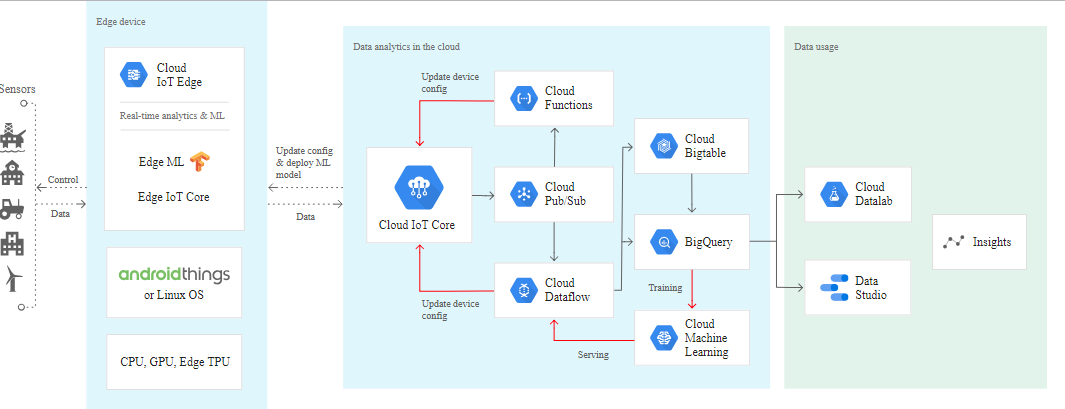
**Setup Guide for FT900 Google Cloud IoT demo**

**I. GOOGLE CLOUD IoT Architecture**

****

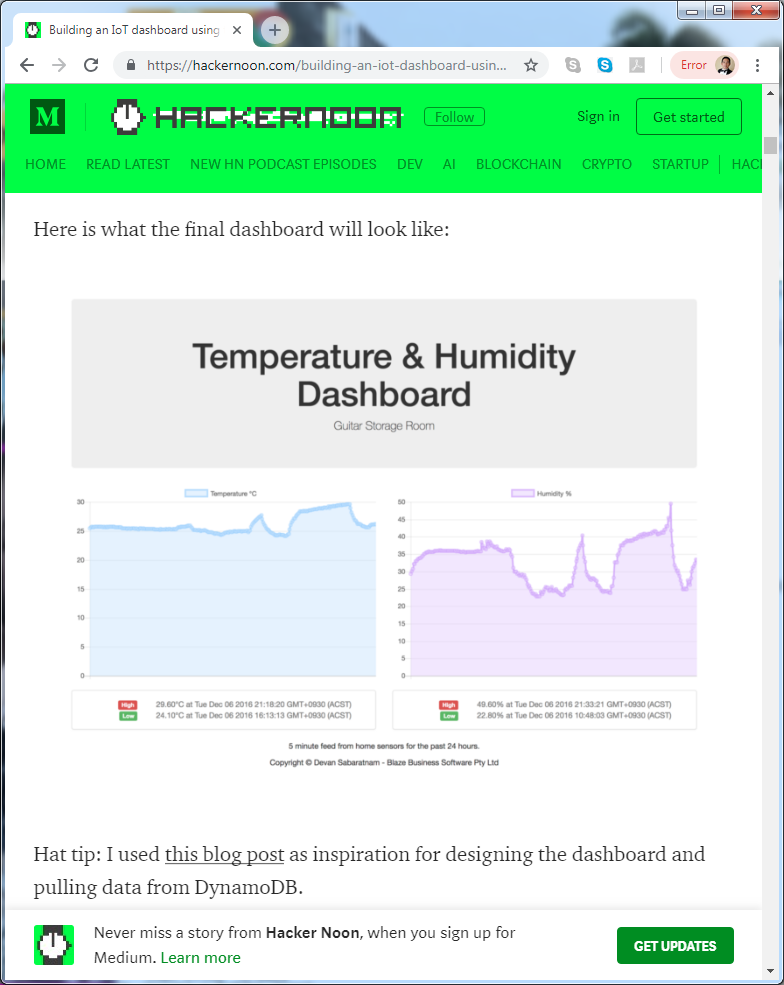
BACK-END: **FT900 -> IoT Core -> Pub/Sub -> Dataflow -> BigQuery**

