



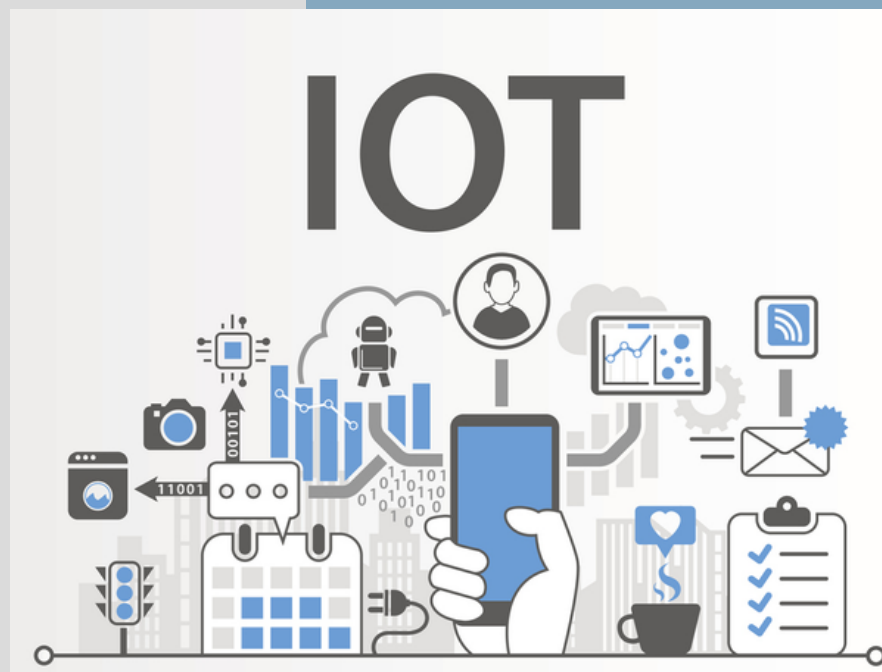
ALEXANDRE B BARRETO

MICROSOFT AZURE HANDS-ONS

IOT FOUNDATIONS

AUTHOR

Alexandre B Barreto
barretoabb@tec.mx



GOAL

This exercise provides a practical example of implementing an IoT architecture using Azure cloud components. During this exercise, the students will learn how to use IoT Hub to manage a set of IoT devices.

This tutorial is based on Veneri, Giacomo, and Antonio Capasso. Hands-on industrial Internet of Things: create a robust industrial IoT infrastructure using industry 4.0. Packt Publishing Ltd, 2018.

AZURE IOT

GOAL

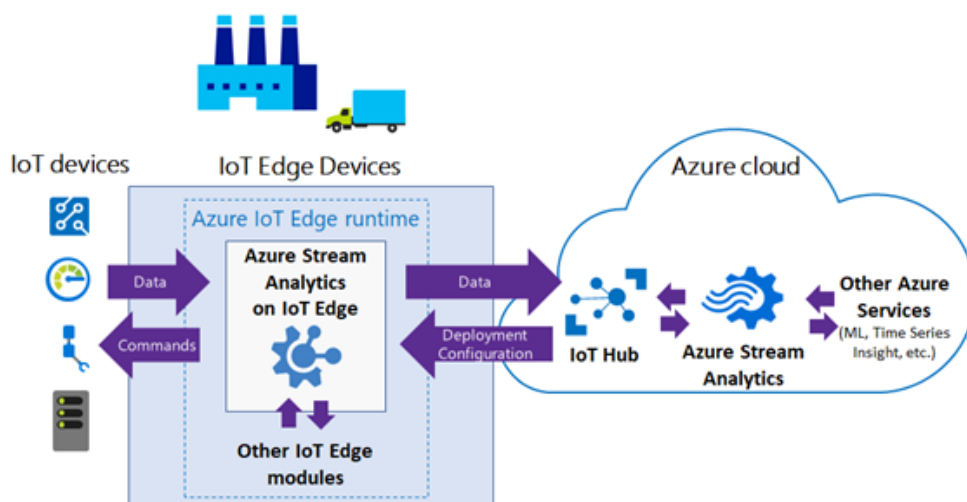
Internet of Things or IoT refers to a collection of managed and platform services across edge and cloud that connect, monitor and control billions of IoT assets. It also includes security and operating systems for devices and equipment and data and analytics that help businesses build, deploy, and manage IoT applications.

