Trong phần trước, mình đã hướng dẫn các bạn cách để kết nối một MQTT Client cụ thể tới dịch vụ Google IoT Core thông qua các chứng chỉ bảo mật tự tạo. Trong phần này mình sẽ cho các bạn thấy khi một thiết bị IoT sử dụng giao thức MQTT kết nối đến Google IoT Core thì hai bên sẽ giao tiếp với nhau thế nào.

**Phần 4: Tương tác giữa các thiết bị IoT với Google IoT Core thông qua giao thức MQTT**

**I. Tương tác giữa MQTT Client với Google IoT Core**

Để thấy được sự tương tác giữa các thiết bị IoT với Google IoT Core hoạt động như thế nào thông qua giao thức MQTT. Chúng ta sẽ sử dụng một MQTT Client (Ở đây ta dùng MQTT.fx) để thực hiện việc kết nối và trao đổi dữ liệu với Cloud. Việc trao đổi dữ liệu này sẽ được thực hiện trung gian thông qua các topic.

Đầu tiên, chúng ta sẽ phân loại các kiểu dữ liệu do Google định nghĩa khi truyền từ thiết bị lên Cloud và từ Cloud truyền xuống thiết bị:

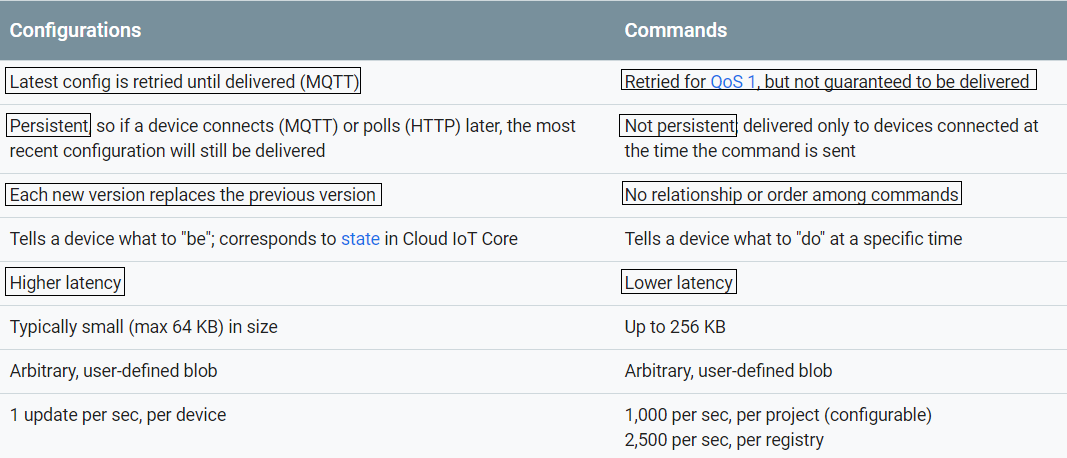
+ Thiết bị -> Cloud:

* *Telemetry*: Các dữ liệu về **môi trường xung quanh** mà thiết bị đo được như nhiệt độ, độ ẩm,...
* *State*: Các dữ liệu phản ánh **trạng thái hiện tại của bản thân thiết bị** như pin, ram trống, nhiệt độ chip,...

+ Cloud -> Thiết bị:

* *Configuration*: Các dữ liệu dùng để **cấu hình thiết bị** (Dựa vào dữ liệu state mà thiết bị báo cáo lên Cloud).
* *Command*: Các dữ liệu dùng để **ra lệnh** cho thiết bị thực hiện (Dựa vào dữ liệu telemetry mà thiết bị báo cáo lên Cloud).

