Name: subhash polisetti

For Internet of Things (IoT) devices, connecting to the Internet is kind of a requirement. Connecting to the Internet allows the devices to work with each other and with backend services. The underlying network protocol of the Internet **is TCP/IP. Built on top of the TCP/IP stack, MQTT (Message Queue Telemetry Transport) has become the standard for IoT communications. MQTT can also run on SSL/TLS, which is a secure protocol built on TCP/IP, to ensure that all data communication between devices are encrypted and secure.**

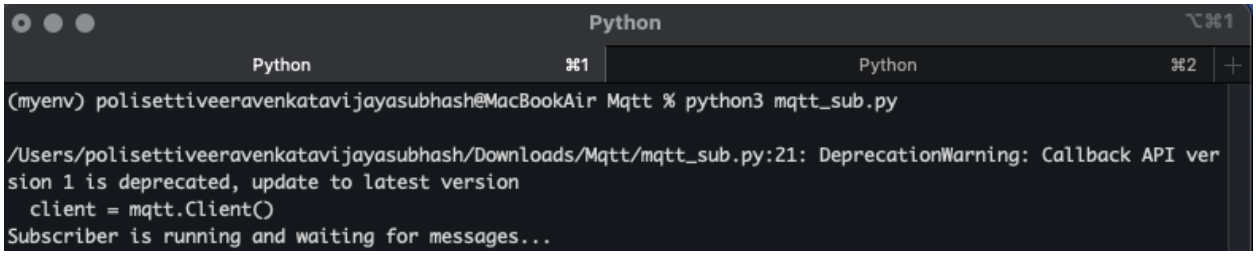
MQTT was **originally invented and developed by IBM in the late 1990's. Its** original application was to link sensors on oil pipelines with satellites. As its name suggests, it is a messaging protocol that supports asynchronous **communication between parties. An asynchronous messaging protocol de-couples the message sender and receiver in both space and** time, and hence is scalable in unreliable network environments.

Please install mosquitto and run the Publisher and Subscriber applications 10000 in an **asynchronous manner. That is, run producer 1000000 times and capture message count.** Now run consumer and make sure the same count is there!

