

# 物聯網技術與應用

Internet of Things:  
Technologies and Applications

LAB4 :  
Google Cloud







































# Outline

- The Introduction of Google Cloud Platform
- The Google Cloud IoT Core
- Using IoT Core gateways with a Pi
- Assignment

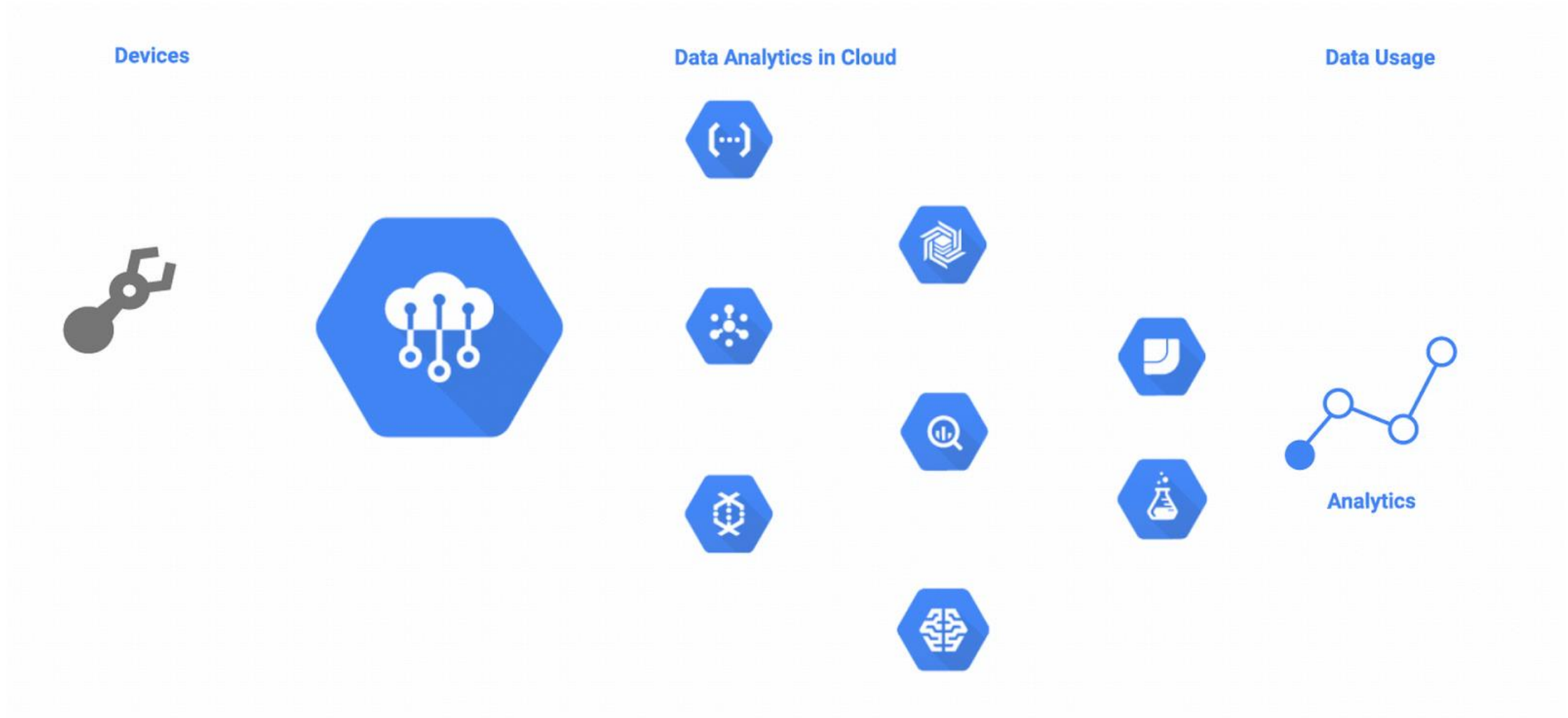
# Google Cloud Platform



Google Cloud

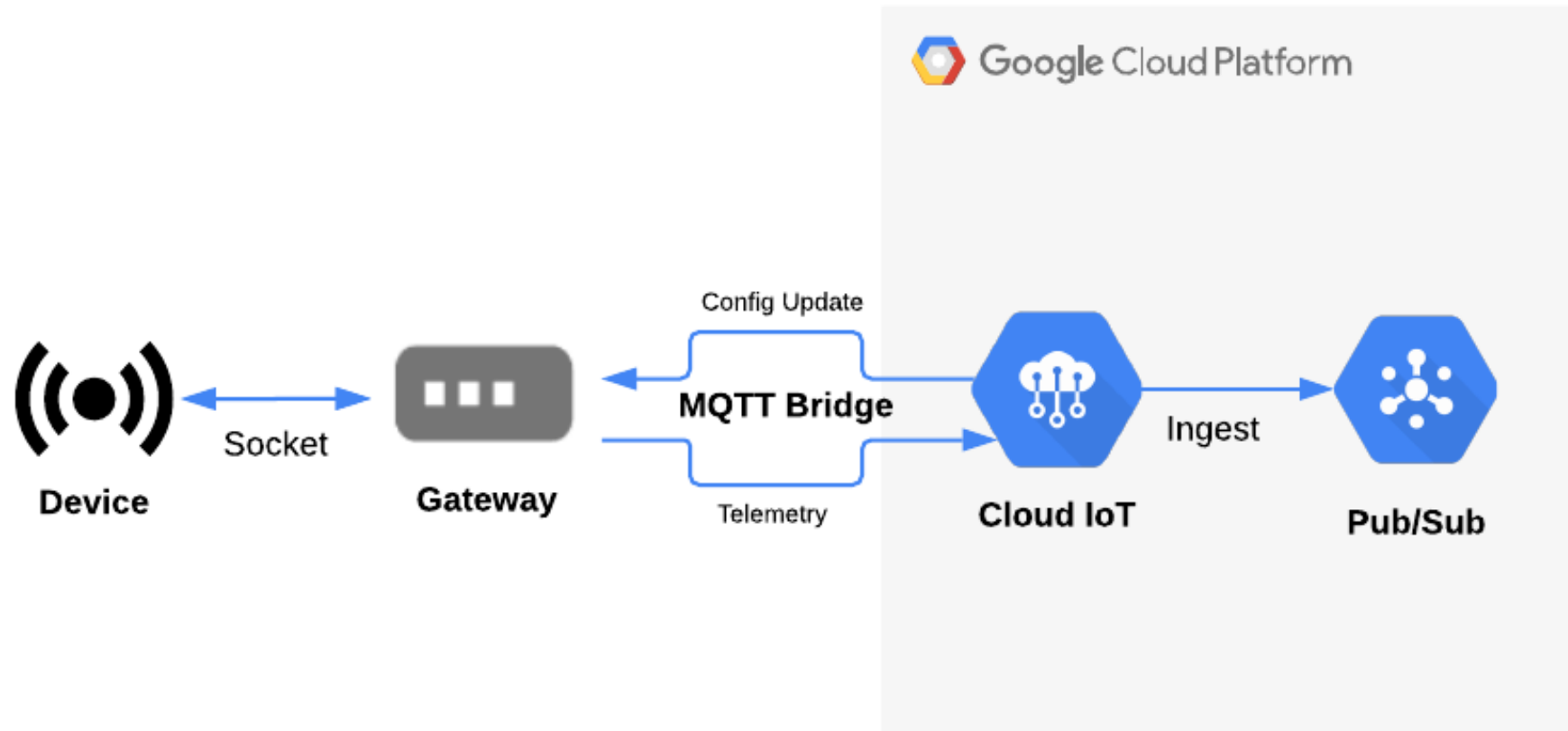
Compute						
	Compute Engine	Kubernetes Engine	App Engine	Cloud Functions		
Management						
	Cloud Console	Stackdriver	Trace	Logging	Debugger	Monitoring
Networking						
	Cloud Load Balancing	Cloud CDN	Cloud DNS	Firewall Rules	Cloud Interconnect	Cloud VPN
Storage & Databases						
	Cloud Bigtable	Cloud Datastore	Cloud Spanner	Cloud SQL	Cloud Storage	
Big Data						
	BigQuery	Cloud Dataflow	Cloud Dataprep	Cloud Dataproc	Cloud IoT Core	Cloud Pub/Sub
Identity & Security						
	Cloud IAM	Cloud Endpoints	VPC	Identity Aware Proxy	KMS	Data Loss Prevention
Machine Learning						
	Cloud ML	Natural Language API	Cloud Speech API	Cloud Vision API	Cloud Translate API	

# Google Cloud IoT Core



# Using IoT Core gateways with a Pi

## ➤ Architecture



# Using IoT Core gateways with a Pi

- Create project
- Enable IoT Core and Pub/Sub APIs
- Create registry
- Set up your gateway
- Create Device
- Managing devices through configuration updates

# Create project (Browser)

## Step 1

Google Cloud

選用 Google 的理由 解決方案 產品 定價 開始使用 聯絡我們

🔍 📄 文件 支援 中文 - 繁體 控制台

### 打造新一代的產品與服務。軟體強化，速度升級。

- ✓ 使用 Google 的核心基礎架構，以及資料分析和機器學習技術
- ✓ 安全無虞且功能完善，適合所有企業使用
- ✓ 致力提供開放原始碼環境與領先業界的成本效益

聯絡銷售人員 前往主控台

#### 具有前瞻性的基礎架構

透過 Google 的基礎架構為使用者提供快速、可靠且安全無虞的服務。

#### 強大的資料技術和分析能力

使用 Google Cloud 的資料分析產品，輕鬆地擷取、管理、處理資料，並透過圖表加以呈現。

#### 採用無伺服器技術：讓您專心編寫程式碼

從原型設計到實際推出，您都不必擔心負荷能力、穩定性或效能方面的問題。

提供建構產品與服務及調度資源所需的一切要素

# Step 2

Google Cloud Platform my-project

資訊主頁 活動 建議

專案資訊

- 專案名稱: my-project
- 專案 ID: [redacted]
- 專案編號: [redacted]

[將使用者新增至這項專案](#)

[前往專案設定](#)

資源

這項專案沒有資源

追蹤項目

過去 7 天內沒有任何追蹤資料

[開始使用 Trace](#)

開始使用

- API 瀏覽及啟用 API
- 部署預先建立的解決方案
- 新增動態記錄到執行中的應用程式
- 透過錯誤報告功能監控錯誤
- 部署 Hello World 應用程式

API API

要求 (每秒要求數)

1.0/s  
0.8/s  
0.6/s  
0.4/s  
0.2/s  
0

9:30 9:45 10 下午 10:15

[前往 API 總覽](#)

Google Cloud Platform 服務狀態

所有服務狀態皆正常

[前往 Cloud 狀態資訊主頁](#)

計費功能

預估費用 TWD \$0.00

帳單週期: 2021/10/1 至 2021/10/26

計費方式導覽

[查看詳細扣款記錄](#)

Monitoring

- 建立我的資訊主頁
- 設定快訊政策
- 建立運作時間檢查

[查看所有資訊主頁](#)

[前往 Monitoring](#)

