Trong phần trước mình đã giới thiệu cho các bạn cách để tạo ra cặp public/ private key và một chuỗi JWT hoàn chỉnh. Ở phần này chúng ta sẽ đi tìm hiểu tuần tự các bước thực hiện kết nối một thiết bị IoT cụ thể tới dịch vụ Google IoT Core dựa trên các thông số đó.

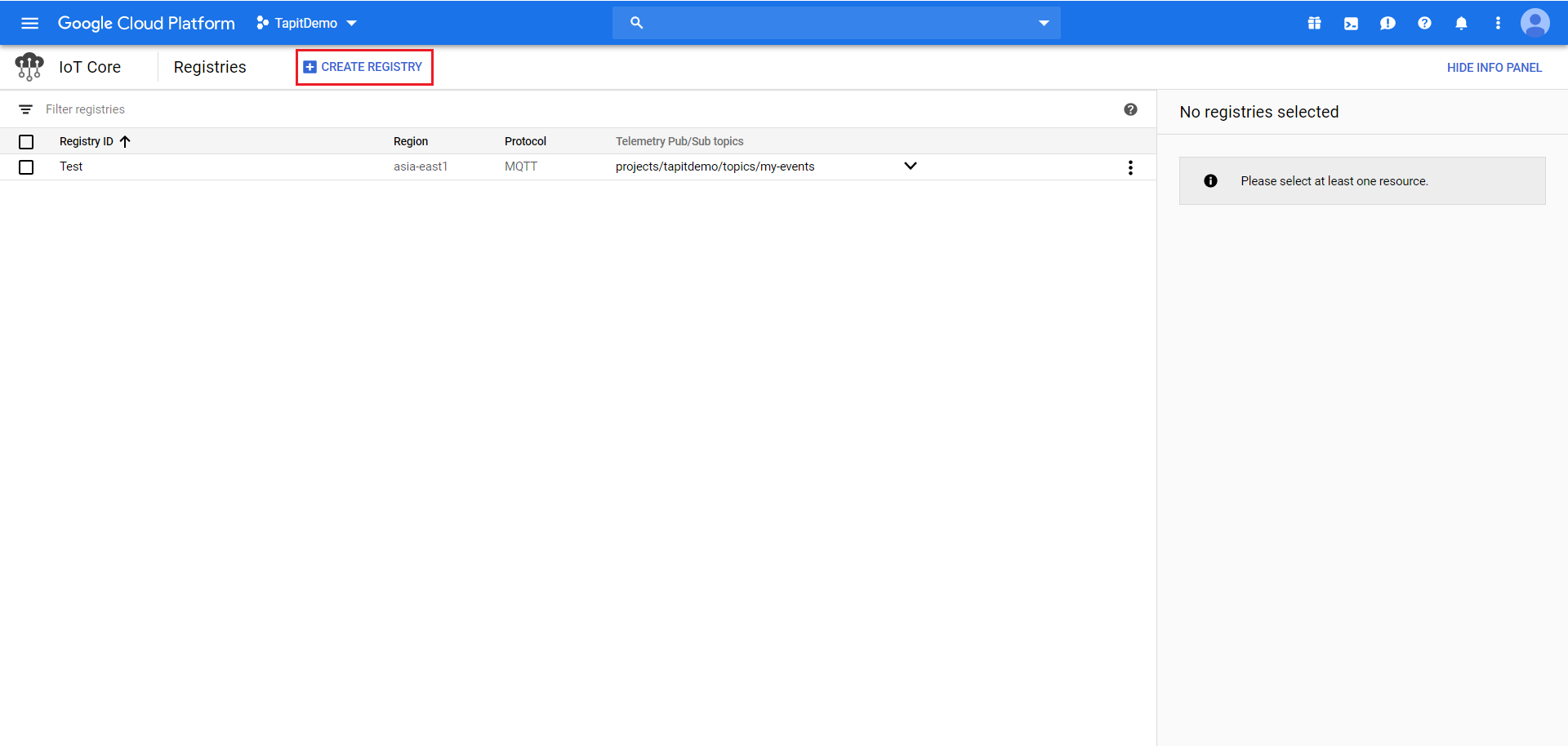
**Phần 3: Hướng dẫn đăng ký và kết nối thiết bị tới Google IoT Core**

