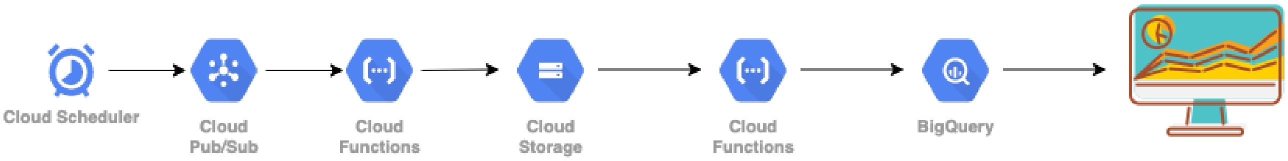
# Use Python and Google Cloud to Schedule API data Import into BigQuery

In this tutorial, I'm going to show you how to set up a serverless data pipeline in GCP that will do the following.

1. Schedule the download of a csv file from the internet
2. Import the data into BigQuery

Note - This tutorial generalizes to any similar workflow where you need to import data into BigQuery.

Here's the workflow. There are a few moving parts, but it's not too complicated.



**The Workflow**

We will need to create a few things to get started. You'll need a Google Cloud account and a project, as well as API access to Cloud Storage, Cloud Functions, Cloud Scheduler, Pub/Sub, and BigQuery.

Here's a summary of what we're going to build.

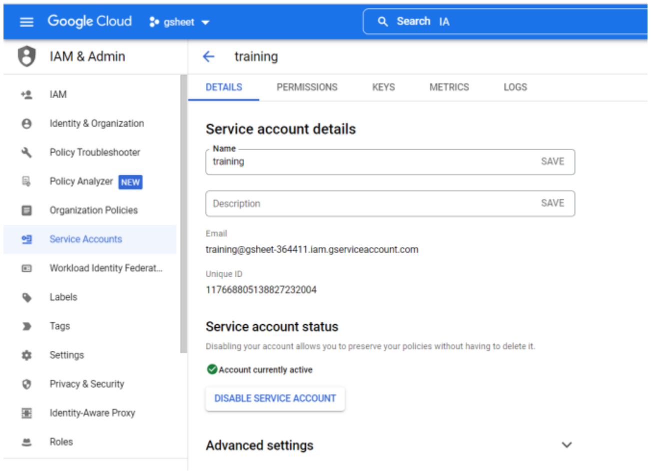
1. One BigQuery dataset and table
2. One Cloud Functions
3. One Cloud Scheduler job

All the code examples are in this GitHub repo.

Even though this example is for the Thailand air quality dataset, it can be used for any situation where you import data into BigQuery.

1. Create new GCP project or used an existing project.

2. create a service account and create json key file that uses for remote access from the local host



Service account details

Name: training

Description

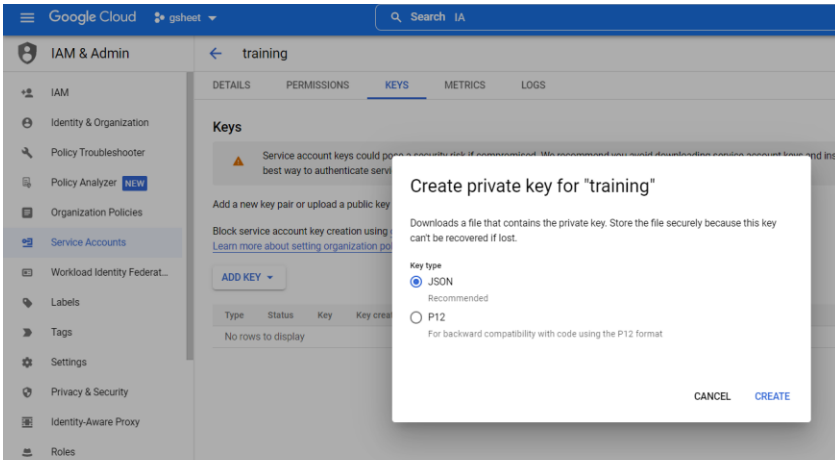
Email
training@gsheet-364411.iam.gserviceaccount.com

Unique ID
117668805138827232004

Service account status

Disabling your account allows you to preserve your policies without having to delete it.

