

Google Cloud Platform Integration

Particle has teamed up with Google to create a 1st-class integration with Google Cloud Platform. Google Cloud Platform enables developers to build, test, and deploy applications on Google's highly scalable and reliable infrastructure.



Send device data from Particle into Google Cloud Platform with ease

Streaming data from Particle devices into Google Cloud Platform creates countless opportunities to add intelligence and robustness to your IoT product. Here are some examples of things you could do with Particle + Google Cloud Platform:

- Store data from a fleet of devices in a durable, long-term hosted database
- Combine data sources from an entire fleet into unified data flows that provide product-wide business intelligence
- Take action in the physical world based on results of predictive analysis on a historical data set

Google's Cloud Platform is comprised of dozens of products & services, providing you with a variety of potential IoT applications.

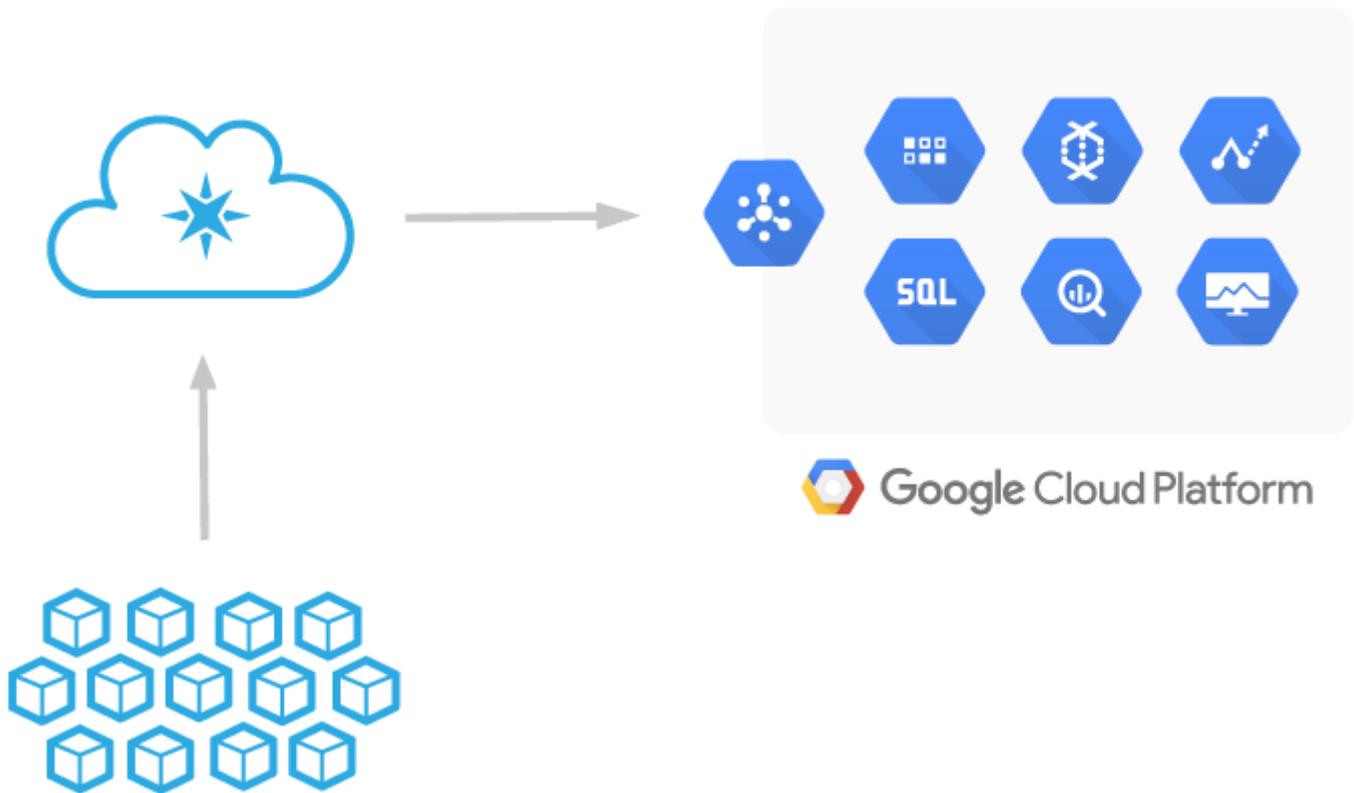
Some of the most relevant Google Cloud Platform products to use with Particle are:

- Cloud Datastore: A highly-scalable NoSQL database for your web and mobile applications
- BigQuery: A fast, economical and fully managed data warehouse for large-scale data analytics
- SQL: A fully managed MySQL database service
- Cloud Dataflow: A fully-managed cloud service and programming model for batch and streaming big data processing
- Prediction API: Machine learning to analyze your data and make predictions

Many of these products relate to data storage. For more information on finding the right data storage option for your particular use case, check out the docs on choosing a storage option. This tutorial will focus on the first example use case: storing data from Particle devices into Google Cloud Platform Datastore.

Preconfiguration in Google Cloud Platform

Regardless of which Google Cloud Platform product(s) you'd like to use, streaming data into Google Cloud Platform requires an entry point. Google Cloud Pub/Sub provides a scalable, flexible and reliable entry point for Particle device data into Google Cloud Platform.



Device data gets routed through the Particle Cloud into Google Cloud Pub/Sub. You can then use that data in any number of Google Cloud Platform products.

The Particle cloud will *publish* device data to a Pub/Sub *topic*. Once inside, Google Cloud Platform, any number of *subscribers* can listen for messages sent to the topic, and relay them to other Google Cloud Platform products & services. Once you have enabled the Google Cloud Platform integration, the Particle Cloud will begin listening for Particle publishes with a specific event name from your fleet of devices, and forward the data payload onto your Google Cloud Pub/Sub topic.

Before setting up the integration in Particle, there are a few pre-configuration steps that you must do inside Google Cloud Platform. These instructions are also included in the Particle Console when you enable the Google Cloud Platform integration.

Sign up for a Google Cloud Platform Account

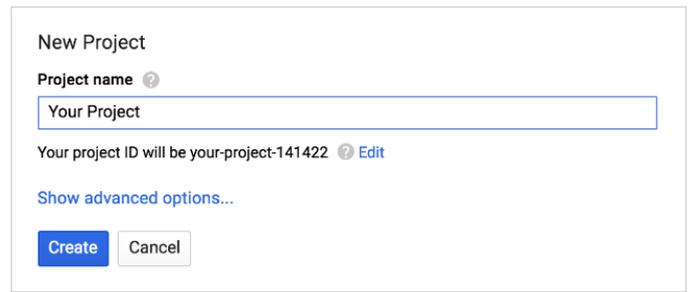
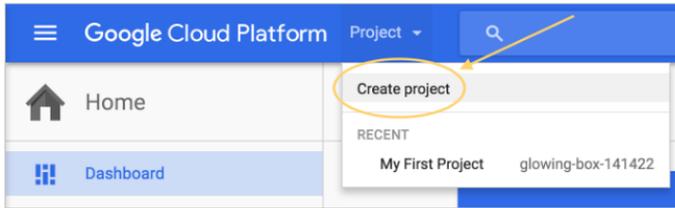
If you don't already have one, you'll need to sign up for a Google Cloud Platform Account to use this integration.

Visit the Google Cloud Platform signup page. Click the "Try it Free" button, and enter the requested information. Upon signup, you will be enrolled in a 60-day free trial of the service, with \$300 in Google Cloud Platform credit.

You can use the Google Cloud Platform Pricing Calculator to estimate costs of any Google Cloud Platform product.

Create a Google Cloud Platform Project

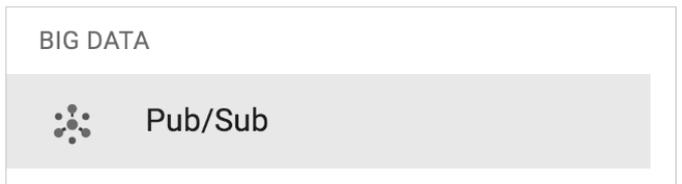
In your Google Cloud Platform console, find the project dropdown in the top navigation bar, and select "Create project." Then, give your project a name. When you're done, click "Create."



