

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng:

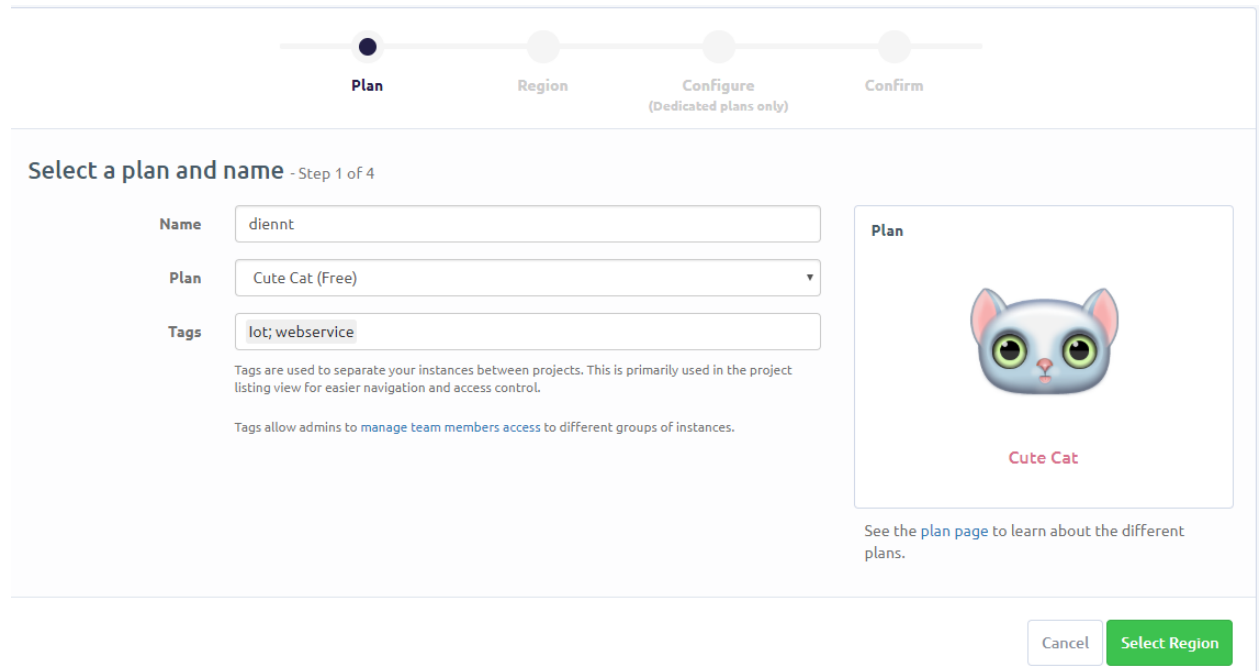
- Tạo và thiết lập tại khoản trên CloudMQTT.
- Lập trình giao tiếp với CloudMQTT.

PHẦN I

Bài 1 (1 điểm): Cài đặt tài khoản CloudMQTT

Bước 1: vào trang <https://www.cloudmqtt.com/>

Bước 2: Đăng ký tài khoản



The screenshot shows the 'Plan' step of the CloudMQTT account creation process. At the top, a progress bar indicates four steps: Plan (active), Region, Configure (Dedicated plans only), and Confirm. Below the progress bar, the heading 'Select a plan and name - Step 1 of 4' is displayed. The form contains three input fields: 'Name' with the value 'diennt', 'Plan' with a dropdown menu showing 'Cute Cat (Free)', and 'Tags' with the value 'iot; webservice'. Below the tags field, there is explanatory text: 'Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control. Tags allow admins to manage team members access to different groups of instances.' To the right of the form, there is a 'Plan' section featuring a cute cat illustration and the text 'Cute Cat'. Below this, a link says 'See the plan page to learn about the different plans.' At the bottom right, there are two buttons: 'Cancel' and 'Select Region'.

Bước 3: Chọn vùng

No credit card Please add a credit card if you want to subscribe to a paid plan

Plan Region Configure (Dedicated plans only) Confirm

Select a region and data center - Step 2 of 4

Data center EU-West-1 (Ireland)

aws

« Back Cancel Confirm

Bước 4: Xác thực thông tin

Confirm new instance - Step 4 of 4

Plan

Cute Cat

Total: \$0/month

Name: diennt
Provider: Amazon Web Services
Region: EU-West-1 (Ireland)
Tags: lot; webservice

