## **10.4 Testing the MQTT Broker connection**

In Section 10.3, Figure 14, I configured the MQTT broker credentials within the Arduino codebase, enabling the Arduino to establish a connection with the broker. This setup allowed real-time monitoring of the sensor data from the ultrasonic sensor when tested on the vehicle to be published to the broker. I used **MQTT-Explorer** as the MQTT client to subscribe to the sensor data and monitor the published messages. (Thomas Nordquist 2025)

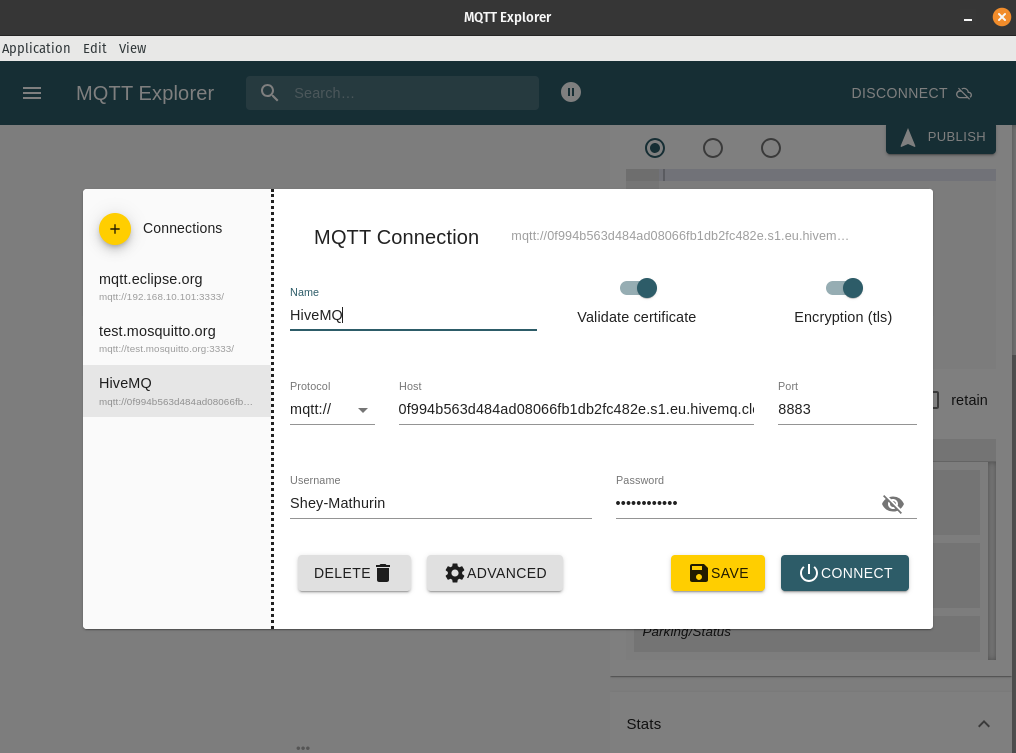


Figure 21

* This figure shows the MQTT client application MQTT-Explorer, by adding a ‘new connection’ located on the left side, the Hive MQTT Broker credentials are typed where specified, to monitor or subscribe to sensor data. In MQTT-Explorer, TLS (Transport Layer Security) must be enabled to ensure the connection between the client and the broker is secure encrypting any data transferred over the network. The Certificate validation checks if the server is trustworthy to verify and accept the digital certificate when logging in with the username and password as this is a serverless setup.



Figure 22

* When entering the MQTT-Explorer application, once logged in, the logs can be seen in the command line. As this is the Snap package version of MQTT-explorer, it acts as an application, so all logs can be seen in the command line as seen in the figure above. It shows the certificate being successfully validated and all the details inputted before specifying the MQTT protocol.

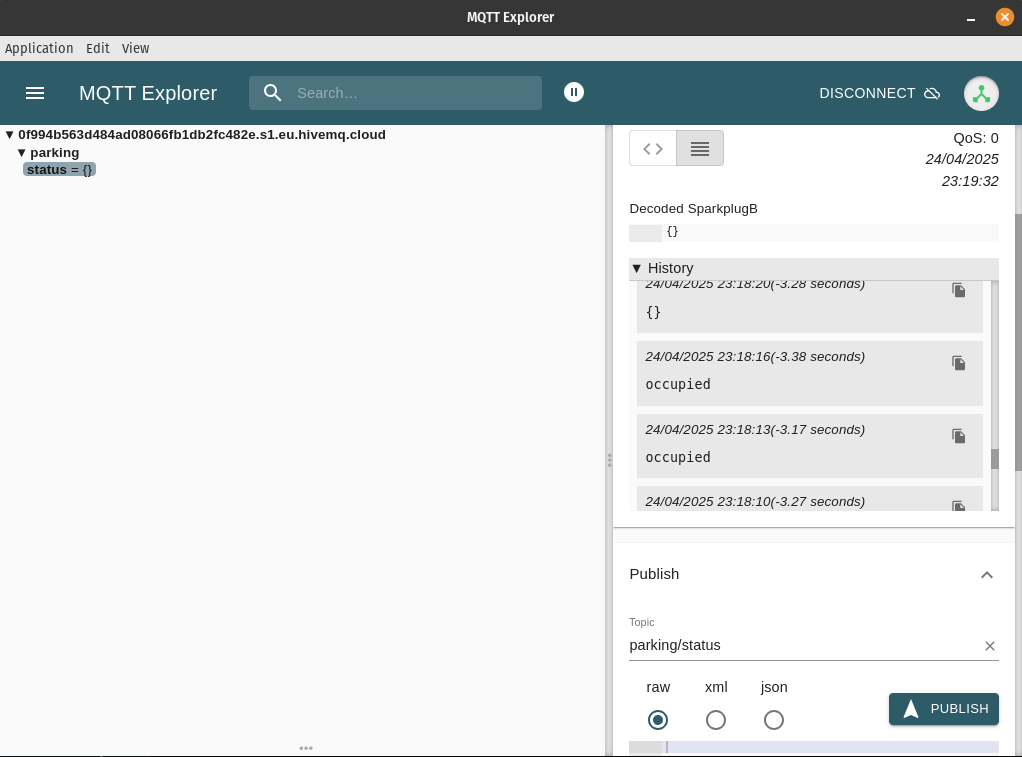


Figure 23

* For MQTT-Explorer to gain access to the sensor data it needs to be subscribed to a topic. In this case, it was ‘parking/status’ when this is inputted the status of whether a car is parked or not is not parked is stored in ‘parking’. When the vehicle is parked it will show ‘occupied’, when the vehicle is not present it will show ‘{}’ meaning the space is empty which is shown in the history logs in the figure above.