Azure IoT is a platform proposed by Microsoft to connect multiple devices, enable telemetry, store measures, run and develop analytics, and visualize results. The critical components of Azure IoT are the following:

- IoT Hub
- Stream Analytics
- Azure Data Lake
- Data Lake Analytics
- Time Series Insights

The general-purpose components of Azure are the following:

- Machine Learning (ML)
- Analytics Power BI



AZURE HUB



IoT Hub is a Platform-as-a-Services (PaaS) managed service hosted in the cloud. It acts as a central message hub for bi-directional communication between an IoT application and its devices. It supports the MQTTS, HTTPS, and AMQP protocols.

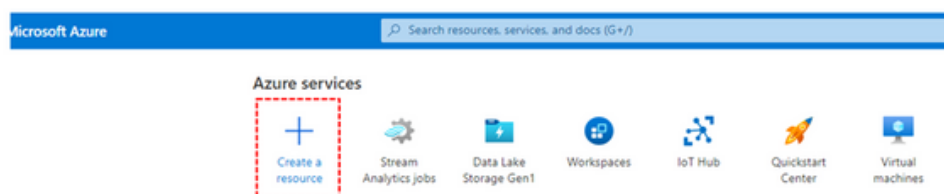
Azure IoT Hub enables IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution backend. IoT Hub supports multiple messaging patterns such as device-to-cloud telemetry, file upload from devices, and request-reply to methods to control your devices from the cloud. IoT Hub monitors and tracks events such as device creation, device failures, and device connections.

In summary, IoT Hub enables:

- Establish bi-directional communication with billions of IoT devices
- Enhance security with per-device authentication
- Provision devices at scale w/ IoT Hub Device Provisioning Service
- Manage devices at scale with device management

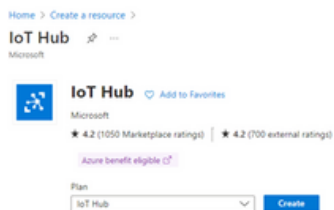
TASK 1 CREATE AN IOT HUB INSTANCE

Log in to Microsoft Azure^[1]: <https://portal.azure.com/>
Click on "Create a Resource".



[1] It is required that you did Activity 6.2.3. Fundamentos en Microsoft Azure.

Type "IoT Hub" and click enter. After clicking on the "Create" button.



Type the following information and ensure that you select the correct parameters. Next, click on the "Networking" button.

- **Subscription:** Azure for Students (any other selection will charge a monthly value for the use of the service).
- **Resource Group:** A resource group is a collection of resources that share the same lifecycle, permissions, and policies. Define a friendly name that you will use to save all components used in this activity.
- **IoT Hub Name:** select a friendly name for the IoT Hub component.
- **Region:** select one (remember that some region is not free charged).

This screenshot shows the "Project details" section of the IoT Hub creation wizard. It includes a breadcrumb trail "Home > Create a resource > IoT Hub" and tabs for "Basics", "Networking", "Management", "Add-ons", "Tags", and "Review + create". The "Basics" tab is active. The text says: "Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. Learn more". Under "Project details", it says: "Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources." There are two dropdown menus: "Subscription" (set to "Azure for Students") and "Resource group" (set to "(New) iot_6_2_4" with a "Create new" link below it). Under "Instance details", there are two more dropdowns: "IoT hub name" (set to "iot-hub-624" with a green checkmark) and "Region" (set to "East US").This screenshot shows the navigation buttons at the bottom of the IoT Hub creation wizard. There are three buttons: "Review + create" (blue), "< Previous" (grey), and "Next: Networking >" (grey). The "Next: Networking >" button is highlighted with a red dashed border.

In the Networking tab, ensure that you select **public access** and click the "**Management**" button.

Next, you need to configure the Management section.