\* So sánh giữa *Configurations và Commands:*



Từ các kiểu dữ liệu mà Google chia ra, chúng ta sẽ làm việc với 4 loại topics có path bắt đầu bằng /devices/DEVICE-ID/:

+ Các topic từ thiết bị -> Cloud:

* /devices/DEVICE-ID/events/{subFolders}
* /devices/DEVICE-ID/state

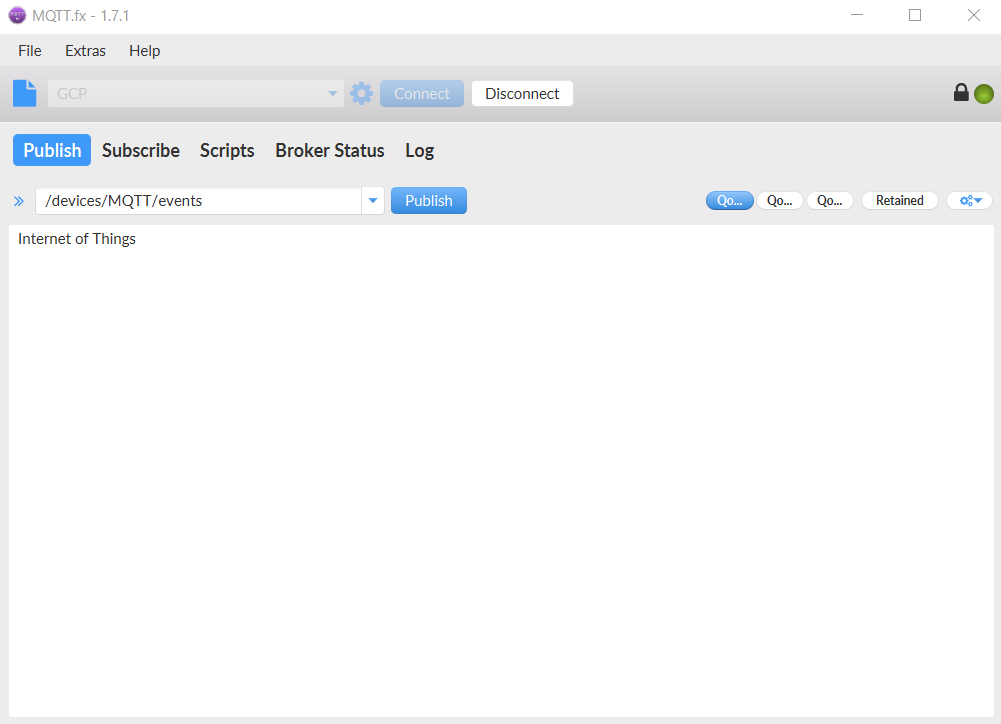
+ Các topic từ Cloud -> Thiết bị:

* /devices/DEVICE-ID/config
* /devices/DEVICE-ID/commands/#
* Ở thiết bị IoT, Google IoT Core chỉ cho phép chúng ta **publish** dữ liệu vào 2 topics là */devices/DEVICE-ID/events/{subFolders}* và */devices/DEVICE-ID/state*, nếu chúng ta publish dữ liệu ngoài 2 topics này thì giao thức MQTT sẽ ngay lập tức ngắt kết nối.
* Còn về **subscribe** thì Google IoT Core cũng chỉ cho phép chúng ta subscribe vào 2 topics là */devices/DEVICE-ID/config* và */devices/DEVICE-ID/commands/#* (bắt buộc phải có wildcard #). Bằng việc subscribe vào wildcard topic này, thiết bị IoT không chỉ nhận dữ liệu từ topic */devices/DEVICE-ID/commands* mà nó còn nhận được cả từ subfolder */devices/DEVICE-ID/commands/{subFolder}*. Lý do cho việc này là tại vì hiện tại Google IoT Core chưa hỗ trợ cho thiết bị có thể subscribe vào 1 subFolder cụ thể nào cả.

**II. Thao tác với các topic trong Google IoT Core**

**1. Publish message từ thiết bị IoT lên Cloud**

Chúng ta sẽ publish 1 gói tin bất kỳ tới topic */devices/DEVICE-ID/events/{subFolder}* hoặctopic */devices/DEVICE-ID/state:*



Sau khi nhấn chọn **Publish** thì gói tin sẽ thông qua giao thức MQTT mà gửi tới topic */devices/DEVICE-ID/events* nằm trong dịch vụ Google IoT Core. Tuy nhiên như mình đã nói ở bài viết trước trước, gói tin sẽ không được lưu trữ ở dịch vụ Google IoT Core, mà nó sẽ tự động chuyển tiếp và lưu trữ ở dịch vụ khác có tên là Cloud Pub/Sub. Các gói tin này sẽ lưu trữ trong Cloud Pub/Sub với topic có định dạng:

*project/PROJECT-ID/topics/{Topic you defined in registry}*

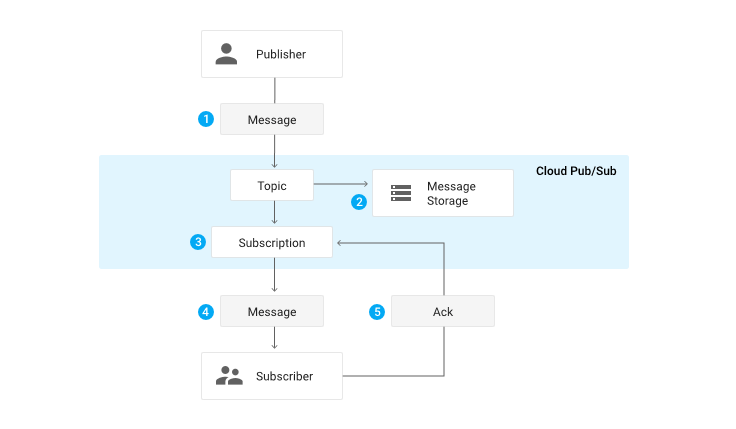
Tiếp theo, để có thể quan sát các gói tin mà thiết bị IoT gửi lên Cloud, chúng ta có các cách sau:

* Sử dụng gcloud command-line được tích hợp sẵn có trong [Cloud SDK](https://cloud.google.com/sdk/) của Google.
* Sử dụng các [thư viện client](https://cloud.google.com/apis/docs/cloud-client-libraries) được Google hỗ trợ như (java/python/nodejs/golang)
* Sử dụng **Console** của Cloud Pub/Sub để tạo **Topic** và **Subscription**

Để đơn giản, phần này mình sẽ sử dụng giao diện **Console** của Cloud Pub/Sub để demo cho các bạn xem các gói tin từ thiết bị gửi lên Cloud (Cụ thể thiết bị ở đây là phần mềm MQTT.fx).

Vì mặc định lúc chúng ta khởi tạo một registry trong Google IoT Core thì Google bắt buộc dịch vụ này phải liên kết với một hay nhiều topic khác ở trong dịch vụ Cloud Pub/Sub. Vậy nên khi thao tác với Google IoT Core thì chúng ta đã tự tạo ra các topic trong Cloud Pub/Sub. Việc của chúng ta bây giờ là tạo **Subscription** cho các topics này. **Subscription** chính là tên đại diện cho các dòng tin nhắn từ một hoặc nhiều topic cụ thể nào đó.

Vì bài viết này chỉ giới hạn trong nền tảng dịch vụ Google IoT Core, thế nên mình sẽ không trình bày quá sâu về Cloud Pub/Sub. Các bạn có thể tham khảo các bài viết trên mạng hoặc document của Google về Cloud Pub/Sub. Ở đây mình xin tóm tắt tổng quan về Cloud Pub/Sub để các bạn dễ hình dung.

