COVID-19 ANALYSIS ON GCP

Data Science for Management – LM91

Università di Catania

Introduction

The COVID-19 pandemic of 2019-2020 is the current pandemic of the so-called “new coronavirus disease“, caused by the SARS-CoV-2 virus.

Patients experience flu-like symptoms such as dermatitis, fever, dry cough, tiredness, difficulty breathing. In the most serious cases, often found in subjects already burdened by previous pathologies, pneumonia develops, acute renal failure, up to even death.

Patients also have leukopenia and lymphocytopenia. A test to diagnose SARS-CoV-2 infection has been available since mid-January 2020, but specific treatments and vaccines are currently being researched and tested.

The healings are spontaneous and the treatments are mainly aimed at managing symptoms and supporting vital functions even if some antiviral drugs already used to contrast other infections have been tested.

The aim of this project is to analyze data about COVID-19 using Google Cloud Platform.

The dataset

Data are stored in two GitHub repositories:

- Dipartimento di Protezione Civile (Region and Province) that stores data about:

Region

Fields	Type	Description
data	datetime	Date of update
nuovi_casi	integer	New positive cases
totale_casi	integer	Total number of cases
totale_positivi	integer	Total number of positive cases
tamponi	integer	Total number of tests
tamponi_test_antigenico_rapido	integer	Number of rapid tests
totale_positivi_test_antigenico_rapido	integer	Number of positive related to rapid tests
tamponi_test_molecolare	integer	Number of PCR tests
totale_positivi_test_molecolare	integer	Number of positive related to PCR tests
ingressi_terapia_intensiva	integer	Admission to intensive care
terapia_intensiva	integer	Number of cases in intensive care
totale_ospedalizzati	integer	Total number of hospitalized cases
ricoverati_con_sintomi	integer	Total number of hospitalized with symptoms

Province

Fields	Type	Description
data	datetime	Date of update
denominazione_regione	string	Region name
denominazione_provincia	string	Province name
totale_casi	integer	Total number of cases

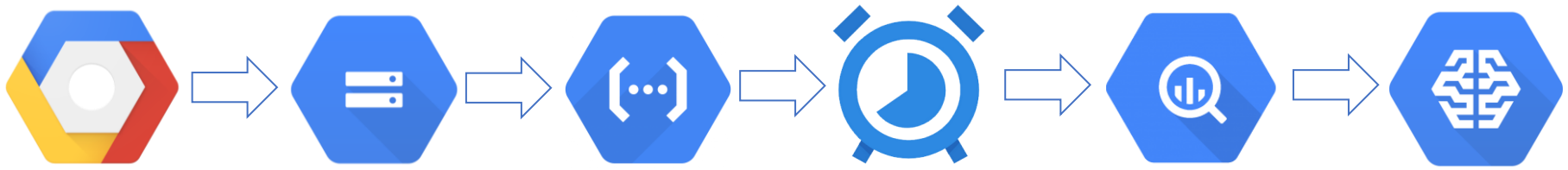
The dataset

- Commissario straordinario per l'emergenza Covid-19 (about vaccines) that stores all the data related to vaccinations:

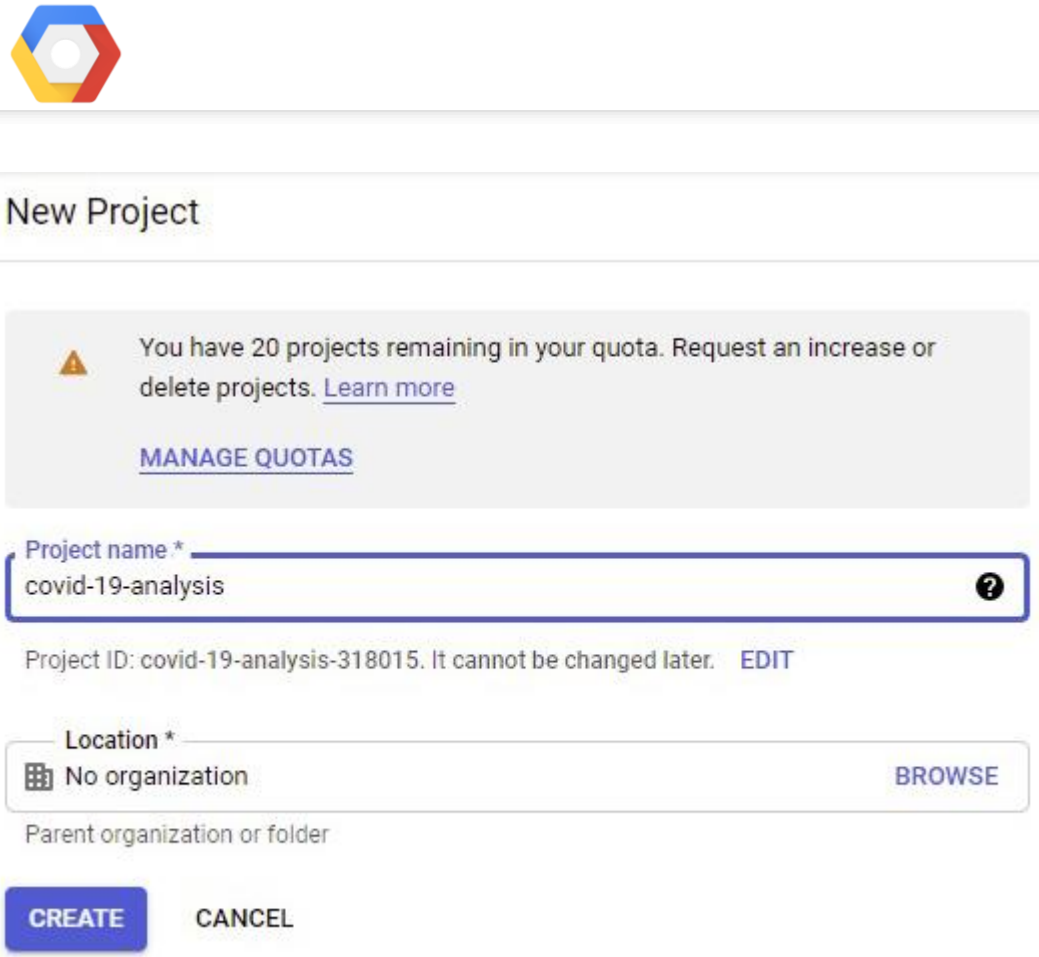
Fields	Type	Description
fornitore	string	Full name of the vaccine supplier
data_somministrazione	datetime	Date of administration
fascia_anagrafica	string	Age range of the subjects to which the vaccine was administered
sesso_maschile	integer	Total of male subjects to which the vaccine has been administered by day, region and age group
sesso_femminile	integer	Total of female subjects to which the vaccine has been administered by day, region and age group
prima_dose	integer	Number of first shots
seconda_dose	integer	Number of second shots
nome_regione	string	Standard area name (where a bilingual naming required)

The workflow

Here we describe the cloud workflow to download and update the data automatically from GitHub repository, the analysis using Notebooks and the data visualization

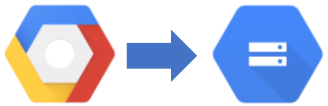


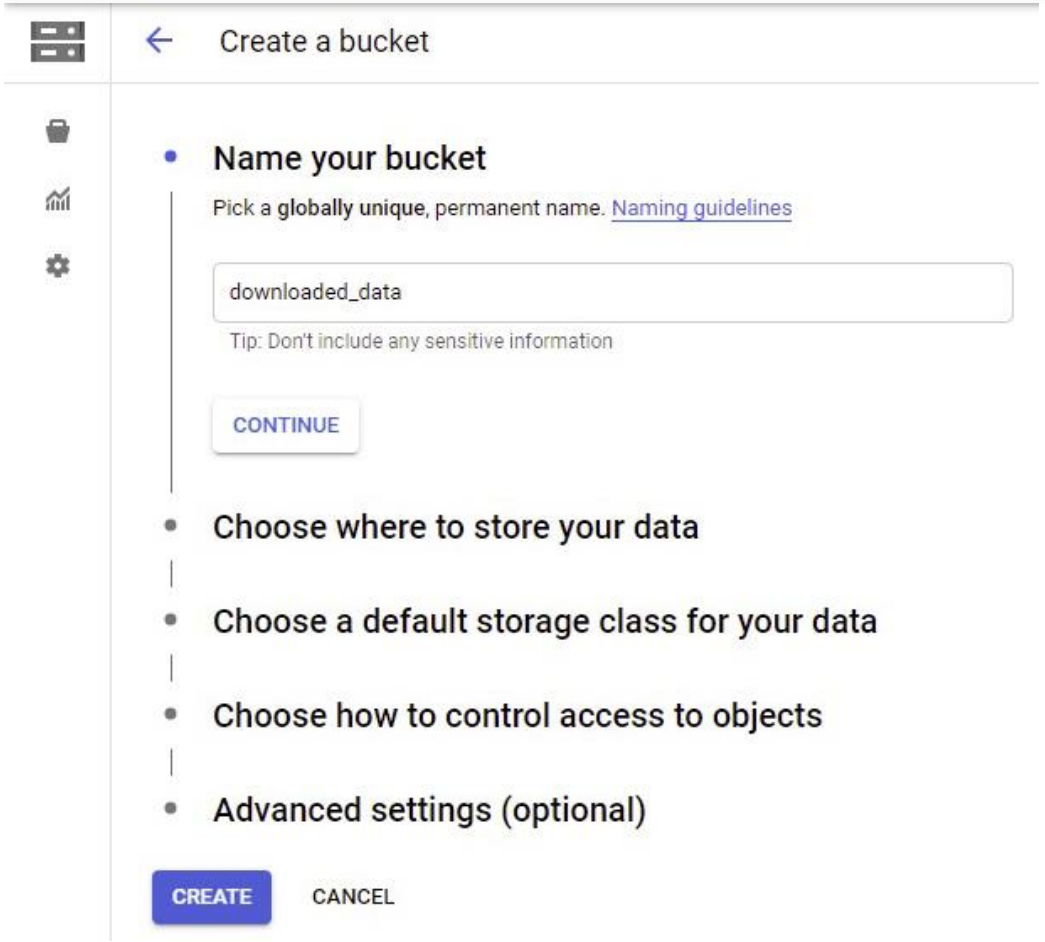
The workflow – Google Cloud Platform



In Google Cloud create a new project naming it ‘’covid-19-analysis’’