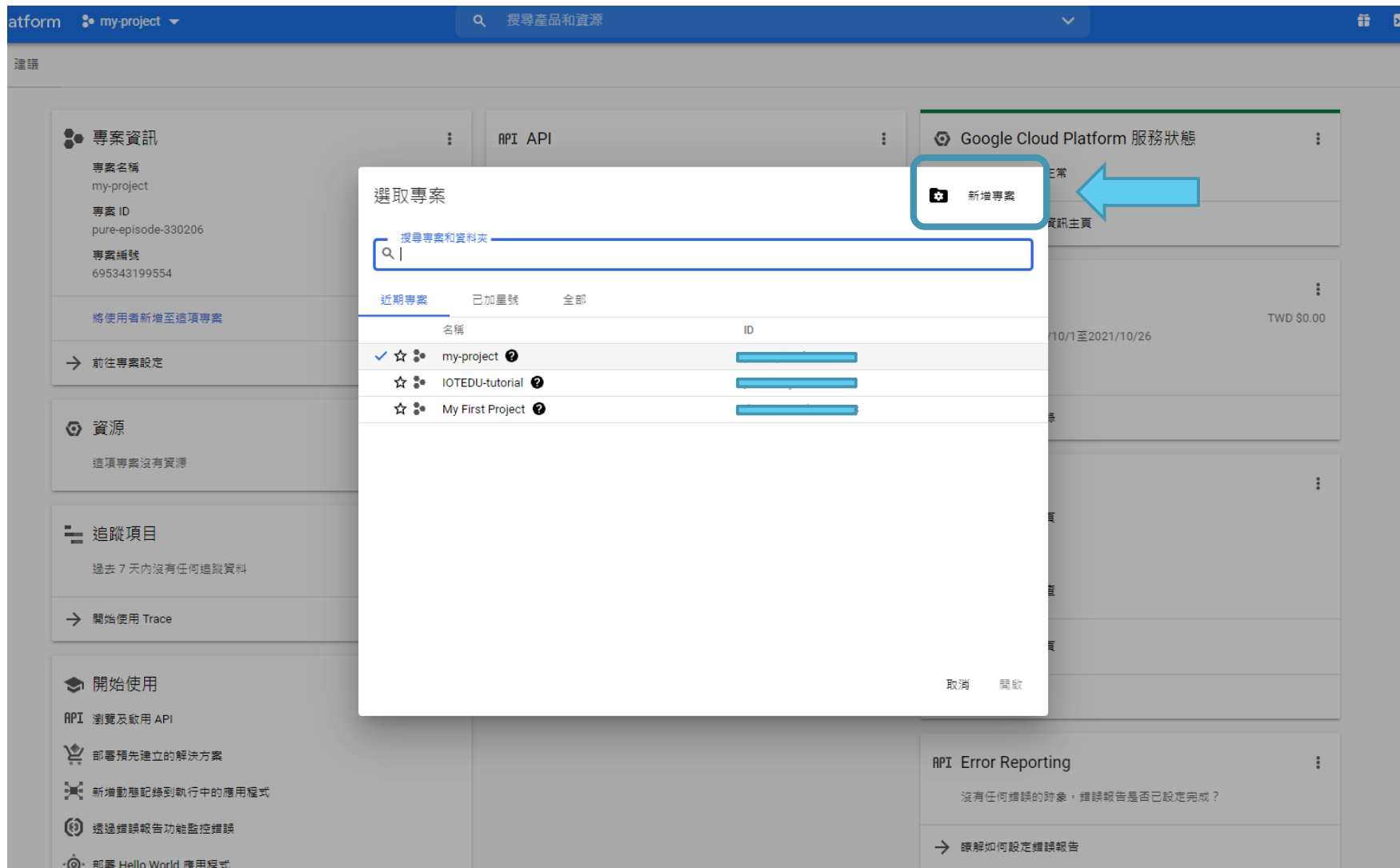
API Error Reporting

沒有任何錯誤的跡象。錯誤報告是否已設定完成?

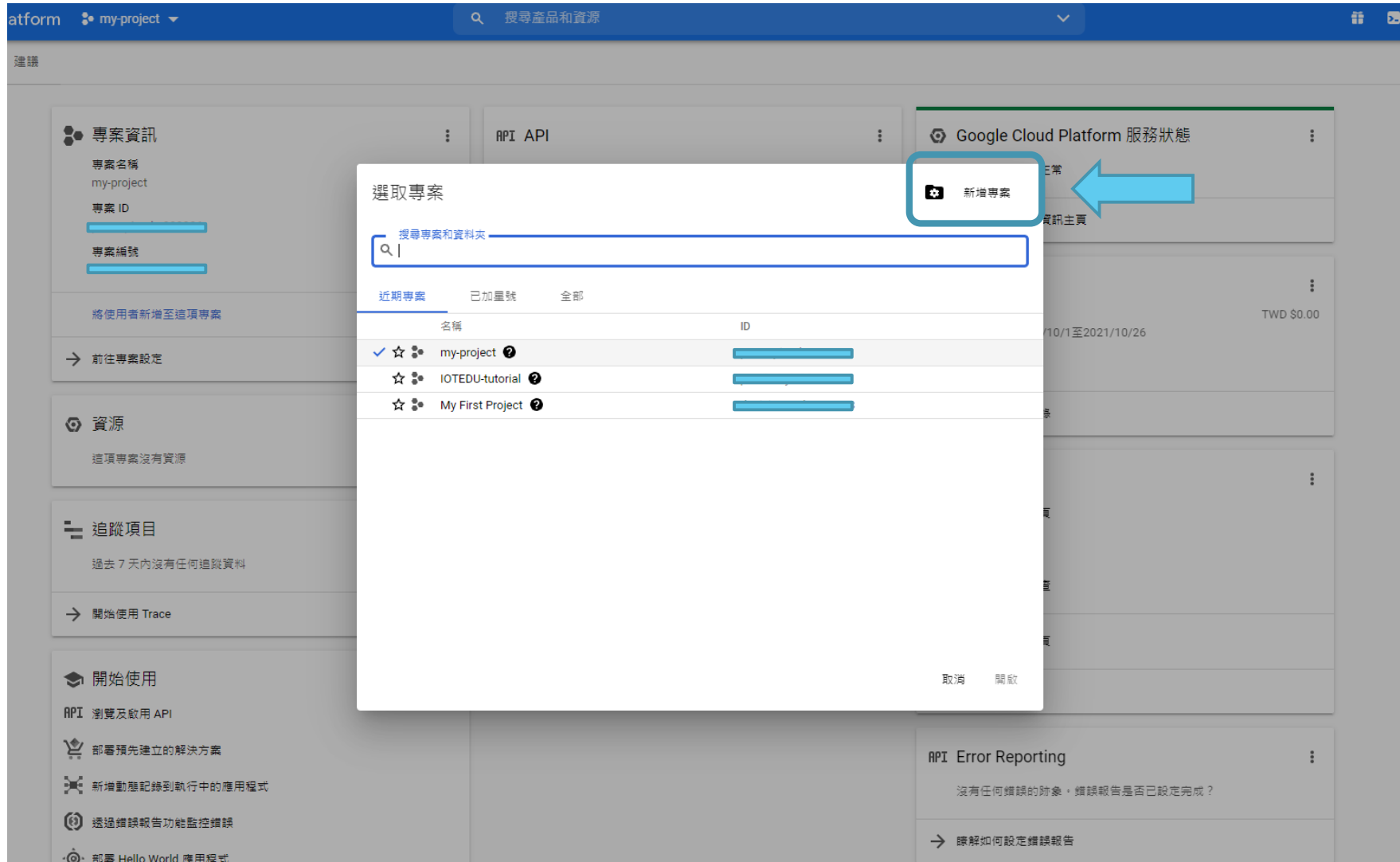
[瞭解如何設定錯誤報告](#)




# Step 3



# Step 4




# Step 5

 Google Cloud Platform

搜尋產品和資源

新增專案

 您的配額還可供建立 5 projects。建議您要求增加配額或刪除專案。 [瞭解詳情](#)

[MANAGE QUOTAS](#)


專案名稱 \*

my-project

?

專案 ID : dogwood 。專案 ID 設定完成後即無法變更。 [編輯](#)

位置 \*

 無機構

[瀏覽](#)

上層機構或資料夾

建立

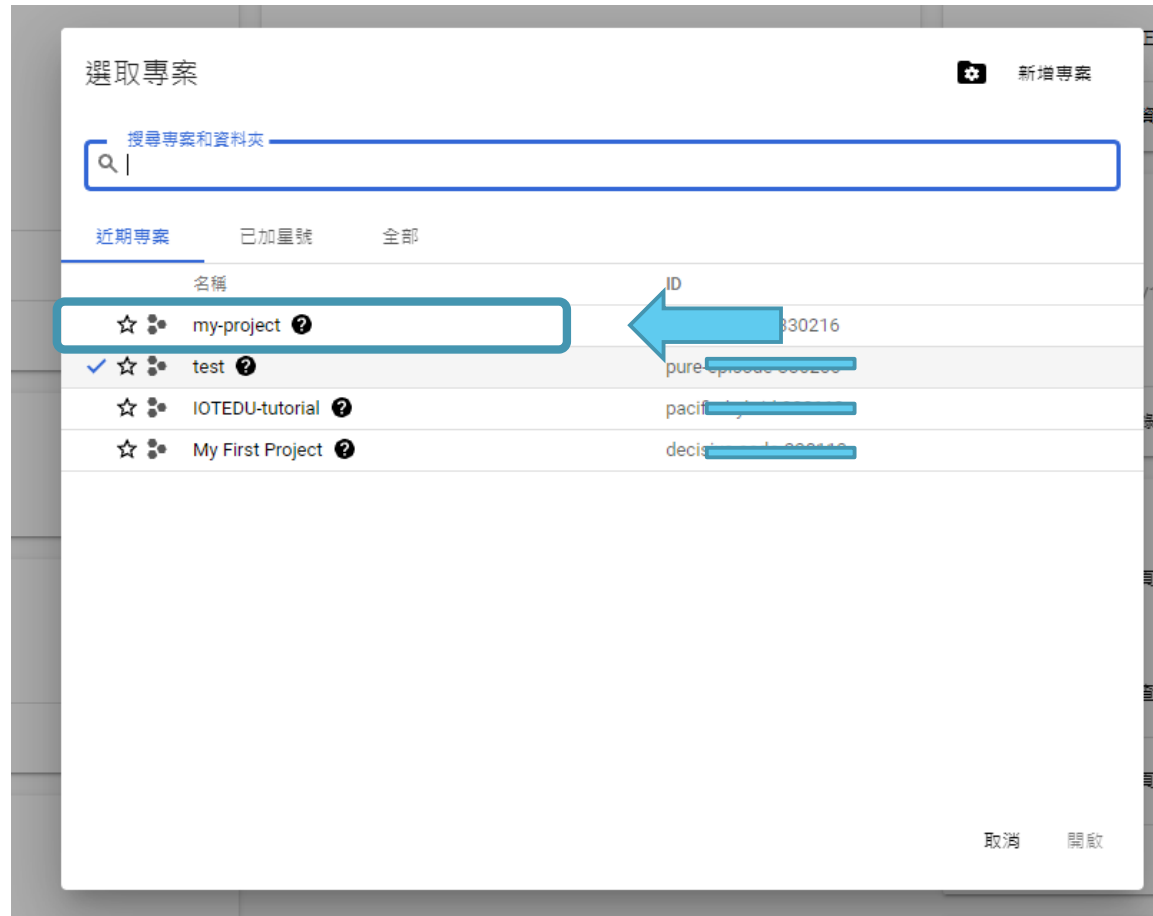
取消

# Step 6

The screenshot displays the Google Cloud Platform (GCP) console interface. At the top, the navigation bar shows the 'test' project selected, highlighted by a red box and a red arrow. Below the navigation bar, the main content area is divided into several sections:

- 專案資訊 (Project Information):** Displays the project name 'test', project ID, and project number. It includes a link to '將使用者新增至這項專案' (Add users to this project) and a button to '前往專案設定' (Go to project settings).
- 資源 (Resources):** Indicates that there are no resources for this project.
- 追蹤項目 (Tracked Items):** Shows that there is no tracking data for the last 7 days. It includes a button to '開始使用 Trace' (Start using Trace).
- 開始使用 (Get Started):** Provides links to 'API 瀏覽及啟用 API' (Browse and enable APIs), '部署預先建立的解決方案' (Deploy pre-built solutions), '新增動態記錄到執行中的應用程式' (Add dynamic logging to running applications), '透過錯誤報告功能監控錯誤' (Monitor errors using error reporting), and '部署 Hello World 應用程式' (Deploy Hello World application).
- API API:** A section for API resources, showing a graph of '要求 (每秒要求數)' (Requests (per second)) over time. It includes a button to '前往 API 總覽' (Go to API overview).
- Google Cloud Platform 服務狀態 (Google Cloud Platform Service Status):** Indicates that all services are operational. It includes a link to '前往 Cloud 狀態資訊主頁' (Go to Cloud status information page).
- 計費功能 (Billing):** Shows the estimated cost (TWD \$0.00) and the billing period (2021/10/1 to 2021/10/26). It includes a link to '查看詳細扣款記錄' (View detailed billing records).
- Monitoring:** Provides links to '建立我的資訊主頁' (Create my dashboard), '設定快訊政策' (Set alert policy), '建立運作時間檢查' (Create uptime check), and '查看所有資訊主頁' (View all dashboards). It includes a button to '前往 Monitoring' (Go to Monitoring).
- API Error Reporting:** Indicates that there are no error reports. It includes a link to '瞭解如何設定錯誤報告' (Learn how to set up error reporting).

# Step 7



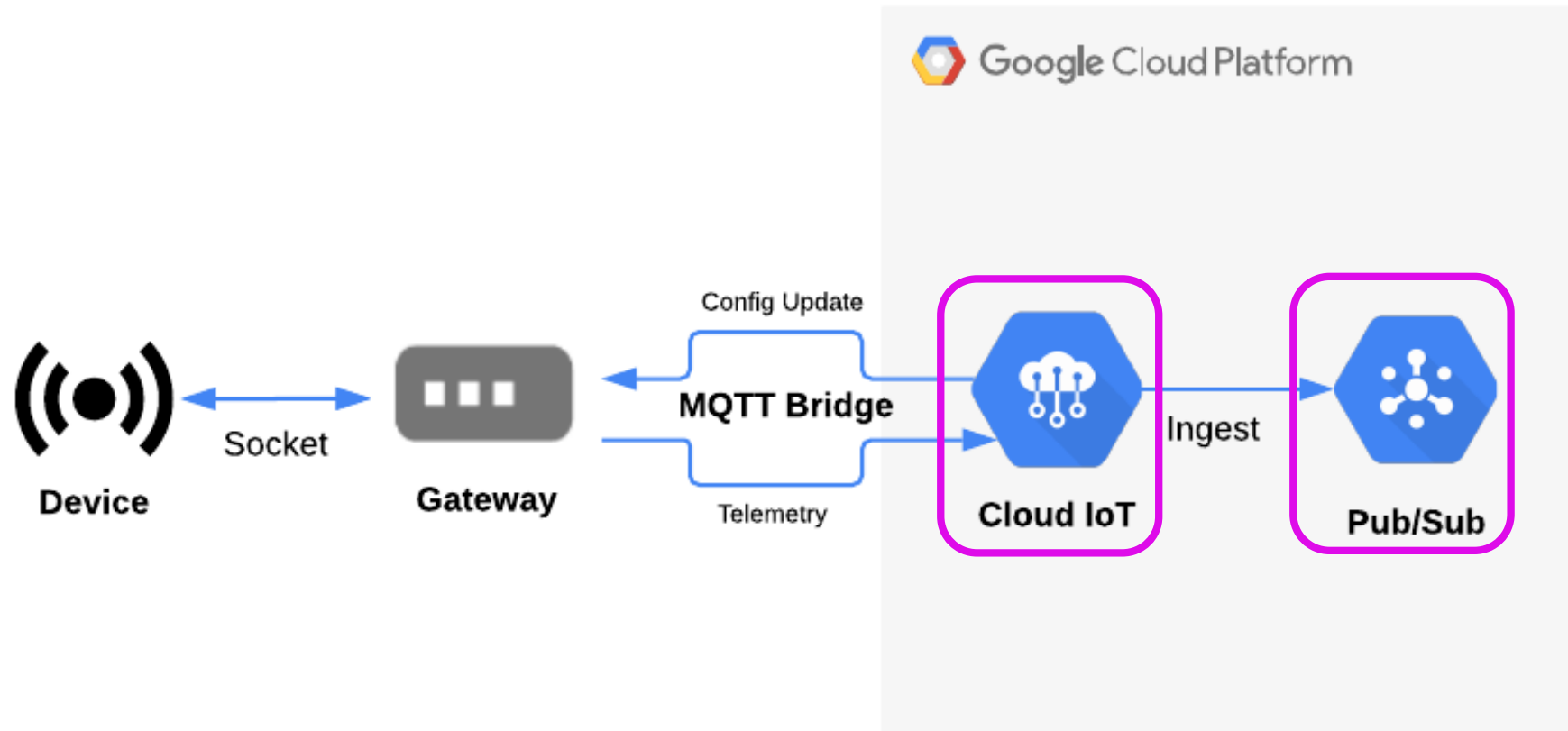
# Step 8

The screenshot displays the Google Cloud Platform (GCP) console interface. At the top, a blue header bar contains the 'Google Cloud Platform' logo, a dropdown menu labeled 'my-project' (highlighted with a red box), a search bar, and utility icons. Below the header, the main content area is divided into several sections:

- 專案資訊 (Project Information):** Displays project details for 'my-project', including the Project ID and Project Number. It includes a link to '前往專案設定' (Go to Project Settings).
- 資源 (Resources):** Indicates that no resources are currently available for this project.
- 追蹤項目 (Tracked Items):** Shows that there is no tracking data available for the last 7 days, with a link to '開始使用 Trace' (Start using Trace).
- 開始使用 (Get Started):** Provides links for 'API 瀏覽及啟用 API' (Browse and enable APIs) and '部署預先建立的解決方案' (Deploy pre-built solutions).
- API API:** A section for API usage, currently showing 'No data is available for the selected time frame.' with a link to '前往 API 總覽' (Go to API Overview).
- Google Cloud Platform 服務狀態 (Google Cloud Platform Service Status):** Confirms that all services are operational, with a link to '前往 Cloud 狀態資訊主頁' (Go to Cloud Status Information page).
- 計費功能 (Billing):** Shows the estimated cost as TWD \$0.00 for the period 2021/10/1 to 2021/10/26, with a link to '查看詳細扣款記錄' (View detailed billing records).
- Monitoring:** Offers options to '建立我的資訊主頁' (Create my dashboard), '設定快訊政策' (Set alert policy), and '建立運作時間檢查' (Create uptime check), with a link to '前往 Monitoring' (Go to Monitoring).
- API Error Reporting:** A section for reporting errors.

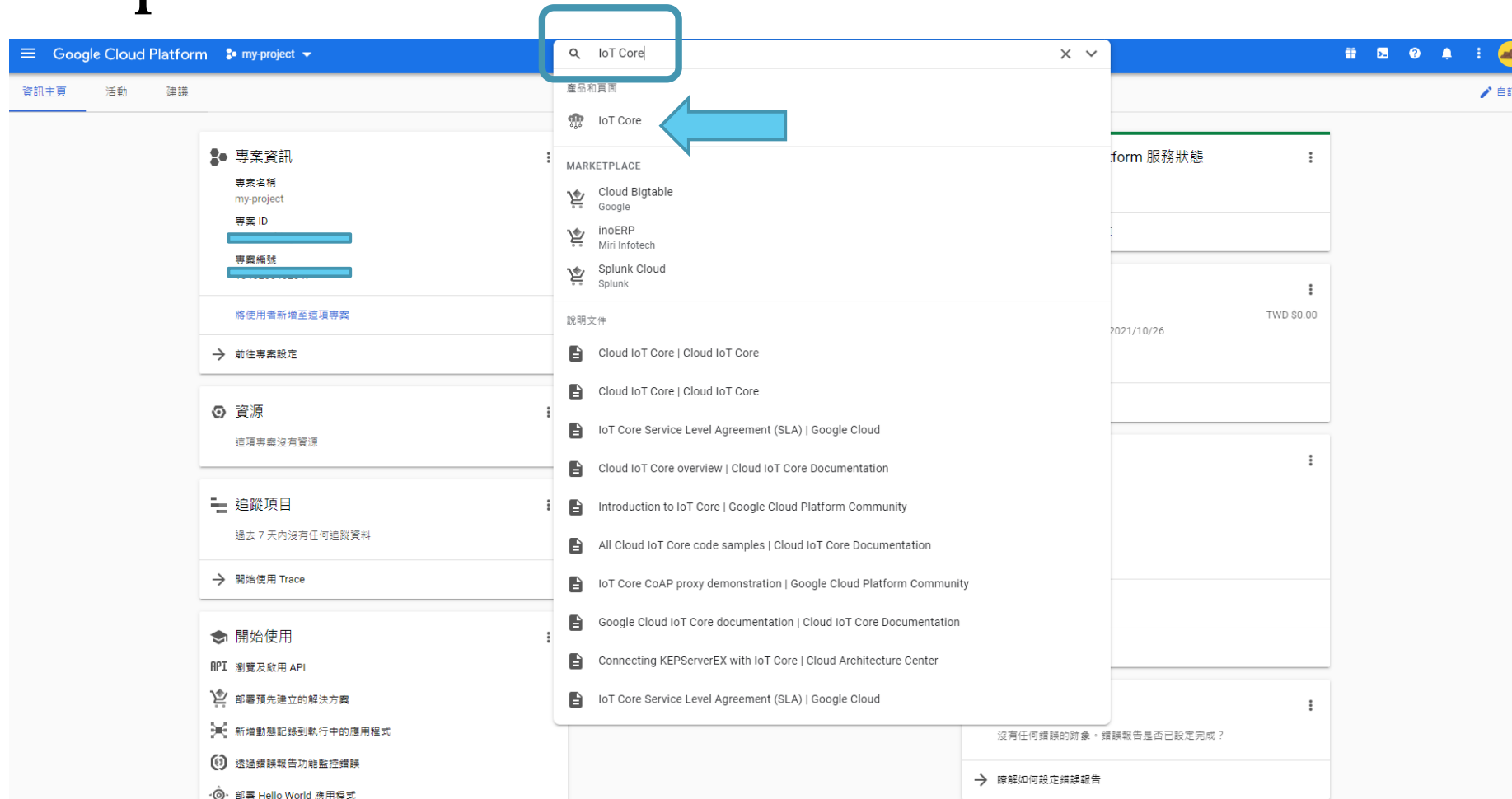
# Enable IoT Core and Pub/Sub APIs(Browser)

## ➤ Architecture





# Enable IoT Core and Pub/Sub APIs(Browser)


## Step 1






# Step 2

 Google Cloud Platform 





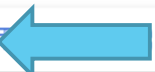
## Google Cloud IoT API

Google Enterprise API

Registers and manages IoT (Internet of Things) devices that connect to the Google Cloud Platform.