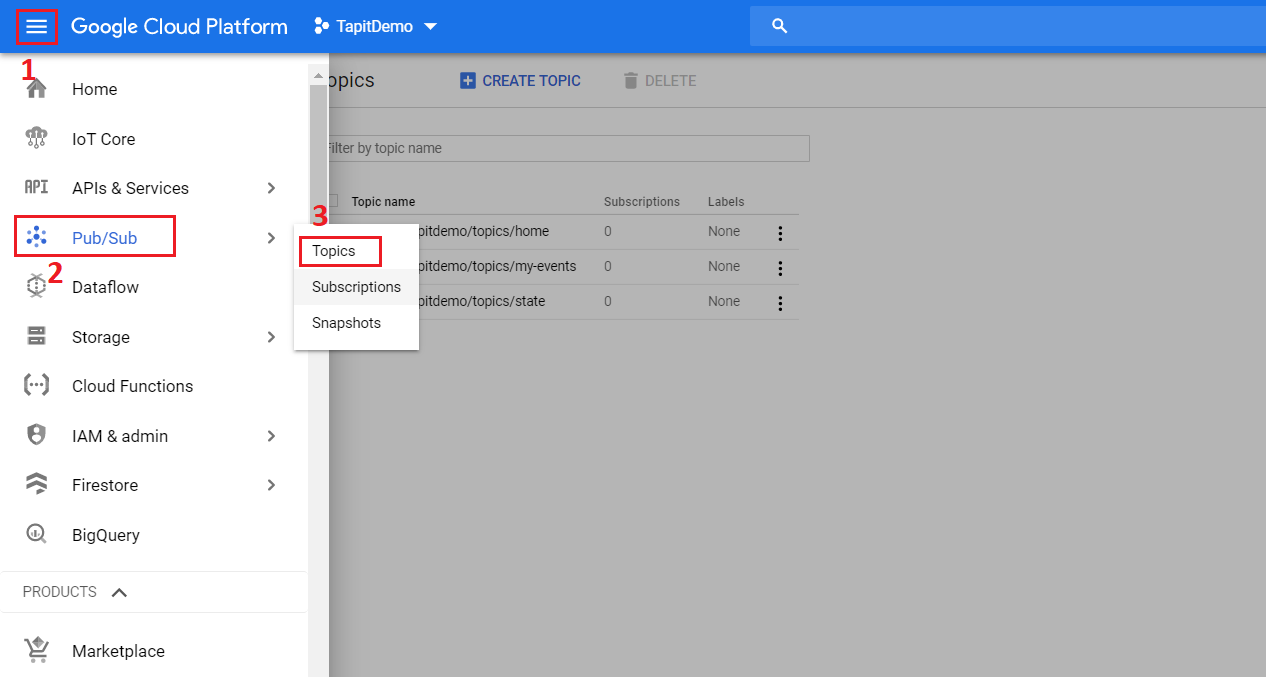
****

*Luồng xử lý message trong Cloud Pub/Sub*

1. **Pulisher**tạo **Topic**trêndịch vụ **Cloud Pub/Sub** và gửi messsagelên **Topic**.
2. **Message**sẽ được lưu trữ tại **Message Storage**cho đến khi tin nhắn được chuyển đến cho **subscribers**và nhận được phản hồi **(ACK)** từ **subscribers.**
3. **Cloud Pub/Sub**  ẽ chuyển tiếp messagetừ **topic**đến hàng đợi của tất cả **subscriptions**của nó. Mỗi **subscriptions**sau khi được tin nhắn, nó sẽ đẩy (**push**) **message**đó về cho **subscriber**. Hoặc **subscriber**có thể tự kéo (**pull**) **message**từ một **subscription**nào đó.
4. **Subscriber**nhận được messageđang chờ xử lý từ **subscription**.   
   **Subscriber**sẽ xác nhận (**ACK**) từng tin nhắn từ **subscription**.
5. Khi **subscription**nhận được tin nhắn xác nhận (**ACK**) của **message**đó thì thì **message**đó sẽ được xóa khỏi hàng đợi của **subscription.**