**I. Hướng dẫn đăng ký thiết bị với registry và liên kết Public key với device**

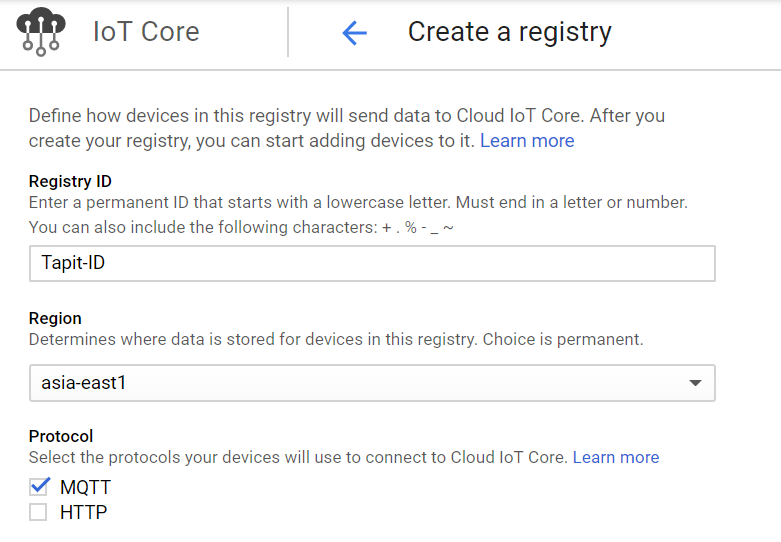
Các thiết bị IoT kết nối đến dịch vụ Google IoT Core được xem là các “device” trong “registry”. Registry là nơi cho phép chúng ta lưu trữ các thiết bị IoT có những tính chất gần giống nhau cùng kết nối tới Google IoT Core.

Để đơn giản hóa thì chúng ta có thể xem *project* như một ngôi nhà, bên trong ngôi nhà lại chia ra các căn phòng như phòng ngủ, phòng khách,... tương ứng với các phòng đó chính là các *registry*. Và những thiết bị gói gọn trong các căn phòng đó sẽ được xem là các *device*.

Để bắt đầu, tại giao diện console của GCP các bạn chọn ***CREATE REGISTRY.***



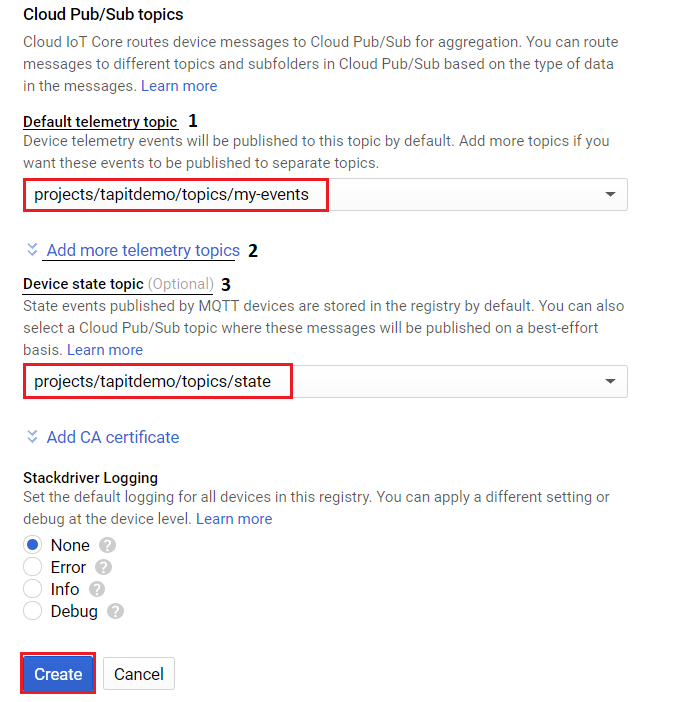
Chúng ta có thể tạo 1 Registry ID bất kỳ (NodeMCU, MyHome,..) nhưng không được phép trùng với bất kỳ registry nào khác nằm bên trong project mà chúng ta đang làm việc, ở đây mình ví dụ đặt tên là Tapit-ID. Tại phần *region* mình sẽ chọn *asia-east1* để tối ưu độ trễ của đường truyền dữ liệu. Còn *protocol* mình sẽ làm việc với giao thức MQTT (Đây là giao thức phổ biến trong việc tương tác machine-to-machine).



