

Since i did not have a Raspberry pi when doing this test, i decided to use my laptop as the host. Since it was connected to the internet, the python socket that is a python library that i am using to fetch the IP address of my laptop

For MQTT communication, i am using the public broker on port 1883 and i installed a python script to ensure that there is connection to the broker and the IP address of the Laptop has been published to a topic.

For subscribing to the topic, we are to use a tool like MQTT explorer to write another python script to subscribe to the same topic and verify that the IP address is being published.

STEP BY STEP PROCESS.

1. **WE ARE GOING TO INSTALL THE REQUIRED PYTHON LIBRARIES.**
 - We are going to install the paho-mqtt library for MQTT communication.

Command:

Pip install paho-mqtt

```
PS C:\Users\user\mywork> cd "C:\mywork\iot_intership"
PS C:\mywork\iot_intership> pip install paho-mqtt
Collecting paho-mqtt
  Downloading paho_mqtt-2.1.0-py3-none-any.whl.metadata (23 kB)
  Downloading paho_mqtt-2.1.0-py3-none-any.whl (67 kB)
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-2.1.0
PS C:\mywork\iot_intership>
```

PAHO-MQTT is a library that helps devices to communicate using MQTT (message Queueing Telemetry Transport) – that is a lightweight protocol used in IOT systems.

MQTT – is like a post office for devices. Its uses a broker (a central server) to send and receive message between devices. Clients can publish messages or subscribe to topics (categories of message).

ROLE. – is the tool (library) you use to write python programs that can send or receive message via MQTT. It handles all the technical details.

2. **WE WILL THEN INITIALIZATION GIT**

```
Successfully installed paho-mqtt-2.1.0
PS C:\mywork\iot_intership> git init
Initialized empty Git repository in C:/mywork/iot_intership/.git/
PS C:\mywork\iot_intership> git config --global user.name "mwangi"
PS C:\mywork\iot_intership> git config --global user.email "mwangiwanjiku033@gmail.com"
PS C:\mywork\iot_intership> git remote add origin https://github.com/mwangi/mqtt.git
PS C:\mywork\iot_intership>
```

3. WE ARE THEN GOING TO CREATE THE PYTHON FILE THAT HELPS IN CONNECTING TO THE BROKER.

the file is known as the mqtt_ip_publisher.py

```
import paho.mqtt.client as mqtt
import time
```

```
BROKER = "broker.emqx.io"
PORT = 1883
TOPIC = "gateway/ip"
RETRY_DELAY = 5
```

```
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected successfully to the MQTT broker.")
    else:
        print(f"Failed to connect, return code {rc}")
```

```
# Create an MQTT client instance
client = mqtt.Client()
```

```
# Assign the on_connect callback function
client.on_connect = on_connect
```

```
# Attempt to connect to the broker
while True:
    try:
        client.connect(BROKER, PORT, 60)
        break
    except Exception as e:
        print(f"Connection failed: {e}. Retrying in {RETRY_DELAY} seconds...")
        time.sleep(RETRY_DELAY)
```

```
# Start the MQTT client loop
client.loop_start()
```

FETCH THE DYNAMIC IP ADDRESS OF THE PC

```
import socket
def get_ip_address():
    try:
        ip_address = socket.gethostbyname(socket.gethostname())
        return ip_address
    except Exception as e:
        print(f'Error retrieving IP address: {e}')
        return None
```

CALL THIS FUNCTION AFTER SUCCESSFULLY CONNECTING TO THE BROKER:

```
# Fetch the dynamic IP address
ip_address = get_ip_address()
if ip_address:
    print(f'IP Address: {ip_address}')
```

PUBLISH THE IP ADDRESS TO THE MQTT TOPIC

```
# Publish the IP address to the MQTT topic
if ip_address:
    client.publish(TOPIC, ip_address)
    print(f'IP Address published to topic '{TOPIC}': {ip_address}')
else:
    print("No IP address to publish.")
```

RUNNING THE SCRIPT.

After running the script, the output in the terminal by the program should be an IP address of the PC.

We should first run the *ipconfig* command on the terminal of the PC, then check the output for the IP of the program for the IP address of the system then check whether the IPv4 address of the system is similar to what we have on the output terminal of the python program.

```
C:\> Command Prompt

Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 10:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::2cb1:3af1:5851:bee1%5
IPv4 Address. . . . . : 192.168.1.6
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1%5
                             192.168.1.1

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

C:\Users\user>
```

we can see that from the above diagram that shows the output of the command prompt of my PC, we can see that the IPv4 address is 192.168.1.6 and is similar to the screenshot of the python program from the script i created.

```
mqtt_ip_publisher.py X
mqtt_ip_publisher.py > ...
40 client.loop_start()
41
42 # Fetch the dynamic IP address
43 ip_address = get_ip_address()
44
45 # Publish the IP address to the MQTT topic
46 if ip_address:
47     client.publish(TOPIC, ip_address)
48     print(f"IP Address published to topic '{TOPIC}': {ip_address}")
49 else:
50     print("No IP address to publish.")
51
```

```
PS C:\mywork\iot_intership> & C:\Users\user\.virtualenvs\storeFront_Django_Tutorial-qZwR4MIE\Scripts\python.exe c:\mywork\iot_intership\mqtt_ip_publisher.py
c:\mywork\iot_intership\mqtt_ip_publisher.py:25: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
IP Address published to topic 'gateway/ip': 192.168.1.6
PS C:\mywork\iot_intership>
```

To simulate the dynamic functionality of the system, i tried to connect the PC to my mobile hotspot so that to get a different IP address of the PC and demonstrate whether it is being received by the python program.

The second program output from the cmd is:

```
Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::ca0d:703:926d:fa41%5
    IPv4 Address. . . . . : 172.20.10.6
    Subnet Mask . . . . . : 255.255.255.240
    Default Gateway . . . . . : 172.20.10.1

Ethernet adapter Bluetooth Network Connection:
```

The second program output from the python program is:

```
PS C:\mywork\iot_intership> & C:/Users/user/.virtualenvs/storeFront_Django_Tutorial-qZwR4MIE/Scripts/python.exe c:/mywork/iot_intership.py
c:\mywork\iot_intership\mqtt_ip_publisher.py:25: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
IP Address published to topic 'gateway/ip': 192.168.1.6
● PS C:\mywork\iot_intership> & C:/Users/user/.virtualenvs/storeFront_Django_Tutorial-qZwR4MIE/Scripts/python.exe c:/mywork/iot_intership.py
c:\mywork\iot_intership\mqtt_ip_publisher.py:25: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
IP Address published to topic 'gateway/ip': 172.20.10.6
○ PS C:\mywork\iot_intership> █
```