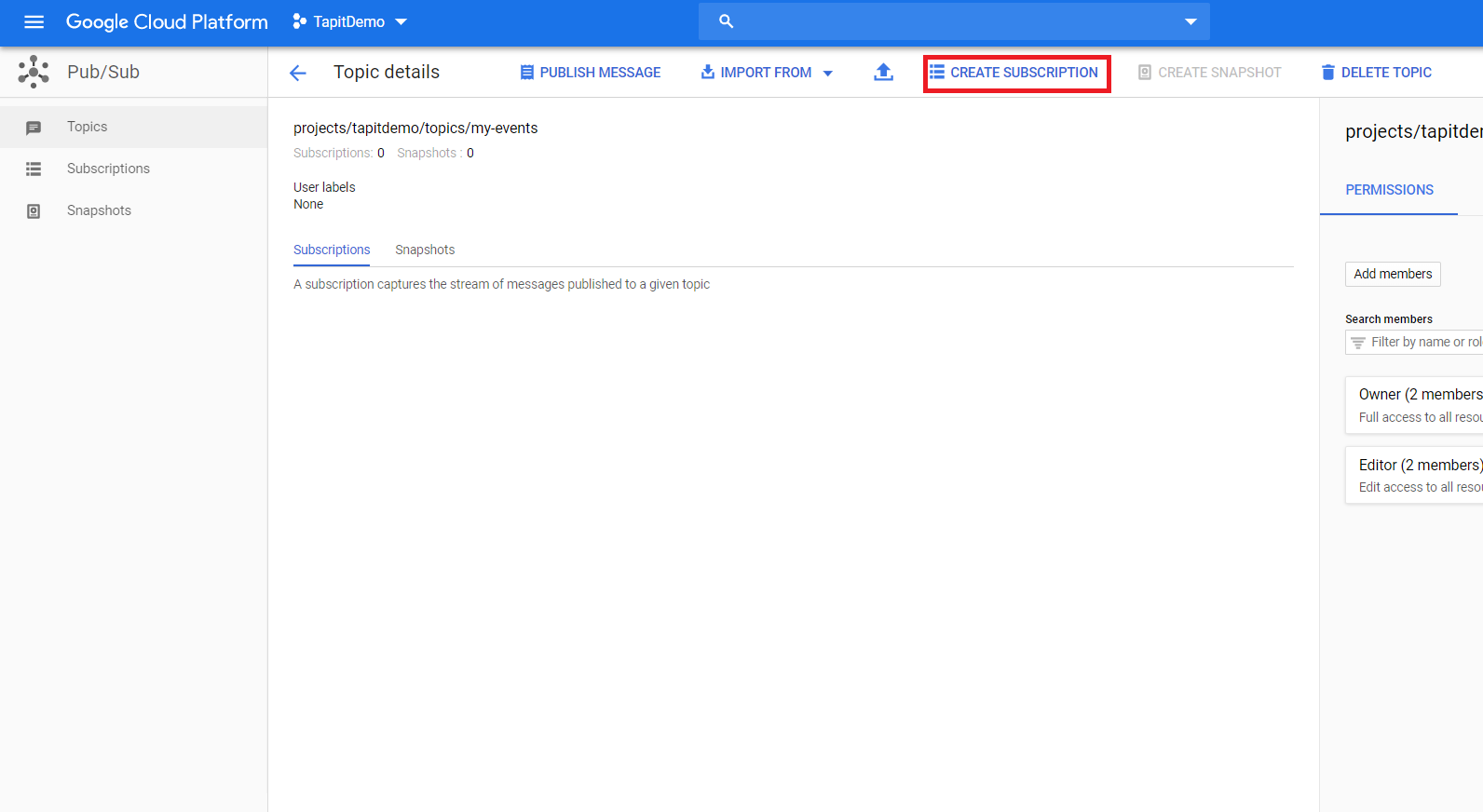
Sau khi tổng quan về mô hình của Cloud Pub/Sub, các bạn có thể dễ dàng thấy rằng để đọc được các gói tin trong các topic, các bạn cần tạo cho topic đó một hoặc nhiều subscription.

Tại giao diện Console, các bạn chuyển hướng tới dịch vụ Cloud Pub/Sub như sau:

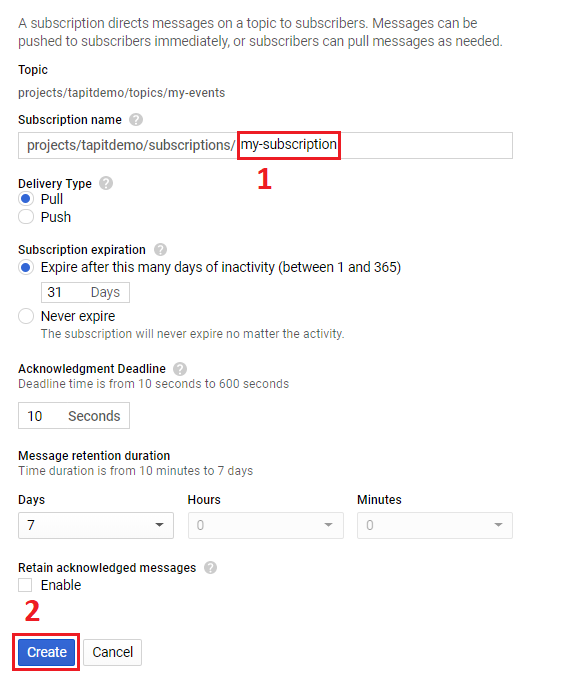


Giao diện Console sẽ hiển thị cho chúng ta tên các topic đã khởi tạo khi đăng ký registry trên Google IoT Core.