Mỗi registry mà chúng ta đăng ký, Google IoT Core bắt buộc chúng ta phải liên kết tới một hay nhiều topics nằm trong một dịch vụ khác có tên là Cloud Pub/Sub (Mình sẽ nói về dịch vụ này ở bài viết sau). Tổ chức của Google IoT Core chính là khi một gói tin gửi từ thiết bị IoT lên Google IoT Core, Google IoT Core sẽ không lưu trữ gói tin đó mà nó sẽ chuyển tiếp gói tin đó tới một dịch vụ khác tên là Cloud Pub/Sub.

Vậy nên nhiệm vụ của chúng ta khi tạo một registry bất kỳ là phải xác định trước gói tin đó sẽ được chuyển tiếp tới những topic nào nằm trong dịch vụ Cloud Pub/Sub.

Ở đây, Google IoT Core đã định nghĩa sẵn cho chúng ta 3 loại topic nằm trong cloud Pub/Sub mà dựa vào đó chúng ta sẽ chia luồng dữ liệu sao cho hợp lý:

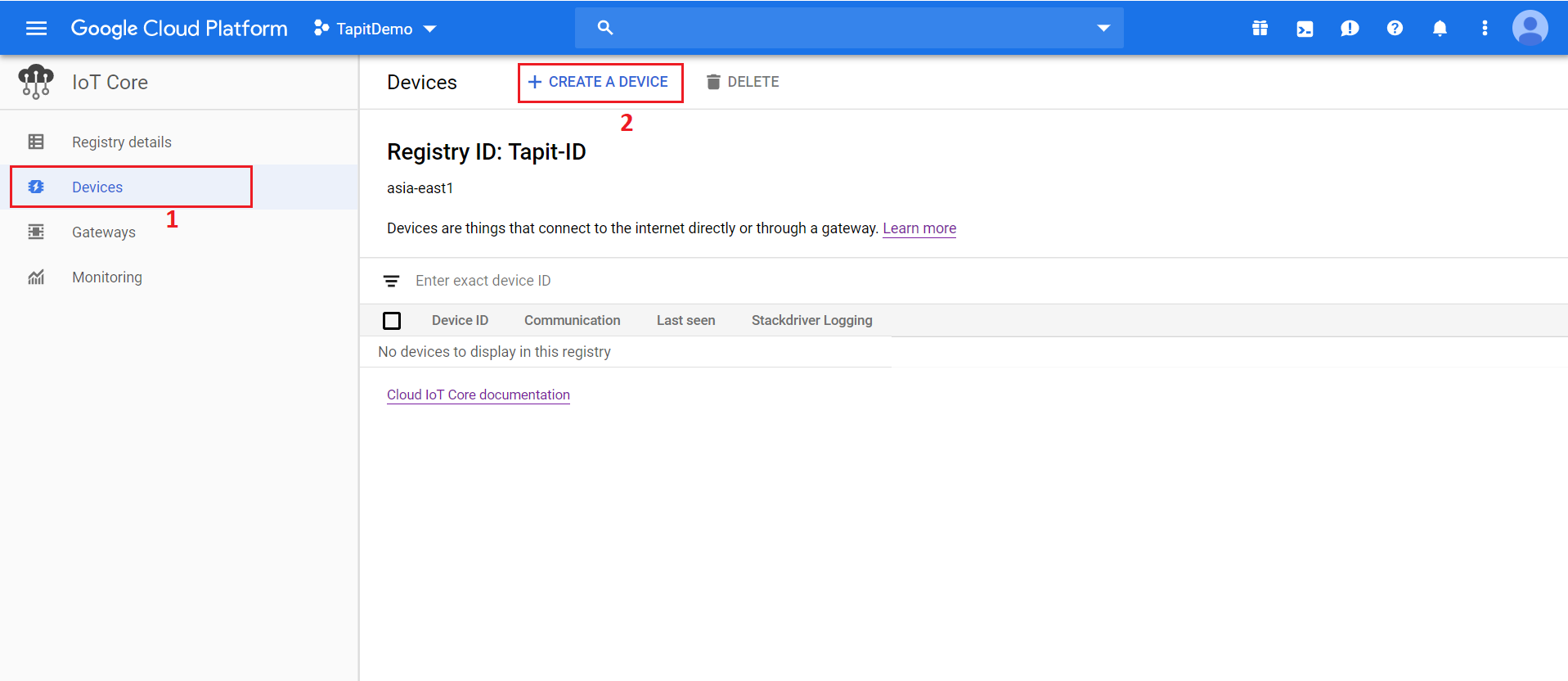


* ***Default telemetry***: Topic này **bắt buộc** phải có khi khởi tạo 1 registry. Mục đích của topic này là dùng để gửi các dữ liệu về **môi trường xung quanh** mà thiết bị thu thập được