Create a Pub/Sub Topic with the correct permissions

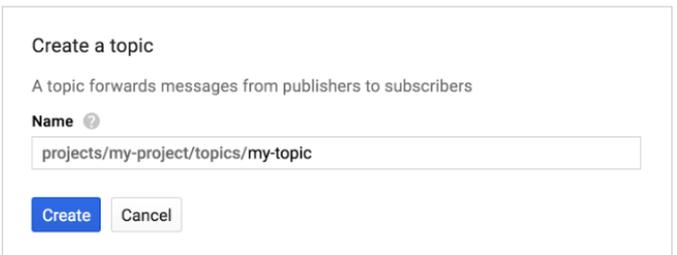
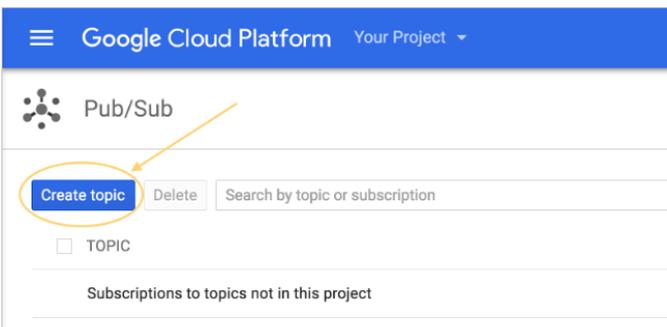
Navigate to Pub/Sub

Click the hamburger menu icon in the top navigation bar. Then, from the slide out menu, select "Pub/Sub" from the list of Google Cloud Platform products.



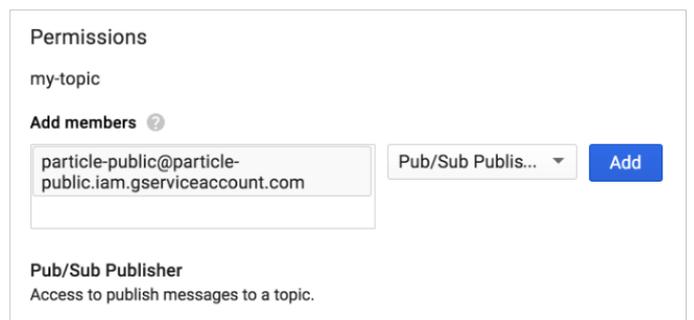
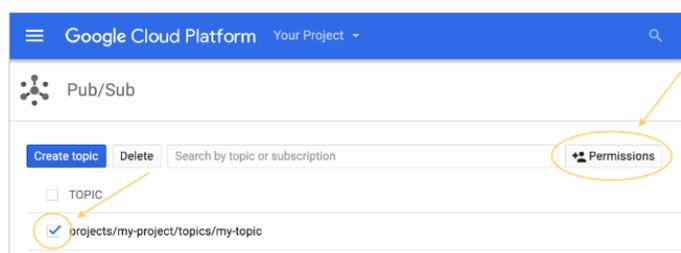
Create a Topic

If this is your first time using Pub/Sub, you will need to enable the Pub/Sub API before you can create a topic. Once the API is enabled for your account, click on the "Create topic" button from the Pub/Sub dashboard. Then, give your topic a name. Google automatically prefixes your topic name to ensure that you will publish to the right place.



Give Particle Permission to Publish to your Topic

Once created, check the checkbox next to your topic name, and click on the "Permissions" button. Then, in the "Add members" box, paste in **particle-public@particle-public.iam.gserviceaccount.com**. Choose **Pub/Sub Publisher** from the the "Select a role" dropdown. When you're done, click "Add." Congrats, you're all done!