✓ Account currently active



Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and ins
best way to authenticate servi

Add a new key pair or upload a public key

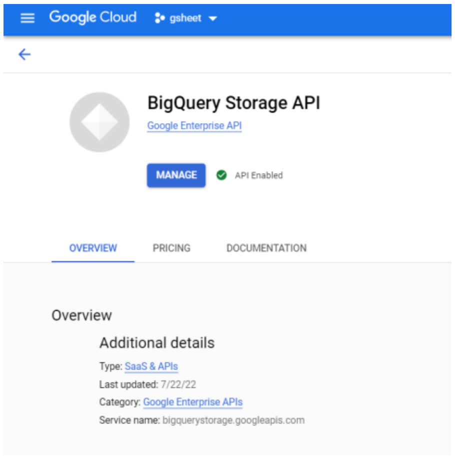
Block service account key creation using
Learn more about setting organization po

Create private key for "training"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type
● JSON
Recommended
○ P12
For backward compatibility with code using the P12 format

## 3. Enable BigQuery API



BigQuery Storage API
Google Enterprise API

✓ API Enabled

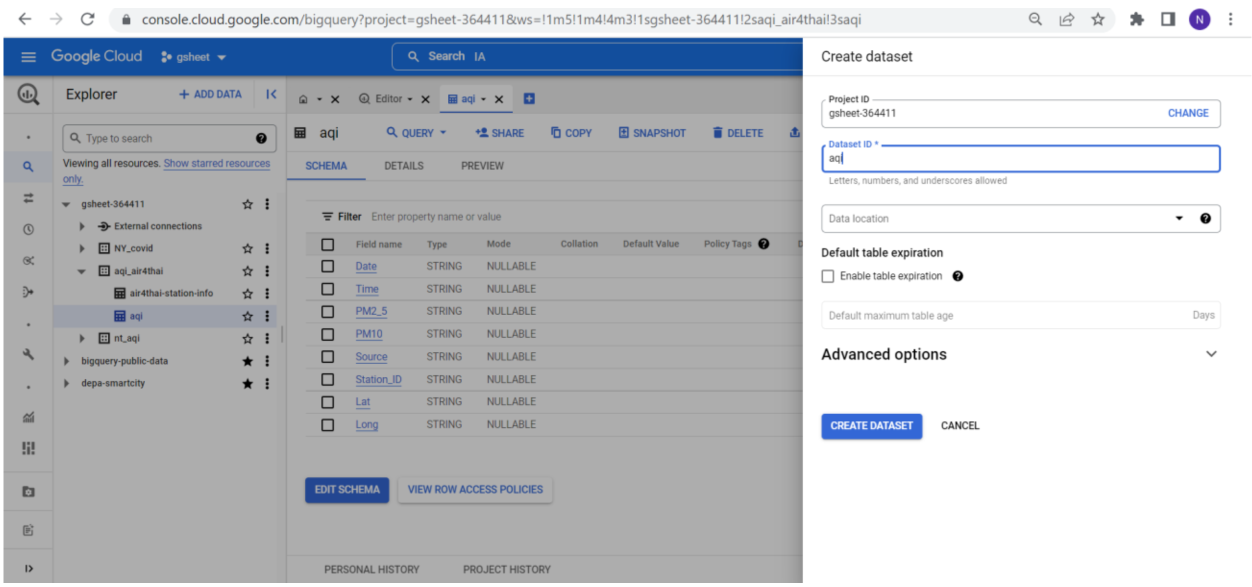
Overview

Additional details

Type: SaaS & APIs
Last updated: 7/22/22
Category: Google Enterprise APIs
Service name: bigquerystorage.googleapis.com

## 4. Create A BigQuery Dataset and Table

Creating a BigQuery dataset and table is very simple. Just remember that you first create a dataset, then create a table.



| □ | Field name | Type | Mode | Collation | Default Value | Policy Tags |
|---|---|---|---|---|---|---|
| □ | Date | STRING | NULLABLE | | | |
| □ | Time | STRING | NULLABLE | | | |
| □ | PM2_5 | STRING | NULLABLE | | | |
| □ | PM10 | STRING | NULLABLE | | | |
| □ | Source | STRING | NULLABLE | | | |
| □ | Station_ID | STRING | NULLABLE | | | |
| □ | Lat | STRING | NULLABLE | | | |
| □ | Long | STRING | NULLABLE | | | |