* ***Device telemetry***: Topic này **không bắt buộc** phải có khi khởi tạo 1 registry. Tuy nhiên nếu chúng ta muốn khởi tạo thì nhấn vào *Add more telemetry topics*. Mục đích của các topic này cũng giống như topic Default telemetry là dùng để gửi các dữ liệu về **môi trường xung quanh**. Tuy nhiên khi sử dụng các topic này thì chúng ta có thể phân luồng dữ liệu cụ thể từ một thiết bị hơn.
* ***Device state***: Topic này **không bắt buộ**c phải có khi khởi tạo 1 registry, mục đích của topic này là dùng để gửi các dữ liệu về **trạng thái hiện tại của thiết bị IoT** lên Cloud.

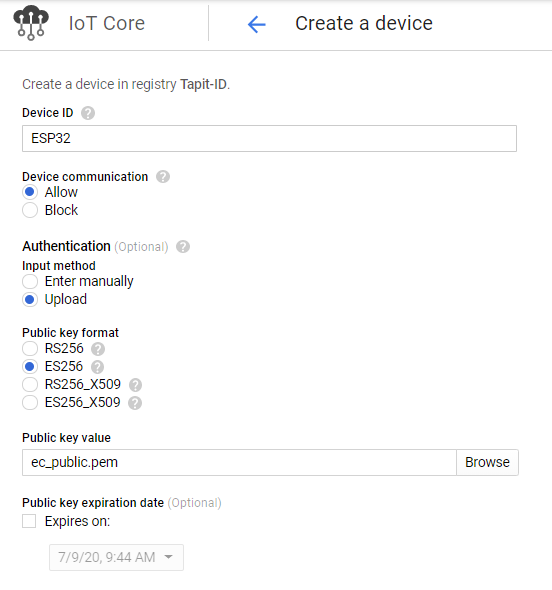
\* **Lợi ích của subFolder:** Giả sử bạn mong muốn thiết bị của bạn publish nhiều loại dữ liệu khác nhau lên cloud như nhiệt độ môi trường, độ ẩm hoặc dữ liệu logging,... Bằng cách hướng các luồng dữ liệu này vào từng subFolder riêng lúc cấu hình cho thiết bị của bạn, bạn có thể loại bỏ nhu cầu không cần phải tách dữ liệu thành các mục khác nhau sau khi dữ liệu đã ở trong Cloud Pub/Sub.