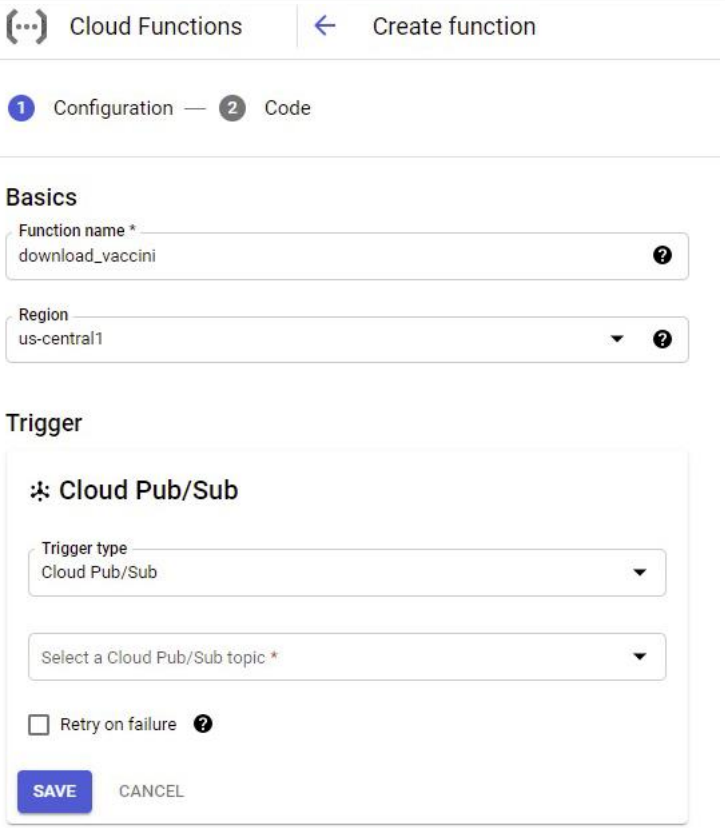
The workflow – Google Cloud Storage





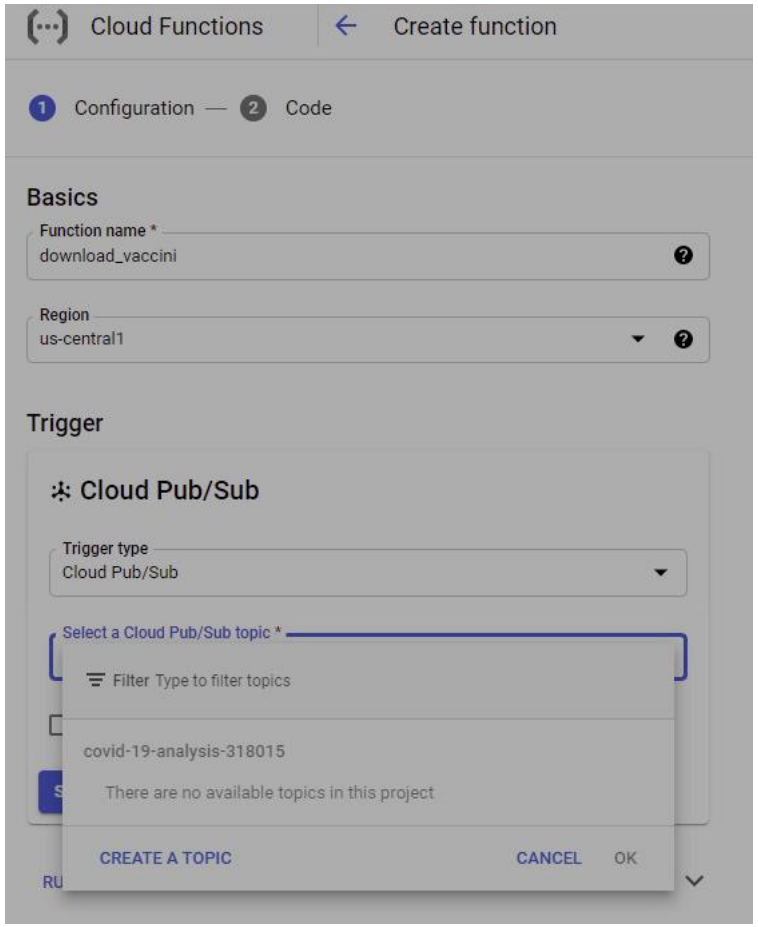
1. In Google Cloud Storage create a new bucket and naming it ‘’downloaded_data’’
2. Leave everything default and click on CREATE to create the bucket

The workflow – Google Cloud Functions



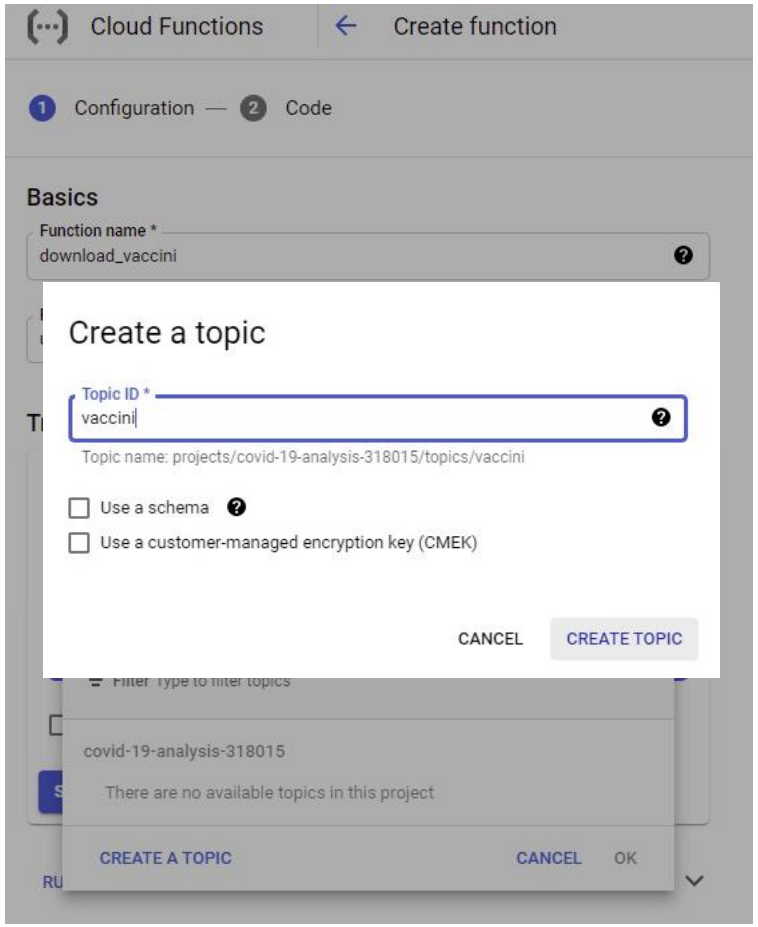
1. In Google Cloud Functions we create a new functions and name it as ‘’download_vaccini’’
2. Select Cloud Pub/Sub as trigger

The workflow – Google Cloud Functions



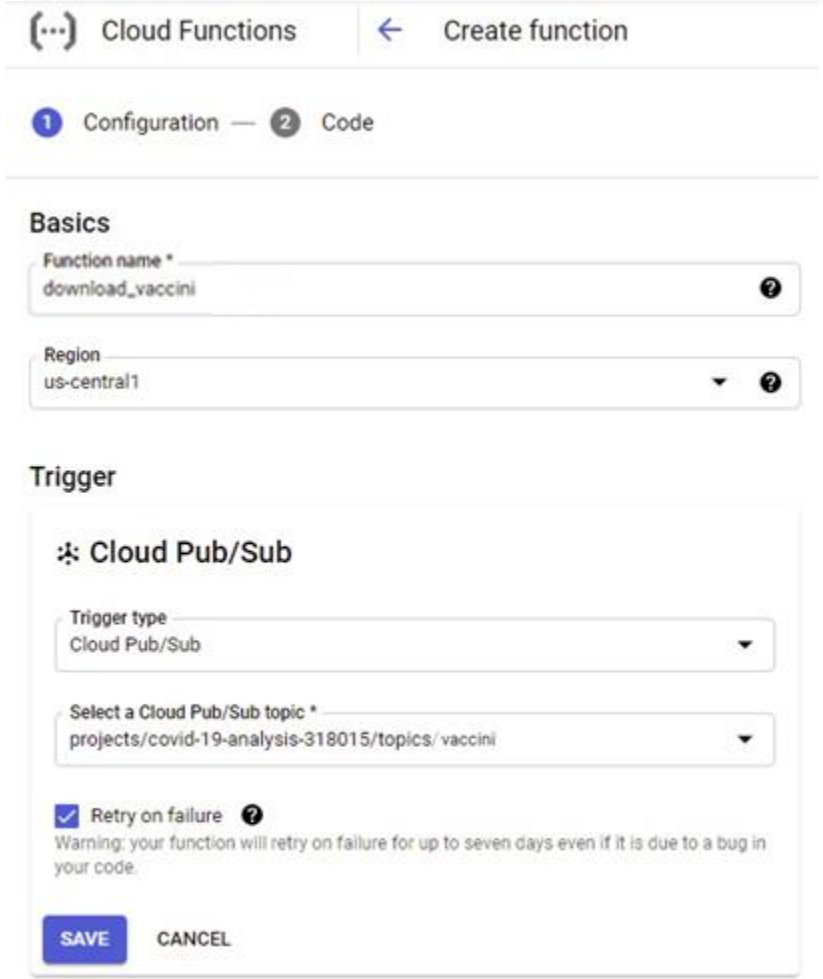
1. In Google Cloud Functions we create a new functions and name it as ‘’download_vaccini’’
2. Select Cloud Pub/Sub as trigger
3. We need to select the topic, in our case we need to create a new one clicking on CREATE A TOPIC

The workflow – Google Cloud Functions



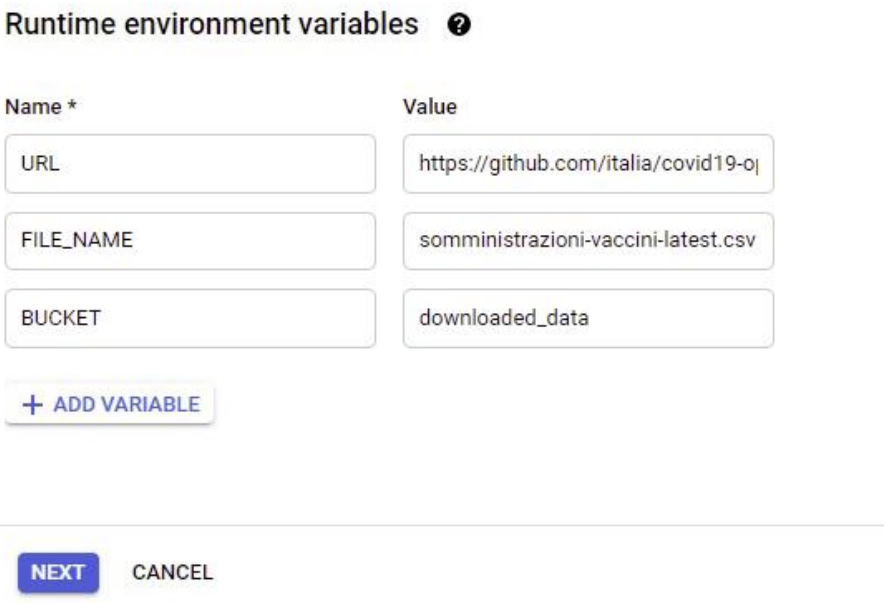
1. In Google Cloud Functions we create a new functions and name it as ‘’download_vaccini’’
2. Select Cloud Pub/Sub as trigger
3. We need to select the topic, in our case we need to create a new one clicking on CREATE A TOPIC
4. We name the Topic ID as ‘’vaccini’’ and CREATE TOPIC

The workflow – Google Cloud Functions



1. In Google Cloud Functions we create a new functions and name it as ‘’download_vaccini’’
2. Select Cloud Pub/Sub as trigger
3. We need to select the topic, in our case we need to create a new one clicking on CREATE A TOPIC
4. We name the Topic ID as ‘’vaccini’’ and CREATE TOPIC
5. We want also to retry on failure

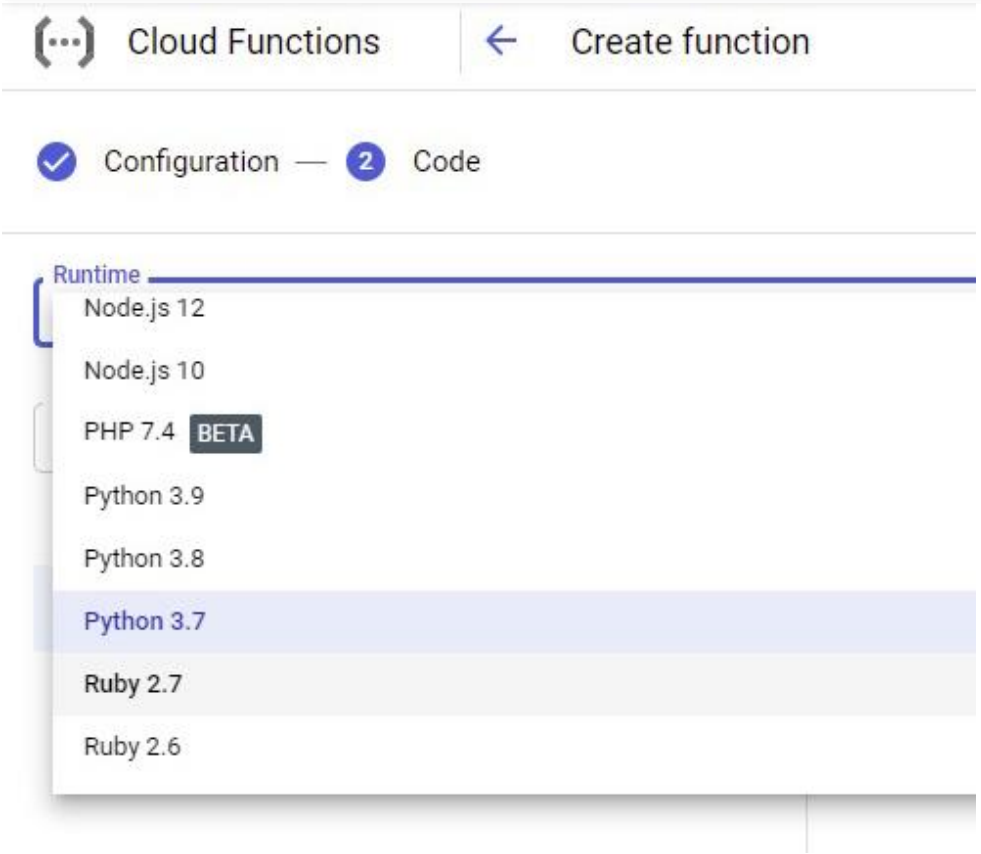
The workflow – Google Cloud Functions



1. In Google Cloud Functions we create a new functions and name it as ‘’download_vaccini’’
2. Select Cloud Pub/Sub as trigger
3. We need to select the topic, in our case we need to create a new one clicking on CREATE A TOPIC
4. We name the Topic ID as ‘’vaccini’’ and CREATE TOPIC
5. We want also to retry on failure
6. Add three environment variables
   - URL
   - FILE_NAME
   - BUCKET