啟用

試用



總覽

定價

說明文件

### 總覽

Registers and manages IoT (Internet of Things) devices that connect to the Google Cloud Platform.

[瞭解詳情](#)

### 其他詳細資料

類型: [SaaS & APIs](#)

上次更新時間: 2021/7/23

類別: [Big data](#), [Google Enterprise APIs](#)

服務名稱: [cloudiot.googleapis.com](#)

# Step 3

Google Cloud Platformmy-projectIoT Core

IoT Core登錄檔+ 建立登錄檔

隱藏資訊面板

登錄檔登錄登錄檔

登錄檔 ID ↑

區域

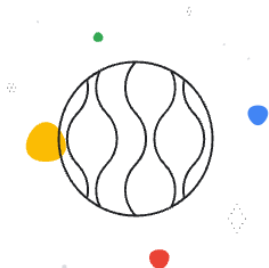
通訊協定

遙測 Pub/Sub 主題

沒有可顯示的登錄檔

未選取任何登錄檔

請至少選取一項資源。



您尚未針對這項專案建立登錄檔。  
按一下 [+ 建立登錄檔] 按鈕即可新增登錄檔。

# Step 4

## Repeat Step 1~3 to enable Pub/Sub

 Pub/Sub

主題

訂閱項目

快照

結構定義

Lite 保留項目

Lite 主題

Lite 訂閱項目

主題

建立主題

刪除

篩選 篩選主題

主題 ID ↑

加密金鑰

主題名稱

標籤

保留

沒有可顯示的主題

選取主題

權限

標籤

儲存空間政策

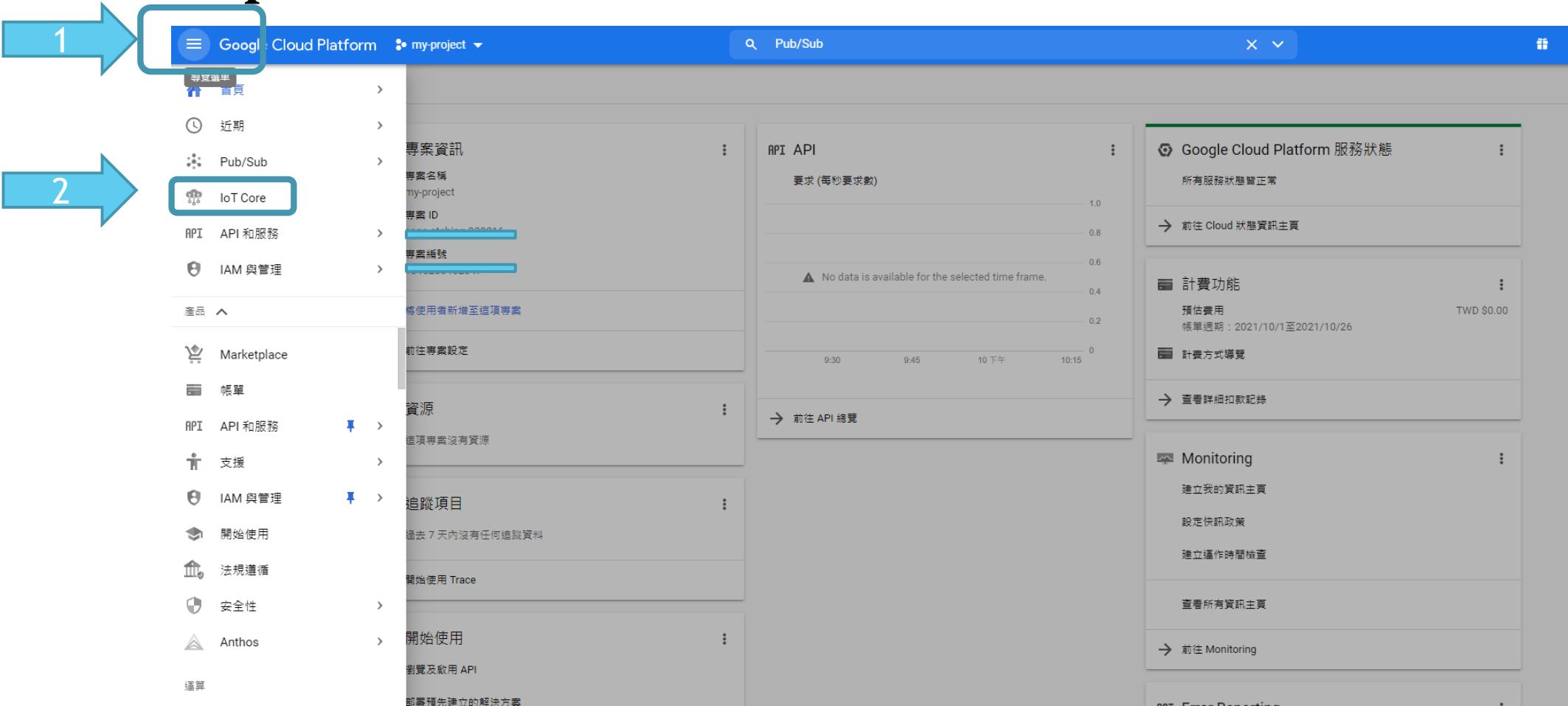
活動

1

請至少選取一項資源。

# Create registry

## Step 1



# Step 2



您尚未針對這項專案建立登錄檔。  
按一下 [+ 建立登錄檔] 按鈕即可新增登錄檔。

# Step 3

Google Cloud Platform

my-project

IoT Core

← 建立登錄檔

定義這個登錄檔中的裝置如何將資料傳送至 Cloud IoT Core。建立登錄檔之後，您就能開始在當中新增裝置。[瞭解詳情](#)

### 登錄檔屬性

登錄檔 ID \*

my-registry

登錄檔的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、-、%、-、\_ 和 ~

地區

asia-east1

指定這個登錄檔中裝置的資料儲存位置，一旦選定就無法變更。

### Cloud Pub/Sub 主題

Cloud IoT Core 會將裝置訊息轉送至 Cloud Pub/Sub，以便進行匯總。您可以依據訊息中的資料類型，將訊息轉送至 Cloud Pub/Sub 中的不同主題和子資料夾。[瞭解詳情](#)

選取 Cloud Pub/Sub 主題

無

根據預設，系統會將裝置遙測事件發布至這個主題。

### 其他主題

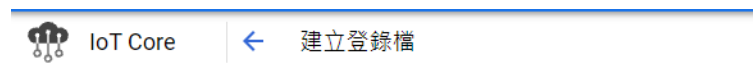
+ 新增

✓ 顯示進階選項

建立

取消

# Step 4



定義這個登錄檔中的裝置如何將資料傳送至 Cloud IoT Core。建立登錄檔之後，您就能開始在當中新增裝置。[瞭解詳情](#)

## 登錄檔屬性

登錄檔 ID \*

my-registry

登錄檔的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、-、%、\_、. 和 ~

地區

asia-east1

指定這個登錄檔中裝置的資料儲存位置，一旦選定就無法變更。

## Cloud Pub/Sub 主題

Cloud IoT Core 會將裝置訊息轉送至 Cloud Pub/Sub，以便進行匯總。您可以依據訊息中的資料類型，將訊息轉送至 Cloud Pub/Sub 中的不同主題和子資料夾。[瞭解詳情](#)

1

選取 Cloud Pub/Sub 主題

無

根據預設，系統會將裝置遙測事件發布至這個主題。

## 其他主題

+ 新增

▼ 顯示進階選項

建立 取消

## Cloud Pub/Sub 主題

Cloud IoT Core 會將裝置訊息轉送至 Cloud Pub/Sub，以便進行匯總。您可以依據訊息中的資料類型，將訊息轉送至 Cloud Pub/Sub 中的不同主題和子資料夾。[瞭解詳情](#)

2

選取 Cloud Pub/Sub 主題

Filter 輸入內容以篩選主題

無

sage-etching-330216

There are no available topics in this project

建立主題 取消 確定

3

建立主題

將主題新增至 Pub/Sub，以便在裝置登錄檔中使用。

主題 ID \*

gateway-telemetry

主題名稱：projects/sage-etching-330216/topics/gateway-telemetry

☐ 使用結構定義 ?


☐ 設定訊息保留時間 (付費功能) ?

☐ 使用客戶自行管理的加密金鑰 (CMEK)

4

取消 建立主題

# Step 5

 IoT Core < 建立登錄檔

定義這個登錄檔中的裝置如何將資料傳送至 Cloud IoT Core。建立登錄檔之後，您就能開始在當中新增裝置。 [瞭解詳情](#)

### 登錄檔屬性

登錄檔 ID \*

my-registry

登錄檔的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、.、%、-、\_ 和 ~。

地區

asia-east1

指定這個登錄檔中裝置的資料儲存位置，一旦選定就無法變更。

### Cloud Pub/Sub 主題

Cloud IoT Core 會將裝置訊息轉送至 Cloud Pub/Sub，以便進行匯總。您可以依據訊息中的資料類型，將訊息轉送至 Cloud Pub/Sub 中的不同主題和子資料夾。 [瞭解詳情](#)

選取 Cloud Pub/Sub 主題

projects/sage-etching-330216/topics/gateway-telemetry

根據預設，系統會將裝置遙測事件發布至這個主題。

### 其他主題

+ 新增

顯示進階選項

 IoT Core < 建立登錄檔

選取 Cloud Pub/Sub 主題

projects/sage-etching-330216/topics/gateway-telemetry

根據預設，系統會將裝置遙測事件發布至這個主題。

### 其他主題

+ 新增

### 裝置狀態主題 (選填)

系統會盡可能將裝置狀態資料發布至您選取的主題，如果您的裝置使用的是 MQTT 通訊協定，則系統會一併將這類資料發布至預設的 MQTT 狀態主題。 [瞭解詳情](#)

選取 Cloud Pub/Sub 主題

無

### 通訊協定

選取裝置用於連線至 Cloud IoT Core 的通訊協定。 [瞭解詳情](#)

☒ MQTT

☒ HTTP

1

2



# Step 6

## 裝置狀態主題 (選填)

系統會盡可能將裝置狀態資料發布至您選取的主題。如果您的裝置使用的是 MQTT 通訊協定，則系統會一併將這類資料發布至預設的 MQTT 狀態主題。[瞭解詳情](#)

選取 Cloud Pub/Sub 主題

Filter 輸入內容以篩選主題

無

sage-etching-330216

There are no available topics in this project

建立主題

取消

確定

## Cloud Logging

設定這個登錄檔中所有裝置的預設記錄功能，您可以在裝置層級套用不同設定或偵錯。[瞭解詳情](#)

☒ 已停用

不儲存裝置資料。

☐ 錯誤

擷取裝置錯誤，例如失敗的連線嘗試和發布作業，但不包含驗證錯誤。

☐ 全部

## 建立主題

將主題新增至 Pub/Sub，以便在裝置登錄檔中使用。

主題 ID \*

gateway-state

?

