

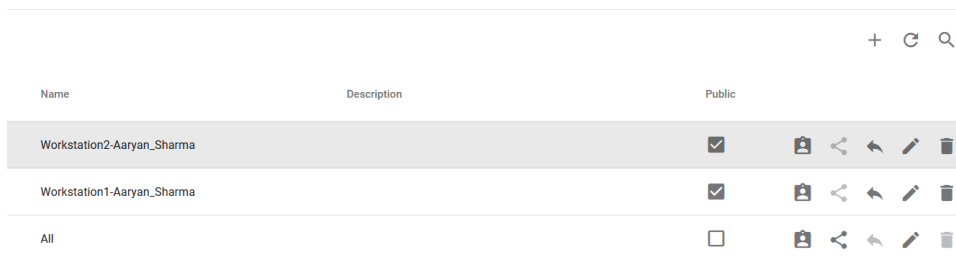
IOT Assignment 2

Name: Aryan Sharma

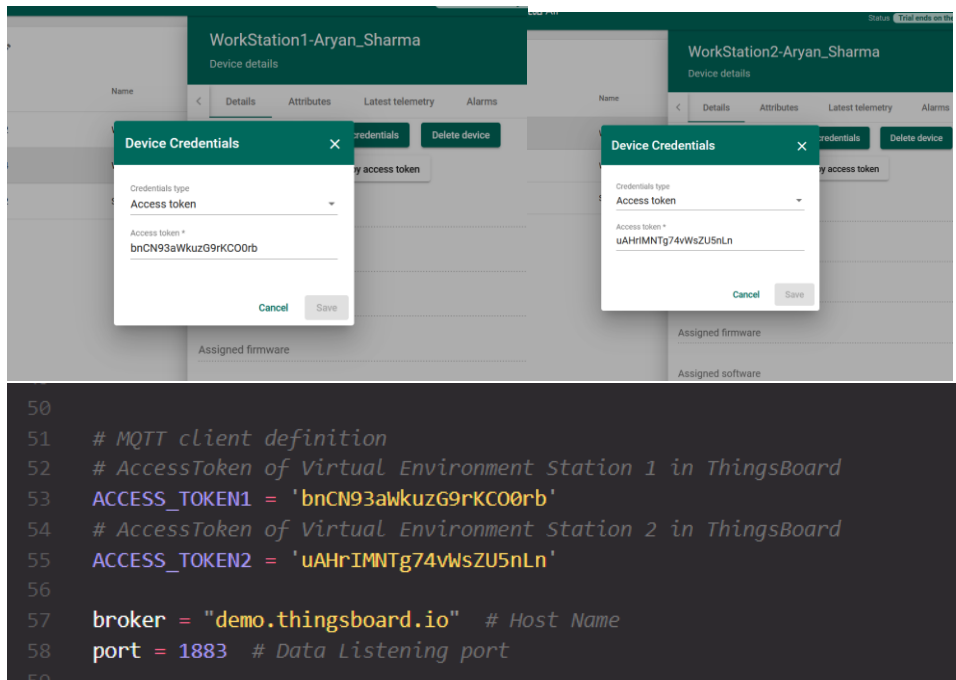
SUID: 401140505

A station gathers data (such temperature, rain height, etc.) from various sensors and publishes its status using a MQTT Broker*, hosted by the Cloud Platform (Thingsboard dashboard).

For you to see the dashboard, create two stations, each with a separate token and port that must be open and setup. Check that the port number, token 2, and broker where your dashboard will be hosted are operational and accessible. Compare the access token here with the token in the code.



Name	Description	Public
Workstation2-Aaryan_Sharma		<input checked="" type="checkbox"/>
Workstation1-Aaryan_Sharma		<input checked="" type="checkbox"/>
All		<input type="checkbox"/>