Create dataset

Project ID: gsheet-364411 CHANGE

Dataset ID *: aqi

Letters, numbers, and underscores allowed

Data location

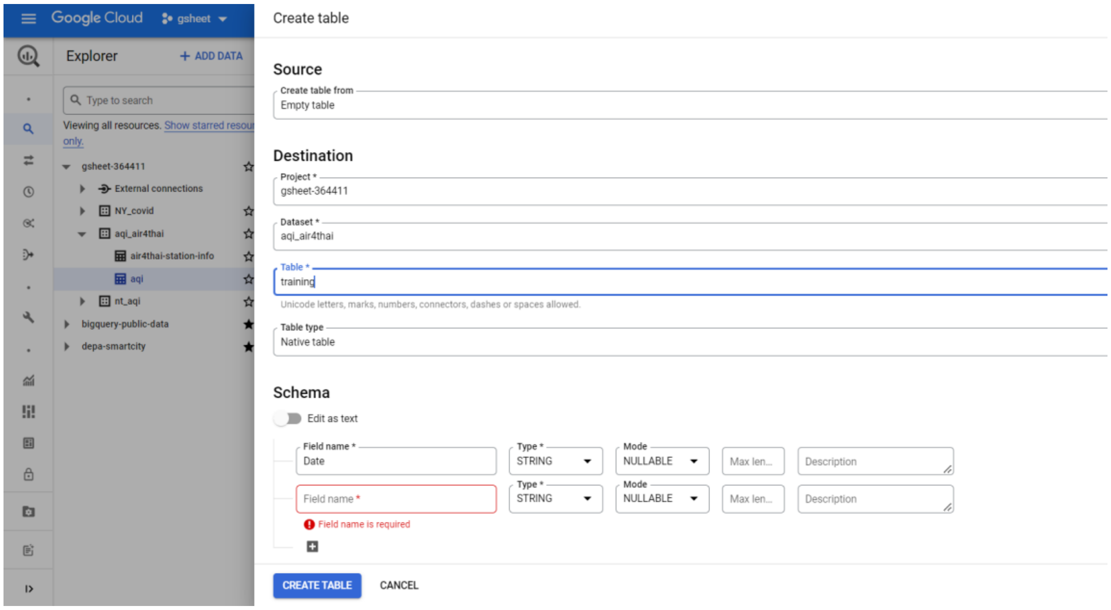
Default table expiration

□ Enable table expiration

Default maximum table age Days

Advanced options

CREATE DATASET CANCEL



Create table

Source

Create table from: Empty table

Destination

Project *: gsheet-364411

Dataset *: aqi_air4thai

Table *: training

Unicode letters, marks, numbers, connectors, dashes or spaces allowed.

Table type: Native table

Schema

Edit as text

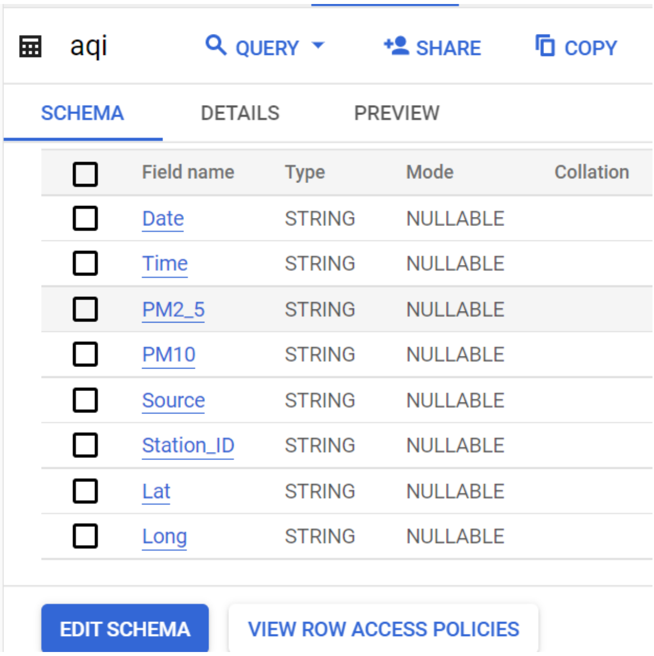
| Field name * | Type * | Mode | Max len... | Description |
|---|---|---|---|---|
| Date | STRING | NULLABLE | | |
| | STRING | NULLABLE | | |

Field name is required

CREATE TABLE CANCEL

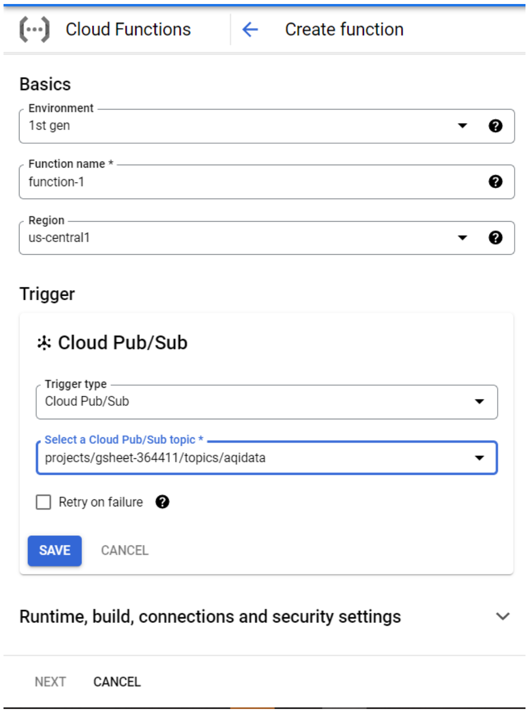
When you create your BigQuery table, you'll need to create a schema with the following fields. These BigQuery fields match the fields in the Thailand air quality json API's header.