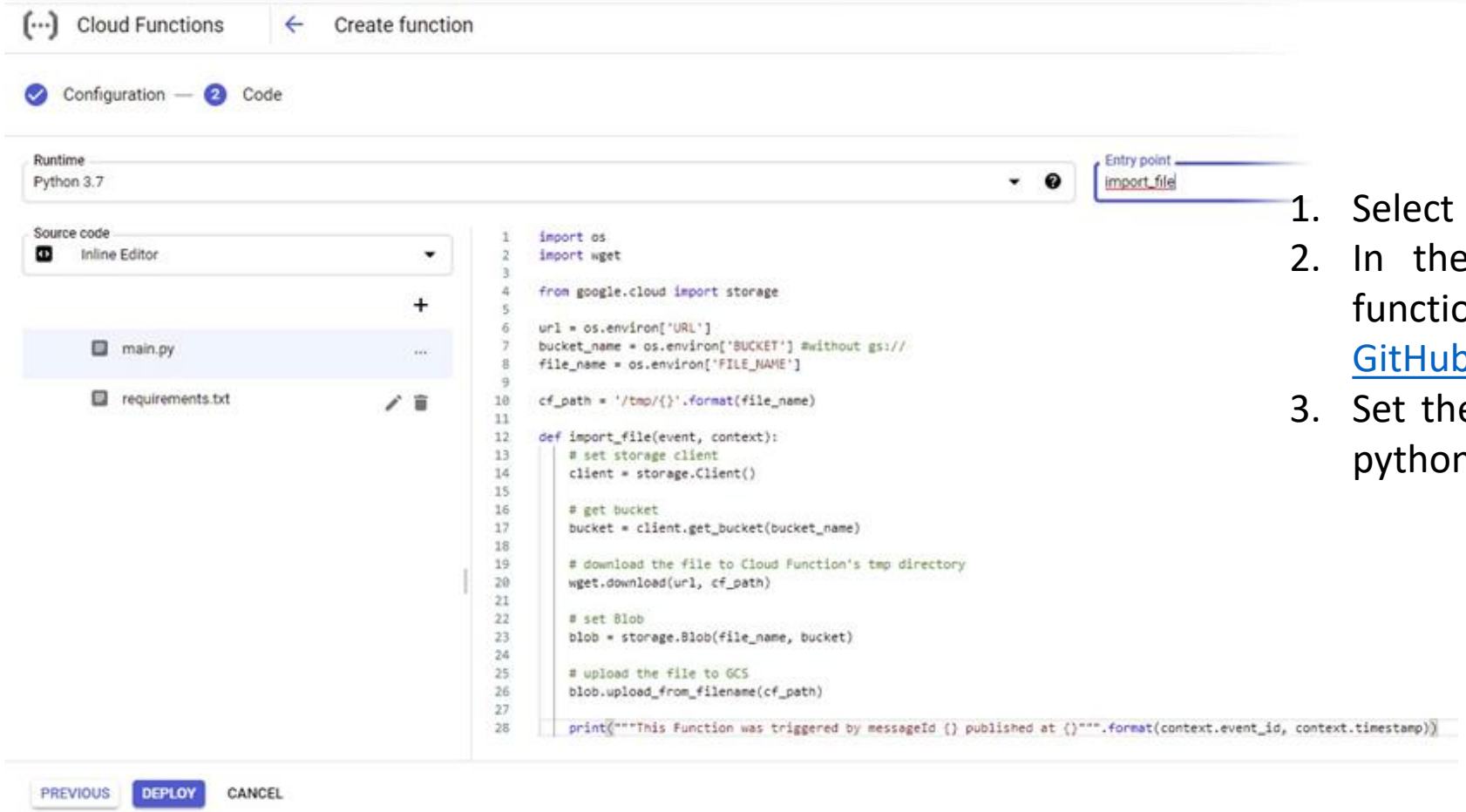
   Click on NEXT

The workflow – Google Cloud Functions



1. Select Python 3.7 as runtime

The workflow – Google Cloud Functions

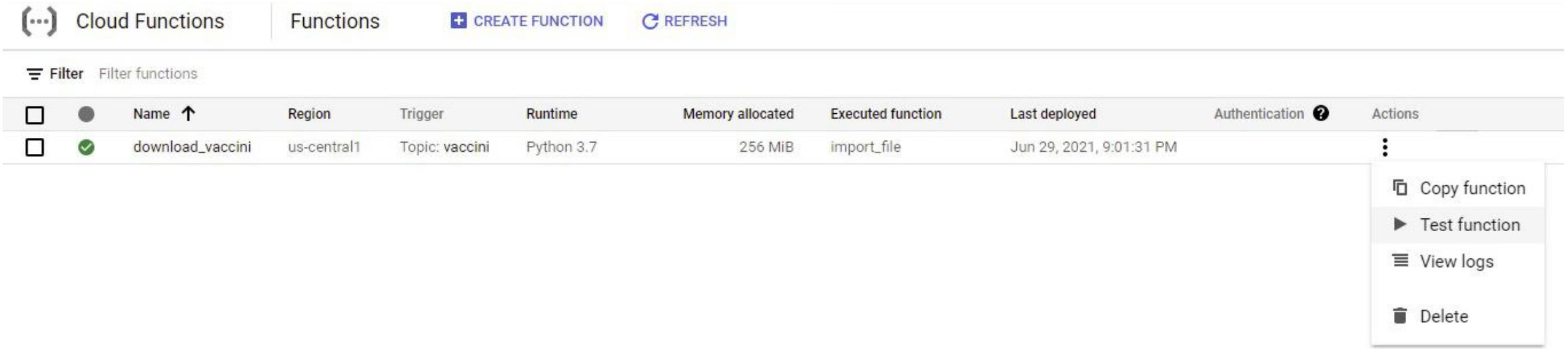


1. Select Python 3.7 as runtime
2. In the main.py script we code the function to download the file from the [GitHub repository]
3. Set the entrypoint as the name of the python function

The workflow – Google Cloud Functions

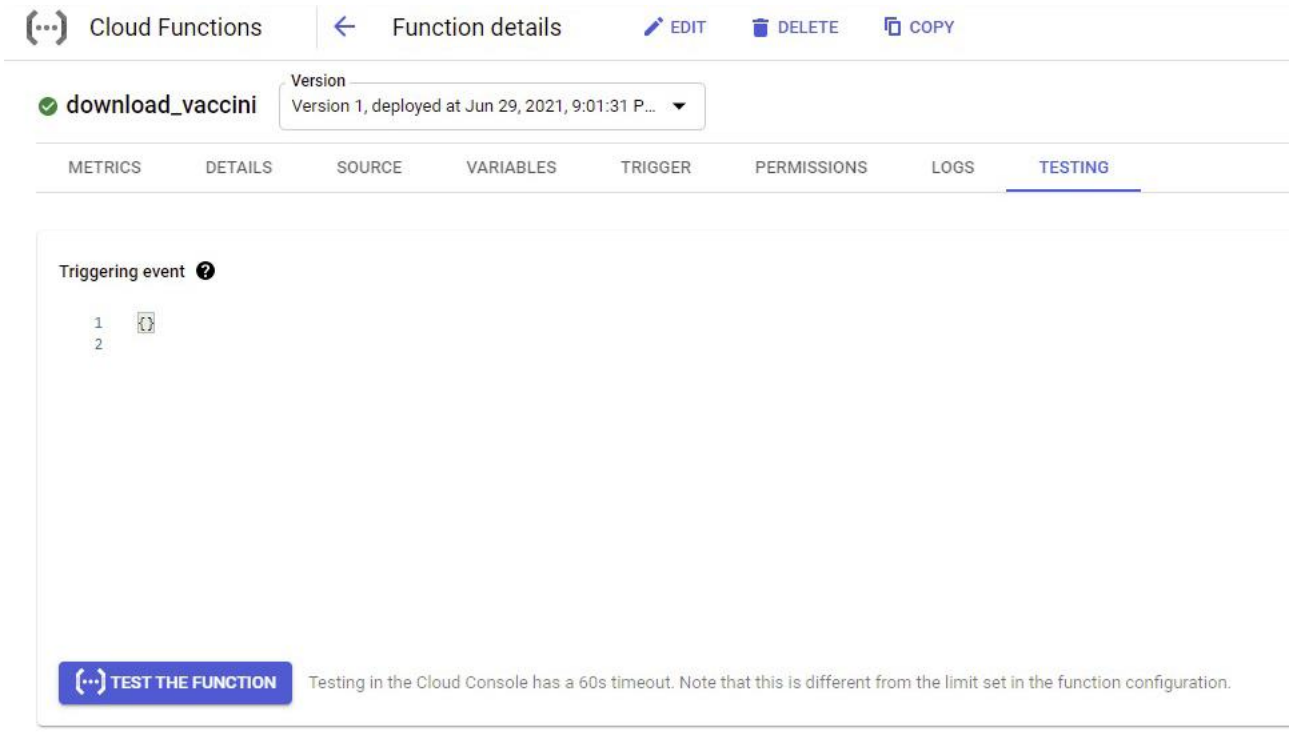
1. Select Python 3.7 as runtime
2. In the main.py script we code the function to download the file from the GitHub repository
3. Set the entrypoint as the name of the python function
4. Insert the function dependencies
5. DEPLOY the function

The workflow – Google Cloud Functions



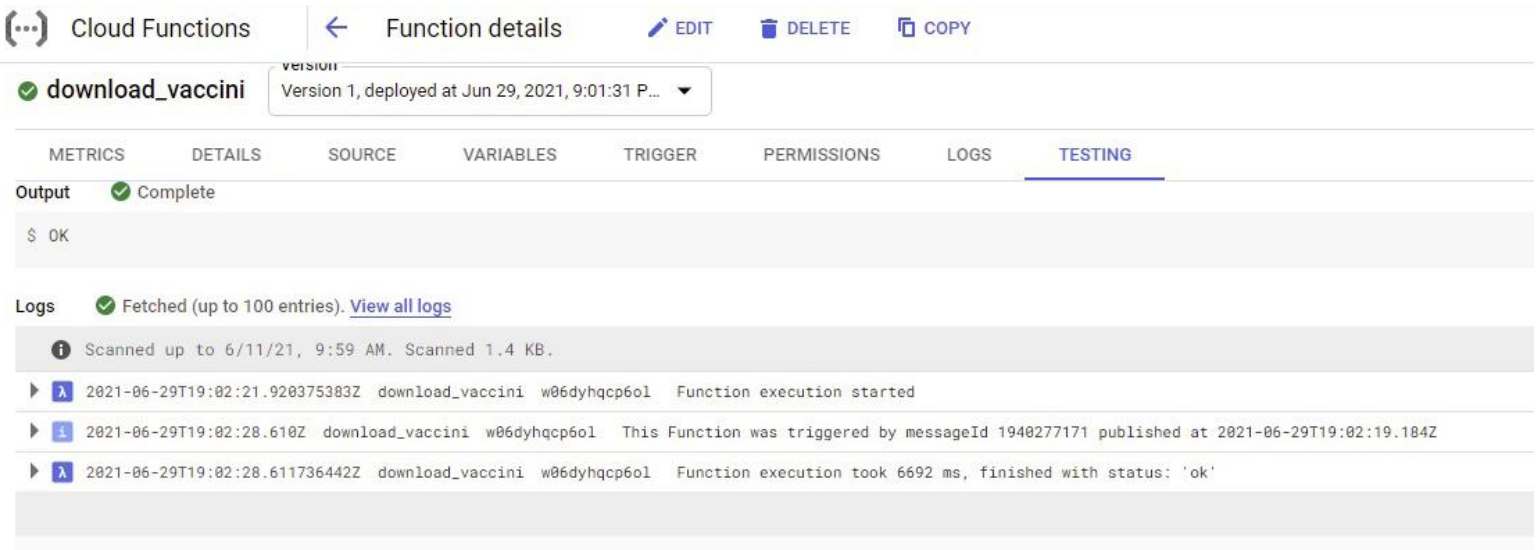
The function has been correctly created but not deployed yet.
Just click on Test function to firstly deploy the function

