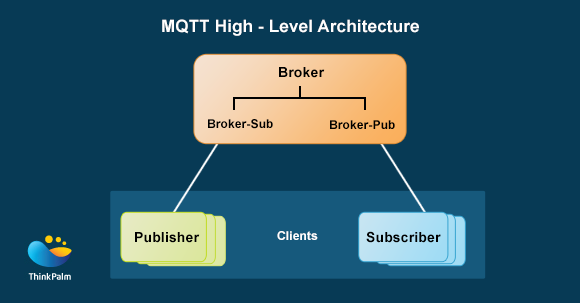
Tìm hiểu về giao thức MQTT ?

# I/ MQTT(Message Queue Telemetry Transport)

1. Khái niệm

Đây là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe (xuất bản – theo dõi), sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

Kiến trúc mức cao (high-level) của MQTT gồm 2 phần chính là Broker và Clients.



Trong đó, broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ chính của broker là nhận mesage từ publisher, xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể. Nhiệm vụ phụ của broker là nó có thể đảm nhận thêm một vài tính năng liên quan tới quá trình truyền thông như: bảo mật message, lưu trữ message, logs,...

Client thì được chia thành 2 nhóm là publisher và subscriber. Client là các software components hoạt động tại edge device nên chúng được thiết kế để có thể hoạt động một cách linh hoạt (lightweight). Client chỉ làm ít nhất một trong 2 việc là publish các message lên một topic cụ thể hoặc subscribe một topic nào đó để nhận message từ topic này.

II/Máy chủ MQTT (mosquitto)



Mosquitto là một MQTT Broker, hay là một máy chủ MQTT. Nó cho phép xuất bản và đăng ký bản tin bằng các câu lệnh mosquitto\_pub và mosquitto\_pub.

Bài viết này hướng dẫn cài đặt và sử dụngMosquitto trên Ubuntu 18.04

1. Cài đặt Mosquitto  
   Đầu tiên ta chạy lệnh sau :

$ sudo apt update

Cài đặt Mosquitto bằng cách sử dụng apt install:

$ sudo apt install mosquitto mosquitto-clients

Theo mặc định, Ubuntu sẽ khởi động dịch vụ Mosquitto sau khi cài đặt. Chạy thử các lệnh sau để kiểm tra kiểm tra hệ thống để đảm bảo mosquitto đã được chạy

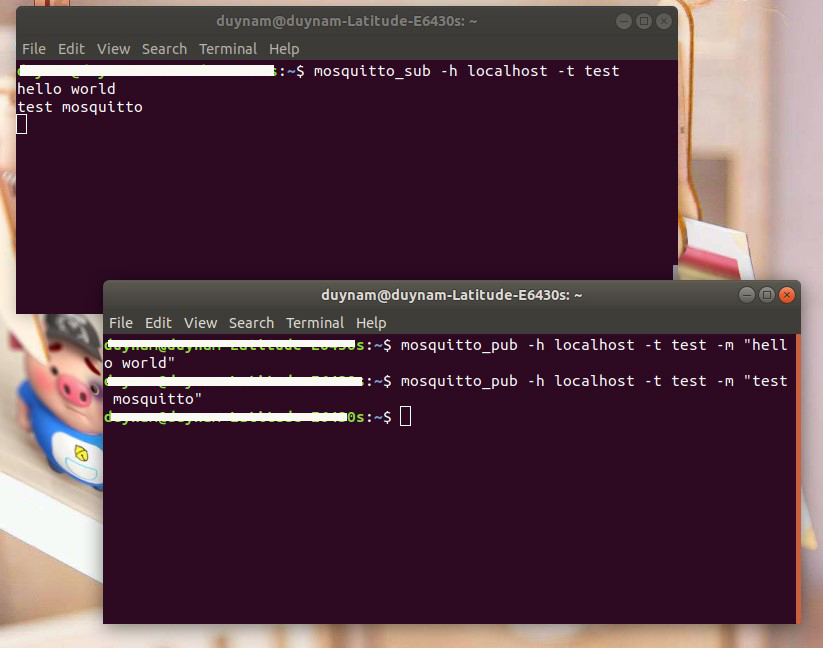
– Subscirbe topic:

$ mosquitto\_sub -h localhost -t test

– Publish topic:

$ mosquitto\_pub -h localhost -t test -m "hello world"

Quá trình sẽ giống như sau:



1. Bảo mật máy chủ MQTT:

Việc bảo mật cho hệ thống là tối quan trọng trong bất kỳ trường hợp nào. Bảo mật còn quyết định sự sống còn của hệ thống.

Hãy cấu hình Mosquitto để sử dụng mật khẩu. Mosquitto bao gồm một tiện ích để tạo một tệp mật khẩu đặc biệt được gọi là mosquitto\_passwd. Lệnh này sẽ nhắc bạn nhập mật khẩu cho tên người dùng được chỉ định và đặt kết quả vào /etc/mosquitto/passwd.

B1: mở file /etc/mosquitto/conf.d/default.conf và điền vào đoạn code sau

$ sudo nano /etc/mosquitto/conf.d/default.conf

allow\_anonymous false

password\_file /etc/mosquitto/passwd

* allow\_anonymous false : sẽ vô hiệu hóa tất cả các kết nối không được xác thực
* password\_file :dòng này cho Mosquitto biết nơi tìm thông tin mật khẩu và người dùng.

Lưu và thoát tập tin.

B2: Tạo user ,password:

$ sudo mosquitto\_passwd -b /etc/mosquitto/passwd **<username>** <password>

Chú ý : username và password bạn nhập sẽ liên quan đến code bên dưới.Ở đây tôi đặt là user: sammy và password: 123456

Khởi động lại Mosquitto và kiểm tra các thay đổi :

$ sudo systemctl restart mosquitto

Bây giờ nếu cố gắng xuất bản một tin nhắn mà không có mật khẩu ví dụ như :

$ mosquitto\_pub -h localhost -t "test" -m "hello world"

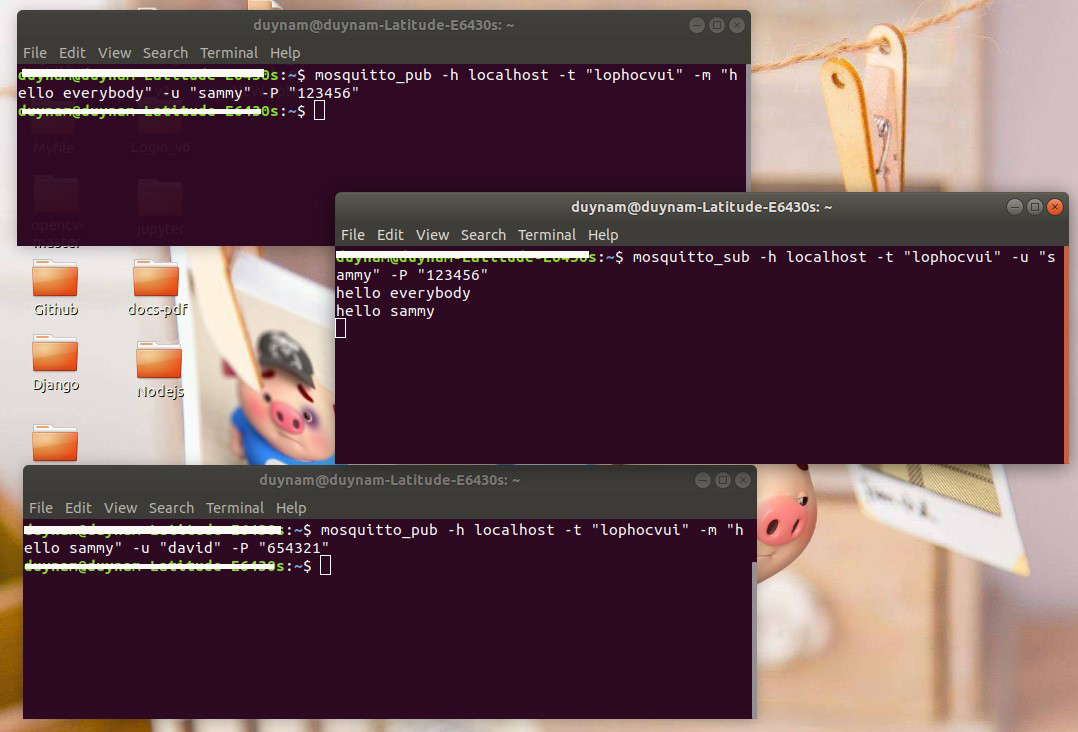
Hệ thống sẽ báo lỗi sau:

Connection Refused: not authorised.

Error: The connection was refused.

Để có thể puslish và subscribe bạn cần phải có tên tài khoản và mật khẩu mà bạn đã khai báo cho máy chủ MQTT biết ở bước trên:

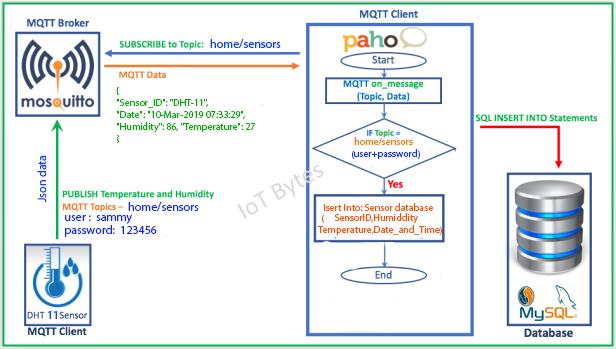
VD : khi 2 user “sammy” và “david” cùng trao đổi thông tin riêng tư trên một topic tên là “lophocvui” .Khi “david” gửi tin nhắn đến máy chủ MQTT và “sammy” nhận được tin nhắn đó nó sẽ được bảo mật 2 lớp. Lớp đầu tiên là muốn kết nối được đến máy chủ họ phải có user và mật khẩu đã khai báo với máy chủ. Lớp thứ 2 là phải subcribe đến đúng “topic” mà người gửi gửi cho người nhận. Quá trình đó diễn ra như sau :



3. Gửi giá trị cảm biến thông qua MQTT lưu vào database

Yêu cầu đề ra :

Viết code python lấy random các giá trị cảm biến vd : Humidity,Temperature,Date\_Time truyền dữ liệu thông qua MQTT dưới dạng chuỗi Json và lưu vào databse.