- Pricing and Scale Tier[1]: **changing the default value in this option is essential. It would be best if you had to select the F1: Free Tier. The other option will be to charge you!**
- The other options keep the default.
- Finally, click on the "Review + create" button.


[1] To know more about sizing and tiers: <https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-scaling>.


IoT hub ...
Microsoft

Basics Networking **Management** Add-ons Tags Review + create

Each IoT hub is provisioned with a certain number of units in a specific tier. The tier and number of units determine the maximum daily quota of messages that you can send. [Learn more](#)

Scale tier and units

Pricing and scale tier * ⓘ F1: Free tier 
[Learn how to choose the right IoT hub tier for your solution](#)

Number of F1 IoT hub units ⓘ  1
Determines how your IoT hub can scale. You can change this later if your needs increase.

Pricing and scale tier ⓘ	F1	Device-to-cloud-messages ⓘ	Enabled
Messages per day ⓘ	8,000	Message routing ⓘ	Enabled
Cost per month	0.00 USD	Cloud-to-device commands ⓘ	Enabled
Defender for IoT ⓘ	Not eligible	IoT Edge ⓘ	Enabled
Device updates ⓘ	Eligible	Device management ⓘ	Enabled

Role-based access control

Change the permission model to Azure role-based access control (RBAC) only, or to a combination of shared access policies and RBAC. [Learn more](#)

☐ RBAC only

☒ Shared access policy + RBAC

[Review + create](#) [< Previous: Networking](#) [Next: Add-ons >](#)

Azure will validate your information to check for any misconfiguration. If everything is ok, a new screen will show. **You need to confirm that you do not be charged.** If everything is ok, click on the "**create**" button.

Allow public network access	Enabled
Management	
Pricing and scale tier	F1
Number of F1 IoT hub units	1
Messages per day	8,000
Device-to-cloud partitions	2
Cost per month	0.00 USD
Defender for IoT	Disabled

Now, Azure will create the IoT Hub component. This process can take a few minutes to be finished. The message "Deployment is in Progress" is shown to you during this process. A new message Your deployment is complete signalizes that the component is ready to be used.

✓ Your deployment is complete



Deployment name: `iot-hub-624-10410539`
Subscription: [Azure for Students](#)
Resource group: `iot_6_2_4`

Start time: 10/4/2022, 10:53:15 AM

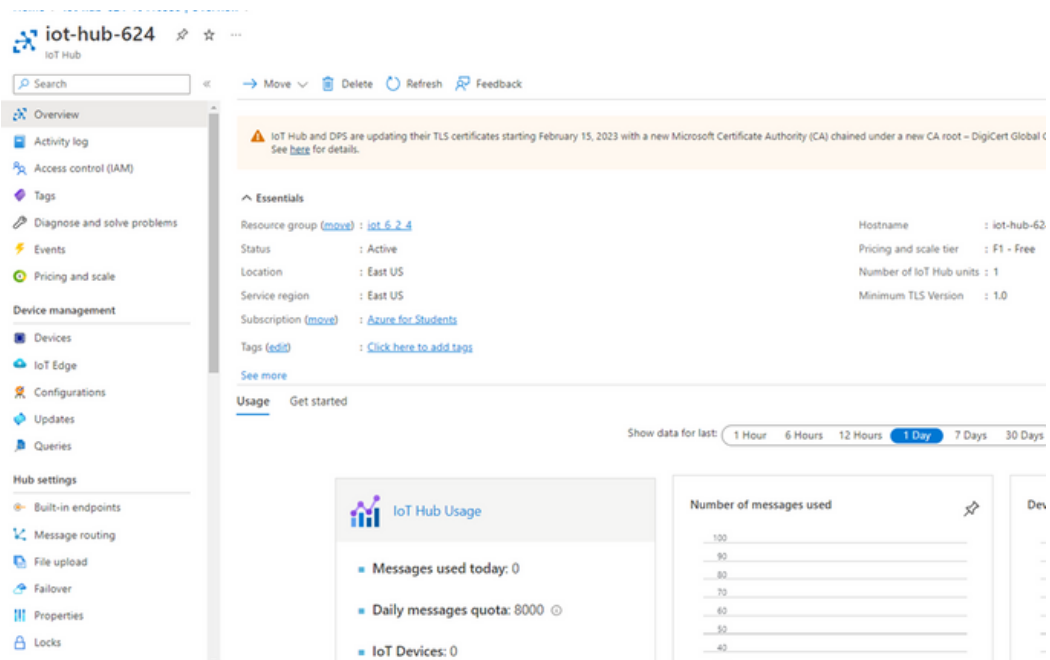
Correlation ID: `e131c9f3-1cb9-48e6-bdb0-f3ee3efe69b2`