The workflow – Google Cloud Functions



1. Leave everything default just click on TEST THE FUNCTION

The workflow – Google Cloud Functions



1. Leave everything default just click on TEST THE FUNCTION
2. The function execution started
3. When the function is finished return the status ‘’ok’’

The workflow – Google Cloud Functions

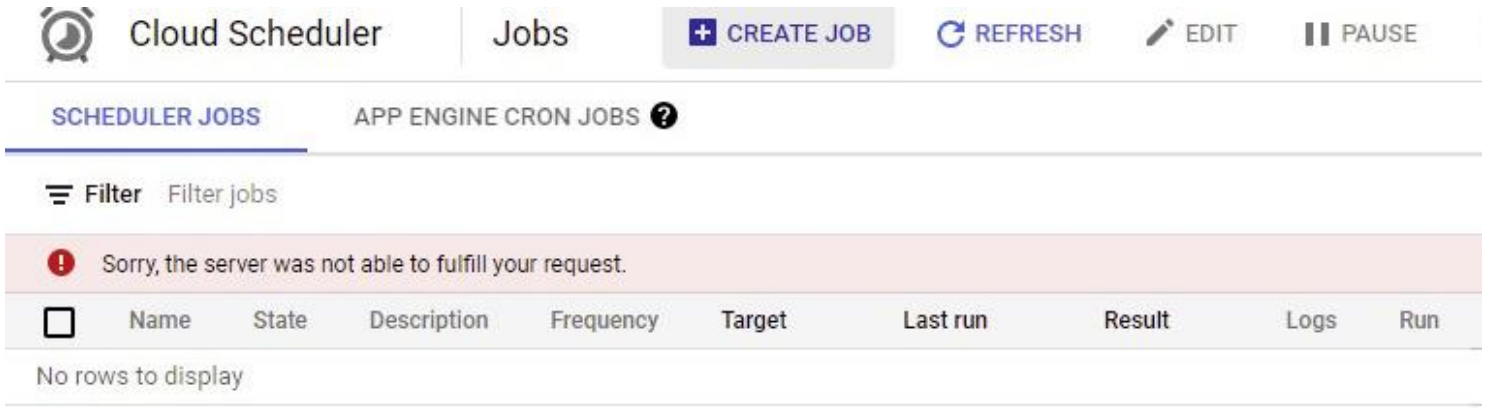
Cloud Functions | Functions | CREATE FUNCTION | REFRESH

Filter: Filter functions

	Name	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
☐	✓ download_province	us-central1	Topic: province	Python 3.7	256 MiB	import_file	Jun 29, 2021, 9:05:28 PM		⋮
☐	✓ download_regioni	us-central1	Topic: regioni	Python 3.7	256 MiB	import_file	Jun 29, 2021, 9:07:36 PM		⋮
☐	✓ download_vaccini	us-central1	Topic: vaccini	Python 3.7	256 MiB	import_file	Jun 29, 2021, 9:01:31 PM		⋮

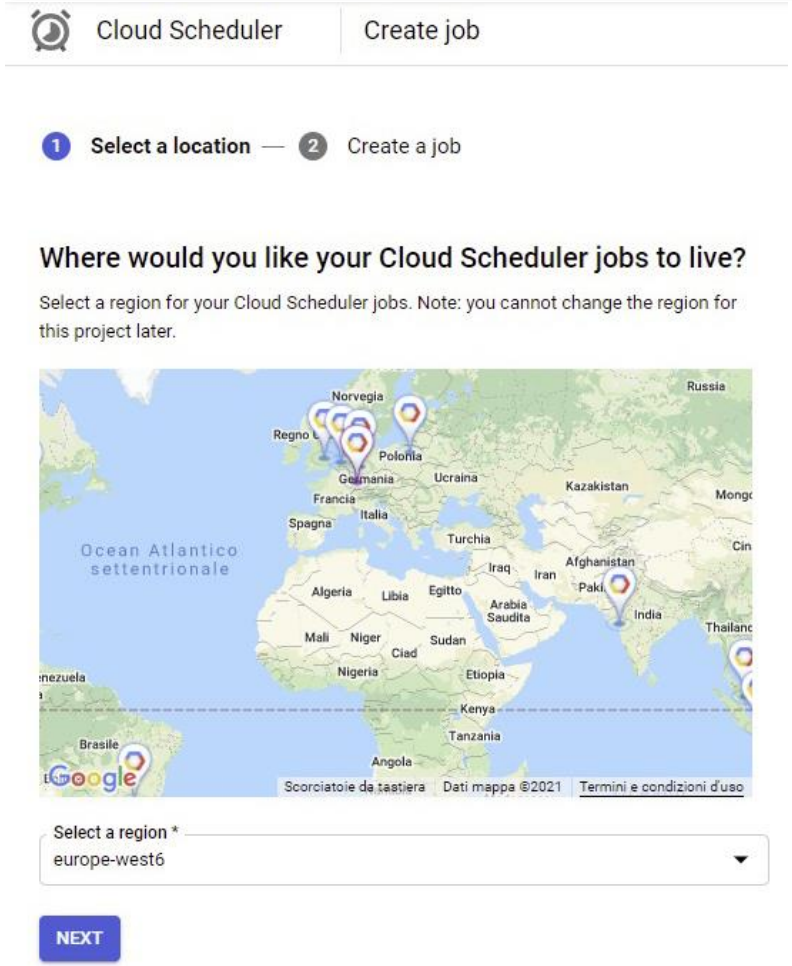
1. Leave everything default just click on TEST THE FUNCTION
2. The function execution started
3. When the function is finished return the status ‘’ok’’
4. Replicate the previous steps for the other files

The workflow – Google Cloud Scheduler



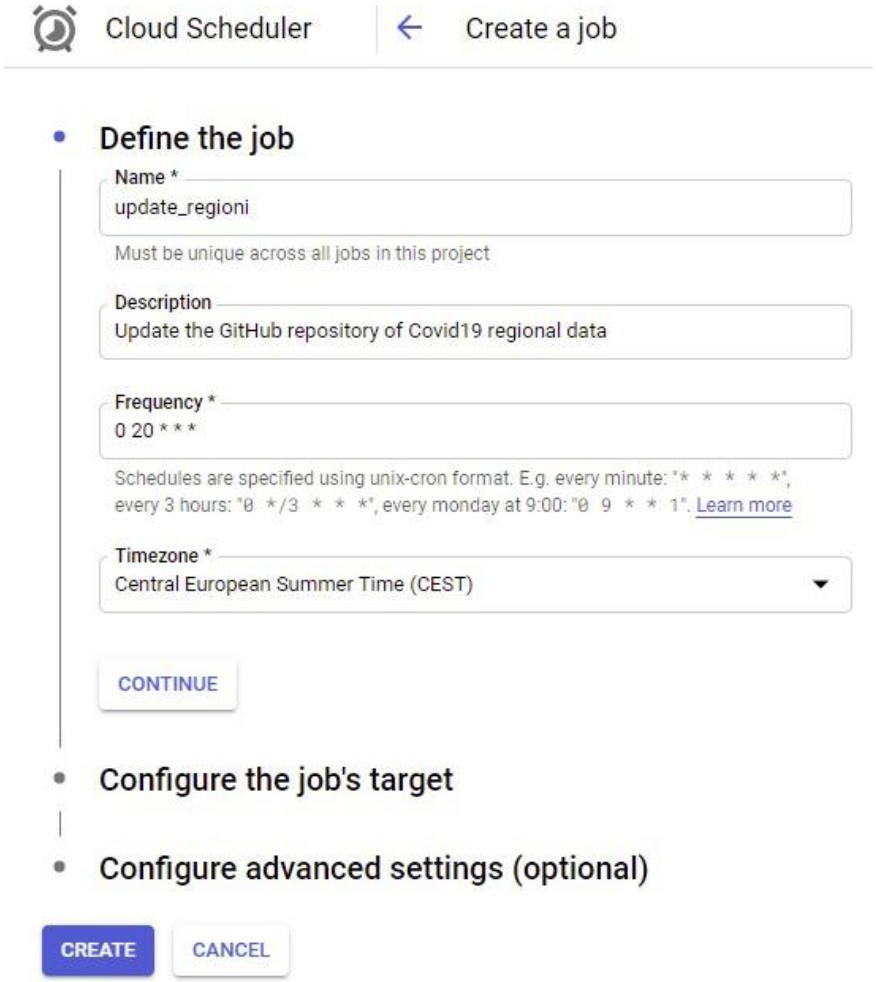
1. Click on CREATE JOB to create a new schedule

The workflow – Google Cloud Scheduler



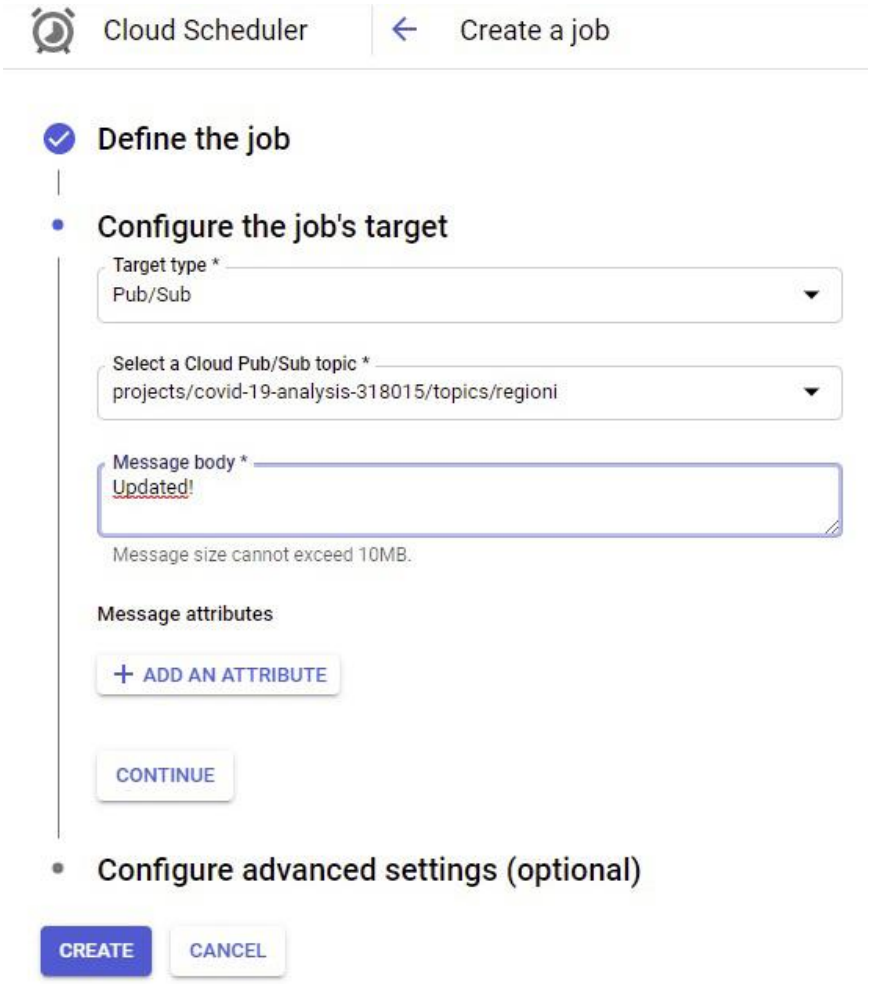
1. Click on CREATE JOB to create a new schedule
2. Choose where the Cloud Scheduler jobs live