Bước 5: Hoàn tất thông tin

CloudMQTT

List all instances

diennt@fpt.edu.vn

Instances

Instance successfully created

lot; webservice

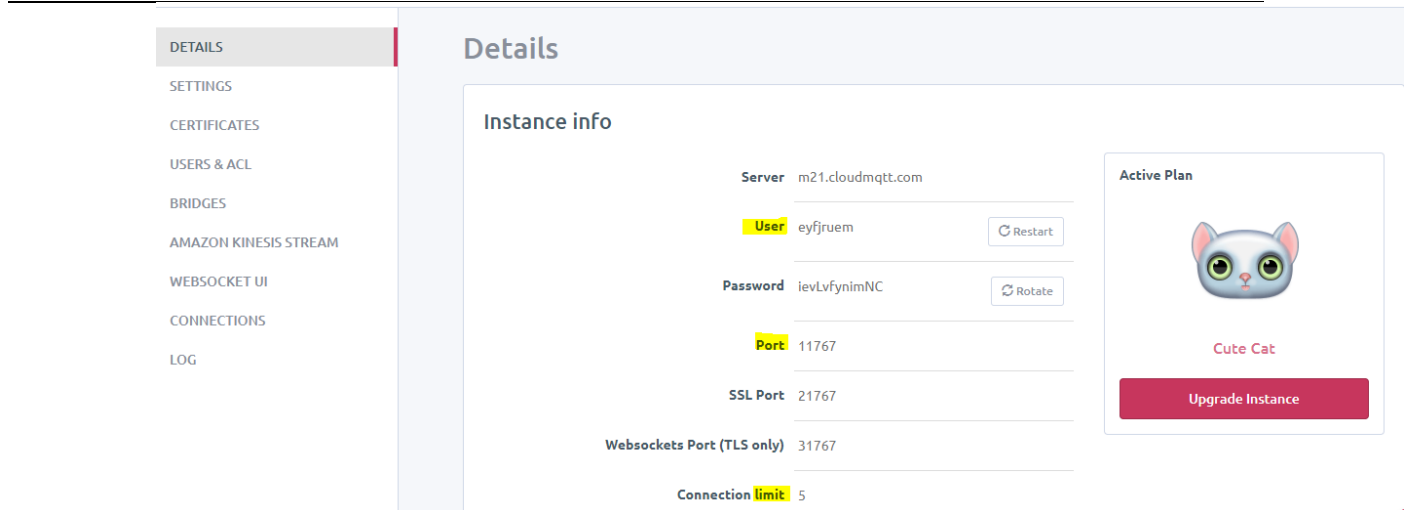
Name	Plan	Datacenter	Actions
diennt	Cat	Amazon Web Services EU-West-1 (Ireland)	Edit

Bước 6: Trong **User & ACL** điền tên user là esp8266, password là 123456, sau đó **Add**

Bước 7 : Trong New Rule chọn user như hình, tên topic đặt demo, tick chọn Read,Write

Type	Pattern	Read/Write
<input checked="" type="radio"/> Pattern <input type="radio"/> Topic	esp8266	<input checked="" type="checkbox"/> Read Access? <input checked="" type="checkbox"/> Write Access?

Bước 8 : Chú ý thông tin quan trọng



Sinh viên chụp lại kết quả thực hiện và nộp sản phẩm **bai1.docx**

Bài 2 (3 điểm): Sinh viên sử dụng tiếp bài 1, tiến hành lập trình.

Chuẩn bị:

- ESP8266 (NodeMCU)
- Arduino IDE
- Thư viện Pubsubclient: <https://github.com/knolleary/pubsubclient/>
- MQTTlens cho Chrome:
<https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlgmdigohjobjm>
- Cách kết nối cũng như dùng Arduino cho ESP8266.

Code tham khảo:

- Để ESP8266 có thể publish và subscribe dữ liệu lên MQTT broker thì cần phải có thư viện MQTT: <https://github.com/knolleary/pubsubclient/>

#include

<ESP8266WiFi.h>

#include <PubSubClient.h>

// Cập nhật thông tin

// Thông tin về wifi

#define ssid "ten_wifi"

```

#define password "password"
// Thông tin về MQTT Broker

#define mqtt_server "m21.cloudmqtt.com" // Thay bằng thông tin của
bạn

#define mqtt_topic_pub "demo" //Giữ nguyên nếu bạn tạo topic tên
là demo

#define mqtt_topic_sub "demo"

#define mqtt_user "esp8266" //Giữ nguyên nếu bạn tạo user là
esp8266 và pass là 123456

#define mqtt_pwd "123456"


const uint16_t mqtt_port = 11767; //Port của CloudMQTT


WiFiClient espClient;
PubSubClient client(espClient);


long lastMsg = 0;
char msg[50];
int value = 0;


void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);

```

```

client.setCallback(callback);
}
// Hàm kết nối wifi
void setup_wifi() {
    delay(10);
    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());
}
// Hàm call back để nhận dữ liệu
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();
}

