

# Step 6 Consume Data from IoT Core to Google BigQuery

Google IoTCore support several ways to consume data.

## Google Cloud SDK

---

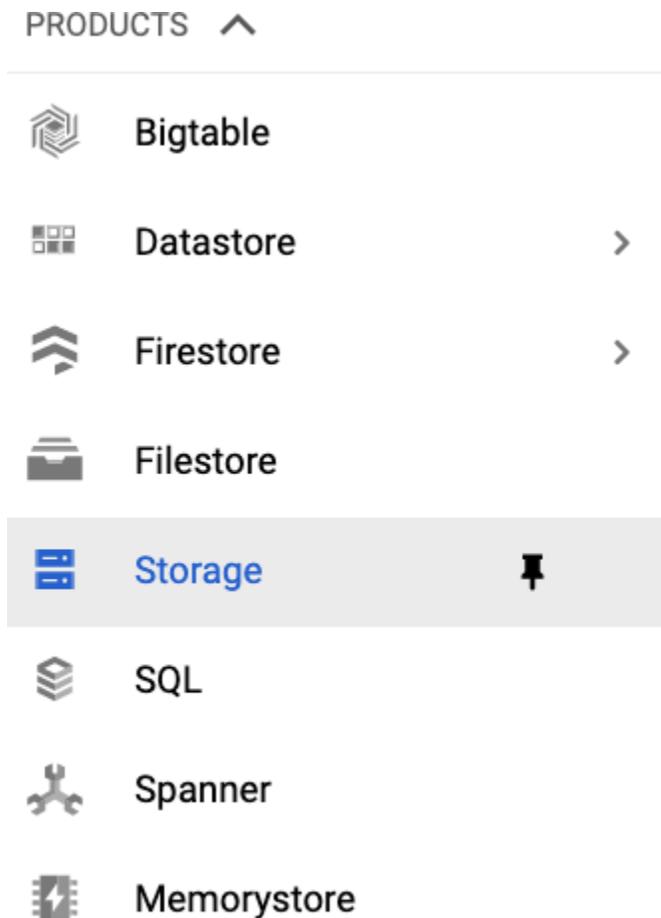
Google provide you with a client libraries for all major languages to push and pull data from Pub/Sub see the link for more information: <https://cloud.google.com/sdk/docs/>

## Google Cloud Services

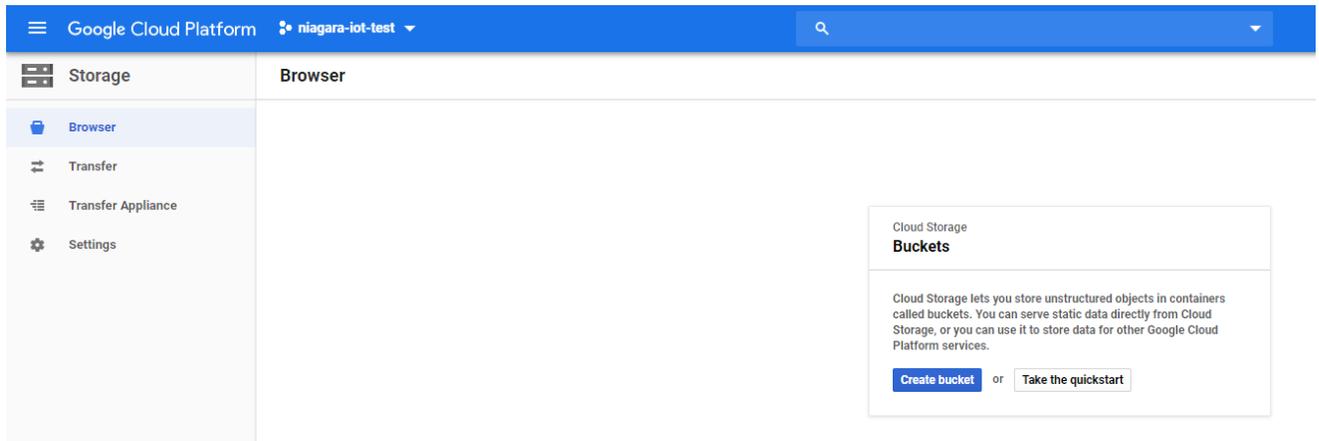
---

Google IoTCore has full integration with all major Google Cloud services, in this section we will focus on **how to push data to BigQuery**

