

# Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery

<https://github.com/celiamuriel-google/aemet-to-bigquery.git>

|   |           |
|---|-----------|
| <b>Introduction</b>   | <b>1</b>  |
| <b>Get API Key</b>  | <b>2</b>  |
| <b>Where to run the software to retrieve data from AEMET</b>                            | <b>4</b>  |
| Prepare the environment to work   | 5         |
| Create BigQuery objects   | 7         |
| DDLs  | 7         |
| Description of the tables   | 8         |
| todasestaciones   | 8         |
| observacion-convencional  | 10        |
| <b>Shell Scripts to retrieve the meteorological information and load it in BigQuery</b> | <b>17</b> |
| <b>Store data files in a Cloud Storage bucket</b>                                       | <b>17</b> |
| <b>References</b>   | <b>21</b> |
| AEMET   | 21        |
| GCP Services  | 21        |
| Using GCP SDK   | 22        |
| Geographic Data   | 22        |

## Introduction

This document provides the scripting necessary to download meteorological information from the Spanish Meteorological Agency ([AEMET](#)) from the [AEMET OpenData](#) project and upload it to BigQuery to use it as part of the data analytics.

AEMET provides data in plain JSON format. It contains geographic information, both in DMS (Degree, Minute, Second) and decimal latitude and longitude.

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



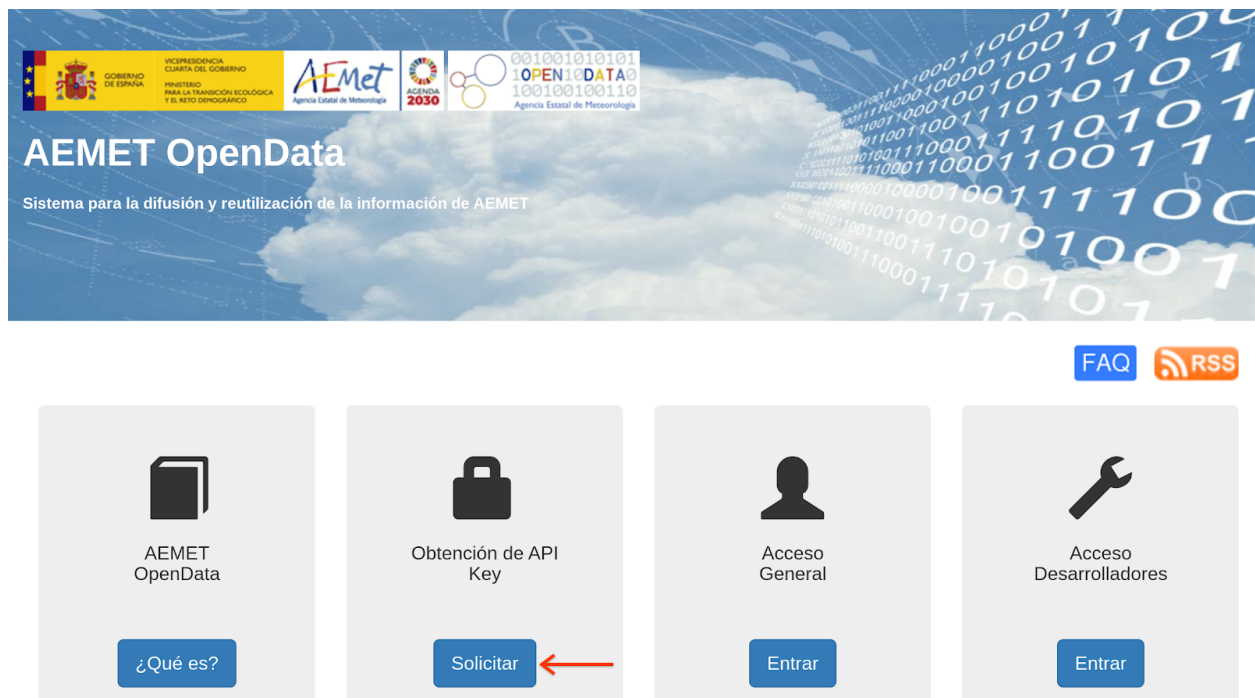
Data is uploaded to staging tables (stg). Then it needs to be transformed to be used for reporting. The master tables with the data necessary for reporting has a “t\_” prefix.

This document also explains in detail the setup to run the scripting.

It was done on May 4th, 2020, with the Generally Available features on the different services used for this exercise.

## Get API Key

Navigate to [AEMET OpenData](#) and click on “Solicitar” in the “Obtención de API Key” box.