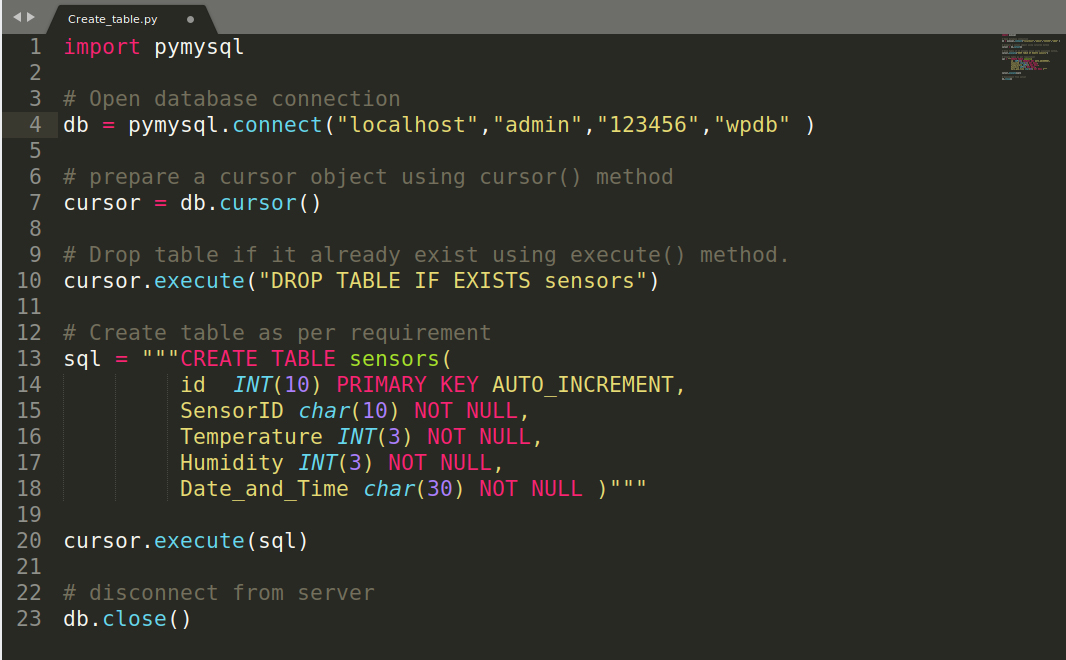


Để giải quyểt ta chia thành các file với các chức năng như sau:

Create\_tables.py => Publish.py => Subscribe.py => Get\_Data\_to\_DB.py

* B1: Khởi tạo database

Ta có thể khởi tạo dabase bằng câu lệnh sql hoặc có thể khởi tạo bằng file python **Create\_tables.py** sau:



* Bước 2: Pusblish giá trị ngẫu nhiên bằng MQTT

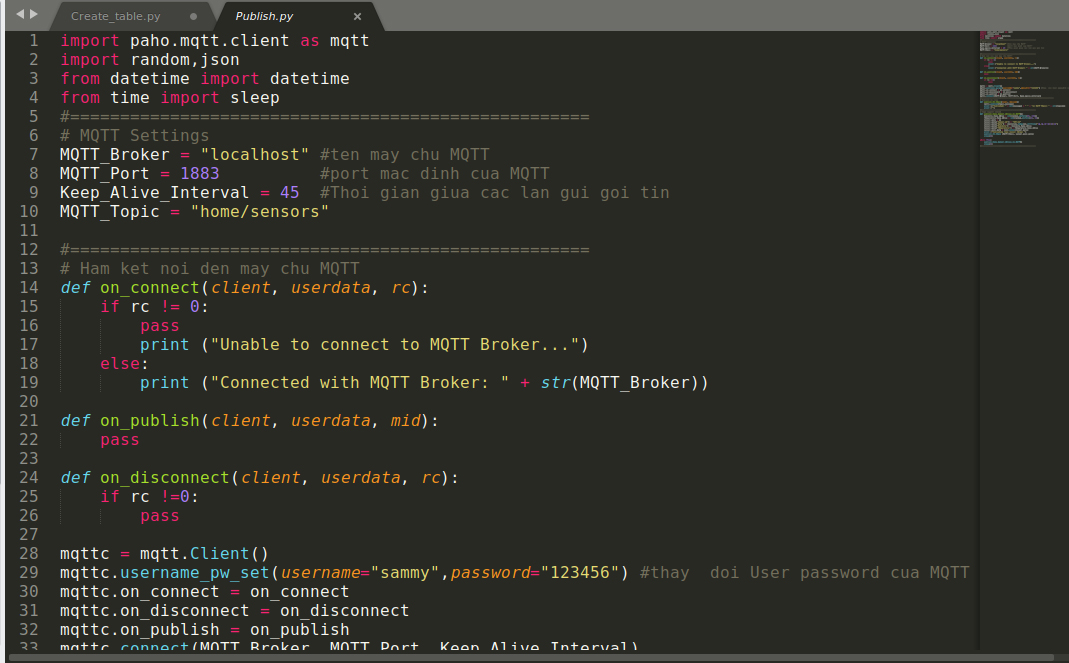
Một số khái niệm:

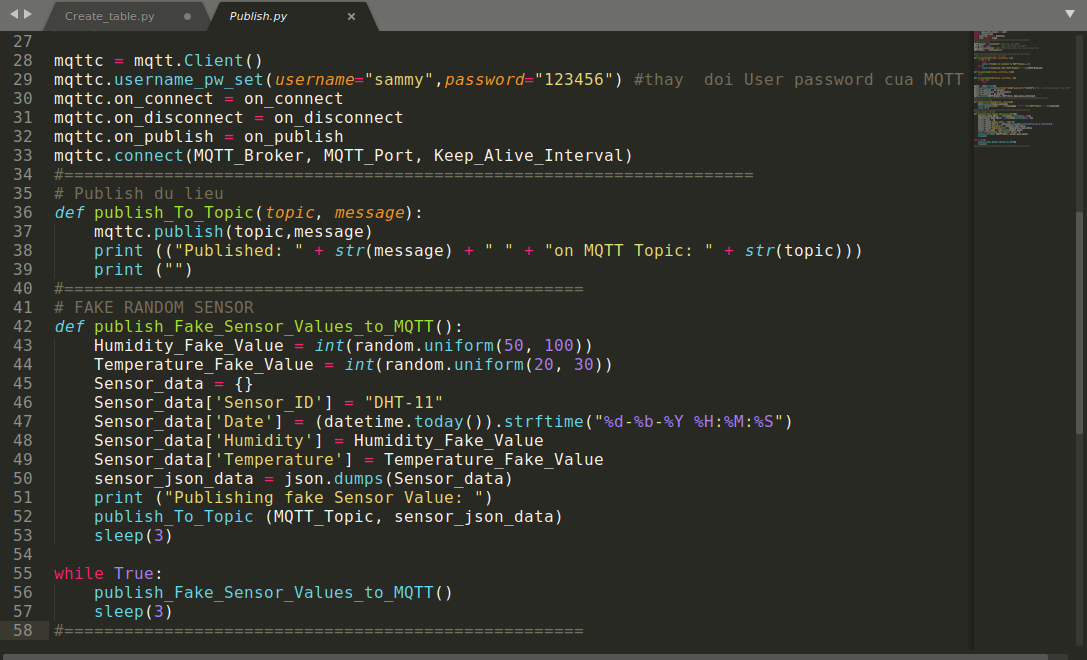
* JSON : là chữ viết tắt của Javascript Object Notation, đây là một dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được, bạn có thể sử dụng lưu nó vào một file, một record trong CSDL rất dễ dàng. JSON có định dạng đơn giản, dễ dàng sử dụng và truy vấn hơn XML rất nhiều nên tính ứng dụng của nó hiện nay rất là phổ biến.
* Paho-mqtt: thư viện Python hỗ trợ làm việc với MQTT

Để publish dữ liệu ta có thể dùng câu lệnh với mosquito như sau:

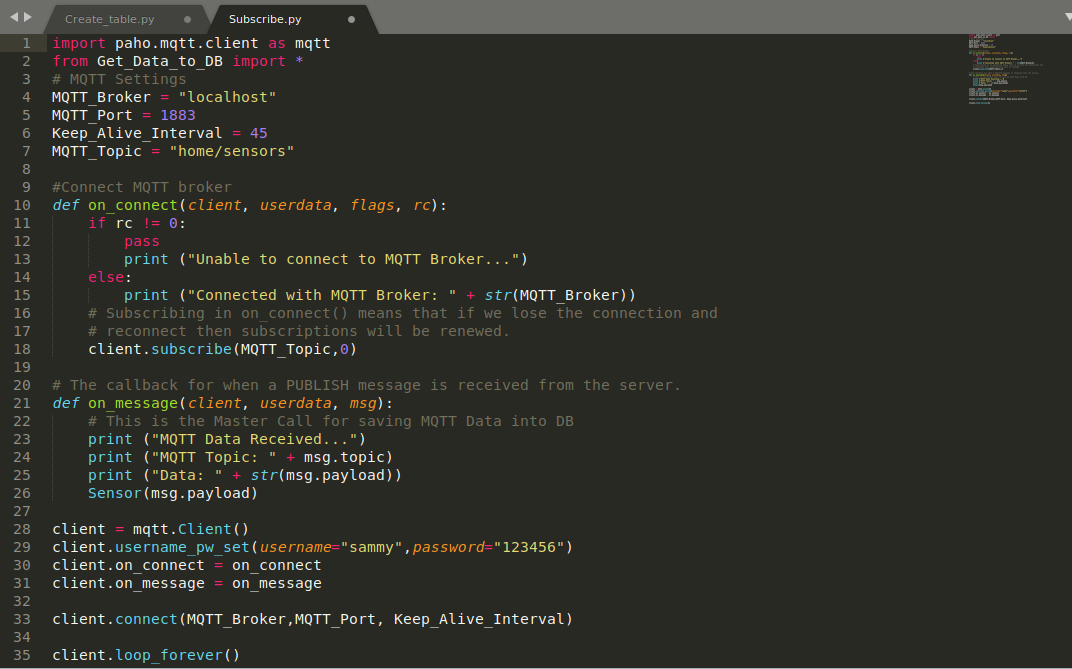
$ mosquitto\_pub -h localhost -t "home/sensor" -m '{"Sensor\_ID": "DHT-11", "Date": "10-Mar-2019 07:33:29", "Humidity": 86, "Temperature": 27} ' -u "sammy" -P "123456"