主題名稱：projects/sage-etching-330216/topics/gateway-state

☐ 使用結構定義 ?

☐ 設定訊息保留時間 (付費功能) ?

☐ 使用客戶自行管理的加密金鑰 (CMEK)

3

取消

建立主題

# Step 7

Google Cloud Platform

my-project

IoT Core

← 建立登錄檔

定義這個登錄檔中的裝置如何將資料傳送回 Cloud IoT Core。建立登錄檔之後，您就能開始在舊中新增裝置。[瞭解詳情](#)

登錄檔屬性

登錄檔 ID \*

my-registry

登錄檔的永久 ID，長度介於 3 至 255 個字元之間，而且只能包含英文字母。您也可以使用數字和下列字元：+、-、%、.、\_ 和 ~。

地區

asia-east1

指定這個登錄檔中裝置的資料儲存位置，一旦選定就無法變更。

Cloud Pub/Sub 主題

Cloud IoT Core 會將裝置訊息轉送至 Cloud Pub/Sub，以便進行匯總。您可以依據訊息中的資料類型，將訊息轉送至 Cloud Pub/Sub 中的不同主題和子資料夾。[瞭解詳情](#)

選取 Cloud Pub/Sub 主題

projects/sage-etching-330216/topics/gateway-telemetry

根據預設，系統會將裝置遙測事件發布至這個主題。

其他主題

+ 新增

裝置狀態主題 (選填)

系統會盡可能將裝置狀態資料發布至您選取的主題，如果您的裝置使用的是 MQTT 通訊協定，則系統會一併將這類資料發布至預設的 MQTT 狀態主題。[瞭解詳情](#)

選取 Cloud Pub/Sub 主題

projects/sage-etching-330216/topics/gateway-state

通訊協定

選取裝置用於連接到 Cloud IoT Core 的通訊協定。[瞭解詳情](#)

☒ MQTT

☒ HTTP

## Cloud Logging

設定這個登錄檔中所有裝置的預設記錄功能，您可以在裝置資訊頁面用不同設定來選擇。[瞭解詳情](#)

- ☒ 已停用  
不儲存裝置資料。
- ☐ 錯誤  
擷取裝置錯誤，例如失敗的連線嘗試和發布作業，但不包含設備錯誤。
- ☐ 資訊  
僅限 MQTT。擷取裝置錯誤 (設備錯誤除外) 並包含所有生命週期事件，例如裝置連線與中斷連線事件。
- ☐ 偵錯  
以極為詳盡的記錄格式式擷取所有裝置活動，適合用於裝置疑難排解。

## CA 憑證 (選填)

### 輸入方法

- ☒ 手動輸入
- ☐ 上傳

憑證值

建立這個登錄檔之後，您就能新增更多憑證。

隱藏進階選項

建立

High Speed ISCC  
Communication & Computing

26

# Step 8

Google Cloud Platform

my-project

搜尋產品和資源

IoT Core

登錄檔詳細資料

裝置

閘道

監控

登錄檔詳細資料

登錄檔 ID : my-registry

地區	asia-east1
通訊協定	MQTT
	HTTP
Cloud Logging	已停用 <a href="#">查看記錄</a>

### Cloud Pub/Sub 主題

一個登錄檔中可以有 1 或多個用於發布裝置遙測和狀態事件的主題。 [新增或編輯主題](#)

### Pub/Sub topics

主題名稱	Topic type ?	子資料夾
projects/sage-etching-330216/topics/gateway-telemetry	預設遙測主題	—
projects/sage-etching-330216/topics/gateway-state	裝置狀態	—

CA CERTIFICATES

# Set up your gateway (Raspberry Pi)

## Step 1

```
git clone https://github.com/andykuo8766/GCP_Pi.git
```

```
pi@raspberrypi:~ $ git clone https://github.com/andykuo8766/GCP_Pi.git
Cloning into 'GCP_Pi'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 47 (delta 8), reused 38 (delta 7), pack-reused 0
Unpacking objects: 100% (47/47), done.
```

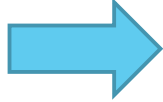
```
cd GCP_Pi/cloud-iot-gateways-rpi/
./generate_keys.sh
```

```
pi@raspberrypi:~ $ cd GCP_Pi/cloud-iot-gateways-rpi/
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ ./generate_keys.sh
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $
```

# Step 2

Copy the contents of rsa\_public.pem

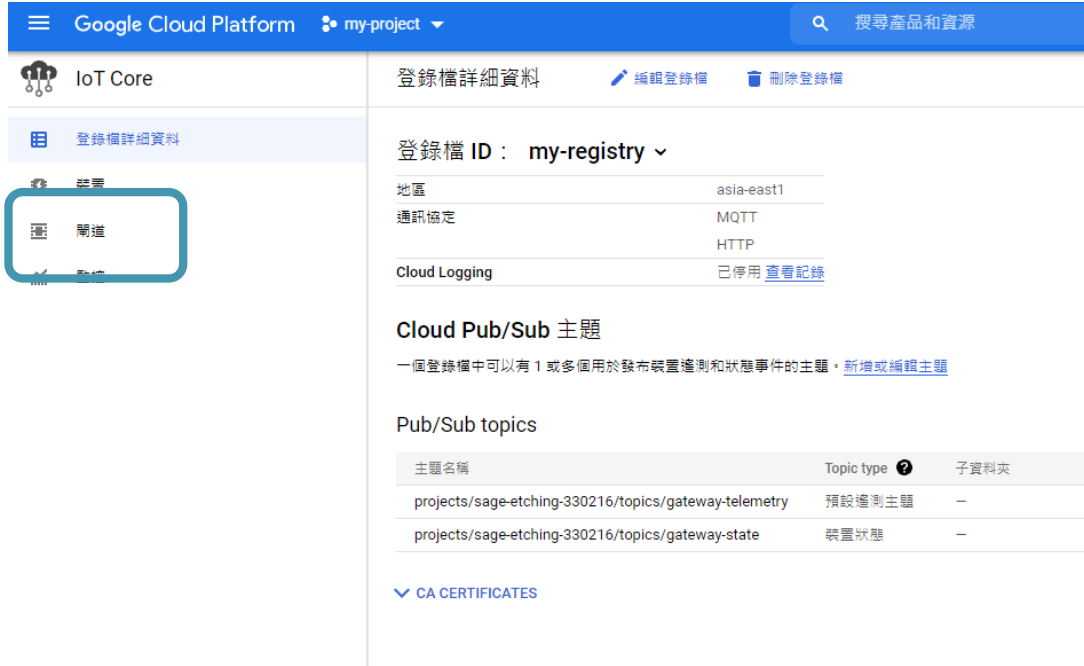
```
cat rsa_public.pem
```



```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ cat rsa_public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAR/Lx1swoTo0yBUbegUvW
Ua7bhKBK0TIjjcacJRjrrWTRSKYIrGxBAqboxl+h7TNM3yBUJP4YMDyYwVIKb2lq
QkfK7F9dr/iJzgDz2HdID/po8jXVIAejlh0yGsY9HZ5UaPiWX6YUbI02NcoEyqGz
Ulp3PMi0l31xmwmnbnsCeGKySSSnJwfW+0RrUjVtha3afd0x1+P9bPaDsqA4o4jLZ
iD2UzXe+W780+IIhCioh6ba9E/w2jFrKuMgAdET+u8c6ZgwSRt71u+7qUREltaTG
E5UAaVHGRBvWeSFvPYIXQjo3Jw+l0izTMtGPdZG+mymVFFnJ/7DbGnJA787EdVvG
vQIDAQAB
-----END PUBLIC KEY-----
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $
```

# Change to browser

## Step 3



Google Cloud Platform my-project

IoT Core

登錄檔詳細資料 編輯登錄檔 刪除登錄檔

登錄檔 ID : my-registry

地區 asia-east1

通訊協定 MQTT

Cloud Logging 已停用 [查看記錄](#)

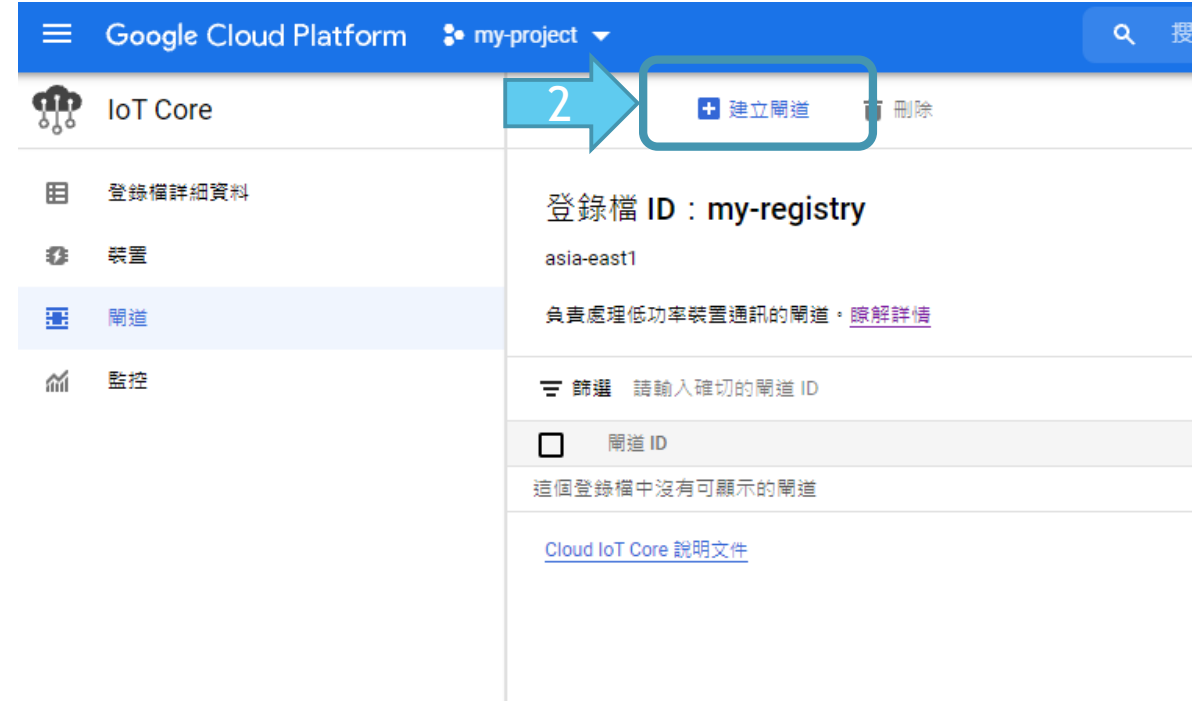
Cloud Pub/Sub 主題

一個登錄檔中可以有 1 或多個用於發布裝置遙測和狀態事件的主題，[新增或編輯主題](#)