```

// Hàm reconnect thực hiện kết nối lại khi mất kết nối với MQTT

Broker

```
void reconnect() {
    // Chờ tới khi kết nối
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Thực hiện kết nối với mqtt user và pass
        if (client.connect("ESP8266Client",mqtt_user, mqtt_pwd)) {
            Serial.println("connected");
            // Khi kết nối sẽ publish thông báo
            client.publish(mqtt_topic_pub, "ESP_reconnected");
            // ... và nhận lại thông tin này
            client.subscribe(mqtt_topic_sub);
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Đợi 5s
            delay(5000);
        }
    }
}

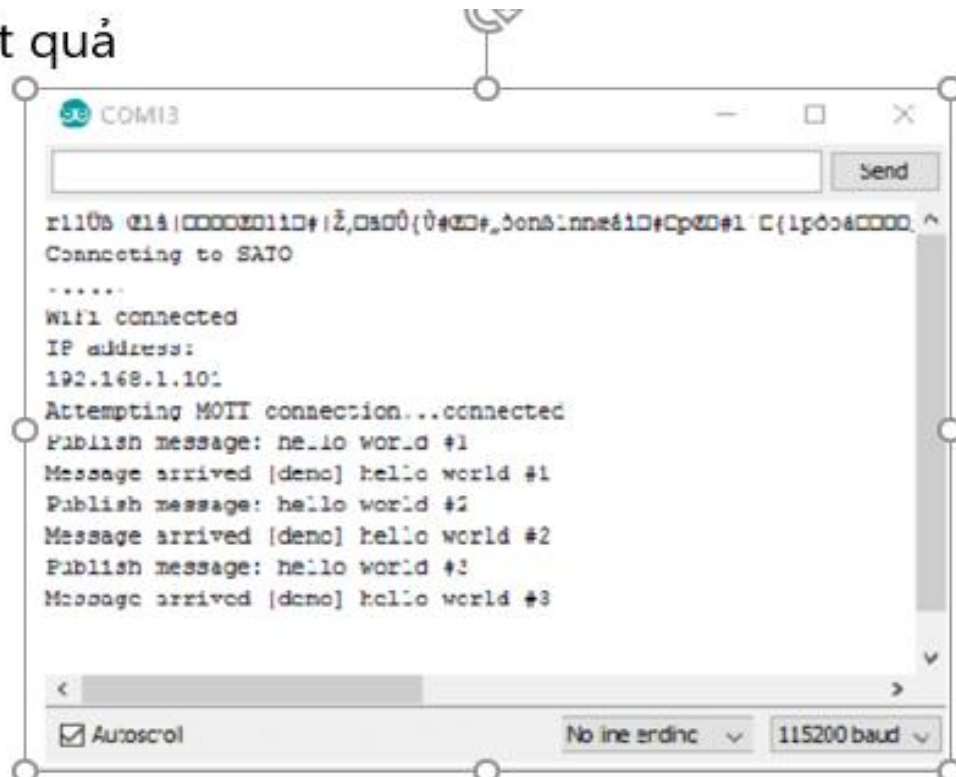
void loop() {
    // Kiểm tra kết nối
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}
```


```
// Sau mỗi 2s sẽ thực hiện publish dòng hello world lên MQTT
broker
```

```
    long now = millis();

    if (now - lastMsg > 2000) {
        lastMsg = now;
        ++value;
        snprintf(msg, 75, "hello world #%ld", value);
        Serial.print("Publish message: ");
        Serial.println(msg);
        client.publish(mqtt_topic_pub, msg);
    }
}
```

Kết quả



 Sinh viên chụp lại kết quả thực hiện và nộp sản phẩm **bai2.docx**

PHẦN II

Bài 3 (4 điểm): Lập trình để ESP8266 thực hiện bật/tắt LED có sẵn trên kit khi Websocket UI Send message xuống.

Code tham khảo:

```
#include
<ESP8266WiFi.h>

#include <PubSubClient.h>
// Cập nhật thông tin
// Thông tin về wifi
#define ssid "ten_wifi"
#define password "password"
// Thông tin về MQTT Broker
#define mqtt_server "m12.cloudmqtt.com" // Thay bằng thông tin của
bạn
#define mqtt_topic_pub "demo" //Giữ nguyên nếu bạn tạo topic tên
là demo
#define mqtt_topic_sub "demo"
#define mqtt_user "esp8266" //Giữ nguyên nếu bạn tạo user là
esp8266 và pass là 123456
#define mqtt_pwd "123456"

const uint16_t mqtt_port = 10769; //Port của CloudMQTT
const byte ledPin = D0;
```

```
WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
int value = 0;

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
  pinMode(ledPin, OUTPUT);
}
// Hàm kết nối wifi
void setup_wifi() {
  delay(10);
  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());
}
// Hàm call back để nhận dữ liệu
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++) {
        char receivedChar = (char)payload[i];
        Serial.print(receivedChar);
        if (receivedChar == '1')
            // Kiểm tra nếu tin nhận được là 1 thì bật LED và ngược lại
            digitalWrite(ledPin, HIGH);
        if (receivedChar == '0')
            digitalWrite(ledPin, LOW);
    }
    Serial.println();
}
// Hàm reconnect thực hiện kết nối lại khi mất kết nối với MQTT
Broker
void reconnect() {
    // Chờ tới khi kết nối
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");

```

```
// Thực hiện kết nối với mqtt user và pass
if (client.connect("ESP8266Client",mqtt_user, mqtt_pwd)) {
    Serial.println("connected");
    // Khi kết nối sẽ publish thông báo
    client.publish(mqtt_topic_pub, "ESP_reconnected");
    // ... và nhận lại thông tin này
    client.subscribe(mqtt_topic_sub);
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Đợi 5s
    delay(5000);
}
}
}

void loop() {
    // Kiểm tra kết nối
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}
```

 Sinh viên chụp lại kết quả dung thực hiện : **bai3.docx**

Bài 4 (2 điểm): Giảng viên có thể cho thêm bài tập