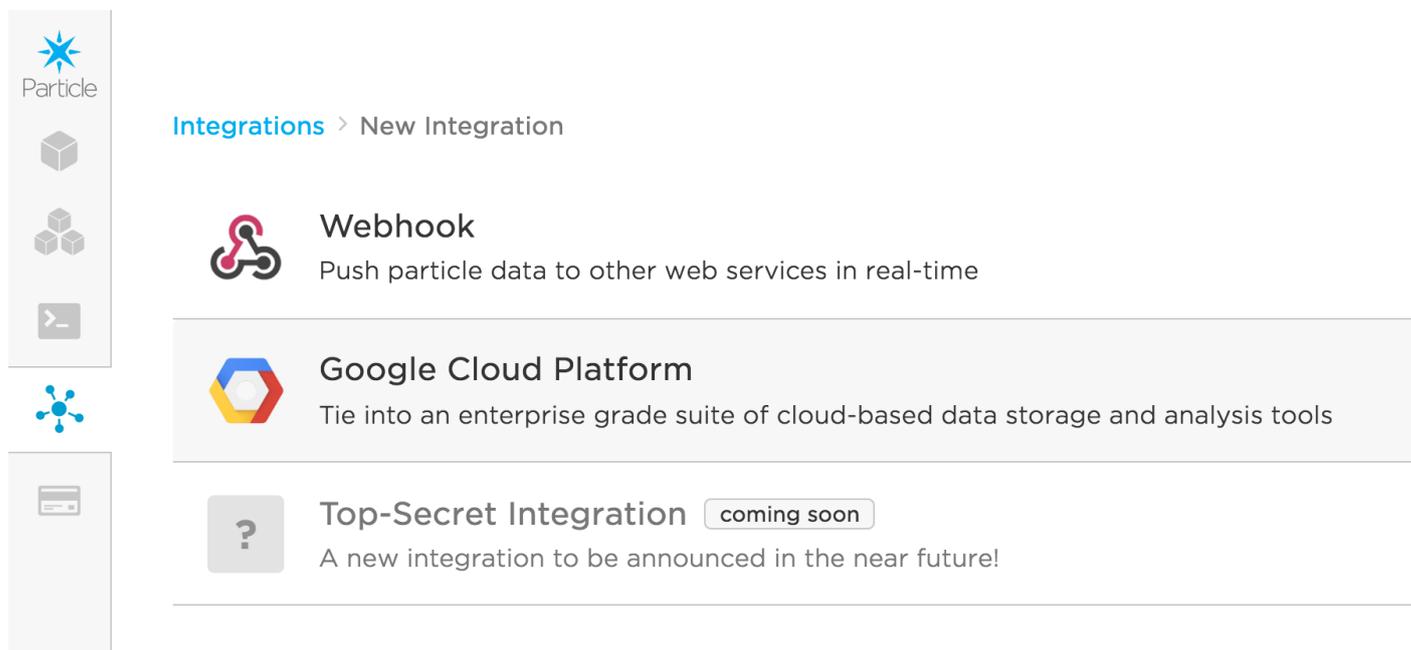
Enabling the Integration

Particle Console

Now that you've done all of the pre-configuration required, you are now ready to enable the Google Cloud Platform integration on the Particle Console.

Start by going to the integrations hub by clicking on the integrations icon in the sidebar. If you'd like to enable the integration for a product, you'll need to visit the integrations hub for the desired product. Do this by clicking the products icon () in the sidebar, finding your product, then clicking on the integrations icon () in the product context.

Once in the integrations hub, click on the "New Integration" button. From the list of available integrations, click on "Google Cloud Platform."



You'll see a reminder that setup is required before continuing to enable the integration. If you have followed the steps outlined in Preconfiguration in Google Cloud Platform, you should be good to go. Click the "I have done all these things" button to advance.

The next step is configuring the integration. Fill out the following fields:

- **Event Name:** The name of the event that will trigger publishing an event to Google Cloud Platform. This is the name of your event when you call `Particle.publish()` in your firmware.
- **Google Cloud Pub/Sub Topic:** The name of your topic that you created in your Google Cloud Pub/Sub account. This topic forwards messages from a publisher (the Particle cloud) to subscribers. The topic name is prefixed by `projects/{project_name}/topic/`.
- **Device:** Select which of your devices will trigger publishing to Google. If you'd like the publish to trigger from any of the devices you own, select 'Any.' For product-level integrations, you can instead choose if you'd like the response to the integration to be routed back to the device in the fleet that originally triggered the publish.



Particle

Integrations > New Integration > Google Cloud Platform

Docs Setup

Event Name ⓘ
my_event

Google Cloud Pub/Sub Topic ⓘ
projects/my-project/topics/my-topic

Device ⓘ
Any

ENABLE INTEGRATION

Click "Enable Integration." You have now successfully told the Particle cloud to stream data to Google Cloud Platform! Make sure that the integration is configured properly by clicking on the **TEST** button (available for integrations **not** scoped to a single device).

[Integrations](#) > [View Integration](#)



Google Cloud Platform

Event:	foo	Topic:	my-topic	TEST
ID:	57c7139d329dd26729d3651e	Created:	August 31st, 2016	

Clicking **TEST** will attempt to send a test event to the Google Cloud Platform Topic. This serves as a gut-check to determine if the integration is configured successfully

Did the test return a successful response? Great! Let's move onto firmware. If you can't get the test to succeed, double check that you used the correct topic name, and that you gave the topic the correct permissions to allow Particle to publish to it on your behalf.