Pub/Sub topics

主題名稱	Topic type	子資料夾
projects/sage-etching-330216/topics/gateway-telemetry	預設遙測主題	—
projects/sage-etching-330216/topics/gateway-state	裝置狀態	—

CA CERTIFICATES



Google Cloud Platform my-project

IoT Core

登錄檔詳細資料 裝置 開道 監控

登錄檔 ID : my-registry

asia-east1

負責處理低功率裝置通訊的開道，[瞭解詳情](#)

篩選 請輸入確切的開道 ID

開道 ID

這個登錄檔中沒有可顯示的開道

[Cloud IoT Core 說明文件](#)

# Step 4

 IoT Core

 登錄檔詳細資料  
 裝置  
 閘道  
 監控

← 建立閘道

1

閘道 ID \*  
my-gateway

閘道的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、-、%、\_、. 和 ~

閘道中繼資料 (選填)  
您可以設定自訂的閘道中繼資料，例如製造商和位置等。這些資料可用來查詢這個登錄檔中的閘道。 [瞭解詳情](#)

+ 新增屬性

▼ 通訊、CLOUD LOGGING、驗證

建立 取消

 IoT Core

 登錄檔詳細資料  
 裝置  
 閘道  
 監控

← 建立閘道

閘道 ID \*  
my-gateway

閘道的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、-、%、\_、. 和 ~

閘道中繼資料 (選填)  
您可以設定自訂的閘道中繼資料，例如製造商和位置等。這些資料可用來查詢這個登錄檔中的閘道。 [瞭解詳情](#)

+ 新增屬性

2

▼ 通訊、CLOUD LOGGING、驗證

建立 取消

# Step 5

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ cat rsa public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAR/Lx1swoTo0yBUbegUvW
Ua7bhKBK0TIjccacJRjrrWTRSKYIrGxBAqboxl+h7TNM3yBUJP4YMDyYwVIKb2lq
QkfK7F9dr/iJzgDz2HdID/po8jXVIAejlh0yGsY9HZ5UaPiwX6YUbI02NcoEyqGz
Ulp3PMi0l31xmwmbsCeGKySSSnJwfw+0RrUjVtha3afd0x1+P9bPaDsqA4o4jLZ
iD2UzXe+W780+IIhCioh6ba9E/w2jFrKuMgAdET+u8c6ZgwSRt71u+7qUREltaTG
E5UAaVHGRBvWeSFvPYIXQjo3Jw+l0izTMtGPdZG+mymVFFnJ/7DbGnJA787EdVvG
vQIDAQAB
-----END PUBLIC KEY-----
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $
```

 IoT Core

登錄權詳細資料

裝置

閘道

監控

← 建立閘道

封鎖閘道後，Google Cloud 會拒絕閘道傳出的所有通訊內容。如果閘道發生問題或未經過設定，您可以考慮封鎖閘道。

☒ 允許

☐ 封鎖

Cloud Logging

選擇這個閘道的記錄權設定，這項設定只會覆蓋該閘道的登錄權預設值。[瞭解詳情](#)

☒ 使用登錄權預設定

☐ 已停用

不儲存閘道資料。

☐ 錯誤

擷取閘道錯誤，例如失敗的連線嘗試和發布作業，但不包含驗證錯誤。

☐ 資訊

僅限 MQTT，擷取閘道錯誤 (驗證錯誤除外) 並包含所有生命週期事件，例如閘道連線與中斷連線事件。

☐ 偵錯

以極為詳盡的記錄陳述式擷取所有閘道活動，適合用於閘道疑難排解。

驗證 (選填)

指定要用來驗證這個閘道的公開金鑰。您可以將金鑰欄位留空。不過如未提供金鑰，閘道就無法連結至 Google Cloud。[瞭解詳情](#)

輸入方法

☒ 手動輸入

☐ 上傳

公開金鑰格式

RS256

公開金鑰值

E5UAaVHGRBvWeSFvPYIXQjo3Jw+l0izTMtGPdZG+mymVFFnJ/7DbGnJA787EdVvG  
vQIDAQAB  
-----END PUBLIC KEY-----

High Speed IASLLC  
Communication & Computing

32



# Step 6

公開金鑰格式  
RS256

公開金鑰值  
E5UAaVHGRBvWeSFvPYIXQjo3Jw+l0izTMtGPdZG+mymVFFnJ/7DbGnJA787EdVv  
G  
vQIDAQAB  
-----END PUBLIC KEY-----

公開金鑰到期日 (選填)  
☐ 到期日 :

日期 香港時間

## 裝置驗證方式

決定 Cloud IoT Core 如何驗證及信任與這個閘道相關聯的裝置。 [瞭解詳情](#)

- ☒ 僅限關聯  
信任繫結至這個閘道的任何裝置。
- ☐ 僅限裝置憑證  
信任各個裝置前，請先檢查裝置的 JWT。
- ☐ 關聯與裝置憑證  
信任各個裝置前，請先檢查裝置的 JWT 和繫結的閘道。

^ 通訊、CLOUD LOGGING、驗證



# Step 7

 IoT Core

登錄欄詳細資料

裝置

閘道

監控

← 閘道詳細資料

編輯閘道

更新設定

傳送指令

封鎖通訊

刪除

閘道 ID: my-gateway

數字 ID

登錄欄

Cloud Logging

通訊

3180530026230800

my-registry

登錄欄預設值 [查看記錄欄](#)

已核准

詳細資料

已繫結的裝置

設定和狀態

驗證

最新活動

活動訊號 (僅限 MQTT)	—
上次收到遙測事件的時間	—
上次收到閘道狀態事件的時間	—
上次傳送設定的時間	—
可用區設定 ACK (僅限 MQTT)	—
錯誤	—

授權方法

僅限關聯

閘道中繼資料

您可以在閘道設定中新增或編輯中繼資料。 [編輯閘道](#)。

# Step 8




A screenshot of a form with three input fields. The first field is labeled '專案名稱 \*' (Project Name \*) and contains the text 'my-project'. To the right of this field is a '儲存' (Save) button. The second field is labeled '專案 ID' (Project ID) and contains the text 'my-staging-000016'. This field is highlighted with a blue box and a blue arrow labeled '3' points to it. The third field is labeled '專案編號:' (Project Number:) and contains the text '1040200402047'.

# Change to Raspberry Pi

## Step 9

```
export GOOGLE_CLOUD_PROJECT = <Project ID(Copy from Step8)>
printenv
```



```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ export GOOGLE_CLOUD_PROJECT=egg-etching-220216
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ printenv
SHELL=/bin/bash
LANGUAGE=en_US.UTF-8
GOOGLE_CLOUD_PROJECT=egg-etching-220216
NO_AT_BRIDGE=1
PWD=/home/pi/GCP_Pi/cloud-iot-gateways-rpi
LOGNAME=pi
XDG_SESSION_TYPE=tty
```

專案名稱 \*  
my-project 儲存

專案 ID  
egg-etching-220216

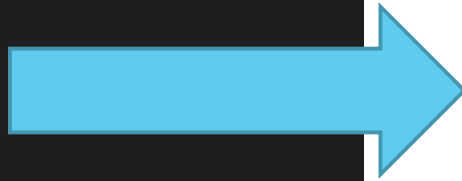
專案編號:  
101020102017

# Step 10

## vim run-gateway

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ vim run-gateway
```

```
python3 ./cloudiot_mqtt_gateway.py \
--registry_id=my-registry \
--gateway_id=my-gateway \
--cloud_region=us-central1 \
--project_id=$GOOGLE_CLOUD_PROJECT \
--private_key_file=rsa_private.pem \
--algorithm=RS256 \
--ca_certs=roots.pem \
--mqtt_bridge_hostname=mqtt.googleapis.com \
--mqtt_bridge_port=8883 \
--jwt_expires_minutes=1200
```



```
python3 ./cloudiot_mqtt_gateway.py \
--registry_id=my-registry \
--gateway_id=my-gateway \
--cloud_region=asia-east1 \
--project_id=$GOOGLE_CLOUD_PROJECT \
--private_key_file=rsa_private.pem \
--algorithm=RS256 \
--ca_certs=roots.pem \
--mqtt_bridge_hostname=mqtt.googleapis.com \
--mqtt_bridge_port=8883 \
--jwt_expires_minutes=1200
```

: wq

Type **:wq** and hit Enter

# Step 11

```
wget https://pki.goog/roots.pem
```

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ wget https://pki.goog/roots.pem
--2021-10-27 02:54:00-- https://pki.goog/roots.pem
Resolving pki.goog (pki.goog)... 216.239.32.29, 2001:4860:4802:32::1d
Connecting to pki.goog (pki.goog)|216.239.32.29|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 71214 (70K) [application/x-pem-file]
Saving to: 'roots.pem'

roots.pem                               100%[=====>] 69.54K  --.-KB/s    in 0.04s

2021-10-27 02:54:01 (1.62 MB/s) - 'roots.pem' saved [71214/71214]
```

```
python3 -m venv env
source env/bin/activate
pip install -r requirements-gateway.txt
```

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ python3 -m venv env
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ source env/bin/activate
(env) pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ pip install -r requirements-gateway.txt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting cryptography==2.4.1 (from -r requirements-gateway.txt (line 1))
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
Using cached https://www.piwheels.org/simple/cryptography/cryptography-2.4.1-cp37-cp37m-linux_armv7l.whl
Collecting paho-mqtt==1.4.0 (from -r requirements-gateway.txt (line 2))
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
Downloading https://www.piwheels.org/simple/paho-mqtt/paho_mqtt-1.4.0-py3-none-any.whl (48kB)
100% | 51kB 90kB/s
Collecting pyjwt==1.6.4 (from -r requirements-gateway.txt (line 3))
  Cache entry deserialization failed, entry ignored
```

# Step 12

```
source run-gateway
```

```
(env) pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ source run-gateway
Creating JWT using RS256 from private key file rsa_private.pem
on_publish, userdata None, mid 1
Unable to find key 1
connect status False
on_connect Connection Accepted.
on_subscribe: mid 2, qos (1,)
Received message '' on topic '/devices/my-gateway/config' with Qos 1
Nobody subscribes to topic /devices/my-gateway/config
on_subscribe: mid 3, qos (0,)
```

Keep this process running while you proceed through the next steps.

Recommend that you use a new terminal.