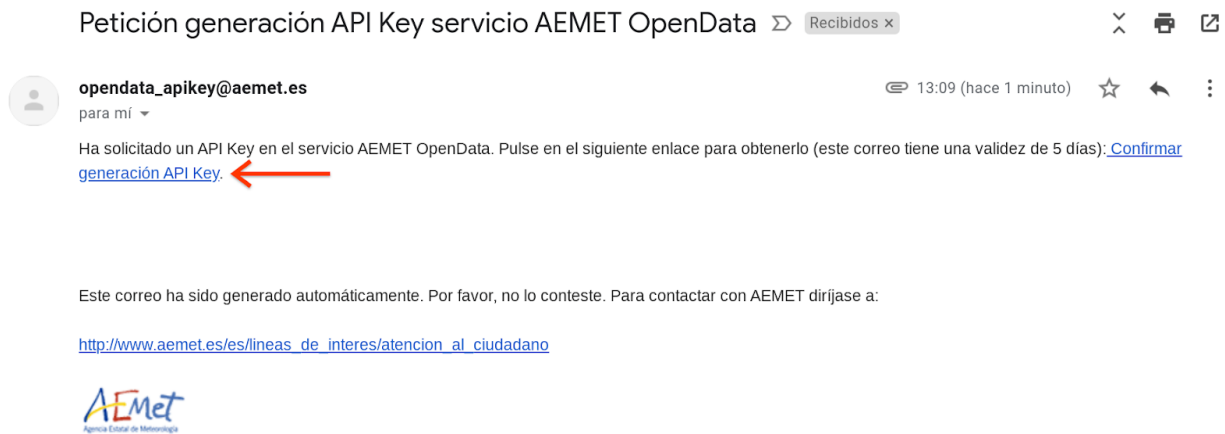


Insert an email address, check the “I’m not a robot box”, and click in “Enviar”.

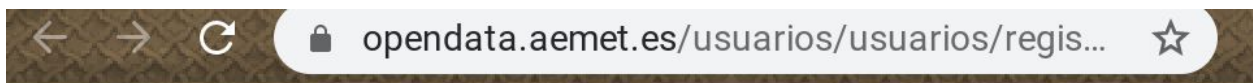
Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



You should receive an email. Click on “Confirmar generación de API Key”.



You get a message confirming that you generated the API Key:



Su API Key se ha generado correctamente, se le enviará en un correo.

You receive a second email with the API Key:

## Alta en el servicio AEMET Open



**opendata\_apikey@aemet.es**

para mí ▼



inglés ▼



español ▼

[Traducir mensaje](#)

Alta en el servicio AEMET OpenData. Su API Key es:

evJhbGciOiJIUzI1NiJ9.evJzdWliOiJibXVvaWVsbUBnb

## Where to run the software to retrieve data from AEMET

AEMET provides client [application examples](#) and [Codegen to automatically generate the API in several programming languages](#) to retrieve the meteorological data. We are going to create several Shell scripts to download the data from AEMET, format it, upload to BigQuery and run the ETL to transform the data for analysis and store it in the master tables.

The Shell scripts can run on:

- [Cloud Shell](#),
- A [compute engine](#) with the appropriate size if Cloud Shell is too small or for security reasons,
- [Cloud App Engine](#),
- [Cloud Function](#), or
- [Cloud Run](#).

See [here](#) how to choose between the different serverless services.

You'd choose one or another depending on your actual requirements.

For the demonstration purposes of this document, we will run our scripting from Cloud Shell.