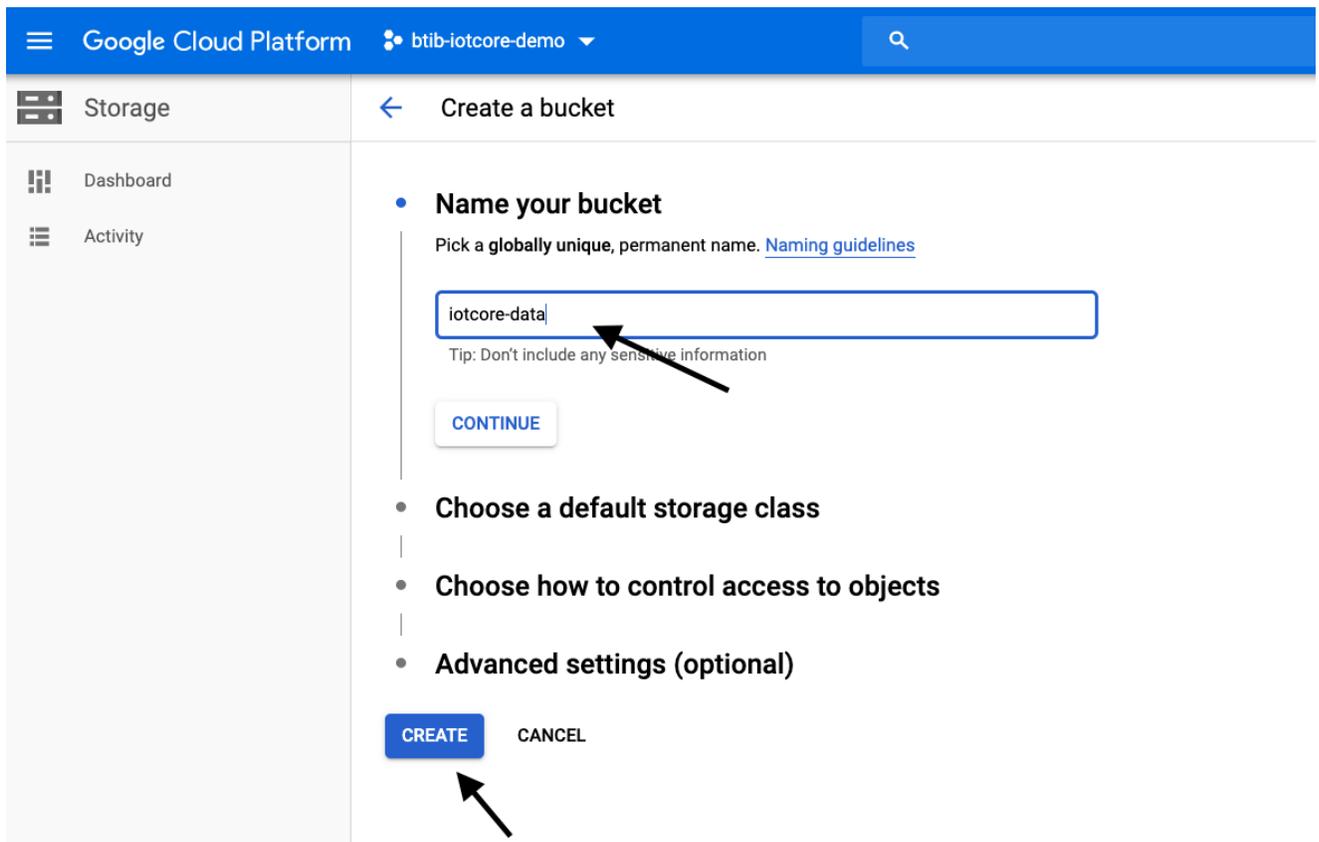
1. Go to the cloud storage service.