Create the BigQuery table, which should have a schema that looks like this.



| □ | Field name | Type | Mode | Collation |
|---|---|---|---|---|
| □ | Date | STRING | NULLABLE | |
| □ | Time | STRING | NULLABLE | |
| □ | PM2_5 | STRING | NULLABLE | |
| □ | PM10 | STRING | NULLABLE | |
| □ | Source | STRING | NULLABLE | |
| □ | Station_ID | STRING | NULLABLE | |
| □ | Lat | STRING | NULLABLE | |
| □ | Long | STRING | NULLABLE | |

5. Create google cloud function, this example used trigger type as cloud pub/sub then fill all variables and click save and next.



Basics

Environment: 1st gen

Function name *: function-1

Region: us-central1

Trigger

Cloud Pub/Sub

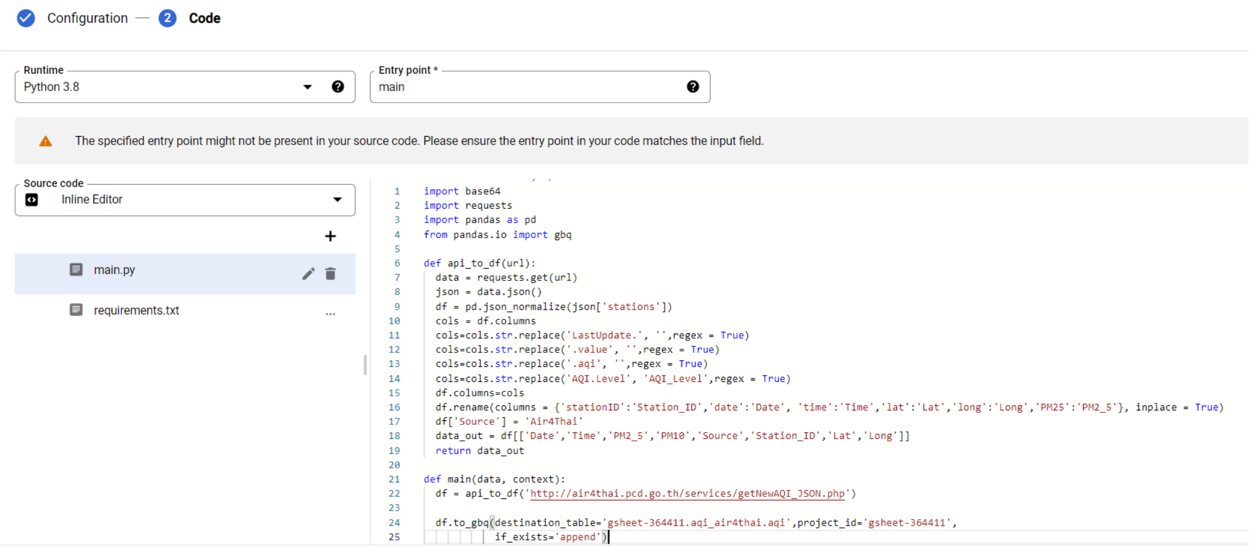
Trigger type: Cloud Pub/Sub

Select a Cloud Pub/Sub topic *: projects/gsheet-364411/topics/aqidata

□ Retry on failure

6. select runtime as Python 3.8, write your python code and necessary python package in requirement.txt

Note *entry point should be the same as your main function



Runtime: Python 3.8

Entry point *: main

The specified entry point might not be present in your source code. Please ensure the entry point in your code matches the input field.