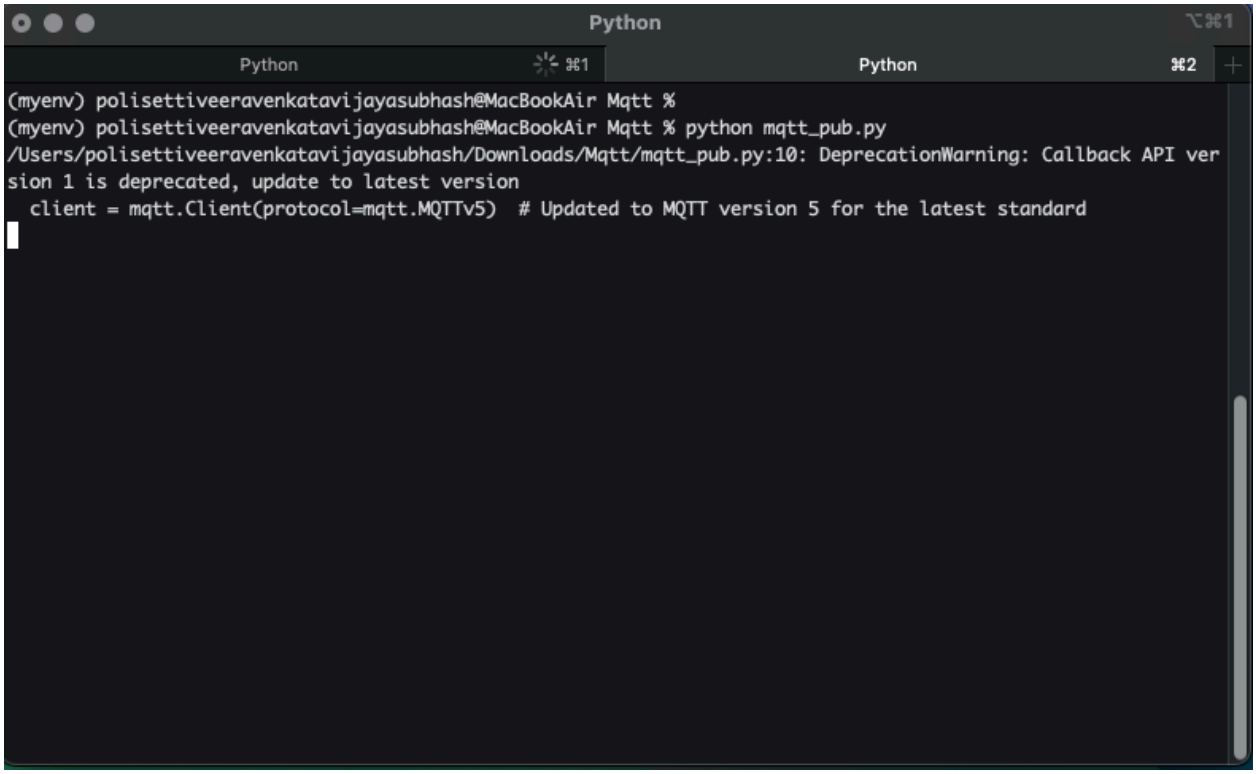
Implementation:-

Github repo: https://github.com/subhashpolisetti/Python-MQTT-Message-Stream

Step-1: **Run the subscriber** to start listening for messages:



Step-2: **Run the publisher** to start sending 1,000,000 messages



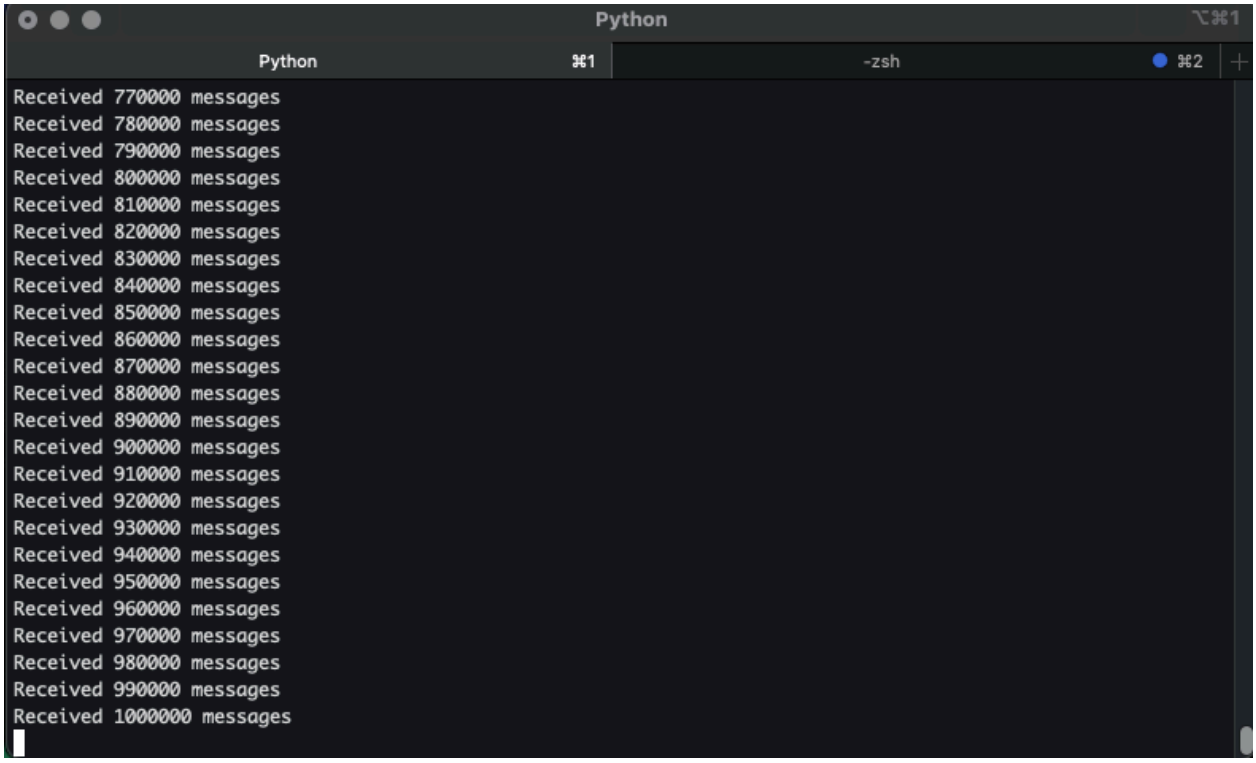
The subscriber is receiving the messages.

```
(myenv) polisettiveeravenkatavijayasubhash@MacBookAir Mqtt % python3 mqtt_sub.py

/Users/polisettiveeravenkatavijayasubhash/Downloads/Mqtt/mqtt_sub.py:21: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Subscriber is running and waiting for messages...
Received 10000 messages
Received 20000 messages
Received 30000 messages
Received 40000 messages
Received 50000 messages
Received 60000 messages
Received 70000 messages
Received 80000 messages
Received 90000 messages
Received 100000 messages
Received 110000 messages
Received 120000 messages
Received 130000 messages
Received 140000 messages
Received 150000 messages
Received 160000 messages
Received 170000 messages
Received 180000 messages
Received 190000 messages
```