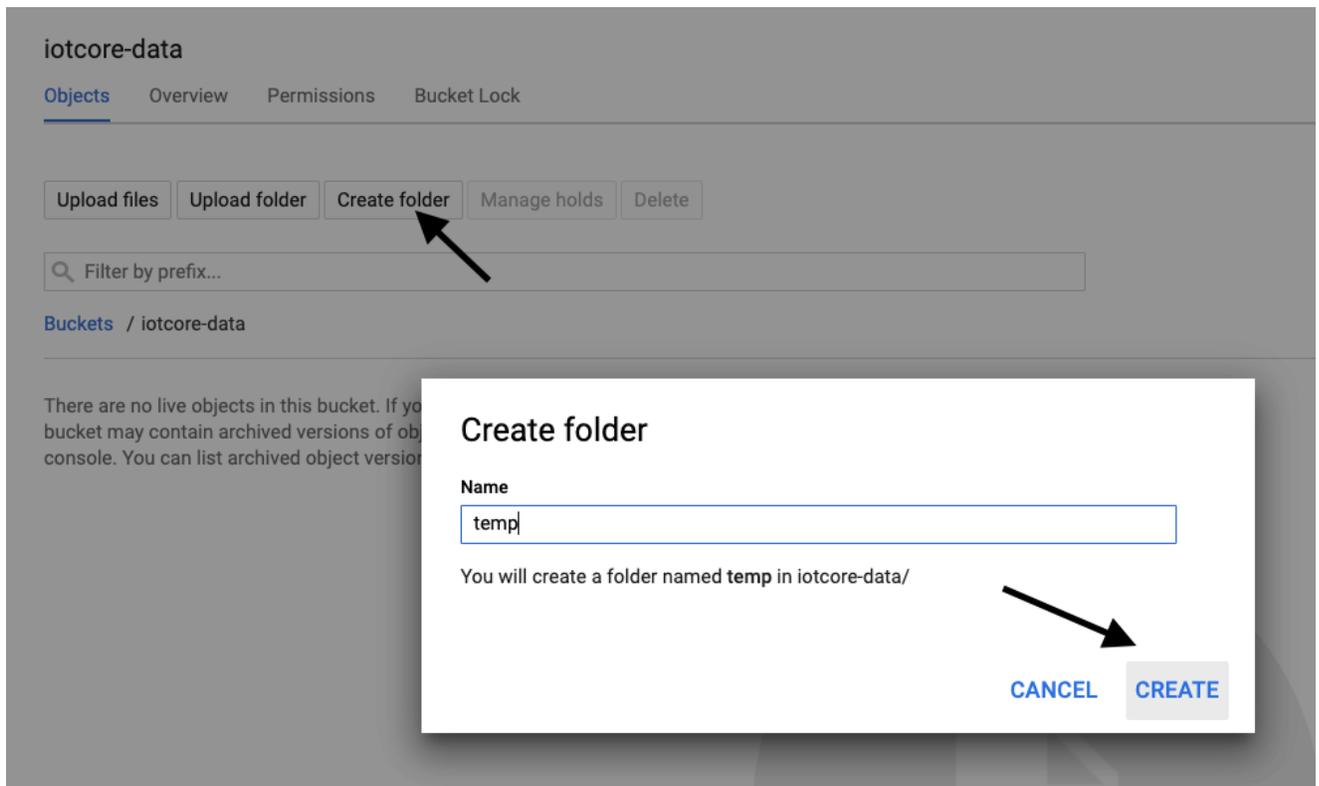
2. Create a new bucket.