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>

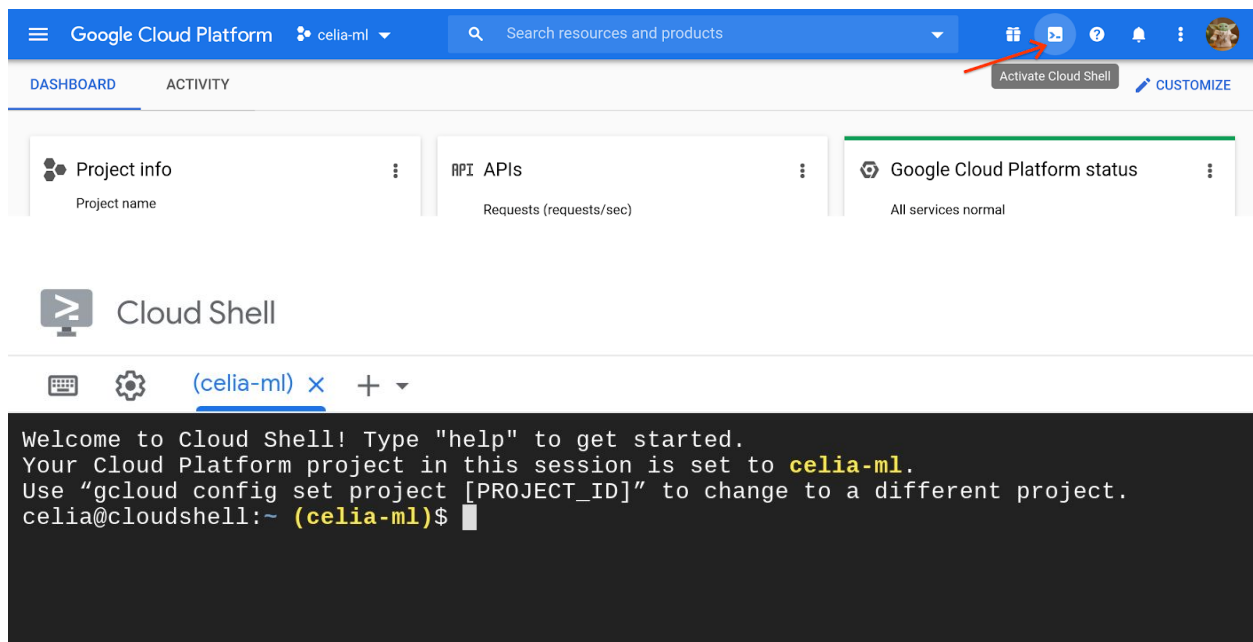


## Prepare the environment to work

Cloud Shell has a Debian distribution and you can find [here](#) how to use it.

Check if the [Cloud SDK](#) is installed. If not, [install](#) it and [set it up](#).

Launch Cloud Shell in your project (or a compute engine, a Cloud App Engine, a Cloud Function or Cloud Run).



Update the package lists for upgrades for packages that need upgrading, as well as new packages that have just come to the repositories.

```
sudo apt-get update
```

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamurael-google/aemet-to-bigquery.git>



Cloud Shell



(celia-ml) x + ▾

```
Your Cloud Platform project in this session is set to celia-ml.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
celia@cloudshell:~ (celia-ml)$ sudo apt-get update
*****
You are running apt-get inside of Cloud Shell. Note that your Cloud Shell
machine is ephemeral and no system-wide change will persist beyond session end.

To suppress this warning, create an empty ~/.cloudshell/no-apt-get-warning file.
The command will automatically proceed in 5 seconds or on any key.

Visit https://cloud.google.com/shell/help for more information.
*****
Ign:1 http://deb.debian.org/debian stretch InRelease
Get:2 http://security.debian.org/debian-security stretch/updates InRelease [94.3 kB]
Get:3 http://deb.debian.org/debian stretch-updates InRelease [91.0 kB]
Ign:4 http://ftp.debian.org/debian jessie InRelease
Ign:5 http://ftp.debian.org/debian stretch InRelease
Get:6 http://ftp.debian.org/debian stretch-backports InRelease [91.8 kB]
Hit:7 http://packages.cloud.google.com/apt gcsfuse-stretch InRelease
Hit:8 http://repo.mysql.com/apt/debian stretch InRelease
Hit:9 http://deb.debian.org/debian stretch Release
Get:10 http://ftp.debian.org/debian stretch-updates InRelease [91.0 kB]
Hit:11 http://ftp.debian.org/debian jessie Release
Hit:13 http://ftp.debian.org/debian stretch Release
Hit:14 http://storage.googleapis.com/bazel-apt stable InRelease
Hit:15 https://download.docker.com/linux/debian stretch InRelease
Hit:16 https://packages.microsoft.com/repos/microsoft-debian-stretch-prod stretch InRelease
Get:17 https://packages.sury.org/php stretch InRelease [6,760 B]
Hit:12 https://apt.llvm.org/stretch llvm-toolchain-stretch-9 InRelease
Get:19 http://ftp.debian.org/debian stretch-backports/main Sources.diff/Index [27.8 kB]
Get:20 http://ftp.debian.org/debian stretch-backports/main amd64 Packages.diff/Index [27.8 kB]
Get:21 http://ftp.debian.org/debian stretch-backports/main Sources 2020-04-18-1402.01.pdiff [235 B]
Get:21 http://ftp.debian.org/debian stretch-backports/main Sources 2020-04-18-1402.01.pdiff [235 B]
Get:22 http://ftp.debian.org/debian stretch-backports/main amd64 Packages 2020-04-18-1402.01.pdiff [219 B]
Get:22 http://ftp.debian.org/debian stretch-backports/main amd64 Packages 2020-04-18-1402.01.pdiff [219 B]
Get:25 https://packages.sury.org/php stretch/main amd64 Packages [136 kB]
Fetched 567 kB in 2s (239 kB/s)
Reading package lists... Done
celia@cloudshell:~ (celia-ml)$
```

[jq](#) is like sed to JSON files. Type `jq --help` to know if it is already installed.