Firmware

Now that the integration is enabled in the Particle cloud, the final step required to get data streaming into Google Cloud Platform is to flash a device with some firmware that publishes the targeted event. Head over to the Particle Web IDE, Local IDE, or whichever IDE you are using for firmware development.

If you're already working on a firmware application, just make sure you include a `Particle.publish()` with the event name matching the event used to enable the Google Cloud Platform integration above. Otherwise, if you need some sample firmware, paste in the below code into your firmware app:

```
// The on-board LED
int led = D7;

void setup() {
```

```

pinMode(led, OUTPUT);
}
void loop() {
  // Turn the LED Off
  digitalWrite(led, HIGH);
  // Publish an event to trigger the integration
  // Replace "my-event" with the event name you used when configuring the integration
  // Replace "test-data" with the real data you'd like to send to Google Cloud Platform
  Particle.publish("my-event", "test-data", PRIVATE);
  // Wait for 3 seconds
  delay(3000);
  // Turn the LED off
  digitalWrite(led, LOW);
  delay(3000);
}

```

The above code will publish an event every 6 seconds, when the on-board LED turns on. In reality, you might publish an event containing the readings from a temperature or humidity sensor every few minutes, when motion is detected, or when a user interacts with the device by pushing a button. This firmware is entirely meant as a sample to illustrate the minimum code needed to stream data into Google Cloud Platform.

Go ahead and flash the firmware with the `Particle.publish()` that will trigger the integration to a Particle device.

Once confident in the firmware, you can stream data from large numbers of devices by rolling out the firmware to a product fleet. Remember that this requires creating the integration under the product scope, allowing any device in the product fleet to trigger the it.

Congrats! This is all you need to get the integration working end-to-end. Your device will now begin to publish the targeted event, which will signal to the Particle cloud to stream the contents of the event to Google Cloud Platform.

Confirming the data reaches Google Cloud Platform

To ensure that the data is successfully being published to Google Cloud Platform, you can use the Google Cloud CLI. To use the Google Cloud CLI, click on the terminal icon on the right side of the top navigation bar as shown below:



After launching the CLI, you can verify messages coming through by creating a *subscription* to your Google Cloud Pub/Sub topic name. Any other Google Cloud Platform product that you eventually use will also be subscribers of your topic's messages. This subscriber, however, will just be used to acknowledge that messages are entering Google Cloud Platform successfully.

To create a subscription:

```
~$ gcloud beta pubsub subscriptions create test_sub --topic [topic_name]
```

Where `topic_name` is the name of your Google Cloud Pub/Sub topic (don't include the `projects/{project_name}/topic/` prefix, just the topic name). This subscriber will now listen for all messages published to your topic.

Now, to pull some published messages:

```
~$ gcloud beta pubsub subscriptions pull test_sub --auto-ack --max-messages 100
```

If things are looking good, you should see a table like this:

DATA	MESSAGE_ID	ATTRIBUTES
null	1	device_id=123abc event=test published_at=[timestamp]
null	2	device_id=123abc event=test published_at=[timestamp]
null	3	device_id=123abc event=test published_at=[timestamp]

Nice! Everything looks to be wired up correctly. We've proven that we are now successfully getting data from a Particle device -> Particle Cloud -> Google Cloud Pub/Sub. From here, it's up to you how you'd like to leverage Google Cloud Platform's products & services to add value to your connected product. The rest of this tutorial will focus on specific IoT use cases for Google Cloud Platform.