FRONT-END: **Node.JS script-> BigQuery 🡨 [OK: CURRENT STATUS]**

FRONT-END: **Browser -> [Dashboard webpage (w/Node) in] Storage -> BigQuery 🡨 [NG: TODO]**

TODO: Integrate the provided Node.JS script, bigqueryclient.js to sample AWS IoT Dashboard at

<https://hackernoon.com/building-an-iot-dashboard-using-the-onion-omega-and-amazon-aws-a3520f850c9>

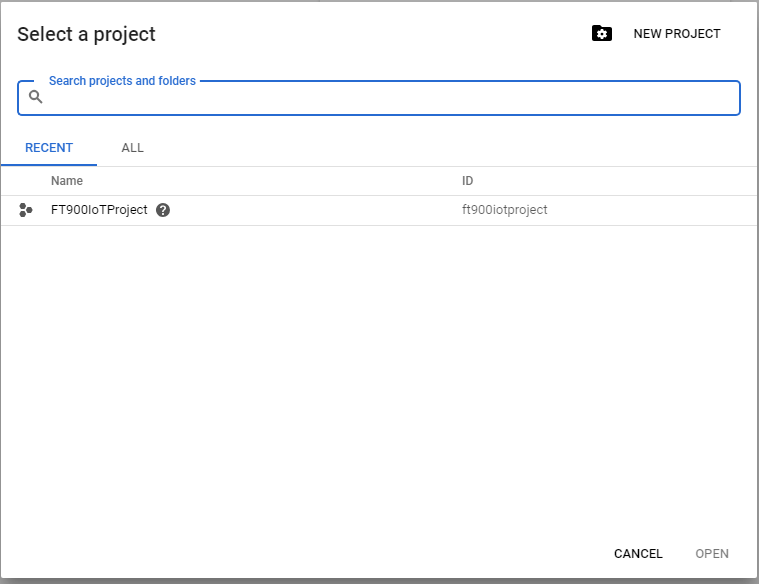


**II. BACKEND CLOUD Setup Guide:**

A. Create a Project "FT900IoTProject"

1. Click "Select a project" then click "New Project".

2. Type Project Name and click Create.



B Create a Registry "FT900RegistryID"

1. Select IoT Core and click "Enable API"

2. Click "Create a device registry"

3. Fill in the details:

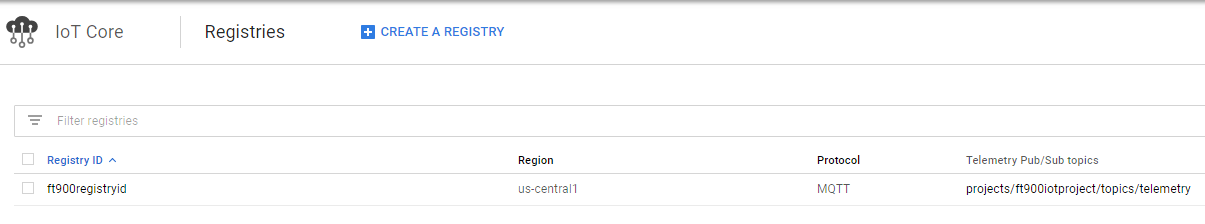
Registry ID: FT900RegistryID

Region: us-central1

Protocol: MQTT

Default telemetry topic: projects/ft900iotproject/topics/telemetry

Then click "Create"