Hoặc publish giá trị lên tự động bằng code Python .Tạo file Publish.py với nội dung sau:

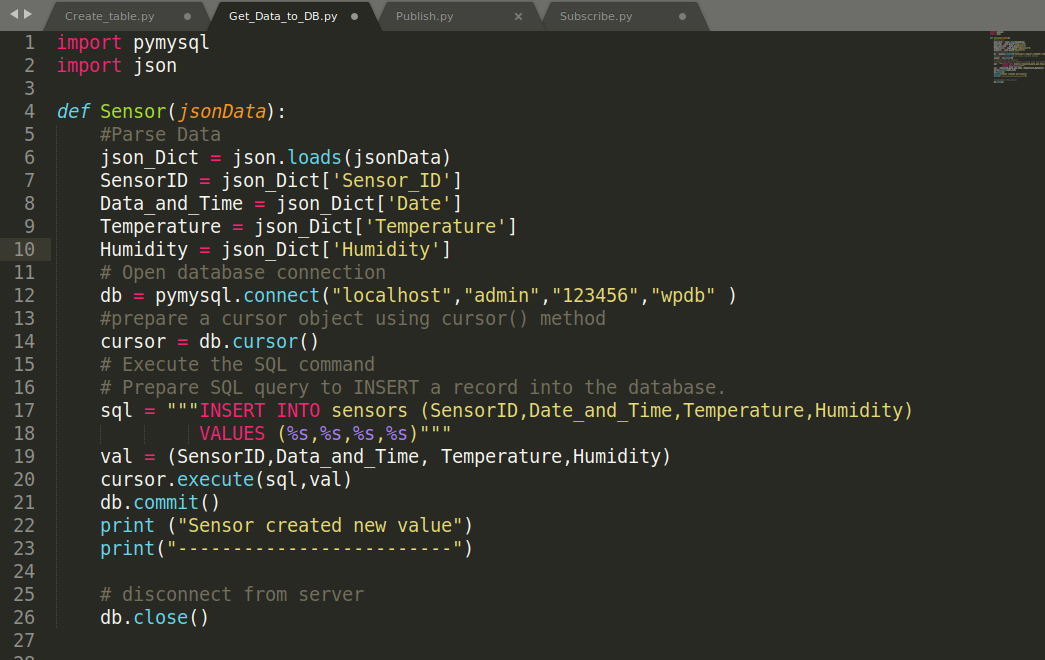




* Bước 3: Tạo file Subscribe.py để nhận dữ liệu Publish :



* Bước 4: Tạo file Get\_Data\_to\_DB.py lưu giá trị vào database:



Và đây là kết quả ,dữ liệu sẽ tự động được thêm vào database:

