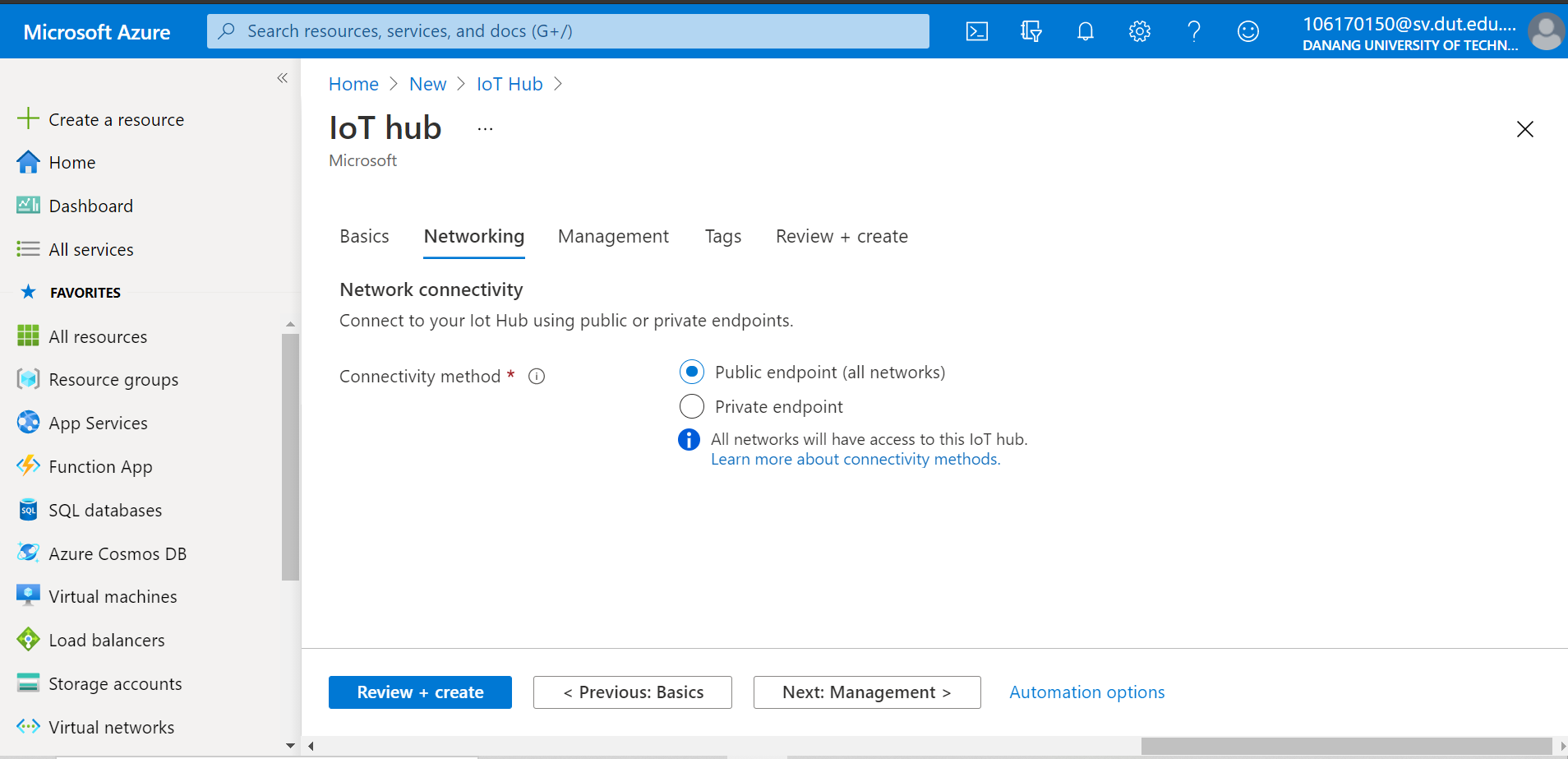
Bước 4: Hoàn thành các trường như sau: *Subscription, Resource Group, Region, IoT Hub Name.*

Sau đó chọn *Next: Networking* để tiếp tục tạo IoT hub của bạn.

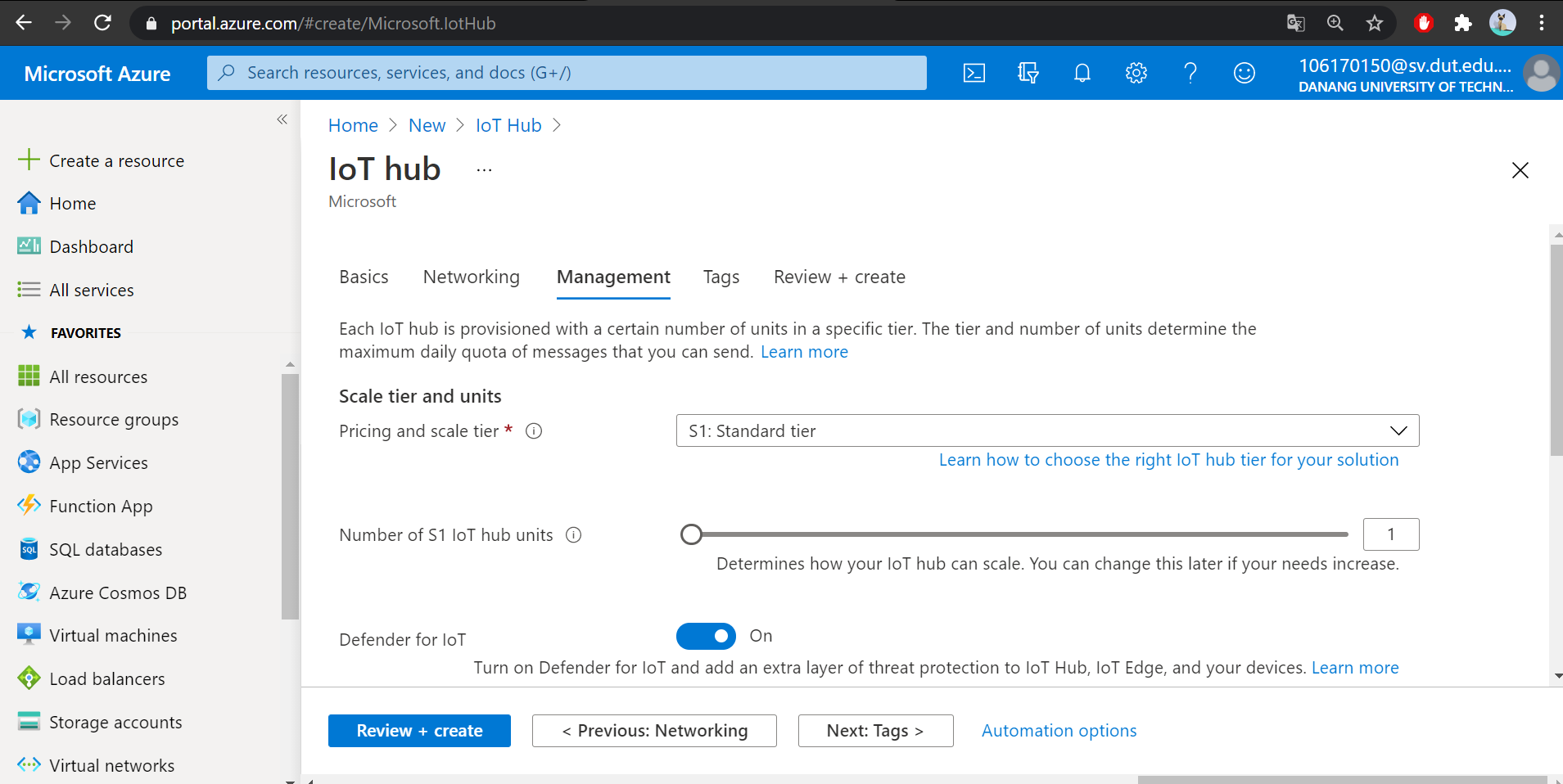


Bước 5: Chọn endpoints để kết nối với các IoT hub của bạn.

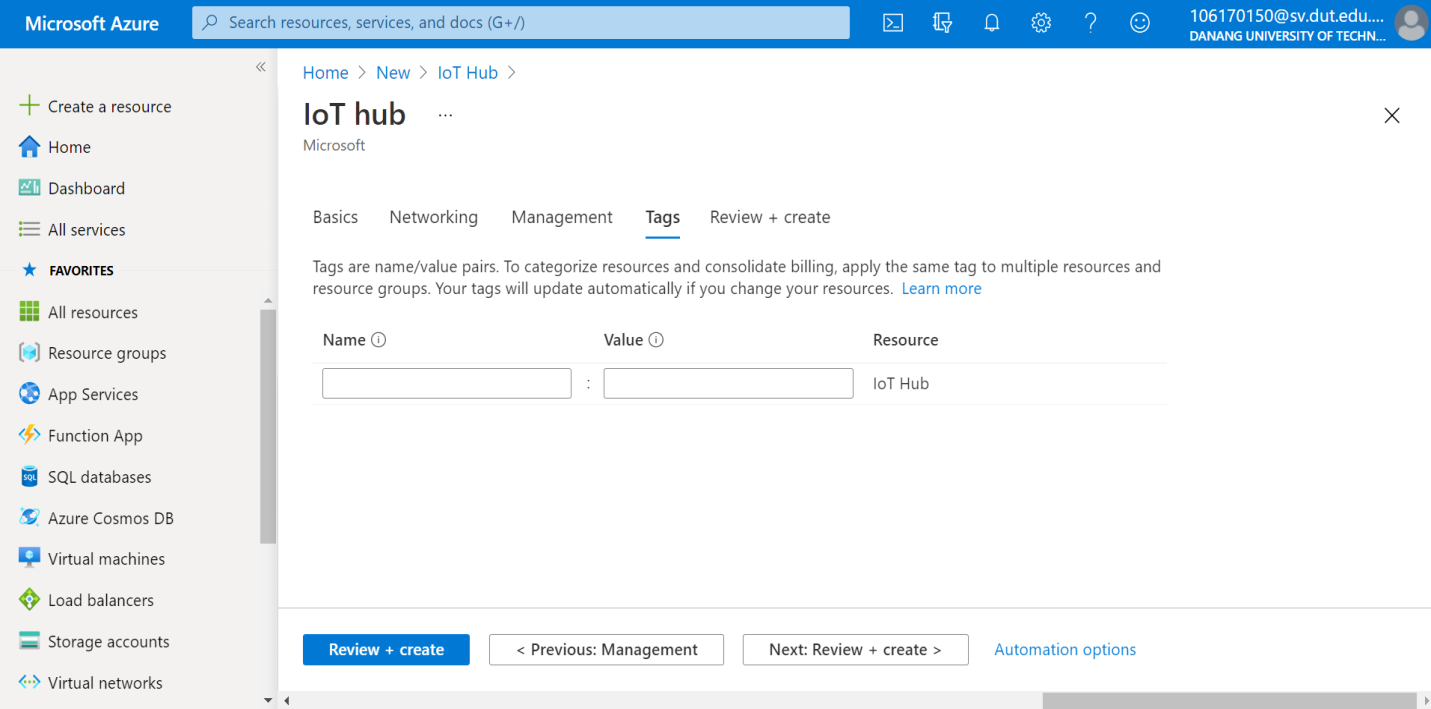
Sau đó chọn *Next: Management* để tiếp tục tạo IoT hub.