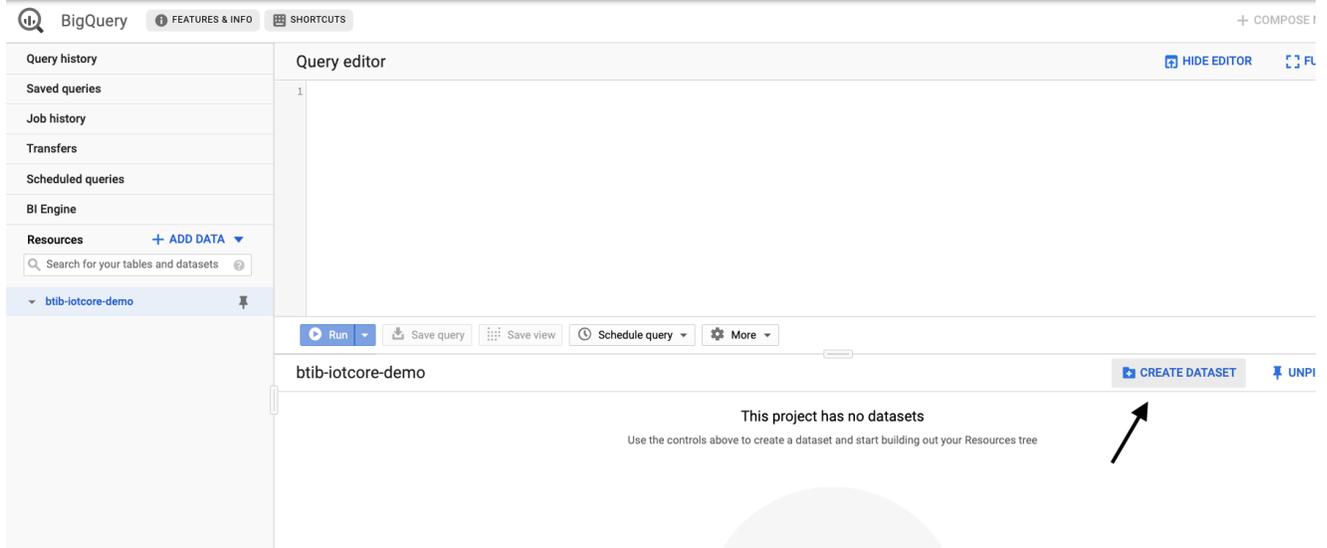
3. Give the bucket a name and hit "Create" (this bucket will be used by the DataFlow service to export IoTCore data to BigQuery)



4. Create a temp folder in this bucket.



5. Open the BigQuery service (in the left panel), select your resource (on the left menu), then create a dataset.



6. Give it a name and click on "Create"

## Create dataset

**Dataset ID**

**Data location (Optional) ?**

**Default table expiration ?**

Never

Number of days after table creation:

7. Select the dataset on the left and create a table.

The screenshot shows the Databricks workspace interface. On the left, the 'Resources' sidebar is open, showing a search bar and a list of resources under the 'btib-iotcore-demo' folder. The 'iotcoredata' dataset is selected and highlighted in blue. An arrow points to this selection. In the main workspace area, the breadcrumb path is 'btib-iotcore-demo:iotcoredata'. Below this, there are sections for 'Description' (None) and 'Labels' (None). At the top right of the main area, there is a '+ CREATE TABLE' button and a 'SHARE D...' button. An arrow points to the '+ CREATE TABLE' button.

8. Give the table a name and add these columns (those are json fields of the message sent by niagara).

## Create table

### Source

Create table from:

Empty table ▼

### Destination

Project name

btib-iotcore-demo ▼

Dataset name

iotcoredata ▼

Table type ?

Native table ▼

Table name

events

### Schema

Edit as text

Name

pointId 

Type

STRING ▼

Mode

NULLABLE ▼

✕

status

STRING ▼

NULLABLE ▼

✕

timestamp

TIMESTAMP ▼

NULLABLE ▼

✕

value

STRING ▼

NULLABLE ▼

✕

[+ Add field](#)

### Partition and cluster settings

Partitioning: ?

No partitioning ▼

Clustering order (optional): ?