Ở đây mình có một topic tên là *projects/tapitdemo/topics/my-events* mà mình đã tạo ở phần trước, đây chính là topic trong Cloud Pub/Sub lưu trữ các dữ liệu *telemetry event* từ thiết bị IoT gửi tới, chúng ta sẽ nhấn chọn topic này. Ngay lập tức trang console sẽ trả cho ta thông tin về topic, tiếp tục các bạn chọn **CREATE SUBSCRIPTION** như hình dưới:



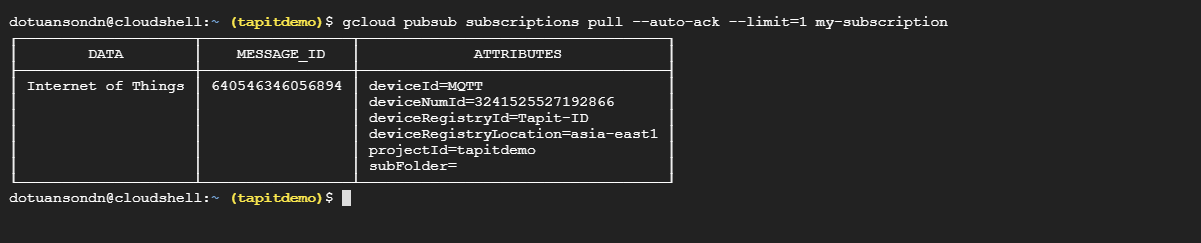
Các bạn có thể đặt bất kỳ tên gì ở trường **Subscription Name**, ở đây mình đặt là *my-subscription*. Các thông số khác các bạn không cần quan tâm, chúng ta sẽ cuộn xuống và nhấn chọn **CREATE**.



Sau khi đã khởi tạo xong subscription cho topic mà chúng ta muốn quan sát message. Các bạn mở **gcloud command-line** của google ra, sau đó các bạn điền command-line này vào gcloud:

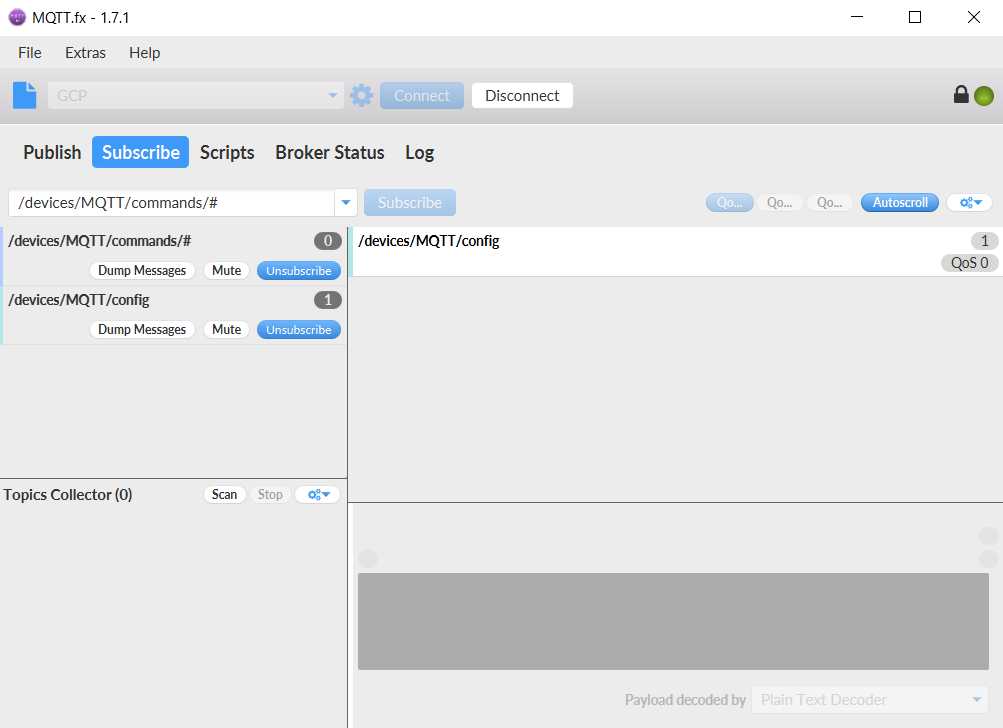
*gcloud pubsub subscriptions pull --auto-ack --limit=1 my-subscription*