Ở đây mình demo 2 topic là **default telemtry topic**với path *projects/tapitdemo/topics/my-events* và **device state topic**với path *projects/tapitdemo/topics/state*. Sau đó nhấn chọn **CREATE***.*

Sau khi khởi tạo Registry xong, ở thanh menu registry của giao diện Console, chúng ta chọn **Devices***,* sau đó nhấn chọn **CREATE A DEVICE***.*



Tại giao diện đăng ký, các bạn có thể đặt tên bất kỳ cho Device ID nhưng lưu ý không được trùng tên với các Device ID khác trong registry, ở đây mình đặt tên Device ID là “ESP32”.



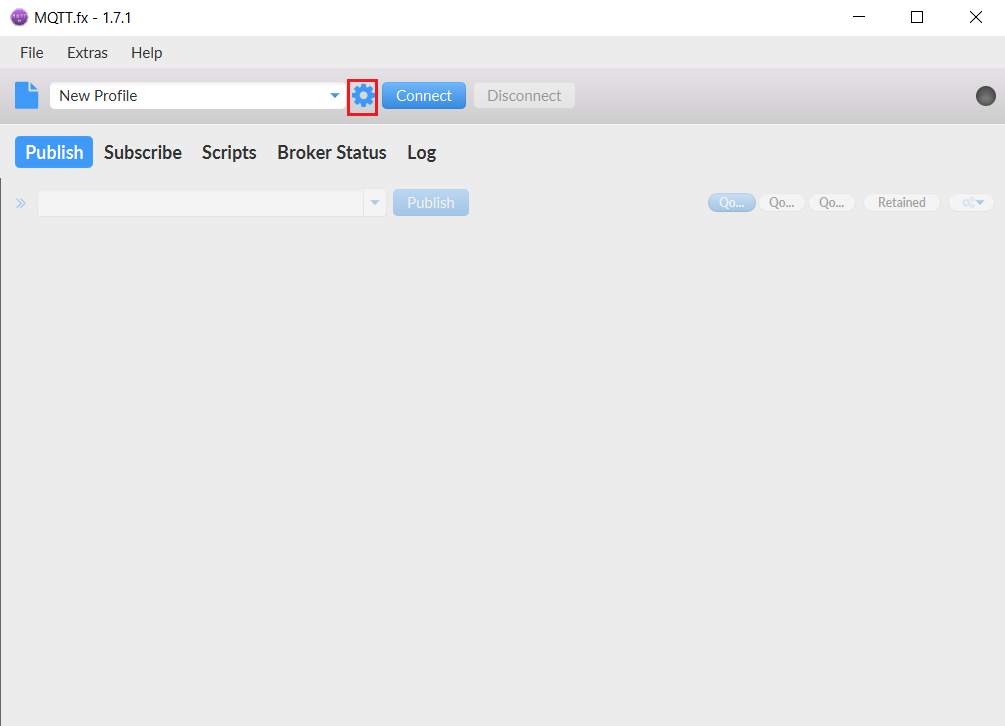
Ở phần **Authentication**, các bạn có thể chọn Input method là *Enter manually* hoặc *Upload*. Ở đây mình chọn là *Upload* vì file Public Key mình đã tải về máy rồi nên mình chỉ cần tải lại lên Google IoT Core là xong. Lưu ý phần *Public key format* các bạn chọn mục tương ứng với thuật toán đã khởi tạo Public key. Các thông số còn lại chúng ta không cần quan tâm, các bạn cuộn xuống dưới rồi nhấn **CREATE**.