Clustering order determines the sort order of the data. Clustering can only be used on a partitioned table, and works with tables partitioned either by column or ingestion time.

Comma-separated list of fields to define clustering order (up to 4)

Create table

Cancel

- Now go to the pub/sub service.
- Choose the topic you want to export to BigQuery.

Pub/Sub		Topics	<a href="#">+ CREATE TOPIC</a>	<a href="#">DELETE</a>
<a href="#">Dashboard</a>		<a href="#">Filter table</a>		
<a href="#">Activity</a>		<input type="checkbox"/> Topic name ↑		Encryption
		<input type="checkbox"/> events		Google-managed
		<input type="checkbox"/> metadata		Google-managed

11. Hit Export to BigQuery.

← Topic details    + PUBLISH MESSAGE    🔍 PULL MESSAGES    📄 IMPORT    📤 EXPORT    🗑️ DELETE

**events**

- BigQuery
- Cloud Storage text file
- Cloud Storage Avro file

1 hour   6 hours   1 c

**Publish message count**

**Publish bytes**

12. Now in the DataFlow

- Give a name to the job
- Choose a region where the data pipeline will be created
- Under "BigQuery output table" set your table path
- Under Temporary location set the path of the folder we created before on storage service.
- Then start the job: [Example: gs://iotcore-data/temp](gs://iotcore-data/temp)



### Create a Dataflow job to export data from Cloud Pub/Sub Topic to BigQuery

**Job name**

Must be unique among running jobs. Use lowercase letters, numbers, and hyphens (-).

**Cloud Dataflow template**

A pipeline that ingests a Cloud Pub/Sub stream of JSON-encoded messages from a Pub/Sub Topic, performs a transform via a user defined JavaScript function, and writes to a pre-existing BigQuery table.

**Required Parameters**

**Regional endpoint**

Choose where to deploy Cloud Dataflow workers and store metadata for the job.

**Cloud Pub/Sub input topic**

Cloud Pub/Sub topic to read the input from, in the format of 'projects/<project>/topics/<topic>'

**BigQuery output table**

BigQuery table location (<project>:<dataset>.<table\_name>) to write the output to. The table's schema must match the input JSON objects.

**Temporary location**

Path and filename prefix for writing temporary files. ex: gs://MyBucket/tmp

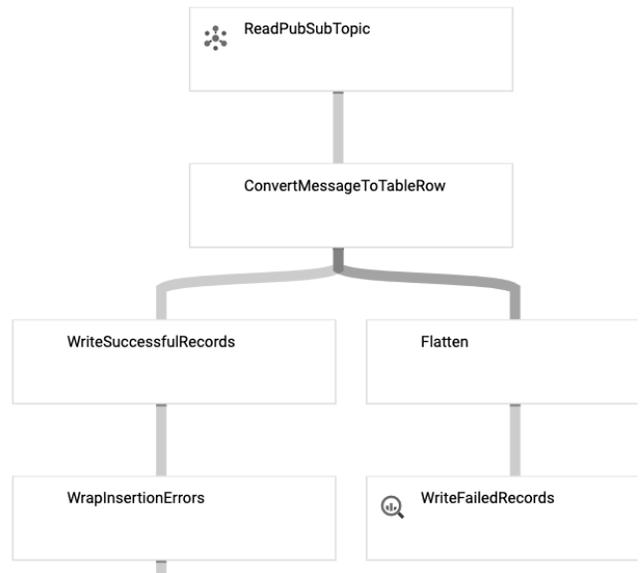
Optional parameters

Once you run this job, you can view its status on the next screen to confirm that no errors occurred and all data exported successfully. You can also stop it at any time.

This streaming pipeline will cost you between \$0.40 and \$1.20 per hour in the us-central1 region...