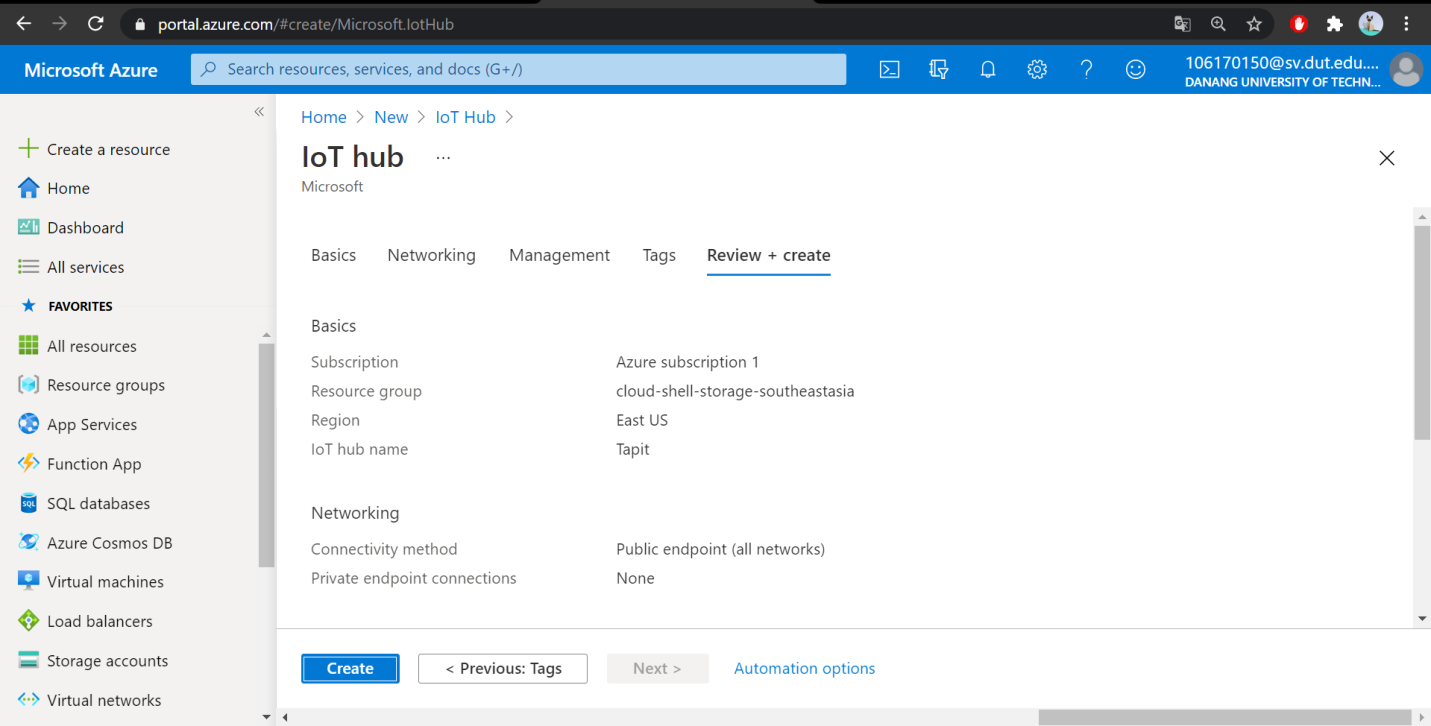
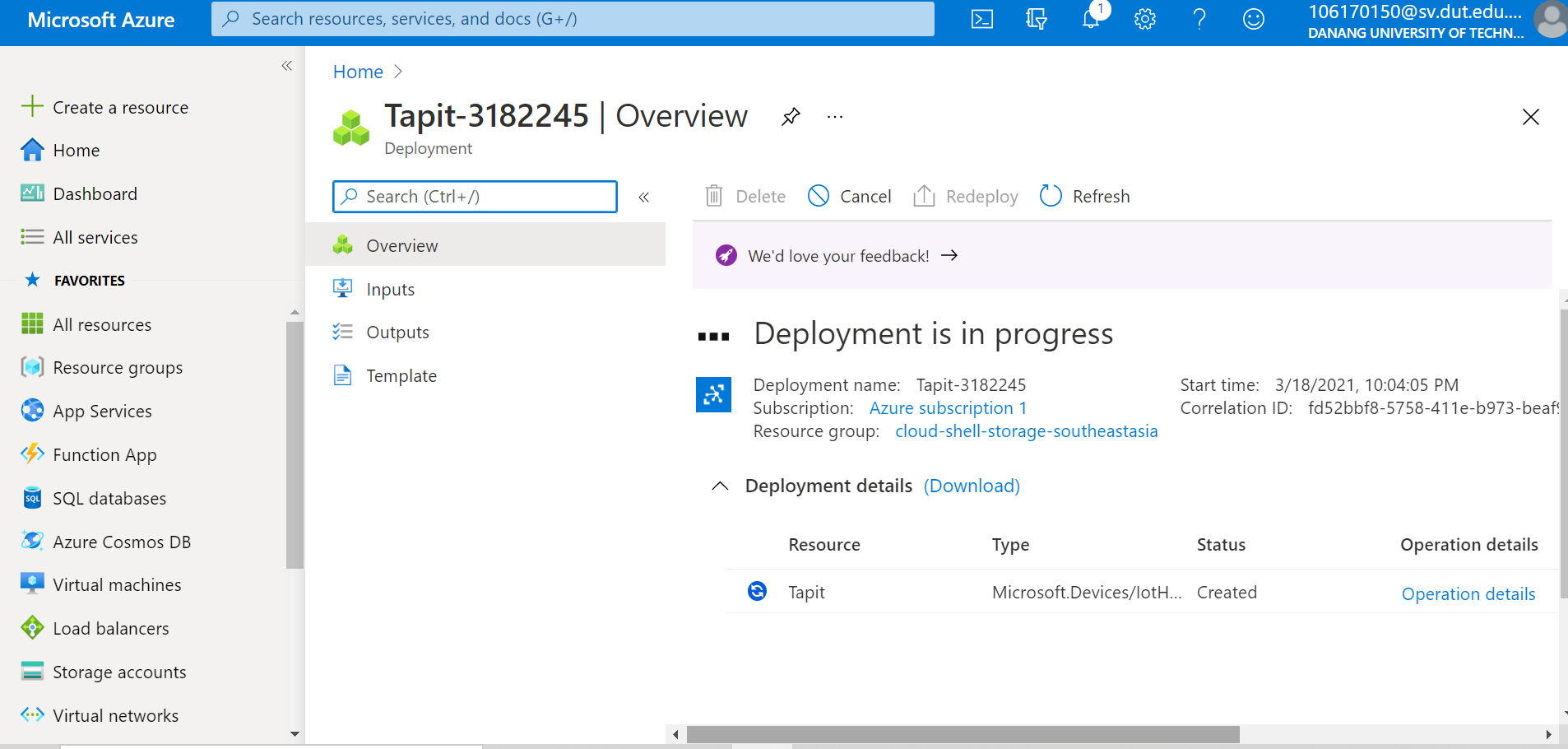


Bước 6: Bạn có thể chấp nhận cài đặt mặc định tại đây. Chọn ***Next: Tag*** để tiếp tục.



Bước 7: Chọn*Next: Review + create* để xem lại các lựa chọn của bạn



Vậy là bạn đã tạo một IoT hub thành công

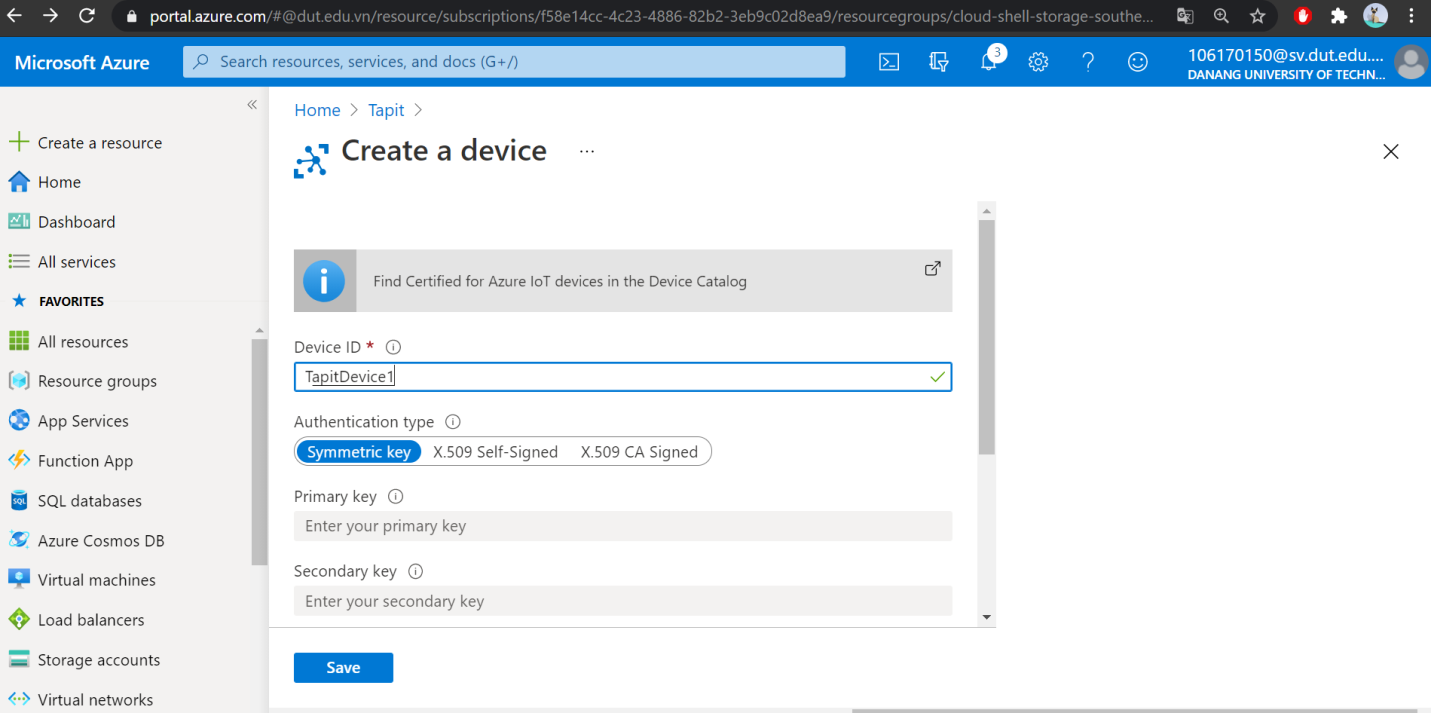
***III Đăng ký thiết bị đầu tiên:***

Các thiết bị kết nối với Azure IoT hub được xem là các “device”.Các device này sẽ lưu trữ thông tin trạng thái thiết bị nhờ vậy mà chúng ta có thể thực hiện truy vấn (cập nhật,truy xuất,xóa,...).

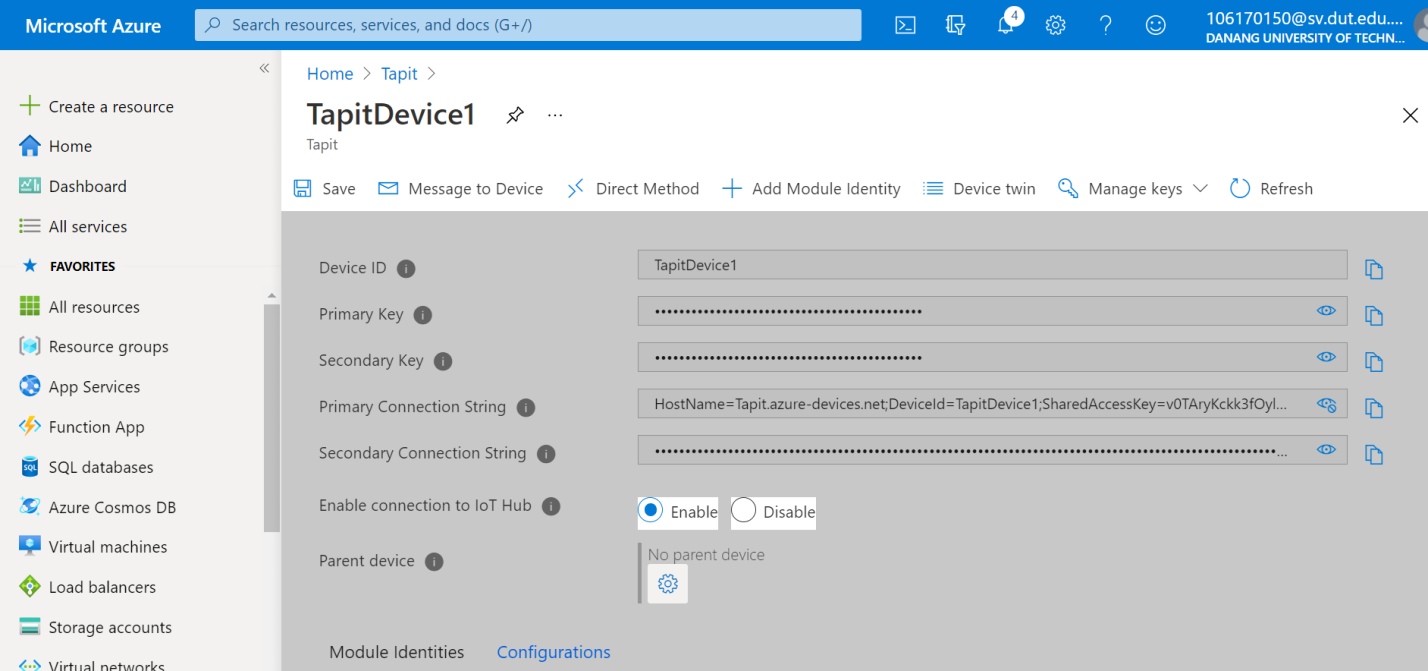
Có 3 cách để đăng ký một thiết bị là đăng ký trực tiếp ngay trên *Portal,* sử dụng  *Azure Cloud Shell* hoặc *Visual Studio Code*.Với bài viết này nhóm sẽ hướng dẫn các bạn cách đăng ký một thiết bị bằng Portal thông qua các bước như sau:

Bước 1: Đăng nhập vào <https://portal.azure.com/> và điều hướng đến IoT hub của bạn.

Bước 2:Chọn *IoT Devices* từ menu sau đó nhập tên Devices vào mục *Device ID*. Chúng ta có thể tạo Device bất kì nhưng không được phép trùng tên với các Device khác trong cùng một IoT hub,ở đây mình ví dụ đặt tên là TapitDevice1.

Bước 3:Chọn save để hoàn tất quá trình tạo một thiết bị mới.

Sau khi quá trình tạo một thiết bị thành công, chúng ta click vào thiết bị mới tạo tại *IoT Devices* và sao chép chuỗi *Primary Connection String* để phục vụ cho việc kết nối sau này.