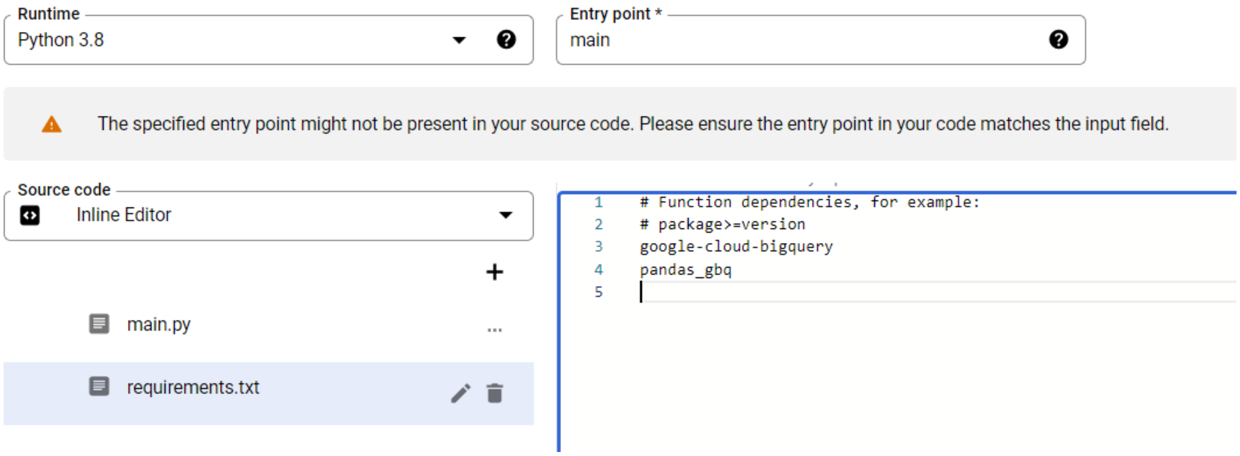
Source code: Inline Editor

```python
import base64
import requests
import pandas as pd
from pandas.io import gbq

def api_to_df(url):
  data = requests.get(url)
  json = data.json()
  df = pd.json_normalize(json['stations'])
  cols = df.columns
  cols=cols.str.replace('LastUpdate.', '',regex = True)
  cols=cols.str.replace('.value', '',regex = True)
  cols=cols.str.replace('.aqi', '',regex = True)
  cols=cols.str.replace('AQI.Level', 'AQI_Level',regex = True)
  df.columns=cols
  df.rename(columns = {'stationID':'Station_ID','date':'Date', 'time':'Time','lat':'Lat','long':'Long','PM25':'PM2_5'}, inplace = True)
  df['Source'] = 'Air4Thai'
  data_out = df[['Date','Time','PM2_5','PM10','Source','Station_ID','Lat','Long']]
  return data_out

def main(data, context):
  df = api_to_df('http://air4thai.pcd.go.th/services/getNewAQI_JSON.php')

  df.to_gbq(destination_table='gsheet-364411.aqi_air4thai.aqi',project_id='gsheet-364411',
            if_exists='append')
```



Runtime: Python 3.8

Entry point *: main

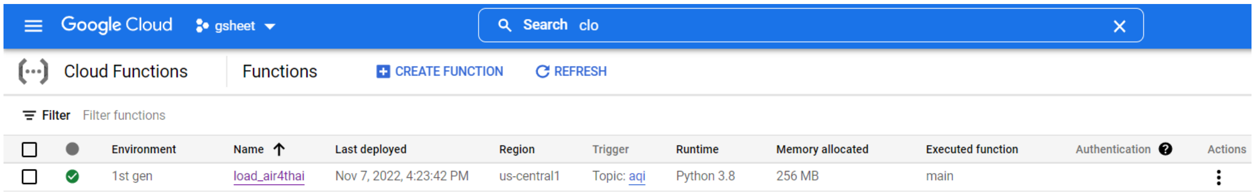
The specified entry point might not be present in your source code. Please ensure the entry point in your code matches the input field.

Source code: Inline Editor

```
# Function dependencies, for example:
# package>=version
google-cloud-bigquery
pandas_gbq
```