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



```
celia@cloudshell:~/aemet (celia-m1)$ jq --help
jq - commandline JSON processor [version 1.5.1-a5b5cbe]
Usage: jq [options] <jq filter> [file...]

jq is a tool for processing JSON inputs, applying the
given filter to its JSON text inputs and producing the
filter's results as JSON on standard output.
The simplest filter is ., which is the identity filter,
copying jq's input to its output unmodified (except for
formatting).
For more advanced filters see the jq(1) manpage ("man jq")
and/or https://stedolan.github.io/jq

Some of the options include:
-c          compact instead of pretty-printed output;
-n          use `null` as the single input value;
-e          set the exit status code based on the output;
-s          read (slurp) all inputs into an array; apply filter to it;
-r          output raw strings, not JSON texts;
-R          read raw strings, not JSON texts;
-C          colorize JSON;
-M          monochrome (don't colorize JSON);
-S          sort keys of objects on output;
--tab       use tabs for indentation;
--arg a v   set variable $a to value <v>;
--argjson a v set variable $a to JSON value <v>;
--slurpfile a f set variable $a to an array of JSON texts read from <f>;
See the manpage for more options.
celia@cloudshell:~/aemet (celia-m1)$
```

If jq is not installed, do it.

```
sudo apt-get install jq
```

Clone this GitHub repository, where you have the BigQuery objects you need to create, and the scripts to download, process and upload the meteorological information in Spain.

```
git clone https://github.com/celiamuriel-google/aemet-to-bigquery.git
```

## Create BigQuery objects

### DDLs<sup>1</sup>

Before loading the meteorological information in BigQuery for the first time, we need to create the dataset and objects where we are going to store the data. The next commands are meant to create the database objects using the [bq command-line tool](#) from the [Cloud Shell](#). You can also use the [BigQuery console](#) and run the DDLs from there.

---

<sup>1</sup> DDL stands for Data Definition Language.



Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



If you want BigQuery to automatically delete data older than a certain number of days, add the table `--default_table_expiration` flag to the `bq mk dataset`, or the `--expiration` flag to the `bq mk table`.

Remember to replace `[PROJECT_ID]` in the DDL scripts by your actual GCP Project ID.

## Description of the tables

AEMET provides metadata files which describe the data we download from their website.

We load the data in staging tables (`stg_[name of the downloaded file]`). Then we add the load date (`fecha_carga`), calculate a GEOGRAPHY field with the longitude and latitude provided and load these new fields along with the ones in the staging tables in the master ones (`t_[name of the downloaded file]`).

In some cases we rename the fields in the master tables to a more intuitive name. For example, `hr` as `humedad_rel`.

This section includes the metadata as we downloaded on May 4th, 2020. When necessary, we explain in detail about some fields.

### todasestaciones

```
{
  "unidad_generadora": "Servicio del Banco de Datos Nacional de
Climatología",
  "periodicidad": "1 vez al día",
  "descripcion": "Inventario de estaciones para el apartado Valores
Climatología",
  "formato": "application/json",
  "copyright": "AEMET. Autorizado el uso de la información y su
reproducción citando a AEMET como autora de la misma.",
  "notaLegal": "http://www.aemet.es/es/nota_legal",
  "campos": [
    { "id": "latitud",
      "descripcion": "latitud de la estación",
      "tipo_datos": "string",
      "requerido": true
    },
    { "id": "provincia",
      "descripcion": "provincia donde reside la estación",
```



Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



```
"tipo_datos": "string",
  "requerido": true
},
{"id": "indicativo",
"descripcion": "indicativo climatológico de la estación",
"tipo_datos": "string",
  "requerido": true
},
{"id": "altitud",
"descripcion": "altitud de la estación ",
"tipo_datos": "string",
  "requerido": true
},
{"id": "nombre",
"descripcion": "ubicación de la estación",
"tipo_datos": "string",
  "requerido": true
},
{"id": "indsinop",
"descripcion": "Indicativo sinóptico",
"tipo_datos": "string",
  "requerido": true
},
{"id": "longitud",
  "descripcion": "longitud de la estación",
  "tipo_datos": "string",
  "requerido": true
}
]
}
```

The weather station coordinates are expressed as follows:

1. Latitude (latitud in Spanish): degrees (2 digits) + minutes (2 digits) + seconds (2 digits) + [North (Norte, N) or South (Sur, S) of the Equator]. There are neither separators nor space among the different elements.
2. Longitude (longitud): degrees (2 digits) + minutes (2 digits) + seconds (2 digits) + [East (Este, E) or West (Oeste, W) of Greenwich]. There are neither separators nor space among the different elements.
3. Altitude (altitud): metres above mean sea level (MAMSL) - "metros sobre el nivel del mar (msnm)" in Spanish.

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



We load these fields in BigQuery as `latitud_DMS`, `longitud_DMS` (Degree, Minute, Second) and `altitud`. Then we transform them in their digital format (`latitud_dec`, `longitud_dec` and `altitud` fields respectively).