C. Create a Device "FT900Device1"

1. Click "Create device"

2. Fill in the details:

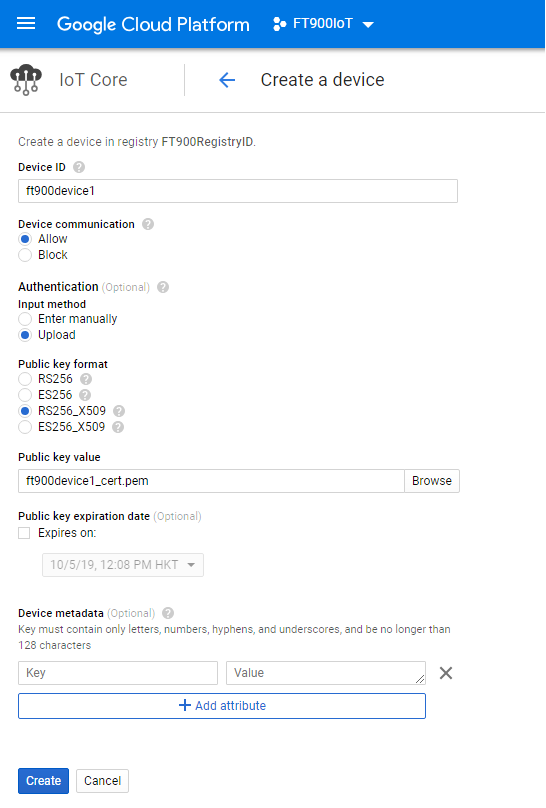
Device ID: "FT900Device1"

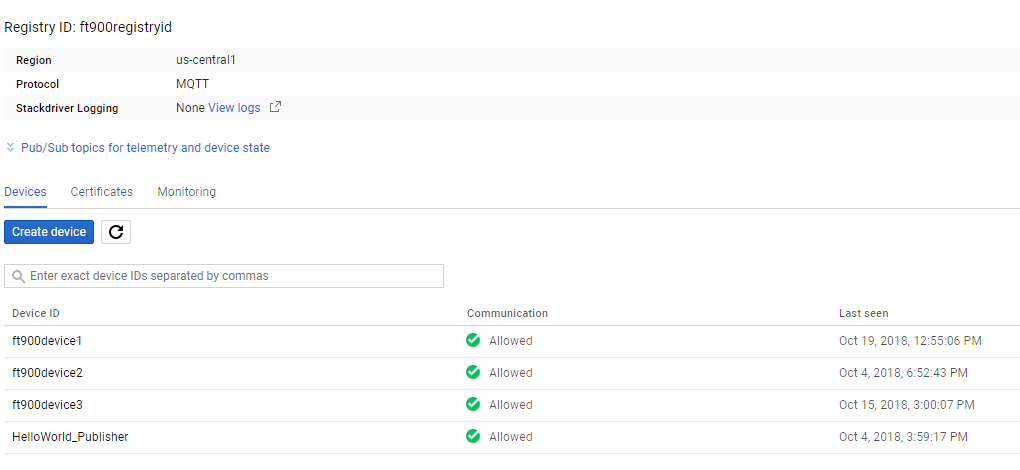
Input Method: Upload

Public Key Format: RS256\_X509

Public Key Value: Point to "ft900device1\_cert.pem"

Then click "Create"





D. Link IoTCore to PubSub

1. Select Pub/Sub and click "Enable API"

E. Create a Storage bucket "ft900storagebucket"

1. Select Storage and click "Create bucket"

2. Fill in the details:

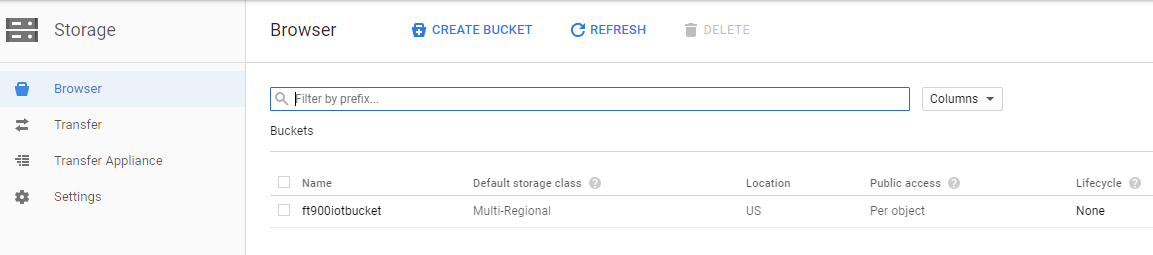
Name: "ft900storagebucket"