The workflow – Google Cloud Scheduler



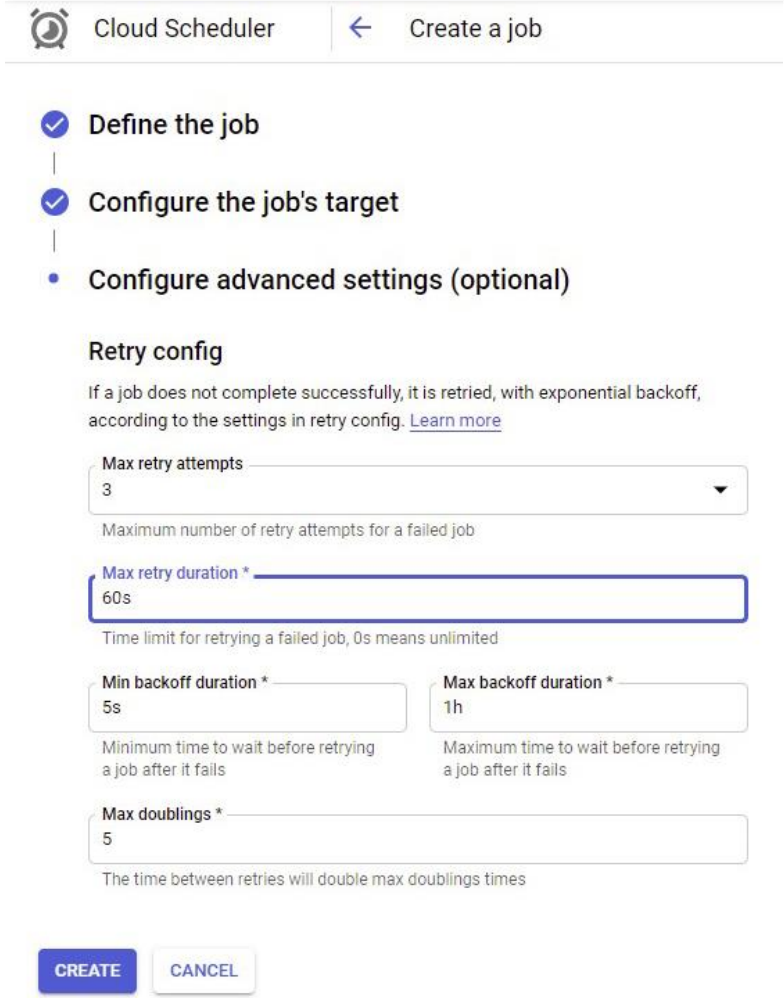
1. Click on CREATE JOB to create a new schedule
2. Choose where the Cloud Scheduler jobs live
3. Put name, description, frequency and timezone (in our case it runs every day at 20 PM)

The workflow – Google Cloud Scheduler



1. Click on CREATE JOB to create a new schedule
2. Choose where the Cloud Scheduler jobs live
3. Put name, description, frequency and timezone (in our case it runs every day at 20 PM)
4. Select Pub/Sub as target
5. Select the topic previously created
6. Insert the Message Body

The workflow – Google Cloud Scheduler



1. Click on CREATE JOB to create a new schedule
2. Choose where the Cloud Scheduler jobs live
3. Put name, description, frequency and timezone (in our case it runs every day at 20 PM)
4. Select Pub/Sub as target
5. Select the topic previously created
6. Insert the Message Body
7. Insert how many attempts and max retry durations
8. CREATE the schedule

The workflow – Google Cloud Scheduler

Cloud Scheduler | Jobs | CREATE JOB | REFRESH | EDIT | PAUSE | RESUME | DELETE

SCHEDULER JOBS | APP ENGINE CRON JOBS

Filter: Filter jobs

Name	State	Description	Frequency	Target	Last run	Result	Logs	Run
update_regioni	Enabled	Update the GitHub repository of Covid19 regional data	0 20 * * * (Europe/Rome)	Topic : projects/covid-19-analysis-318015/topics/regioni		Has not run yet	View	RUN NOW

The job has been scheduled but not run yet.
Click on RUN NOW to execute the job!

The workflow – Google Cloud Scheduler

Cloud Scheduler | Jobs | CREATE JOB | REFRESH | EDIT | PAUSE | RESUME | DELETE

SCHEDULER JOBS | APP ENGINE CRON JOBS

Filter: Filter jobs

Name	State	Description	Frequency	Target	Last run	Result	Logs	Run
update_regioni	Enabled	Update the GitHub repository of Covid19 regional data	0 20 * * * (Europe/Rome)	Topic : projects/covid-19-analysis-318015/topics/regioni	Jun 29, 2021, 9:12:33 PM	Success	View	RUN NOW

The job has been scheduled correctly and executed successfully!

The workflow – Google Cloud Scheduler

Cloud Scheduler | Jobs | CREATE JOB | REFRESH | EDIT | PAUSE | RESUME | DELETE

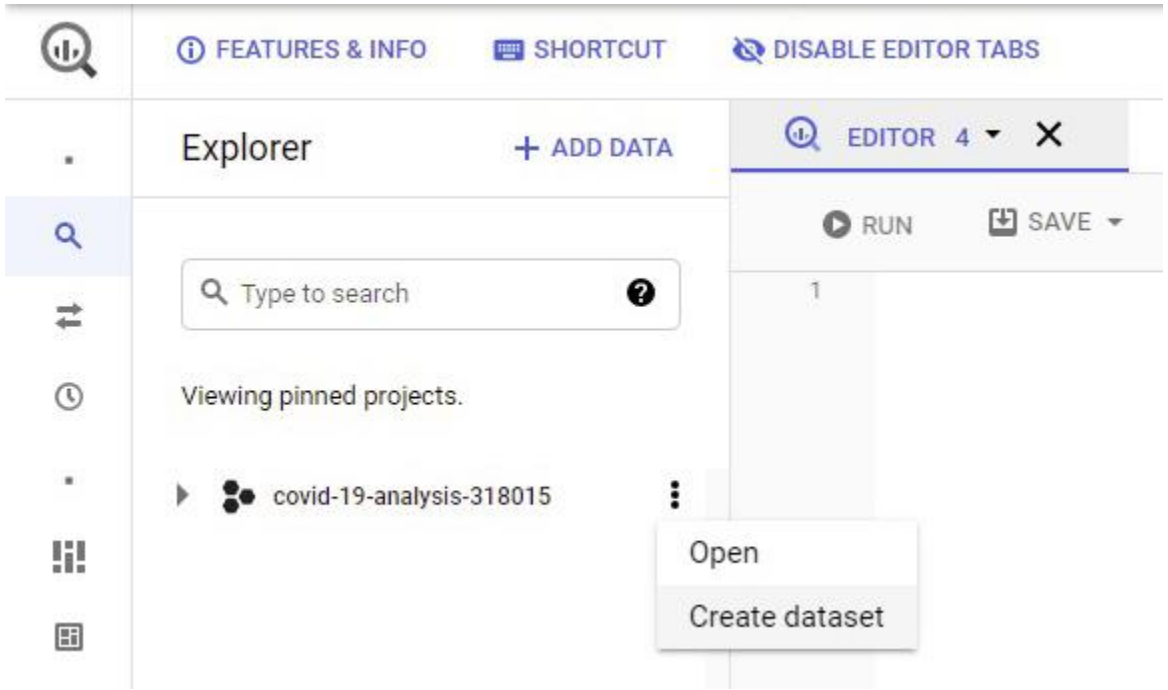
SCHEDULER JOBS | APP ENGINE CRON JOBS

Filter: Filter jobs

Name	State	Description	Frequency	Target	Last run	Result	Logs	Run
update_province	Enabled	Update the GitHub repository of Covid19 province data	0 20 * * * (Europe/Rome)	Topic : projects/covid-19-analysis-318015/topics/province	Jun 29, 2021, 9:13:43 PM	Success	View	RUN NOW
update_regioni	Enabled	Update the GitHub repository of Covid19 regional data	0 20 * * * (Europe/Rome)	Topic : projects/covid-19-analysis-318015/topics/regioni	Jun 29, 2021, 9:12:33 PM	Success	View	RUN NOW
update_vaccini	Enabled	Update the GitHub repository of Covid19 vaccines data	0 20 * * * (Europe/Rome)	Topic : projects/covid-19-analysis-318015/topics/vaccini	Jun 29, 2021, 9:15:01 PM	Success	View	RUN NOW

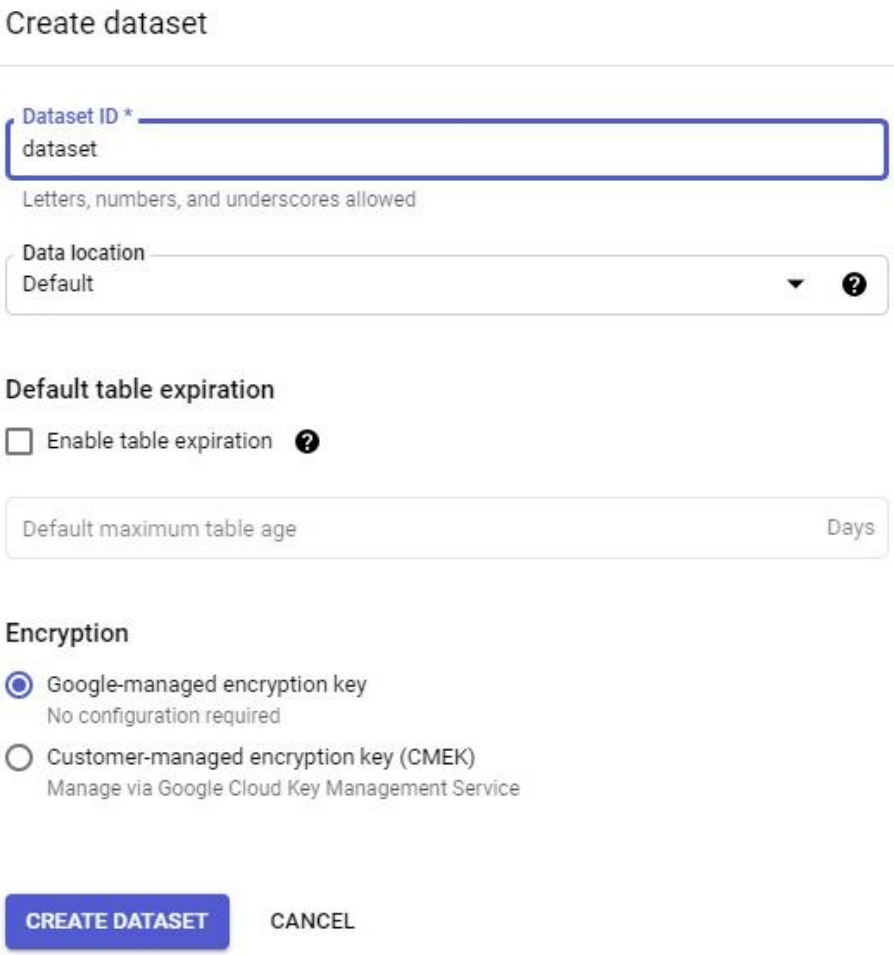
Repeat for the other schedules!