Verify the Message Count

- The subscriber terminal should display message counts every 10,000 messages until it reaches 1,000,000.
- This confirms that the subscriber received all published messages.



Successfully Recieved 1000000 messages.

Detailed Steps:

Step 1: Install Mosquitto

Step 2: Install the `paho-mqtt` Python Package

Step 3: Write the Publisher Script
This script will connect to the Mosquitto broker and publish 1,000,000 messages.

output.txt | mqtt_pub.py | mqtt_sub.py

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

# MQTT settings
broker = "localhost"  # Use 'localhost' if Mosquitto is running on your machine
port = 1883           # Default port for MQTT
topic = "test/topic"  # Topic to publish messages to

# Create a new MQTT client instance using the latest API
client = mqtt.Client(protocol=mqtt.MQTTv5)  # Updated to MQTT version 5 for the

# Define callback for connection
def on_connect(client, userdata, flags, rc, properties=None):
    if rc == 0:
        print("Connected to the broker successfully.")
    else:
        print("Failed to connect, return code %d\n", rc)

client.on_connect = on_connect

# Connect to the broker
client.connect(broker, port)

# Publish 1,000,000 messages
message_count = 1000000

for i in range(message_count):
    message = f"Message {i+1}"
    client.publish(topic, payload=message, qos=0)

    # Optional: Adjust this sleep time if messages are sent too fast for the su
    time.sleep(0.001)

print("Publishing complete!")

# Disconnect from the broker
client.disconnect()
```

Step-4: Write the Subscriber Script

This script will connect to the broker and subscribe to the same topic, counting the messages it receives.

output.txt | mqtt_pub.py | mqtt_sub.py

```python
import paho.mqtt.client as mqtt

# MQTT settings
broker = "localhost"
port = 1883
topic = "test/topic"

# Initialize message counter
message_count = 0

# Callback function for when a message is received
def on_message(client, userdata, message):
    global message_count
    message_count += 1

    # Print count every 10,000 messages to monitor progress
    if message_count % 10000 == 0:
        print(f"Received {message_count} messages")

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

# Assign callback function
client.on_message = on_message

# Connect to the broker
client.connect(broker, port)

# Subscribe to the topic
client.subscribe(topic)

# Start the loop to process received messages
print("Subscriber is running and waiting for messages...")
client.loop_forever()
```

Step 5: Run the Mosquitto Broker

brew services start mosquitto

Step 6: Run the Subscriber Script

python3 mqtt_sub.py

Step 7: Run the Publisher Script

python3 mqtt_pub.py

Step 8: Verify the Message Count.

**********end**********