## observacion-convencional

```
{
  "unidad_generadora": "Servicio de Observaci\u00f3n",
  "periodicidad": "continuamente",
  "formato": "application/json",
  "copyright": "\u25c0 AEMET. Autorizado el uso de la informaci\u00f3n y su
reproducci\u00f3n citando a AEMET como autora de la misma.",
  "notaLegal": "http://www.aemet.es/es/nota_legal",
  "campos": [{
    "id": "idema",
    "descripcion": "Indicativo climatol\u00f3gico de la estaci\u00f3n
meteorol\u00f3gica autom\u00e1tica",
    "tipo_datos": "string",
    "requerido": true
  },
  {
    "id": "lon",
    "descripcion": "Longitud de la estaci\u00f3n meteorol\u00f3gica
(grados)",
    "tipo_datos": "float",
    "requerido": true
  },
  {
    "id": "lat",
    "descripcion": "Latitud de la estaci\u00f3n meteorol\u00f3gica
(grados)",
    "tipo_datos": "float",
    "requerido": true
  },
  {
    "id": "alt",
    "descripcion": "Altitud de la estaci\u00f3n en metros",
    "tipo_datos": "float",
    "requerido": true
  },
  {
    "id": "ubi",
    "descripcion": "Ubicaci\u00f3n de la estaci\u00f3n. Nombre de la
estaci\u00f3n",
    "tipo_datos": "string",
    "requerido": true
  }
  ]
}
```

```
    },
    {
      "id": "fint",
      "descripcion": "Fecha hora final del periodo de observaci\u00f3n, se trata de datos del periodo de la hora anterior a la indicada por este campo (hora UTC)",
      "tipo_datos": "string (AAAA-MM-DDTHH:MM:SS)",
      "requerido": false
    },
    {
      "id": "prec",
      "descripcion": "Precipitaci\u00f3n acumulada, medida por el pluvi\u00f3metro, durante los 60 minutos anteriores a la hora indicada por el periodo de observaci\u00f3n 'fint' (mm, equivalente a l/m2)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "pacutp",
      "descripcion": "Precipitaci\u00f3n acumulada, medida por el disdr\u00f3metro, durante los 60 minutos anteriores a la hora indicada por el periodo de observaci\u00f3n 'fint' (mm, equivalente a l/m2)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "pliqtp",
      "descripcion": "Precipitaci\u00f3n l\u00edquida acumulada durante los 60 minutos anteriores a la hora indicada por el periodo de observaci\u00f3n 'fint' (mm, equivalente a l/m2)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "psolt",
      "descripcion": "Precipitaci\u00f3n s\u00f3lida acumulada durante los 60 minutos anteriores a la hora indicada por el periodo de observaci\u00f3n 'fint' (mm, equivalente a l/m2)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "vmax",
      "descripcion": "Velocidad m\u00e1xima del viento, valor m\u00e1ximo del viento mantenido 3 segundos y registrado en los 60 minutos anteriores a la hora indicada por el periodo de observaci\u00f3n 'fint' (m/s)",
```

```
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "vv",
        "descripcion": "Velocidad media del viento, media escalar de
las muestras adquiridas cada 0,25 1 segundo en el periodo de 10 minutos
anterior al indicado por 'fint' (m/s)",
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "vmaxu",
        "descripcion": "Velocidad m xima del viento (sensor
ultras nico), valor m ximo del viento mantenido 3 segundos y registrado
en los 60 minutos anteriores a la hora indicada por el periodo de
observaci n 'fint' (m/s)",
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "vvu",
        "descripcion": "Velocidad media del viento (sensor
ultras nico), media escalar en el periodo de 10 minutos anterior al
indicado por 'fint' de las muestras adquiridas cada 0,25 1 segundo
(m/s)",
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "dv",
        "descripcion": "Direcci n media del viento, en el periodo de
10 minutos anteriores a la fecha indicada por 'fint' (grados)",
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "dvu",
        "descripcion": "Direcci n media del viento (sensor
ultras nico), en el periodo de 10 minutos anteriores a la fecha indicada
por 'fint' (grados)",
        "tipo_datos": "float",
        "requerido": false
    },
    {
        "id": "dmax",
```

```
    "descripcion": "Dirección del viento máximo registrado en los  
60 minutos anteriores a la hora indicada por 'fint' (grados)",  
    "tipo_datos": "float",  
    "requerido": false  
  },  
  {  
    "id": "dmaxu",  
    "descripcion": "Dirección del viento máximo registrado en los  
60 minutos anteriores a la hora indicada por 'fint' por el sensor  
ultrasónico (grados)",  
    "tipo_datos": "float",  
    "requerido": false  
  },  
  {  
    "id": "stdvv",  
    "descripcion": "Desviación estándar de las muestras  
adquiridas de velocidad del viento durante los 10 minutos anteriores a la  
fecha dada por 'fint' (m/s)",  
    "tipo_datos": "float",  
    "requerido": false  
  },  
  {  
    "id": "stddv",  
    "descripcion": "Desviación estándar de las muestras  
adquiridas de la dirección del viento durante los 10 minutos anteriores a  
la fecha dada por 'fint' (grados)",  
    "tipo_datos": "float",  
    "requerido": false  
  },  
  {  
    "id": "stdvvu",  
    "descripcion": "Desviación estándar de las muestras  
adquiridas de velocidad del viento durante los 10 minutos anteriores a la  
fecha dada por 'fint' obtenido del sensor ultrasónico de viento instalado  
junto al convencional (m/s)",  
    "tipo_datos": "float",  
    "requerido": false  
  },  
  {  
    "id": "stddvu",  
    "descripcion": "Desviación estándar de las muestras  
adquiridas de la dirección del viento durante los 10 minutos anteriores a  
la fecha dada por 'fint' obtenido del sensor ultrasónico de viento  
instalado junto al convencional (grados)",  
    "tipo_datos": "float",  
    "requerido": false  
  }
```