The workflow – Google BigQuery



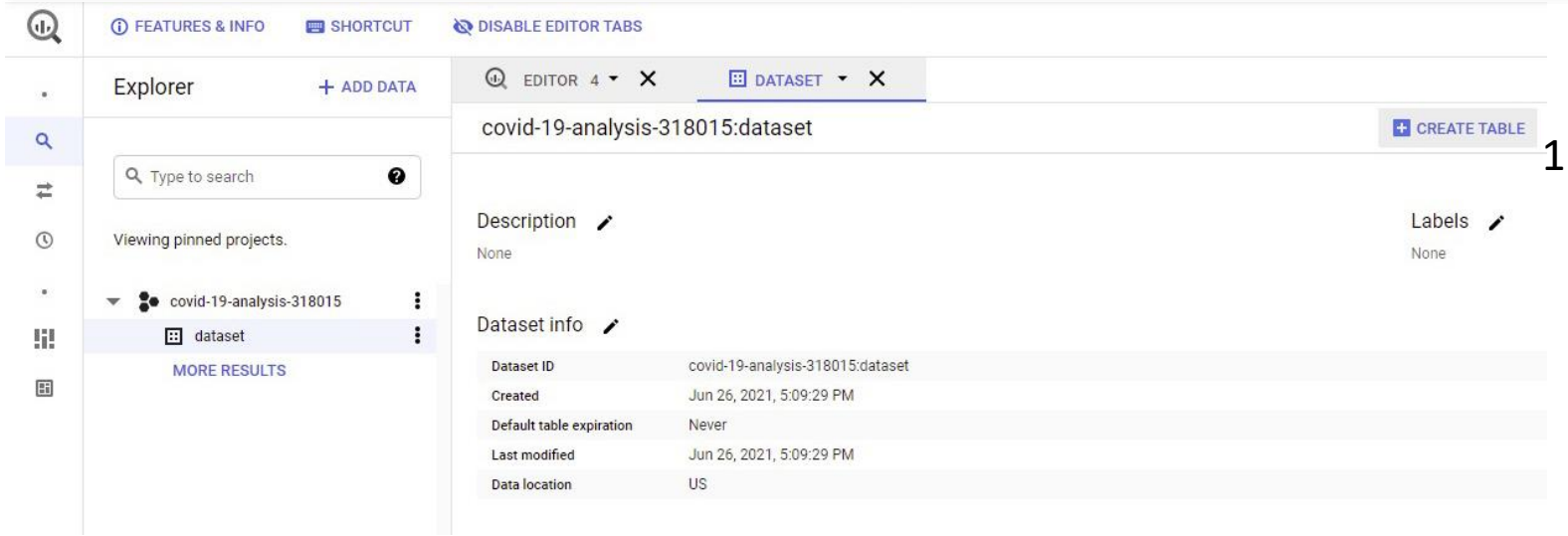
1. In Google BigQuery we select the project and Create dataset

The workflow – Google BigQuery



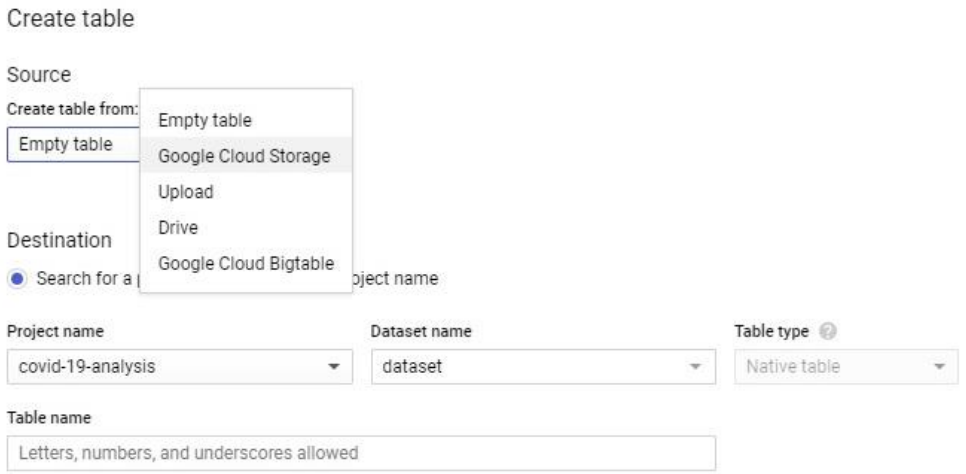
1. In Google BigQuery we select the project and Create dataset
2. Insert the id for Dataset ID and we choose ‘’dataset’’
3. Click on CREATE DATASET to create the dataset

The workflow – Google BigQuery



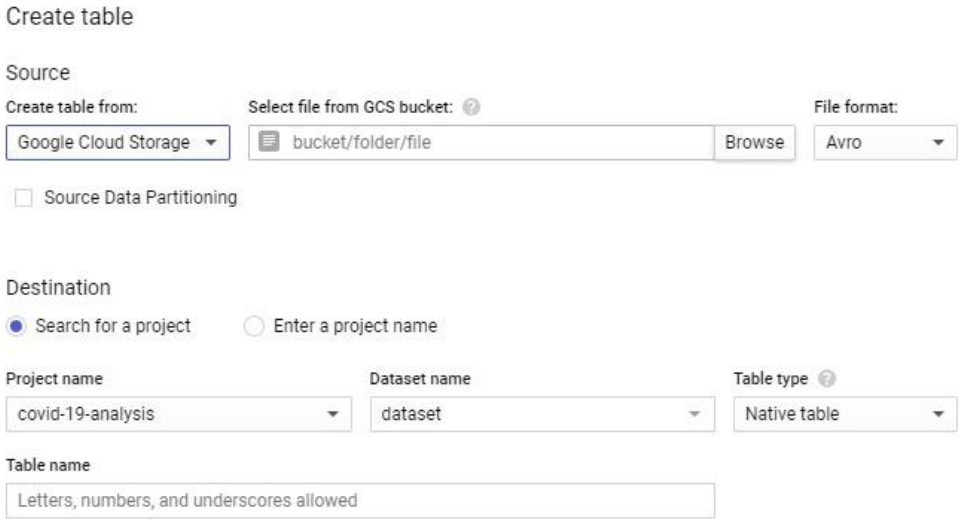
1. CREATE TABLE to create a table in the dataset

The workflow – Google BigQuery



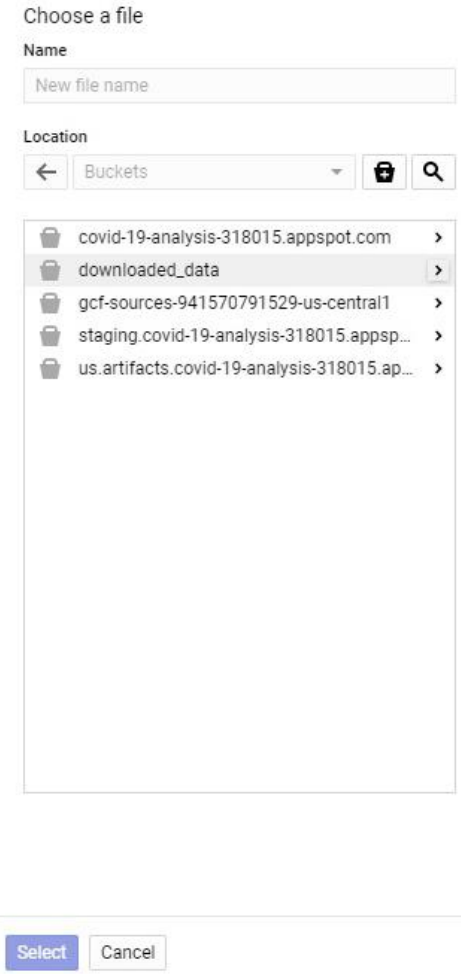
1. CREATE TABLE to create a table in the dataset
2. As source we use Google Cloud Storage

The workflow – Google BigQuery



1. CREATE TABLE to create a table in the dataset
2. As source we use Google Cloud Storage and Browse the file in the bucket

The workflow – Google BigQuery

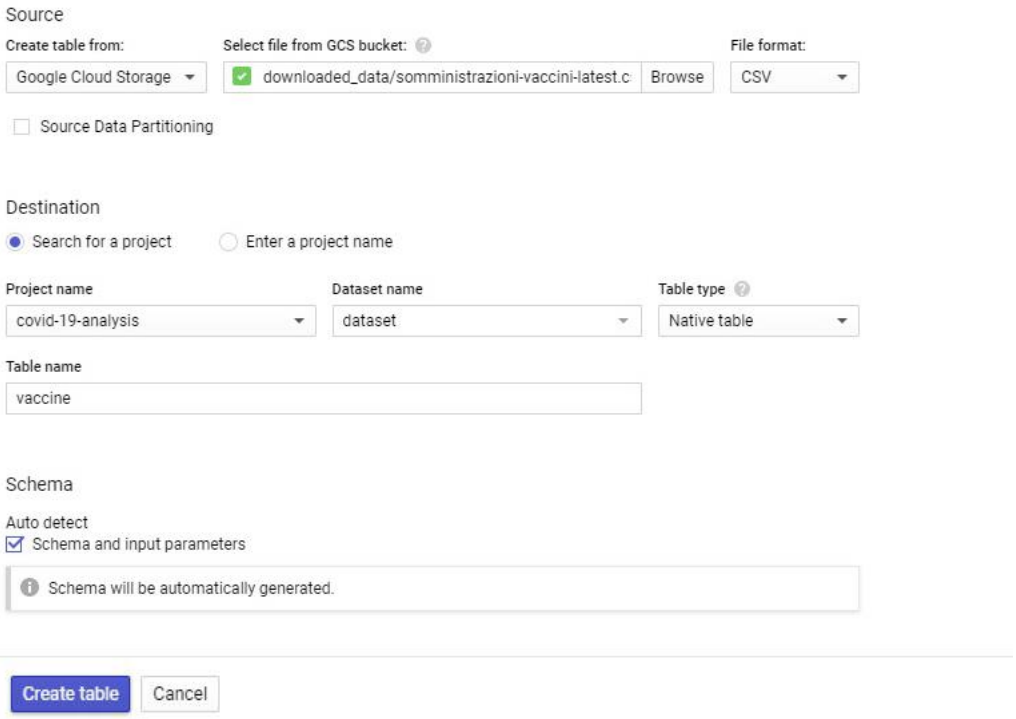


1. CREATE TABLE to create a table in the dataset
2. As source we use Google Cloud Storage and Browse the file in the bucket
3. We select the Bucket

The workflow – Google BigQuery

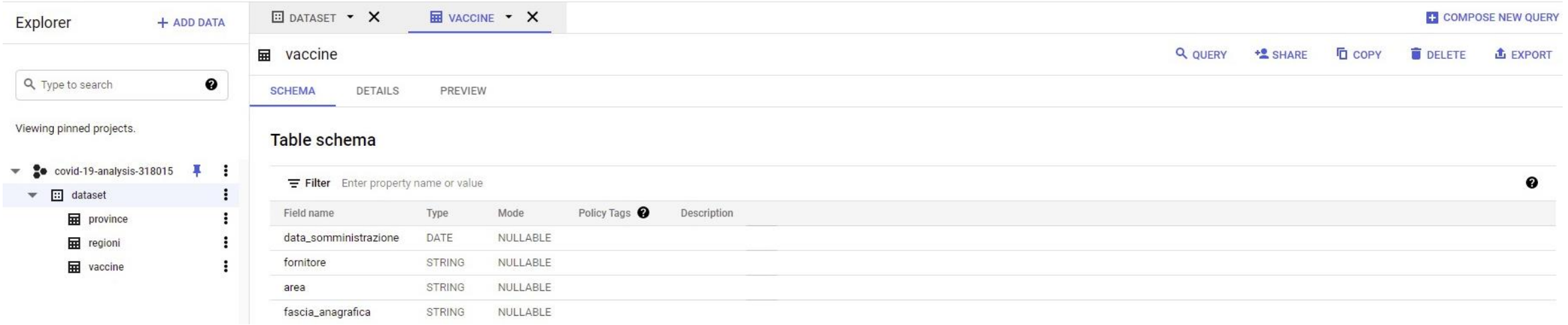
1. CREATE TABLE to create a table in the dataset
2. As source we use Google Cloud Storage and Browse the file in the bucket
3. We select the Bucket and the corresponding file