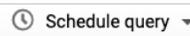
More



13. Type the query below in the query console and run it, you should see your data.

## Query editor

```
1 SELECT
2 *
3 FROM
4 btib-iotcore-demo.iotcoredata.events
5 LIMIT
6 1000
```

## Query results

Query complete (1.4 sec elapsed, 0 B processed)

Job information **Results** JSON Execution details

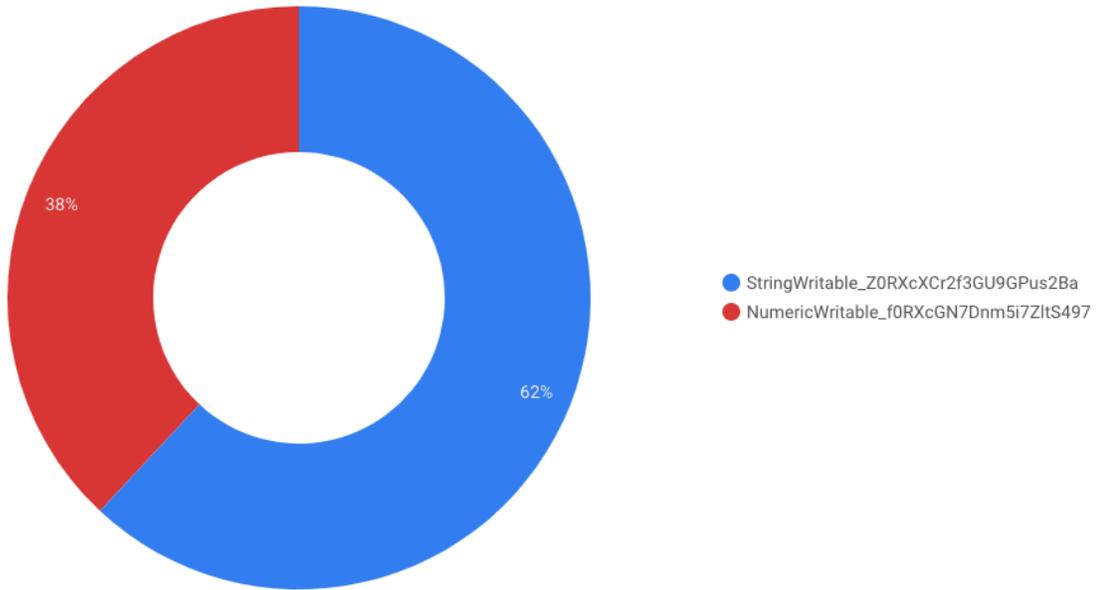
Row	pointId	status	timestamp	value
1	StringWritable_Z0RXcXCr2f3GU9GPus2Ba	{ok} @ 10	2019-07-31 15:33:10.785 UTC	8.40
2	StringWritable_Z0RXcXCr2f3GU9GPus2Ba	{ok} @ 10	2019-07-31 15:30:21.623 UTC	6.12
3	NumericWritable_f0RXcGN7Dnm5i7ZItS497	{ok} @ 10	2019-07-31 15:31:33.949 UTC	6.7021137809609179
4	NumericWritable_f0RXcGN7Dnm5i7ZItS497	{ok} @ 10	2019-07-31 15:32:45.283 UTC	2.3066844948135179
5	NumericWritable_f0RXcGN7Dnm5i7ZItS497	{ok} @ 10	2019-07-31 15:30:18.579 UTC	1.0266762244881613
6	StringWritable_Z0RXcXCr2f3GU9GPus2Ba	{ok} @ 10	2019-07-31 15:31:18.669 UTC	4.91
7	NumericWritable_f0RXcGN7Dnm5i7ZItS497	{ok} @ 10	2019-07-31 15:32:48.342 UTC	7.8854610699522665
8	StringWritable_Z0RXcXCr2f3GU9GPus2Ba	{ok} @ 10	2019-07-31 15:33:55.506 UTC	1.02

## Tips

- To visualize your data click on the button **EXPLORE WITH DATA STUDIO** (between the query editor and the console with the results)

Add a chart ▾

Filter  +



- To prepare, clean and transform your data you can use a Dataprep service (ETL service).