

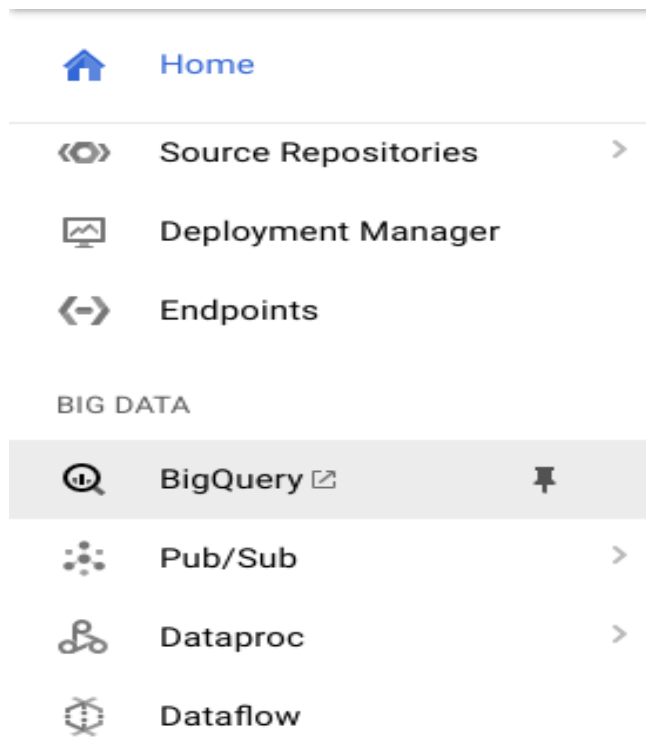
CREATE A SERVERLESS IOT DATA PROCESSING



CREATE A BIG QUERY TABLE:

Big Query is a serverless, highly scalable, low-cost enterprise data warehouse and will be an ideal option to store data being streamed from IoT devices while also allowing an analytics dashboard to query the information.

Let's create a table that will hold all the IoT weather data. Select Big Query from the Cloud console. **This will open Big Query in a new window** (don't close the original window as you'll need to access it again).



Click on the down arrow icon next to your project name and then select "Create new dataset"

Enter "weather Data" for the Dataset, select a location where it will be stored and Click "OK".

A screenshot of the 'Create Dataset' dialog box in Google Cloud Platform. The dialog has a title bar 'Create Dataset' with a close button (X) on the right. It contains three main sections: 'Dataset ID' with a text input field containing 'weatherData' and a help icon (?); 'Data location' with a dropdown menu showing 'US' and a help icon (?); and 'Data expiration' with two radio buttons: 'Never' (selected) and 'In' followed by an empty text input field and the word 'days.'. At the bottom, there are two buttons: 'OK' (blue) and 'Cancel' (gray).

now have a data warehouse setup to receive your weather data.

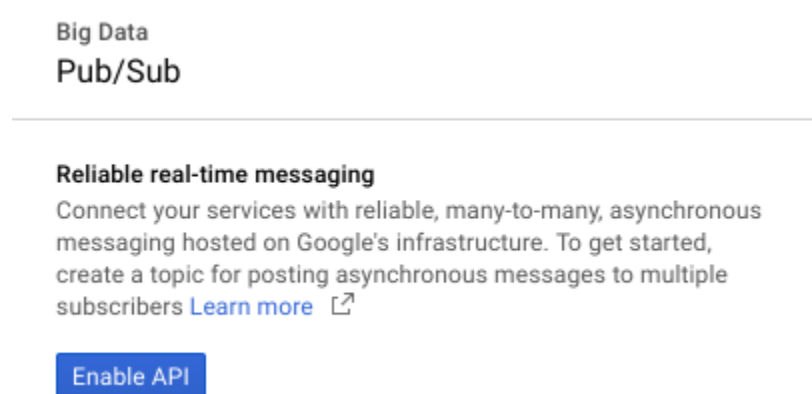
CREATE A PUB/SUBTOPIC:

Cloud pub/sub is a simple, reliable, scalable foundation for stream analytics and event-driven computing systems. As a result, it is perfect for handling incoming IOT messages and then allowing downstream systems to process them.

If you are still in the window for Big Query, switch back to the Cloud Console. If you closed the Cloud Console, go to <https://console.cloud.google.com>

From the Cloud Console, select Pub/Sub and then Topics.

If you see an Enable API prompt, click the Enable API button.




Click on the Create a topic button


Big Data Pub/Sub


Reliable real-time messaging

Connect your services with reliable, many-to-many, asynchronous messaging hosted on Google's infrastructure. To get started, create a topic for posting asynchronous messages to multiple subscribers [Learn more](#) ↗

Create a topic

 Pub/Sub

 Topics


 Subscriptions

Topics [+ CREATE TOPIC](#) [DELETE](#)

Filter by topic name

<input type="checkbox"/> Topic name	Subscriptions	
<input type="checkbox"/> projects/iot2analytics/topics/weatherdata	0	⋮

After the key upload is complete, it should appear in the Cloud Storage browser.