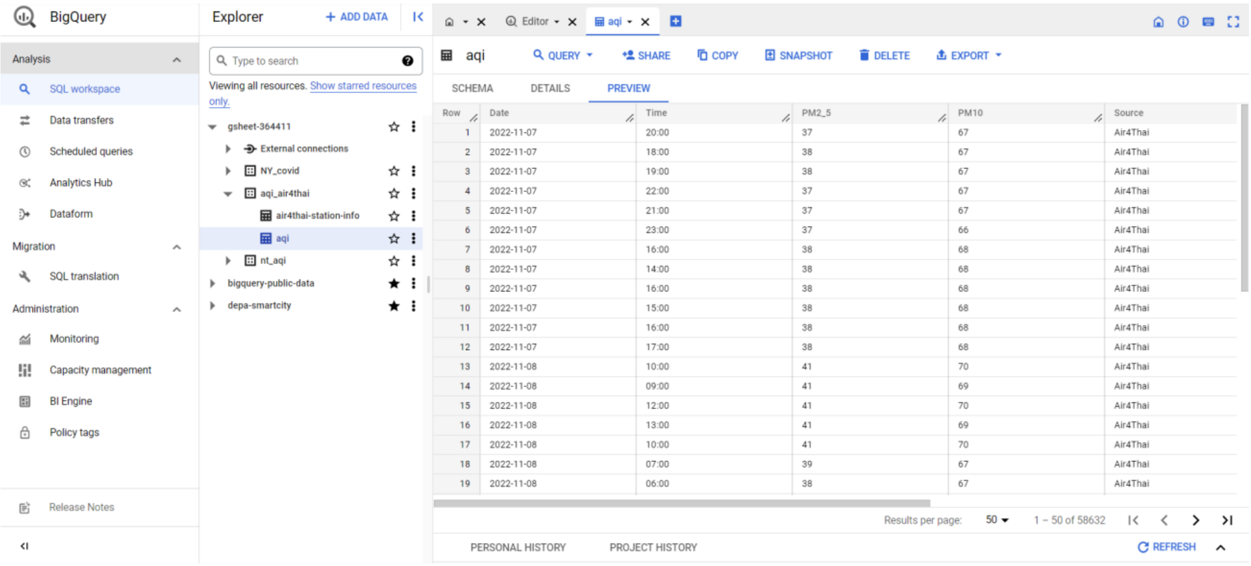
7. deploy your google cloud function, it will taking 2-3 minute.

8. if no error or malfunction, your function will appear green color



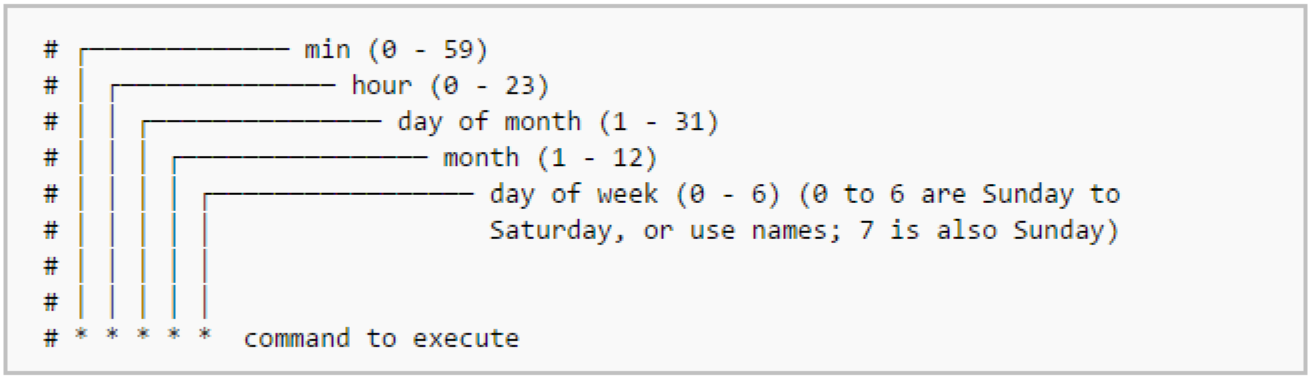
| □ | ● | Environment | Name ↑ | Last deployed | Region | Trigger | Runtime | Memory allocated | Executed function | Authentication | Actions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ✓ | 1st gen | load_air4thai | Nov 7, 2022, 4:23:42 PM | us-central1 | Topic: aqi | Python 3.8 | 256 MB | main | | ⋮ |

9. check your data in your BigQuery table, your function get the data from API and writes it to BigQuery directly.