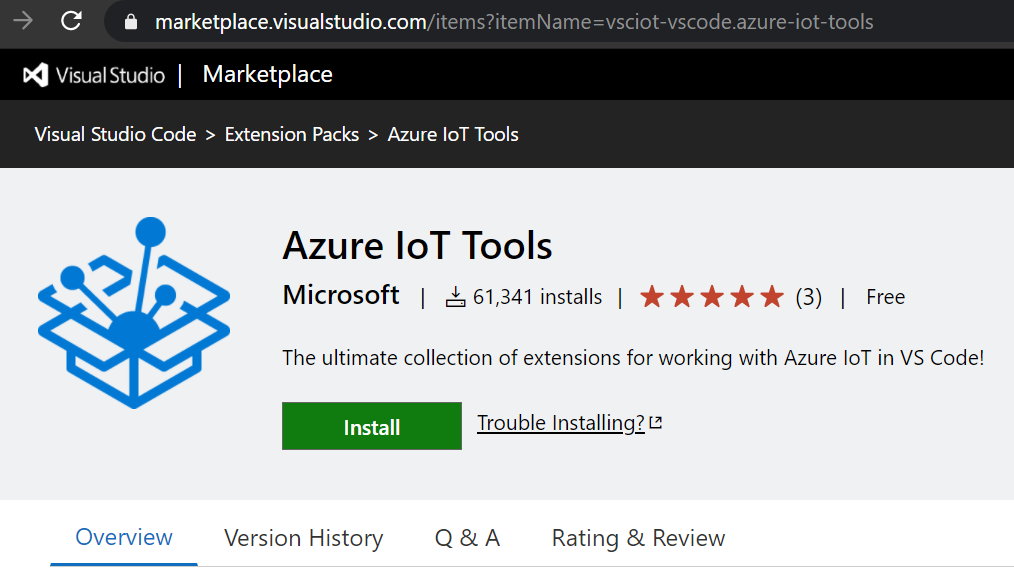


***IV Cài đặt các công cụ hỗ trợ::***

**1 Azure IoT Tools:**

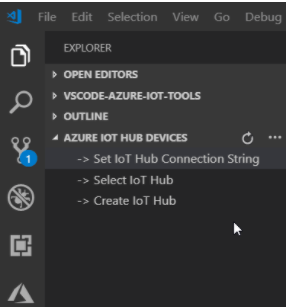
Azure IoT Tools là một tiện ích mở rộng của Visual Studio Code, nó giúp quản lý IoT Hub và phát triển ứng dụng IoT một cách dễ dàng hơn. Để cài đặt ta truy cập link bên dưới và click vào Install

<https://marketplace.visualstudio.com/items?itemName=vsciot-vscode.azure-iot-tools>



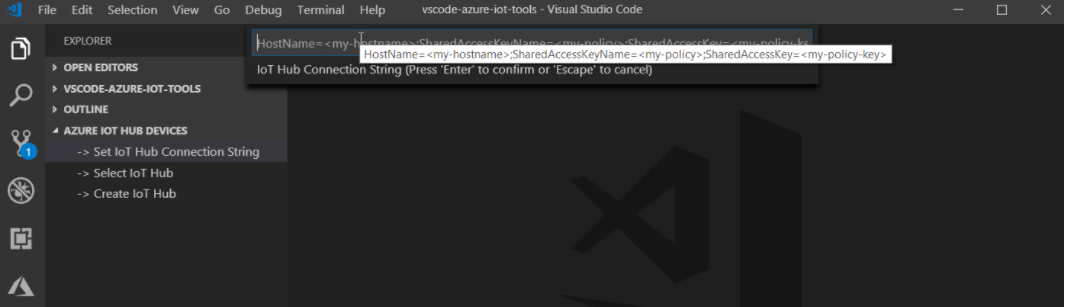
Sau khi cài đặt bạn có thể thiết lập Azure IoT Hub thông qua chuỗi Connection String mà bạn đã lưu lại khi đăng kí thiết bị từ trước bằng cách:

Bước 1: Tại *Explorer* của VS Code,chúng ta Click vào "*Azure IoT Hub Devices*" ở góc dưới cùng bên trái. Rồi sau đó chọn "*Set IoT Hub Connection String*"

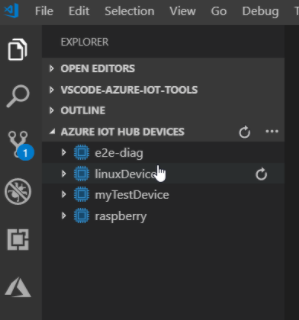


Bước 2: Sau đó bạn nhập chuỗi *Connection String* đã được lưu trước đó.Chuỗi này sẽ có dạng:

*HostName=<my-hub>.azure-devices.net;SharedAccessKeyName=<my-policy>;SharedAccessKey=<my-policy-key>*



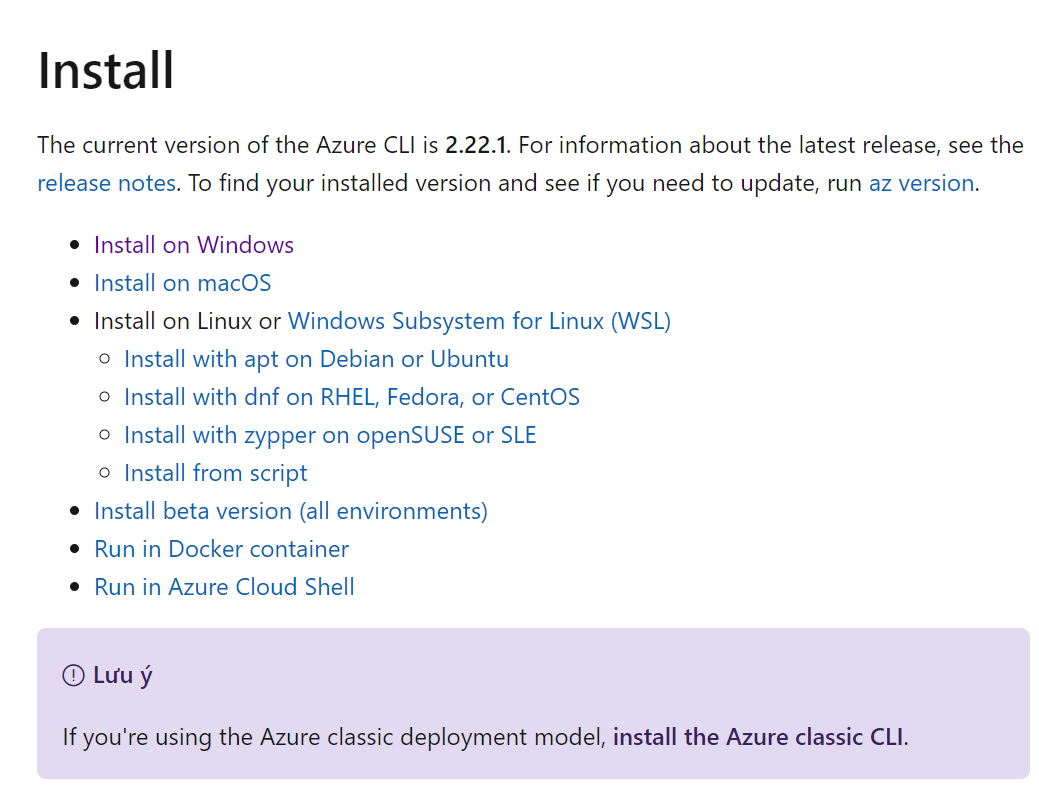
Bước 3: Danh sách các Device bạn đã tạo sẽ xuất hiện. Như vậy là bạn đã cài đặt Azure IoT tool thành công.



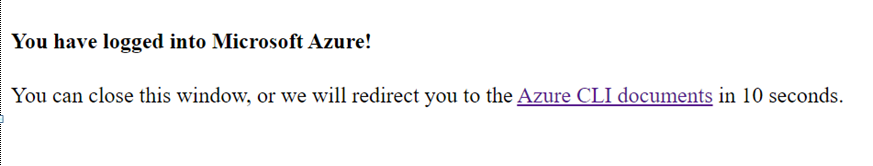
**2 Azure CLI:**

Với công cụ Azure CLI chúng ta có thể sử dụng các lệnh để tạo và quản lý các Azure resources. Azure CLI đã có sẵn để cài đặt trong các hệ điều hành như Windows, macOS và Linux và nó Cũng có thể chạy trong Docker và Azure Cloud Shell. Nhờ vậy mà Azure CLI giúp bạn dễ dàng làm việc dễ dàng với các ngôn ngữ lập trình và môi trường phần mềm khác nhau.

Để cài đặt chúng ta truy cập vào link <https://docs.microsoft.com/vi-vn/cli/azure/install-azure-cli?view=azure-cli-latest> và lựa chọn sao cho phù hợp với hệ điều hành mà ta sử dụng.



Sau khi cài đặt bạn thử chạy lệnh : “*az login*” .Lúc này màn hình hiển thị như hình bên dưới là bạn đã thành công.



Như vậy, Azure IoT Tools và Azure CLI là hai công cụ hỗ trợ đắt lực và cần thiết cho các bạn khi ta làm việc Azure IoT hub.Thông qua hai công cụ này có thể giúp chúng ta quản lý, sử dụng Azure IoT hub của mình một cách dễ dàng hơn đặc biệt là đối với các bạn bước đầu tiếp xúc với Azure.

Chúc các bạn cài đặt các công cụ hỗ trợ này thành công.

**Hướng dẫn kết nối thiết bị với IoT Hub qua giao thức MQTT**

IoT Hub cho phép thiết bị kết nối với “IoT Hub endpoints” qua các phiên bản:

* MQTT v3.1.1 on port 8883
* MQTT v3.1.1 over WebSocket on port 443.

Một thiết bị có thể sử dụng giao thức MQTT để kết nối với IoT hub thông qua một trong hai cách dưới đây:

* Thư viện trong Azure IoT SDKs: được xây dựng sẵn để dễ dàng kết nối với IoT hub một cách nhanh chóng
* Sử dụng trực tiếp giao thức MQTT

Trong phạm vi bài viết mình sẽ hướng dẫn cách thứ 2, mình sẽ sử dụng trực tiếp giao thức MQTT để kết nối với IoT hub

Thiết bị khi kết nối đến IoT Hub bắt buộc phải hỗ trợ bảo mật TLS/SSL và có thể xác thực danh tính thông qua một trong hai cách là: cơ chế Username/Password hoặc sử dụng chứng chỉ X5.509

Trong bài viết này mình sẽ hướng dẫn theo bảo mật theo kiểu xác thực danh tính theo cơ chế **Username/Password** sử dụng MQTT.fx.

1. Sử dụng MQTT.fx

Giao thức MQTT cung cấp các trường như ClientId, Username, Password, và được định nghĩa trong IoT hub như sau:

* Trường ClientId: sử dụng tên deviceId lúc khởi tạo thiết bị
* Trường Username: {iothubhostname}/{device\_id}/?api-version=2018-06-30

Trong đó:

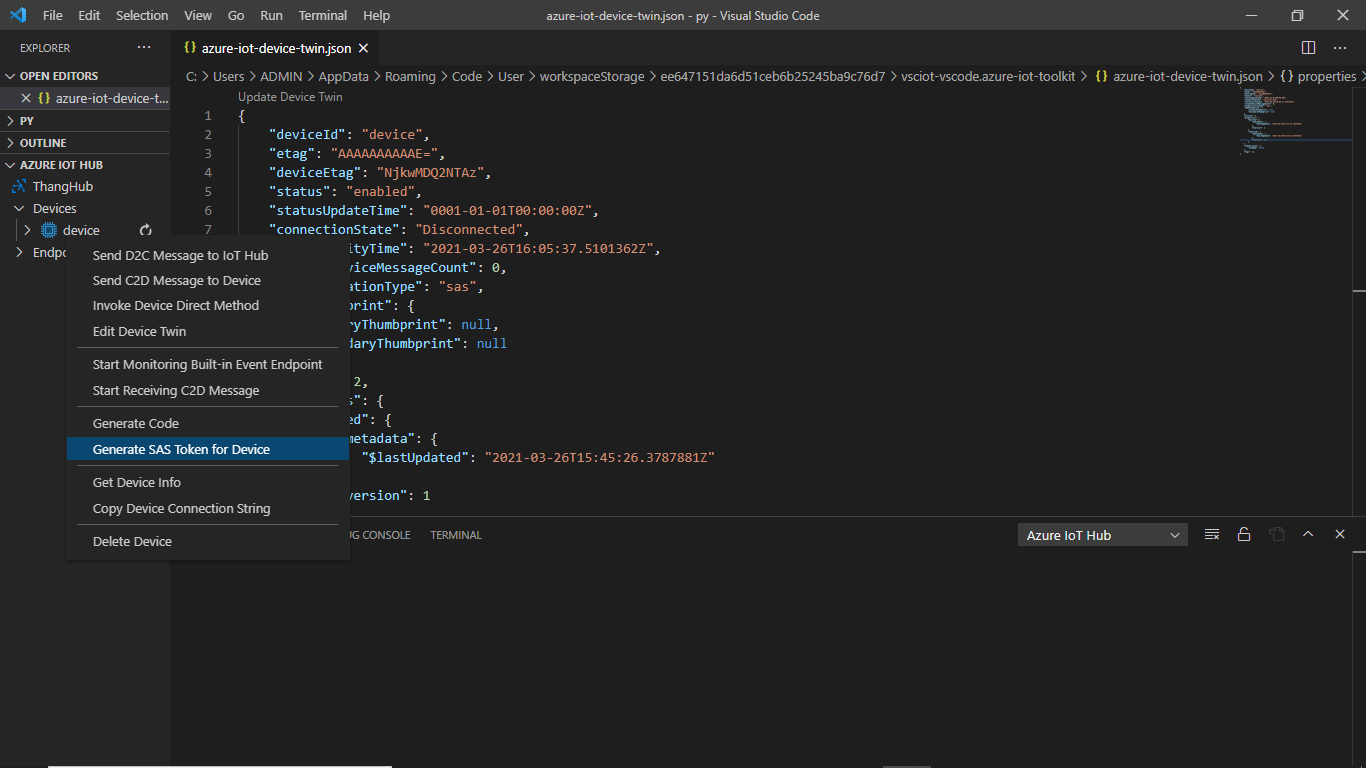
* {iothubhostname}: là tên host IoT hub
* {device\_id}: tên thiết bị

Ví dụ nếu tên IoT hub của bạn là **ThangHub.azure-devices.net**, tên thiết bị là **device** thì trường Username là **ThangHub.azure-devices.net/device/?api-version=2018-06-30**

* Trường Password: sử dụng mã SAS (Shared Access Signature), có thể tạo bằng cách sử dụng công cụ hỗ trợ đã giới thiệu ở **PHẦN 1** là Azure IoT tools hoặc Azure CLI