The workflow – Google BigQuery



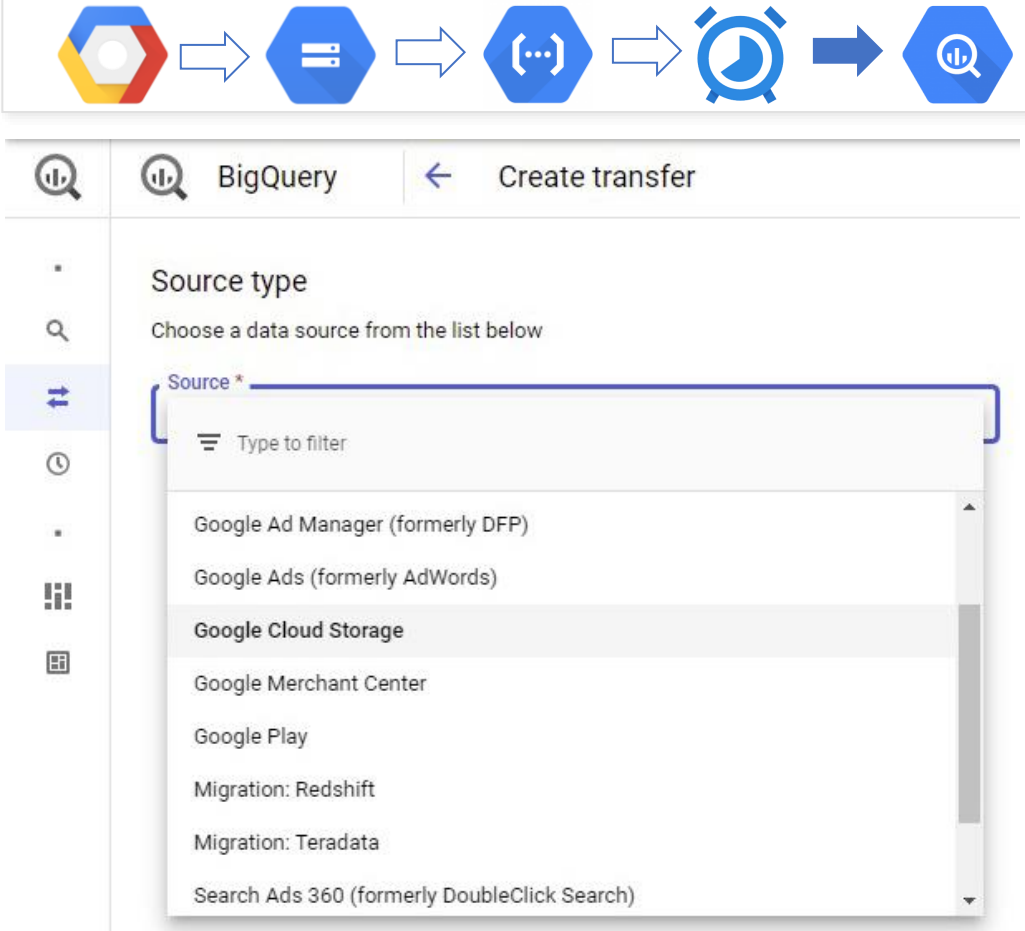
1. CREATE TABLE to create a table in the dataset
2. As source we use Google Cloud Storage and Browse the file in the bucket
3. We select the Bucket and the corresponding file
4. Insert the table name and auto-detect the schema
5. Click on CREATE TABLE to create the table
6. Repeat the previous steps for the other files

The workflow – Google BigQuery



Tables have been created correctly!

The workflow – Google BigQuery (Data transfer)



1. Select the type of Data Transfer (in our case Google Cloud Storage)

The workflow – Google BigQuery (Data transfer)

Source type

Choose a data source from the list below

Source *
Google Cloud Storage

This is the Google Cloud Storage configuration. Learn more

Transfer config name

Display name *
transfer_vaccine

Schedule options

Start now Start at set time

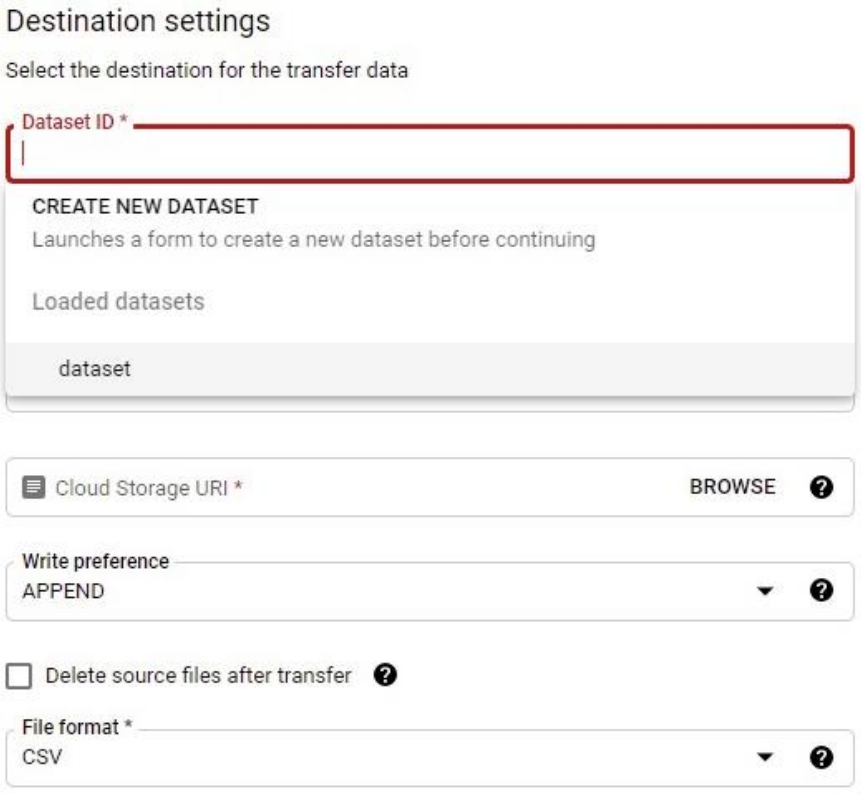
Repeats *
Daily

Start date and run time *
6/30/21, 8:15 PM CEST

Every day at 20:15:00 Europe/Paris

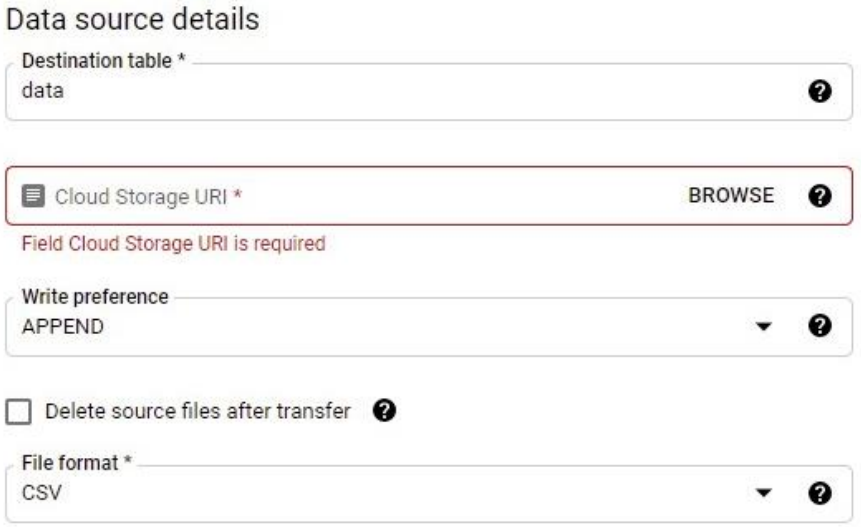
1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the transfer name and schedule options

The workflow – Google BigQuery (Data transfer)



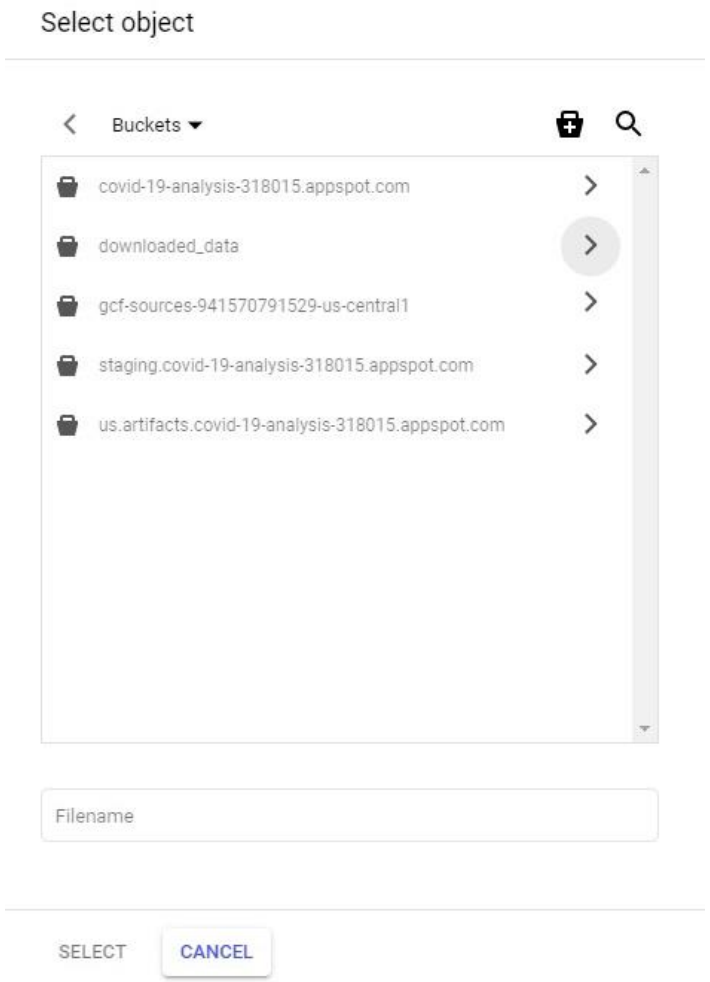
1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the dataset ID

The workflow – Google BigQuery (Data transfer)



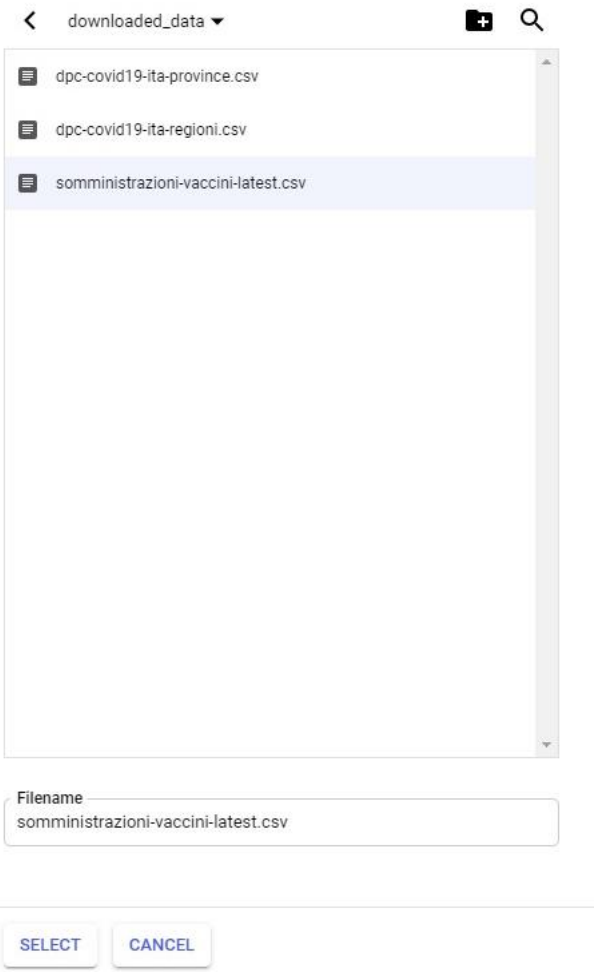
1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the destionation dataset ID
4. Select the destionation table
5. Select the Google Cloud Storage URI

The workflow – Google BigQuery (Data transfer)



1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the destionation dataset ID
4. Select the destionation table
5. Select the Google Cloud Storage URI
6. Select the Bucket

The workflow – Google BigQuery (Data transfer)



1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the destionation dataset ID
4. Select the destionation table
5. Select the Google Cloud Storage URI
6. Select the Bucket and the file

The workflow – Google BigQuery (Data transfer)

