Comentario técnico: CTC-099
Componente: **Google Cloud Platform**
Autor: Sergio R. Caprile, Senior R&D Engineer

Revisiones	Fecha	Comentarios
0	23/06/20	

En el CTC-086 charlamos sobre la “Internet de *nuestras* cosas” y propusimos una solución escalable basada en MQTT, a la vez que recomendamos recurrir a los grandes proveedores para una red de envergadura. Analizaremos ahora una parte de la implementación de ese tipo de soluciones en uno de esos grandes proveedores. En esta oportunidad es Google, y nos referimos a la Google Cloud Platform. En particular, desarrollaremos la utilización de Cloud IoT Core, el producto de conectividad basado en MQTT y HTTP que nos permite concentrar la información proveniente de nuestros dispositivos.

Índice de contenido

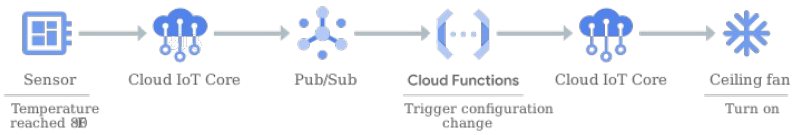
Breve descripción de GCP (desde nuestro punto de vista)....1
Breve descripción de Cloud IoT Core....2
MQTT bridge....3
Autenticación y seguridad....3
Tópicos....4
Manejo de datos....4
Eventos....4
Estado....4
Manejo de dispositivos....4
Configuraciones....4
Comandos....5
Estado....5
Actualizaciones de firmware....5
Forma de utilización....5
Conexión y operación desde un dispositivo....6
Recepción y utilización de datos de telemetría en el resto de la plataforma....6
Recepción y utilización de datos de estado en el resto de la plataforma....7
Cambio de configuración y envío de comandos en el resto de la plataforma....7
Autenticación y Seguridad en API....7
Logging....7
Esquema de precios....8

Breve descripción de GCP (desde nuestro punto de vista)

Google Cloud Platform es una plataforma de servicios (Google Cloud Services), los cuales se proveen mediante la red de datacenters de Google. Se trata mayormente de servicios de infraestructura de sistemas, mediante los cuales es posible desde servir páginas web hasta almacenar petabytes, y ejecutar aplicaciones.

Todos esos servicios cuentan con interfaces de modo de poder recibir y entregar sus datos de y a otros servicios, formando así toda una plataforma sobre la que podemos desarrollar nuestro sistema.[1]
Por otra parte, el tipo de sistemas que nos interesa como desarrolladores de productos de hardware y bajo la óptica de la Internet de las cosas (nuestras), es aquél con conectividad. Las APIs que presentan estos servicios están mayormente diseñadas para sistemas de gran envergadura, para los sistemas dedicados, y en particular el paraguas de la IoT, existe un servicio que concentra el ingreso de datos a la plataforma: Cloud IoT Core. Éste permite enviar y recibir mensajes para manejar dispositivos y recibir datos de telemetría, incorpora una base de dispositivos (*device registry*) y nos presenta un *protocol bridge* que acepta MQTT y HTTP para el diálogo con éstos. El vínculo entre ambos mundos, es decir, entre la red de dispositivos servida por Cloud IoT Core y toda la plataforma de sistemas, lo realiza Cloud Pub/Sub, un servicio que traduce esos datos al formato que manejan los demás servicios.
Sintetizando:

- Cloud IoT Core recibe los datos de telemetría y estado y maneja los dispositivos
- A través de Cloud Pub/Sub se desacoplan esos datos del protocolo de acceso y se inyectan en la plataforma
- A partir de aquí, por ejemplo, Google Cloud Functions permite ejecutar scripts en respuesta a una publicación, generando un flujo de datos hacia otro servicio como por ejemplo Firebase, BigTable, BigQuery, etc; o mismo generando una acción en un dispositivo, reingresando en Cloud IoT Core:



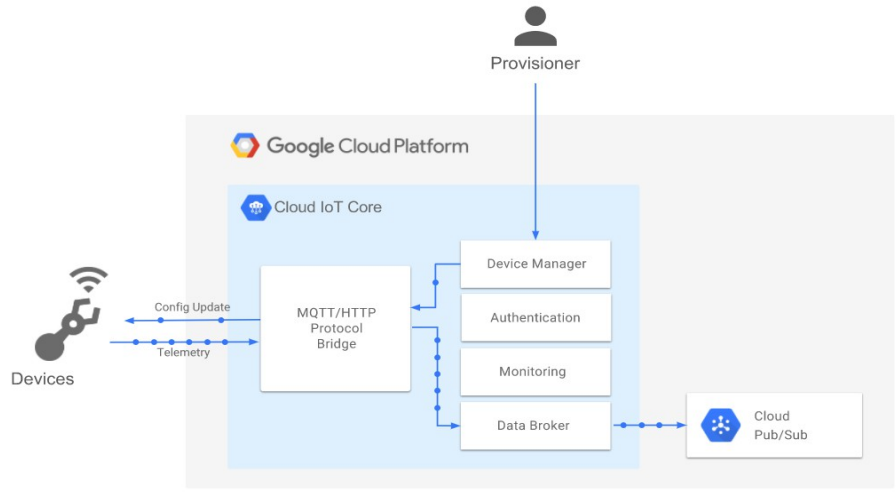
- Si la cantidad de datos a mover es muy elevada, por ejemplo, en vez de Cloud Functions podemos utilizar Cloud Dataflow para dirigir todo un flujo a un servicio

Una vez dentro de la plataforma, los datos pueden ser procesados por otros productos, analizados, graficados, o lo que necesitemos hacer con ellos con cualquiera de los productos disponibles en la plataforma, de los cuales los expertos de sistemas conocen mucho más de lo que podamos comentar aquí.

Breve descripción de Cloud IoT Core

Como comentamos, Cloud IoT Core[2] es el servicio de conectividad que nos permite concentrar la información proveniente de nuestros dispositivos y disponer de un modo de manejarlos. Como punto de acceso desde la red, cuenta con dos bridges para el acceso de dispositivos: MQTT y HTTP, aunque en el resto de este texto nos concentraremos solamente en MQTT[3]. Como centro de manejo de dispositivos, nos presenta una base de dispositivos, la *device registry*:

1 https://cloud.google.com/docs/overview
2 https://cloud.google.com/iot-core
3 para quienes deseen conocer más detalles sobre MQTT, remitimos al CTC-087



MQTT bridge

La implementación del bridge MQTT es ad hoc, es decir, no se trata de un broker abierto y de propósitos generales sino un puente de entrada a la plataforma, y como tal reviste sus características distintivas. Sintetizando, las diferencias al momento de escribir este documento son:

- No soporta retención ni persistencia
  - los mensajes no se retienen, aunque los que se entregan a Cloud Pub/Sub pueden ser retenidos allí
  - las sesiones inician sin estado previo
- Sólo se permiten tópicos prefijados
- La versión de MQTT requerida es 3.1.1
- No soporta QoS 2
- No soporta mensajes de última voluntad (LWT)

Autenticación y seguridad

Tanto en el bridge MQTT como en el HTTP, Cloud IoT Core usa autenticación por esquema asimétrico (clave pública/privada). En el caso particular de MQTT, al conectarse:

- El dispositivo usa su clave privada para firmar un JSON Web Token (JWT)[4], el cual pasa al broker como password y oficia de prueba de identidad.
- El broker verifica la autenticidad de la firma valiéndose de la clave pública, almacenada durante la creación del dispositivo en la *device registry*.
- La conexión es encriptada mediante TLS, el dispositivo autentica al broker mediante su certificado.[5]

Dado que entre la información intercambiada en el proceso de autenticación intervienen fecha y hora, se recomienda sincronizar los dispositivos con los servidores NTP de Google.

Opcionalmente, es posible además operar con TLS en modo de doble autenticación, aunque Google no provee certificados.[6]

La principal diferencia respecto a otros brokers es la utilización de JWT, esto del lado del dispositivo significa que en vez de un password fijo debemos disponer de una función que lo genere en el momento. Afortunadamente existe soporte al respecto.[7]

4 https://tools.ietf.org/html/rfc7519
5 https://cloud.google.com/iot/docs/concepts/device-security
6 https://cloud.google.com/iot/docs/how-tos/credentials/verifying-credentials
7 https://cloud.google.com/iot/docs/how-tos/credentials/jwts

Tópicos

Los únicos tópicos posibles son los siguientes, donde {device-id} corresponde al identificador del dispositivo como se lo crea en la *device registry*:

función	sentido	tópico
Telemetría	dispositivo → nube	/devices/{device-id}/events
Estado	dispositivo → nube	/devices/{device-id}/state
Configuración	nube → dispositivo	/devices/{device-id}/config
Comandos	nube → dispositivo	/devices/{device-id}/commands