**II. Hướng dẫn kết nối MQTT.fx với Google IoT Core thông qua giao thức MQTT**

Vậy là chúng ta đã biết cách làm thế nào để liên kết public key với 1 device cụ thể trên Google IoT Core. Bây giờ chúng ta sẽ đi vào việc thực hiện kết nối giữa thiết bị IoT với Cloud sử dụng cặp private key - public key tương ứng. Ở đây mình sẽ sử dụng công cụ MQTT.fx như là một MQTT Client để demo. Chúng ta có thể sử dụng các thông số của device-ID “ESP32” đã khởi tạo để bắt đầu, nhưng để trực quan cho các bạn mình sẽ tạo thêm 1 device-ID khác có tên là “MQTT” để kết nối.

Đầu tiên các bạn tải file cài đặt MQTT.fx về máy, sau khi thực hiện xong các bước cài đặt, chúng ta sẽ đi vào bước thiết lập cấu hình tham số sao cho MQTT.fx có thể kết nối được với Google IoT Core.

Tại giao diện MQTT.fx, các bạn nhấn vào biểu tượng răng cưa (biểu tượng setting).

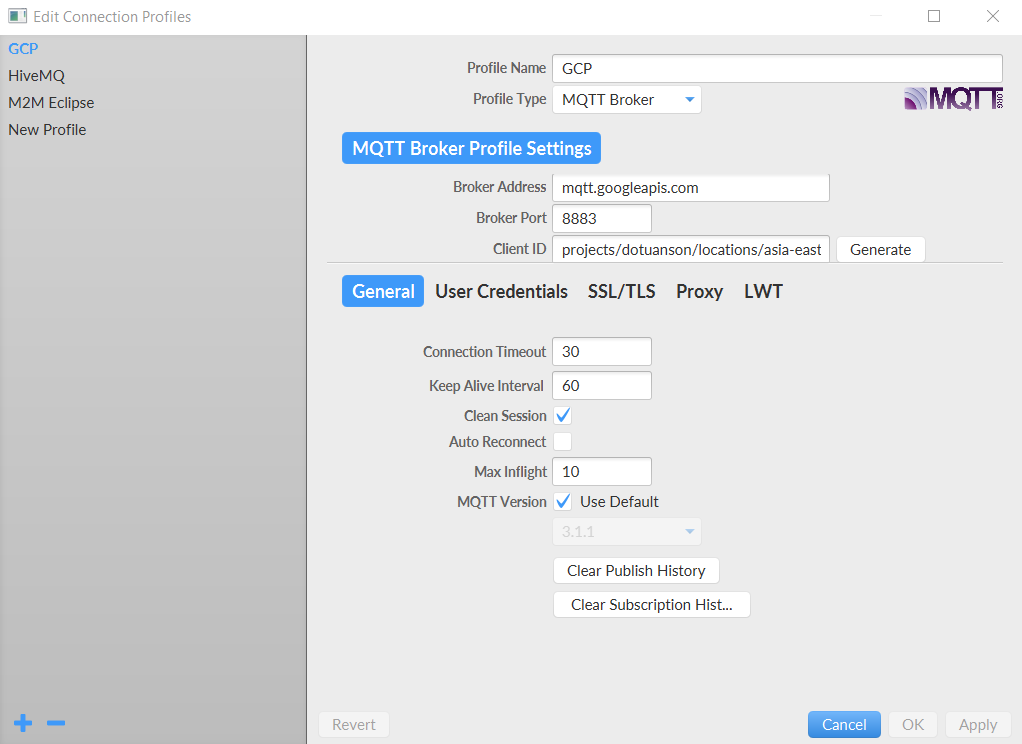


Ở thanh điều hướng bên trái, các bạn chọn New Profile và bắt đầu việc cấu hình.