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



```
    },
    {
      "id": "hr",
      "descripcion": "Humedad relativa instantánea del aire correspondiente a la fecha dada por 'fint' (%)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "inso",
      "descripcion": "Duración de la insolación durante los 60 minutos anteriores a la hora indicada por el período de observación 'fint' (horas)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "pres",
      "descripcion": "Presión instantánea al nivel en el que se encuentra instalado el barómetro y correspondiente a la fecha dada por 'fint' (hPa)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "pres_nmar",
      "descripcion": "Valor de la presión reducido al nivel del mar para aquellas estaciones cuya altitud es igual o menor a 750 metros y correspondiente a la fecha indicada por 'fint' (hPa)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "ts",
      "descripcion": "Temperatura suelo, temperatura instantánea junto al suelo y correspondiente a los 10 minutos anteriores a la fecha dada por 'fint' (grados Celsius)",
      "tipo_datos": "float",
      "requerido": false
    },
    {
      "id": "tss20cm",
      "descripcion": "Temperatura subsuelo 20 cm, temperatura del subsuelo a una profundidad de 20 cm y correspondiente a los 10 minutos anteriores a la fecha dada por 'fint' (grados Celsius)",
      "tipo_datos": "float",
```

```
    "requerido": false
  },
  {
    "id": "tss5cm",
    "descripcion": "Temperatura subsuelo 5 cm, temperatura del
subsuelo a una profundidad de 5 cm y correspondiente a los 10 minutos
anteriores a la fecha dada por 'fint' (grados Celsius)",
    "tipo_datos": "float",
    "requerido": false
  },
  {
    "id": "ta",
    "descripcion": "Temperatura instantánea del aire
correspondiente a la fecha dada por 'fint' (grados Celsius)",
    "tipo_datos": "float",
    "requerido": false
  },
  {
    "id": "tpr",
    "descripcion": "Temperatura del punto de roc o calculado
correspondiente a la fecha 'fint' (grados Celsius)",
    "tipo_datos": "float",
    "requerido": false
  },
  {
    "id": "tamin",
    "descripcion": "Temperatura m nima del aire, valor m nimo de
los 60 valores instant neos de 'ta' medidos en el per odo de 60 minutos
anteriores a la hora indicada por el per odo de observaci n 'fint'
(grados Celsius)",
    "tipo_datos": "float",
    "requerido": false
  },
  {
    "id": "tamax",
    "descripcion": "Temperatura m xima del aire, valor m ximo de
los 60 valores instant neos de 'ta' medidos en el per odo de 60 minutos
anteriores a la hora indicada por el per odo de observaci n 'fint'
(grados Celsius)",
    "tipo_datos": "float",
    "requerido": false
  },
  {
    "id": "vis",
```



```
      "descripcion": "Visibilidad, promedio de la medida de la  
visibilidad correspondiente a los 10 minutos anteriores a la fecha dada  
por 'fint' (Km)",  
      "tipo_datos": "float",  
      "requerido": false  
    },  
    {  
      "id": "geo700",  
      "descripcion": "Altura del nivel de la superficie de  
referencia barométrica de 700 hPa calculado para las estaciones con  
altitud mayor de 2300 metros y correspondiente a la fecha indicada por  
'fint' (m geopotenciales)",  
      "tipo_datos": "float",  
      "requerido": false  
    },  
    {  
      "id": "geo850",  
      "descripcion": "Altura del nivel de la superficie de  
referencia barométrica de 850 hPa calculado para las estaciones con  
altitud mayor de 1000 metros y menor o igual a 2300 metros y  
correspondiente a la fecha indicada por 'fint' (m geopotenciales)",  
      "tipo_datos": "float",  
      "requerido": false  
    },  
    {  
      "id": "geo925",  
      "descripcion": "Altura del nivel de la superficie barométrica  
de 925 hPa calculado para las estaciones con altitud mayor de 750 metros y  
y menor o igual a 1000 metros correspondiente a la fecha indicada por  
'fint' (m geopotenciales)",  
      "tipo_datos": "float",  
      "requerido": false  
    },  
    {  
      "id": "rviento",  
      "descripcion": "Recorrido del viento durante los 60 minutos  
anteriores a la fecha indicada por 'fint' (Hm)",  
      "tipo_datos": "float",  
      "requerido": false  
    },  
    {  
      "id": "nieve",  
      "descripcion": "Espesor de la capa de nieve medid en los 10  
minutos anteriores a la a la fecha indicada por 'fint' (cm)",  
      "tipo_datos": "float",  
      "requerido": false  
    }
```