**Đối với sử dụng Azure IoT tools:**

* Tại tab AZURE IOT HUB phía bên trái của Visual Studio Code
* Nhấn chuột phải vào thiết bị bạn muốn tạo mã SAS, và chọn Generate SAS Token for Device
* Đặt thời hạn ( Ví dụ: 9000 giờ tức là 1 năm) và nhấn Enter



Mã SAS được tạo ra có định dạng:

SharedAccessSignature sig={signature-string}&se={expiry}&sr={URL-encoded-resourceURI}

Chúng ta sử dụng mã này tại trường Password

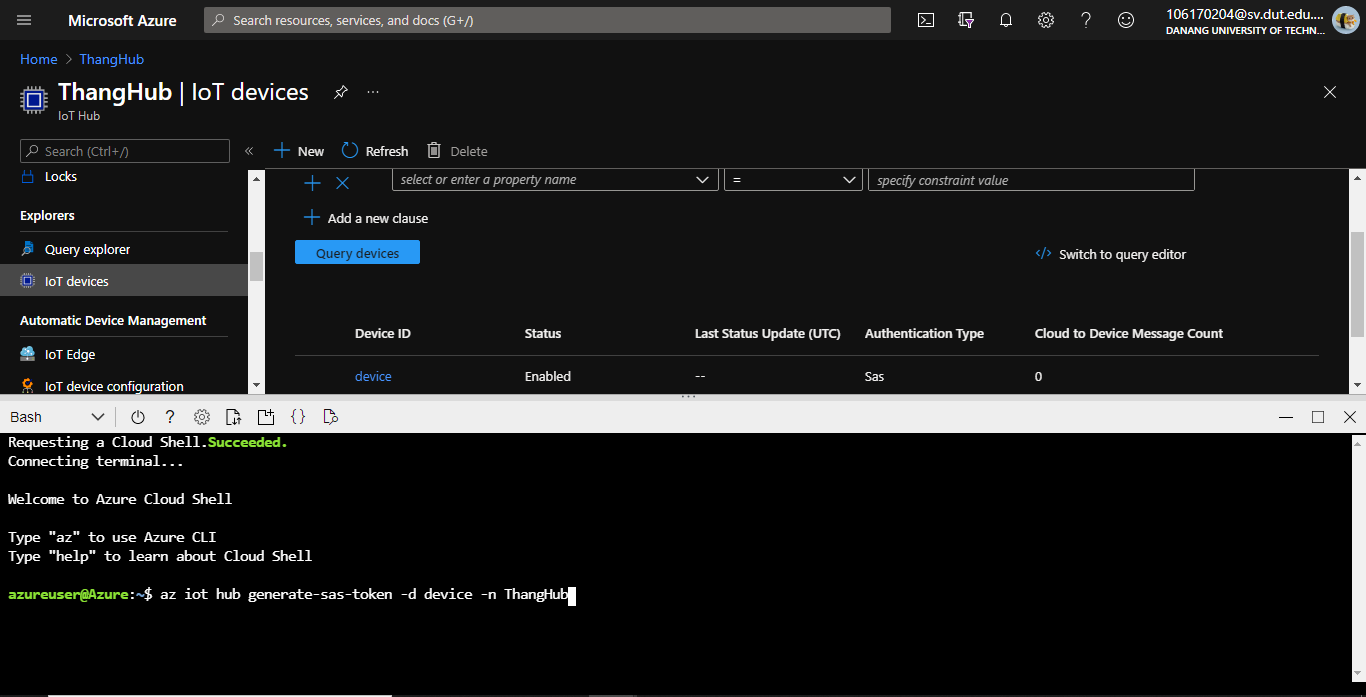
**Đối với sử dụng Azure CLI:**

Nhập với lệnh: az iot hub generate-sas-token -d {device\_id} -n {iothub\_name}

Trong đó:

{device\_id}: tên thiết bị để kết nối

{iothub\_name}: tên IoT hub



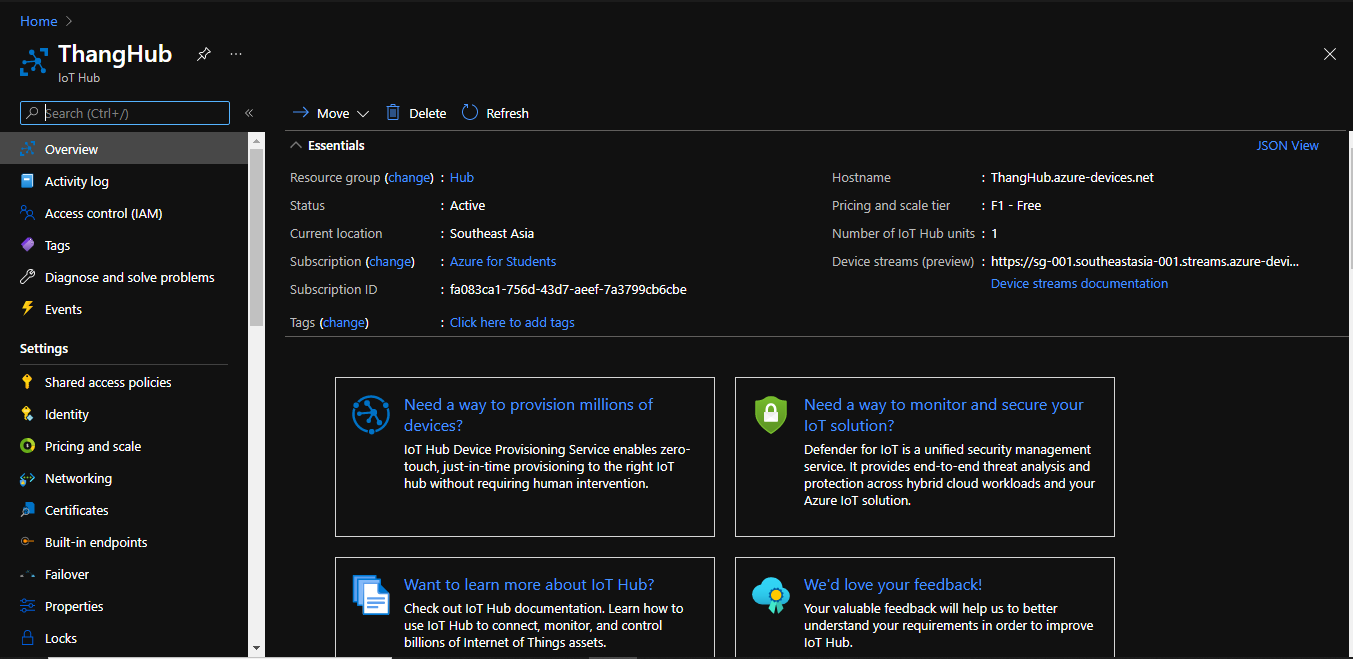
Và nhấn Enter để chạy lệnh, mã SAS được tạo ra và hiện ở command.

Tại MQTT.fx:

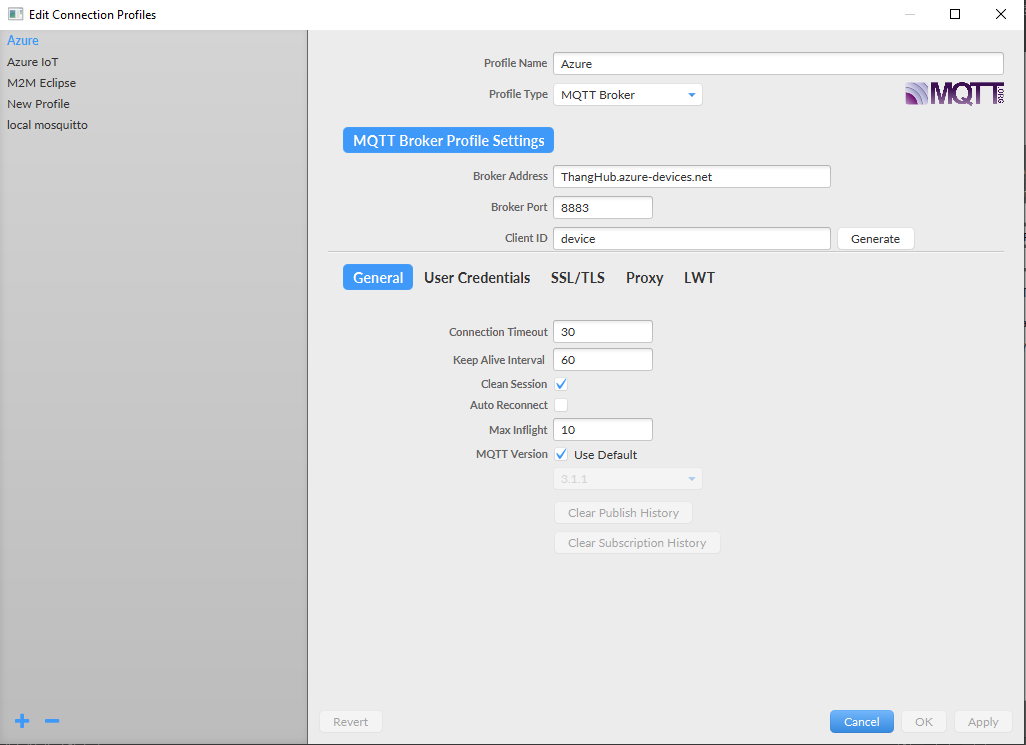
Tại mục MQTT Broker Profile Settings:

* Broker Address: là địa chỉ host name IoT hub của bạn
* Broker Port: sử dụng port 8883
* Client ID: tên thiết bị IoT hub

Chúng ta có thể lấy địa chỉ host name của mình tại mục Overview



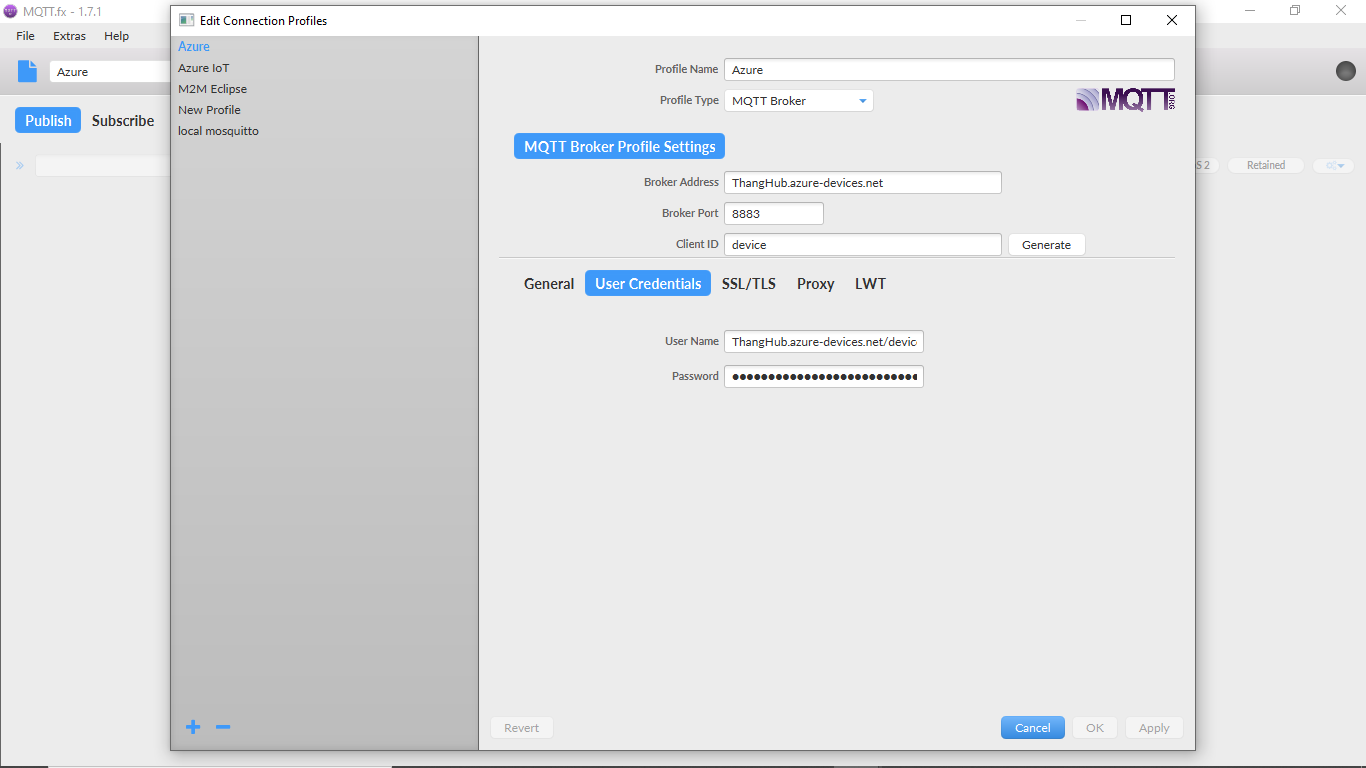
Các thông số sau khi điền tại mục MQTT Broker Profile Settings