* ***Profile Name***: Các bạn có thể đặt bất kỳ tên gì cho Profile Name, ở đây mình đặt là GCP.
* ***Profile Type***: Các bạn có thể chọn mục Google Cloud IoT cho việc cài đặt dễ dàng hơn, tuy nhiên nó sẽ bỏ qua bước khởi tạo JWT và Client ID. Vì mình muốn trực quan hơn cho mọi người nên mình sẽ demo mục MQTT Broker.
* ***Broker Address***: mqtt.googleapis.com
* ***Broker Port***: 8883 hoặc 443.
* ***Client ID***: Các bạn thay thế các thông số bôi đỏ dưới đây, bốn thông số này có thể tìm thấy trên Google IoT Core sau khi đã thiết lập device và registry như ở các bài viết trước.

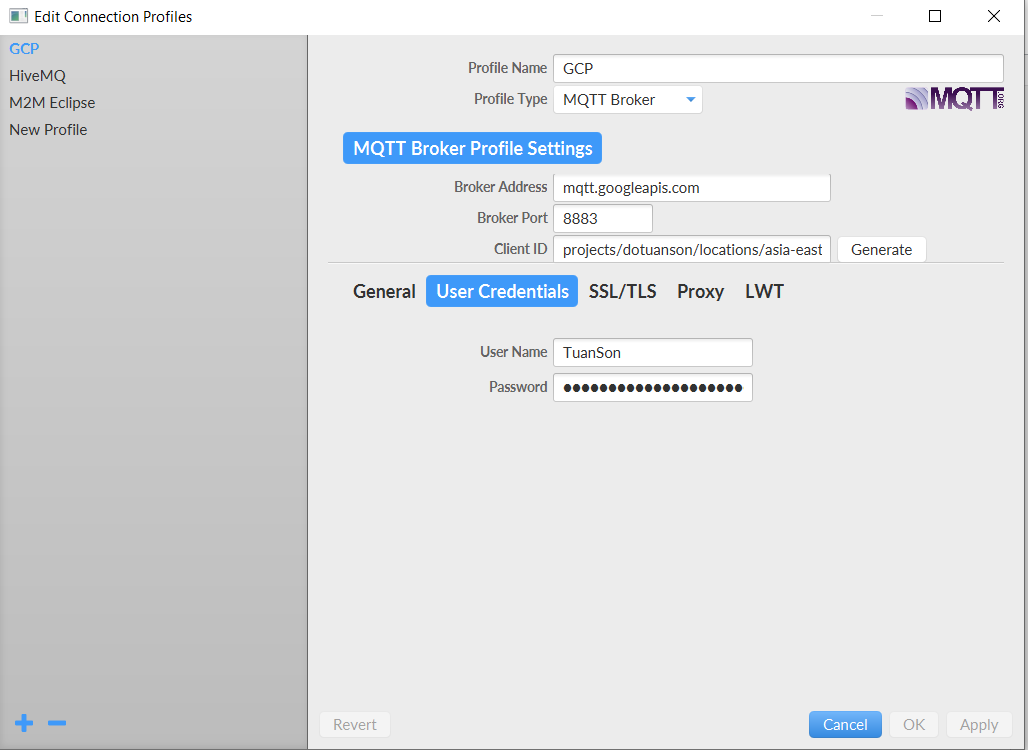
projects/**PROJECT-ID**/locations/**REGION**/registries/**REGISTRY-ID**/devices/**DEVICE-ID**

Các thông số mục **General** chúng ta tạm không quan tâm, chúng ta sẽ chuyển hướng sang mục khác có tên là **User Credentials.**