Browser ↑ UPLOAD FILES ↑ UPLOAD FOLDER CREATE FOLDER REFRESH SHARE PUBLICLY DELETE						
Filter by prefix...						
Buckets / keystore-iot2analytics						
<input type="checkbox"/> Name	Size	Type	Storage class	Last modified	Share publicly	
<input type="checkbox"/>  lot2Analytics-a68b9ef614d3.json	2.28 KB	application/octet-stream	Multi-Regional	1/29/18, 1:42 PM	<input type="checkbox"/>	⋮

Upload 1 of 1 complete			—	×
lot2Analytics-a68b9ef614d3.json	Finished	×		

CREATE A CLOUD FUNCTION:

Cloud computing has made possible fully serverless models of computing where logic can be spun up on-demand in response to events originating from anywhere. For this lab, a Cloud Function will start each time a message is published to the weather topic, will read the message and then store it in BigQuery.

Cloud Functions		Overview	CREATE FUNCTION	REFRESH	DELETE	COPY
Filter functions						Columns
<input type="checkbox"/>	Name ^	Region	Trigger	Memory allocated	Executed function	Last deployed
<input type="checkbox"/>	function-weatherPubSubToBQ	us-central1	topic: weatherdata	256 MB	subscribe	1/29/18, 3:34 PM

We just connected Pub/Sub to Big Query via Functions.

START THE DATA PIPELINE:

Might need to enable compute API.

dataflow-gcs-to-pubsub

LOGS

Read Text Data

Running

Write to PubSub

Running

Job

Job summary

Job name

dataflow-gcs-to-pubsub

Job ID

2018-02-01_08_54_59-14441071159444143835

Region

us-central1

Job status

Running

Stop job

SDK version

Apache Beam SDK for Java 2.2.0

Job type

Batch

Start time

Feb 1, 2018, 9:55:00 AM

Elapsed time

40 sec

Autoscaling

...

Resource metrics

Current vCPUs

0

Total vCPU time

0 vCPU hr

Current memory

0 B

Total memory time

0 GB hr

Current PD

0 B

Total PD time

0 GB hr

Current SSD PD

0 B

Total SSD PD time

0 GB hr

Total shuffle usage

-

CHECK THAT DATA IS FLOWING:

CLOUD FUNCTION LOGS:

Ensure that the Cloud Function is being triggered by Pub/Sub

gcloud beta functions logs read function-weatherPubSubToBQ

New Query ? Query Editor UDF Editor ×

```
1 SELECT * FROM [iot2analytics:weatherData.weatherDataTable] LIMIT 1000
```

RUN QUERY Save Query Save View Format Query Show Options Query complete (1.2s elapsed, 0 B processed) ✓

Ctrl + Enter: run query, Tab or Ctrl + Space: autocomplete.

Results Details Download as CSV Download as JSON Save as Table Save to Google Sheets

Row	sensorID	timecollected	zipcode	latitude	longitude	temperature	humidity	dewpoint	pressure
1	s-testing	2018-01-30 20:02:11.000 UTC	94043	37.421655	-122.085637	70.86	18.33	41.46	24.19
2	s-testing	2018-01-30 20:11:13.000 UTC	94043	37.421655	-122.085637	70.85	18.24	41.42	24.18
3	s-testing	2018-01-30 20:06:12.000 UTC	94043	37.421655	-122.085637	70.89	18.28	41.47	24.19
4	s-testing	2018-01-30 20:09:13.000 UTC	94043	37.421655	-122.085637	70.86	18.23	41.43	24.18
5	s-testing	2018-01-30 20:08:12.000 UTC	94043	37.421655	-122.085637	70.88	18.27	41.46	24.18
6	s-testing	2018-01-30 20:15:14.000 UTC	94043	37.421655	-122.085637	70.87	18.26	41.45	24.18
7	s-testing	2018-01-30 20:01:11.000 UTC	94043	37.421655	-122.085637	70.9	18.58	41.59	24.19
8	s-testing	2018-01-30 20:14:14.000 UTC	94043	37.421655	-122.085637	70.83	18.22	41.39	24.18
9	s-testing	2018-01-30 20:13:14.000 UTC	94043	37.421655	-122.085637	70.86	18.25	41.43	24.18

Table JSON [First](#) [< Prev](#) Rows 1 - 9 of 16 [Next >](#) [Last](#)

CREATE A DATA STUDIO DASHBOARD:

Google data studio turns our data into informative dashboards and reports that are easy to read, easy to share, and fully customizable.

weatherDataTable Field Editing in Reports: ON USING OWNER'S CREDENTIALS ? CREATE REPORT

← EDIT CONNECTION +

Index	Field	Type	Aggregation	Description
1	zipcode	123 Number	None	
2	dewpoint	123 Number	Average	
3	timecollected	Date Hour (YYYYMMDD...)	None	
4	latitude	123 Number	None	
5	temperature	123 Number	None	
6	humidity	123 Number	Average	
7	pressure	123 Number	Average	
8	sensorID	88C Text	None	
9	longitude	123 Number	Average	

