

Exercise 3: Data transmission via MQTT

Learning outcome: To get familiar with Wireless Data Transmission via MQTT

Todo:

- Read BMP280 data via I2C Interface in real time
- Connect Raspberry Pi Pico with Wi-Fi
- Get familiar with MQTT Server on HiveMQ
- Publish the data to MQTT server

Implementation

1. Read BMP280 sensor data via I2C interface in real-time

- Follow the work done in Exercise 1 for that (*code available in Exercise 1 folder*)

2. Connect Raspberry Pi Pico with Wi-Fi

- **Write Script to Connect to Wi-Fi:**
 - Write a script to connect the Pico to your Wi-Fi network using the network module.
 - Configure the SSID and password for your Wi-Fi network.
- **Run the Script:**
 - Save and run the script on Pico to test and verify the connection.
 - Ensure Pico connects to the Wi-Fi network successfully by checking for an IP address assignment.

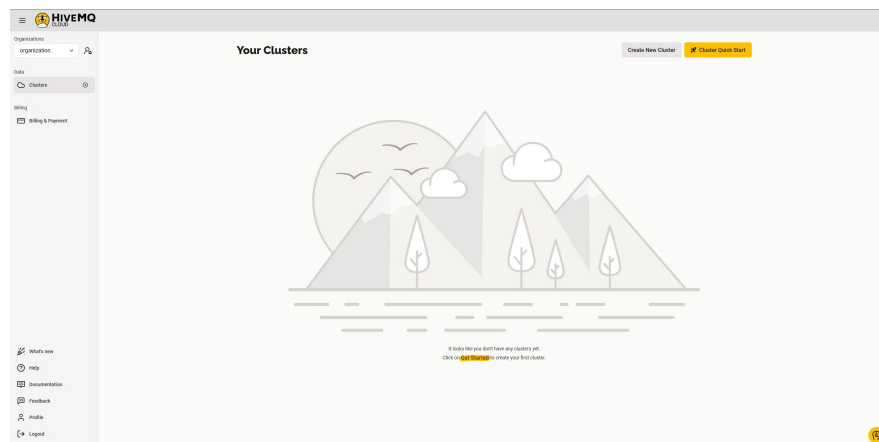
3. Create an MQTT server

For creating a MQTT sever to publish and subscribe to data via topics, we have multiple options available:

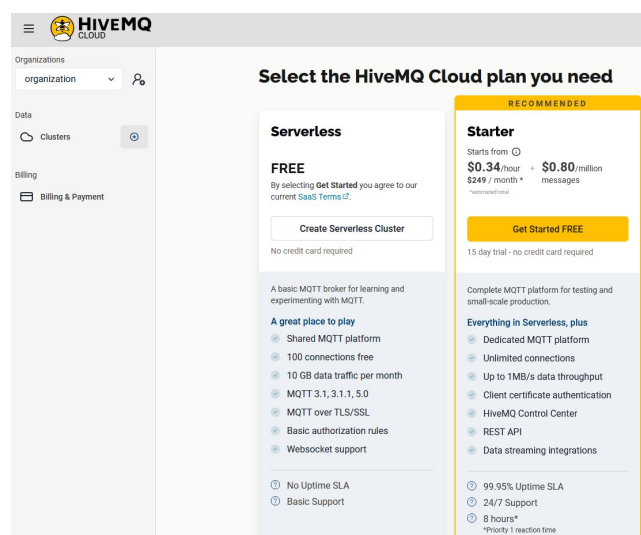
- Install mosquitto MQTT broker
- Utilize MQTT test client on AWS IOT Core
- **Create a MQTT Cloud sever (*Cloud provided by AWS anyway*) on HiveMQ**

For this exercise we will utilize MQTT cloud server created on HiveMQ.

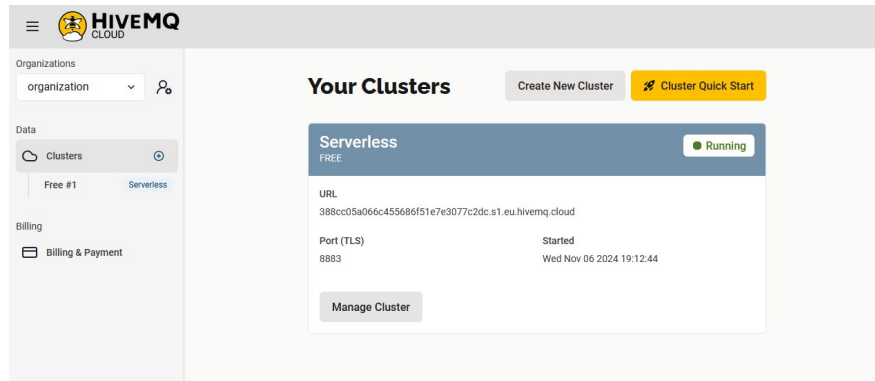
- **Create your account on HiveMQ**
 - Go to [HiveMQ website](#)
 - Start free > Sign up FREE NOW (*HiveMQ Cloud*)
 - Sign up
- **Login to your account**
 - You gonna see the dashboard after log in



