MQTT Broker Setup and Client Codes

Tulasi Sainath Polisetty

November 8, 2023

Mosquitto MQTT Broker Setup Commands

Below is a summary of the commands used for setting up the Mosquitto MQTT broker:

```
# Update packages and install Mosquitto and Mosquitto clients
sudo apt-get update
sudo apt-get install mosquitto mosquitto-clients

# Check the status of Mosquitto service
sudo systemctl status mosquitto

# Install net-tools for netstat (if needed)
sudo apt install net-tools

# Check Mosquitto listening ports
sudo netstat -tulpn | grep mosquitto

# Create a new configuration file in /etc/mosquitto/conf.d/
sudo nano /etc/mosquitto/conf.d/default.conf
```

The contents of the ‘default.conf‘ file:

```
listener 1883
allow_anonymous true
```

```
# Restart Mosquitto service to apply the new configuration
sudo systemctl restart mosquitto
```

AWS Security Group Configuration

This part of the setup was done via the AWS Management Console:

- Set inbound rules to allow traffic on port 1883.

Connectivity Test

To test the connectivity to the MQTT broker:

```
telnet your-ec2-public-ip 1883
# or
nc -vz your-ec2-public-ip 1883
```

Verbose Logging Configuration

To enable verbose logging in Mosquitto (if needed for troubleshooting), add the following lines to the Mosquitto configuration file:

```
log_type all
connection_messages true
log_timestamp true
```

Publish Code for Arduino Nano IoT

This is the code for publishing messages to an MQTT topic using an Arduino Nano IoT:

```
#include <SPI.h>
#include <WiFiNINA.h>
#include <PubSubClient.h>

// Update these with your network credentials
const char* ssid = "yourSSID";
const char* password = "yourPASSWORD";
const char* mqttServer = "yourMQTTbrokerIP";

int status = WL_IDLE_STATUS;
WiFiClient wifiClient;
PubSubClient client(wifiClient);

void setup() {
  // Initialize serial and wait for the port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect.
  }

  // Attempt to connect to WiFi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, password);
    // Wait 10 seconds for connection:
    delay(10000);
  }
  // Connected to WiFi
  Serial.println("Connected to WiFi");
  client.setServer(mqttServer, 1883);
}

void loop() {
```

```
  // Ensure the client is connected
  if (!client.connected()) {
    // Connect to the MQTT server
    if (client.connect("arduinoClient")) {
      // Once connected, publish a message
      client.publish("outTopic", "hello world");
    }
  }
  client.loop();
}
```

Subscribe Code for ESP8266

This is the code for subscribing to MQTT topics using an ESP8266 module:

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with your network credentials
const char* ssid = "yourSSID";
const char* password = "yourPASSWORD";
const char* mqttServer = "yourMQTTbrokerIP";

WiFiClient espClient;
PubSubClient client(espClient);

void setup_wifi() {
  delay(10);
  // Connect to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

```

```
42 void reconnect() {
43   // Loop until we’re reconnected
44   while (!client.connected()) {
45     Serial.print("Attempting MQTT connection...");
46     // Attempt to connect
47     if (client.connect("ESP8266Client")) {
48       Serial.println("connected");
49       // Once connected, resubscribe
50       client.subscribe("inTopic");
51     } else {
52       Serial.print("failed, rc=");
53       Serial.print(client.state());
54       Serial.println(" try again in 5 seconds");
55       // Wait 5 seconds before retrying
56       delay(5000);
57     }
58   }
59 }
60
61 void setup() {
62   Serial.begin(115200);
63   setup_wifi();
64   client.setServer(mqttServer, 1883);
65   client.setCallback(callback);
66 }
67
68 void loop() {
69   if (!client.connected()) {
70     reconnect();
71   }
72   client.loop();
73 }
```