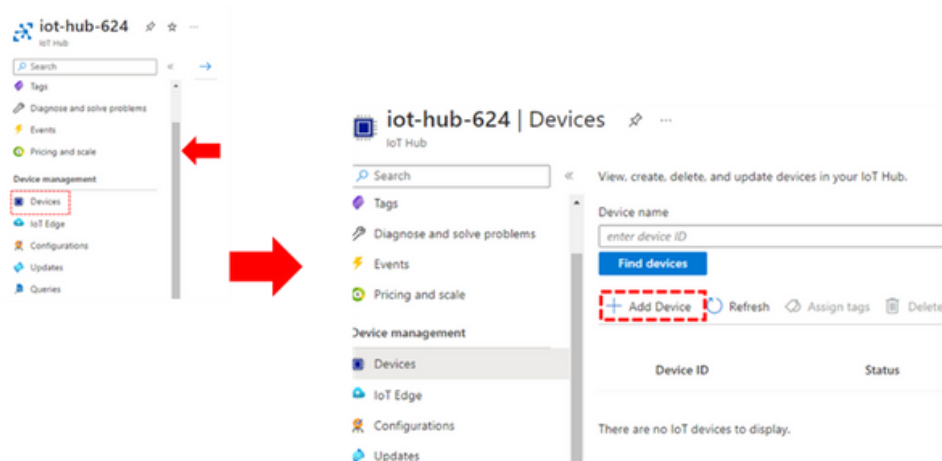
▼ Deployment details

TASK 2 REGISTERING A NEW DEVICE

The next step is registering a new device in the IoT Hub. To perform this task, click on the **"Go to the resource"** button to open the options related to the IoT Hub component. Now you can see your IoT Hub Dashboard (`iot-hub-624`). Using this Dashboard, you can have basic information about devices managed by the Hub, such as change policies, check events, etc.



To add a new component, use the scrollbar at the left until finding the **"Devices"** option. On the next screen, click on the **"Add Device"** button.



In the opened form, define a **device ID**, which is the Device identity used for device authentication and access control, and keep the other information with its default values.

[Home](#) > [iot-hub-624-10410539 | Overview](#) > [iot-hub-624 | Devices](#) >

Create a device ...

i Find Certified for Azure IoT devices in the Device Catalog

Device ID * ⓘ

iot-device-001

Authentication type ⓘ

Symmetric key X.509 Self-Signed X.509 CA Signed

Auto-generate keys ⓘ

☒

Connect this device to an IoT hub ⓘ

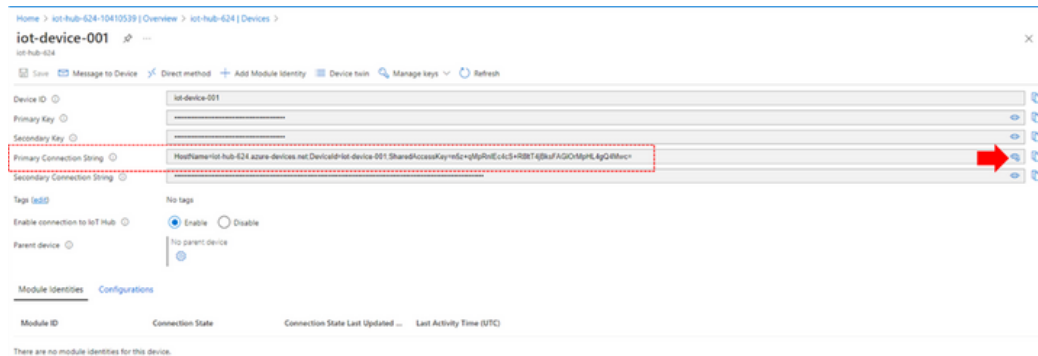
Enable Disable

Parent device ⓘ

No parent device

[Set a parent device](#)