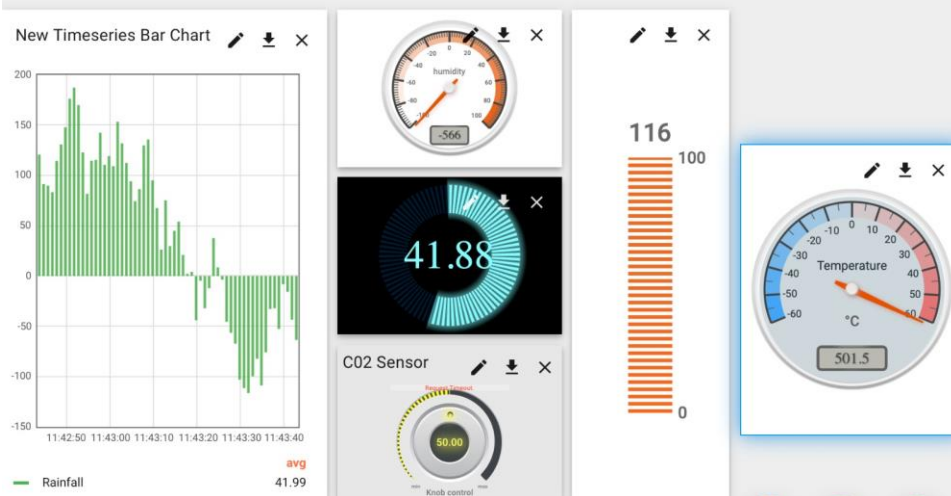
Create a payload, then release it: We may establish a connection between the client and our broker using host:"demo.thingsboard.io", port:1883 and topic:"v1/devices/me/telemetry." It is

constructed using Python and sent in JSON format. We must enter our access token in the "username" field on the connection form to complete it. Remote connectivity is crucial since only then will the payload display live data and dashboard readings. Additionally, download the required libraries to speed up the connection to the cloud service.

First, we download httpie and use the command line tool httpie to make HTTP requests with JSON payloads (For Windows). You can use cURL for mac

Real-time modifications require the WebSocket API and its authorization, which may be deduced from: Using this bash script and an X-Authorization token, http POST <http://demo.thingsboard.io/api/auth/login> 'Content-Type:application/json' 'Accept:application/json' username=iotassignment2@thingsboard.org password=aryan

Workstation1-Aaryan_Sharn



orkstation1-Aaryan_Sharma



Below code is a Python script that sends simulated telemetry data to two virtual environment stations in ThingsBoard, an IoT platform. It uses the paho-mqtt library to establish MQTT connections and send the data as payloads to the ThingsBoard cloud. First, the required libraries are imported, including paho-mqtt for MQTT communication, time for time-related

functions, and random for generating random values for the simulated telemetry data. Next, several functions are defined to generate random values for temperature, humidity, wind direction, wind intensity, and rain height. These functions are called by another function named `get_payload` to generate a JSON payload containing the telemetry data. Afterward, the script defines two MQTT clients and connects them to the ThingsBoard cloud using access tokens specific to the two virtual environment stations. The client objects also have the `on_publish` callback function assigned to handle publishing events. Finally, the script enters a loop that generates new payloads every five seconds and sends them to the respective virtual environment stations using the MQTT clients. The payloads are printed to the console, and the loop continues indefinitely until the script is stopped.

```
def get_payload(): # Generate the payload to send
    payload = '{"Temperature":'
    payload += get_temperature()
    payload += ', "Humidity":'
    payload += get_humidity()
    payload += ', "Wind direction":'
    payload += get_wind_direction()
    payload += ', "Wind intensity":'
    payload += get_wind_intensity()
    payload += ', "Rain height":'
    payload += get_rain_height()
    payload += '}'
    return payload

60 client1 = paho.Client("control1") # Create client1 object
61 client1.on_publish = on_publish # Assign function to callback of client1
62 # Assign to client1 the access token of Virtual Environment Station 1 in ThingsBoard
63 client1.username_pw_set(ACCESS_TOKEN1)
64 # Establish connection using the func connect from the class client
65 client1.connect(broker, port, keepalive=60)
66
67 client2 = paho.Client("control1") # Create client2 object
68 client2.on_publish = on_publish # Assign function to callback of client2
69 # Assign to client2 the access token of Virtual Environment Station 2 in ThingsBoard
70 client2.username_pw_set(ACCESS_TOKEN2)
71 # Establish connection using the func connect from the class client
72 client2.connect(broker, port, keepalive=60)
73
# Sending data
while True:
    payload1 = get_payload() # Get a new payload1
    payload2 = get_payload() # Get a new payload2
    # Send the payload 1 to client1 topic
    ret = client1.publish("v1/devices/me/telemetry", payload1)
    # Send the payload 2 to client2 topic
    ret = client2.publish("v1/devices/me/telemetry", payload2)
    print("Please check LATEST TELEMETRY field of your devices")
    print(payload1)
    print(payload2)
    time.sleep(5) # Wait 5 second to resend another payload to the devices
```

GitHub Repository: https://github.com/aryansharma111/IOT_Assignment2