Tanto el tópico de telemetría como el de comandos pueden tener subtópicos, a manera de carpetas (*subfolders*) para separar diversos tipos de información de un mismo dispositivo. En el caso de telemetría, es posible re-publicarlos en diferentes tópicos de Cloud Pub/Sub. En el caso de comandos, su utilidad es mayormente para el usuario.
El contenido de los mensajes es arbitrario, en tanto que es responsabilidad del usuario entenderlo del otro extremo. Es decir, la plataforma es transparente tanto para lo que se reporta al sistema como para lo que se envía al dispositivo, se considera binario y no se analiza.
La separación y sentido responde además a particularidades en el manejo de los datos y de los dispositivos, que desarrollamos a continuación.

Manejo de datos

Cloud IoT Core recibe de los dispositivos dos tipos de datos: datos de telemetría, denominados “eventos” (*events*), y datos de estado del dispositivo. Ambos reciben un tratamiento bastante diferente.

Eventos

Este tipo de datos es inmediatamente publicado en Cloud Pub/Sub para su utilización por el resto de la plataforma. No son retenidos dentro de Cloud IoT Core pero pueden serlo en Cloud Pub/Sub. Son utilizados para reportar mediciones y datos que cambian de manera frecuente.

Estado

Este tipo de datos es retenido dentro de Cloud IoT Core hasta un máximo de diez, es decir, mantiene los diez últimos. Son utilizados para reportar cambios en el estado del dispositivo, situaciones que no ocurren de manera frecuente. Opcionalmente pueden además ser publicados en Cloud Pub/Sub para su utilización por el resto de la plataforma; esto ocurre sin garantía de entrega.

Manejo de dispositivos

El manejo de los dispositivos se hace a través de una *device registry*; una base de dispositivos con características afines. Allí podemos crear y borrar dispositivos, guardar información adicional (metadatos), modificar y guardar su configuración, y enviar comandos.[8]

Configuraciones

Una configuración es una cantidad de datos definida de manera arbitraria por el usuario, que se mantiene almacenada en Cloud IoT Core. El contenido no es procesado, pero puede visualizarse y modificarse en la

8 https://cloud.google.com/iot/docs/concepts/devices

consola. La plataforma almacena las últimas diez versiones y es posible seleccionar cualquiera de ellas como activa, siendo enviada al dispositivo cuando éste se conecta.
Una configuración es algo que indica al dispositivo qué debe ser, y guarda una cierta correspondencia con su estado. Se espera que esto no cambie de manera frecuente, siendo su velocidad de actualización limitada al orden del segundo para cada dispositivo. Al momento de escribir este texto, el tamaño máximo de una configuración es de 64KB.

Comandos

Un comando es también una cantidad de datos definida de manera arbitraria por el usuario cuyo contenido no es procesado; pero no es almacenada. Cloud IoT Core envía el comando en el momento de ser solicitado si el dispositivo está conectado, y se pierde si no lo está.
Un comando es algo que indica al dispositivo qué debe hacer en un momento determinado, es algo transitorio y no debe ser parte de algo que persiste en el dispositivo. De igual modo, dada la forma en que son enviados, el operador no debe esperar que el resultado del comando se refleje en datos de estado o telemetría.
La frecuencia de actualización de los comandos está limitada de manera global al orden de 1000 por segundo.[9] Al momento de escribir este texto, el tamaño máximo de un comando es de 256KB.

Estado

Como comentamos, la plataforma almacena los últimos diez estados reportados. Es posible obtenerlos accediendo a la misma.

Actualizaciones de firmware

Esto en sí no es manejado por la plataforma y debe realizarse mediante el esquema soportado, es decir, comandos y configuraciones. La plataforma no contempla otra forma de acceder a los dispositivos, pero tampoco impide que éstos realicen conexiones a otros destinos ni acepten conexiones de otros entes. La infraestructura adicional necesaria corre por cuenta del usuario.