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



```
}  
]  
}
```

## Shell Scripts to retrieve the meteorological information and load it in BigQuery

When you are ready to get the meteorological information, you can execute the `aemet.sh` script.

```
./aemet.sh
```

Comment out those lines with data you are not interested in using for your analysis.

You can schedule the execution of the `aemet.sh` script daily.

The `aemet.sh` script executes the Shell scripts which download every data file from the AEMET OpenData website and loads them in BigQuery. The API Key is hardcoded in the script. This is NOT a good practice for Production. Secure this Key as per your organization's security standard and policy.

## Store data files in a Cloud Storage bucket

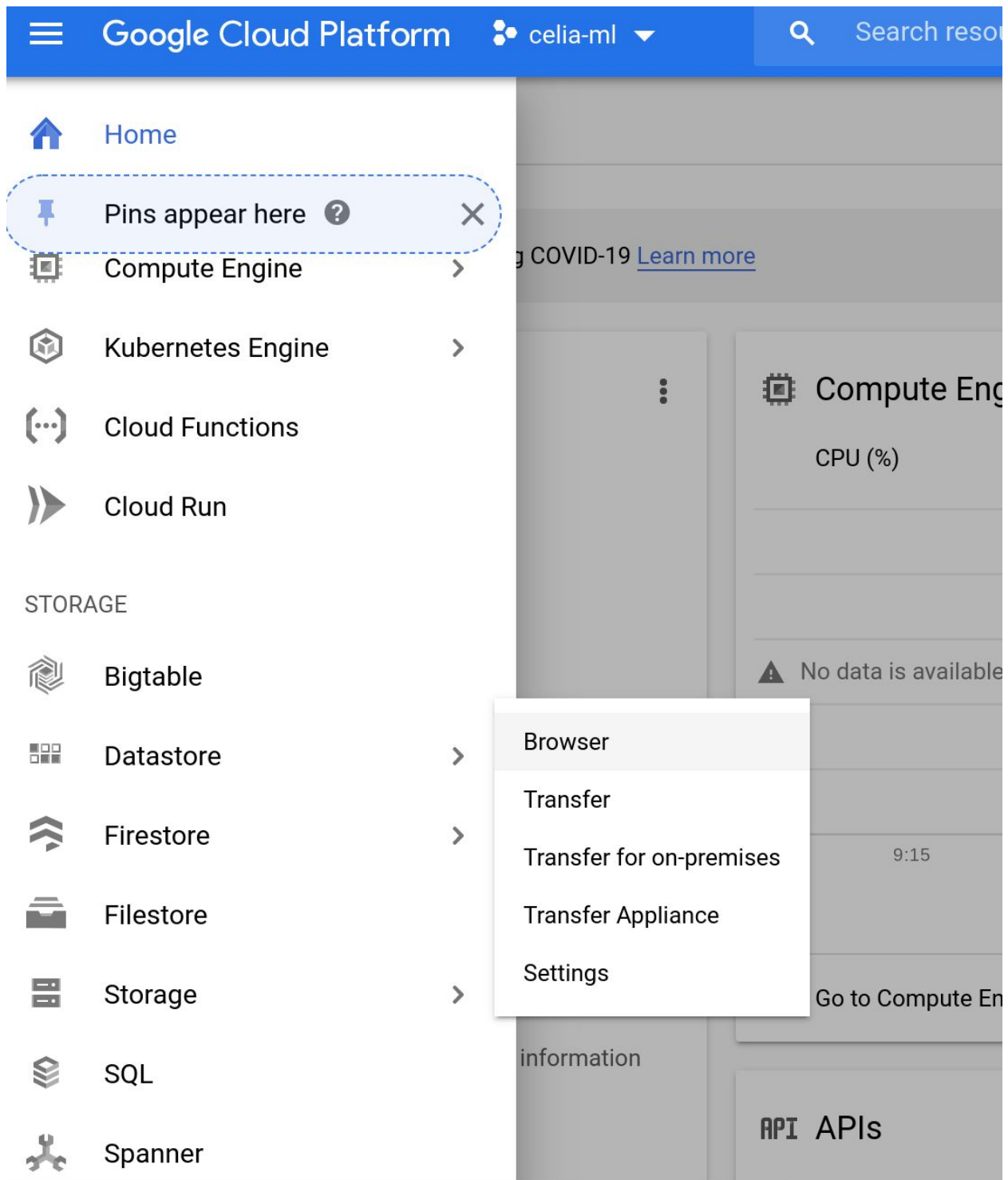
You can create a Cloud Storage bucket to keep the historical data and metadata AEMET files (\*.json). You can add [lifecycle rules](#) to delete old files or move them to a cheaper storage.