Now, you can see it in the device's hub panel. Click on its ID and check the information provided. An essential piece of data is the Primary Connection String, which is based on the primary key used in API calls and allows the device to communicate with IoT Hub. You can check its value click on the view button.



Now, we have the configuration in the hub, which enables a device to send data to our IoT Cloud. The next step is to configure a physical device to send these data. In our example, you implement a simplistic device using JavaScript, emulating a temperature sensor.

To perform this task, download the PYCharm Community and install it on your computer.: <https://www.jetbrains.com/pycharm/download/>

After the installation, open the tools and create a folder named "iot-624" enter and install the node dependencies and azure-iot packages required for this activity.

- mkdir my-iiot-device
- cd my-iiot-device
- pip install azure-iot-device



On the folder defined, download the file named "my-iot-device.py", which is stored in the Git repository: <https://github.com/kabartsjc/iot-foundation-course/tree/main/iot-624-python> This code implements an IoT client that uses an MQTT protocol[1] to send the temperature data, emulated by our sensor, to the IoT Hub. The central part of this code is commented on below:

Line 9-11	Define a set of variables used in the code.
Line 15	It is the most important variable because it defines the connection string. In your code, replace the value for the one shown in the device previously created in the IoT Hub.
Line 18	Instantiate a client object.
Line 21	Connect the client to the IoT Hub
Line 23 – 35	The method simulates the temperature and sends it to the IoT Hub.
Line 37	Send the message to IoT Hub (async)

[1] You will study the MQTT protocol in future classes.

After you save the file in the recent folder, you need to run the code and see that it works. To run our code, press the following button.

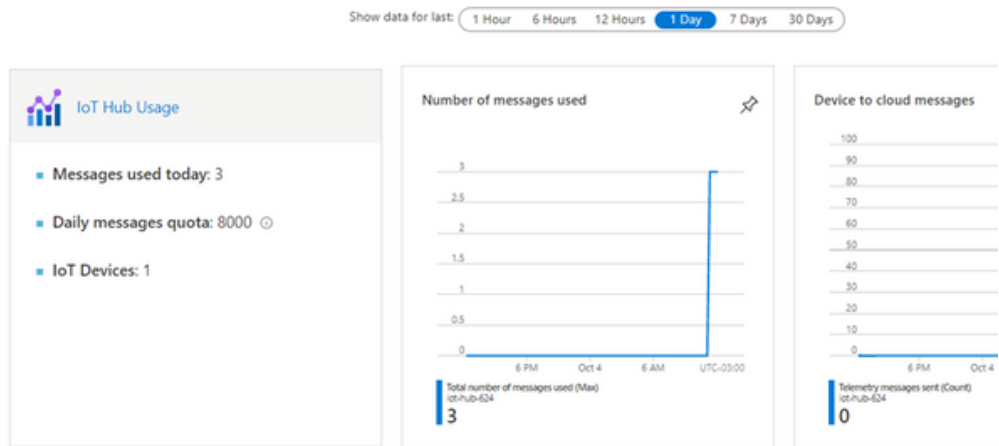


```

my-iot-device x
D:\git-repo\iot-foundation-course\iot-624-python\venv\Sc
sending message #1
sending message #2
sending message #3
sending message #4
sending message #5
sending message #6
sending message #7
sending message #8

```

As shown in the previous Figure, the code prints the log message, indicating that the code works, and sends the temperature every two seconds. Also, we need to check in the IoT Hub. To prevent this information, open the Dashboard and review the information in the IoT Hub Usage panel.



Also, you can check the Metrics panel of the IoT Hub, where you can set the parameter you want to analyze, as shown in the Figure below.