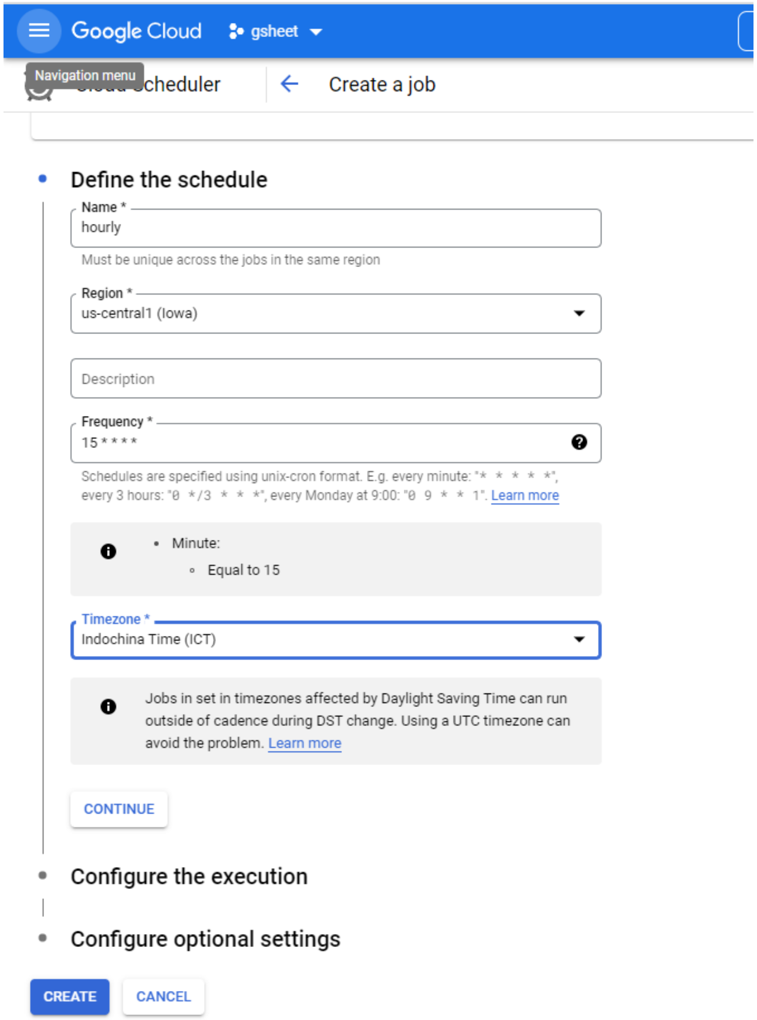
| Row | Date | Time | PM2_5 | PM10 | Source |
|---|---|---|---|---|---|
| 1 | 2022-11-07 | 20:00 | 37 | 67 | Air4Thai |
| 2 | 2022-11-07 | 18:00 | 38 | 67 | Air4Thai |
| 3 | 2022-11-07 | 19:00 | 38 | 67 | Air4Thai |
| 4 | 2022-11-07 | 22:00 | 37 | 67 | Air4Thai |
| 5 | 2022-11-07 | 21:00 | 37 | 67 | Air4Thai |
| 6 | 2022-11-07 | 23:00 | 37 | 66 | Air4Thai |
| 7 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 8 | 2022-11-07 | 14:00 | 38 | 68 | Air4Thai |
| 9 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 10 | 2022-11-07 | 15:00 | 38 | 68 | Air4Thai |
| 11 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 12 | 2022-11-07 | 17:00 | 38 | 68 | Air4Thai |
| 13 | 2022-11-08 | 10:00 | 41 | 70 | Air4Thai |
| 14 | 2022-11-08 | 09:00 | 41 | 69 | Air4Thai |
| 15 | 2022-11-08 | 12:00 | 41 | 70 | Air4Thai |
| 16 | 2022-11-08 | 13:00 | 41 | 69 | Air4Thai |
| 17 | 2022-11-08 | 10:00 | 41 | 70 | Air4Thai |
| 18 | 2022-11-08 | 07:00 | 39 | 67 | Air4Thai |
| 19 | 2022-11-08 | 06:00 | 38 | 67 | Air4Thai |

10. Create A Cloud Scheduler job - Cloud Scheduler is a GCP utility that allows you to schedule tasks. If you've used cron, then you'll have a good idea of how this works. If not, no worries. Cron is very simple to figure out. You basically just need to know that cron goes something like this. You've got 5 options that you can populate however you like. In our example, we'll set a Cloud Scheduler cron frequency for every ten minutes (this is obviously overkilled for a data set that updates once a day, but since we're in example-land, the point stands).



```
# ┌───────────── min (0 - 59)
# │ ┌────────────── hour (0 - 23)
# │ │ ┌─────────────── day of month (1 - 31)
# │ │ │ ┌──────────────── month (1 - 12)
# │ │ │ │ ┌───────────────── day of week (0 - 6) (0 to 6 are Sunday to
# │ │ │ │ │                  Saturday, or use names; 7 is also Sunday)
# │ │ │ │ │
# │ │ │ │ │
# * * * * *  command to execute
```

Cloud Scheduler needs to trigger something to execute. For this tutorial, we'll target the Pub/Sub topic that we created in the Cloud Function we deployed above. Once you get the frequency populated, hit create (or update). Here's what your Cloud Scheduler should look like.