Example Use Cases

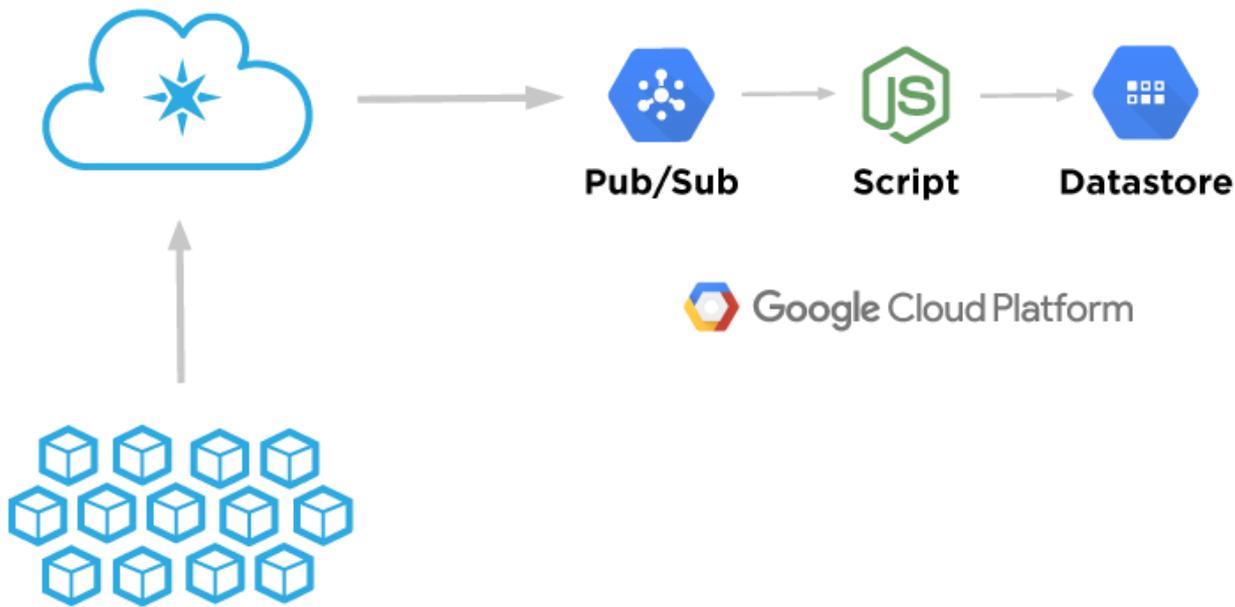
Storing Data in a Datastore Database

One relatively universal application of the Particle & Google Cloud Platform integration is the ability to store device data into a long-term database. Google Cloud offers many different data storage options within the platform, but this tutorial will show you how to get data into a Google Cloud Datastore.

Datastore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. A key advantage of using Datastore is the fact that its *schemaless*: this means it is flexible in the data structure it will accept, and does not require configuration in advance. Google Cloud Datastore also features high availability of reads and writes, massive scalability and performance, and automatic data encryption. For all of Datastore's features, check out the docs.

Before starting with this tutorial, it is important that you first follow the setup steps above, both preconfiguring in Google Cloud Platform and enabling the integration.

For this example, we will be running a small Node.js script to act as an intermediary to subscribe to a Google Cloud Pub/Sub topic, and pass the data into a Datastore database. Events will flow from devices, to the Particle Cloud, and into Google Cloud Pub/Sub. Then, the Node script will funnel those events into Datastore. See below for the high-level architecture of what this example will entail:



Creating a Private Key

The Node script will need to have the correct permissions to both subscribe to Google Cloud Platform topics as well as insert data into Datastore. For this to work, we'll need to create a private key that your script will use to authenticate. Google refers to these private keys for external server authentication as service accounts.

To create a service account:

1. Open the list of credentials in the Google Cloud Platform Console.
OPEN LIST OF CREDENTIALS
2. Click **Create credentials**
3. Select **Service account key**
4. Click the drop-down box below *Service account*, then click **New service account**
5. Enter a name for the service account in **Name**
6. From the **Role** dropdown, select **Project > Editor**
7. Use the default **Service account ID** or generate a different one
8. Select **JSON** as the **Key type**
9. Click **Create**

Now, a **Service account created** window is displayed and the private key for the **Key type** you selected should be downloaded automatically. This file is the private key that your script will use to authenticate with Google Cloud Platform.