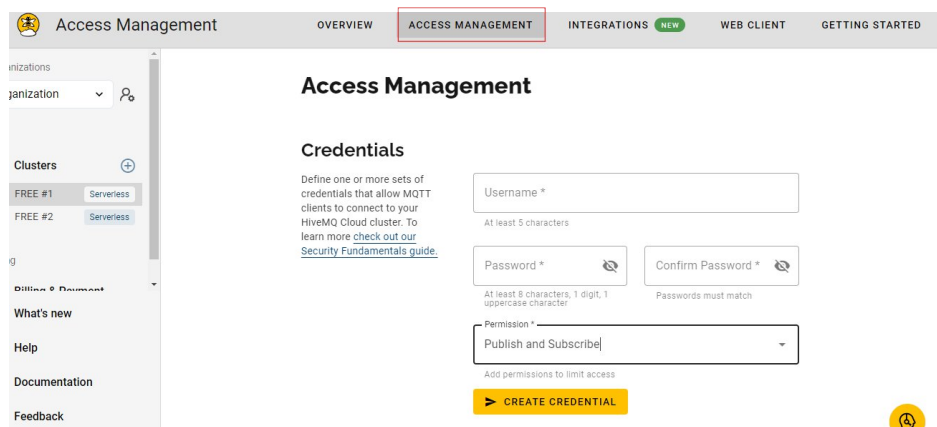
- **Create MQTT Cloud server**
 - Click the \oplus symbol near the *Clusters* on left panel
 - Choose the Free option



- The server created successfully, as shown below



- Click Manage Cluster, you will see the *cluster details*. The Cluster URL is the one you gonna use later.
- **Define/Create credentials access to MQTT Server**
 - In the Access Management tab, you need to specify the username and password for your cluster. These credential information will be utilized in the code later.
 - Remember to choose “Publish and Subscribe” in the *Permission*.



- Click Create Credential

4. Publish the data to MQTT server

- **Install required libraries:**

Install the following libraries on your Pico board for MQTT setup, **micropython-umqtt.robust**, **micropython-umqtt.simple**

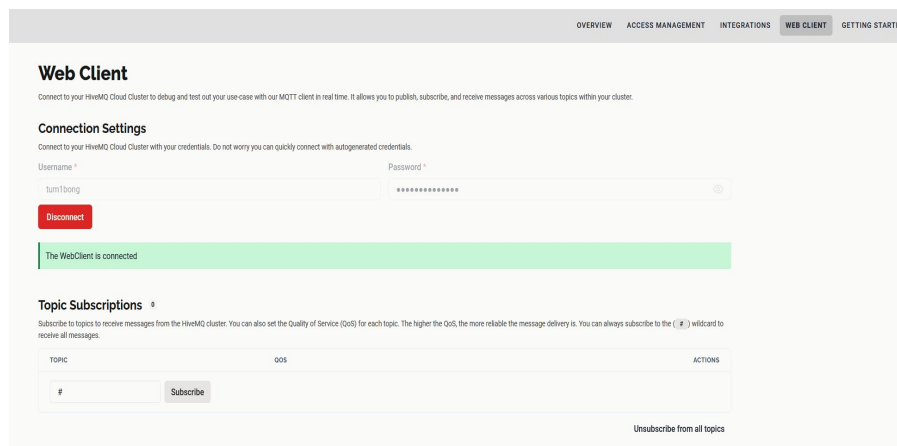
NOTE: If you face any issue with installing these libraries with the latest version of micropython installed on you pico board you can try installing them with the **v1.21.0 (2023-10-05) .uf2** version. The latest one has some bugs and it missing few libraries such as ussl,uzlib and more.

- **Write Script to Publish Data**

- Create a MQTT client with credentials of server/cluster URL, port, username, password, and SSL (Secure Socket Layer) configurations, etc., (**NOTE:** remember to import the **MQTTClient** from **umqtt.simple**).
- Connect to the MQTT Cloud server with **.connect()** function.
- Publish the sensor data to a specific topic at regular intervals with **.publish(topic, value)**

- **Verify if data being published**

- Go to Web Client tab
- Connect to your HiveMQ Cloud Cluster with your credentials (username, password)



- Click Subscribe
 - “#” means you subscribe to all topics
 - Or you just type the topic name you want to subscribe
- Now, just wait and see if the message is published and displayed

Topic Subscriptions ¹

Subscribe to topics to receive messages from the HiveMQ cluster. You can also set the Quality of Service (QoS) for each topic. The higher the QoS, the more reliable the message delivery is. You can always subscribe to the (#) wildcard to receive all messages.

TOPIC	QOS	ACTIONS
<div>h #</div>	QoS: 0	
<div><input type="text" value="Topic Name"/> <input type="button" value="Subscribe"/></div>		

[Unsubscribe from all topics](#)

Messages ⁶

Send and see messages that are published to the topics you are subscribed to. If you cannot see any messages, make sure you are subscribed to the correct topics. You can always subscribe to the (#) wildcard to receive all messages.

MESSAGE	TOPIC	QOS	ACTIONS
<div><input type="text" value="Your message"/></div>	<div><input type="text" value="Topic"/></div>	QoS: 0	<input type="button" value="Send Message"/>
25.37	<div>h tuml_picow/temp</div>	0	
102168.3	<div>h tuml_picow/pressure</div>	0	
102169.1	<div>h tuml_picow/pressure</div>	0	
25.38	<div>h tuml_picow/temp</div>	0	
102169.3	<div>h tuml_picow/pressure</div>	0	
25.39	<div>h tuml_picow/temp</div>	0	

In the next exercise session, we will utilize Node Red for connecting different node and servers for receiving and transmitting data. SOOOOO....STAY TUNED!!!!