# Create Device(Browser)

## Step 1



Google Cloud Platform my-project

IoT Core

登錄檔詳細資料 編輯登錄檔 刪除登錄檔

登錄檔 ID: my-registry

地區: asia-east1

通訊協定: MQTT

Cloud Logging: 已停用 [查看記錄](#)

Cloud Pub/Sub 主題

一個登錄檔中可以有 1 或多個用於發布裝置遙測和狀態事件的主題。 [新增或編輯主題](#)

Pub/Sub topics

主題名稱	Topic type	子資料夾
projects/sage-etching-330216/topics/gateway-telemetry	預設遙測主題	-
projects/sage-etching-330216/topics/gateway-state	裝置狀態	-

CA CERTIFICATES



Google Cloud Platform my-project

IoT Core

登錄檔詳細資料 裝置 開道 監控

登錄檔 ID: my-registry

asia-east1

裝置為直接連上網際網路或透過開道連上網際網路的實體設備。 [瞭解詳情](#)

篩選 請輸入確切的裝置 ID

裝置 ID

這個登錄檔中沒有可顯示的裝置

[Cloud IoT Core 說明文件](#)



# Step 2

## 驗證 (選填)

指定要用來驗證這部裝置的公開金鑰。您可以將金鑰欄位留空。不過如未提供金鑰，裝置就無法連結至 Google Cloud。 [瞭解詳情](#)

### 輸入方法

☒ 手動輸入

☐ 上傳

公開金鑰格式

RS256

公開金鑰值

E5UAaVHGRBvWeSFvPYIXQjo3Jw+l0izTMtGPdZG+mymVFFnJ/7DbGnJA787EdVv  
G  
vQIDAQAB  
-----END PUBLIC KEY-----

公開金鑰到期日 (選填)

☐ 到期日：

日期

香港時間



通訊、CLOUD LOGGING、驗證

建立

取消

← 建立裝置

裝置 ID \*

led-light

裝置的永久 ID，長度介於 3 至 255 個字元之間，而且開頭須為英文字母。您也可以使用數字和下列字元：+、-、%、\_、~ 和 ~

## 裝置中繼資料 (選填)

您可以設定自訂的裝置中繼資料，例如製造商和位置等。這些資料可用來查詢這個登錄檔中的裝置。 [瞭解詳情](#)

+ 新增屬性

✓ 通訊、CLOUD LOGGING、驗證

建立

取消

# Step 3

**IoT Core**

登錄檔詳細資料  
裝置  
**閘道**  
監控

**閘道** + 建立閘道 刪除

登錄檔 ID : my-registry  
asia-east1  
負責處理低功耗裝置通訊的閘道。 [瞭解詳情](#)

篩選 請輸入確切的閘道 ID

☐ 閘道 ID  
☐ my-gateway

[Cloud IoT Core 說明文件](#)

閘道詳細資料 編輯閘道 更新設定 傳送指令 封鎖通訊 刪除

閘道 ID: my-gateway

數字 ID	登錄檔	Cloud Logging	通訊
3180530026230800	my-registry	登錄檔預設值 <a href="#">查看記錄檔</a>	已核准

[閘道資料](#) **已繫結的裝置** 設定和狀態 驗證

最新活動

活動訊號 (僅限 MQTT)	2021年10月27日 上午3:10:29
上次收到遙測事件的時間	2021年10月27日 上午2:59:25
上次收到閘道狀態事件的時間	—
上次傳送設定的時間	2021年10月27日 上午2:59:26
可用區設定 ACK (僅限 MQTT)	2021年10月27日 上午2:59:26
錯誤	—

授權方法  
僅限關聯

閘道中繼資料  
您可以在閘道設定中新增或編輯中繼資料。 [編輯閘道](#)。

# Step 4

1

閘道 ID: my-gateway

數字 ID	登錄檔	Cloud Logging	通訊
3180530026230800	my-registry	登錄檔預設值 <a href="#">查看記錄檔</a>	已核准

詳細資料 已繫結的裝置 設定和狀態 驗證

繫結裝置

解除繫結

篩選 請輸入確切的裝置 ID

<input type="checkbox"/>	裝置 ID
閘道沒有任何繫結的裝置。請對裝置進行繫結，在裝置與閘道間建立關聯。	

2

將裝置繫結至閘道

從「my-registry」登錄檔中選取要繫結至「my-gateway」閘道的裝置。

在您將裝置繫結至閘道之後，裝置就能與 Cloud IoT Core 交換 MQTT/HTTP 訊息。請注意，每個閘道可繫結的裝置數量上限為 10,000 部。

篩選 輸入確切的裝置 ID，並以半形逗號分隔

<input checked="" type="checkbox"/>	裝置 ID	通訊	上次出現時間	Cloud Logging
<input checked="" type="checkbox"/>	<a href="#">led-light</a>	✔ 已允許	—	登錄檔預設值

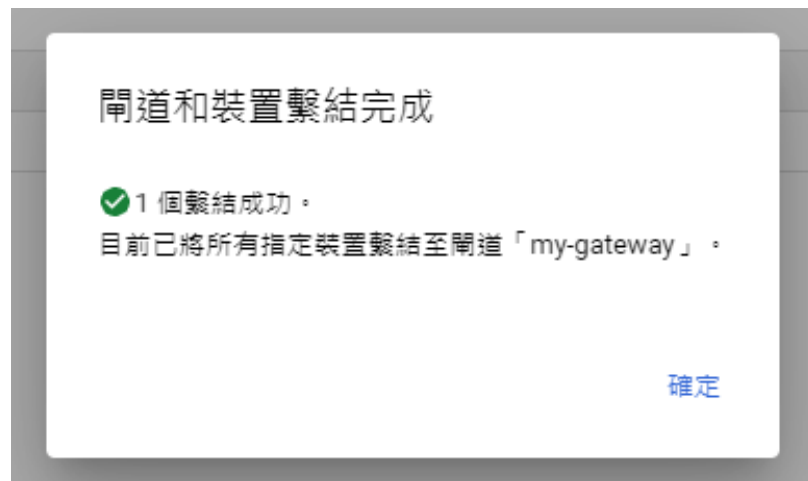
已選取 1 部裝置

取消

繫結

3

# Step 5



## 閘道 ID: my-gateway

數字 ID      登錄權      Cloud Logging      通訊  
3180530026230800      my-registry      登錄權預設值 [查看記錄檔](#)      已核准

詳細資料      已繫結的裝置      設定和狀態      驗證

繫結裝置

解除繫結

篩選 請輸入確切的裝置 ID

<input type="checkbox"/> 裝置 ID	通訊	上次出現時間	Cloud Logging
<input type="checkbox"/> led-light	✔ 已允許	—	登錄權預設值

# Managing devices through configuration updates (Raspberry Pi)

## Step 1

ifconfig

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b8:27:eb:5f:92:ef txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 47 bytes 2482 (2.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 47 bytes 2482 (2.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.137.187 netmask 255.255.255.0 broadcast 192.168.137.255
    inet fe80::db88:b249:7321:8ac0 prefixlen 64 scopeid 0x20<link>
    ether 22:fd:b0:2e:da:05 txqueuelen 1000 (Ethernet)
    RX packets 135844 bytes 180510302 (172.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 65315 bytes 21324762 (20.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

# Step 2

vim led-light.py

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ vim led-light.py
```

```
from __future__ import print_function
import socket
import sys

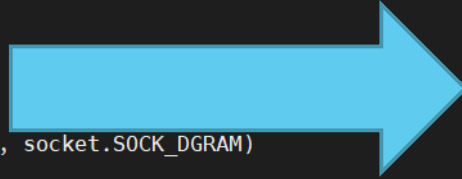
import RPi.GPIO as GPIO
from colors import bcolors

LED_IOPIN = 14
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(LED_IOPIN, GPIO.OUT)

ADDR = ''
PORT = 10000
# Create a UDP socket
client_sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_address = (ADDR, PORT)

device_id = sys.argv[1]
if not device_id:
    sys.exit('The device id must be specified.')
print('Bringing up device {}'.format(device_id))
```



```
from __future__ import print_function
import socket
import sys

import RPi.GPIO as GPIO
from colors import bcolors

LED_IOPIN = 14
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(LED_IOPIN, GPIO.OUT)

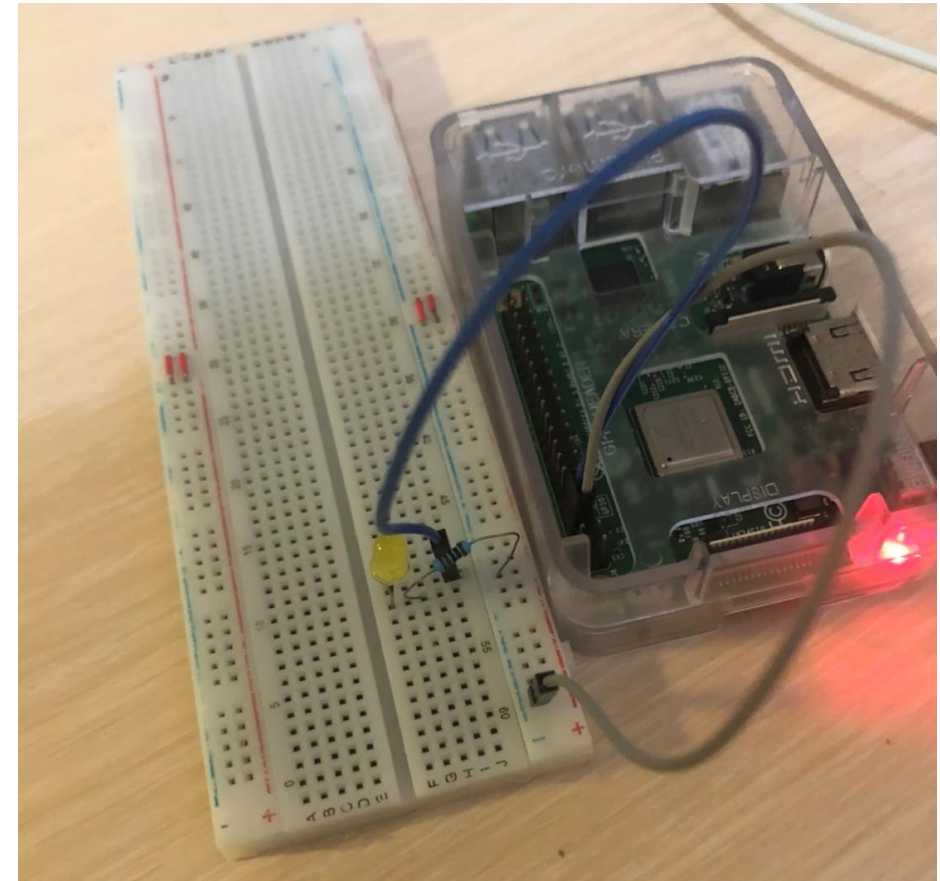
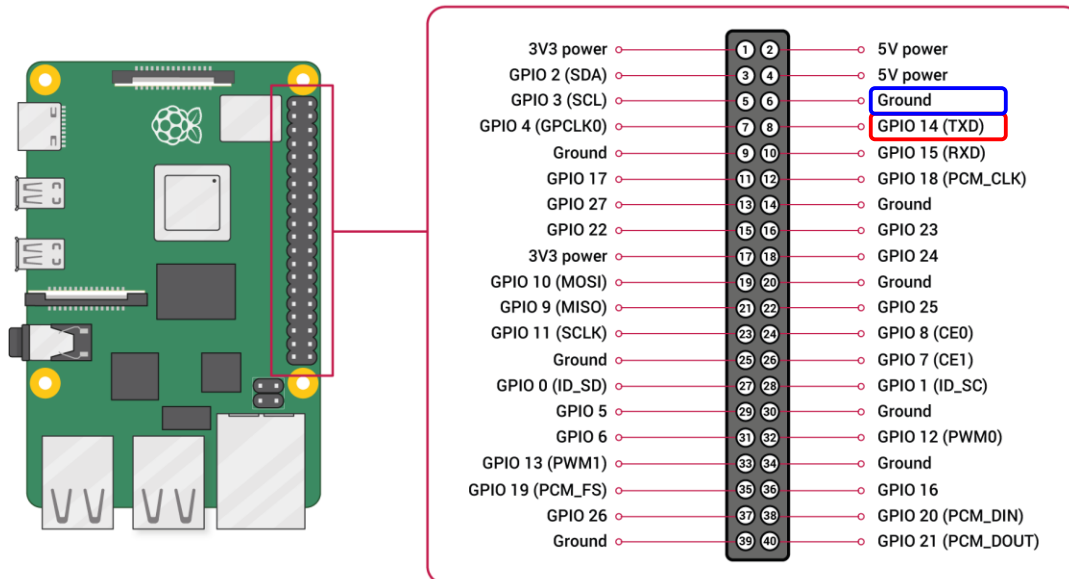
ADDR = '192.168.137.187'
PORT = 10000
# Create a UDP socket
client_sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_address = (ADDR, PORT)