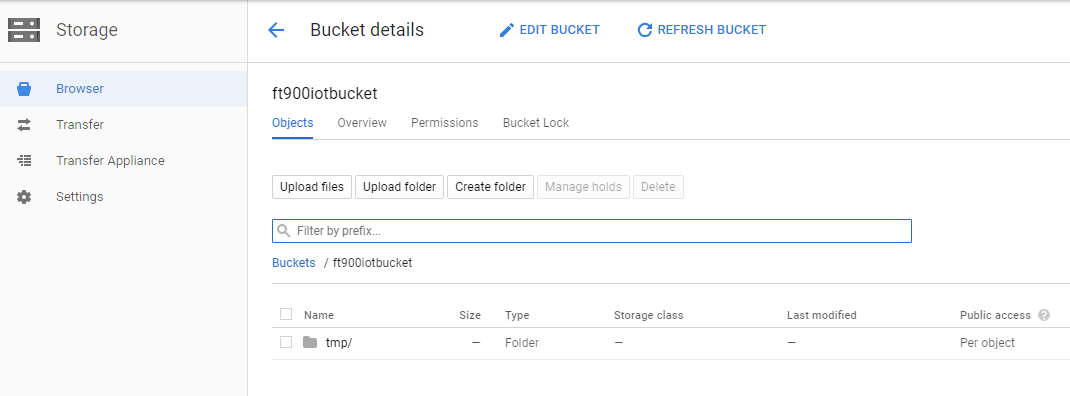
Default storage class: Multi-regional

Location: United States

Then click "Create"

3. Create a folder "tmp"





F. Create a BigQuery table "ft900iotproject:dataset.timeseriestable"

1. Select BigQuery

2. Click "FT900IoTProject" and click "Create Dataset"

Fill in the details:

Dataset ID: dataset

Default table expiration: 1 day

Then click "Create dataset".

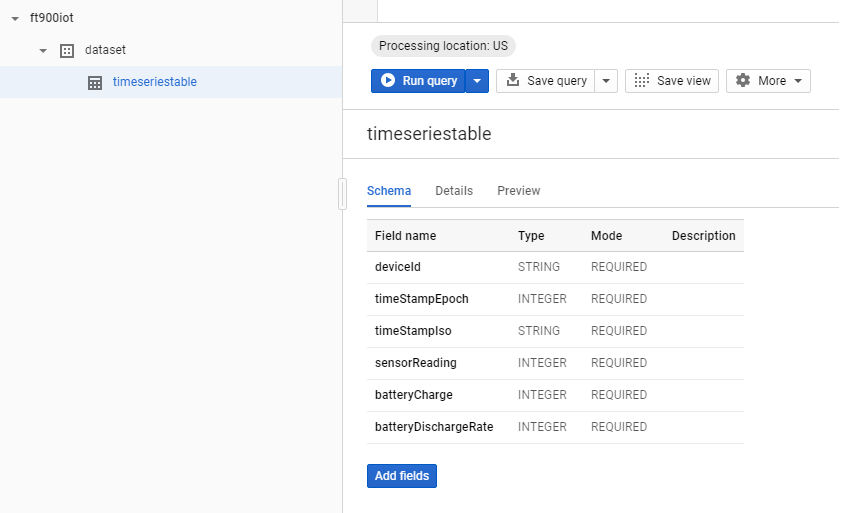
3. Under the dataset created, click "Create Table"

Fill in the details:

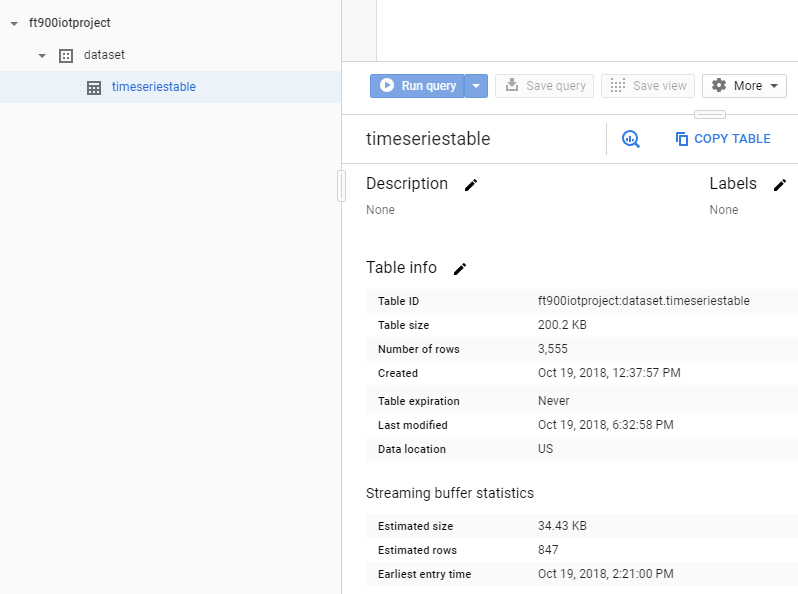
Destination table: timeseriestable

Add the following fields as in the screenshot below:

Then click "Create table"



4. Update the table expiration to NEVER.



G. Create a Dataflow job to link Pub/Sub to BigQuery

1. Select Dataflow

2. Click "Create Job From Template"

Fill in the details:

Job name: ft900dataflow

Cloud Dataflow template: PubSub to BigQuery

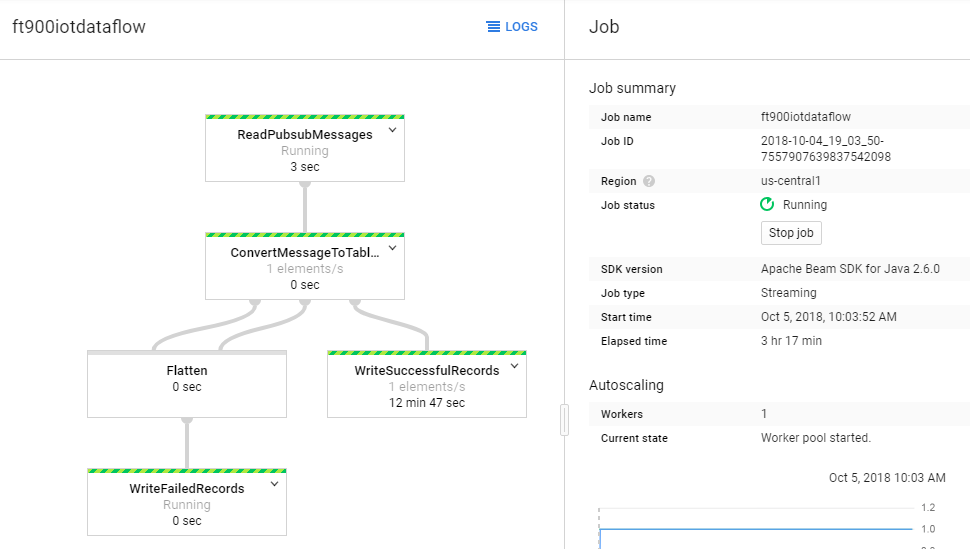
Regional endpoint: us-central1

Cloud Pub/Sub input topic: projects/ft900iotproject/topics/telemetry

BigQuery output table: ft900iotproject:dataset.timeseriestable

Temporary Location: gs://ft900iotbucket/tmp

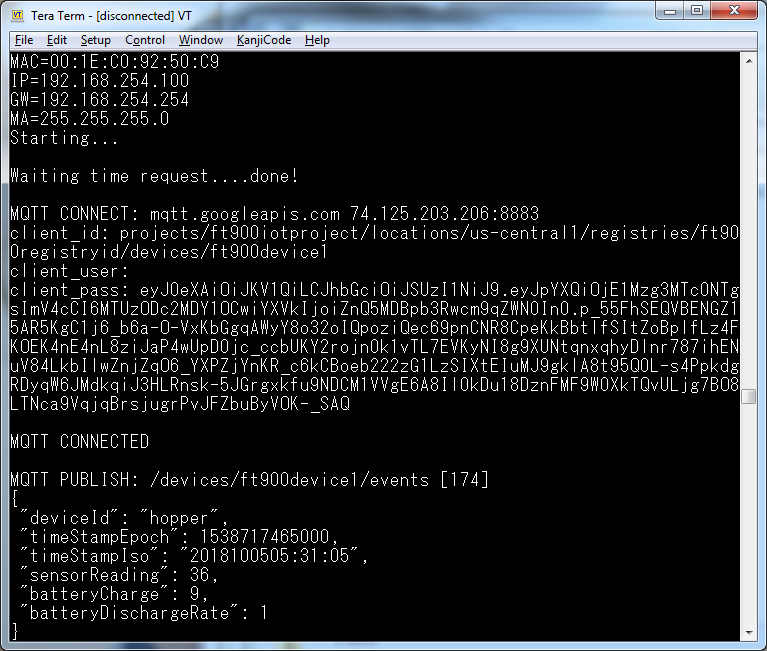
Then click "Run job"