Forma de utilización

Antes de comenzar, recomendamos enfáticamente olvidarse de todos los tutoriales y artículos que se encuentran diseminados por la Internet. Para comprender la forma de trabajar con esta plataforma desde Cloud IoT Core, simplemente hay que seguir el inicio rápido (*quickstart*) oficial de Google para Cloud IoT Core.[10] Clarificaremos a continuación algunos puntos que nos han suscitado dudas

- Proyectos
  - Un proyecto es la unidad administrativa que engloba a todos los otros elementos, lo primero que necesitamos hacer es crear un proyecto
- Asegurarse de que la facturación está habilitada
  - Si estás en el período de prueba, es decir, has aceptado la sugerencia de probar la plataforma, lo está. No es necesario buscarlo.
- Habilitar las APIs de Cloud IoT Core y Cloud Pub/Sub (*Enable the Cloud IoT Core and Cloud Pub/Sub APIs)*
  - Solamente debemos seleccionar que las queremos usar y se habilitan, no es necesario que creemos ninguna credencial de acceso todavía. Esto se hará en una red real cuando configuremos el acceso desde otros productos para generar las operaciones sobre los dispositivos que nuestra aplicación requiera. Por el momento, no lo necesitamos.
- Configurar el entorno y requisitos previos a la instalación del software

9 https://cloud.google.com/iot/docs/how-tos/commands
10 https://cloud.google.com/iot/docs/quickstart

    - prestar atención a la versión de Python
        - en nuestro caso tenemos varias versiones y el instalador reconoció sin problemas la que le sirve, pero por las dudas deberíamos saber cuál o cuáles tenemos
    - No es necesario instalar ningún paquete adicional, solamente necesitamos gcloud para lo que vamos a hacer con Cloud IoT Core
- Al crear la *device registry*
    - creamos el tópico para eventos, las claves son gestionadas por Google (*Google managed key*)
    - para ver las opciones de MQTT que indica el *quickstart* debemos seleccionar opciones avanzadas (*Advanced topics*)
    - podemos habilitar logs por *registry* o por dispositivo, por ahora no lo haremos
- Cuando se nos pregunte la región que queremos usar... la elección de la zona es una cuestión de cercanía, disponibilidad, costo. Elegimos `us-central1` para las pruebas

A continuación, el *quickstart* nos indica un ejemplo en JavaScript que podemos clonar de github para simular un dispositivo, y luego finaliza indicándonos cómo observar la información almacenada en Cloud Pub/Sub.

Conexión y operación desde un dispositivo

Llegando al final, el *quickstart* nos indica un ejemplo en JavaScript que podemos clonar de github. Dicho ejemplo

- se conecta al broker
- publica datos en el tópico de telemetría (eventos)
- puede publicar datos en el tópico de estado
- se suscribe al tópico de configuración solicitando QoS=1
- se suscribe al tópico de comandos solicitando QoS=0

En síntesis, tiene toda la información necesaria para operar con la plataforma, ya que en esencia eso es lo que debemos hacer en un dispositivo; el resto corresponde a lo que nuestra aplicación requiera. A fines ilustrativos, el ejemplo realiza lo siguiente:

```
Google Cloud IoT Core MQTT example.
connect
Publishing message: my-registry/my-device-payload-1
Config message received:
Config message received:
Publishing message: my-registry/my-device-payload-2
Publishing message: my-registry/my-device-payload-3
[...]
Publishing message: my-registry/my-device-payload-24
Publishing message: my-registry/my-device-payload-25
Closing connection to MQTT. Goodbye!
close
```

Para más datos sobre la forma de conexión y envío de datos de telemetría y estado por un dispositivo, podemos consultar la documentación.[11]

Recepción y utilización de datos de telemetría en el resto de la plataforma

Como hemos comentado, los datos de telemetría son publicados en Cloud Pub/Sub y debemos extraerlos de allí con el servicio que deseemos utilizar, suscribiéndonos al tópico que resultó configurado al crear la *device registry*. Al final del *quickstart* nos encontramos con un ejemplo de cómo utilizar *gcloud* (el comando instalado por el SDK) para obtener dichos datos. En esencia, lo que hace *gcloud* es utilizar el API de Cloud Pub/Sub para traer los mensajes desde la nube a nuestra computadora. Observaremos un diagrama con los campos del mensaje:

11 https://cloud.google.com/iot/docs/how-tos/mqtt-bridge

DATA	MESSAGE_ID	ATTRIBUTES	DELIVERY_ATTEMPT
my-registry/my-device-payload-1	1265880308770517	deviceId=my-device deviceNumId=2792707136353885 deviceRegistryId=my-registry deviceRegistryLocation=us-central1 projectId=test-gcp-280019 subFolder=	

Vemos que el campo DATA contiene el mensaje enviado por nuestro dispositivo (observar lo que envió el ejemplo), junto con información administrativa adicional.
En el campo ATTRIBUTES, subFolder incluirá cualquier subtópico que hayamos utilizado.
Probablemente nos haya llamado la atención que no hay *timestamps*, información de fecha y hora. Es posible obtenerla en Cloud Functions del contexto de una función al momento de ser invocada, disparada por un mensaje en un tópico configurado.