Define the schedule

Name *
hourly

Must be unique across the jobs in the same region

Region *
us-central1 (Iowa)

Description

Frequency *
15 * * * *

Schedules are specified using unix-cron format. E.g. every minute: "* * * * *", every 3 hours: "0 */3 * * *", every Monday at 9:00: "0 9 * * 1". Learn more

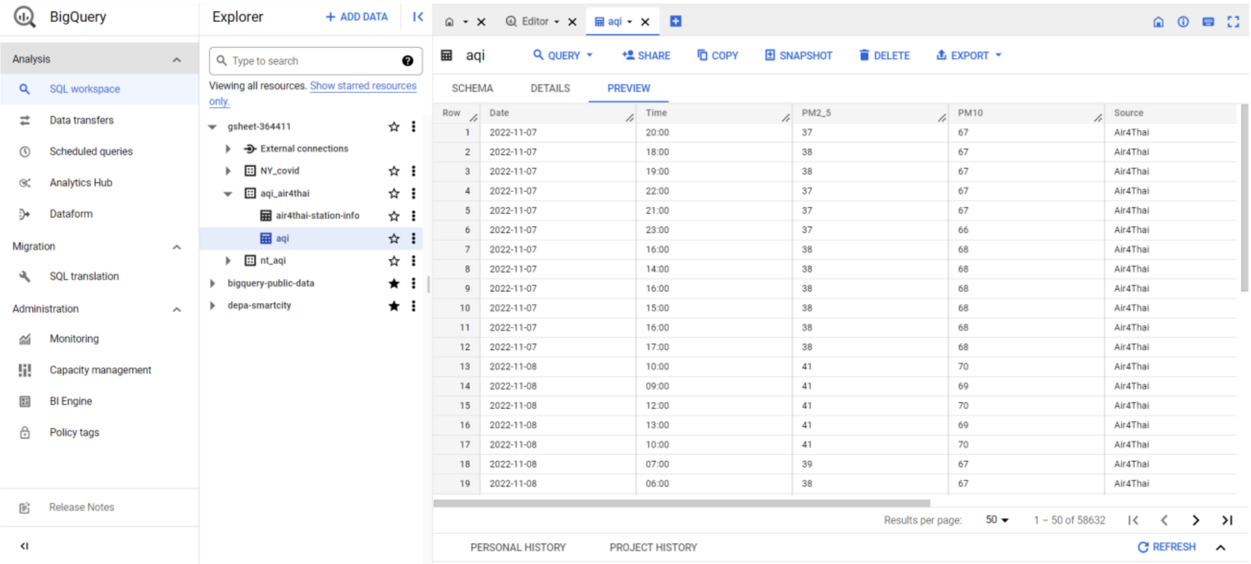
- Minute:
  - Equal to 15

Timezone *
Indochina Time (ICT)

Jobs in set in timezones affected by Daylight Saving Time can run outside of cadence during DST change. Using a UTC timezone can avoid the problem. Learn more

Then in the Cloud Scheduler console, hit “Run” to test the workflow. If everything works, you should see your BigQuery table populated with data.

Let’s See the Data, Now that we’ve got the workflow pieced together and working, let’s have a look at the air quality data in BigQuery. Here’s what you should see in BigQuery that hourly update.



| Row | Date | Time | PM2_5 | PM10 | Source |
|---|---|---|---|---|---|
| 1 | 2022-11-07 | 20:00 | 37 | 67 | Air4Thai |
| 2 | 2022-11-07 | 18:00 | 38 | 67 | Air4Thai |
| 3 | 2022-11-07 | 19:00 | 38 | 67 | Air4Thai |
| 4 | 2022-11-07 | 22:00 | 37 | 67 | Air4Thai |
| 5 | 2022-11-07 | 21:00 | 37 | 67 | Air4Thai |
| 6 | 2022-11-07 | 23:00 | 37 | 66 | Air4Thai |
| 7 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 8 | 2022-11-07 | 14:00 | 38 | 68 | Air4Thai |
| 9 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 10 | 2022-11-07 | 15:00 | 38 | 68 | Air4Thai |
| 11 | 2022-11-07 | 16:00 | 38 | 68 | Air4Thai |
| 12 | 2022-11-07 | 17:00 | 38 | 68 | Air4Thai |
| 13 | 2022-11-08 | 10:00 | 41 | 70 | Air4Thai |
| 14 | 2022-11-08 | 09:00 | 41 | 69 | Air4Thai |
| 15 | 2022-11-08 | 12:00 | 41 | 70 | Air4Thai |
| 16 | 2022-11-08 | 13:00 | 41 | 69 | Air4Thai |
| 17 | 2022-11-08 | 10:00 | 41 | 70 | Air4Thai |
| 18 | 2022-11-08 | 07:00 | 39 | 67 | Air4Thai |
| 19 | 2022-11-08 | 06:00 | 38 | 67 | Air4Thai |

Results per page: 50 1 – 50 of 58632