H. Verify FT900 MQTT messages are saved in BigQuery

1. Run FT900 IoT Demo for 1 minute.

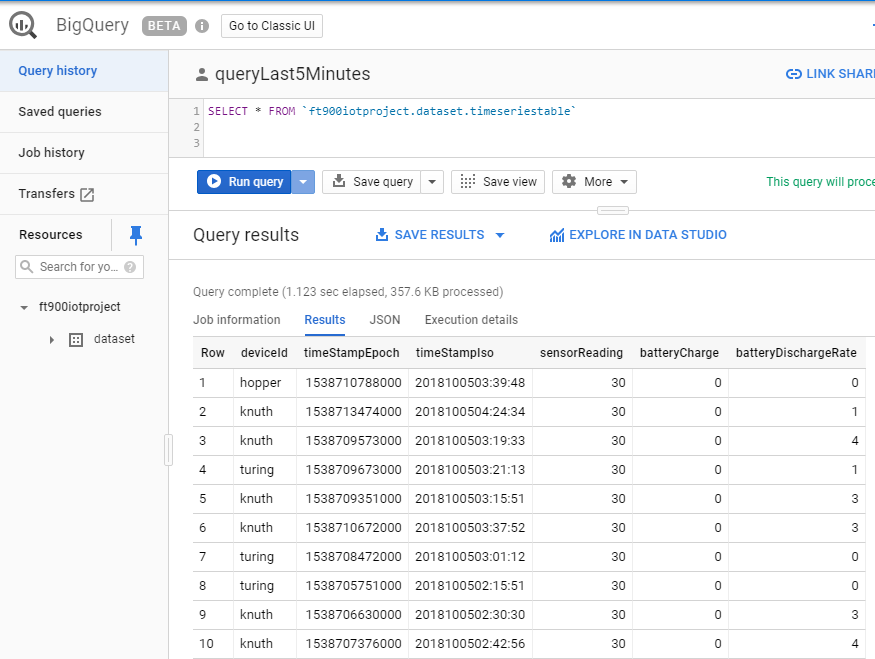
(make sure MQTT\_BROKER\_GCP\_IOT and SAMPLE\_DEVICE\_1 is selected)



2. Go to BigQuery and run the following query

SELECT \* FROM `ft900iotproject.dataset.timeseriestable`

Verify that the data appears as screenshot below.



**III. FRONTEND CLOUD Setup Guide:**

A. Install the prerequisites

1. Google Cloud SDK

2. NodeJS

B. Create a service account

1. gcloud iam service-accounts create bigqueryclient

2. gcloud projects add-iam-policy-binding ft900iotproject --member "serviceAccount:bigqueryclient@ft900iotproject.iam.gserviceaccount.com" --role "roles/bigquery.admin"

3. gcloud iam service-accounts keys create bigqueryclient.json --iam-account [bigqueryclient@ft900iotproject.iam.gserviceaccount.com](mailto:bigqueryclient@ft900iotproject.iam.gserviceaccount.com)

C. Update and run the Node.JS script "bigqueryclient.js":

1. Update the values for

projectId = "ft900iotproject";

dataSetId = "dataset";

tableId = "timeseriestable";

credentialPrivateKey =

"-----BEGIN PRIVATE KEY-----\nMIIEvgI…RuY\n-----END PRIVATE KEY-----\n";

credentialEmail = "bigqueryclient@ft900iotproject.iam.gserviceaccount.com";

2. Run FT90x for 1-2 minutes

3. Run the script using the command "node bigqueryclient.js"