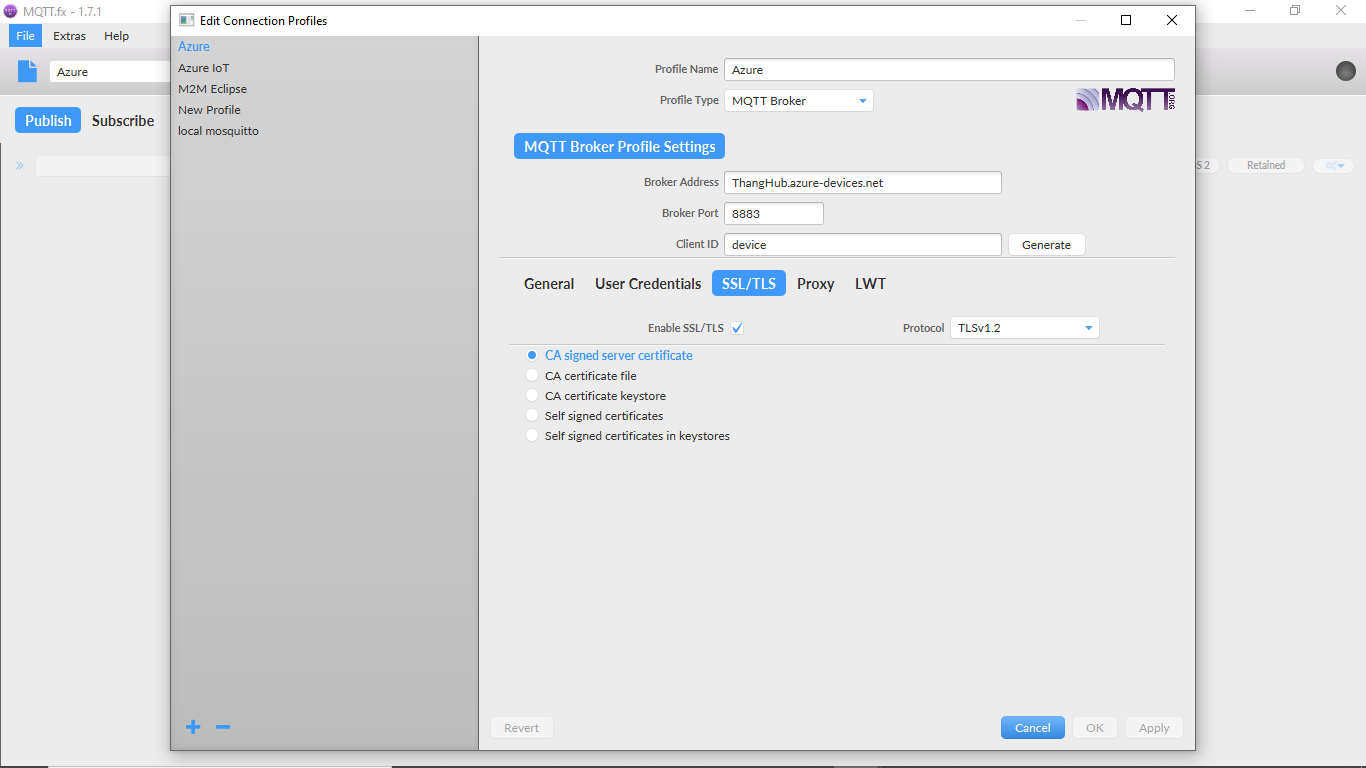
Chúng ta tạm không quan tâm đến các thông số General

Tại mục User Credentials:

* User Name: {iothubhostname}/{device\_id}/?api-version=2018-06-30
* Password: mã SAS được tạo ở phía trên



Vì chúng ta đang sử dụng bảo mật SSL/TLS nên tích chọn “Enable SSL/TLS”



Chúng ta nhấn “OK” để hoàn tất việc cấu hình.

Để kết nối tới thiết bị IoT hub chúng ta nhấn CONNECT.

Nếu thành công thì sẽ có biểu tượng ổ khóa và ô chấm màu xanh



Như vậy qua bài này, mình đã giới thiệu cho các bạn cách thiết lập các thông số, cũng như cách kết nối với IoT hub qua giao thức MQTT.fx. Phần tới mình sẽ hướng dẫn tương tác giữa MQTT client (sử dụng MQTT.fx) với IoT Hub qua giao thức MQTT

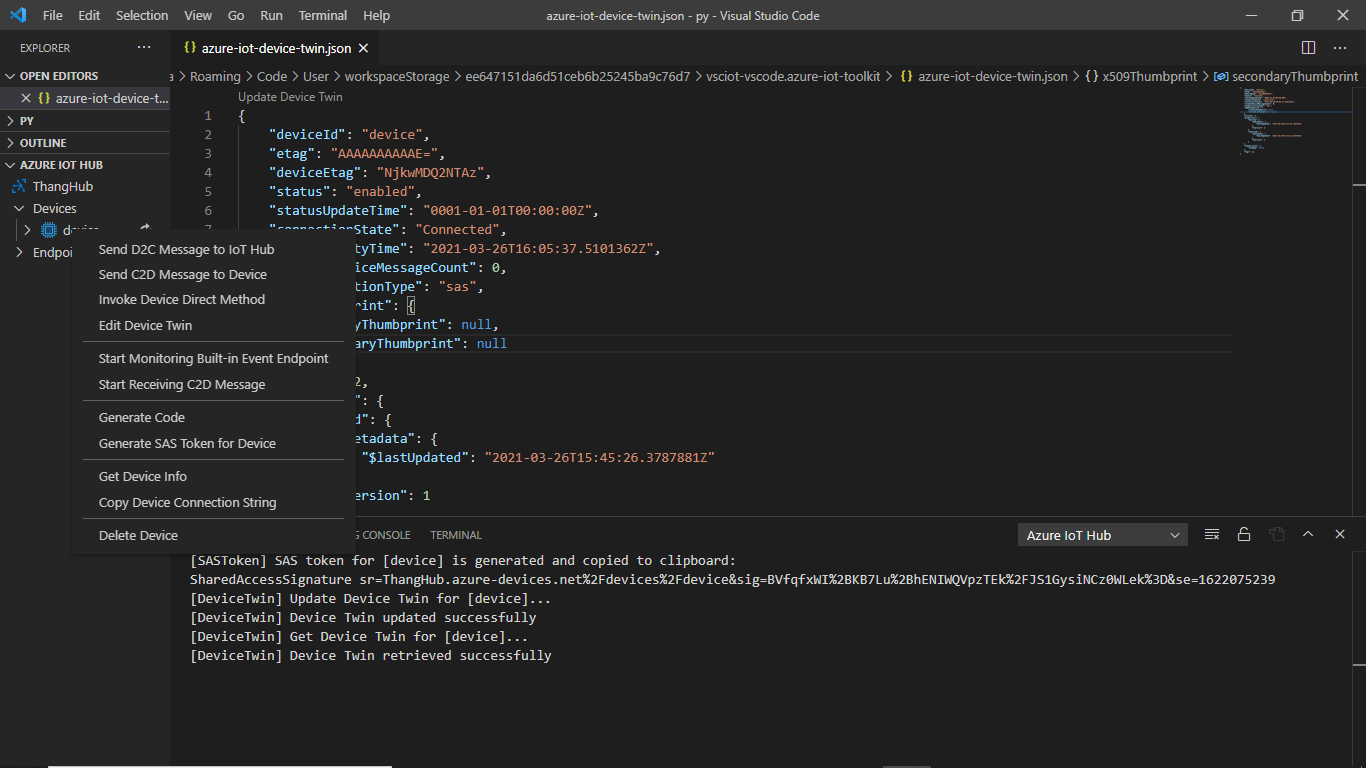
**TƯƠNG TÁC GIỮA CÁC THIẾT BỊ VỚI IOT HUB**

Qua bài viết **phần 2** và **phần 3** chúng ta đã tìm hiểu về kiến trúc cơ bản của IoT hub gồm: Device Endpoint và Device Twin và cái topic đi kèm. Bài viết này mình sẽ thực hành gửi, nhận dữ liệu, tương tác giữa MQTT client và IoT hub. Trong đó MQTT.fx đóng vai trò là một MQTT client và Azure IoT tools đóng vai trò đại diện IoT hub Endpoint ( có thể tương tác với thiết bị)

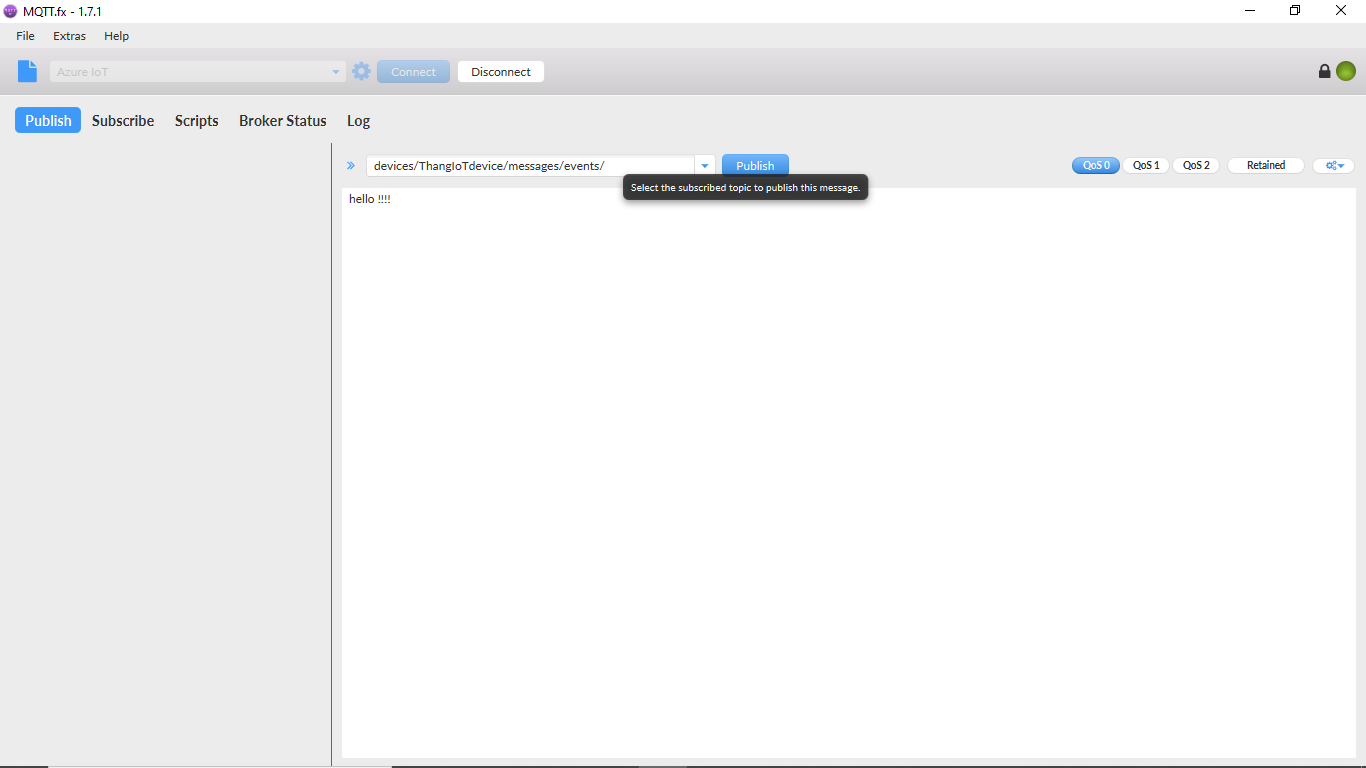
1. Thực hành tương tác với Endpoint

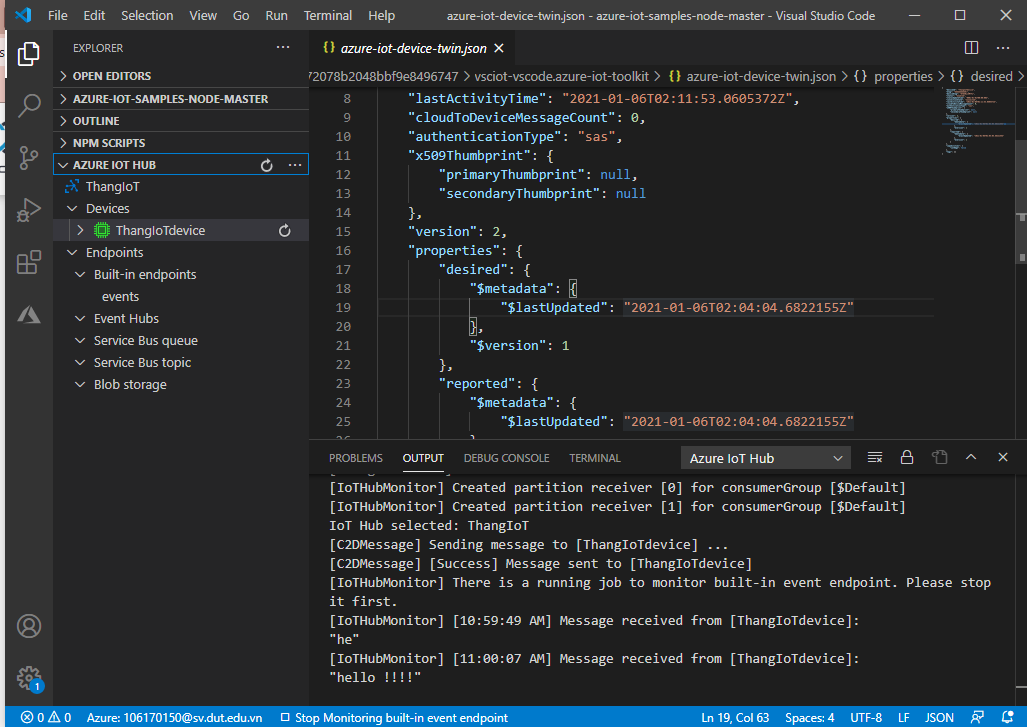
Iot hub có tích hợp sẵn 1 endpoint ( built-in endpoint) mà các dịch vụ back-end có thể sử dụng để đọc dữ liệu được. Khi gói tin được gửi từ thiết bị Iot hub của bạn thì mặc định được chuyển đến built-in endpoint này. Endpoint này giao tiếp với thiết bị qua các giao thức như MQTT, AMQP và HTTPS.