Dòng lệnh này sẽ trả về cho chúng ta gói tin “Internet of Things” nằm trong *my-subscription* với các **ATTRIBUTES** và **MESSAGE\_ID** như sau:



**2. Subscribe message từ Cloud xuống thiết bị IoT**

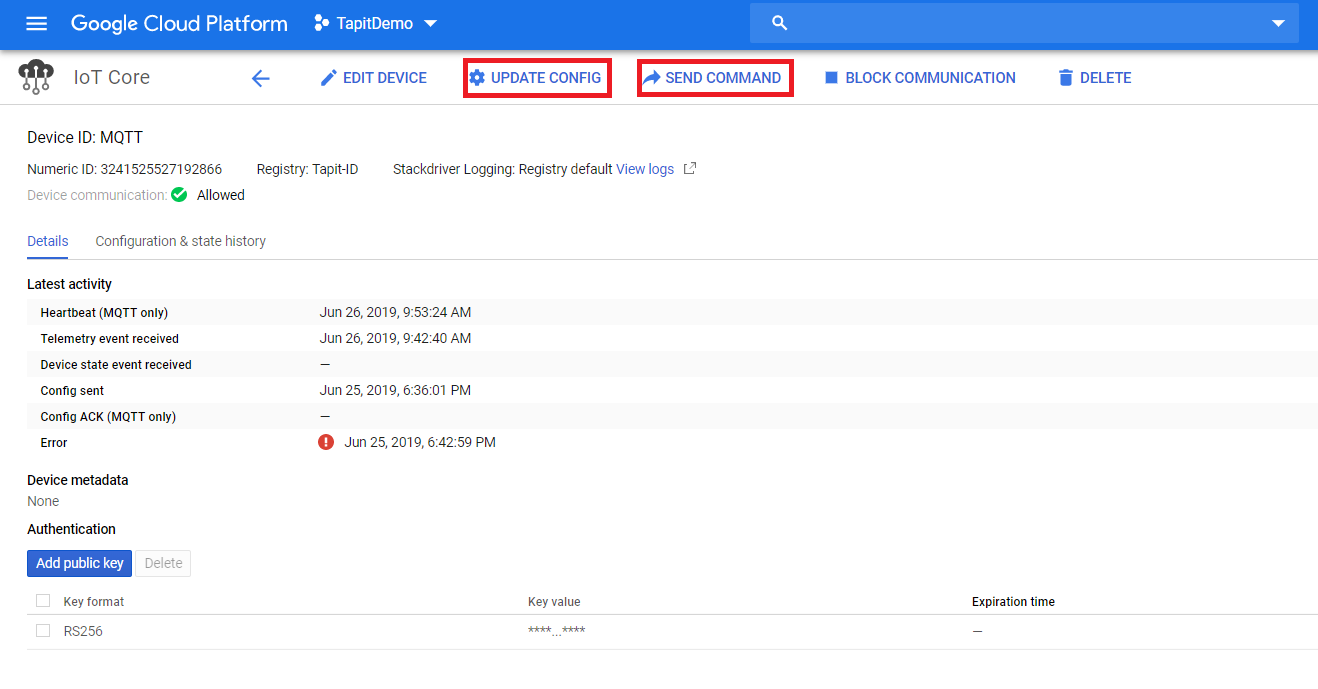
Google IoT Core cho phép MQTT Client subscribe 2 topics:



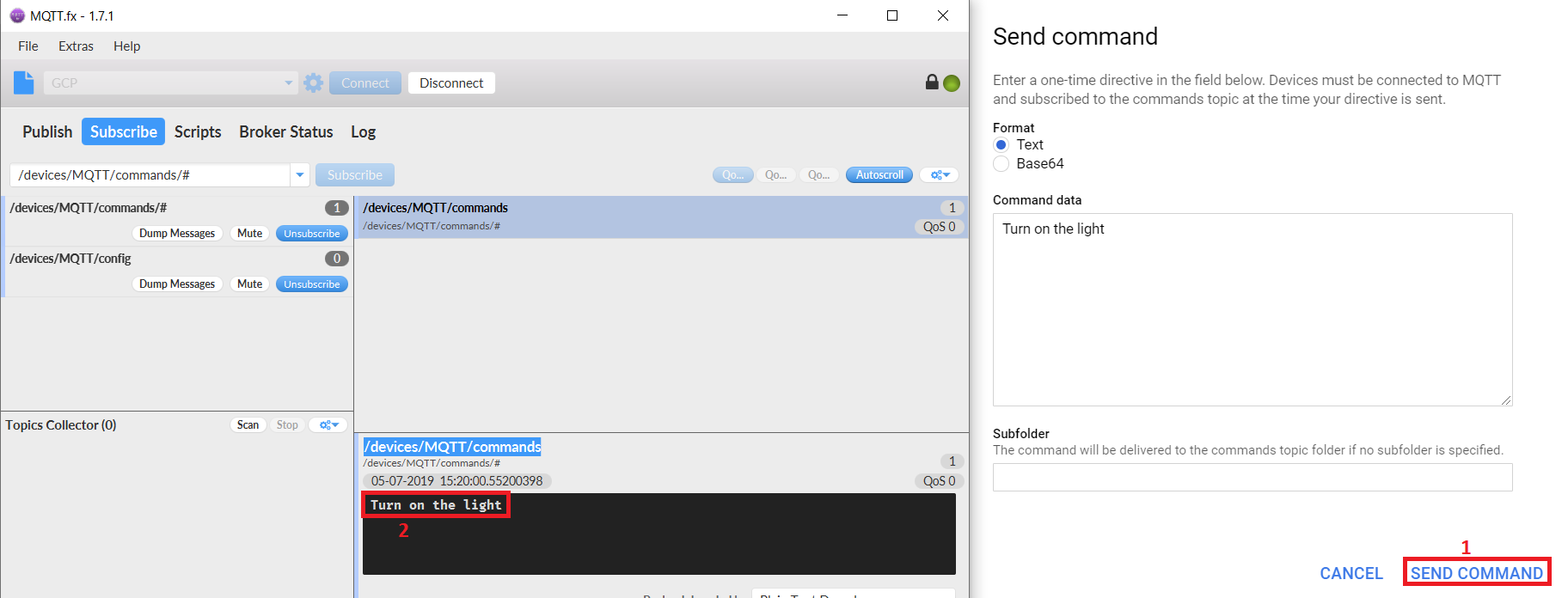
Để có thể gửi dữ liệu command hoặc config từ cloud xuống thiết bị của bạn, chúng ta có thể dùng **Console**, **gcloud** hoặc thông qua **API** mà Google cung cấp. Để trực quan, mình sẽ sử dụng **Console** của Google.

Đầu tiên chúng ta vào GCP Console, ở thanh điều hướng bên trái, chúng ta chọn dịch vụ **IoT Core**. Chọn **registry ID** mà chúng ta muốn thao tác. Ở thanh registry menu bên trái, chúng ta nhấn **Devices** và chọn **device ID** mà chúng ta muốn gửi command hoặc config.

Để gửi command hoặc config, chúng ta chọn các mục tương ứng như hình sau:



Ở đây mình ví dụ sau khi MQTT.fx đã subscribe vào topic có path */devices/MQTT/commands/#* thì khi ta **SEND COMMAND** gói tin “Turn on the light” từ Google IoT Core xuống cho MQTT.fx, chúng ta sẽ nhận được chính xác gói tin đó ở MQTT.fx



Từ phần một tới giờ chúng ta đã đi tìm hiểu về các khái niệm cũng như làm việc với các thiết bị IoT thông qua giao thức MQTT nhưng chỉ dừng lại ở mức demo một tool MQTT client. Trong phần tiếp theo, nhóm TAPIT IoTs sẽ hướng dẫn các bạn cách để kết nối một MCU là ESP32 tới dịch vụ Google IoT Core xem có gì khó khăn không nhé!