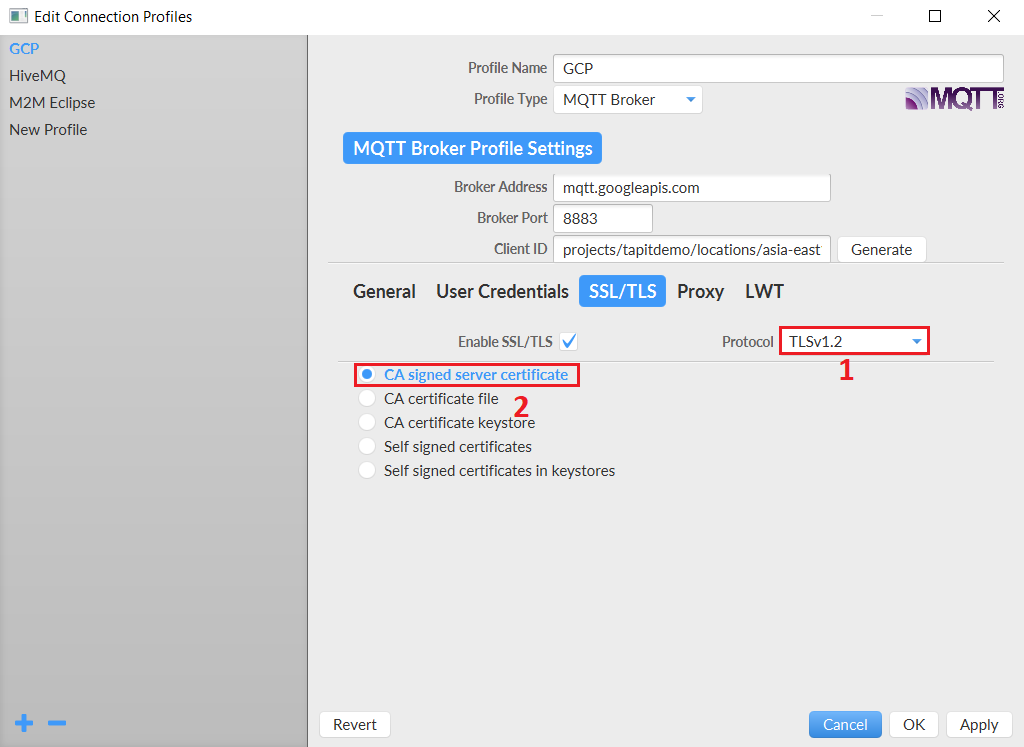


Ở mục **User Credentials**, chúng ta có thực hiện cấu hình như sau:

* ***User Name***: Các bạn có thể đặt tên bất kỳ, tại vì Google IoT Core không quan tâm trường này. Tuy nhiên, các bạn lưu ý **không được bỏ trống** mục này vì MQTT.fx quy định không cho phép gửi trường Password nếu như không có trường User Name, dẫn đến việc thực hiện kết nối bị lỗi.
* ***Password***: Copy chuỗi JWT và paste vào mục này. Lưu ý chuỗi JWT phải nằm trong khoản thời gian “iat” và “exp” (Trong trường payload) thì Google IoT Core mới chấp nhận chuỗi JWT này.



Cuối cùng chúng ta chuyển hướng sang mục **SSL/TLS**, hiện tại Google IoT Core chỉ hỗ trợ chuẩn **TLS1.2** nên ở mục **Protocol** chúng ta chỉ có thể chọn **TLSv1.2**. Tiếp theo ta **chọn CA signed server certificate**, đây là chứng chỉ tự server Google cấp cho chúng ta và Google sẽ tự xác thực chứng chỉ này mà không thông qua bên thứ ba nào.



Sau khi đã cấu hình xong, chúng ta nhấn **OK**. Bây giờ thì chúng ta đã hoàn toàn có thể thực hiện việc kết nối tới Google IoT Core thông qua giao thức MQTT. Để kết nối MQTT.fx tới dịch vụ Google IoT Core, các bạn nhấn chọn nút **CONNECT**.

Như vậy là qua phần 2 và 3, mình đã giới thiệu cho các bạn sơ bộ cách để cấu hình cũng như thiết lập các thông số của một client cụ thể khi muốn kết nối tới dịch vụ Google IoT Core thông qua giao thức MQTT. Trong phần tới, chúng ta sẽ đi sâu vào cơ chế hoạt động cũng như sử dụng hiệu quả sự tương tác giữa các thiết bị IoTs với dịch vụ đám mây này. Chúc các bạn thành công!