Tại Azure IoT tools ta nhấn chuột phải vào thiết bị và chọn Start Monitoring Built-in Event Endpoint để có thể tương tác với thiết bị

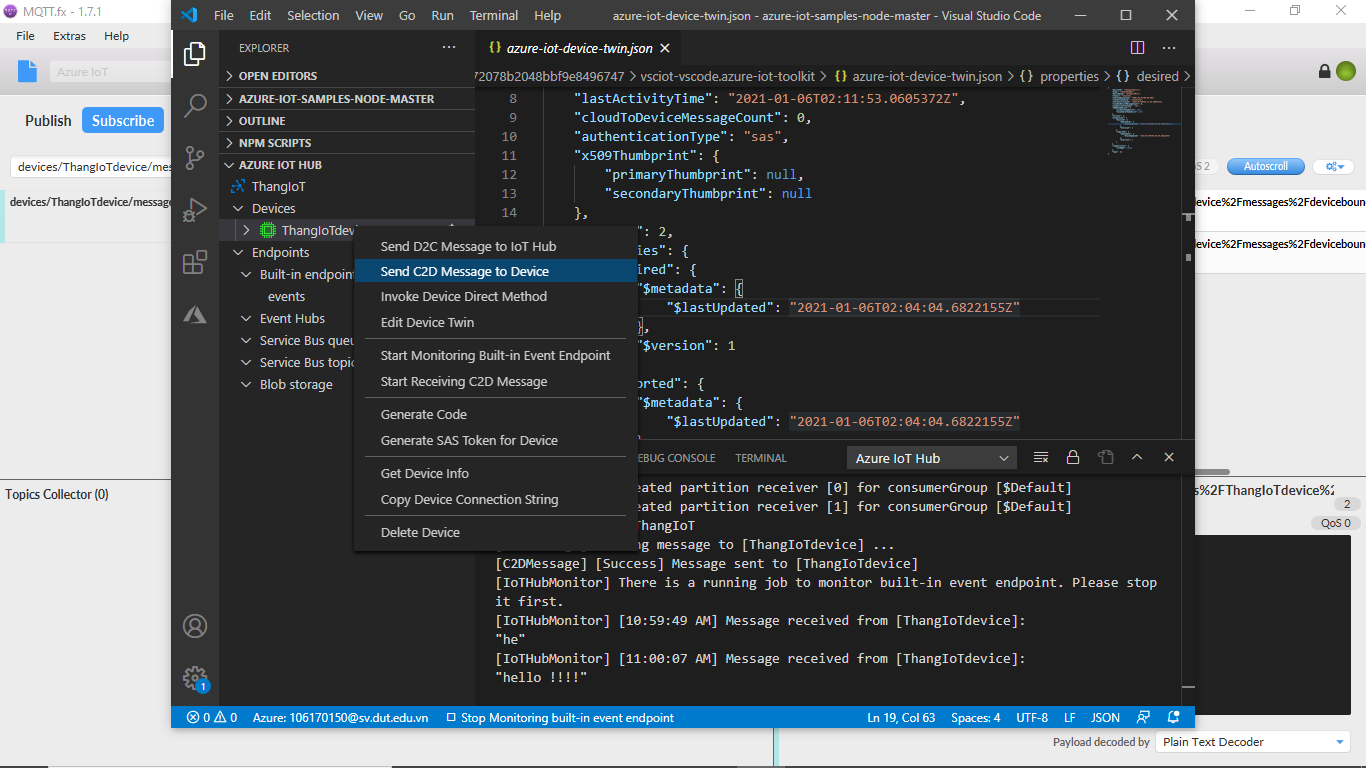


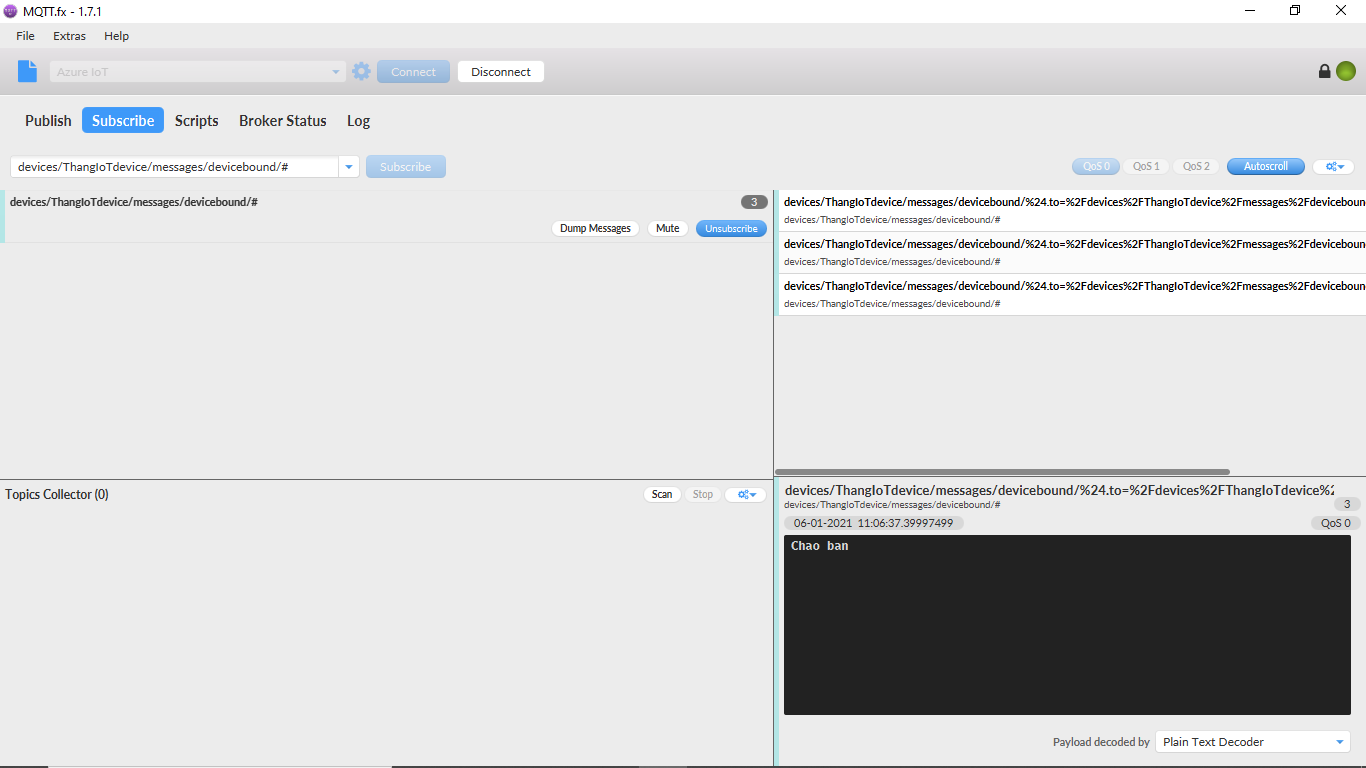
Client publish vào topic devices/{device\_id}/messages/events/ để gửi dữ liệu từ thiết bị đến Endpoint





Client subscribe topic: devices/{device\_id}/messages/devicebound/# để nhận dữ liệu từ IoT hub gửi về thiết bị





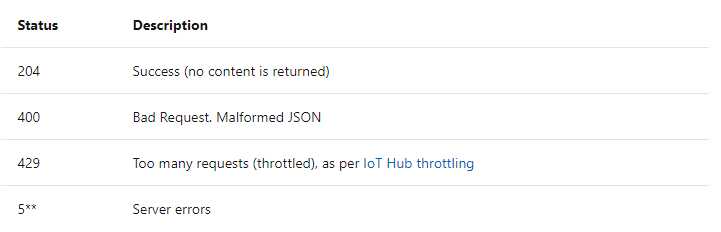
1. Thực hành tương tác với Device Twin

**Cập nhật dữ liệu qua trường Reported** :

Device Twin cho phép mình cập nhật dữ liệu qua trường Reported

* Trước tiên, client phải subcribe topic $iothub/twin/res/# để nhận phản hồi từ IoT hub.
* Client gửi gói tin theo định dạng Json vào topic $iothub/twin/PATCH/properties/reported/?$rid={request id} để cập nhật dữ liệu
* IoT hub gửi gói tin phản hồi về client: $iothub/twin/res/{status}/?$rid={request id}

Trong đó {status} là mã thông báo



Nếu {status} là 204 có nghĩa đã cập nhật dữ liệu thành công

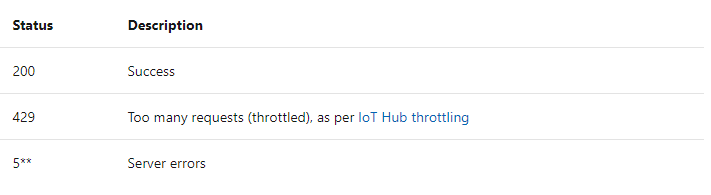




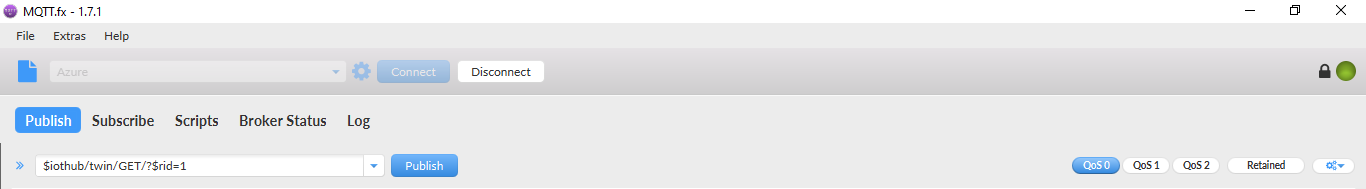
**Truy xuất dữ liệu:**

Đầu tiên, client subscribes đến topic $iothub/twin/res/# để nhận được phản hồi. Tiếp theo chúng ta gửi một gói tin trống đến **topic $iothub/twin/GET/?$rid= {request id}.** Sau đó IoT hub sẽ gửi một gói tin phản hồi có chứa dữ liệu trong Device Twin trong topic $iothub/twin/res/{status}/?$rid= {request id}

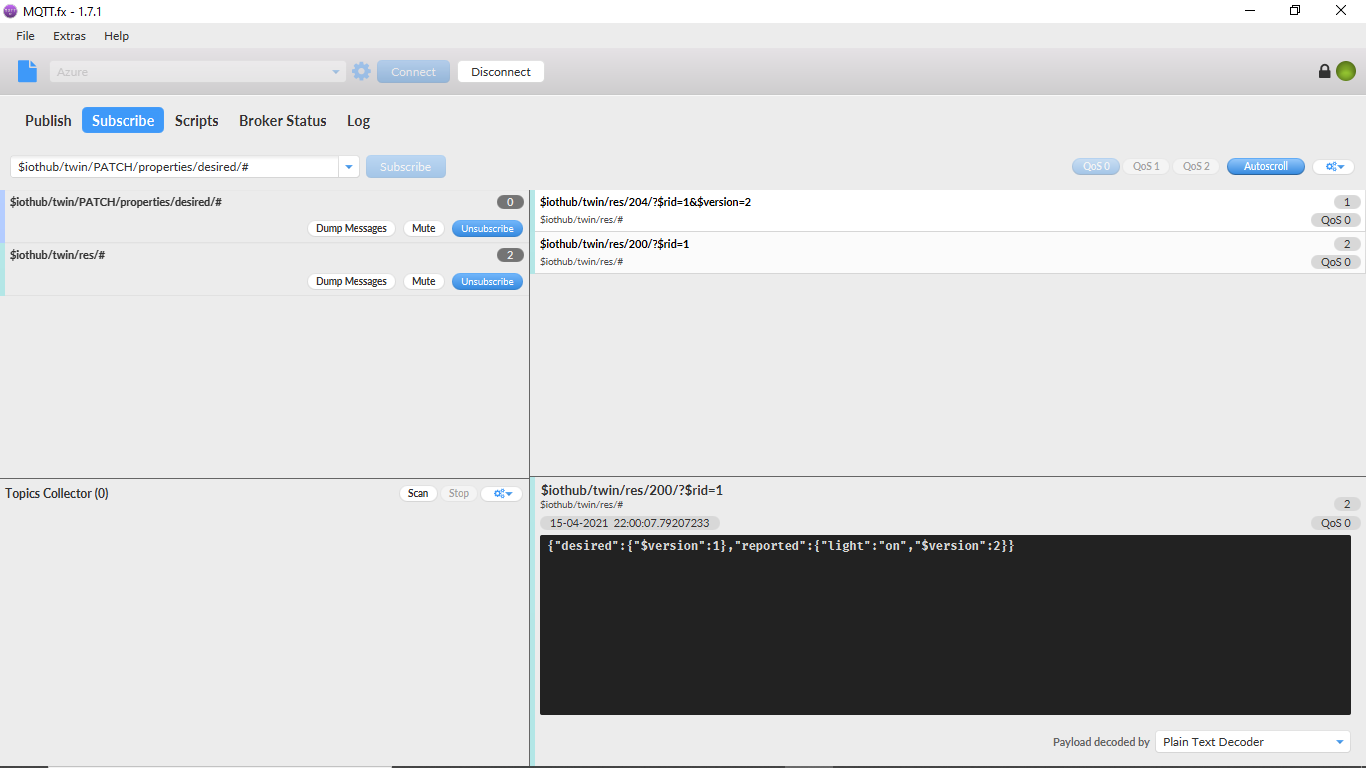
Trong đó {status} là mã thông báo



Nếu {status} là 200 thì chúng ta đã truy xuất thành công



Publish gói tin trống



Nhận dữ liệu

**Nhận thông báo khi cập nhật trường Desired:**

Khi client được kết nối, IoT hub sẽ gửi thông báo qua topic $iothub/twin/PATCH/properties/desired/?$version={new version}

1. Kết nối ESP8266 và IoT hub sử dụng giao thức MQTT

* Cài đặt thư viện:

Sử dụng hai thư viện chính là:

* + ESP8266WiFi
  + PubSubClient

Để ESP8266 kết nối với IoT hub chúng ta sử dụng thư viện ESP8266WiFi

Để ESP8266 có thể tương tác với IoT hub thông qua giao thức MQTT như một client thì chúng ta sẽ sử dụng thư viện có tên PubSubClient. Các bạn có thể tải thư viện ở đây.

Sau khi cài đặt thư viện, có một thay đổi nhỏ mà ta cần phải lưu ý nằm trong thư viện PubSubClient. Ở thư mục chứa thư viện PubSubClient, mở file PubSubClient.h ra sẽ tìm thấy định nghĩa:

**#define MQTT\_MAX\_PACKET\_SIZE 128**

Cấu hình này mặc định các dữ liệu trả về từ IoT hub thông qua các topic chỉ là 128Byte, kích thước của các topic từ IoT hub có thể lên tới 256KB nên nếu dữ liệu trả về lớn hơn 128Byte có thể làm mất dữ liệu. Để khắc phục vấn đề này chúng ta sẽ tăng giá trị của biến MQTT\_MAX\_PACK\_SIZE lên như sau:

**#define MQTT\_MAX\_PACKET\_SIZE 1024**

* **Viết chương trình kết nối ESP8266 đến IoT hub**
  + Cài đặt và kết nối Wifi
  + Khởi tạo MQTT Client để kết nối tới IoT hub:
    - Khởi tạo DEVICE\_ID, HOST NAME, Username/Password
    - Subscribe, publish các topic để truyền nhận dữ liệu

Về việc khởi tạo các thông số trên các bạn có thể xem lại **Phần 3**

Sau khi đã cài đặt xong thư viện cũng như lấy được các thông tin cần thiết, chúng ta sẽ viết chương trình kết nối cho ESP8266 theo đoạn chương trình mẫu dưới đây:

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// Set up wifi

const char\* ssid = "<YOUR\_SSID>";

const char\* password = "<YOUR\_WIFI\_PASSWORD>";

// Set up IoT hub

#define DEVICE\_ID "device"

const char MQTT\_HOST[] = "<Hostname>";

// ex: xxxxx.azure-devices.net

const int MQTT\_PORT = 8883;

const char MQTT\_USER[] = "<xxxx.azure-devices.net/<deviceId>/?api-version=2018-06-30>";

const char MQTT\_PASS[] = "<SharedAccessSignature sr={URL-encoded-resourceURI}&sig={signature-string}&se={expiry}>";

const char MQTT\_PUB\_TOPIC[] = "$iothub/twin/PATCH/properties/reported/?$rid=1";

const char MQTT\_SUB\_TOPIC[] = "$iothub/twin/res/#";

WiFiClientSecure espClient;

PubSubClient client(espClient);

void setup() {

Serial.begin(115200);

setup\_wifi();

espClient.setInsecure();

client.setServer(MQTT\_HOST, MQTT\_PORT);

client.setCallback(callback);

mqtt\_connect();

}

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void callback(char\* topic, byte\* payload, unsigned int length)

{

Serial.print("Received. topic=");

Serial.println(topic);

for (int i = 0; i < length; i++) {

Serial.print((char)payload[i]);

}

Serial.print("\n");

}

void mqtt\_connect()

{

while (!client.connected()) {

Serial.print("MQTT connecting ... ");

if (client.connect(DEVICE\_ID, MQTT\_USER, MQTT\_PASS)) {

Serial.println("connected.");

client.subscribe(MQTT\_SUB\_TOPIC);

client.publish(MQTT\_PUB\_TOPIC, "");

} else {

Serial.print("failed, status code =");

Serial.print(client.state());

Serial.println(". Try again in 5 seconds.");

Serial.print(DEVICE\_ID);

/\* Wait 5 seconds before retrying \*/

delay(5000);

}

}

}

//----------------------------------------------------

long messageSentAt = 0;

int dummyValue = 0;

char pubMessage[128];

void mqttLoop() {

if (!client.connected())

{

mqtt\_connect();

}

client.loop();

unsigned long now = millis();

if (now - messageSentAt > 10000)

{

messageSentAt = now;

sprintf(pubMessage, "{\"count\":\"%d\"}", dummyValue++);

Serial.print("Publishing message to topic ");

Serial.println(MQTT\_PUB\_TOPIC);

Serial.println(pubMessage);

client.publish(MQTT\_PUB\_TOPIC, pubMessage);

Serial.println("Published.");

}

}

void loop()

{

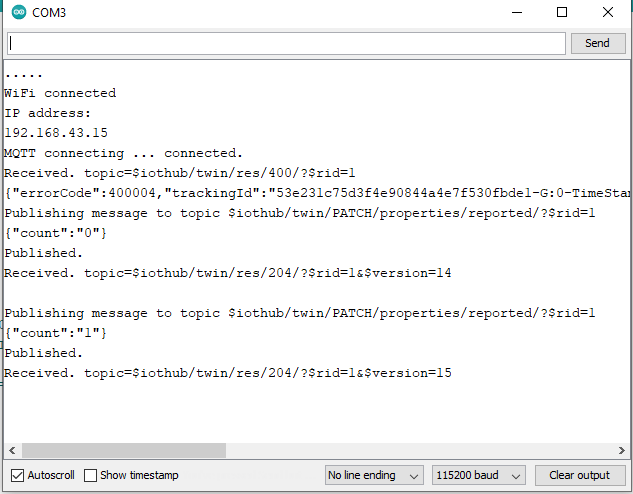
mqttLoop();

}

Giải thích chương trình: Sau mỗi 10 giây, ESP8266 sẽ publish một chuỗi có định dạng JSON tới trường desired của topic **$iothub/twin/PATCH/properties/reported/?$rid=1**

Đồng thời chúng ta cũng sẽ subscribe **topic $iothub/twin/res/#** để có thể nhận được thông báo chúng ta đã gửi gói tin thành công hay chưa.

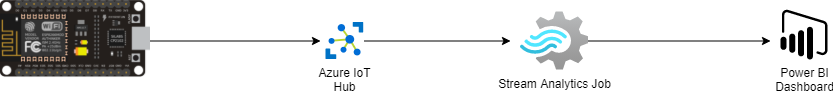
Sau khi nạp code thành công chúng ta mở Serial Monitor trong Arduino IDE ra để quan sát (lưu ý cài đặt mức baudrate là 115200). Dữ liệu nhận được sẽ như sau:



Như vậy qua phần này hy vọng các bạn hiểu thêm được về IoT hub endpoint thông qua tương tác giữa chúng với thiết bị. Mình mong qua chuỗi bài hướng dẫn “Xây dựng hệ thống Internet of Things sử dụng nền tảng Mircrosoft Azure IoT hub” giúp các bạn có cái nhìn tổng quan hơn về IoT platform cũng như có được kiến thức căn bản nhất để xây dựng các ứng dụng IoTs kết hợp với nền tảng Microsoft Azure cho riêng mình.

**Trực quan hóa dữ liệu cảm biến theo thời gian thực từ Azure IoT Hub**

Sơ đồ:



Mình sử dụng Microsoft Power BI để trực quan hóa dữ liệu cảm biến thời gian thực mà Azure IoT Hub nhận được. Để làm như vậy, mình cấu hình một Azure Stream Analytics để sử dụng dữ liệu từ Azure IoT Hub và định tuyến dữ liệu đó đến tập dữ liệu trong Power BI.

Microsoft Power BI là một công cụ trực quan hóa dữ liệu mà bạn có thể sử dụng để thực hiện các hoạt động tự phục vụ và kinh doanh doanh nghiệp (BI) trên các tập dữ liệu lớn. Azure Stream Analytics là một dịch vụ phân tích thời gian thực, được quản lý hoàn toàn, được thiết kế để giúp bạn phân tích và xử lý các luồng dữ liệu chuyển động nhanh có thể được sử dụng để nhận thông tin chi tiết, tạo báo cáo hoặc kích hoạt cảnh báo và hành động.

Trong chương này, mình sẽ hướng dẫn thực hiện các tác vụ sau:

* Tạo một consumer group trong IoT Hub
* Tạo và định cấu hình công việc Azure Stream Analytics để đọc cảm biến nhiệt độ từ comsumer group của bạn và gửi đến Power BI.
* Tạo báo cáo về dữ liệu nhiệt độ trong Power BI.

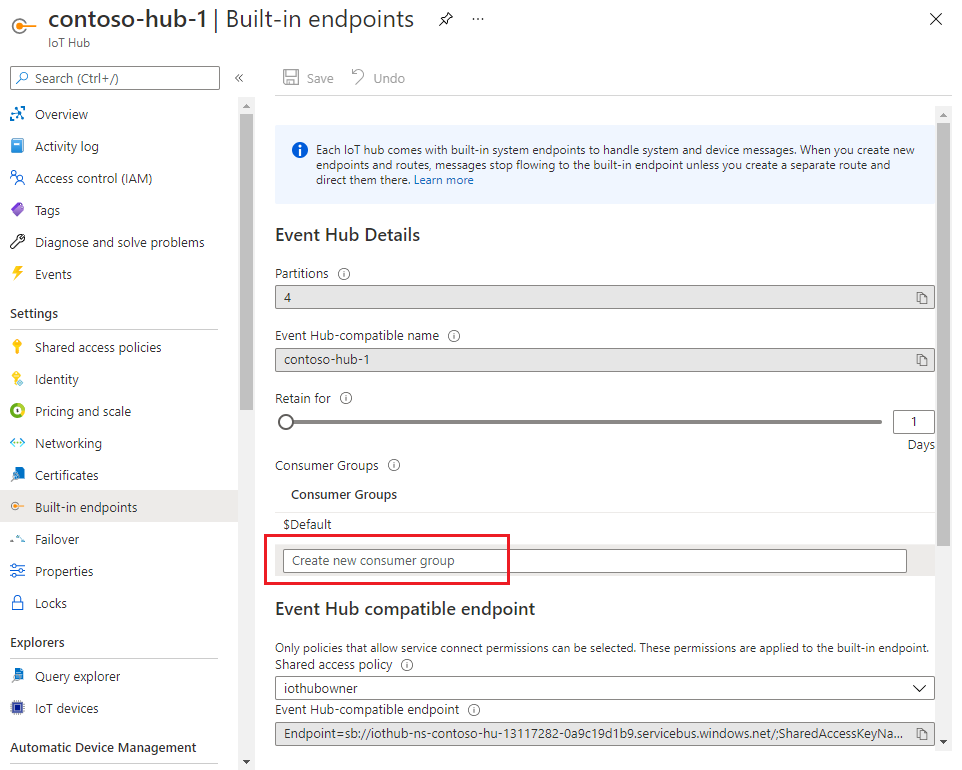
**Tạo một consumer group trong IoT Hub:**

**Consumer group** cung cấp các chế độ xem độc lập vào luồng sự kiện cho phép các ứng dụng và dịch vụ Azure sử dụng dữ liệu một cách độc lập từ cùng một Event Hub endpoint.

Trong phần này, bạn thêm một comsumer group vào điểm cuối tích hợp sẵn của IoT Hub của bạn, được sử dụng trong hướng dẫn này để lấy dữ liệu từ điểm cuối.

Để tạo một consumer group trong IoT hub làm theo các bước sau:

* Trong Azure portal mở IoT hub của bạn
* Ở phần bên trái chọn **Built-in endpoints**, nhập tên Consumer group mà bạn muốn tạo vào ô bên dưới



**Tạo, cấu hình và chạy một Stream Analytics job**:

Bắt đầu từ việc tạo Stream Analytics job, sau đó bạn định nghĩa inputs, outputs và truy vấn để xuất dữ liệu.

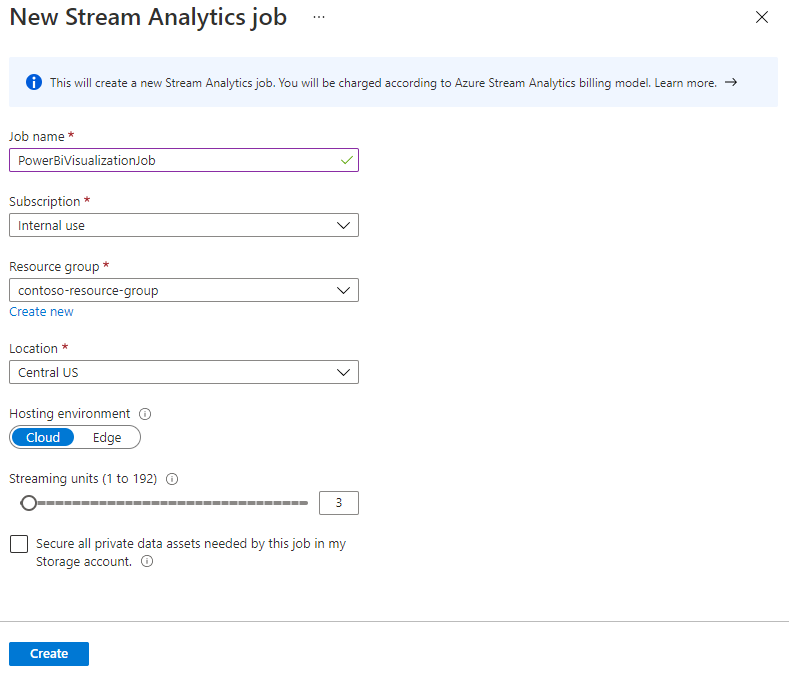
**Tạo Stream Analytics job:**

* Ở Azure portal, chọn Create a resource, tìm kiếm Stream Analytics Job tại ô tìm kiếm và chọn Create.
* Nhập thông tin cho job:

**Job name:** tên của job. Tên này là duy nhất trên toàn cầu.

**Resource Group:** sử dụng cùng Resource group mà IoT hub của bạn sử dụng.

**Location:** sử dụng cùng vị trí mà Resource group của bạn.

- 

* Chọn Create

**Thêm input vào Stream Analytics job:**

* Mở Stream Analytics job
* Phía dưới Job topology, chọn Inputs
* Trong phần Inputs, chọn Add stream input, sau đó chọn IoT Hub của bạn. Một bảng thông tin xuất hiện bao gồm:

**Input alias:** Bí danh duy nhất cho đầu vào (input)

**Select IoT Hub from your subscription**

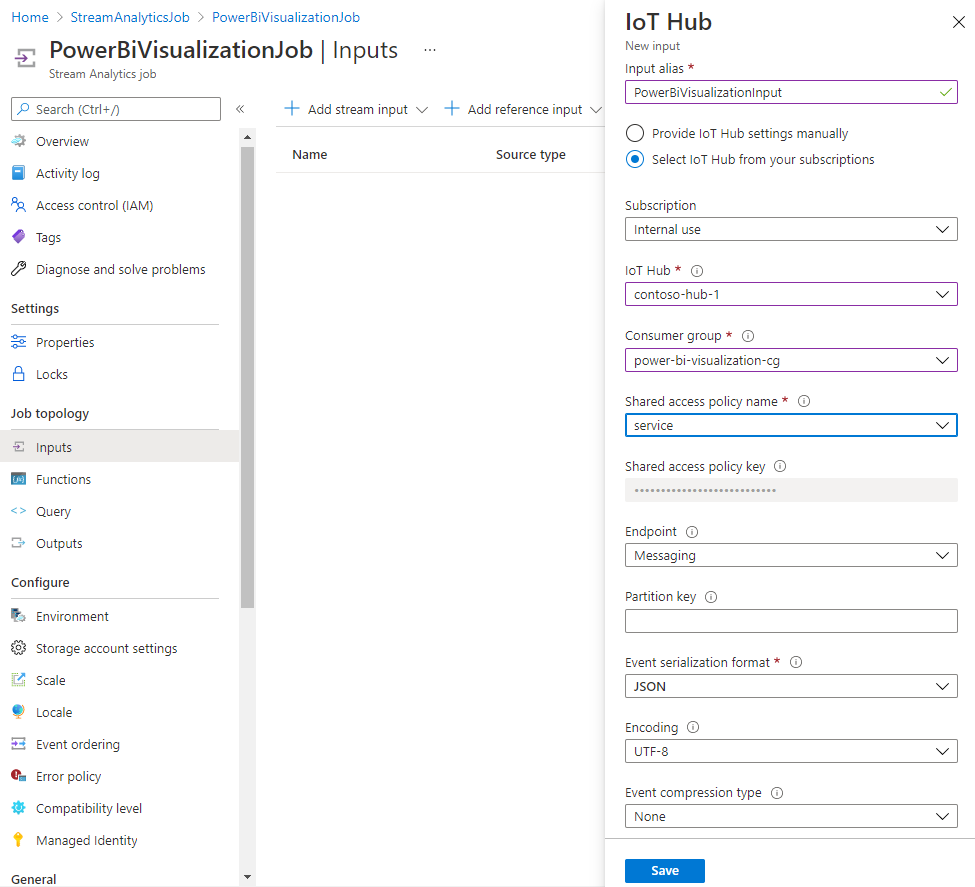
**Subscription:** Chọn subscription của bạn

**IoT hub:** Chọn IoT hub mà bạn muốn truyền dữ liệu vào

**Endpoint:** Chọn Massaging

**Shared access policy name:** Chọn tên chính sách truy cập mà bạn muốn Stream Analytics job sử dụng. Ở đây mình sử dụng **service**

**Consumer group:** Chọn Consumer group mà bạn đã tạo ở phía trên.

****

* Chọn Save

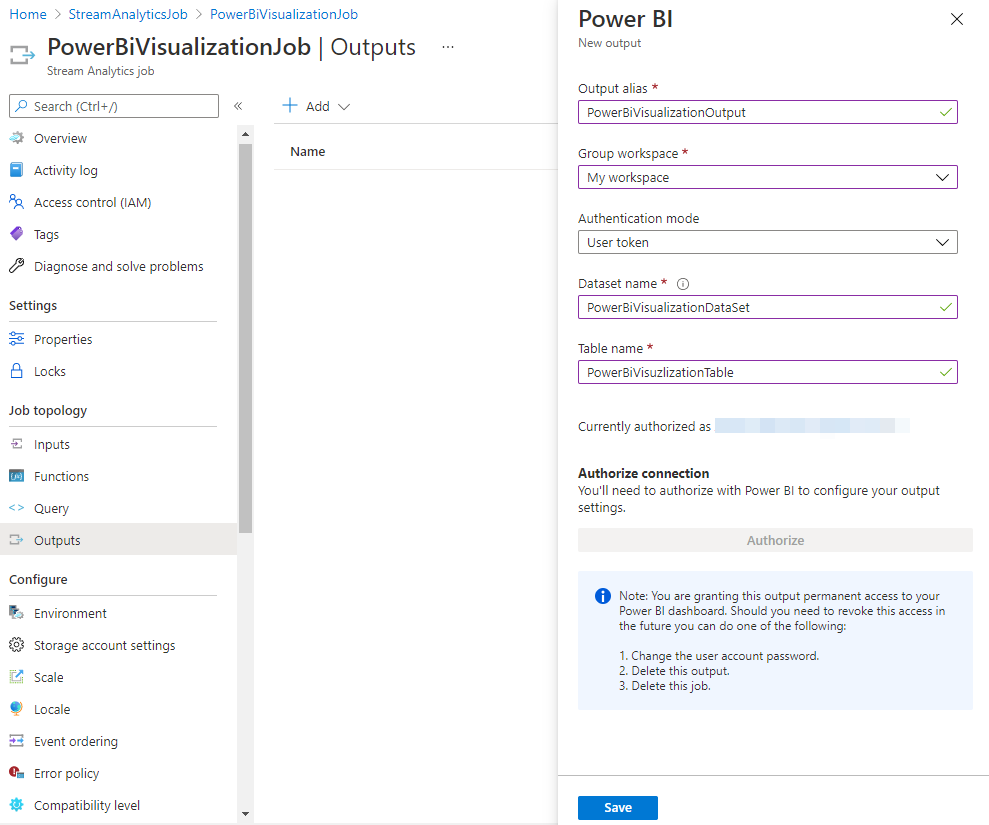
**Thêm output vào Stream Analytics job:**

* Dưới phần Job topology, chọn Outputs

Trong bảng Outputs, chọn Add, và sau đó chọn Power BI.

Và điền các thông tin sau cho Output:

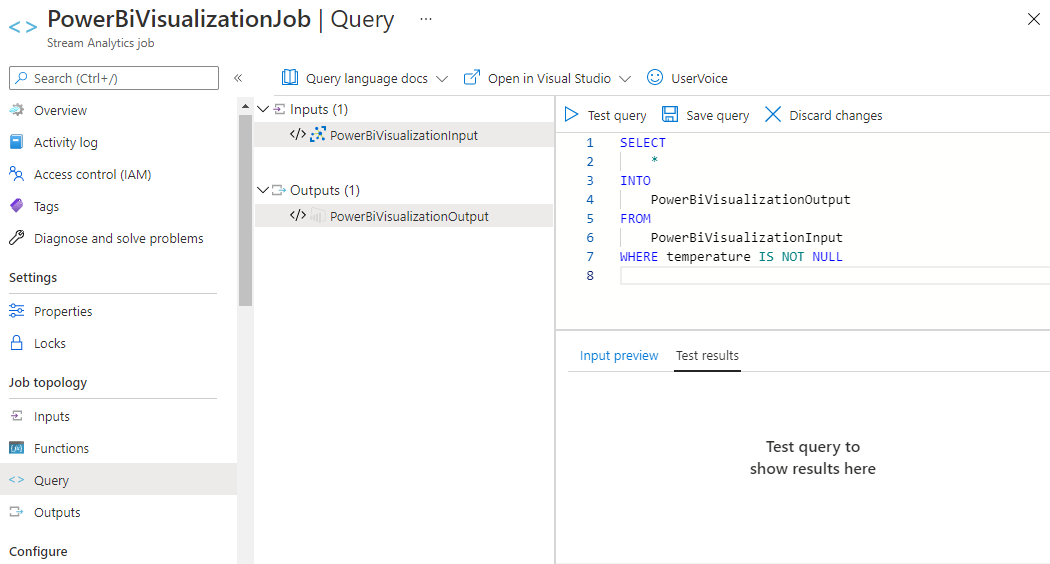
* **Output alias:** Bí danh duy nhất cho đầu ra (output)
* **Group workspace:** Chọn workspace
* **Dataset name:** Nhập trên Dataset
* **Table name:** Nhật tên bảng



* Chọn Save

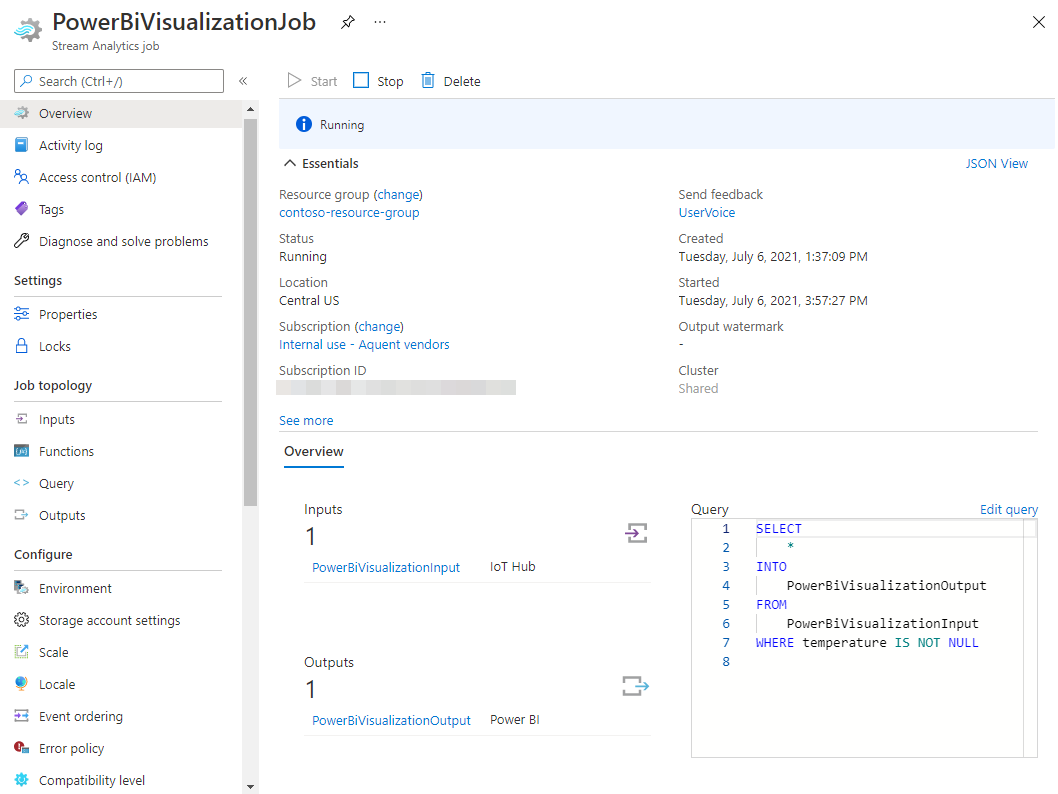
**Cấu hình truy vấn của Stream Analytics job:**

* Dưới phần Job topology, chọn Query
* Thay [YourInputAlias] bởi input alias
* Thay [YourOutputAlias] bởi output alias



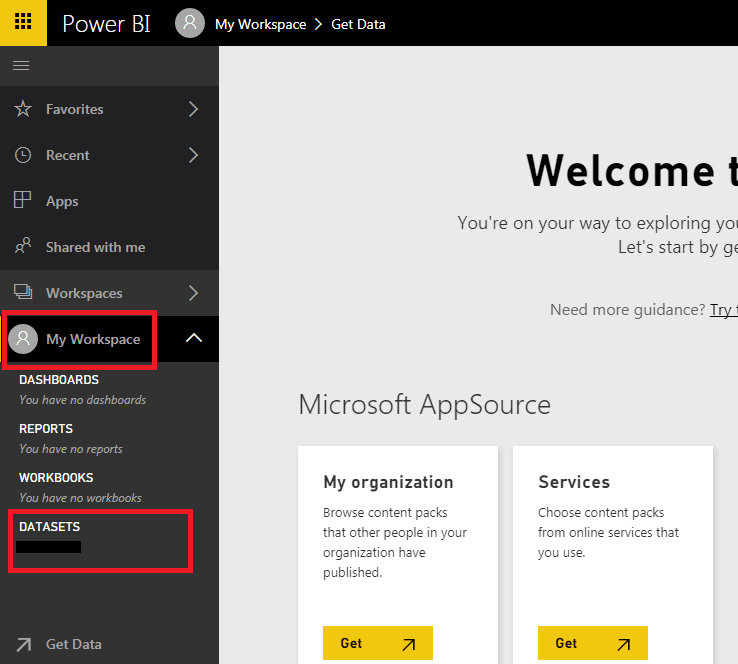
**Chạy Stream Analytics job:**

Trong Stream Analytics job, chọn Overview, sau đó chọn Start > Now > Start. Khi công việc bắt đầu thành công, trạng thái công việc sẽ chuyển từ Stopped sang Running.



**Tạo và xuất bản báo cáo Power BI để trực quan hóa dữ liệu:**

* Kết nối thiết bị của bạn để gửi dữ liệu đến Trung tâm IoT của bạn
* Đăng nhập vào tài khoản Power BI.
* Chuyển đến Group Workspace mà bạn đã chọn khi tạo đầu ra cho Stream Analytics Job. Trong trường hợp của chúng tôi, đó là Không gian làm việc của tôi.



* Chọn dữ liệu và biểu diễn