Recepción y utilización de datos de estado en el resto de la plataforma

Como hemos mencionado, Cloud IoT Core almacena datos de estado y opcionalmente puede publicarlos (a mejor esfuerzo, sin garantías[12]) en Cloud Pub/Sub.
Para obtenerlos de Cloud Pub/Sub, nos suscribiremos en nuestro producto al tópico que hayamos configurado.
Para obtenerlos de Cloud IoT Core, podemos hacerlo manualmente a través de la Cloud Console, de *gcloud*, o desde otros servicios/aplicaciones utilizando la API.[13]

Cambio de configuración y envío de comandos en el resto de la plataforma

Desde el punto de vista del dispositivo, se trata de información que le llega por uno de los mencionados tópicos.
Desde el punto de vista de la aplicación, se trata de modificar el contenido de la configuración almacenada en la *device registry* y de mandar un mensaje determinado, respectivamente.
Interactivamente, podemos hacerlo a través de la Cloud Console o de *gcloud*. Desde otros servicios/aplicaciones podemos hacerlo utilizando la API. En ambos casos, la documentación oficial incluye ejemplos en varios lenguajes de programación.[14][15]

Autenticación y Seguridad en API

El acceso mediante la API utiliza un sistema de autenticación y autorización denominado Cloud Identity and Access Management (IAM)[16], es posible fijar roles y asignar credenciales que permiten regular el acceso.

Logging

Hemos visto, aunque no lo desarrollamos, que es posible generar logs en la *device registry* registrando el acceso y posibles errores en las acciones de los dispositivos sobre Cloud IoT Core. Para más información podemos consultar la documentación oficial.[17] Es posible también enviar logs al servicio Cloud Logging.[18]
Existe además un servicio adicional, Google Cloud's Operations Suite (anteriormente llamado Stackdriver), diseñado para realizar el monitoreo de toda nuestra infraestructura.[19] Es posible enviar datos de log desde los dispositivos a este producto utilizando Cloud Functions, y por ejemplo identificando los eventos de log publicándolos con un subtópico en particular.

12 El dato se publica en el tópico mencionado, pero si nadie lo reclama, no se reintenta la publicación.
13 https://cloud.google.com/iot/docs/how-tos/config/getting-state
14 https://cloud.google.com/iot/docs/how-tos/config/configuring-devices
15 https://cloud.google.com/iot/docs/how-tos/commands
16 https://cloud.google.com/iot/docs/how-tos/iam
17 https://cloud.google.com/iot/docs/how-tos/device-logs
18 https://cloud.google.com/logging
19 https://cloud.google.com/products/operations

Esquema de precios

Si bien esto parece algo fuera de lugar, es información importante a la hora de decidir qué reportamos y cómo lo hacemos. La razón es que cada proveedor cobra por el servicio de una forma diferente, y Google no es la excepción. La forma en que el proveedor identifica la unidad de tarifado tiene influencia en cómo deberemos empaquetar la información en caso que deseemos minimizar el costo. Nos referimos aquí solamente a Cloud IoT Core, otros productos tienen su esquema e incluso hay disponible un estimador para poder tener una idea de los costos involucrados en nuestro proyecto.[20] Los siguientes son algunos detalles al momento de escribir este texto:

- La unidad mínima es de 1024 bytes. Si enviamos mensajes más largos, pagamos de manera proporcional, pero si enviamos mensajes más cortos, Google nos cobrará 1024 bytes. Esto a simple vista parece poco importante pero a fin de mes ese espacio que no usamos se multiplica considerablemente dada la gran cantidad de mensajes que se envía y es necesario tenerlo en cuenta.
- Las conexiones y suscripciones se cobran como un mensaje.
- Las confirmaciones de recepción (ACKs) de MQTT se cobran como un mensaje, por ejemplo enviar una configuración de 10 bytes genera cargos por 2048 bytes (1024 por el mensaje y 1024 por el ACK).
- Los pings usados para mantener viva la conexión, se cobran como un mensaje; poner un tiempo más corto para responder rápidamente ante una desconexión, se paga.
- No se cobra si el tráfico mensual es inferior a los 250MB.
- Enviar entre 250MB y 250GB mensuales cuesta entre US$ 1,125 y US$ 1125, respectivamente.

Para más detalles se sugiere consultar la documentación oficial.[21]

20 https://cloud.google.com/pricing/list
21 https://cloud.google.com/iot/pricing