device_id = sys.argv[1]
if not device_id:
    sys.exit('The device id must be specified.')
print('Bringing up device {}'.format(device_id))
```

# Step 3

Connect the LED to the Raspberry Pi **GPIO Pin 14** and **ground** using an appropriate resistor.



# Step 4

## source run-led-light

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ source run-led-light
Bringing up device led-light
Send data: { "device" : "led-light", "action":"detach" }
sending "{ "device" : "led-light", "action":"detach" }"
waiting for response
received: "{ "device": "0", "command": "4", "status" : "ok" }"
Response: { "device": "0", "command": "4", "status" : "ok" }
Send data: { "device" : "led-light", "action":"attach" }
sending "{ "device" : "led-light", "action":"attach" }"
waiting for response
received: "{ "device": "led-light", "command": "attach", "status" : "ok" }"
Response: { "device": "led-light", "command": "attach", "status" : "ok" }
Send data: { "device" : "led-light", "action":"event", "data" : "LED is online" }
sending "{ "device" : "led-light", "action":"event", "data" : "LED is online" }"
waiting for response
received: "{ "device": "led-light", "command": "event", "status" : "ok" }"
Response: { "device": "led-light", "command": "event", "status" : "ok" }
Send data: { "device" : "led-light", "action":"subscribe" }
sending "{ "device" : "led-light", "action":"subscribe" }"
waiting for response
received: "{ "device": "led-light", "command": "subscribe", "status" : "ok" }"
Response: { "device": "led-light", "command": "subscribe", "status" : "ok" }
```



# Change to browser

## Step 5

 IoT Core

 設備詳情與管理

 裝置

 頻道

 監控

 裝置詳細資訊

 編輯裝置

 更新設定

 傳送指令

 封鎖通訊

 刪除

裝置 ID: led-light

數字 ID

3284129026052221

登錄檔

my-registry

Cloud Logging

登錄檔預設值 [查看記錄檔](#)

通訊

已核准

詳細資料

設定和狀態驗證

最新活動

裝置中繼資料

# Step 6

更新設定

請在下方輸入新設定。如果您使用 MQTT，系統會在裝置連線時將這項設定推送到裝置。  
[瞭解詳情](#)

格式

☒ 文字

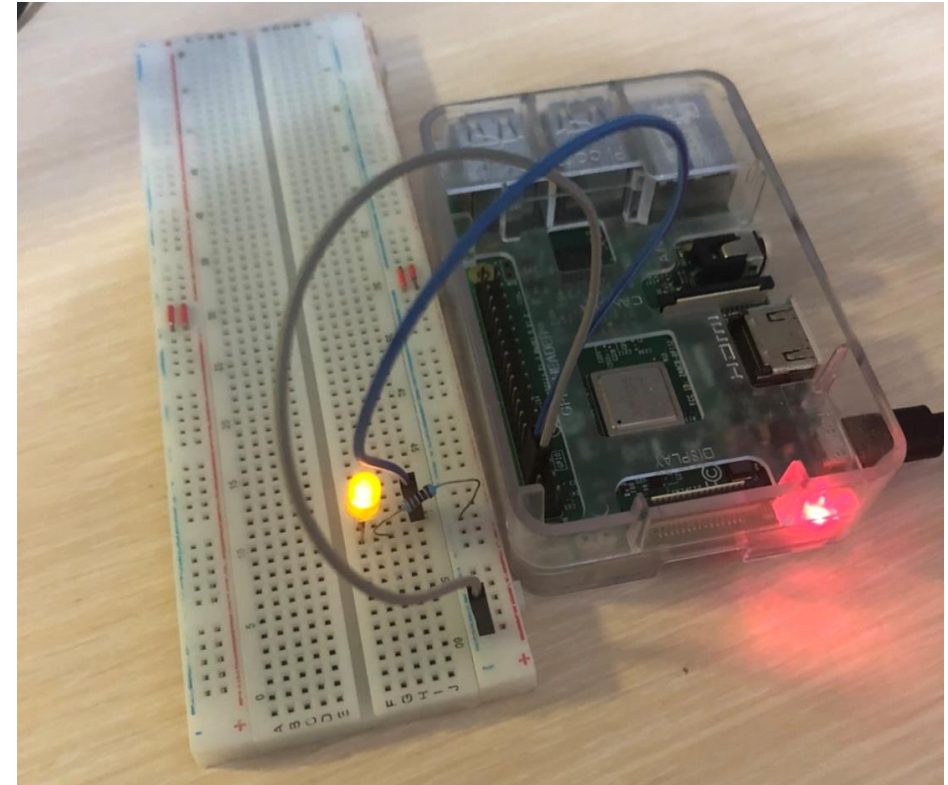
☐ Base64

2/32000

1 設定 ON

2 傳送

傳送至裝置



# Step 7

更新設定

請在下方輸入新設定。如果您使用 MQTT，系統會在裝置連線時將這項設定推送到裝置。  
[瞭解詳情](#)

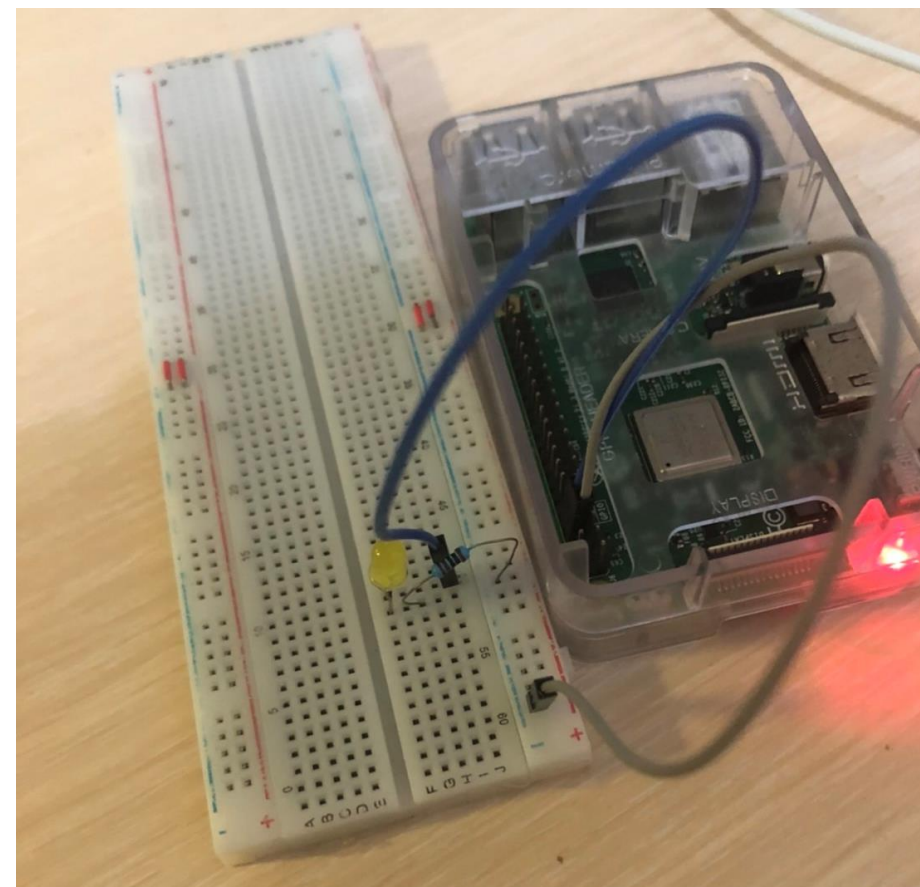
格式

☒ 文字

☐ Base64

1 設定 OFF 3/32000

2 傳送至裝置



# Assignment

## Part I (Publishing telemetry events through the gateway)

- Set up a DHT22 sensor to send telemetry from the sensor through the gateway to IoT Core.
  - Repeat steps from the previous section, "Managing devices through configuration updates", but use thermostat as the Device ID.
  - Edit thermostat.py by adding the IP address of your gateway on line 27: ADDR = "
  - Wire the DHT22 sensor to the Raspberry Pi as described in the setup section of this [tutorial](#).
  - Run the following from a terminal on the Raspberry Pi:

```
source run-thermostat
```

- If everything is done correctly, you should see the temperature on that terminal updating once per second.

```
pi@raspberrypi:~/GCP_Pi/cloud-iot-gateways-rpi $ source run-thermostat
Bringing up device thermostat
Bring up device 1
Send data: { "device" : "thermostat", "action":"detach" }
sending: "{ \"device\" : \"thermostat\", \"action\":\"detach\" }"
waiting for response
received: "{ \"device\": \"0\", \"command\": \"4\", \"status\" : \"ok\" }"
Response { \"device\": \"0\", \"command\": \"4\", \"status\" : \"ok\" }
Send data: { "device" : "thermostat", "action":"attach" }
sending: "{ \"device\" : \"thermostat\", \"action\":\"attach\" }"
waiting for response
received: "{ \"device\": \"thermostat\", \"command\": \"attach\", \"status\" : \"ok\" }"
Response { \"device\": \"thermostat\", \"command\": \"attach\", \"status\" : \"ok\" }
>>Temp: 20.100, Hum: 50.500 <<
```

# Assignment

## Part II (Create a subscription to your telemetry topic to view data)

- Open the Pub/Sub dashboard.
- Click the three-dot menu button next to the telemetry topic that you created earlier, and click New subscription.
- Enter a subscription name, such as my-subscription.
- Make sure that Delivery Type is set to Pull, and leave everything else as-is.
- Click the Create button to create the subscription.
- Click the Activate Cloud Shell icon in the upper right area of the Cloud Console window.
- In Cloud Shell, enter the following:



```
gcloud pubsub subscriptions pull my-subscription --auto-ack --limit=100
```

```
andy8766kuo@cloudshell:~ (sage-etching-330216)$ gcloud pubsub subscriptions pull my-subscription --auto-ack --limit=100
DATA: RPI Gateway started.
MESSAGE_ID: 3331685803830714
ORDERING_KEY:
ATTRIBUTES: deviceId=my-gateway
deviceNumId=3180530026230800
deviceRegistryId=my-registry
deviceRegistryLocation=asia-east1
projectId=[REDACTED]
subFolder=
DELIVERY_ATTEMPT:
DATA: temperature=20.100, humidity=50.500
```