Data source details

Destination table *
data

Cloud Storage URI *
downloaded_data/somministrazioni-vaccini-latest.csv BROWSE

Write preference
MIRROR

Delete source files after transfer

File format *
CSV

Transfer Options

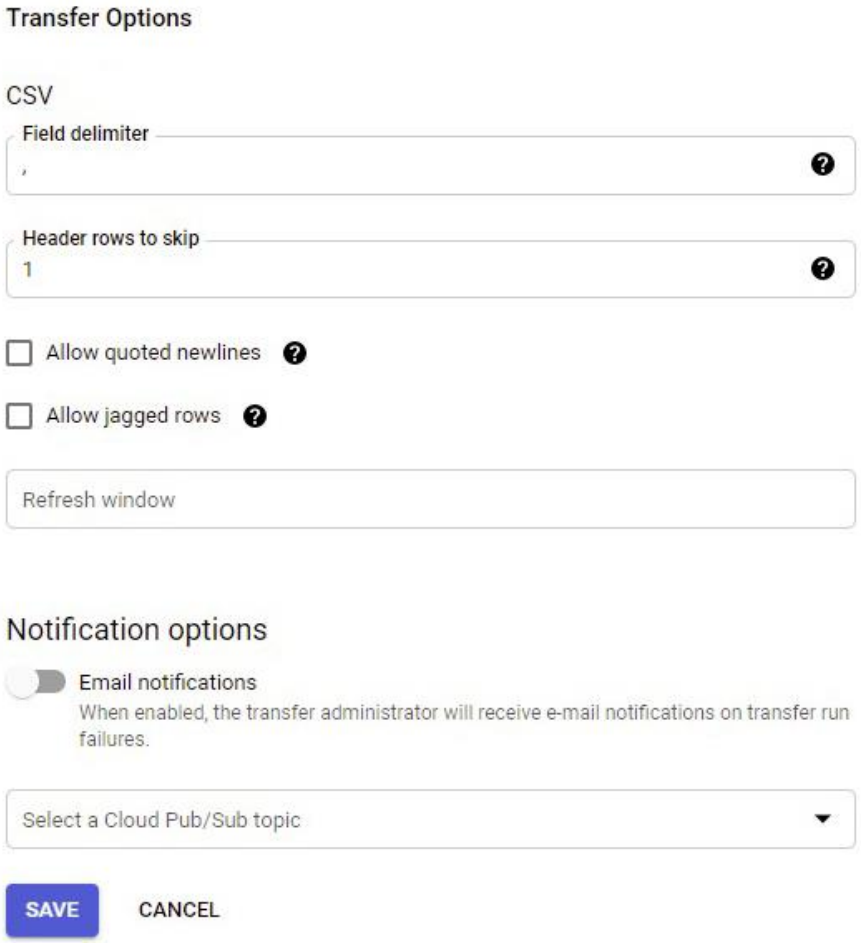
All Formats

Number of errors allowed
0

Decimal target types

1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the destionation dataset ID
4. Select the destionation table
5. Select the Google Cloud Storage URI
6. Select the Bucket and the file
7. To overwrite the entire table select MIRROR

The workflow – Google BigQuery (Data transfer)



1. Select the type of Data Transfer (in our case Google Cloud Storage)
2. Insert the name and schedule options
3. Select the destionation dataset ID
4. Select the destionation table
5. Select the Google Cloud Storage URI
6. Select the Bucket and the file
7. To overwrite the entire table select MIRROR
8. Insert the delimiter character and skip the first header row
9. Click on SAVE

The workflow – Google BigQuery (Data transfer)

BigQuery ← Data transfers + CREATE TRANSFER

Filter Filter transfer configs

	Display name	Source	Schedule (UTC)	Region	Destination dataset	Next scheduled UTC+2	Actions
	transfer_vaccine	Google Cloud Storage	every day 18:15	us	dataset	June 30, 2021 at 8:15:00 PM UTC+2	

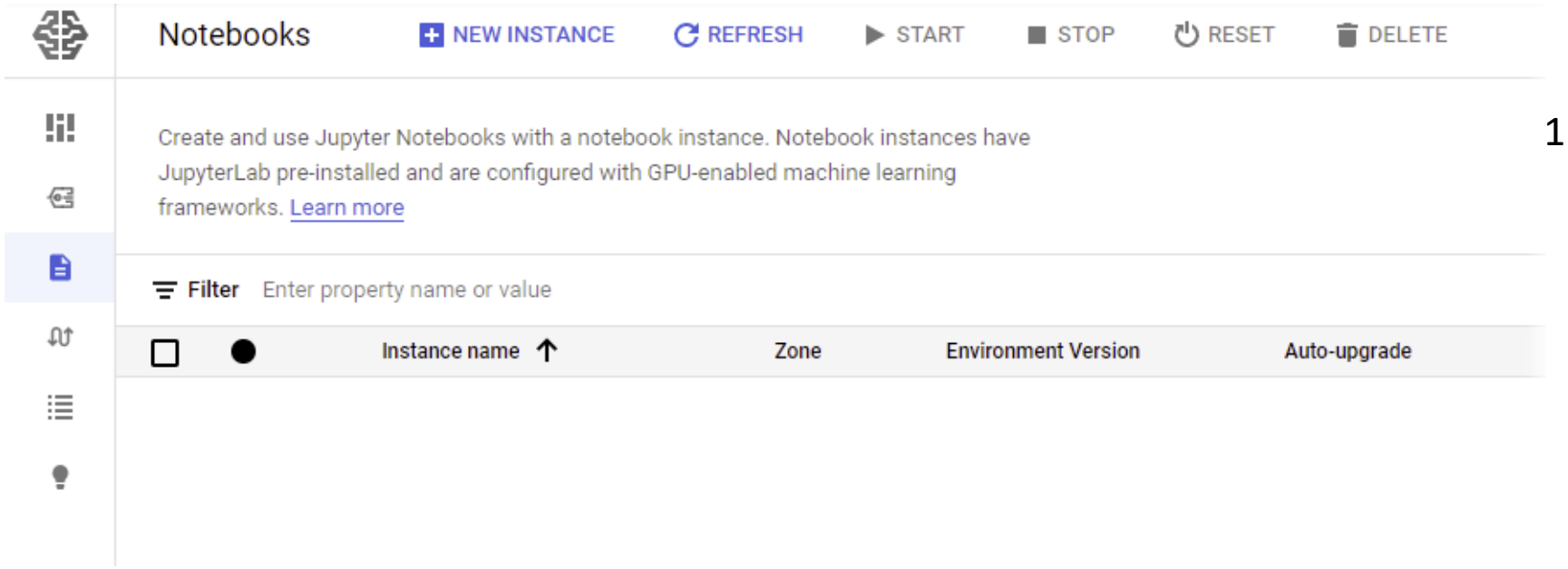
The data transfer from the file in the bucket to the table in the dataset has been created correctly and scheduled!
Repeat again the previous steps for the other data transfers!

BigQuery ← Data transfers + CREATE TRANSFER

Filter Filter transfer configs

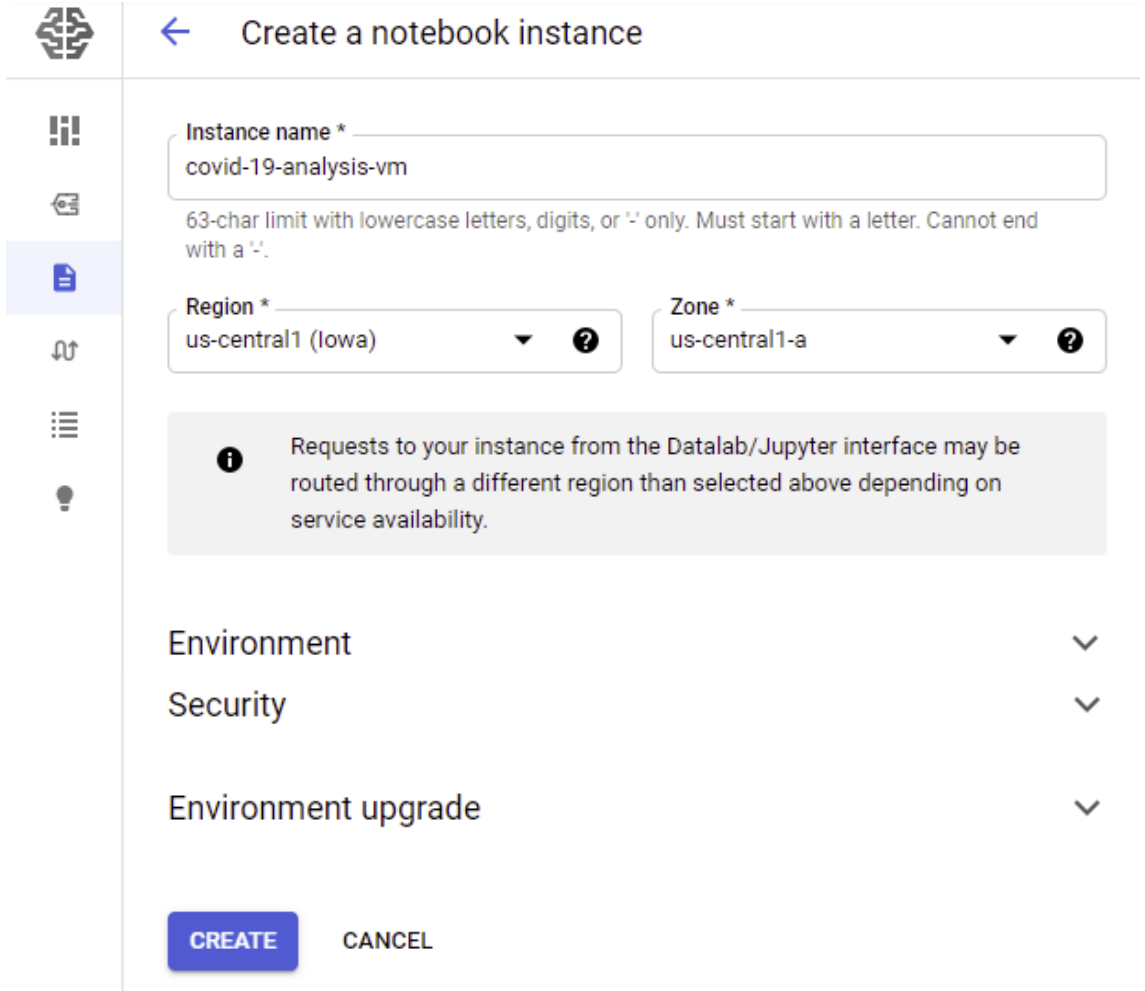
	Display name	Source	Schedule (UTC)	Region	Destination dataset	Next scheduled UTC+2	Actions
	transfer_vaccine	Google Cloud Storage	every day 18:15	us	dataset	June 30, 2021 at 8:15:00 PM UTC+2	
	transfer_province	Google Cloud Storage	every day 18:15	us	dataset	June 30, 2021 at 8:15:00 PM UTC+2	
	transfer_regioni	Google Cloud Storage	every day 18:15	us	dataset	June 30, 2021 at 8:15:00 PM UTC+2	

The workflow – Google AI Platform (Notebooks)



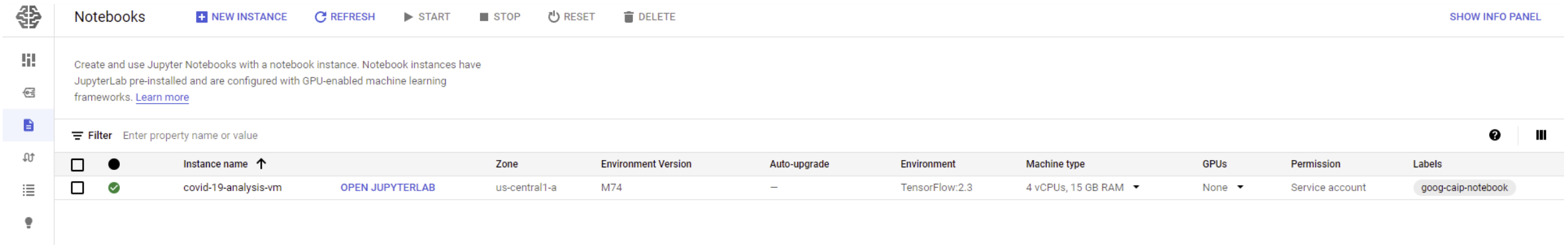
1. Create a NEW INSTANCE

The workflow – Google AI Platform (Notebooks)



1. Create a NEW INSTANCE
2. Insert Instance Name
3. Leave all parameters to Default
4. CREATE

The workflow – Google AI Platform (Notebooks)



The virtual machine has been created correctly and
we can open JupyterLab!