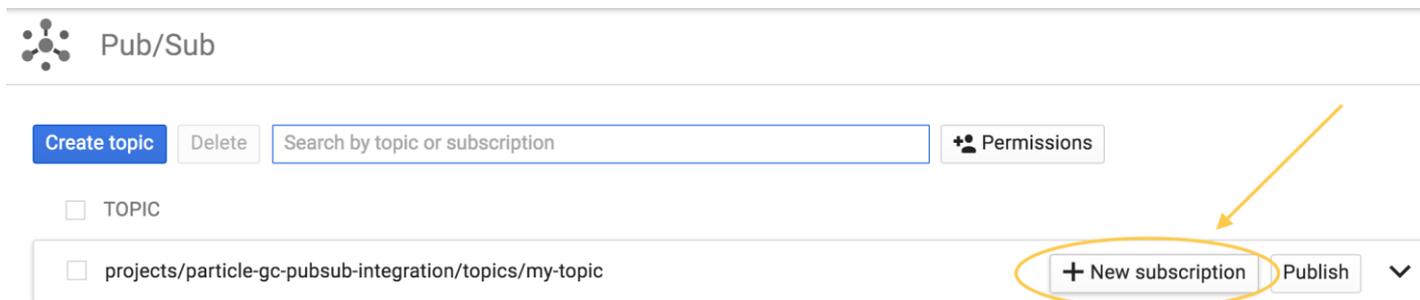
NOTE: Do not share this private key, or check it into Git. Keep it secret.

Creating a Pub/Sub Subscription

The next thing you'll need is a Google Cloud Pub/Sub subscription. Remember that the Particle Cloud *publishes to a topic* in Google Cloud. A *subscriber* listens for messages published to a topic. We'll need a subscriber in order to funnel events from the topic to the Datastore database.

To create a topic,

1. Visit the Pub/Sub Homepage within the Google Cloud Platform Console.
2. Find your topic in the list and click **New Subscription**. It is important that this topic name matches the topic you added as part of enabling the Google Cloud Platform Integration on Particle.



The "New Subscription" button will appear when hovering over your topic

1. Give your subscription a **Name**, and set the **Delivery Type** to **Pull**. Your subscription name will automatically be prefixed just as your topic was.
2. Click **Create**

Running the Node Script

We'll need an intermediary to allow the Particle device data to flow from Google Cloud Pub/Sub to Datastore. This is where the Node script comes into play. It's job is simple:

- Subscribe to events published to your Pub/Sub Topic
- Route those events into Datastore

With the help of our friends at Google Cloud Platform, we've created an open-source repository that has all the code you need to do just this.

CHECK OUT THE REPO

The README of the repository has all the information you need to get up and running. You'll need the private key file and the subscription name you created earlier.

Checking out the data

If all goes well, you should see output like this when running the script:

```
Particle event received from Pub/Sub!  
{ gc_pub_sub_id: '62230504*****',  
  device_id: '3e003f0005473*****',  
  event: 'led-off',  
  data: null,  
  published_at: '2016-09-13T17:08:46.549Z' }
```

```
Particle event stored in Datastore!  
{ gc_pub_sub_id: '62230504*****',  
  device_id: '3e003f0005473*****',  
  event: 'led-off',  
  data: null,
```

```
published_at: '2016-09-13T17:08:46.549Z' }
```

This confirms that the data has successfully been captured in Datastore. After some events have come through successfully, head over to Datastore.

Here's an example of how the Particle data in Datastore will look:

Kind

MessageID Filter entities

Number of columns to display 50

<input type="checkbox"/> Name/ID	data	device_id	event	published_at
<input type="checkbox"/> name=62211010816762	null	3e003f000	led-off	2016-09-12T22:16:47.128Z
<input type="checkbox"/> name=62211028779959	null	3e003f000	led-on	2016-09-12T22:16:42.624Z
<input type="checkbox"/> name=62211028875157	null	3e003f000	led-on	2016-09-12T22:17:27.638Z
<input type="checkbox"/> name=62211028885922	null	3e003f000	led-on	2016-09-12T22:17:33.641Z
<input type="checkbox"/> name=62211031787163	null	3e003f000	led-off	2016-09-12T22:17:23.137Z
<input type="checkbox"/> name=62211054123889	null	3e003f000	led-off	2016-09-12T22:16:56.128Z
<input type="checkbox"/> name=62211054179991	null	3e003f000	led-off	2016-09-12T22:17:20.136Z
<input type="checkbox"/> name=62211082086996	null	3e003f000	led-on	2016-09-12T22:17:15.636Z
<input type="checkbox"/> name=62211082628506	null	3e003f000	led-on	2016-09-12T22:17:00.632Z

Note the following:

- data is the data send with the Particle.publish() event from a device
- device_id is the ID of the device that sent the data
- event is the Particle event name that triggered the integration
- published_at is a timestamp of when the event was sent

Congratulations! You are storing Particle device data in Google Cloud Platform.