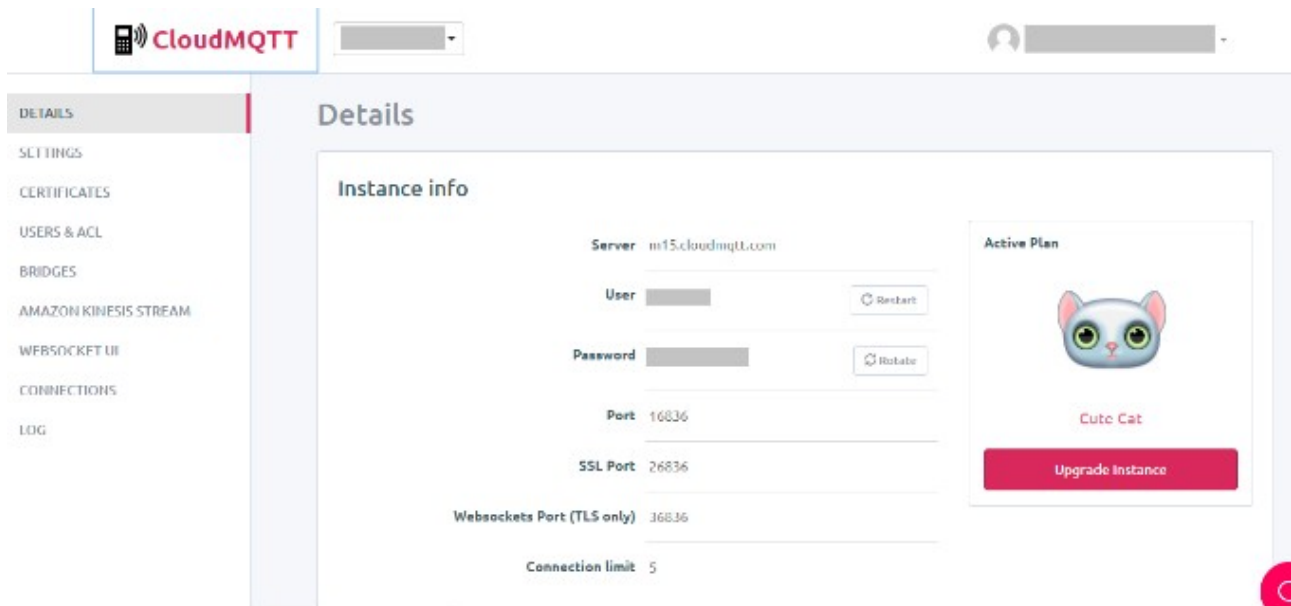


Broker MQTT na cloud

Para esse tipo de projeto é comum a utilização de brokers na nuvem, onde alguns oferecem contas gratuitas com algumas limitações.

Inicialmente esperava utilizar a io.adafruit.com, porém tive problemas para conseguir postar no feed, então parti para outro broker, o cloudmqtt.com.



Aplicação para o ESP32

Para validar o conceito, busquei alguns exemplos na internet e cheguei ao código abaixo, onde ocorrem ações básicas como:

- estabelece conexão na rede wi-fi;

```
1  #include <PubSubClient.h>
2  #include <WiFi.h>
3
4  const char* ssid = "rede";
5  const char* password = "senha";
6  const char* mqttServer = "m15.cloudmqtt.com";
7  const int mqttPort = 16836;
8  const char* mqttUser = "usuario";
9  const char* mqttPassword = "senha";
10 int LED_BUILTIN = 2;
11 WiFiClient espClient;
12 PubSubClient client(espClient);
13
```

- conecta ao broker;

```
20 while (WiFi.status() != WL_CONNECTED) {
21     delay(500);
22     Serial.print("Connecting to WiFi:");
23     Serial.println(ssid);
24 }
25
26 Serial.println("Connected to the WiFi network");
27 Serial.println("");
28 Serial.println("IP address: ");
29 Serial.println(WiFi.localIP());
30 client.setServer(mqttServer, mqttPort);
31 client.setCallback(callback);
32
33 while (!client.connected()) {
34     Serial.println("Connecting to MQTT...");
35     String clientId = "ESP32Client-";
36     clientId += String(random(0xffff), HEX);
37     if (client.connect(clientId.c_str(), mqttUser, mqttPassword )) {
38         Serial.println("connected");
39     } else {
40         Serial.print("failed with state ");
41         Serial.print(client.state());
42         delay(2000);
43     }
44 }
```

- posta uma mensagem de hello world no tópico esp/test;

```
46 Serial.print("Tentando enviar a mensagem");
47 client.publish("esp/test", "Hello from ESP32");
48 client.subscribe("esp/test");
49
```