In the [Google Cloud Platform portal](#), create a Cloud Storage bucket in your project, and access it from Cloud Shell. To do this, go to Storage → Browse<sup>2</sup>.

---

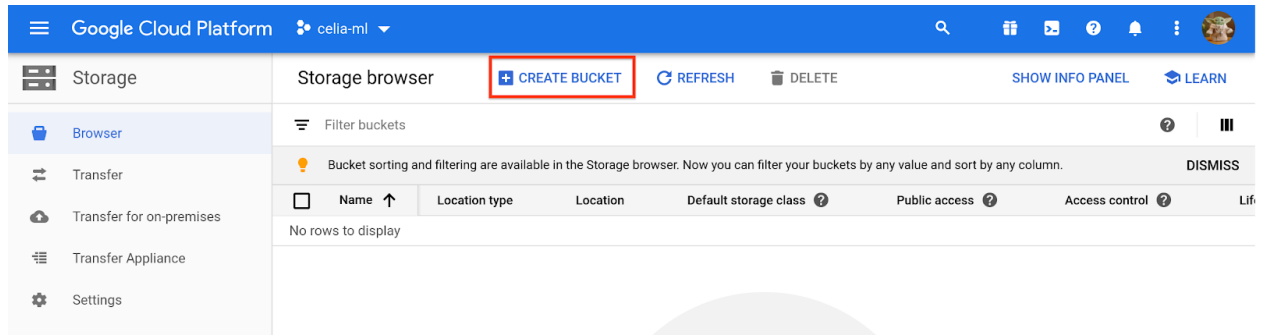
<sup>2</sup> It can also be created from the Cloud Shell with the command [gsutil mb](#).

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



Click on "Create bucket".

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



Provide the requested information for the bucket. Note the bucket name must be globally unique. A way to achieve this is to use the [Project ID](#)<sup>3</sup> as the bucket name. Choose the region where you want your bucket. We are going to leave the rest of settings by default.

<sup>3</sup> You can find the Project info in:

1. The GCP Home page,
2. Navigation Menu → IAM & Admin → Settings, or
3. Running `gcloud projects list` in the Cloud Shell.

## ← Create a bucket

### ✓ Name your bucket

Pick a **globally unique**, permanent name. [Naming guidelines](#)

celia-ml-aemet

Tip: Don't include any sensitive information

CONTINUE

### ✓ Choose where to store your data

This permanent choice defines the geographic placement of your data and affects cost, performance, and availability. [Learn more](#)

#### Location type

- ☒ Region  
Lowest latency within a single region
- ☐ Dual-region  
High availability and low latency across 2 regions
- ☐ Multi-region  
Highest availability across largest area

#### Location

europa-west1 (Belgium) ▼

Once the bucket is created, you can click on “Upload files” from your computer, or upload the file using the command [gsutil cp](#) in the Cloud Shell.

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



## References

### AEMET

[AEMET](#), Spanish Meteorological Agency.

[AEMET OpenData](#).

[AEMET OpenData Documentation](#).

[AEMET OpenData Client Application Examples](#).

[AEMET OpenData Codegen](#) - to automatically create client applications in different programming languages.

### GCP Services

[Cloud Shell](#). [Google Cloud](#).

[Using Cloud Shell](#). [Google Cloud](#).

[Compute Engine](#). [Google Cloud](#).

[Cloud App Engine](#). [Google Cloud](#).

[Cloud Functions](#). [Google Cloud](#).

[Cloud Run](#). [Google Cloud](#).

[What is serverless?](#) [Google Cloud](#).

[BigQuery documentation](#). [Google Cloud](#).

[Cloud Storage](#). [Google Cloud](#).

[Object Lifecycle Management](#). [Google Cloud](#).

Retrieve meteorological information from AEMET (DMS geographic data) and load it into BigQuery  
<https://github.com/celiamuriel-google/aemet-to-bigquery.git>



## Using GCP SDK

[Installing Google Cloud SDK](#). [Cloud SDK](#).

[Initializing Cloud SDK](#). [Cloud SDK](#).

[Using the bq command-line tool](#). [Google Cloud](#).

[Creating and Managing Projects](#). [Google Cloud](#).

[mb - Make buckets](#). [gsutil tool](#). [Google Cloud](#).

[cp - Copy files and objects](#). [gsutil tool](#). [Google Cloud](#).

## Geographic Data

Stephen P. Morse. [Translate latitude and longitude from degrees+minute+second to decimal](#).

[Simple features](#). [Wikipedia](#).

[jq](#)